A3

Shiqi Zhou

```
#set path
getwd()
```

```
## [1] "/Users/zhoushiqi/Desktop/613/A3/A3.R"
```

```
mainpath = "/Users/zhoushiqi/Desktop/613/A3/A3.R"
setwd(mainpath)
getwd()
```

```
## [1] "/Users/zhoushiqi/Desktop/613/A3/A3.R"
```

```
datapath = "/Data.A3"

#import datasets
DATA <- list.files(paste0(mainpath,datapath))
for (i in 1:length(DATA)) {
  assign(sub(".csv","",DATA[i]), fread(paste0(paste0(mainpath,datapath),"/",DATA[i]),encoding = "UTF-8")
}
```

Exercise 1: Basic Statistics

```
names(datsss)
```

```
## [1] "V1"          "schoolname"  "schoolcode"  "sssdistrict" "ssslong"
## [6] "ssslat"
```

```
names(datjss)
```

```
## [1] "V1"          "jssdistrict" "point_x"     "point_y"
```

```
names(datstu_v2)
```

```
##  [1] "V1"          "score"       "agey"        "male"        "schoolcode1"
##  [6] "schoolcode2" "schoolcode3" "schoolcode4" "schoolcode5" "schoolcode6"
## [11] "choicepgm1"  "choicepgm2"  "choicepgm3"  "choicepgm4"  "choicepgm5"
## [16] "choicepgm6"  "jssdistrict" "rankplace"
```

1.1 Number of students, schools, programs

```
# number of students
nrow(datstu_v2)
```

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

```
# Number of schools(total)
length(unique(datsss$schoolcode))
```

```
## [1] 898
```

```
# Number of school(student choices)
length(na.omit(unique(unlist(datstu_v2[,5:10]))))
```

```
## [1] 640
```

```
# Number of program
length(na.omit(unique(unlist(datstu_v2[,11:16]))))
```

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

1.2 Number of choices (school,program)

```
datstu = mutate(datstu_v2,choice1=paste(schoolcode1, choicepgm1, sep=" "),
                choice2=paste(schoolcode2, choicepgm2, sep=" "),
                choice3=paste(schoolcode3, choicepgm3, sep=" "),
                choice4=paste(schoolcode4, choicepgm4, sep=" "),
                choice5=paste(schoolcode5, choicepgm5, sep=" "),
                choice6=paste(schoolcode6, choicepgm6, sep=" "))
a = select(datstu,choice1,choice2,choice3,choice4,choice5,choice6)
a1 = gather(a, 'key', 'value')
a1 = subset(a1, value!= "NA")
length(unique(na.omit(a1$value)))
```

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

1.3 Number of students applying to at least one senior high schools in the same district to home

```
#this data has a schoolname with symbol and cannot get length
filter(datsss,schoolcode==30403)
```

```
##      V1                           schoolname schoolcode
## 1: 1373      ABAKRAMPA SENIOR HIGH TECHNICAL      30403
## 2: 1583      ABAKRAMPA SENIOR HIGH TECHNICAL      30403
## 3: 1710      ABAKRAMPA SENIOR HIGH TECHNICAL      30403
## 4: 2009 \xd8?ABAKRAMPA SENIOR HIGH TECHNICAL      30403
## 5: 3463                                           30403
## 6: 3668                                           30403
## 7: 4455                                           30403
## 8: 6015                                           30403
## 9: 6050                                           30403
##                             sssdistrict   ssslong   sssslat
## 1: Abura/Asebu/Kwamankese (Abura Dunkwa) -1.197088 5.130001
## 2: Abura/Asebu/Kwamankese (Abura Dunkwa) -1.197088 5.130001
## 3: Abura/Asebu/Kwamankese (Abura Dunkwa) -1.197088 5.130001
## 4: Abura/Asebu/Kwamankese (Abura Dunkwa) -1.197088 5.130001
## 5:                    Abura/Asebu/ Kwaman        NA       NA
## 6:                    Abura/Asebu/ Kwaman        NA       NA
## 7:                    Abura/Asebu/ Kwaman        NA       NA
## 8:                    Abura/Asebu/ Kwaman        NA       NA
## 9:                    Abura/Asebu/ Kwaman        NA       NA
```

```
datsss <- filter(datsss,V1!=2009)

#link the school district and home district
b=select(mutate(datsss,str1=str_length(schoolname),str2=str_length(sssdistrict)),-V1)
b1=unique(filter(group_by(b,schoolcode),str1==max(str1)))
b2=filter(group_by(b1,schoolcode),str2==max(str2))

schoolcd=ungroup(select(b2,schoolname,schoolcode,sssdistrict,ssslong,ssslat))

b3=schoolcd

b3$schoolcode1=b3$schoolcode
b3$schoolcode2=b3$schoolcode
b3$schoolcode3=b3$schoolcode
b3$schoolcode4=b3$schoolcode
b3$schoolcode5=b3$schoolcode
b3$schoolcode6=b3$schoolcode
b3$sssdistrict1=b3$sssdistrict
b3$sssdistrict2=b3$sssdistrict
b3$sssdistrict3=b3$sssdistrict
b3$sssdistrict4=b3$sssdistrict
b3$sssdistrict5=b3$sssdistrict
b3$sssdistrict6=b3$sssdistrict


choicetime=1:6
stusss=datstu
for (i in choicetime) {
  schlcd=paste0("schoolcode","",choicetime[i])
  ssdist=paste0("sssdistrict","",choicetime[i])
  atest=select(b3,schlcd,ssdist)
  stusss=left_join(stusss, atest, by=schlcd)
}
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(schlcd)` instead of `schlcd` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(ssdist)` instead of `ssdist` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
b4=stusss
b4$schlhm1 <- ifelse(b4$sssdistrict1==b4$jssdistrict,1,0)
b4$schlhm2 <- ifelse(b4$sssdistrict2==b4$jssdistrict,1,0)
b4$schlhm3 <- ifelse(b4$sssdistrict3==b4$jssdistrict,1,0)
b4$schlhm4 <- ifelse(b4$sssdistrict4==b4$jssdistrict,1,0)
b4$schlhm5 <- ifelse(b4$sssdistrict5==b4$jssdistrict,1,0)
b4$schlhm6 <- ifelse(b4$sssdistrict6==b4$jssdistrict,1,0)
b4$schlhm = b4$schlhm1+b4$schlhm2+b4$schlhm3+b4$schlhm4+b4$schlhm5+b4$schlhm6

nrow(filter(b4,schlhm>=1))
```

```
## [1] 250826
```

