#Exercise 1----------

```
#1 Number of students
library(tidyverse)
map(datstu,~sum(is.na(.)))
q1.1 = nrow(datstu)
cat('1.1 The number of students is:', q1.1, '\n')
```

1.1 The number of students is: 340823

```
#2 Number of Schools
number_of_schools <- unique(datsss$schoolcode)
number_of_schools <- data.frame(number_of_schools)
q1.2 = nrow(number_of_schools)
cat('1.2 The number of school is:',q1.2,'\n')
```

1.2 The number of school is: 898

```
#3 Number of programs
prgm = select(datstu, choicepgm1, choicepgm2, choicepgm3, choicepgm4, choicepgm5, choicepgm6, X)
prgm_Total = gather(prgm, program, progname, choicepgm1, choicepgm2, choicepgm3, choicepgm4, choicepgm5, choicepgm6)
prgm_Total_Unique = unique(prgm_Total)
cat('1.3 The number of program is:', nrow(prgm_Total_Unique), '\n')
```

1.3 The number of program is: 2044938

```
#4 Number of choices (school,program)
school = datstu %>% select(schoolcode1, schoolcode2, schoolcode3, schoolcode4, schoolcode5, schoolcode6, X) %>% gather (, schoolcode, schoolcode1, schoolcode2, schoolcode3, schoolcode4, schoolcode5, schoolcode6)
choices = cbind(school[,3], prgm_Total[,3])
choices_unique = unique(choices)
cat('1.4 The number of choices is:', nrow(choices_unique), '\n')
```

1.4 The number of choices is: 3086

```
#5 Missing test score
missing_test_score = sum(is.na(datstu$score) == 'TRUE', na.rm = TRUE)
cat('1.5 The number of missed test score is:', missing_test_score, '\n')
```

1.5 The number of missed test score is: 179887

```
#6 Apply to the same school (different programs)
f1 <- function(x) {length(x[!is.na(x)]) - length(unique(x[!is.na(x)]))}
same_school = apply(datstu[, 5:10], MARGIN = 1, FUN = f1)
length(same_school[same_school != 0])
cat('1.6 The number of students apply to the same school but different programs:', length(same_school[same_school != 0]), '\n')
```

1.6 The number of students apply to the same school but different programs: 120071

```
#7 Apply to less than 6 choices
less_than_6 = sum(is.na(datstu$schoolcode6)=='TRUE', na.rm = TRUE)
cat('1.7 The number of student apply to less than 6 choices is:', less_than_6, '\n')
```

1.7 The number of student apply to less than 6 choices is: 17088

#Exercise 2----------

```
datstu$admitted_by_schoolcode=ifelse(datstu$rankplace==1, datstu$schoolcode1,
                                     ifelse(datstu$rankplace==2, datstu$schoolcode2,
                                     ifelse(datstu$rankplace==3, datstu$schoolcode3,
                                     ifelse(datstu$rankplace==4, datstu$schoolcode4,
                                     ifelse(datstu$rankplace==5, datstu$schoolcode5,
                                     ifelse(datstu$rankplace==6, datstu$schoolcode6, NA))))))
datstu$admitted=ifelse(datstu$rankplace==1, datstu$choicepgm1,
                       ifelse(datstu$rankplace==2, datstu$choicepgm2,
                       ifelse(datstu$rankplace==3, datstu$choicepgm3,
                       ifelse(datstu$rankplace==4, datstu$choicepgm4,
                       ifelse(datstu$rankplace==5, datstu$choicepgm5,
                       ifelse(datstu$rankplace==6, datstu$choicepgm6, NA))))))
data_raw=datstu %>%
  group_by(admitted) %>%
  summarise(schoolcode=admitted_by_schoolcode,                    minscore=min(score),
average=mean(score), number=n())
dataraw=data_raw %>%
  rename(school_program = admitted)
dataraw=unique(dataraw)
datsss$X=NULL
datsss=unique(datsss)
data_q2<-merge(x=dataraw,y=datsss,by="schoolcode",all.x=TRUE)
data_q2=na.omit(data_q2)
names(data_q2)[names(data_q2) == "minscore"] <- "Cutoff"
names(data_q2)[names(data_q2) == "average"] <- "Quality"
names(data_q2)[names(data_q2) == "number"] <- "Size"
```

