Laboratorio # 3, Modelos de regresión lineal

Definir el directorio

```
dir <- "C:/Users/Oscar/Desktop/Galileo/Trimestre2/Econometria en
R/Tareas/laboratorio3"
setwd(dir)
getwd()
```

```
## [1] "C:/Users/Oscar/Desktop/Galileo/Trimestre2/Econometria en
R/Tareas/laboratorio3"
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

```
## Loading required package: lattice
```

Lectura de archivo

```
df <- read.csv("Admisions.csv")
```

Ejercicio #1: utilizando R realice una función que dado un dataframe cualquiera de dos columnas, donde la primera (índice 1) sea el valor de la variable independiente (X) y la segunda sea el valor de una variable dependiente (Y), devuelva una lista con los siguientes elementos: 1) Un arreglo con los valores de los estimadores para $\beta!$ y $\beta"$. 2) El valor del coeficiente de determinación $r\#$ del modelo. 3) El coeficiente de correlación r (raíz cuadrada de $r\#$). 4) Un arreglo con los valores de los residuos. 5) Una gráfica con la nube de puntos y la recta de regresión del modelo.

```r
regresion_lineal <- function(x,y) {
  n <- length(x)  # Número de observaciones
  x_mean <- mean(x)  # Media de x
  y_mean <- mean(y)  # Media de y
  xy_mean <- mean(x * y)  # Media de x*y
  x_squared_mean <- mean(x^2)  # Media de x^2

  # Coeficientes de la regresión lineal
  beta_1 <- (xy_mean - x_mean * y_mean) / (x_squared_mean - x_mean^2)
  beta_0 <- y_mean - beta_1 * x_mean

  # valores ajustados y los residuos
  y_pred <- beta_0 + beta_1 * x
  residual <- y - y_pred

  # Coeficiente de correlación (R-cuadrado)
  SS_total <- sum((y - y_mean)^2)
  SS_residual <- sum(residual^2)
  R_squared <- ( 1 - SS_residual / SS_total)
  R_squared_ <- ( 1 - SS_residual / SS_total)^(1/2)

  # Dataframe para el diagrama de dispersión
  data <- data.frame(x = x, y = y, y_pred = y_pred)

  # Creación del gráfico de dispersión con la línea de la recta
  plot <- ggplot(data, aes(x, y)) +
    geom_point(color = "blue") +
    geom_line(aes(x, y_pred), color = "red") +
    geom_text(x = max(x), y = max(y), label = paste("Beta_0: ",
round(beta_0, 4)), hjust = 2, vjust = 1) +
    geom_text(x = max(x), y = max(y), label = paste("Beta_1: ",
round(beta_1, 4)), hjust = 3.5, vjust = 2.5) +
    geom_text(x = max(x), y = max(y), label = paste("corr: ",
round(R_squared_,4)), hjust = 4.4, vjust = 4 ) +
    labs(x = "Variable X", y = "Variable Y", title = "Regresión Lineal")
+
    theme_minimal()


  # Resultados
  resultados <- list(fitted_values = y_pred,
                     residual = residual,
                     coefficients = c(beta_0, beta_1),
                     r_squared = R_squared,
                     correlation = R_squared_,
                     plot = plot)
  return(resultados)
}
```

```
reg_lin <-  regresion_lineal( df$TOEFL.Score, df$Chance.of.Admit)
```

fitted values

```
head(reg_lin$fitted_values,25)
```

```
##  [1] 0.9204454 0.7182101 0.6630550 0.7733652 0.6446700 0.8652903
0.7549801
##  [8] 0.6078999 0.6262849 0.7365951 0.6998250 0.7917502 0.8101352
0.7549801
## [15] 0.6630550 0.6814400 0.7182101 0.6998250 0.7733652 0.6262849
0.7182101
## [22] 0.8469053 0.8836754 0.9388304 0.9388304
```

Residuals

```
head(reg_lin$residual,25)
```

```
##  [1] -0.0004454151  0.0417899260  0.0569450190  0.0266348329
0.0053300500
##  [6]  0.0347096779 -0.0049801361  0.0721001120 -0.1262849190 -
0.2865951051
## [11] -0.1798250430  0.0482498019 -0.0301352291 -0.1349801361 -
0.0530549810
## [16] -0.1414400120 -0.0582100740 -0.0498250430 -0.1433651671 -
0.0062849190
## [21] -0.0782100740 -0.1469052911  0.0563246469  0.0111695539
0.0311695539
```

Coeficientes de la regresión Beta_0 y Beta_1

```
print(reg_lin$coefficients)
```

```
## [1] -1.24898824  0.01838503
```

```
print(paste("Coeficiente de determinación", round (reg_lin$r_squared,4))
)
```

