UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN

Aplicación de R como una herramienta de estadística descriptiva para el análisis de una base de datos de un sistema de información académico.

TESIS

QUE PARA OBTENER EL TITULO DE :

Licenciado en Matemáticas Aplicadas y Computación

PRESENTA:

ARACELI RAMÍREZ GARCÍA

ASESOR: M. en C. JOSÉ ANTONIO CORIA FERNÁNDEZ
COASESOR: Dr. SERGIO VICTOR CHAPA VERGARA

MARZO 2010

Agradecimientos

Agradezco a la Universidad Nacional Autónoma de México y a su Facultad de Estudios Superiores Acatlán por brindarme todo el apoyo para concluir con este trabajo.

Sin ningún orden de preferencia agradezco:

- A *Dios* por permitirme una vez más concluir mis metas y conocer a valiosos amigos.
- A mis padres, por el apoyo incondicional que me han brindado para superarme.
- A mis hermanos, por escucharme y motivarme.
- A mis primos Sergio Ramírez Sánchez y Ricardo Aguilar Ramírez, por su apoyo y comprensión que me dieron en todo momento.
- A mi asesor Mtro. José Antonio Coria Fernández, por todos sus consejos, enseñanzas, dedicación y paciencia a lo largo de este trabajo.
- A Javier Garduño Cimental, por tus palabras de aliento y superación durante la elaboración de este trabajo.
- A mis amigos porque de ellos he encontrado el apoyo incondicional para lograr cada uno de mis objetivos.
- A mis sinodales, por sus valiosos comentarios y dedicación en la revisión de este trabajo: Dr. Sergio Víctor Chapa, Mtro. Jaime Ramírez Muñoz, Lic. Sandra Mendoza Ortíz, Mtro. Sergio Alejandro Matías Hernández.

Índice general

Índice de figuras

Introducción

Ante el vertiginoso crecimiento de la información, se hace prácticamente imposible para una persona extraer conclusiones, tendencias y patrones a partir de datos crudos. Por lo que, la visualización de la información tiene un aporte significativo en la exploración de conjuntos de datos, debido a que tiene como objetivo la representación perceptual tanto de estos con parámetros múltiples como de las tendencias y las relaciones subyacentes que existen entre ellos.

Esta tesis presenta el uso del software libre R (bajo la licencia de opensource)[1] empleado como una herramienta para efectuar análisis estadísticos en información almacenada en bases de datos. El propósito de emplear R es lograr una conexión de éste con una Base de datos y de esta manera extraer información y manipularla de tal forma que se pueda construir un análisis descriptivo de la información conseguida. La elaboración de este trabajo contribuye a la automatización para generar indicadores estadísticos[2] que apoyan en la toma de decisiones.

La realización de este trabajo surge de la necesidad del departamento que se encarga de la administración y monitoreo de la Base de datos SINAC [3], de requerir constantemente de reportes estadísticos que indiquen de forma numérica y visual el comportamiento de los datos y que además el software que se utilice para realizar estas tareas sea de ambiente libre ya que SINAC se encuentra montado sobre PostgreSQL. A pesar de que ya existe un trabajo previo [10] “ Sistema de consulta Estadística con un Ambiente Gráfico utilizando Tecnología Dinámica ” que es empleado en la automatización del proceso de obtener información estadística; un requerimiento indispensable es contar con una herramienta puramente estadística que a través de la interacción con la Base de datos sirva de interfaz para proporcionar los reportes estadísticos solicitados. Una alternativa para satisfacer estas necesidades es por medio del empleo de R que es un software que provee de

[1] Software libre. Software de código abierto y no necesariamente gratuito

[2] Medidas de tendencia central, medidas de dispersión

[3] Sistema de Información Académica

diversas técnicas estadísticas y gráficas que además se distribuye gratuitamente. Esta es la razón por la cual se ha desarrollado este proyecto.

El alcance de este proyecto ha sido sólo en el área de estadística descriptiva puesto que el principal problema era obtener resultados que involucraran medidas estadísticas visuales y en este sentido una de las principales ramas de la estadística que aborda gran parte de la estadística visual es la estadística descriptiva.

La solución al problema consistió en la búsqueda de una librería que permitiera la conexión de la Base de datos PostgreSQL con R. Una vez conectada a R se realizaron las consultas requeridas y se pasaron en forma de objetos a R para después manipular estos objetos. Durante este proyecto se realizarón varias pruebas de conversión datos con el objetivo de obtener los tipos de datos con los que se pudieran calcular medidas estadísticas y hacer gráficas de ellas.

Con el desarrollo de este proyecto se logra el uso de software libre como herramienta en el área de estadística computacional, pero también sirve como base para futuros trabajos en la extracción y automatización de información de Bases de datos mediante R.

El uso de software libre presenta las siguientes ventajas: no se incurre en un costo para su adquisición, se tiene a disposición el código fuente de los programas (lo que da la flexibilidad de realizar modificaciones para su adaptación), se tiene soporte para distintas plataformas y se cuenta con un gran número de fuentes de información mundial (páginas Web, foros de discusión, listas de suscripción, grupos de desarrollo, entre otros).

Éste procedimiento de conexión de base de datos con R puede ser usado para conectar cualquier otra base de datos sin importar que sistema manejador de base de datos se requiera de forma que, se pueda procesar y obtener medidas estadísticas de una gran cantidad de información contenida en una consulta en el menor tiempo posible. Esta tesis se encuentra organizada de la siguiente manera:

El **capitulo 1** presenta información teórica como, conceptos y fórmulas de estadística descriptiva. También se expone un ejemplo práctico que muestra el procedimiento de cálculo para medidas estadística descriptiva.

El **capítulo 2** presenta los elementos teóricos de R: tipos de objetos, funciones estadísticas y funciones gráficas.

El **capítulo 3** describe la Base de datos académica SINAC; su diseño conceptual, relacional y descripción de los metadatos.

El **capítulo 4** presenta el desarrollo de la tesis: conexión de la Base de datos, se realizan consultas y se ejecutan en R como un objeto para conseguir análisis estadísticos con ellas. También se presentan las conclusiones obtenidas.

Capítulo 1

Fundamentos teóricos de estadística descriptiva

La estadística descriptiva es una disciplina encargada de la aplicación de métodos para coleccionar datos estadísticos, que pueden estar ordenados de diversas formas por ejemplo, en tablas que representan medidas numéricas o gráficas las cuales son herramientas que permiten el análisis del comportamiento de este tipo de datos. De esta forma permite realizar deducciones para la toma de decisiones. Estos datos estadísticos pueden ser derivados de una infinidad de acontecimientos; el precio del petróleo en un determinado periodo de años, el nivel de desnutrición en una determinada población, el número de muertes causadas por accidentes automovilísticos, por citar algunos ejemplos. Para obtener el resultado del comportamiento de cada uno de los ejemplos anteriores es necesario realizar observaciones cualitativas o bien cuantitativas (dependiendo del tipo de datos) de los mismos, pero antes es necesario tener presente algunos conceptos teóricos de estadística descriptiva.

A continuación se presentan algunos conceptos teóricos de estadística descriptiva útiles.

- Dato. Es un objeto o sujeto con características o atributos.
- Los **individuos** son personas, animales u objetos descritos en un conjunto de datos.
- Observación atípica. Observación que no comparte las mismas características del resto de las observaciones.

- Población. Es un conjunto de datos(individuos, objetos, medidas) que comparten alguna característica en común observable, se denota con una **N** mayúscula.
- Muestra. Es un subconjunto de datos representativos de una población, se denota con **n** minúscula.
- Categoría. Conjunto de características o atributos que comparte una agrupación de datos en especial.
- Variable. En estadística una variable es cualquier característica observable o medible en un individuo, que puede adoptar diferentes valores en cada uno de los casos de estudio. Las variables se clasifican según su escala de medición, esta depende del tipo de variable.
- Variables cualitativas o categóricas. Este tipo de variables no se pueden operar numéricamente, sin embargo, son útiles para describir un conjunto de datos que pertenecen a un grupo particular. Se clasifican en ordinales y nominales.
    - Ordinales. Es la variable cualitativa en donde los valores adoptan un orden jerárquico establecido.
    - Nominales. Es la variable cualitativa en donde los valores no toman ningún orden o jerarquía.
- Variables cuantitativas o intervalares. Son aquellas a las cuales posible asignar valores numéricos y realizar operaciones aritméticas. Estas se clasifican en discretas y continuas.
    - Variable discreta. Es aquella en la que se puede contar su conjunto de valores posibles, es decir, toman valores enteros.
    - Variable continua. Es la variable en la que el conjunto de sus posibles resultados puede tomar valores infinitos en algún intervalo dado.
- Frecuencia. Es la colección del número de observaciones que pertenecen a cada una de las categorías de un conjunto de datos en cuestión.
- Frecuencia absoluta. Es el numero de veces que se presenta una determinada categoría o valor de una variable, la suma total de de frecuencias absolutas es igual al tamaño de la población en observación.

- Frecuencia relativa. La frecuencia relativa de una categoría dada es la proporción entre el número total de veces que se presenta esa categoría y el tamaño de la población.

- Frecuencia acumulada absoluta. Es la frecuencia absoluta de esa categoría más la frecuencia absoluta de las categorías anteriores o menores a la misma.

- Frecuencia acumulada relativa. Es la suma de todas las frecuencias relativas de todas las categorías menores o iguales a dicha categoría entre el tamaño de la población.

- Ojiva. Es la representación gráfica de una distribución acumulada, se obtiene tomando los límites de la clase (eje horizontal) y las frecuencias acumuladas, frecuencias relativas acumuladas o frecuencias porcentuales acumuladas en el eje vertical. La ojiva permite ver cuántas observaciones se encuentran por encima o debajo de ciertos valores.

- Polígono de frecuencias. Es una representación gráfica de una distribución de frecuencias. Este se elabora a partir de los puntos medios de cada clase localizados en las tapas superiores de los rectángulos utilizados en los histogramas de las gráficas.

- Sesgo. Es el grado de asimetría en una distribución.

- Asimetría o simetría. Es la falta de simetría en una distribución. La asimetría se mide con el sesgo, de este modo se puede observar que tipo de asimetría tiene la distribución.

    - Simétrica. La distribución queda dividida en dos partes iguales con la misma frecuencia tato a la izquierda como a la derecha.
    - Asimétrica negativa o sesgada a la izquierda. La distribución tiene una cola más larga a la izquierda.
    - Asimétrica positiva o sesgada a la derecha. La distribución tiene una cola más larga ala derecha.

- Curtosis. Es una medida de forma o apuntamiento que proporciona información acerca de como se distribuyen los datos en relación a una distribución normal. Se definen tres tipos de distribuciones dependiendo de su grado de curtosis:

- Distribución mesocúrtica, es simétrica en forma de campana
- Distribucion leptocúrtica, es más apuntada (picuda).
- Distribución Platicúrtica, es menos apuntada (plana).

- Coeficiente de curtosis, analiza el grado de apuntamiento de una distribución.

1.1. Métodos que sirven de apoyo en la interpretación descriptiva de datos

La presentación de datos estadísticos es una tarea importante dentro de la estadística descriptiva, ya que posibilita organizar, analizar y simplificar la información de forma rápida y accesible. Los métodos utilizados para presentar información estadística son: el método gráfico y el método numérico.

1.1.1. Método gráfico

El método gráfico es una técnica visual que nos permite analizar y hacer juicios respecto a las características de los datos, tales como la variabilidad y la tendencia; entre las técnicas visuales existentes se encuentran las tablas de frecuencia, histogramas, diagramas circulares, gráficas de barras, diagramas de dispersión entre otras. Las gráficas más frecuentes para la descripción de datos cualitativos son: las gráficas de barras y las gráficas circulares, mientras que en la descripción de datos cuantitativos es más usual emplear el histograma.

A continuación se da una breve descripción de las características de cada una de las técnicas visuales mencionadas anteriormente.

- **Tablas de frecuencias**. Es una tabla que tiene como finalidad clasificar y organizar los datos de acuerdo a la categoría a la que pertenecen, indicando cuantas veces se repite la variable en cada categoría de esta forma, se puede obtener tanto la frecuencia absoluta como la frecuencia relativa. Una tabla de frecuencias también es conocida como una distribución de frecuencias y al igual que los histogramas también existen diferentes tipos de distribuciones de frecuencias:

  - **Distribución de frecuencias simples**. Es el número de veces que se presenta un número en un conjunto inicial de datos organizado de menor a mayor.

- **Distribución de frecuencias por intervalos**. Es la organización de los datos mediante intervalos o clases.
- **Distribución de frecuencias acumuladas**. Muestra la suma de frecuencias para cada uno de los límites superiores de cada clase.
- **Distribución porcentual acumulativa**. Indica el porcentaje de la totalidad de los datos que son menores o iguales que un límite superior dado

- **Diagramas circulares o de pastel**. Ayudan a interpretar la frecuencia relativa (porcentajes) de la clase a partir del ángulo central de cada rebanada del pastel.

- **Gráficas de barras**. Es un grafo que representa las frecuencias asociadas a las categorías de un conjunto de datos.

- **Diagramas de dispersión.** Son útiles para analizar si existe relación entre dos variables cuantitativas y así determinar el tipo de relación.

- **Histograma**. Es el método gráfico para representar frecuencias a través de la altura de las barras y realizar comparaciones de las mismas, los histogramas se consideran simétricos si su mitad izquierda es exactamente igual a su mitad derecha.

  Los histogramas pueden tomar diferentes formas dependiendo del conjunto de datos en cuestión:

  - **Unimodal**. Representa la frecuencia con que ocurre un valor en una observación de un conjunto de individuos del mismo tipo; gráficamente se percibe como una campana, es decir, sólo existe un pico en el histograma. Este tipo de histogramas puede tener sesgo positivo si su cola derecha, está muy alargada en comparación con cola izquierda, y tiene sesgo negativo si esta muy extendida hacia la izquierda.
  - **Bimodal**. Se genera cuando se encuentran dos valores distintos de individuos u objetos que ocurren con mayor frecuencia en un conjunto de datos, por lo que la gráfica que se forma tiene dos picos diferentes. Por ejemplo, un histograma de calificaciones de la materia de estadística 1 de los alumnos de MAC presentaría un pico a determinada altura representativa para los alumnos que si estudiaron, y otro con la altura que representa a los alumnos que no estudiaron.

- **Multimodal**. Es un histograma con más de dos picos.

1.1.2. Elaboración de tablas de frecuencias e histogramas

Las tablas de frecuencias como los histogramas son dos herramientas de mayor uso dentro de la estadística descriptiva y son dependientes una de la otra, esto es la construcción de un Histograma de frecuencias depende de la elaboración previa de la tabla de frecuencias de los datos en observación. En este contexto, es importante tener presente que existen diferentes alternativas para calcular el número de clases que serán utilizados en el momento de elaborar una tabla de frecuencia; la elección de esta depende del criterio de quien este realizando los cálculos estadísticos. A continuación se mencionan algunas de las opciones que permiten calcular el número de clases:

- Regla de Sturges $k = 1 + 3{,}322 log n$
- Raíz cuadrada de los datos $\sqrt{n}$
- Regla empírica. Si se realizan pocas observaciones se pueden emplear 5 clases aproximadamente si son muchos pueden ser, 8, 10, 12, ó 15 no más.

1.2. Métodos numéricos

Como se mencionó en el tema 1.1.1, es posible analizar información acerca de los datos a través de métodos gráficos, sin embargo, también es de gran ayuda disponer de otra clase de métodos como son los métodos numéricos. Debido a que, los métodos numéricos tienen como objetivo presentar información resumida y cuantificada que describa las características del conjunto (no importa si es un conjunto de datos agrupado o sin agrupar) de datos en estudio.

1.2.1. Medidas de tendencia central

Las medidas de tendencia central son medidas descriptivas numéricas provenientes de un conjunto de datos, que permiten crear un panorama precedente respecto a la localización de los datos. Además ayudan a obtener un análisis más formal a través de la interpretación de un sólo valor que represente a todos los datos. Las medidas de tendencia central más conocidas son la media aritmética, la mediana y la moda.

Media o media aritmética. Es el valor promedio aritmético para un conjunto dado de **n** observaciones, $x_1, x_2, ...x_n$.

Una característica de la media es su sensibilidad(puede variar) a la influencia de por lo menos un dato atípico, ya sea muy grande o muy pequeño; cuando esto sucede la media no es el valor apropiado para el calculo de un promedio. En este caso la mediana puede ser una buena alternativa.

La media poblacional se denota mediante la letra griega μ y es el promedio de todas las **N** observaciones de un población.

