Econ 613 Assignment 1

Xiaolin Ding

28/02/21

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
datstu = read.csv("file:///Users/DXL/Desktop/Econ613/datstu.csv")
datjss = read.csv("file:///Users/DXL/Desktop/Econ613/datjss.csv")
datsss = read.csv("file:///Users/DXL/Desktop/Econ613/datsss.csv")
```

Part 1 Missing Data

Exercise 1 Missing Data

Number of students

```
length(datstu[,1])
```

```
## [1] 340823
```

Number of schools

```
length(unique(unlist(datstu[,5:10])))
```

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

Number of programs

```
length(unique(unlist(datstu[,11:16])))
```

```
## [1] 33
```

Number of choices

```
df = rbind(setNames(datstu[,c(5,11)], c("schoolcode","choicepgm")),
           setNames(datstu[,c(6,12)], c("schoolcode","choicepgm")),
           setNames(datstu[,c(7,13)], c("schoolcode","choicepgm")),
           setNames(datstu[,c(8,14)], c("schoolcode","choicepgm")),
           setNames(datstu[,c(9,15)], c("schoolcode","choicepgm")),
```

```
            setNames(datstu[,c(10,16)], c("schoolcode","choicepgm")))
dim(df %>% group_by_all %>% summarise())[1]
```

```
## `summarise()` has grouped output by 'schoolcode'. You can override using the `.groups` argument.
```

```
## [1] 3086
```

Missing test score

```
sum(is.na(datstu[,2]))
```

```
## [1] 179887
```

Apply to the same school (different programs)

```
count = 0
num = 1:dim(datstu)[1]
for (i in num) {
  if(sum(duplicated(datstu[i,5:10]))>0)  count = count+1
}
print(count)
```

```
## [1] 0
```

Apply to less than 6 choices

```
count = 0
for (i in 1:dim(datstu)[1]) {
  if(sum(is.na(datstu[i,5:10]))>0)  count = count+1
}
print(count)
```

```
## [1] 17734
```

Exercise 2 Data

df2 is the required school level dataset.

```
rankindex = which(datstu[,18]<=6)
rankplace = datstu[rankindex,18]
ssscode = c()
choicepgm = c()
score = c()
jssname = c()
for (i in 1:length(rankplace)){
  ssscode = append(ssscode, datstu[rankindex[i],rankplace[i]+4])
  choicepgm = append(choicepgm, toString(datstu[rankindex[i], rankplace[i]+10]))
  score = append(score, datstu[rankindex[i],2])
  jssname = append(jssname, toString(datstu[rankindex[i],17]))
}
df1 = data.frame(ssscode, choicepgm, score, jssname, rankplace)
summary = df1 %>% group_by(ssscode, choicepgm) %>%
  summarise(cutoff = min(score), quality = mean(score), size = n())
```

```
## `summarise()` has grouped output by 'ssscode'. You can override using the `.groups` argument.
```

```
ssscode = summary %>% pull(ssscode)
df2 = data.frame(ssscode)
```

```r
df2$choicepgm = summary %>% pull(choicepgm)
df2$sssname = datsss[,2][match(df2[,1], datsss[,3])]
df2$sssdistrict = datsss[,4][match(df2[,1], datsss[,3])]
df2$ssslon = datsss[,5][match(df2[,1], datsss[,3])]
df2$ssslat = datsss[,6][match(df2[,1], datsss[,3])]
df2$cutoff = summary %>% pull(cutoff)
df2$quality = summary %>% pull(quality)
df2$size = summary %>% pull(size)
df2[1:20,]
```