1.4 Number of students each senior high school admitted

```
stusss$admitted_schoolcode=ifelse(stusss$rankplace==1, stusss$schoolcode1,
                                  ifelse(stusss$rankplace==2, stusss$schoolcode2,
                                         ifelse(stusss$rankplace==3, stusss$schoolcode3,
                                                ifelse(stusss$rankplace==4, stusss$schoolcode4,
                                                       ifelse(stusss$rankplace==5, stusss$schoolcode5,
                                                              ifelse(stusss$rankplace==6, stusss$schoolc
stusss$admitted_pgm=ifelse(stusss$rankplace==1, stusss$choicepgm1,
                           ifelse(stusss$rankplace==2, stusss$choicepgm2,
                                  ifelse(stusss$rankplace==3, stusss$choicepgm3,
                                         ifelse(stusss$rankplace==4, stusss$choicepgm4,
                                                ifelse(stusss$rankplace==5, stusss$choicepgm5,
                                                       ifelse(stusss$rankplace==6, stusss$choicepgm6, NA
stusss$admitted=ifelse(stusss$rankplace==1, stusss$choice1,
                       ifelse(stusss$rankplace==2, stusss$choice2,
                              ifelse(stusss$rankplace==3, stusss$choice3,
                                     ifelse(stusss$rankplace==4, stusss$choice4,
                                            ifelse(stusss$rankplace==5, stusss$choice5,
                                                   ifelse(stusss$rankplace==6, stusss$choice6, NA))))))

(schladmit <- summarise(group_by(stusss,admitted_schoolcode),size=n(),cutoff=min(score),maxscore=max(sc
```

```
## # A tibble: 518 x 5
##    admitted_schoolcode  size cutoff maxscore quality
##                  <int> <int>  <int>    <int>   <dbl>
##  1               10101   398    284      397    320.
##  2               10102   248    343      456    394.
##  3               10103   443    316      421    354.
##  4               10104   220    245      371    297.
##  5               10105   346    260      428    351.
##  6               10106   395    293      407    340.
##  7               10107   306    281      393    312.
##  8               10108   318    248      378    304.
##  9               10109   300    257      362    282.
## 10               10110   535    343      461    408.
## # ... with 508 more rows
```

```
(c1=schladmit[,1:2])
```

```
## # A tibble: 518 x 2
##    admitted_schoolcode  size
##                  <int> <int>
##  1               10101   398
##  2               10102   248
##  3               10103   443
##  4               10104   220
##  5               10105   346
##  6               10106   395
##  7               10107   306
##  8               10108   318
##  9               10109   300
## 10               10110   535
## # ... with 508 more rows
```

1.5 The cutoff of senior high schools (the lowest score to be admitted)

```
(c2=select(schladmit,admitted_schoolcode,cutoff))
```

```
## # A tibble: 518 x 2
##    admitted_schoolcode cutoff
##                  <int>  <int>
##  1               10101    284
##  2               10102    343
##  3               10103    316
##  4               10104    245
##  5               10105    260
##  6               10106    293
##  7               10107    281
##  8               10108    248
##  9               10109    257
## 10               10110    343
## # ... with 508 more rows
```

1.6 The quality of senior high schools (the average score of students admitted)

```
(c3=select(schladmit,admitted_schoolcode,quality))
```

```
## # A tibble: 518 x 2
##    admitted_schoolcode quality
##                  <int>   <dbl>
##  1               10101    320.
##  2               10102    394.
##  3               10103    354.
##  4               10104    297.
##  5               10105    351.
##  6               10106    340.
##  7               10107    312.
##  8               10108    304.
##  9               10109    282.
## 10               10110    408.
## # ... with 508 more rows
```

Exercise 2: Data

```
#2.1 the district where the school is located
#2.2 the latitude and longitude of the district
#2.3 cutoff (the lowest score to be admitted)
#2.4 quality (the average score of the students admitted)
#2.5 size (number of students admitted)
colnames(schladmit)[1] = 'schoolcode'
(schldat=left_join(schoolcd,schladmit,by="schoolcode"))
```

```
## # A tibble: 898 x 9
```

```
##    schoolname      schoolcode sssdistrict   ssslong ssslat  size cutoff maxscore
##    <chr>                <int> <chr>           <dbl>  <dbl> <int>  <int>    <int>
##  1 WESLEY GIRLS H~      30107 Cape Coast M~  -1.31    5.15   380    394      469
##  2 HOLY CHILD SEN~      30103 Cape Coast M~  -1.31    5.15   220    393      464
##  3 ST. PETER'S SE~      21003 Kwahu South ~  -0.636   6.62   280    372      468
##  4 PRESBY BOYS SE~      10111 Ga West (Ama~  -0.398   5.66   600    371      462
##  5 MFANTSIPIM SEN~      30104 Cape Coast M~  -1.31    5.15   400    373      464
##  6 ABURI GIRLS SE~      20301 Akwapim Sout~  -0.268   5.83   450    385      457
##  7 OPOKU WARE SEN~      50110 Kumasi Metro   -1.60    6.68   440    387      464
##  8 ACHIMOTA SENIO~      10110 Accra Metrop~  -0.197   5.61   535    343      461
##  9 ARCHBISHOP POR~      40103 Shama/Ahanta~  -1.62    5.08   236    377      447
## 10 ST. ROSE'S SEN~      21103 Kwaebibirem ~  -0.799   6.13   225    388      450
## # ... with 888 more rows, and 1 more variable: quality <dbl>
```