Mediana. Es el punto medio de las n observaciones ordenadas de menor a mayor magnitud o viceversa y comúnmente se describe con el símbolo $\tilde{x}$.

1. Si n es impar existe un único valor medio, es decir, un 50 % de los datos se encuentra por debajo de este valor y el otro 50 % de los datos se encuentra por arriba del mismo.

2. Si n es par existen dos valores medios que se promedian, la mediana es útil cuando se encuentran datos atípicos en una muestra que no pueden ser tratados con la media.

Moda. Es el valor u observación que se repite con mayor frecuencia en un conjunto de n observaciones, $x_1, x_2, ...x_n$, en términos gráficos es el pico más alto de la distribución de frecuencia relativa. Al igual que la mediana, la moda puede utilizarse cuando existe la presencia de valores atípicos pero también es utilizada cuando el objetivo es conocer el valor más común en una distribución.

1.2.2. Medidas de dispersión

Aún cuando las medidas de tendencia central permiten la obtención de características importantes de los datos, no logran expresar ciertos aspectos como, la distancia de los datos respecto a un valor central o la concentración de datos en un determinado recorrido de los datos. En términos generales las medidas de dispersión miden la distancia de los datos respecto a la media; de esta forma, cuando se tiene un valor observado mayor que la media, la dispersión será positiva, por lo contrario, se obtendrá una desviación negativa, encontrando así la variabilidad de los datos observados respecto a su media. Las medidas de dispersión de uso más frecuente son: rango o recorrido, la desviación media y la desviación estándar.

La elección de alguna de las medidas de dispersión depende de que tan buena es esta para representar de la mejor forma los datos en una situación dada por

ejemplo, cuando se tienen tienen distintas muestras o poblaciones la media puede ser idéntica, sin embargo, la dispersión de los datos respecto a la media puede variar, en consecuencia se podrá realizar un estudio más detallado con las medidas de dispersión.

Rango o recorrido. Es la diferencia entre el dato mayor y el dato menor en un conjunto de datos. Una desventaja del intervalo es que sólo depende de las dos observaciones extremas por tanto, se pierde fidelidad respecto al grado de variación, sin embargo, es una forma sencilla y rápida de obtener una idea de cuanta variación existe en los datos.

Desviación media o desviación absoluta media

Son las desviaciones calculadas mediante la diferencia de cualquier valor de un conjunto de datos dado con respecto a la media. La desviación media para datos no agrupados, es el promedio en valor absoluto de las distancias entre los datos y la media, es decir, se suman todas las desviaciones positivas;en este caso si todas las desviaciones son iguales la dispersión será nula, de lo contrario, mientras la desviación media absoluta sea mayor, mayor será la dispersión; esta medida de dispersión es poco usual debido a su complejidad para realizar operaciones algebraicas, como consecuencia es preferible emplear otra medida de dispersión como la varianza.

Varianza. La varianza es la media aritmética de las desviaciones de los datos obtenidos de la variable con respecto de su media aritmética elevados al cuadrado, es decir, las desviaciones,$(x_1 - \bar{x})^2, (x_2 - \bar{x})^2, (x_n - \bar{x})^2$ son elevadas en unidades cuadradas con el objetivo de que sean positivas y que se eviten dificultades en los cálculos. La varianza muestral es utilizada en el análisis de muestras y la varianza poblacional en poblaciones.

Desviación estándar (se denota por σ o s). Es la medida de dispersión, correspondiente a la diferencia entre cualquier valor y la media aritmética, es el cuadrado de la varianza.

1.2.3. Medidas de localización o Posición relativa

Las medidas de localización o medidas de posición relativa son aquellas que ayudan al tratamiento de datos atípicos que afectan tanto a la media como a la mediana. Las medidas de localización dividen en partes iguales las observaciones de una distribución que se encuentra ordenada de menor a mayor, con el objetivo de encontrar un valor para el cual una porción especifica de la distribución queda contenida o por debajo de dicho valor, dependiendo del número de divisiones los

cuantiles se clasifican en : cuartiles, deciles, percentiles y quintiles, siendo los cuartiles, deciles y percentiles los más usados.

Cuartiles . Dividen en cuatro partes iguales un conjunto de datos, es decir, se encuentran tres valores que se indican de la siguiente manera:

- Primer cuartil Q_1. Representa el 25 % de los datos.
- Segundo cuartil Q_2. Es la mediana, por tanto es el valor que representa el 50 % de los datos.
- Tercer cuartil Q_3. Representa el 75 % de los datos.

Deciles D_k. Son nueve valores que dividen a los datos en diez partes iguales, por tanto cada una de las partes representa un 10 % de los mismos, el quinto quincuagésimo percentil corresponden también a la mediana .

Percentiles P_k. Representan 99 valores que dividen los datos en 100 partes iguales.

Quintiles . Son cuatro valores que dividen a los datos en cinco partes iguales.

Capítulo 2

Conceptos básicos de R

R es un proyecto iniciado por Robert Gentleman y Ross Ihaka del Departamento de Estadística de la Universidad de Auckland en 1993. R comenzó como un experimento tratando de usar métodos de Lisp implementados para elaborar un pequeño banco de pruebas que se podría utilizar como ensayo para construir un ambiente estadístico. A principios de este proyecto, se tomó la decisión de utilizar la sintaxis de S, lenguaje desarrollado por los laboratorios Bell. En Agosto de 1993 se utilizarón algunas copias binarias de R en Statlib y se anunció esta noticia por medio de la lista de correo de S-noticias; fue así como un número de personas cogió los códigos binarios y ofreció su regeneración. El más persistente de estos fue Martin Mächler, quien en Junio de 1995 sugirió lanzar el código fuente de R como "software libre"; bajo los términos de la GNU General Public Licence.
Actualmente R se encuentra bajo responsabilidad del grupo R Development Core Team y aún sigue siendo un proyecto vivo, debido a que, sus usuarios tienen el permiso de modificar o extender el código fuente dependiendo de las necesidades que a los mismos convenga, de esta forma contribuyen al desarrollo e innovación dinámica en diversas áreas de la investigación científica. R es un esfuerzo de colaboración por medio de Internet para intercambiar ideas y obtener resultados, esto se lleva acabo mediante foros de discusión y listas de correos.

2.1. Características de R

La instalación de R se puede realizar de dos formas en los diferentes sistema operativo existentes: una por medio de comandos en la consola, cabe mencionar que con esta opción se requiere del servicio de Internet; la otra forma se realiza

desde un disco de instalación, con R previamente descargado. En linux la ejecución de comandos es necesario si no se desea trabajar con alguna interfaz gráfica y cuando se requiere de la instalación de librerías adicionales, en cuyo caso se utilizan los siguientes comandos.

- R -g TK o sudo R y después el comando install.packages("nombre")

También existen interfaces gráficas (GUI) que permiten manipular datos de forma amigable, es decir, se encuentran menús para cargar paquetes, funciones, obtener resúmenes estadísticos, distribuciones de probabilidad y generar gráficas. Sin embargo, la instalación de estas GUIS se realizan de forma independiente ya que, el paquete básico de R no las incluye. Para efectos de este proyecto se trabajo por medio de comandos en consola, por lo que, no existe ningún menú de ayuda visible.

Tipos de estadísticas con R	
Tema	**Paquete**
Tablas de Frecuencias	agricolae
Distribuciones discretas	stats
Distribuciones continuas	stats
Inferencia Estadística	stats
Simetría y curtosis	FBasics
Analis multivariado	mva
Modelos de regresión	R, Hmisc, Design, lasso, VGAM, pda
Modelos lineales y no lineales	nlme
Numéros aleatorios	Survey
Arboles de decisión	ada, Bayes Tree, rpart randomForest, rpart.permutation
Regresiones	rpart, randomForest, rpart.permutation
Análisis de correspondencia	MASS
Analisis de sobrevivencia	survival
Estadística Espacial	spatial

2.1.1. Comparación de R contra otros lenguajes

La siguiente tabla muestra una comparación de R respecto a otros softwares estadísticos.

Requerimientos	R 2.7	SPSS 17	S-PLUS 8
Costo	$0	$27,800	136,363.6
S.O	Linux, Unix Windows Mac OSX	Windows Red Hat Enterprise Mac OSX10.4 Debían 4.0	Windows Sun Solaris 2.8 Sun Solaris 2.9 Sun Solaris 2.10 Red Hat Enterprise
Espacio de Almacenamiento	180.4 MB	312.7 MB	994 MB
Acceso a Bases de datos	Acces, Oracle SQL Server PostgreSQL, MySQL	Oracle, SQL Server DB2, Acces	Oracle, SQL Server Acces
Visualización (GUI)	R Commander, RKWard JGR, pmg	SPSS Base	Interfaz gráfica de usuario

2.2. Objetos, expresiones y modos

Los objetos en R son variables sobre las cuales se realizan diferentes operaciones, tales como, asignación de valores, parámetros, etc. Éstos objetos tienen dos características primordiales, el nombre y contenido; estas dos características estarán descritas por atributos que a su vez se caracterizan por tener dos propiedades: tipo (numérico, carácter, complejo y lógico) y longitud (número de elementos en el objeto).

Los objetos que son creados en cada sesión de R se almacenan en un archivo y forman un espacio de trabajo (workspace) para que posteriormente puedan ser utilizados; una vez que se termina una sesión de R, ofrece la opción de salvar el espacio de trabajo (objetos) creados en la sesión, el usuario determina si los salva o no según sea su beneficio. Existen tres funciones básicas para el uso de objetos, ***mode(objeto)***, ***str(objeto)*** y ***length(objeto)***, la primera función permite conocer el tipo de objeto, en el caso de ***str*** despliega la información contenida en el mismo, y la tercera función permite saber la longitud del objeto con el que se esta trabajando.

Las expresiones en R son variables, operadores, llamadas a funciones, entre otras expresiones; la forma en que se trabaja con las expresiones es evaluando cada una de estas con el objetivo de obtener un resultado y analizar el efecto del mismo, para que una expresión se ejecute necesariamente tiene que interactuar sobre el objeto con el que se este trabajando que como ya se mencionó pueden ser de diferentes tipos.

Un modo en R se refiere a los tipos de atributos que constituyen un objeto, por ejemplo, si declaramos un vector de números entonces el modo o tipo de atributos que constituyen el objeto es numérico, sí el vector es de caracteres entonces el modo es carácter.

2.2.1. Operadores en R

La siguiente tabla muestra los operadores que existen en R, para más información consultar [18].

Aritméticos	**Comparativos**	**Lógicos**
+ Adición	<menor que	! x negación NOT
- Sustración	>mayor que	x & y; Y lógico AND
* Multiplicación	<= menor o igual que	x && y; Y lógico
/ División	>= mayor o igual que	x | y O lógico
Ôotencia	== igual	x || y O lógico
% % Módulo	!= diferente	xor(x, y) O exclusivo
% / % División entre enteros		
%* % Producto interno		

2.2.2. Tipos de Objetos

Las clases de objetos que se trabajan en este lenguaje son:

1. Vectores.
2. Matrices y arrays(arreglos).
3. Factor.
4. Lista.
5. Data frame o marco de datos.
6. Funciones.

Vectores

Un vector es una variable, cuyos elementos son del mismo tipo,los elementos de un vector pueden ser de tipo numérico, carácter ó lógico. En R los vectores pueden ser manipulados con operaciones simples de aritmética, la forma más simple para crear un vector es por medio de la función ***c()*** (acrónimo de concatenación) que permite concatenar vectores. Un vector carácter debe ir especificado entre comillas simples o dobles para que pueda ser reconocido como una cadena de ; en este trabajo se hace uso de las dobles comillas (“ “). Los vectores lógicos pueden tomar los valores Falso (F) o Verdadero (V).

Ejemplo 1. Generar un vector de 3 elementos tipo caracter.

```
 c("a", "b", "c")
  [1] "a" "b" "c"
```

Ejemplo 2. En el siguiente ejemplo, primero se crea el vector **x** y **y**, posteriormente se genera un vector llamado **z** donde los argumentos son el vector **x** y **y**, el resultado es la concatenación de los mismos.

```
 x <- c("Javier")
 y <- c("Araceli")
 x
[1] "Javier"
 y
[1] "Araceli"
 z <- c(x,y)
 z
[1] "Javier"  "Araceli"
```

Una observación importante es que la concatenación de los vectores debe ser entre variables del mismo tipo, si se concatenan variables de tipo numérico con variables caracter la concatenación se realiza obligando a las variables a ser de un mismo tipo sin marcar error, en este caso la salida es el tipo más alto de los componentes, sin embargo, el resultado no es el deseado.

Ejemplo 3. En el siguiente ejemplo, se crea la variable ***x*** de tipo caracter con 3 elementos y la variable ***y*** de tipo numérico con 2 elementos; el resultado de la concatenación es el vector ***z*** de tipo caracter, esto se debe a que el número de elementos de tipo caracter es mayor al número de elementos de tipo numérico.

```
  x<- c("a", "b", "c")
  x
```

```
 [1] "a" "b" "c"
 y<- c(1,2)
 y
[1] 1 2
 z <- c(x,y)
 z
[1] "a" "b" "c" "1" "2"
```

La función **str()** muestra el tipo de datos que contiene el objeto que le es pasado como argumento.

```
 z <- c(x,y)
 z
[1] "a" "b" "c" "1" "2"
 str(z)
 chr [1:5] "a" "b" "c" "1" "2"
```

Matrices y arrays (arreglos)

Las matrices son vectores que se encuentran indexados por uno o más índices, es un arreglo de elementos ordenados en filas y columnas. Los arreglos pueden tener una, dos o más dimensiones. Los elementos de una matriz y de un arreglo son del mismo tipo; sus elementos pueden ser de tipo numérico, caracter, complejo y lógico.
Existen diferentes formas de crear una matriz, una forma es mediante el atributo **dim**, por medio de la función **matrix** y por último a través de la función ***array***. Para convertir un vector en una matriz se debe agregar el atributo dim a dicho vector.
Ejemplo 4. Se tiene un vector llamado ***b*** de 8 elementos, para convertir el vector ***b*** en un vector de dimensional (matriz) asignamos el vector al atributo ***dim***.

```
 b<-1:8
 dim(b)<-c(2,4)
 b
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```

El resultado del ejemplo es una matriz de dimensión (2X4) (2 filas y 4 columnas).

Función matrix.

Esta función permite ingresar una matriz de valores dados o de valores específicos que ya existen.

Sintaxis:

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,
       dimnames = NULL)
```

Donde:

data= Vector de datos.
nrow= Número de filas.
ncol= Número de columnas.
byrow= Indica si los valores en el campo data deben llenar por columnas (FALSE) o bien por filas (TRUE).
dimnames= Permite asignar nombres a las filas y columnas.
Ejemplo 5. Crear una matriz de 2 filas y 2 columnas con el vector de datos c(2,4,5,7).

```
 matrix(data = c(2,4,5,7), nrow= 2, ncol = 2, byrow = TRUE,
    dimnames = NULL)
    [,1] [,2]
[1,]    2    4
[2,]    5    7
```

Ejemplo 6. También se puede crear una matriz con 8 elementos, se puede escribir como sigue.

```
 matrix(1:8, 2, 4)
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```

Función array()

La función ***array()*** es otra forma de obtener una matriz, sin embargo, su construcción es más estricta debido a que el vector de datos debe coincidir con el vector de dimensiones, en el caso de que el vector de datos sea menor respecto al vector de dimensiones los valores para la matriz se repiten tantas veces sea necesario hasta obtener los mismos elementos en ambos vectores a este proceso se le conoce como reciclado.

Sintaxis:

```
array(data = NA, dim = length(data), dimnames = NULL)
```

Donde:

data= Vector de datos.
dim= Vector de dimensiones.
dimnames= Asignación de nombres a las filas y columnas.
Ejemplo 7. Crear una matriz de dimensión 2x4.

```
 array(1:8, dim=c( 2, 4))
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```

Ejemplo 7.1. Reciclado, el vector de datos es de 6 elementos y el vector de dimensiones indica 8 elementos, como el vector de datos es menor que el de dimensiones entonces se repiten los dos primeros valores (1 y 2) para completar los 8 elementos.

```
  array(1:6, dim=c( 2, 4))
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    1
[2,]    2    4    6    2
```

Funciones rbind() y cbind()

Las funciones ***rbind()*** y ***cbind()*** son funciones que sirven de apoyo para la construcción de nuevas matrices a partir de la concatenación de otras matrices o arreglos y/o vectores.

