Econ 613 A1

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(knitr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.1
## v readr   1.4.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
datstu = read.csv("datstu.csv")
datjss = read.csv("datjss.csv")
datsss = read.csv("datsss.csv")
```

EX1

```
# number of students
nrow(datstu)
```

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

```
# number of schools
schoolcodes = select(datstu, starts_with("schoolcode"))
n_distinct(flatten(schoolcodes), na.rm = TRUE)
```

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

```
# number of programs
programs = select(datstu, starts_with("choicepgm"))
n_distinct(flatten(programs), na.rm = TRUE)
```

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

```
# number of choices
schools_all = flatten(schoolcodes)
```

```r
programs_all = flatten(programs)

sp1 = datstu %>% group_by(schoolcode1,choicepgm1) %>% summarise(Count = n())
```

```
## `summarise()` has grouped output by 'schoolcode1'. You can override using the `.groups` argument.
```

```r
sp2 = datstu %>% group_by(schoolcode2,choicepgm2) %>% summarise(Count = n())
```

```
## `summarise()` has grouped output by 'schoolcode2'. You can override using the `.groups` argument.
```

```r
sp3 = datstu %>% group_by(schoolcode3,choicepgm3) %>% summarise(Count = n())
```

```
## `summarise()` has grouped output by 'schoolcode3'. You can override using the `.groups` argument.
```

```r
sp4 = datstu %>% group_by(schoolcode4,choicepgm4) %>% summarise(Count = n())
```

```
## `summarise()` has grouped output by 'schoolcode4'. You can override using the `.groups` argument.
```

```r
sp5 = datstu %>% group_by(schoolcode5,choicepgm5) %>% summarise(Count = n())
```

```
## `summarise()` has grouped output by 'schoolcode5'. You can override using the `.groups` argument.
```

```r
sp6 = datstu %>% group_by(schoolcode6,choicepgm6) %>% summarise(Count = n())
```

```
## `summarise()` has grouped output by 'schoolcode6'. You can override using the `.groups` argument.
```

```r
sp = bind_rows(sp1, sp2, sp3, sp4, sp5, sp6)
num_sp = sp %>% group_by(schoolcode1, choicepgm1) %>% summarise()
```

```
## `summarise()` has grouped output by 'schoolcode1'. You can override using the `.groups` argument.
```

```r
num_sp = num_sp %>% rename(schoolcode = schoolcode1, choicepgm = choicepgm1)
nrow(num_sp)
```

```
## [1] 2546
```

```r
# number of missing test score
sum(is.na(datstu$score))
```

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

```r
# number of students who apply to the same school(different programs)
same_school = schoolcodes %>% filter(schoolcode1 == schoolcode2 |
                                     schoolcode1 == schoolcode3|
                                     schoolcode1 == schoolcode4|
                                     schoolcode1 == schoolcode5|
                                     schoolcode1 == schoolcode6|
                                     schoolcode2 == schoolcode3|
                                     schoolcode2 == schoolcode4|
                                     schoolcode2 == schoolcode5|
                                     schoolcode2 == schoolcode6|
                                     schoolcode3 == schoolcode4|
                                     schoolcode3 == schoolcode5|
                                     schoolcode3 == schoolcode6|
                                     schoolcode4 == schoolcode5|
                                     schoolcode4 == schoolcode6|
                                     schoolcode5 == schoolcode6
                                     )
nrow(same_school)
```

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

```r
# apply to less than 6 choices
less = schoolcodes %>% filter(is.na(schoolcode1)| is.na(schoolcode2)
                       | is.na(schoolcode3) | is.na(schoolcode4)|
                         is.na(schoolcode5) |is.na(schoolcode6))
nrow(less)
```

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

EX2

```r
ssss = datsss %>% select(schoolcode, sssdistrict, ssslong, ssslat)%>% distinct()
ssss = ssss[complete.cases(ssss), ]
school_data = left_join(num_sp, ssss, by = "schoolcode")

rank1 = datstu %>% filter(rankplace == 1)
rank1 = rank1 %>% select(X, agey, male, score,
                         schoolcode1,
                         choicepgm1,
                         jssdistrict)%>% rename(ad.schoolcode = schoolcode1,
                                                ad.choicepgm = choicepgm1)

rank2 = datstu %>% filter(rankplace == 2)
rank2 = rank2 %>% select(X,agey, male,
                         score, schoolcode2,
                         choicepgm2, jssdistrict) %>% rename(
                           ad.schoolcode = schoolcode2,
                           ad.choicepgm = choicepgm2)