Exercise 3: Distance

```
d=select(stusss, V1, choice1:choice6)
d=gather(d, 'key', 'value', -V1)

datjss=select(datjss, -V1)
stujss=left_join(stusss, datjss, by="jssdistrict")

d2=left_join(d, stujss, by=c("V1"="V1"))
d3=select(d2, V1, key, value, jssdistrict, point_x, point_y)
d3=cbind(d3, colsplit(d3$value," ",c("schoolcode","program")))

stujssdis=left_join(d3, schoolcd, by=c("schoolcode"="schoolcode"))
stujssdis$distance=sqrt(
  (69.172*(stujssdis$ssslong-stujssdis$point_x)*cos(stujssdis$point_y/57.3))^2+(69.172*(stujssdis$sssla
)

schldis=select(filter(stujssdis,!is.na(stujssdis$distance)), jssdistrict, point_x, point_y,
               schoolname, sssdistrict, ssslat, ssslong, distance)

head(stujssdis,20)
```

```
##    V1     key                 value                          jssdistrict
## 1   1 choice1   50112 Home Economics Bosomtwe/Atwima/Kwanwoma (Kuntanase)
## 2   2 choice1     70102 General Arts                          Ho Municipal
## 3   3 choice1         50702 Business                 Kwabre (Mamponteng)
## 4   4 choice1      90501 Visual Arts           Kassena/Nankani (Navrongo)
## 5   5 choice1   51802 Home Economics             Atwima Mponua (Nyinahin)
## 6   6 choice1     10102 General Arts                          Kumasi Metro
## 7   7 choice1     80301 General Arts             Nanumba North (Bimbilla)
## 8   8 choice1     40301 General Arts                  Jomoro (Half Assini)
## 9   9 choice1         21303 Business                     East Akim (Kibi)
## 10 10 choice1     80101 General Arts            Ejura/Sekyedumase (Ejura)
## 11 11 choice1   51802 Home Economics               Sekyere West (Mampong)
## 12 12 choice1 100201 General Science           Kassena/Nankani (Navrongo)
## 13 13 choice1         30603 Business                          Agona Swedru
## 14 14 choice1         80101 Business               Tolon Kunbungu (Tolon)
## 15 15 choice1        90301 Technical                   Accra Metropolitan
## 16 16 choice1     40903 General Arts          Mpohor-Wassa East (Daboase)
```

```
## 17 17 choice1     80102 General Arts            Ejura/Sekyedumase (Ejura)
## 18 18 choice1     10401 General Arts                   Ga West (Amasaman)
## 19 19 choice1      60301 Agriculture           Wassa Amenfi (Asankragwa)
## 20 20 choice1    100102 General Arts                                Bole
##       point_x   point_y schoolcode         program
## 1  -1.5627517  6.559323      50112  Home Economics
## 2   0.5261422  6.717607      70102    General Arts
## 3  -1.5414201  6.806778      50702        Business
## 4  -1.2174410 10.909423      90501     Visual Arts
## 5  -2.1771805  6.549507      51802  Home Economics
## 6  -1.5971872  6.682060      10102    General Arts
## 7  -0.1417642  8.816774      80301    General Arts
## 8  -2.8032203  5.069508      40301    General Arts
## 9  -0.4543442  6.178558      21303        Business
## 10 -1.3679653  7.462874      80101    General Arts
## 11 -1.1800768  7.199565      51802  Home Economics
## 12 -1.2174410 10.909423     100201 General Science
## 13 -0.7552425  5.617353      30603        Business
## 14 -1.1097199  9.527246      80101        Business
## 15 -0.1971153  5.607396      90301       Technical
## 16 -1.6975694  5.330796      40903    General Arts
## 17 -1.3679653  7.462874      80102    General Arts
## 18 -0.3975105  5.664688      10401    General Arts
## 19 -2.3020179  5.725518      60301     Agriculture
## 20 -2.2666752  8.629696     100102    General Arts
##                                             schoolname
## 1            KUMASI SENIOR HIGH./TECH. SCHOOL, KUMASI
## 2                      MAWULI SENIOR HIGH. SCHOOL, HO
## 3            SIMMS SENIOR HIGH. COMM. SCHOOL, FAWOADE
## 4          NAVRONGO SENIOR HIGH SCH, JAMANIA-NAVRONGO
## 5            TWENEBOA KODUA SENIOR HIGH. SCH., KUMAWU
## 6         ST. MARY'S SENIOR HIGH. SCHOOL, KORLE GONNO
## 7                   SALAGA SENIOR HIGH SCHOOL, SALAGA
## 8                    NSEIN SENIOR HIGH SCHOOL - NSEIN
## 9                        ABUAKWA STATE COLLEGE, KIBI
## 10                  TAMALE SENIOR HIGH SCHOOL, TAMALE
## 11           TWENEBOA KODUA SENIOR HIGH. SCH., KUMAWU
## 12                  NANDOM SENIOR HIGH SCHOOL, NANDOM
## 13    OBRACHIRE SENIOR HIGH./TECH. SCHOOL, OBRACHIRE
## 14                  TAMALE SENIOR HIGH SCHOOL, TAMALE
## 15 SANDEMA SENIOR HIGH/TECHNICAL SCH, BILINSA-SANDEMA
## 16       FIASEMAN SENIOR HIGH. SCHOOL, BENKYIM TARKWA
## 17                  GHANA SENIOR HIGH. SCHOOL, TAMALE
## 18               GHANATTA SENIOR HIGH. SCHOOL, DODOWA
## 19                BEREKUM SENIOR HIGH SCHOOL, BEREKUM
## 20                          WA SENIOR HIGH SCHOOL, WA
##                    sssdistrict    ssslong    ssslat   distance
## 1                  Kumasi Metro -1.5971872  6.682060   2.577169
## 2                  Ho Municipal  0.5261422  6.717607   0.000000
## 3          Kwabre (Mamponteng) -1.5414201  6.806778   0.000000
## 4   Kassena/Nankani (Navrongo) -1.2174410 10.909423   0.000000
## 5     Sekyere East (Effiduase) -0.8442360  7.210829  91.765769
## 6           Accra Metropolitan -0.1971153  5.607396  96.602382
## 7          East Gonja (Salaga) -0.5339396  8.729157  26.816957
```

```
## 8            Nzema East (Axim) -2.3118021  5.141226  33.864677
## 9             East Akim (Kibi) -0.4543442  6.178558   0.000000
## 10                     Tamale -0.7843482  9.383351  43.097181
## 11    Sekyere East (Effiduase) -0.8442360  7.210829  23.047843
## 12                      Lawra -2.8009412 10.546398 107.597001
## 13 Awutu/Efutu/Senya (Winneba) -0.5086389  5.544896  16.986853
## 14                     Tamale -0.7843482  9.383351  22.228458
## 15            Builsa (Sandema) -1.3374945 10.557073  88.643584
## 16         Wassa West (Tarkwa) -1.9888532  5.276049  20.066721
## 17                     Tamale -0.7843482  9.383351  43.097181
## 18        Dangme West (Dodowa)  0.5123865  5.786251  62.640244
## 19                    Berekum -2.6317439  7.503565  27.086959
## 20               Wa Municipal -2.2850304 10.030622  11.718883
```

