Assignment1

Luyi Huang

2021/2/25

Exercise 1 Reading the data

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
summary(stu)
```

```
##        X              score            agey            male
##  Min.   :     1   Min.   :158.0    Min.   : 9.00   Min.   :0.000
##  1st Qu.: 85207   1st Qu.:252.0    1st Qu.:16.00   1st Qu.:0.000
##  Median :170412   Median :283.0    Median :17.00   Median :1.000
##  Mean   :170412   Mean   :291.1    Mean   :17.13   Mean   :0.549
##  3rd Qu.:255618   3rd Qu.:324.0    3rd Qu.:18.00   3rd Qu.:1.000
##  Max.   :340823   Max.   :469.0    Max.   :57.00   Max.   :1.000
##                   NA's   :179887   NA's   :650
##   schoolcode1       schoolcode2       schoolcode3       schoolcode4
##  Min.   :  10101   Min.   :  10101   Min.   :  10101   Min.   :  10101
##  1st Qu.:  21502   1st Qu.:  21502   1st Qu.:  21502   1st Qu.:  21502
##  Median :  50105   Median :  50107   Median :  50113   Median :  50202
##  Mean   : 239365   Mean   : 244223   Mean   : 264627   Mean   : 315661
##  3rd Qu.:  61201   3rd Qu.:  61202   3rd Qu.:  61202   3rd Qu.:  61203
##  Max.   :9100501   Max.   :9100501   Max.   :9100501   Max.   :9100501
##  NA's   :102       NA's   :163       NA's   :195       NA's   :406
##   schoolcode5       schoolcode6       choicepgm1         choicepgm2
##  Min.   :  10101   Min.   :  10101   Length:340823      Length:340823
##  1st Qu.:  21201   1st Qu.:  21203   Class :character   Class :character
##  Median :  50204   Median :  50204   Mode  :character   Mode  :character
##  Mean   :  47539   Mean   :  47354
##  3rd Qu.:  60801   3rd Qu.:  60704
##  Max.   :9100101   Max.   :9090401
##  NA's   :17140     NA's   :17088
##   choicepgm3         choicepgm4         choicepgm5         choicepgm6
##  Length:340823      Length:340823      Length:340823      Length:340823
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
##  jssdistrict          rankplace
##  Length:340823      Min.   : 1.00
##  Class :character   1st Qu.: 1.00
##  Mode  :character   Median : 3.00
##                     Mean   :15.45
##                     3rd Qu.: 4.00
##                     Max.   :99.00
##                     NA's   :179888
```

Overall, there are 340823 students in the dataset, there are 179886 missing score in the dataset.

```
st<-stu %>%
  filter()
dat<- data.frame(stack(stu[5:10]))
length(unique(dat$values))
```

```
## [1] 641
```

There are 641 schools.

```
dat<- data.frame(stack(stu[11:16]))
length(unique(dat$values))-1
```

```
## [1] 32
```

There are 32 programs.

```
#replace null in program with NA
stu[stu == ""] = NA
```

```
#paste to the choice
cols <- c( 'schoolcode1' , 'choicepgm1' )
# create a new column `x` with the three columns collapsed together
stu$paste1 <- apply( stu[ , cols ] , 1 , paste , collapse = "-" )
cols <- c( 'schoolcode2' , 'choicepgm2' )
# create a new column `x` with the three columns collapsed together
stu$paste2 <- apply( stu[ , cols ] , 1 , paste , collapse = "-" )

cols <- c( 'schoolcode3' , 'choicepgm3' )
# create a new column `x` with the three columns collapsed together
stu$paste3 <- apply( stu[ , cols ] , 1 , paste , collapse = "-" )

cols <- c( 'schoolcode4' , 'choicepgm4' )
# create a new column `x` with the three columns collapsed together
stu$paste4 <- apply( stu[ , cols ] , 1 , paste , collapse = "-" )

cols <- c( 'schoolcode5' , 'choicepgm5' )
# create a new column `x` with the three columns collapsed together
stu$paste5 <- apply( stu[ , cols ] , 1 , paste , collapse = "-" )

cols <- c( 'schoolcode6' , 'choicepgm6' )
# create a new column `x` with the three columns collapsed together
stu$paste6 <- apply( stu[ , cols ] , 1 , paste , collapse = "-" )
```

