UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

INTRODUCCIÓN A LA MANIPULACIÓN DE DATOS CON POSTGRESQL Y R

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

A C T U A R I A

P R E S E N T A:

KARINA LIZETTE GAMBOA PUENTE

DIRECTOR DE TESIS:
DR. MIGUEL EHÉCATL MORALES TRUJILLO
2017

Datos del alumno:
Gamboa
Puente
Karina Lizette
Universidad Nacional Autónoma de México
Facultad de Ciencias
Actuaría
308052318

Datos del tutor:
Dr.
Miguel Ehécatl
Morales
Trujillo

Datos del sinodal 1:
M. en I.
Gerardo
Avilés
Rosas

Datos del sinodal 2:
L. en C. C.
Erick Orlando
Matla
Cruz

Datos del sinodal 3:
L. en C. C.
Marisol
Flores
Castro

Datos del sinodal 4:
M. en C.
Benjamin
Figueroa
Solano

Datos del trabajo escrito:
“Introducción a la manipulación de datos con PostgreSQL y R."
175 pág.
2017

A mi Padre y Dios,
quien quitó y dio,
quien ha dirigido, sostenido y prosperado mis pasos.
¡A Él sea la gloria!

Agradecimientos

Es imposible no reconocer que el logro de este trabajo y de lo que representa, está aunado a la concurrencia de distintas personas en esta etapa de mi vida.

Primeramente, agradezco a mis padres, quienes han dado su vida sin escatimar nada, han tenido enorme paciencia y confianza en mí y por quienes me he convertido en la mejor versión de mi hasta el día de hoy. A mi madre, agradezco por ser mi mayor ejemplo de esfuerzo, valor e integridad, por guiarme siempre de la mejor forma posible, con alta moral y rectitud, por ser mi amiga y confidente, por sostenerme desde mi primer aliento, a través de los malos y peores días hasta hoy.

A mi tía Paty, quien es para mí como una segunda madre y quien ha tenido un papel invaluable dentro de mi formación, con apoyo incondicional y gran amor.

A mi gran amigo José Arias, quien no fue un maestro más, sino mi mentor. Quien no sólo me instruyó en las ciencias matemáticas, sino se ocupo de mí como persona, no teniendo en poco mi esfuerzo por aprender. Mi gratitud por ofrecerme su conocimiento, por darme su confianza, tiempo y atención, por ayudarme a construir un puente sobre la enfermedad que me permitió continuar hacia mí meta de ser profesionista. Su respaldo fue decisivo en mi decisión de no abandonar mi carrera y en el desarrollo de este trabajo.

A Luis Quiroz con todo mi cariño y reconocimiento, por su acompañamiento en mis éxitos, fracasos, frustraciones y alegrías, por compartir conmigo su vida y tiempo, por crear cosas mejores en él para darme, por diseñar sus horarios en torno a mis necesidades, cuidar de mi mientras dormía y esperar pacientemente a que despertase para repetir y continuar con las lecciones matemáticas. Por ser mi profesor más constante y entrañable, que me enseñaba con amor, dedicación y paciencia aún las cosas que él mismo debía aprender por mí. Por alentarme, apoyarme, cuidar de mi y sostenerme por largos años. YTLDDMV.

A Oscar Bringas, por quien descubrí cuanto amaba aprender, quien pese a mi resistencia hizo salir muchas cosas buenas de mí, quien se convirtió en un pilar importante en mi crecimiento profesional y personal. Su acompañamiento, cariño, tolerancia, ejemplo, palabras de aliento y consuelo, fueron estímulos que me ayudaron a descubrirme como una persona capaz de hacer esto y más. Mi gratitud por quedarse, por enseñarme, por creer en mí e impulsarme repetida y pacientemente. Por compartir sin reservas las cosas maravillosas que hay en su mente y corazón.

A Alejandra Valdez, Leonor Lozada, Estephani Galindo y Monserrat Mireles, mis invaluables amigas, que fueron un sostén a lo largo de la carrera, en mis buenos y peores días, quienes durante mis obligadas "siestas" cuidaban de mí y hacían de todo para mantenerme despierta en clase, me pasaban sus apuntes, estudiaban conmigo, soportaban mis crisis y frustraciones, me regañaban y cuidaban de mi corazón. A ellas que contribuyeron como nadie más e incondicionalmente en cada uno de los aspectos de mi vida convirtiéndose en mi familia. Haber concluido este trabajo es obra de su apoyo y ánimo en cada paso del camino. Las quiero.

A cada uno de mis sinodales por su tiempo, paciencia y contribuciones.

Un agradecimiento de forma especial a mis tutores Miguel Morales y Erick Matla, por ver en mí el potencial para llevar a cabo un proyecto como este, por guiarme y corregirme en el camino, por su pronta respuesta y atención a mis avances, por su apoyo y disposición constante. Me considero afortunada por tener tutores como ellos, son un gran ejemplo para mi, gracias por su tiempo y confianza.

A cada profesor y compañero que de una u otra forma contribuyeron a mi formación cuya lista seria imposible citar.

Finalmente, gracias UNAM, por regalarme los mejores años de mi vida, a mis mejores amigos, tantos aprendizajes, el deporte que amo y sobre todo, gracias por permitirme encontrar mi pasión.

"La gota de agua rompe la piedra, no por su fuerza sino por su constancia."

Índice general

Lista de figuras **11**

1. Introducción **13**
1.1. Contexto . . . 13
1.2. Problema a resolver . . . 14
1.3. Objetivo General . . . 15
1.4. Metodología . . . 15
1.5. Especificaciones . . . 17
1.5.1. Insumo principal . . . 17
1.5.2. Prácticas . . . 17
1.6. Resultados esperados . . . 22

2. Conceptos Básicos **23**
2.1. Bases de Datos . . . 23
2.2. Herramientas de manipulación de datos . . . 26
2.2.1. Definición de software y lenguaje de programación . . . 26
2.2.2. Sistemas Manejadores de Bases de Datos . . . 27
2.2.3. Software estadístico y de manipulación de datos . . . 28
2.3. Probabilidad para explotación de datos . . . 30
2.3.1. Eventos, espacios muestrales y medida de probabilidad . . . 30
2.3.2. Teoría de conjuntos . . . 31
2.3.3. Relaciones entre eventos . . . 32
2.3.4. Probabilidad condicional e independencia . . . 32
2.3.5. Variables aleatorias . . . 33
2.3.6. Funciones de distribución y densidad . . . 34
2.3.7. Distribuciones conjuntas y marginales . . . 36
2.3.8. Distribuciones condicionales . . . 38
2.3.9. Esperanza, varianza, covarianza y coeficiente de correlación . . . 38
2.4. Estadística para explotación de datos . . . 43
2.4.1. Tipos de Variables . . . 43
2.4.2. Análisis exploratorio y otras medidas de tendencia central y dispersión . . . 45
2.4.3. Estimación puntual . . . 49
2.4.4. Pruebas de hipótesis . . . 50
2.4.5. Estadística no paramétrica . . . 52

3. Práctica R **55**
3.1. Objetivo . . . 55
3.2. Introducción . . . 55
3.3. Conexión PostgreSQL y R . . . 55
3.4. Ejercicios . . . 60

4. Práctica S **61**
4.1. Objetivo . . . 61
4.2. Introducción . . . 61
4.3. Perfiles en R . . . 63
4.4. Ejercicios . . . 71

5. Práctica T **73**
5.1. Objetivos . . . 73
5.2. Introducción . . . 73
5.3. Distribuciones condicionales y dependencia . . . 74
5.4. Ejercicios . . . 86

6. Práctica U **87**
6.1. Objetivos . . . 87
6.2. Introducción . . . 87
6.2.1. Medidas de tendencia central, dispersión y posición . . . 87
6.2.2. Diagrama de caja . . . 89
6.3. Análisis de variables cuantitativas . . . 89
6.3.1. Gráficas de estrellas . . . 98
6.4. Ejercicios . . . 103
6.5. Anexo . . . 104

7. Práctica K **105**
7.1. Objetivos . . . 105
7.2. Introducción . . . 105
7.3. Iniciando un proyecto con KNIME . . . 108
7.3.1. Nodos de acceso. . . . 109
7.3.2. Nodos de manipulación. . . . 111
7.3.3. Nodos de análisis. . . . 112
7.3.4. Nodos de visualización. . . . 116
7.3.5. Nodos de despliegue. . . . 118
7.4. Ejercicios . . . 122

8. Práctica L **123**
8.1. Objetivos . . . 123
8.2. Introducción . . . 123
8.3. Limpieza . . . 123
8.4. Ejercicios . . . 133

9. Práctica P **135**
9.1. Objetivos . . . 135
9.2. Introducción . . . 135
9.3. Python y operaciones básicas . . . 138
9.4. Asignación y tipos de datos básicos . . . 139
9.5. Listas . . . 141
9.6. Funciones y métodos . . . 142
9.7. Soluciones . . . 145

10.Práctica Q **147**
10.1. Objetivos . . . 147
10.2. Introducción . . . 147
10.3. Paquetes . . . 148
10.4. NumPy . . . 149
10.5. Matrices . . . 151
10.6. Estadística Básica . . . 152
10.7. Soluciones . . . 154

11.Resultados **157**
11.1. Verificación de las prácticas . . . 157
11.2. Socialización de las prácticas . . . 158
11.3. Validación de las prácticas . . . 159
11.4. Recolección y análisis de resultados . . . 161
11.4.1. Comentarios de los alumnos . . . 165
11.5. Interpretación de resultados . . . 166
11.6. Mejoras identificadas . . . 167

12.Conclusiones **169**

A. NFL-ONEFA **173**

Índice de figuras

2.1. Ejemplo de modelo jerárquico. 24
2.2. Ejemplo de modelo en red. 25
2.3. Ejemplo de modelo relacional. 25
2.4. Desviación estándar. 40
2.5. Interpretación de la covarianza. 41
2.6. Coeficiente de correlación. 42
2.7. Tipo de variables. 44
2.8. Gráfica circular. 46
2.9. Gráfica de barras. 46
2.10. Diagrama de caja. 47
2.11. Histograma. 47
2.12. Diagrama de dispersión. 47
2.13. Gráfica de caras (Chernoff). 48
2.14. Gráfica de estrellas. 48
2.15. Región crítica [1]. 52

7.1. Software KNIME. 106
7.2. Nodos. 106
7.3. Almacén de nodos. 107
7.4. Proyecto nuevo. 108
7.5. Flujo de trabajo. 108
7.6. Nodo 1, selección y configuración. 109
7.7. Nodo 2, configuración. 110
7.8. Nodo 2, resultado. 110
7.9. Nodo 3, configuración y resultado. 111
7.10. Unión de nodo 4. 111
7.11. Nodo 4, configuración y resultado. 112
7.12. Unión de nodo 5. 112
7.13. Nodo 5, configuración. 113
7.14. Nodo 5, resultado. 113
7.15. Unión de nodo 8. 113
7.16. Nodo 6, resultados. 114
7.17. Unión de nodo 8. 114
7.18. Nodo 7 y 8, configuración. 115
7.19. Nodo 8, resultados. 115
7.20. Nodo 8, resultados. 116
7.21. Unión de nodos 10 y 12. 116
7.22. Nodo 10, resultado. 117
7.23. Nodo 12, configuración. 117
7.24. Nodo 12, resultado. 118

7.25. Unión de nodos 14, 15, 16 y 17. . . . 118
7.26. Nodo 14, configuración. . . . 119
7.27. Nodo 14, resultado. . . . 120
7.28. Nodo 15, configuración. . . . 120
7.29. Nodos 14, 15, 16 y 17, resultado. . . . 121

8.1. Nodo 1, configuración. . . . 124
8.2. Nodo 1, resultados. . . . 124
8.3. Nodo 2, configuración. . . . 125
8.4. Nodo 2, resultado. . . . 125
8.5. Nodo 3, configuración. . . . 126
8.6. Nodo 3, resultado. . . . 126
8.7. Nodo Missing Value 127
8.8. Nodo 4, configuración. . . . 128
8.9. Nodo 4, resultados. . . . 129
8.10. Nodo 4, resultado 2. . . . 129
8.11. Nodo 5, configuración. . . . 130
8.12. Nodo 5, resultado. . . . 131
8.13. Nodo 7, configuración. . . . 132

9.1. Selección de lenguaje en compilador online JDOODLE. . . . 136
9.2. Compilador Online JDOODLE. . . . 137
9.3. Selección de lenguaje en compilador online Repl.it. . . . 137
9.4. Compilador Online Repl.it. . . . 138

11.1. Participación de los alumnos por práctica. . . . 159
11.2. Resultados de la práctica R. . . . 162
11.3. Resultados de la práctica S. . . . 162
11.4. Resultados de la práctica T. . . . 163
11.5. Resultados de la práctica U. . . . 163
11.6. Resultados de la práctica K. . . . 164
11.7. Información extra requerida. . . . 164

A.1. Diagrama ONEFA 175

1 | Introducción

1.1. Contexto

Desde tiempos antiguos el hombre ha utilizado diferentes mecanismos para almacenar datos relevantes de su acontecer diario, esto con la finalidad de hacer frente a sus necesidades de información. Hace 30 o 40 años, las formas más comunes de almacenamiento seguían siendo archiveros y sistemas manuales en los que era necesario gran cantidad de papel, espacio físico, bodegas o almacenes. En ellos se recaudaban datos sobre cosechas, nacimientos, muertes y censos en general. Acceder a estos registros era difícil y su búsqueda, tardada. Sin embargo, los mecanismos para el almacenamiento de datos han evolucionado a la par de la tecnología; del papel, a los sistemas de archivos y las bases de datos, por lo que la situación cambió con la aparición de los sistemas computarizados. Se dejó de usar papel, bodegas y personal para utilizar una computadora y un disco duro.

La era actual se ha caracterizado por un avance tecnológico y de digitalización importante, el creciente uso de las computadoras, así como las redes globales, han generado que los datos fluyan rápidamente y su acumulación propicia que las herramientas para gestionar bases de datos tengan un lugar importante y decisivo. Las bases de datos entonces, se han vuelto indispensables para el trabajo y gestión de las grandes empresas, así como para investigaciones sociales, de mercado, clínicas, entre otras cosas.

Aunque existen distintos modelos de bases de datos, las bases relacionales son las más utilizadas hoy en día en empresas y para investigación. Este tipo de bases de datos utiliza un modelo basado en tablas, lo que facilita la consulta y compresión de su contenido, también tienen la habilidad de relacionar tablas con diferentes tipos de datos gracias a llaves, esto hace que el trabajo con este modelo sea eficiente. Para trabajar con este tipo de bases existen diferentes sistemas manejadores de bases de datos (SMBD) y programas especializados para la explotación de las mismos. Cada una de estas herramientas tienen diferentes usos. Los SMBDs posibilitan el manejo de grandes volúmenes de datos en poco tiempo, se puede disponer de datos de manera simultánea para más de un usuario, permiten el trabajo con diferentes estructuras de datos organizados e incluso dan la oportunidad de modificar dichas estructuras; también les ahorran a los usuarios los detalles del almacenamiento físico de los datos.

Por otro lado, el software y los programas especializados en la explotación de datos tienen una gran capacidad para procesarlos, permiten utilizar diferentes técnicas para el análisis de éstos, hacen posible la inferencia estadística y asociación entre variables, algunos de ellos permiten crear modelos de predicción o discriminación y tienen la ventaja de incluir cálculos y fórmulas propias del usuario a través de un lenguaje de programación, además de poder utilizar funciones de diferentes paqueterías para la explotación de los mismos.

Es por ello que este proyecto se centra en la integración de software, lenguajes de progra-

mación y SMBDs con el fin de explotar de una mejor manera los datos almacenados en una base de datos.

1.2. Problema a resolver

Actualmente se almacenan grandes cantidades de datos, la búsqueda de conocimiento de la población, usuario o clientes se ha reflejado en la necesidad de almacenar toda actividad que se lleve a cabo, principalmente a través de un sistema, aplicación o página web. Considerando lo anterior, es importante reconocer que con la gran cantidad y variedad de registros que se acumulan, no basta con acceder fácilmente a ellos y tenerlos a la mano. Podría decirse que la producción y almacenamiento masivo de datos no necesariamente puede ser considerada información, por lo que es necesario no sólo incursionar en su manejo, sino tener conocimientos y herramientas que permitan obtener de los datos, información valiosa y provechosa, con análisis lógico, modelos matemáticos y análisis de las variables que se encuentren.

De esta manera se podrían obtener datos para administrar de una mejor manera, aumentar la productividad, reducir costos y para tomar decisiones competentes según el área y cuestión. Todo esto podría confirmar que ya no es suficiente el manejo de las bases de datos, hoy en día es necesario que exista una interacción con otras herramientas que permitan diferentes formas y técnicas para manipular y explotar datos.

En este momento existen diversas herramientas de gran utilidad que pueden optimizar los tiempos de trabajo, la revolución digital también ofrece la creación y desarrollo de diversos software que en conjunto aportan al profesionista que los conoce y domina, una capacidad sobresaliente para hacer su trabajo. Siguiendo en esta línea, se debe mencionar que el perfil de los actuarios y de estudiantes de carreras afines de la Facultad de Ciencias, formados con bases matemáticas de manera lógico-estructurada, permite que se desenvuelvan en diversas áreas, entre ellas estadística y probabilidad. Esto, aunado a conocimientos de programación, les otorga un perfil adecuado para la administración y explotación de grandes cantidades de datos, aportando análisis cuantitativo, elaborando estrategias para la interpretación de los resultados en diversas áreas o investigaciones.

En contraste con lo anterior, en mi estancia como alumna dentro de la Facultad de Ciencias resultó complicado conocer y mucho más aprender todas las herramientas computacionales. Existen cursos de programación que abren el panorama, donde se puede conocer C++, Java o Visual FoxPro, luego, con otras asignaturas, el alumno se adapta a los programas que el profesor indique que debe manejar. En materias de estadística, se puede interactuar con R, Excel o SPSS. En alguna otra materia como Análisis Numérico, se puede conocer Matlab o Python. A pesar de eso, ya sea por tiempo, mayor dedicación a la parte teórica o por el límite de materias optativas que un alumno tiene derecho a inscribir, un alumno promedio de la carrera de Actuaría, apenas escuchará de algunos de los muchos programas que puede utilizar y en ocasiones no alcanzará a cubrir los conocimientos necesarios para usar esos paquetes con destreza.

1.3. Objetivo General

El objetivo general de la materia optativa de Bases de Datos[1] es conocer y dominar los principales conceptos subyacentes al campo del diseño, construcción y explotación eficiente de bases de datos relacionales.

De la misma forma, este trabajo tiene el objetivo de complementar específicamente la parte de explotación eficiente de bases de datos, generando recursos didácticos que permitan introducir a los alumnos que tienen conocimientos previos de estadística y probabilidad, en algunas de las técnicas y herramientas que pueden usarse para el modelado, consulta y explotación de bases de datos. Lo anterior, les permitirá transformar el contenido de éstas, en conocimientos útiles y de interés para las diversas áreas actuariales en las que puedan desarrollarse.

Cabe señalar, que a la par con el objetivo anterior, se busca generar interés en cualquier persona que encuentre útiles estas herramientas en su área de trabajo o especialidad, exhortando a que se profundice en ellas. Adicionalmente se utiliza el lenguaje de SQL como recurso para el desarrollo de este material (reforzando el quinto tema del temario de Bases de Datos ("Lenguaje de consulta estructurado SQL").

Finalmente, si bien no está explícito en el temario, es relevante y necesario que el contenido de la asignatura permita al alumno integrar herramientas estadísticas y de programación con las bases de datos, ya que diversas empresas donde se analizan e interpretan datos, como son las organizaciones del sector financiero y asegurador, son lugares de trabajo potenciales para los egresados de la carrera de Actuaría.

1.4. Metodología

Dada la gran cantidad de datos que actualmente se almacenan, las actividades relacionadas a su explotación aumentan en demanda. Por ello, se requiere aprovechar las herramientas actuales que permitan dichas actividades.

En muchas ocasiones al no tener consciencia de la existencia o el uso de dichas herramientas, un alumno no puede explotar sus conocimientos previos así como los datos que tiene a su disposición de diferentes maneras. Es por ello que a través de prácticas se planea introducir al lector en el uso de estas herramientas, así como en el beneficio de su combinación y empleo de diferentes técnicas para la manipulación y explotación de datos.

La metodología utilizada para la selección de las herramientas y el diseño de las prácticas se presenta en los siguientes puntos:

1. **Identificación de las herramientas de software y SMBDs candidatos a utilizar.**

   Inicialmente se hablará de las herramientas necesarias para la manipulación y explotación de bases de datos, describiendo las características y ventajas de cada tipo de herramienta en su sección correspondiente.

   Dentro del conjunto de estas herramientas tenemos diferentes Sistemas Manejadores o Gestores de Bases de Datos, lenguajes de programación y software estadísticos y de mi-

[1] Temario de la materia optativa de Bases de Datos consultado en Agosto 2017, http://www.fciencias.unam.mx/licenciatura/asignaturas/2017/143

nería de datos, de los cuales se eligieron los que satisfagan mayormente las necesidades de este trabajo y puedan funcionar de manera complementaria.

2. **Selección del software y SMBD a utilizar.**

   Por otra parte, la elección de programas y herramientas que se utilizarán para el desarrollo de este trabajo, se basó en procurar una relación con diferentes carreras impartidas en la Facultad de Ciencias. Se pretende que estudiantes de Ciencias de la Computación, Matemáticas o Actuaría puedan sentirse cómodos e identificados con el uso de los instrumentos seleccionados y sean fácilmente reconocidos.

   Además, se busca que la forma de abordar los temas y la interacción entre las herramientas pueda ser útil para ser tomada en cuenta como una parte complementaria a los cursos de Bases de Datos que se imparten en la Facultad de Ciencias.

   De esta manera, la selección del SMBD se llevó a cabo con base en herramientas que se trabajan en cursos de Bases de Datos de la Facultad; aprovechando la plataforma DreamSpark y el convenio de Microsoft con la UNAM, **SQL Server** resultó uno de los SMBD elegidos; por otro lado, la necesidad de tener una herramienta de licencia pública nos llevó a elegir **PostgreSQL** como SMBD alterno.

   PostgreSQL además de ser de licencia pública, es un programa eficiente y fácil de administrar. De igual manera utiliza SQL para realizar consultas. Por otra parte, puede operar sobre distintas plataformas, incluyendo Linux y Windows, lo cual lo convierte en una herramienta versátil para los estudiantes, además de ser conocida y confiable.

   En segunda instancia, se eligieron los lenguajes de programación **R** y **Python** así como el software correspondiente para cada uno de ellos.

   La razón para elegir R se sustentó en que, es un lenguaje de programación conocido dentro del ámbito estadístico de la Facultad de Ciencias. Adicionalmente, al ser un lenguaje estadístico tiene todo lo necesario para el análisis de datos, variedad de gráficos que pueden utilizarse con versatilidad, así como pruebas estadísticas ya implementadas. Otro rasgo importante es que permite la creación de nuevas funciones y métodos que el usuario requiera crear. Finalmente, R tiene incluidas funciones que se suelen utilizar en otros programas y tiene la ventaja de leer y exportar archivos en diferentes formatos.

   Usualmente, se utiliza R Studio como una plataforma para el lenguaje R. R Studio es un software libre y multiplataforma, lo que permite que cualquier persona pueda trabajar con él en diferentes sistemas operativos.

   Simultáneamente Python es un lenguaje de alto nivel, y así como R, cuenta con distintos paquetes para análisis y visualización de datos. Puede ser usado en diferentes sistemas operativos y actualmente está en incremento su uso para minería de datos

   Finalmente, se mencionará KNIME, es una plataforma para minería de datos que permite al usuario crear de forma visual flujos de datos y ejecutar pasos de análisis que pueden incorporar código de R o Python, entre otros.

3. **Diseño y creación de las prácticas para el uso del software y SMBD elegidos.**

   El diseño de las prácticas buscará que el alumno se familiarice con cada uno de los programas, se explicarán algunos comandos y sintaxis de los mismos, mientras se generan recursos didácticos que permitan al alumno aplicar métodos estadísticos y probabilísticos sobre bases de datos estructuradas.

   Se utilizarán bases de datos cuyas características permitan hacer análisis estadístico y cuyos resultados permitan al alumno comprender de mejor manera lo expuesto.

   En cada una de las prácticas se realizará un análisis distinto con el contenido de la base de datos, haciendo uso de las diferentes herramientas propuestas. Se hará un manual o script con los comandos comentados que se utilizarán para dicho ejercicio. Después de ello el alumno podrá imitar y resolver el problema en contextos similares. El objetivo será que el alumno pueda ejercitarse en el uso de las herramientas y al manipular los datos pueda dar interpretaciones adecuadas y útiles.

4. **Verificación y validación de prácticas.**

   Las prácticas creadas serán verificadas por dos profesores de la Facultad de Ciencias expertos en Bases de Datos y se validará en un grupo piloto. El objetivo de la validación será demostrar la utilidad del contenido mostrado en las prácticas.

5. **Socialización de las prácticas.**

   Las prácticas creadas se distribuirán de manera electrónica a los alumnos. También estarán disponibles en la página del curso de Bases de Datos de manera que, de ser requerido por otros profesores interesados, puedan utilizarlas.

1.5. Especificaciones

Para facilitar la utilización de las prácticas creadas, a continuación se presentan sus objetivos particulares, el insumo principal para utilizarlas y los recursos necesarios.

1.5.1. Insumo principal

Para las prácticas se decidió utilizar una base de datos con la que los alumnos hubieran trabajado previamente. La base de datos a utilizar se llama NFL-ONEFA, la cual contiene información relacionada a una liga de futbol americano.

Para su uso dentro de las prácticas, la base fue poblada con información de partidos de 10 temporadas. Las tablas que son de interés para este trabajo son las especializaciones de los jugadores, las tablas que contienen información sobre los partidos y el clima de las ciudades a los que cada equipo esta asociado. En el Anexo A se presentan los supuestos y el diagrama de la base de datos.

1.5.2. Prácticas

Se crearán 3 tipos de prácticas, con distintos esquemas y objetivos:

- **Prácticas con lenguaje R.** Las prácticas R, S, T, y U tienen el objetivo de integrar las habilidades adquiridas en distintas asignaturas de la carrera de Actuaría, específicamente

Probabilidad y Estadística, con los conocimientos que se vayan adquiriendo del curso de Bases de Datos. Se presentaran aplicaciones para explotar los datos a los que se tengan acceso, para este fin se utilizará el lenguaje R y el software R Studio.

La estructura de dichas prácticas contiene un objetivo particular, una introducción al tema que se desarrollará con conceptos básicos para la realización de la misma, una parte práctica donde se hace uso del lenguaje R y finalmente, ejercicios similares a los ejemplos expuestos para reforzar lo aprendido.

- Práctica R
  - Objetivos:
    a) Describir las ventajas de integrar PostgreSQL y R.
    b) Explicar los pasos para realizar la conexión entre PostgreSQL y R.
    c) Realizar consultas dentro de R para manipular datos de una instancia de PostgreSQL
  - Precondiciones:
    a) Conocimientos del lenguaje R (declaración de variables, carga de de paqueterias, implementación de gráficas, ciclos y manipulación de tablas).
    b) Dominio de las instrucciones JOIN, UNION, ORDER BY y GROUP BY de SQL.
    c) Noción del concepto de subconsultas
  - Insumos:
    a) R Studio, PostgreSQL y base de datos NFL-ONEFA.
  - Postcondiciones:
    a) El alumno será capaz de ejecutar la conexión entre PostgreSQL y R.
    b) El alumno será capaz de extraer conjuntos de datos a través de R para su análisis.
- Práctica S
  - Objetivos:
    a) Introducir al alumno en uno de los conceptos base del análisis de correspondencias: perfiles.
    b) Implementar método gráfico de perfiles en lenguaje R.
    c) Interpretar los resultados gráficos.
  - Precondiciones:
    a) Conocimientos del lenguaje R (declaración de variables, carga de paqueterías, implementación de gráficas y manipulación de tablas).
  - Insumos:
    a) R Studio.
    b) Conjunto de datos “equipos-partidos-climas.csv” obtenido en la práctica R.

- Postcondiciones:
  a) El alumno será capaz de realizar un análisis de perfiles e intepretación de las gráficas correspondientes a la técnica.
  b) El alumno obtendrá un criterio y visión analítico sobre el comportamiento de los datos contenidos en NFL-ONEFA.

- Práctica T
  - Objetivos:
    a) Comprender la importancia del concepto de dependencia e independencia entre variables aleatorias.
    b) Aprender la importancia del uso de gráficas con distribuciones condicionales para la toma de decisiones.
    c) Usar e interpretar la prueba Ji-cuadrada para independencia de variables.
    d) Emplear de código en lenguaje R para creación de tablas, distribuciones condicionales y prueba de hipótesis.
  - Precondiciones:
    a) Conocimientos del lenguaje R (declaración de variables, carga de de paqueterias, implementación de gráficas y manipulación de tablas).
    b) Conocimientos básicos de probabilidad y estadística (distribuciones conjuntas, condicionales y pruebas de hipótesis).
  - Insumos:
    a) R Studio.
    b) Conjunto de datos "equipos-partidos-climas.csv" obtenido en la práctica R.
  - Postcondiciones:
    a) El alumno será capaz de comprender la importancia de los conceptos de dependencia e independencia.
    b) El alumno será capaz de aplicar e interpretar la prueba Ji-cuadrada sobre tablas de contingencia.
    c) El alumno será capaz de identificar gráficamente relaciones importantes entre variables.

- Práctica U
  - Objetivos:
    a) Introducir al alumno en el análisis exploratorio y descriptivo de variables cuantitativas.
    b) Implementación de métodos gráficos en lenguaje R para el análisis de variables cuantitativas.
    c) Interpretación de gráficos y coeficiente de correlación de Pearson.
  - Precondiciones:
    a) Conocimientos del lenguaje R (declaración de variables, carga de de paqueterias, implementación de gráficas, ciclos y manipulación de tablas).
    b) Dominio de las instrucciones JOIN, UNION, ORDER BY, GROUP BY, LEFT OUTER JOIN, COUNT, SUM y NATURAL JOIN de SQL.
    c) Noción del concepto de subconsultas.
    d) Conocimientos básicos de probabilidad y estadística (medidas de tendencia central, correlación de Pearson e independencia).
  - Insumos:

a) R Studio, PostgreSQL y base de datos NFL-ONEFA.
b) Consulta U (contenida en el anexo de la Práctica U).

- Postcondiciones:
  a) El alumno sabrá realizar análisis exploratorio de variables cuantitativas.
  b) El alumno será capaz de representar gráficamente resúmenes de información de los datos cuantitativos y la asociación entre variables.
  c) El alumno aprenderá nuevas formas de representación visual de tres o más variables cuantitativas.

- **Prácticas con KNIME** Las prácticas K y L tienen el objetivo de presentar al alumno el software KNIME, mostrando una alternativa de herramienta estadística para la manipulación y visualización de datos. Las prácticas se enfocan en la interacción del alumno con el programa y la presentación de los elementos de KNIME. Se espera generar interés para el aprendizaje de un software nuevo. Ambas prácticas comparten la estructura de las prácticas en R.

  - Práctica K
    - Objetivos:
      a) Que el alumno conozca y se familiarice con el software KNIME.
      b) Describir de forma clara, sintetizada y visual elementos básicos del software KNIME.
      c) Proporcionar ejemplos del uso de KNIME con bases de datos.
    - Precondiciones:
      a) Conocimientos de estadística (regresión lineal, correlación de Pearson y medidas de tendencia central).
      b) Dominio de las instrucciones JOIN, UNION, ORDER BY, GROUP BY, LEFT OUTER JOIN, COUNT, SUM y NATURAL JOIN de SQL.
      c) Noción del concepto de subconsultas.
    - Insumos:
      a) PostgreSQL, KNIME, consulta U (anexo de la Práctica U) y base NFL-ONEFA.
    - Postcondiciones:
      a) El alumno será capaz de establecer una conexión entre KNIME Y PostgreSQl.
      b) El alumno conocerá la manera de realizar consultas y extraer conjuntos de datos de una base de datos en PostreSQL para ser manipulada por KNIME.
      c) El alumno identificará la clasificación de los nodos (herramientas) de KNIME.
      d) El alumno aprenderá la configuración de los distintos tipos de nodos y el procedimiento para la exportación de archivos.
  - Práctica L
    - Objetivos:
      a) Identificar posibles herramientas de KNIME para limpieza de bases de datos.
      b) Ejemplificar el uso de los nodos básicos para análisis de valores perdidos.
    - Precondiciones:

a) Dominio de las instrucciones JOIN, UNION, ORDER BY, GROUP BY, LEFT OUTER JOIN, COUNT, SUM y NATURAL JOIN de SQL.
b) Noción del concepto de subconsultas.
c) Conocimiento de los diferentes tipos de nodos de KNIME.
d) Conocimiento de la conexión PostgreSQL y KNIME para extracción de conjuntos de datos.

- Insumos:
  a) Consulta U (contenida en el anexo de la Práctica U).
  b) Base de datos con datos faltantes (opcional).
- Postcondiciones:
  a) El alumno conocerá una de las herramientas de KNIME para la imputación de datos y sus diferentes configuraciones.

- **Prácticas con lenguaje Python.** Las prácticas P y Q tienen el objetivo de enseñar desde cero la estructura y sintaxis del lenguaje Python, sentando las bases para iniciar al alumno en la manipulación de datos con este lenguaje. Estas prácticas buscan ser el parte aguas para el aprendizaje de un nuevo lenguaje de programación que tiene alta relevancia en la ciencia de datos.

  Dado que el enfoque es distinto al resto de las prácticas, la estructura de ambas se basa en un objetivo particular, una introducción (no sobre técnicas estadísticas sino sobre el conocimiento del lenguaje de programación) y una parte práctica que a su vez presenta ejercicios para repasar lo explicado conforme se va avanzando. Finalmente, las prácticas P y Q contienen una sección de respuestas a los ejercicios, ya que se asume que a diferencia de R, Python es un lenguaje nuevo.

  - Práctica P
    - Objetivos:
      a) Aprender la sintaxis del lenguaje Python.
      b) Conocer las estructuras básicas para la manipulación de datos.
      c) Que el alumno practique sus conocimientos del lenguaje Python, conforme van obteniendo experiencia en el lenguaje.
    - Precondiciones:
      a) Tener nociones básicas de algún lenguaje de programación.
    - Insumos:
      a) Terminal en línea (A elegir por el alumno).
    - Postcondiciones:
      a) El alumno conocerá la sintaxis de Python.
      b) El alumno conocerá las operaciones básicas en Python, asignación y tipos de datos, estructuras de las listas, así como funciones y métodos.
  - Práctica Q
    - Objetivos:
      a) Que el alumno progrese en la comprensión de la sintaxis del lenguaje Python.
      b) Que el alumno refuerce los conocimientos adquiridos en la práctica P.
      c) Adquirir conocimientos sobre la importación y uso de paquetes dentro de Python.
    - Precondiciones:

a) Práctica P

- Insumos:
  a) Terminal en línea (A elegir por el alumno).
- Postcondiciones:
  a) El alumno conocerá la manera de importar paquetes en Python.
  b) El alumno aprenderá las funciones básicas del paquete NumPy.
  c) El alumno conocerá la estructura de las matrices dentro de Python.
  d) El alumno aprenderá funciones de estadística básica.

Los insumos y PDF's de las prácticas se encuentran disponibles para su uso en el siguiente URL: *132.247.127.131:8080/BD/*

1.6. Resultados esperados

Al culminar la aplicación en el grupo piloto se espera:

- Poder obtener conclusiones y documentación para la mejora de las prácticas como apoyo didáctico.
- Potencializar las capacidades de los estudiantes en el manejo de bases de datos.
- Ampliar el panorama de los estudiantes respecto a las herramientas para explotación y manipulación de datos.
- Que los alumnos puedan integrar las distintas herramientas para optimizar sus recursos.
- Garantizar que al concluir con las prácticas el alumno haya adquirido las herramientas necesarias para implementar las técnicas vistas con diferentes bases de datos.
- Finalmente, que las prácticas creadas tengan el potencial para convertirse en una herramienta complementaria para cursos de Bases de Datos.

2 | Conceptos Básicos

2.1. Bases de Datos

Los datos son una representación abstracta de una variable. Elmasri y Navathe los explica como hechos conocidos que pueden grabarse y tienen un significado implícito [4]. Entonces, una base de datos se define como un conjunto de datos que están relacionados entre sí. Esta descripción es sencilla y concreta, sin embargo, para que un "conjunto de datos" sea una base de datos, estos deben de cumplir con ciertas características [4]:

- El conjunto de datos no debe de ser aleatorio (es decir, los datos deben ser homogéneos).
- Los datos recopilados deben de tener un significado o finalidad.
- Los datos se visualizan como una tabla.
- Los datos están organizados.

El inicio de las bases de datos, como las conocemos ahora, tiene origen en tiempos muy antiguos, desde antes que la idea estuviera relacionada con el uso de las computadoras. Las bases de datos existen desde que hubo algún tipo de registro en piedra u otro material para guardar o seguir el rastro de ciertos datos, hasta la existencia de las bibliotecas. Conforme aumentó la cantidad de datos que se debían almacenar, empezó a ser necesaria la ayuda de máquinas. Herman Hollerith fue el creador de la máquina automática de tarjetas perforadas en 1884, posterior a eso se inventó una máquina censadora y en los años 50 se crearon las cintas magnéticas. Años después las computadoras empezaron a usarse y gracias a ellas, actualmente podemos concebir una base de datos como un sistema de archivos electrónicos, donde se almacenan datos de manera ordenada con un propósito en particular.

Dicho esto, ¿por qué actualmente son tan relevantes las bases de datos? A lo largo de la historia, el avance tecnológico juega un papel crucial en el desarrollo de la sociedad; la era digital, las redes globales e Internet, abren actualmente un extenso panorama para la producción, diversidad y almacenamiento de datos. Estos, se generan a un velocidad impresionante y en cantidades masivas por lo que el conocimiento sobre su apropiado manejo es cada vez más requerido.

Hoy en día, casi todo puede ser reducido a símbolos dentro de una base de datos (direcciones, registros de producción, opiniones, enfermedades, referencias de clientes, etc.); sin embargo, estos datos carecen de sentido sin un contexto donde se conviertan en elementos básicos para la creación de información. Es la búsqueda de la misma, lo que hace más relevante el uso de las bases de datos, sin ellas, muchos datos podrían pasar inadvertidos, pero gracias a su almacenamiento, pueden convertirse en información de gran utilidad. Entonces, el objetivo ya no solo es recaudar datos, sino convertirlos en algo de valor para una empresa, negocio o investigación.

En consecuencia, las bases de datos se usan cotidianamente en empresas, negocios, asociaciones de salud, de educación o cultura, hospitales y gobierno, para inventariar, ordenar, analizar o administrar datos, para hacer investigaciones en comunicación, sociales, clínicas o de mercado. Todo ello con la finalidad de consultar datos, poder comparar información, optimizar procesos o buscar patrones para sustentar la toma de decisiones.

Ahora bien, hay una gran variedad de modelos de datos. Existen las bases de datos *jerárquicas* que son aquellas en donde los datos se almacenan en una estructura de árbol invertido, donde cada uno de sus nodos depende de un único nodo padre y todos a su vez dependen de un nodo llamado raíz (Figura 2.1[1])

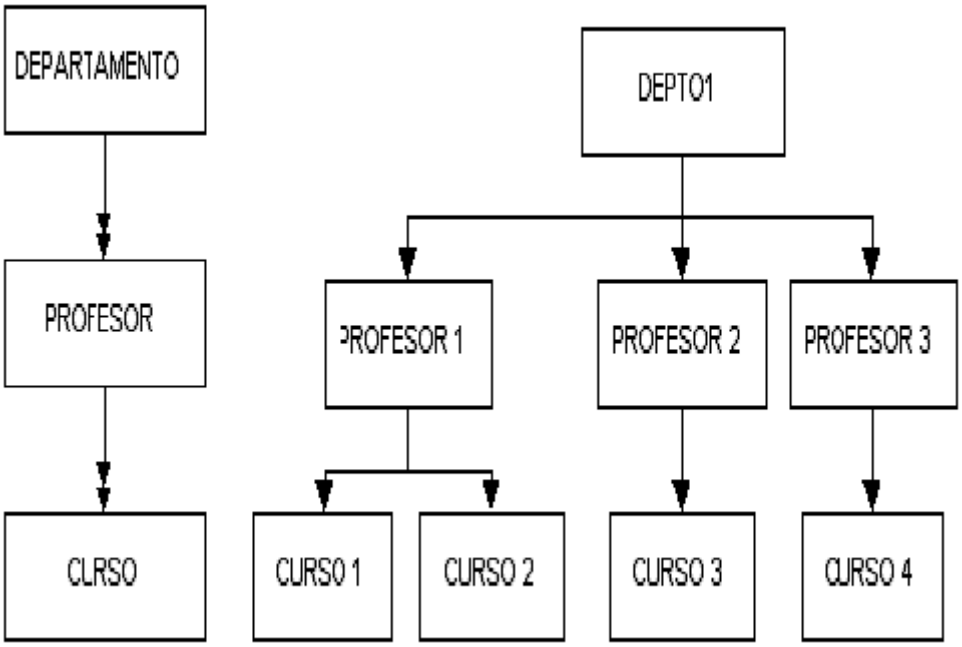


Figura 2.1: Ejemplo de modelo jerárquico.

[1]https://goo.gl/pEkDXD

Las bases de datos *en red* retoman la estructura del modelo jerárquico, pero agregan conexiones entre nodos hijos y padres para que de esta manera, se pueda acceder a un nodo en particular, no necesariamente de manera descendente sino por vías alternas (Figura 2.2[2]).

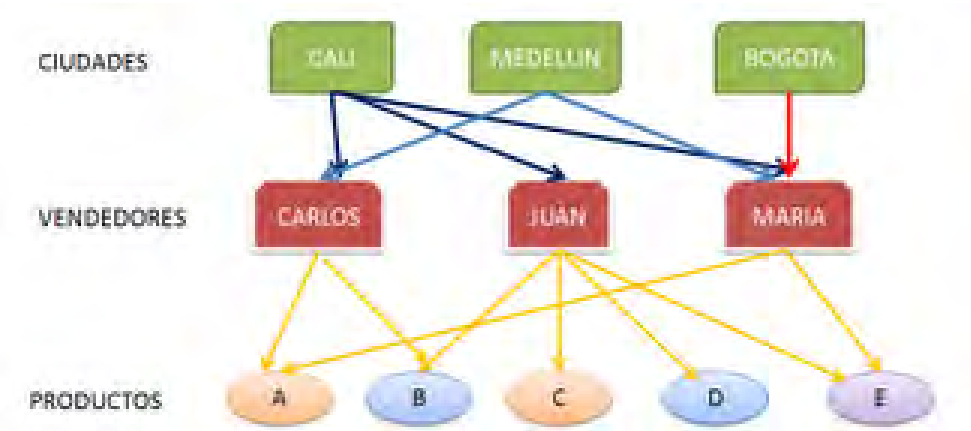


Figura 2.2: Ejemplo de modelo en red.

Las bases de datos *relacionales* tienen ventaja sobre las dos anteriores al tener un esquema basado en tablas bidimensionales y llaves que permiten la relación entre ellas. Además de tener una base matemática que incluye operadores de teoría de conjuntos, su fácil comprensión y facilidad para consulta de datos hace de este modelo el más usado hoy en día (Figura 2.3[3]).

Figura 2.3: Ejemplo de modelo relacional.

El modelo relacional de bases de datos fue una idea expuesta en 1970 por Edgar Frank Codd quien publicó una serie de reglas en su artículo “A Relational Model of data for Large Shared Banks". La característica del modelo relacional es, como su nombre lo dice, las relaciones, que son consideradas tan importantes como los datos mismos. Silberschatz describe este modelo como uno que utiliza un grupo de tablas para representar datos y las relaciones entre ellos. Cada tabla tiene varias columnas y cada una de éstas un nombre único. Este modelo es un ejemplo de bases de datos que tienen una estructura y un formato fijo, es decir, cada tabla tiene un tipo de dato en particular y cada tupla tiene un número de atributos asentado [11].

[2]https://goo.gl/yG52mr
[3]https://goo.gl/PBKJ9J

Como se ha dicho, este modelo, además de las designaciones comunes de tabla, tupla y columna, se hallan conceptos peculiares para referirse a cada una de ellas. A una tabla se le conoce como *relación*, ya que los datos dentro de ella tienen un nexo. A cada fila dentro de una tabla se le denomina como *tupla*, esta es vista como una secuencia de valores agrupados que por su naturaleza deben ir juntos[4], finalmente a las columnas de una tabla se les nombra *atributos*, que son los elementos que componen a una tupla.

Hasta ahora se ha hablado ampliamente de dos elementos clave. El primero son bases de datos, es decir, nuestra *información fuente* y el segundo son los estudiantes de la Facultad de Ciencias, que consideramos el *recurso humano*. En sistemas de información de la era actual, faltaría mencionar un último elemento, según Cohen [2] este es, *equipo computacional.*

2.2. Herramientas de manipulación de datos

En esta sección se podrán apreciar descripciones y definiciones más puntuales de algunas herramientas que pueden utilizarse para la manipulación de datos. Se mencionará las tareas especificas para las que se emplean así como ejemplos de cada herramienta.

2.2.1. Definición de software y lenguaje de programación

En los últimos años han nacido lenguajes de programación y variedad de software cuyo objetivo es brindar herramientas que permitan solucionar problemas específicos. Por esta razón es necesario explicar los siguientes conceptos.

- **Lenguajes de programación.** Un lenguaje de programación es un sistema destinado a la comunicación entre el hombre y la computadora, a través de símbolos y reglas que se convierten en instrumentos con los cuales se puede crear software.

  Dependiendo de las necesidades y recursos, los lenguajes de programación poseen diferentes características[5]:

  1. Utilidad: fácil de aprender, fácil de usar por un programador experimentado.
  2. Rendimiento: velocidad de ejecución de los programas, velocidad de ejecución del compilador.
  3. Portabilidad y flexibilidad a posibilidad de desarrollar el lenguaje y su implementación, existencia de bibliotecas de funciones, clases, etc.
  4. Continuidad: continuidad del fabricante, continuidad del lenguaje, continuidad de implementación, existencia de una norma internacional para definir el lenguaje, conformidad de implementación con respecto a la norma, existencia de varios fabricantes para un mismo lenguaje.

  Dicho de otra forma, un lenguaje de programación es un conjunto de palabras y ordenes estructuradas que nos permiten dar instrucciones a una máquina para que cree programas y software a nuestra conveniencia. Es la manera en que pedimos que se lleven a cabo tareas particulares con las condiciones que especifiquemos.

[4] http://progra.usm.cl/apunte/materia/tuplas.html
[5] https://goo.gl/jZirJS

- **Software**. Un software es un equipamiento o sistema lógico que posee un dispositivo tecnológico. El mismo está compuesto por programas capaces de realizar tareas específicas. Según su funcionalidad pueden ser clasificados en tres tipos[6]:

    1. *Software de sistema:* este grupo clasifica a los programas que dan al usuario la capacidad de relacionarse con el sistema, para entonces ejercer control sobre el hardware. El software de sistema también se ofrece como soporte para otros programas.

    2. *Software de programación:* programas directamente diseñados como herramientas que le permiten a un programador el desarrollo de programas informáticos. Influyen en su utilización diferentes técnicas utilizadas y lenguaje de programación específico.

    3. *Software de aplicación:* son aquellos programas diseñados para la realización de una o más tareas específicas a la vez, pudiendo ser automáticos o asistidos.

Existe una gran variedad de software especializado en diferentes áreas y con características diferentes. En este trabajo, hablaremos de software acorde al tema de bases de datos así como para el análisis y explotación de los datos dentro de ellas.

2.2.2. Sistemas Manejadores de Bases de Datos

En primera instancia se debe mencionar que los Sistemas Manejadores de Bases de Datos (SMBD) nacen de la practicidad y carácter del modelo relacional de bases de datos (aunque existen también para manejar otro tipo de modelos). Un SMBD según Elmasri es "una colección de programas que permite a los usuarios crear y mantener una base de datos. Además, facilita los procesos de definición, construcción, manipulación y compartición de bases de datos entre varios usuarios y aplicaciones" [4].

Los SMBD son herramientas que administran grandes cantidades de datos y permiten el acceso a estos conforme a peticiones de usuarios, dan facilidades para que, en caso de ser requerido, más de uno tenga acceso a los datos al mismo tiempo. Además, controlan las actividades que se permiten a cada usuario, autorizando o denegando la realización de tareas específicas, lo cual ayuda a garantizar que las modificaciones de los datos contenidos, de la estructura de la base y cualquier otro aspecto no será realizado por algún usuario que no esté capacitado para ejecutar dicha modificación. De esta manera se minimiza el riesgo de pérdida de datos o el mal manejo de la base de datos. Estos programas también permiten limitar la visibilidad de datos, logrando con esto que, a pesar de estar almacenados en la base a la que un usuario tiene acceso, determinados datos sigan siendo confidenciales para él.

Una gran ventaja de estos programas es que permiten la organización y reducción del espacio de almacenamiento, así como la definición de restricciones de integridad. Además, dan control sobre la coherencia de los mismos y valida las operaciones entre ellos, de manera que se asegura la eficiencia y estructura de la base y validez de los datos.

Más aún, los SMBD garantizan seguridad, ya que estos programas permiten realizar respaldos para que, en caso de ser necesario, de una manera simple se puedan restaurar y recuperar bases de datos completas.

[6] http://concepto.de/software/

2.2.3. Software estadístico y de manipulación de datos

El software estadístico se compone de programas que están especialmente diseñados para satisfacer necesidades de análisis de datos, tanto cualitativos como cuantitativos. Si bien, hay programas que no necesariamente fueron creados con este propósito, pueden tener funciones y realizar cálculos que puedan ser utilizados en este sentido.

Al ser indispensables las bases de datos en los diferentes ámbitos, el uso de estos programas también se vuelve esencial para llegar a conclusiones en investigaciones de todo tipo, por ejemplo en ciencias sociales, investigaciones de mercado, medicina, psicología, analistas de crimen o fraudes, estudios de opinión entre otros.

La ventaja de este tipo de programas es que al ser de uso específico, contienen grandes cantidades de técnicas, gráficos y funciones que permiten la realización de diferentes tipos de análisis. En algunos de estos programas se pueden recodificar variables según las necesidades del usuario, también incluyen funciones para hacer análisis multivariado, análisis exploratorio, estadística no paramétrica, así como modelos de diferentes tipos, como regresión lineal o logística. Más aún, pueden contener funciones para análisis de varianza e intervalos de confianza en estimaciones.

Otro rasgo del software mencionado es que, en su mayoría, tienen incluidas herramientas de visualización de datos. Se sabe que ya no puede limitarse el análisis de datos solo a más datos, por lo que las representaciones visuales hacen que el análisis de datos, interpretación y comparación sea más sencilla. Además, la gran variedad de gráficos como tablas, gráficos de barra, líneas, circulares, de dispersión, burbujas, o diferentes figuras así como los tridimensionales, pueden ser utilizados con versatilidad de manera que la información sea más clara y comprensible.

Finalmente se debe mencionar que algunos de estos programas tienen su propio lenguaje de programación, si se conocen estos lenguajes se tiene la capacidad de implementar funciones que no están hechas o simplemente que son requeridas por el usuario para necesidades específicas, creando y utilizando diferentes procedimientos estadísticos simultáneamente.

En el siguiente esquema se muestran ejemplos de las herramientas mencionadas:

- Heramientas
  - SMBDs
    - -Relacionales
      - MySQL
      - ORACLE
      - SQL Server
      - Microsoft Access
      - Visual FoxPro
      - PostgreSQL
      - Apache Derby
      - SQLite
      - DB2
      - MariaDB
    - - No relacionales
      - MongoBD
      - Cassandra
      - CouchDB
  - Lenguajes de Programación
    - Javascript
    - Java
    - PHP
    - Python
    - C++
    - Ruby
    - C
    - C#
    - R
    - Scala
    - Go
    - Haskell
    - Matlab
    - Visual Basic
  - Software estadístico y de minería de datos
    - Spps
    - R
    - SAS
    - SPSS
    - Stata
    - Scala
    - Minitab
    - Matlab
    - KNIME
    - EViews

2.3. Probabilidad para explotación de datos

Generalmente se usan expresiones del tipo “Probablemente el campeón del torneo de este año sea...", “Hay un 95 % de probabilidad de que llueva” de manera intuitiva, y es que a diario, con juegos de casino, el cálculo de una prima de seguro, una decisión médica o el estado del tiempo, cuando las cosas no pueden ser previstas experimentamos algo donde la probabilidad está presente.

Pero, ¿qué es la probabilidad? La probabilidad es un área de las matemáticas, su concepción está ligada a la idea del azar y es aplicada en muchas áreas: física, otras ramas de las matemáticas, química, sociología, economía, política, medicina, etc. La teoría de la probabilidad se encarga del estudio de fenómenos aleatorios. Gracias a ella se pueden resolver problemas y modelar situaciones reales donde hay incertidumbre.

Es importante tener clara la diferencia entre los fenómenos deterministas y los fenómenos aleatorios, ya que este trabajo esta enfocado a la explotación de bases de datos, se deberá saber en qué contexto utilizar cada herramienta y la metodología estadística adecuada de manera efectiva.

Un fenómeno es determinista cuando podemos adelantarnos a un resultado sin antes haber realizado un experimento. Es decir, cuando se puede predecir con exactitud lo que pasará bajo ciertas condiciones.

En contraste con lo anterior, un fenómeno aleatorio se presenta cuando no es posible conocer los resultados antes de realizar el experimento, es decir, cuando no hay certeza qué es lo que sucederá.

Tener claras las ideas anteriores y los conceptos básicos de esta área tienen un papel crucial en la aplicación de la inferencia estadística, ya que una decisión fundada en la información de una muestra aleatoria puede estar equivocada sin un adecuado discernimiento de las leyes elementales de probabilidad.

2.3.1. Eventos, espacios muestrales y medida de probabilidad

Se define al *espacio muestral* como el conjunto de todos los elementos posibles que puede tener un experimento [9]. Se denota generalmente con la letra Ω.

Se entiende por *evento* a cualquier subconjunto del espacio muestral. Dicho de otra forma los eventos son los resultados del experimento u observaciones [5].

Para dejar claros los siguientes conceptos, considere el ejemplo clásico de la literatura: si se considera el experimento de lanzar un dado, claramente los resultados posibles a salir en un tiro serían 1, 2, 3, 4, 5 y 6; en este ejemplo, se puede apreciar que el espacio muestral es $\Omega = \{1, 2, 3, 4, 5, 6\}$ y como ejemplo de evento, se podría definir al conjunto $A = \{2, 4, 6\}$ que sería el evento de obtener un número par como resultado del experimento.

Con cualquier experimento entonces, podemos ver que siempre hay incertidumbre, por ello se asigna una "medida" de probabilidad entre 0 y 1, de manera que si se está seguro que el evento ocurrirá, decimos que es 1 o 100 % probable, en cambio si se está seguro que el evento no va a ocurrir, entonces se le asigna cero. De aquí, se tiene el enfoque clásico de probabilidad, donde la probabilidad de un evento es calculada como:

$$\mathbb{P}(A) = \frac{Número\, de\, casos\, favorables\, para\, A}{Número\, de\, casos\, posibles}$$

Formalmente una medida de probabilidad es una función que va de los eventos al [0,1] y que cumple:

- $\mathbb{P}(\Omega) = 1$.
- $\mathbb{P}(\varnothing) = 0$.
- Si $A_1, A_2, A_3, ...$ es una familia de conjuntos mutuamente excluyentes, entonces[7] $\mathbb{P}\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mathbb{P}(A_n)$.