	schoolcode	school_program	Cutoff	Quality	Size	schoolname	sssdistrict	ssslong	ssslat
1	10101	Home Economics	190	280.0537	14994	EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
3	10101	Agriculture	188	274.7164	13168	EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
5	10101	General Science	158	334.1679	15125	EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
7	10101	Business	194	297.6537	30304	EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
9	10101	Visual Arts	173	295.9655	9989	EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
11	10101	General Arts	194	298.1614	46517	EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
13	10102	General Arts	194	298.1614	46517	ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO	Accra Metropolitan	-0.19711526	5.607396
15	10102	Visual Arts	173	295.9655	9989	ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO	Accra Metropolitan	-0.19711526	5.607396
17	10102	General Science	158	334.1679	15125	ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO	Accra Metropolitan	-0.19711526	5.607396
19	10102	Home Economics	190	280.0537	14994	ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO	Accra Metropolitan	-0.19711526	5.607396
22	10103	Agriculture	188	274.7164	13168	WESLEY GRAMMAR SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
24	10103	General Arts	194	298.1614	46517	WESLEY GRAMMAR SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
26	10103	Home Economics	190	280.0537	14994	WESLEY GRAMMAR SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
28	10103	Business	194	297.6537	30304	WESLEY GRAMMAR SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
30	10103	General Science	158	334.1679	15125	WESLEY GRAMMAR SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
32	10103	Visual Arts	173	295.9655	9989	WESLEY GRAMMAR SCHOOL, DANSOMAN	Accra Metropolitan	-0.19711526	5.607396
33	10104	General Arts	194	298.1614	46517	HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA	Accra Metropolitan	-0.19711526	5.607396
35	10104	General Science	158	334.1679	15125	HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA	Accra Metropolitan	-0.19711526	5.607396
37	10104	Visual Arts	173	295.9655	9989	HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA	Accra Metropolitan	-0.19711526	5.607396
39	10104	Home Economics	190	280.0537	14994	HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA	Accra Metropolitan	-0.19711526	5.607396

#Exercise 3----------

```
datjss=select(datjss, -X)
datstu_jss=left_join(datstu, datjss, by="jssdistrict")
```

```
datstu_jss_sss=left_join(datstu_jss, datsss, by=c("admitted_by_schoolcode"="schoolcode"))
datstu_jss_sss$distance = sqrt((69.172*(datstu_jss_sss$ssslong-datstu_jss_sss$point_x)*cos(datstu_jss_sss$point_y/57.3))^2+(69.172*(datstu_jss_sss$ssslat-datstu_jss_sss$point_y)^2))
```