```
#replace choices that only have school and NA with NA
stu$paste1[which(str_sub(stu$paste1,-2,-1)=="NA") ] <- NA
stu$paste2[which(str_sub(stu$paste2,-2,-1)=="NA") ] <- NA
stu$paste3[which(str_sub(stu$paste3,-2,-1)=="NA") ] <- NA
stu$paste4[which(str_sub(stu$paste4,-2,-1)=="NA")] <- NA
stu$paste5[which(str_sub(stu$paste5,-2,-1)=="NA")] <- NA
stu$paste6[which(str_sub(stu$paste6,-2,-1)=="NA") ] <- NA
stu$paste1[which(str_sub(stu$paste1,1,2)=="NA") ] <- NA
stu$paste2[which(str_sub(stu$paste2,1,2)=="NA") ] <- NA
stu$paste3[which(str_sub(stu$paste3,1,2)=="NA")] <- NA
stu$paste4[which(str_sub(stu$paste4,1,2)=="NA")] <- NA
stu$paste5[which(str_sub(stu$paste5,1,2)=="NA")] <- NA
stu$paste6[which(str_sub(stu$paste6,1,2)=="NA") ] <- NA
```

```
dat<- data.frame(stack(stu[19:ncol(stu)]))
length(unique(unlist(na.omit((stu[19:ncol(stu)])))))
```

```
## [1] 2759
```

There are 2759 choices for students in the dataset.

```
cols=c('schoolcode1','schoolcode2','schoolcode3','schoolcode4','schoolcode5','schoolcode6')
ll=stu
ll <- cbind(ll, count1 = apply(ll[,cols], 1, function(x) sum(duplicated(na.omit(x)),na.rm=TRUE)))
count111<-ll %>%
  filter(count1>=1)
length(count111$X)
```

```
## [1] 120071
```

There are overall 120071 students that apply to the same school.

```
cols=c('paste1','paste2','paste3','paste4','paste5','paste6')
ll <- cbind(ll, count2 = apply(ll[,cols], 1, function(x) length(na.omit(x))))
count12<-ll %>%
  filter(count2<6)
length(count12$X)
```

```
## [1] 21001
```

There are 21001 students that apply to less than 6 different choices. ## Exercise 2 Data

```
stu1=read.csv("D:/ECON613/Assignments/A1/dat/datstu.csv")
stu1<- drop_na(stu1)
```

```
#adding admission information
for(i in 1:length(stu1$rankplace)) {       # for-loop over rows
  n <- stu1[i, 18]
  if(n <7){
    stu1$admission[i]<-stu1[i,n+4]
```

```
    stu1$program[i]<-stu1[i,n+10]}

  else{
       stu1$admission[i]="no admission"
       stu1$program[i]="no program"
  }
}
```

```
stu1<-stu1[which(stu1$admission!="no admission"),]
school<-stu1 %>%
  group_by(admission,program) %>%
  dplyr::summarise(cutoff=min(score),
                   size=n_distinct(X),
            quality=(mean(score)))
```

```
## `summarise()` regrouping output by 'admission' (override with `.groups` argument)
```

```
colnames(school)[1]<-"schoolcode"
head(school,20)
```

```
## # A tibble: 20 x 5
## # Groups:   schoolcode [6]
##    schoolcode program         cutoff  size quality
##    <chr>      <chr>            <int> <int>   <dbl>
##  1 100101     General Arts       198    78    244.
##  2 100101     Home Economics     199    40    229.
##  3 100101     Technical          201    49    235.
##  4 100102     Agriculture        273    87    293.
##  5 100102     Business           283    85    303.
##  6 100102     General Arts       291    86    311.
##  7 100102     General Science    273    89    299.
##  8 100102     Home Economics     262    44    279
##  9 100102     Visual Arts        250    42    273.
## 10 100104     General Arts       319    43    336.
## 11 100104     General Science    313    44    334.
## 12 100104     Home Economics     282    45    309.
## 13 100105     Business           251    76    268.
## 14 100105     General Arts       258    77    275.
## 15 100105     Home Economics     242    79    258.
## 16 100106     Agriculture        223    40    241.
## 17 100106     Business           238    39    254.
## 18 100106     General Arts       248    40    269.
## 19 100201     Business           288    76    315.
## 20 100201     General Arts       319    39    339
```

```
sss$schoolcode=as.character(sss$schoolcode)
ssu1=sss[-c(1)]
ssu1=drop_na(ssu1)
school=left_join(school, unique(ssu1), by = c("schoolcode"="schoolcode"))
head(school,20)
```