2.3.2. Teoría de conjuntos

Ya que los eventos son representados como conjuntos, se recordarán algunas propiedades básicas de la teoría de conjuntos [8]:

- $A \cup B = B \cup A$ y $A \cap B = B \cap A$ (conmutatividad).
- $A \cup (B \cup C) = (A \cup B) \cup C = A \cup B \cup C$ (asociatividad de uniones).
- $A \cap (B \cap C) = (A \cap B) \cap C = A \cap B \cap C$ (asociatividad de intersecciones).
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (distributividad I).
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (distributividad II).
- $A - B = A \cap B^c$.
- $A \cup \varnothing = A$ y $A \cap \varnothing = \varnothing$.
- $A \cup \Omega = \Omega$ y $A \cap \Omega = A$.
- $(A \cup B)^c = A^c \cap B^c$ (primera Ley de Morgan).
- $(A \cap B)^c = A^c \cup B^c$ (segunda Ley de Morgan).
- Para cualesquiera conjuntos A y B se cumple , $A = (A \cap B) \cup (A \cap B^c)$.

[7] Se dice que dos conjuntos son *mutuamente excluyentes o disjuntos* si no tienen resultados en común, en otras palabras $A_1 \bigcap A_2 = \varnothing$.

2.3.3. Relaciones entre eventos

Teniendo claras las ideas anteriores, se pueden dar algunas reglas básicas de probabilidad:

- $\mathbb{P}(A - B) = \mathbb{P}(A) - \mathbb{P}(A \cap B)$.
- $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$.
- $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.
- $\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$.
- $\mathbb{P}(A - B) = \mathbb{P}(A) - \mathbb{P}(B)$ si $B \subseteq A$.
- Si $B \subset A$, $\mathbb{P}(B) \leq \mathbb{P}(A)$.
- $\mathbb{P}(A) = \mathbb{P}(A \cap B) + \mathbb{P}(A \cap B^c)$.
- $\mathbb{P}\left(\bigcup_{n=1}^{\infty} A_n\right) \leq \sum_{n=1}^{\infty} \mathbb{P}(A_n)$ (desigualdad de Boole).

2.3.4. Probabilidad condicional e independencia

Un concepto de vital importancia en la teoría de probabilidad es el de independencia. Una forma simple de entenderlo sería considerar el evento "que llueva" y el evento "al nacer, el género de un bebé sea femenino". Si se reflexiona en esto, es claro pensar que los eventos no tienen relación, es decir, son independientes.

Entonces, se dice que *dos eventos son independientes* si el hecho de que uno ocurra no afecta la probabilidad de la ocurrencia del otro. Si esta condición se cumple entonces:

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B).$$

A continuación presentamos la idea que da origen a esta definición, que está relacionada con el concepto de probabilidad condicional, donde informalmente se restringe el espacio muestral a un conjunto B de manera que se calcula la probabilidad del evento A dado que B sucedió. Considere el siguiente ejemplo: supongamos que la probabilidad de elegir al azar a un estudiante dentro de cuidad universitaria es equitativa para el género, es decir $\frac{1}{2}$ para mujer y $\frac{1}{2}$ para hombre. El evento A será la probabilidad de escoger a una mujer. Pero, ¿qué sucede si se condiciona este evento a la Facultad de Ingeniería? El evento sería descrito como "seleccionar una mujer dentro de C.U. dado que estudia en la Facultad de Ingeniería". Sabemos que durante años la mayoría de los ingenieros son hombres, por lo tanto, la probabilidad de escoger a una mujer disminuye. De esta manera el espacio muestral se reduce y la probabilidad de mi evento se afecta. Esto es la *probabilidad condicional*.

Lo anterior motiva a definir la probabilidad condicional de un evento A dado un evento B, $\mathbb{P}(A \mid B)$, de la siguiente manera:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Con las dos definiciones anteriores tenemos las propiedades siguientes[8]

- **Regla del producto** Si $A_1, A_2, ..., A_n$ son eventos tales que la probabilidad de su intersección es mayor que cero, entonces:

$$\mathbb{P}(A_1 \cap ... \cap A_n) = \mathbb{P}(A_1)\mathbb{P}(A_2 \mid A_1)\mathbb{P}(A_3 \mid A_1 \cap A_2)...\mathbb{P}(A_n \mid A_1 \cap ... \cap A_{n-1}).$$

- **Teorema de probabilidad total** Si $B_1, B_2, B_3, ..., B_n$ eventos que forman una partición del espacio muestral Ω, tal que $\mathbb{P}(B_i) > 0$, entonces:

$$\mathbb{P}(A) = \sum_{i=1}^{n} \mathbb{P}(A \mid B_i)\mathbb{P}(B_i).$$

- **Teorema de Bayes** Teorema de probabilidad total Si $B_1, B_2, B_3, ..., B_n$ eventos que forman una partición de Ω, tal que $\mathbb{P}(B_i) > 0$, entonces:

$$\mathbb{P}(B_j \mid A) = \frac{\mathbb{P}(A \mid B_j)\mathbb{P}(B_j)}{\sum_{i=1}^{n} \mathbb{P}(A \mid B_i) \sum_{i=1}^{n} \mathbb{P}(B_i)}.$$

  Una versión alternativa del Teorema de Bayes para solo dos eventos.:

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \mid B)\mathbb{P}(B)}{\mathbb{P}(A)}.$$

2.3.5. Variables aleatorias

Dado un experimento aleatorio cualquiera, una *variable aleatoria*, abreviada como v.a. es una transformación X del espacio muestral al conjunto de números reales, esto es [9]:

$$X : \Omega \longrightarrow \mathbb{R}.$$

Al realizar un experimento, se obtiene un resultado ω dentro del espacio muestral Ω. Y al transformar este resultado con la variable aleatoria X se obtiene un numero real $X(\omega) = x$

Veámoslo desde otra perspectiva, cuando realizamos un experimento, muchas veces nos interesará no el resultado en si del experimento sino una función de él. Por ejemplo, al lanzar dos dados, puede que nos interese que el valor de la suma sea 7, y no las maneras en que se obtuvo este resultado $(1, 6), (2, 5), (3, 4), (4, 3), (5, 2)$ o $(6, 1)$.

Informalmente, una *variable aleatoria* es una función cuyo dominio es el espacio muestral y su codominio son los números reales que representan cantidades de interés de nuestro experimento aleatorio [10].

Dicho de otra manera, una variable aleatoria es una función que va del espacio muestral a los reales y modela "adecuadamente" los eventos a los cuales se les quiere calcular probabilidad[9].

Es necesario esclarecer la diferencia entre diversos tipos de variables aleatorias. Las v.a. que toman un número finito o infinito contable de valores se denomina *variable aleatoria discreta*

[8] Las demostraciones se pueden consultar en [9]

[9] En sentido estricto la definición formal involucra conceptos de imagen inversa y sigma álgebras, pero dicha definición no será útil para el presente trabajo.

mientras que una que toma un número infinito no contable de valores se llama variable *aleatoria continua*[10]. Un ejemplo de v.a. discreta es el número de hijos de una familia, el resultado al tirar un dado. Como ejemplos de v.a. continua son el resultado de un generador de números aleatorios entre 0 y 1 o el tiempo de duración de una pila.

2.3.6. Funciones de distribución y densidad

Dentro de la teoría de probabilidad existen dos funciones que proveen información acerca de una variable aleatoria. En estas funciones se representan al mismo tiempo el espacio muestral, asi como las probabilidades de dichos eventos [9].

Una *función de distribución* asocia a cada valor de la variable la probabilidad **acumulada** hasta ese valor y se denota $F_X(x)$.

La distribución $F_X(x)$ es una función que cumple las siguientes características:

- Es monótona no decreciente.
- Es continua por la derecha.
- $\lim_{x\to\infty} F_X(x) = 1$.
- $\lim_{x\to-\infty} F_X(x) = 0$.

Por otro lado la *función de densidad* de una variable aleatoria tiene una interpretación distinta dependiendo del tipo de variable que se maneje. Esta función describe la probabilidad según la cual dicha variable tomará determinado valor o estará dentro de algún intervalo de valores y se denota $f_X(x)$.

Para variables aleatorias discretas

Para una v.a. discreta X la función de densidad es:

$$f_X(x) = \mathbb{P}(X = x) \text{ para } x \in X.$$

Se entiende por la probabilidad de que la variable aleatoria X tome el valor de x. De esta manera se asigna una probabilidad a cada x. La función de densidad para variables discretas cumple:

- $f_X(x) \geq 0$.
- $\sum f_X(x) = 1$.

Para una v.a. discreta X la función de distribución puede ser calculada:

$$F_X(x) = \mathbb{P}(X \leq x) = \sum_{n\leq x} \mathbb{P}[X = n].$$

[10]Ésta no es la definición formal de variable aleatoria continua. Una variable aleatoria es continua si su función de distribución (que se definirá más adelante) es continua.

Ejemplo [8]: Supongamos que se lanza una moneda dos veces de manera que el espacio muestral es $\Omega = CC, CS, SC, SS$, y sea X una v.a. que cuenta el número de caras que pueden resultar. Entonces los valores posibles para X según nuestro espacio muestral, $X = 0$ si el resultado es SS, $X = 1$ si es CS o SC y $X = 2$ si es CC. Por lo tanto la función de densidad de X es:

$$f_X(x) = \begin{cases} \frac{1}{4} & x = 0 \\ \frac{1}{2} & x = 1 \\ \frac{1}{4} & x = 2 \end{cases}$$

La interpretación es sencilla: la probabilidad de tener cero caras es igual a un $\frac{1}{4}$, ya que solo hay 1 caso favorable de 4. La probabilidad de que el resultado sea una cara es $\frac{1}{2}$, ya que hay 2 casos favorables de 4. Finalmente sólo hay 1 caso favorable para que haya dos caras, por lo tanto su probabilidad es $\frac{1}{4}$.

La función de distribución para este ejemplo es la siguiente:

$$F_X(x) = \begin{cases} 0 & -\infty < x < 0 \\ \frac{1}{4} & 0 \leq x < 1 \\ \frac{3}{4} & 1 \leq x < 2 \\ 1 & 2 \leq x < \infty \end{cases}$$

La interpretación es la siguiente: la probabilidad de que el número de caras sea menor a cero, es cero. La probabilidad de que el número de caras sea menor estricto que 1, es lo mismo que la probabilidad de que el número sea cero y por lo tanto $\frac{1}{4}$. La probabilidad de que el número de caras sea igual o mayor a dos es 1, ya que esto abarca todas las posibilidades.

Para variables aleatorias continuas

Para una v.a. continua la probabilidad de que X tome exactamente un valor generalmente es cero, por lo tanto no es posible definir la función de densidad de la misma manera que para una variable discreta. Para el caso de las variables continuas, la función de densidad (si existe) es aquella que satisface la igualdad:

$$\mathbb{P}(X \in (a, b)) = \int_a^b f_X(x)dx \quad \text{para cualquier intervalo (a,b).}$$

La función de densidad de una v.a. continua cumple:

- $f_X(x) \geq 0$, para toda $x \in \mathbb{R}$.
- $\int_{-\infty}^{\infty} f_X(x)dx = 1$.

Observación: Cualquier función que cumpla las dos propiedades anteriores se llamará función de densidad sin necesidad de involucrar una variable aleatoria [9].

La función de distribución, análogamente a la de la v.a. discreta está dada de la siguiente forma:

$$F_X(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^{x} f_X(t)dt.$$

Ejemplo [1]: La variable aleatoria que representa la proporción de accidentes automovilísticos en Estados Unidos, tiene la siguiente función de densidad:

$$f(x) = \begin{cases} 42x(1-x)^5 & 0 < x \leq 1 \\ 0 & para \quad cualquier \quad otro \quad valor. \end{cases}$$

¿Cuál es la probabilidad de que no más del 25 % de los accidentes sean fatales? (En otras palabras, ¿cuál es $\mathbb{P}(X \leq 0.25)$?

$$\int_0^{.25} 42x(1-x)^5 dx = 0.5551.$$

Por lo tanto la probabilidad es del 55.51 %[11].

2.3.7. Distribuciones conjuntas y marginales

Hasta este momento me ha limitado a hablar de una sola variable aleatoria, sin embargo bajo el contexto de las bases de datos es importante prever que habrá momentos en los que se tenga más de una variable aleatoria de interés y será necesario tener resultados de diferentes experimentos simultáneamente. Por ello es importante estudiar el comportamiento de las variables en conjunto.

Por simplicidad, consideraremos el caso de dos variables aleatorias. Los casos más comunes son aquellos donde ambas variables son discretas o ambas continuas, sino fuese de esta manera pueden hacerse las modificaciones apropiadas.

Caso discreto

Si X y Y son dos variables aleatorias discretas, se define la *función de probabilidad conjunta* como:

$$f_{X,Y}(x,y) = \mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x \cap Y = y).$$

donde:
1. $f_{X,Y}(x,y) \geq 0$.
2. $\sum_x \sum_y f_{X,Y}(x,y) = 1$. Es decir la suma sobre los valores de la distribución conjunta es 1.

Una función de probabilidad conjunta para dos variables puede representarse de la siguiente manera:

Los x_i son todos los posibles valores que puede tomar la v.a. X y los valores y_i para Y. Y de esta manera puede obtenerse la probabilidad conjunta por ejemplo, de que $X = x_1$ y $Y = y_2$ que está dada por $f_{XY}(x_1, y_2)$.

Por otra parte, se le llama *función de probabilidad marginal* cuando a partir de la función de probabilidad conjunta obtenemos la función de probabilidad de una sola variable, eliminando el efecto de la otra variable, es decir, obtenemos las probabilidades de los valores de X sin importar el valor de Y o viceversa. Dicho de otra manera, la densidad de probabilidad de una sola variable.

De esta manera, la probabilidad de que $X = x_1$ se obtiene sumando todos los valores de la fila de x_1, así mismo para cada valor de x_i de manera que:

[11] Para consultar el desarrollo, página 62 de [1]

Cuadro 2.1: Probabilidad Conjunta

XY	y_1	y_2	...	y_n	Totales
x_1	$f_{XY}(x_1, y_1)$	$f_{XY}(x_1, y_2)$	...	$f_{XY}(x_1, y_n)$	$f_X(x_1)$
x_2	$f_{XY}(x_2, y_1)$	$f_{XY}(x_2, y_1)$	...	$f_{XY}(x_2, y_n)$	$f_X(x_2)$
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
x_m	$f_{XY}(x_m, y_1)$	$f_{XY}(x_m, y_1)$	...	$f_{XY}(x_m, y_n)$	$f_X(x_m)$
Totales	$f_Y(y_1)$	$f_Y(y_2)$	...	$f_Y(y_n)$	1

$$\mathbb{P}(X = x_i) = f_X(x_i) = \sum_k f_{XY}(x_i, y_k).$$

Análogamente para la probabilidad de que $Y = y_1$ se tendría sumando sobre los valores de las columnas correspondientes:

$$\mathbb{P}(Y = y_i) = f_Y(y_i) = \sum_i f_{XY}(x_i, y_k).$$

Así debe notarse que,
- $f_X(x_i) = 1$.
- $f_Y(y_k) = 1$.

Caso continuo

El caso donde ambas variables son continuas es análogo al caso discreto y se obtiene al reemplazar la suma por integrales. Entonces, la función de probabilidad conjunta para dos v.a. continuas X y Y se define:

$$\mathbb{P}(X \leq a, Y \leq b) = \int_{-\infty}^{a} \int_{-\infty}^{b} f_{XY}(x, y) dy dx.$$

donde:
1. $f_{X,Y}(x, y) \geq 0$.
2. $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{XY}(x, y) dy dx = 1$.

Finalmente para ambos casos, continuo y discreto, si las variables son independientes se cumple que:

$$f_{XY}(x, y) = f_X(x) f_Y(y).$$

Y, como se dijo en la sección de independencia para el caso discreto, se interpreta como la probabilidad conjunta de $X = x$ y $Y = y$ es igual al producto de la probabilidad de $X = x$ con la probabilidad de que $Y = y$.

2.3.8. Distribuciones condicionales

Como ya se mencionó,

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \text{ cuando } \mathbb{P}(B) > 0$$

Entonces, si X y Y son variables aleatorias discretas:

$$\mathbb{P}(Y = y \mid X = x) = \frac{f_{XY}(x, y)}{f_X(x)}$$

Análogamente para $\mathbb{P}(X = x \mid Y = y)$.

De manera que se puede ampliar la idea para *función de probabilidad condicional*:

$$f_{X|Y}(y \mid x) = \frac{f_{XY}(x, y)}{f_X(x)}.$$

Dada la idea anterior, se puede llevar la definición para v.a. continuas [8], por ejemplo, la probabilidad de que Y este entre a y b cuando $X < x$ sería:

$$\mathbb{P}(a < Y < b \mid X < x) = \int_a^b f(y \mid x)dy.$$

2.3.9. Esperanza, varianza, covarianza y coeficiente de correlación

Existen muchas medidas características de las distribuciones de probabilidad, cada una de estas medidas aportan información sobre el comportamiento de una variable aleatoria. Un concepto de importancia en probabilidad y estadística es el de esperanza, varias de las medidas más frecuentes están basadas en ella aunque pueden tener significados diferentes (posición central, dispersión, asimetría, etc).

Esperanza

La *esperanza* representa el valor promedio de una variable aleatoria después de un número grande de experimentos [1], también es llamada *media, valor esperado o valor promedio.*

Para una variable aleatoria X que tiene como función de densidad a $f_X(x)$, el valor esperado es denotado como $\mathbb{E}(X)$ o μ_x y se calcula como:

$$\mathbb{E}(X) = \begin{cases} \sum_x x f_x(x) & \text{si } X \text{ es discreta,} \\ \\ \int_{-\infty}^{\infty} x f_x(x) & \text{si } X \text{ es continua.} \end{cases}$$

En general, el valor esperado de una función $g(x)$ de una variable aleatoria X está dado por:

$$\mathbb{E}[g(X)] = \begin{cases} \sum_x g(x) f_x(x) & \text{si } X \text{ es discreta,} \\ \\ \int_{-\infty}^{\infty} g(x) f_x(x) & \text{si } X \text{ es continua.} \end{cases}$$

La esperanza de una variable aleatoria X no es una función de X sino un número fijo y una propiedad de la distribución de probabilidad de X.

Ejemplo [8]: Sea X la variable aleatoria que representa la cantidad de dinero que puede ganarse con cualquier lanzamiento de un dado según el siguiente cuadro:

Cuadro 2.2: Juego con dado

	1	2	3	4	5	6
x_i	\$0	+\$20	\$0	+\$40	\$0	-\$30
$f_x(x)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

De esta manera, la pregunta a responder es ¿cuál es la ganancia que se puede esperar al jugar? Calculando la esperanza de X se tiene:

$$\mathbb{E}(X) = (0)(\tfrac{1}{6}) + (20)(\tfrac{1}{6}) + (0)(\tfrac{1}{6}) + (40)(\tfrac{1}{6}) + (0)(\tfrac{1}{6}) - (30)(\tfrac{1}{6}) = 5.$$

Se deduce entonces que el jugador puede esperar una ganancia de \$5.00.

Propiedades de la esperanza: Si X y Y son variables aleatorias con esperanza finita y c una constante. Entonces:

1. $\mathbb{E}(c) = c$.
2. $\mathbb{E}(cX) = c\mathbb{E}(X)$.
3. $\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y)$.

Varianza y desviación estándar

La *varianza* es otra característica numérica asociada a variables aleatorias. Se denota por $Var(X)$ o σ_x^2 y se define por:

$$\sigma_x^2 = Var(X) = \mathbb{E}[(x-\mu)^2] = \begin{cases} \sum_x (x-\mu)^2 f_x(x) & \text{si } X \text{ es discreta,} \\ \\ \int_{-\infty}^{\infty} (X-\mu)^2 f_x(x) & \text{si } X \text{ es continua.} \end{cases}$$

Propiedades de la varianza:[12] Si X y Y son dos variables aleatorias y c una constante. Entonces:

1. $Var(X) \geq 0$.
2. $Var(c) = 0$.
3. $Var(cX) = c^2 Var(X)$.
4. $Var(X + c) = Var(X)$.
5. $Var(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X)$[13].
6. En general, $Var(X + Y) \neq Var(X) + Var(Y)$.

[12]Se pueden ver las demostraciones en [9].
[13]Esta es la manera habitual de calcular la varianza.

La varianza es una medida de la dispersión de la distribución de probabilidad de una variable aleatoria. Por ejemplo, en el caso continuo si la mayor parte del área por debajo de la curva de la distribución se encuentra cercana a la media, la varianza será pequeña; si la mayor parte del área se encuentra dispersa alrededor de la media, la varianza será grande.

A la raíz cuadrada positiva de la varianza se le da el nombre de *desviación estándar* y se denota por σ, este valor indica en promedio cuanto se alejan los datos de la media (Figura 2.4 [14]).

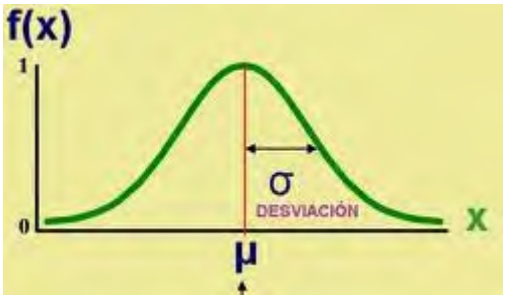


Figura 2.4: Desviación estándar.

Los resultados anteriores pueden ampliarse para dos o mas variables, de manera que las esperanzas son:

$$\mu_x = \mathbb{E}[X] = \sum_x \sum_y x f(x,y) \quad \text{si } X \text{ y } Y \text{ son discretas,}$$
$$\mu_x = \mathbb{E}[X] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x,y) dxdy \quad \text{si } X \text{ y } Y \text{ son continuas.}$$

$$\mu_y = \mathbb{E}[Y] = \sum_x \sum_y y f(x,y) \quad \text{si } X \text{ y } Y \text{ son discretas,}$$
$$\mu_y = \mathbb{E}[Y] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y f(x,y) dxdy \quad \text{si } X \text{ y } Y \text{ son continuas.}$$

Y las varianzas son:

$$\sigma_x^2 = \sum_x \sum_y (x-\mu_x)^2 f(x,y) \quad \text{si } X \text{ y } Y \text{ son discretas,}$$
$$\sigma_x^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-\mu_x)^2 f(x,y) dxdy \quad \text{si } X \text{ y } Y \text{ son continuas.}$$
$$\sigma_y^2 = \sum_x \sum_y (y-\mu_y)^2 f(x,y) \quad \text{si } X \text{ y } Y \text{ son discretas,}$$
$$\sigma_y^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (y-\mu_y)^2 f(x,y) dxdy \quad \text{si } X \text{ y } Y \text{ son continuas.}$$

Covarianza

En el caso de dos variables X y Y aparece otra medida importante, la *covarianza*, ésta es una medida del grado en que dos variables aleatorias se mueven linealmente en la misma dirección o en direcciones opuestas la una respecto a la otra. Está definida como:

$$\sigma_{XY} = Cov(X,Y) = \mathbb{E}[(x-\mu_x)(y-\mu_y)].$$

[14] https://goo.gl/3MgNsG

Por lo tanto:

$$\sigma_{XY} = Cov(X,Y) = \sum_x \sum_y (x-\mu_x)(y-\mu_y)f(x,y) \quad \text{si } X \text{ y } Y \text{ son discretas,}$$

$$\sigma_{XY} = Cov(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\mu_x)(y-\mu_y)f(x,y)dxdy \quad \text{si } X \text{ y } Y \text{ son continuas.}$$

De este modo:

- Si $\sigma_{XY} \geqslant 0$ hay dependencia lineal positiva, es decir a grandes valores de X le corresponden valores grandes de Y.
- Si $\sigma_{XY} \leqslant 0$ hay dependencia lineal negativa, es decir a grandes valores de X corresponden valores pequeños de Y.
- Finalmente, si $\sigma_{XY} = 0$ se dice que no hay existencia de una relación lineal entre X y Y.

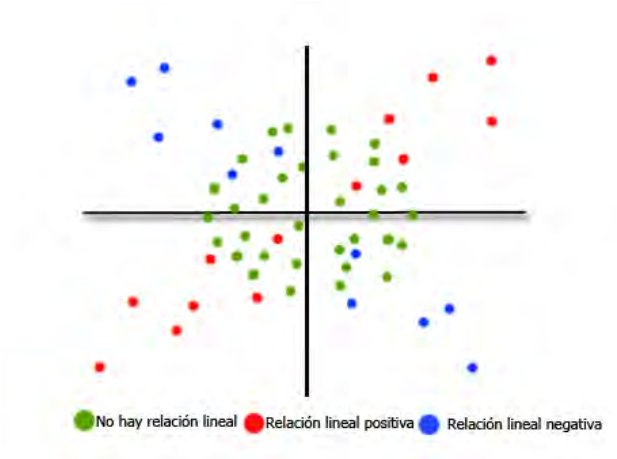


Figura 2.5: Interpretación de la covarianza.

A continuación algunas propiedades importantes sobre de la covarianza. Si X, Y y Z son variables aleatorias con varianza finita y a una constante. Entonces:

1. $Cov(X,Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$[15].
2. $Cov(X,Y) = Cov(Y,X)$.
3. $Cov(X,X) = Var(X)$.
4. $Cov(a,X) = 0$.
5. $Cov(aX,Y) = aCov(X,Y)$.
6. $Cov(X+Y,Z) = Cov(X,Z) + Cov(Y,Z)$.
7. Si X y Y son variables aleatorias independientes, entonces $Cov(X,Y) = 0$. El recíproco es falso.

[15]Esta suele ser la manera habitual de calcular la covarianza.

Coeficiente de correlación

Si la covarianza se divide por el producto de las desviaciones estándar de X y Y, el resultado es un valor que recibe el nombre de *coeficiente de correlación* y es denotado de la siguiente forma:

$$\rho = \frac{Cov(X,Y)}{\sigma_x \sigma_x}.$$

El coeficiente de correlación es una medida estandarizada de la asociación lineal que existe entre las variables X y Y en relación con sus dispersiones [1].

El valor del coeficiente puede variar entre -1 y 1. Si $\rho > 0$ se dice que hay una correlación lineal positiva, es decir, al aumentar los valores de una variable aumentan los de la segunda, en particular, si $\rho = 1$ se dice que hay una relación lineal positiva perfecta. En sentido contrario, si $\rho < 0$, las variables se relacionan en sentido inverso y si $\rho = -1$ entonces se dice que tienen una correlación negativa perfecta. Finalmente, cuando $\rho = 0$, solo se puede decir que no hay una relación lineal entre las variables, sin embargo, esto no implica de ninguna manera que las variables sean independientes.

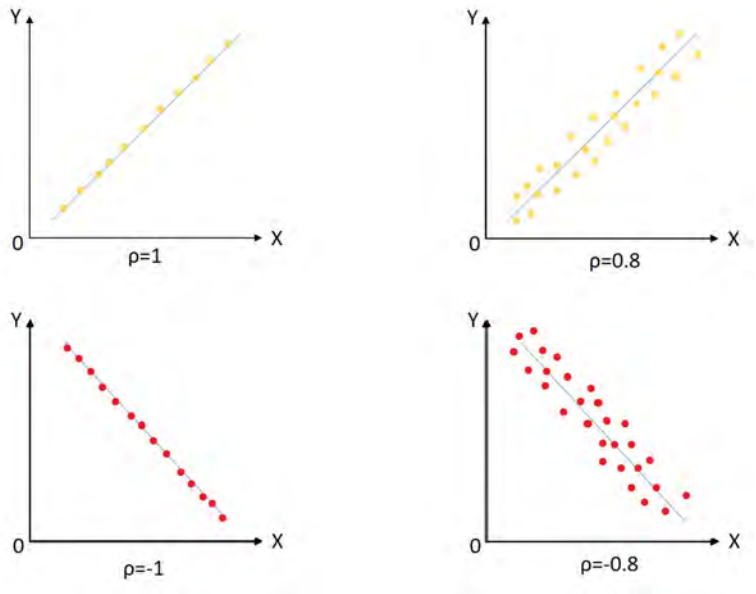


Figura 2.6: Coeficiente de correlación.

Diferencias entre varianza, covarianza y coeficiente de correlación.

- La varianza es una medida de la dispersión o variación de los valores de la variable aleatoria alrededor de la media.
- La covarianza una medida del tipo de relación lineal entre dos variables.
- La correlación indica la fuerza y dirección de la asociación entre dos variables aleatorias en forma de relación lineal.

2.4. Estadística para explotación de datos

La estadística es la rama de las matemáticas que recolecta, organiza, analiza e interpreta conjuntos de datos para obtener inferencias o conclusiones basadas en el cálculo de probabilidades para la toma de decisiones.

Frecuentemente se está interesado en obtener conclusiones respecto a un grupo grande de individuos u objetos, al conjunto entero de elementos a estudiar se le llama *población*. Por otro lado, en la mayoría de las ocasiones resulta imposible estudiar una población de interés entera, por lo que solamente se examina a una pequeña parte de ella, a la que se le llama *muestra*.

La estadística se divide en dos grandes áreas, una de ellas es la *estadística descriptiva*, está formada por procedimientos empleados para resumir y describir las características principales de un conjunto de datos [7]. Por otra parte, la *estadística inferencial* hace referencia a un conjunto de técnicas que permiten hacer inferencias inductivas con determinado grado de confianza acerca de características poblacionales, a partir de información contenida en una muestra sacada de esta población.

2.4.1. Tipos de Variables

El elemento básico de la estadística lo forman los *datos*, éstos provienen de las variables de las cuales expresan algún tipo de característica. Una *variable* es una característica que cambia o varía para diferentes personas u objetos de una población en estudio.

Por ejemplo, en un estudio la variable puede ser el color de ojos, y el dato es café, azul o verde según la persona.

De esta manera, es importante identificar el tipo de valores que pueden tomar las variables ya que de esto depende el tipo de análisis estadístico que tiene sentido realizar. La clasificación de las variables con base en sus datos es la siguiente:

Variables cuantitativas Son aquellas que tienen un valor numérico y que expresan una medida. Éstas a su vez, pueden ser:

- Continuas: son aquellas que pueden tomar cualquier valor real dentro de un intervalo por lo que para cualquier par de valores siempre se puede encontrar un valor intermedio.
- Discretas: son las variables que solo pueden tomar valores enteros.

Variables cualitativas Éstas son las que expresan una cualidad o característica, pueden ser expresadas numéricamente, sin embargo, no se pueden hacer operaciones con ellas.

- Dicotómicas: son aquellas que sólo pueden tomar dos valores.
- Nominales: son datos que corresponden a categorías que por su naturaleza no admiten un orden.
- Ordinales: son valores de la variable que corresponden a un tipo de evaluación subjetiva, de manera que tienen un orden o jerarquía.

En el siguiente diagrama se muestran algunos ejemplos:

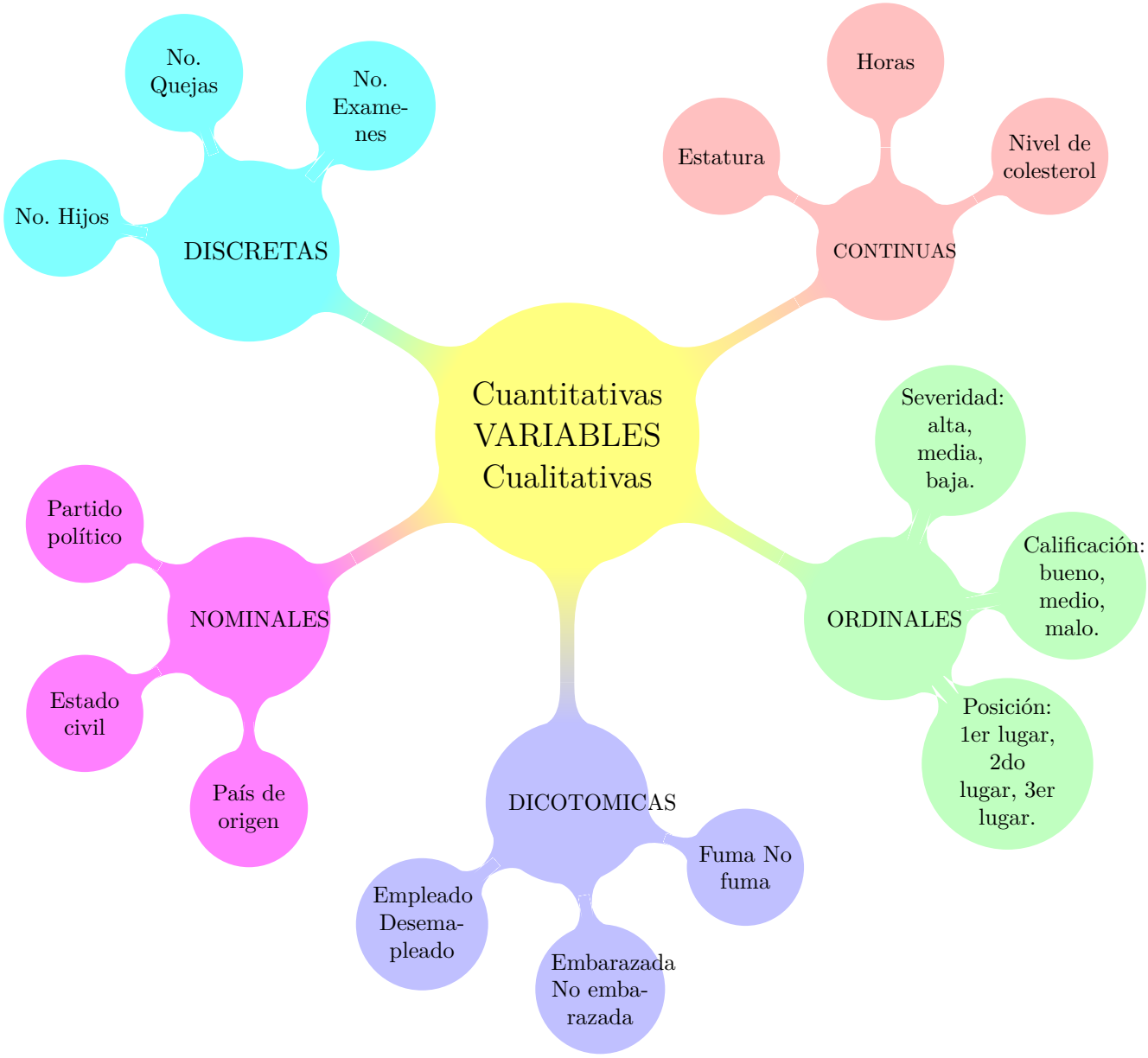


Figura 2.7: Tipo de variables.

2.4.2. Análisis exploratorio y otras medidas de tendencia central y dispersión

El análisis exploratorio o estadística descriptiva busca resumir e identificar las características de un conjunto de datos a través de una cantidad reducida de tablas, gráficos y números. Para ello se pueden calcular ciertas medidas que se describen a continuación.

Las **medidas de tendencia central**, son medidas estadísticas que sirven como puntos de referencia para resumir en un solo valor cierta información de un conjunto de datos. Representan un valor en torno al cual se hallan el resto de los datos. Dos de estas medidas son la media y la mediana.

La *media aritmética* es un punto de referencia estándar, también llamada promedio. Es calculada de la siguiente forma:

$$\overline{X} = \tfrac{1}{n} \sum_{i=1}^{n} x_i.$$

La *mediana* es el valor que separa a los datos en dos grupos con la misma cantidad de datos, cuando están ordenados de tal manera que el 50 % de los datos son mayores a este valor y el otro 50 % son menores. Es denotada en algunos libros por $\widetilde{X}$ y se calcula de la siguiente forma en una muestra ordenada:

$$\widetilde{X} = \begin{cases} \frac{1}{2}\left[x\left(\frac{n}{2}\right) + x\left(\frac{n}{2}+1\right)\right] & \text{si } n \text{ es par,} \\ \\ x\left[\frac{n+1}{2}\right] & \text{si } n \text{ es impar.} \end{cases}$$

La *moda* es el valor observado con mayor frecuencia, este valor puede no existir o, en ciertos casos, puede no ser única.

Existen también las **medidas de dispersión** como la varianza y la desviación estándar. La *varianza* (S^2) es el promedio del cuadrado de las distancias entre cada observación y la media aritmética, corregido por un factor de insesgamiento por lo que se divide entre $n-1$ y no entre n. La *desviación estándar* es la raíz cuadrada de la varianza y nos indica en promedio cuánto se alejan los datos de la media.

$$S^2 = \tfrac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2.$$

Otro tipo de medidas descriptivas son las **medidas de posición**, las cuales dividen a un conjunto de datos de una distribución en base a si exceden cierto valor.

Un ejemplo de medida de posición son los *cuantiles*, los cuales son valores que dividen a un conjunto de datos en partes iguales. Los cuantiles más utilizados son los *cuartiles*, los cuales dividen a los datos en cuatro grupos con el mismo número de elementos. Son denotados por Q_1, Q_2 (mediana) y Q_3 y determinan los valores correspondientes al 25 %, al 50 % y al 75 % de los datos.

Con todas estas medidas se pueden construir gráficos donde se pueda apreciar los datos como un todo e identificar sus características. El tipo de gráfico que se utilice depende del tipo de variable que se quiera representar.

Gráficas para datos cualitativos

Gráfica de pastel o circular. Es un gráfico utilizado comúnmente para representar frecuencias relativas (proporciones) o porcentajes.

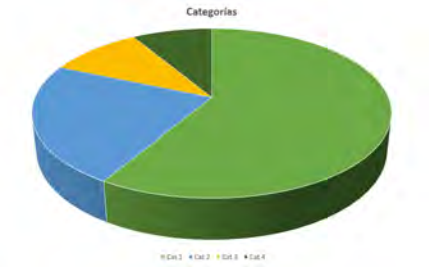


Figura 2.8: Gráfica circular.

Gráfica de barras. La gráfica de barras se suele utilizar para comparar las alturas de barras de medidas de categorías. Se pueden representar frecuencias absolutas o relativas según sea la preferencia.

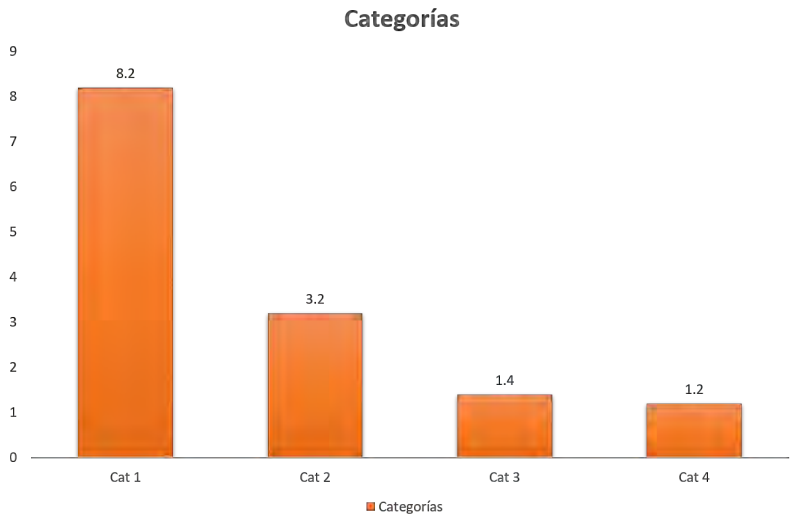


Figura 2.9: Gráfica de barras.

Gráficas para datos cuantitativos

Diagrama de caja. Un diagrama de caja es una gráfica utilizada comúnmente para representar el resumen de una variable cuantitativa, en este gráfico se visualizan los datos mínimos, máximos, los cuartiles y en algunas ocasiones los valores extremos (outliers) de la siguiente manera:

Figura 2.10: Diagrama de caja.

Histograma. Un histograma es una gráfica que se puede utilizar para evaluar la forma y dispersión de datos continuos. Sirve confirmar supuestos o distinguir la distribución de frecuencias.

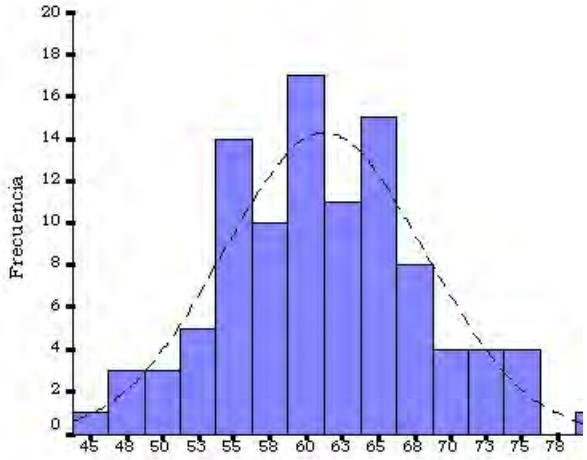


Figura 2.11: Histograma.

Diagrama de dispersión. El diagrama de dispersión es una herramienta gráfica que sirve para describir el comportamiento conjunto de dos variables, de manera que pueda identificarse una posible relación entre ellas y facilitando la interpretación de la misma.

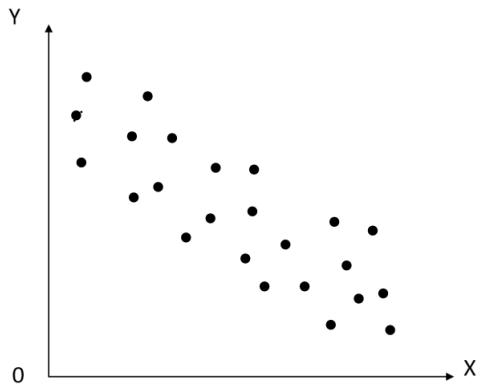


Figura 2.12: Diagrama de dispersión.

Gráficos multivariados. Existen además variedad de gráficos multivariados con distintos usos, sin embargo, con uso descriptivo puede mencionarse la gráfica de estrellas o segmentos y la gráfica de "caras" , que permiten ver de manera conjunta más de dos variables. Donde el tamaño o color de cada cara por sujeto de estudio cambia respecto a sus características.

Figura 2.13: Gráfica de caras (Chernoff).

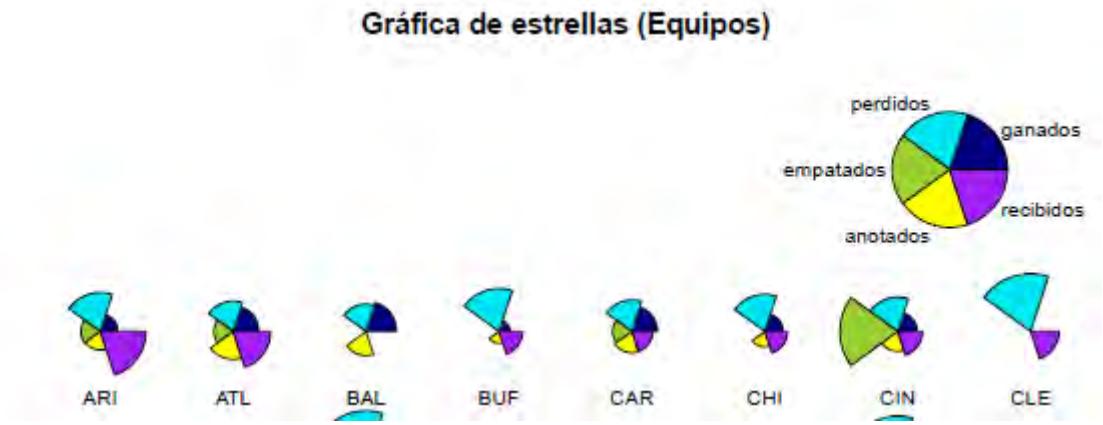


Figura 2.14: Gráfica de estrellas.

2.4.3. Estimación puntual

Cuando se conoce de manera exacta la distribución probabilística que sigue una muestra de una población, es sencillo calcular las probabilidades y otras medidas como la media, la varianza, etc., asociadas a dicha distribución. Sin embargo, en la mayor parte de los casos no se conoce la distribución exacta, por lo que dependiendo de la situación, puede suceder que únicamente se conozca la distribución pero no los parámetros, o bien no se tenga ninguna idea de la distribución. La estadística inferencial propone técnicas que permiten estimar la verdadera distribución de una población a partir de un conjunto limitado, pero representativo de datos (muestra).

Una de las técnicas más utilizadas en la estadística inferencial es la *estimación puntual*. El objetivo de la estimación puntual es utilizar la información que proporciona la muestra, para decidir de manera precisa cuál es la distribución de la muestra. Cuando se utiliza la estimación puntual, es común suponer que se conoce la distribución de la cual proviene la muestra, mas se supone que los parámetros son desconocidos; por lo que se utiliza la información que proporciona la muestra para poder dar una estimación de los parámetros.

Muestras aleatorias

Una *muestra aleatoria* (m.a.) es una colección de variables aleatorias $X_1, ..., X_n$ para las cuales se asumen dos condiciones: independencia e idéntica distribución de probabilidad [9].

Las muestras aleatorias son la materia prima de la inferencia estadística.

Estadísticas

Cuando se obtiene una muestra aleatoria, generalmente casi cualquier función de ella será una v.a. y es llamada *estadística o estimador* cuando ésta no depende de parámetros desconocidos, por ejemplo:

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Entonces, $\overline{X}$ es una función de la muestra, por tanto una variable aleatoria y una *estadística*.

La estimación de parámetros involucra el uso de los datos muestrales en conjunción con una estadística [1]. Existen dos maneras para llevar a cabo lo anterior:

- **Estimación puntual.**

  Se basa en datos muestrales para calcular un único número *(estimador puntual)* $\hat{\theta}$, que estime el parámetro buscado, denotado comúnmente como θ.

- **Estimación por intervalo.**

  De la misma forma, con base en datos muestrales se calculan dos números para formar un intervalo dentro del cual se espera esté contenido el parámetro θ.

Una característica que un buen estimador puntual debe cumplir es el ser **insesgado.** Un estimador es insesgado cuando la esperanza de su distribución coincide con la esperanza de la población, es decir, $\mathbb{E}(\hat{\theta}) = \theta$.

Método de estimación puntual: máxima verosimilitud

Uno de los métodos más relevantes de estimación puntual es el de *máxima verosimilitud*. Este método consiste en obtener el valor de θ que maximice la función de verosimilitud $L(\theta)$.

Si $X_1, ..., X_n$ es una m.a. de una población con función de densidad $f(x;\theta)$. La función de verosimilitud de la muestra es denotada por $L(\theta)$ y se define como la función de densidad conjunta vista como función de los parámetros [9], es decir:

$$L(\theta) = f_{X_1,...,X_n}(x_1, ..., x_n; \theta).$$

El estimador de máxima verosimilitud se obtiene en la mayoría de los casos, utilizando técnicas de cálculo diferencial: obtener las derivadas e igualar el sistema a cero para encontrar los puntos críticos. Sin embargo, como el máximo de una función se obtiene en el mismo punto que en el logaritmo de la función, muchas veces es más conveniente maximizar la logverosimilitud, que es el logaritmo de la verosimilitud[16].

2.4.4. Pruebas de hipótesis

El concepto de *prueba de hipótesis* tiene una fuerte relación con el concepto de estimación.

La esencia de una prueba de hipótesis es decidir si una afirmación tiene respaldo en evidencia obtenida a través de una muestra aleatoria. Generalmente la afirmación involucra algún parámetro o alguna distribución y la decisión se toma considerando si los datos muestrales apoyan estadísticamente a la afirmación y, si ésta es mínima, entonces se rechaza la afirmación [1].

Considérese el siguiente ejemplo: se tiene interés en el tiempo promedio que se necesita para terminar una unidad en una línea de armado. Bajo condiciones de producción estándares, el objetivo es que el tiempo promedio de armado por cada unidad sea de 10 minutos (ésta será la hipótesis estadística). La evidencia se encontrará en una muestra aleatoria de tamaño n obtenida de la línea de armado. Nótese que no es de interés conocer la estimación de tiempo que tarda en armarse una unidad sino determinar si el valor μ es 10. A la afirmación de que $\mu = 10$ se le llama *hipótesis nula* y se denota:

$$H_0 : \mu = 10.$$

Cuando sólo se específica un valor para la hipótesis nula, también se dice que es una *hipótesis simple*, de otra forma se le conoce como *hipótesis compuesta*, por ejemplo, si se hubiera propuesto $H_0 : \mu \geq 10$ o $H_0 : \mu \leq 10$. Una hipótesis nula debe considerarse como verdadera a menos que exista suficiente evidencia en contra [1]. En otras palabras, sólo se rechazará que el tiempo promedio de armado sea 10 si la evidencia difiere marcadamente de la afirmación.

[16]Para ver el método ejemplificado, revisar [9], página 81.

Para construir una regla de decisión apropiada, también se necesita establecer una *hipótesis alternativa,* que refleje el valor posible o el intervalo de valores del parámetro de interés. La hipótesis alternativa se denota como H_1 o H_a, para el ejemplo que se ha manejado, podríamos decir que el gerente de la planta sospecha que el promedio de armado es mayor a 10, entonces:

$$H_0 : \mu = 10 \text{ vs } H_1 : \mu > 10.$$

Al hacer una prueba de hipótesis deben considerarse los posibles errores que pueden originarse según la decisión tomada. Si se rechaza una hipótesis cuando debería ser aceptada, se dice que se comete un *error del Tipo I.* Si por el contrario, se acepta una hipótesis que debería ser rechazada, se dice que se comete un *error del Tipo II.*

Para que cualquier regla de decisión sea buena se debe diseñar alguna forma que minimice errores, lo cual no es sencillo. Sin embargo, en la práctica se acostumbra a fijar la probabilidad del error tipo I y minimizar la probabilidad del error tipo II.

	Hipótesis nula	
Decisión	Verdadera	Falsa
Rechazar	Error tipo I	Decisión correcta
No rechazar	Decisión correcta	Error tipo II

Cuadro 2.3: Tipo de error.

Un ejemplo para entender la gravedad de los dos tipos de errores es el siguiente: si se estuviera en un tribunal donde se llevara a cabo un juicio, se sabe que el acusado es inocente hasta que demuestre lo contrario, entonces la hipótesis nula es la inocencia. ¿Qué tipo de error es el más grave? Si la hipótesis nula es cierta y se rechaza (error tipo I) un inocente puede ser declarado culpable, pero si es culpable un agresor puede ser dejado en libertad al equivocarse el jurado (error tipo II). Se suele considerar más grave el error tipo I.

La probabilidad máxima con la que en una prueba de hipótesis se puede cometer un error del Tipo I se llama nivel de significancia del ensayo. Esta probabilidad se denota frecuentemente por α. En la práctica se acostumbra usar $\alpha = 0{,}05$ o $\alpha = 0{,}01$, es decir, se espera que la probabilidad de error sea del 1 % o 5 %, aunque igualmente pueden emplearse otros valores.

La decisión se basa en alguna estadística que recibe el nombre de *estadística de prueba.* Para ciertos valores de la estadística de prueba, la decisión será rechazar la hipótesis nula y por lo tanto tomar la alternativa. La región de valores en donde se rechaza la hipótesis nula se conoce como *región crítica de la prueba.* Por ejemplo, para la situación planteada anteriormente, la hipótesis nula $\mu = 10$, para un tamaño de muestra n, supóngase que se decide rechazar la hipótesis si se observa un valor de la media muestral (la estadística de prueba) $\overline{X} \geq 12$. El valor crítico sería 12 y el conjunto de valores mayores a él constituyen la región crítica de la prueba.

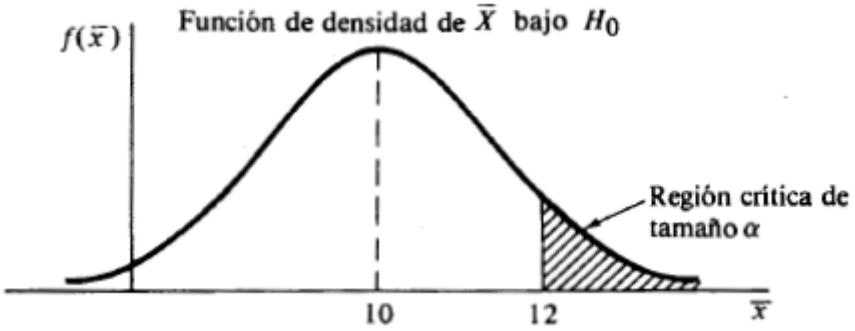


Figura 2.15: Región crítica [1].

Tal como se ve en la imagen anterior, la parte sombreada representa la región crítica. De manera que si la estadística de prueba toma un valor dentro de ella, la hipótesis nula se rechazaría.

Otra manera habitual de tomar una decisión es usando el *p-value*. El *p-value* o *valor p* es la probabilidad de que el estadístico de prueba tome un valor más extremo que el observado bajo la hipótesis nula. En un sentido amplio el p-value es una medida de la "credibilidad" de la hipótesis nula, de manera que las probabilidades más bajas proporcionan evidencia más fuerte en contra de la misma. Por tanto, cuando el p-value es menor igual que el nivel de significancia, se rechaza H_0.

2.4.5. Estadística no paramétrica

Las pruebas paramétricas asumen las distribuciones subyacentes a los datos. Por ende, deben cumplirse algunas condiciones de validez para las pruebas.

La estadística no paramétrica estudia las pruebas donde las distribuciones subyacentes no pueden conocerse a *priori*, las técnicas no paramétricas permiten probar diferentes hipótesis donde los procesos paramétricos no son viables.

Dentro de la estadística no paramétrica existen diversas pruebas como son las pruebas de bondad de ajuste, de independencia y de homogeneidad.

Las pruebas de bondad de ajuste tienen la finalidad de distinguir si un conjunto de datos se ajusta de manera correcta a una distribución específica de probabilidad. En este tipo de pruebas se comparan los valores observados con los esperados teóricamente bajo la supuesta distribución.

Las pruebas de independencia tienen el objetivo de vislumbrar si hay una dependencia entre variables cualitativas que se definen en filas y columnas (dentro de tablas de contingencia) de manera que se contrastan los valores reales con los esperados bajo hipótesis de independencia.

Algunas de las principales pruebas no paramétricas, así como algunas de sus aplicaciones son las siguientes[17]:

- **Prueba χ^2 de Pearson**. Esta prueba tiene 3 facetas, de bondad de ajuste (donde se contrasta si los datos provienen de una distribución particular), de homogeneidad (donde se compara si dos poblaciones son homogéneas) e independencia. Ya que en este trabajo se usará particularmente esta prueba, se explicará con más detalle posteriormente.

[17] https://goo.gl/v29QJ7

- **Prueba de los signos**. Esta prueba contrasta hipótesis sobre la posición central (mediana) de una distribución poblacional para analizar si dos muestras provienen de la misma distribución de probabilidad.
- **Prueba de rangos de Wilcoxon**. Esta prueba al igual que la prueba de los signos, es usada para hacer pruebas de hipótesis acerca de la mediana, compara el rango medio de dos muestras y cuenta la magnitud como la dirección de los puntajes de diferencia.
- **Coeficiente de correlación de Spearman**. Es una medida de la correlación (asociación o independencia).
- **Prueba exacta de Fisher**. La prueba de Fisher es particular para tablas de contingencia de $2x2$, la prueba determina si dos conjuntos de datos difieren en proporciones.
- **Prueba de la mediana**. Esta es una técnica para decidir si dos grupos independientes difieren en sus medidas, es decir si fueron extraídos de poblaciones diferentes.
- **Prueba de Kruskal-Wallis**. Hace análisis de varianza igual que la prueba anterior, permite saber si dos muestras son de diferentes poblaciones.
- **Prueba de Anderson-Darling**. Esta prueba analiza si los datos de una muestra provienen de una distribución específica.
- **Prueba de Friedman**. Esta prueba hace un análisis de varianza, de manera que compara para dos muestras si hay o no diferencia entre los grupos.
- **Prueba de Kendall**. Esta prueba mide el grado de asociación entre 3 o más variables.
- **Prueba de Kolmogorov-Smirnov**. Esta prueba es de bondad de ajuste, permite analizar si los datos siguen una distribución en específica.
- **Prueba de Wald-Wolfowitz**: Esta prueba esta referida a dos muestras independientes, permite contrastar la hipótesis de que ambas muestras proceden de la misma población.