rank3 = datstu %>% filter(rankplace == 3)
rank3 = rank3 %>% select(X, agey, male,
                         score, schoolcode3,
                         choicepgm3, jssdistrict)%>% rename(
                           ad.schoolcode = schoolcode3,
                           ad.choicepgm = choicepgm3)

rank4 = datstu %>% filter(rankplace == 4)
rank4 = rank4 %>% select(X, agey, male,
                         score, schoolcode4,
                         choicepgm4, jssdistrict)%>% rename(
                           ad.schoolcode = schoolcode4,
                           ad.choicepgm = choicepgm4)

rank5 = datstu %>% filter(rankplace == 5)
rank5 = rank5 %>% select(X, agey, male,
                         score, schoolcode5,
                         choicepgm5, jssdistrict)%>% rename(
                           ad.schoolcode = schoolcode5,
                           ad.choicepgm = choicepgm5)

rank6 = datstu %>% filter(rankplace == 6)
rank6 = rank6 %>% select(X, agey, male,
                         score, schoolcode6,
```

```
                         choicepgm6, jssdistrict)%>% rename(
                           ad.schoolcode = schoolcode6,
                           ad.choicepgm = choicepgm6)

admitted = bind_rows(rank1, rank2, rank3, rank4, rank5, rank6)
addata = admitted %>% group_by(ad.schoolcode,
         ad.choicepgm) %>% summarise(cutoff = min(score),
                                     quality = mean(score), size = n())
```

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

```
school_data = left_join(school_data, addata, by = c("schoolcode" = "ad.schoolcode",
                                                    "choicepgm" = "ad.choicepgm"))
```

Table 1: School level data

schoolcode	choicepgm	sssdistrict	ssslong	ssslat	cutoff	quality	size
10101	Agriculture	Accra Metropolitan	-0.1971153	5.607396	288	310.1429	49
10101	Business	Accra Metropolitan	-0.1971153	5.607396	305	324.8600	100
10101	General Arts	Accra Metropolitan	-0.1971153	5.607396	316	330.0900	100
10101	General Science	Accra Metropolitan	-0.1971153	5.607396	299	329.1000	50
10101	Home Economics	Accra Metropolitan	-0.1971153	5.607396	284	300.5714	49
10101	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	296	311.5400	50
10102	General Arts	Accra Metropolitan	-0.1971153	5.607396	388	404.9773	88
10102	General Science	Accra Metropolitan	-0.1971153	5.607396	389	406.4143	70
10102	Home Economics	Accra Metropolitan	-0.1971153	5.607396	363	377.1111	45
10102	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	343	370.9333	45
10103	Agriculture	Accra Metropolitan	-0.1971153	5.607396	316	333.1316	38
10103	Business	Accra Metropolitan	-0.1971153	5.607396	341	357.9664	119
10103	General Arts	Accra Metropolitan	-0.1971153	5.607396	349	362.5812	117
10103	General Science	Accra Metropolitan	-0.1971153	5.607396	335	353.5625	80
10103	Home Economics	Accra Metropolitan	-0.1971153	5.607396	320	336.0408	49
10103	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	343	357.9500	40
10104	General Arts	Accra Metropolitan	-0.1971153	5.607396	302	320.1273	55
10104	General Science	Accra Metropolitan	-0.1971153	5.607396	245	283.3636	55
10104	Home Economics	Accra Metropolitan	-0.1971153	5.607396	264	285.8545	55
10104	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	273	298.3273	55

EX3

```
admitted = left_join(admitted, datjss%>%select(-X), by = "jssdistrict")

admitted = left_join(admitted, ssss%>% select(-sssdistrict),
                     by = c("ad.schoolcode"="schoolcode"))

admitted = admitted %>% rename(jsslong = point_x, jsslat = point_y)

admitted = admitted %>% mutate(distance= sqrt(
  (69.172 * (ssslong - jsslong)* cos(jsslat/57.3))^2 +
    (69.172*(ssslat - jsslat))^2 ) )

dis = admitted %>% group_by(ad.schoolcode, ad.choicepgm) %>%
  summarise(avg_distance = mean(distance))
```

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

