# Equipo 2

# Integrantes

1. Nancy Mejía Dominguez
2. Norma Angélica Zúñiga Pacheco
3. Margarita Palacios A.
4. Israel Sandoval Grajeda
5. Alejandro Núñez Vilchis
6. Luis Josue Díaz Alvarez
7. Cristina Pérez Martínez
8. Vanesa Domínguez Esquivel

**SCRIPT 1**

**#Descripción de la base (Pendiente)**

Estuve viendo un poco los datos y veo que son estadísticas de equipos de béisbol. Aunque conozco lo básico del juego, la información no me hace tanto sentido pero creo que de inicio está totalmente transpuesta, es decir, las variables están en cada renglón y las observaciones son las columnas.

Comenten si alguien nota algo diferente.

Es una base de datos relacionada a juegos de baseball, que tiene las siguientes variables:

• Equipo (Team)

• Liga (League)

• Año (Year)

• Carreras anotadas (RS)

• Ejecuciones permitidas (RA)

• Victorias (W)

• Porcentaje en base (OBP)

• Porcentaje de slugging (SLG)

• Promedio de bateo (BA)

• Playoffs (Playoffs)

• Clasificación en temporada (RankSeason)

• Clasificación en los Playoffs (RankPlayoffs)

• Juegos jugados (G)

• Porcentaje base del oponente (OOBP)

• Porcentaje de slugging del oponente (OSLG)

**# 1.-Cargar la base de datos**

*#Cargar librerias*

library(tidyr)

library(dplyr)

library(readr)

**# Importación de datos:**

datos<-read.csv2("baseball\_raw.csv", header=T)

View(datos)

datos

**# 2.- Ordenar la base de datos**

**# Ordenar y transformar**

*#Transformar y ordenar base*

*Este sería el código más simple*

datos\_ordenados<-datos%>%

pivot\_longer(X1:X1232,

names\_to="Observaciones",

values\_to="Valor") %>%

pivot\_wider(names\_from=X, values\_from=Valor)%>%

select(Observaciones:OSLG)

View(datos\_ordenados)

También se podría utilizar el “group by” dentro del código, aunque en este caso no es necesario porque ya están agrupados por default

***#Generar archivo con base ordenada***

write\_csv(datos\_ordenados, "baseball ordenada.csv")

**# 2.1 Elimine las columnas ‘Rango en temporada’ y ‘Clasificación en los Playoffs’**

#Eliminación de las columnas “RankSeason” y “RankPlayoffs”

sin\_columnas <- datos\_ordenados%>%

select(!RankSeason & !RankPlayoffs)

View (sin\_columnas)

**# 3 . Elimine las observaciones que contienen valores faltantes.**

sin\_na <- sin\_columnas%>%

drop\_na()

View(sin\_na)

**#4. Utilice la variable de carreras anotadas (RS) para detectar y eliminar los**

**outliers de la base.**

Con lo siguiente:

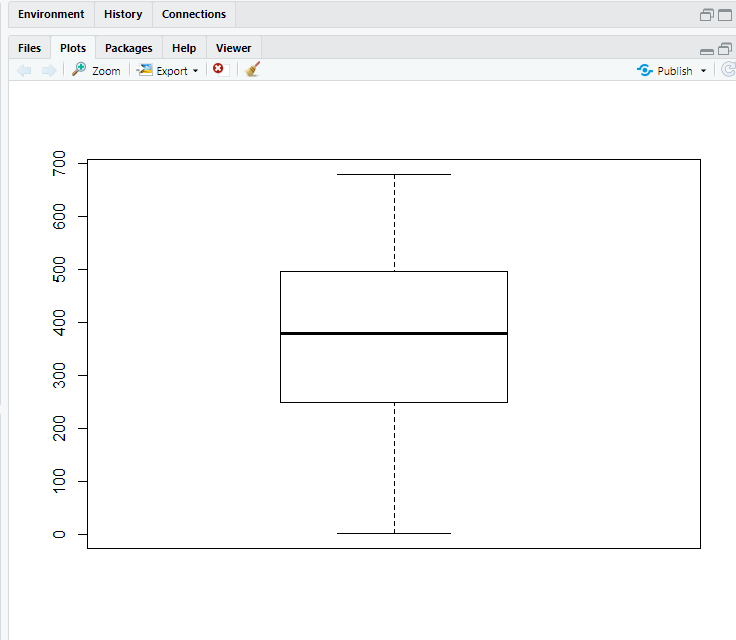
class(sin\_na$RS)

nume <- as.numeric(sin\_na$RS)#Tiene que ser numerico

boxplot(nume)

#Me sale así:

#Pero no observo outliers, qué me faltó?