Exercise 4: Dimensionality Reduction

```
#4.1 Recode the schoolcode into its first three digits (substr). Call this new variable scode rev
e=mutate(stusss,scode_rev1=substr(schoolcode1,1,3),
         scode_rev2=substr(schoolcode2,1,3),
         scode_rev3=substr(schoolcode3,1,3),
         scode_rev4=substr(schoolcode4,1,3),
         scode_rev5=substr(schoolcode5,1,3),
         scode_rev6=substr(schoolcode6,1,3))

#4.2 Recode the program variable into 4 categories,arts (general arts and visual arts),
#    economics (business and home economics), science (general science) and others. Call this new varia
e$pgm_rev1=ifelse(e$choicepgm1=="General Arts","arts",
                  ifelse(e$choicepgm1=="Visual Arts","arts",
                         ifelse(e$choicepgm1=="Business","economics",
                                ifelse(e$choicepgm1=="Home Economics","economics",
                                       ifelse(e$choicepgm1=="General Science","science","others")))))

e$pgm_rev2=ifelse(e$choicepgm2=="General Arts","arts",
                  ifelse(e$choicepgm2=="Visual Arts","arts",
                         ifelse(e$choicepgm2=="Business","economics",
                                ifelse(e$choicepgm2=="Home Economics","economics",
                                       ifelse(e$choicepgm2=="General Science","science","others")))))
e$pgm_rev3=ifelse(e$choicepgm3=="General Arts","arts",
                  ifelse(e$choicepgm3=="Visual Arts","arts",
                         ifelse(e$choicepgm3=="Business","economics",
                                ifelse(e$choicepgm3=="Home Economics","economics",
                                       ifelse(e$choicepgm3=="General Science","science","others")))))
e$pgm_rev4=ifelse(e$choicepgm4=="General Arts","arts",
                  ifelse(e$choicepgm4=="Visual Arts","arts",
                         ifelse(e$choicepgm4=="Business","economics",
                                ifelse(e$choicepgm4=="Home Economics","economics",
                                       ifelse(e$choicepgm4=="General Science","science","others")))))
e$pgm_rev5=ifelse(e$choicepgm5=="General Arts","arts",
                  ifelse(e$choicepgm5=="Visual Arts","arts",
                         ifelse(e$choicepgm5=="Business","economics",
                                ifelse(e$choicepgm5=="Home Economics","economics",
                                       ifelse(e$choicepgm5=="General Science","science","others")))))
e$pgm_rev6=ifelse(e$choicepgm6=="General Arts","arts",
```

```
                  ifelse(e$choicepgm6=="Visual Arts","arts",
                         ifelse(e$choicepgm6=="Business","economics",
                                ifelse(e$choicepgm6=="Home Economics","economics",
                                       ifelse(e$choicepgm6=="General Science","science","others")))))

#4.3 Create a new choice variable choice rev.
e = mutate(e,choice_rev1=paste(scode_rev1, pgm_rev1, sep=" "),
           choice_rev2=paste(scode_rev2, pgm_rev2, sep=" "),
           choice_rev3=paste(scode_rev3, pgm_rev3, sep=" "),
           choice_rev4=paste(scode_rev4, pgm_rev4, sep=" "),
           choice_rev5=paste(scode_rev5, pgm_rev5, sep=" "),
           choice_rev6=paste(scode_rev6, pgm_rev6, sep=" "))

#4.4 Recalculate the cutoff and the quality for each recoded choice.
e$revad_schoolcode=ifelse(e$rankplace==1, e$scode_rev1,
                          ifelse(e$rankplace==2, e$scode_rev2,
                                 ifelse(e$rankplace==3, e$scode_rev3,
                                        ifelse(e$rankplace==4, e$scode_rev4,
                                               ifelse(e$rankplace==5, e$scode_rev5,
                                                      ifelse(e$rankplace==6, e$scode_rev6, NA))))))
e$revad_pgm=ifelse(e$rankplace==1, e$pgm_rev1,
                   ifelse(e$rankplace==2, e$pgm_rev2,
                          ifelse(e$rankplace==3, e$pgm_rev3,
                                 ifelse(e$rankplace==4, e$pgm_rev4,
                                        ifelse(e$rankplace==5, e$pgm_rev5,
                                               ifelse(e$rankplace==6, e$pgm_rev6, NA))))))
e$revad=ifelse(e$rankplace==1, e$choice_rev1,
               ifelse(e$rankplace==2, e$choice_rev2,
                      ifelse(e$rankplace==3, e$choice_rev3,
                             ifelse(e$rankplace==4, e$choice_rev4,
                                    ifelse(e$rankplace==5, e$choice_rev5,
                                           ifelse(e$rankplace==6, e$choice_rev6, NA))))))

e=ungroup(e)

length(unique(e$revad))
```

```
## [1] 426
```

```
(e1 <- summarise(group_by(e,revad),size=n(),cutoff=min(score),quality=mean(score)))
```

```
## # A tibble: 426 x 4
##    revad          size cutoff quality
##    <chr>         <int>  <int>   <dbl>
##  1 100 arts       1200    194    276.
##  2 100 economics  1364    195    264.
##  3 100 others      630    191    246.
##  4 100 science     441    228    305.
##  5 101 arts       3201    243    340.
##  6 101 economics  2674    205    326.
##  7 101 others      770    257    313.
##  8 101 science    1499    203    369.
##  9 102 arts       1066    216    316.
## 10 102 economics   730    206    309.
```

```
## # ... with 416 more rows
```

```
#4.5 Consider the 20,000 highest score students.
e2=e[order(e$score,decreasing=T),]
e2[20000,2]
```

```
##    score
## 1:   355
```

```
e3=filter(e2,score>=355)
```

The rest of the assignment uses the recoded choices and the 20,000 highest score students.

construct dataset used in Ex5 & 6

```
colnames(e1)[1]="choice"
#choose related variables
f = select(e3,V1,score,scode_rev1,pgm_rev1,choice_rev1)
#delete NA in scode_rev
f=filter(f,!is.na(scode_rev1))
#combining with quality
colnames(f)[5]="choice"
f = left_join(f,e1,by="choice")
```

```
#choice number of choice1
choice1=unique(f$choice)
choice1=data.frame(sort(choice1,decreasing=F))
colnames(choice1)[1]="choice"
choice2=left_join(choice1,e1,by="choice")
choice3=select(choice2,choice,quality)

#choice matrix
ni=nrow(f)#num of people
nj=nrow(choice3)#num of choice
Y=class.ind(f$choice)
```

Exercise 5: First Model

#Using the new data with recoded choices, #we want to understand the effect of the student test score on his first choice.