```
## # A tibble: 20 x 9
```

```
## # Groups:   schoolcode [6]
##    schoolcode program cutoff  size quality schoolname sssdistrict ssslong ssslat
##    <chr>      <chr>    <int> <int>   <dbl> <chr>      <chr>         <dbl>  <dbl>
##  1 100101     Genera~    198    78    244. WA SENIOR~ Wa Municip~   -2.29   10.0
##  2 100101     Home E~    199    40    229. WA SENIOR~ Wa Municip~   -2.29   10.0
##  3 100101     Techni~    201    49    235. WA SENIOR~ Wa Municip~   -2.29   10.0
##  4 100102     Agricu~    273    87    293. WA SENIOR~ Wa Municip~   -2.29   10.0
##  5 100102     Busine~    283    85    303. WA SENIOR~ Wa Municip~   -2.29   10.0
##  6 100102     Genera~    291    86    311. WA SENIOR~ Wa Municip~   -2.29   10.0
##  7 100102     Genera~    273    89    299. WA SENIOR~ Wa Municip~   -2.29   10.0
##  8 100102     Home E~    262    44    279  WA SENIOR~ Wa Municip~   -2.29   10.0
##  9 100102     Visual~    250    42    273. WA SENIOR~ Wa Municip~   -2.29   10.0
## 10 100104     Genera~    319    43    336. LASSIE-TU~ Wa Municip~   -2.29   10.0
## 11 100104     Genera~    313    44    334. LASSIE-TU~ Wa Municip~   -2.29   10.0
## 12 100104     Home E~    282    45    309. LASSIE-TU~ Wa Municip~   -2.29   10.0
## 13 100105     Busine~    251    76    268. ISLAMIC S~ Wa Municip~   -2.29   10.0
## 14 100105     Genera~    258    77    275. ISLAMIC S~ Wa Municip~   -2.29   10.0
## 15 100105     Home E~    242    79    258. ISLAMIC S~ Wa Municip~   -2.29   10.0
## 16 100106     Agricu~    223    40    241. T. I. AHM~ Wa Municip~   -2.29   10.0
## 17 100106     Busine~    238    39    254. T. I. AHM~ Wa Municip~   -2.29   10.0
## 18 100106     Genera~    248    40    269. T. I. AHM~ Wa Municip~   -2.29   10.0
## 19 100201     Busine~    288    76    315. NANDOM SE~ Lawra         -2.80   10.5
## 20 100201     Genera~    319    39    339  NANDOM SE~ Lawra         -2.80   10.5
```

Exercise 3 Distance

```
ssu1=unique(sss[-c(1,2)])
ssu1=drop_na(ssu1)
jsu1=drop_na(jss[-c(1)])
dis=left_join(stu1, jsu1, by = c("jssdistrict"="jssdistrict"))
```

```
dis=left_join(dis, ssu1, by = c("admission"="schoolcode"))
```

```
dis$dis=sqrt((69.172*(dis$ssslong-dis$point_x)*cos(dis$point_y/57.3))^2+(69.712*(dis$ssslat-dis$point_y
```

```
head(dis$dis,20)
```

```
##  [1]  45.60072   0.00000   0.00000   0.00000  23.08895  39.71062  55.52640
##  [8]  14.04710   0.00000  14.04710  14.04710   0.00000   0.00000   0.00000
## [15]   0.00000   0.00000 192.50774  25.91758  55.52640   0.00000
```

Exercise 4 Descriptive Characteristics

```
c=school[c(1,2,3,4,5)]
cols <- c( 'admission' , 'program' )
# create a new column `x` with the three columns collapsed together
stu1$admissionchoice <- apply( stu1[ , cols ] , 1 , paste , collapse = "-" )
cols <- c( 'schoolcode' , 'program' )
c$admissionchoice <- apply( c[ , cols ] , 1 , paste , collapse = "-" )
sch=left_join(stu1, c, by = c("admissionchoice"="admissionchoice"))
```

```r
head(sch,20)
```

```
##         X score agey male schoolcode1 schoolcode2 schoolcode3 schoolcode4
## 1  179888   249   16    0       30905       30902       30902       30903
## 2  179888   249   16    0       30905       30902       30902       30903
## 3  179890   254   19    1       31201       30403       30304       30402
## 4  179890   254   19    1       31201       30403       30304       30402
## 5  179891   277   17    0       30105       30109       30402       30403
## 6  179891   277   17    0       30105       30109       30402       30403
## 7  179892   236   16    0       31201       30304       30403       30403
## 8  179892   236   16    0       31201       30304       30403       30403
## 9  179893   237   18    1       30403       30603       30203     9030401
## 10 179893   237   18    1       30403       30603       30203     9030401
## 11 179894   262   16    0       30902       31201       31101       31202
## 12 179894   262   16    0       30902       31201       31101       31202
## 13 179895   249   15    1       10502       10502       10503       10111
## 14 179895   249   15    1       10502       10502       10503       10111
## 15 179896   229   17    0       30403       30303       30304       30302
## 16 179896   229   17    0       30403       30303       30304       30302
## 17 179897   219   18    1       30109       31201       30305       30110
## 18 179897   219   18    1       30109       31201       30305       30110
## 19 179898   227   17    0       30303       30703       30403       30504
## 20 179898   227   17    0       30303       30703       30403       30504
##    schoolcode5 schoolcode6      choicepgm1   choicepgm2   choicepgm3
## 1        30403       30801 General Science General Arts General Arts
## 2        30403       30801 General Science General Arts General Arts
## 3        30402       30303    General Arts  Agriculture General Arts
## 4        30402       30303    General Arts  Agriculture General Arts
## 5        30303       30201        Business General Arts     Business
## 6        30303       30201        Business General Arts     Business
## 7        30110       30305    General Arts General Arts General Arts
## 8        30110       30305    General Arts General Arts General Arts
## 9        30504       31204    General Arts  Visual Arts    Technical
## 10       30504       31204    General Arts  Visual Arts    Technical
## 11       30903       30403    General Arts General Arts General Arts
## 12       30903       30403    General Arts General Arts General Arts
## 13       30403       30403     Visual Arts General Arts  Visual Arts
## 14       30403       30403     Visual Arts General Arts  Visual Arts
## 15       30701       30702    General Arts General Arts General Arts
## 16       30701       30702    General Arts General Arts General Arts
## 17       30108       30403    General Arts General Arts General Arts
## 18       30108       30403    General Arts General Arts General Arts
## 19       30403       30902     Agriculture  Agriculture  Agriculture
## 20       30403       30902     Agriculture  Agriculture  Agriculture
##                  choicepgm4   choicepgm5      choicepgm6
## 1           Home Economics General Arts General Science
## 2           Home Economics General Arts General Science
## 3                 Business General Arts     Agriculture
## 4                 Business General Arts     Agriculture
## 5           Home Economics     Business    General Arts
## 6           Home Economics     Business    General Arts
## 7             General Arts General Arts    General Arts
## 8             General Arts General Arts    General Arts
```