Si se quisiera profundizar en cada una de las pruebas o en estadística no paramétrica, se puede consultar [3] y [6].

Prueba χ^2 de Pearson

La *prueba* χ^2, particularmente para su uso de prueba de independencia, se usa sobre tablas de contingencia[18] para evaluar si existe relación o no entre dos variables cualitativas. Esta prueba mide la discrepancia entre una distribución observada y otra teórica (bajo la hipótesis de independencia), sobre una tabla de contingencia como la siguiente:

[18]Una tabla de contingencia es una tabla que cuenta las observaciones (o frecuencias) de diferentes variables. Las filas y las columnas de las tablas corresponden a variables categóricas.

X (variable renglón)	Y (variable columna) y_1	y_2	...	y_J	Total
x_1	n_{11}	n_{12}	...	n_{1J}	$n_{1\bullet}$
x_2	n_{21}	n_{22}	...	n_{2J}	$n_{2\bullet}$
...	...	...		...	...
x_I	n_{I1}	n_{I2}	...	n_{IJ}	$n_{I\bullet}$
Total	$n_{\bullet 1}$	$n_{\bullet 2}$	...	$n_{\bullet J}$	n=$n_{\bullet\bullet}$

Cuadro 2.4: Tabla de contingencia.

Las hipótesis son:

$$H_0 = \textit{Las variables son independientes.}$$
$$H_1 = \textit{Las variables no son independientes.}$$

El estadístico de prueba que se utiliza es el siguiente:

$$X^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}};\ E_{ij} = \frac{n_{i\bullet} n_{\bullet j}}{n_{\bullet\bullet}}.$$

Donde:
E_i=Valores esperados.
O_i=Valores observados o reales.

Los grados de libertad para la prueba son gl=$(I-1)(J-1)$, es decir, la región crítica para la toma de la decisión es:

$$X^2 \geq \chi^2_{(I-1)(J-1)}.$$

A continuación presentamos un ejemplo[19]: se desea determinar si existe dependencia entre la práctica de algún deporte y la depresión bajo un nivel de significancia del 5 %, por lo que se seleccionó una muestra aleatoria de 100 jóvenes con los siguientes resultados:

	Sin depresión	**Con depresión**
Deportista	38	9
No deportista	31	22

Se calculan los valores esperados E_{ij}:

	Sin depresión	**Con depresión**	Total
Deportista	32.43	14.57	47
No deportista	36.57	16.43	53
Total	69	31	100

Bajo la prueba χ^2, se calcula su estadístico:

$$X^2 = \frac{(38-32.43)^2}{32.43} + \frac{(9-14.57)^2}{14.57} + \frac{(31-36.57)^2}{36.57} + \frac{(22-16.43)^2}{16.43} = 5.82$$

El valor anterior se compara con el percentil de la distribución χ^2 (con un grado de libertad), $\chi^2_{(2-1)(2-1),0.95\,\%} = 3.84$.

Por lo tanto el valor de la estadística es superior al valor crítico (es decir se encuentra dentro de la región crítica), por lo tanto se rechaza la hipótesis de independencia y por consecuencia se asume que existe una relación entre la depresión y los hábitos de deporte de una persona.

[19]Tómado de https://goo.gl/q6FB5Y

3 | Práctica R - Introducción a PostgreSQL en R

3.1. Objetivo

- Describir las ventajas de integrar PostgreSQL y R.
- Explicar los pasos para realizar la conexión entre PostgreSQL y R.
- Realizar consultas dentro de R para manipular datos de una instancia de PostgreSQL.

3.2. Introducción

Actualmente se acostumbra manejar grandes volúmenes de datos, sin embargo, en muchas ocasiones si no se tiene el equipo necesario, la cantidad de datos pueden exceder los recursos disponibles de nuestro sistema computacional. Dada esta situación es útil disponer de otras opciones para optimizar los recursos que se tienen.

PostgreSQL como ya se ha dicho antes, es un SMBD poderoso y con muchas capacidades lo que lo convierte en una buena opción para manipular grandes cantidades de datos.

¿Qué pasaría si pudiéramos tener esa ventaja y al mismo tiempo poder hacer análisis y explotación de datos con otro programa?

R es conocido dentro del ámbito estadístico y tiene lo necesario para el análisis de datos: contiene variedad de gráficos que pueden utilizarse con versatilidad, además de pruebas estadísticas ya implementadas. Sin embargo, al cargar bases de datos grandes, la memoria puede ser saturada y no dejar los recursos suficientes para que los cálculos puedan llevarse a cabo.

Actualmente existe una conexión entre R y PostgreSQL que permite manipular y explorar grandes volúmenes de datos sin que esto merme el rendimiento de nuestro equipo. Tener la posibilidad de manejar una gran base de datos en pequeñas consultas, aún cuando con el tiempo su tamaño crezca, permitirá al resto del programa tener disponibles los recursos computacionales para realizar cálculos y manipulación de los datos.

3.3. Conexión PostgreSQL y R

Para establecer la conexión es necesario tener instalado R, RStudio y PostgreSQL. Garantizado eso, el primer requerimiento para la conexión es instalar en RStudio el paquete RPostgreSQL, esto se puede hacer con el comando *install.packages("RPostgreSQL")*.

A continuación, se escribe el siguiente código en R:

```
#Cargamos el paquete ya instalado y declaramos el controlador
library(RPostgreSQL)

drv <- dbDriver("PostgreSQL")

#Utilizando la conexión anterior, el nombre de la base (en este caso ONEFA2)
#contenida en PostgreSQL, se especifíca el puerto, host, usuario y contraseña
conexion <- dbConnect(drv, dbname = "ONEFA2",
                      host = "localhost", port = 5432,
                      user = "postgres", password = "123456")
```

Después de esto, se puede realizar cualquier consulta que se desee, incluso el paquete 'DBI' en R permite consultar si existe una tabla con cierto nombre dentro de la base con la que se hizo la conexión. Por ejemplo:

```
library(DBI)
dbExistsTable(conexion, "equipo")

## [1] TRUE
```

Lo que confirma que la base con la que se hizo la conexión contiene una tabla llamada "equipo", de esta manera se puede tener ayuda para las consultas que se quieran realizar.

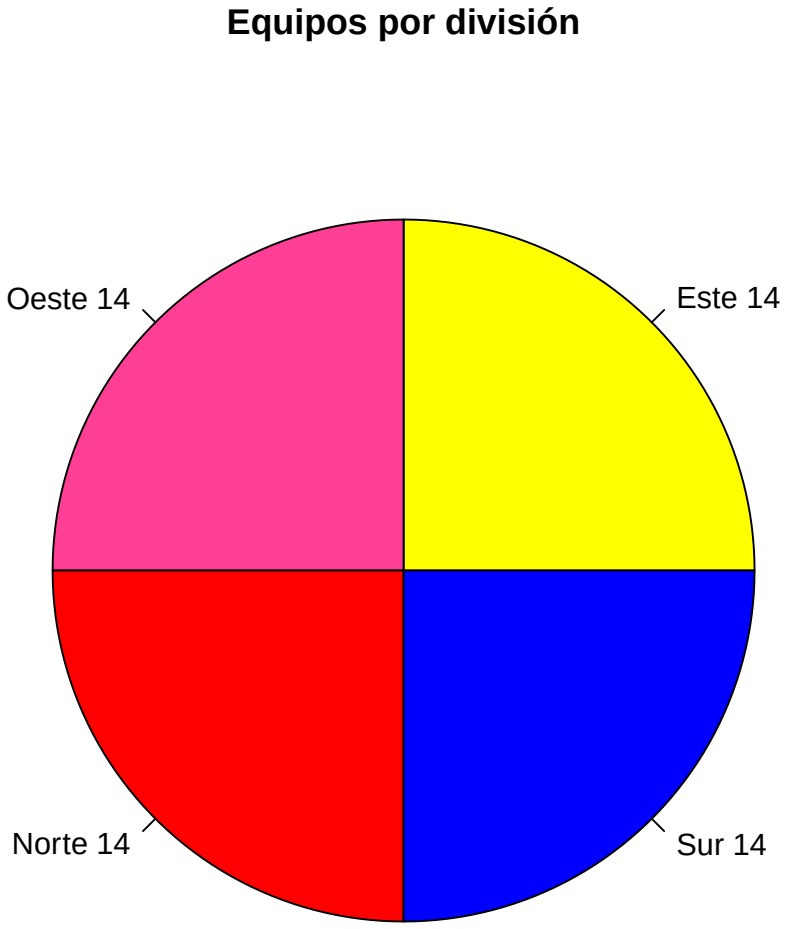
Hecha la conexión, ¿cómo se hace una consulta desde R? Se utiliza el comando *dbGetQuery*. La ventaja de la conexión es que utiliza SQL, el mismo lenguaje que se usa en PostgreSQL

Por ejemplo ¿cuántos equipos hay pertenecientes a cada división?

```
dbGetQuery(conexion,"SELECT division, COUNT(id_equipo)
                     FROM equipo
                     GROUP BY division")

##   division count
## 1     Este    14
## 2    Oeste    14
## 3    Norte    14
## 4      Sur    14
```

Se pueden guardar las consultas como cualquier otra variable y manipularse. Por ejemplo, una consulta con datos cuantitativos puede ser presentada gráficamente como se muestra a continuación :

```
#Consulta de número de equipos por división
a<- dbGetQuery(conexion,"SELECT division, COUNT(id_equipo)
                         FROM equipo
                         GROUP BY division")

library(grDevices)
division<-paste(a$division, a$count)
pie(a$count,col=c("yellow", "violetred1", "red", "blue"), labels=c(division),
    main="Equipos por división")
```



Un ejemplo de interés es conocer cuántos partidos ha ganado cada equipo, si jugaron de local o visitante, o los climas a los que pertenece ese equipo y el lugar donde jugaron. Estas consultas se pueden resolver de la siguiente manera:

```
base <- dbGetQuery(conexion,
"SELECT id_partido, tab.id_equipo, marcador, c_local, clima AS c_equipo,
         0 AS ganador, 0 AS clima_partido
  FROM (
          SELECT id_partido, id_equipo_v AS id_equipo, marcador_v AS marcador,
          0 AS c_local
          FROM partido
          UNION
          SELECT id_partido, id_equipo_l AS id_equipo, marcador_l AS marcador,
          1 AS c_local
          FROM partido
        ) AS tab JOIN equipo_ciudad ON tab.id_equipo=equipo_ciudad.id_equipo
          JOIN ciudad ON equipo_ciudad.id_ciudad=ciudad.id_ciudad
  ORDER BY id_partido")
```

Con lo anterior se guardó la consulta en la variable 'base', ¿cuál es el resultado? :

```
#Solo los primeros 20 registros
head(base,20)

##    id_partido id_equipo marcador c_local c_equipo ganador clima_partido
## 1           1        SF       16       0 Templado       0             0
## 2           1       NYG       13       1     Frio       0             0
## 3           2        SD       34       0 Caluroso       0             0
## 4           2       CIN        6       1     Frio       0             0
## 5           3       NYJ       37       0     Frio       0             0
## 6           3       BUF       31       1     Frio       0             0
## 7           4       CHI       27       1     Frio       0             0
## 8           4       MIN       23       0     Frio       0             0
## 9           5       DET       21       0     Frio       0             0
## 10          5       MIA       49       1 Caluroso       0             0
## 11          6       CAR       10       1 Templado       0             0
## 12          6       BAL        7       0     Frio       0             0
## 13          7       JAX       25       1 Caluroso       0             0
## 14          7       IND       28       0     Frio       0             0
## 15          8        GB       37       1     Frio       0             0
## 16          8       ATL       34       0 Templado       0             0
## 17          9        KC       40       0 Templado       0             0
## 18          9       CLE       39       1     Frio       0             0
## 19         10       WAS       31       1     Frio       0             0
## 20         10       ARI       23       0 Caluroso       0             0
```

A pesar de la consulta, faltará llenar las nuevas variables creadas, por lo que el paso siguiente es implementar el código que compara los marcadores sobre los partidos que tienen el mismo id y asigna un 1 al partido que tuvo el marcador más alto y asigna un cero en el caso contrario. También se asigna el clima correspondiente a cada partido.

```
#LLenado de las dos columnas finales

attach(base) #Permite utilizar las variables de la base

#Llenado de las nuevas variables:
for (j in 1: 4005)      #Recorre todos los registros
{
  aux<-0
  for (i in 1:8010  )   #Recorrido sobre la variable partido
  {
    if (id_partido[i]==j) #Se comparan los marcadores del mismo partido
    {
      if (marcador[i]>aux)
        aux<-marcador[i]
      if (c_local[i]==1)
        clima_j<-c_equipo[i] #El clima del local se le asigna al partido
    }
  }
  for (i in 1:8010  )
  {

    if (id_partido[i]==j)   #Asigna 1 al ganador del partido
    {
      base$clima_partido[i]<-clima_j
      if (marcador[i]==aux)
        base$ganador[i]<-1
    }

  }
}



##Quitando los empates (solo hay 7 de 4005 partidos donde los hubo)
h=c()
for (i in 1:4005){
    k<-base[base$id_partido==i,]
  if (k[1,3]==k[2,3]){
    h<-c(h,k[1,1])}
}
base=base[-(which(base$id_partido%in%h)),]
```

Finalmente, después de ejecutar el código, el resultado es el siguiente:

```
head(base,10)

##    id_partido id_equipo marcador c_local c_equipo ganador clima_partido
## 1           1        SF       16       0 Templado       1          Frio
## 2           1       NYG       13       1     Frio       0          Frio
## 3           2        SD       34       0 Caluroso       1          Frio
## 4           2       CIN        6       1     Frio       0          Frio
## 5           3       NYJ       37       0     Frio       1          Frio
## 6           3       BUF       31       1     Frio       0          Frio
## 7           4       CHI       27       1     Frio       1          Frio
## 8           4       MIN       23       0     Frio       0          Frio
## 9           5       DET       21       0     Frio       0      Caluroso
## 10          5       MIA       49       1 Caluroso       1      Caluroso
```

Ahora se tiene un 'pedazo' de información de toda la base de ONEFA en R y se puede trabajar con ella como se desee. Hecho esto se termina la conexión, con el propósito de dejar de consumir recursos, a través de los siguientes comandos:

```
dbDisconnect(conexion)
dbUnloadDriver(drv)
```

Finalmente, se puede guardar la base de datos en cierta ubicacioón en formato .csv para usarla posteriormente en caso de ser requerido, sin abrir nuevamente la conexión con postgreSQL, esto puede hacerse usando el comando *write.csv* de la siguiente manera:

```
write.csv(base, file="Direccion/../.../nombredelarchivo.csv" )
```

3.4. Ejercicios

- Establezca la conexión entre PostgreSQL y R.
- Genere las siguientes consultas a través de la conexión creada:
  - Ciudades y el número de equipos asociados.
  - Todos los datos de los 5 estadios con mayor capacidad.
- Replique la consulta que se explicó en la práctica y ejecute el código para obtener las dos variables nuevas y por ende la tabla llamada 'base' para usar en R.
- Termine la conexión entre PostgreSQL y R.
- A través del comando *write.csv* guarde la consulta con el nombre "equipos-partidos-climas.csv", de esta manera se sabrá que contiene la base al cargarla posteriormente para su uso.

4 | Práctica S - Introducción al análisis de correspondencias: perfiles.

4.1. Objetivo

- Introducir al alumno en uno de los conceptos base del análisis de correspondencias: *perfiles.*
- Implementación del método gráfico de perfiles en lenguaje R.
- Interpretación de los resultados gráficos.

4.2. Introducción

El análisis de correspondencias es una técnica estadística exploratoria para *variables categóricas* y se aplica principalmente a tablas de contingencia.

Una variable categórica también denominada variable cualitativa, es una variable que no puede medirse numéricamente ni en una magnitud de jerarquía. Los valores de una variable cualitativa son adjetivos o características, por ejemplo, sexo: hombre o mujer, color de cabello etc.

Una tabla de contingencia es una tabla que cuenta las observaciones (o frecuencias) de diferentes variables. Las filas y columnas de las tablas corresponden a las variables categóricas[1]. Estas tablas se utilizan generalmente para analizar la asociación entre dos o más variables y a menudo incluyen explorar los perfiles de cada fila y de cada columna, es decir, las probabilidades condicionales.

La técnica de análisis de correspondencias tiene como objetivo analizar la interrelación entre variables y mostrar, desde un punto de vista gráfico, las relaciones de dependencia o independencia entre las variables.

En el capítulo 2 se habló de las distribuciones conjuntas de variables aleatorias discretas, bien, la tabla de contingencia puede ser vista de la siguiente forma:

[1] https://goo.gl/S1CnC2

Cuadro 4.1: Tabla de contingencia

X (variable renglón)	Y (variable columna) y_1	y_2	...	y_J	Total
x_1	n_{11}	n_{12}	...	n_{1J}	$n_{1\bullet}$
x_2	n_{21}	n_{22}	...	n_{2J}	$n_{2\bullet}$
...	...	...		...	...
x_I	n_{I1}	n_{I2}	...	n_{IJ}	$n_{I\bullet}$
Total	$n_{\bullet 1}$	$n_{\bullet 2}$	...	$n_{\bullet J}$	n=$n_{\bullet\bullet}$

Perfiles

El concepto de *perfil* (un vector de frecuencias relativas) es de suma importancia, ya que la suma de sus elementos es 1 o 100 %. Es decir, este vector representa la distribución condicional dada la categoría correspondiente. De esta manera se tiene:

Perfil renglón:

$$\left(\frac{n_{i1}}{n_{i\bullet}}, \frac{n_{i2}}{n_{i\bullet}}, ..., \frac{n_{iJ}}{n_{i\bullet}}\right) \text{ para } i = 1, 2, ..., I$$

La tabla correspondiente es la siguiente:

Cuadro 4.2: Perfiles Renglón

X (variable renglón)	Y (variable columna) y_1	y_2	...	y_J	Total
x_1	$\frac{n_{11}}{n_{1\bullet}}$	$\frac{n_{12}}{n_{1\bullet}}$	...	$\frac{n_{1J}}{n_{1\bullet}}$	1
x_2	$\frac{n_{21}}{n_{2\bullet}}$	$\frac{n_{22}}{n_{2\bullet}}$	...	$\frac{n_{23}}{n_{2\bullet}}$	1
...	...	...		...	...
x_I	$\frac{n_{I1}}{n_{I\bullet}}$	$\frac{n_{I2}}{n_{I\bullet}}$	...	$\frac{n_{IJ}}{n_{I\bullet}}$	1

Perfil columna

$$\left(\frac{n_{1j}}{n_{\bullet j}}, \frac{n_{2j}}{n_{\bullet j}}, ..., \frac{n_{IJ}}{n_{\bullet j}}\right) \text{ con } j = 1, 2, ..., J$$

La tabla correspondiente es la siguiente:

Cuadro 4.3: Perfiles Columna

X (variable renglón)	Y (variable columna) y_1	y_2	...	y_J
x_1	$\frac{n_{11}}{n_{\bullet 1}}$	$\frac{n_{12}}{n_{\bullet 2}}$	...	$\frac{n_{1J}}{n_{\bullet J}}$
x_2	$\frac{n_{21}}{n_{\bullet 1}}$	$\frac{n_{22}}{n_{\bullet 2}}$	...	$\frac{n_{23}}{n_{\bullet J}}$
...	...	...		...
x_I	$\frac{n_{I1}}{n_{\bullet 1}}$	$\frac{n_{I2}}{n_{\bullet 2}}$	...	$\frac{n_{IJ}}{n_{\bullet J}}$
Total	1	1		1

Las gráficas de ambas tablas sirven para ver la similitud entre los perfiles al comparar sus categorías.

4.3. Perfiles en R

```
base<-read.csv(file=""Direccion/../equipos-partidos-climas.csv"", header=TRUE)
base<-base[,-4]
```

Para los propósitos de esta práctica se utilizará el archivo generado con la consulta de la práctica R (en la misma se encuentra el código para obtenerla), este conjunto de datos contiene el id de cada partido, los equipos que lo jugaron así como su marcador, si el equipo resultó con un marcador mayor en un partido, se le asignó un 1 en la variable ganador. Por otra parte, se tienen dos columnas respecto al clima, una de ellas es sobre el clima de la cuidad a la que pertenece el equipo (clima_equipo) y la otra el clima de la ciudad donde se jugó el partido (clima_partido). A continuación, se muestran los primeros 28 registros:

```
head(base,24)
```

```
##     X id_partido id_equipo c_local c_equipo ganador clima_partido
## 1   1          1        SF       0 Templado       1          Frio
## 2   2          1       NYG       1     Frio       0          Frio
## 3   3          2        SD       0 Caluroso       1          Frio
## 4   4          2       CIN       1     Frio       0          Frio
## 5   5          3       NYJ       0     Frio       1          Frio
## 6   6          3       BUF       1     Frio       0          Frio
## 7   7          4       CHI       1     Frio       1          Frio
## 8   8          4       MIN       0     Frio       0          Frio
## 9   9          5       DET       0     Frio       0      Caluroso
## 10 10          5       MIA       1 Caluroso       1      Caluroso
## 11 11          6       CAR       1 Templado       1      Templado
## 12 12          6       BAL       0     Frio       0      Templado
## 13 13          7       JAX       1 Caluroso       0      Caluroso
## 14 14          7       IND       0     Frio       1      Caluroso
## 15 15          8        GB       1     Frio       1          Frio
## 16 16          8       ATL       0 Templado       0          Frio
## 17 17          9        KC       0 Templado       1          Frio
## 18 18          9       CLE       1     Frio       0          Frio
## 19 19         10       WAS       1     Frio       1          Frio
## 20 20         10       ARI       0 Caluroso       0          Frio
## 21 21         11       TEN       1 Templado       1      Templado
## 22 22         11       PHI       0     Frio       0      Templado
## 23 23         12        NO       0 Templado       1      Caluroso
## 24 24         12        TB       1 Caluroso       0      Caluroso
```

El objetivo es analizar si hay relación entre los partidos ganados y el clima del lugar donde se jugaron, por lo tanto nos importan las variables: id_partido, ganador y clima_partido.

Se empezará analizando los **partidos ganados** de todos los equipos por el clima del lugar donde se jugaron los partidos.

```
BaseG<-base[base$ganador==1,]     #Hacemos un filtro para partidos ganados.
tablaE<-table(BaseG$id_equipo,BaseG$clima_partido);tablaE

##
##       Caluroso Frio Templado
##   ARI       80   17       19
##   ATL       13   24       97
##   BAL       12  119       14
##   BUF       11   80       10
##   CAR       19   19       96
##   CHI       10   95       12
##   CIN        7  101       12
##   CLE        3   60       13
##   DAL       78   40       15
##   DEN       16  109       27
##   DET        6   72        7
##   GB        10  135       16
##   HOU       13   17       79
##   IND       15  124       33
##   JAX       61   21       12
##   KC         9   20       91
##   MIA       67   31       11
##   MIN        4  101       15
##   NE        18  175       14
##   NO        18   20       97
##   NYG       16  103       16
##   NYJ       17   92       10
##   OAK       10   14       64
##   PHI       14  115       15
##   PIT       13  144       10
##   SD        82   27       25
##   SEA       12  115       23
##   SF        11   20       82
##   STL       12   16       60
##   TB        59   21       25
##   TEN       12   22       81
##   WAS        9   80       11
```

La tabla anterior es una tabla de contingencia de frecuencias absolutas, da el número de partidos ganados por equipo y clima. Dice que, tomando un equipo en particular, en 10 temporadas (ya que es la cantidad de datos que tenemos en nuestra base relacional), Dalas (DAL) tuvo 78 partidos ganados en clima caluroso, 40 en frio y 15 en templado.

```
tablaE["DAL",] #Solo Dallas

## Caluroso     Frio Templado
##       78       40       15
```

Aunque ya se tiene la tabla de contingencia aún no se han obtenido las probabilidades. El comando *prop.table* permite obtener las proporciones de toda la tabla. Sin embargo, eso nos daría la distribución conjunta. ¿Eso es lo qué queremos? La respuesta es no, lo que es de interés es conocer los perfiles de los equipos y sus partidos ganados respecto al clima donde los jugaron. La variable equipo (id_equipo) es nuestra variable renglón, por lo tanto, en la función se indicará que se desean obtener las proporciones por renglón (1), si se deseara obtener los perfiles columna se especificaría el segundo parámetro como 2.

```
PR<-prop.table(tablaE,1); PR #Perfiles renglon

##
##         Caluroso       Frio   Templado
##   ARI 0.68965517 0.14655172 0.16379310
##   ATL 0.09701493 0.17910448 0.72388060
##   BAL 0.08275862 0.82068966 0.09655172
##   BUF 0.10891089 0.79207921 0.09900990
##   CAR 0.14179104 0.14179104 0.71641791
##   CHI 0.08547009 0.81196581 0.10256410
##   CIN 0.05833333 0.84166667 0.10000000
##   CLE 0.03947368 0.78947368 0.17105263
##   DAL 0.58646617 0.30075188 0.11278195
##   DEN 0.10526316 0.71710526 0.17763158
##   DET 0.07058824 0.84705882 0.08235294
##   GB  0.06211180 0.83850932 0.09937888
##   HOU 0.11926606 0.15596330 0.72477064
##   IND 0.08720930 0.72093023 0.19186047
##   JAX 0.64893617 0.22340426 0.12765957
##   KC  0.07500000 0.16666667 0.75833333
##   MIA 0.61467890 0.28440367 0.10091743
##   MIN 0.03333333 0.84166667 0.12500000
##   NE  0.08695652 0.84541063 0.06763285
##   NO  0.13333333 0.14814815 0.71851852
##   NYG 0.11851852 0.76296296 0.11851852
##   NYJ 0.14285714 0.77310924 0.08403361
##   OAK 0.11363636 0.15909091 0.72727273
##   PHI 0.09722222 0.79861111 0.10416667
##   PIT 0.07784431 0.86227545 0.05988024
##   SD  0.61194030 0.20149254 0.18656716
##   SEA 0.08000000 0.76666667 0.15333333
##   SF  0.09734513 0.17699115 0.72566372
##   STL 0.13636364 0.18181818 0.68181818
##   TB  0.56190476 0.20000000 0.23809524
##   TEN 0.10434783 0.19130435 0.70434783
##   WAS 0.09000000 0.80000000 0.11000000
```

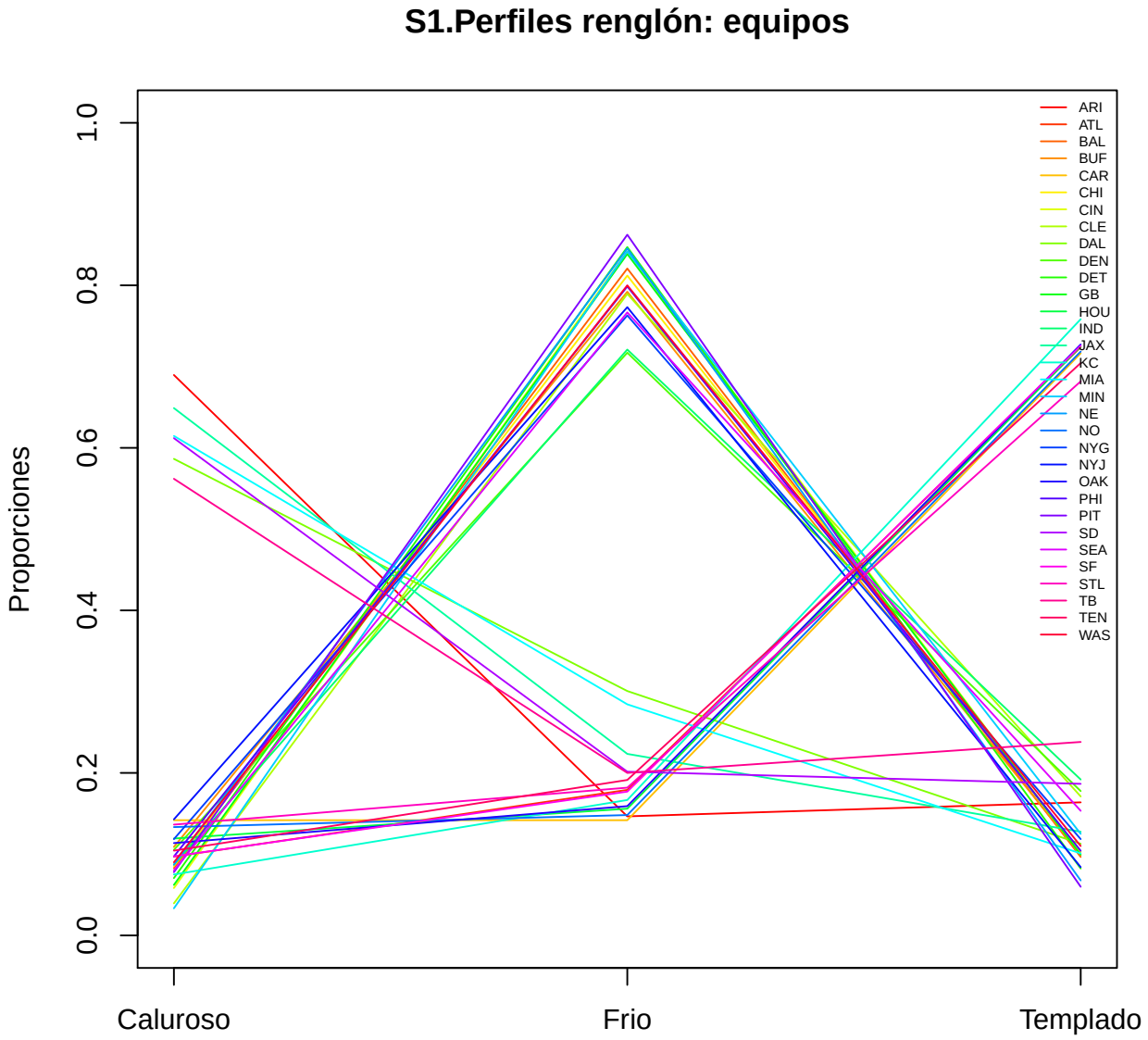
Si se tuviera duda a cerca de las proporciones, es posible corroborar la distribución condicionada por renglón con el siguiente comando que se encuentra en la biblioteca de 'vcd', el comando da la suma de renglones y columnas:

```
library(vcd)

mar_table(PR)

##           Caluroso       Frio   Templado TOTAL
##   ARI   0.68965517  0.1465517 0.16379310     1
##   ATL   0.09701493  0.1791045 0.72388060     1
##   BAL   0.08275862  0.8206897 0.09655172     1
##   BUF   0.10891089  0.7920792 0.09900990     1
##   CAR   0.14179104  0.1417910 0.71641791     1
##   CHI   0.08547009  0.8119658 0.10256410     1
##   CIN   0.05833333  0.8416667 0.10000000     1
##   CLE   0.03947368  0.7894737 0.17105263     1
##   DAL   0.58646617  0.3007519 0.11278195     1
##   DEN   0.10526316  0.7171053 0.17763158     1
##   DET   0.07058824  0.8470588 0.08235294     1
##   GB    0.06211180  0.8385093 0.09937888     1
##   HOU   0.11926606  0.1559633 0.72477064     1
##   IND   0.08720930  0.7209302 0.19186047     1
##   JAX   0.64893617  0.2234043 0.12765957     1
##   KC    0.07500000  0.1666667 0.75833333     1
##   MIA   0.61467890  0.2844037 0.10091743     1
##   MIN   0.03333333  0.8416667 0.12500000     1
##   NE    0.08695652  0.8454106 0.06763285     1
##   NO    0.13333333  0.1481481 0.71851852     1
##   NYG   0.11851852  0.7629630 0.11851852     1
##   NYJ   0.14285714  0.7731092 0.08403361     1
##   OAK   0.11363636  0.1590909 0.72727273     1
##   PHI   0.09722222  0.7986111 0.10416667     1
##   PIT   0.07784431  0.8622754 0.05988024     1
##   SD    0.61194030  0.2014925 0.18656716     1
##   SEA   0.08000000  0.7666667 0.15333333     1
##   SF    0.09734513  0.1769912 0.72566372     1
##   STL   0.13636364  0.1818182 0.68181818     1
##   TB    0.56190476  0.2000000 0.23809524     1
##   TEN   0.10434783  0.1913043 0.70434783     1
##   WAS   0.09000000  0.8000000 0.11000000     1
##   TOTAL 6.15853095 16.4876637 9.35380537    32

PR["DAL",] #Solo pido las proporciones de Dallas.

##  Caluroso      Frio  Templado
## 0.5864662 0.3007519 0.1127820
```

Entonces, la interpretación para Dallas es: el 58.64 % de los partidos ganados fueron en clima caluroso, el 30.07 % los ganó en frío y el 11.27 % en clima templado.

Una vez obtenida la distribución condicionada y con ella los perfiles renglón, se grafica la matriz transpuesta:

```
matplot(t(PR),type="l",ylab="Proporciones",ylim=c(0,1),
        #Tipo de línea, límites y nombre del eje Y
        lty=1,col=rainbow(32),
        #Cuántos tipos de líneas y numero de colores en arcoiris.
        xaxt="n", #Se quitan las categorías por default
        main="S1.Perfiles renglón: equipos") #Título
axis(1,at=1:length(colnames(PR)),labels=colnames(tablaE))
#Se agrega las categorías de la tabla PR, tantas como haya
legend("topright",rownames(PR),lty=1,col=rainbow(32),bty="n",border="white",
cex=.5)#Se agregan las leyendas con los mismos colores y tipos de línea.
```



Notemos que el perfil de cada uno de los 32 equipos fue graficado. Treinta y dos perfiles es demasiada información para una sola gráfica, sin embargo, es claro que hay equipos con perfiles semejantes.

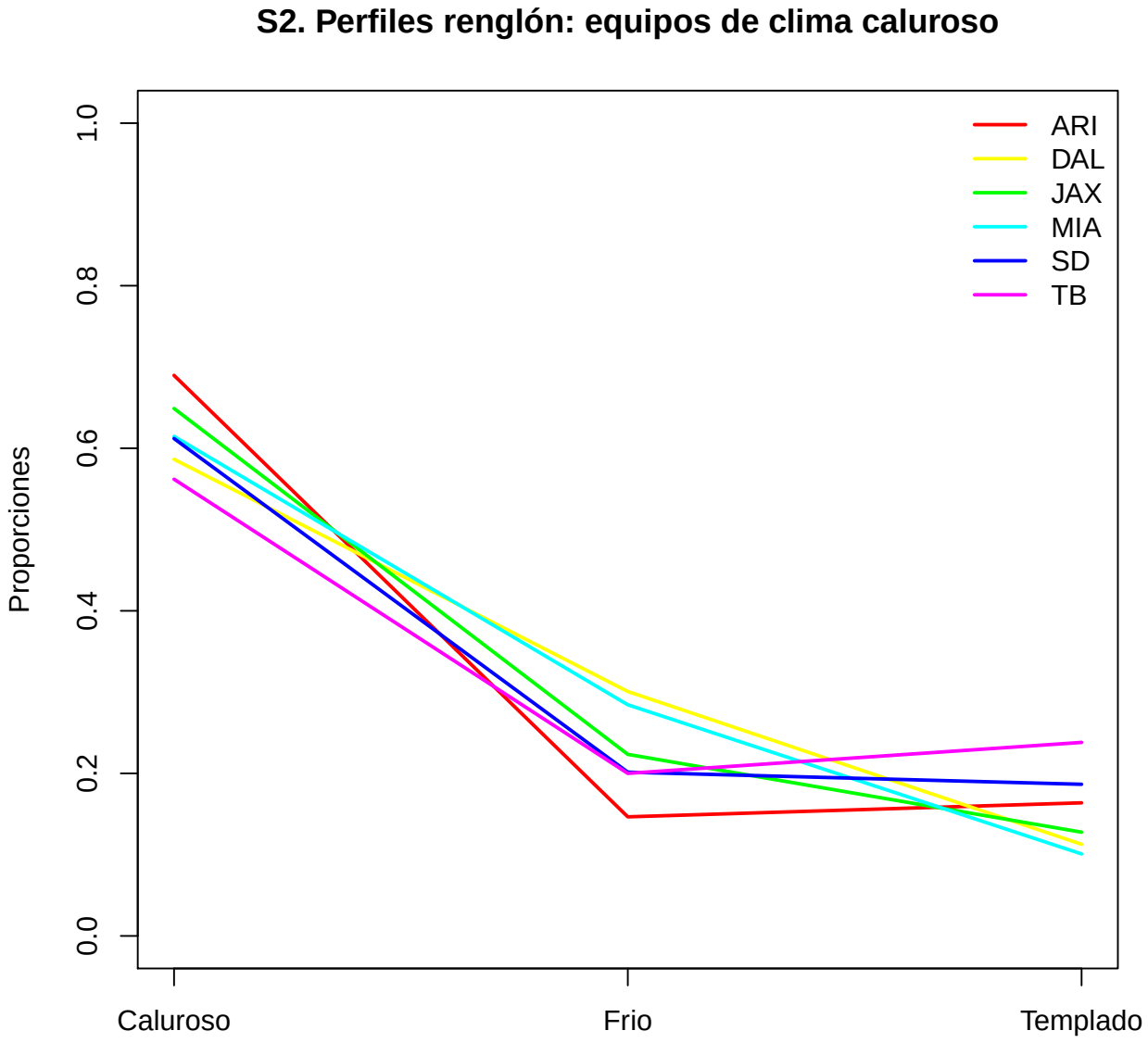
¿Qué variable definirá el tipo de perfil? ¿Cuál es una variable intuitiva para este problema? La respuesta es el clima al que el equipo pertenece (c_equipo), por lo tanto se hacen 3 filtros

diferentes y se utiliza la misma técnica:

```
#Filtros: Partidos ganados y tipo de clima de los equipos.
baseCalor<-base[base$c_equipo=="Caluroso" & base$ganador==1,]

baseTemplado<-base[base$c_equipo=="Templado"& base$ganador==1,]

baseFrio<-base[base$c_equipo=="Frio" & base$ganador==1,]
```

Perfiles para equipos de clima cálido.

```
tablaC<-table(as.vector(baseCalor$id_equipo),as.vector(baseCalor$clima_partido))
tablaC

##
##       Caluroso Frio Templado
##   ARI       80   17       19
##   DAL       78   40       15
##   JAX       61   21       12
##   MIA       67   31       11
##   SD        82   27       25
##   TB        59   21       25

PRC<-prop.table(tablaC,1);PRC #Perfil renglón equipos clima caluroso.

##
##        Caluroso      Frio  Templado
##   ARI 0.6896552 0.1465517 0.1637931
##   DAL 0.5864662 0.3007519 0.1127820
##   JAX 0.6489362 0.2234043 0.1276596
##   MIA 0.6146789 0.2844037 0.1009174
##   SD  0.6119403 0.2014925 0.1865672
##   TB  0.5619048 0.2000000 0.2380952

matplot(t(PRC),type="l",ylab="Proporciones",ylim=c(0,1),
        #Tipo línea, límites y nombre del eje Y
        lty=1, col=rainbow(6),
        #cuántos tipos de líneas y número de colores en arcoiris
        xaxt="n",
        #Quitamos las categorias que pone la gráfica por default
        lwd=2,
        #Grosor, si no se pone por default es 1
        main="S2. Perfiles renglón: equipos de clima caluroso")

axis(1,at=1:length(colnames(tablaC)),labels=colnames(tablaC))
#Agrego las categorias de la tabla, tantas como hay
legend("topright",rownames(tablaC),lty=1,col=rainbow(6),bty="n",lwd=2)
```

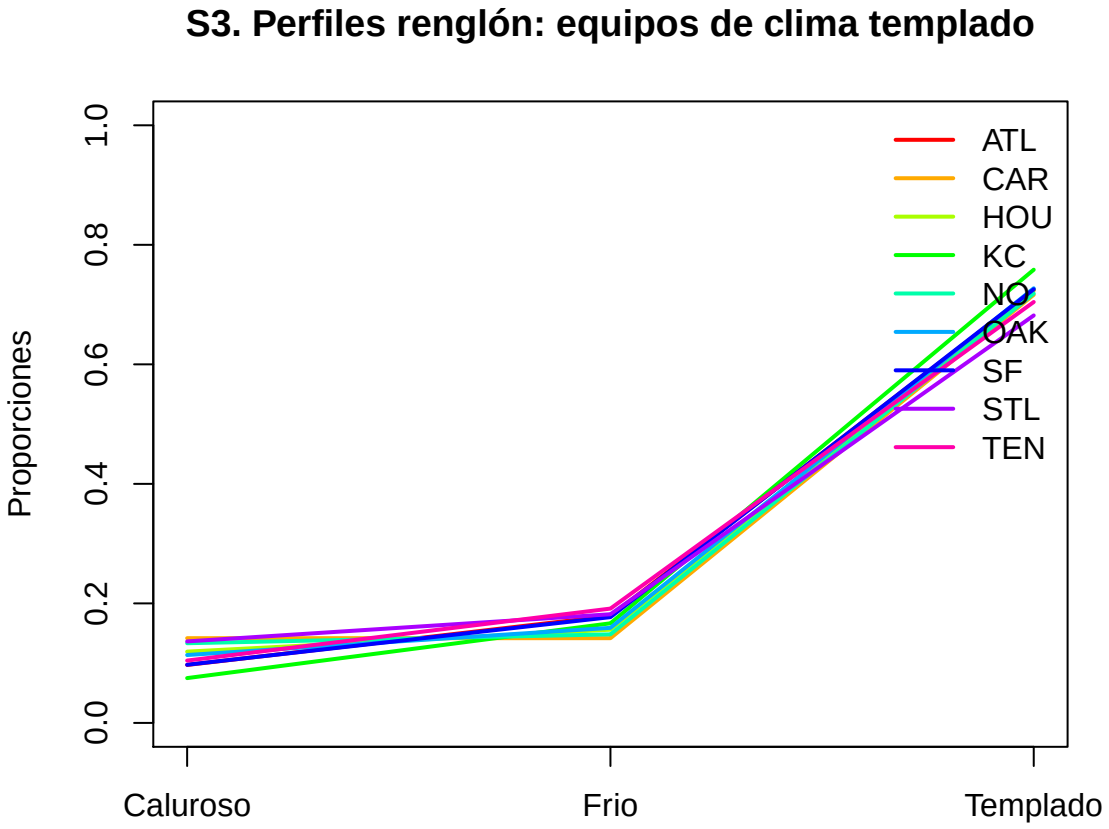


¿Cómo se interpreta la gráfica anterior? No se debe olvidar lo que representan las frecuencias, por lo tanto: los *partidos ganados* de los equipos que pertenecen a un clima cálido fueron jugados en mayoría (alrededor del 60 %) en su mismo clima, mientras que el resto de los partidos ganados en clima diferente están entre el 20 % y 40 %.

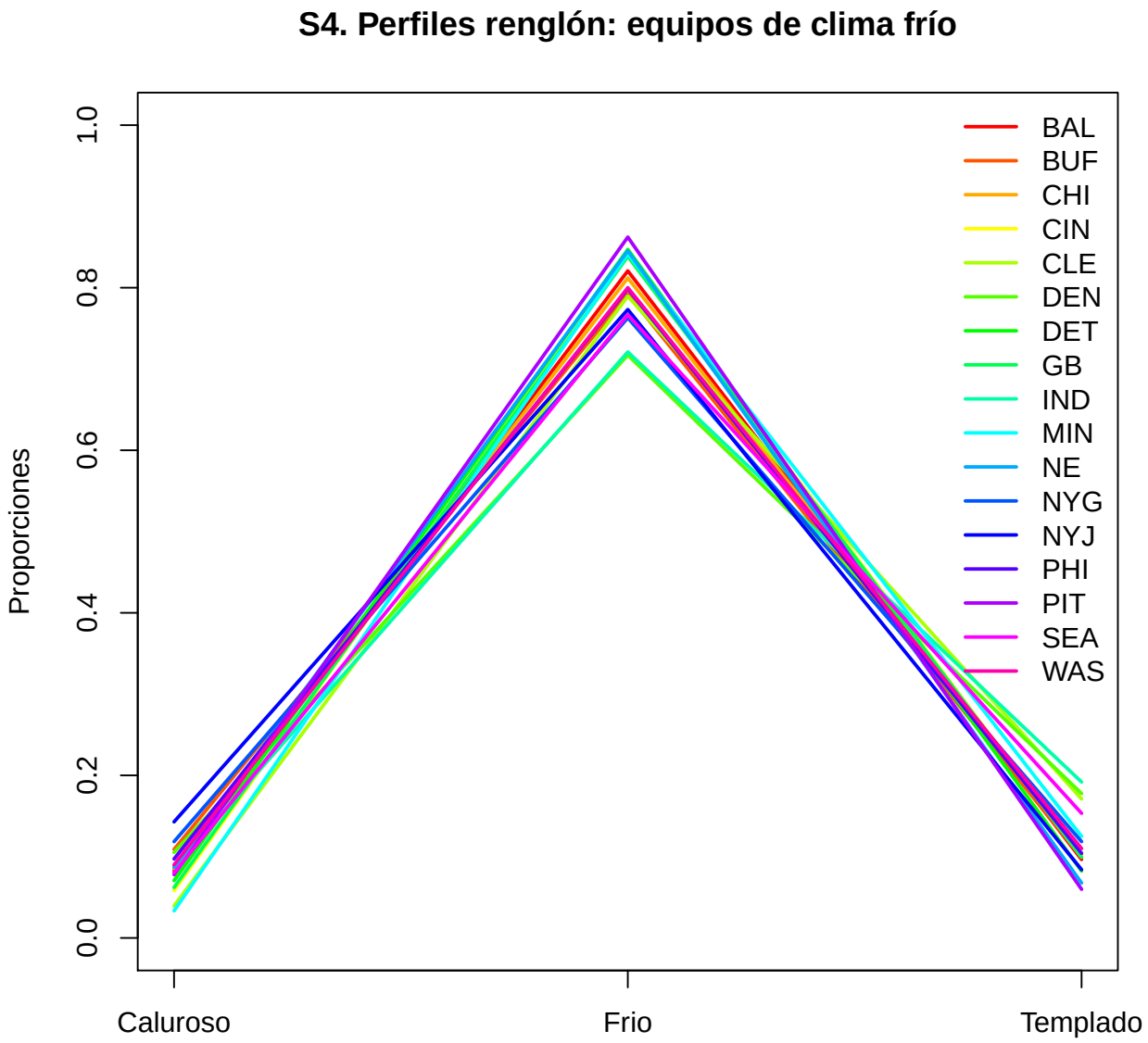
Análogamente al código anterior se tienen las siguientes gráficas para los equipos de climas templado y frío.

Perfiles para equipos de clima templado.

```
##
##       Caluroso Frio Templado
##   ATL       13   24       97
##   CAR       19   19       96
##   HOU       13   17       79
##   KC         9   20       91
##   NO        18   20       97
##   OAK       10   14       64
##   SF        11   20       82
##   STL       12   16       60
##   TEN       12   22       81
##
##         Caluroso       Frio   Templado
##   ATL 0.09701493 0.17910448 0.72388060
##   CAR 0.14179104 0.14179104 0.71641791
##   HOU 0.11926606 0.15596330 0.72477064
##   KC  0.07500000 0.16666667 0.75833333
##   NO  0.13333333 0.14814815 0.71851852
##   OAK 0.11363636 0.15909091 0.72727273
##   SF  0.09734513 0.17699115 0.72566372
##   STL 0.13636364 0.18181818 0.68181818
##   TEN 0.10434783 0.19130435 0.70434783
```



Por lo tanto, los partidos ganados de equipos de clima templado fueron jugados casi en 80 % en su mismo clima, mientras que al rededor del 20 % fueron en clima frío y menor proporción en clima cálido.



4.4. Ejercicios

- Interprete la gráfica S4.
- Repita el análisis para partidos perdidos condicionado por tipo de clima.
- Interprete las 3 gráficas.

5 | Práctica T - Dependencia y distribuciones condicionales.

5.1. Objetivos

- Comprender la importancia del concepto de dependencia e independencia entre variables aleatorias.
- Aprender la importancia del uso de gráficas con distribuciones condicionales para la toma de decisiones.
- Uso e interpretación de la prueba Ji-cuadrada para independencia de variables.
- Empleo de código en lenguaje R para creación de tablas, distribuciones condicionales y prueba de hipótesis.

5.2. Introducción

La utilización de tablas de contingencia y distribuciones condicionales son técnicas exploratorias de bases de datos, principalmente para variables de tipo categórico. La implementación de estas tablas permite usar distintos tipos de pruebas de hipótesis o crear distribuciones condicionales para descubrir o confirmar asociaciones de dependencia o independencia entre variables.

El concepto de independencia es de vital importancia en la teoría de la probabilidad. Informalmente se dice que *dos eventos son independientes* si el hecho de que uno ocurra no afecta la probabilidad de la ocurrencia del otro.
La prueba χ^2 es utilizada comúnmente en tablas de contingencia y contrasta si dos variables discretas son independientes o no. La idea de esta prueba es rechazar la hipótesis nula si los valores reales difieren mucho de los valores esperados.

Cuadro 5.1: Tabla de contingencia

	Y (variable columna)				
X (variable renglón)	y_1	y_2	...	y_J	Total
x_1	n_{11}	n_{12}	...	n_{1J}	$n_{1\bullet}$
x_2	n_{21}	n_{22}	...	n_{2J}	$n_{2\bullet}$
...	...	...		...	...
x_I	n_{I1}	n_{I2}	...	n_{IJ}	$n_{I\bullet}$
Total	$n_{\bullet 1}$	$n_{\bullet 2}$	...	$n_{\bullet J}$	n=$n_{\bullet\bullet}$

Las hipótesis de esta prueba son:

$$H_0\textbf{(Hipótesis nula)} = \textit{Las variables son independientes.}$$
$$H_a\textbf{(Hipótesis alternativa)} = \textit{Las variables no son independientes.}$$

Y el estadístico de prueba que se utiliza es el siguiente:

$$\chi^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}};\ E_{ij} = \frac{n_{i\bullet} n_{\bullet j}}{n_{\bullet\bullet}}$$

Donde:
E_i=Valores esperados.
O_i=Valores observados o reales.

Los grados de libertad para la prueba son gl=(I-1)(J-1). Para decidir se utiliza el p - *value*, a un nivel de significativa estándar, usualmente del 5 % y cuando $p - value \leq .05$ se rechaza la hipótesis nula y se toma la alternativa, en caso contrario se acepta la hipótesis nula.

Es importante recalcar que esta prueba contrasta la independencia, no es una medida de fuerza de asociación. Adicionalmente, para que esta prueba sea correcta, las frecuencias esperadas deben ser mayores que 5, aunque en la práctica se permite un 20 % de frecuencias por debajo de 5. Finalmente cuando el valor de n $=n_{\bullet\bullet}$ es demasiado grande, la prueba puede dar resultados de asociación incluso cuando posiblemente la asociación no es significativa.

Finalmente es importante comprender la importancia de la asociación entre variables, ya que, aunque que se pueden hacer supuestos sobre el comportamiento de una de ellas, éste puede mostrarse afectado al verse involucrada una segunda o tercera variable.

5.3. Distribuciones condicionales y dependencia

En esta práctica se volverá a trabajar con el conjunto de datos obtenido en la práctica R, dado que ya se tenía guardada en un archivo *.csv* se facilitará la carga tras obtenerla con la conexión de R y PostgreSQL. Esta base contiene el id de cada partido, los equipos que lo jugaron, así como su marcador, si el equipo resultó con un marcador mayor en un partido, se le asignó un 1 en la variable ganador. Por otra parte, se tienen dos columnas respecto al clima, una de ellas es sobre el clima del equipo y la otra el clima donde se jugó el partido. A continuación, se muestran los primeros 20 registros.

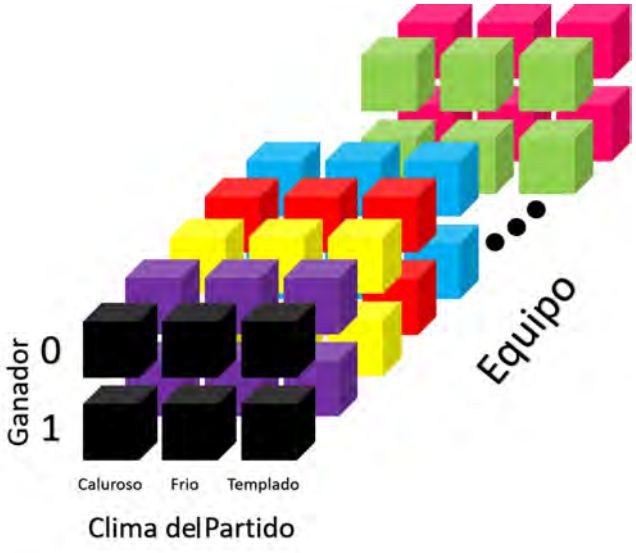
```
base<-read.csv("Direccion/../equipos-partidos-climas.csv")
```

```
##    X id_partido id_equipo marcador c_local c_equipo ganador clima_partido
## 1  1          1        SF       16       0 Templado       1          Frio
## 2  2          1       NYG       13       1     Frio       0          Frio
## 3  3          2        SD       34       0 Caluroso       1          Frio
## 4  4          2       CIN        6       1     Frio       0          Frio
## 5  5          3       NYJ       37       0     Frio       1          Frio
## 6  6          3       BUF       31       1     Frio       0          Frio
## 7  7          4       CHI       27       1     Frio       1          Frio
## 8  8          4       MIN       23       0     Frio       0          Frio
## 9  9          5       DET       21       0     Frio       0      Caluroso
```

```
##  10 10          5       MIA       49       1 Caluroso       1      Caluroso
##  11 11          6       CAR       10       1 Templado       1      Templado
##  12 12          6       BAL        7       0      Frio       0      Templado
##  13 13          7       JAX       25       1 Caluroso       0      Caluroso
##  14 14          7       IND       28       0      Frio       1      Caluroso
##  15 15          8        GB       37       1      Frio       1          Frio
##  16 16          8       ATL       34       0 Templado       0          Frio
##  17 17          9        KC       40       0 Templado       1          Frio
##  18 18          9       CLE       39       1      Frio       0          Frio
##  19 19         10       WAS       31       1      Frio       1          Frio
##  20 20         10       ARI       23       0 Caluroso       0          Frio
```

En la práctica S se analizaron los perfiles y distribuciones condicionales, sin embargo, sólo se analizaron los casos de partidos ganados y perdidos por separado. En esta ocasión interesa conocer alguna forma de tomar decisiones. ¿Qué pasaría si se pudiera tener acceso a esta base y se planea apostar? ¿Qué decisión se tomaría en un partido? ¿Con base en qué?

Vamos a utilizar una tabla de contingencia de 3 vías, que en la práctica puede visualizarse como un cubo y analizaremos información por 'rebanadas', es decir, condicionando primero por equipo y luego por la variable de interés: el clima.