```
##    ssscode       choicepgm                                 sssname
## 1    10101     Agriculture        EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN
## 2    10101        Business        EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN
## 3    10101    General Arts        EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN
## 4    10101 General Science        EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN
## 5    10101  Home Economics        EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN
## 6    10101     Visual Arts        EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN
## 7    10102    General Arts   ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO
## 8    10102 General Science   ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO
## 9    10102  Home Economics   ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO
## 10   10102     Visual Arts   ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO
## 11   10103     Agriculture               WESLEY GRAMMAR SCHOOL, DANSOMAN
## 12   10103        Business               WESLEY GRAMMAR SCHOOL, DANSOMAN
## 13   10103    General Arts               WESLEY GRAMMAR SCHOOL, DANSOMAN
## 14   10103 General Science               WESLEY GRAMMAR SCHOOL, DANSOMAN
## 15   10103  Home Economics               WESLEY GRAMMAR SCHOOL, DANSOMAN
## 16   10103     Visual Arts               WESLEY GRAMMAR SCHOOL, DANSOMAN
## 17   10104    General Arts HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA
## 18   10104 General Science HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA
## 19   10104  Home Economics HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA
## 20   10104     Visual Arts HOLY TRINITY CATHEDRAL SENIOR HIGH SCH, ACCRA
##           sssdistrict     ssslon   ssslat cutoff  quality size
## 1  Accra Metropolitan -0.1971153 5.607396    288 310.1429   49
## 2  Accra Metropolitan -0.1971153 5.607396    305 324.8600  100
## 3  Accra Metropolitan -0.1971153 5.607396    316 330.0900  100
## 4  Accra Metropolitan -0.1971153 5.607396    299 329.1000   50
## 5  Accra Metropolitan -0.1971153 5.607396    284 300.5714   49
## 6  Accra Metropolitan -0.1971153 5.607396    296 311.5400   50
## 7  Accra Metropolitan -0.1971153 5.607396    388 404.9773   88
## 8  Accra Metropolitan -0.1971153 5.607396    389 406.4143   70
## 9  Accra Metropolitan -0.1971153 5.607396    363 377.1111   45
## 10 Accra Metropolitan -0.1971153 5.607396    343 370.9333   45
## 11 Accra Metropolitan -0.1971153 5.607396    316 333.1316   38
## 12 Accra Metropolitan -0.1971153 5.607396    341 357.9664  119
## 13 Accra Metropolitan -0.1971153 5.607396    349 362.5812  117
## 14 Accra Metropolitan -0.1971153 5.607396    335 353.5625   80
## 15 Accra Metropolitan -0.1971153 5.607396    320 336.0408   49
## 16 Accra Metropolitan -0.1971153 5.607396    343 357.9500   40
## 17 Accra Metropolitan -0.1971153 5.607396    302 320.1273   55
## 18 Accra Metropolitan -0.1971153 5.607396    245 283.3636   55
## 19 Accra Metropolitan -0.1971153 5.607396    264 285.8545   55
## 20 Accra Metropolitan -0.1971153 5.607396    273 298.3273   55
```

Exercise 3 Distance

df3 is the required dataset for the distance between junior high school and senior high school.

```
jssname = unique(datjss[,2])
sssname = unique(datsss[,4])
jsslon = datjss[,3][match(jssname, datjss[,2])]
jsslat = datjss[,4][match(jssname, datjss[,2])]
ssslon = datsss[,5][match(sssname, datsss[,4])]
ssslat = datsss[,6][match(sssname, datsss[,4])]
dist = c()
jssandsss = c()
for (i in 1:length(jssname)){
  for (j in 1:length(sssname)){
    d = sqrt((69.172*(ssslon[j]-jsslon[i])*cos(jsslat[i]/57.3))^2
             +(69.172*(ssslat[j])-jsslat[i]))^2
    dist = append(dist, d)
    jssandsss = append(jssandsss, paste(toString(jssname[i]), "&",
                                        toString(sssname[j])))
  }
}
df3 = data.frame(jssandsss, dist)
df3[1:20,]
```

