HW1

Carina\_Sun

2024-02-20

knitr::opts\_chunk$set(echo = TRUE)  
  
  
library(psych)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ ggplot2::%+%() masks psych::%+%()  
## ✖ ggplot2::alpha() masks psych::alpha()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tinytex)  
library(markdown)  
library(skimr)  
  
# 2. Read in your file and Assign it to the name grecollege   
grecollege <- read.csv("grecollege.csv")  
  
# 3. Check the structure of the data using str  
str(grecollege)

## 'data.frame': 400 obs. of 4 variables:  
## $ admit: num 0 1 1 1 0 1 1 0 1 0 ...  
## $ gre : num 380 660 800 640 520 760 560 400 540 700 ...  
## $ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...  
## $ rank : num 3 3 1 4 4 2 1 2 3 2 ...

# 4. Change admit into a factor using as.factor  
grecollege$admit <- as.factor(grecollege$admit)  
  
# 5. Check the structure using str again and confirm it is a factor.   
str(grecollege)

## 'data.frame': 400 obs. of 4 variables:  
## $ admit: Factor w/ 2 levels "0","1": 1 2 2 2 1 2 2 1 2 1 ...  
## $ gre : num 380 660 800 640 520 760 560 400 540 700 ...  
## $ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...  
## $ rank : num 3 3 1 4 4 2 1 2 3 2 ...

# 6. Index all columns  
names(grecollege)

## [1] "admit" "gre" "gpa" "rank"

# 7. Index the 40th row, first two columns  
grecollege[40, 1:2]

## admit gre  
## 40 1 520

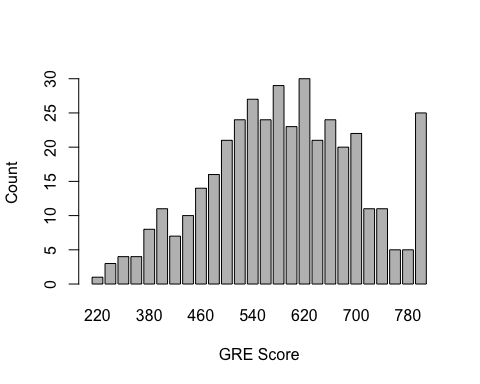
# 8. Use the R commands describe or glimpse to get descriptive statistics for all variables, including mean, sd, var, and n’s.  
describe(grecollege)

## vars n mean sd median trimmed mad min max range skew  
## admit\* 1 400 1.32 0.47 1.0 1.27 0.00 1.00 2 1.00 0.78  
## gre 2 400 587.70 115.52 580.0 589.06 118.61 220.00 800 580.00 -0.14  
## gpa 3 400 3.39 0.38 3.4 3.40 0.40 2.26 4 1.74 -0.21  
## rank 4 400 2.49 0.94 2.0 2.48 1.48 1.00 4 3.00 0.10  
## kurtosis se  
## admit\* -1.39 0.02  
## gre -0.36 5.78  
## gpa -0.60 0.02  
## rank -0.91 0.05

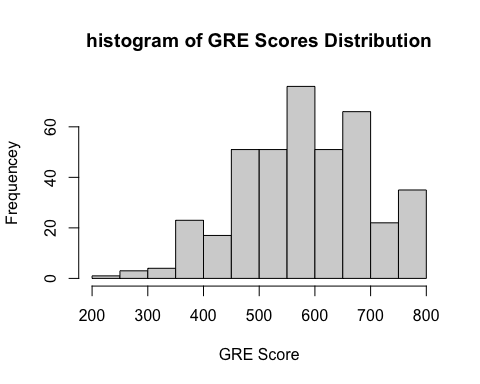
# 9. Use describeBY to group by admit to get descriptive statistics.  
describeBy(grecollege, group = grecollege$admit)

##   
## Descriptive statistics by group   
## group: 0  
## vars n mean sd median trimmed mad min max range skew  
## admit 1 273 1.00 0.00 1.00 1.00 0.00 1.00 1 0.00 NaN  
## gre 2 273 573.19 115.83 580.00 574.25 118.61 220.00 800 580.00 -0.10  
## gpa 3 273 3.34 0.38 3.34 3.35 0.40 2.26 4 1.74 -0.07  
## rank 4 273 2.64 0.92 3.00 2.68 1.48 1.00 4 3.00 -0.03  
## kurtosis se  
## admit NaN 0.00  
## gre -0.38 7.01  
## gpa -0.55 0.02  
## rank -0.89 0.06  
## ------------------------------------------------------------   
## group: 1  
## vars n mean sd median trimmed mad min max range skew  
## admit 1 127 2.00 0.00 2.00 2.00 0.00 2.00 2 0.00 NaN  
## gre 2 127 618.90 108.88 620.00 620.39 118.61 300.00 800 500.00 -0.17  
## gpa 3 127 3.49 0.37 3.54 3.51 0.40 2.42 4 1.58 -0.53  
## rank 4 127 2.15 0.92 2.00 2.07 1.48 1.00 4 3.00 0.44  
## kurtosis se  
## admit NaN 0.00  
## gre -0.35 9.66  
## gpa -0.38 0.03  
## rank -0.64 0.08

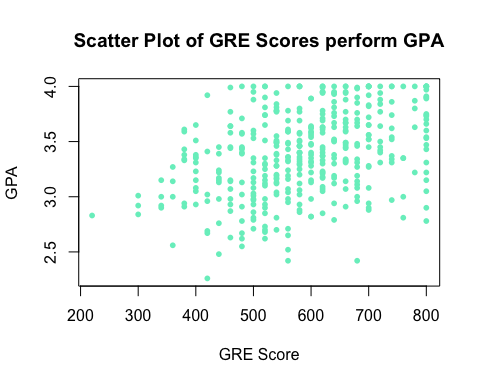
# 10. Using counts and barplot create a barplot for gre and label axes.  
gre\_counts <- table(grecollege$gre)  
barplot(gre\_counts, xlab = "GRE Score", ylab = "Count")



# 11. Create histogram for GRE with main labels and axes labels  
hist(grecollege$gre, main = "histogram of GRE Scores Distribution", xlab = "GRE Score", ylab = "Frequencey")



# 12. Create a scatterplot for gre and gpa with main labels and axes. Change any of the options for colors, cex size, points, main and sub labels font sizes, bold, etc.  
plot (grecollege$gre, grecollege$gpa, main = "Scatter Plot of GRE Scores perform GPA", xlab = "GRE Score", ylab = "GPA", col = "aquamarine2", cex = 0.8, pch = 16)



# 13. Use lm to get a linear model with gpa as the DV and gre as the IV  
#a. Use Summary to get a print out of the lm   
  
lm\_model <- lm(gpa ~ gre, data = grecollege)  
summary(lm\_model)

##   
## Call:  
## lm(formula = gpa ~ gre, data = grecollege)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.08675 -0.22435 -0.00015 0.24809 0.76176   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.6458978 0.0913100 28.977 < 2e-16 \*\*\*  
## gre 0.0012660 0.0001525 8.304 1.6e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3518 on 398 degrees of freedom  
## Multiple R-squared: 0.1477, Adjusted R-squared: 0.1455   
## F-statistic: 68.95 on 1 and 398 DF, p-value: 1.596e-15

#b. INTERPRET the coefficient and results.  
# The Intercept is the baseline GPA when GRE equals to 0 is 2.6458978, and the Gre coefficient is small and significantly positive, as P <0.001. Their equation is GPA=0.0012660×GRE+2.6458978. However, since the R-squared is low, other factors should also be included in the model predicting GPA.