**MP:**

#Yo intenté de esta manera:

boxplot(sin\_na$RS)

#pero me sale el siguiente error:

Error in x[floor(d)] + x[ceiling(d)] :

non-numeric argument to binary operator

#Lo resolví de esta forma

#Como lo hicieron antes, había que convertir a numerico, lo gracioso es que tuve que quitar los NA antes y después de convertir porque me salia error.

outliers <- boxplot(sin\_na$RS)$out

datos\_finales <- filter(sin\_na, !RS %in% c(outliers))

#Al preguntar qué tipo de variable es “Carreras anotadas (RS)”, me dice que es de tipo character, no numérica.

> class(sin\_na$RS)

[1] "character"

#Entonces, hay que convertirla en variable numérica.

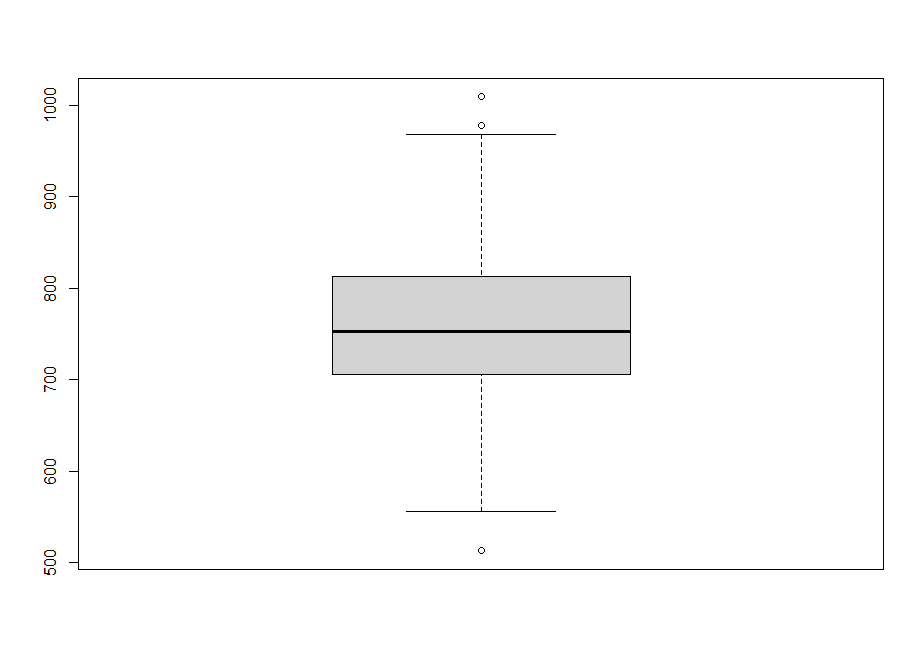
#Genero una nueva variable llamada “RS\_num” que va a ser numérica:

RS\_num <- as.numeric (sin\_na$RS)

#Y ahora sí, ploteo esta variable:

boxplot(RS\_num)

#Sale lo siguiente:



#Es congruente con lo que indica el resumen de la variable:

> summary (RS\_num)

Min. 1st Qu. Median Mean 3rd Qu. Max.

513.0 706.0 752.5 759.5 813.0 1009.0

#Pregunto cuáles son los outliers:

> boxplot(RS\_num) $out

[1] 513 978 1009

# Elimine las observaciones que contengan valores faltantes

final<-sin\_na %>%

filter(513 < RS & RS < 978)

Considerando el código anterior, para poder eliminar los outliers primero es necesario tener la variable numérica, la sugerencia de código es la siguiente

final<-sin\_na%>%

mutate(RS\_numerica=as.numeric(RS)%>%

filter(RS\_numerica>513,RS\_numerica<978)

#Conviene cambiar algunas variables a tipo numéricas para hacer análisis descriptivo, excepto Playoffs, Team y League. Estas podrían ser factores

sin\_na[c(4:10,12:14)] <- sin\_na[c(4:10,12:14)] %>% mutate(across(where(is.character),as.numeric))

sin\_na[c(2,3,11)] <- sin\_na[c(2,3,11)] %>% mutate(across(where(is.character),as.factor))

str(sin\_na)

**#5. Guarde la base limpia y ordenada en un nuevo archivo.**

write\_csv(sin\_na, "Beisbol\_final.csv")

**#6. Análisis exploratorio.Guía:**

**A) ¿Qué tipo de variables contiene la base?**

La base contiene variables cualitativas y cuantitativas.Las variables “Team” y “League” son cualitativas, mientras que las variables “RS”, “RA”, “W”, “OBP”, “SLG”, “BA”, “Playoffs”, “G”, “OOBP” y “OSLG” son cuantitativas.