También se puede omitir alguna de las 3 variables y analizar el comportamiento de dos de ellas. Por ejemplo, puede interesarnos simplemente ¿Cuántos partidos ganó cada equipo? ¿Cuál es su proporción de partidos perdidos? Este tipo de preguntas se pueden resolver con ayuda de la siguiente tabla:

```
gvsp<-table(base$ganador,base$id_equipo);gvsp
```

```
##
##     ARI ATL BAL BUF CAR CHI CIN CLE DAL DEN DET  GB HOU IND JAX  KC MIA
##   0 132 117 112 139 119 129 124 165 115 102 158 100 138  91 149 127 133
##   1 116 134 145 101 134 117 120  76 133 152  85 161 109 172  94 120 109
##
##     MIN  NE  NO NYG NYJ OAK PHI PIT  SD SEA  SF STL  TB TEN WAS
##   0 126  64 115 118 132 156 110  94 116 112 136 154 140 131 144
##   1 120 207 135 135 119  88 144 167 134 150 113  88 105 115 100
##
```

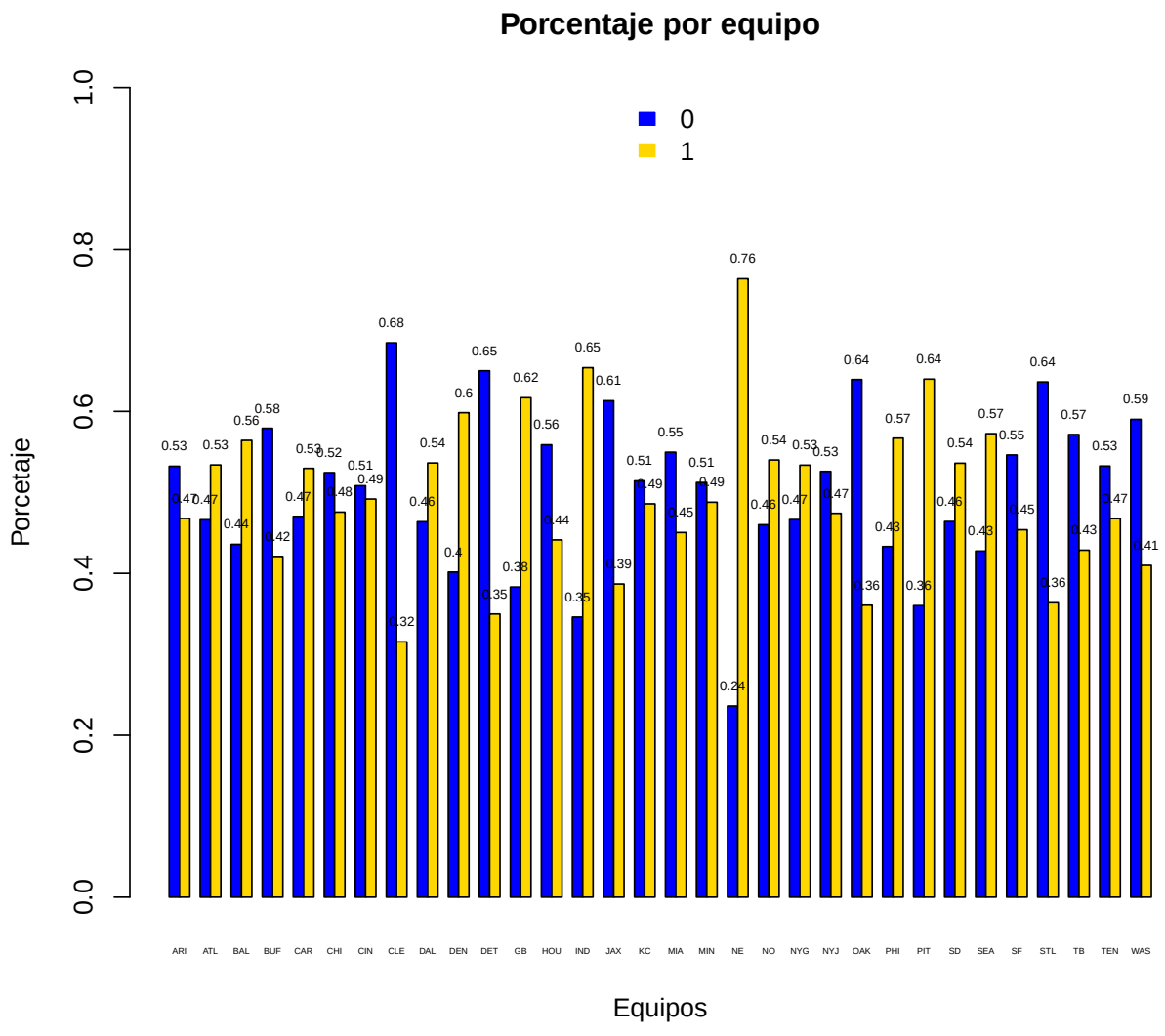
La tabla anterior nos permite ver las frecuencias absolutas, pero aún no tenemos las proporciones por cada equipo. Se condiciona utilizando el comando *prop.table* y se elige condicionar por columna (2 en el segundo parámetro de la función).

Interpretando un caso en particular, Dallas perdió 115 partidos y ganó un total de 133.

```
Condicional_gvsp<-prop.table(gvsp,2); Condicional_gvsp
```

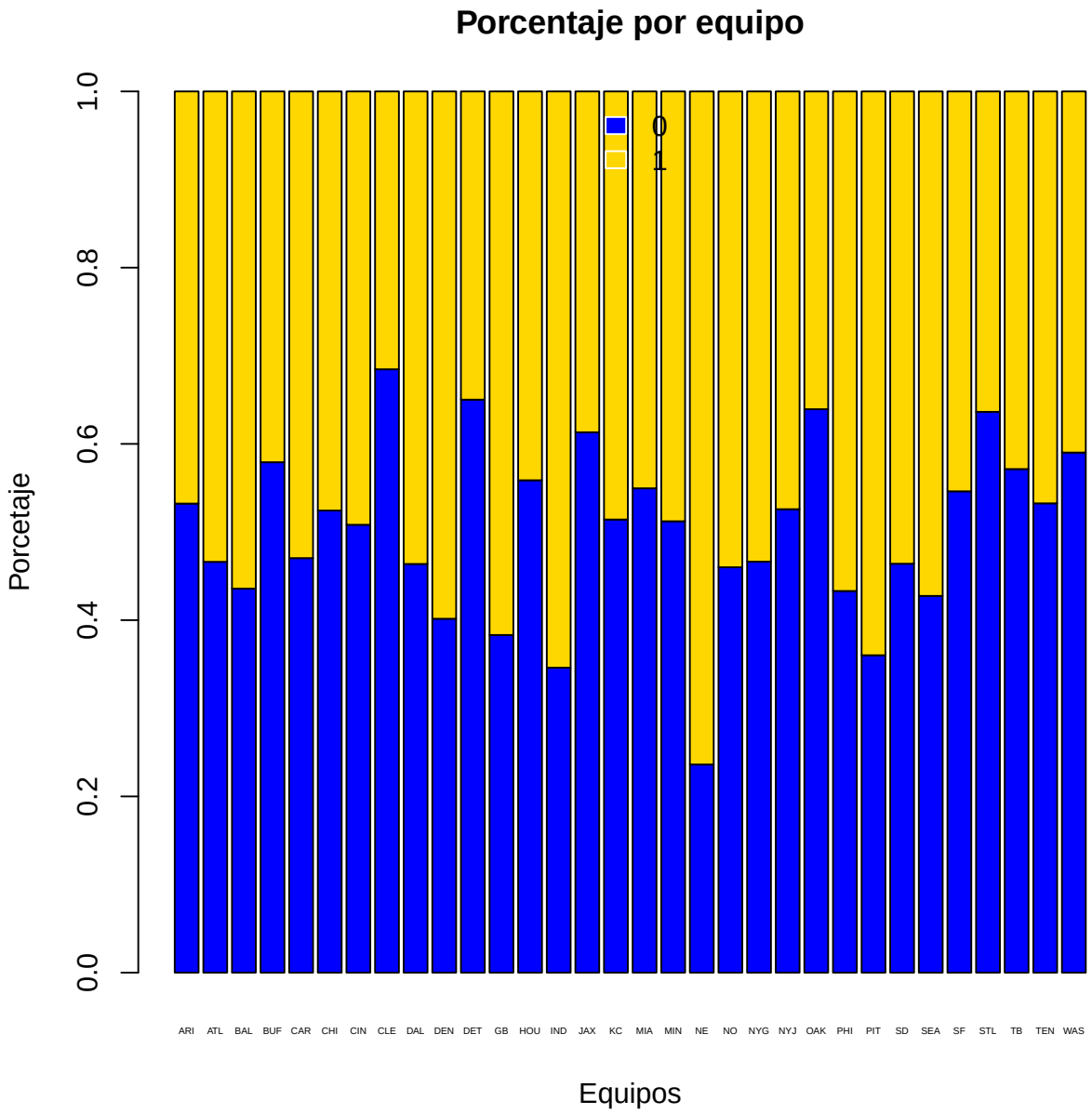
```
##
##           ARI       ATL       BAL       BUF       CAR       CHI       CIN
##   0 0.5322581 0.4661355 0.4357977 0.5791667 0.4703557 0.5243902 0.5081967
##   1 0.4677419 0.5338645 0.5642023 0.4208333 0.5296443 0.4756098 0.4918033
##
##           CLE       DAL       DEN       DET        GB       HOU       IND
##   0 0.6846473 0.4637097 0.4015748 0.6502058 0.3831418 0.5587045 0.3460076
##   1 0.3153527 0.5362903 0.5984252 0.3497942 0.6168582 0.4412955 0.6539924
##
##           JAX        KC       MIA       MIN        NE        NO       NYG
##   0 0.6131687 0.5141700 0.5495868 0.5121951 0.2361624 0.4600000 0.4664032
##   1 0.3868313 0.4858300 0.4504132 0.4878049 0.7638376 0.5400000 0.5335968
##
##           NYJ       OAK       PHI       PIT        SD       SEA        SF
##   0 0.5258964 0.6393443 0.4330709 0.3601533 0.4640000 0.4274809 0.5461847
##   1 0.4741036 0.3606557 0.5669291 0.6398467 0.5360000 0.5725191 0.4538153
##
##           STL        TB       TEN       WAS
##   0 0.6363636 0.5714286 0.5325203 0.5901639
##   1 0.3636364 0.4285714 0.4674797 0.4098361
##
```

```
### GRAFICAMOS, nombre de la tabla, titulo, colores
#Se indica que se quieren las barras una al lado de la otra,
#Nombre de los ejes, límites y tamaño de la letra.

b<-barplot(Condicional_gvsp,main="Porcentaje por equipo",
           col=c("blue","gold"),
           beside=TRUE,
           ylab = "Porcetaje",
           xlab="Equipos",
           ylim=c(0,1),
           cex.names = .3)

legend("top", rownames(Condicional_gvsp),
       fill=c("blue","gold"),bty="n",
       border="white", cex=1)

#Nombre de las leyendas, localización, colores y tamaño
text(x=b,y=Condicional_gvsp,
     labels=c(round(Condicional_gvsp,2)),
     pos=3,col="black",cex=.5)
```



Particularmente para Dallas, la gráfica muestra que ha perdido el 46.37 % de sus partidos y ha ganado el 53.62 %.

Una ventaja de graficar porcentajes o probabilidades condicionales es que por cada equipo se espera completar el 100 % lo que hace de la siguiente gráfica más útil visualmente para comparar las proporciones. Para obtenerla se cambia *beside=FALSE* en el código anterior.



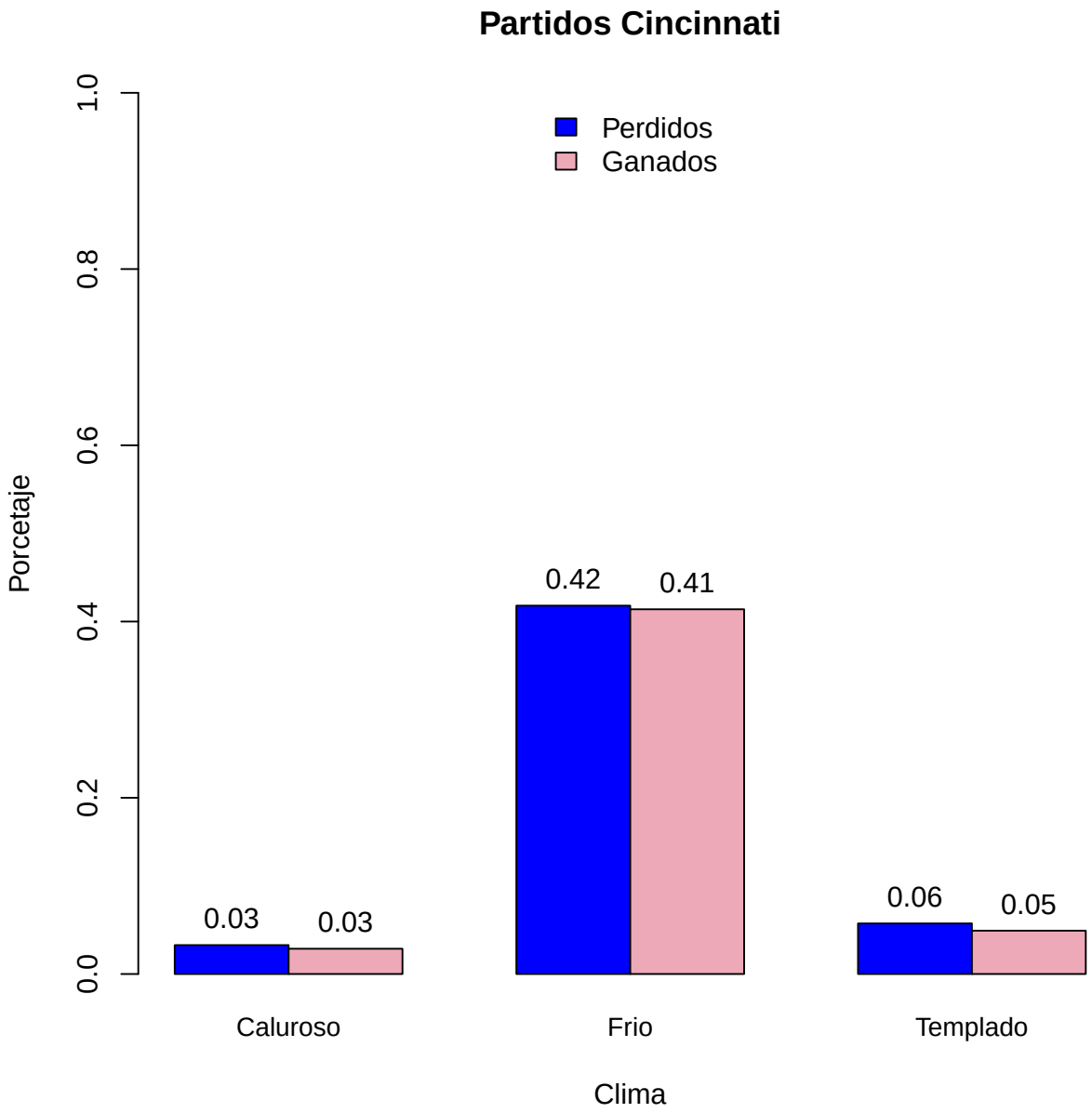
¿Cuál es el equipo que más gana? ¿cuál es el equipo que más pierde? Parece que los equipos ganan similarmente, sin embargo, hay un equipo que sobresale de los demás al ganar: New England (NE), un equipo que tiende a perder: Cleveland (CLE) y un ejemplo de equipo que parece que gana y pierde por igual: Cincinnati (CIN). A continuación, se prosigue a contestar la pregunta ¿El clima influirá en ellos?

Hagamos un análisis del cubo de datos por equipo:

```
##Cubo de Datos (incluyendo clima)
cd<-table(base$ganador, base$clima_partido,base$id_equipo)
```

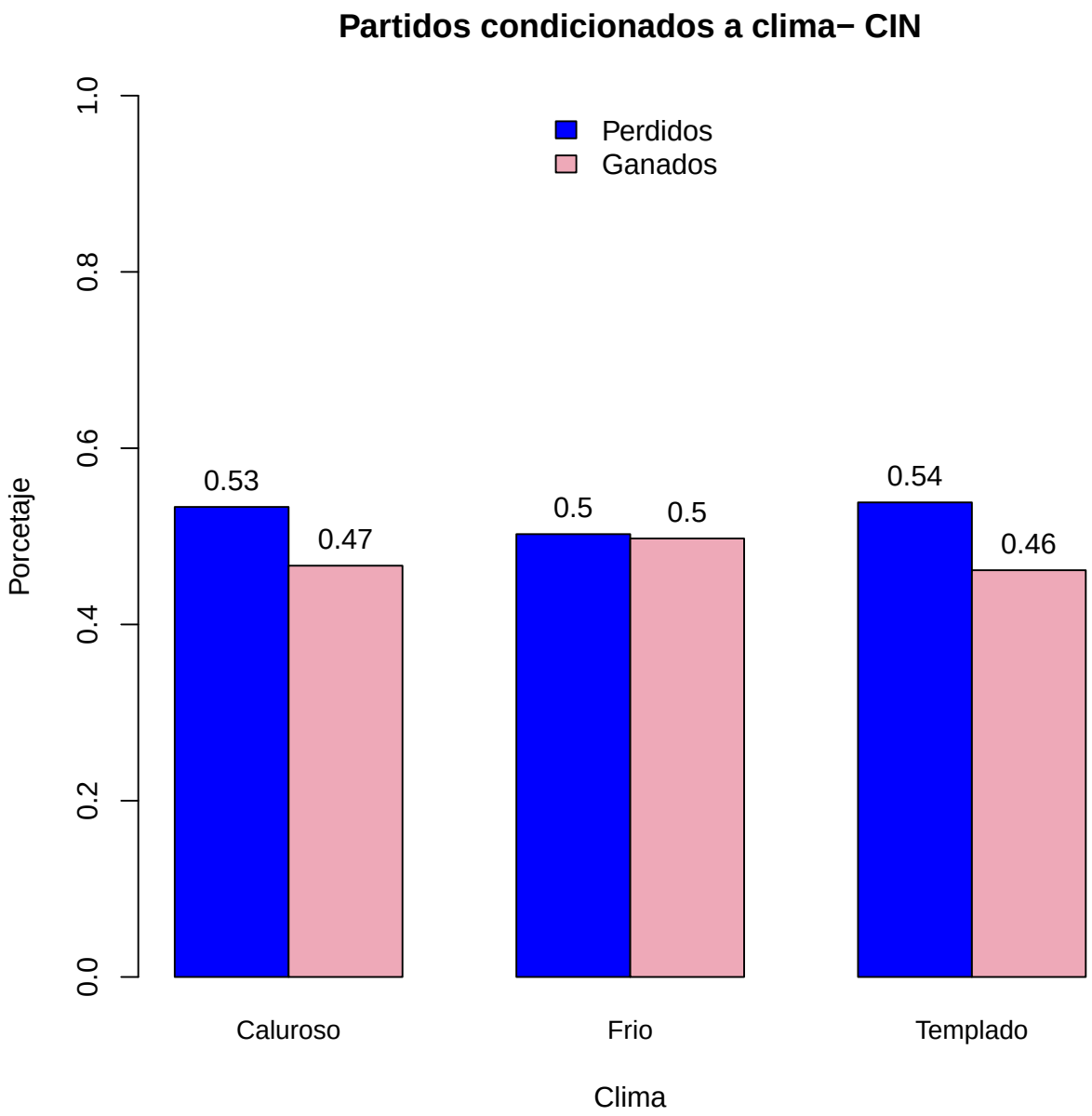
De la variable "cd" (cubo de datos) podemos elegir las variables que se quieran usar, por ejemplo, Cincinnati, y posteriormente graficar las proporciones:

```
##
#Elegimos solo la "rebanada" de Cincinnati
cin<-cd[,,"CIN"]; cin

##
##     Caluroso Frio Templado
##   0        8  102       14
##   1        7  101       12

cin.conjunta<-prop.table(cin)
#Se renombran los ceros y unos
rownames(cin.conjunta)<-c("Perdidos","Ganados")

#se definen los colores a usar
colr=c("blue","pink2")

### GRAFICAMOS, nombre de la tabla, titulo, colores
#Se indica que se quieren las barras una al lado de la otra,
#Nombre de los ejes, límites y tamaño de la letra.

b<-barplot(cin.conjunta,
           main="Partidos Cincinnati",
           horiz=FALSE,
           col=colr,
           beside=TRUE,
           ylab = "Porcetaje",
           xlab="Clima",
           ylim=c(0,1),
           cex.names = .9)

#Nombre de las leyendas, localización, colores y tamaño
text(x=b,y=cin.conjunta,
     labels=c(round(cin.conjunta,2)),
     pos=3,cex=1)

legend("top",rownames(cin.conjunta),
       fill=colr,bty="n",border="black",
       cex=1)
```



Parece que el clima no afecta en nada a este equipo que sigue ganando casi en proporción 50-50,ahora se plantean otras preguntas, ¿y si se condiciona por clima? ¿habrá alguna variación?

```
cin.condicional<-prop.table(cin,2)
rownames(cin.condicional)<-c("Perdidos","Ganados")
cin.condicional

##
##              Caluroso      Frio  Templado
##   Perdidos 0.5333333 0.5024631 0.5384615
##   Ganados  0.4666667 0.4975369 0.4615385
##
```

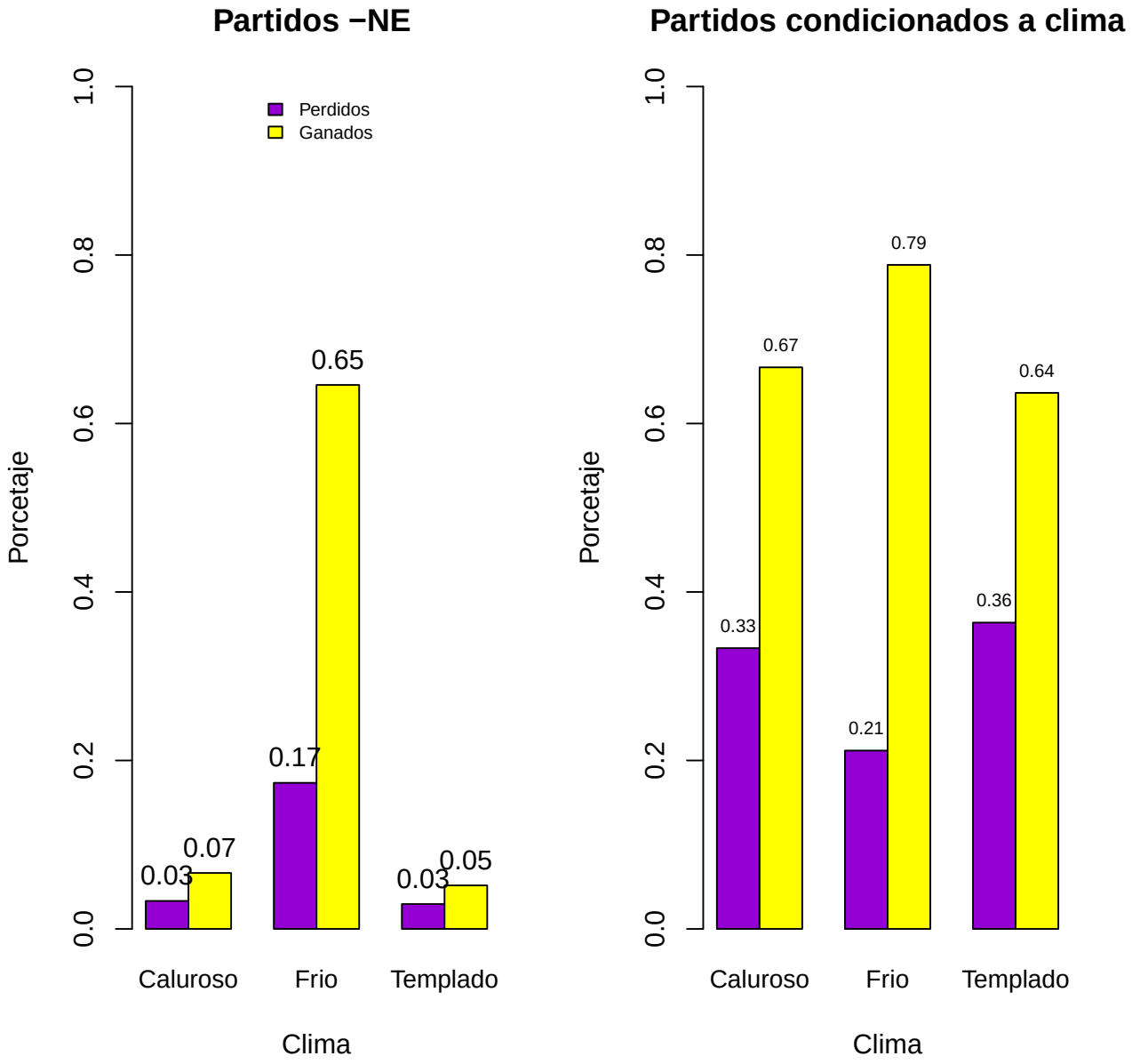


No hay relación entre los partidos ganados y el clima. Esto último se comprueba con la siguiente prueba de hipótesis: la prueba χ^2 se aplica con el comando *chisq.test(tabla de contingencia)*. Como ya se explicó anteriormente, si el p-value es menor a .05, se dice que hay dependencia o asociación entre las variables, formalmente: "se rechaza la hipótesis nula de independencia".

```
chisq.test(cin)

##
##  Pearson's Chi-squared test
##
## data:  cin
## X-squared = 0.15991, df = 2, p-value = 0.9232
```

Con el resultado obtenido se confirma que no hay relación entre el resultado de sus partidos y el clima donde se juegue.

A continuación se realiza el mismo análisis para New England (NE):



Aunque hay una ligera variación en el resultado de los partidos en clima frío (10 % mas que en clima caluroso o templado) la tendencia de este equipo es a ganar en cualquiera de los tres tipos de climas. Por lo tanto no hay relación entre las variables del clima y el resultado. A continuación, se comprueba esta última afirmación formalmente, utilizando la prueba χ^2.

```
chisq.test(ne)

##
##  Pearson's Chi-squared test
##
## data:  ne
## X-squared = 4.1308, df = 2, p-value = 0.1268
```

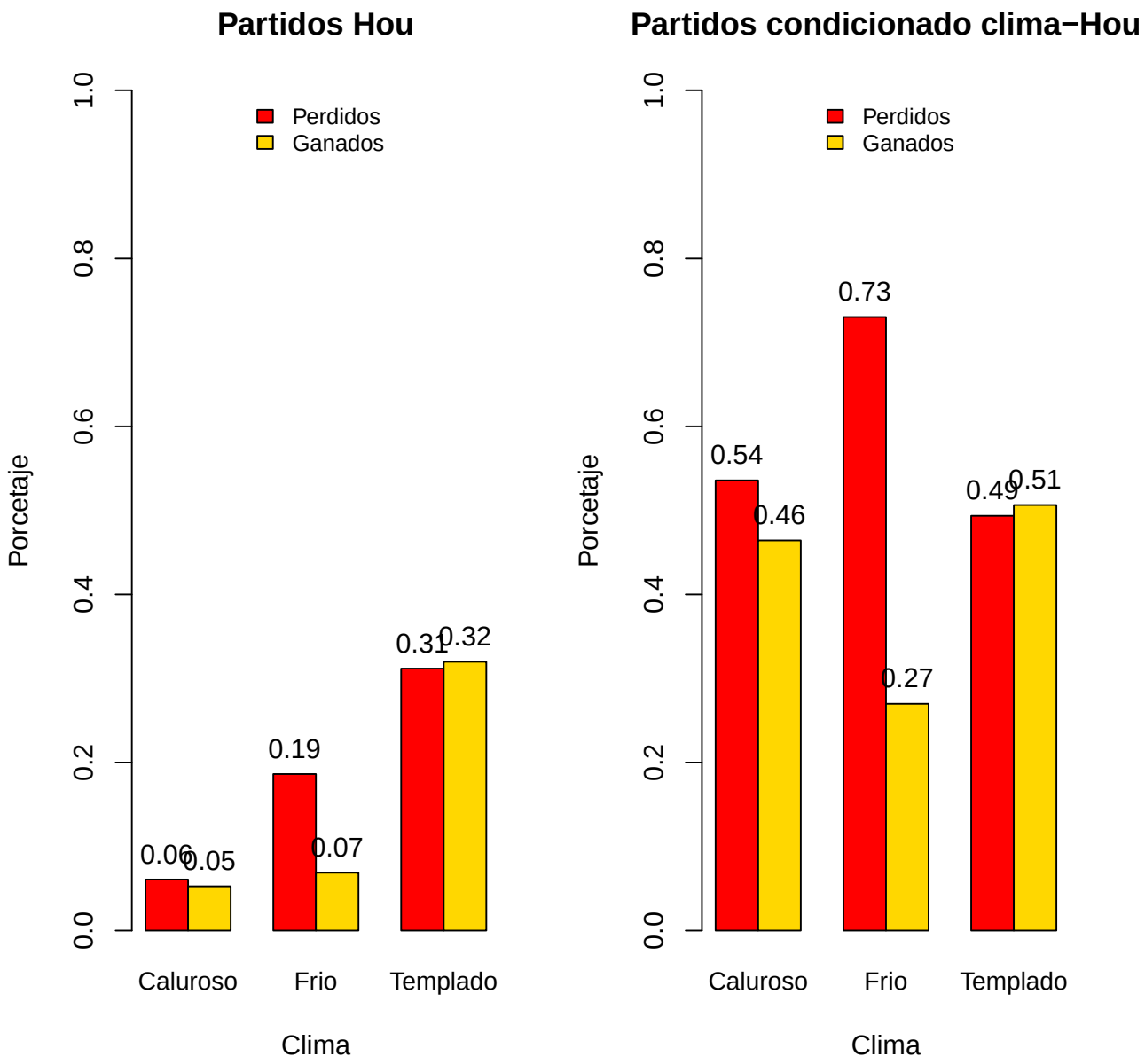
Se confirma que no existe una relación. ¿La intuición falló? ¿El clima no tiene nada que ver con el resultado de un partido o cómo rinde un equipo?

La siguiente lista contiene los p-value de las pruebas χ^2 para el resto de los equipos:

```
PRUEBA

##
## $SF
## [1] "0.034"                "Si le afecta el clima"
## $NYG
## [1] "0.807"                "No le afecta el clima"
## $SD
## [1] "0.066"                "No le afecta el clima"
## $CIN
## [1] "0.923"                "No le afecta el clima"
## $NYJ
## [1] "0.223"                "No le afecta el clima"
## $BUF
## [1] "0.8"                  "No le afecta el clima"
## $CHI
## [1] "0.571"                "No le afecta el clima"
## $MIN
## [1] "0.206"                "No le afecta el clima"
## $DET
## [1] "0.323"                "No le afecta el clima"
## $MIA
## [1] "0.171"                "No le afecta el clima"
## $CAR
## [1] "0.026"                "Si le afecta el clima"
## $BAL
## [1] "0.981"                "No le afecta el clima"
## $JAX
## [1] "0.019"                "Si le afecta el clima"
## $IND
## [1] "0.327"                "No le afecta el clima"
## $GB
## [1] "0.387"                "No le afecta el clima"
## $ATL
## [1] "0.365"                "No le afecta el clima"
## $KC
## [1] "0"                    "Si le afecta el clima"
## $CLE
## [1] "0.103"                "No le afecta el clima"
## $WAS
## [1] "0.6"                  "No le afecta el clima"
## $ARI
## [1] "0"                    "Si le afecta el clima"
## $TEN
## [1] "0.078"                "No le afecta el clima"
# $PHI
```

```
## [1] "0.733"                "No le afecta el clima"
## $NO
## [1] "0.101"                "No le afecta el clima"
## $TB
## [1] "0.886"                "No le afecta el clima"
## $DEN
## [1] "0.98"                 "No le afecta el clima"
## $STL
## [1] "0.197"                "No le afecta el clima"
## $OAK
## [1] "0.11"                 "No le afecta el clima"
## $SEA
## [1] "0.019"                "Si le afecta el clima"
## $HOU
## [1] "0.006"                "Si le afecta el clima"
## $DAL
## [1] "0.275"                "No le afecta el clima"
## $NE
## [1] "0.127"                "No le afecta el clima"
## $PIT
## [1] "0.015"                "Si le afecta el clima"
```

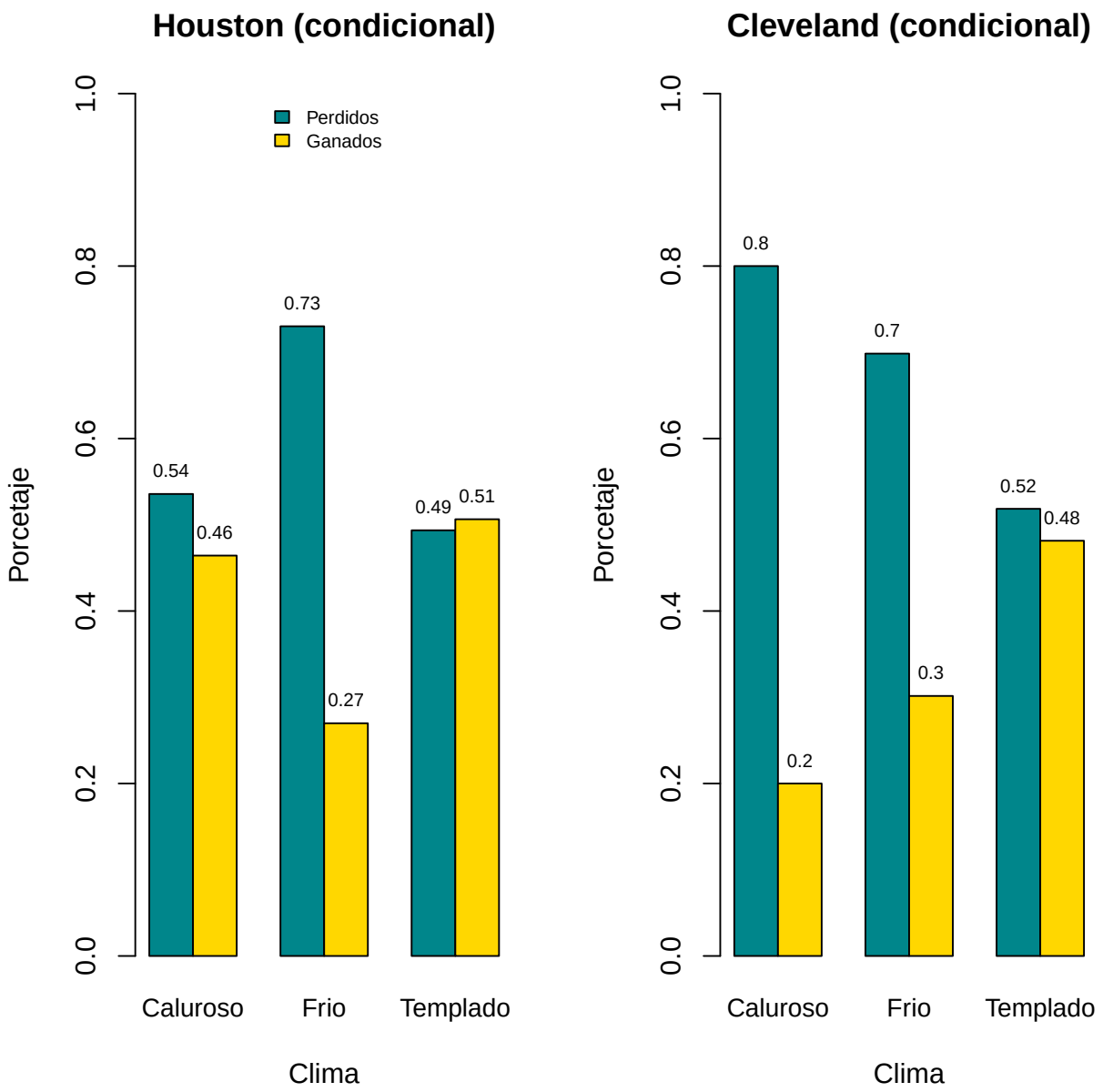
Como se puede observar, hay equipos en los que si hay relación en el resultado del partido y el clima donde se juega. Con la ayuda de esta lista y el p-value, se hace el análisis para uno de los equipos que parecen verse afectados por el clima, primero el caso de Houston:



Interpretación de ambas gráficas: En la primera gráfica se observa que los partidos (ganados y perdidos) que jugó Houston fueron en su mayoría en clima templado y en cantidad mucho menor en clima caluroso. Por otro lado, la gráfica condicional nos brinda la siguiente información: Houston tiende a perder mucho más en clima frío, un 19 % más que en clima caluroso. Sin embargo en clima templado, es casi equiprobable para este equipo ganar o perder. Si este fuera el equipo de su predilección, tendría mas oportunidades de ganar una apuesta si el juego fuese en un clima templado.

De esta manera y al dar la correcta interpretación de las gráficas dadas sus distribuciones (conjunta o condicional), se puede ayudar a tomar decisiones.

Supongamos que hay un juego entre Houston y Cleveland, tenemos acceso a las gráficas condicionadas y el partido va a jugarse en una cuidad de clima caluroso ¿Por quién apostaría? ¿Y si fuera en templado?



Claramente Houston tiene mayor probabilidad de ganar que Cleveland en un clima caluroso. Sin embargo, en clima templado prácticamente es un 50-50 para los equipos. ¿Qué decidirías tú?

5.4. Ejercicios

- Elija dos equipos diferentes para los cuales replicar el análisis anterior.
- Grafique las distribuciones conjuntas y condicionales para ambos equipos.
- Realice la prueba ji-cuadrada y diga si hay independencia entre las variables para cada equipo.
- Decida, si se jugara un partido entre dichos equipos ¿Por cuál apostaría en cada uno de los climas?

6 | Práctica U - Análisis exploratorio de variables cuantitativas.

6.1. Objetivos

- Introducir al alumno en el análisis exploratorio y descriptivo de variables cuantitativas.
- Implementación de métodos gráficos en lenguaje R para el análisis de variables cuantitativas.
- Interpretación de gráficos y coeficiente de correlación de Pearson.

6.2. Introducción

6.2.1. Medidas de tendencia central, dispersión y posición

Las *medidas de tendencia central*, son medidas estadísticas que sirven como puntos de referencia para resumir en un solo valor cierta información de un conjunto de datos. Representan un valor en torno al cual se hallan el resto de los datos. Dos de estas medidas son la media y la mediana.

La **media aritmética** es un punto de referencia estándar, también llamada promedio. Se calcula de la siguiente forma:

$$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

La **mediana** es el valor que separa a los datos en dos grupos con la misma cantidad de datos, cuando están ordenados de tal manera que el 50 % de los datos son mayores a este valor y el otro 50 % son menores.

Las *medidas de dispersión*, por otro lado, miden el grado de variabilidad de los datos, de manera que buscan señalar cuánto se alejan los datos a un cierto número, que generalmente es la media aritmética.

En esta práctica se usará la media y la mediana como ejemplos de medidas de tendencia central; mientras que se utilizará la varianza y la desviación estándar como ejemplos de medidas de dispersión.

La **varianza** (S^2) es el promedio del cuadrado de las distancias entre cada observación y la media aritmética, corregido por un factor de insesgamiento por lo que R se divide entre $n - 1$.

La **desviación estándar** es la raíz cuadrada de la varianza y nos indica en promedio cuánto se alejan los datos de la media.

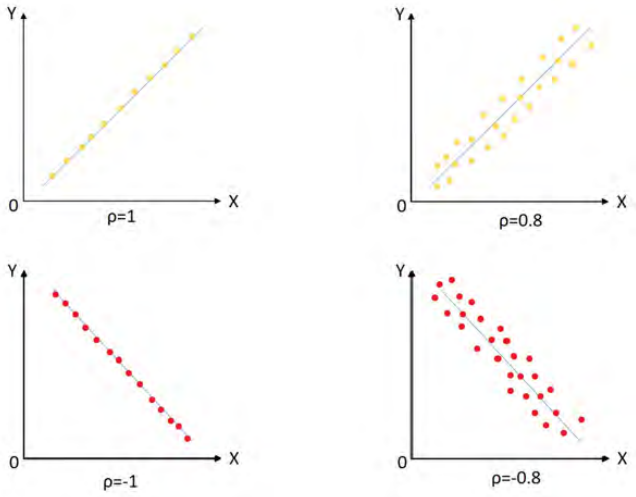
Otro tipo de medidas descriptivas son las *medidas de posición*, las cuales dividen a un conjunto de datos de una distribución con base en si exceden cierto valor.

Un ejemplo de medida de posición son los cuantiles, los cuales son valores que dividen a un conjunto de datos en partes iguales. Los cuantiles más utilizados son los cuartiles, los cuales dividen a los datos en cuatro grupos con el mismo número de elementos. Son denotados por Q_1, Q_2 (mediana) y Q_3 y determinan los valores correspondientes al 25 %, al 50 % y al 75 % de los datos.

Finalmente, en esta práctica se utilizará el **coeficiente de correlación de Pearson**, el cual es un indicador de la relación lineal entre variables aleatorias cuantitativas y es calculado de la siguiente forma:

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{(n-1)S_x S_y}$$

El valor del coeficiente puede variar entre -1 y 1. Si $r > 0$ se dice que hay una correlación positiva, es decir, al aumentar los valores de una variable aumentan los de la segunda, en particular, si $r = 1$ se dice que hay una relación lineal positiva perfecta. En sentido contrario, si $r < 0$, las variables se relacionan en sentido inverso y si r=-1 entonces se dice que tienen una correlación negativa perfecta. Finalmente, cuando $r = 0$, solo se puede decir que no hay una relación lineal entre las variables, sin embargo, esto no implica de ninguna manera que las variables sean independientes.



6.2.2. Diagrama de caja

Un diagrama de caja es una gráfica utilizada comúnmente para representar el resumen de una variable cuantitativa, en este gráfico se visualizan los datos mínimos, máximos, los cuartiles y en algunas ocasiones los valores extremos (outliers) de la siguiente manera:

Así pueden apreciarse en qué intervalos hay mayor cantidad de datos y si hay algún sesgo.

6.3. Análisis de variables cuantitativas

En esta práctica se utiliza la conexión PostgreSQL y R (explicada con detalle en la práctica R) mediante la cual se obtiene la siguiente consulta (el código de la misma se encuentra en el anexo de la práctica):

```
##    mi_equipo ganados perdidos empatados anotados recibidos
## 1        ARI     116      132         1     5289      5829
## 2        ATL     134      117         1     5844      5658
## 3        BAL     145      112         0     5618      4873
## 4        BUF     101      139         0     4893      5368
## 5        CAR     134      119         1     5452      5307
## 6        CHI     117      129         0     5111      5347
## 7        CIN     120      124         3     5368      5390
## 8        CLE      76      165         0     4196      5466
## 9        DAL     133      115         0     5628      5438
## 10       DEN     152      102         0     6084      5576
## 11       DET      85      158         0     4989      6082
## 12        GB     161      100         1     6855      5686
## 13       HOU     109      138         0     5024      5557
## 14       IND     172       91         0     6580      5718
## 15       JAX      94      149         0     4695      5509
## 16        KC     120      127         0     5501      5459
## 17       MIA     109      133         0     4803      5157
## 18       MIN     120      126         1     5561      5532
## 19        NE     207       64         0     7612      5130
## 20        NO     135      115         0     6506      5993
## 21       NYG     135      118         0     5735      5628
## 22       NYJ     119      132         0     5038      5269
## 23       OAK      88      156         0     4756      6050
## 24       PHI     144      110         1     6142      5419
## 25       PIT     167       94         1     6139      4984
## 26        SD     134      116         0     6178      5451
## 27       SEA     150      112         1     6095      5287
## 28        SF     113      136         1     5041      5596
## 29       STL      88      154         1     4506      5859
## 30        TB     105      140         0     4806      5253
```

```
## 31       TEN     115      131          0     5214      5724
## 32       WAS     100      144          1     4883      5547
```

¿Qué tipo de datos se tiene? Esta consulta contiene los nombres de los equipos, el total de partidos ganados, perdidos y empatados, así como la suma de todos los puntos recibidos y anotados a lo largo de los juegos durante las 15 temporadas registradas en la base de datos. Esta consulta contiene una recopilación de datos continuos o variables cuantitativas.

Se puede obtener un resumen de la información de la consulta con el comando *summary()*, este comando se utiliza para realizar análisis descriptivos de varios tipos e imprime un resumen estadístico completo dependiendo de la variable del argumento. En este caso, al ser variables (con excepción de la variable "mi equipo") cuantitativas, devuelve la frecuencia mínima y máxima, el primer, segundo y tercer cuartil así como la media o promedio de los datos.

```
summary(perfiles_cuanti)
```

```
##
##   mi_equipo            ganados         perdidos       empatados
##  Length:32          Min.   : 76.0   Min.   : 64.0   Min.   :0.0000
##  Class :character   1st Qu.:108.0   1st Qu.:114.2   1st Qu.:0.0000
##  Mode  :character   Median :120.0   Median :126.5   Median :0.0000
##                     Mean   :124.9   Mean   :124.9   Mean   :0.4375
##                     3rd Qu.:137.2   3rd Qu.:138.2   3rd Qu.:1.0000
##                     Max.   :207.0   Max.   :165.0   Max.   :3.0000
##
##     anotados      recibidos
##  Min.   :4196   Min.   :4873
##  1st Qu.:4965   1st Qu.:5337
##  Median :5410   Median :5488
##  Mean   :5504   Mean   :5504
##  3rd Qu.:6087   3rd Qu.:5665
##  Max.   :7612   Max.   :6082
##
```

El comando anterior reconoce el tipo de variable, así que para la variable "mi equipo" al no ser una variable cuantitativa, sino cualitativa, no se obtuvo la información que se obtiene para las otras variables. Para omitir esa variable basta con especificar [-1] para quitar la primera columna, de la siguiente forma:

```
summary(perfiles_cuanti[-1])
```

Se puede interpretar la tabla o puede hacerse un análisis individual para cada variable. Por ejemplo, para **partidos ganados**:

```
summary(perfiles_cuanti$ganados)
```

```
##
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    76.0   108.0   120.0   124.9   137.2   207.0
##
##
```

La tabla anterior se puede interpretar de la siguiente forma::

- **Min.** El mínimo de partidos ganados es de 76.
- **Max.** El máximo de partidos ganados es de 207.
- **1st Qu.** El 25 % de los equipos tienen menos de 108 partidos ganados.
- **Median.** La mayoría de los equipos tienen alrededor de 120 partidos ganados (El 50 % de los equipos tienen por debajo de 120 partidos ganados y el otro 50 % esta por encima).
- **Mean.** El promedio por todos los equipos es de 124.9 partidos ganados.
- **3rd Qu.** El 75 % de los equipos está por debajo de los 137.2 partidos ganados.

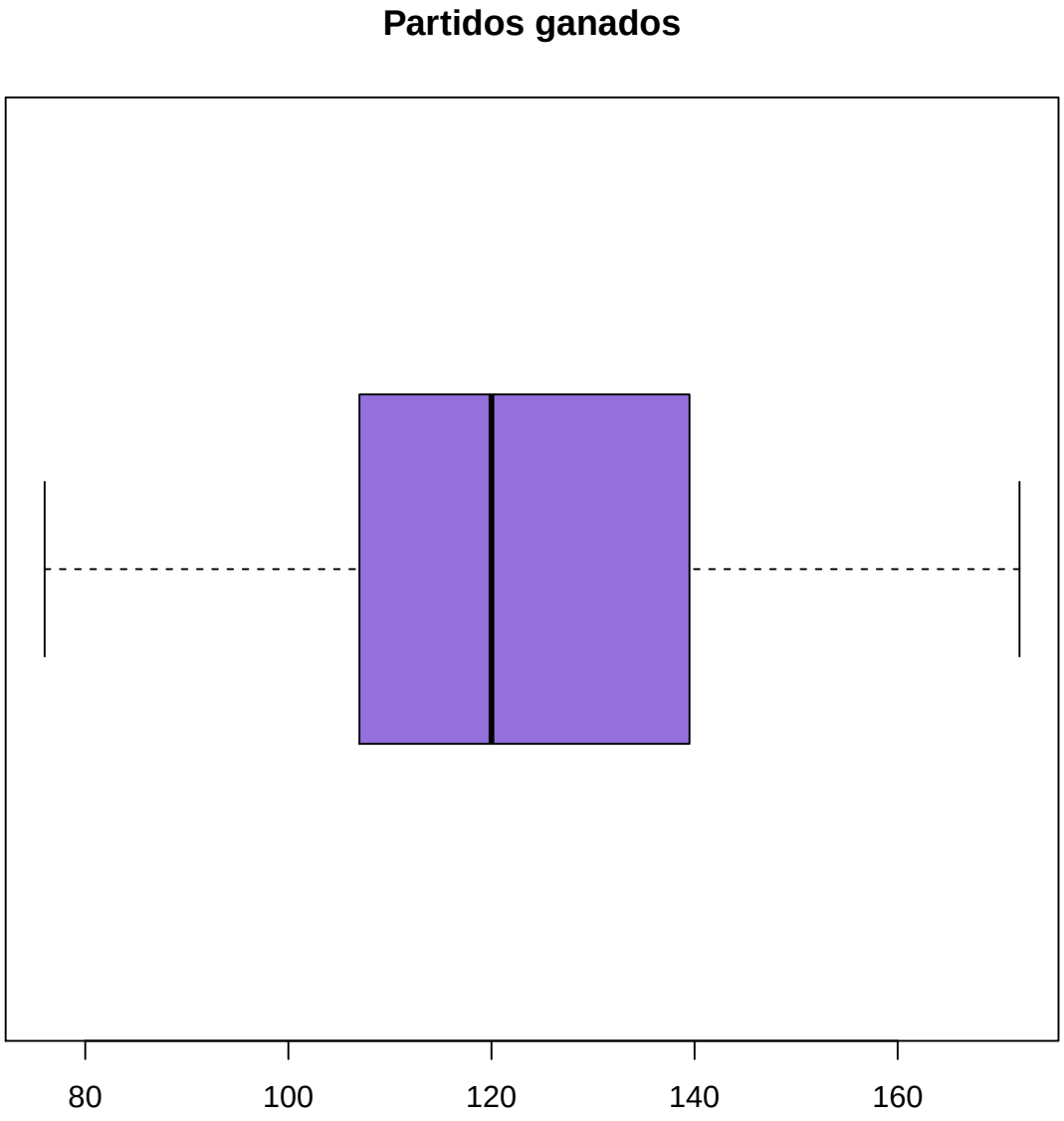
También se pueden obtener la varianza y desviación estándar con los comandos *var()* y *sd()* como se muestra a continuación, siguiendo con el ejemplo de partidos ganados:

```
# Matriz de varianzas y covarianzas para todas las variables
#var(perfiles_cuanti[-1])

#Solo para partidos ganados:
var(perfiles_cuanti$ganados)

## [1] 804.125

sd(perfiles_cuanti$ganados)

## [1] 28.3571
```

Entonces la varianza es de 804.125 partidos ganados y la desviación estándar es de 28.35 partidos. Dicho de otra manera, se espera que los equipos tengan alrededor de 96.55 y 153.25 partidos ganados (media ± desviación estándar).

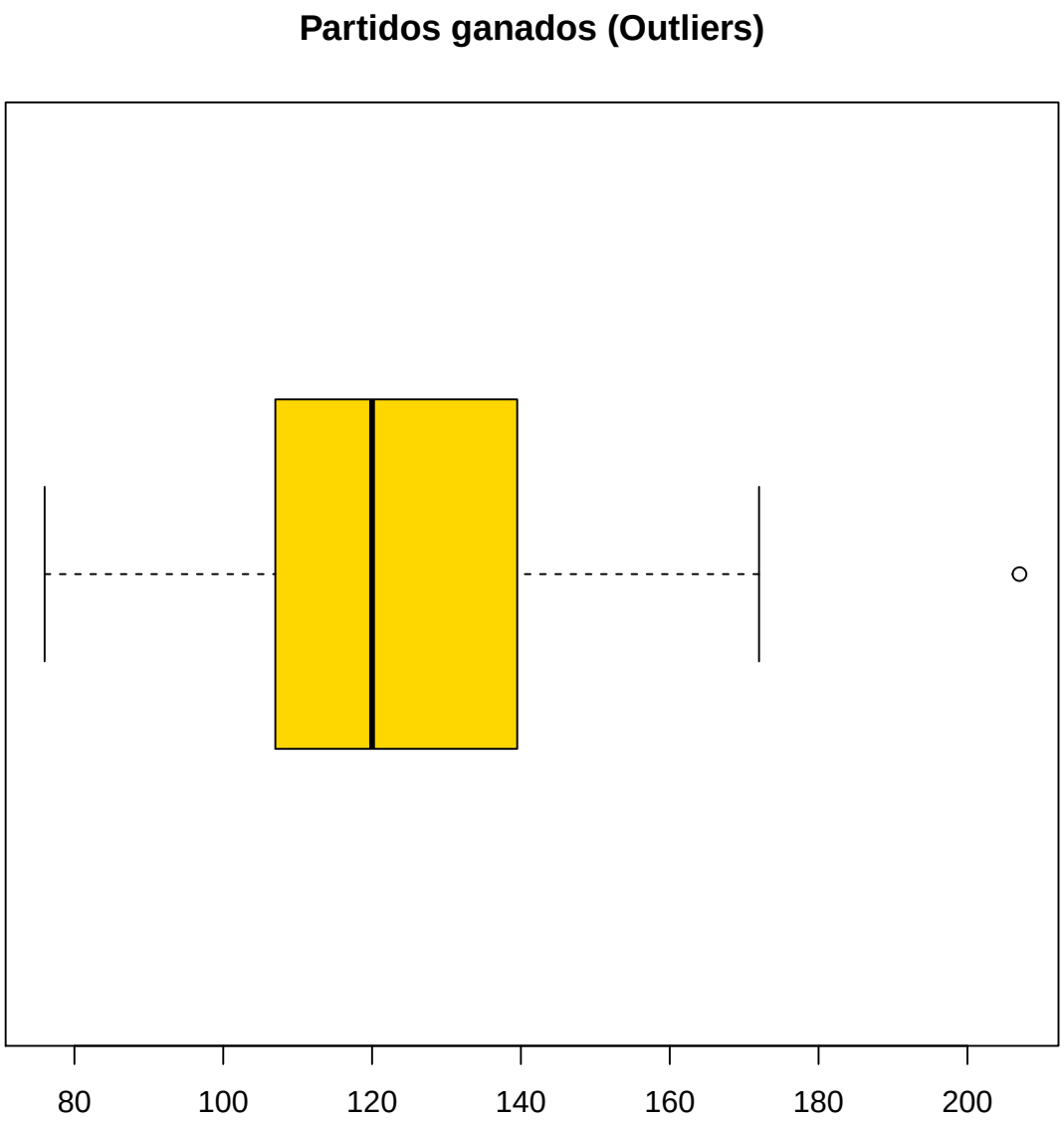
¿Cómo se pueden visualizar todos estos resultados de manera gráfica? El diagrama de caja o gráfica de caja proporciona los datos previamente obtenidos con la función *summary*() de una manera visual:

```
 #sin datos atipicos (Summary)

boxplot(perfiles_cuanti$ganados, #Variable
        main="Partidos ganados", #Titulo
        col="mediumpurple",  #Color de la caja
        horizontal = T,  #Se quiere la gráfica horizontal
        outline = FALSE) #No aparecen outliers
```



El comando *boxplot* tiene la posibilidad de reconocer outliers, de esta manera se puede dar un mejor resumen según sea el caso; lo único que se debe hacer es cambiar *outline = TRUE* en el código anterior para obtener:

```
boxplot(perfiles_cuanti$ganados, #Variable
        main="Partidos ganados (Outliers)", #Título
        col="gold",  #Color de la caja
        horizontal = T,  #Se quiere la gráfica horizontal
        outline = TRUE) #Aparecen outliers
```



¿Cuál es la importancia de los *outliers* o valores atípicos o extremos? Un *outlier* es un elemento de los datos que es significativamente diferente a los otros. Estos datos pueden causar problemas al hacer algún tipo de modelo o al interpretar la infomación, ya que suelen tener influencia en la media o promedio, por lo que un dato de este tipo puede llegar a sesgar la información.

