



PRACTICA 2

SEMINARIO DE SISTEMAS 2

Bayron Romeo
Axpuac Yoc

201314474
JUNIO 2020

**BAYRON ROMEO
AXPUAC YOC**

PRACTICA 2

GUATEMALA JUNIO 2020

RESUMEN

En el presente documento se da a conocer una solución de Business Intelligence por medio del lenguaje de programación R, El concepto ‘Business Intelligence’, o inteligencia empresarial, se refiere a la utilización de datos en una empresa para facilitar la toma de decisiones dentro de la misma. Es un conjunto de estrategias y herramientas enfocadas al análisis de datos de una empresa mediante el análisis de datos existentes. El business intelligence se caracteriza por mostrar el estado pasado y presente de la organización. Así, pues, recae en las manos de los directivos utilizar este conocimiento para tomar decisiones estratégicas. Un proyecto de business intelligence puede ser una tarea compleja por lo que se requiere una buena organización. Para organizar nuestro sistema de Business Intelligence, consecuentemente, deberemos tener en cuenta los tres elementos:

1. Evaluar nuestras fuentes de datos, que pueden ser internas o externas:

- Accesibilidad
- Fiabilidad
- Calidad
- Posibilidades de integración

2. Convertirlas en información:

- Contextualizando: cuándo, cómo y por qué se han generado.
- Categorizando: cómo se pueden medir y clasificar.
- Calculando: procesarlos si es necesario.
- Corrigiendo: eliminar errores, duplicados e inconsistencias.
- Condensando: definir criterios de agregación, resumir

3. Definir el conocimiento que podremos obtener:

- Comparando la información con otra.
- Haciendo predicciones, buscando interrelaciones...

Para esta práctica tenemos en nuestras manos cinco documentos con información a analizar, el primero documento consiste en análisis de ventas de una empresa, el archivo numero dos nos permitirá analizar un estudio de cardio, el tercer archivo de nuestro sistema nos permitirá analizar diferentes datos sobre asesinatos a nivel internacional, el cuarto archivo es un análisis de futbolistas y sus equipos, por último tenemos un archivo en el cual se lleva a cabo el estudio de relaciones entre tres variables estrés, depresión y ansiedad.

LENGUAJE R

R es un entorno de software libre (licencia GNU GPL) y lenguaje de programación interpretado, es decir, ejecuta las instrucciones directamente, sin una previa compilación del programa a instrucciones en lenguaje máquina. El término entorno, en R, se refiere a un sistema totalmente planificado y coherente, en lugar de una acumulación de herramientas específicas e inflexibles, como suele ser el caso en otros softwares de análisis de datos. Este entorno es comúnmente utilizado para la computación estadística y gráfica, ya que dispone de una amplia variedad de técnicas estadísticas (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, agrupamiento, etc.) y gráficas. Funciona en plataformas UNIX y sistemas similares (incluidos FreeBSD y Linux), Windows y MacOS. Su desarrollo actual es responsabilidad del R Development Core Team. Forma parte de un proyecto colaborativo y abierto donde los usuarios pueden publicar paquetes que extienden su configuración básica (repositorio oficial de paquetes). Además, se puede descargar gratis a través del siguiente enlace: <https://www.r-project.org/>.



Características de R

- Manejo y almacenamiento efectivo de los datos.
- Un conjunto de operadores para la realización de cálculos con matrices.
- Una gran colección de herramientas para el análisis de datos.
- Utilidades gráficas para la visualización de datos.
- Un lenguaje de programación bien desarrollado que incluye saltos condicionales, bucles, funciones recursivas, utilidades para la entrada y salida de datos, etc.
- Tiene un formato de documentación basado en LaTeX, que se utiliza para proporcionar documentación completa tanto en formato físico como digital.

El lenguaje de programación R se integra bien con otros lenguajes de programación como C, C++ o Fortran para tareas de análisis de datos computacionalmente intensivas (alto consumo de recursos como CPU y RAM). Además, puede integrarse con distintas bases de datos y existen bibliotecas que facilitan su utilización desde lenguajes de programación interpretados como Perl y Python.

Uso de R en Big Data

En el ámbito del Big Data se utiliza para la manipulación, procesamiento y visualización gráfica de los datos. R nos permite:

- Crear visualizaciones de datos de alta calidad.
- Crear dashboards para visualizar y analizar datos.
- Crear informes automáticos.
- Disponer de herramientas de análisis estadístico para ahondar en el conocimiento de los datos.

R es algo más que un lenguaje de programación. El usuario no programa propiamente, sino que utiliza R interactivamente: ensaya, se equivoca y vuelve a probar. Solo cuando termina el ciclo y el resultado es satisfactorio, produce un resultado final que, generalmente, no es un programa, sino un informe.

Se utiliza en todas las fases de análisis de datos:

- Adquisición de los datos de las fuentes disponibles: bases de datos, archivos de texto, etc.
- Preparación de los datos: eliminación de duplicados, datos incorrectos, valores extremos, etc.
- Análisis de los datos: construcción de modelos predictivos, de clasificación, de agrupamiento...
- Comunicación de los resultados: realización de informes para presentación de los resultados y conclusiones.
- Aplicación de los resultados obtenidos: por ejemplo, utilización de modelos predictivos desarrollados para en función de una serie de datos históricos (datos de entrenamiento y test del modelo) predecir ciertas salidas.

Las características y diferentes aplicaciones de R lo convierten en una herramienta básica para los analistas de datos.



RStudio es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo. RStudio está disponible para Windows, Mac y Linux o para navegadores conectados a RStudio Server o RStudio Server Pro (Debian / Ubuntu, RedHat / CentOS, y SUSE Linux). RStudio tiene la misión de proporcionar el entorno informático estadístico R. Permite un análisis y desarrollo para que cualquiera pueda analizar los datos con R.

Características

IDE construido exclusivo para R

- El resaltado de sintaxis, auto completado de código y sangría inteligente.
- Ejecutar código R directamente desde el editor de código fuente.
- Salto rápido a las funciones definidas.

Colaboración

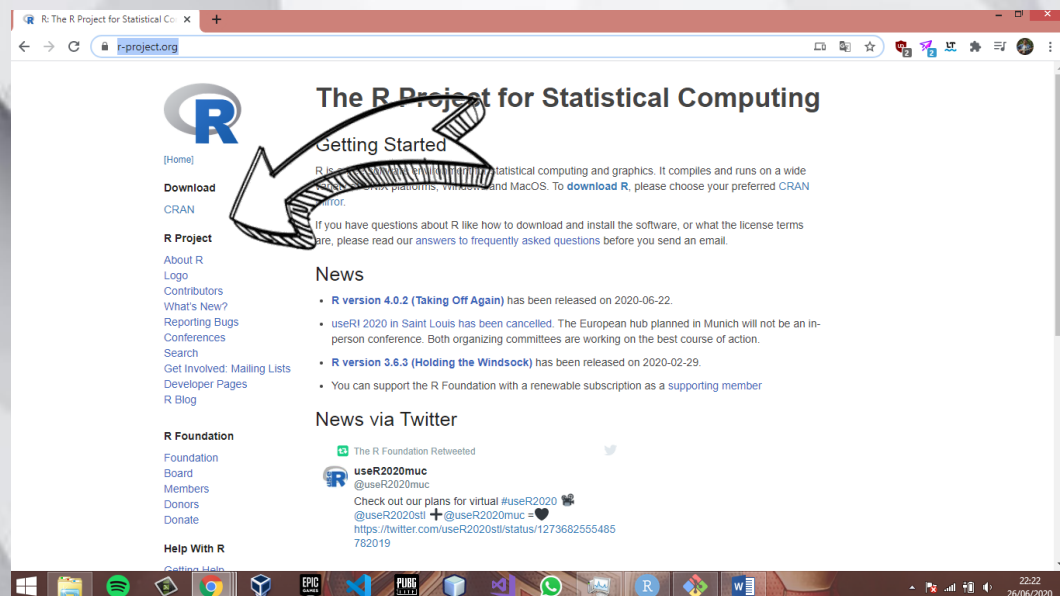
- Documentación y soporte integrado.
- Administración sencilla de múltiples directorios de trabajo mediante proyectos.
- Navegación en espacios de trabajo y visor de datos.
- Potente autoría y depuración.

Depurador interactivo para diagnosticar y corregir los errores rápidamente.

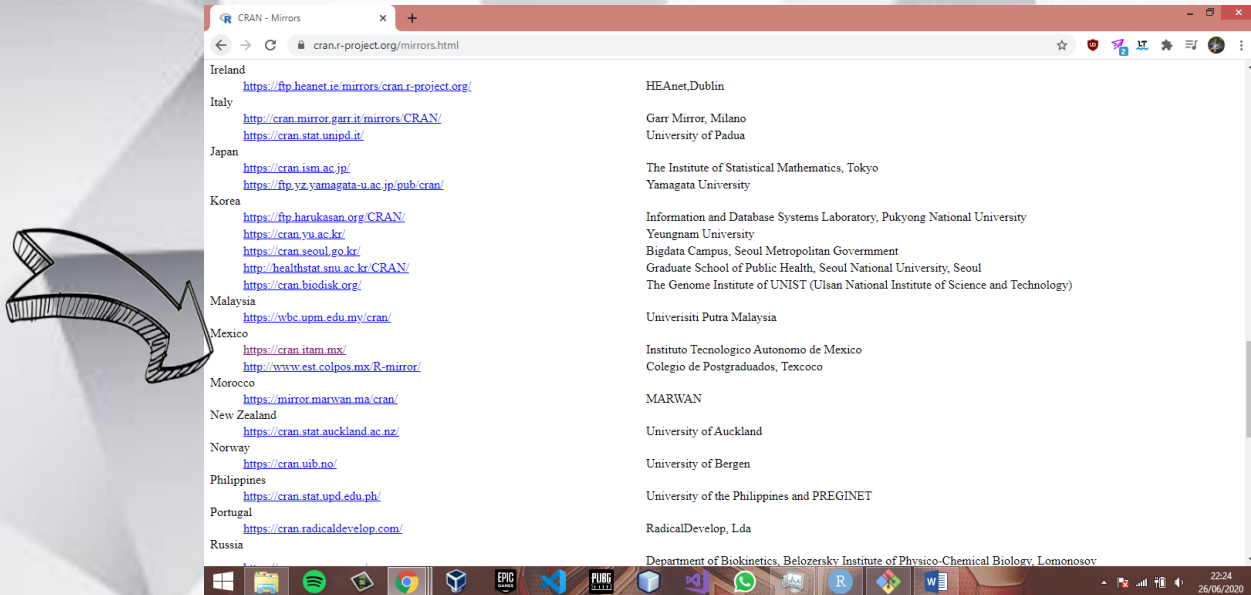
- Herramientas de desarrollo extensas.
- Autoría con Sweave y R Markdown.