**B) ¿Cuántos equipos hay por liga?**

**C) ¿Existe relación entre el equipo y el número de victorias por año?**

**D) ¿Hay relación entre las ejecuciones permitidas y la liga?**

**E) ¿La distribución del promedio de bateo cambia de acuerdo a la liga?.**

**F) ¿El porcentaje base del oponente está relacionado con las victorias?.**

**G) ¿La distribución del promedio de bateo es distinta para cada liga?**

#Exploración numérica

str(datos1)

summary(datos1)

#Exploración gráfica

plot(datos1)

**SCRIPT 2**

**library(ggplot2)**

**library(dplyr)**

**#1.- Lea la base de datos ordenada que guardó en el ejercicio 5.**

**datos1 <- read.csv("C:/Users/CRISTINA/Documents/cursito/baseball ordenada.csv", header=T)**

**View(datos1)**

**#2. A ajuste un modelo de regresión lineal que ajuste las carreras anotadas para**

**los registros antes del 2010:**

**A) Obtenga la base que contenga solamente a los registros antes del**

**2010.**

**base\_para\_regL <- datos1%>%**

**filter(Year<2010)**

**B) Use un modelo lineal que ajuste las carreras anotadas. Elija entre las**

**variables ‘porcentaje en base’, ‘porcentaje de slugging’ y ‘año’.**

**(Puede ser una o mas variables)**

**#Correlacion entre las variables**

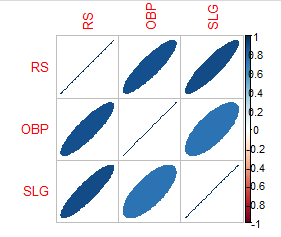
**correlaciones <- base\_para\_regL %>%**

**select( RS,OBP,SLG) %>%**

**cor( )**

**library(corrplot)**

**corrplot( correlaciones, method = "ellipse" )**

****

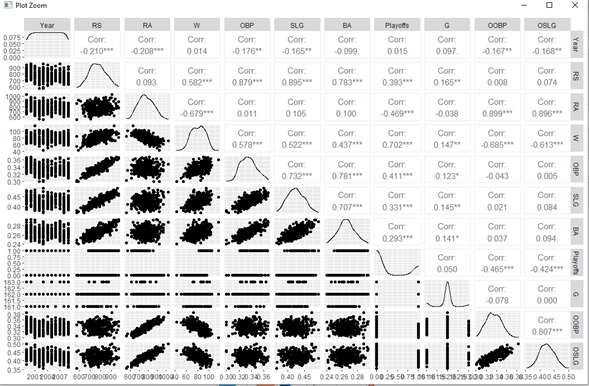
**library(GGally)**

**correlacion\_pairs <- base\_para\_regL %>%**

**select\_if(is.numeric) %>%**

**ggpairs( )**

**correlacion\_pairs**

****

**#Variable dependiente: carreras anotadas**

**#variable independiente: porcentaje en base**

**#Carreras anotadas en funcion del porcentaje en slugging**

**modelo1 <- lm(RS~SLG, base\_para\_regL)**

**modelo1**

**#Call:**

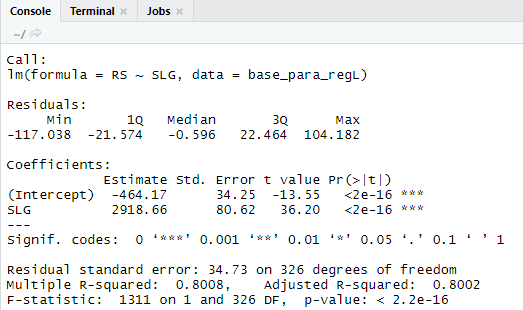
**#lm(formula = RS ~ SLG, data = base\_para\_regL)**