```
## 9  Mech. Eng. Craft Pract. General Arts       Technical
## 10 Mech. Eng. Craft Pract. General Arts       Technical
## 11            General Arts     Business     Agriculture
## 12            General Arts     Business     Agriculture
## 13             Visual Arts General Arts    General Arts
## 14             Visual Arts General Arts    General Arts
## 15          Home Economics General Arts    General Arts
## 16          Home Economics General Arts    General Arts
## 17            General Arts General Arts    General Arts
## 18            General Arts General Arts    General Arts
## 19             Agriculture  Agriculture     Agriculture
## 20             Agriculture  Agriculture     Agriculture
##                              jssdistrict rankplace admission      program.x
## 1                            Agona Swedru         5     30403   General Arts
## 2                            Agona Swedru         5     30403   General Arts
## 3  Abura/Asebu/Kwamankese (Abura Dunkwa)         2     30403    Agriculture
## 4  Abura/Asebu/Kwamankese (Abura Dunkwa)         2     30403    Agriculture
## 5  Abura/Asebu/Kwamankese (Abura Dunkwa)         4     30403 Home Economics
## 6  Abura/Asebu/Kwamankese (Abura Dunkwa)         4     30403 Home Economics
## 7  Abura/Asebu/Kwamankese (Abura Dunkwa)         3     30403   General Arts
## 8  Abura/Asebu/Kwamankese (Abura Dunkwa)         3     30403   General Arts
## 9         Ajumako/Enyan/Essiam (Ajumako)         1     30403   General Arts
## 10        Ajumako/Enyan/Essiam (Ajumako)         1     30403   General Arts
## 11            Twifo Hemang (Twifo Praso)         6     30403    Agriculture
## 12            Twifo Hemang (Twifo Praso)         6     30403    Agriculture
## 13           Awutu/Efutu/Senya (Winneba)         5     30403   General Arts
## 14           Awutu/Efutu/Senya (Winneba)         5     30403   General Arts
## 15                 Mfantsiman (Saltpond)         1     30403   General Arts
## 16                 Mfantsiman (Saltpond)         1     30403   General Arts
## 17 Abura/Asebu/Kwamankese (Abura Dunkwa)         6     30403   General Arts
## 18 Abura/Asebu/Kwamankese (Abura Dunkwa)         6     30403   General Arts
## 19                 Mfantsiman (Saltpond)         3     30403    Agriculture
## 20                 Mfantsiman (Saltpond)         3     30403    Agriculture
##         admissionchoice schoolcode      program.y cutoff size  quality
## 1    30403-General Arts      30403   General Arts    208   35 242.0857
## 2    30403-General Arts      30403   General Arts    208   35 242.0857
## 3     30403-Agriculture      30403    Agriculture    219   15 241.9333
## 4     30403-Agriculture      30403    Agriculture    219   15 241.9333
## 5  30403-Home Economics      30403 Home Economics    215    8 248.3750
## 6  30403-Home Economics      30403 Home Economics    215    8 248.3750
## 7    30403-General Arts      30403   General Arts    208   35 242.0857
## 8    30403-General Arts      30403   General Arts    208   35 242.0857
## 9    30403-General Arts      30403   General Arts    208   35 242.0857
## 10   30403-General Arts      30403   General Arts    208   35 242.0857
## 11    30403-Agriculture      30403    Agriculture    219   15 241.9333
## 12    30403-Agriculture      30403    Agriculture    219   15 241.9333
## 13   30403-General Arts      30403   General Arts    208   35 242.0857
## 14   30403-General Arts      30403   General Arts    208   35 242.0857
## 15   30403-General Arts      30403   General Arts    208   35 242.0857
## 16   30403-General Arts      30403   General Arts    208   35 242.0857
## 17   30403-General Arts      30403   General Arts    208   35 242.0857
## 18   30403-General Arts      30403   General Arts    208   35 242.0857
## 19    30403-Agriculture      30403    Agriculture    219   15 241.9333
## 20    30403-Agriculture      30403    Agriculture    219   15 241.9333
```