```
## [1] "Coeficiente de determinación 0.6276"
```

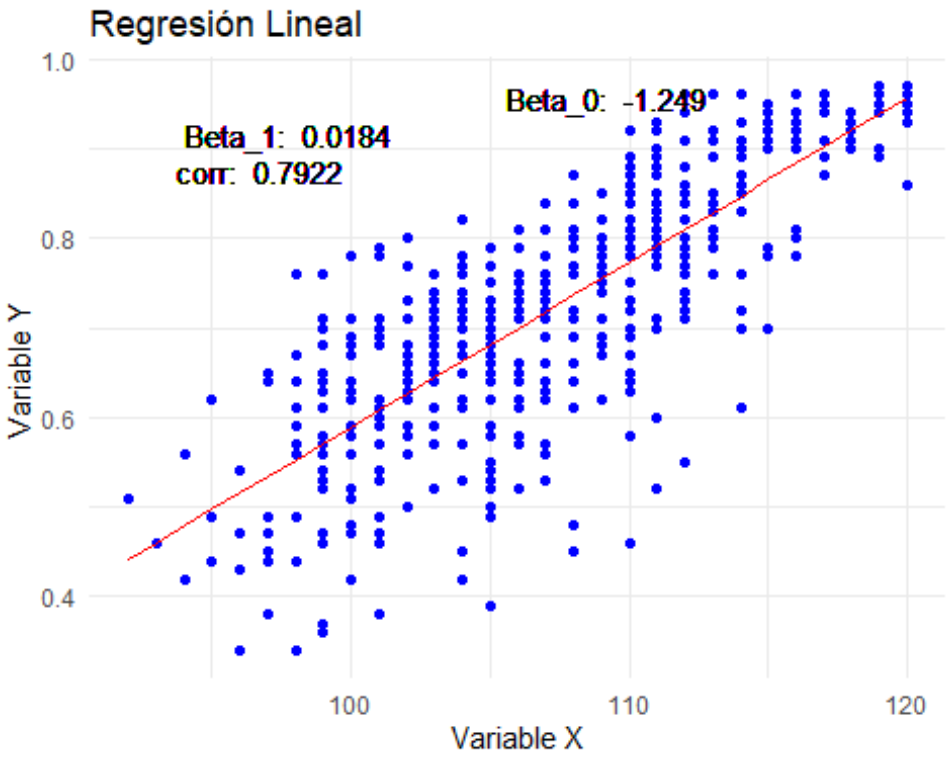
```
print("-------------------------------------------------------------")
```

```
## [1] "-------------------------------------------------------------"
```

```
print(paste("Coeficiente de correlación: ",
round(reg_lin$correlation,4)))
```

```
## [1] "Coeficiente de correlación:  0.7922"
```

```
print(reg_lin$plot)
```



Ejercicio #2: Para este ejercicio se le solicita que desarrolle las siguientes actividades utilizando RStudio Con el dataset Admissions adjunto a este laboratorio realice lo siguiente:

1. Realice un análisis estadístico sobre todas las variables del dataset, recuerde que pude usar la función summary().

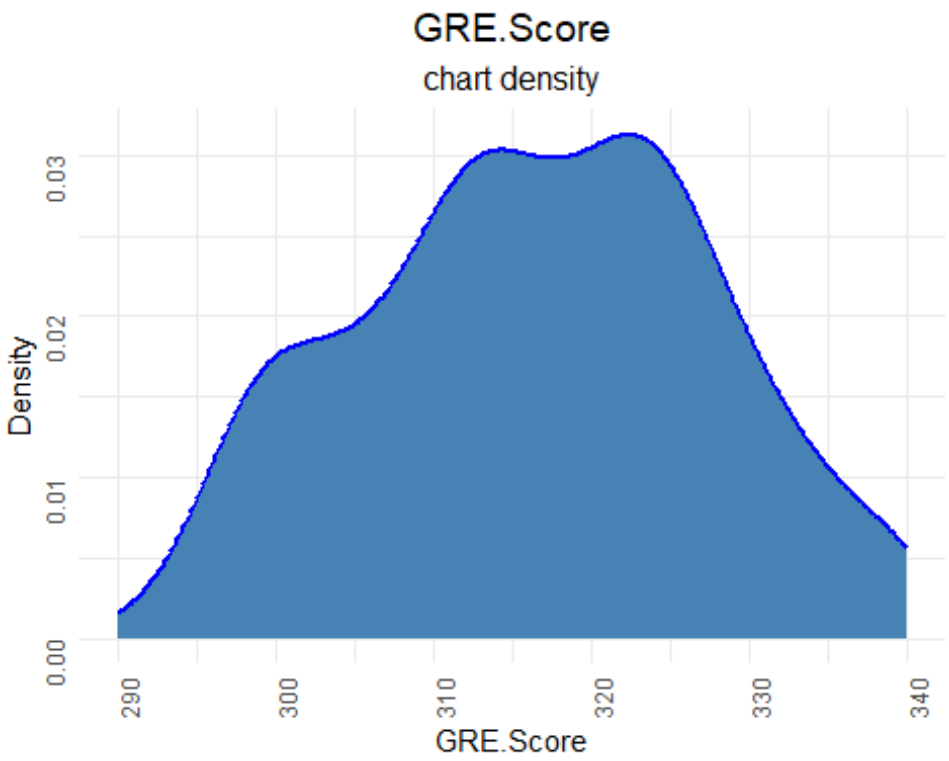
```
summary(df)
```

