Regression\_hw1

JiminLee\_2229027

2023-10-02

## Applied questions: Data analysis of body fat dataset(60 pts)

install.packages('gridExtra', dependencies=TRUE, repos = "http://cran.us.r-project.org")

## Warning in readRDS(dest): lzma decoder corrupt data

##   
## The downloaded binary packages are in  
## /var/folders/zl/ljg24f\_15m93w7f2rt14c2n80000gn/T//Rtmpo9NNZF/downloaded\_packages

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 1.0.2  
## ✔ tibble 3.2.1 ✔ dplyr 1.1.3  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.2  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(gridExtra)

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

Consider the dataset contained in ‘BODY FAT.TXT’. Please read carefully the description of this dataset and the guidelines for this part of the homework before performing the analyses in R and writing your results (file datasetguide.pdf).

setwd('/Users/jimin/Desktop/데스크탑 - 이지민의 MacBook Air/지민/ewha/2023-2/Regression/week3/')

body.fat = read.table('body\_fat.txt', header=TRUE)  
head(body.fat, 5)

## Density SiriBFperc Over45 Weight Height NeckC ChestC AbdomenC HipC ThighC  
## 1 1.0708 12.3 0 154.25 67.75 36.2 93.1 85.2 94.5 59.0  
## 2 1.0853 6.1 0 173.25 72.25 38.5 93.6 83.0 98.7 58.7  
## 3 1.0414 25.3 0 154.00 66.25 34.0 95.8 87.9 99.2 59.6  
## 4 1.0751 10.4 0 184.75 72.25 37.4 101.8 86.4 101.2 60.1  
## 5 1.0340 28.7 0 184.25 71.25 34.4 97.3 100.0 101.9 63.2  
## KneeC AnkleC BicepsC ForearmC WristC  
## 1 37.3 21.9 32.0 27.4 17.1  
## 2 37.3 23.4 30.5 28.9 18.2  
## 3 38.9 24.0 28.8 25.2 16.6  
## 4 37.3 22.8 32.4 29.4 18.2  
## 5 42.2 24.0 32.2 27.7 17.7

### a. (10 points) Re-compute body fat percentage using the ‘Density’ variable and Siri’s equation:

body fat percentage = (495/Density) − 450 Note that the values are percentages, so they must be between 0 and 100; change any negative value to 0, and any value > 100 to 100. Are there any erroneous values in the variable ‘SiriBF- Perc’? If so, employ your recomputed variable in further analyses. [Hint: the variable ‘SiriBF- Perc’ has only one decimal place. Before comparing it to your recomputed variable, you will need to round your recomputed variable. You can do so with the command round(variable,digits=1)] Along with body fat percentage, consider the variables weight, height and abdomen circumference. Subsample the dataset in order to have only the variables of interest [Hint: use the command dataset[,columns to keep]]. Produce a scatter plot matrix of the subsampled dataset [Hint: use the command plot(subsampled dataset)]. Looking at this will help you see if there are units (men) for which some of the measurements contain obvious mistakes. Are there any? If so, remove those units in further analyses.

# recompute body fat percentage  
compute.bodyfat = function(x){  
 y = (495/x) - 450  
 return (round(y, 1))  
}  
  
# find errors of dataset SiriBF- Perc  
re.SiriBFperc = apply(body.fat[c('Density')], 2, compute.bodyfat) # recomputed body fat perc  
  
# del.errors : 0 이하면 0으로, 100 초과하면, 100으로 만들어 주는 함수   
del.errors = function(x){  
 if (x >= 0 & x<= 100) return (x)  
 else if(x < 0) return (0)  
 else return (100)  
}  
  
fixed.SiriBFperc = apply(re.SiriBFperc, 1, del.errors) # error fixed ver of re.SiriBFperc  
  
# 오류를 제거한 버전과 원본 버전의 차이를 비교해 몇 개의 잘못된 데이터가 있었는지 확인  
print(length(re.SiriBFperc))

## [1] 252

print(sum(re.SiriBFperc == fixed.SiriBFperc))

## [1] 251

# 기존 SiriBFperc 열에 오류가 있는지 확인한다.   
SiriBFperc = body.fat[c('SiriBFperc')]  
SiriBFperc.col = apply(SiriBFperc, 1, del.errors)  
  