```
##                                                       jssandsss       dist
## 1                   South Dayi (Kpeve) & Cape Coast Municipal 11185.7177
## 2                  South Dayi (Kpeve) & Kwahu South (Mpraeso)  3811.1231
## 3                     South Dayi (Kpeve) & Ga West (Amasaman)  2116.0197
## 4                 South Dayi (Kpeve) & Akwapim South (Nsawam)  1466.8271
## 5                           South Dayi (Kpeve) & Kumasi Metro 15849.4046
## 6                     South Dayi (Kpeve) & Accra Metropolitan  1155.6714
## 7   South Dayi (Kpeve) & Shama/Ahanta/East (Sekondi/Takoradi) 16193.4655
## 8                     South Dayi (Kpeve) & Kwaebibirem (Kade)  5206.8809
## 9                  South Dayi (Kpeve) & Mfantsiman (Saltpond)  7319.2100
## 10                               South Dayi (Kpeve) & Sunyani 30871.8245
## 11                South Dayi (Kpeve) & New Juaben (Koforidua)  1622.3182
## 12              South Dayi (Kpeve) & Akwapim North (Akropong)  1066.9940
## 13                          South Dayi (Kpeve) & Ho Municipal   937.7219
## 14                South Dayi (Kpeve) & Sekyere West (Mampong)  9592.1997
## 15 South Dayi (Kpeve) & Abura/Asebu/Kwamankese (Abura Dunkwa)  9673.5341
## 16                                  South Dayi (Kpeve) & Tema   701.1159
## 17           South Dayi (Kpeve) & Awutu/Efutu/Senya (Winneba)  2801.6951
## 18  South Dayi (Kpeve) & Bosomtwe/Atwima/Kwanwoma (Kuntanase) 15259.1070
## 19                                South Dayi (Kpeve) & Kpando   487.5670
## 20                     South Dayi (Kpeve) & Asutifi (Kenyasi) 35512.3207
```

Exercise 4 Descriptive Characteristics

df5 is the required dataset differentiating by ranked choice.

```
df4 = df1
df4$sssname = datsss[,4][match(df4[,1], datsss[,3])]
jssandsss = c()
for (i in 1:length(df4$jssname)){
```

```
  jssandsss = append(jssandsss, paste(toString(df4$jssname[i]), "&",
                                       toString(df4$sssname[i])))
}
df4$jssandsss = jssandsss
df4$dist = df3[,2][match(df4$jssandsss, df3[,1])]
summary1 = df4 %>% group_by(rankplace) %>%
  summarise(cutoff = min(score), qualitymean = mean(score),
            qualitysd = sd(score), distmean = mean(dist),
            distsd = sd(dist))
rankplace = summary1 %>% pull(rankplace)
df5 = data.frame(rankplace)
df5$cutoff = summary1 %>% pull(cutoff)
df5$qualitymean = summary1 %>% pull(qualitymean)
df5$qualitysd = summary1 %>% pull(qualitysd)
df5$distmean = summary1 %>% pull(distmean)
df5$distsd = summary1 %>% pull(distsd)
df5
```

```
##   rankplace cutoff qualitymean qualitysd distmean   distsd
## 1         1    165    313.6368  56.41016       NA       NA
## 2         2    173    302.4478  49.04344 1639.649 3330.171
## 3         3    190    288.6138  42.41799 1388.500 2936.618
## 4         4    185    276.7714  37.50909 1207.323 2722.688
## 5         5    198    252.7439  30.44706 1304.525 2404.929
## 6         6    158    251.1727  28.94855 1250.332 2149.005
```

df6 is the required dataset differentiating by student test score quantiles.

```
summary2 = df4 %>%
  summarise(quantile = quantile(score, c(0.25, 0.5, 0.75)))
quantile = summary2 %>% pull(quantile)
scorequantile = c()
for (i in 1:length(df4$score)){
  if (df4$score[i]<=quantile[1]){
    scorequantile[i] = "25th"
  }
  else if (df4$score[i]>quantile[1] && df4$score[i]<=quantile[2]){
    scorequantile[i] = "25th-50th"
  }
  else if (df4$score[i]>quantile[2] && df4$score[i]<=quantile[3]){
    scorequantile[i] = "50th-75th"
  }
  else {
    scorequantile[i] = "75th-100th"
  }
}
df4$scorequantile = scorequantile
summary3 = df4 %>% group_by(scorequantile) %>%
  summarise(cutoff = min(score), qualitymean = mean(score),
            qualitysd = sd(score), distmean = mean(dist),
            distsd = sd(dist))
scorequantile = summary3 %>% pull(scorequantile)
df6 = data.frame(scorequantile)
df6$cutoff = summary3 %>% pull(cutoff)
df6$qualitymean = summary3 %>% pull(qualitymean)
```

```
df6$qualitysd = summary3 %>% pull(qualitysd)
df6$distmean = summary3 %>% pull(distmean)
df6$distsd = summary3 %>% pull(distsd)
df6
```