Sintaxis:

```
X <- cbind(arg 1, arg 2, arg 3, ...)

X <- rbind(arg 1, arg 2, arg 3, ...)
```

En estas funciones los argumentos pueden ser vectores de cualquier longitud ó matrices con el mismo número de filas y se tiene que considerar que cuando los argumentos de las funciones en cuestión se encuentran formados por un vector y una matriz, el vector no debe ser de mayor longitud que el número de filas de la matriz. Si es menor, se repiten los valores hasta tener la misma longitud de la matriz. Si los argumentos únicamente son vectores y tienen diferentes longitudes se recicla el más corto hasta igualarse con el tamaño del más grande.
Ejemplo 8. El siguiente ejemplo muestra la matriz **r** formada por el vector **v** y la matriz **mat**, sin embargo, el resultado que se obtiene es erróneo pues sólo toma los dos primeros elementos del vector y además envía un mensaje de error, en donde indica que el número de columnas que se generan con respecto a la matriz no es igual al número de columnas que generarían los elementos del vector que serían 6 columnas.

```
 v <- c(5:10)
 v
[1]  5  6  7  8  9 10
 mat <- matrix (1:4, 2,2)
 mat
     [,1] [,2]
[1,]    1    3
[2,]    2    4
r <- rbind(v, mat)
Warning message:
In rbind(v, mat) :
  number of columns of result is not a multiple of vector length
      (arg 1)
 r
  [,1] [,2]
```

```
v    5    6
     1    3
     2    4
```

Ejemplo 8.1. Uso de ***rbind*** para dos vectores de diferentes longitudes.

```
 l <- c(1:10)
 m <- c(2)

 k <- rbind(m, l)
 k
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    2    2    2    2    2    2    2    2    2     2
[2,]    1    2    3    4    5    6    7    8    9    10
```

El resultado es una matriz de 2x10, como el vector **m** es menor que el vector **l** entonces el vector **m** es reciclado hasta igualar el número de elementos que tiene el vector **l**.

La función ***cbind*** se utiliza cuando se desea sustituir columnas por filas y ***rbind*** realiza la tarea inversa.

Ejemplo 8.2. Sustituir las columnas **A**, **B** y **C** por filas.

```
 cbind(A=1:3, B=4:6, C=7:9)
     A B C
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
 rbind(A=1:3, B=4:6, C=7:9)
  [,1] [,2] [,3]
A    1    2    3
B    4    5    6
C    7    8    9
```

Dentro de las operaciones básicas que se pueden realizar con matrices se encuentran las siguientes.

Operaciones básicas con matrices	
Operador	Descripción
A + B	Suma de matrices
A - B	Resta de matrices
A %* %B	Producto de matrices
A * B	Producto de matrices cuadradas, elemento por elemento
crossprod(A,B)	Producto cruzado de matrices

Factor

Es una estructura de vectores que facilita el manejo de variables categóricas, con la cual no se realizan operaciones, es decir, es una variable cualitativa que permite trabajar con variables descriptivas, el objetivo de los factores es especificar, ordenar y enumerar una clasificación de un vector en particular.

Sintaxis:

```
factor(x, levels = sort(unique(x), na.last = TRUE), labels =
    levels, exclude = NA, ordered = is.ordered(x))
```

Donde:

x = Vector de datos.
levels= Posibles niveles que puede tomar el factor, por defecto toma los valores de x como niveles y los ordena en orden alfabético.
labels= Asigna el nombre de los niveles.
ordered= Indica si los niveles del factor se encuentran ordenados.
exclude= Excluye valores de x.
Ejemplo 10. Elaborar un factor.

```
colores <- c("rosa", "rojo", "blanco", "azul", "rosa", "blanco")
 factorcolores <- factor(colores)
 factorcolores
[1] rosa   rojo   blanco azul   rosa   blanco
Levels: azul blanco rojo rosa
```

Como se observa los niveles que toma el factor son los valores del vector colores y los ordena alfabéticamente.

Ejemplo 10.1. Función factor utilizando la opción labels.

```
factorfrutas<-factor(1:3, labels=c("melón", "mango", "guaraná"))
 factorfrutas
[1] melón   mango   guaraná
Levels: melón mango guaraná
```

Los niveles que toma el factor **factorfrutas** son 3 porque el vector **1:3** contiene tres elementos, pero como a esos tres elementos ya se les asignó un nombre ahora cada uno de los niveles toma el nombre de cada uno de los elementos del vector **1:3**.

Ahora bien, si se utilizan los dos parámetros levels y labels es necesario que ambos sean de igual o de mayor longitud que el vector de datos.

Ejemplo 10.2. Crear un factor con el mismo número de elementos tanto para el vector de datos, ***levels*** y ***labels***.

```
 factorfrutas<-factor(1:3, 1:3, c("melón", "mango", "guaraná"))
 factorfrutas
[1] melón   mango   guaraná
 factorfrutas<-factor(1:3, 1:3, c("melón", "mango", "guaraná"))
 factorfrutas
[1] melon   mango   guaraná
Levels: melón mango guaraná
Levels: melón mango guaraná
```

Ejemplo 10.3. El siguiente ejemplo ilustra de forma general el comportamiento de la función factor.

Se supone que se realiza una encuesta a vendedores de frutas acerca de las frutas de temporada en cierta estación del año; las repuestas son las siguientes.

```
 frutastemporada<-factor(c("melón", "mango", "melón","guayaba",
                           "guayaba", "melón","manzana","mango"))
 frutastemporada
[1] melón   mango   melón   guayaba guayaba melón   manzana
   mango
Levels: guayaba mango manzana melón
```

Como se puede observar los niveles del factor toma el nombre de las frutas sin repetir ninguna, de esta forma, existen cuatro niveles aunque el factor esta construido de ocho respuestas.

Lista

Una lista es una generalización del vector, sin embargo, sus elementos pueden ser una colección ordenada de diferentes estructuras (vectores, matrices, listas, funciones, etc.). Los componentes de una lista siempre se encuentran numerados.

Sintaxis:

```
list(vector, listas, funciones, arreglos o matrices)
```

Ejemplo 11. Construir una lista con tres componentes cualesquiera.

```
 primerLista <- list(c("Javier","Araceli"), c(26,24), mean)
[[1]]
[1] "Javier"  "Araceli"
[[2]]
[1] 26 24
[[3]]
function (x, ...)
UseMethod("mean")
<environment: namespace:base>
```

Asignación de nombres a listas

Es posible asignar nombres a los componentes de una lista con los cuales también pueden ser accedidos.

Sintaxis:

```
Nombredelalista\$nombredelcomponente
```

Ejemplo 12. Asignar nombres a la lista primerLista.

```
 primerLista<- list(c("Javier","Araceli"), c(26,24), mean)
 names(primerLista)<-c("alumnos","edad","función")
 primerLista
\$alumnos
```

```
[1] "Javier"  "Araceli"
$edad
[1] 26 24
$función
function (x, ...)
UseMethod("mean")
<environment: namespace:base>
```

La lista **primerLista** esta formada de tres argumentos, un vector carácter, un vector numérico y una función; al vector carácter se le asigna el nombre alumnos, al vector numérico el nombre edad y a la función ***mean*** se le asigna el nombre función.
Ahora acceder al primer componente de la lista **primerLista**.

```
 primerLista$edad
[1] 26 24
```

Cuando se requiere almacenar el nombre de los componentes en otra variable se sugiere utilizar el nombre de los mismos entre corchetes dobles.
Ejemplo 13. Almacenar el nombre del primer componente (alumnos) en la variable ***x***.

```
   x<-"alumnos";primerLista[[x]]
  [1] "Javier"  "Araceli"
```

En este ejemplo la variable ***x*** contiene el nombre del componente alumnos, es decir, si se ejecuta ***x*** el resultado es alumnos.

```
 x
[1] "alumnos"
```

Marco de datos (Data Frames)

Un marco de datos es un contenedor de datos, es decir, como una tabla o matriz de datos cuyos componentes son uno o más vectores de la misma longitud. Las variables en un marco de datos son columnas que se describen por un nombre para identificarlas y es la única fila de nombres.

Es recomendable la aplicación de la estructura data frame() cuando se desea describir matrices de datos, donde la relación son individuos(filas)x variables(columnas).

Una lista puede transformarse en un data frame, sin embargo, se debe considerar las siguientes restricciones para los componentes:

- Los componentes deben ser vectores, matrices numéricas u otros dataframes.
- Los vectores deben ser de la misma longitud y pueden ser numérico caracter ó lógico.
- Las matrices deben tener el mismo número de filas.

Los estructura data frame considera como factores los datos que no son numéricos, con tantos niveles como valores distintos encuentre.

Sintaxis:

```
data.frame(nombre de la variable = elementos de la variable,
    row.names = NULL, check.rows = FALSE, check.names = TRUE,
    stringsAsFactors = default.stringsAsFactors())
```

Donde:

Nombre de la variable = Elementos de la variable = Son argumentos predefinidos de valor o etiqueta=etiqueta. Los nombres de los componentes son creados de acuerdo a las etiquetas.
row.names= Asignación de nombres a las filas.
check.rows= Si es verdadero (TRUE), comprueba las filas para saber si hay consistencia de longitud y nombres.
check.names= Si es TRUE comprueba que los nombres de las variables sean sintácticamente correctos y que no estén duplicados.
StringsAsFactors = Covierte un vector caracter a factor cuando el argumento de default.stringsAsFactors() es FALSE.
Ejemplo 14. Contruir un dataframe utilizando el atributo row.names.

```
data.frame(row.names=c("Araceli","Ara", "Arlette", "Javier",
          "Fernando"), cbind(Edad=20:24, Hijos=1:5))
         Edad Hijos
Araceli    20     1
Ara        21     2
Arlette    22     3
Javier     23     4
Fernando   24     5
```

Ejemplo 14.1. Extraer sólo la primer columna del data frame anterior.

```
 e<- data.frame(row.names=c("Araceli","Ara", "Arlette",
                "Javier", "Fernando"), cbind(Edad=20:24, Hijos
                   =1:5))
e[-2]
          Edad
Araceli     20
Ara         21
Arlette     22
Javier      23
Fernando    24
```

Función read.table()

La función ***read.table()*** es la forma más práctica de crear un data frame a partir de un archivo existente en una hoja de calculo ya sea de excel u openOffiice, sin embargo, la hoja de cálculo debe ser guardada con extensión .txt o bien con .csv y en el respectivo directorio donde se este ejecutando R.

Sintaxis:

```
 read.table(file, header = FALSE, sep = "")
```

Donde:

file = El nombre (entre comillas) del archivo del cual se están leyendo los datos.
header= Valor lógico que indica si debe aparecer el nombre de cada una de las columnas.
sep= Separador de columnas que por defecto es un espacio en blanco pero también puede ser tabulador o coma ",".

Ejemplo 15. Extraer el archivo prueba.csv.

```
 read.table("prueba.csv", header=TRUE, sep=",")
    Alumnos Edad Hijos
1   Arceli    24     1
2   Javier    26     1
3 Fernando    30     3
4  Arlette    24     2
5  Eduardo    25     1
```

El archivo prueba.csv se encuentra guardado en el mismo directorio donde esta instalado R, por lo que, no se indica ninguna ruta de referencia al directorio donde se encuentra.

Funciones y argumentos de R

Es una clase de objeto que se crea con la finalidad de ser utilizada en un proceso posterior o bien una función existente de R.
En R existen funciones estadísticas ya definidas, sin embargo, tiene la flexibilidad de que el usuario elabore sus propias funciones. Para que una función sea reconocida y ejecutada en R es necesario que siempre este acompañada de paréntesis, aún cuando la función no tenga argumentos; si en una función se omiten los paréntesis entonces el resultado obtenido al ejecutar la función es el propio código de la función. En R una función creada también es un objeto, la sintaxis usada en R para crear funciones se realiza mediante la asignación de objetos y se utiliza el modo ***function***.

Sintaxis:

```
NombreFución <-function(arg 1, arg 2, ...)
                        {cuerpo de la función}
```

Donde:

NombreFunción= es un nuevo objeto y es también el nombre de la función creada.
<- = Es el operador de asignación apunta hacia el objeto que recibe el valor de la expresión en este caso el objeto al que apunta es: NombreFunción.
()= El contenido de los paréntesis son los argumentos de la función.
Cuerpo de la función= En esté lugar se definen los comandos y operaciones. Los comandos individuales van separados por un punto y coma (;).

En general el nombre para una función puede ser cualquier cosa que al usuario se le ocurra siempre y cuando no utilice el nombre de las funciones que ya existen.

Los argumentos en R son objetos de diferentes tipos (“datos”, fórmulas, expresiones...) y algunos definidos por defecto en alguna función. Cuando el argumento no se encuentra predefinido por una función, se deben proporcionar los argumentos a la misma.

Funciones para el control de una función

- return(). Termina el flujo de la evaluación de una función en cualquier proceso; usualmente va acompañada de la estructura condicional if.
- warning(). Imprime un mensaje de advertencia en un caso en particular; por ejemplo, si en una evaluación de una función los valores para los que se quiere evaluar la función no son permitidos para la misma.
- Stop(). Para le ejecución de una función e imprime un mensaje de error.

Asignación de valores a objetos

La asignación de valores a evaluar en una variable u objeto se lleva acabo por medio del operador **<** y un guión medio **-**, es importante que los dos caracteres estén juntos uno seguido del otro, de lo contrario, el significado del operador de asignación (<-) será diferente.

Por ejemplo se tiene el objeto x y se le asigna un valor de 2, sin embargo, el espacio que existe entre los caracteres **<** y **–** , impide que se lleve acabo la operación, por ejemplo.

```
 x< -2
[1] FALSE
```

Las variables construidas y utilizadas en R pueden ser reutilizadas para realizar operaciones aritméticas, esto se logra a través de la llamada de dichas variables, por ejemplo, se realiza la siguiente operación aritmética (en este caso suma), con el uso de la variable ***x*** que tiene asignado el valor 1.

```
 x<-1
 x
[1] 1
 x+x
[1] 2
```

Conversión de objetos

Debido al uso frecuente de objetos de diferente tipo, en ocasiones habrá la necesidad de convertir el tipo de objeto para poder trabajar de manera adecuada los datos. La conversión se realiza de la siguiente forma.

Sintaxis:

```
as.tipo_de_objeto_a_obtener(objeto que se va transformar)
```

Ejemplos 16. ***b*** es un objeto de tipo matriz de 2x4, se desea convertirla a un objeto de tipo dataframe y posteriormente se convertirá a un objeto de tipo numérico.

```
   b
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
 as.factor(b)
[1] 1 2 3 4 5 6 7 8
Levels: 1 2 3 4 5 6 7 8
 as.numeric(b)
[1] 1 2 3 4 5 6 7 8
```

Una descripción más general acerca de la conversión de objetos se encuentra en la siguiente tabla.

conversión a	Función	Reglas
númerico	as.numeric	FALSE → 0
		TRUE → 1
		"1", "2",...→ 1,2...
		."A",... → NA
lógico	as.logical	0 → FALSE
		"FALSE", "F"→ FALSE
		"TRUE", "T"→ TRUE
		otos caracteres → NA
caracter	as.character	1,2... → "12"
		"FALSE", "F"→ FALSE
		"TRUE", "T"→ TRUE

Tabla 2.1: Conversión de objetos

2.3. Indexación

Para acceder a alguno de los componentes de objetos de tipo listas, matrices y dataframe se hace referencia a la numeración de dichos componentes y para ello se utilizan dobles corchetes. El corchete simple se utiliza en caso de que se requiera conocer alguno de los elementos de un componente.

La indexación para matrices y dataframes se lleva acabo por medio de los subíndices [i, j] (í – ésima fila y j- ésima columna). Con el sistema de indexación es posible realizar las siguientes tareas.

Cambiar un elemento del objeto. Se realiza asignando el nuevo valor para el elemento a cambiar.

Sintaxis:

```
objeto [[índice_del_elemento_acambiar]] <-nuevo_valor
```

Eliminar una o más filas y columnas.