INSTALACIÓN DE HERRAMIENTAS

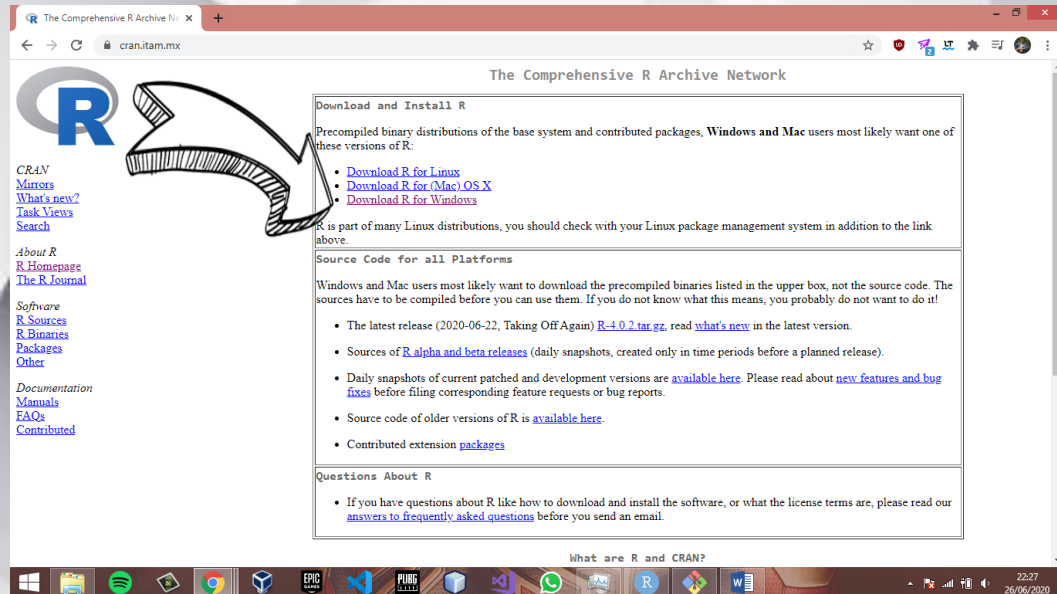
- 1) Instalación de R: para instalar primero debemos ingresar al siguiente enlace <https://www.r-project.org/> ingresamos a la opción CRAN.



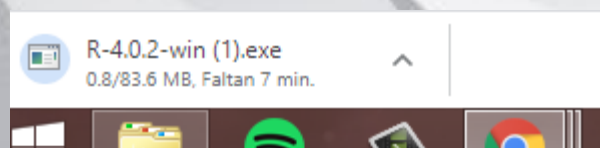
- 2) Dentro de la nueva página a la que accedimos seleccionaremos el servidor al cual nos acoplamos más.



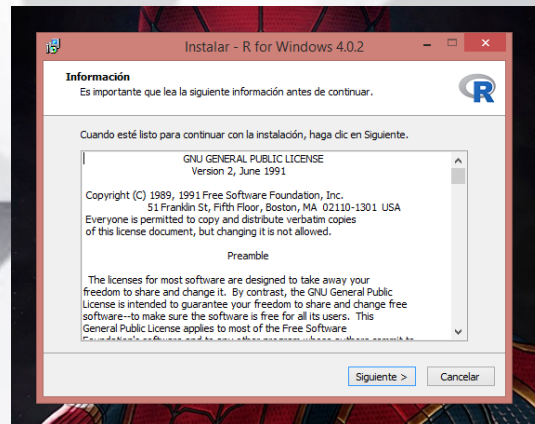
- 3) Luego descargaremos el paquete para nuestro sistema operativo, en esta oportunidad descargaremos el paquete para el sistema de Windows.



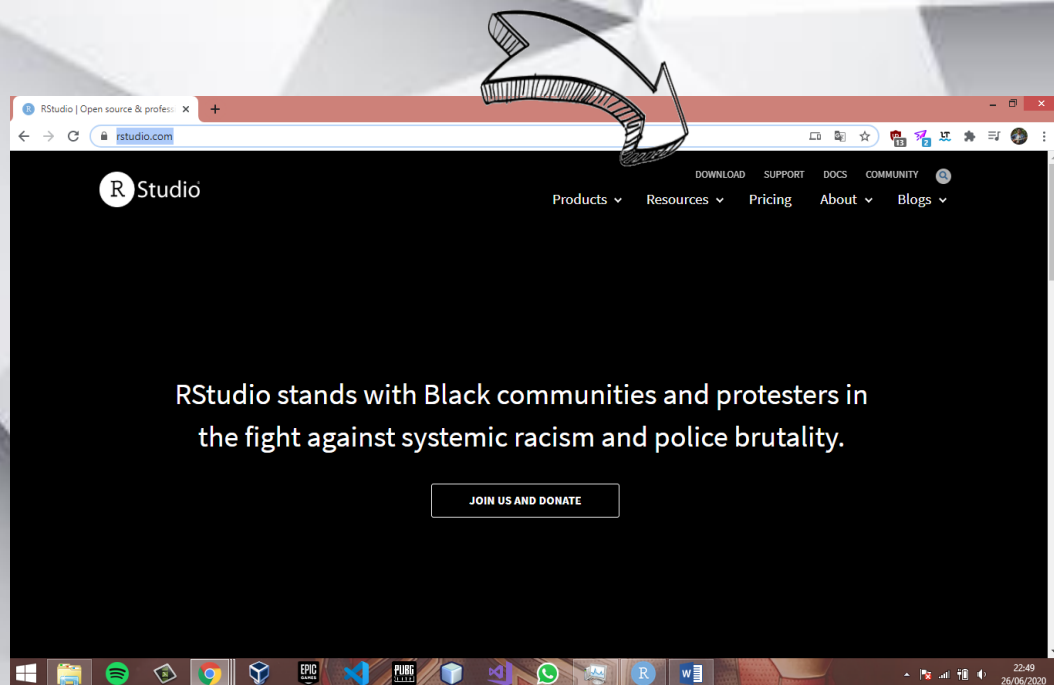
- 4) Se descargará el paquete y esperamos.



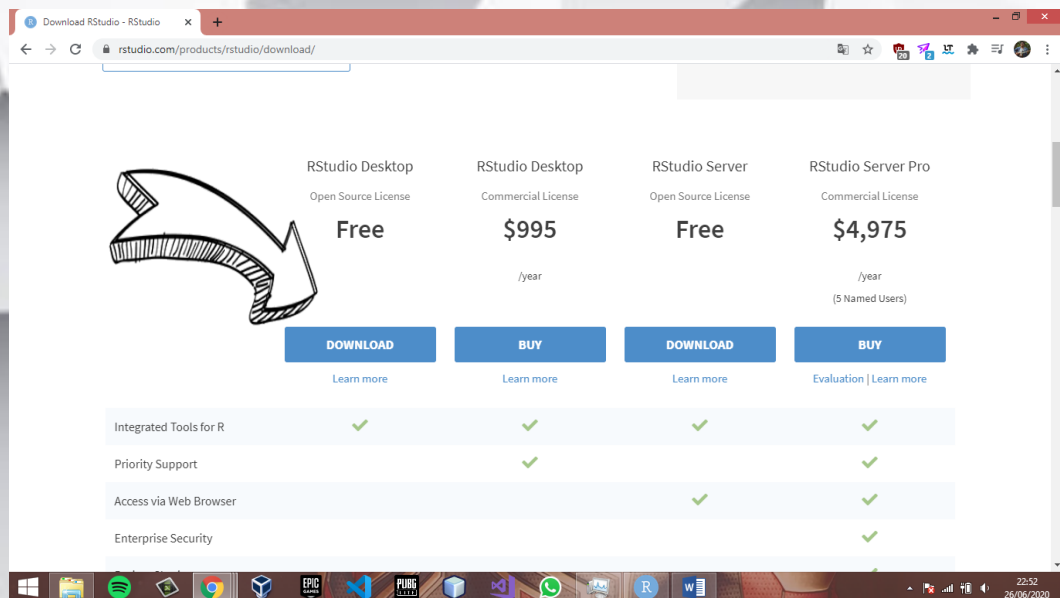
- 5) Para instalar en nuestro sistema operativo el lenguaje R, ejecutamos el archivo .exe que se descargó en el paso anterior, y como la mayoría de software en Windows damos siguiente, a cada uno de las opciones que se nos brindan.



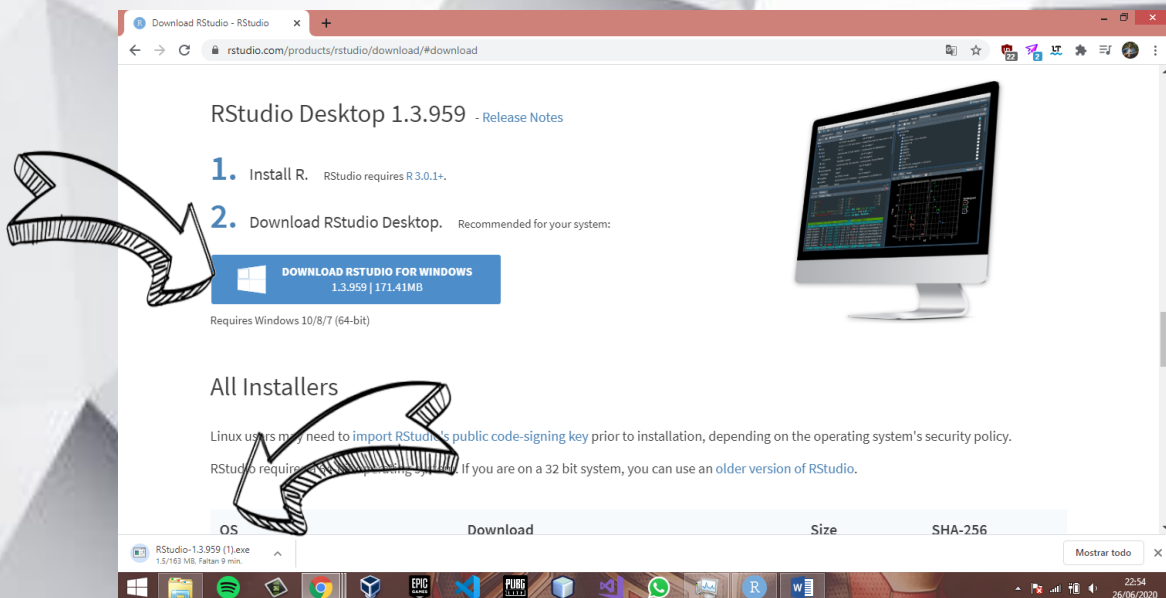
- 6) Para instalar Rstudio primero nos tenemos que dirigir <https://rstudio.com/> y seleccionamos la pestaña DOWNLOAD



7) Dentro de la nueva pestaña seleccionaremos la opción Free.