```
##   scorequantile cutoff qualitymean qualitysd distmean   distsd
## 1          25th    158    237.5496 12.809987       NA       NA
## 2     25th-50th    257    272.7115  9.477293 1396.992 3168.023
## 3     50th-75th    290    308.5783 11.720250 1433.082 2922.843
## 4    75th-100th    331    366.6053 27.260338 1893.068 3206.596
```

Part 2 Data Creation

Exercise 5 Data Creation

```
x1 = runif(10000,1,3)
x2 = rgamma(10000,shape=3,scale=2)
x3 = rbinom(10000,size=1,prob=0.3)
epsilon = rnorm(10000,2,1)
y = 0.5 + 1.2*x1 - 0.9*x2 + 0.1*x3 + epsilon
ydum = as.numeric(y>mean(y))
```

Exercise 6 OLS

The correlation between y and x1

```
cor(x1, y)
```

```
## [1] 0.2084384
```

The correlation between Y and X1 is 0.2, which has the same sign as 1.2.

Creat matrices X and Y

```
x = as.matrix(cbind(x1, x2, x3))
intercept <- rep(1, nrow(x))
Y = as.matrix(y)
X = as.matrix(cbind(intercept, x))
```

Calculate the coefficients on this regression

```
betas = solve(t(X) %*% X) %*% t(X) %*% Y
betas
```

```
##                  [,1]
## intercept  2.50645367
## x1         1.21115528
## x2        -0.90151763
## x3         0.07486742
```

Calculate the standard errors using the standard formulas of the OLS

```
residuals = Y - X %*% betas
p = ncol(X) - 1
df = nrow(X) - p - 1
```

```
res_var = sum(residuals^2) / df
beta_cov = res_var * solve(t(X) %*% X)
beta_se = sqrt(diag(beta_cov))
beta_se
```

```
##   intercept          x1          x2          x3
## 0.040504297 0.017368339 0.002871835 0.021816527
```

Exercise 7 Discrete Choice

Probit Model

```
probit = glm(ydum ~ x1 + x2 + x3, family = binomial(link = "probit"))
```

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

```
summary(probit)
```

```
##
## Call:
## glm(formula = ydum ~ x1 + x2 + x3, family = binomial(link = "probit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3809  -0.0897   0.0078   0.2397   3.2479
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.97659    0.09863  30.180   <2e-16 ***
## x1           1.24721    0.04539  27.475   <2e-16 ***
## x2          -0.91105    0.01875 -48.596   <2e-16 ***
## x3           0.04875    0.04834   1.008    0.313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13685.5  on 9999  degrees of freedom
## Residual deviance:  4210.7  on 9996  degrees of freedom
## AIC: 4218.7
##
## Number of Fisher Scoring iterations: 8
```

Both x1 and x3 increase the probability that ydum = 1, while x2 decreases the probability that ydum = 1. Only x3 is not significant.

Logit Model

```
logit = glm(ydum ~ x1 + x2 + x3, family = binomial(link = "logit"))
```

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

```
summary(logit)
```

```
##
## Call:
## glm(formula = ydum ~ x1 + x2 + x3, family = binomial(link = "logit"))
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.11999  -0.13039   0.03858   0.25393   3.03619
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.31169    0.18465  28.767   <2e-16 ***
## x1           2.25453    0.08455  26.666   <2e-16 ***
## x2          -1.63647    0.03744 -43.705   <2e-16 ***
## x3           0.10035    0.08704   1.153    0.249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13685.5  on 9999  degrees of freedom
## Residual deviance:  4226.9  on 9996  degrees of freedom
## AIC: 4234.9
##
## Number of Fisher Scoring iterations: 7
```

Both x1 and x3 increase the probability that ydum $= 1$, while x2 decreases the probability that ydum $= 1$. Only x3 is not significant.

Linear Model