```
dist=dis[c(1,26)]
sch=left_join(sch, dist, by = c("X"="X"))
head(sch)
```

```
##        X score agey male schoolcode1 schoolcode2 schoolcode3 schoolcode4
## 1 179888   249   16    0       30905       30902       30902       30903
## 2 179888   249   16    0       30905       30902       30902       30903
## 3 179890   254   19    1       31201       30403       30304       30402
## 4 179890   254   19    1       31201       30403       30304       30402
## 5 179891   277   17    0       30105       30109       30402       30403
## 6 179891   277   17    0       30105       30109       30402       30403
##   schoolcode5 schoolcode6      choicepgm1   choicepgm2   choicepgm3
## 1       30403       30801 General Science General Arts General Arts
## 2       30403       30801 General Science General Arts General Arts
## 3       30402       30303    General Arts  Agriculture General Arts
## 4       30402       30303    General Arts  Agriculture General Arts
## 5       30303       30201        Business General Arts     Business
## 6       30303       30201        Business General Arts     Business
##       choicepgm4   choicepgm5      choicepgm6
## 1 Home Economics General Arts General Science
## 2 Home Economics General Arts General Science
## 3       Business General Arts     Agriculture
## 4       Business General Arts     Agriculture
## 5 Home Economics     Business    General Arts
## 6 Home Economics     Business    General Arts
##                             jssdistrict rankplace admission      program.x
## 1                          Agona Swedru         5     30403   General Arts
## 2                          Agona Swedru         5     30403   General Arts
## 3 Abura/Asebu/Kwamankese (Abura Dunkwa)         2     30403    Agriculture
## 4 Abura/Asebu/Kwamankese (Abura Dunkwa)         2     30403    Agriculture
## 5 Abura/Asebu/Kwamankese (Abura Dunkwa)         4     30403 Home Economics
## 6 Abura/Asebu/Kwamankese (Abura Dunkwa)         4     30403 Home Economics
##        admissionchoice schoolcode      program.y cutoff size  quality      dis
## 1   30403-General Arts      30403   General Arts    208   35 242.0857 45.60072
## 2   30403-General Arts      30403   General Arts    208   35 242.0857 45.60072
## 3    30403-Agriculture      30403    Agriculture    219   15 241.9333  0.00000
## 4    30403-Agriculture      30403    Agriculture    219   15 241.9333  0.00000
## 5 30403-Home Economics      30403 Home Economics    215    8 248.3750  0.00000
## 6 30403-Home Economics      30403 Home Economics    215    8 248.3750  0.00000
```

```
sch %>%
    group_by(rankplace) %>%
    dplyr::summarise(cutoff_mean=mean(cutoff),
                     cutoff_sd=sd(cutoff),
                     quality_mean=mean(quality),
                     quality_sd=sd(quality),
        distance_mean=mean(dis),
              distance_sd=sd(dis) )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 7
```

```
##   rankplace cutoff_mean cutoff_sd quality_mean quality_sd distance_mean
##       <int>       <dbl>     <dbl>        <dbl>      <dbl>         <dbl>
## 1         1        282.     58.9          309.       52.3          NA
## 2         2        276.     51.2          302.       44.6          34.2
## 3         3        261.     43.8          289.       37.5          28.8
## 4         4        248.     37.7          277.       31.8          23.0
## 5         5        211.      8.18         252.       12.8          32.6
## 6         6        211.      8.19         249.       11.2          32.1
## # ... with 1 more variable: distance_sd <dbl>
```

```
sch<-sch %>% group_by(rankplace) %>%
mutate(qurtile=ntile(score, 4))
```

```
c=sch %>%
    group_by(rankplace, qurtile) %>%
    dplyr::summarise(cutoff_mean=mean(cutoff),
                     cutoff_sd=sd(cutoff),
                     quality_mean=mean(quality),
                     quality_sd=sd(quality),
        distance_mean=mean(dis),
              distance_sd=sd(dis) )
```

```
## `summarise()` regrouping output by 'rankplace' (override with `.groups` argument)
```