Sintaxis:

```
 objeto[-fila/columna_a_eliminar]
```

Ejemplo 17. Extraer el primer componente de la lista **primerLista**.

```
 primerLista[[1]]
[1] "Javier"  "Araceli"
```

Ejemplo 18. Extraer el primer elemento del segundo componente de la lista **primerLista**.

```
 primerLista[[2]][1]
[1] 26
```

2.3.1. Uso de una matriz como índices

Hasta ahora se ha mencionado que una matriz son vectores indexados por lo que también se le conoce como variable indexada. Una variable indexada se compone de vectores indices los cuales son útiles para extraer una colección de

elementos particulares y también para asignar un vector a una colección de elementos. Los índices de una matriz deben ir entre corchetes y separados por comas, fuera del corchete va el nombre de la variable indexada.
Ejemplo 18.1. Extraer la segunda fila del ejemplo anterior.

```
 b<-1:8
 dim(b) <- c(2,4)
 b
      [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8

b[2,]
[1] 2 4 6 8
```

Para resolver este problema se indica el índice que se refiere a la segunda fila, en este caso es 2 y se expresa de la siguiente forma $b[2,]$, la coma después del 2 indica que sólo se extraen los elementos de la segunda fila y ningún elemento correspondiente a las columnas.
Ejemplo 19. Extraer el elemento de la segunda fila y la segunda columna b[2,2].

```
 b[2,2]
[1] 4
```

2.4. Gráficos en R

Un aspecto fundamental a considerar es la presentación de información a través de gráficos que a diferencia de otras formas de presentación de datos, ésta permite entender e interpretar el comportamiento de los datos de forma clara y sencilla; aún cuando los datos son complejos. En éste trabajo la utilidad de gráficos es primordial para el análisis del comportamiento de los datos. R maneja una variedad de funciones gráficas, mencionar todas en éste trabajo sería imposible pero se pueden consultar ejecutando el comando a ***demo(graphics)***.

2.4.1. Presentación de un Gráfico

La presentación estándar de un gráfico en R se encuentra ubicado dentro de márgenes, las coordenadas del gráfico describen en unidades de datos los ejes ***x*** y

y, esto es, que cada coordenada tanto del eje ***x*** como del eje ***y*** se caracterizan por tener un dato. Una ventaja de R con respecto al nombre que el usuario le asigna a los títulos de los ejes, es que las coordenadas en los márgenes se especifican en líneas de texto escritas en forma perpendicular en el eje vertical ***y*** y en forma horizontal para el eje ***x***. El objetivo de esta disposición del gráfico es que el usuario pueda observar la descripción (títulos) de las etiquetas de ambos ejes. Existen varias alternativas para modificar un gráfico, por ejemplo, cambiar el título principal del diagrama que contiene el gráfico, cambiar el subtítulo que se ubica en la parte inferior del mismo o bien eliminar las etiquetas de los ejes.

El siguiente ejemplo muestra como llevar acabo las modificaciones de título, subtítulo y etiquetas de los ejes [1].

```
png("ejemplo1")
 y <- runif(100,0,2)
 x <- runif(100,0,2)
 plot (x, y, main="titulo principal", sub="subtitulo",
        xlab="y-etiqueta", ylab="x-etiqueta")
        dev.off()
```

También se pueden agregar los puntos y líneas como un texto dentro del gráfico con:

```
 text(0.5,0.5,"text  at  (0.5,0.5)")
 abline(h=.5,v=.5)
```

La función abline posiciona el texto en los puntos especificados como argumento, la función tex permite escribir texto seguido de las coordenadas donde se colocará.
Además los márgenes de un gráfico también se pueden observar mediante la función ***mtext***; los valores de esta función sólo se pueden modificar con la función par() esto es:

```
png("ejemplo2")
plot (x, y, main="titulo principal", sub="subtitulo",
        xlab="y-etiqueta", ylab="x-etiqueta")
          text(0.6,0.6,"text at (0.6,0.6)")
          abline(h=.6,v=.6)
        for (side in 1:4) mtext(-1:4,side=side,at=.8,line=-1:5)
```

[1] La función ***runif*** sirve para generar números aleatorios de una distribución uniforme donde el intervalo es un límite mínimo y un límite máximo

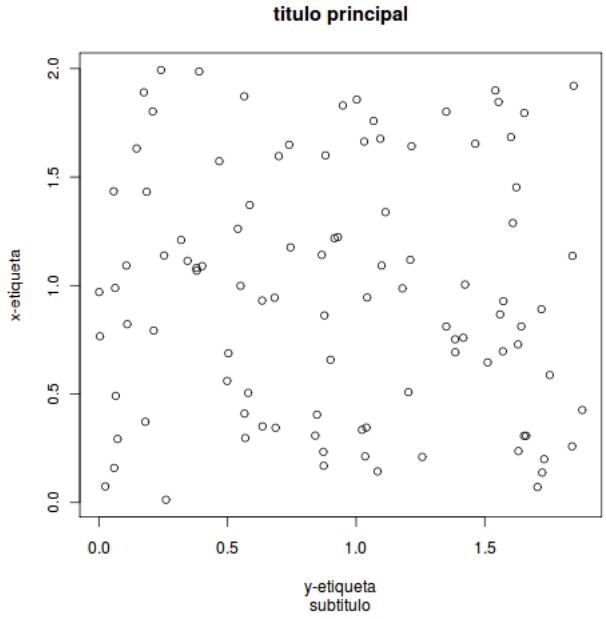

Figura 2.1: Títulos y etiquetas

```
 mtext(paste("side",1:4), side=1:4, line=-1,font=2)
dev.off()
```

En estructura de la función anterior se realiza un para colocar los números de los márgenes del -1 a 4 cada uno de los cuatro lados, en una escala centrada a .8 del usuario, después le agrega una etiqueta con un número correspondiente a cada lado y se les asigna el color negro **(font=2)**.

2.4.2. Procedimientos gráficos con la función par

La función ***par()***, se utiliza para controlar los parámetros de un diagrama o dispositivo gráfico en uso, con ella podemos fijar o consultar los parámetros de un gráfico mediante llamadas a la función ***par*** y de esta forma personalizar el aspecto de un diagrama; los argumentos de la función no son cualquier variable, éstos son parámetros gráficos que ya existen y sólo son llamados por la función par.

Sintaxis:

```
par(nombre=valor)
```

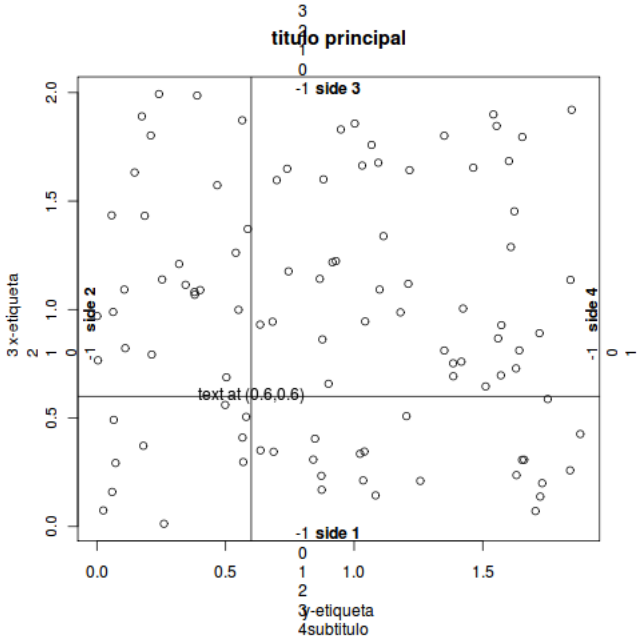

Figura 2.2: Márgenes medidos en líneas de texto

Donde:
nombre= Es el nombre del parámetro.
Valor= Es el valor que se le asigna al parámetro.
Ejemplo. Modificar los siguientes parámetros:

- oma. Márgenes exteriores del gráfico.
- bg. Color del fondo del gráfico.

```
png("parametros.png")
 y <- -x^2
 x <- -3:3
par(oma = c(2, 0, 2, 0), bg= "gray")
plot(x,y,  col.lab= "red", type = "l", lwd= 2, col= "blue", ylab
   = "eje Y", xlab= "eje X" )
title("Márgenes y fondo del gráfico", col.main = "red")
dev.off()
```

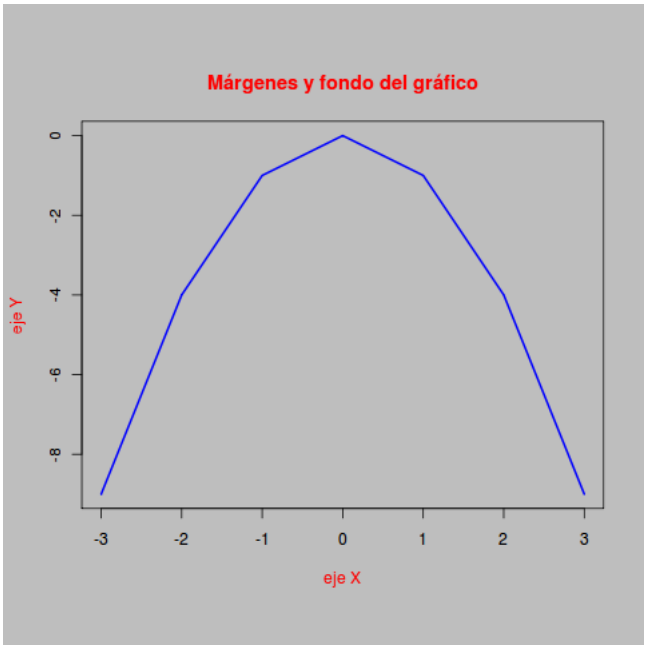

Figura 2.3: Parámetros gráficos

2.4.3. Tipos de parámetros gráficos

Los parámetros en R son diversos y permiten controlar la apariencia de los gráficos; se utilizan dependiendo de las necesidades del gráfico y del usuario, por ejemplo, para algunos gráficos habrá que modificar el color, el tipo de línea o el tamaño de la fuente, de tal forma que que sea presentable.

Parámetros que modifican márgenes de un gráfico o figura

- mai: Vector numérico c(inferior, superior, izquierdo, derecho) que determina el ancho de los márgenes en pulgadas.
- mar: Vector numérico c(inferior, superior, izquierdo, derecho), el cual también proporciona el ancho de los márgenes de la figura pero medido en líneas.
- oma: Vector numérico de la forma c(inferior, superior, izquierdo, derecho), especifica el tamaño en líneas de los márgenes exteriores.

- omi: Tiene el mismo efecto que oma, sólo que da el tamaño de los margenes exteriores en pulgadas.
- omd: Es un vector de la forma **c(x1, x2, y1, y2)** para especificar la región de margen externo como fracción (en [0, 1]) de la región de dispositivo.

Parámetros para elementos de un gráfico

- col: Color para ejes, texto, imágenes, líneas, puntos y rellenos; para obtener los colores deseados se puede indicar el nombre del color en inglés y entre comillas, o bien en términos de componentes RGB. [2]
- font: Entero que indica la fuente del texto,si es posible para el dispositivo gráfico en uso se podrán utilizar los siguientes parámetros.
  - 1 corresponde a texto plano (por defecto),
  - 2 corresponde a texto en negrita,
  - 3 corresponde a texto en itálica,
  - 4 corresponde a texto itálica- negrita.
- cex: Vector numérico para especificar la cantidad por la cual el texto y los símbolos gráficos deberán ser escalados respecto al valor establecido por defecto.

Parámetros para ejes

- lab: Vector numérico de la forma c(x,y, len), donde los valores de $\boldsymbol{x}$ y $\boldsymbol{y}$ indican las marcas para el eje $\boldsymbol{x}$ y $\boldsymbol{y}$, el tercer valor es la longitud de las etiquetas de los ejes $\boldsymbol{x}$ y $\boldsymbol{y}$ medida en caracteres.
- las: Número que indica la orientación de las etiquetas de los ejes.
  - 0: paralelo al eje,
  - 1: horizontal,
  - 2: perpendicular,
  - 3: vertical.

[2] Modelo RRGGBB, cada par RR, GG, BB toma valores desde 00 hasta FF

Para fijar un parámetro éste se tiene que especificar como argumento de la función ***par*** , o bien, pasarlo como una lista de valores etiquetados a dicha función.
La función ***par()*** sin argumentos devuelve una lista de todos los parámetros del gráfico y sus valores.
Ejemplo 1, si se desea cambiar los margenes de un gráfico, se escribe lo siguiente, par(mai=c(5,5,5,5)). En el caso anterior ***mai*** es el nombre de los parámetros, enseguida se encuentran los valores respectivos para sus ejes.

Exportar gráficos.

Los gráficos realizados en R se pueden exportar en diferentes formatos.
Sintaxis:

```
opciones("")
plot()
dev.off()
```

Donde:
opciones= Función para obtener un gráfico de formato png.
()= Nombre que se le va asignar al gráfico.
plot()= Ejecución del gráfico.
dev.off()= Finaliza la ejecución del gráfico.

Formato gráfico	Función
Adobe PostScript Adobe PDF	postscript() pdf()
LATEX PicTEX	pictex()
XFIG	xfig()
Conversión de archivos GhostScript	bitmap()
PNG	png()
JPEG	jpeg()

2.5. Funciones de estadística descriptiva aquí utilizadas

El uso de las funciones gráficas se encuentra relacionado con el tipo de objeto en uso, por tanto, si se tiene un objeto de variables categóricas y se desea dibujar

un histograma con la función **hist** R enviará un mensaje de error, entonces significa que la función adecuada para obtener el histograma es barplot(), lo mismo pasará con otro tipo de variable en la tabla 2.2. Respecto a los argumentos para cada una de las siguientes funciones existe varios, los de mayor importancia y que se utilizan para todos son los que se mencionan en la lista.

- col: Indica el color a ser usado en las gráficas, en este caso se puede pasar como un vector indicando los colores deseados.
- main, sub: Título y subtítulo de gráfico.
- xlab: Etiquetas o nombres para el eje X.
- ylab: Etiquetas o nombres para el eje Y.
- xlim: Delimita el rango de valores en el eje X.
- ylim: Delimita el rango de valores en el eje Y.
- axes: Argumento lógico. Si es TRUE, dibuja el correspondiente eje.
- plot: Argumento lógico. Si es FALSE no se dibuja el gráfico.
- legend.text: Vector de texto usado para construir la legenda del dibujo, o valor lógico que indica si esta se debe incluir.

Función	Tipo de variable
barplot()	genérica
pie()	categórica
hist()	continua
boxplot	continua por categorías
plot()	categórica

Tabla 2.2: Gráficos descriptivos

2.5.1. barplot()

Función que permite obtener un gráfico de barras cuando se se trabaja con datos categóricos, algunos argumentos aceptados por esta función se listan a continuación.

Sintaxis:

```
1 barplot(height, width = 1, space = NULL,
2             names.arg = NULL, legend.text = NULL, beside =
                  FALSE,
3             horiz = FALSE, angle = 45, col = NULL, border = par
                  ("fg")...)
```

- height: Puede ser un vector o matriz de valores que se emplearán para la representación de las barras del diagrama. Si el argumento es un vector, el diagrama se compone de una secuencia de barras rectangulares con una altura determinada por los valores del propio vector. En cambio, si la altura es una matriz y el argumento besides es FALSE entonces, al lado de cada barra del gráfico corresponde a una columna de altura. Si este argumento es una matriz y el argumento besides es TRUE, entonces los valores de cada columna se yuxtaponen en lugar de apilarlos.
- width: Argumento opcional que determina el ancho de las barras.
- space: Para fijar el espacio entre barras, como una fracción del ancho promedio de éstas. Puede especificarse con un sólo número o un número por barra.
- names.arg: Un vector con etiquetas que se colocan debajo de cada barra o grupo de barras. Si se omite, entonces los nombres son tomados de los atributos de nombres que estén contenidos en el objeto especificado en el argumento height.
- horiz = FALSE: Indica si las barras deben dibujarse horizontal o verticalmente, por defecto se dibujan verticalmente.
- legend.txt: Un vector de texto para construir una leyenda para el gráfico. Sólo es útil si el argumento height es una matriz, en cuyo caso las leyendas corresponderán a sus filas.

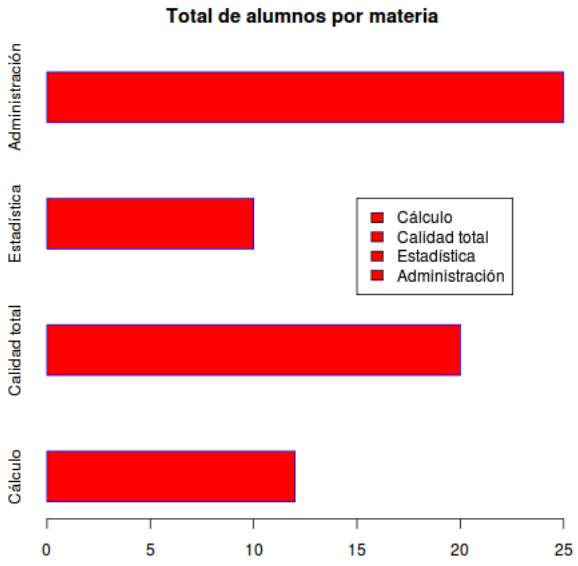

Figura 2.4: Función barplot

- beside: Un valor lógico. Si es FALSE las columnas del argumento height serán representadas por barras apiladas y si es TRUE entonces, serán representadas por barras yuxtapuestas.
- border: Color del borde de cada una de las barras.
- col: Color de las barras.