Para nuestro ejemplo, se investigará si el dato atípico de la base influye en las medidas de tendencia central. Primero se averigua cuál es el equipo que contiene este dato:

```
perfiles_cuanti[perfiles_cuanti$ganados==max(perfiles_cuanti$ganados),]

##    mi_equipo ganados perdidos empatados anotados recibidos
## 19        NE     207       64         0     7612      5130
```

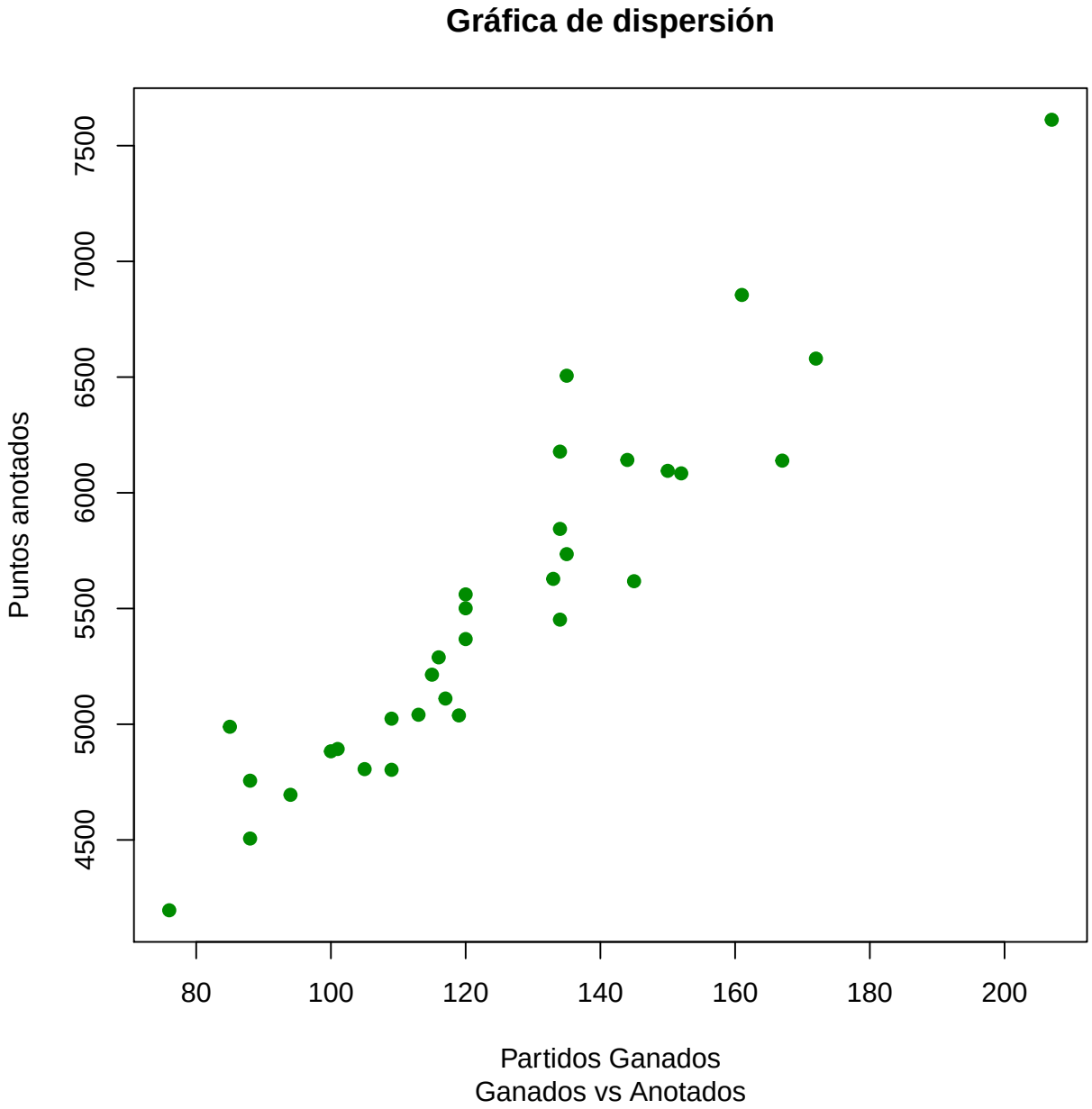
Posteriormente se guarda en otra variable la consulta sin la información de este equipo y se comparan los resúmenes de los resultados:

```
sin19<-perfiles_cuanti[-19,]
#Sin el dato extremo
summary(sin19$ganados)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    76.0   107.0   120.0   122.3   135.0   172.0

#Con el dato extremo
summary(perfiles_cuanti$ganados)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    76.0   108.0   120.0   124.9   137.2   207.0
```

Se puede observar que los datos prácticamente no varían y el promedio solo cambia por 3 unidades, lo que indica que a pesar de que New England gana significativamente más, no hace una diferencia en el análisis completo de los equipos.

¿Qué otro análisis es posible hacer? El comando *plot()* crea gráficas de dispersión, las cuales permiten ver tendencias entre datos de la siguiente manera:

```
x<-perfiles_cuanti$ganados
y<-perfiles_cuanti$anotados

#Graficando
#Varianles, título, subtítulo
#Nombres de los ejes
#Color de los puntos, tamaño y tipo
plot(x,y,
     main="Gráfica de dispersión",
     sub= "Ganados vs Anotados",
     xlab="Partidos Ganados",
     ylab="Puntos anotados",
     col="green4", cex = 1,pch = 19)
```



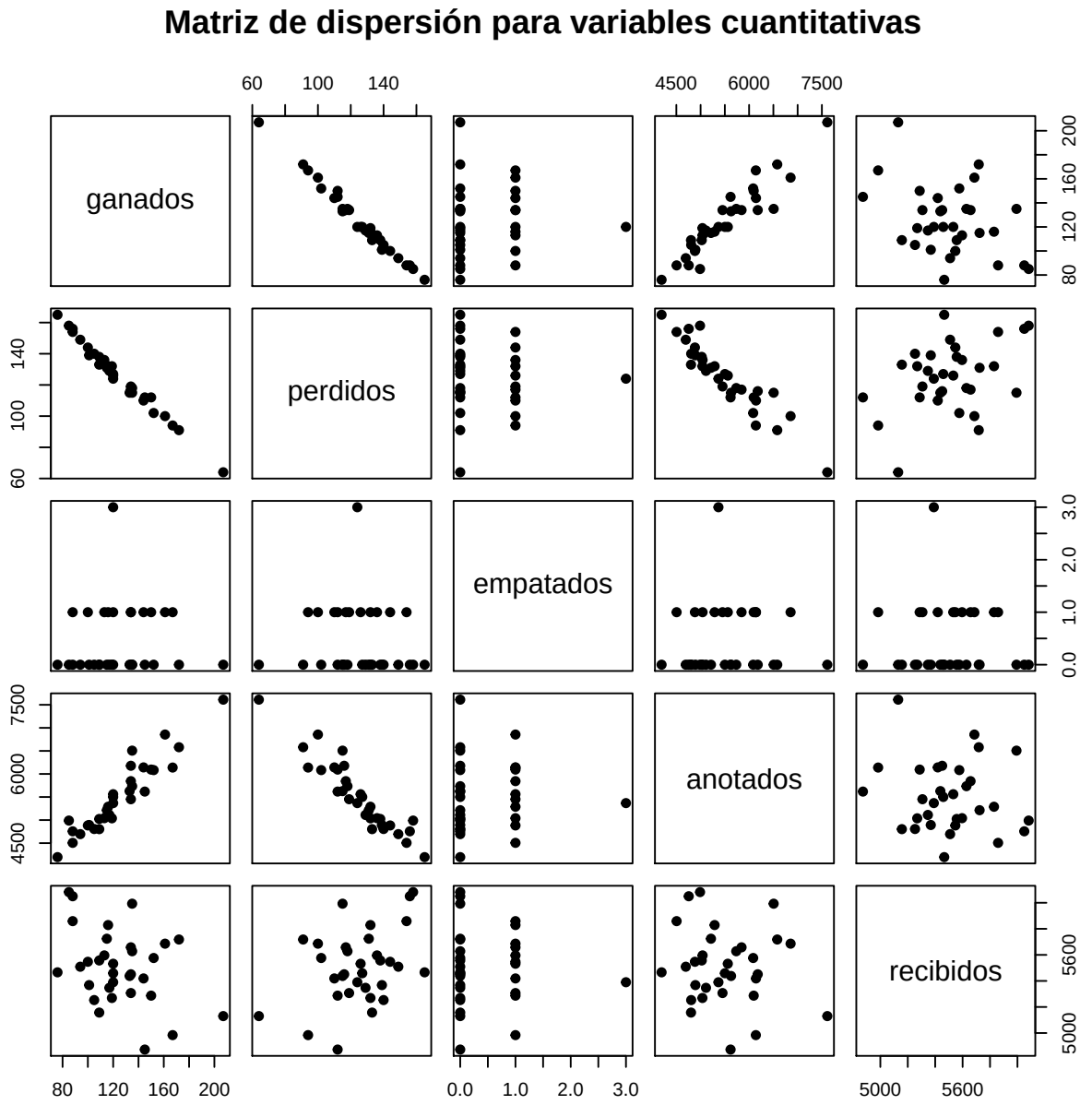
En la gráfica se puede ver que hay una clara relación entre los partidos ganados y los puntos anotados, lo cual tiene sentido ya que mientras más partidos un equipo haya ganado seguramente sumará más puntos a los ya anotados. Intuitivamente estas variables tienen una correlación alta. Se comprueba con el comando *cor()* que calcula por *default* el coeficiente de correlación de Pearson:

```
cor(x,y)

## [1] 0.9389939
```

El coeficiente expresa una correlación muy alta, cercana a 1, por lo tanto estas variables tienen una correlación lineal positiva muy alta.

R proporciona un comando *pairs()* para hacer el análisis previamente descrito para cada posible pareja de las diferentes variables de una base de datos (se debe recordar que esto sólo tiene sentido para variables cuantitativas).

```
  #Base con variables cuantitativas, titulo
  #Tamaño, estilo y color de los puntos, tamaño de las etiquetas.
pairs(perfiles_cuanti[-1],
      main="Matriz de dispersión para variables cuantitativas",
      cex = 1, pch = 19, col="black", cex.labels =1.5)
```

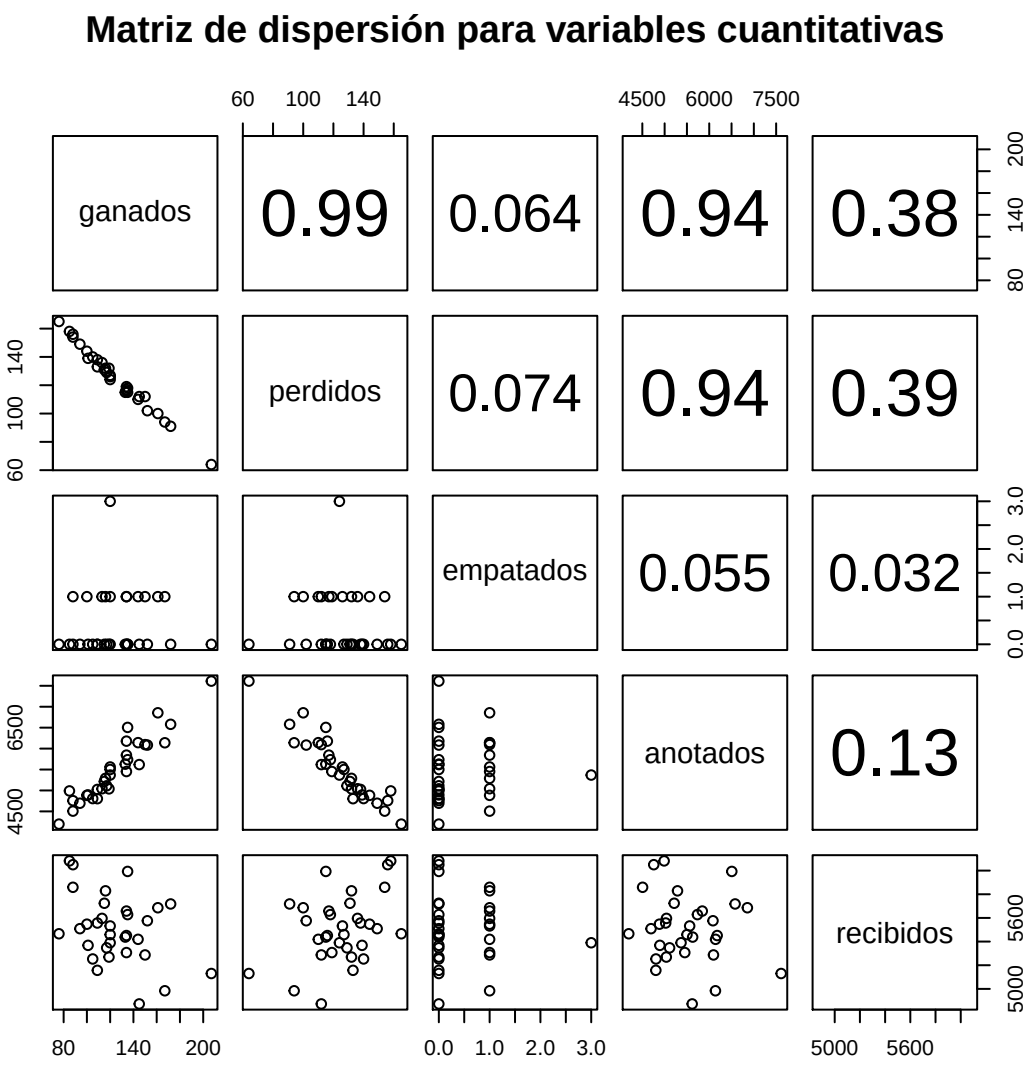


La gráfica anterior devuelve las gráficas de dispersión para parejas de las variables. Se pueden apreciar las correlaciones lineal negativas entre partidos perdidos y ganados, lo cual tiene sentido ya que mientras más partidos se ganan menos se pierden, esta información es redundante, sin embargo ejemplifica bien el uso de este tipo de gráfica y la interpretación del coeficiente de Pearson.

Al mismo tiempo, se aprecia la correlación entre las variables anotados con partidos perdidos, esta correlación es negativa como se puede intuir, ya que mientras más se anota, se debería perder menos.

Si se quisiera obtener toda la información, visual y numérica en un solo gráfico se puede utilizar la función *panel.cor*() como argumento en la función *pairs*()

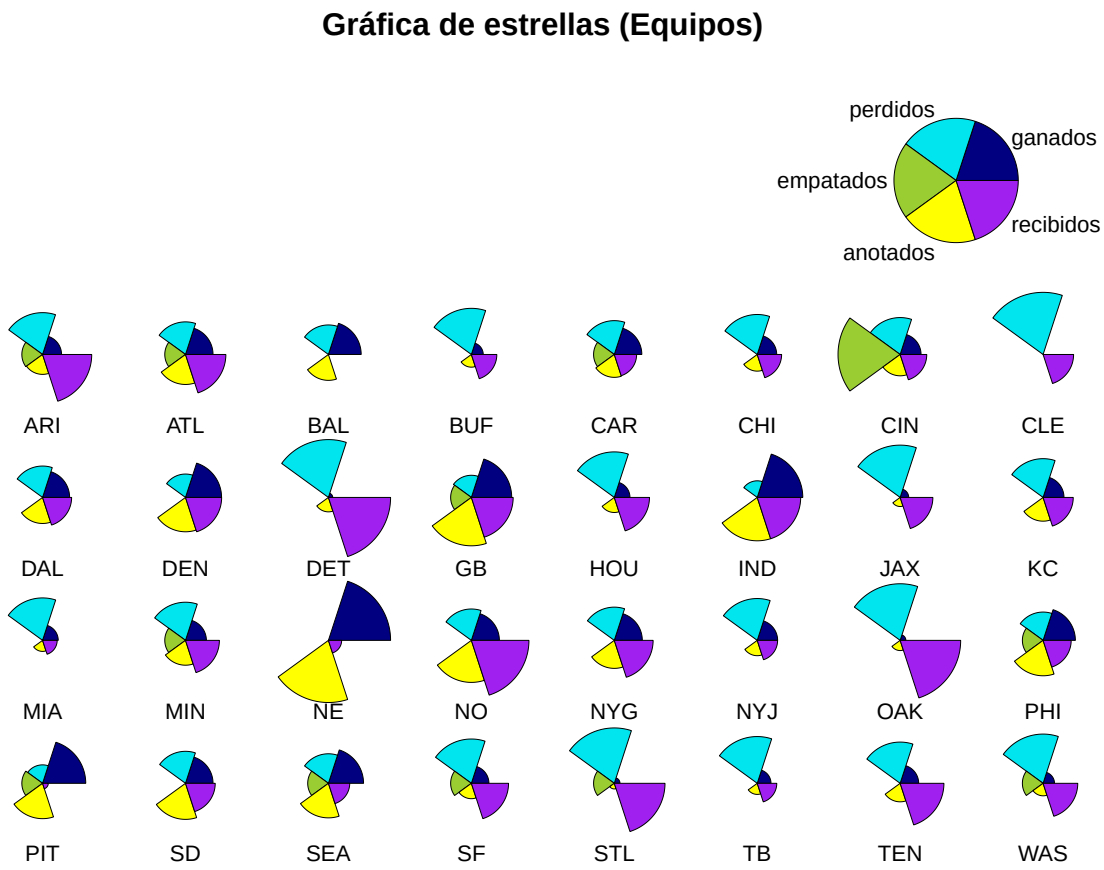
```r
#Correlacion
panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
usr <- par("usr"); on.exit(par(usr))
par(usr = c(0, 1, 0, 1))
r <- abs(cor(x, y))
txt <- format(c(r, 0.123456789), digits=digits)[1]
txt <- paste(prefix, txt, sep="")
if(missing(cex.cor))
cex <- 0.8/strwidth(txt)
text(0.5, 0.5, txt, cex = cex)
}


pairs(perfiles_cuanti[-1],
      main="Matriz de dispersión para variables cuantitativas",
      upper.panel=panel.cor) #Se utiliza la función en el panel superior.
```



Se puede observar que, hay correlación fuerte entre partidos ganados y perdidos (.99) y como se mencionó antes, entre partidos ganados y los puntos anotados (.94) Por otra parte, aunque parezca extraño, hay una correlación débil entre los puntos recibidos y los partidos perdidos. Esta gráfica permite conocer de un sólo vistazo mucha de la información que puede ser relevante para análisis más complejos.

6.3.1. Gráficas de estrellas

¿Qué pasa si se requiere conocer a los equipos con características semejantes? Se puede hacer un análisis equivalente al que se hizo de perfiles en la práctica S pero con variables cuantitativas. Para esto existen gráficos ya diseñados, uno de ellos es el siguiente:



Esta gráfica permite comparar a los equipos con base en sus variables, la escala es tomada en relación al mayor valor dentro de la variable. Por ejemplo, “CIN” es el equipo que empató más veces, 3 para ser exactos, por lo que en la variable empatados, el segmento correspondiente es el más grande.

¿Qué equipos se parecen? Detroit (DET) y Oakland (OAK) son dos que parece comparten características. Se puede confirmar comparando a ambos equipos dentro de la siguiente consulta:

```
##    mi_equipo ganados perdidos empatados anotados recibidos
## 11       DET      85      158         0     4989      6082
## 23       OAK      88      156         0     4756      6050
```

Como se puede observar, tal como se reconoció en la gráfica de segmentos, los equipos tienen características semejantes, ganaron y perdieron casi el mismo número de partidos. Y sus puntos anotados y recibidos son parecidos.

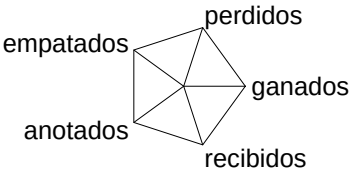
La gráfica anterior es resultado del siguiente código:

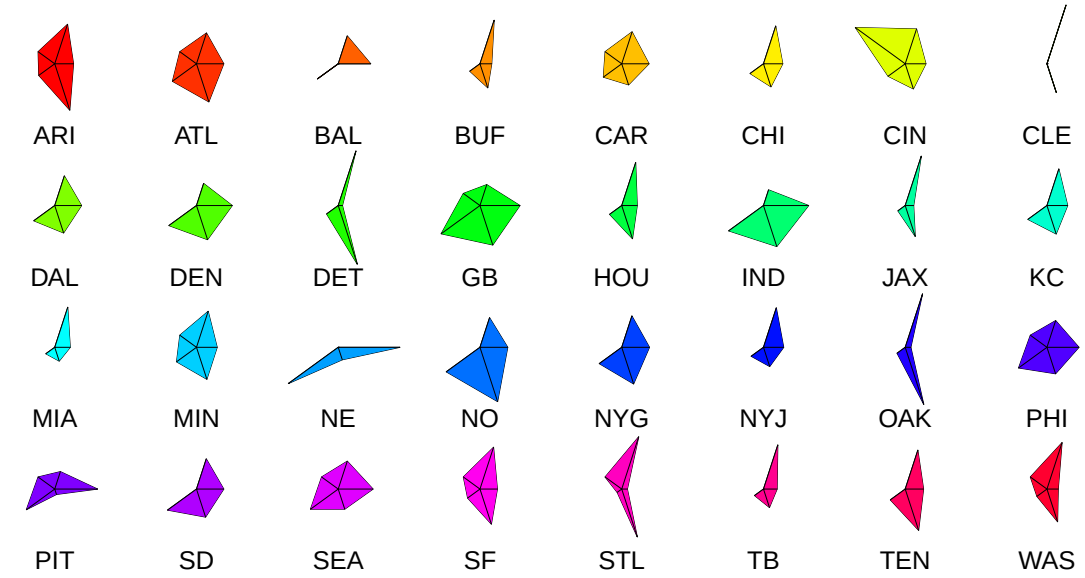
```
#gráficas de estrellas
#Vector con los colores que se usarán posteriormente
colores<-c("navyblue","turquoise2", "yellowgreen", "yellow","purple")
a<-perfiles_cuanti[-1] # "a" Solo contiene variables cuantitativas


#Graficando
#Datos título, tipo de gráfica: segmentos, número de renglones
#localización de la llave. etiquetas de las estrellas, colores,

stars(a,main="Gráfica de estrellas (Equipos)",
      draw.segments = T, nrow = 4,
      key.loc=c(17,12),
      labels = perfiles_cuanti$mi_equipo,
      col.segments = colores )
```

Sin embargo, existen variantes de éste gráfico, basta con modificar los argumentos *full* = *F* o *draw.segments* = *F* dentro de la función *stars* para obtener las siguientes gráficas.

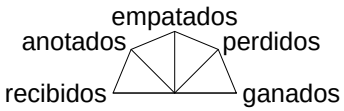
```
stars(a,col.stars=rainbow(32),main="Gráfica de estrellas 2",
      key.loc=c(17,12), labels = perfiles_cuanti$mi_equipo, nrow = 4)

stars(a,col.stars=rainbow(32),main="Gráfica de estrellas 3",
    key.loc=c(17,12), full = F, labels = perfiles_cuanti$mi_equipo, nrow = 4)
```

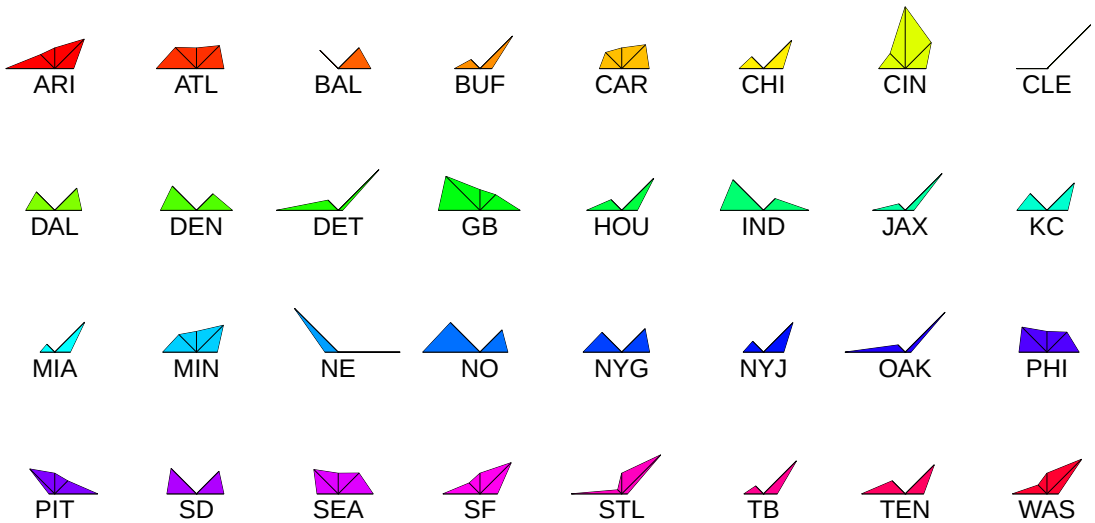
Gráfica de estrellas 2





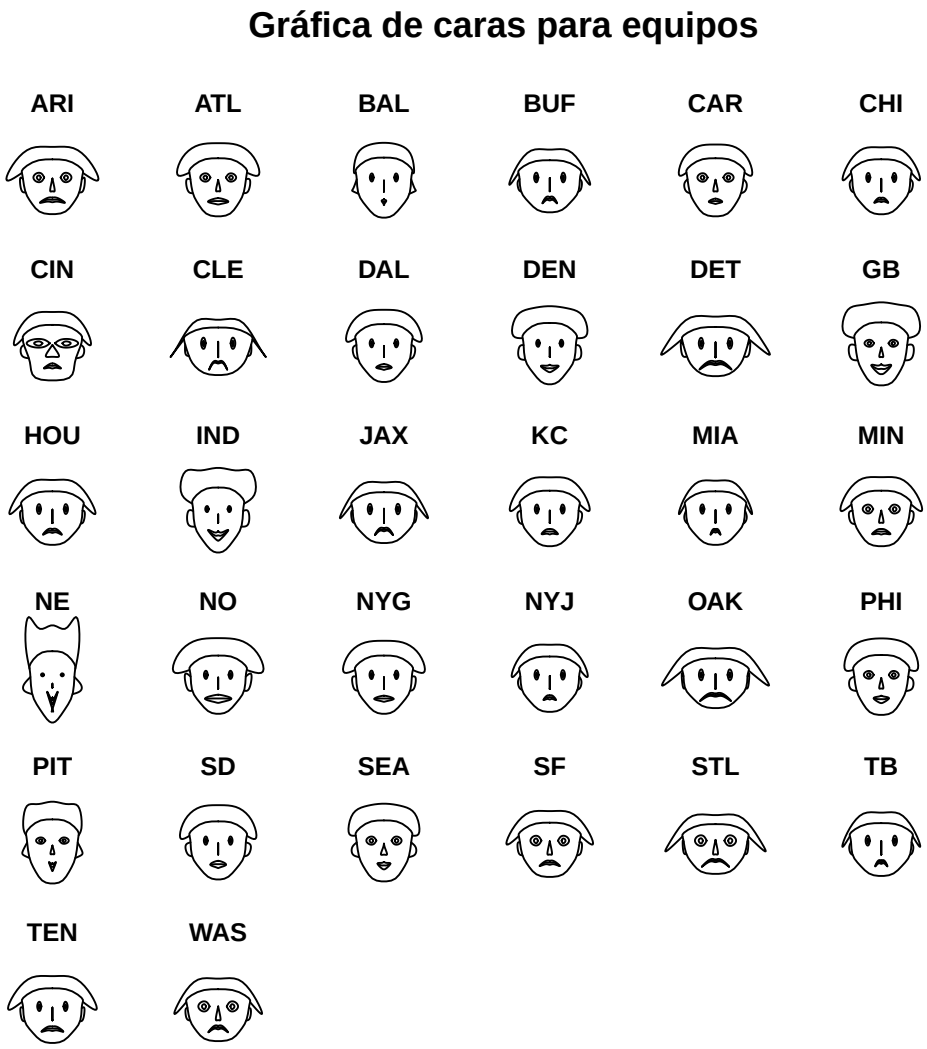
Gráfica de estrellas 3

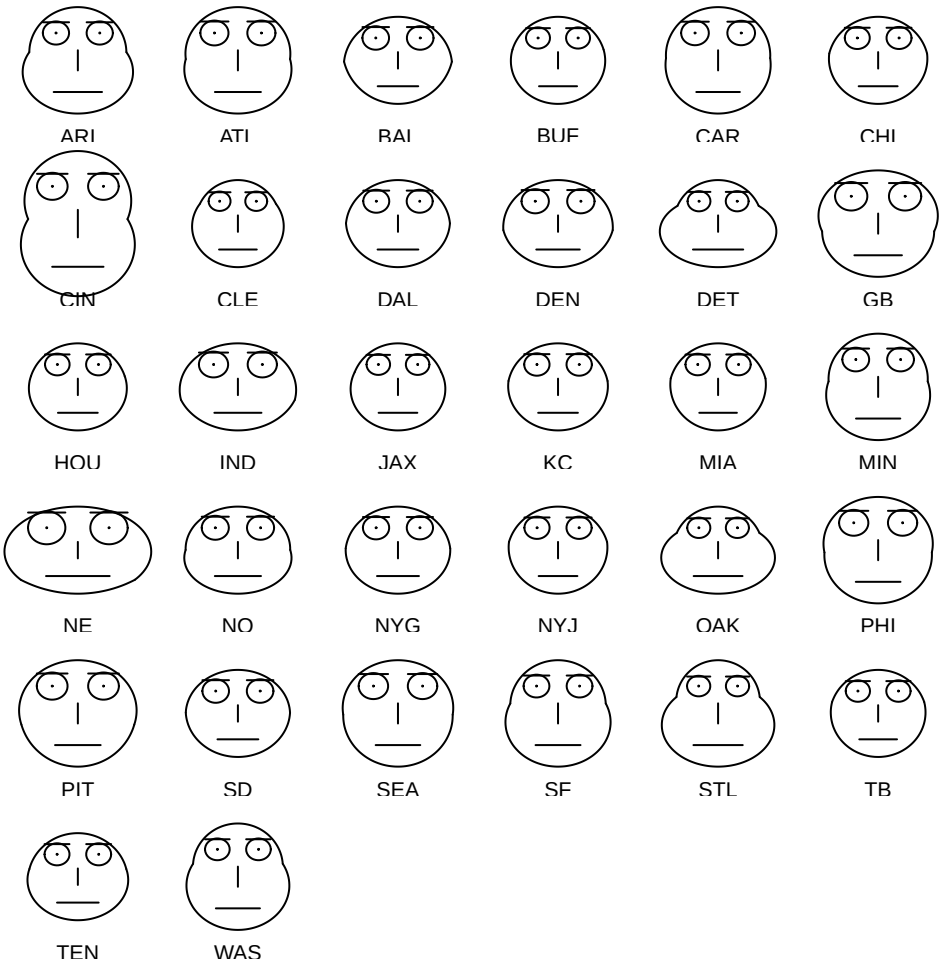




También existen gráficas de "caritas" (gráficas de Chernoff), igualmente diseñadas para buscar a los sujetos que tienen características semejantes. Para estas gráficas se requieren los paquetes *"aplpack"* y *TeachingDemos* y al igual que el tipo anterior de gráficas, mientras las caras se parezcan más, las variables de los sujetos serán mas parecidas. Este tipo de gráfica es útil cuando se tiene muchas variables, las cuales pueden ser asignadas al tamaño de la nariz, orejas, ojos o rostro.

```
library(aplpack)

faces(a, main="Gráfica de caras para equipos")


library(TeachingDemos)

faces2(a, labels = perfiles_cuanti$mi_equipo)
```



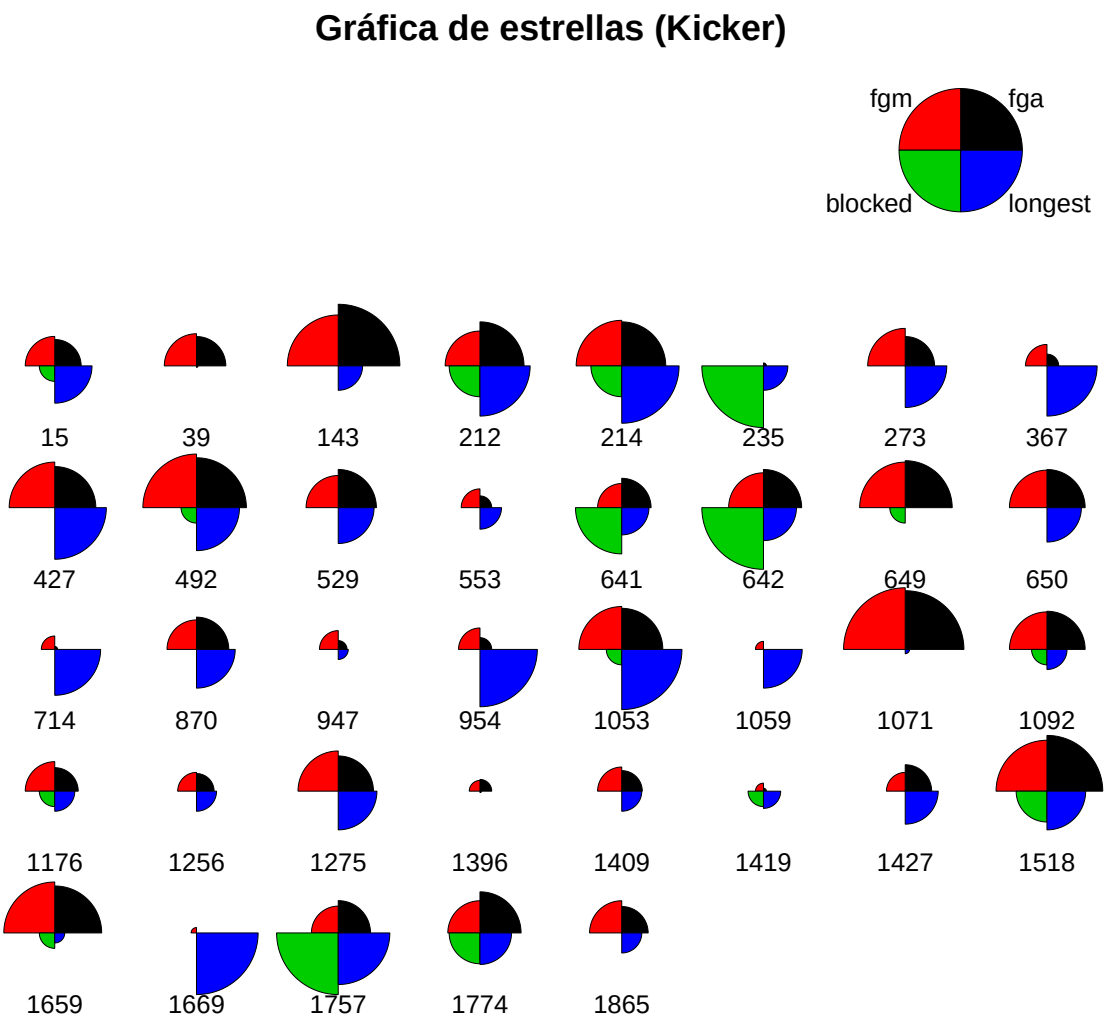


¿Cuál le gusta más? ¿Cuál le parece más útil visualmente? ¿En qué casos cree que pueden ser útiles este tipo de gráficas?

Otra aplicación:

Supongamos que un equipo no dispone de su pateador por un accidente (su pateador tiene el id_ jugador #954) se tiene la posibilidad de elegir otro de los 36 restantes. ¿A cuál elegiría? ¿Por qué? Recuerde que *fga* son goles de campo intentados, *fgm* son goles de campo anotados, *blocked* son goles de campo bloqueados y longest es el gol de campo más largo anotado en su carrera.

```
#Consulta de la tabla kicker
base <- dbGetQuery(conexion, "SELECT * FROM kicker")

#Quitando variables que no son cuantitativas
b<-base[,c(3:6)]

#Grafica de estrellas (segmentos)
stars(b,main="Gráfica de estrellas (Kicker)",
      draw.segments = T, nrow = 5,
      key.loc=c(17,15), labels = base$id_jugador)
```



6.4. Ejercicios

- Basado en la gráfica “Kicker”, responda: ¿A qué kicker elegiría y por qué?
- Para los jugadores que tienen la posición de “punter”:
  - Obtenga el resumen de cada una de sus variables e interprete.
  - Obtenga la gráfica de caja para dos de sus variables, ¿Existen outliers? ¿Qué pásа si los elimina?
  - ¿Existe correlacion entre las variables? ¿De qué tipo?
  - Obtenga la gráfica que más le guste para observar perfiles y diga que jugadores se parecen.

6.5. Anexo

```
-- concentrado total
SELECT mi_equipo, (ganados_l + ganados_v) AS ganados, (perdidos_l + perdidos_v) AS perdidos, (empatados_l + empatados_v) AS empatados, (puntos_anotados_l + puntos_anotados_v) AS anotados, (puntos_recibidos_l + puntos_recibidos_v) AS recibidos
FROM (
SELECT mi_equipo, ganados_l, perdidos_l, empatados_l, puntos_anotados_l, puntos_recibidos_l
FROM (SELECT id_equipo_l AS mi_equipo, COUNT(id_equipo_l) AS ganados_l
      FROM partido
      WHERE marcador_l > marcador_v
      GROUP BY id_equipo_l) AS ganados_local NATURAL JOIN (SELECT id_equipo_l AS mi_equipo, COUNT(id_equipo_l) AS perdidos_l
                FROM partido
                WHERE marcador_l < marcador_v
                GROUP BY id_equipo_l) AS perdidos_local NATURAL JOIN (SELECT id_equipo AS mi_equipo, COUNT(id_equipo_l) AS empatados_l
                            FROM equipo LEFT OUTER JOIN (SELECT *
                                        FROM partido
                                        WHERE marcador_l = marcador_v) AS tmp_empatados ON equipo.id_equipo = tmp_empatados.id_equipo_l
                                        GROUP BY id_equipo) AS empatados_local NATURAL JOIN (SELECT id_equipo_l AS mi_equipo, SUM(marcador_l) AS puntos_anotados_l
                                                    FROM partido
                                                    GROUP BY id_equipo_l) AS puntos_anotados_local NATURAL JOIN (SELECT id_equipo_l AS mi_equipo, SUM(marcador_v) AS puntos_recibidos_l
                                                                FROM partido
                                                                GROUP BY id_equipo_l) AS puntos_recibidos_local) AS concentrado_local NATURAL JOIN (
SELECT mi_equipo, ganados_v, perdidos_v, empatados_v, puntos_anotados_v, puntos_recibidos_v
FROM (SELECT id_equipo_v AS mi_equipo, COUNT(id_equipo_v) AS ganados_v
      FROM partido
      WHERE marcador_v > marcador_l
      GROUP BY id_equipo_v) AS ganados_visitante NATURAL JOIN (SELECT id_equipo_v AS mi_equipo, COUNT(id_equipo_v) AS perdidos_v
                  FROM partido
                  WHERE marcador_v < marcador_l
                  GROUP BY id_equipo_v) AS perdidos_visitante NATURAL JOIN (SELECT id_equipo AS mi_equipo, COUNT(id_equipo_v) AS empatados_v
                              FROM equipo LEFT OUTER JOIN (SELECT *
                                          FROM partido
                                          WHERE marcador_l = marcador_v) AS tmp_empatados ON equipo.id_equipo = tmp_empatados.id_equipo_v
                                          GROUP BY id_equipo) AS empatados_visitante NATURAL JOIN (SELECT id_equipo_v AS mi_equipo, SUM(marcador_v) AS puntos_anotados_v
                                                      FROM partido
                                                      GROUP BY id_equipo_v) AS puntos_anotados_visitante NATURAL JOIN (SELECT id_equipo_v AS mi_equipo, SUM(marcador_l) AS puntos_recibidos_v
                                                                  FROM partido
                                                                  GROUP BY id_equipo_v) AS puntos_recibidos_visitante) AS concentrado_visitante;
```

7 | Práctica K - Introducción al software KNIME

7.1. Objetivos

- Que el alumno conozca y se familiarice con el software KNIME.
- Describir de forma clara, sintetizada y visual elementos básicos del software KNIME.
- Proporcionar ejemplos del uso de KNIME con bases de datos.

7.2. Introducción

KNIME Analytics Platform es un software de código libre *(open source)* desarrollado sobre la plataforma de Eclipse[1], programado en Java. Es un programa multiplataforma, por lo que puede ser utilizado en cualquier sistema operativo (Linux, Windows, Mac).

KNIME tiene una interfaz visual que es libre de código, es decir, no se basa en *scripts* sino en **nodos**: elementos que contienen algoritmos y se conectan entre sí para el procesamiento y análisis de datos en secuencia.

Este software es rápido de implementar e intuitivo para aprender, tiene la capacidad de explorar y expandir rápidamente grandes volúmenes de datos.

Por otra parte, KNIME tiene la facultad de incorporar código implementado en R o Python y también puede conectarse con Sistemas Manejadores de Bases de Datos como MySQL, SQL Server, SQLite y PostgreSQL.

El espacio de trabajo de KNIME se compone del *Explorador KNIME*, el *Editor del Flujo de trabajo*, el *Repositorio de Nodos*, la *Descripción del Nodo*, la *Consola*, y el panel de *Outline* como se muestra en la imagen siguiente:

[1]Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por si misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada (https://goo.gl/ub00Af).

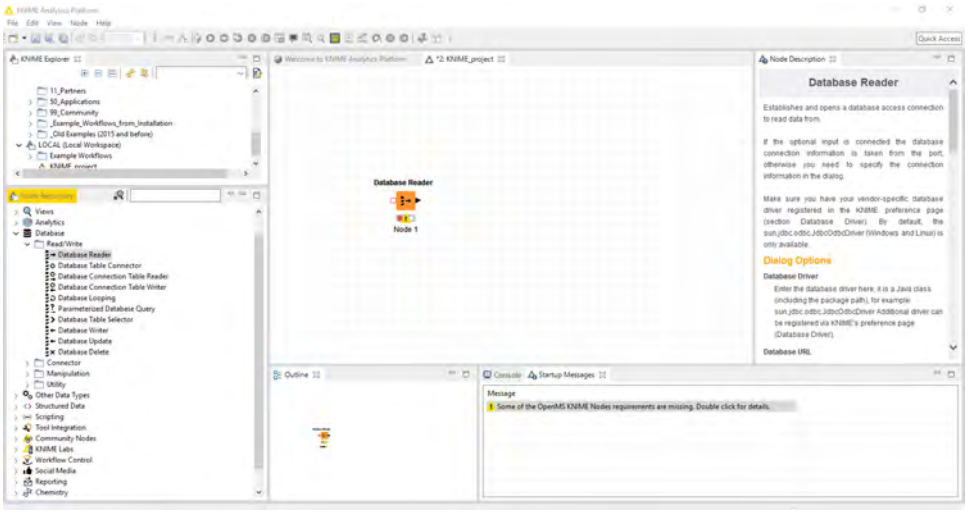


Figura 7.1: Software KNIME.

El repositorio o almacén de nodos contiene cada uno de los nodos que pueden utilizarse. Estos están organizados de manera clasificada, dependiendo del tipo de aportación que pueda hacerse a través de sus algoritmos para el análisis de datos, ya sea manipulación de filas o columnas en una base una base de datos, gráficas descriptivas, técnicas de análisis multivariado como cluster o componentes principales y creación de modelos como regresiones.

De esta manera el nodo deseado puede arrastrase desde el *Repositorio* hasta el panel del *Flujo de Trabajo* donde puede configurarse (con F6 o click derecho sobre el mismo), además, al seleccionar cualquier nodo, en el panel de *Descripción* aparecerá la descripción del nodo con sus opciones de parámetros. Por otra parte, la *consola* mostrará los errores de ejecución en caso de haberlos.

Los atributos de un nodo son:

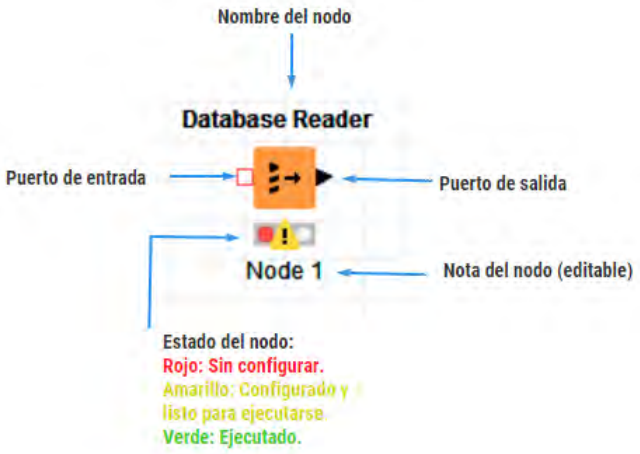


Figura 7.2: Nodos.

La posible desventaja de KNIME es que al tener poca difusión puede que no se encuentren foros de soporte técnico o ejemplos de cada una de las herramientas que KNIME ofrece, como sucede con otros programas más conocidos. Sin embargo, la siguiente infografía o *"acordeón"* puede ayudar a darse una idea de los nodos básicos y sus aplicaciones.

Figura 7.3: Almacén de nodos.

7.3. Iniciando un proyecto con KNIME

Se empezará creando un nuevo proyecto o flujo de trabajo de la siguiente manera: sobre la barra de tareas, se tendrá que seguir la siguiente secuencia de pasos, ir al menú File, elegir las opciones *File − > New − > New KNIME Workflow*, donde se le pondrá nombre al proyecto tal como se muestra a continuación.

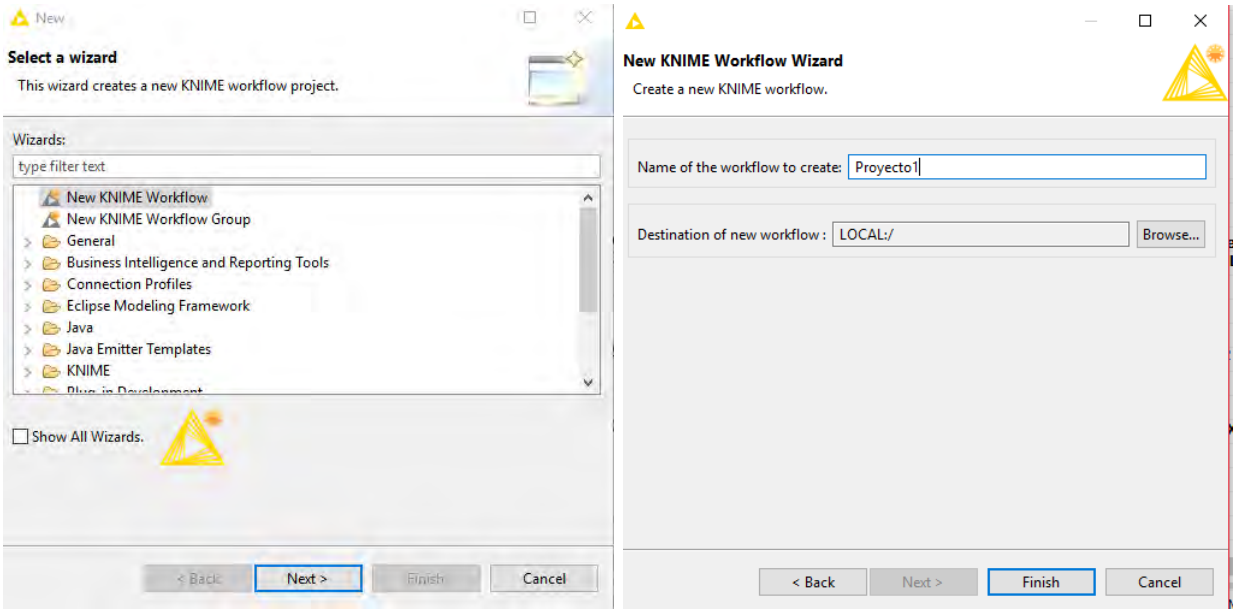


Figura 7.4: Proyecto nuevo.

Para introducirse en el programa, se utilizarán ejemplos sencillos de la manipulación de nodos de cada categoría: Acceso, Manipulación, Análisis, Visualización y Despliegue a través del siguiente flujo de trabajo, el cuál es el resultado final de cada uno de los nodos que se explicarán a lo largo de la práctica:

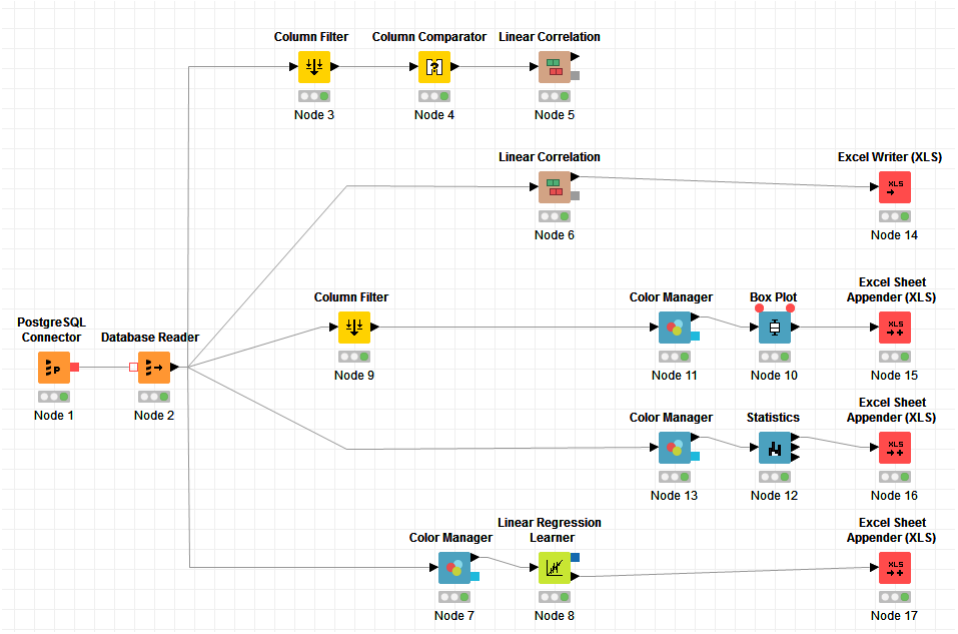


Figura 7.5: Flujo de trabajo.

7.3.1. Nodos de acceso.

Los nodos de **acceso** permiten importar y leer datos de bases de datos en diferentes formatos para su manipulación así como crear conexiones con distintos Sistemas Manejadores de Bases de Datos mediante las cuales se pueden extraer tablas mediante consultas con lenguaje SQL.

Se empezará con una conexión entre PostgreSQL y KNIME, utilizando el nodo *PostreSQL Connector* y el nodo que permitirá leer y manejar las tablas que se desee: *Database Reader*.

Posteriormente se configurará el nodo 1 proporcionando los datos nombre del Host, nombre de la base de datos a la queremos tener acceso, el usuario y la contraseña. Cuando el nodo ya está configurado, el color del estado será amarillo y puede ser ejecutado inmediatamente si se da click derecho sobre el nodo ya configurado y seleccionando la opción *execute* o presionando F7.

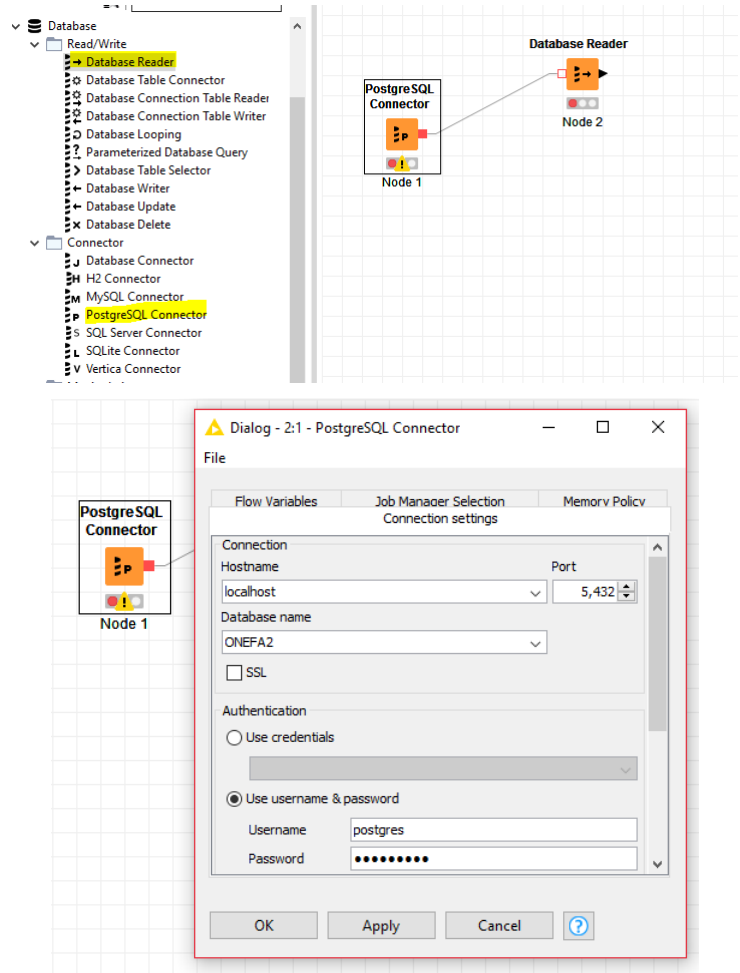


Figura 7.6: Nodo 1, selección y configuración.

Ya que el nodo está ejecutado se puede configurar el nodo 2, donde se usará la consulta de la práctica U. Al ejecutarse se puede dar click derecho sobre el nodo y elegir la opción de *Data from Database* para mostrar la salida del algoritmo ejecutado.

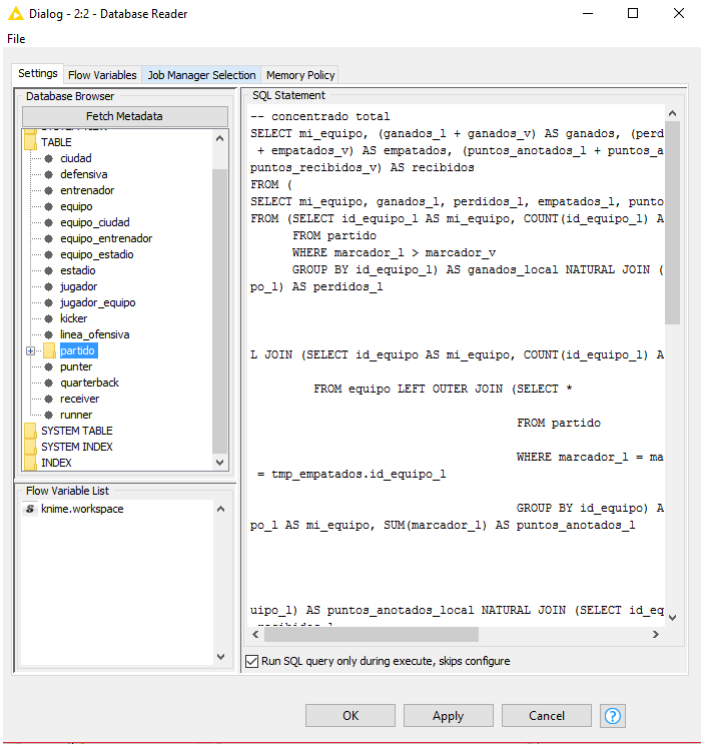
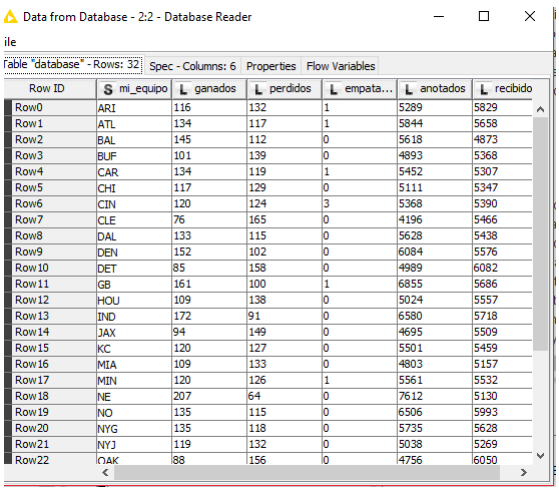


Figura 7.7: Nodo 2, configuración.