**#Coefficients:**

**#(Intercept) SLG**

**# -464.2 2918.7**

**summary(modelo1)**

****

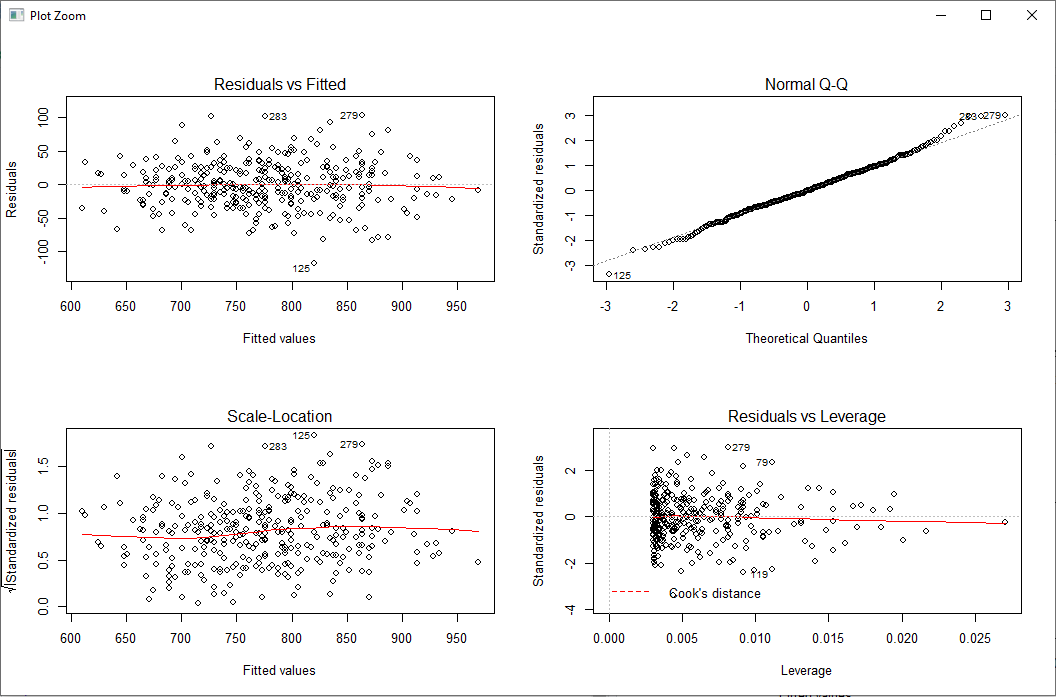
**C) Evalúe los supuestos principales del modelo que obtuvo en el inciso**

**anterior.**

**#############SUPUESTOS############################**

**par(mfrow = c(2,2))**

**plot(modelo1)**

****

**#Gráfico: "Residuals vs Fitted"**

**#Los puntos se ven dispersos,tienen independencia,(SUPUESTO 6)**

**#es decir, errores independientes de los predictores**

**#Gráfico: Normal Q-Q"Standarized Residuals Vs Theoretical Quantiles" ](debe de ser una línea recta), (supuesto 8)**

**#Normalidad: errores normales con varianza constante**

**#Gráfico: Scale- Location:Homocedasticidad, linea horizontal, (supuesto 7)**

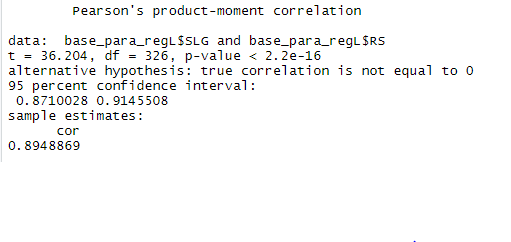
**#Gráfico: "Residuals vs Leverage":para los outliers, en la base no hay**

**### PRINCIPALES SUPUESTOS**

**####1.Correcta especificación del modelo lineal**

**cor.test(x = base\_para\_regL$SLG ,**

**y = base\_para\_regL$RS, method = "pearson")**

****

**###2.No hay colinealidad perfecta**

**relacion <- ggplot( data=base\_para\_regL , aes( x =SLG, y =RS)) +**

**geom\_point( color = "darkblue" )+**

**geom\_smooth(method = "lm", se = FALSE, color = "darkred")+**

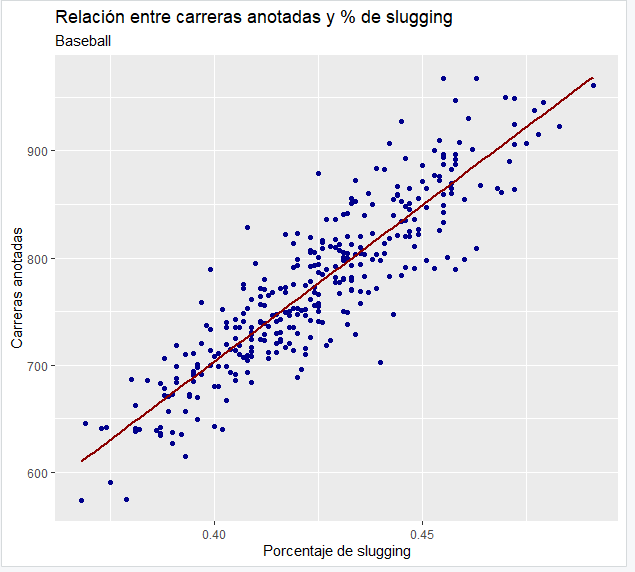
**labs( title = "Relación entre carreras anotadas y % de slugging",**