# 오류를 제거한 버전과 원본 버전의 차이를 비교해 몇 개의 잘못된 데이터가 있었는지 확인  
print(length(SiriBFperc.col))

## [1] 252

print(sum(SiriBFperc == SiriBFperc.col))

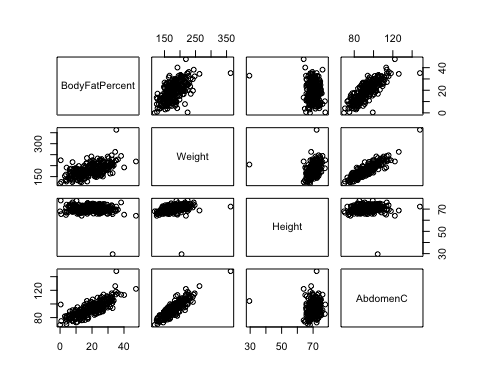
## [1] 252

# 오류가 있는 데이터가 없다.

# new sub dataset : including bodyfatpercent, weight, height, abdomenC  
sub.bodyfat = cbind(fixed.SiriBFperc, body.fat[,c('Weight', 'Height','AbdomenC')])  
colnames(sub.bodyfat)[1] = 'BodyFatPercent' # col name 변경   
head(sub.bodyfat, 5)

## BodyFatPercent Weight Height AbdomenC  
## 1 12.3 154.25 67.75 85.2  
## 2 6.1 173.25 72.25 83.0  
## 3 25.3 154.00 66.25 87.9  
## 4 10.4 184.75 72.25 86.4  
## 5 28.7 184.25 71.25 100.0

plot(sub.bodyfat)



# outliers 솎아내기 : 인덱스 파악  
which(sub.bodyfat[,'Weight'] >= 250) # 39, 41

## [1] 39 41

which(sub.bodyfat[,'Height'] < 50) # 42

## [1] 42

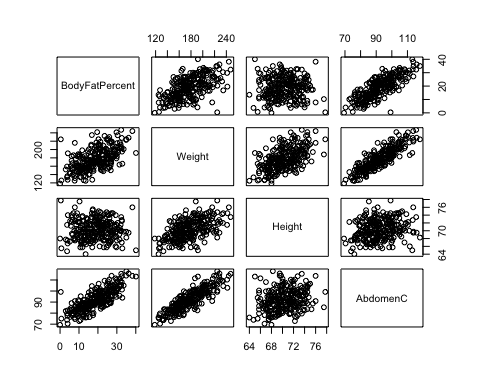
which(sub.bodyfat[,'AbdomenC'] > 130) # 39

## [1] 39

which(sub.bodyfat[,'BodyFatPercent'] >= 45) # 216

## [1] 216

sub.bodyfat.1 = sub.bodyfat[-c(39,41,42,216),] # delete outliers   
  
plot(sub.bodyfat.1)



### b. (20 points) Produce numerical and graphical summaries for body fat percentage, weight, height and abdomen circumference. Do the distributions appear symmetric and bell-shaped, or are they skewed? What can you observe about the variability of the different variables? Are there some very extreme values in any of the distributions? Based on these data, is there evidence that the average body fat percentage in the male population exceeds 25%? How about evidence that the average weight in the male population not equals to 180 pounds? Perform hypothesis tests to answer these questions, and provide the p-values.

# numercial summary + variable's variability(range, IQR, var, std)  
summary(sub.bodyfat)

## BodyFatPercent Weight Height AbdomenC   
## Min. : 0.0 Min. :118.5 Min. :29.50 Min. : 69.40   
## 1st Qu.:12.4 1st Qu.:159.0 1st Qu.:68.25 1st Qu.: 84.58   
## Median :19.2 Median :176.5 Median :70.00 Median : 90.95   
## Mean :19.1 Mean :178.9 Mean :70.15 Mean : 92.56   
## 3rd Qu.:25.3 3rd Qu.:197.0 3rd Qu.:72.25 3rd Qu.: 99.33   
## Max. :47.5 Max. :363.1 Max. :77.75 Max. :148.10