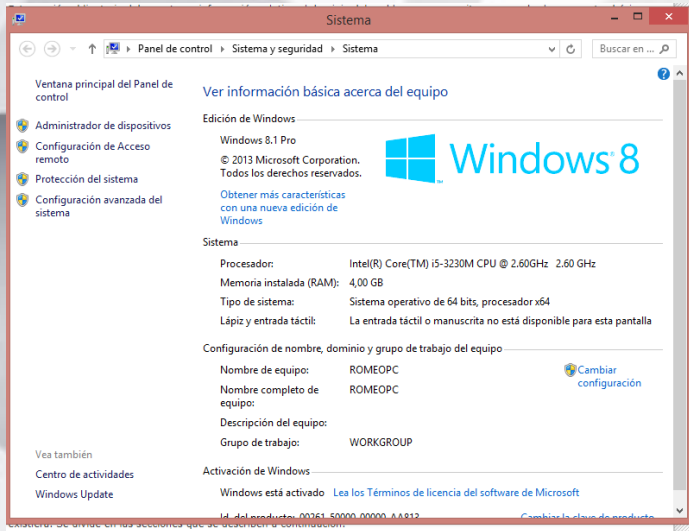


8) Descargamos la versión 1.3.959



9) Al finalizar la descarga, ejecutamos el .exe, y le damos siguiente a cada uno de las cuestiones que se nos solicita el software.

REQUISITOS DEL SISTEMA



Nuestro análisis de bigdata por medio R se realizado en un ordenador marca HP14 con un sistema operativo Windows 8.1 Pro, este ordenador cuenta con una memoria RAM de 4GB y un procesador Intel® Core™ i5 de 2.60 GHz, con un tipo de sistema de 64 bits. Estas características son las mínimas que se recomiendan para la implementación del proyecto de análisis de bigdata de la empresa. Características del software:

Fácil instalación y Mantenimiento:

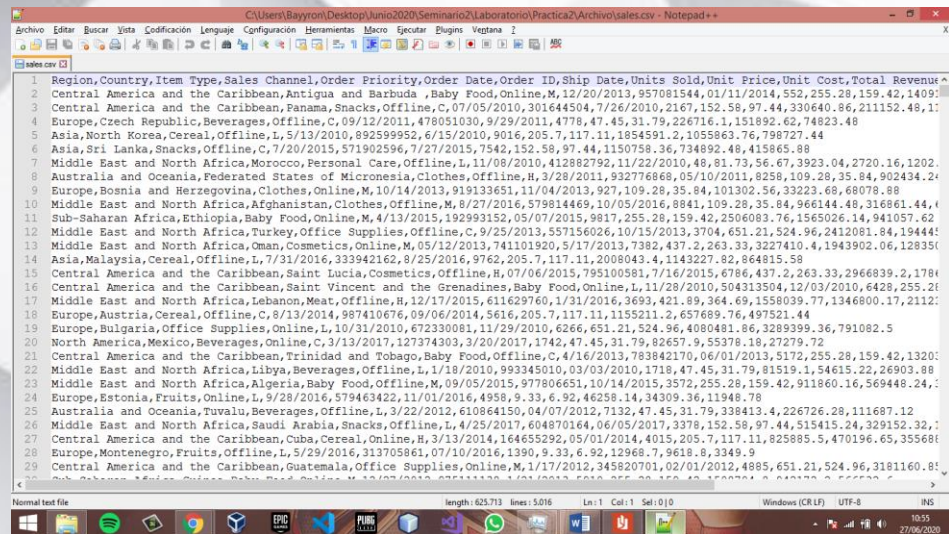
- Completamente basado en R
- Acceso inmediato
- No requiere instalación o mantenimiento continuo
- Apoyo y entrenamiento al usuario
- Guía de Inicio rápido para los usuarios
- Compatible con cualquier plataforma
- Funciona en Windows, Mac OS y Linux con un navegador web

Requerimientos mínimos de hardware

- 2 GHz procesador de 64 bits
- 4 GB de memoria RAM
- 25 GB de Disco duro.

TRANSFORMACIONES Y ACCIONES

1. Ejercicio No. 1 El primer ejercicio de este proyecto consiste en el análisis de un archivo título Sales, dicho archivo contiene la información de las ventas realizadas de una empresa, dichas ventas fueron realizadas a diferentes regiones, diferentes fechas, y con gran diversidad de productos, dichos productos son descritos por su nombre, categoría precio de venta, costo entre otros atributos.



Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue
Central America and the Caribbean	Antigua and Barbuda	Baby Food	Online	M	12/20/2013	957081544	01/11/2014	552,255.28	159.42	1409.1	1409.1
Central America and the Caribbean	Panama	Snacks	Offline	C	07/05/2010	301644504	7/26/2010	2167,152.58	97.44	330640.86	211152.48
Europe	Czech Republic	Beverages	Offline	C	09/12/2011	478051030	9/29/2011	4778,47.45	31.79	226716.1	151892.62
Asia	North Korea	Cereal	Offline	L	5/13/2010	892599952	6/15/2010	6056,205.7	117.11	1854591.2	1055863.76
Asia	Sri Lanka	Snacks	Offline	C	7/19/2015	571902846	11/22/2015	44,1150758.36	734692.48	59665.88	799727.44
Middle East and North Africa	Morocco	Personal Care	Offline	L	11/08/2010	412882792	11/22/2010	48,81.73	56.67	3923.04	2720.16
Australia and Oceania	Federated States of Micronesia	Clothes	Offline	H	3/28/2011	932776868	05/10/2011	8258,109.28	35.84	902434.24	161202.16
Europe	Bosnia and Herzegovina	Clothes	Online	M	10/14/2013	919133651	11/04/2013	927,109.28	35.84	101302.56	33223.68
Middle East and North Africa	Afghanistan	Clothes	Offline	M	8/27/2016	579814469	10/05/2016	8841,109.28	35.84	966144.48	316861.44
Sub-Saharan Africa	Ethiopia	Baby Food	Online	M	4/13/2015	192993152	05/07/2015	9817,255.28	159.42	2506083.76	1565026.14
Middle East and North Africa	Turkey	Office Supplies	Offline	C	9/25/2013	557156026	10/15/2013	3704,651.21	524.96	2412081.84	19444.8
Middle East and North Africa	Oman	Cosmetics	Online	M	05/12/2013	741101920	5/17/2013	7382,437.2	263.33	3227410.4	1943902.06
Asia	Malaysia	Cereal	Offline	L	7/31/2016	333942162	8/25/2016	9762,205.7	117.11	2008043.4	1143227.82
Central America and the Caribbean	Saint Lucia	Cosmetics	Offline	H	07/06/2015	795100581	7/16/2015	6786,437.2	263.33	2966839.2	1784.8
Central America and the Caribbean	Saint Vincent and the Grenadines	Baby Food	Online	L	11/28/2010	504313504	12/03/2010	6428,255.28	159.42	101302.56	33223.68
Middle East and North Africa	Lebanon	Meat	Offline	H	12/17/2015	611629760	1/31/2016	3693,421.89	364.69	1550399.77	1346800.17
Europe	Austria	Cereal	Offline	C	8/13/2014	987410676	09/06/2014	5616,205.7	117.11	1155211.2	657689.76
Europe	Bulgaria	Office Supplies	Online	L	10/31/2010	672330081	11/29/2010	6266,651.21	524.96	4080481.86	3289399.36
North America	Mexico	Beverages	Online	C	3/13/2017	127374303	3/20/2017	1742,47.45	31.79	82657.9	55378.19
Central America and the Caribbean	Trinidad and Tobago	Baby Food	Offline	C	4/16/2013	783842170	06/01/2013	5172,255.28	159.42	1320.8	1320.8
Middle East and North Africa	Libya	Beverages	Offline	L	1/18/2010	993345010	03/03/2010	1718,47.45	31.79	81519.1	54615.22
Middle East and North Africa	Algeria	Baby Food	Offline	M	09/05/2015	977806651	10/14/2015	3572,255.28	159.42	911860.16	569448.24
Europe	Estonia	Fruits	Online	L	9/28/2016	579463422	11/01/2016	4958,9.33	6.92	46258.14	34309.36
Australia and Oceania	Tuvalu	Beverages	Offline	L	3/22/2012	610864150	04/07/2012	7132,47.45	31.79	338413.4	226726.28
Middle East and North Africa	Saudi Arabia	Snacks	Offline	L	4/25/2017	604870164	06/05/2017	3378,152.58	97.44	515415.24	329152.32
Central America and the Caribbean	Cuba	Cereal	Online	H	3/13/2014	164655292	05/01/2014	4015,205.7	117.11	825885.5	470196.65
Europe	Montenegro	Fruits	Offline	L	5/28/2016	313705861	07/10/2016	1390,9.33	6.92	12968.7	9618.8
Central America and the Caribbean	Guatemala	Office Supplies	Online	M	1/17/2012	345820701	02/01/2012	4885,651.21	524.96	311160.81	161202.16

Para este ejercicio se realizaron tres reportes.