```
c
```

```
## # A tibble: 24 x 8
## # Groups:   rankplace [6]
##    rankplace qurtile cutoff_mean cutoff_sd quality_mean quality_sd distance_mean
##        <int>   <int>       <dbl>     <dbl>        <dbl>      <dbl>         <dbl>
##  1         1       1        224.      17.6         257.       14.1          NA
##  2         1       2        253.      26.2         282.       20.7          33.1
##  3         1       3        291.      33.9         315.       28.5          37.9
##  4         1       4        360.      36.0         380.       32.5          43.4
##  5         2       1        222.      16.4         255.       13.5          28.7
##  6         2       2        253.      23.5         281.       18.5          33.5
##  7         2       3        287.      27.7         311.       22.7          34.8
##  8         2       4        343.      28.9         362.       25.2          39.8
##  9         3       1        218.      14.3         251.       12.3          25.1
## 10         3       2        241.      21.5         270.       16.2          28.6
## # ... with 14 more rows, and 1 more variable: distance_sd <dbl>
```

Exercise 5 Data Creation

```
set.seed(11)
x_1=runif(10000, min = 1, max = 3)
x_2=rgamma(10000, shape=3,scale = 2)
x_3=rbinom(10000, size=1, prob=0.3)
epsilon=rnorm(10000,2,1)
X=cbind(x_1,x_2,x_3,epsilon)
```

```r
X1=as.data.frame(X)
```

```r
X1=X1%>% mutate(y=0.5+1.2*x_1-0.9*x_2+0.1*x_3+epsilon) %>%
       mutate(ydum = case_when(y>mean(y) ~ 1,
                               TRUE ~ 0))
```

Exercise 6 OLS

```r
#correlation
cor(X1$x_1,X1$y)
```

```
## [1] 0.2018868
```

The correlation is 0.21, which is quite different from 1.2.

```r
beta_1=cov(X1$x_1,X1$y)/var(X1$x_1)
beta_2=cov(X1$x_2,X1$y)/var(X1$x_2)
beta_3=cov(X1$x_3,X1$y)/var(X1$x_3)
alpha=mean(X1$y)-beta_1*mean(X1$x_1)-beta_2*mean(X1$x_2)-beta_3*mean(X1$x_3)
rbind(beta_1,beta_2,beta_3,alpha)
```

```
##              [,1]
## beta_1  1.1693805
## beta_2 -0.8970377
## beta_3  0.1387792
## alpha   2.5427421
```

```r
X2=X1[-c(4,5,6)]
X2$intercept=1
y=X1$y
```

```r
ma=data.matrix(X2, rownames.force = NA)
solve(t(ma) %*% ma) %*%t(ma)%*% y
```

```
##                 [,1]
## x_1        1.2063891
## x_2       -0.8983352
## x_3        0.1194532
## intercept  2.4820035
```

```r
sqrt(diag(1*solve((t(ma) %*% ma))))
```

```
##         x_1         x_2         x_3   intercept
## 0.017409264 0.002894051 0.021791159 0.040766398
```

Exercise 7 Discrete Choice

consider probit using package for check

```
X4=X1[-c(4,5)]
myprobit <- glm(ydum ~ x_1+x_2+x_3, family = binomial(link = "probit"),
    data = X4)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
myprobit
```

```
##
## Call:  glm(formula = ydum ~ x_1 + x_2 + x_3, family = binomial(link = "probit"),
##     data = X4)
##
## Coefficients:
## (Intercept)          x_1          x_2          x_3
##      2.9758       1.2068      -0.9132       0.1945
##
## Degrees of Freedom: 9999 Total (i.e. Null);  9996 Residual
## Null Deviance:       13720
## Residual Deviance: 4326  AIC: 4334
```

optim:

```
Probit_LL <- function(y,x,par) {
    Phi = pnorm(x %*% par)
    phi = dnorm(x %*% par)

    n = length(y)
    k = length(par)

    # Computing the log-likelihood
    f = sum(y*log(Phi)) + sum((1-y)*log(1-Phi))
    f = -f

    return(f)
}
Probit_LL_g <- function (y,x,par) {
    Phi = pnorm(x %*% par) # Phi is Cumulative probability
    phi = dnorm(x %*% par) # phi is Probability Density

    n = length(y)           # sample size
    k = length(par)         # number of coefficients

    g = t(matrix(rep(phi/Phi,k),nrow=n)*x) %*% y -
        t(matrix(rep(phi/(1-Phi),k),nrow=n)*x) %*% (1-y)
    g = -g

    return(g)
}
```

```
X <- as.matrix(cbind(1,X4[c(1:3)]))
Y<-as.matrix(X4[c(4)])
beta <-c( 0.1, 0.1, 0.1,0.1)
```

```r
result <- optim(par = beta, Probit_LL, y = Y, x = X, gr = Probit_LL_g,
                method = "BFGS",  hessian=TRUE)
```