5.1 Propose a model specification. Write the Likelihood function.

```
#here use multinomial logit model in ex5, test score is not the same for each student
v=select(f,score)
test=v

for (i in 1:245) {
  temp=rbind.data.frame(v,test)
  v=temp
}
```

```r
mlogit_ll<-function(beta){

  #Create the constant as instructed
  intercept1=cbind(0,matrix(rep(beta[1:245],each=ni),ni,nj-1))
  intercept2=cbind(0,matrix(rep(beta[246:490],each=ni),ni,nj-1))
  #Use the lecture definition of conditional logit to compute the likelihood
  XB=intercept1+intercept2*v
  eXB=exp(XB)
  teXB=rowSums(eXB)
  prob=eXB/teXB

  #Compute the neg log likelihood for each choice using the choice matrix
  ll=-sum(Y*log(prob))
  return(ll)
}
```

5.2 Estimate parameters and compute the marginal effect of the proposed model.

```r
#Estimate parameters
set.seed(0)
model1 <- optim(runif(490,-1,1),mlogit_ll,method="BFGS")
model1$par
```

```
> model1 <- optim(runif(490,-1,1),mlogit_ll,method="BFGS")
> model1$par
  [1]  0.659754255 -0.827209250 -0.914685762 -0.302518382  0.084672020  0.218921727 -0.457255803 -0.589538136
  [9] -0.236735703 -0.054883769  0.671035900 -0.757057007  0.351616250 -0.006629560  0.804475496  0.102527934
 [17] -0.742870647 -0.115980429 -0.615404607 -0.130198756 -0.549733194  0.922192564 -0.103145070  0.554289971
 [25] -0.683560451  0.733617165 -0.587708788 -0.644100634 -0.670217707  0.130537955  0.454362042  0.751837978
 [33]  0.416848768 -0.045073578  0.640559987 -0.966775578  0.992222202  0.269789672 -0.143049238 -0.943214134
 [41]  0.506665401 -0.582732049  0.998910810  0.814498386  0.424930064  0.461865768 -0.055024158  0.725021374
 [49] -0.669169755  0.240855851 -0.415337210 -0.087716532 -0.908721582 -0.644676496 -0.884135864  0.888820467
 [57] -0.314513051  0.035901141  0.250490671 -0.525470293  0.032247829  0.613083655 -0.303470531  0.717374906
 [65] -0.931122478  0.941994292  0.490220272 -0.453489516  0.354212203 -0.304105064  0.894041043 -0.322752450
 [73] -0.936573592 -0.292828405 -0.225752824 -0.286105680  0.919937212 -0.232523590  0.092675242  0.850646952
 [81]  0.833719054 -0.514534014  0.434181678 -0.302107928  0.108566701  0.485944826  0.639396637  0.739314530
 [89] -0.928269731 -0.559951193 -0.266053581 -0.388607116  0.455660915  0.399457532  0.820121854  0.691205219
 [97]  0.557089280 -0.198313144  0.153609318 -0.847281108  0.745949701  0.912571471  0.011062160  0.848732728
[105] -0.730972935 -0.522525994 -0.020074756 -0.051879806 -0.130011682 -0.921502080  0.288662030  0.431134400
[113] -0.296666557  0.893113498  0.486213014 -0.898428159  0.960729562 -0.564355095 -0.927607996 -0.478592269
[121]  0.440831572 -0.504211638 -0.530168417 -0.957976314 -0.470758745  0.383603239 -0.930661218 -0.874427741
[129] -0.204587816 -0.019943150  0.262097495  0.446808625 -0.846468415 -0.157107839  0.947453372  0.627611567
[137] -0.552464554 -0.020515388 -0.977319614 -0.482795406 -0.090802456  0.481953376  0.981320024 -0.335590108
[145]  0.889582811  0.923706840  0.798554999 -0.014568330  0.569870956  0.606238094  0.355535247  0.162555157
[153] -0.338663825 -0.997370687 -0.868396072 -0.828096165 -0.973780566 -0.328508751 -0.761471367  0.187190891
[161] -0.925374649 -0.980856416 -0.676807748  0.663797976  0.533685509 -0.454439367 -0.623673402 -0.548476330
[169] -0.876059265 -0.880199518 -0.669062268 -0.854024228 -0.914228216  0.044922545  0.578462979  0.389506696
[177] -0.866847905 -0.964084507 -0.115683153 -0.684795834  0.437720232  0.407883665  0.770452610 -0.349477970
[185]  0.941763133  0.969640533 -0.921630565  0.787142576  0.644798303  0.448489386 -0.422159724  0.003897209
[193] -0.143305997  0.220032032  0.852226348 -0.519753485 -0.456236332  0.470007293  0.503091746  0.861183842
[201] -0.074539450  0.721382353 -0.375927161 -0.583921973  0.841654934 -0.894526232 -0.177689138 -0.087317570
[209] -0.168279505  0.340706903  0.513808160  0.823302629  0.639904172 -0.812700295 -0.644187837  0.284429160
[217]  0.748565988  0.184626701 -0.475135616  0.767504528 -0.624621307 -0.022266035 -0.980359026 -0.279131914
[225] -0.115676618 -0.748541557  0.248729095 -0.395137337 -0.520725544 -0.282740473  0.785820825  0.981254530
[233]  0.464842395  0.985576177  0.079578724 -0.132396476  0.929347348 -0.703360714 -0.867013888 -0.951497024
[241]  0.224846484  0.393217496  0.602283774 -0.724539432 -0.680430355  0.977667186 -0.202708609 -0.624176368
[249] -0.044543518  0.088468045 -0.343684039  0.999861187  0.055563672 -0.732530984 -0.122919864  0.241587880
[257] -0.594175176 -0.731660455  0.553667996  0.271899697 -0.435727407 -0.617481088 -0.468926560  0.061617586
[265]  0.369721808 -0.233433213  0.909975997 -0.763286839 -0.921799889  0.009010066  0.156965110  0.678607816
[273]  0.308889951  0.889056680  0.023315259 -0.830157882  0.989541019 -0.197111760  0.791292593  0.753716824
[281] -0.407030404 -0.734816181 -0.468178185 -0.988544914  0.639203643 -0.650676642  0.686161104 -0.666575667
[289]  0.528038870  0.037876175 -0.454629393 -0.313593728  0.410150680  0.183070602 -0.061241925 -0.327246789
[297]  0.935674857  0.291683767 -0.607397662 -0.285639230  0.026036164  0.765993481 -0.694664869 -0.539076805
[305] -0.674022843  0.228693718  0.003272888 -0.623762224 -0.184570295  0.164776094 -0.382141870 -0.038068720
[313]  0.878789585 -0.907918904  0.069723459 -0.114678435  0.812730939  0.762084763 -0.673517767  0.504381531
```

```
[321] -0.194025596  0.113905387 -0.839661436 -0.766862815  0.850385122  0.757688481 -0.673584746  0.063127803
[329]  0.859584337 -0.348568012 -0.678410738  0.899415836 -0.075285903 -0.558458225 -0.995961535  0.238385207
[337] -0.194988156  0.800065221  0.011763969  0.931970301 -0.945026770 -0.241173574 -0.226510676 -0.324358549
[345] -0.271136297  0.277511939 -0.934171345  0.392385177  0.631128349  0.014845707 -0.774814748  0.812646602
[353] -0.173322094 -0.052488424  0.427428360 -0.516257579 -0.400718745  0.408957643  0.543349896 -0.972319319
[361]  0.102627907  0.304922381  0.863321416  0.380783410 -0.492465722 -0.177923260 -0.754217774 -0.736066889
[369]  0.280236016 -0.307153136 -0.154486791  0.878817761 -0.232221538 -0.542829738  0.380557372 -0.980284165
[377]  0.372261789  0.583349192  0.014986977 -0.383847502  0.474424555 -0.390569513  0.549417404 -0.775772724
[385]  0.595487400 -0.017997665  0.665836350  0.516846310  0.804293122 -0.443119974 -0.108614300 -0.455968529
[393] -0.340163349  0.727201308 -0.780284093 -0.195486201 -0.187849813 -0.249611453 -0.645999807 -0.468539600
[401]  0.319100027  0.632111010 -0.157631311 -0.679871392 -0.422779745  0.604258017 -0.671508636  0.579585470
[409]  0.048429940 -0.359929995 -0.449645824  0.603796076 -0.791951138  0.676244533 -0.841474253  0.435128134
[417]  0.729235653  0.957116416 -0.501747585 -0.950278619 -0.554043536 -0.755949001 -0.229521150 -0.831339051
[425] -0.434836200  0.133164918 -0.342915395  0.886949365 -0.245392972  0.906631225 -0.453494265  0.374660312
[433] -0.726016391  0.620421543  0.525209384 -0.115190332 -0.086648611  0.360910178 -0.851204182  0.293258150
[441]  0.579828223 -0.553366987 -0.918112019 -0.230703273 -0.311815692 -0.107262616 -0.226333661  0.836635831
[449] -0.651964913 -0.700258875  0.226398458 -0.965513336 -0.867779031  0.557678083  0.871312254 -0.585591721
[457]  0.286686314 -0.337240850  0.910843182  0.634378314  0.480552286 -0.554588046 -0.908550567 -0.266946841
[465]  0.482786050  0.867012500  0.346419900  0.402714215  0.695250816  0.412313310  0.717615519 -0.108711018
[473]  0.353269984 -0.744350435  0.472343070 -0.865509703 -0.438399861 -0.176764633  0.886849994  0.954685066
[481] -0.922266830  0.175455554 -0.939976646  0.130562417 -0.803545538  0.260581577  0.637003669 -0.299916674
[489] -0.732032650 -0.838113243
```