```
school_dis = left_join(school_data, dis, by = c("schoolcode" = "ad.schoolcode",
                                                 "choicepgm" = "ad.choicepgm"))
```

Table 2: Schools distances

schoolcode	choicepgm	sssdistrict	ssslong	ssslat	cutoff	quality	size	avg_distance
10101	Agriculture	Accra Metropolitan	-0.1971153	5.607396	288	310.1429	49	20.105568
10101	Business	Accra Metropolitan	-0.1971153	5.607396	305	324.8600	100	7.570936
10101	General Arts	Accra Metropolitan	-0.1971153	5.607396	316	330.0900	100	7.587697
10101	General Science	Accra Metropolitan	-0.1971153	5.607396	299	329.1000	50	11.305599
10101	Home Economics	Accra Metropolitan	-0.1971153	5.607396	284	300.5714	49	7.282156
10101	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	296	311.5400	50	13.431514
10102	General Arts	Accra Metropolitan	-0.1971153	5.607396	388	404.9773	88	12.408223
10102	General Science	Accra Metropolitan	-0.1971153	5.607396	389	406.4143	70	18.022590
10102	Home Economics	Accra Metropolitan	-0.1971153	5.607396	363	377.1111	45	9.339753
10102	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	343	370.9333	45	14.156339
10103	Agriculture	Accra Metropolitan	-0.1971153	5.607396	316	333.1316	38	12.143439
10103	Business	Accra Metropolitan	-0.1971153	5.607396	341	357.9664	119	13.935070
10103	General Arts	Accra Metropolitan	-0.1971153	5.607396	349	362.5812	117	17.483647
10103	General Science	Accra Metropolitan	-0.1971153	5.607396	335	353.5625	80	25.615886
10103	Home Economics	Accra Metropolitan	-0.1971153	5.607396	320	336.0408	49	29.216339
10103	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	343	357.9500	40	10.613046
10104	General Arts	Accra Metropolitan	-0.1971153	5.607396	302	320.1273	55	3.372069
10104	General Science	Accra Metropolitan	-0.1971153	5.607396	245	283.3636	55	9.474863
10104	Home Economics	Accra Metropolitan	-0.1971153	5.607396	264	285.8545	55	3.806261
10104	Visual Arts	Accra Metropolitan	-0.1971153	5.607396	273	298.3273	55	3.320343

EX4

```
# rank 1 choice
sp1 = left_join(sp1,
```

```r
                school_dis %>% select(
                  schoolcode, choicepgm, cutoff, quality, avg_distance),
                by = c("schoolcode1" = "schoolcode","choicepgm1" = "choicepgm"))

# cutoff
cmean1 = mean(sp1[,"cutoff"][!is.na(sp1[,"cutoff"])])
csd1 = sd(sp1[,"cutoff"][!is.na(sp1[,"cutoff"])])
# quality
qmean1 = mean(sp1[,"quality"][!is.na(sp1[,"quality"])])
qsd1 = sd(sp1[,"quality"][!is.na(sp1[,"quality"])])
# distance
dmean1 = mean(sp1[,"avg_distance"][!is.na(sp1[,"avg_distance"])])
dsd1 = sd(sp1[,"avg_distance"][!is.na(sp1[,"avg_distance"])])
# rank 2 choice
sp2 = left_join(sp2,
                school_dis %>% select(
                  schoolcode, choicepgm, cutoff, quality, avg_distance),
                by = c("schoolcode2" = "schoolcode","choicepgm2" = "choicepgm"))

# cutoff
cmean2 = mean(sp2[,"cutoff"][!is.na(sp2[,"cutoff"])])
csd2 = sd(sp2[,"cutoff"][!is.na(sp2[,"cutoff"])])
# quality
qmean2 = mean(sp2[,"quality"][!is.na(sp2[,"quality"])])
qsd2 = sd(sp2[,"quality"][!is.na(sp2[,"quality"])])
# distance
dmean2 = mean(sp2[,"avg_distance"][!is.na(sp2[,"avg_distance"])])
dsd2 = sd(sp2[,"avg_distance"][!is.na(sp2[,"avg_distance"])])

# rank 3 choice
sp3 = left_join(sp3,
                school_dis %>% select(
                  schoolcode, choicepgm, cutoff, quality, avg_distance),
                by = c("schoolcode3" = "schoolcode","choicepgm3" = "choicepgm"))

# cutoff
cmean3 = mean(sp3[,"cutoff"][!is.na(sp3[,"cutoff"])])
csd3 = sd(sp3[,"cutoff"][!is.na(sp3[,"cutoff"])])
# quality
qmean3 = mean(sp3[,"quality"][!is.na(sp3[,"quality"])])
qsd3 = sd(sp3[,"quality"][!is.na(sp3[,"quality"])])
# distance
dmean3 = mean(sp3[,"avg_distance"][!is.na(sp3[,"avg_distance"])])
dsd3 = sd(sp3[,"avg_distance"][!is.na(sp3[,"avg_distance"])])