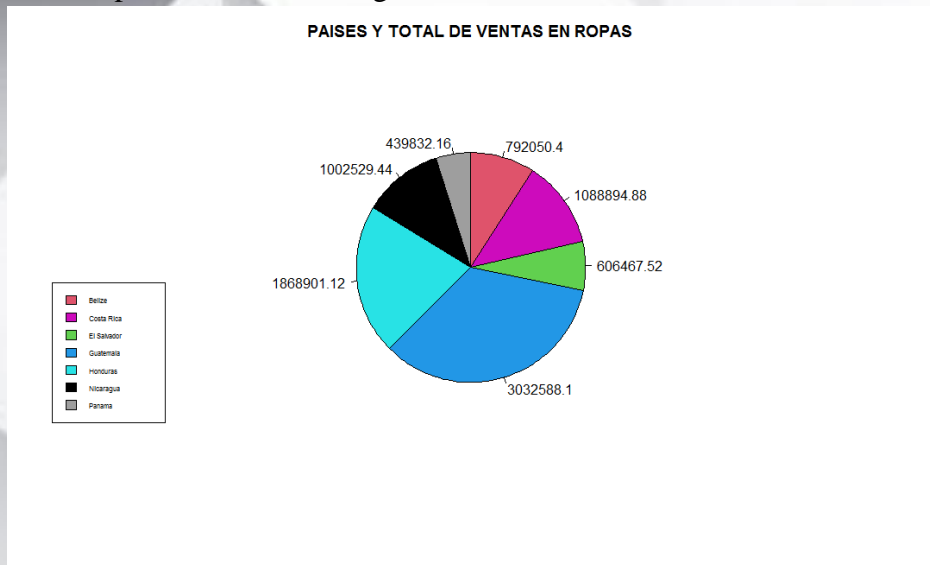
- Primer Reporte: El primer reporte es una gráfica de pie que muestre el total de ganancias de los países de Centroamérica a partir de las ventas de ropa. Para este primer reporte se utilizó el siguiente código.

```
#install.packages("readr") <- se importa desde la consola
#file.choose() <- sirve para obtener las rutas
print("Romeo Axpuc")
library(ggplot2)
#Guardamos el path del primer archivo
archivo <- "C:\\Users\\Bayron\\Desktop\\Junio2020\\Seminario2\\Laboratorio\\Practica2\\Archivo\\sales.csv"

informacion <- read.csv(archivo)
#head(informacion)
columnasReporte1 <- c("Region", "Country", "Item.Type", "Total.Profit")
datos1 <- informacion[columnasReporte1]
#datos1
datosCentroAmerica <- datos1[datos1$Region == "Central America and the Caribbean",]
datosCentroAmerica <- datosCentroAmerica[datosCentroAmerica$Item.Type == "Clothes",]
columnasReporte1 <- c("Country", "Total.Profit")
datosCentroAmerica <- datosCentroAmerica[columnasReporte1]
datosCentroAmerica <- aggregate(datosCentroAmerica$Total.Profit, by=list(Paises=datosCentroAmerica$Country), FUN=sum)
datosCentroAmerica <- datosCentroAmerica[ datosCentroAmerica$Paises == "Guatemala" | datosCentroAmerica$Paises == "Honduras" |
datosCentroAmerica
Ejex <- datosCentroAmerica["Paises"]
Ejey <- datosCentroAmerica["x"]
pie(datosCentroAmerica$x, labels = datosCentroAmerica$x, clockwise = TRUE, main = "PAISES Y TOTAL DE VENTAS EN ROPAS", col = da
legend("bottomleft", datosCentroAmerica$Paises, cex = 0.5, fill = datosCentroAmerica$x)
```

Para la funcionalidad de código anterior debemos instalar el paquete readr y ggplot2, creamos una variable llamada archivo la cual contendrá la ubicación del archivo que vamos a utilizar, por medio de la función read.csv guardamos

los datos del archivo en una variable titulada datos1, esta variable se tomara como nuestro dataframe, creamos una nueva variable llamada datoscentroamerica en esta variable guardamos la información de datos1 filtrada, los filtros a utilizar fueron que la región sea centro america y del caribe, el segundo filtro es que los productos sean de tipo “Clothes”, obtenemos la suma de las ganancias de cada uno de los países, creamos un nuevo dataframe con los atributos que nos interesan, en este caso Country y Total de ventas, luego aplicamos un nuevo filtro el que permitirá únicamente países de Centro America, separamos los datos para la realización de una gráfica de pie, obteniendo el siguiente resultado.



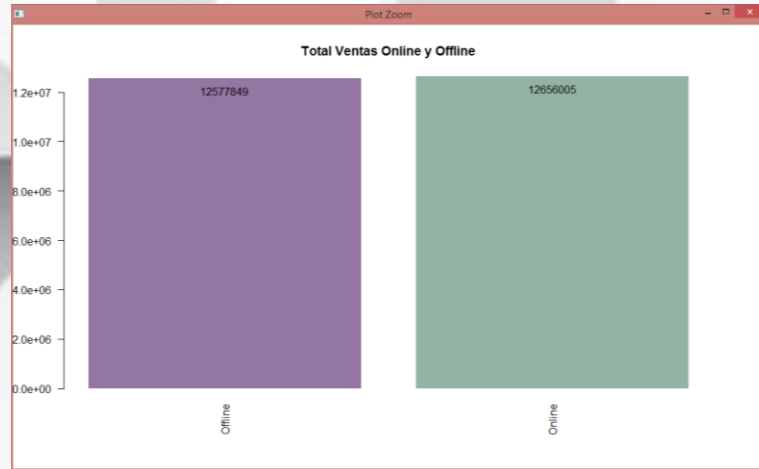
- Segundo Reporte: El segundo reporte consiste en mostrar por medio de una gráfica de barras, que compare el número de artículos de las ventas en línea contra las ventas que no son en línea. Para este primer reporte se utilizó el siguiente código

```
informacion <- read.csv(archivo)
columnasReporte1 <- c("Sales.Channel", "Units.Sold")
datos1 <- informacion[columnasReporte1]
datos1 <- cbind(datos1, numero = datos1$Units.Sold)
datos1 <- aggregate(datos1$numero, by=list(Sales.Channel=datos1$Sales.Channel), FUN=sum)
#head(datos1, n = 101)
# Data
data <- data.frame(
  name = datos1$Sales.Channel,
  average = datos1$x,
  number = datos1$x
)

# Increase bottom margin
par(mar=c(6,4,4,4))
# Basic Barplot
my_bar <- barplot(data$average, border=F, names.arg=data$name,
  las=2,
  col=c(rgb(0.3,0.1,0.4,0.6), rgb(0.3,0.5,0.4,0.6), rgb(0.3,0.9,0.4,0.6), rgb(0.3,0.9,0.4,0.6)),
  main="Total Ventas Online y Offline")
text(my_bar, data$average+0.4, paste(" ", data$number, sep=""), cex=1)
```

Para la elaboración del reporte número dos, primero almacenamos la información del archivo utilizado en la variable información, seguidamente

determinamos las columnas a utilizar de este archivo, los cuales son Sales Channel y Unit Sold, canales de ventas y unidades vendidas, luego por medio de una función de agregación sumamos el total de unidades vendidas por el medio offline y online, con los datos obtenidos procedemos a graficar, dando como resultado:



- Tercer Reporte: El tercer reporte consiste en mostrar por medio de una gráfica el año con más órdenes de prioridad M. Para este primer reporte se utilizó el siguiente código:

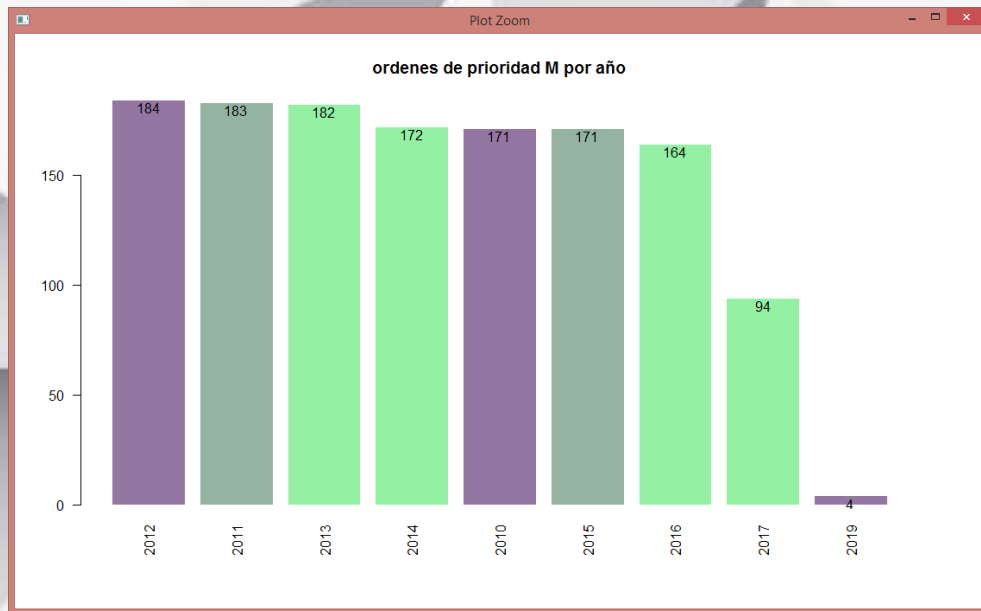
```
informacion <- read.csv(archivo)
columnasReporte1 <- c("Order.Priority","Order.Date")
datos1 <- informacion[columnasReporte1]
datos1 <- cbind(datos1,anio= format(as.Date(datos1$Order.Date, "%m/%d/%Y"),"%Y"),numero = 1)
datos1[datos1$Order.Priority == 'M',]
datos1 <- datos1[c("anio","numero")]
datos1 <- aggregate(datos1$numero, by=list(Anios=datos1$anio), FUN=sum)
datos1 <- datos1[with(datos1,order(-datos1$x)),]

# Data
data <- data.frame(
  name = datos1$Anios,
  average = datos1$x,
  number = datos1$x
)

# Increase bottom margin
par(mar=c(6,4,4,4))
# Basic Barplot
my_bar <- barplot(data$average , border=F, names.arg=data$name ,
  las=2 ,
  col=c( rgb(0.3,0.1,0.4,0.6) , rgb(0.3,0.5,0.4,0.6) , rgb(0.3,0.9,0.4,0.6) , rgb(0.3,0.9,0.4,0.6) ) ,
  main="ordenes de prioridad M por año" )

text(my_bar, data$average+0.4 , paste("\n", data$number, sep="") ,cex=1)
```

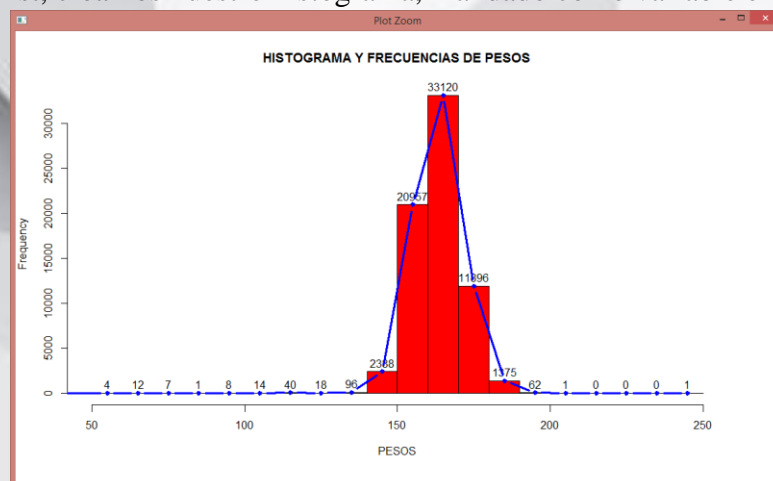
Para este reporte primero guardamos la información del archivo Sales, a la variable información, determinamos que columnas utilizaremos para el reporte, en esta oportunidad utilizaremos Order.Priority y Order Date, luego por de cbind crearemos un nuevo dataframe para obtener el año de cada una de las ventas, utilizaremos únicamente la columna anios y número (1), luego sumaremos por anios las ordenes realizadas y graficamos, obteniendo el siguiente reporte.