Ejemplo 2.

```
1  png("e")
2  alumnos<-c(12,20,10,25)
3  materias<-c("Cálculo","Calidad total","Estadística","
       Administración")
4   barplot(height= alumnos, width=2, space=1.5, beside = TRUE,
5   main = "Total de alumnos por materia", names.arg=materias,
6   col= "red", border= "blue", horiz = TRUE)
7   legend(15,15, materias, text.col= "black", fill= "red")
8   dev.off()
```

2.5.2. pie()

Esta función es una gran herramienta para generar gráficas circulares.
Sintaxis:

```
pie(x, labels = names(x), edges = 200, radius = 0.8,
        clockwise = FALSE, init.angle = if(clockwise) 90 else
            0,
        density = NULL, angle = 45,  border = NULL,
        lty = NULL, main = NULL, ...)
```

- x: Cantidades numéricas positivas que representan el área de cada una de las rebanadas del pastel.
- labels: Vector de caracteres para asignarle nombres a cada una de las rebanadas.
- edges: Aproxima la línea exterior circular mediante un polígono con el número de lados especificado.
- radius: La torta o pastel es dibujada centrada en una caja cuadrada cuyos lados se mueven en el rango de -1 a 1. Si se usan etiquetas largas puede ser necesario usar radios más pequeños.
- clockwise: Argumento lógico, TRUE o FALSE, que indica si la representación es en sentido positivo del reloj o no, el sentido positivo es la opción por defecto (FALSE).
- init.angle: Número que indica el ángulo inicial en el cual se representará el diagrama (en grados). El valor por defecto es 0 (es decir, a las 3 en punto) a menos que el argumento clockwise esté configurado como TRUE entonces, init.angle será por defecto 90 (grados), (es decir, 12 en punto).
- density: Dibuja porciones ralladas por pulgada. El valor predeterminado es NULL, significa que no se dibujan líneas de rallado.
- angle: Pendiente de las líneas de rallado, dado como un ángulo en grados (a la izquierda).
- border y lty: Indican el color para el borde de cada una de las rebanadas.

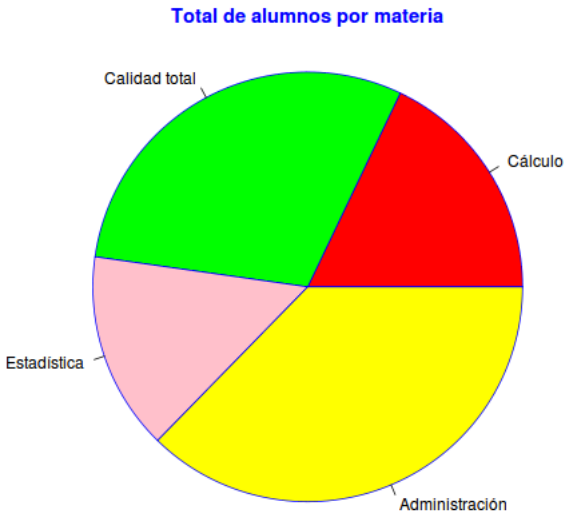

Figura 2.5: Función pie

```
1  png("pie")
2  alumnos<-c(12,20,10,25)
3  materias<-c("Cálculo","Calidad total","Estadística","
       Administración")
4  pie(x=alumnos, labels=materias, radius= 1,
5  main = "Total de alumnos por materia",, font.main=2, col.main=
       "blue",
6  col= c("red", "green", "pink", "yellow"), border= "blue")
7  dev.off()
```

2.5.3. plot()

La función ***plot*** produce gráficos simples, sus argumentos pueden ser vectores, funciones o dataframes; esta función también se puede personalizar llamando a diversos parámetros.

Sintaxis:

plot(x, y, ...)

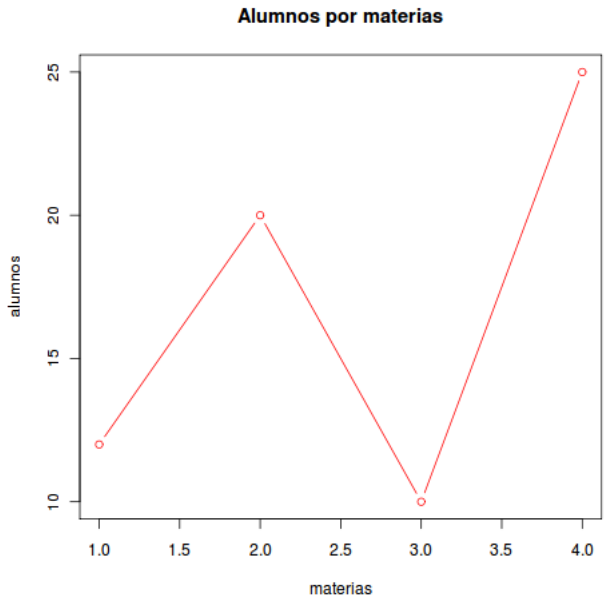

Figura 2.6: Función plot

- x: Coordenadas para el eje x
- y: Coordenadas para el eje y, este es un argumento opcional.
- type: Tipo de gráfico, los valores que puede tomar **type** son p(puntos), l(lineas), b(puntos conectados por líneas), o(líneas ó puntos superpuestos), h(líneas verticales en lugar de las barras), s(gráficos en escalones), S(Otros gráficos en escalas), n(no dibuja nada).

Ejemplo 3.

```
1    png("p")
2       alumnos<-c(12,20,10,25)
3       plot(alumnos,main = "Alumnos por materias", xlab= "
           materias",
4       ylab= "alumnos", type="b", col= "red")
5    dev.off()
```

```
1   png("cos")
```

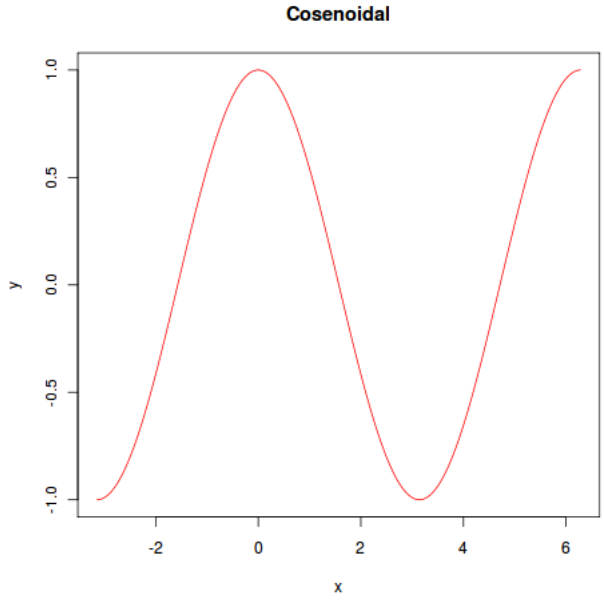

Figura 2.7: Función plot

```
2         x<-seq(-pi, 2*pi, by=0.01)
3         y<-cos(x)
4         plot(x,y, main="Cosenoidal",type="l", col="red")
5   dev.off()
```

2.5.4. hist()

. La función ***hist()*** trabaja con vectores numéricos para obtener histogramas, en R el número de clases se calcula mediante fórmula de Sturges, pero también se puede indicar el numero de clases(barras) a dibujar.

- x: vextor de datos para dibujar el histograma.
- breaks: Un número o vector de valores que indiquen el número de clases o intervalos.
- freq: El tipo de valor es lógico, en este caso si es "TRUE", el histograma representa las frecuencias absolutas de cada clase, de lo contrario, representará las frecuencias relativas de cada clase.

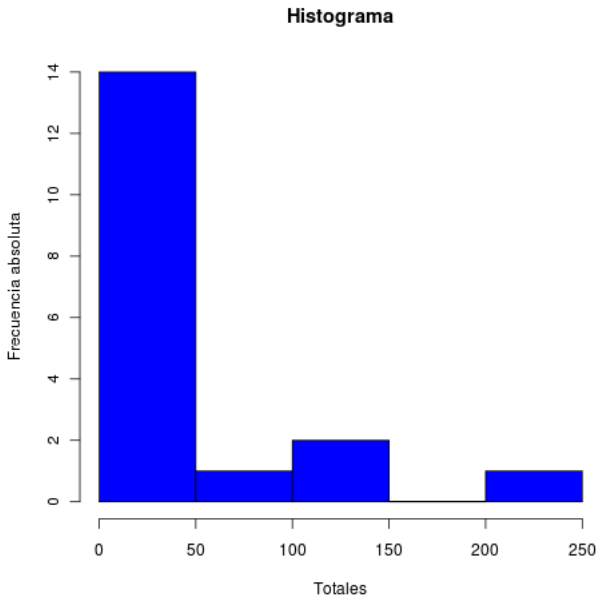

Figura 2.8: Función hist

- probability = !freq: Este argumento representa un alias para ***freq***, de este modo consigue ser compatible con S.
- nclass: Un número entero que indica el número de clases, es el equivalente a ***breaks***, pero se utiliza para que exista compatibilidad con S. **Ejemplo 4**

```
1  png("h")
2  hist(becados\$total, breaks = 5, col= "blue",
3  main = "Histograma", ylab= "Frecuencia absoluta",
4  xlab= "Totales")
5  dev.off()
```

Capítulo 3

Descripción del sistema de información académico

Actualmente la implementación de sistemas de información computacional (SI), son indispensables en empresas y organizaciones, debido a que el objetivo es contar con un instrumento que brinde disponibilidad y confiabilidad de la información, y que además sirva de apoyo en la toma de decisiones para el mejoramiento de dichas empresas u organizaciones. Un sistema de información es un conjunto de procesos y herramientas que a través de su interacción logran transformar, procesar y organizar la información de tal forma, que el flujo de la misma sea favorable en el momento que se requiera.

Los procesos esenciales que lleva acabo un sistema de información son los siguientes:

Entrada de información. Es el proceso de ingresar la información más relevante que podrá ser utilizada en cualquiera de los demás procesos, estos datos son obtenidos por medio de algún dispositivo de entrada, como puede ser un teclado, cd, memoria USB, entre otros.

Almacenamiento de la información. Consiste en recopilar la información y guardarla en una computadora pero más especificamnete en el sistema de información con el cual se este trabajando.

Procesamiento de la información. Transformación y manipulación de los datos de acuerdo a las necesidades de la petición de información que se realice en ese momento.

Salida de información. Obtención impresa de reportes ya sea por un medio visual, en papel, dispositivos de salida (cds, memoria USB, impresoras, etc) y en algunas ocasiones en voz. Generalmente las salidas son reportes impresos a través de una pantalla o en papel, esto depende de cual va a ser su uso.

Este capítulo describe el sistema de información Sistema de Información Académica (SINAC), que pertenece al CINVESTAV propuesto por el Dr. Sergio V. Chapa Vergara, tiene como objetivo principal:

> " contar con un Sistema computacional para la administración académica y para la planeación estratégica. Además, que permita realizar cualquier tipo de consulta estadística a una base de datos y representar la información obtenida de las consultas en forma gráfica."

- Un medio implementado tecnológicamente para grabar, almacenar y distribuir expresiones lingüísticas,
- Así como para extraer conclusiones a partir de dichas expresiones.

3.1. Diseño conceptual de la base de datos

Todo diseño conceptual de bases de datos tiene como propósito proporcionar un esquema que describa los aspectos comunes de los datos de tal forma que, representen la realidad de manera simple (que tanto los diseñadores como los usuarios de la Base de datos puedan entenderlo). Además en el diseño conceptual las especificaciones de los conceptos deben ser formales, es decir, los conceptos deben tener una interpretación única, precisa y bien definida.

El diseño conceptual de la Base de Datos SINAC se muestra en la siguiente tabla:

TABLAS DE LA BASE DE DATOS	
Tabla	Descripción
Adscripciones	Adscripciones de los alumnos(Sección, Departamento, Unidad).
Alumnos	Datos personales de los alumnos.
Áreas	Áreas que agrupan a los diferentes departamentos del CINVESTAV.
Becas	Becas (Monto, Duración, Institución Becaria).
Ciudades	Ciudades (en su caso, delegaciones y municipios) del país.
Departamentos	Departamentos por unidades del CINVESTAV.
Depeconomica	Dependencia económica de los alumnos (trabajará o no durante su estancia).
EntidadesFed	Nombre de las entidades federativas.
Especialidades	Especialidades por unidad CINVESTAV
Extranjeros	Información de alumnos extranjeros.
Investigadores	Datos personales de los investigadores.
Invtesis	Información de las tesis realizadas.
Nacionalidades	Nacionalidades de alumnos e investigadores del CINVESTAV.
Países	Países y nacionalidades.
Secciones	Nombre de las Secciones por departamentos.
Unidades	Unidades del CINVESTAV.

3.1.1. Esquema relacional de la base de datos

Los esquemas relacionales son una herramienta fundamental en el diseño conceptual de una Base de datos debido a que la interpretación y consultas que se realicen a la Base de datos para la extracción de información depende de la **relación** entre entidades, la cual debe estar dada por medio de un atributo en común entre ellas, [1]Los elementos principales en un esquema entidad relación son: entidades, interrelaciones y atributos. Una entidad es un arreglo bidimencional constituida por filas (tuplas) y columnas (atributos).

[1]las entidades comúnmente son conocidas como tablas, sin embargo, el término oficial para nombrar una tabla es **entidad.**

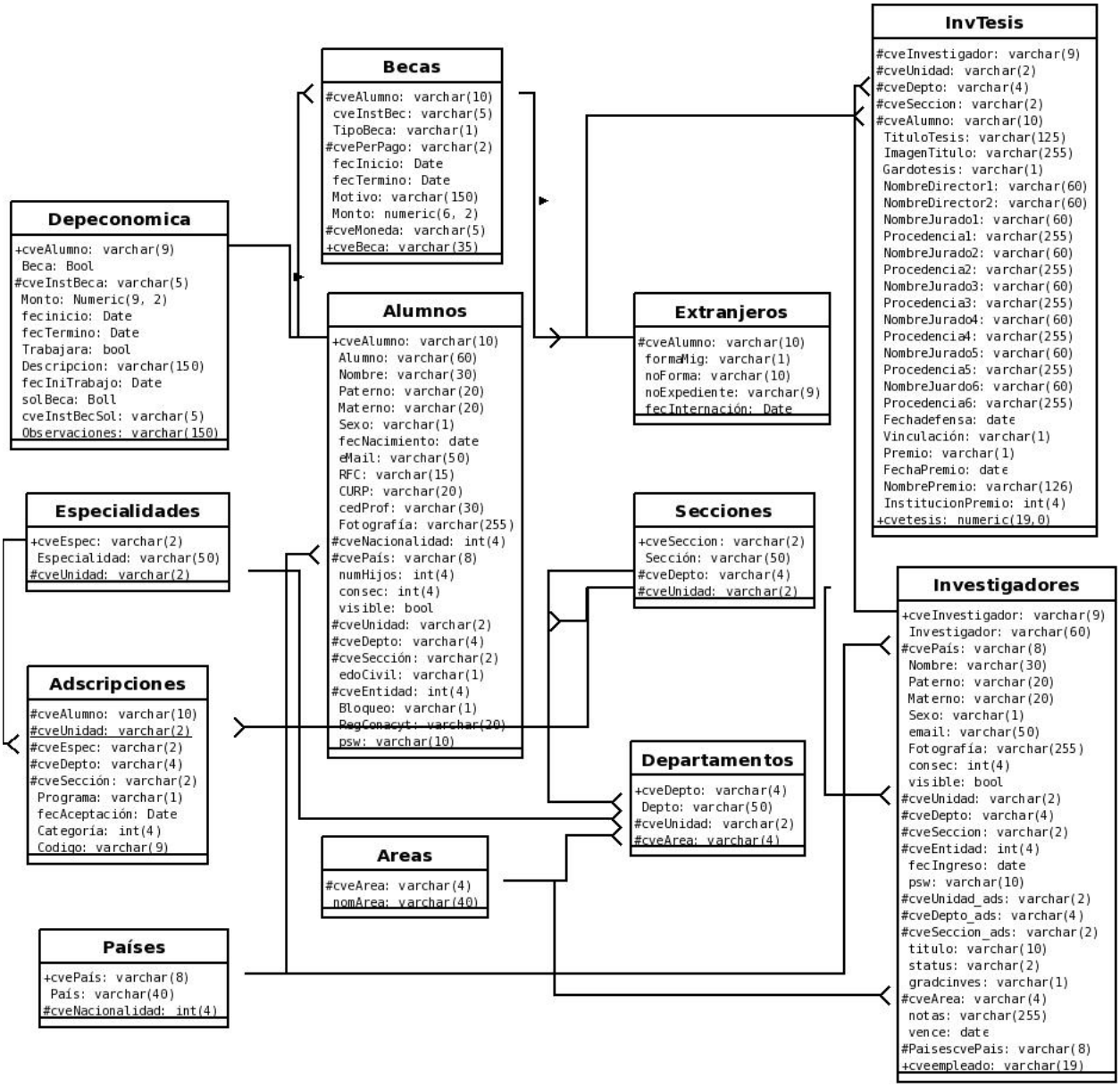

Figura 3.1: Diagrama Entidad relación

.