```
##    Serial.No.      GRE.Score      TOEFL.Score    University.Rating
##  Min.   :  1.0   Min.   :290.0   Min.   : 92.0   Min.   :1.000
##  1st Qu.:125.8   1st Qu.:308.0   1st Qu.:103.0   1st Qu.:2.000
##  Median :250.5   Median :317.0   Median :107.0   Median :3.000
##  Mean   :250.5   Mean   :316.5   Mean   :107.2   Mean   :3.114
##  3rd Qu.:375.2   3rd Qu.:325.0   3rd Qu.:112.0   3rd Qu.:4.000
##  Max.   :500.0   Max.   :340.0   Max.   :120.0   Max.   :5.000
##       SOP             LOR             CGPA          Research
##  Min.   :1.000   Min.   :1.000   Min.   :6.800   Min.   :0.00
##  1st Qu.:2.500   1st Qu.:3.000   1st Qu.:8.127   1st Qu.:0.00
##  Median :3.500   Median :3.500   Median :8.560   Median :1.00
##  Mean   :3.374   Mean   :3.484   Mean   :8.576   Mean   :0.56
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:9.040   3rd Qu.:1.00
##  Max.   :5.000   Max.   :5.000   Max.   :9.920   Max.   :1.00
##  Chance.of.Admit
##  Min.   :0.3400
##  1st Qu.:0.6300
##  Median :0.7200
##  Mean   :0.7217
```

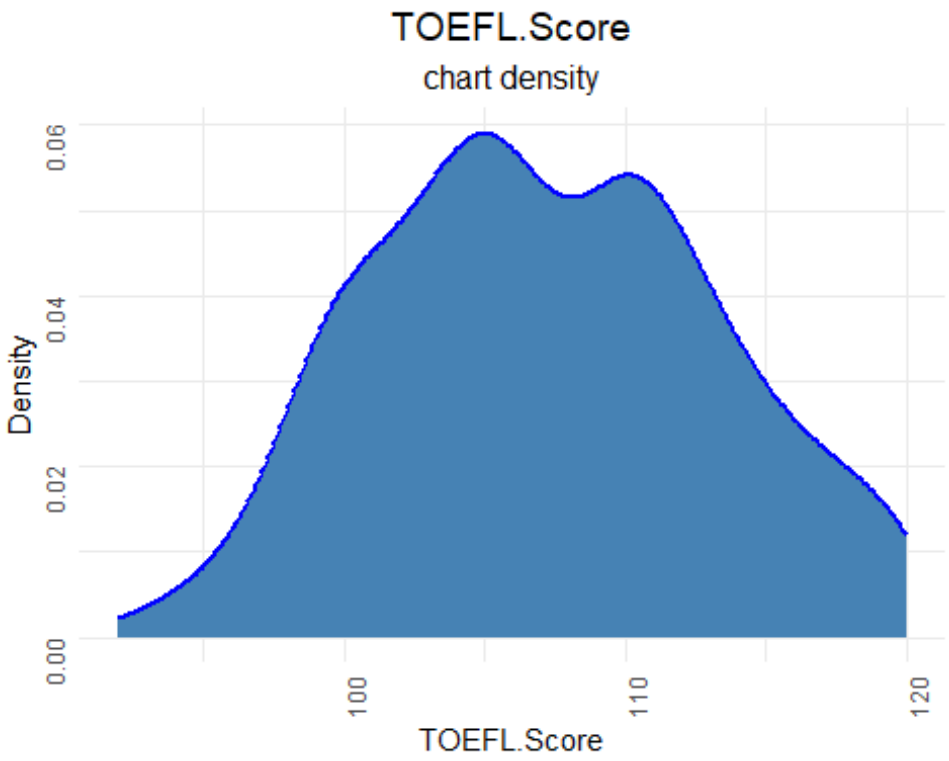
```
##  3rd Qu.:0.8200
##  Max.   :0.9700
```

2. Realice una gráfica de densidad para cada una de las variables numéricas en el dataset: GRE.Score, TOEFEL.Score, CGPA y Chance of Admit.

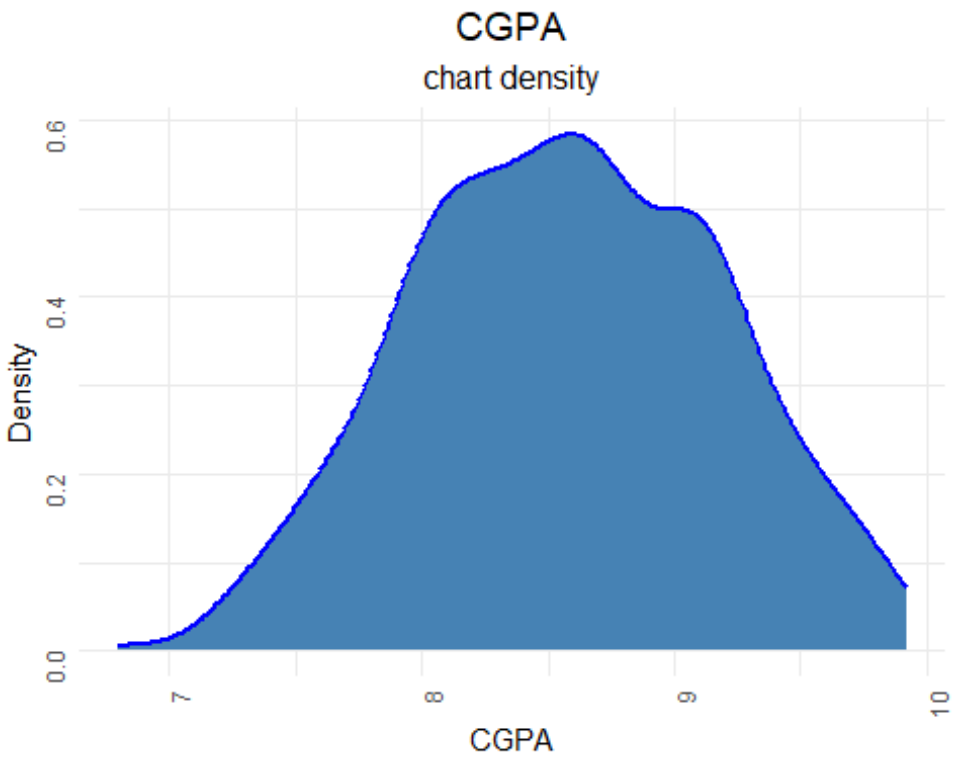
```
df %>%
    ggplot(aes(x=GRE.Score, y=after_stat(density)))+
    geom_density(col = "blue", lwd=1, fill = "steel blue")+
    theme_minimal()+
    labs(x= "GRE.Score", y = "Density", title = "GRE.Score", subtitle
="chart density")+
    theme(axis.text = element_text(angle = 90, hjust = 1),
          plot.title = element_text(hjust = 0.5, size= 14),
          plot.subtitle = element_text(hjust = 0.5, size=12))
```



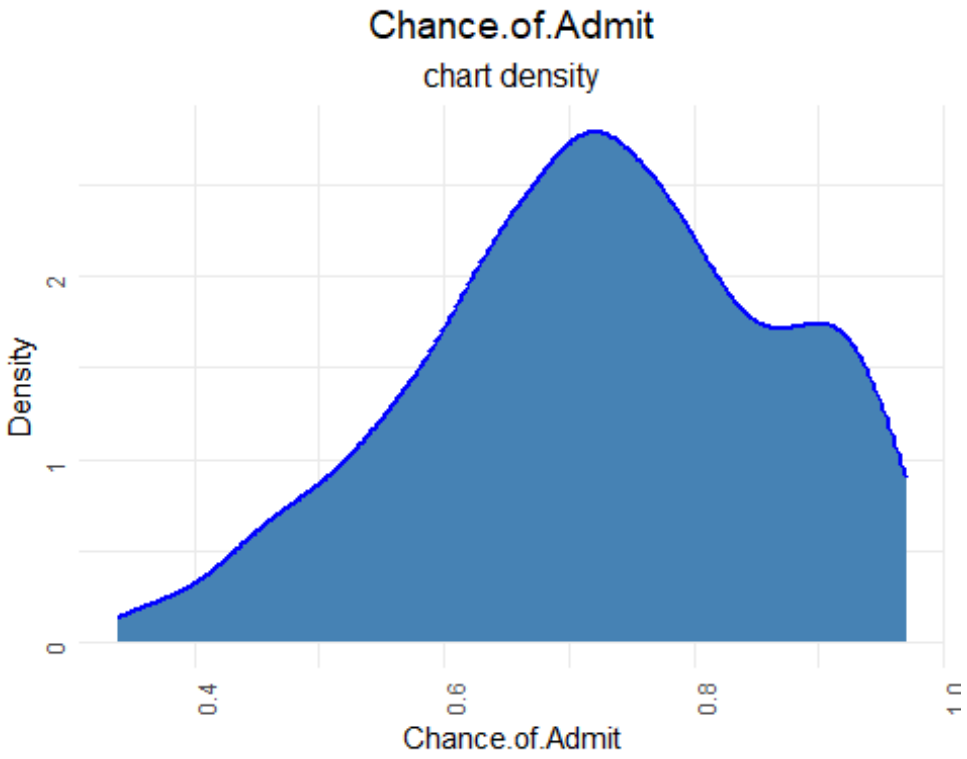