```r
result
```

```
## $par
## [1]  2.9757717  1.2068172 -0.9132392  0.1945398
##
## $value
## [1] 2163.236
##
## $counts
## function gradient
##       64       16
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##            [,1]      [,2]      [,3]      [,4]
## [1,]  2152.2916  4215.439 12303.392  637.7372
## [2,]  4215.4388  8950.198 24924.221 1255.9213
## [3,] 12303.3916 24924.221 74140.958 3741.1570
## [4,]   637.7372  1255.921  3741.157  637.7372
```

consider logit

```r
mylogit <- glm(ydum ~ x_1+x_2+x_3, family = binomial(link = "logit"),
    data = X4)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
mylogit
```

```
##
## Call:  glm(formula = ydum ~ x_1 + x_2 + x_3, family = binomial(link = "logit"),
##     data = X4)
##
## Coefficients:
## (Intercept)          x_1          x_2          x_3
##      5.3383       2.1664      -1.6382       0.3394
##
## Degrees of Freedom: 9999 Total (i.e. Null);  9996 Residual
## Null Deviance:       13720
## Residual Deviance: 4340  AIC: 4348
```

optim

```
negLogLik = function(beta){
 -sum(-Y*log(1 + exp(-(X%*%beta))) - (1-Y)*log(1 + exp(X%*%beta)))
}
logistic_opt = optim(par = beta, negLogLik, hessian=TRUE, method = "BFGS")
```

```
logistic_opt
```

```
## $par
## [1]  5.3383407  2.1664523 -1.6381967  0.3394252
##
## $value
## [1] 2169.926
##
## $counts
## function gradient
##       56       16
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]      [,2]      [,3]      [,4]
## [1,]  668.8163 1310.8143  3851.550  197.6340
## [2,] 1310.8143 2785.5823  7811.702  390.9947
## [3,] 3851.5498 7811.7021 23215.909 1167.8799
## [4,]  197.6340  390.9947  1167.880  197.6340
```

linear model:

```
compCost<-function(X, y, par){
  m <- length(y)
  J <- sum((X%*%par- y)^2)/(2*m)
return(J)
}
theta<-c( 0.1, 0.1, 0.1,0.1)
```

```
optim(par = theta, fn = compCost, X = X, y = Y, method = "BFGS")
```

```
## $par
## [1]  0.88586265  0.14548432 -0.10453518  0.02844438
##
## $value
## [1] 0.05456722
##
## $counts
## function gradient
##       32       21
##
## $convergence
```

```
## [1] 0
##
## $message
## NULL
```

```
mylinear <- lm(ydum ~x_1+x_2+x_3 ,data = X4)
mylinear
```

```
##
## Call:
## lm(formula = ydum ~ x_1 + x_2 + x_3, data = X4)
##
## Coefficients:
## (Intercept)          x_1          x_2          x_3
##     0.88608      0.14541     -0.10454      0.02845
```

```
summary(mylinear)
```

```
##
## Call:
## lm(formula = ydum ~ x_1 + x_2 + x_3, data = X4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90508 -0.26342  0.05629  0.25224  1.44029
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  0.8860824  0.0134701   65.782  < 2e-16 ***
## x_1          0.1454060  0.0057524   25.278  < 2e-16 ***
## x_2         -0.1045441  0.0009563 -109.327  < 2e-16 ***
## x_3          0.0284527  0.0072003    3.952 7.82e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3304 on 9996 degrees of freedom
## Multiple R-squared:  0.5571, Adjusted R-squared:  0.557
## F-statistic:  4191 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```
summary(mylogit)
```

```
##
## Call:
## glm(formula = ydum ~ x_1 + x_2 + x_3, family = binomial(link = "logit"),
##     data = X4)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2262  -0.1380   0.0382   0.2575   2.9368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  5.33833    0.18791  28.409  < 2e-16 ***
## x_1          2.16645    0.08176  26.499  < 2e-16 ***
## x_2         -1.63819    0.03749 -43.703  < 2e-16 ***
## x_3          0.33942    0.08504   3.991 6.57e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13718.6  on 9999  degrees of freedom
## Residual deviance:  4339.9  on 9996  degrees of freedom
## AIC: 4347.9
##
## Number of Fisher Scoring iterations: 7
```

```
summary(myprobit)
```

```
##
## Call:
## glm(formula = ydum ~ x_1 + x_2 + x_3, family = binomial(link = "probit"),
##     data = X4)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5457  -0.0976   0.0075   0.2441   3.0989
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.97577    0.10044  29.628  < 2e-16 ***
## x_1          1.20682    0.04411  27.357  < 2e-16 ***
## x_2         -0.91324    0.01887 -48.393  < 2e-16 ***
## x_3          0.19454    0.04740   4.104 4.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13718.6  on 9999  degrees of freedom
## Residual deviance:  4326.5  on 9996  degrees of freedom
## AIC: 4334.5
##
## Number of Fisher Scoring iterations: 8
```

The coeffcient for x_3 in probit and logit model are both insignificant while other coeffcients are statistically significant. The model fit in linear probability model is not so good whill the other too has large deviance which means that the model of fit in these models are fine. However, the coeffcient in linear probability model is quite similiar to the original parameter we used to calculate Y. Also, the coeff for logit and probit are quite similiar but we can guess that the marginal effect of these two models will be quite similiar to the OLS and also the parameters we used to generate the data.

Exercise 8 Marginal Effects

marginal effect of probit