3.2. Descripción de los metadatos

Los metadatos o "datos sobre datos", es un catálogo de datos que permite consultar tanto información detallada como procedimientos de estos datos. A continuación se describe el archivo de metadatos de la Base de datos del CINVESTAV.

3.2.1. Tabla de adscripciones

A continuación se presenta la tabla adscripciones sobre la cual se realizarán consultas acerca de la sección, departamento y programa al que pertenecen los alumnos, tanto de maestría como de doctorado.

TABLA DE ADSCRIPCIONES			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveAlumno	Char	Clave del alumno
20	cveUnidad	Char	Clave que identifica cada una de las unidades
30	cveEspec	Char	Clave de la especialidad
40	cveDepto	Char	Clave del Departamento
50	cveSección	Char	Clave de la Sección
60	Programa	Char	Grado académico adscrito; M o D
70	fecAceptacion	Date	Fecha de aceptación
80	Categoría	Inte	Categoría del alumno
90	Codigo	Char	Código de preclave
100	uuidAdscripción	Char	Identificador de adscripción

3.2.2. Tabla para Alumnos

La tabla alumnos presenta información personal de los mismos, de la cual se extraerán consultas.

TABLA ALUMNOS			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveAlumno	Char	Clave del alumno
20	Alumno	Char	Nombre del alumno (completo)
30	Nombre	Char	Nombre(s) del alumno
40	Paterno	Char	Apellido Paterno
50	Materno	Char	Apellido Materno
60	sexo	Char	Sexo del alumno (masculino-femenino)
70	fecNacimiento	Date	Fecha de Nacimiento
80	eMail	Char	Dirección de correo electrónico
90	RFC	Char	Registro federal de contribuyentes
100	CURP	Char	Clave única de registro de registro de población
110	cedProf	Char	Cédula profesional
120	Fotografía	Char	Fotografía del alumno
130	cveNacionalidad	Char	clave que identifica el tipo de nacionalidad
140	cvePais	Char	Clave que identifica a cada uno de los países
150	numHijos	Inte	Número de hijos del alumno
160	Consec	Inte	Número consecutivo para la asignación de las claves provisionales de los alumnos
170	Visible	Logi	Permite establecer visibles los registros mientras mientras tengan el valor ”yes”
180	cveUnidad	Char	Clave que identifica a cada una de las unidades
190	cveDepto	Char	Clave del Departamento
200	cveSeccion	Char	Clave de la Sección
210	edoCivil	Char	Clave del estado civil

3.2.3. Tabla de áreas

La tabla áreas permitirá realizar consultas acerca de información correspondiente a las diferentes áreas de investigación a las cuales se encuentran inscritos tanto alumnos como investigadores.

TABLA ÁREAS			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveArea	Char	Clave que identifica a cada una de las áreas
20	nomArea	Char	Nombre del área

3.2.4. Tabal de departamentos

La tabla departamentos será utilizada para realizar consultas acerca del departamento al que pertenecen alumnos e investigadores.

TABLA DEPARTAMENTOS			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveDepto	Char	Clave del Departamento
20	Depto	Char	Nombre del departamento
30	cveUnidad	Char	Clave que identifica a cada una de las unidades
40	cveArea	Char	Clave que identifica a cada una de las áreas

3.2.5. Tabla de dependencias

La siguiente tabla sera utilizada para elaborar consultas que permitan extraer información acerca del respaldo económico de los alumnos durante la maestría o

el doctorado.

TABLA DEPECONOMICA			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveAlumno	Char	Clave del alumno
20	Beca	Logi	Pregunta si el alumno cuenta con Beca
30	cveInstBec	Char	Clave de la institución becaria
40	Monto	Deci-2	Monto recibido por la beca
50	fecInicio	Date	Fecha de inicio de la percepción de la beca
60	fecTermino	Date	Fecha de termino
70	Trabajara	Logi	Si el alumno va a trabajar durante la realización de sus estudios
80	Descripción	Char	Lugar a donde va a trabajar el alumno en caso de así especificarlo
90	fecIniTrabajo	Date	Fecha de inicio de trabajo
100	solBeca	Logi	Pregunta si el alumno solicitará beca
110	cveInstBecSol	Char	Clave de Institución
120	Observaciones	Char	Observaciones

3.2.6. Tabla de especialidades

La siguiente tabla describe las especialidades a las que se encuentran adscritos alumnos e investigadores.

TABLA DE ESPECIALIDADES			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveEspec	Char	Clave de la especialidad
20	Especialidad	Char	Nombre de la especialidad
30	cveUnidad	Char	Clave que identifica a cada una de las unidades

3.2.7. Tabla de países

La siguiente tabla presenta información acerca de la procedencia (país y nacionalidad) de los alumnos.

TABLA PAÍSES			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cvePaís	Char	Clave que identifica a cada uno de los países
20	País	Char	Nombre del país
30	cveNacionalidad	Char	Clave que identifica el tipo de nacionalidad
40	Nacionalidad	Char	Nombre de la nacionalidad relacionada con las entidades Alumnos y Datos PersonalesInv

3.2.8. Tabla de alumnos extranjeros

La siguiente tabla presenta información personal de alumnos extranjeros inscritos en alguna maestría o doctorado.

TABLA EXTRANJEROS			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveAlumno	Char	Clave del alumno
20	fromaMig	Char	Forma migratoria del alumno
30	noForma	Char	Número de forma migratoria
40	noExpediente	Char	Número de expediente
50	fecInternación	Date	Fecha de internación

3.2.9. Tabla de las secciones

La siguiente tabla describe información acerca de la sección a la que pertenecen los departamentos existentes.

TABLA SECCIONES			
Orden	Nombre del campo	Tipo de Dato	Descripción
10	cveseccion	Char	Clave de la Sección
20	Seccion	Char	Nombre de la Sección
30	cveDepto	Char	Clave del Departamento al que pertenece la sección

Capítulo 4

Análisis e interpretación estadística de la base de datos con R

En el capitulo 1 y 2 se describieron los conceptos básicos tanto de estadística descriptiva como de R. En el presente capítulo se emplean estos conceptos mediante una aplicación que ejemplifica visualmente el comportamiento de los datos.

4.1. Conexión de la base de datos a PostgreSQL

La conexión de la base de datos de PostgreSQL a R se realizó a través del paquete **RODBC** basado en el estándar de acceso a Bases de datos, Open Database Connectivity (ODBC).

RODBC funciona mediante diversos métodos e instrucciones que permiten llevar a cabo la conexión de R con el manejador de bases de datos PostgreSQL. Los comandos para instalar RODBC es: **install.packages(RODBC")** si el equipo se encuentra conectado a internet el paquete será descargado directamente desde el sitio oficial de R- CRAN de lo contrario lo buscará en el sistema de archivos del propio, por lo que, si no se tiene servicio de Internet es conveniente descargarlo en el sistema operativo previamente y guardarlo en el directorio en donde se encuentra ubicado el lenguaje R.

4.2. Funciones de R para establecer la conexión de la base de datos

Las principales funciones de R para manipular la base de datos por medio del paquete RODBC, son las siguientes:

odbcConnect . Realiza la conexión al sistema manejador de base de datos PostgreSQL indicando el origen de los datos como es, parámetros, ruta y driver de PostgreSQL. El siguiente código representa la configuración del archivo **odbcConnect**.

```
[ODBC Data Sources]
aracelidata = PostgreSQL

[aracelidata]
ReadOnly = 0
Driver = /usr/lib/odbc/psqlodbcw.so
Servername = localhost
Username = ara
Password = a4a
Database = tesisdb

[ODBC]
InstallDir = /usr/lib
```

odbcinst . Contiene un conjunto de drivers para realizar la interfaz de Bases de Datos.

```
[PostgreSQL]
Description     = PostgreSQL driver for Linux & Windows
Driver          = /usr/lib/odbc/psqlodbcw.so
Setup           = /usr/lib/odbc/libodbcpsqlS.so
```

4.2.1. Ejecución de RODBC

- Para empezar a trabajar con el paquete RODBC se ejecuta el siguiente comando para cargar la librería:
    - library(RODBC)

- Abre la conexión a la Base de Datos PostgrSQL.
    - channel <- odbcConnect(" aracelidata")
- Lista las tablas de la Base de Datos.
    - sqlTables(channel)
- Ejecuta una sentencia SQL a la Base de Datos y devuelve como resultado un objeto de tipo marco de datos(data frame).
    - sqlQuery(channel, "select")
- Cierra la conexión de la Base de Datos
    - odbcClose(channel) o close(channel)

4.2.2. Obtención de estadísticas de la base de datos SINAC

A partir de la base de datos académica SINAC se realizarán las siguientes consultas y se elaborará un estudio descriptivo del comportamiento de los datos respecto a:

- El promedio de edad de:
    - Alumnos inscritos en la licenciatura
    - Total de:
- Alumnos becados
- Investigadores
- Tesis realizadas por doctorado y por maestrías

La siguiente gráfica corresponde a la edad de los alumnos inscritos en las maestrías del Centro de Investigación y Estudios Avanzados (CINVESTAV), en donde el promedio de edad es **33.71**. Otras estadísticas son:

```
                edad
  Dato menor    :19.00
  1rd cuantil   :28.00
  Mediana       :32.00
  3rd Cuantil   :37.00
  Dato mayor    :90.00
```

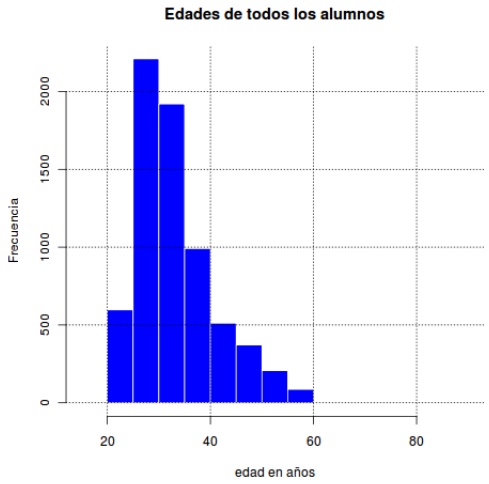

Figura 4.1: Histograma para edad

En la gráfica correspondiente a la función de densidad 4.2 para la variable edad, se observa que la curva es aproximadamente normal.

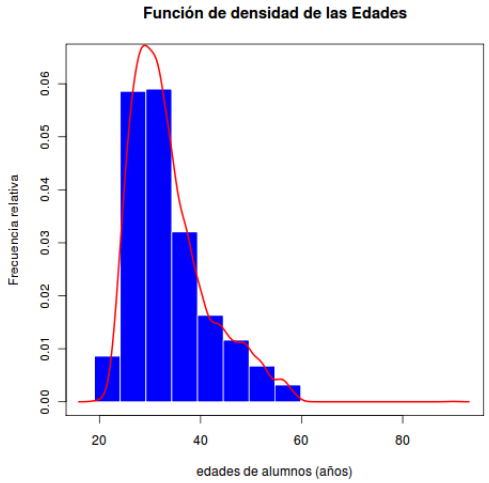

Figura 4.2: Función de densidad

En el polígono de frecuencia absolutas 4.3 para la variable edad se observa que las edades con mayor frecuencia en la que los alumnos estudian una maestría son 27.5, 32.5 y 37.5 (puntos medios de las barras del histograma).

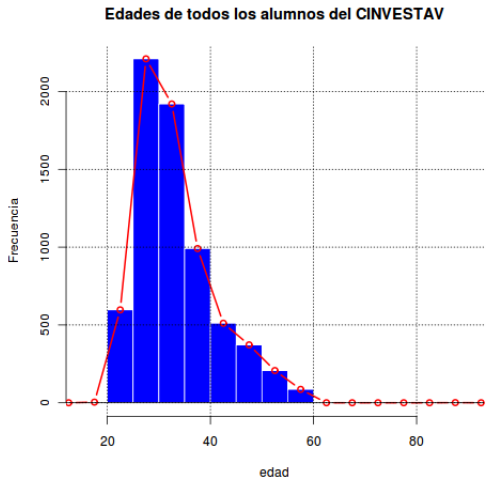

Figura 4.3: Polígono de frecuencias

La ojiva 4.4 muestra las probabilidades que corresponden a un rango o un valor determinado de edad, por ejemplo, de 25 a 30 años de edad existe un 40 % de todos los alumnos inscritos.

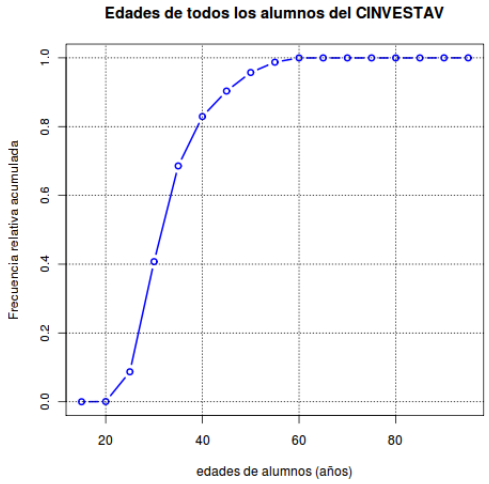

Figura 4.4: Ojiva

La gráfica de barras 4.5 representa la cantidad de todos los alumnos becados por departamento, en este caso el departamento de Ingeniería Eléctrica tiene un número mayor de alumnos becados, le siguen el departamento de Saltillo(IM)[1] y el departamento de Genética y Biología Molecular, los departamentos con menor número de alumnos becados son Computación, Fisiología, Biofísica y neurociencias. Con la tabla se puede verificar que efectivamente los departamentos antes mencionados corresponden a la observación gráfica.

```
                                    depto total
1                            Saltillo (IC)    42
2     Infectómica y Patogénesis Experimental    39
3  Laboratorio de Tecnologías de Información    12
4                                    Física    97
5                           Biología Celular    12
6                                Computación     1
7                             Saltillo (IM)   128
8                            Ecología Humana    16
9                       Ingeniería Eléctrica   241
```

[1] Saltillo: Ingeniería Mecánica

```
10                          Control Automático     2
11                       Biomedicina Molecular    47
12              Sección Externa de Toxicología     9
13               Genética y Biología Molecular   118
14     Fisiología, Biofísica y Neurociencias      1
15                       Matemática Educativa    12
16                 Investigaciones Educativas    10
17                                 Bioquímica    11
18                                  Querétaro    31
```

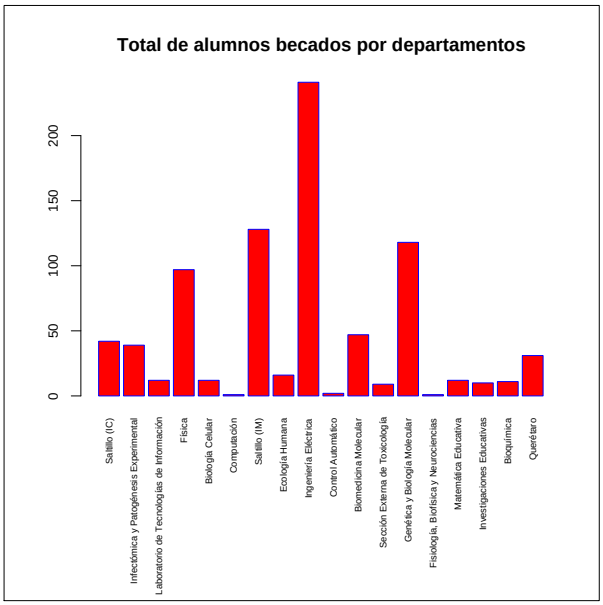

Figura 4.5: Gráfica de barras

La gráfica de pastel 4.6 representa los porcentajes de tesis realizadas por áreas, estas representan la categoría a la que pertenece cada una de las maestrías, las áreas con mayor porcentaje de tesis presentadas son: Tecnología y Ciencias de la Ingeniería 33 %, Ciencias Biológicas y de la Salud 33 %; el área con menor tesis registradas es el área nueva con el 5 %.