```
df %>%
    ggplot(aes(x=TOEFL.Score, y=after_stat(density)))+
    geom_density(col = "blue", lwd=1, fill = "steel blue")+
    theme_minimal()+
    labs(x= "TOEFL.Score", y = "Density", title = "TOEFL.Score", subtitle
="chart density")+
    theme(axis.text = element_text(angle = 90, hjust = 1),
          plot.title = element_text(hjust = 0.5, size= 14),
          plot.subtitle = element_text(hjust = 0.5, size=12))
```




```
df %>%
    ggplot(aes(x=CGPA, y=after_stat(density)))+
    geom_density(col = "blue", lwd=1, fill = "steel blue")+
    theme_minimal()+
    labs(x= "CGPA", y = "Density", title = "CGPA", subtitle ="chart
density")+
    theme(axis.text = element_text(angle = 90, hjust = 1),
          plot.title = element_text(hjust = 0.5, size= 14),
          plot.subtitle = element_text(hjust = 0.5, size=12))
```

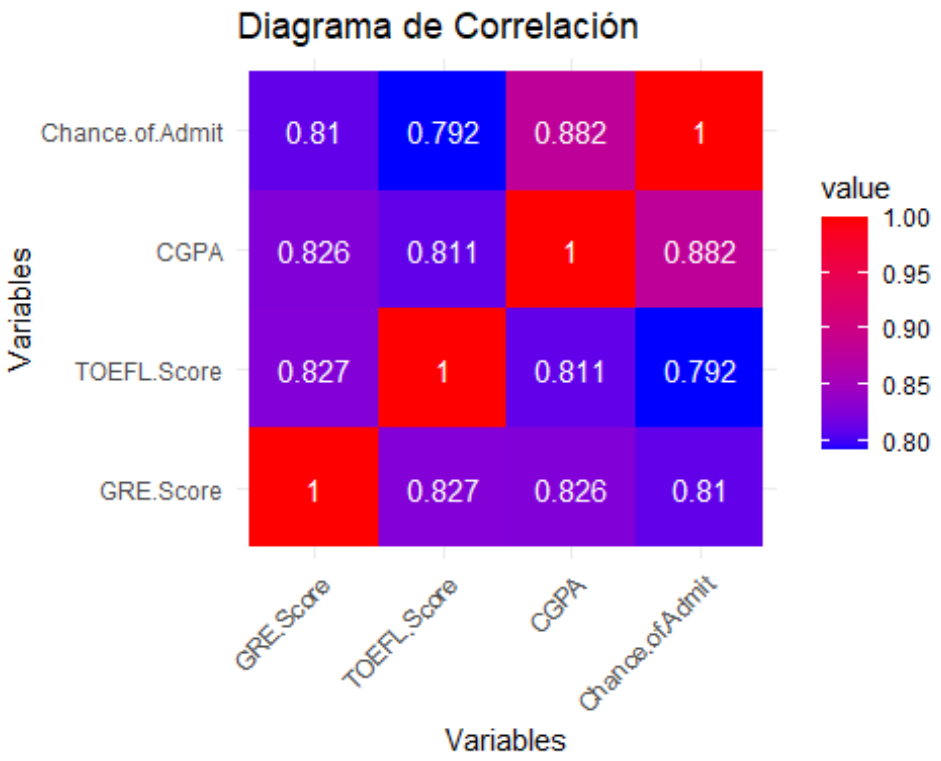



```
df %>%
    ggplot(aes(x=Chance.of.Admit, y=after_stat(density)))+
    geom_density(col = "blue", lwd=1, fill = "steel blue")+
    theme_minimal()+
    labs(x= "Chance.of.Admit", y = "Density", title = "Chance.of.Admit",
subtitle ="chart density")+
    theme(axis.text = element_text(angle = 90, hjust = 1),
          plot.title = element_text(hjust = 0.5, size= 14),
          plot.subtitle = element_text(hjust = 0.5, size=12))
```



3. Realice una gráfica de correlación entre las variables del inciso anterior.

```
correlation_plot <- function(df) {
  correlation_matrix <- cor(df)

  # Convertir la matriz de correlación en un dataframe
  correlation_df <- reshape2::melt(correlation_matrix)
  correlation_df$correlation <- round(correlation_df$value, 3)
  ggplot(data = correlation_df, aes(x = Var1, y = Var2, fill = value)) +
    geom_tile() +
    scale_fill_gradient(low = "blue", high = "red") +
    geom_text(aes(label = correlation), color = "white") +
    labs(title = "Diagrama de Correlación", x = "Variables", y =
"Variables") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

correlacion <- df %>% select(GRE.Score,TOEFL.Score, CGPA,
Chance.of.Admit)