apply(sub.bodyfat, 2, sd) # standard deviation

## BodyFatPercent Weight Height AbdomenC   
## 8.432132 29.389160 3.662856 10.783077

apply(sub.bodyfat, 2, var) # variance

## BodyFatPercent Weight Height AbdomenC   
## 71.10085 863.72272 13.41651 116.27475

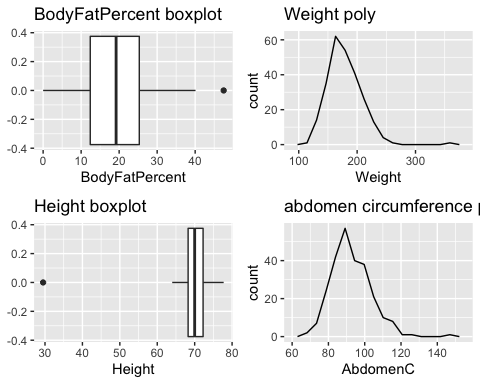
apply(sub.bodyfat, 2, range) # range : min, max value

## BodyFatPercent Weight Height AbdomenC  
## [1,] 0.0 118.50 29.50 69.4  
## [2,] 47.5 363.15 77.75 148.1

apply(sub.bodyfat, 2, IQR) # IQR

## BodyFatPercent Weight Height AbdomenC   
## 12.90 38.00 4.00 14.75

# graphical summary   
g1 = ggplot(sub.bodyfat, aes(x=BodyFatPercent)) + geom\_boxplot() + ggtitle('BodyFatPercent boxplot')  
g2 = ggplot(sub.bodyfat, aes(x=Weight)) + geom\_freqpoly(bins=16) + ggtitle('Weight poly')  
g3 = ggplot(sub.bodyfat, aes(x=Height)) + geom\_boxplot() + ggtitle('Height boxplot')  
g4 = ggplot(sub.bodyfat, aes(x=AbdomenC)) + geom\_freqpoly(bins=16) + ggtitle('abdomen circumference poly')  
  
grid.arrange(g1,g2,g3,g4, ncol=2,nrow=2) # 4개의 그래프를 한 번에 출력

 BodyFatPercent : slightly left skewed. Weight, AbdomenC : right skewed. Height : one extreme outlier, slightly right skewed.

Outlier은 존재한다. 구체적으로 어떤 outlier가 존재하는지는 (a)에서 이미 찾았기에 추가적으로 서술하지 않는다. 하단에 진행한 평균에 대한 가설검정은 outlier를 제거한 sub.bodyfat.1로 진행한다.

# 모집단의 분포를 모르지만, n=252로 z-검정을 사용한다.  
  
z.test = function(mu, mean, n, sd, alternative="two.sided", conf.level){  
 test.statistics = round((mean - mu) / (sd / (n^0.5)), 2)  
 alpha = (1 - conf.level)  
   
 if (alternative == 'two.sided'){  
 p.value = 2 \* pnorm(abs(test.statistics), lower.tail = FALSE)  
 }  
 else if (alternative == 'less'){  
 p.value = pnorm(abs(test.statistics))  
 }  
 else if (alternative == 'greater'){  
 p.value = pnorm(abs(test.statistics), lower.tail = FALSE)   
 }  
 else {  
 print('Wrong Alternative. Alternative should be two.sided, less, or greater.')  
 return ()  
 }  
   
 cat('Statistical Hypothesis Test in conf.level = ',conf.level,'\n')  
 cat('--------------------------------------------------\n')  
 cat('alpha : ', alpha, '\n')  
 cat('test.statistics : ', test.statistics, '\n')  
 cat('p.value : ', round(p.value, 4), '\n')  
 cat('--------------------------------------------------\n')  
   
 if (p.value <= alpha) cat('H0(null hypothesis) is rejected.')  
 else cat('H0(null hypothesis) is not rejected.')  
}

Body Fat Percent에 대한 가설 검정(0.05 유의수준에서) H0 : mu = 0.25 H1 : mu > 0.25

z.test(mu = 0.25,   
 mean = mean(sub.bodyfat.1[,'BodyFatPercent']),   
 n = length(sub.bodyfat.1[,'BodyFatPercent']),   
 sd = sd(sub.bodyfat.1[,'BodyFatPercent']),   
 alternative = 'greater',   
 conf.level = 0.95)

## Statistical Hypothesis Test in conf.level = 0.95   
## --------------------------------------------------  
## alpha : 0.05   
## test.statistics : 35.94   
## p.value : 0   
## --------------------------------------------------  
## H0(null hypothesis) is rejected.