marginal effect

EXIT THIS PROGRAM FREQUENTLY, the compute space is limited to compute this.

#==========

Exercise 6: Second Model

Using the new data with recoded choices,

we want to understand the effect of the school quality on the first choice.

#6.1 Propose a model specification. Write the Likelihood function.

#here use conditional logit model in ex6, school quality is the same for all student

#6.2 Estimate parameters and compute marginal effect of the proposed model.

#optimization

```
> clogit <- optim(runif(246,-1,1),clogit_ll,method="BFGS")
> clogit$par
  [1]  0.878273412 -0.414102554 -0.671346851 -0.201794888 -0.080849176 -0.131938303  0.034019653  0.692491505
  [9] -0.889671427  0.108354122  0.376550475  0.316115107  0.326685451 -0.055531595  0.939056332 -0.195605875
 [17]  0.699104208  0.513289815  0.065202439  0.748299322 -0.065769769 -0.983743089  0.455534009  0.433178958
 [25] -0.625147258  0.292134554  0.083958560 -0.329358480  0.275817446  0.658402128  0.417950397 -0.302899298
 [33] -0.743344250 -0.223843022  0.856355097  0.608781543  0.517393614  0.914499778  0.987827757  0.212881989
 [41] -0.941245669 -0.327109284 -0.444683827 -0.765604900 -0.913563482 -0.259380429 -0.326243382 -0.652694893
 [49]  0.243546559 -0.204312745  0.911351537  0.306698886 -0.342512567 -0.605706573 -0.769315350  0.991930957
 [57] -0.241446550  0.123976185  0.465436035  0.741611106  0.144340519 -0.977927858  0.812630523  0.541307263
 [65] -0.234990753 -0.811908222 -0.900692832  0.642324636  0.658648609  0.309465752 -0.734344374 -0.316380198
 [73]  0.462743158  0.814582832  0.392393999 -0.516841555  0.288214457 -0.438499573  0.915273041 -0.683208087
 [81] -0.163322017 -0.495980405 -0.811219465  0.655435539  0.050611007  0.335495255 -0.183444480  0.685179807
 [89]  0.474610936 -0.303551198  0.897876355  0.293358380 -0.929444466  0.192896911 -0.169363997 -0.846205924
 [97]  0.056097758  0.924666617  0.417480102  0.106951407 -0.514086782  0.556085058  0.303881887  0.660491408
[105]  0.297101895 -0.040328411 -0.009871789 -0.240254817 -0.099029126  0.628503501  0.857554452 -0.705037910
[113]  0.499643330  0.951314699  0.949584927 -0.298748860 -0.212101886  0.901902018 -0.786703361  0.869520233
[121] -0.307675800  0.066121212  0.077588599  0.429435899 -0.188419000 -0.694423717 -0.319534477  0.253309695
[129] -0.885254643  0.703335280 -0.574709308  0.536496861 -0.727024828 -0.350269719  0.242152588 -0.488035503
[137]  0.269751598 -0.028655786  0.876353833  0.715003074 -0.258232925 -0.371596340  0.657068720 -0.096316974
[145] -0.368243176 -0.804382925 -0.870198927  0.378914733  0.336101208  0.809093296 -0.396613459  0.865617409
[153] -0.596033965  0.584757512 -0.550738982 -0.938486858  0.724068082  0.370215010  0.884149819  0.351707521
[161]  0.686240297 -0.276211630 -0.215268230  0.135374795 -0.809695733 -0.612431857  0.176132786  0.503008336
[169]  0.734477574 -0.256408526  0.597629099 -0.883371215  0.246871419 -0.286717187  0.175855865  0.827569284
[177] -0.601115634 -0.261832752  0.342816656  0.536289101  0.044496552  0.656150036  0.054192389  0.003509650
[185] -0.160053360 -0.275403433 -0.753142179 -0.403676941 -0.446647021  0.582912063  0.556362604 -0.712425439
[193]  0.031052301  0.194480971  0.011686048 -0.227800734 -0.147803798 -0.976480531  0.838663541 -0.841120614
[201]  0.014748510  0.640343234  0.196790836 -0.151692948  0.118620541  0.578188944 -0.664569484  0.940903460
[209] -0.052993803  0.859486412  0.801878541  0.501764376  0.353137545  0.296026892 -0.853506261 -0.152883163
[217]  0.061648728  0.885409525  0.424449119  0.448981151 -0.059742765 -0.759435488  0.566195342 -0.123685206
[225] -0.137088040 -0.945004242 -0.706876314 -0.154809684  0.534274349 -0.990467109  0.207191383  0.811154521
[233]  0.413323064 -0.474925681  0.702152439 -0.332788970  0.156568535 -0.134453740 -0.896809354  0.459606576
[241]  0.096340829  0.502443971 -0.898457671  0.429932425 -0.404611086 -0.433045581
```