# Generar el diagrama de correlación
correlation_plot(correlacion)
```

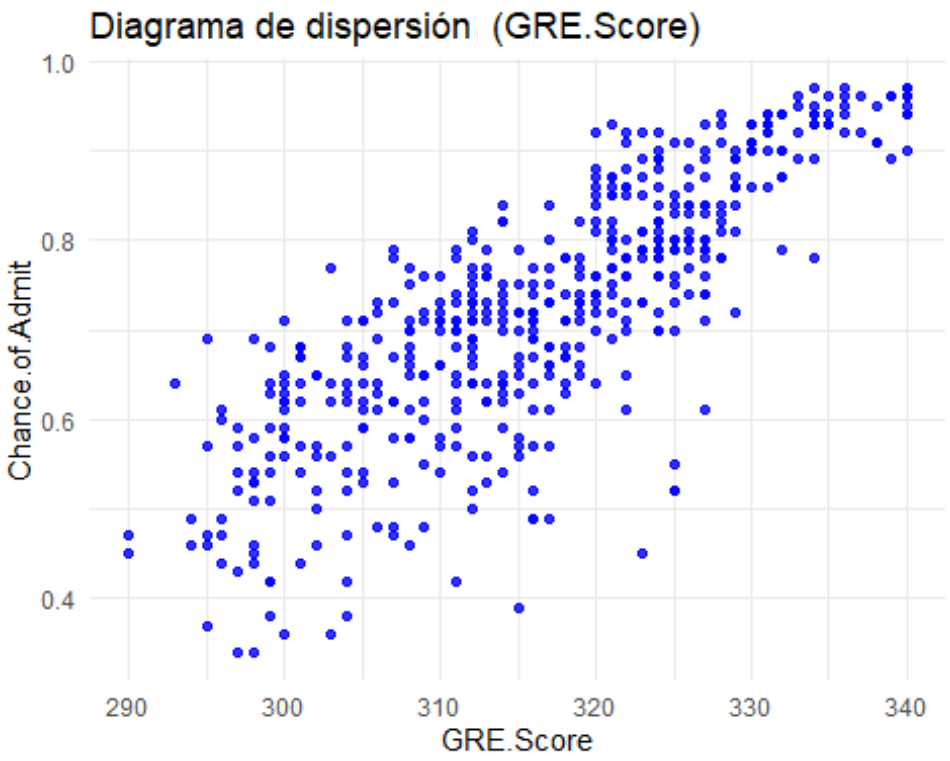


4. Realice comentarios sobre el análisis estadístico de las variables numéricas y la gráfica de correlación.

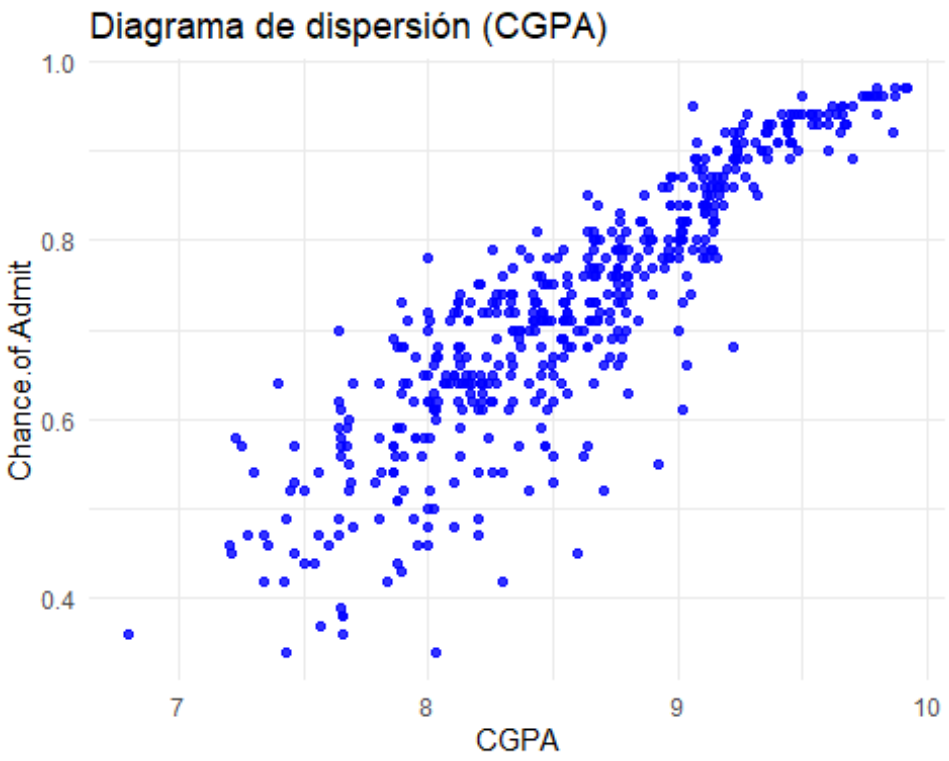
De acuerdo con la matriz de correlación se puede verificar que la variable dependiente y = chance of Admit tiene correlación positiva con las variables independientes, la correlación esta por arriba de 0.78, esto quiere decir que las variables tienen una correlación fuerte.

5. Realice un scatter plot (nube de puntos) de todas las variables numéricas contra la variable Chance of Admit.

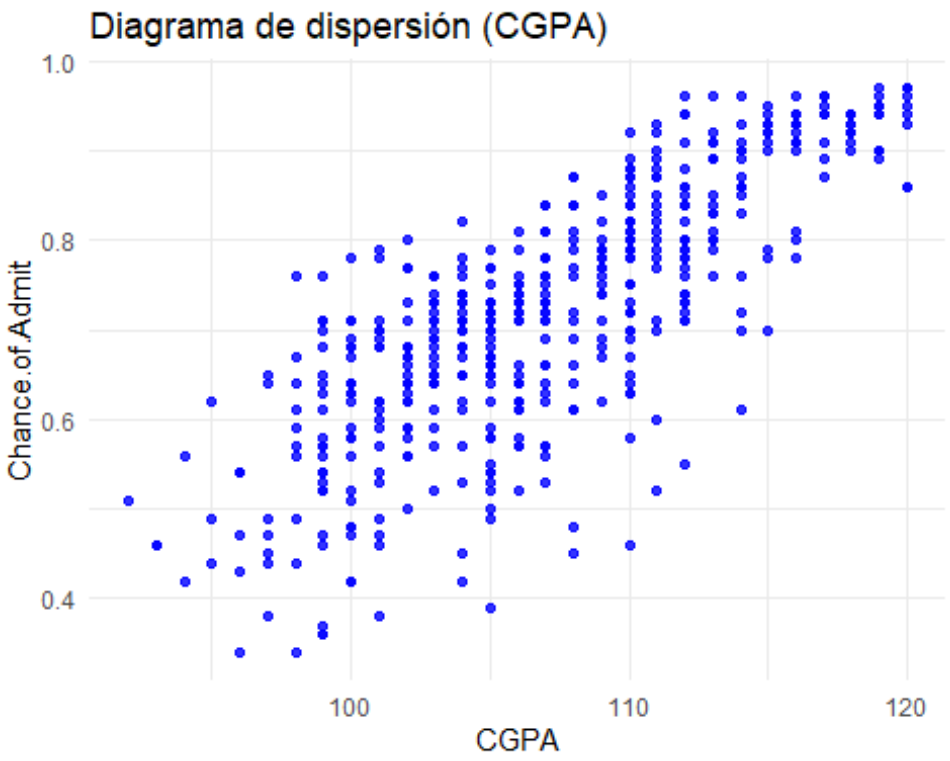
```
df_2 <- df %>% select(GRE.Score,TOEFL.Score, CGPA, Chance.of.Admit)