# rank 4
sp4 = left_join(sp4,
                school_dis %>% select(
                  schoolcode, choicepgm, cutoff, quality, avg_distance),
                by = c("schoolcode4" = "schoolcode","choicepgm4" = "choicepgm"))

# cutoff
cmean4 = mean(sp4[,"cutoff"][!is.na(sp4[,"cutoff"])])
```

```r
csd4 = sd(sp4[,"cutoff"][!is.na(sp4[,"cutoff"])])
# quality
qmean4 = mean(sp4[,"quality"][!is.na(sp4[,"quality"])])
qsd4 = sd(sp4[,"quality"][!is.na(sp4[,"quality"])])
# distance
dmean4 = mean(sp4[,"avg_distance"][!is.na(sp4[,"avg_distance"])])
dsd4 = sd(sp4[,"avg_distance"][!is.na(sp4[,"avg_distance"])])


# rank 5
sp5 = left_join(sp5,
                school_dis %>% select(
                  schoolcode, choicepgm, cutoff, quality, avg_distance),
                by = c("schoolcode5" = "schoolcode","choicepgm5" = "choicepgm"))

# cutoff
cmean5 = mean(sp5[,"cutoff"][!is.na(sp5[,"cutoff"])])
csd5 = sd(sp5[,"cutoff"][!is.na(sp5[,"cutoff"])])
# quality
qmean5 = mean(sp5[,"quality"][!is.na(sp5[,"quality"])])
qsd5 = sd(sp5[,"quality"][!is.na(sp5[,"quality"])])
# distance
dmean5 = mean(sp5[,"avg_distance"][!is.na(sp5[,"avg_distance"])])
dsd5 = sd(sp5[,"avg_distance"][!is.na(sp5[,"avg_distance"])])

# rank 6
sp6 = left_join(sp6,
                school_dis %>% select(
                  schoolcode, choicepgm, cutoff, quality, avg_distance),
                by = c("schoolcode6" = "schoolcode","choicepgm6" = "choicepgm"))

# cutoff
cmean6 = mean(sp6[,"cutoff"][!is.na(sp6[,"cutoff"])])
csd6 = sd(sp6[,"cutoff"][!is.na(sp6[,"cutoff"])])
# quality
qmean6 = mean(sp6[,"quality"][!is.na(sp6[,"quality"])])
qsd6 = sd(sp6[,"quality"][!is.na(sp6[,"quality"])])
# distance
dmean6 = mean(sp6[,"avg_distance"][!is.na(sp6[,"avg_distance"])])
dsd6 = sd(sp6[,"avg_distance"][!is.na(sp6[,"avg_distance"])])

# make table
describe_rank = matrix(c(cmean1, csd1, qmean1, qsd1, dmean1, dsd1,
                       cmean2, csd2, qmean2, qsd2, dmean2, dsd2,
                       cmean3, csd3, qmean3, qsd3, dmean3, dsd3,
                       cmean4, csd4, qmean4, qsd4, dmean4, dsd4,
                       cmean5, csd5, qmean5, qsd5, dmean5, dsd5,
                       cmean6, csd6, qmean6, qsd6, dmean6, dsd6),
                       ncol = 6, byrow = TRUE)
colnames(describe_rank) = c("cutoff_mean", "cutoff_sd", "quality_mean",
                            "quality_sd", "distance_mean", "distance_sd")
rownames(describe_rank) = c("rank1", "rank2", "rank3", "rank4", "rank5",
                            "rank6")
```

Table 3: Descriptive characteristics1

	cutoff_mean	cutoff_sd	quality_mean	quality_sd	distance_mean	distance_sd
rank1	255.6668	50.25398	283.2153	44.88691	31.31638	25.16919
rank2	255.7365	50.31093	283.3278	44.90970	31.29904	25.12976
rank3	255.6959	50.28441	283.2957	44.87948	31.28798	25.11730
rank4	255.6609	50.27969	283.2341	44.90437	31.28649	25.15790
rank5	249.6318	45.65402	277.7417	40.32279	30.08643	24.35934
rank6	248.9744	44.59776	276.9881	39.25010	29.34648	22.97542