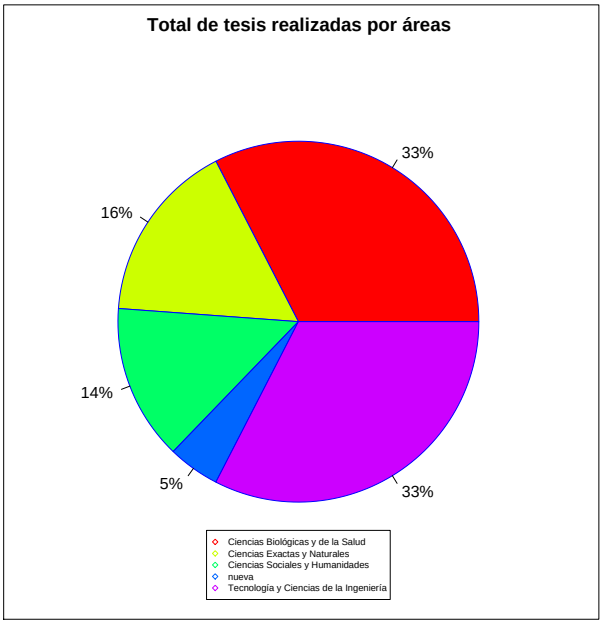

Figura 4.6: Total de tesis por área

.

La gráfica siguiente representa el total de alumnos hombres y el total de alumnos mujeres por departamento en donde se observa que la cantidad de alumnos hombres por departamento es mayor que la cantidad de alumnos mujeres.

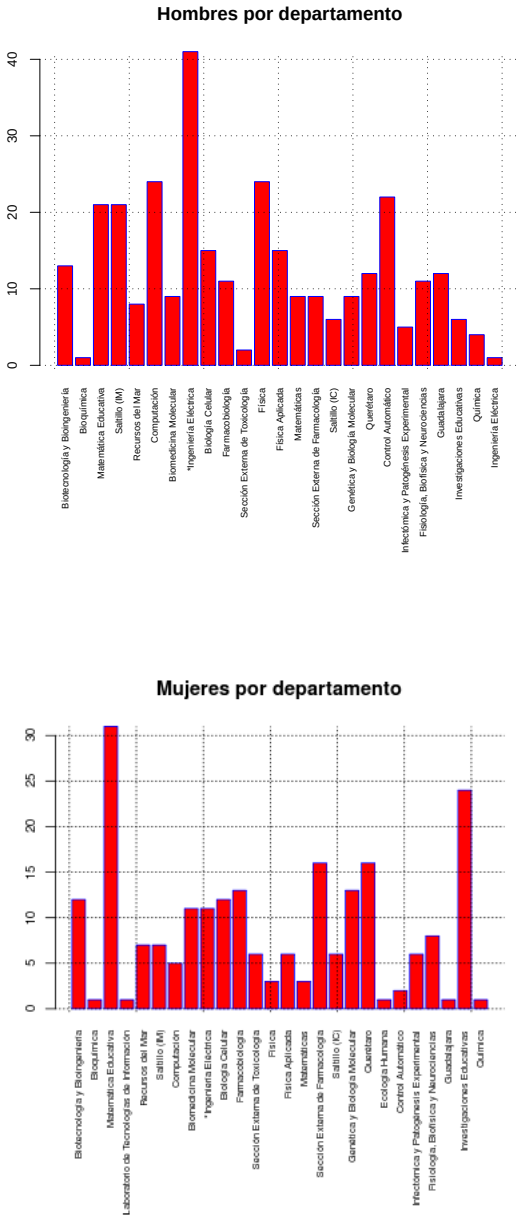

Figura 4.7: Alumnos por género inscritos por departamento

.

La siguiente gráfica presenta el total de tesis realizadas por grado en cada área, se observa que la cantidad de tesis realizadas en la maestría es mucho mayor que la cantidad de tesis realizadas por doctorado; esto sucede debido a que entre más alto es el nivel de estudios menos alumnos hay.

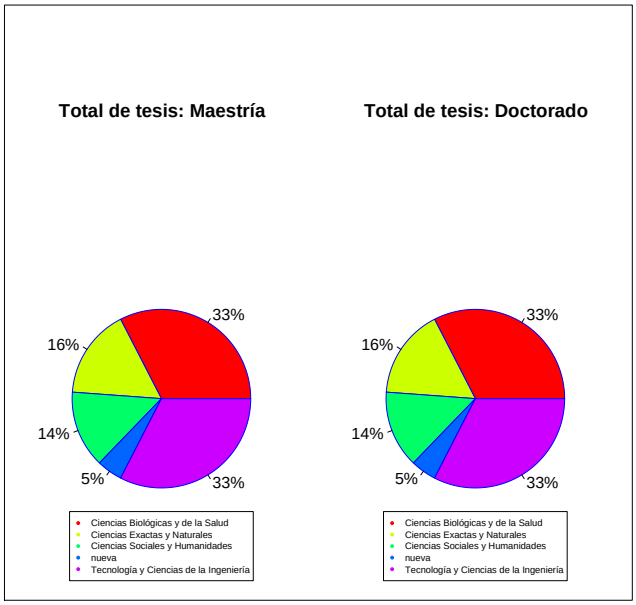

Figura 4.8: Total de tesis por área

.

Si se presenta la gráfica de pastel del total de tesis por grado se encuentra que el porcentajes de la cantidad de tesis por maestría y la cantidad de tesis por doctorado son iguales.

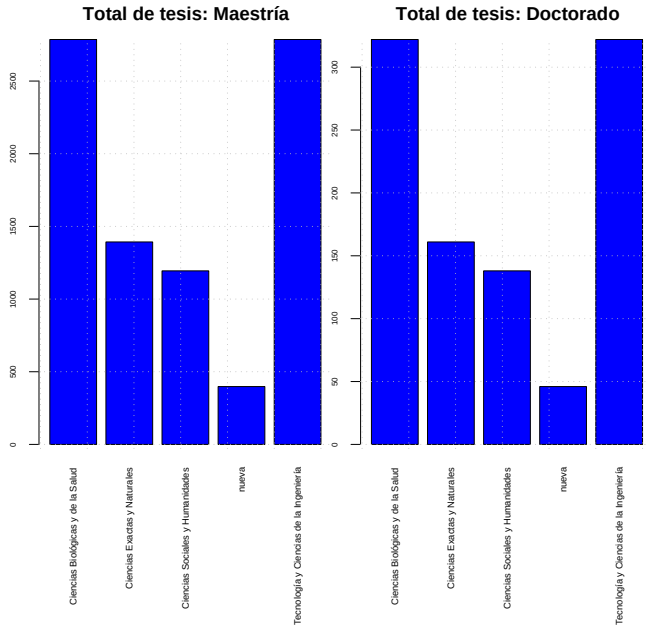

Figura 4.9: Gráfica de barras

.

En la gráfica 4.9 se presenta la misma información que en la gráfica circular 4.8, sin embargo, a en la gráfica de barras se puede observar la diferencia que existe entre las tesis realizadas en la maestría y las tesis realizadas en el doctorado, también se observa que ambas gráficas de barras presentan un comportamiento similar en cada departamento.

Conclusiones

El uso de la herramienta de software libre R facilita manipular datos extraídos directamente de base de datos relacionales (la mayoría de manejadores son relacionales) y aplicar métodos para obtener representaciones visuales de gráficos que apoyan en la toma de decisiones. Una de las ventajas de emplear R en el análisis estadístico es la flexibilidad para construir gráficos atractivos, lo cual se obtiene a partir de parámetros gráficos que son pasados a las funciones gráficas; una vez que se obtiene una gráfica con los parámetros deseados éste código puede ser reutilizado o empleado como una plantilla para realizar más gráficos.

La manipulación y conocimiento de R depende de los requerimientos de cada persona , ya que existe una gran variedad de paquetes con distintas tareas que pueden ser utilizados. Una de las desventajas de R es el uso de comandos, debido a que se tiene que familiarizar con la sintaxis y secuencia de estos. Aunque actualmente existen interfaces gráficas como RKWard y Rcomander, aunque aún no están completamente desarrolladas.

R permite la conexión directa de bases de datos a través de la librería RODBC. A partir de la conexión se elaboran consultas en PosgreSQL que son almacenadas en R para su posterior manipulación y así generar reportes estadísticos descriptivos y visuales.

Los resultados obtenidos del análisis de la información de SINAC, permitieron dar una visión global del comportamiento de estudiantes, egresados e investigadores y así determinar la tendencia actual de las áreas que conforman el CINVESTAV, y donde es necesario tomar acciones que mejoren el desarrollo de la institución.

ANEXO A

Con el objetivo de obtener resúmenes de medidas estadísticas y gráficas se realizarón diversas consultas a la Base de datos.

Consulta para total de alumnos por nacionalidad

```
select substr(CURP,12,2) as edo, count(CURP) as total from
   Alumnos where cveNacionalidad='1' group by substr(CURP,12,2)
   having substr(CURP,12,2)!='' and substr(CURP,12,2)!='-';
```

Consulta de alumnos por genero en cada departamento

```
select alumnos.cvealumno, depto, to_char(fecaceptacion, 'yyyy')
   as anio from adscripciones, alumnos, departamentos where
   alumnos.cvealumno=adscripciones.cvealumno and alumnos.
   cvedepto= departamentos.cvedepto and  to_char(fecaceptacion,
   'yyyy')='2008' and cvenacionalidad='1' and sexo='F';
```

Consulta de alumnos por genero femenino expresado en totales para cada departamento

```
select depto, count(alumnos.cvealumno) as total   from
   adscripciones, alumnos, departamentos where alumnos.cvealumno
   =adscripciones.cvealumno and alumnos.cvedepto= departamentos.
   cvedepto and  to_char(fecaceptacion, 'yyyy')='2008' and
   cvenacionalidad='1' and sexo='F' group by depto;
```

Consulta de alumnos por genero masculino expresado en totales para cada departamento

```
select depto, count(alumnos.cvealumno) as total from
   adscripciones, alumnos, departamentos where alumnos.cvealumno
   =adscripciones.cvealumno and alumnos.cvedepto= departamentos.
   cvedepto and  to_char(fecaceptacion, 'yyyy')='2008' and
   cvenacionalidad='1' and sexo='M' group by depto;
```

Por nacionalidad(mexicanos, extranjeros)

```
select depto, count(alumnos.cvealumno) as total from
   adscripciones, alumnos, departamentos where alumnos.cvealumno
   =adscripciones.cvealumno and alumnos.cvedepto= departamentos.
   cvedepto and to_char(fecaceptacion, 'yyyy')='2008' and
   cvenacionalidad <> '1' and sexo='M' group by depto;

select depto, count(alumnos.cvealumno) as total from
   adscripciones, alumnos, departamentos where alumnos.cvealumno
   =adscripciones.cvealumno and alumnos.cvedepto= departamentos.
   cvedepto and to_char(fecaceptacion, 'yyyy')='2008' and
   cvenacionalidad <> '1' and sexo='F' group by depto;
```

Tesis por investigador

```
select investigador, count(invtesis.cveinvestigador) from
   investigadores, invtesis where investigadores.cveinvestigador
   = invtesis.cveinvestigador group by invtesis.cveinvestigador,
    investigador;
```

Tesis por investigador y tipo de tesis

```
select investigador, count(invtesis.cveinvestigador), gradotesis
    from investigadores, invtesis where investigadores.
   cveinvestigador= invtesis.cveinvestigador group by invtesis.
   cveinvestigador, investigadores.investigador, invtesis.
   gradotesis order by investigador;
```

Total de tesis por área de investigación

```
select  nomarea, count(invtesis.cveinvestigador) from
   departamentos, invtesis,areas where departamentos.cvearea=
   areas.cvearea   group by nomarea order by nomarea;
```

Total de tesis por Maestría

```
select  nomarea, count(invtesis.cveinvestigador) from
   departamentos, invtesis,areas where departamentos.cvearea=
   areas.cvearea and gradotesis='M'  group by  nomarea order by
   nomarea;
```

Total de tesis por Doctorado

```
 select  nomarea, count(invtesis.cveinvestigador) from
    departamentos, invtesis,areas where departamentos.cvearea=
    areas.cvearea and gradotesis='D' group by nomarea order by
    nomarea;
```

ANEXO B

Para llevar acabo los cálculos estadísticos se convirtieron los objetos (datos) extraídos de las consultas debido a que el tipo de objeto que devuelve la consulta una vez que se pasa a R se obtiene como un arreglo de datos (dataframe) y en la mayoría de los casos el tipo de objetos debe ser numérico o caracter. A continuación se presenta el código correspondiente a las estadísticas en R.

Conversión del tipo de objeto para Total de alumnos por Estado

```
tmpb <- estado$edo
tmpb
Conversión de la cadena estado$edo a vector
tmpb <- c(estado$edo)
tmpb
tmpb <- estado$edo
```

Conversión a caracter y después a numérico

```
 tmpb <- as.character(estado$edo)
 tmp2 <- as.numeric(estado$total)
```

Gráficas de barras para Total de alumnos por estado Muestra gráfica

```
barplot(tmp2,names.arg=tmpb, main="Total de alumnos por estado",
col=heat.colors(6), border= "blue", xlab="Estados",
ylab="Alumnos");
```

Exportar a formato pdf

```
pdf("gb.pdf")
```

```
gb<- barplot(tmp2,names.arg=tmpb, main="Total de alumnos por
    estado", col=heat.colors(6), border= "blue", xlab="Estados",
    ylab="Alumnos");
dev.off()
```

Gráfica circular o de pastel para total de alumnos por estado

```
pdf("gcc.pdf")
gcc<-pie(tmp2, labels=tmpb, main="Total de alumnos por estado",
    radius=0.9, col=heat.colors(8), border= "blue",
 xlab="Estados", ylab="Alumnos");
dev.off()
```

Gráfica de líneas para total de alumnos por estado

```
  pdf("gl.pdf")
  gl<-plot((tmp2), type= "l", col= "red")
  dev.off()
```

Grafica para edades

```
edadlineasp<-sqlQuery(channel, "select substr(to_char(age(
    current_date, fecnacimiento), 'YYYY-MM-DD'), 3,2) as edad
    from alumnos where substr(to_char(age(current_date,
    fecnacimiento), 'YYYY-MM-DD'),3,2)>='19'");
png("edadhist.png")
hist(edadlineasp$edad, xlab="edad", ylab="Frecuencia", main="
    Edades de todos los alumnos", col= 4, border="white")
grid(col="black")
dev.off()
```

Poligono de frecuencias

```
png("pf.png")
```

Cargando la librería agricolae para obtener:

- Tablas de efrecuencias
- Ojivas

- Polígono de frecuencias

```
library(agricolae)
gfreq<-hist(edadlineasp$edad, xlab="edad", ylab="Frecuencia",
   main="Edades de todos los alumnos del CINVESTAV", col= 4,
   border="white")
polygon.freq(gfreq,frequency=1,col="red", type="b", lwd = 2)
grid(col="black")
dev.off()
```

Ojiva de frecuencias para edades

```
png("ojiva.png")
gfreq<-hist(edadlineasp$edad, xlab="edad", ylab="Frecuencia",
   main="Edades", col= 4, border="white", plot=FALSE)
ojiva.freq(gfreq, col="blue", type="b", lwd = 2, xlab="edades de
    alumnos (años)", ylab="Frecuencia relativa acumulada", main=
   "Edades de todos los alumnos del CINVESTAV")
grid(col="black")
dev.off()
```

Gráfica de frecuencias=histograma

```
png("gf.png")
gfrec<-graph.freq(edadlineas$edad, plot=TRUE, col=4)
polygon.freq(gfrec,frequency=1,col="red", type="b")
grid(col="black")
dev.off()
```

Ojiva para el tipo graph.freq

```
gfrec<-graph.freq(edadlineas$edad, plot=FALSE, col=4)
ojiva.freq(gfreq, col="red", type="b", xlab="edades de alumnos
   en años", ylab="Frecuencia relativa acumulada")
```

Curva de histograma: función de densidad

```
png("fd.png")
gfrec<-graph.freq(edadlineasp$edad, frequency=3, xlab="edades de
    alumnos en años", ylab="Frecuencia relativa",  main="Función
    de densidad de las Edades", col= 4, border="white", plot=
   TRUE) lines(density(edadlineasp$edad), col="red", lwd=2)
```

```
dev.off()
```