marginal effect
EXIT THIS PROGRAM FREQUENTLY, the compute space is limited to compute this.

#================================
in excercise 5 and 6, I code to compute marginal effect in two ways,
but both of them are spending too much time and the compute space is limited to these two questions.
so, I left the coding here without results.
#================================

#================================
Exercise 7: Counterfactual simulations

#7.1 Explain and justify, which model (first or second model) you think is appropriate to conduct this exercise.
#use conditional model here (second model),
#because the the relative quality are different while omitting "others" programs
#for first model, thee result is based on pair comparasion, each group with base group.
#not suitable for omitting part of the choices.

#7.2 Calculate choice probabilities under the appropriate model.
#7.3 Simulate how these choice probabilities change when these choices are excluded

#calculate the new parameters

```
> clogit_omit <- optim(runif(196,-1,1),clogit_ll_omit,method="BFGS")
> clogit_omit$par
  [1]  0.78541984  0.27621029  0.10898623  0.53265990 -0.80627741  0.49603298  0.75043598 -0.98466281  0.86983397
 [10] -0.18606868  0.57752678  0.95225525 -0.33319491  0.57442964 -0.82127817 -0.91260020  0.02986748 -0.90382134
 [19] -0.82160774  0.37665104 -0.93032293 -0.13817815  0.51858810  0.66359391 -0.23507497  0.23110964 -0.31002670
 [28]  0.06455755 -0.47998179  0.14572124  0.81169336  0.13204376 -0.88702636 -0.79577251 -0.34331629  0.98120101
 [37]  0.42873068 -0.84205163  0.22843290 -0.01353265 -0.46180469 -0.41377827 -0.04944340  0.92815709 -0.15273580
 [46]  0.68819419  0.90765470 -0.42705295  0.08169880  0.16275664 -0.48742621  0.75616794  0.76460478  0.97466937
 [55] -0.79675820  0.23755000  0.40946754 -0.70282314  0.78162118 -0.67465940 -0.75626706  0.95511371  0.25422053
 [64] -0.96160114  0.65832829  0.99775495  0.73619103 -0.20617785  0.87248357 -0.35045421  0.94489512  0.12519691
 [73]  0.15865808  0.64938071 -0.60311104  0.23797927 -0.55087077 -0.94955254  0.51301025  0.27631389 -0.88426217
 [82]  0.34759563  0.38308954  0.49045792 -0.75187626 -0.69765643 -0.64513875 -0.98226106  0.26757729 -0.25449823
 [91] -0.44455990 -0.65498300 -0.59257772 -0.55625824 -0.49313699 -0.65980462  0.84335353 -0.30937308 -0.82978158
[100] -0.24436439  0.08668704  0.57402185 -0.35937294  0.67607704  0.99694097 -0.57311402 -0.12540702 -0.24254001
[109]  0.93115342  0.28223300  0.61227067  0.83812438 -0.88275510 -0.04471429  0.39376558 -0.91065698  0.51370863
[118]  0.18334745  0.02028093 -0.09479647 -0.94903607  0.92368722  0.97525488  0.26871912 -0.14891731 -0.46674819
[127] -0.83886102 -0.33971687  0.78810021 -0.38664960 -0.19515232  0.14758280  0.19397677  0.14003428 -0.56385742
[136]  0.64025349 -0.92668844  0.25116563 -0.13304212 -0.28614772  0.54733528 -0.65309590 -0.01358205  0.47437782
[145]  0.02200205 -0.04982617  0.17579988  0.14507476  0.30983548  0.73025111 -0.20630725  0.92023231 -0.23824549
[154] -0.65982344 -0.53679024  0.89984042 -0.83298368 -0.61391197 -0.85060941  0.21003202 -0.56831421  0.17597452
[163] -0.54220479 -0.21814964 -0.30418817  0.83675880 -0.77592175 -0.49004771 -0.62675186  0.49108705  0.86593756
[172] -0.01057621  0.28998736 -0.64839483  0.20598929  0.77551533 -0.01958119 -0.96323285 -0.64121842 -0.40121772
[181] -0.44607188 -0.87678115 -0.28935730  0.15407551  0.07006312  0.20854566 -0.02770204 -0.73674686  0.92914010
[190]  0.48492315 -0.26632123 -0.69834941 -0.54612065  0.49023362 -0.89881524  0.18798659
```