Row ID	mi_equipo	ganados	perdidos	empata...	anotados	recibido
Row0	ARI	116	132	1	5289	5829
Row1	ATL	134	117	1	5844	5658
Row2	BAL	145	112	0	5618	4873
Row3	BUF	101	139	0	4893	5368
Row4	CAR	134	119	1	5452	5307
Row5	CHI	117	129	0	5111	5347
Row6	CIN	120	124	3	5368	5390
Row7	CLE	76	165	0	4196	5466
Row8	DAL	133	115	0	5628	5438
Row9	DEN	152	102	0	6084	5576
Row10	DET	85	158	0	4989	6082
Row11	GB	161	100	1	6855	5686
Row12	HOU	109	138	0	5024	5557
Row13	IND	172	91	0	6580	5718
Row14	JAX	94	149	0	4695	5509
Row15	KC	120	127	0	5501	5459
Row16	MIA	109	133	0	4803	5157
Row17	MIN	120	126	1	5561	5532
Row18	NE	207	64	0	7612	5130
Row19	NO	135	115	0	6506	5993
Row20	NYG	135	118	0	5735	5628
Row21	NYJ	119	132	0	5038	5269
Row22	OAK	88	156	0	4756	6050

Figura 7.8: Nodo 2, resultado.

7.3.2. Nodos de manipulación.

Hasta ahora se han utilizado nodos de acceso, a continuación se utilizarán nodos de **manipulación**. ¿Qué puede hacerse con ellos? Este tipo de nodos permiten manejar y transformar las tablas que se hayan leído a través de los nodos de acceso, de manera que pueden hacerse filtros, cambios en los tipos de variables, renombre de las mismas así como cambio de datos de una columna o renglón, agrupaciones, entre otras cosas.

Dentro de esta práctica se utilizará en primera instancia el nodo de "Column Filter" (nodo 3), la configuración permite elegir las variables columna por las que se quiere hacer el filtro, se eligen las variables *id_ equipo, anotados y recibidos* y puede verse el resultado como en el nodo anterior.

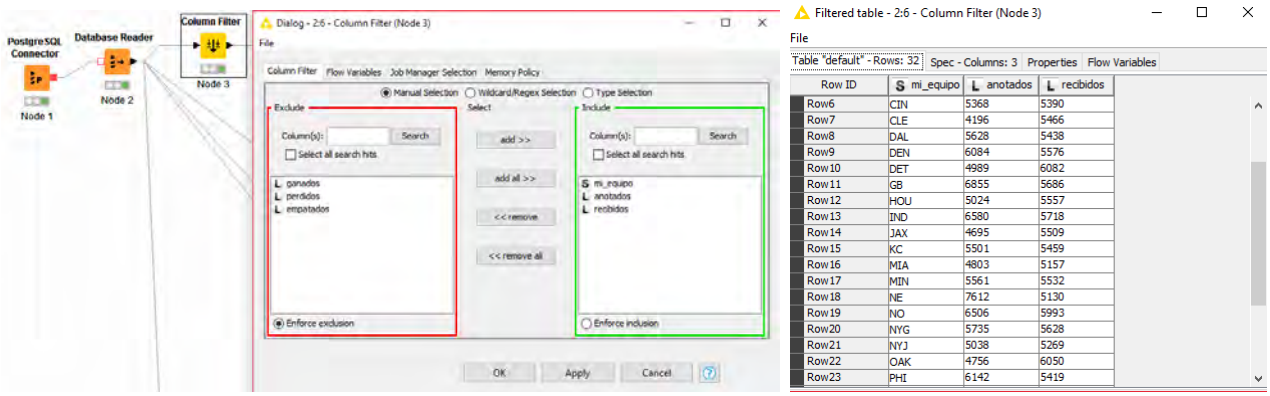


Figura 7.9: Nodo 3, configuración y resultado.

Otro ejemplo de nodo de manipulación es el de *Column Comparator*, supongamos que queremos asignar a cada equipo las variables categóricas "mal equipo" y "buen equipo" ¿cuál sería una posible forma de hacerlo? para este caso, se usará el filtro hecho anteriormente y se conectará con el nuevo nodo.

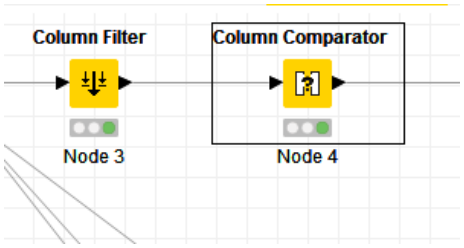


Figura 7.10: Unión de nodo 4.

La configuración de este nodo se basa en seleccionar las columnas que se compararán y el operador que se requiera. La condición para comparar, en este caso, será que los puntos recibidos sean mayores que los puntos anotados; si el operador resulta *verdadero* se puede elegir una opción para agregar en una columna nueva.

En este ejemplo, se utilizará la opción definida por el usuario: “buen equipo” si la condición es verdadera y “mal equipo” si es falsa. A continuación se muestra la ventana de configuración y el resultado del nodo ejecutado.

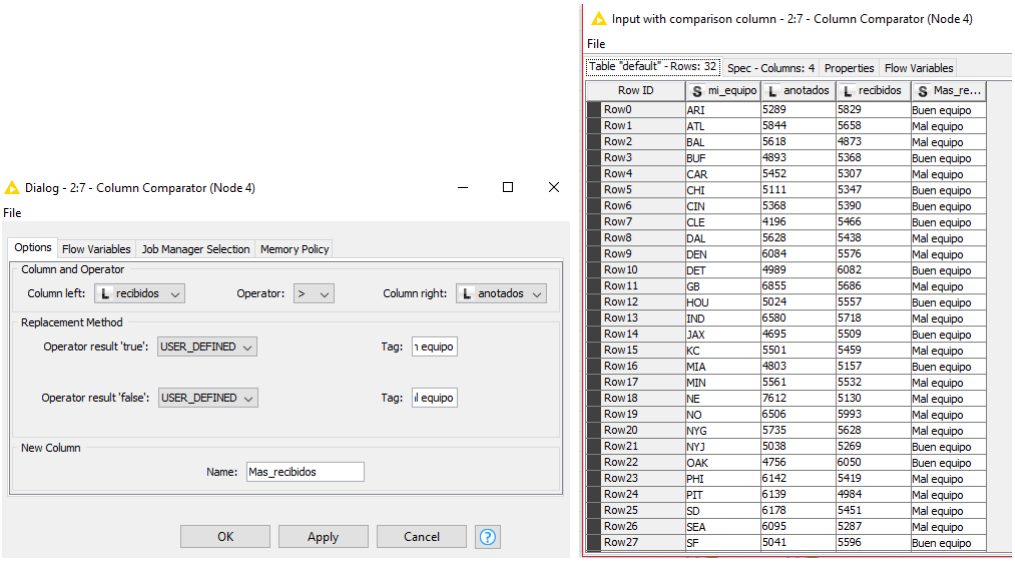


Figura 7.11: Nodo 4, configuración y resultado.

7.3.3. Nodos de análisis.

Una vez ilustrados los nodos de manipulación, continuamos con los nodos de **análisis**. Estos contienen algoritmos con diferentes técnicas de estadística descriptiva, inferencial, no paramétrica y bayesiana. También contienen técnicas de análisis multivariado, muestreo, teoría de decisiones, así como modelos probabilísticos.

Hasta el momento se han utilizado dos nodos de manipulación; ahora, se explicarán dos ejemplos de nodos de análisis: *“Linear Correlation”* y *“Linear Regression Learner”*.

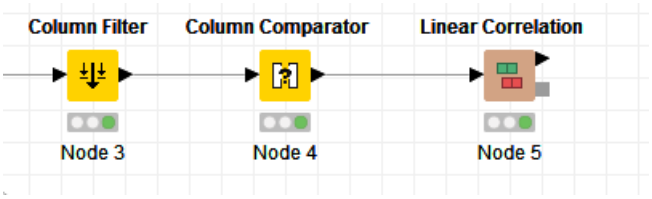


Figura 7.12: Unión de nodo 5.

Linear Correlation calcula para cada par de variables columna seleccionadas el coeficiente de correlación. A continuación se muestra su configuración al elegir las columnas *anotados* y *recibidos*, así como el resultado obtenido:

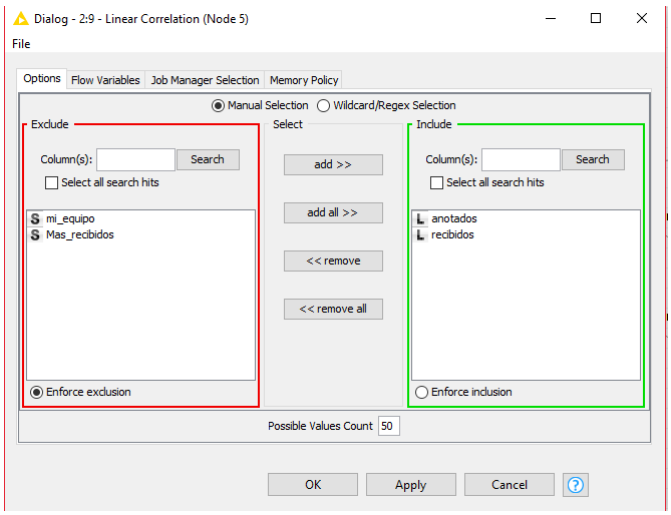


Figura 7.13: Nodo 5, configuración.

Row ID	anotados	recibidos
anotados	1	-0.132
recibidos	-0.132	1

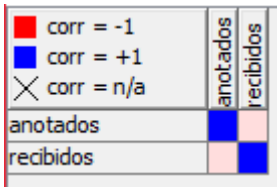


Figura 7.14: Nodo 5, resultado.

Este nodo de análisis tiene dos resultados, uno de ellos es la tabla numérica del coeficiente de correlación y una tabla que indica la fuerza de la relación lineal. Si el color es rojo indica que hay una correlación lineal negativa alta entre las variables, si es un poco mas claro indica una correlación negativa débil y análogamente para el color azul y una correlación lineal positiva.

De esta manera se puede observar que no hay correlación lineal entre las dos variables elegidas. ¿Qué hay de las demás variables? Se utilizará el mismo nodo para hacer el análisis de todas las variables, es decir, se evitará el filtro y se configurará de igual manera:

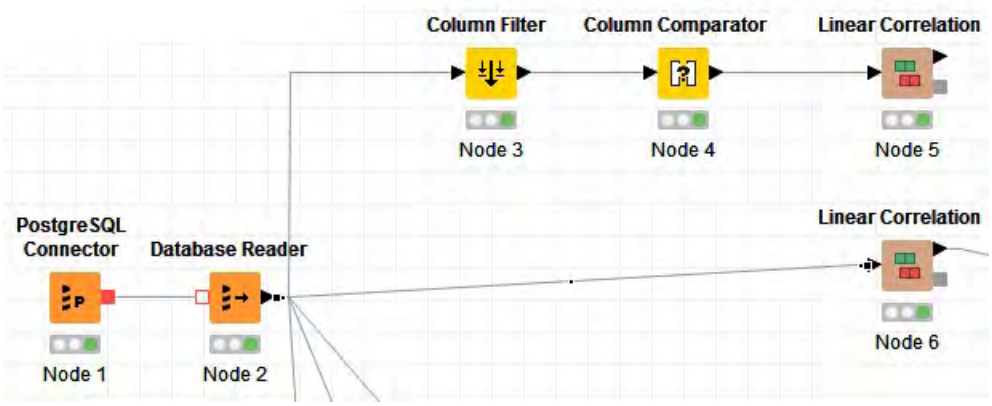


Figura 7.15: Unión de nodo 8.

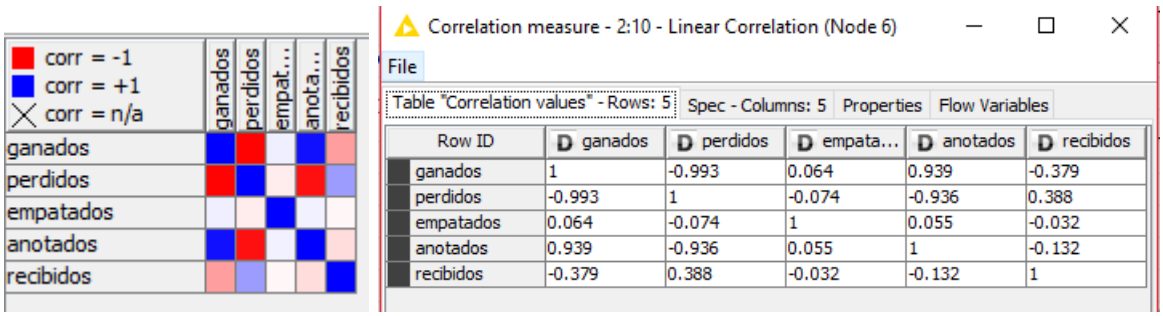


Figura 7.16: Nodo 6, resultados.

Se puede apreciar como es que hay correlación lineal negativa entre partidos perdidos y ganados, de igual manera entre partidos perdidos y puntos anotados. Por otra parte, existe correlación lineal positiva entre partidos ganados y puntos anotados. Esta última conclusión permite que tenga sentido utilizar un nuevo nodo de análisis: *"Linear Regression Learner"*.

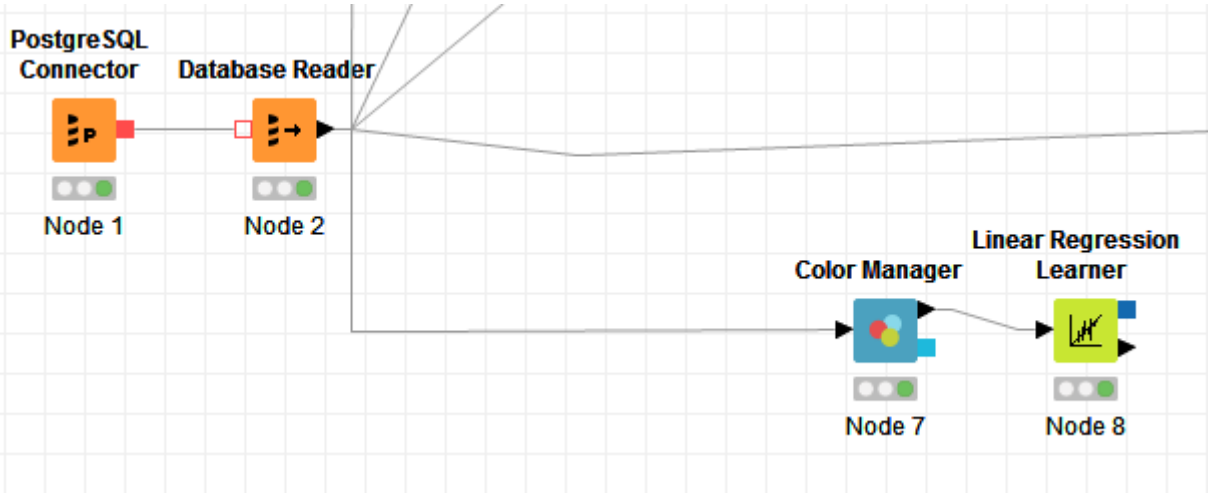


Figura 7.17: Unión de nodo 8.

Este nodo genera una regresión lineal de la base de datos (podría configurarse para ser una regresión multivariada). Para este ejemplo, sabiendo previamente de la correlación lineal entre partidos ganados y puntos anotados, se elegirán las dos variables relacionadas, poniendo como variable "respuesta" los puntos anotados. Además, se une el nodo a uno que permite dar colores a lo gráficos por valores nominales o columnas numéricas (nodo 7).

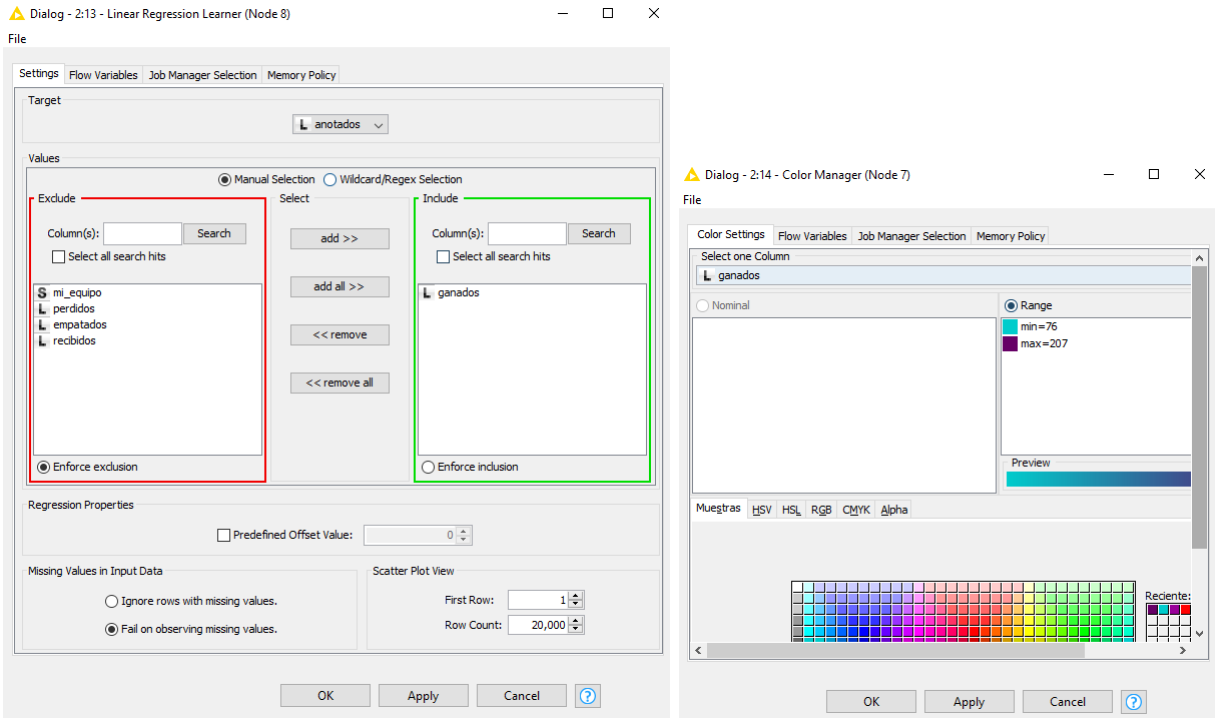


Figura 7.18: Nodo 7 y 8, configuración.

Ejecutando ambos nodos se obtiene la siguiente regresión lineal y los parámetros correspondientes:

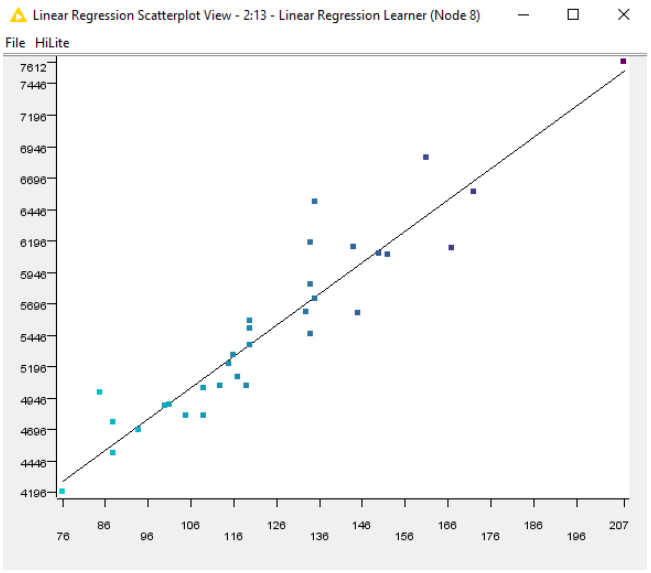


Figura 7.19: Nodo 8, resultados.

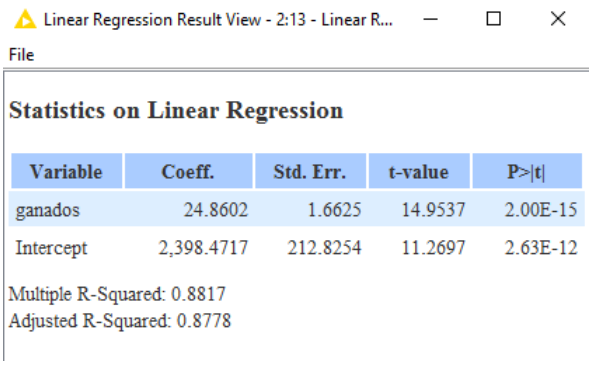


Figura 7.20: Nodo 8, resultados.

El modelo es: y=24.86x+2398.47 donde y es la variable que representa a los puntos anotados y x a los partidos ganados. El modelo tiene R^2 de .8, por lo que, la calidad del modelo es muy bueno.

7.3.4. Nodos de visualización.

Habiendo utilizado nodos de análisis, se proseguirá a ejemplificar los nodos de **visualización**. Estos nodos permiten reflejar los datos de las tablas o bases de manera gráfica. Se utilizarán "Box plot" y "Statistics" para ejemplificar.

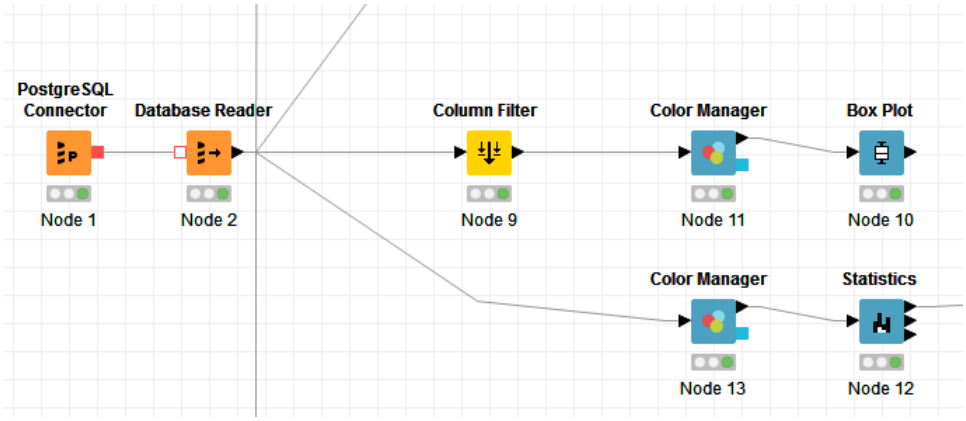


Figura 7.21: Unión de nodos 10 y 12.

El nodo de "Box plot" proporciona el algoritmo para obtener el gráfico de caja con las medidas respectivas. Para utilizarlo se decidió usar un nodo de manipulación creando un filtro con las variables 'Ganados y Perdidos" y el nodo de visualización puede ejecutarse con una configuración por *default*. De manera que el resultado es la siguiente gráfica:

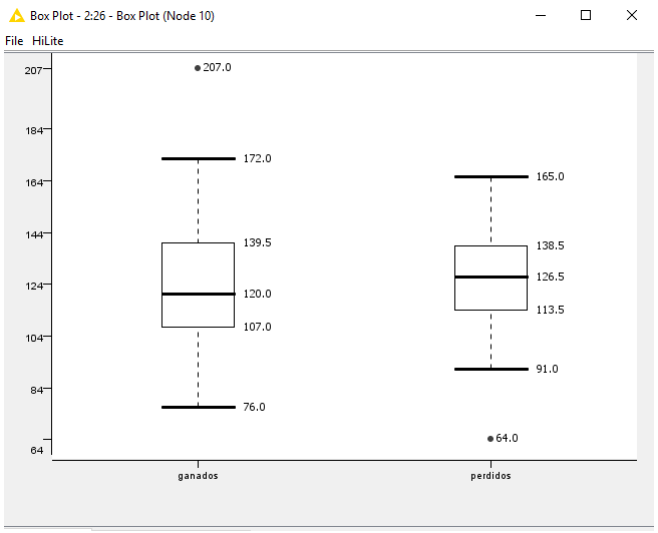


Figura 7.22: Nodo 10, resultado.

Como se ha explicado en la práctica U, la manera de interpretar este gráfico es la siguiente: el mínimo de partidos ganados es de 76, en promedio cada equipo ganó en el total de las temporadas 120 juegos y existe un equipo que tiene un dato atípico, es decir, el número de sus partidos ganados se aleja mucho del promedio que han tenido los demás equipos, con un total de 207 victorias. Análogamente para la gráfica de caja de partidos perdidos.

Por otra parte, el nodo "Statistics" puede dar de otra manera el resumen de toda la tabla para el total de las variables, para mostrarlo se une directamente al nodo 2 que contiene la consulta hecha. La configuración se basa en elegir las variables para las cuales se quiera hacer el análisis, ya que la variables id_equipo no es numérica, se excluye del análisis.

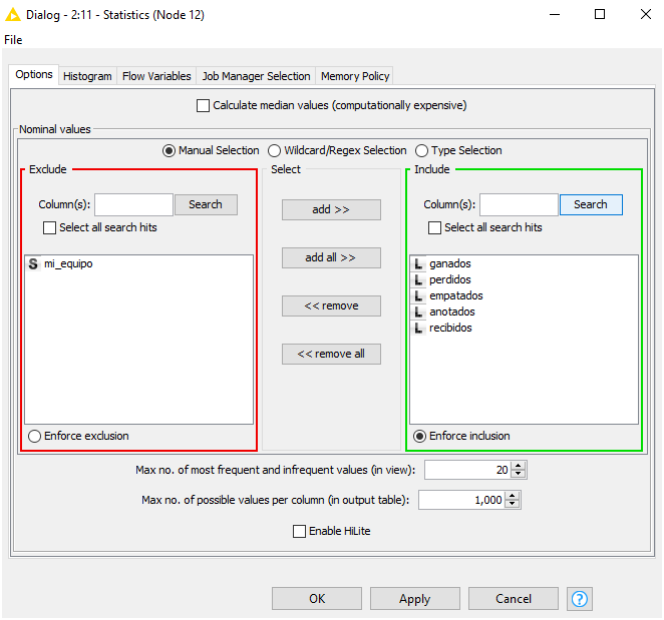


Figura 7.23: Nodo 12, configuración.

Se puede apreciar que el resumen de la información por variables incluye además un histograma.

File

Numeric | Nominal | Top/bottom

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
ganados	76	124.9375	?	207	28.3571	0.713	0.9438	0	0	0	76 207
perdidos	64	124.9375	?	165	21.5106	-0.5075	0.8674	0	0	0	64 165
empatados	0.0	0.4375	?	3	0.669	1.966	5.4721	0	0	0	0 3
anotados	4,196	5,504.4375	?	7,612	750.7631	0.7695	0.6144	0	0	0	4.196 7.612
recibidos	4,873	5,504.4375	?	6,082	283.7231	0.0608	0.1227	0	0	0	4.873 6.082

Figura 7.24: Nodo 12, resultado.

7.3.5. Nodos de despliegue.

Los nodos de despliegue o implementación permiten exportar, sobrescribir en las tablas o guardar información en distintos formatos.

Para ejemplificar su uso, se guardarán los datos de los dos análisis hechos (Regresión lineal y correlación) así como los datos de los nodos de visualización. De esta manera se conectan 4 nodos nuevos:

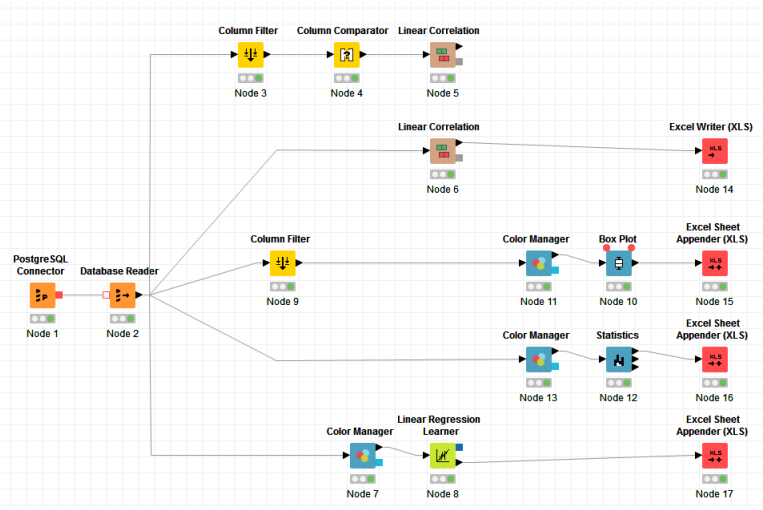


Figura 7.25: Unión de nodos 14, 15, 16 y 17.

El nodo 14 (“Excel Writer”) permite crear un archivo de Excel con los datos numéricos del análisis, su configuración se basa en dar la ruta donde se desea guardar el archivo, el nombre y condiciones como: si se quiere sobrescribir sobre alguno existente, si se quiere tener los encabezados de columnas y renglones o la información de qué variables se quiere guardar. A continuación se muestra su configuración:

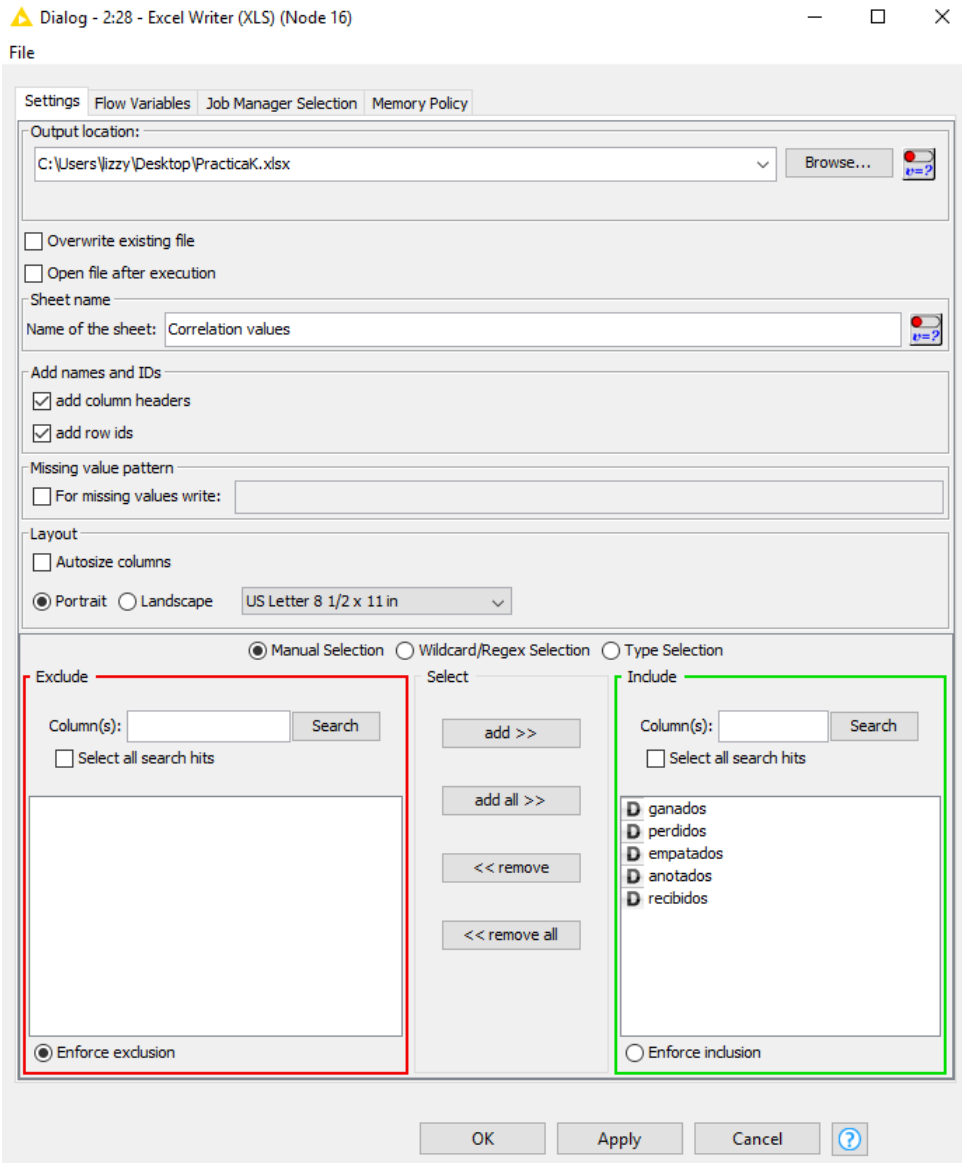


Figura 7.26: Nodo 14, configuración.

El resultado será la creación de un nuevo archivo, con los datos y la hoja renombrada como se indicó.

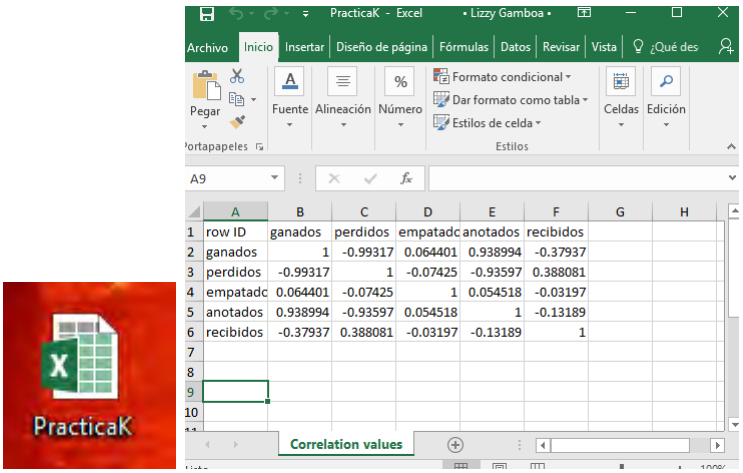


Figura 7.27: Nodo 14, resultado.

Posterior a la creación del archivo, el nodo “Excel Sheet Appender (XLS)” coloca la información requerida en una hoja específica de un libro en Excel ya existente. La configuración es similar al nodo anterior, siendo la única excepción no dar un nombre nuevo al archivo sino seleccionar uno ya existente.

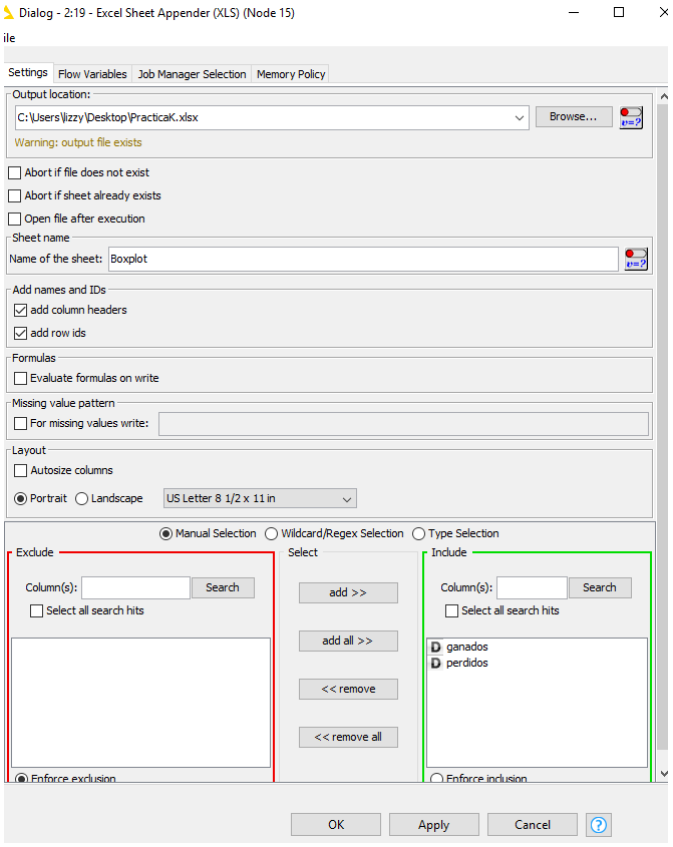


Figura 7.28: Nodo 15, configuración.

De la misma manera para los nodos 16 y 17, obteniendo el siguiente resultado:

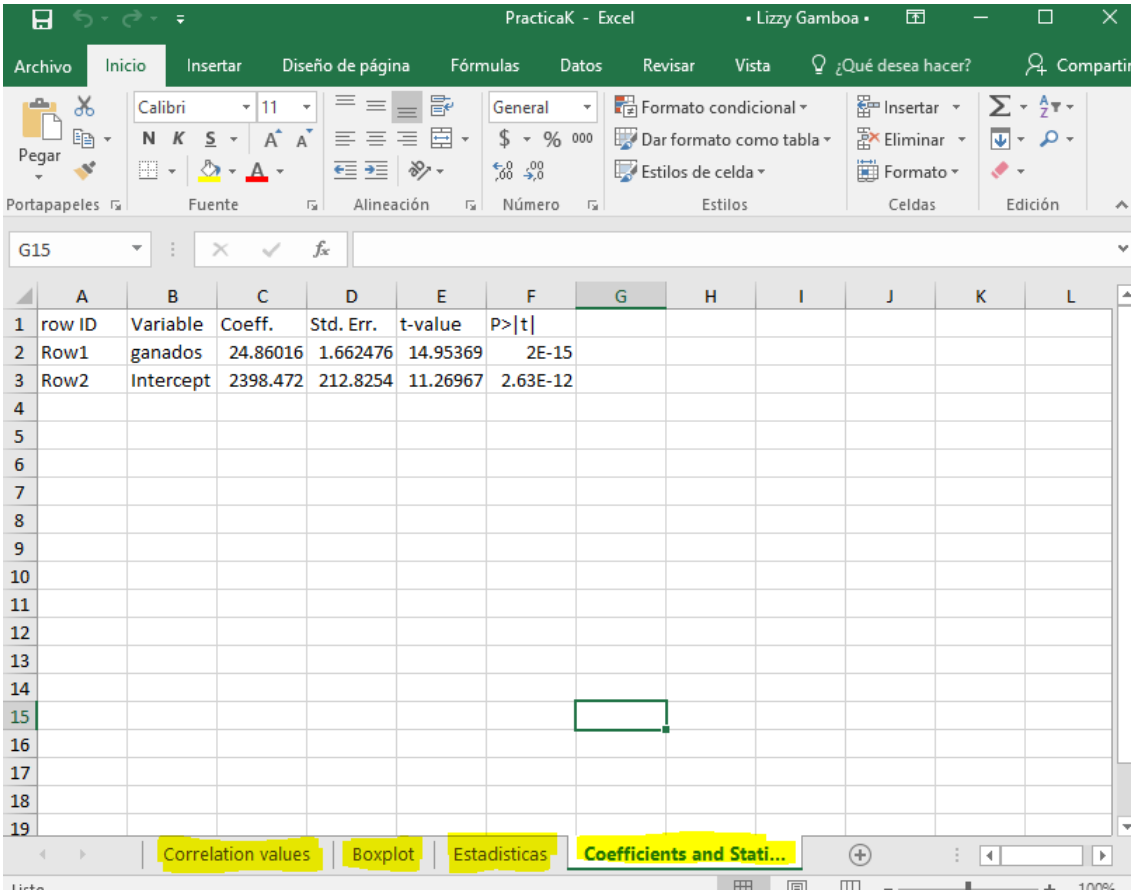


Figura 7.29: Nodos 14, 15, 16 y 17, resultado.

Se puede apreciar que los datos de la regresión lineal fueron guardados en la hoja nombrada *Coefficients and Statistics*. Desafortunadamente KNIME aún no cuenta con una manera de exportar las imágenes, cuando se guardan se hacen como *screenshot*.

Como se puede apreciar, el trabajo dentro de la plataforma KNIME es fluido e intuitivo, si se quisiera hacer análisis más complejo de bases de datos, bastaría con conocer las características y necesidades de entrada de cada nodo, por lo que se puede agradecer a los creadores la buena documentación en la parte de “Node Description”.

7.4. Ejercicios

- Establezca la conexión de Knime con PostgreSQL:

  • Realice dos consultas (ciudad con mayor cantidad de equipos asociados, nombre de los entrenadores que han participado en todos los equipo).

  • Utilice un nodo de visualización distinto a los vistos en la práctica e interprete.

- Asigne una calificación a cada jugador de Punter dependiendo de sus estadísticas en la variable punts (superestrella/élite/regular/malo/...).

  Hint: Utilice un nodo tipo Binner.

- Responda ¿qué nodo usaría si requiere buscar duplicados en la tabla Jugador?

- Genere salidas a archivos Excel de las 3 consultas que realizó así como de sus resultados.

- Grafique los *touchdowns* (tds) lanzados por los “Quarterbacks". Elija la gráfica que considere más adecuada.

8 | Práctica L - Limpieza y análisis de valores perdidos con KNIME

8.1. Objetivos

- Identificar posibles herramientas de KNIME para limpieza de bases de datos.
- Ejemplificar el uso de los nodos básicos para análisis de valores perdidos.

8.2. Introducción

Al hacer análisis estadístico, la mayor parte del tiempo se trabajará con bases de datos provenientes de diferentes experimentos y encuestas. Lo anterior implica tratar con tasas de no respuesta, es decir, cierto porcentaje de información que no pudo obtenerse. En otros casos, por el mal diseño de una base datos o por razones variadas, se presentarán faltantes de información que podrán ser tratados con técnicas de imputación. Por otro lado, en variadas ocasiones también tendrá que tratarse con bases de datos "sucias", es decir, bases de datos que contienen información repetida o mal categorizada.

Lo anterior representa un reto importante para el análisis y modelado de datos, ya que si no se cuenta con la información necesaria, puede hacerse un mal análisis de los datos y en consecuencia obtener resultados no veraces.

En esta práctica se mostrará cómo usar el software KNIME para detectar los llamados *missing values* o valores faltantes. El criterio para tratar con ellos dependerá de la persona a cargo del análisis, del contexto y las técnicas estadísticas conocidas.

8.3. Limpieza

Se utilizará una base de datos reducida de la obtenida en la práctica S, con solo 25 partidos y por ende, 50 registros. Para manipularla dentro de KNIME se utilizará el nodo de acceso *CSV Reader*.

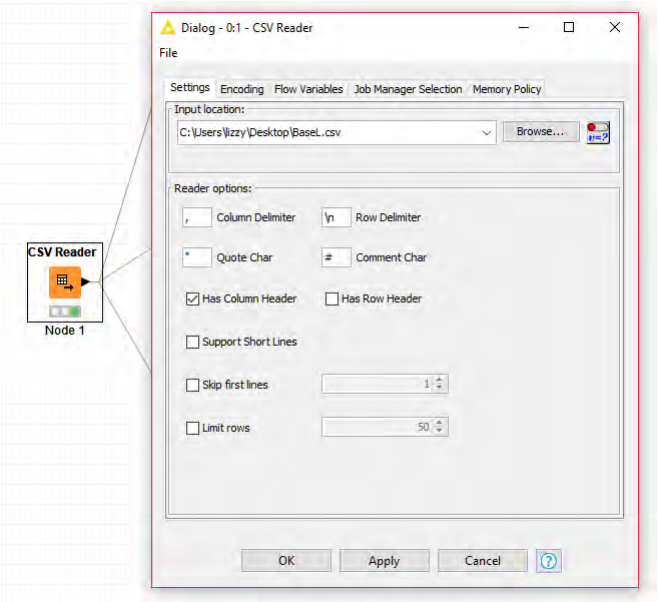
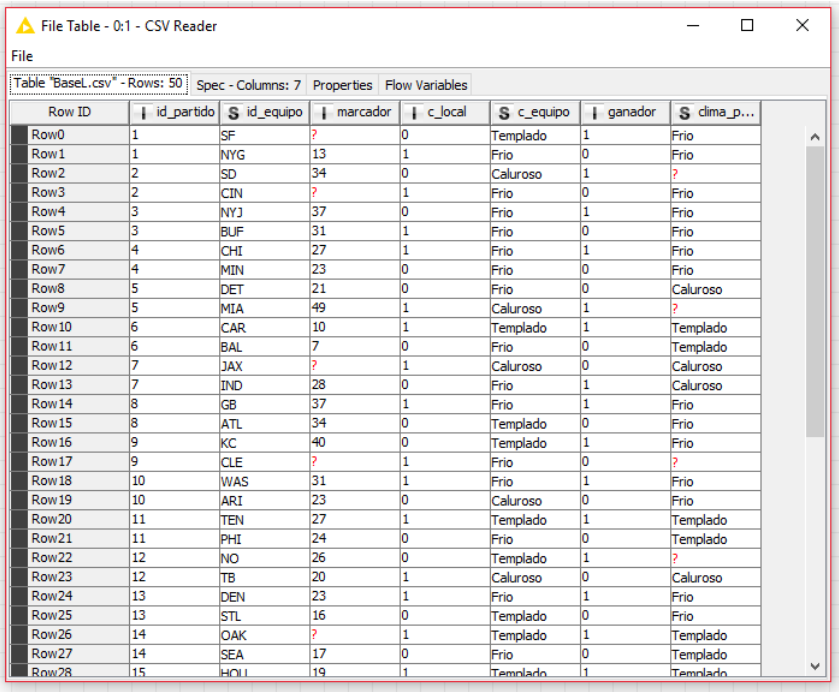


Figura 8.1: Nodo 1, configuración.

De esta manera se puede leer la base de datos como se muestra a continuación. Puede apreciarse la falta de datos indicado por un ? en color rojo en las variables marcador y clima_partido.



Row ID	id_partido	id_equipo	marcador	c_local	c_equipo	ganador	clima_p...
Row0	1	SF	?	0	Templado	1	Frio
Row1	1	NYG	13	1	Frio	0	Frio
Row2	2	SD	34	0	Caluroso	1	?
Row3	2	CIN	?	1	Frio	0	Frio
Row4	3	NYJ	37	0	Frio	1	Frio
Row5	3	BUF	31	1	Frio	0	Frio
Row6	4	CHI	27	1	Frio	1	Frio
Row7	4	MIN	23	0	Frio	0	Frio
Row8	5	DET	21	0	Frio	0	Caluroso
Row9	5	MIA	49	1	Caluroso	1	?
Row10	6	CAR	10	1	Templado	1	Templado
Row11	6	BAL	7	0	Frio	0	Templado
Row12	7	JAX	?	1	Caluroso	0	Caluroso
Row13	7	IND	28	0	Frio	1	Caluroso
Row14	8	GB	37	1	Frio	1	Frio
Row15	8	ATL	34	0	Templado	0	Frio
Row16	9	KC	40	0	Templado	1	Frio
Row17	9	CLE	?	1	Frio	0	?
Row18	10	WAS	31	1	Frio	1	Frio
Row19	10	ARI	23	0	Caluroso	0	Frio
Row20	11	TEN	27	1	Templado	1	Templado
Row21	11	PHI	24	0	Frio	0	Templado
Row22	12	NO	26	0	Templado	1	?
Row23	12	TB	20	1	Caluroso	0	Caluroso
Row24	13	DEN	23	1	Frio	1	Frio
Row25	13	STL	16	0	Templado	0	Frio
Row26	14	OAK	?	1	Templado	1	Templado
Row27	14	SEA	17	0	Frio	0	Templado
Row28	15	HOU	19	1	Templado	1	Templado

Figura 8.2: Nodo 1, resultados.

El primer nodo a utilizar para el análisis de valores perdidos es un nodo de manipulación llamado *Missing Value Column Filter*, este nodo elimina las columnas de la base de datos que contengan más valores faltantes que un cierto porcentaje dado. Es posible configurar, por cada columna, el porcentaje de valores faltantes que permitirá decidir si la columna se filtra o no.

Ya que la opinión sobre el porcetaje permitido de valores faltantes en una base de datos varía respecto al autor, se ejemplificará para las dos opciones usuales: 5 % y 10 %.

El Nodo 2 filtrará todas las variables con un porcentaje permitido de 10 % por variable.

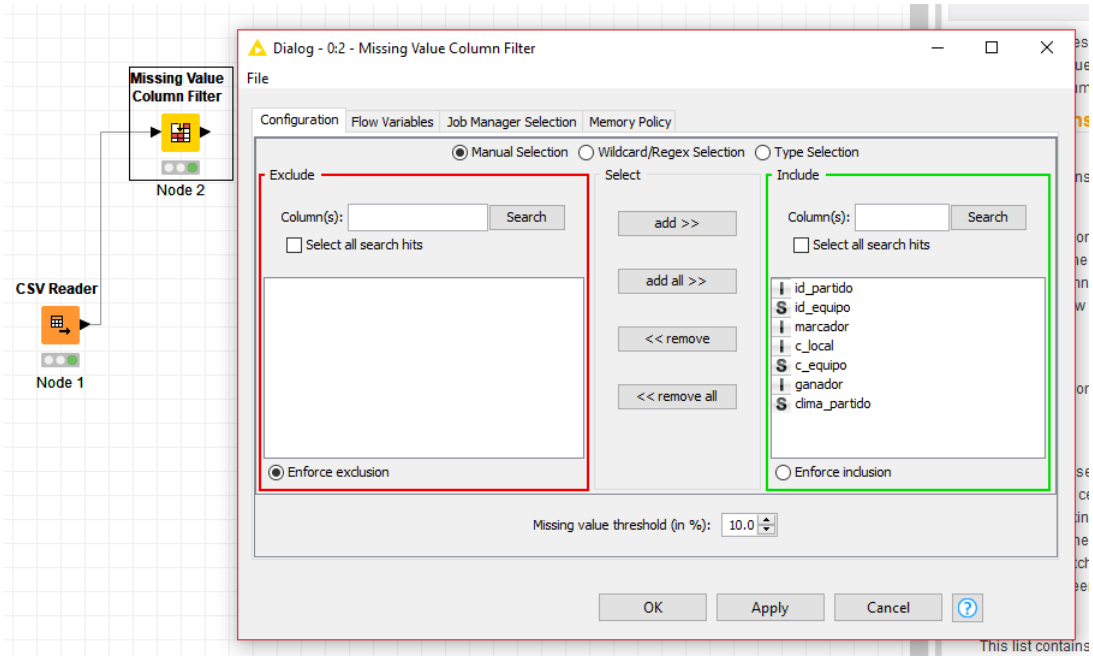
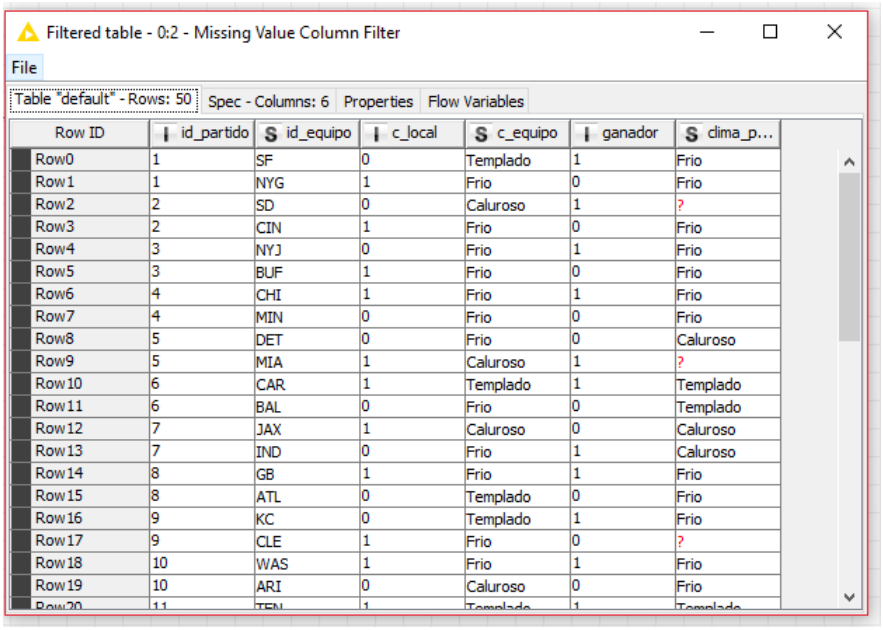


Figura 8.3: Nodo 2, configuración.



Row ID	id_partido	id_equipo	c_local	c_equipo	ganador	clima_p...
Row0	1	SF	0	Templado	1	Frio
Row1	1	NYG	1	Frio	0	Frio
Row2	2	SD	0	Caluroso	1	?
Row3	2	CIN	1	Frio	0	Frio
Row4	3	NYJ	0	Frio	1	Frio
Row5	3	BUF	1	Frio	0	Frio
Row6	4	CHI	1	Frio	1	Frio
Row7	4	MIN	0	Frio	0	Frio
Row8	5	DET	0	Frio	0	Caluroso
Row9	5	MIA	1	Caluroso	1	?
Row10	6	CAR	1	Templado	1	Templado
Row11	6	BAL	0	Frio	0	Templado
Row12	7	JAX	1	Caluroso	0	Caluroso
Row13	7	IND	0	Frio	1	Caluroso
Row14	8	GB	1	Frio	1	Frio
Row15	8	ATL	0	Templado	0	Frio
Row16	9	KC	0	Templado	1	Frio
Row17	9	CLE	1	Frio	0	?
Row18	10	WAS	1	Frio	1	Frio
Row19	10	ARI	0	Caluroso	0	Frio

Figura 8.4: Nodo 2, resultado.

Como puede observarse, la variable marcador no se muestra en el resultado, por lo tanto esta columna tiene más del 10 % de valores faltantes. Sin embargo, la variable clima_partido sigue apareciendo.

A continuación, configuramos el Nodo 3 bajo el supuesto que el porcentaje permitido de valores faltantes es del 5 %.