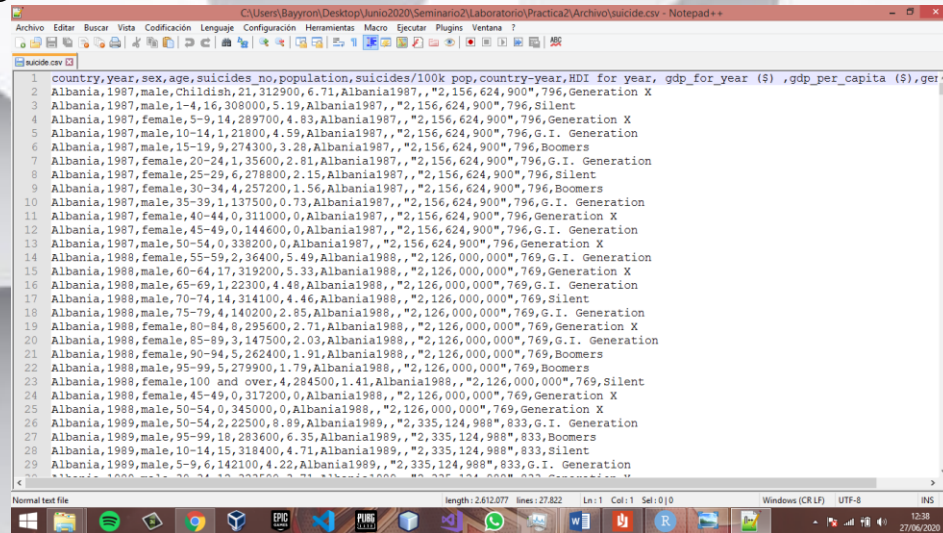
2. Ejercicio No. 2. Este ejercicio únicamente se trabajó un reporte, dicho reporte consiste muestra un histograma unido con un polígono de frecuencias sobre el peso de un estudio realizado. El código utilizado para este reporte fue el siguiente:

```
1 print("Romeo Axpuc")
2 library(ggplot2)
3 #Guardamos el path del primer archivo
4 archivo <- "C:\\Users\\Bayron\\Desktop\\Junio2020\\Seminario2\\Laboratorio\\Practica2\\Archivo\\cardio_train.csv"
5 informacion <- read.csv(archivo)
6 #head(informacion)
7 columnasReporte2 <- c("id","height")
8 datos <- informacion[columnasReporte2]
9
10 h= hist(datos$height, main = "HISTOGRAMA Y FRECUENCIAS DE PESOS", col="red", xlab="PESOS", labels = T)
11 lines(c(0,h$mids),c(0,h$counts), type = "b", pch = 20, col = "blue", lwd = 3)
```

Para la realización de este reporte primero debemos importar la librería ggplot2, luego por medio de la variable archivo guardaremos la ubicación del archivo, por medio de la función read.csv obtenemos la información del archivo y hacemos mención que únicamente utilizaremos las columnas “id” y “height” de nuestro archivo y luego de la función hist, creamos nuestro histograma, mandado como variable el peso.



3. Ejercicio No. 3. Para este enunciado se trabajó con el archivo suicide, el cual tiene la siguiente estructura.



Para el primer reporte de este archivo se solicita una tabla de frecuencias de los rangos de los suicidios en Guatemala (todas las edades), por lo cual se utilizó el siguiente código.

```
#Guardamos el path del primer archivo
archivo <- "C:\\Users\\Bayyron\\Desktop\\Junio2020\\Seminario2\\Laboratorio\\Practica2\\Archivo\\suicide.csv"
datos1 <- read.csv(archivo)
datos1 <- datos1[c("country", "age", "suicides_no")]
datos1 <- subset.data.frame(datos1, datos1$country == "Guatemala" & datos1$suicides_no != "NA")
datos1 <- datos1[c("age", "suicides_no")]
edades <- datos1$age
edades <- replace(edades, edades == "1-4", "01-04")
edades <- replace(edades, edades == "5-9", "05-09")
edades <- replace(edades, edades == "100 and over", "99+")
valores <- datos1$suicides_no

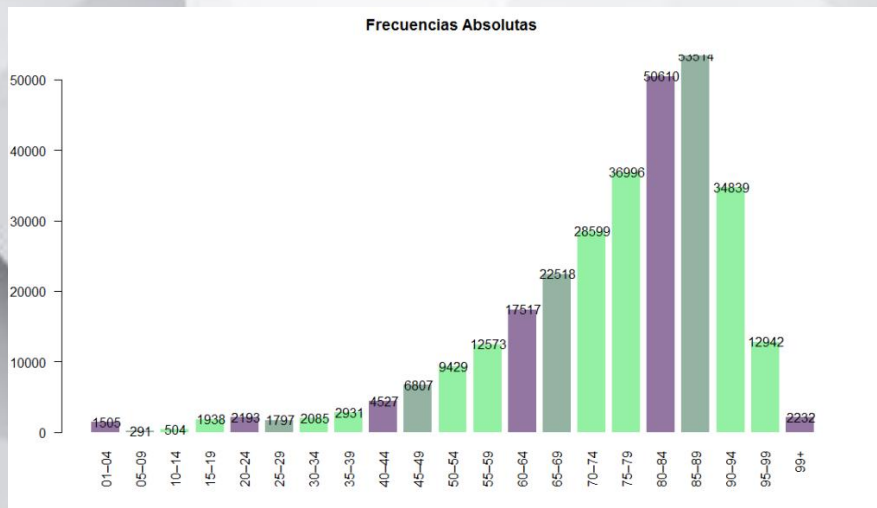
tablafinal <- data.frame(edades, valores)
tablafinal <- aggregate(x=tablafinal$valores, by=list(edades=tablafinal$edades), FUN=sum)
tablafinal <- tablafinal[order(tablafinal$edades, decreasing = FALSE),]
tablafinal
datostabla <- tablafinal$edades
frecuencias <- tablafinal$x
frecuenciaAcumulada <- cumsum(unname(tablafinal$x))
#Frecuencia relativa
fi <- round(prop.table(frecuencias), 5)
#Frecuencia relativa acumulada
Fi <- round(cumsum(prop.table(frecuencias)), 5)
reportefinal <- data.frame(datostabla, frecuencias, frecuenciaAcumulada, fi, Fi)
tg <- tableGrob(reportefinal)
grid.draw(tg)
```

Primero guardamos la dirección de nuestro archivo en una variable, obtenemos la información del archivo por medio de la función `read.csv`, luego seleccionaremos las columnas "country", "age", "suicides_no", luego utilizamos un filtro, dicho filtro nos obtendrá únicamente los datos de Guatemala, de este resultado únicamente, nos quedamos con las edades y el número de suicidios, luego verificamos que el número de suicidios sea un número válido, por medio de una agregación obtenemos la suma de casos por edades en Guatemala y graficamos la tabla.

	datostabla	frecuencias	frecuenciaAcumulada	fi	Fi
1	01-04	1505	1505	0.00491	0.00491
2	05-09	291	1796	0.00095	0.00586
3	10-14	504	2300	0.00165	0.00751
4	15-19	1938	4238	0.00633	0.01383
5	20-24	2193	6431	0.00716	0.02099
6	25-29	1797	8228	0.00587	0.02686
7	30-34	2085	10313	0.00681	0.03366
8	35-39	2931	13244	0.00957	0.04323
9	40-44	4527	17771	0.01478	0.05801
10	45-49	6807	24578	0.02222	0.08023
11	50-54	9429	34007	0.03078	0.11101
12	55-59	12573	46580	0.04104	0.15205
13	60-64	17517	64097	0.05718	0.20923
14	65-69	22518	86615	0.07350	0.28273
15	70-74	28599	115214	0.09335	0.37609
16	75-79	36996	152210	0.12077	0.49685
17	80-84	50610	202820	0.16520	0.66206
18	85-89	53514	256334	0.17468	0.83674
19	90-94	34839	291173	0.11372	0.95047
20	95-99	12942	304115	0.04225	0.99271
21	99+	2232	306347	0.00729	1.00000

Para el segundo reporte graficaremos la frecuencia absoluta, ya que tenemos los datos de la tabla anterior únicamente graficaremos un sistema de barras con las edades y las frecuencias absolutas, a continuación se da a conocer el código y la gráfica resultante.