#Exercise 4----------

```
a <- data.frame(datstu$schoolcode1,datstu$choicepgm1)
b <- data.frame(datstu$schoolcode2,datstu$choicepgm2)
c <- data.frame(datstu$schoolcode3,datstu$choicepgm3)
d <- data.frame(datstu$schoolcode4,datstu$choicepgm4)
e <- data.frame(datstu$schoolcode5,datstu$choicepgm5)
f <- data.frame(datstu$schoolcode6,datstu$choicepgm6)

colnames(a)[colnames(a) == "datstu.schoolcode1"] <- "schoolcode"
colnames(a)[colnames(a) == "datstu.choicepgm1"] <- "choicepgm"
colnames(b)[colnames(b) == "datstu.schoolcode2"] <- "schoolcode"
colnames(b)[colnames(b) == "datstu.choicepgm2"] <- "choicepgm"
colnames(c)[colnames(c) == "datstu.schoolcode3"] <- "schoolcode"
colnames(c)[colnames(c) == "datstu.choicepgm3"] <- "choicepgm"
colnames(d)[colnames(d) == "datstu.schoolcode4"] <- "schoolcode"
colnames(d)[colnames(d) == "datstu.choicepgm4"] <- "choicepgm"
colnames(e)[colnames(e) == "datstu.schoolcode5"] <- "schoolcode"
colnames(e)[colnames(e) == "datstu.choicepgm5"] <- "choicepgm"
colnames(f)[colnames(f) == "datstu.schoolcode6"] <- "schoolcode"
colnames(f)[colnames(f) == "datstu.choicepgm6"] <- "choicepgm"

dataselected <- bind_rows(a,b,c,d,e,f)
numberofchoices <- unique(dataselected)
dataselected <- [!(is.na(numberofchoices$schoolcode)),]
nrow(numberofchoices)

datsss2<-datsss
datsss2<-datsss2[!(is.na(datsss2$ssslong)),]
datsss2<-datsss2[,-1]
datsss2<-as.data.frame(unique(datsss2))

datanew<-merge(x=numberofchoices,y=datsss2,by="schoolcode",all.x = TRUE, all.y = FALSE)

datstunew<-datstu
```

```
datstunew<-datstunew[!(is.na(datstunew$rankplace)),]

for (i in 1:nrow(datstunew)){
   if (datstunew$rankplace[i]==1){
      datstunew$schoolcode[i]<-datstunew$schoolcode1[i]
      datstunew$choicepgm[i]<-datstunew$choicepgm1[i]
   }
   if (datstunew$rankplace[i]==2){
      datstunew$schoolcode[i]<-datstunew$schoolcode2[i]
      datstunew$choicepgm[i]<-datstunew$choicepgm2[i]
   }
   if (datstunew$rankplace[i]==3){
      datstunew$schoolcode[i]<-datstunew$schoolcode3[i]
      datstunew$choicepgm[i]<-datstunew$choicepgm3[i]
   }
   if (datstunew$rankplace[i]==4){
      datstunew$schoolcode[i]<-datstunew$schoolcode4[i]
      datstunew$choicepgm[i]<-datstunew$choicepgm4[i]
   }
   if (datstunew$rankplace[i]==5){
      datstunew$schoolcode[i]<-datstunew$schoolcode5[i]
      datstunew$choicepgm[i]<-datstunew$choicepgm5[i]
   }
   if (datstunew$rankplace[i]==6){
      datstunew$schoolcode[i]<-datstunew$schoolcode6[i]
      datstunew$choicepgm[i]<-datstunew$choicepgm6[i]
   }
}
datstunew<-datstunew[!(datstunew$rankplace == 99),]

data_final<-datstunew %>%
   group_by(schoolcode,choicepgm) %>%
   summarise(cutoff=min(score),quality = mean(score),size = n())

data_final<-merge(x=datanew,y=data_final,by= c("schoolcode", "choicepgm"))

datanew2<-merge(x=datstunew,y=data_final,by= c("schoolcode", "choicepgm"))
datanew2<-merge(x=datanew2,y=datjss,by="jssdistrict",all.x = TRUE, all.y = FALSE)

colnames(datanew2)[colnames(data3) == "point_x"] <- "jsslong"
colnames(datanew2)[colnames(data3) == "point_y"] <- "jsslat"

datanew2$distance <- 0
```

```
datanew2<-datanew2[!(is.na(datanew2$score)),]
datanew2<-datanew2[!(is.na(datanew2$distance)),]

datanew2 %>%
  group_by(rankplace) %>%
  summarise(cutoff=min(score),quality = mean(score),distance=mean(distance))
```

```
# A tibble: 6 x 4
  rankplace cutoff quality distance
*     <int>  <int>   <dbl>    <dbl>
1         1    165    314.        0
2         2    173    302.        0
3         3    190    289.        0
4         4    185    277.        0
5         5    198    253.        0
6         6    158    251.        0
```

```
install.packages("devtools")
devtools::install_github("moodymudskipper/cutr")
library (cutr)
datanew2$quantile <- smart_cut(datanew2$score, 4, "g", output = "numeric")
datanew2$quantile <- replace(datanew2$quantile, datanew2$quantile==1, "0%-25%")
datanew2$quantile <- replace(datanew2$quantile, datanew2$quantile==2, "25%-50%")
datanew2$quantile <- replace(datanew2$quantile, datanew2$quantile==3, "50%-75%")
datanew2$quantile <- replace(datanew2$quantile, datanew2$quantile==4, "75%-100%")
datanew2 %>%
  group_by(quantile) %>%
  summarise(cutoff=min(score),quality = mean(score),distance=mean(distance))
```

```
# A tibble: 4 x 4
  quantile cutoff quality distance
* <chr>     <int>   <dbl>    <dbl>
1 0%-25%      158    237.        0
2 25%-50%     256    272.        0
3 50%-75%     289    308.        0
4 75%-100%    330    366.        0
```

 # Part2

```
rm(list = ls())
```

#Exercise 5----------

```
set.seed(123)
x1 <- runif(10000, min = 1, max = 3)
x1 <- as.matrix(x1)
x2 <- rgamma(10000, shape = 3, rate = 1/2)
x2 <- as.matrix(x2)
x3 <- rbinom(10000, 1, 0.3)
x3 <- as.matrix(x3)
epsilon <- rnorm(10000, mean=2, sd=1)
```

```
epsilon <- as.matrix(epsilon)

y <- 0.5 + 1.2*x1 - 0.9*x2 + 0.1*x3 + epsilon
ydum <- y
for (i in 1:10000){
   ydum[i] <- 0
   if (y[i]>mean(y)){
      ydum[i] <- 1
   }
}

databook <- data.frame(cbind(y,ydum,x1,x2,x3,epsilon))
names(databook)[names(databook) == "X1"] <- "y"
names(databook)[names(databook) == "X2"] <- "ydum"
names(databook)[names(databook) == "X3"] <- "x1"
names(databook)[names(databook) == "X4"] <- "x2"
names(databook)[names(databook) == "X5"] <- "x3"
names(databook)[names(databook) == "X6"] <- "epsilon"
```