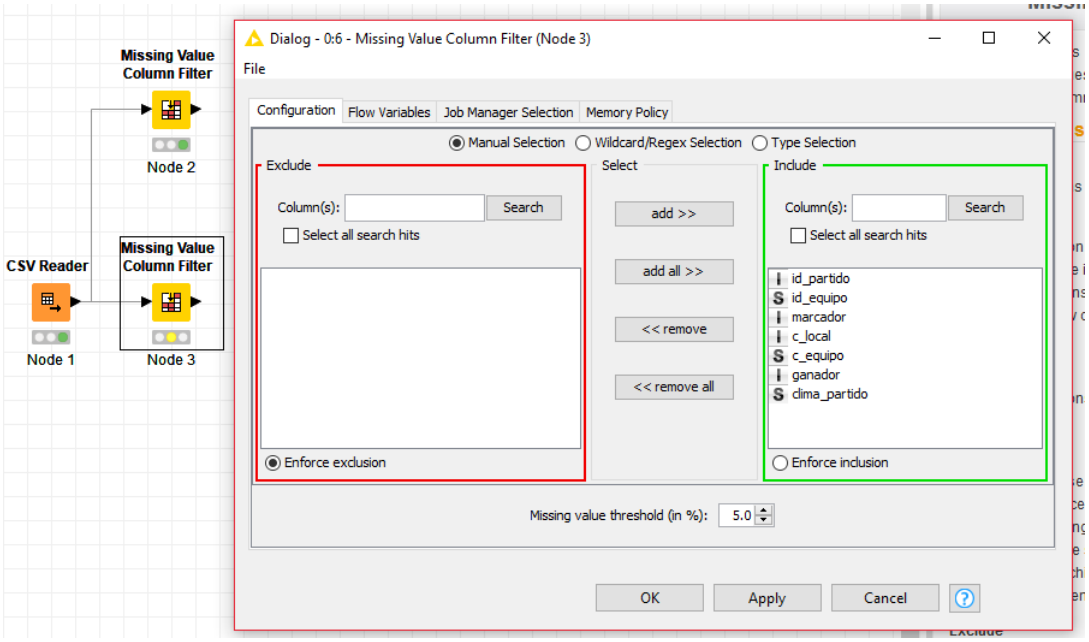
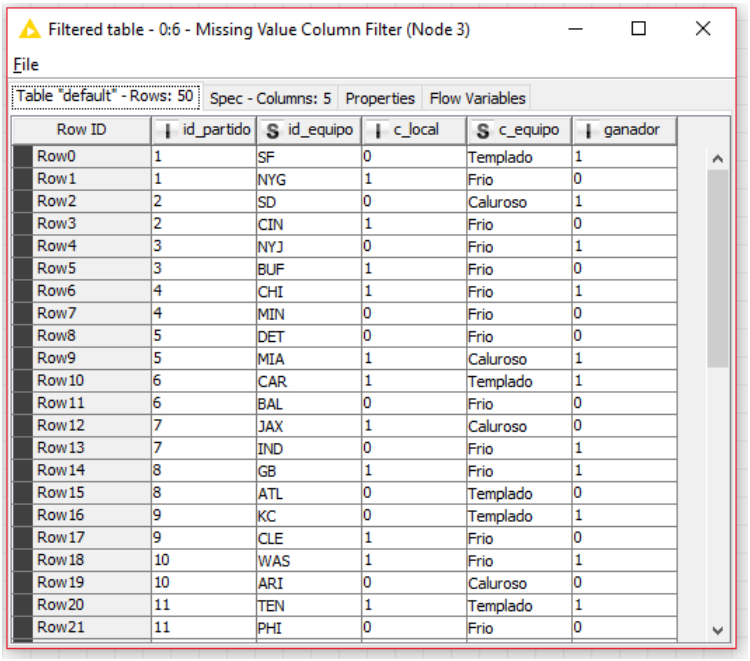


Figura 8.5: Nodo 3, configuración.



Row ID	id_partido	id_equipo	c_local	c_equipo	ganador
Row0	1	SF	0	Templado	1
Row1	1	NYG	1	Frio	0
Row2	2	SD	0	Caluroso	1
Row3	2	CIN	1	Frio	0
Row4	3	NYJ	0	Frio	1
Row5	3	BUF	1	Frio	0
Row6	4	CHI	1	Frio	1
Row7	4	MIN	0	Frio	0
Row8	5	DET	0	Frio	0
Row9	5	MIA	1	Caluroso	1
Row10	6	CAR	1	Templado	1
Row11	6	BAL	0	Frio	0
Row12	7	JAX	1	Caluroso	0
Row13	7	IND	0	Frio	1
Row14	8	GB	1	Frio	1
Row15	8	ATL	0	Templado	0
Row16	9	KC	0	Templado	1
Row17	9	CLE	1	Frio	0
Row18	10	WAS	1	Frio	1
Row19	10	ARI	0	Caluroso	0
Row20	11	TEN	1	Templado	1
Row21	11	PHI	0	Frio	0

Figura 8.6: Nodo 3, resultado.

Cuando se reduce el porcentaje a 5 % también se elimina la columna de clima_partido (Figura 8.6).

Ya que se han identificado en qué variables o columnas hay valores perdidos, el siguiente paso será decidir qué hacer con ellos. En ciertos casos se puede optar por eliminar las tuplas correspondientes y en otros casos se decide por rellenar valores faltantes con distintas técnicas.

El nodo llamado *Missing Value* ayuda a manejar los valores perdidos encontrados en las celdas de la base de datos. Su configuración se aplica a todas las columnas que se seleccionen.

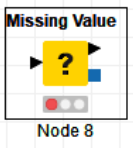


Figura 8.7: Nodo Missing Value

Es importante mencionar que las diversas maneras en la que el nodo *Missing Value* trata los valores faltantes son aplicables únicamente para datos categóricos o cuantitativos. A continuación se explica cada una de las opciones que ofrece este nodo para tratar los datos faltantes:

- **Mean.** Calcula el valor medio de todas las celdas que no faltan y reemplaza los valores perdidos con la media calculada.
- **Moving Average.** Semejante al anterior, pero reemplaza los valores perdidos con la media móvil.
- **Fix Value (Double).** Sustituye valores perdidos por un doble especificado por el usuario.
- **Maximum.** Encuentra el valor más grande de la columna y reemplaza todos los valores faltantes con él.
- **Rounded Mean.** Semejante a la primera y segunda opción, calcula el valor medio de todas las celdas que no faltan en una columna y reemplaza los valores faltantes por esta media, pero redondeada.
- **Fix Value (Integer).** Reemplaza valores perdidos con un número entero dado por el usuario.
- **Minimum.** Encuentra el valor más pequeño de la columna y reemplaza todos los valores faltantes con él.
- **Most Frequent Value.** Calcula el valor más frecuente en una columna y sustituye los valores faltantes por él.
- **Previous.** Reemplaza los valores faltantes por el último valor no faltante encontrado en la columna para la que está configurado.
- **Remove Row.** Elimina las filas que tienen un valor faltante en la columna para la que está configurado.
- **Median.** Calcula el valor mediano de la columna y reemplaza todos los valores faltantes con él (si la base es demasiado grande, usar esta opción puede ser costoso computacionalmente, ya que la base necesita ser ordenada para encontrar dicho valor).
- **Linear Interpolation.** Reemplaza los valores faltantes por la interpolación lineal entre el último valor no faltado encontrado y el siguiente.

- **Fix Value (String).** Reemplaza los valores que faltan con una cadena especificada por el usuario.
- **Average Interpolation.** Reemplaza los valores faltantes por el valor promedio del valor no faltado encontrado anterior y siguiente en la columna para la que está configurado.

Se usará la opción **Mean** para imputar los valores faltantes de la columna de marcador como se ve en la Figura 8.8.

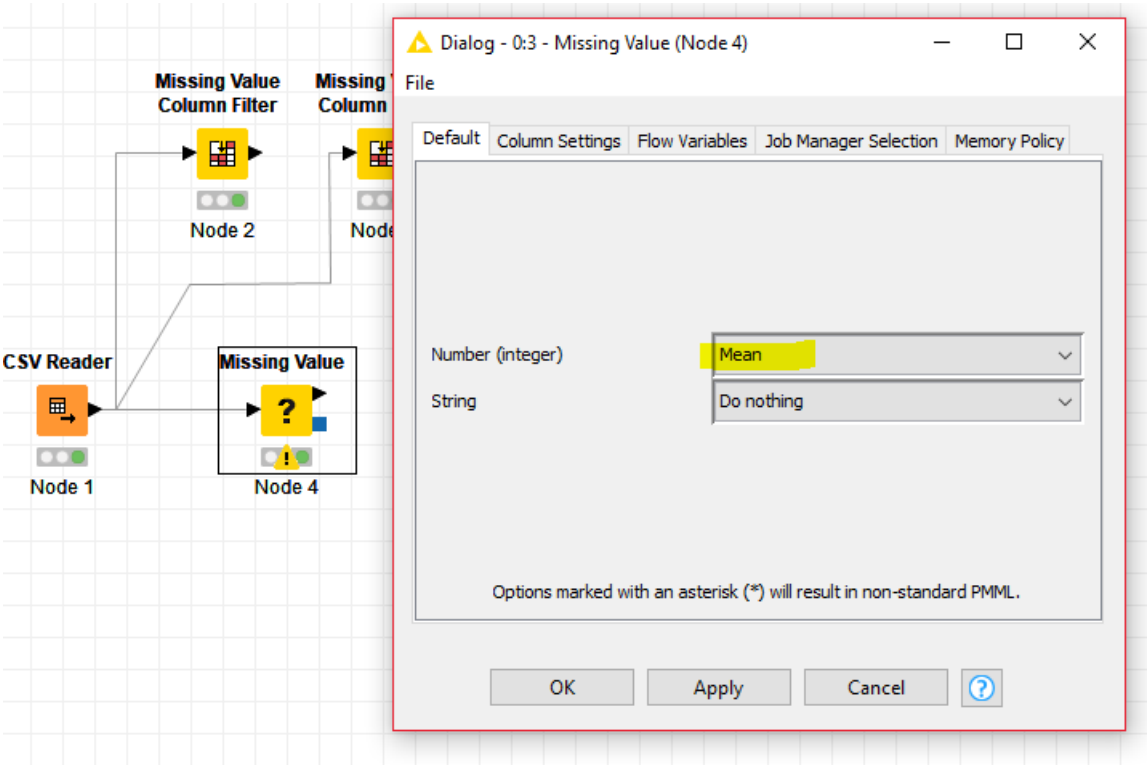
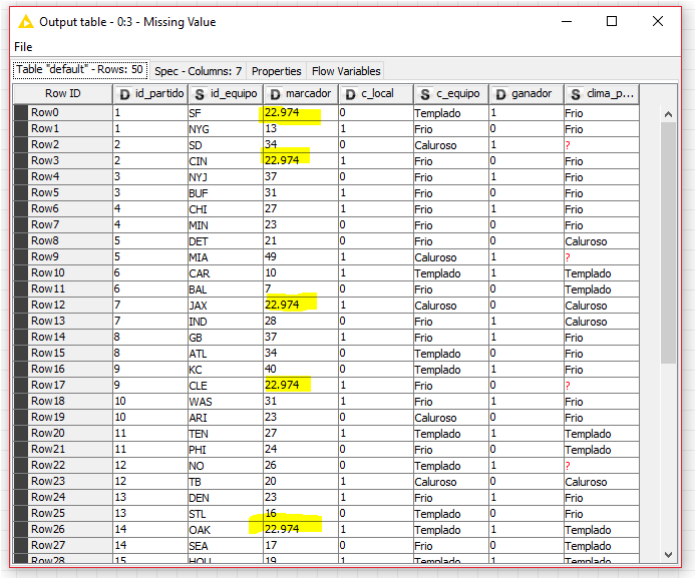


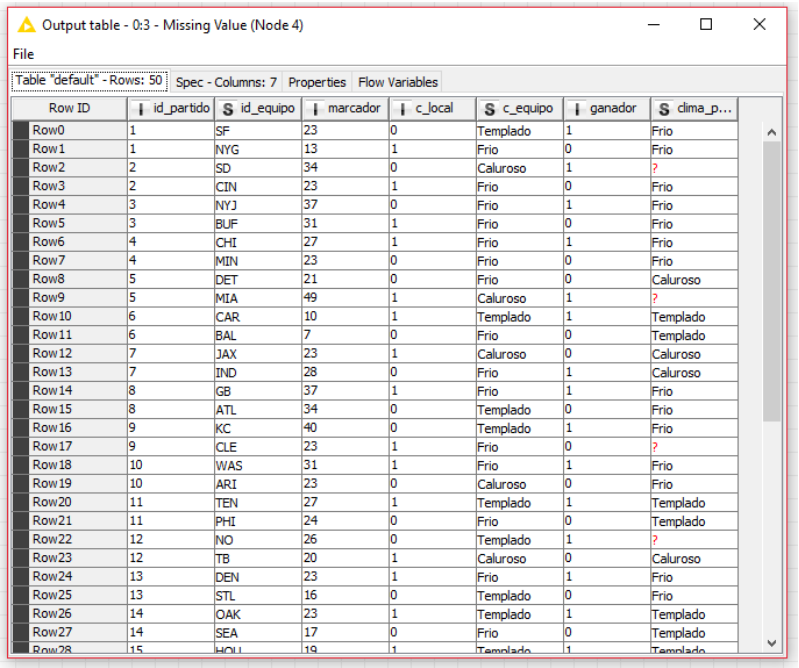
Figura 8.8: Nodo 4, configuración.



Row ID	id_partido	id_equipo	marcador	c_local	c_equipo	ganador	clima_p...
Row0	1	SF	22.974	0	Templado	1	Frio
Row1	1	NYG	13	1	Frio	0	Frio
Row2	2	SD	34	0	Caluroso	1	?
Row3	2	CIN	22.974	1	Frio	0	Frio
Row4	3	NYJ	37	0	Frio	1	Frio
Row5	3	BUF	31	1	Frio	0	Frio
Row6	4	CHI	27	1	Frio	1	Frio
Row7	4	MIN	23	0	Frio	0	Frio
Row8	5	DET	21	0	Frio	0	Caluroso
Row9	5	MIA	49	1	Caluroso	1	?
Row10	6	CAR	10	1	Templado	1	Templado
Row11	6	BAL	7	0	Frio	0	Templado
Row12	7	JAX	22.974	1	Caluroso	0	Caluroso
Row13	7	IND	28	0	Frio	1	Caluroso
Row14	8	GB	37	1	Frio	1	Frio
Row15	8	ATL	34	0	Templado	0	Frio
Row16	9	KC	40	0	Templado	1	Frio
Row17	9	CLE	22.974	1	Frio	0	?
Row18	10	WAS	31	1	Frio	1	Frio
Row19	10	ARI	23	0	Caluroso	0	Frio
Row20	11	TEN	27	1	Templado	1	Templado
Row21	11	PHI	24	0	Frio	0	Templado
Row22	12	NO	26	0	Templado	1	?
Row23	12	TB	20	1	Caluroso	0	Caluroso
Row24	13	DEN	23	1	Frio	1	Frio
Row25	13	STL	16	0	Templado	0	Frio
Row26	14	OAK	22.974	1	Templado	1	Templado
Row27	14	SEA	17	0	Frio	0	Templado
Row28	15	HOU	19	1	Templado	1	Templado

Figura 8.9: Nodo 4, resultados.

¿Tiene sentido haber usado la opción **Mean**? Siempre se debe tener en cuenta el contexto de los datos, por lo tanto, como un marcador no puede tener decimales, se cambia la opción a **Rounded Mean** y el resultado será el siguiente:



Row ID	id_partido	id_equipo	marcador	c_local	c_equipo	ganador	clima_p...
Row0	1	SF	23	0	Templado	1	Frio
Row1	1	NYG	13	1	Frio	0	Frio
Row2	2	SD	34	0	Caluroso	1	?
Row3	2	CIN	23	1	Frio	0	Frio
Row4	3	NYJ	37	0	Frio	1	Frio
Row5	3	BUF	31	1	Frio	0	Frio
Row6	4	CHI	27	1	Frio	1	Frio
Row7	4	MIN	23	0	Frio	0	Frio
Row8	5	DET	21	0	Frio	0	Caluroso
Row9	5	MIA	49	1	Caluroso	1	?
Row10	6	CAR	10	1	Templado	1	Templado
Row11	6	BAL	7	0	Frio	0	Templado
Row12	7	JAX	23	1	Caluroso	0	Caluroso
Row13	7	IND	28	0	Frio	1	Caluroso
Row14	8	GB	37	1	Frio	1	Frio
Row15	8	ATL	34	0	Templado	0	Frio
Row16	9	KC	40	0	Templado	1	Frio
Row17	9	CLE	23	1	Frio	0	?
Row18	10	WAS	31	1	Frio	1	Frio
Row19	10	ARI	23	0	Caluroso	0	Frio
Row20	11	TEN	27	1	Templado	1	Templado
Row21	11	PHI	24	0	Frio	0	Templado
Row22	12	NO	26	0	Templado	1	?
Row23	12	TB	20	1	Caluroso	0	Caluroso
Row24	13	DEN	23	1	Frio	1	Frio
Row25	13	STL	16	0	Templado	0	Frio
Row26	14	OAK	23	1	Templado	1	Templado
Row27	14	SEA	17	0	Frio	0	Templado
Row28	15	HOU	19	1	Templado	1	Templado

Figura 8.10: Nodo 4, resultado 2.

El Nodo *Missing Value* permite configurar distintas opciones para cada columna en su segunda ventana. En la configuración del Nodo 5 se aplica lo que se decidió en el paso anterior y se usa **Rounded Mean** para la columna de marcador.

Por otra parte, pueden elegirse también diferentes opciones para los valores faltantes en la columna clima_partido, en este caso al corroborar que la columna apenas tiene alrededor el 5 % de valores faltantes puede elegirse la opción de eliminar las correspondientes tuplas, **Remove Row**.

A continuación se muestra la configuración por columna y el resultado:

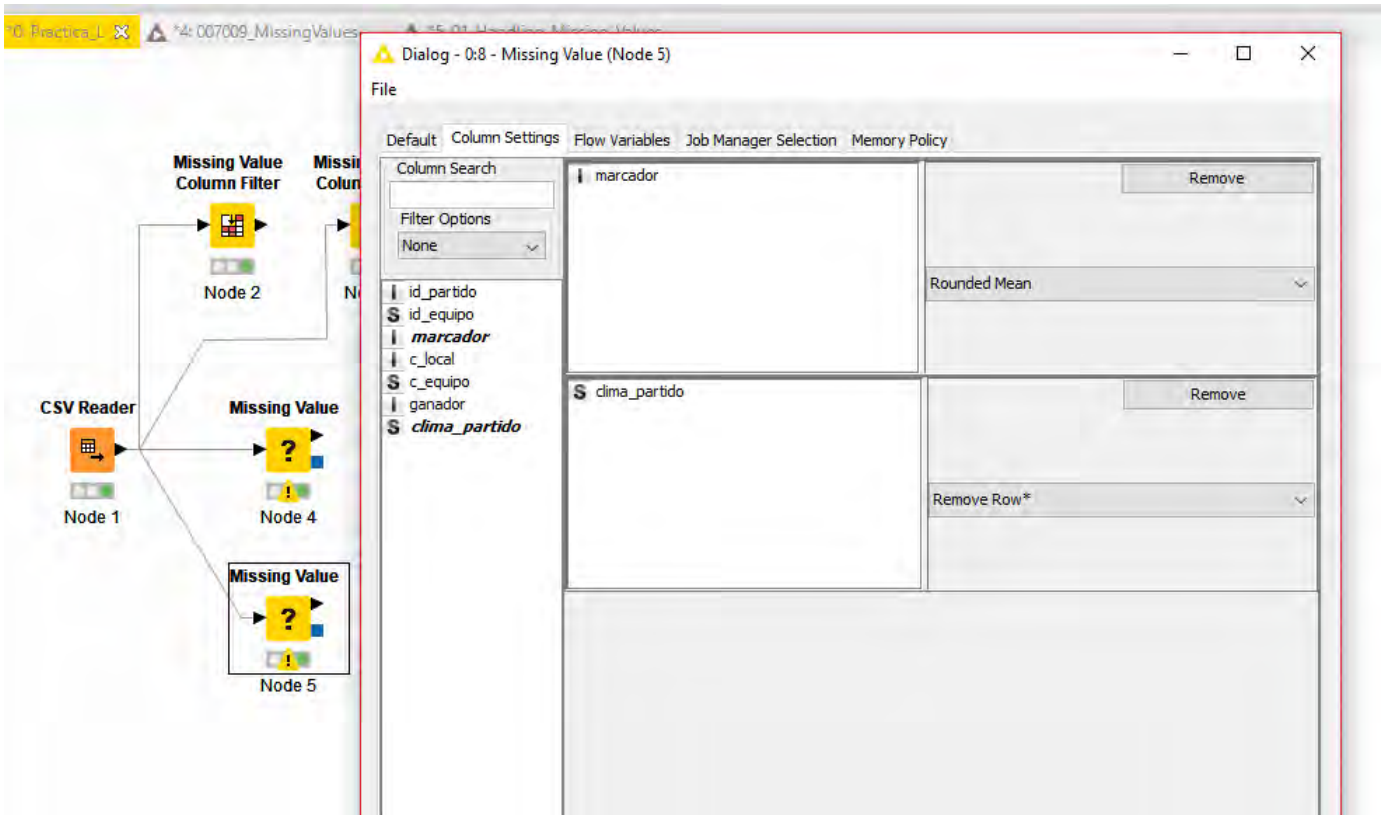


Figura 8.11: Nodo 5, configuración.

Output table - 0:8 - Missing Value (Node 5)

File

Table "default" - Rows: 46 Spec - Columns: 7 Properties Flow Variables

Row ID	id_partido	id_equipo	marcador	c_local	c_equipo	ganador	clima_p...
Row0	1	SF	23	0	Templado	1	Frio
Row1	1	NYG	13	1	Frio	0	Frio
Row3	2	CIN	23	1	Frio	0	Frio
Row4	3	NYJ	37	0	Frio	1	Frio
Row5	3	BUF	31	1	Frio	0	Frio
Row6	4	CHI	27	1	Frio	1	Frio
Row7	4	MIN	23	0	Frio	0	Frio
Row8	5	DET	21	0	Frio	0	Caluroso
Row10	6	CAR	10	1	Templado	1	Templado
Row11	6	BAL	7	0	Frio	0	Templado
Row12	7	JAX	23	1	Caluroso	0	Caluroso
Row13	7	IND	28	0	Frio	1	Caluroso
Row14	8	GB	37	1	Frio	1	Frio
Row15	8	ATL	34	0	Templado	0	Frio
Row16	9	KC	40	0	Templado	1	Frio
Row18	10	WAS	31	1	Frio	1	Frio
Row19	10	ARI	23	0	Caluroso	0	Frio
Row20	11	TEN	27	1	Templado	1	Templado
Row21	11	PHI	24	0	Frio	0	Templado
Row23	12	TB	20	1	Caluroso	0	Caluroso
Row24	13	DEN	23	1	Frio	1	Frio
Row25	13	STL	16	0	Templado	0	Frio
Row26	14	OAK	23	1	Templado	1	Templado
Row27	14	SEA	17	0	Frio	0	Templado
Row28	15	HOU	19	1	Templado	1	Templado
Row29	15	DAL	10	0	Caluroso	0	Templado
Row30	16	NE	30	1	Frio	1	Frio
Row31	16	PIT	14	0	Frio	0	Frio
Row32	17	IND	13	1	Frio	0	Frio
Row33	17	MIA	23	0	Caluroso	1	Frio
Row34	18	JAX	23	0	Caluroso	1	Templado
Row35	18	KC	16	1	Templado	0	Templado
Row36	19	TB	25	0	Caluroso	1	Frio
Row37	19	BAL	23	1	Frio	0	Frio
Row38	20	DAL	21	1	Caluroso	1	Caluroso
Row39	20	TEN	13	0	Templado	0	Caluroso
Row40	21	CIN	7	0	Frio	0	Frio
Row41	21	CLE	20	1	Frio	1	Frio
Row42	22	NE	23	0	Frio	1	Frio
Row43	22	NYJ	7	1	Frio	0	Frio
Row44	23	CAR	31	1	Templado	1	Templado
Row45	23	DET	23	0	Frio	0	Templado
Row46	24	NO	35	1	Templado	1	Templado
Row47	24	GB	23	0	Frio	0	Templado
Row48	25	CHI	14	0	Frio	1	Templado
Row49	25	ATL	23	1	Templado	0	Templado

Figura 8.12: Nodo 5, resultado.

Como puede observarse en la Figura 8.12, los cambios fueron realizados. Para que los cambios se apliquen y se conserven, se usará un nuevo nodo llamado **Missing Value (Apply)** donde se selecciona la política de memoria requerida y se ejecuta el nodo.

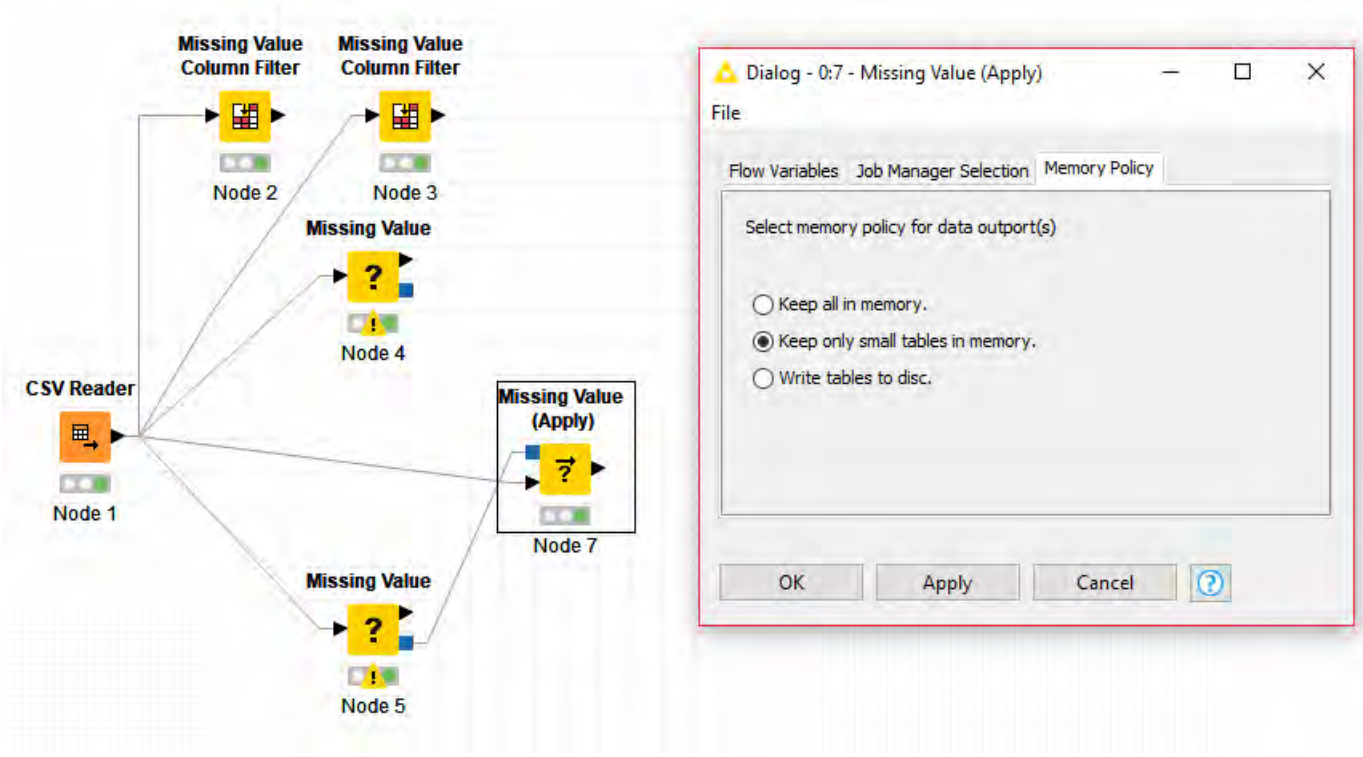


Figura 8.13: Nodo 7, configuración.

KNIME, como se ha mostrado, permite examinar los datos para manejar los valores perdidos, así como para estimar e imputar los valores que faltan mediante distintos algoritmos.

En resumen, una forma de hacer un análisis para poder limpiar las bases de datos respecto a valores faltantes se puede resumir al resolver las siguientes preguntas.

1. ¿Dónde se encuentran los valores perdidos?
2. ¿Qué cantidad de valores perdidos hay? ¿Qué porcentaje representan?
3. ¿Es conveniente prescindir de los registros con datos faltantes?
4. Si se opta por sustituir los valores perdidos, ¿qué método es el mejor y qué posibles consecuencias puede tener dicha elección?

8.4. Ejercicios

- Busque una base de datos que contenga datos perdidos.

  En caso de no tener acceso a otra base de datos, de la base obtenida en la práctica S, borre algunos registros de las variables clima_partido y marcador utilizando conocimientos de la práctica K (nodo tipo *Binner*). Si opta por esta opción, especifique el criterio que utilizó para borrar los registros.

- Replique el análisis respondiendo las preguntas mostradas anteriormente.

- Responda ¿Son suficiente para resolver el problema de los missing values, los nodos explicados en la práctica? ¿Por qué? ¿Qué conocimientos cree que se deban tener previamente?

9 | Práctica P - Introducción a Python para la manipulación de datos.

9.1. Objetivos

- Aprender la sintaxis del lenguaje Python.
- Conocer las estructuras básicas para la manipulación de datos.
- Que el alumno practique sus conocimientos del lenguaje Python, conforme van obteniendo experiencia en el lenguaje.

9.2. Introducción

Python es un lenguaje de programación de alto nivel orientado a objetos creado con el propósito de ser un lenguaje ágil y sencillo por lo que hace hincapié en la productividad y legibilidad del mismo.

Este lenguaje puede ser utilizado en cualquier sistema operativo y puede usarse para procesar texto, números e imágenes, así como también puede ser empleado para la lectura y la escritura en MySQL y PostgreSQL o para el desarrollo de aplicaciones.

Dada la versatilidad del lenguaje, actualmente Python también es uno de los lenguajes más usados para la ciencia de datos y la visualización de los mismos. Las bibliotecas destacadas para esta materia son Pandas, NumPy, SciPy, Scikit Learn, Matplotlib.

Python puede descargarse directamente desde su sitio web[1] y puede utilizarse el entorno de desarrollo o compiladores que se prefiera. Otra posibilidad es utilizar alguna terminal en línea, que son servicios web que permiten editar, compilar y ejecutar código de diversos lenguajes y paquetes desde un navegador y tienen la ventaja de no tener que instalar ningún programa.

Existe una amplia variedad de terminales en línea, de las cuales se muestra a continuación una lista:

- JDOODLE: compilador y editor en línea que soporta 63 lenguajes incluidos R y Python.
  https://www.jdoodle.com
- Browxy y Compilejava: son compiladores libres de Java.
  http://browxy.com y *https://www.compilejava.net/*

[1] https://www.python.org/

- Repl.it: compilador y editor que soporta 36 lenguajes entre ellos Python.

  http://repl.it

- Tutorialspoint: Entorno de desarrollo y aplicaciones, donde se puede crear programas en más de 80 lenguajes de programación, compilar, ejecutar y compartirlos a través de la web.

  https://www.tutorialspoint.com/codingground.htm

- Ideone: es una herramienta que te permite compilar el código fuente y lo ejecuta en línea en más de 60 lenguajes de programación, incluidos R y Python.

  http://ideone.com/

Para la práctica actual, se sugiere usar alguno de los dos compiladores siguientes: JDOODLE, donde puede seleccionarse el lenguaje de programación a utilizar como se muestra en la Figura 9.1 :

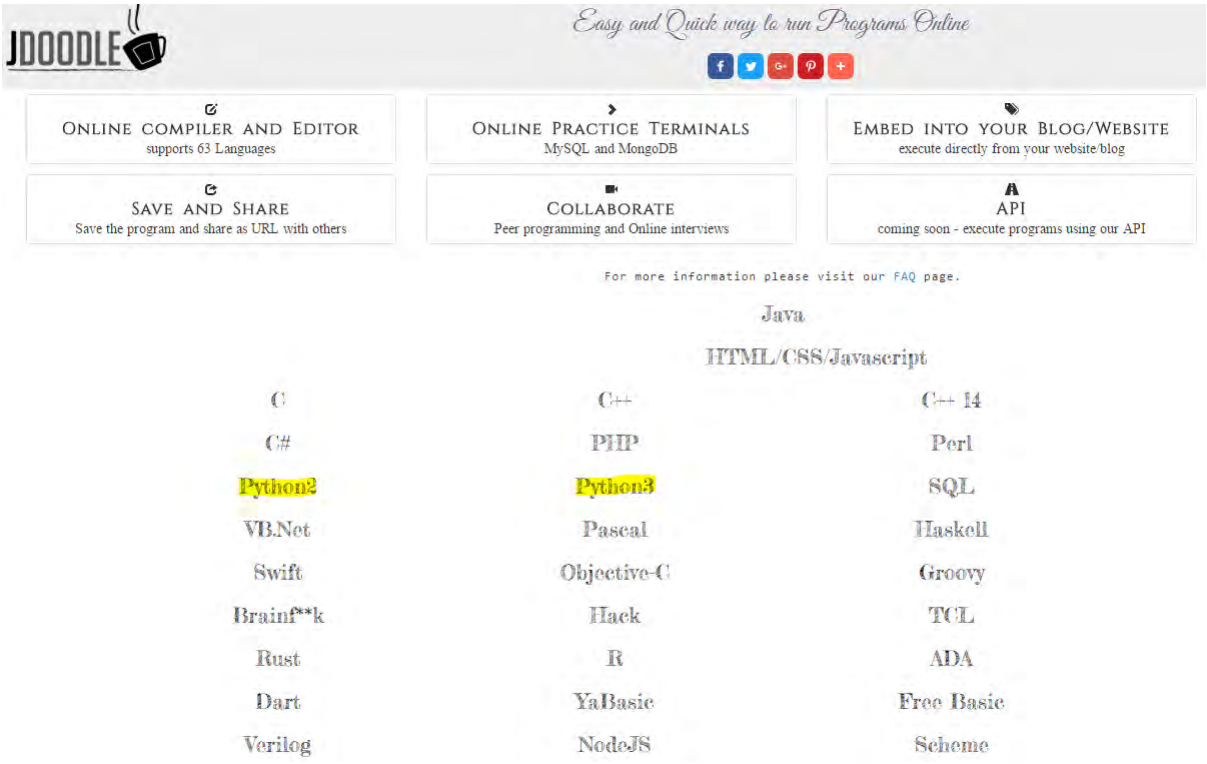


Figura 9.1: Selección de lenguaje en compilador online JDOODLE.

Donde podrá editarse código y ejecutarse como se muestra a continuación:

Figura 9.2: Compilador Online JDOODLE.

O si se prefiere, puede utilizarse Repl.it, del cual puede seleccionarse el lenguaje de manera similar:

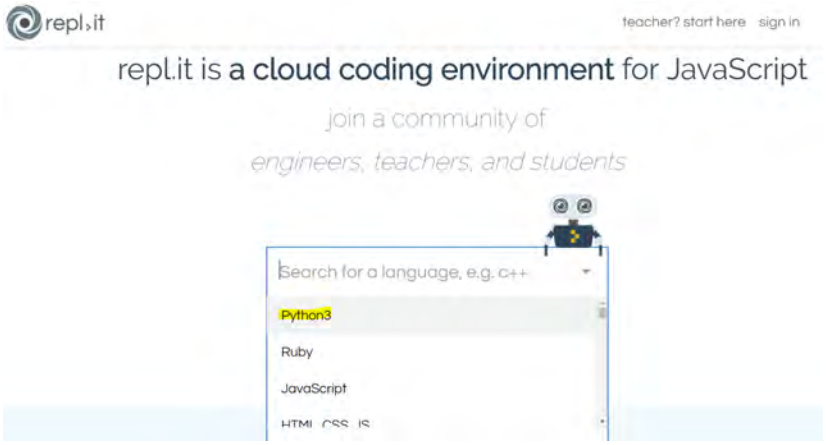


Figura 9.3: Selección de lenguaje en compilador online Repl.it.

De esta manera se aprecia el editor y compilador:

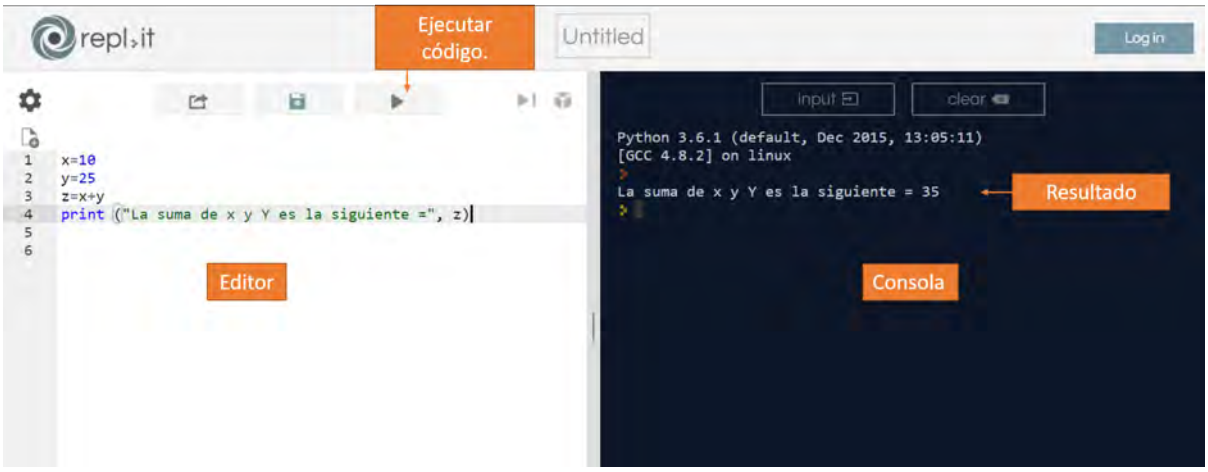


Figura 9.4: Compilador Online Repl.it.

9.3. Python y operaciones básicas

Adicionalmente a las operaciones básicas de suma, resta, multiplicación y división, Python tiene soporte para operaciones más avanzadas, tales como exponentes y cálculo de residuos de cocientes (módulos), por lo que se puede usar fácilmente como calculadora:

```
# Suma y resta
print(10+ 13 )
print(54 - 31)

# Producto y cociente
print(6 * 5)
print(526 / 2)

# Exponenciacion
print(5**2)

# Modulo
print(18 % 7)
```

```
Out[2]:23
Out[3]:23
Out[6]:30
Out[7]:263.0
Out[10]:25
Out[13]:4
```

Como es posible apreciar en el ejemplo, es necesaria la función *print()* para poder visualizar los resultados de cada operación. Así también el signo de # permite escribir comentarios en el código para su documentación sin que Python los tome en cuenta al momento de ejecutar el script.

Ejercicio 1: Suponga que tiene $100 y que puede invertirlos con una tasa de interés compuesto cada año de 10 %. ¿Cúal será el valor de los $100 después de 7 años?

9.4. Asignación y tipos de datos básicos

En Python, se puede asignar un valor a una variable con el signo = como en este ejemplo:

```
#Asignacion
x=5
print(x)
```

```
Out[3]:5
```

Ahora puede utilizarse el nombre de la variable x, en lugar del valor real 5. Es importante recalcar que en Python = significa asignación, no igualdad.

Ejercicio 2: Declare una variable llamada *factor* igual a 1.10 y una variable llamada *ahorro* con valor de 100. Calcule la cantidad de dinero que se tendrá por la inversión de 7 años con las variables ahorro y factor. Almacene el resultado en una variable llamada *resultado* e imprímala.

En los ejercicios anteriores se trabajó con números enteros (variable ahorro) y con números reales (variable factor). Estos son dos tipos de datos básicos en Python: **int** (de integer o entero) y **float** que hace referencia a los números reales o con parte fraccionaria.

Para asignar un dato de tipo float, como se hizo antes, se escribe la parte entera, seguida de un punto y la parte decimal. Otra manera de hacer es utilizando notación científica y añadir una e para indicar un exponente base 10. Por ejemplo:

```
numeroreal=7.91e-3
print (numeroreal)
```

```
Out[2]:0.00791
```

Existen otros tipos de datos comunes, por ejemplo, los datos del tipo *str* o cadenas, que son un tipo de dato que se utiliza para representar texto, siempre escrito entre comillas simples o dobles. Otro ejemplo de tipo de dato es **bool** o boolean, el cual representa valores lógicos y solo puede ser true (verdadero) o false (falso).

Se puede corroborar el tipo de dato que contiene cada variable con la función *type()* como se muestra a continuación:

```
#Esto es una cadena
c = "Hola Mundo"

#Esto es un entero
e = 23

#Esto es un numero con parte fraccionaria
d = 5.6

#Esto es un boolean o bool
k = True

print(c)
print(type(c))
print(type(d))
print(type(k))
```

```
Out[13]:Hola Mundo
Out[15]:<class 'str'>
Out[15]:<class 'float'>
Out[16]:<class 'bool'>
```

Ya que se han visto diferentes tipos de datos bajo los operadores comunes, es intuitivo que entre números se efectuarán las operaciones deseadas, sin embargo, ¿qué pasa con la cadenas?

```
cadena="Hola hola"
print (cadena)

# Repeticion
cadena1 = "cadena" * 3
print (cadena1)
print (type (cadena1))

# Concatenacion
nombre = "Karina"
apellido = "Caballero"
nombre_completo = nombre + " " + apellido
print (nombre_completo)
print (type (nombre_completo))

print ("Tamano de cadena '", nombre_completo, "' es:", len(nombre_completo))

# acceder a rango de la cadena
print (nombre_completo[3:13])
```

```
Out[2]:Hola hola
Out[6]:cadenacadenacadena
Out[7]:<class 'str'>
Out[13]:Karina Caballero
Out[14]:<class 'str'>
Out[16]:Tamano de cadena ' Karina Caballero ' es: 16
Out[19]:ina Caball
```

Como se pudo observar usando el operador + en tipos de datos cadena el resultado es la unión de los mismos, lo cual puede ser útil cuando tenemos resultados guardados en distintos tipos de variables.

Ejercicio 3: Imprima el siguiente código, ¿qué le dice el error?

```

print("Si inviertes $"+ ahorro+ " podras tener $"+ resultado + " al final de 7
    periodos)
```

Para que el código anterior funcione se tiene que hacer una conversión de variables, de float a cadena. Para lograr esta conversión, se utiliza la función *str()* para así poder unir las variables del mismo tipo. Existen funciones similares como *int(), float() y bool()* las cuales convierten los datos de Python a otro tipo de dato. A continuación se muestra un ejemplo.

```
ahorro=100
resultado=ahorro * 1.10 ** 7
print("Si inviertes $"+str(ahorro)+" podras tener $"+str(resultado)+" al final
    de 7 periodos.")
```

```

Si inviertes $100 podras tener $194.87171000000012 al final de 7 periodos.
```

De esta manera se unen variables del mismo tipo. Para averiguar qué tipo de dato contiene una variable siempre se puede recurrir a la función *type()* como se hizo anteriormente.

Ejercicio 4: Cree la variable cadena pi="3.1415926" y utilice la función *float()* para almacenar en una variable llamada *area* el cálculo del área de un círculo cuyo radio es 7 cm. e imprima *area.* ¿Qué sucede si se escribe print(pi*3)?

9.5. Listas

Hasta ahora, los tipos de variables que se han visto almacenan o representan un único dato. Esto puede tener algunos inconvenientes cuando se tienen muchos datos del mismo tipo. Sin embargo, una de las estructuras básicas de Python da solución a este aspecto.

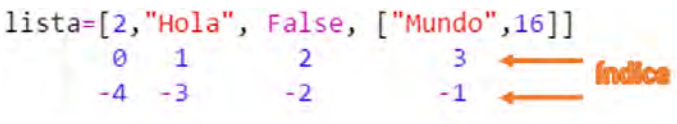
Una **lista** es una colección de datos y es una de las estructuras mas versátiles en Python ya que, a diferencia de los tipos de datos int o bool, una lista es un tipo de variable que puede agrupar tipos de datos compuestos. Una lista puede ser escrita como un conjunto de valores entre corchetes separados entre ellos por comas. Por ejemplo:

```
#Lista con datos str
familia=["Liz","Ale","Papa","Mama"]
print(familia)

#Lista con datos float
alturas=[1.58,1.80,1.76,1.60]
print(alturas)

#Lista compuesta
familia2=["Liz",1.58,"Ale",1.80,"Papa",1.76,"Mama",1.60]
print(familia2)

#Lista de listas
familia3=[["Liz",1.58], ["Ale",1.80],["Papa",1.76],["Mama",1.60]]
print(familia3)

#Lista de variables
a="Mi familia esta formada por"
b="Y nuestras alturas son"
familia4= [a,familia,b, alturas]
print(familia4)
```

```
Out[3]:['Liz', 'Ale', 'Papa', 'Mama']

Out[7]:[1.58, 1.8, 1.76, 1.6]

Out[11]:['Liz', 1.58, 'Ale', 1.8, 'Papa', 1.76, 'Mama', 1.6]

Out[15]:[['Liz', 1.58], ['Ale', 1.8], ['Papa', 1.76], ['Mama', 1.6]]

Out[21]:['Mi familia esta formada por', ['Liz', 'Ale', 'Papa', 'Mama'], 'Y
    nuestras alturas son', [1.58, 1.8, 1.76, 1.6]]
```

Ejercicio 5: Cree una lista con los nombres de su familia y otra con las alturas de cada miembro. Utilice ambas para crear una nueva lista llamada *mifamilia* donde estén contenidas las anteriores.

Las listas pueden ser muy útiles gracias a la forma en que puede accederse a sus elementos. Para empezar, los elementos de una lista tienen asignado un índice con el cual pueden referenciarse, de esta manera dependiendo de la posición en la cual están pueden ser seleccionados. Por ejemplo:



En la lista anterior, lista[0] y lista[-4] hacen referencia al mismo elemento de la lista. Algunos ejemplos de manipulación de listas se muestran a continuación:

```
lista=[2,"Hola", False, ["Mundo",16]]

#Acceder a un elemento
a=lista[1]
print (a)

#Acceso a un elemento en lista anidada
b=lista[3][0]
print(b)

#Darle un  nuevo valor e un elemento
lista[1]="Hello"
print(lista)

#Obtener un rango de elementos
c=lista[1:3]
print(c)

d=lista[3:]
print(d)

e=lista[:3]
print(e)

#Borrar elementos de la lista
del(lista[0])
print(lista)
```

```
Out[5]:Hola

Out[9]:Mundo

Out[13]:[2, 'Hello', False, ['Mundo', 16]]

Out[17]:['Hello', False]

Out[20]:[['Mundo', 16]]

Out[23]:[2, 'Hello', False]

Out[27]:['Hello', False, ['Mundo', 16]]
```

Ejercicio 6: Cree la siguiente lista:

```
lis=["Edades",[18,23,45,23,45,38,16], "nombres", ["Carol", "Lizzy", "Roy", "
    Dany", "Jorge", "Oscar","Erick"]]
```

E imprima lo siguiente:

- El primer y segundo elemento de lis.
- Las edades del 23 al 38.
- Los 3 últimos nombres de la sublista.

9.6. Funciones y métodos

En Python todo es un objeto, cualquier “cosa” a la que se le pueda atribuir cualidades o atributos, como tamaño o tipo, tiene la capacidad de ser manipulado con funciones o métodos.

Python ofrece una variedad de funciones ya incorporadas para ayudar al manejo de datos. Algunas de ellas ya se mencionaron anteriormente como lo son las funciones print(), type(), str(), float() y bool(). La manera general de llamar las funciones es la siguiente:

```
Var_resultado = nombre_funcion ( Var_entrada )
```

Algunas funciones básicas se explican a continuación:

- print(): imprime los valores de una secuencia.
- len(): devuelve el número de elementos que contiene un conjunto de datos.
- max(): devuelve el elemento más grande de un conjunto de datos.
- min(): devuelve el elemento más pequeño de un conjunto de datos.
- help(): inicia una sesión de ayuda interactiva sobre cualquier función que esté en el argumento.
- sorted(). devuelve una lista con los elementos de la lista original en orden ascendente.

Ejercicio 7: Revise la documentación de complex() escribiendo:

```
help(complex)
```

• ¿Qué devuelve esta función? ¿Qué argumentos necesita?
• Utilice la función *max()* y *min()* para saber los valores extremos del segundo elemento de la lista del ejercicio 6.
• Utilice la función *sorted()* para ordenar los nombres de lista del ejercicio 6. ¿Con qué criterio los ordena?

Un método es la forma de definir una determinada acción que realiza un objeto. La manera general de llamar un método es la siguiente:

```
var_nueva=objeto.nombremetodo()
```

Existen métodos exclusivos de los datos tipo string. A continuación un ejemplo de los principales.

```

#### EJEMPLO DE CADENA ######
cadena="este eS mi ejemplo de CADENA"


######## METODOS DE FORMATO#######
#La cadena con la primera letra en mayusculas
cadena2= cadena.capitalize()
print(cadena2)

#Convertir una cadena a minusculas
cadena3=cadena.lower()
print(cadena3)

#Convertir una cadena a mayusculas
cadena4=cadena.upper()
print(cadena4)

#Convertir mayusculas a minusculas y viceversa
cadena5=cadena.swapcase()
print(cadena5)
```

```
Out[9]: Este es mi ejemplo de cadena
Out[13]: este es mi ejemplo de cadena
Out[17]: ESTE ES MI EJEMPLO DE CADENA
Out[21]: ESTE Es MI EJEMPLO DE cadena
```

Existen algunos otros métodos que puede aplicarse a ciertos objetos, por ejemplo en listas.

```
###### METODOS DE BUSQUEDA ######

#Cuenta el elemento que se indique
#Cuantas ``o" hay
o=cadena.count("o")
print(o)
#cuantas ``a"  hay en minuscula y cuantas en mayuscula?
a=cadena.count("a")
print(a)
A=cadena.count("A")
print(A)


#Devuelve un entero que representa la posicion de el elemento pedido.
indice=cadena.index("o")
print(indice)


## Ejemplo en una lista
Nombres=["Karina", "Alejandra", "Estephani", "Oscar", "Leonor"]
ind_ale=Nombres.index("Alejandra")
print(ind_ale)

Promedios=[8.6,8.5,8.6,8.5,8.9]
print(Promedios.count(8.6))
```

```
Out[6]: 1
Out[9]: 0
Out[11]: 2
Out[16]: 17
Out[22]: 1
Out[25]: 2
```

```
###### METODOS DE  SUSTITUCION #######

Nombres=["Karina", "Alejandra", "Estephani", "Oscar", "Leonor"]
Promedios=[8.6,8.5,8.6,8.5,8.9]
cadena="este eS mi ejemplo de CADENA"


#Agrega un elemento adelante en el objeto que se le indique
Nombres.append("Monse")
Promedios.append(8.2)
print(Nombres)
print(Promedios)

#Elimina el primer elemento del objeto que coincida con la entrada
Nombres.remove("Karina")
print(Nombres)
Promedios.remove(8.6)
print(Promedios)


#Invierte el orden de los elementos del
Nombres.reverse()
```

```
print(Nombres)


#Reemplaza una subcadena
print(cadena.replace("CADENA","texto"))
```

```
Out[11]:['Karina', 'Alejandra', 'Estephani', 'Oscar', 'Leonor', 'Monse']
Out[12]:[8.6, 8.5, 8.6, 8.5, 8.9, 8.2]
Out[16]:['Alejandra', 'Estephani', 'Oscar', 'Leonor', 'Monse']
Out[18]:[8.5, 8.6, 8.5, 8.9, 8.2]
Out[23]:['Monse', 'Leonor', 'Oscar', 'Estephani', 'Alejandra']
Out[27]: este eS mi ejemplo de texto
```

Ejercicio 8: Cree la siguiente lista:

```
Lista2=["hola", [3,4,5,6,7],"ADIOS"]
```

- Agregue en la sublista el numero 8 e imprima.
- En una nueva variable guarde el primer elemento de lista y utilice la función que sea necesaria para poner la palabra en mayúsculas e imprima.
- Ordene la lista de manera inversa e imprima.

9.7. Soluciones

```


#Respuesta 1:
    print(100*1.1**7)

#Respuesta 2:
    factor=1.10
    ahorro=100
    resultado=ahorro*factor**7
    print(resultado)

#Respuesta 3:
    #Los tipos de variables son diferentes.
#Respuesta 4:
    pi="3.1415926"

    area=7**2 * float(pi)
    print(area)


#Respuesta 5:   libre

#Respuesta 6:   print(lis[:2])
    print(lis[1][1:6])
    print(lis[3][4:])

#Respuesta 7:
    #complex() toma dos argumentos: realy e imag. reales un argumento requerido,
            images un argumento opcional. La funcion devuelve un numero complejo
    a partir de    una parte real y una parte imaginaria.

    print(max(lis[1]))


    print(sorted(lis[3]))
```

```
34      #En orden alfabetico.
35
36
37 #Respuesta 8:
38      Lista2[1].append(8)
39
40      print(Lista2)
41
42 L=Lista2[0].capitalize()
43
44      print(L)
45
46 Lista2.reverse()
47
48      print(Lista2)
```

10 | Práctica Q - Introducción a Python para la manipulación de datos II.

10.1. Objetivos

- Que el alumno progrese en la comprensión de la sintaxis del lenguaje Python.
- Que el alumno refuerce los conocimientos adquiridos en la práctica P.
- Adquirir conocimientos sobre la importación y uso de paquetes dentro de Python.

10.2. Introducción

Python puede ampliar sus funcionalidades gracias a sus diferentes paquetes. Un paquete es una carpeta que contiene archivos .py, que no son sino nuevas funciones que pueden ser llamadas para su uso.

Con los años, se han desarrollado multitud de paquetes para la materia de ciencia de datos, que permiten realizar numerosas tareas de tratamiento de datos, visualización, cálculos y aplicaciones científicas específicas. Algunos de ellos son los siguientes[1]:

- NumPy[2]: la característica más potente de NumPy es su matriz n-dimensional. Este paquete también contiene funciones básicas de álgebra lineal, transformadas de Fourier, capacidades avanzadas de números aleatorios y herramientas para la integración con otros lenguajes de bajo nivel como Fortran, C y C ++.
- SciPy[3]: significa Python Científico. SciPy se basa en NumPy. Es uno de los paquetes más útiles con una variedad de módulos de ciencia e ingeniería de alto nivel como transformada de Fourier discreta, Álgebra Lineal, Optimización y matrices dispersas.
- Matplotlib[4]: para el diseño de gran variedad de gráficos, histogramas y líneas.
- Pandas[5]: para operación y manipulación de datos estructurados. Se utiliza ampliamente para minería de datos. Con poco tiempo de antigüedad, Pandas ha sido fundamental para impulsar el uso de Python en la comunidad de científicos de datos.

[1] https://goo.gl/UojNsh y https://goo.gl/WAJhiH
[2] *http://www.numpy.org/*
[3] *http://www.scipy.org/*
[4] *http://matplotlib.org/*
[5] *pandas.pydata.org/*

- SymPy[6]: tiene capacidades de amplio alcance de la aritmética simbólica básica para cálculo, álgebra, matemáticas discretas y de física cuántica. Otra característica útil es la capacidad de formatear el resultado de los cálculos como código LaTeX.

10.3. Paquetes

Antes de usar un paquete, es necesario importarlo y con éste a las funciones contenidas. La forma general de importar un paquete es la siguiente:

```
import paquete

#Con un alias
import paquete as p

#Llamar a una funcion del paquete
paquete.funcion

#Llamar con el alias
p.funcion
```

Ejercicio 1:

• Importe el paquete *math*. De esta manera se puede llamar a la constante *pi* con math.pi.

• Calcule la circunferencia de un círculo de radio 0.43 m. llamando a pi. Guarde el resultado en una variable C e imprima.

• Calcule el área de la circunferencia, guarde el resultado en una variable A e imprima.

• Imprima la cadena R de tal forma que diga: " El área de la circunferencia es: _____ y el área de la circunferencia es: _____ "

Las importaciones generales, como *import math*, hacen que todas las funciones del paquete math estén disponibles. Sin embargo, si sólo se quisiera utilizar una parte específica del paquete, se puede hacer una importación más específica. Por ejemplo:

```
from math import pi

#Subpquetes
```

Ejercicio 2:

• Realice una importación selectiva de la función *sqrt* (raíz cuadrada) del paquete math.

• ¿Cuánto mide el lado más grande de un triángulo cuyos dos lados menores miden 12 y 10 cm.? Imprima el resultado.

Ahora veamos diferentes formas de importar:

Supongamos que se quiere usar la función inv(), la cual se encuentra en el paquete scipy, dentro del subpquete linalg. ¿Cuál es la manera de importar?

```
from scipy.linalg import inv

#con alias para la funcion
from scipy.linalg import inv as mi_inv

#Asi se puede usar la funcion:
mi_inv([[1,2], [3,4]])
```

[6] *http://sympy.org/en/index.html*

10.4. NumPy

NumPy es un paquete fundamental para la manipulación de bases de datos, ya que es un paquete para el cálculo científico. Dispone de un objeto matriz llamado *(array)* o arreglo, funciones para realizar cálculos entre elementos o matrices, así como operaciones de álgebra lineal.