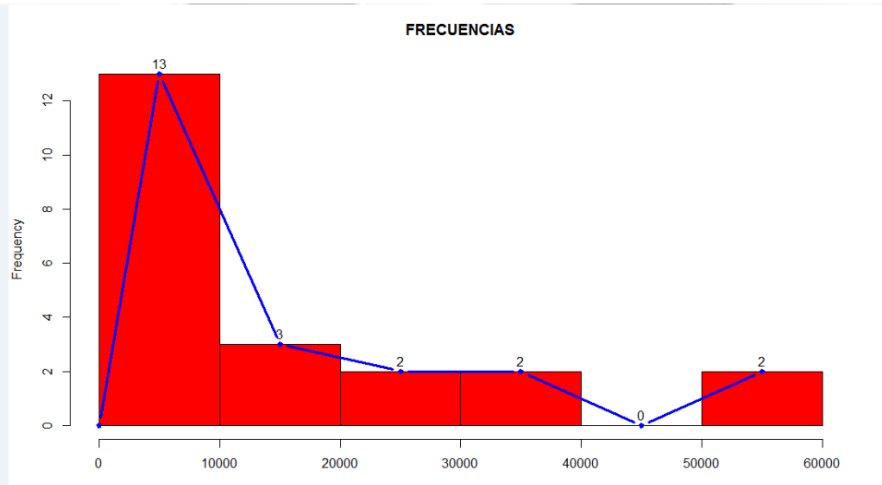
```
# Data
data <- data.frame(
  name = reportefinal$datostabla ,
  average = reportefinal$frecuencias,
  number = reportefinal$frecuencias
)
par(mar=c(6,4,4,4))
my_bar <- barplot(data$average , border=F , names.arg=data$name ,
  las=2 ,
  col=c(rgb(0.3,0.1,0.4,0.6) , rgb(0.3,0.5,0.4,0.6) , rgb(0.3,0.9,0.4,0.6) , rgb(0.3,0.9,0.4,0.6) ) ,
  main="Frecuencias Absolutas" )
text(my_bar, data$average+0.4 , paste(" ", data$number, sep="") , cex=1)
```



Para el tercer reporte graficaremos la frecuencia, ya que tenemos los datos de la tabla anterior únicamente graficaremos la información por medio de un polígono de

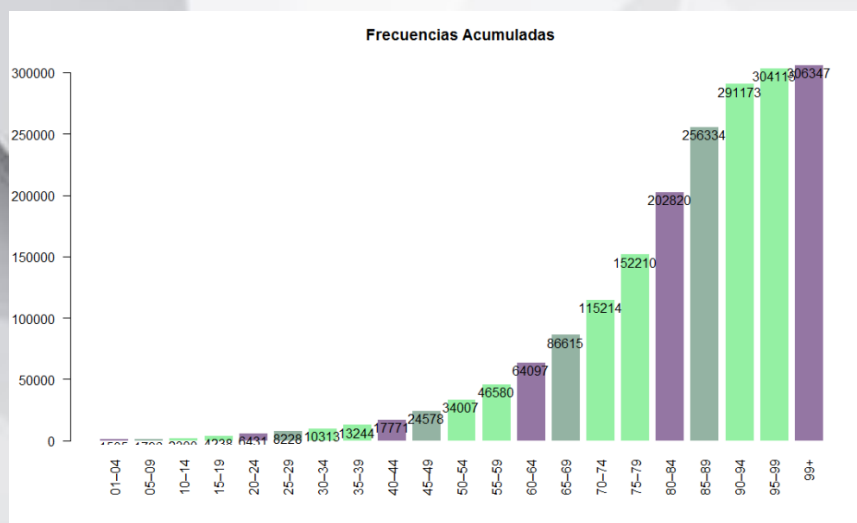
frecuencias los datos correspondientes, a continuación se da a conocer el código y la gráfica resultante.

```
h= hist(reportefinal$frecuencias, main = "FRECUENCIAS", col="red", xlab="FRECUENCIAS", labels = T)
lines(c(0,h$mids),c(0,h$counts), type = "b", pch = 20, col = "blue", lwd = 3)
```

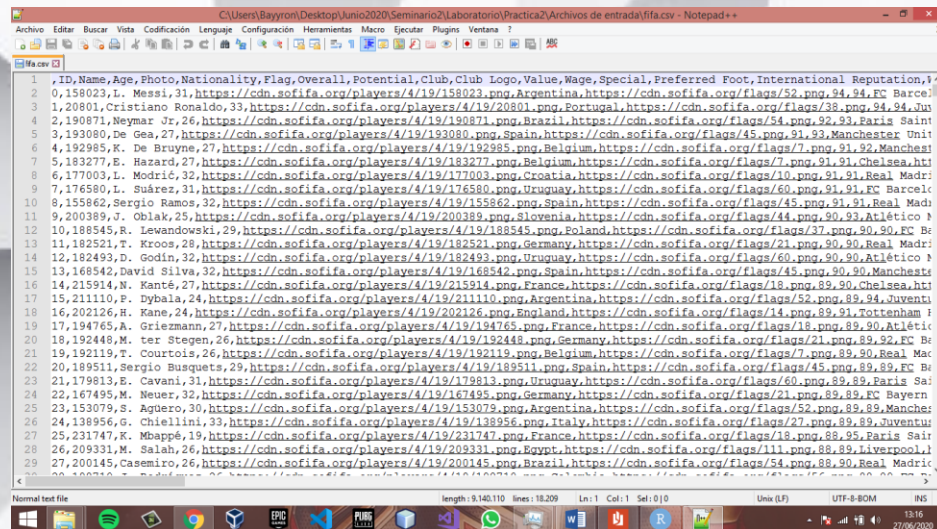


El cuarto reporte es un diagrama el cual muestra el resultado de las frecuencias acumuladas, estos datos fueron obtenidos del primer reporte realizado en este enunciado.

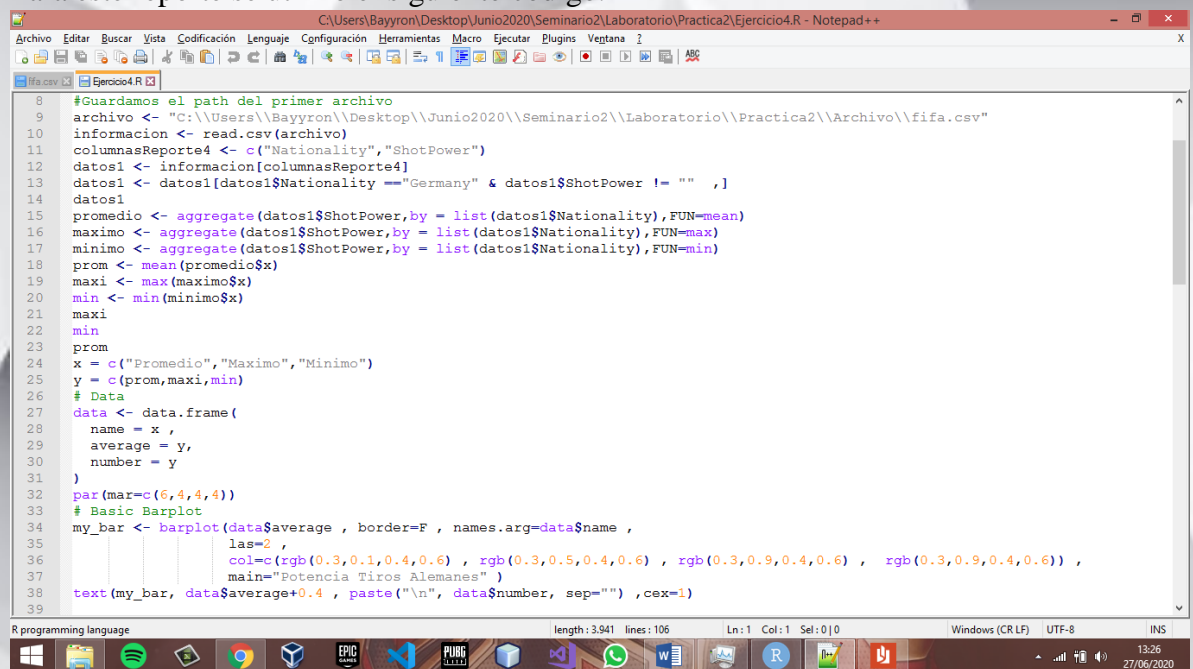
```
# Data
data <- data.frame(
  name = reportefinal$datostabla ,
  average = reportefinal$frecuenciaAcumulada,
  number = reportefinal$frecuenciaAcumulada
)
# Increase bottom margin
par(mar=c(6,4,4,4))
# Basic Barplot
my_bar <- barplot(data$average , border=F, names.arg=data$name ,
  las=2 ,
  col=c(rgb(0.3,0.1,0.4,0.6) , rgb(0.3,0.5,0.4,0.6) , rgb(0.3,0.9,0.4,0.6) , rgb(0.3,0.9,0.4,0.6)) ,
  main="Frecuencias Acumuladas" )
text(my_bar, data$average+0.4 , paste("\n", data$number, sep="") ,cex=1)
```



4. Ejercicio No. 4. Para este ejercicio utilizaremos el archivo Fifa, cuya estructura es el siguiente:



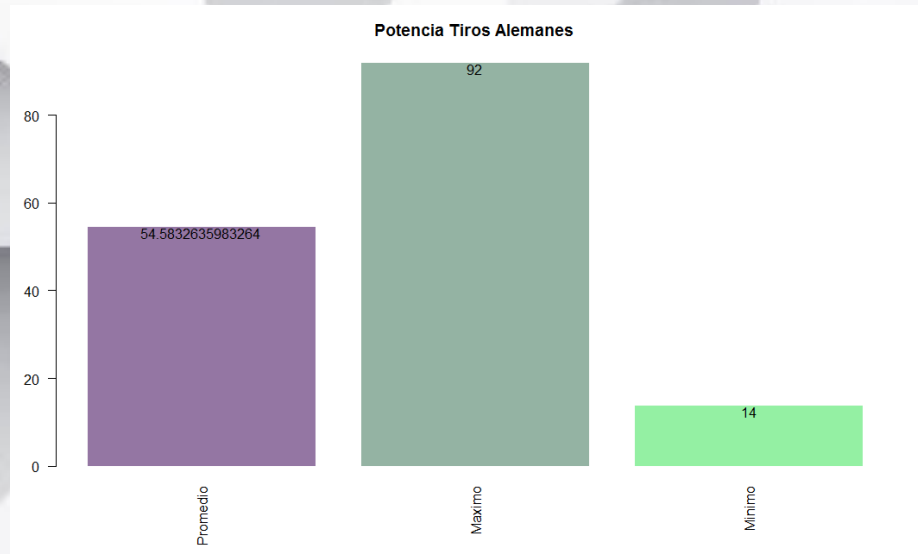
El primer reporte de este enunciado se solicita una gráfica de barras de la potencia de disparo promedio, máximo y mínimo de los jugadores con nacionalidad alemana. Para este reporte se utilizó el siguiente código.



```
8 #Guardamos el path del primer archivo
9 archivo <- "C:\\Users\\Bayron\\Desktop\\Junio2020\\Seminario2\\Laboratorio\\Practica2\\Archivo\\fifa.csv"
10 informacion <- read.csv(archivo)
11 columnasReporte4 <- c("Nationality", "ShotPower")
12 datos1 <- informacion[columnasReporte4]
13 datos1 <- datos1[datos1$Nationality == "Germany" & datos1$ShotPower != "", ]
14 datos1
15 promedio <- aggregate(datos1$ShotPower, by = list(datos1$Nationality), FUN=mean)
16 maximo <- aggregate(datos1$ShotPower, by = list(datos1$Nationality), FUN=max)
17 minimo <- aggregate(datos1$ShotPower, by = list(datos1$Nationality), FUN=min)
18 prom <- mean(promedio$x)
19 maxi <- max(maximo$x)
20 min <- min(minimo$x)
21 maxi
22 min
23 prom
24 x = c("Promedio", "Maximo", "Minimo")
25 y = c(prom, maxi, min)
26 # Data
27 data <- data.frame(
28   name = x,
29   average = y,
30   number = y
31 )
32 par(mar=c(6,4,4,1))
33 # Basic Barplot
34 my_bar <- barplot(data$average, border=F, names.arg=data$name,
35   las=2,
36   col=c(rgb(0.3,0.1,0.4,0.6), rgb(0.3,0.5,0.4,0.6), rgb(0.3,0.9,0.4,0.6), rgb(0.3,0.9,0.4,0.6)),
37   main="Potencia Tiros Alemanes")
38 text(my_bar, data$average+0.4, paste("\n", data$number, sep=""), cex=1)
39
```