#compare the probability

#prob with all choices

```
> all_prob[1,]
  [1] 0.003344055 0.008048273 0.002210197 0.001708878 0.002732971 0.003084332 0.002930713 0.003459776 0.006683726
 [10] 0.001373707 0.003726756 0.004873121 0.004587335 0.004636082 0.003163416 0.008552644 0.002749938 0.006728070
 [19] 0.005587176 0.003569361 0.007067335 0.003131194 0.001250372 0.005273626 0.005157042 0.001789680 0.004478637
 [28] 0.003636940 0.002405663 0.004406151 0.006459722 0.005079102 0.002470165 0.001590168 0.002673373 0.007873788
 [37] 0.006147009 0.005610152 0.008345178 0.008980108 0.004137394 0.001304655 0.002411080 0.002143629 0.001555161
 [46] 0.001341275 0.002580037 0.002413169 0.001741052 0.004266231 0.002726098 0.008318947 0.004544342 0.002374226
 [55] 0.001824813 0.001549402 0.009017031 0.002626724 0.003785433 0.005326105 0.007020225 0.003863311 0.001257664
 [64] 0.007536928 0.005745928 0.002643737 0.001484794 0.001358650 0.006356696 0.006461314 0.004556933 0.001604544
 [73] 0.002437088 0.005311782 0.007551657 0.004950943 0.001994399 0.004461114 0.002156927 0.008351634 0.001688729
 [82] 0.002840165 0.002036441 0.001485817 0.006440587 0.003517657 0.004677105 0.002783585 0.006635035 0.005375196
 [91] 0.002468555 0.008207600 0.004484121 0.001320142 0.004055529 0.002823056 0.001434732 0.003537011 0.008430455
[100] 0.005076714 0.003721533 0.001999900 0.005831471 0.004531559 0.006473232 0.004500939 0.003211878 0.003311206
[109] 0.002629857 0.003028765 0.006269444 0.007883237 0.001652264 0.005511449 0.008658131 0.008643167 0.002480438
[118] 0.002704947 0.008240707 0.001522693 0.007978133 0.002458394 0.003572642 0.003613847 0.005137775 0.002769772
[127] 0.001669894 0.002429413 0.004308087 0.001379788 0.006756597 0.001882263 0.005718354 0.001616332 0.002355880
[136] 0.004260288 0.002052685 0.004379505 0.003249588 0.008032839 0.006835894 0.002582999 0.002306169 0.006451114
[145] 0.003036991 0.002313915 0.001496009 0.001400718 0.004884656 0.004679940 0.007510315 0.002249191 0.007947056
[154] 0.001842549 0.006001093 0.001927927 0.001308259 0.006898143 0.004842345 0.008095707 0.004753550 0.006642075
[163] 0.002536975 0.002696396 0.003828829 0.001488083 0.001812582 0.003988108 0.005530026 0.006970324 0.002587716
[172] 0.006078836 0.001382389 0.004280439 0.002510462 0.003987004 0.007650366 0.001833210 0.002573718 0.004711474
[181] 0.005717166 0.003496214 0.006445190 0.003530278 0.003355812 0.002849464 0.002539026 0.001574664 0.002233360
[190] 0.002139425 0.005990029 0.005833089 0.001640102 0.003449525 0.004061958 0.003383363 0.002662814 0.002884583
[199] 0.001259486 0.007735713 0.001442047 0.003393740 0.006344114 0.004071351 0.002873386 0.003765214 0.005961804
[208] 0.001720500 0.008568456 0.003171455 0.007898482 0.007456325 0.005523151 0.004760352 0.004496103 0.001424296
[217] 0.002869968 0.003556699 0.008105912 0.005112218 0.005239182 0.003150123 0.001564785 0.005890727 0.002955001
[226] 0.002915660 0.001299760 0.001649229 0.002864445 0.005705659 0.001241993 0.004113917 0.007525812 0.005055654
[235] 0.002079772 0.006748610 0.002397425 0.003910842 0.002923351 0.001363936 0.005295147 0.003682254 0.005526906
[244] 0.001361690 0.005140326 0.002231275
```

#prob after omit "others"

```
> omit_prob[1,]
  [1] 0.004398533 0.009647423 0.005797816 0.004905010 0.007492712 0.001964020 0.007223242 0.009315754 0.001643139
 [10] 0.010497162 0.003651735 0.007836543 0.011399007 0.003152123 0.007812309 0.001934778 0.001765918 0.004531887
 [19] 0.001781489 0.001934141 0.006410402 0.001734897 0.003830873 0.007388015 0.008540887 0.003477091 0.005542139
 [28] 0.003226004 0.004691857 0.002721788 0.005088546 0.009904254 0.005019422 0.001811662 0.001984761 0.003120380
 [37] 0.011733781 0.006753100 0.001895001 0.005527324 0.004339410 0.002771715 0.002908079 0.004186343 0.011127595
 [46] 0.003775509 0.008753601 0.010901776 0.002869730 0.004772975 0.005175974 0.002701602 0.009369305 0.009448687
 [55] 0.011657390 0.001982806 0.005577948 0.006624259 0.002178089 0.009610845 0.002240304 0.002064739 0.011431637
 [64] 0.005671715 0.001681472 0.008496032 0.011929638 0.009183992 0.003579035 0.010525012 0.003098186 0.011315417
 [73] 0.004985172 0.005154804 0.008420352 0.002406468 0.005580343 0.002535524 0.001701854 0.007346920 0.005798416
 [82] 0.001816677 0.006226825 0.006451809 0.007183085 0.002073825 0.002189372 0.002307425 0.001647090 0.005747979
 [91] 0.003410206 0.002819927 0.002284822 0.002431950 0.002521900 0.002686217 0.002273832 0.010222841 0.003228113
[100] 0.001918396 0.003444940 0.004796843 0.007809124 0.003070677 0.008648172 0.011919932 0.002479748 0.003880112
[109] 0.003451231 0.011160987 0.005832840 0.008113600 0.010169523 0.001819417 0.004206188 0.006521057 0.001769353
[118] 0.007352053 0.005283657 0.004488650 0.004000721 0.001702733 0.011077967 0.011664218 0.005754546 0.003789953
[127] 0.002758047 0.001901057 0.003131631 0.009673316 0.002988051 0.003618714 0.005098028 0.005340118 0.005059690
[136] 0.002502809 0.008343848 0.001741214 0.005654415 0.003850599 0.003303965 0.007603482 0.002289137 0.004339195
[145] 0.007068503 0.004496382 0.004184741 0.005243928 0.005085258 0.005996083 0.009129602 0.003578572 0.011039760
[154] 0.003466084 0.002273789 0.002571478 0.010816918 0.001912263 0.002380615 0.001878853 0.005426547 0.002491679
[163] 0.005244844 0.002557592 0.003536443 0.003244894 0.010155646 0.002024554 0.002694529 0.002350244 0.007187605
[172] 0.010456340 0.004352258 0.005878245 0.002299924 0.005404653 0.009552341 0.004313242 0.001678731 0.002316489
[181] 0.002944836 0.002815667 0.001830318 0.003293378 0.005131236 0.004717760 0.005418487 0.004278356 0.002105439
[190] 0.011138539 0.007143438 0.003370125 0.002187855 0.002547596 0.007181474 0.001790430
```