Imprime tabla de frecuencias

```
Statistics
     tablafreq<-stat.freq(gfreq)
     print(tablafreq)
$variance
[1] 59.65578

$mean
[1] 33.2117

$median
[1] 31.66276

$mode
     [- -]     mode
[1,] 25 30 29.23805

     frequency table full
     round(table.freq(gfreq),2)
Inf Sup   MC   fi  fri   Fi  Fri
   15  20 17.5    4 0.00    4 0.00
   20  25 22.5  597 0.09  601 0.09
   25  30 27.5 2210 0.32 2811 0.41
   30  35 32.5 1920 0.28 4731 0.69
   35  40 37.5  991 0.14 5722 0.83
   40  45 42.5  511 0.07 6233 0.90
   45  50 47.5  372 0.05 6605 0.96
   50  55 52.5  207 0.03 6812 0.99
   55  60 57.5   86 0.01 6898 1.00
   60  65 62.5    0 0.00 6898 1.00
   65  70 67.5    0 0.00 6898 1.00
   70  75 72.5    0 0.00 6898 1.00
   75  80 77.5    0 0.00 6898 1.00
   80  85 82.5    0 0.00 6898 1.00
   85  90 87.5    1 0.00 6899 1.00

intervalo
dato mayor-dato menor= 90-19= 71
summary(edadlineasp)
      edad
 Min.   :19.00
 1st Qu.:28.00
```

```
 Median :32.00
 Mean   :33.71
 3rd Qu.:37.00
 Max.   :90.00
 quantile(edadlineasp$edad)
  0%  25%  50%  75% 100%
  19   28   32   37   90
```

Conversión del tipo de objetos para total becados por departamento

```
becados <- sqlQuery(channel, paste("select p.Depto, count(a.
    alumno) as total from alumnos a, departamentos p,
    depeconomica d where a.cveDepto=p.cveDepto and a.cveAlumno=d.
    cveAlumno and  d.Beca='TRUE' and cvePais='MX' group by p.
    Depto;"));
v<- as.numeric(becados$total)
nomgraf<- as.character(becados$depto)
```

tabla de frecuencias para becados

```
 hi <- hist(v)
 table.freq(hi)
  Inf Sup  MC fi        fri Fi       Fri
    0  50  25 14 0.77777778 14 0.7777778
   50 100  75  1 0.05555556 15 0.8333333
  100 150 125  2 0.11111111 17 0.9444444
  150 200 175  0 0.00000000 17 0.9444444
  200 250 225  1 0.05555556 18 1.0000000
```

Gráfica de total de alumnos becados por departamento

```
pdf("be")
par (oma=c(2,2,2,2))
par(mar=c(10.5,4,4,2))
barplot(v, names.arg=nomgraf, main="Total de alumnos becados por
    departamentos", cex.names=0.6, las=3, cex.axis=0.8, axes=
    TRUE, border="blue", col="red")
box("inner", col="black")
grid(col="black")
dev.off()
```

Conversión del tipo de objetos para la consulta del total por genero en cada departamento

```
genero <-sqlQuery(channel, "select depto, count(alumnos.
    cvealumno) as total from adscripciones, alumnos,
    departamentos where alumnos.cvealumno=adscripciones.cvealumno
     and alumnos.cvedepto=departamentos.cvedepto and to_char(
    fecaceptacion, 'yyyy')='2008' and cvenacionalidad='1' and
    sexo='F' group by depto");
femdepto <- c(genero$depto)
femtotal <- c(genero$total)
femdepto <- as.character(genero$depto)
femtotal <- as.numeric(genero$total)
```

Graficas para el total de sexo femenino por departamento

```
par(oma=c(2,2,2,2))
par(mar=c(10.5,4,4,2))
barplot(femtotal, names.arg=femdepto, main="Total de alumnos de
    género femenino por departamentos", cex.names=0.6, las=3,
    cex.axis=0.8, axes=TRUE, border="blue", col="red")
pdf("generof.pdf")
par (oma=c(1,4,4,.5))
par(mar=c(1,1,1,5))
pie(femtotal, labels=NA, main="Total de alumnos de género
    femenino por departamentos", radius=.8, col=rainbow(10),
    border="blue")
legend("bottom", femdepto, col=rainbow(10),
inset=0.020, pch=20, cex=0.4,  ncol=3)
dev.off()
```

Conversión del tipo de objetos para total masculino

```
genero<-sqlQuery(channel, "select depto, count(alumnos.cvealumno
    ) as total from adscripciones, alumnos, departamentos where
    alumnos.cvealumno=adscripciones.cvealumno and alumnos.
    cvedepto= departamentos.cvedepto and  to_char(fecaceptacion,
    'yyyy')='2008' and cvenacionalidad ='1' and sexo ='M' group
    by depto");
masdepto <- c(genero\$depto)
mastotal <- c(genero\$total)
masdepto <- as.character(genero\$depto)
mastotal <- as.numeric(genero\$total)
```

Gráfica para sexo masculino

```
par (oma=c(2,2,2,2))
par(mar=c(10.5,4,4,2))
barplot(mastotal, names.arg=masdepto, main="sexo masculino por
    departamentos", cex.names=0.6, las=3, cex.axis=0.8, axes=TRUE
    , border="blue", col="red") grid(col="black")
```

Gráfica de total por sexo

```
pdf("comparaxsexo.pdf")
par(mfrow=c(1,2))
par (oma=c(1,1.2,1,.3))
par(mar=c(10.5,.9,2.7,.6))
barplot(mastotal, names.arg=masdepto, main="Hombres por
    departamento", cex.names=0.6, las=3, cex.axis=0.5, axes=TRUE,
     border="black", col="blue") grid(col="gray")
barplot(femtotal, names.arg=femdepto, main="Mujeres por
    departamento", cex.names=0.6, las=3, cex.axis=0.5, axes=TRUE,
     border="black", col="red") grid(col="gray")
dev.off()
```

Conversión de tipo de objetos para la consulta tesis por áreas

```
areas<-sqlQuery(channel, "select nomarea, count(invtesis.
    cveinvestigador) as total from departamentos, invtesis,areas
    where   departamentos.cvearea=areas.cvearea group by nomarea
    order by nomarea");

 tesistotal <- c(areas$total)
 ubicacion <- c(areas$nomarea)
 tesistotal <- as.numeric(areas$total)
 ubicacion <- as.character(areas$nomarea)
```

Gráfica de total de tesis realizadas por áreas

```
pdf("tesisxarea.pdf")
par (oma=c(1,2.5,2,1.5))
par(mar=c(1,.7,2,1))
pct <- round(tesistotal/sum(tesistotal)*100)
lbls<- paste(pct, "%", sep="" )
pie(tesistotal, labels=lbls, main="Total de tesis realizadas por
     áreas", radius=.7, col=rainbow(5), border="blue")
```

```
legend("bottom", col=rainbow(5), inset=0.001, pch=23, cex=0.6,
    ncol=1)
box("inner", col="black")
dev.off()
```

Tesis por maestría

```
gradomaestria<-sqlQuery(channel, "select nomarea, count(invtesis
    .cveinvestigador) as total from departamentos, invtesis,areas
     where departamentos.cvearea=areas.cvearea and gradotesis='M'
     group by nomarea order by nomarea");
areamaestria <-c(gradomaestria$nomarea)
totalmaestria <-c(gradomaestria$total)
areamaestria <-as.character(gradomaestria$nomarea)
totalmaestria <-as.numeric(gradomaestria$total)
```

Gráfica de tesis por maestría

```
pct<-round(totalmaestria/sum(totalmaestria)*100)
lbls<-paste(pct, "%", sep="" )
pie(totalmaestria, labels=lbls, main="Total de tesis: Maestría",
     radius=.7, col=rainbow(5), border="blue")
legend("bottom", areamaestria, col=rainbow(5), inset=0.001, pch
    =20, cex=0.6, ncol=1)
box("inner", col="black")
```

Tesis por doctorado

```
gradodoctorado<-sqlQuery(channel, "select nomarea, count(
    invtesis.cveinvestigador) as total from departamentos,
    invtesis,areas where departamentos.cvearea=areas.cvearea and
    gradotesis='D' group by nomarea order by nomarea");
areadoctorado <-c(gradodoctorado$nomarea)
totaldoctorado <-c(gradodoctorado$total)
areadoctorado <-as.character(gradodoctorado$nomarea)
totaldoctorado <-as.numeric(gradodoctorado$total)
```

Gráfica por doctorado

```
pct <- round(totaldoctorado/sum(totaldoctorado)*100)
lbls<- paste(pct, "%", sep="" )
```

```
pie(totaldoctorado, labels=lbls, main="Total de tesis: Doctorado
    ", radius=.7, col=rainbow(5), border="blue")
legend("bottom", areadoctorado, col=rainbow(5), inset=0.001, pch
    =20, cex=0.6, ncol=1)
box("inner", col="black")
```

Clasificación de tesis por grados (maestría y doctorado): Barras

```
pdf("comparaxgrado.pdf")
par(mfrow=c(1,2))
par (oma=c(1,1.2,1,.3))
par(mar=c(9,.9,2.7,.6))
barplot(totalmaestria, names.arg=areamaestria, main="Total de
    tesis: Maestría", cex.names=0.6, las=3, cex.axis=0.5, axes=
    TRUE, border="black", col="blue")
grid(col="gray")
barplot(totaldoctorado, names.arg=areadoctorado, main="Total de
    tesis: Doctorado", cex.names=0.6, las=3, cex.axis=0.5, axes=
    TRUE, border="black", col="blue")
grid(col="gray")
dev.off()
```

Bibliografía

[1] *Proyecto R UCA*. en línea. http://knuth.uca.es/R/doku.php, consultado: 2008-08-21.

[2] *R programming: Graphical Control*. http://www.r-project.org/.

[3] M. Arai: *A Brief Guide to R for Beginners in Econometrics*. Teaching Material, Department of Economics, Stockholm University, 2004.

[4] Juan Carlos Correa y Nelfi González: *Gráficos Estadísticos con R*, 2002.

[5] P. Dalgaard: *Introductory Statistics with R*. Springer, 2002.

[6] David: *Evaluación de Software: GNU R*, Julio 2008. http://crisol.uc3m.es/index.php/gnur, Consultado: 2008-11-22.

[7] Felipe de Mendiburu : *Algunos topicos en R*, Dic. 2007. http://tarwi.lamolina.edu.pe/~fmendiburu/, Coloquio de R.

[8] Jay L. Devore: *Probabilidad y estadística para ingeniería y ciencias*. THOMSON LEARNING, spanishquinta edición, 2001.

[9] J.E. Freund y J.E. Freund: *Estadística matemática con aplicaciones*. Pearson Educación, 2000.

[10] Alor Hernández Giner: *Sistema de Consulta Estadística con un Ambiente Gráfico utilizando Tecnología Dinámica*. Tesis de Licenciatura, CINVESTAD, México D.F, Septiembre 2001.

[11] J.R. Gonzalez, L. Armengol, X. Sole, E. Guino, J.M. Mercader, X. Estivill y V. Moreno: *SNPassoc: an R package to perform whole genome association studies*. Bioinformatics, 23(5):654, 2007.

[12] David V. Conesa Guillén: *Introducción al entorno R*. Grupo de Estadística Espacial y Temporal en Epidemiología y Medio Ambiente. Sesión 4.

[13] Sergio Alejandro Matias Hernández: *Manual de curso-taller de Itroducción al software estadístico R*. Direccion General de Asuntos del Personal Academico (DGAPA), 2009. pág. 199-201.

[14] P. Kuhnert, B. Venables y A. Cleveland: *An Introduction to R: Software for Statistical Modelling & Computing*, 2005.

[15] Originally Michael Lapsley y from Oct 2002 B. D. Ripley: *RODBC: ODBC Database Access*, 2008. R package version 1.2-3.

[16] J. H. Maindonald: *Using R for Data Analysis and Graphics*, Enero 2008. Consultado: 2008-08-22.

[17] W. Mendenhall, R.J. Beaver y B.M. Beaver: *Introduction To Probability And Statistics*. Thomson Brooks/Cole, 2005.

[18] Emmanuel Paradis: *R para principiantes*, mar 2003. Traducido por: Jorge A. Ahumada.

[19] Joaquín Ortega Sánchez: *Introducción a R: Estructuras de Datos y Gráficos*. Guanajuato, Gto., Mexico, Jun-Jul 2008. Sesión 3. Verano de Probabilidad y Estadística.

[20] Abraham Silberschatz, Henry F. Korth y S. Sudarshan: *Fundamentos de Bases de Datos*. McGraw-Hill, spanishcuarta edición, 2002, ISBN 0-07-228363-7. pág. 5-12.

[21] Development Core Team: *R Data Import/Export*, Agos 2008, ISBN 3-900051-10-0. Consultado: 2008-08, pág. 16-18.

[22] Leonardo Collado Torres y María Gutiérrez Arcelus: *Principios de Estadística*, Feb-Jun 2009. pág. 8-12.

[23] R.E. Walpole y R.H. Myers: *Probabilidad y estadística para ingenieros*. Pearson Educación, 1999.

Bibliografía

[1] *Proyecto R UCA*. en línea. http://knuth.uca.es/R/doku.php, consultado: 2008-08-21.

[2] *R programming: Graphical Control*. http://www.r-project.org/.

[3] M. Arai: *A Brief Guide to R for Beginners in Econometrics*. Teaching Material, Department of Economics, Stockholm University, 2004.

[4] Juan Carlos Correa y Nelfi González: *Gráficos Estadísticos con R*, 2002.

[5] P. Dalgaard: *Introductory Statistics with R*. Springer, 2002.

[6] David: *Evaluación de Software: GNU R*, Julio 2008. http://crisol.uc3m.es/index.php/gnur, Consultado: 2008-11-22.

[7] Felipe de Mendiburu : *Algunos topicos en R*, Dic. 2007. http://tarwi.lamolina.edu.pe/~fmendiburu/, Coloquio de R.

[8] Jay L. Devore: *Probabilidad y estadística para ingeniería y ciencias*. THOMSON LEARNING, spanishquinta edición, 2001.

[9] J.E. Freund y J.E. Freund: *Estadística matemática con aplicaciones*. Pearson Educación, 2000.

[10] Alor Hernández Giner: *Sistema de Consulta Estadística con un Ambiente Gráfico utilizando Tecnología Dinámica*. Tesis de Licenciatura, CINVESTAD, México D.F, Septiembre 2001.

[11] J.R. Gonzalez, L. Armengol, X. Sole, E. Guino, J.M. Mercader, X. Estivill y V. Moreno: *SNPassoc: an R package to perform whole genome association studies*. Bioinformatics, 23(5):654, 2007.

[12] David V. Conesa Guillén: *Introducción al entorno R*. Grupo de Estadística Espacial y Temporal en Epidemiología y Medio Ambiente. Sesión 4.

[13] Sergio Alejandro Matias Hernández: *Manual de curso-taller de Itroducción al software estadístico R*. Direccion General de Asuntos del Personal Academico (DGAPA), 2009. pág. 199-201.

[14] P. Kuhnert, B. Venables y A. Cleveland: *An Introduction to R: Software for Statistical Modelling & Computing*, 2005.

[15] Originally Michael Lapsley y from Oct 2002 B. D. Ripley: *RODBC: ODBC Database Access*, 2008. R package version 1.2-3.

[16] J. H. Maindonald: *Using R for Data Analysis and Graphics*, Enero 2008. Consultado: 2008-08-22.

[17] W. Mendenhall, R.J. Beaver y B.M. Beaver: *Introduction To Probability And Statistics*. Thomson Brooks/Cole, 2005.

[18] Emmanuel Paradis: *R para principiantes*, mar 2003. Traducido por: Jorge A. Ahumada.

[19] Joaquín Ortega Sánchez: *Introducción a R: Estructuras de Datos y Gráficos*. Guanajuato, Gto., Mexico, Jun-Jul 2008. Sesión 3. Verano de Probabilidad y Estadística.

[20] Abraham Silberschatz, Henry F. Korth y S. Sudarshan: *Fundamentos de Bases de Datos*. McGraw-Hill, spanishcuarta edición, 2002, ISBN 0-07-228363-7. pág. 5-12.

[21] Development Core Team: *R Data Import/Export*, Agos 2008, ISBN 3-900051-10-0. Consultado: 2008-08, pág. 16-18.

[22] Leonardo Collado Torres y María Gutiérrez Arcelus: *Principios de Estadística*, Feb-Jun 2009. pág. 8-12.

[23] R.E. Walpole y R.H. Myers: *Probabilidad y estadística para ingenieros*. Pearson Educación, 1999.