**subtitle = "Baseball",**

**x = "Porcentaje de slugging",**

**y = "Carreras anotadas")**

**relacion**

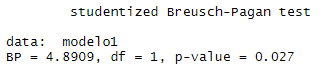
****

**#7. Homocedasticidad:**

**library(lmtest)**

**# H0: Homocedasticidad**

**bptest(modelo1)**

****

**####8. Normalidad de los errores:**

**norm\_errores <- ggplot( modelo1 , aes( x = modelo1$residuals ) )+**

**geom\_histogram( aes( y = ..density.., fill = "deeppink3" ), alpha=0.5, show.legend = FALSE )+**

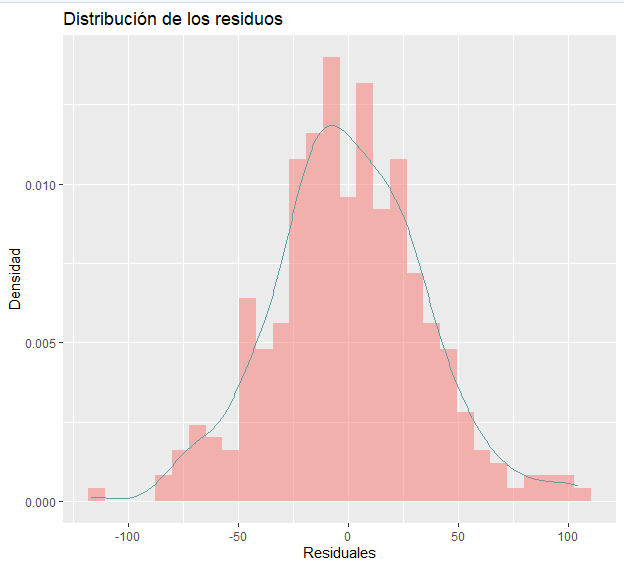
**geom\_density( color = "cadetblue", show.legend = F )+**

**labs( title = "Distribución de los residuos",**

**x = "Residuales",**

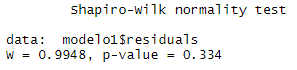
**y = "Densidad" )**

**norm\_errores**

****

**# H0:Normalidad**

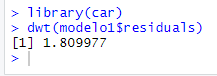
**shapiro.test( modelo1$residuals )**

****

**######9.No correlación entre los errores:existe independencia si la estadística de prueba está entre 1.5 y 2.5**

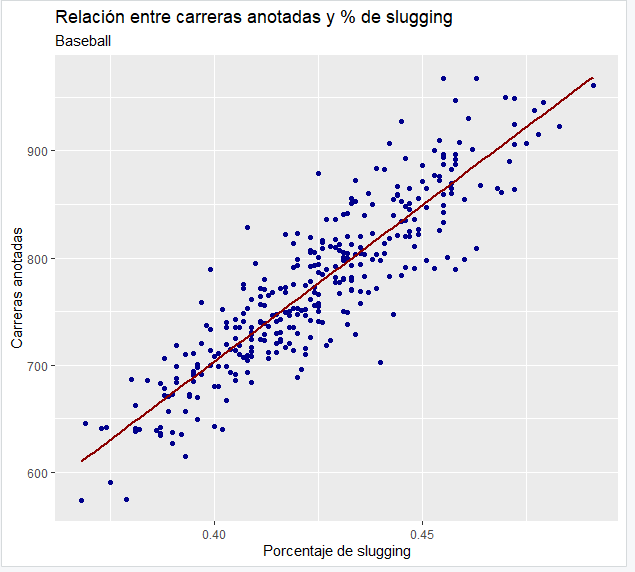
**library(car)**

**dwt(modelo1$residuals)**

****

**D) Grafique la muestra y el plano de regresión que obtuvo. Obs: Si eligió**

**el modelo con 3 variables, seleccione dos de ellas para graficar.**

****

**#3. Elabore un dashboard en Shiny en el que muestre la información más**

**importante que obtuvo en este análisis, además de la gráfica que obtuvo al**

**final del ejercicio anterior.**