ggplot(df_2)+
  geom_point( aes(x= GRE.Score, y = Chance.of.Admit), color = "blue",
alpha = 0.8)+
  labs(x= "GRE.Score", y = "Chance.of.Admit", title = "Diagrama de
dispersión  (GRE.Score)" )+
  theme(plot.title = element_text(hjust = 0.5, size= 14))+
  theme_minimal()
```




```
ggplot(df_2)+
  geom_point( aes(x= CGPA, y = Chance.of.Admit), color = "blue", alpha = 0.8)+
  labs(x= "CGPA", y = "Chance.of.Admit", title = "Diagrama de dispersión (CGPA)" )+
  theme(plot.title = element_text(hjust = 0.5, size= 14))+
  theme_minimal()
```




```
ggplot(df_2)+
  geom_point( aes(x= TOEFL.Score, y = Chance.of.Admit), color = "blue",
alpha = 0.8)+
  labs(x= "CGPA", y = "Chance.of.Admit", title = "Diagrama de dispersión
(CGPA)" )+
  theme(plot.title = element_text(hjust = 0.5, size= 14))+
  theme_minimal()
```



6. Utilizando la función train y trainControl para crear un crossvalidation y le permita evaluar los siguientes modelos:

• Chance of Admit ~ TOEFL.Score. • Chance of Admit ~ CGPA • Chance of Admit ~ GRE.Score. • Chance of Admit ~ TOEFL.Score + CGPA • Chance of Admit ~ TOEFL.Score + GRE.Score. • Chance of Admit ~ TOEFL.Score + CGPA • Chance of Admit ~ TOEFL.Score + CGPA + GRE.Score.

Posteriormente cree una lista ordenando de mejor a peor cual es el mejor modelo en predicción, recuerde que es necesario caclular el RMSE para poder armar correctamente la lista.

```
formulas <- list(
  formula1 = as.formula("Chance.of.Admit ~ TOEFL.Score"),
  formula2 = as.formula("Chance.of.Admit ~ CGPA"),
  formula3 = as.formula("Chance.of.Admit ~ GRE.Score"),
  formula4 = as.formula("Chance.of.Admit ~ TOEFL.Score + CGPA"),
  formula5 = as.formula("Chance.of.Admit ~ TOEFL.Score + GRE.Score"),
  formula6 = as.formula("Chance.of.Admit ~ GRE.Score + CGPA"),
  formula7 = as.formula("Chance.of.Admit ~ TOEFL.Score + CGPA +
GRE.Score")
)

rmse <- function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}
```

```r
results <- list()

# Configurar el control de entrenamiento del crosvalidation
ctrl <- trainControl(method = "cv", number = 10)

# Iterar sobre las fórmulas y entrenar los modelos
for (i in seq_along(formulas)) {
  model <- train(
    formulas[[i]],
    data = df_2,
    method = "lm",
    trControl = ctrl,
    metric = "RMSE"
  )
  results[[i]] <- model
}

# Crear un vector de RMSE y nombres de modelos
rmse_values <- sapply(results, function(x) x$results$RMSE[1])
model_names <- names(results)

# Ordenar los modelos en función del menor RMSE
sorted_indices <- order(rmse_values)
sorted_results <- results[sorted_indices]
sorted_names <- model_names[sorted_indices]

rmse_ <- list()

# Imprimir la lista ordenada con el modelo y su respectivo RMSE
for (i in seq_along(sorted_results)) {
  model_name <- sorted_names[i]
  model_rmse <- rmse_values[sorted_indices[i]]
  print(cat("Model: ", i, model_name, "tRMSE:", model_rmse, "/n"))

}
```

```
## Model:  1 tRMSE: 0.0623516 /nNULL
## Model:  2 tRMSE: 0.06310746 /nNULL
## Model:  3 tRMSE: 0.06341269 /nNULL
## Model:  4 tRMSE: 0.06577437 /nNULL
## Model:  5 tRMSE: 0.07641941 /nNULL
## Model:  6 tRMSE: 0.08237858 /nNULL
## Model:  7 tRMSE: 0.08581024 /nNULL
```

De acuerdo a los resultados proporcionados, el modelo con el menor error RMSE es el Modelo 1, con un valor de 0.0621829. Los modelos se ordenan de mejor a peor rendimiento en función del valor del RMSE, por lo que el Modelo 1 es el que presenta la mejor capacidad predictiva en comparación con los demás modelos evaluados.

*Chance.of.Admit ~ TOEFL.Score + CGPA + GRE.Score

Ejercicio #3:

A continuación se le muestran tres imágenes que muestran los resultados obtenidos de correr la función summary() a dos modelos de regresión lineal, para este ejercicio se le solicita que realice la interpretación de las tablas resultantes. Recuerde tomar en cuenta la signficancia de los parámetros (signfícancia local), la signficancia del modelo (signficancia global), el valor del r^2: y cualquier observación que considere relevante para determinar si el modelo estructuralmente es adecuado o no.

Modelo 1