#Exercise 6----------

```
#6.1
cor(y,x1)
```

 # Correlation between x1 and y is about 0.20, which very different from 1.2.

```
> cor(y,x1)
           [,1]
[1,] 0.216015
```

```
#6.2$6.3$6.4 Regression of Y on X
cons <- rep(1,10000)
X <- cbind(cons,x1,x2,x3)
beta <- solve(t(X)%*%X)%*%t(X)%*%y
rownames(beta)[1] <- 'intercept'
colnames(beta)[1] <- 'est_beta'
sigma2 <- sum((y-X%*%beta)^2)/(nrow(X)-ncol(X))
var <- sigma2*solve(t(X)%*%X)
SE_ols <- sqrt(diag(var))
SE_ols
```

```
> SE_ols
        cons
0.040620200 0.017358550 0.002876599 0.021694530
```

#Exercise 7----------

```
X <- cbind(1,x1,x2,x3)
y <- as.matrix(y)
```

```
probit_loglikelihood <- function(b., y. = ydum, X. = X){
   phi <- pnorm(X.%*%b.)
   phi[phi==1] <- 0.9999 # avoid NaN of log function
   phi[phi==0] <- 0.0001
   f <- sum(y.*log(phi))+sum((1-y.)*log(1-phi))
   f <- -f
   return(f)
}

probit <- optim(par = c(0,0,0,0), probit_loglikelihood)
probit$par
```

```
> probit$par
[1]  3.04344256  1.17216701 -0.90555423 -0.01106539
```

```
# Optimizing Logit
logit_loglikelihood <- function(b., y. = ydum,X. = X){
   gamma <- plogis(X%*%b.)
   f <- sum(y.*log(gamma))+sum((1-y.)*log(1-gamma))
   f <- -f
   return(f)
}
logit<- optim(par = c(0,0,0,0), logit_loglikelihood)
logit$par
```

```
> logit$par
[1]  5.42762617  2.10006305 -1.61854304 -0.01973273
```

```
# Optimizing Linear
linear <- lm(ydum~x1+x2+x3)
summary(linear)
linear$par = c(0.8858236, 0.1461940,-0.1028320, -0.0080531)
linear$par
```

```
> linear$par
[1]  0.8858236  0.1461940 -0.1028320 -0.0080531
```

```
estimation <- cbind(probit$par, logit$par, linear$par)
colnames(estimation) <- c("Probit", "Logit", "Linear")
rownames(estimation) <- c("intercept", "x1", "x2", "x3")
```

	Probit	Logit	Linear
intercept	3.04344256	5.42762617	0.8858236
x1	1.17216701	2.10006305	0.1461940
x2	-0.90555423	-1.61854304	-0.1028320
x3	-0.01106539	-0.01973273	-0.0080531

#Exercise 8----------

```
#Marginal Effect-probit
probit_ME <- function(df){
  result <- glm(ydum ~ x1 + x2 + x3, family=binomial(link = "probit"),df)
  ME <- mean(dnorm(X%*%coef(result)))*coef(result)
  return(ME)
}
probit_ME(databook)
```

```
> probit_ME(databook)
(Intercept)          x1          x2          x3
 0.37324175  0.14380827 -0.11106954 -0.00137997
```

```
#Marginal Effect-logit
logit_ME <- function(df){
  result <- glm(ydum ~ x1 + x2 + x3, family=binomial(link = "logit"),df)
  ME <- mean(dlogis(X%*%coef(result)))*coef(result)
  return(ME)
}
logit_ME(databook)
```

```
> logit_ME(databook)
  (Intercept)          x1          x2          x3
 0.372080184  0.144030901 -0.110975755 -0.001345977
```

```
#SE of Probit and Logit Marginal Effect by using bootstrapse
bootstrapse <- function(n,fun){
  boot_result <- data.frame(result = NA)[-1] #creating empty data.frame
  for (i in 1:n) {
    df.existing <- databook[sample(nrow(databook),size = nrow(databook),replace = T),]
    boot_result <- cbind(boot_result,fun(df.existing))
  }
  return(data.frame(SE = apply(boot_result,1,sd)))
}
bootstrapse(49,probit_ME)
bootstrapse(49,logit_ME)
```

```
> bootstrapse(49,probit_ME)
                        SE
(Intercept) 0.0097193201
x1          0.0045815902
x2          0.0005152664
x3          0.0062492739
```

```
> bootstrapse(49,logit_ME)
                        SE
(Intercept) 0.0100119341
x1          0.0047115112
x2          0.0003874686
x3          0.0068950258
```