Primero guardamos la ubicación de nuestro archivo en una variable y obtenemos la información de este archivo y seleccionamos las columnas “Nationality” y “ShotPower”, filtramos la información y únicamente obtendremos la información de los jugadores de nacionalidad alemana y el dato de la potencia válidos, seguidamente

obtenemos por funciones de agregación el promedio, el máximo y el mínimo, y lo graficamos, obteniendo como resultado el siguiente reporte.

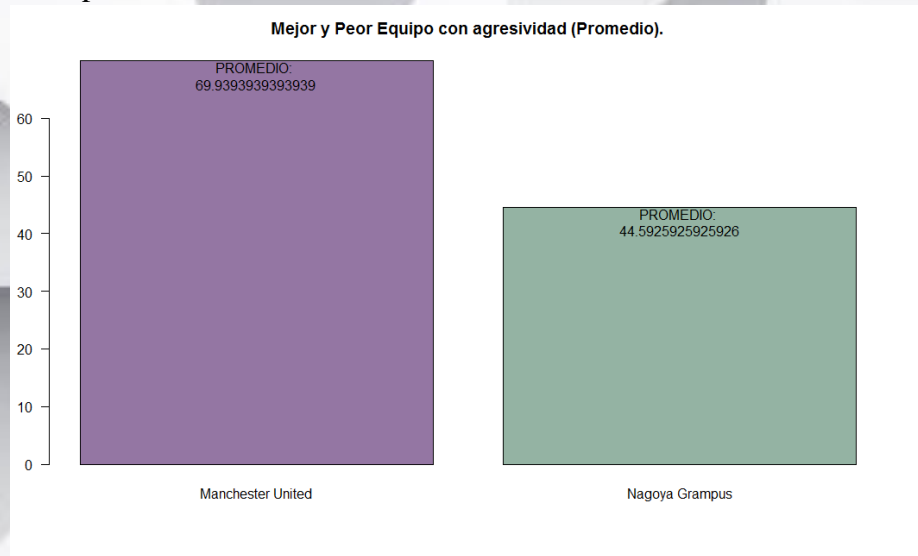


Para el segundo reporte se solicitó una gráfica que muestre al club con la mayor agresividad promedio y al club con la menor agresividad promedio, para ello se utilizó el siguiente código.

```
45 #####
46 columnasReporte4 <- c("Club", "Aggression")
47 datos1 <- informacion[columnasReporte4]
48 datos1[datos1$Club != "" & datos1$Aggression != "", ]
49 datos1 <- aggregate(datos1$Aggression, by = list(datos1$Club), FUN=mean)
50 maximo <- max(datos1$x)
51 posicionMaximo <- which.max(datos1$x)
52 maximoEquipo <- datos1[posicionMaximo,]
53 maximoEquipo
54 minimo <- min(datos1$x)
55 posicionMinima <- which.min(datos1$x)
56 MinimoEquipo <- datos1[posicionMinima,]
57 MinimoEquipo
58 x = c(maximoEquipo$Group.1, MinimoEquipo$Group.1)
59 y = c(maximoEquipo$x, MinimoEquipo$x)
60 # Data
61 data <- data.frame(
62   name = x ,
63   average = y,
64   number = y
65 )
66
67 # Increase bottom margin
68 par(mar=c(6,4,4,4))
69 my_bar <- barplot(data$average , border=T , names.arg=data$name ,
70   las=1 ,
71   col=c(rgb(0.3,0.1,0.4,0.6) , rgb(0.3,0.5,0.4,0.6) , rgb(0.3,0.9,0.4,0.6) , rgb(0.3,0.9,0.4,0.6)) ,
72   main="Mejor y Peor Equipo con agresividad (Promedio).")
73 text(my_bar, data$average+0.3 , paste("\n\nPROMEDIO: \n", data$number, sep="") , cex=1)
```

Primero obtenemos información de la variable información, esta variable contiene los datos del archivo leído con prioridad, luego seleccionaremos las columnas a utilizar en esta oportunidad son “Club” y “Aggression”, luego filtramos los datos, el filtro permitirá utilizar datos que si tengan información relevante, seguidamente por medio de una función de agregación obtenemos el promedio (mean) de la agresividad de cada club, luego obtendremos el máximo y mínimo dato de estos clubes, luego por medio de la función which min y wich max, obtendremos las posiciones de estos

clubes en las tablas y así poder obtener el nombre, procedemos a graficar los dos datos correspondientes.

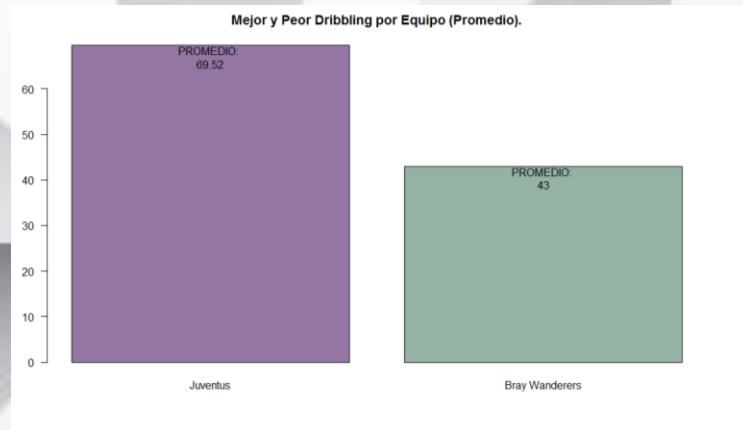


Para el tercer reporte se solicitó una gráfica que muestre al club con el mayor regate promedio y al club con el menor regate promedio., para ello se utilizó el siguiente código.

```
columnasReporte4 <- c("Club", "Dribbling")
datos1 <- informacion[columnasReporte4]
datos1 <- datos1[datos1$Club != "" & datos1$Dribbling != "", ]
datos1 <- aggregate(datos1$Dribbling, by = list(datos1$Club), FUN=mean)
maximo <- max(datos1$x)
posicionMaximo <- which.max(datos1$x)
maximoEquipo <- datos1[posicionMaximo,]
maximoEquipo
minimo <- min(datos1$x)
posicionMinima <- which.min(datos1$x)
MinimoEquipo <- datos1[posicionMinima,]
MinimoEquipo
x = c(maximoEquipo$Group.1, MinimoEquipo$Group.1)
y = c(maximoEquipo$x, MinimoEquipo$x)
# Data
data <- data.frame(
  name = x,
  average = y,
  number = y
)
par(mar=c(6,4,4,4))
my_bar <- barplot(data$average, border=T, names.arg=data$name,
  las=1,
  col=c(rgb(0.3,0.1,0.4,0.6), rgb(0.3,0.5,0.4,0.6), rgb(0.3,0.9,0.4,0.6), rgb(0.3,0.9,0.4,0.6)),
  main="Mejor y Peor Dribbling por Equipo (Promedio).")
text(my_bar, data$average+0.3, paste("\n\nPROMEDIO: \n", data$number, sep=""), cex=1)
```

Primero obtenemos información de la variable información, esta variable contiene los datos del archivo leído con prioridad, luego seleccionaremos las columnas a utilizar en esta oportunidad son “Club” y “Dribbling”, luego filtramos los datos, el filtro permitirá utilizar datos que si tengan información relevante, seguidamente por medio de una función de agregación obtenemos el promedio (mean) de la agresividad de cada club, luego obtendremos el máximo y mínimo dato de estos clubes, luego por medio de la función which min y wich max, obtendremos las posiciones de estos

clubes en las tablas y así poder obtener el nombre, procedemos a graficar los dos datos correspondientes.