```
Call:
lm(formula = ROLL ~ UNEM, data = datavar)

Residuals:
    Min      1Q  Median      3Q     Max
-7640.0 -1046.5   602.8  1934.3  4187.2

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3957.0     4000.1   0.989   0.3313
UNEM          1133.8      513.1   2.210   0.0358 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3049 on 27 degrees of freedom
Multiple R-squared: 0.1531, Adjusted R-squared: 0.1218
F-statistic: 4.883 on 1 and 27 DF,  p-value: 0.03579
```

Prueba de significancia global F $H_0 $: El modelo no es funcional para explicar a Y (x = 0) $H_1 $: el modelo es funcional, al menos un X explica a Y (al menos un coeficiente es diferente de cero)

F-statistic = 4.883 p- value = 0.03579

F_calculado >= F alfa >= valor p 0.05 >= 0.03579 Alfa es mayor al valor p, por lo tanto se rechaza H0 a favor de Ha, el modelo es funcional para explicar Y, al menos uno de los coeficientes es diferente de cero.

r^2 El coeficiente de de correlación esta esta cercano a 10, esto indica que hay una relación debil entre la variable independiente y la dependiente, por lo que no es un buen modelo que explique a ROLL la variable independiente UNEM.

El intercepto no es significativo, y la variable independiente es cercano al máximo nivel de error aceptado del 5% = 0.05, esto quiere decir que este primer modelo no es el optimo para explicar ROLL.

Modelos 2

```
Call:
lm(formula = ROLL ~ UNEM + HGRAD + INC, data = datavar)

Residuals:
     Min       1Q   Median       3Q      Max
-1148.840  -489.712    -1.876   387.400  1425.753

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.153e+03  1.053e+03  -8.691 5.02e-09 ***
UNEM         4.501e+02  1.182e+02   3.809 0.000807 ***
HGRAD        4.065e-01  7.602e-02   5.347 1.52e-05 ***
INC          4.275e+00  4.947e-01   8.642 5.59e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 670.4 on 25 degrees of freedom
Multiple R-squared: 0.9621, Adjusted R-squared: 0.9576
F-statistic: 211.5 on 3 and 25 DF,  p-value: < 2.2e-16
```

Prueba de significancia global F ${H_0} $: El modelo no es funcional para explicar a Y (x = 0) ${H_1} $: el modelo es funcional, al menos un X explica a Y (al menos un coeficiente es diferente de cero)

F-statistic = 211.5 p- value = 2.2e-16

F_calculado >= F alfa >= valor p 0.05 >= 2.2e-16 Alfa es mayor al valor p, por lo tanto se rechaza H0 a favor de Ha, el modelo es funcional para explicar Y, al menos uno de los coeficientes es diferente de cero.

r^2 Siguendo con la misma variable dependiente del modelo #1, pero ahora con más variables independientes, se puede observar que el coeficiente de correlación incrementa arriba del 0.90, esto quiere decir de que la relación de la variable dependiente con las independientes es fuerte y positiva, al agregarle otras tres variables puede ayudar a mejorar a explicar el modelo.

De acuerdo al valor p de cada una de las variables se puede observar que todas aportan al modelo y son significativas que pueden explicar a la variable dependiente.

Modelo 3

```
Call:
lm(formula = Cab.Price ~ Months, data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max
-11.034  -2.305  -1.034   2.764   9.241

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  72.6826     3.2377   22.45 6.92e-10 ***
Months        4.8626     0.3495   13.91 7.18e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.657 on 10 degrees of freedom
Multiple R-squared:  0.9509,    Adjusted R-squared:  0.946
F-statistic: 193.6 on 1 and 10 DF,  p-value: 7.181e-08
```

Prueba de significancia global F $H_0 $: El modelo no es funcional para explicar a Y (x = 0) $H_1 $: el modelo es funcional, al menos un X explica a Y (al menos un coeficiente es diferente de cero)

F-statistic = 193.6 p- value = 7.181e-08

F_calculado >= F alfa >= valor p 0.05 >= 7.181e-08 Alfa es mayor al valor p, por lo tanto se rechaza H0 a favor de Ha, el modelo es funcional para explicar Y, al menos uno de los coeficientes es diferente de cero.

r^2 el coeficiente de correlacion es de 0.95 de la regresión lineal simple, sin embargo se pude observar que el precio (variable dependiente) es explicada por la variable independiente (MONTH), sin embargo puede haber un sobreajuste del modelo que no es malo descartarlo.

Tanto el intercepto como la variable independiente son significativas para el modelo debio a que son menor al error del 0.05.