```
fav = mean(dnorm(predict(myprobit, type = "link")))
marg =  as.matrix(fav * coef(myprobit))
marg
```

```
##                    [,1]
## (Intercept)  0.35735086
## x_1          0.14492279
## x_2         -0.10966795
## x_3          0.02336167
```

```
gr = as.numeric(dnorm(predict(myprobit, type = "link")))
vcv = solve(result$hessian)
se = sqrt(t(marg) %*% vcv %*% marg)
se
```

```
##            [,1]
## [1,] 0.03595188
```

marginal effect of logit

```
fav = mean(dnorm(predict(mylogit, type = "link")))
marg =  as.matrix(fav * coef(mylogit))
marg
```

```
##                    [,1]
## (Intercept)  0.36546907
## x_1          0.14831798
## x_2         -0.11215291
## x_3          0.02323748
```

the marginal effect on probability:

```
library(mfx)
```

```
## Warning: package 'mfx' was built under R version 4.0.3
```

```
## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 4.0.3
```

```
## Loading required package: lmtest
```

```
## Warning: package 'lmtest' was built under R version 4.0.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.0.3
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: betareg

## Warning: package 'betareg' was built under R version 4.0.3
```

```
pro=probitmfx(formula = ydum ~ .,data = X4, atmean = FALSE)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

SE for probit

```
pro$mfxest
```

```
##           dF/dx    Std. Err.           z         P>|z|
## x_1  0.14492279 0.0044036440   32.909742 1.594607e-237
## x_2 -0.10966795 0.0004359352 -251.569368  0.000000e+00
## x_3  0.02323686 0.0056120188    4.140553  3.464699e-05
```

SE for logit:

```
log=logitmfx(formula = ydum~ x_1+x_2+x_3,data = X4, atmean = FALSE)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
log$mfxest
```

```
##           dF/dx   Std. Err.          z         P>|z|
## x_1  0.14489588 0.007692872  18.835084  3.895056e-79
## x_2 -0.10956524 0.004795005 -22.849868 1.465798e-115
## x_3  0.02258671 0.005607601   4.027874  5.628340e-05
```

marginal effect and se for probit:

```
Y=X4$ydum
xm = as.matrix(colMeans(X4))
be=as.matrix(result$par)
x1=as.matrix(cbind(1,X4[c(1:3)]))
fxb= mean(dnorm(x1 %*% as.matrix(result$par)))
mfx = data.frame(mfx=fxb*as.matrix(result$par), se=NA)
vcv = solve(result$hessian)
temp1 = apply(x1,2,function(x)length(table(x))==2)
disch = names(temp1[temp1==TRUE])
k1=4
gr = apply(x1, 1, function(x){
      as.numeric(as.numeric(dnorm(x %*% be))*(diag(k1) - as.numeric(x %*% be)*(be %*% t(x))))
    })
gr = matrix(apply(gr,1,mean),nrow=k1)
mfx$se = sqrt(diag(gr %*% vcv %*% t(gr)))
mfx
```

```
##           mfx           se
## 1  0.35735084 0.0096590700
## 2  0.14492279 0.0044072576
## 3 -0.10966795 0.0004366261
## 4  0.02336166 0.0056647939
```

marginal effect and se for logit

```
Y=X4$ydum
xm = as.matrix(colMeans(X4))
be=as.matrix(logistic_opt$par)
x1=as.matrix(cbind(1,X4[c(1:3)]))
fxb= mean(dnorm(x1 %*% as.matrix(logistic_opt$par)))
mfx = data.frame(mfx=fxb*as.matrix(logistic_opt$par), se=NA)
vcv = solve(logistic_opt$hessian)
temp1 = apply(x1,2,function(x)length(table(x))==2)
disch = names(temp1[temp1==TRUE])
k1=4
gr = apply(x1, 1, function(x){
      as.numeric(as.numeric(dnorm(x %*% be))*(diag(k1) - as.numeric(x %*% be)*(be %*% t(x))))
    })
gr = matrix(apply(gr,1,mean),nrow=k1)
mfx$se = sqrt(diag(gr %*% vcv %*% t(gr)))
mfx
```

```
##           mfx           se
## 1  0.36546922 0.0100230445
## 2  0.14831793 0.0044246150
## 3 -0.11215292 0.0004713743
## 4  0.02323746 0.0057935011
```

They results are quite similiar compared with the package. We can also see that the margianl effect is quite similar to the parameter we used to generate the data.