```r
linear = lm(y ~ x1 + x2 + x3)
summary(linear)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2603 -0.6476 -0.0076  0.6610  3.9310
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  2.506454   0.040504   61.881  < 2e-16 ***
## x1           1.211155   0.017368   69.734  < 2e-16 ***
## x2          -0.901518   0.002872 -313.917  < 2e-16 ***
## x3           0.074867   0.021817    3.432 0.000602 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9984 on 9996 degrees of freedom
## Multiple R-squared:  0.9119, Adjusted R-squared:  0.9119
## F-statistic: 3.45e+04 on 3 and 9996 DF,  p-value: < 2.2e-16
```

A unit increase in x1 decreases y by

```r
summary(linear)$coefficients[2,1]
```

```
## [1] 1.211155
```

A unit increase in x2 decreases y by

```
summary(linear)$coefficients[3,1]
```

```
## [1] -0.9015176
```

A unit increase in x3 increases y by

```
summary(linear)$coefficients[4,1]
```

```
## [1] 0.07486742
```

All the estimated coefficients are significant.

Exercise 8 Marginal Effects

Probit Model

```
probit_flike = function(par,x1,x2,x3,ydum)
{
  xbeta = par[1] + par[2]*x1 + par[3]*x2 + par[4]*x3
  pr = pnorm(xbeta)
  pr[pr>0.999999] = 0.999999
  pr[pr<0.000001] = 0.000001
  like = ydum*log(pr) + (1-ydum)*log(1-pr)
  return(-sum(like))
}
ntry = 100
out = mat.or.vec(ntry,4)
for (i0 in 1:ntry)
{
start = runif(4,-10,10)
probit_res = optim(start,fn=probit_flike,method="BFGS",control=list(trace=0,maxit=1000),x1=x1,x2=x2,x3=
out[i0,] = probit_res$par
}
start = runif(4)
probit_res = optim(start,fn=probit_flike,method="BFGS",control=list(trace=0,maxit=1000),x1=x1,x2=x2,x3=
fisher_info = solve(probit_res$hessian)
probit_sigma = sqrt(diag(fisher_info))
```

The marginal effect of x1 is

```
probit_res$par[2]
```

```
## [1] 1.247215
```

The standard error of the marginal effect of x1 is

```
probit_sigma[2]
```

```
## [1] 0.04529426
```

The marginal effect of x2 is

```
probit_res$par[3]
```

```
## [1] -0.9110562
```

The standard error of the marginal effect of x2 is

```
probit_sigma[3]
```

```
## [1] 0.01876452
```

The marginal effect of x3 is

```
probit_res$par[4]
```

```
## [1] 0.04875106
```

The standard error of the marginal effect of x3 is

```
probit_sigma[4]
```

```
## [1] 0.04815612
```

Logit Model

```
logit_flike = function(par,x1,x2,x3,ydum)
{
  xbeta = par[1] + par[2]*x1 + par[3]*x2 + par[4]*x3
  pr = exp(xbeta)/(1+exp(xbeta))
  pr[pr>0.999999] = 0.999999
  pr[pr<0.000001] = 0.000001
  like = ydum*log(pr) + (1-ydum)*log(1-pr)
  return(-sum(like))
}
ntry = 100
out = mat.or.vec(ntry,4)
for (i0 in 1:ntry)
{
start = runif(4,-10,10)
logit_res = optim(start,fn=logit_flike,method="BFGS",control=list(trace=0,maxit=1000),x1=x1,x2=x2,x3=x3
out[i0,] = logit_res$par
}
start = runif(4)
logit_res = optim(start,fn=logit_flike,method="BFGS",control=list(trace=0,maxit=1000),x1=x1,x2=x2,x3=x3
fisher_info = solve(logit_res$hessian)
logit_sigma = sqrt(diag(fisher_info))
```

The marginal effect of x1 is

```
logit_res$par[2]
```

```
## [1] 2.254552
```

The standard error of the marginal effect of x1 is

```
logit_sigma[2]
```

```
## [1] 0.08455086
```

The marginal effect of x2 is

```
logit_res$par[3]
```

```
## [1] -1.63648
```

The standard error of the marginal effect of x2 is

```r
logit_sigma[3]
```

```
## [1] 0.03744665
```

The marginal effect of x3 is

```r
logit_res$par[4]
```

```
## [1] 0.100355
```

The standard error of the marginal effect of x3 is

```r
logit_sigma[4]
```

```
## [1] 0.0870449
```