```r
# compute the quantile of student test scores
quantile(datstu[,'score'], probs = seq(0, 1, 0.25), na.rm = TRUE)
```

```
##   0%  25%  50%  75% 100%
##  158  252  283  324  469
```

```r
# quantile
q1 = filter(admitted, score <= 252)
q2 = filter(admitted, score > 252 & score <=283)
q3 = filter(admitted, score > 283 & score <= 324)
q4 = filter(admitted, score > 324)
# table
des_quantile = matrix(c(min(q1[,"score"]), mean(q1[,"score"]), sd(q1[,"score"]),
                      mean(q1[,"distance"]), sd(q1[,"distance"]),
                      min(q2[,"score"]), mean(q2[,"score"]), sd(q2[,"score"]),
                      mean(q2[,"distance"], na.rm = TRUE), sd(q2[,"distance"], na.rm = TRUE),
                      min(q3[,"score"]), mean(q3[,"score"]), sd(q3[,"score"]),
                      mean(q3[,"distance"]), sd(q3[,"distance"]),
                      min(q4[,"score"]), mean(q4[,"score"]), sd(q4[,"score"]),
                      mean(q4[,"distance"]), sd(q4[,"distance"])
                      ), ncol = 5, byrow = TRUE)

colnames(des_quantile) = c("cutoff", "quality_mean", "quality_sd", "distance_mean",
                           "distance_sd")
rownames(des_quantile) = c("Q1", "Q2", "Q3", "Q4")
```

Table 4: Descriptive characteristics2

	cutoff	quality_mean	quality_sd	distance_mean	distance_sd
Q1	158	235.1609	11.868473	25.62145	42.73525
Q2	253	267.8348	8.924626	27.97151	47.72807
Q3	284	302.7984	11.707446	31.04465	47.86558
Q4	325	362.3934	28.471819	37.88719	46.20664

EX5-6

Y~X1

```r
# setting a seed
set.seed(1233)
X1 = runif(10000, 1, 3)
X2 = rgamma(10000, 3, scale = 2)
X3 = rbinom(10000, 1, 0.3)
```

```
eps = rnorm(10000, 2, 1)
# Y variables
Y = 0.5 + 1.2 * X1 - 0.9 * X2 + 0.1 * X3 + eps
ydum = as.numeric(Y > mean(Y))
```

```
ones = seq(from = 1, to  = 1, length.out = 10000)
# Y ~ X1
new_X1 = cbind(ones, X1) # add ones to first col of X1
beta1 = solve(t(new_X1) %*% new_X1) %*% t(new_X1) %*% Y
```

```
beta1
```

```
##           [,1]
## ones -3.033850
## X1    1.277808
```

We can see that the correlation coefficient between Y and X1 is 1.2778, which is slightly larger than 1.2, which is the true coefficient. This may due to the number of sample we have, i.e. if we increase the number of observations from 10,000 to 100,000 or 1 000,000 the beta_hat we computed will be closer and closer to 1.2. The difference might also due to the residual, but it will not have big impact.

Y~(1, X1, X2, X3, X4)

```
X = cbind(new_X1, X2, X3)
# beta
beta = solve(t(X) %*% X) %*% t(X) %*% Y
```

```
beta
```

```
##            [,1]
## ones  2.4238161
## X1    1.2105091
## X2   -0.8933401
## X3    0.1655092
```

Standard error

```
Y_hat = X %*% beta
sigma2 = sum((Y-Y_hat)^2)/10000
var_beta = sigma2 * solve(t(X) %*% X)
std_err = sqrt(diag(var_beta))
std_err
```

```
##        ones          X1          X2          X3
## 0.040664988 0.017332244 0.002820772 0.021662514
```

EX7

Linear probability model: $P[y_i = 1|x_i] = x_i^T\beta$

```
result1 = solve(t(X) %*% X) %*% t(X) %*% ydum
result1
```

```
##             [,1]
## ones  0.89233912
## X1    0.13627886
## X2   -0.10106224
## X3    0.02035289
```

These parameters means that, ceteris paribus, for every 1 unit increase of X1, the probability of Ydum = 1 will increase by 0.136; for every 1 unit increase of X2, the probability of Ydum = 1 will decrease by 0.1; for every 1 unit increase of X3, the probability of Ydum = 1 will increase by 0.02.