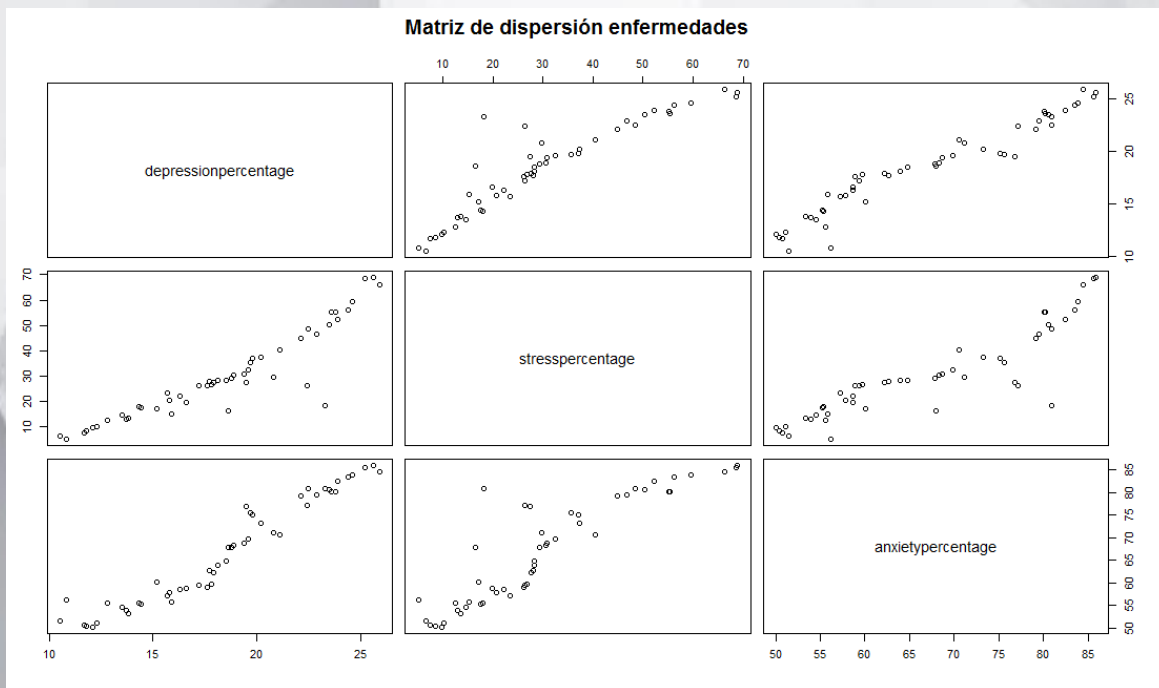
5. Ejercicio No. 5. Para este ejercicio se utilizará el archivo regresión, por medio de este archivo analizaremos la relación entre tres variables, depresión, estrés y ansiedad, el código para estos reportes son los siguientes.

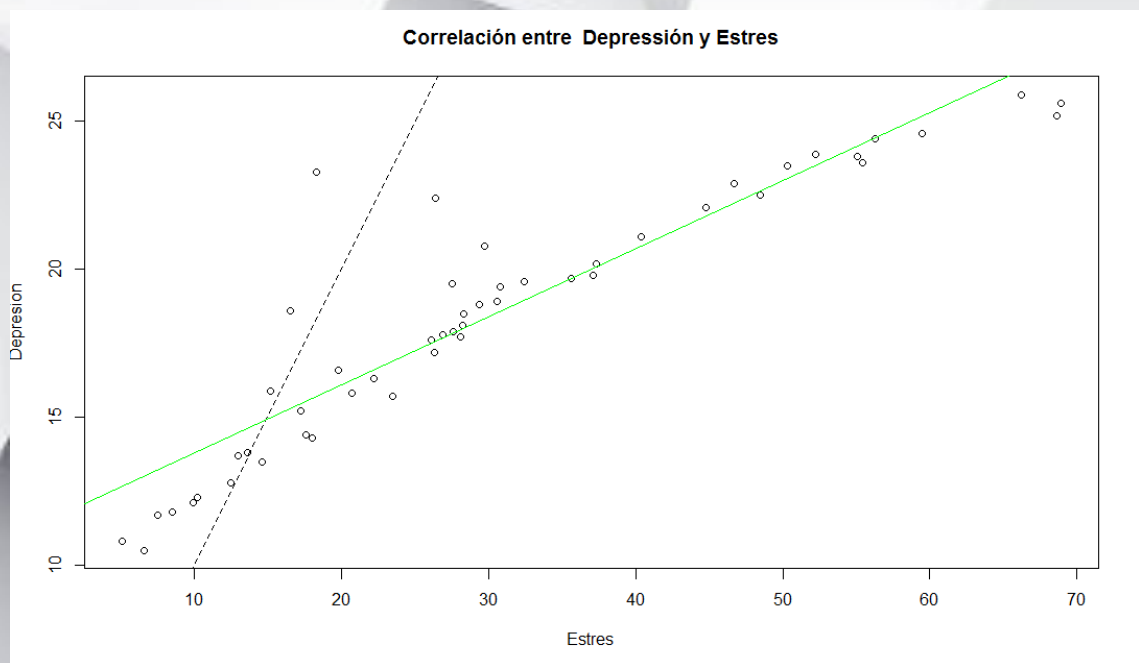
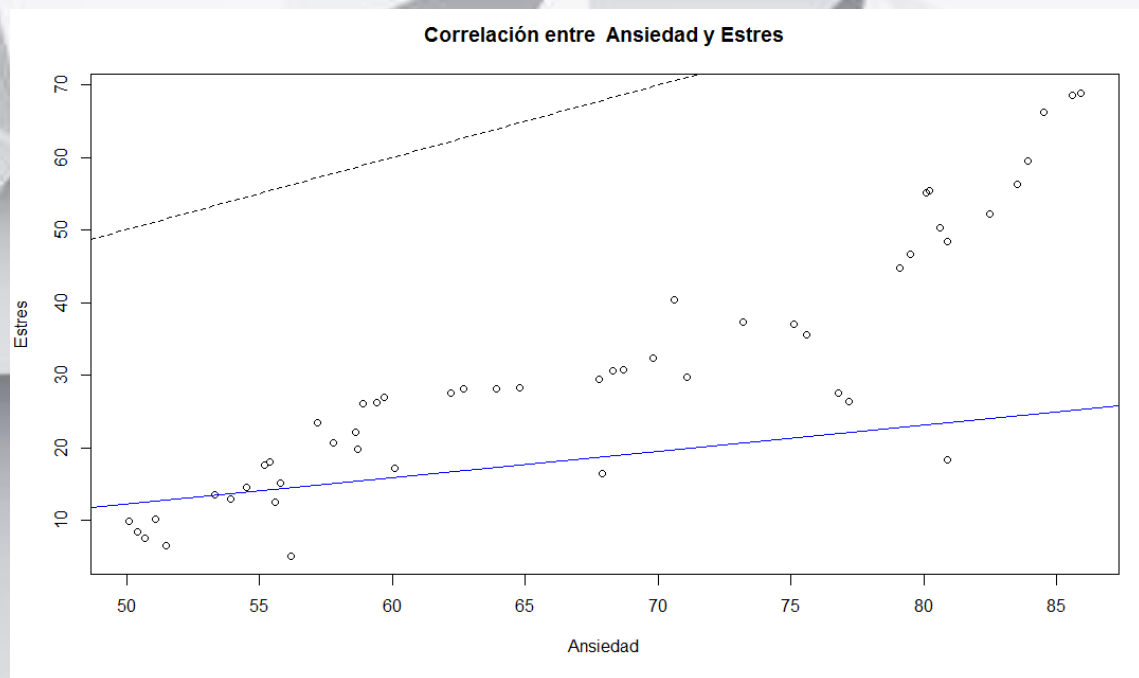
```
#Guardamos el path del primer archivo
archivo <- "C:\\Users\\Bayyron\\Desktop\\Junio2020\\Seminario2\\Laboratorio\\Practica2\\Archivo\\regresion.csv"
informacion <- read.csv(archivo)
informacion
pairs(depressionpercentage ~ stresspercentage + anxietypercentage, data=informacion, main="Matriz de dispersión enfermedades")

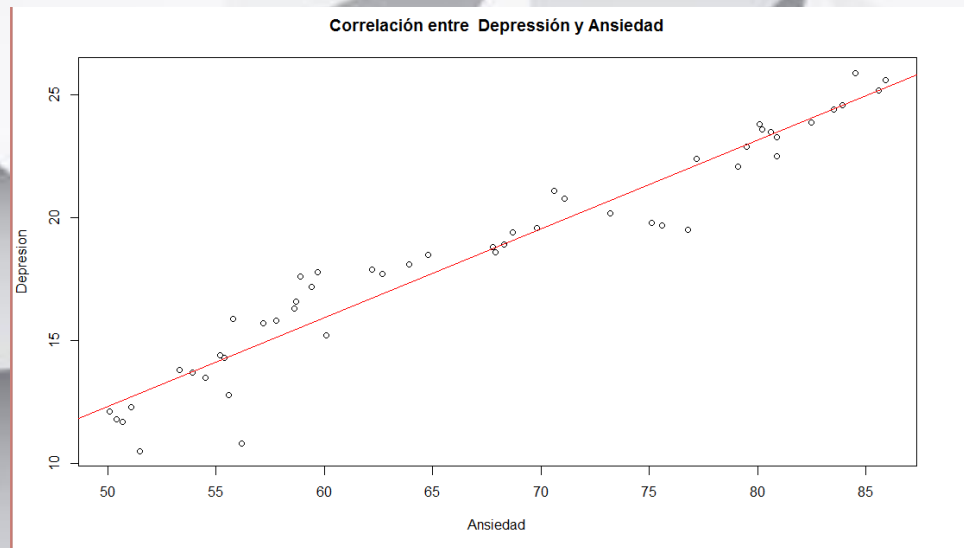
plot(depressionpercentage ~ stresspercentage, data = informacion, xlab = "Estrés", ylab = "Depresión", main="Correlación entre
abline(a = 0, b = 1, lty = 2)
MLatin <- lm(depressionpercentage ~ stresspercentage, data = informacion)
abline(MLatin, col = "green")

plot(depressionpercentage ~ anxietypercentage, data = informacion, xlab = "Ansiedad", ylab = "Depresión", main="Correlación ent
abline(a = 0, b = 1, lty = 2)
MLatin <- lm(depressionpercentage ~ anxietypercentage, data = informacion)
abline(MLatin, col = "red")

plot(stresspercentage ~ anxietypercentage, data = informacion, xlab = "Ansiedad", ylab = "Estrés", main="Correlación entre Ans
abline(a = 0, b = 1, lty = 2)
MLatin <- lm(stresspercentage ~ anxietypercentage, data = informacion)
```







Análisis: analizando la data obtenida, no podemos decir a que estudio está relacionado, es decir que no existe una variable global a examinar, es decir, con que enfermedad o situación se relacionan el estrés, ansiedad y depresión, sin embargo independientemente de esto, podemos analizar cómo interactúan estas tres enfermedades entre sí, primero analicemos la gráfica entre depresión y estrés, podemos decir que están relacionados, que al momento de sufrir depresión puede sufrir estrés y viceversa, lo mismo sucede entre la depresión y ansiedad, al igual que la ansiedad y estrés. Por lo tanto podemos concluir que existe una gran relación entre las tres enfermedades.