Weight average에 대한 가설 검정 (0.05 유의수준에서) H0 : mu = 180 H1 : mu != 180

z.test(mu = 180,   
 mean = mean(sub.bodyfat.1[,'Weight']),   
 n = length(sub.bodyfat.1[,'Weight']),   
 sd = sd(sub.bodyfat.1[,'Weight']),   
 alternative = 'two.sided',   
 conf.level = 0.95)

## Statistical Hypothesis Test in conf.level = 0.95   
## --------------------------------------------------  
## alpha : 0.05   
## test.statistics : -1.44   
## p.value : 0.1499   
## --------------------------------------------------  
## H0(null hypothesis) is not rejected.

### c. (30 points) Compute the correlation between the body fat percentage and each of the variables weight, height and abdomen. Then, fit three separate simple regression models for body fat percentage versus weight, height and abdomen circumference. For each regression model, make sure you provide:

• Model employed; • Least square estiamtes of the parameter β0 and β1; • The equation of the estimated regression line

# the least square estimators  
cal.LSE = function(X, y){  
 beta = solve(t(X)%\*%X)%\*%t(X)%\*%y  
 return (beta)  
}

# body fat percentage & weight  
x1 = sub.bodyfat.1$Weight  
x1 = cbind(1,x1)  
y = sub.bodyfat.1$BodyFatPercent  
  
cat('Least square estiamtes of the parameter β0 and β1\n')

## Least square estiamtes of the parameter β0 and β1

LSE1 = cal.LSE(x1,y)  
print(LSE1)

## [,1]  
## -13.1479428  
## x1 0.1799537

cat('\nThe equation of the estimated regression line\n')

##   
## The equation of the estimated regression line

cat('y = ',LSE1[1], '+', LSE1[2], 'x + e')

## y = -13.14794 + 0.1799537 x + e

# body fat percentage & height  
x2 = sub.bodyfat.1$Height  
x2 = cbind(1, x2)  
  
LSE2 = cal.LSE(x2,y)  
  
cat('Least square estiamtes of the parameter β0 and β1\n')

## Least square estiamtes of the parameter β0 and β1

print(LSE2)

## [,1]  
## 21.30954048  
## x2 -0.03557106

cat('\nThe equation of the estimated regression line\n')

##   
## The equation of the estimated regression line

cat('y = ',LSE2[1], '+', LSE2[2], 'x + e')

## y = 21.30954 + -0.03557106 x + e

# body fat percentage & abodomen  
x3 = sub.bodyfat.1$AbdomenC  
x3 = cbind(1, x3)  
  
cat('Least square estiamtes of the parameter β0 and β1\n')

## Least square estiamtes of the parameter β0 and β1

LSE3 = cal.LSE(x3,y)  
print(LSE3)

## [,1]  
## -42.2187608  
## x3 0.6631156

cat('\nThe equation of the estimated regression line\n')

##   
## The equation of the estimated regression line

cat('y = ',LSE3[1], '+', LSE3[2], 'x + e')

## y = -42.21876 + 0.6631156 x + e

p1 = ggplot(data=sub.bodyfat.1, aes(x=Weight, y=BodyFatPercent, color=Weight)) +   
 geom\_point(size=0.6) +   
 stat\_smooth(method='lm', formula = y~x, geom = 'smooth') +   
 ggtitle('x = weight')  
  
p2 = ggplot(data=sub.bodyfat.1, aes(x=Height, y=BodyFatPercent, color=Height)) +   
 geom\_point(size=0.6) +   
 stat\_smooth(method='lm', formula = y~x, geom = 'smooth') +   
 ggtitle('x = height')  
  
p3 = ggplot(data=sub.bodyfat.1, aes(x=AbdomenC, y=BodyFatPercent, color=AbdomenC)) +   
 geom\_point(size=0.6) +   
 stat\_smooth(method='lm', formula = y~x, geom = 'smooth') +   
 ggtitle('x = AbdomenC')  
  
grid.arrange(p1, p2, p3, ncol = 2, nrow = 2)