```
p = 3
hat_ydum1 = X %*% result1
SSE1 = sum((ydum - hat_ydum1)^2)
SSR1 = sum((hat_ydum1 - mean(ydum))^2)
f1 = (SSR1/p)/(SSE1/(10000-p-1))
f1 > qf(.99, df1=3, df2=10000-4)
```

```
## [1] TRUE
```

The F-test shows that the parameters are significant at 0.01 level.

Probit model: $P[y_i = 1|x_i] = \phi(x_i^T \beta)$

```
log.neg_llh2 = function(X, par2){
  -sum(ydum * log(pnorm(X %*% par2)) + (1 - ydum) * log(1 - pnorm(X %*% par2)) )
}
result2 = optim(par = c(0,0,0,0), fn = log.neg_llh2, X = X)
result2$par
```

```
## [1]  2.8689820  1.1353529 -0.8629948  0.1921023
```

These parameters indicates that, ceteris paribus, an increase of X1 or X3 will increases the probability of ydum = 1, whereas an increase of X2 will decreases the probability of ydum = 1.

```
p = 3
hat_par2 = result2$par
hat_ydum2 = X %*% hat_par2
SSE2 = sum((ydum - hat_ydum2)^2)
SSR2 = sum((hat_ydum2 - mean(ydum))^2)
f2 = (SSR2/p)/(SSE2/(10000-p-1))
f2 > qf(.99, df1=3, df2=10000-4)
```

```
## [1] TRUE
```

The F-test shows that the parameters are significant at 0.01 level.

Logit model: $P[y_i = 1|x_i] = \frac{1}{1+e^{-x_i^T \beta}}$

```
log.neg_llh3 = function(X, par3){
  -sum(ydum * log(1/(1+exp(-X%*%par3))) + (1 - ydum) * log(1 -1/(1+exp(-X%*%par3)) ))
}
result3 = optim(par = c(0,0,0,0), fn = log.neg_llh3, X = X)
result3$par
```

```
## [1]  5.1313522  2.0264261 -1.5436514  0.3476706
```

These parameters indicates that, ceteris paribus, an increase of X1 or X3 will increases the probability of ydum = 1, whereas an increase of X2 will decreases the probability of ydum = 1. Noted that parameters for the logit model are generally larger in magnitude than those for the probit model, as well as those for the linear model.

```
p = 3
hat_par3 = result3$par
hat_ydum3 = X %*% hat_par3
SSE3 = sum((ydum - hat_ydum3)^2)
SSR3 = sum((hat_ydum3 - mean(ydum))^2)
```

```r
f3 = (SSR3/p)/(SSE3/(10000-p-1))
f3 > qf(.99, df1=3, df2=10000-4)
```

```
## [1] TRUE
```

The F-test shows that the parameters are significant at 0.01 level.

EX8

```r
# probit model
# marginal effect(average marginal effect)
me_probit = exp(-(X%*%hat_par2)^2/2) %*% t(hat_par2) / sqrt(2 * pi)
avg_me_probit = colMeans(me_probit)
# logit model
# marginal effect(average marginal effect)
f = 1/(1+exp(- X%*% hat_par3))
me_logit = (f * (1 - f)) %*% t(hat_par3)
avg_me_logit = colMeans(me_logit)
# std error
set.seed(123)
R = 999
nvar = length(hat_par2)
outs2 = mat.or.vec(R,nvar)
outs3 = mat.or.vec(R,nvar)

for (i in 1:R)
{
  samp = sample(1:10000,200,rep=TRUE)
  X_samp = X[samp,]

  probit = exp(-(X_samp %*% hat_par2)^2/2) %*% t(hat_par2) / sqrt(2 * pi)
  outs2[i,] = colMeans(probit)

  f_boo = 1/(1+exp(- X_samp %*% hat_par3))
  logit = (f_boo * (1 - f_boo)) %*% t(hat_par3)
  outs3[i,] = colMeans(logit)
}

sd_probit = apply(outs2, 2, sd)
sd_logit = apply(outs3, 2, sd)
```

Probit model

The average marginal effect:

```r
avg_me_probit
```

```
## [1]  0.35079736  0.13882234 -0.10552047  0.02348881
```

Standard error of the marginal effect:

```r
sd_probit
```

```
## [1] 0.028989533 0.011472135 0.008720103 0.001941091
```

Logit model

The average marginal effect:

```
avg_me_logit
```

```
## [1]  0.35136584  0.13875815 -0.10570047  0.02380651
```

Standard error of the marginal effect:

```
sd_logit
```

```
## [1] 0.030550745 0.012064817 0.009190501 0.002069941
```