Los arreglos de NumPy son una forma más eficiente de almacenar y manipular datos que las estructuras que se han visto hasta ahora. Los arreglos de NumPy son un vector o un conjunto de datos multidimensional.

Veámos un ejemplo de esta estructura:

```
#Se importa Numpy
import numpy as np

#Creando un vector y una lista
vector = np.array([1, 2, 3, 4])
lista = [1, 2, 3, 4]

#Diferencias entre lista y vector:
a=vector+vector
b=lista+lista

print(a)
print(b)
```

```
Out[12]:[2 4 6 8]
Out[13]:[1, 2, 3, 4, 1, 2, 3, 4]
```

```
c=vector*3
d=lista*3

print(c)
print(d)
```

```
Out[4]:[ 3  6  9 12]
Out[5]:[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
```

Como se puede ver, las propiedades de los objetos son diferentes respecto a los operadores aritméticos típicos, ya que los arreglos hacen sus operaciones de manera vectorial.

Veámos operaciones con vectores en Python:

```

a = np.array([34, 22, 14, 13,  7])
b = np.array([21, -10,  0, 11, -5])

#Suma  elemento a elemento
print(a + b)

#Resta elemento a elemento
print(a - b)


#Multiplicar elemento a elemento
print(a * b)

#Division elemento a elemento
print(b/a)

```

```
#Suma o resta de escalar
print(a-10)

#Multiplicacion por escalar
print(b*2)
```

```
Out[6]:[55  12  14  24   2]
Out[9]:[13  32  14   2  12]
Out[13]:[ 714 -220    0  143  -35]
Out[16]:[ 0.61764706 -0.45454545  0.          0.84615385 -0.71428571]
Out[19]:[24  12   4   3  -3]
Out[22]:[ 42 -20   0  22 -10]
```

Ejercicio 3: Se tiene la siguiente lista:

```
#Estaturas de jugadores del equipo de basquetbol en cm
Estaturas= [178, 182, 190, 180, 185]
```

- Haga un vector con la lista.
- Se requieren las estaturas en metros, ¿qué debería hacerse? Imprima las estaturas con la nueva medida.
- Imprima el tipo de variable que tiene las estaturas nuevas.

Ejercicio 4:

- Cree un vector con 5 posibles pesos en kg. para los jugadores.
- Calcule el índice de masa corporal con la siguiente fórmula:

$$\frac{Peso(Kg.)}{Estatura^2(m.)}$$

Guarde los resultados en la variable IMC e imprima.

Para los vectores numpy también se pueden utilizar operaciones lógicas y acceder a sus elementos como en el caso de las listas. A continuación un ejemplo:

```
Lista=[34,4,15,73,16,93,12,65,9]
Vector=np.array(Lista)

#acceder a un elemento
print(Lista[3])
print(Vector[3])

#Acceder a un rango de elementos}
print(Lista[2:5])
print(Vector[2:5])
```

```
Out[5]:73
Out[6]:73
Out[9]:[15, 73, 16]
Out[10]:[15 73 16]
```

```
#Quienes son mayores de edad
MayoresEdad=VectorEdades>18
print(MayoresEdad) # Bool

print(VectorEdades[MayoresEdad])
```

```
Out[3]:[ True False False  True False  True False  True False]
Out[5]:[34 73 93 65]
```

Ejercicio 5: Cree una matriz con booleanos de manera que el elemento de la matriz sea verdadero si el índice de masa corporal es menor a 25. Nombre el nuevo arreglo como Normal e imprima la matriz con los índices que son normales.

10.5. Matrices

Como ya se ha visto, con NumPy podemos crear vectores a partir de listas. Sin embargo estas listas tienen que contener el mismo tipo de variables como se ha ejemplificado hasta ahora.

Por otra parte, también pueden construirse matrices cuando se tiene una lista con sublistas. A continuación un ejemplo:

```

#La siguiente lista contiene la edad y los pesos de 5 chicos.
lista4= [[17, 78.4], [20, 102.7], [25, 98.5], [19, 75.2]]

v_lista=np.array(lista4)

print(v_lista)

#Conocer el tipo de objeto
print(type(v_lista))

#Conocer sus atributos (dimensiones)
print(v_lista.shape)

#Acceder a los elementos
#Primer renglon
print(v_lista[0])

#Segundo renglon
print(v_lista[1])

#Segundo elemento del primer renglon
print(v_lista[0][1])
```

```
Out[7]: [[  17.    78.4]
         [  20.   102.7]
         [  25.    98.5]
         [  19.    75.2]]

Out[10]: <class 'numpy.ndarray'>
Out[13]: (4, 2)
Out[17]: [ 17.   78.4]
Out[20]: [  20.   102.7]
Out[23]: 78.4
```

Ejercicio 6: Considerando la siguiente lista:

```
Jugadores= ([ 1 , 75  ],
[ 2 , 84  ],[ 3 , 64  ],[ 4 , 67  ],
[ 5 , 70  ],[ 6 , 79  ],[ 7 , 93  ],
[ 8 , 63  ],[ 9 , 75  ],[ 10  , 93  ],
[ 11  , 76  ],[ 12  , 92  ],[ 13  , 95  ],
[ 14  , 68  ],[ 15  , 65  ],[ 16  , 72  ],
[ 17  , 79  ],[ 18  , 89  ],[ 19  , 71  ],
[ 20  , 68  ],[ 21  , 72  ],[ 22  , 92  ],
[ 23  , 85  ],[ 24  , 92  ],[ 25  , 65  ],
[ 26  , 61  ],[ 27  , 82  ],[ 28  , 88  ],
[ 29  , 75  ],[ 30  , 70  ],[ 31  , 83  ],
[ 32  , 81  ],[ 33  , 76  ],[ 34  , 82  ],
[ 35  , 71  ],[ 36  , 62  ],[ 37  , 80  ],
[ 38  , 72  ],[ 39  , 65  ],[ 40  , 77  ],
[ 41  , 65  ],[ 42  , 67  ],[ 43  , 83  ],
[ 44  , 63  ],[ 45  , 91  ],[ 46  , 66  ],
[ 47  , 87  ],[ 48  , 74  ],[ 49  , 82  ],
[ 50  , 62  ])
```

Escriba el código necesario para:

- Crear una matriz con nombre PesosJugadores.
- Obtener de la matriz el peso del jugador #45.
- Obtener de la matriz los datos del jugador #35 hasta el #48.

10.6. Estadística Básica

Ya que se aprendió a utilizar NumPy, todo se reduce a hacer uso de las funciones contenidas en el paquete, aquí un ejemplo de las funciones básicas de estadística descriptiva contenidas en NumPy.

```
import numpy as np

matriz=np.array([[54,15,19,52,63,20],
                 [1,38,28,32,3,24],
                 [11,14,63,24,35,57],
                 [42,54,18,11,48,17],
                 [58,31,48,1,34,64]])
print(matriz)

#Promedio
print(np.mean (matriz))    #Toda la matriz
print(np.mean (matriz[0])) #Solo al primer renglon
print(np.mean (matriz[2:]))#Del 3 renglon en adelante

#Media
print(np.median (matriz))    #Toda la matriz
print(np.median (matriz[0])) #Solo al primer renglon
print(np.median (matriz[2:]))#Del 3 renglon en adelante

#Desviacion Estandar
print(np.std (matriz))    #Toda la matriz
print(np.std (matriz[0])) #Solo al primer renglon
print(np.std (matriz[2:]))#Del 3 renglon en adelante

#Coeficiente de Correlacion
print(np.corrcoef (matriz))    #Matriz de correlacion
```

```
print(np.corrcoef(matriz[0],matriz[2]))  #Correlacion entre el primer y 3er
    renglon
print(np.corrcoef(matriz[:,0],matriz[:,2])) #Correlacion entre la primera y
    tercera columna
```

```
Out[8]:
[[54 15 19 52 63 20]
 [ 1 38 28 32  3 24]
 [11 14 63 24 35 57]
 [42 54 18 11 48 17]
 [58 31 48  1 34 64]]

Out[11]:32.6333333333
Out[12]:37.1666666667
Out[13]:35.0

Out[16]:31.5
Out[17]:36.0
Out[18]:34.5

Out[21]:19.4944151547
Out[22]:19.5227787184
Out[23]:19.5590274696

Out[26]:
[[ 1.          -0.72318422 -0.42172958  0.14922015 -0.32334165]
 [-0.72318422  1.          0.15608464 -0.3426786  -0.40108423]
 [-0.42172958  0.15608464  1.         -0.59138129  0.37394718]
 [ 0.14922015 -0.3426786  -0.59138129  1.          0.11016353]
 [-0.32334165 -0.40108423  0.37394718  0.11016353  1.        ]]

Out[27]:
[[ 1.          -0.42172958]
 [-0.42172958  1.        ]]

Out[28]:
[[ 1.          -0.27447197]
 [-0.27447197  1.        ]]
```

Ejercicio 7: Complete de manera correcta el siguiente código:

```

prom= _____________________________
print("El promedio entre los datos del tercer renglon es " +
    _____________________)


media= _____________________
print("La media entre los datos de la cuarta y quinta columna es " +
    _________________)


Corr=np.corrcoef(matriz[3],matriz[0])
print(
    ____________________________________________________________________________)


desv=
print("La _______________ de los datos del ultimo renglon es  " + str(desv**2))
```

10.7. Soluciones

```
1
2 #Respuesta 1:
3
4     import math
5
6
7
8
9     C = 2*math.pi*0.43
10
11    print(C)
12
13
14
15
16    A = math.pi*(0.43**2)
17
18    print(A)
19
20
21
22    R="El area de la circunferencia es: "+ str(C)+" y el area de la
      circunferencia     es:"+str(A)
23
24    print(R)
25
26
27 #Respuesta 3:
28    Estaturas= [178, 182, 190, 180, 185]
29
30
31    vector=np.array(Estaturas)
32
33    nuevas_estaturas=vector/100
34    print(nuevas_estaturas)
35    print(type(nuevas_estaturas))
36
37
38 #Respuesta 4:
39    Peso=np.array([69,78,81,95,78])
40
41
42    IMC=Peso / nuevas_estaturas**2
43
44    print(IMC)
45
46
47 #Respuesta 5:
48    Normal=IMC<25
49    print(IMC[Normal])
50
51 #Respuesta 6:
52    PesosJugadores=np.array(Jugadores)
53    print(PesosJugadores[44][1])
54    print(PesosJugadores[34:48])
55
56 #Respuesta 7:
```

```
57
58      prom= np.mean(matriz[2])
59      print("El promedio entre los datos del tercer renglon es " + str(prom))
60
61
62      media= np.median(matriz[:,3:5])
63      print("La media entre los datos de la cuarta y quinta columna es " + str(
        media))
64
65
66      Corr=np.corrcoef(matriz[3],matriz[0])
67      print("El coeficiente de correlacion entre los datos del primer y cuarto
        renglon     es" + str(Corr))
68
69
70      desv=np.std(matriz[-1])
71      print("La varianza de los datos del utlimo renglon es  " + str(desv**2))
```

11 | Resultados

En este capítulo se muestra el proceso de validación de las prácticas propuestas en el presente trabajo, el instrumento utilizado para validarlas, así como la respuesta de los alumnos y las conclusiones generadas.

11.1. Verificación de las prácticas

Se crearon 3 tipos de prácticas:

- **Prácticas con lenguaje R.** Las prácticas R, S, T y U tienen el objetivo de unir las habilidades adquiridas en distintas materias de la carrera de Actuaría, al combinar el conocimiento que se obtiene en ellas del lenguaje R con el curso de Bases de Datos. Las prácticas mencionadas anteriormente contienen la misma estructura, basada en la hipótesis de que el alumno tiene conocimientos previos del lenguaje de programación antes mencionado y consta de 3 partes:

  1. **Objetivos:** describe las metas y expectativas de la práctica.
  2. **Introducción:** explica de manera concreta la teoría necesaria para el desarrollo de la práctica.
  3. **Parte práctica del tema:** desarrolla una aplicación a modo de ejemplo con el lenguaje especificado anteriormente, junto con los códigos e interpretaciones necesarias, utilizando una base de datos que el alumno trabajó a lo largo de curso de Base de Datos.
  4. **Ejercicios:** la práctica concluye con una serie de ejercicios que permiten al alumno implementar y reforzar el aprendizaje adquirido a través de la parte práctica y teórica.

- **Prácticas con el software KNIME.** Las prácticas K y L, tienen el objetivo de dar a conocer al alumno el software KNIME y sus elementos, se busca que el alumno pueda interesarse en aprender a manejar un software nuevo y abra sus posibilidades, ampliando la perspectiva respecto a programas disponibles para aplicaciones estadísticas, manipulación y visualización de datos. Las prácticas mencionadas tienen una estructura semejante a las anteriores:

  1. **Objetivos:** describe las metas y expectativas de la práctica.
  2. **Introducción:** concentra información sobre el programa y sus elementos básicos de manera concreta, así como posibles ventajas y desventajas de su uso.
  3. **Parte práctica:** desarrolla ejemplos del uso de los elementos del programa para análisis y visualización de datos combinados con los conocimientos de bases de datos.

4. **Ejercicios:** al final de la práctica se despliega una lista con ejercicios donde se exhorta al alumno a internarse en el uso del programa, con problemas y preguntas sencillas que permiten aprender a utilizar los elementos del software.

- Finalmente, **prácticas con lenguaje Python**. Las prácticas P y Q, a diferencia de las anteriores, tienen el propósito de enseñar desde cero la estructura y sintaxis del lenguaje Python para su uso en la manipulación de bases de datos. Por esta razón, la estructura de estas prácticas es diferente a las prácticas precedentes:

  1. **Objetivos:** describe las metas y expectativas de la práctica.
  2. **Introducción:** describe el tipo de lenguaje, diferentes aplicaciones, así como opciones para la realización de la práctica y el aprendizaje.
  3. **Parte práctica:** desarrolla un manual instructivo que contiene una serie de explicaciones, ejemplos y ejercicios separados por temas, que permitan al alumno ser autodidacta y avanzar en el conocimiento del lenguaje al ritmo que él desee.
  4. **Respuestas:** ya que se espera que el alumno sea autodidacta, al final de la práctica se presenta una lista con las respuestas de los ejercicios requeridos en la sección práctica.

Cada una de las prácticas al ser creada, era verificada y corregida por dos profesores de la Facultad de Ciencias expertos en bases de datos: el profesor titular de la materia de Bases de Datos y el profesor adjunto. Realizados los cambios pertinentes eran verificadas de nuevo hasta obtener la versión que satisficiera con su estructura, objetivos, temas y habilidades a desarrollar.

Obtenida la versión final de las prácticas se continuó con la socialización de las mismas.

11.2. Socialización de las prácticas

Las prácticas R, S, T, U y K se distribuyeron de manera electrónica en formato PDF a los alumnos del grupo 9149 del curso de Bases de Datos en el semestre 2017-2.

El grupo de Bases de Datos contaba con 61 alumnos inscritos y 8 oyentes. De los 69 totales, 59 pertenecían a la carrera de Actuaría, 8 a la carrera de Matemáticas y 2 a la carrera de Física.

Las prácticas fueron impartidas en clases presenciales una vez a la semana. Éstas eran presentadas por la autora de esta tesis al grupo en cuestión. En cada sesión se contó con el apoyo del profesor titular. La entrega de los ejercicios propuestos en cada práctica fue opcional para los alumnos, con el beneficio de 0.5 puntos extras por práctica sobre sus exámenes parciales.

De los 69 alumnos mencionados anteriormente participaron un máximo de 38 alumnos, como se muestra en la siguiente gráfica:

Figura 11.1: Participación de los alumnos por práctica.

Por cada práctica presentada, los alumnos respondieron una encuesta, la cual fue diseñada para evaluar la utilidad y el contenido de las prácticas.

11.3. Validación de las prácticas

La encuesta que evaluó las prácticas fue realizada considerando la calidad del contenido, el atractivo del mismo y la estructura, secuencia y claridad de la práctica. De esta manera se creó el siguiente cuestionario:

Evaluación post – práctica.

- Nombre:
- Práctica que evalúas:
- ¿Enviaste el entregable correspondiente a esta práctica?:
  - Sí
  - No

I. Esta práctica me proporcionó la teoría necesaria y los ejemplos suficientes para responder satisfactoriamente los ejercicios propuestos.

1. Totalmente de acuerdo
2. De acuerdo

3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

II. Esta práctica me proporcionó los ejemplos suficientes para responder satisfactoriamente los ejercicios propuestos.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

III. El contenido de la práctica me pareció interesante.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

IV. El contenido de la práctica me pareció aplicable y útil para situaciones que podría enfrentar en el ámbito laboral.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

V. La secuencia del texto y los diferentes puntos que componen la práctica están organizados de manera ordenada.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

VI. El lenguaje con el que se explica la teoría es entendible.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

VII. Los ejemplos de esta práctica me parecieron adecuados y entendibles.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

VIII. Los ejercicios son expresados de manera clara.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

IX. Al realizar la práctica, percibo que aprendí algo nuevo.

1. Totalmente de acuerdo
2. De acuerdo
3. Ni acuerdo ni en desacuerdo
4. En desacuerdo
5. Totalmente en desacuerdo

X. Tuve que buscar más información fuera de la presentada en la práctica.

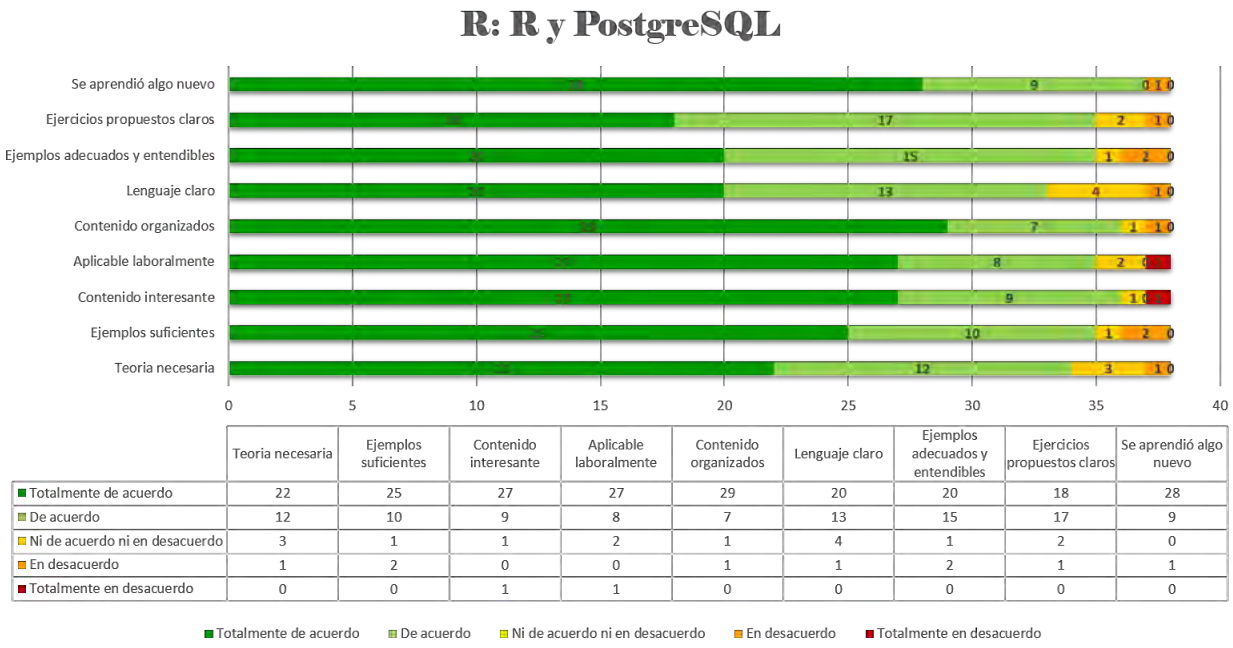
1. Sí
   ¿Sobre qué fue y a qué fuente recurriste? Respuesta abierta
2. No

XI. Comentario y opinión general sobre esta práctica.

Respuesta abierta.

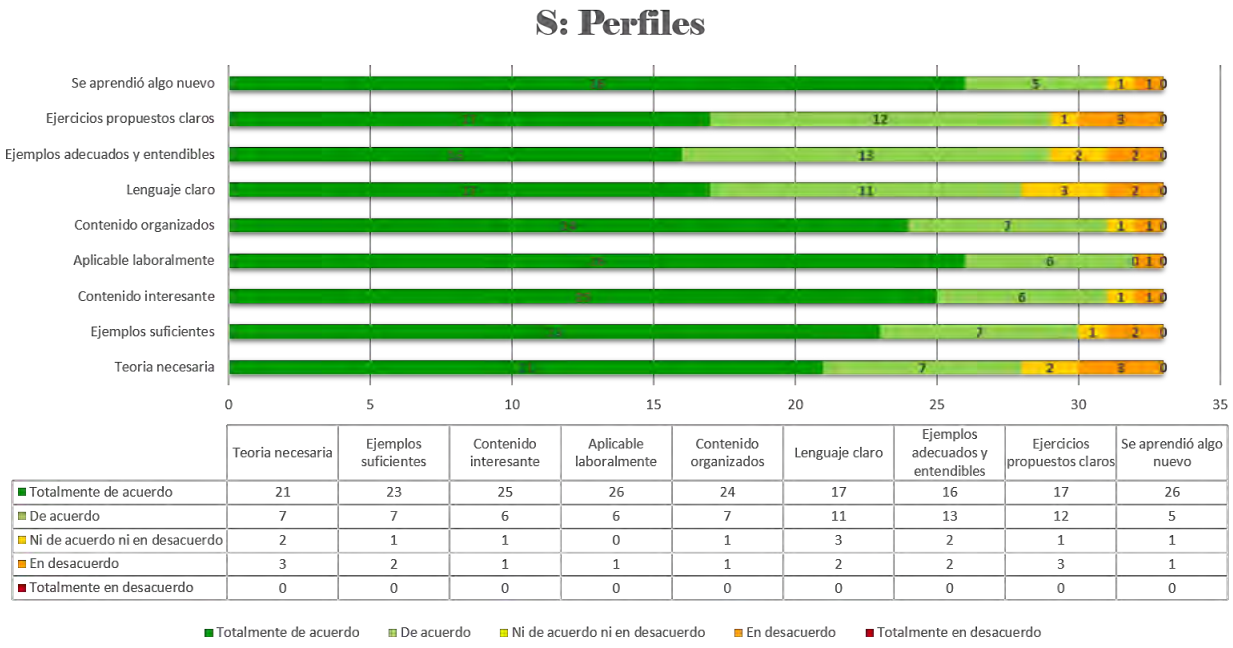
11.4. Recolección y análisis de resultados

Las respuestas del cuestionario creado para la validación fueron almacenadas en una base de datos, la misma que fue utilizada para obtener las siguientes gráficas:



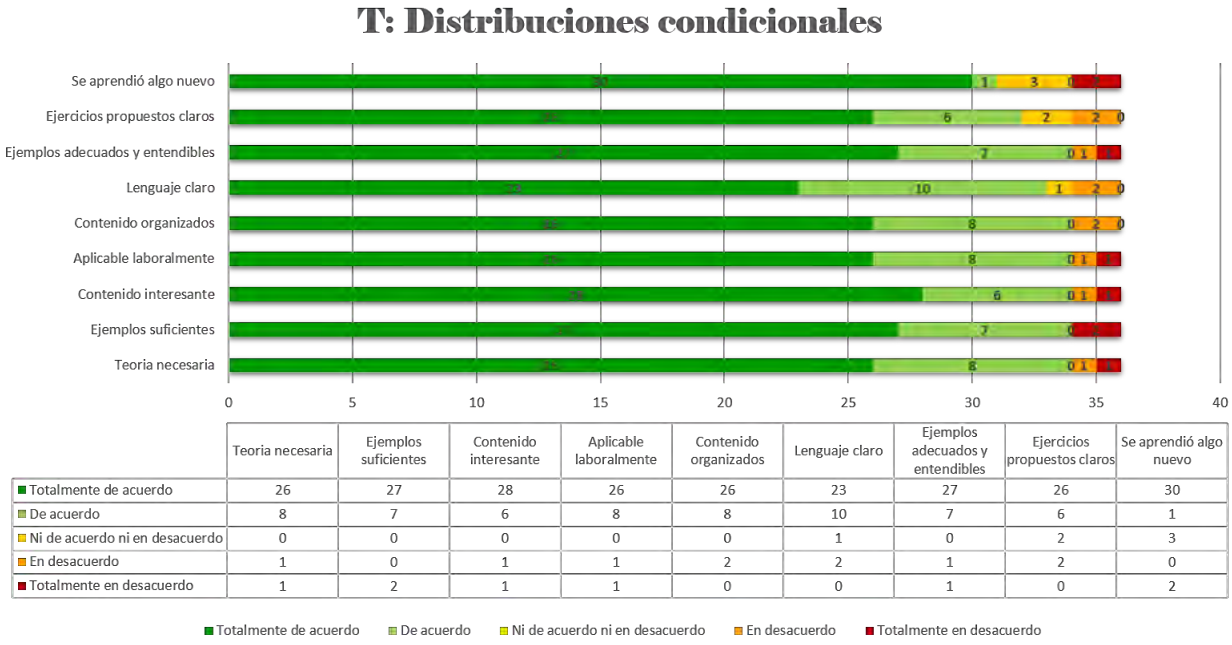
	Teoria necesaria	Ejemplos suficientes	Contenido interesante	Aplicable laboralmente	Contenido organizados	Lenguaje claro	Ejemplos adecuados y entendibles	Ejercicios propuestos claros	Se aprendió algo nuevo
Totalmente de acuerdo	22	25	27	27	29	20	20	18	28
De acuerdo	12	10	9	8	7	13	15	17	9
Ni de acuerdo ni en desacuerdo	3	1	1	2	1	4	1	2	0
En desacuerdo	1	2	0	0	1	1	2	1	1
Totalmente en desacuerdo	0	0	1	1	0	0	0	0	0

Figura 11.2: Resultados de la práctica R.



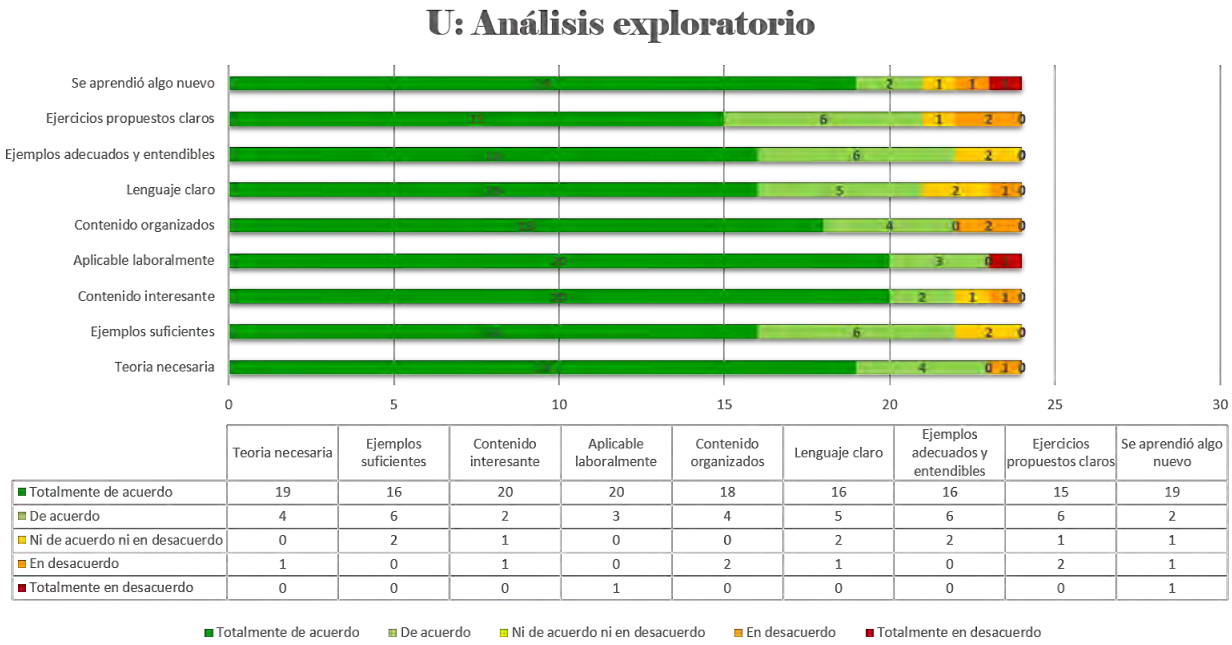
	Teoria necesaria	Ejemplos suficientes	Contenido interesante	Aplicable laboralmente	Contenido organizados	Lenguaje claro	Ejemplos adecuados y entendibles	Ejercicios propuestos claros	Se aprendió algo nuevo
Totalmente de acuerdo	21	23	25	26	24	17	16	17	26
De acuerdo	7	7	6	6	7	11	13	12	5
Ni de acuerdo ni en desacuerdo	2	1	1	0	1	3	2	1	1
En desacuerdo	3	2	1	1	1	2	2	3	1
Totalmente en desacuerdo	0	0	0	0	0	0	0	0	0

Figura 11.3: Resultados de la práctica S.



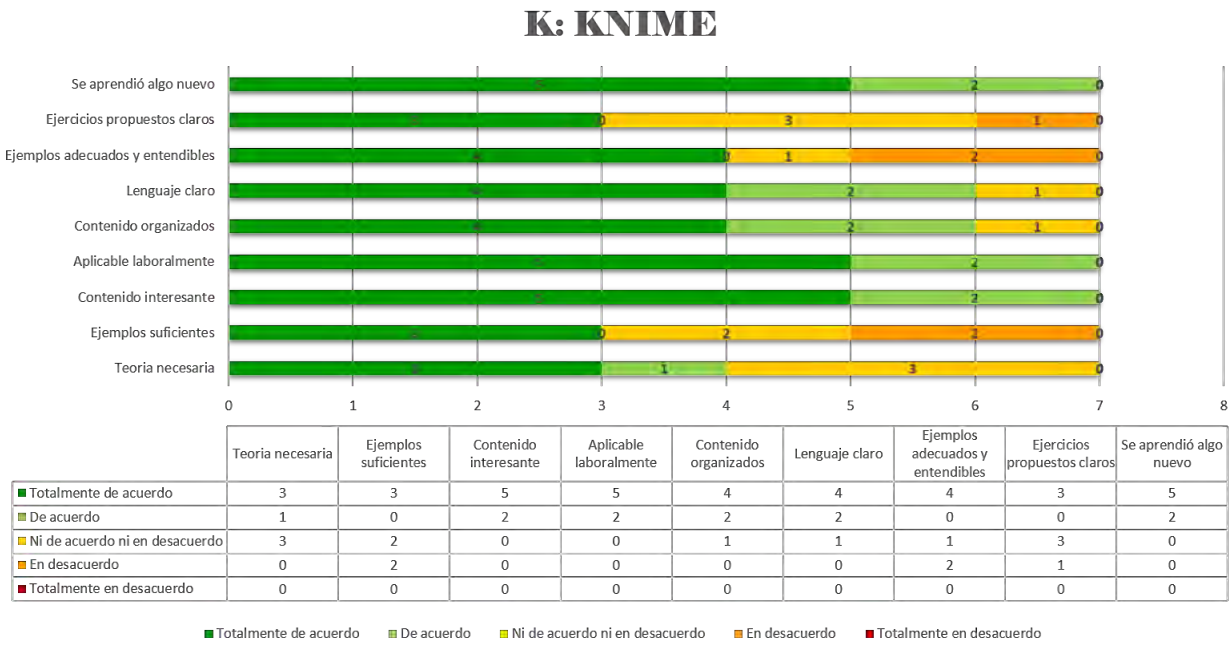
	Teoria necesaria	Ejemplos suficientes	Contenido interesante	Aplicable laboralmente	Contenido organizados	Lenguaje claro	Ejemplos adecuados y entendibles	Ejercicios propuestos claros	Se aprendió algo nuevo
Totalmente de acuerdo	26	27	28	26	26	23	27	26	30
De acuerdo	8	7	6	8	8	10	7	6	1
Ni de acuerdo ni en desacuerdo	0	0	0	0	0	1	0	2	3
En desacuerdo	1	0	1	1	2	2	1	2	0
Totalmente en desacuerdo	1	2	1	1	0	0	1	0	2

Figura 11.4: Resultados de la práctica T.



	Teoria necesaria	Ejemplos suficientes	Contenido interesante	Aplicable laboralmente	Contenido organizados	Lenguaje claro	Ejemplos adecuados y entendibles	Ejercicios propuestos claros	Se aprendió algo nuevo
Totalmente de acuerdo	19	16	20	20	18	16	16	15	19
De acuerdo	4	6	2	3	4	5	6	6	2
Ni de acuerdo ni en desacuerdo	0	2	1	0	0	2	2	1	1
En desacuerdo	1	0	1	0	2	1	0	2	1
Totalmente en desacuerdo	0	0	0	1	0	0	0	0	1

Figura 11.5: Resultados de la práctica U.



	Teoria necesaria	Ejemplos suficientes	Contenido interesante	Aplicable laboralmente	Contenido organizados	Lenguaje claro	Ejemplos adecuados y entendibles	Ejercicios propuestos claros	Se aprendió algo nuevo
Totalmente de acuerdo	3	3	5	5	4	4	4	3	5
De acuerdo	1	0	2	2	2	2	0	0	2
Ni de acuerdo ni en desacuerdo	3	2	0	0	1	1	1	3	0
En desacuerdo	0	2	0	0	0	0	2	1	0
Totalmente en desacuerdo	0	0	0	0	0	0	0	0	0

Totalmente de acuerdo · De acuerdo · Ni de acuerdo ni en desacuerdo · En desacuerdo · Totalmente en desacuerdo

Figura 11.6: Resultados de la práctica K.

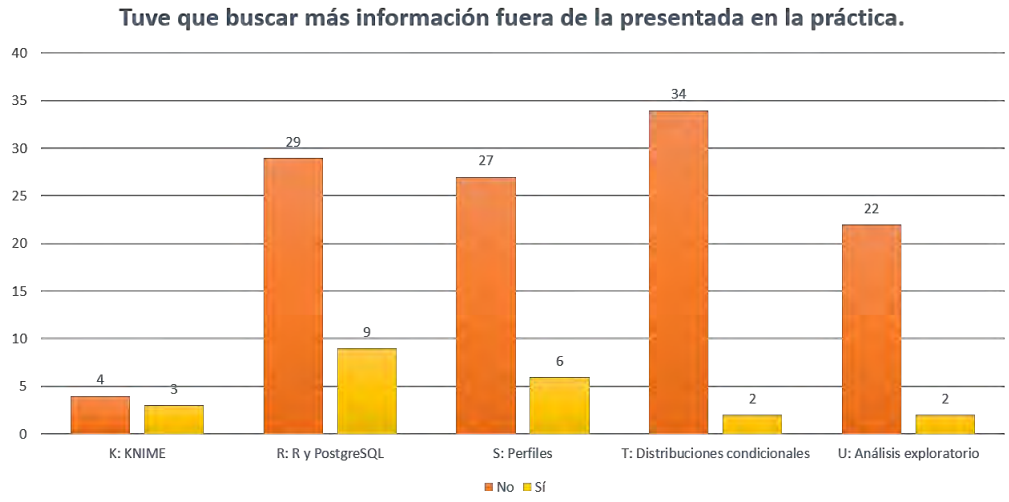


Figura 11.7: Información extra requerida.

11.4.1. Comentarios de los alumnos

Entre los comentarios libres recolectados podemos destacar los siguientes por cada práctica.

I. Práctica R:

- *La práctica me gustó mucho, me parece algo muy útil conectar a R con Postgres y además explicado de manera sencilla. Siento que aprendí algo nuevo y útil.*
- *Una muy buena práctica, sé manejar R pero no tenia idea que R y Postgresql pueden trabajar juntos.*
- *Me hubiera gustado más comentarios en el código.*
- *Muy buena, explicó de manera sencilla la conexión, querys, escritura y las gráficas, sería de gran beneficio para los alumnos contar con más material como este.*

II. Práctica S:

- *Fue una práctica bastante interesante, tiene un buen ejemplo de aplicación para lo aprendido anteriormente, sería perfecto contar con muchos más ejemplos de éstos.*

  Hubo una parte del código que pareció algo confuso sin embargo lo atribuyo a que no he tenido el acercamiento suficiente en R como lo es llevar una práctica de éste estilo, pero es por ésta misma razón que siento que he aprendido algo nuevo y muy aplicable en mi la carrera de Actuaría.
- *Me parece muy buen tema, sin embargo me gustaría hubieran más ejemplos y que hacer cuando tengamos algún error, se que son muchos pero los más comunes podrían ser expuestos.*
- *Buena práctica, todo muy claro, el ejemplo fue interesante. Usé el libro Practical Nonparametric Statistics de W. J. Conover para repasar el tema de tablas de convergencia.*
- *Falto explicación sobre los comandos de la gráfica, color, dominio, etc. Los comentarios al código no me parecieron suficientes.*

III. Práctica T:

- *Ya entendí mejor el código en R y por lo tanto me gustó más esta práctica.*
- *Se complica si no se cuenta con conocimientos de estadística.*
- *La práctica en general esta bien explicada, sólo algunos parámetros en las funciones que se utilizan que no sabía exactamente lo que hacían hasta que me puse a jugar con los valores y listo.*
- *Excelente práctica, aporta herramientas interesantes a pesar de ser sencillas para empezar a explotar los datos obtenidos de una base de datos.*

IV. Práctica U:

- *Me pareció muy interesante y dinámica esta práctica, me parecieron muy curiosas las gráficas de caras.*
- *Fue bastante útil, a pesar de que no he tenido mucha práctica en R creo que sirve como un buen repaso de probabilidad y también para aprender a realizar trabajos en conjunto con SQL+R.*

- *Conforme fueron avanzando las prácticas de R me fui familiarizando con ellas, con los pdf con las "instrucciones" esta práctica me resulto mas fácil y comprensible en comparación con la primera.*

  Sería bueno ir subiendo el nivel conforme avanzamos en prácticas aunque si uno no tiene tantos conocimientos de estadística se "complica" un poco.
- *Me parecería adecuado agregar el significado de las variables a interpretar, debido a que desconozco sobre el tema de football americano.*

V. Práctica K:

- *Deberían de ser obligatorias las prácticas, aprendí mucho con ellas.*
- *Para ser la primera práctica en ese software le faltó más explicación al PDF y mejores ejemplos. Si no hubiera asistido a la clase de demostración me hubiera resultado aún más difícil. Faltó información no tanto sobre el software y sus elementos sino de como manipularlo.*
- *Es interesante saber que hay muchas más cosas con las que puedes conectar pgAdmin.*
- *Es interesante como este programa puede trabajar con la base de datos y todo lo que en la práctica decía que podía hacer, pero en particular siento que falta un poquito más de explicación porque, aunque el folleto que venía acerca de los nodos era bueno, había cosas que en lo personal no me quedaron muy claras.*

11.5. Interpretación de resultados

Las prácticas fueron creadas para satisfacer necesidades actuariales, donde la materia optativa de Bases de Datos pudiera verse estrictamente relacionada con conocimientos y el software R, comúnmente utilizado dentro del ámbito estadístico. De esta manera, al observar el desarrollo, aplicación y culminación de las prácticas, puede decirse lo siguiente respecto a los resultados de la encuesta realizada y comentarios de los alumnos:

- La práctica R fue realizada por 38 alumnos. El 97 % de los comentarios fueron favorables. Una sola persona opinó que la práctica no disponía de contenido interesante y sólo el 23 % de los participantes requirió información extra, la cual no tuvo que ver con dudas sobre el contenido dentro de la práctica, sino con cuestiones técnicas del SMBD y las paqueterías dentro de R.

- El 95 % de las respuestas calificaron positivamente la práctica S, teniendo las mejores calificaciones las ideas de aprender algo nuevo y la aplicación laboral, estando totalmente de acuerdo los alumnos con estas ideas. Tres personas de las 33 que realizaron ésta práctica consideraron que no hubo la teoría necesaria y que a los ejercicios propuestos les faltaba claridad. Dentro de la práctica, 6 personas buscaron información externa a la que contenía el PDF, dentro de este conjunto un alumno requirió reforzar sus conocimientos sobre tablas de contingencia y 3 buscaron información extra sobre los argumentos en las funciones para implementar gráficas.

- La práctica T fue realizada por 36 personas, de las cuales, 2 necesitaron buscar información fuera de la presentada en la práctica, sin embargo, una de esas personas dijo que buscó los nombres de los equipos de fútbol americano en Internet y otra persona busco más información acerca de las gráficas. Dos personas consideraron que no se aprendió nada nuevo, que no hubo la teoría necesaria o era aplicable laboralmente, sin embargo no hicieron comentarios negativos en la sección abierta.

- Sobre la práctica U, fue realizada por 24 personas. De las 24, 20 mencionaron que estaban totalmente de acuerdo con la idea de que el contenido era interesante y que podía aplicarse laboralmente. Una persona tuvo la opinión totalmente contraria y 2 consideraron que los ejercicios propuestos no eran tan claros. En el desarrollo de esta práctica 2 personas requirieron información adicional a la contenida; uno de ellos sobre estadística y otro sobre la documentación de R.

- Finalmente, la práctica K sólo fue realizada por 7 alumnos. Se sospecha que la respuesta baja ante ella fue por ser la última, estando en días de final del semestre y porque era necesario la instalación de software adicional. Sin embargo, los comentarios sobre ella fueron de interés y 5 de los 7 estuvieron totalmente de acuerdo con que el contenido era interesante, era aplicable laboralmente y hubo un aprendizaje nuevo.

- Los comentarios de los alumnos fueron tomados en cuenta desde la primera sesión de socialización para la mejora de cada práctica, como se explica en la siguiente sección.

11.6. Mejoras identificadas

A través de la exposición de las prácticas dentro de la clase, se pudieron identificar fortalezas y aspectos positivos, como la estructura de éstas. Sin embargo, tras el desarrollo de las mismas por parte de los alumnos, escuchar sus dudas más frecuentes y recolectar los datos de las encuestas, se identificaron dos aspectos a mejorar:

- **Cantidad de comentarios dentro del código de programación**. Este aspecto fue tomado en cuenta prontamente y fue una mejora implementada en cada práctica posterior, ampliando las explicaciones de los códigos.

- **Especificaciones acerca del material a entregar por práctica**. Este punto no fue expresado explícitamente en las prácticas, ya que es un rubro que depende de la estrategia que utilice el profesor que aplique las prácticas.

Ahora bien, tomando en cuenta las diferentes carreras a las cuales pertenecían los alumnos del grupo piloto, se recomendaría que el profesor que desee usar el material didáctico creado en esta tesis, al considerar un contexto diferente al supuesto, pueda facilitar a los alumnos el repaso de probabilidad y estadística contenida en el capítulo dos del presente trabajo.

12 | Conclusiones

El objetivo principal de este trabajo fue la generación de recursos didácticos que complementen el curso de Base de Datos que se imparte en la Facultad de Ciencias de la UNAM. Este trabajo permite al alumno vincular los conocimientos que tiene de probabilidad y estadística con los conocimientos que adquiere durante el curso de Base de Datos. Dicha vinculación se obtiene a través de diversos pasos.

Como primer paso, se repasaron diversos conceptos básicos de las materias de Probabilidad y Estadística. A continuación, se le enseñó al alumno a conectar el programa R, con el SMBD PostgreSQL. Posteriormente, se buscó incentivar a los alumnos a conocer diversas herramientas, vinculando PostgreSQL con R y KNIME. Finalmente, con estos programas se estudiaron las diversas herramientas y recursos que se tienen para la explotación de bases de datos.

Tomando en cuenta lo anterior, se tenía la expectativa de que los alumnos contaran con conocimientos previos de probabilidad y estadística, los que por alguna razón no los tenían, tuvieron una ligera dificultad en llevar a cabo las prácticas. Sin embargo, el diseño de las mismas ayudó para que, conforme los alumnos realizaban las prácticas y los ejercicios, pudieran familiarizarse gradualmente con el nuevo conocimiento y éstas lograron ser una aportación a sus saberes. Asimismo, para el resto de los alumnos que participaron se observó un impacto favorable en el interés por la manipulación y explotación de bases de datos en unión con sus aptitudes como actuarios.

Durante la aplicación de las prácticas fue posible ejemplificar el potencial que otorga integrar conocimiento de bases de datos estructuradas, software de minería de datos y lenguajes de programación como Python o R. De esta manera, se proporcionó al alumno herramientas básicas para la manipulación de datos y la integración de un SMBD con software estadísticos.

Como se pudo apreciar en el capítulo de Resultados, a través de las estadísticas y comentarios, en general se observó una respuesta positiva por parte de los alumnos en cada una de las prácticas presentadas. Lograron cumplirse los objetivos respecto al software KNIME y se consiguió incentivar a los alumnos en la búsqueda de herramientas además de las conocidas previamente.

Como conclusión, podemos decir que este trabajo es una aportación para el curso optativo de Bases de Datos de Actuaría. Los recursos didácticos generados pueden ser utilizados provechosamente por alumnos de diferentes carreras de la Facultad de Ciencias, ya que el material generado en este trabajo resultó ser provechoso para los involucrados, de fácil acceso a los alumnos, ilustrativo sobre aplicaciones reales a las que pueden enfrentarse y, sobre todo, esta propuesta permite integrar tres campos de conocimiento ampliamente usados por los actuarios: Probabilidad, Estadística y Bases de Datos.

Esperamos que este trabajo sirva de base para la creación de cursos especializados en áreas

emergentes y de gran importancia como lo es la Ciencia de Datos y para la cual los actuarios contamos con un perfil privilegiado para incursionar en ella.

Personalmente puedo recomendar que la exposición de las prácticas con R y KNIME sean presenciales, una vez a la semana a partir de que los alumnos comiencen a practicar el lenguaje de consultas SQL dentro del curso de Bases de Datos. De esta manera, los alumnos podrán interesarse no sólo en el manejo de R o KNIME, sino también en mejorar su nivel de conocimiento del lenguaje SQL para poder realizar consultas sin requerir copiarlas de las prácticas.

El tiempo de una semana fue suficiente para que los alumnos llevaran a cabo los ejercicios, recomiendo que los resultados de éstos sean entregados en formato PDF anexando las capturas de sus resultados e interpretaciones y sus respectivos códigos debidamente comentados; de tal forma que si existen interpretaciones incongruentes o diferentes, el profesor pueda recurrir al código del alumno y guiarlos acerca de los posibles errores cometidos para que finalmente él pueda tomarlo en cuenta en casos futuros.

Este trabajo fue una oportunidad para compartir mi gusto por la programación y la estadística, presentando a los alumnos que se acercaron a mí, referencias y consejos sobre mi experiencia en las materias de esta área. La respuesta de los alumnos a las prácticas y el interés generado en aprender más herramientas y técnicas para la explotación de bases de datos ha sido gratificante para mí.

Como trabajo futuro, se hará un seguimiento a los comentarios de los alumnos para mejorar las prácticas creadas. Consideramos importante la implementación de sus comentarios para seguir mejorando el contenido educativo del curso de Base de Datos. Una vez aplicadas las sugerencias propuestas, se solicitará la opinión de una nueva generación de alumnos para examinar si los objetivos de las prácticas se siguen cumpliendo, así como para evaluar aquellas prácticas que no pudieron ser probadas por cuestiones de tiempo (prácticas L, P y Q).

Se concluye la presente tesis, señalando que, dada la rápida evolución de la tecnología, este trabajo deberá ser actualizado periódicamente para incluir software, que en su debido momento, vaya ganando relevancia en el ámbito laboral del actuario.

Bibliografía

[1] Canavos C., G. *Probabilidad y estadística, aplicaciones y métodos.* Editorial McGraw-Hill, 1988.

[2] Cohen Karen, D., Asín Lares, E., Lankenau Caballero, D., and Alanis Davila, D. *Sistemas de informacion para los negocios: Un enfoque para la toma de decisiones.* McGraw-Hill/Interamericana, 2005.

[3] Conover, W. J., and Conover, W. J. *Practical nonparametric statistics.* Wiley New York, 1980.

[4] Elmasri, R., Navathe, S. B., Castillo, V. C., Pérez, G. Z., and Espiga, B. G. *Fundamentos de sistemas de bases de datos.* Addison-Wesley, 2002.

[5] Feller, W., and Everest, S. F. *Introducción a la teoría de probabilidades y sus aplicaciones*, vol. 1. Limusa, 1978.

[6] Gibbons, J. D., and Chakraborti, S. *Nonparametric statistical inference fourth edition, revised and expanded*, vol. 168. Marcel Dekker AG, 2003.

[7] Mendenhal, W. *Introducción a la Probabilidad y Estadística.* Cengage Learning, 1972.

[8] Murray, S., and SPIEGEL, M. *Probabilidad y estadística.* Mc. Graw Hill, 1994.

[9] Rincón, L. *Curso elemental de Probabilidad y Estadística.* Facultad de Ciencias UNAM, 2007.

[10] Ross, S. *A first course in probability.* Pearson, 2015.

[11] Silberschatz, A., Korth, H. F., Sudarshan, S., Pérez, F. S., Cordero, A. G., and Fernández, J. C. *Fundamentos de bases de datos.* McGraw-Hill, 2002.

A | NFL-ONEFA

1. Las especializaciones de los jugadores son receptor (Receiver), corredor (Runner), mariscal de campo (Quarterback), pateador de despeje (Punter), pateador de goles de campo (Kicker), línea ofensiva (O-line) y defensiva (Defense).

2. Los jugadores especializados en defensiva tienen asignadas las siguientes estadísticas: tackles, sacks, ff, ints mayores o iguales que cero. Además los sacks (atrapar a un QB rival) tienen que ser menores o iguales que tackles.

3. Los jugadores especializados en kicker tienen asignadas las siguientes estadísticas: fga (goles de campo intentados), fgm (goles de campo anotados), pct (porcentaje de efectividad), blocked (goles de campo bloqueados) y longest (gol de campo más largo anotado en su carrera).

4. Los jugadores especializados en punter tienen asignadas las siguientes estadísticas: punts (patadas realizadas), avrg (promedio de yardas por patada), net (yardas netas totales), longest (patada más larga realizada en su carrera) e in20 (patadas colocadas dentro de la yarda 20 rival).

5. Los jugadores especializados en quarterback tienen asignadas las siguientes estadísticas: att (pases lanzados), comp (pases completados), tds (pases de anotación), ints (pases interceptados) y yds (cantidad total de yardas obtenidas).

6. Los jugadores quarterback son calificados mediante el rating.

7. Los jugadores especializados en runner tienen asignadas las siguientes estadísticas: carries (número de veces que acarrea el balón), yds (cantidad total de yardas obtenidas), avrg (promedio de yardas por acarreo) y tds (acarreos de anotación).

8. Los jugadores especializados en receiver tienen asignadas las siguientes estadísticas: rec (número de veces que atrapó el balón), yds (cantidad total de yardas obtenidas), avrg (promedio de yardas por recepción) y tds (atrapadas de anotación) mayores o iguales que cero, yds sin restricción. Además tds debe ser menor o igual que rec.

9. Los equipos son identificados por un acrónimo de tres letras.

10. Cada equipo pertenece a alguna división dentro de su conferencia, siendo alguna de las siguientes: Norte, Sur, Este, Oeste.

11. El récord de los equipos está compuesto por tres valores: ganados, empatados y perdidos.

12. Una ciudad tiene un nombre y un estado asociados. El estado es un acrónimo de dos letras.

13. Cada ciudad tiene un clima preponderante asignado y puede tomar los siguientes valores: Templado, Frío, Caluroso.

14. Cada equipo está relacionado con una ciudad.
15. Cada partido es jugado por dos equipos.
16. Un partido tiene asignado un equipo local y uno visitante.
17. El marcador está compuesto por los puntos del local y del visitante.

A continuación, se muestra el diagrama de la base (Figura A.1):

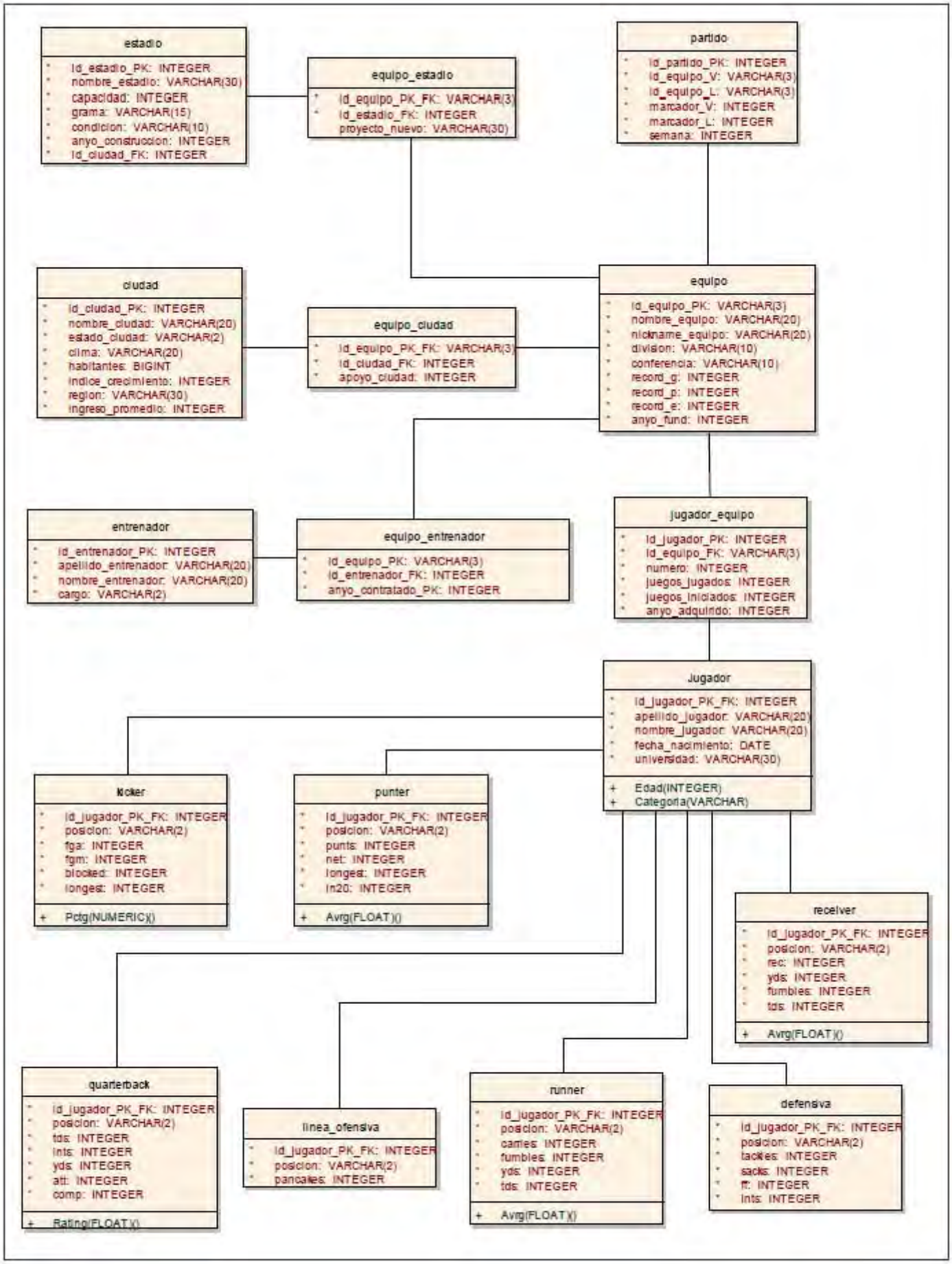


Figura A.1: Diagrama ONEFA