[Workspace loaded from ~/Desktop/.RData]

> #import dataset

> datjss = read.csv("~/Desktop/DUKE/COURSES/SPRING2019/ECON613/Assignments/1/dat/datjss.csv")

> datsss = read.csv("~/Desktop/DUKE/COURSES/SPRING2019/ECON613/Assignments/1/dat/datsss.csv")

> datstu = read.csv("~/Desktop/DUKE/COURSES/SPRING2019/ECON613/Assignments/1/dat/datstu.csv")

>

> #subset school and program data

> studentcode = matrix(datstu[,1], ncol=1)

> schoolcode1 = matrix(datstu[,5], ncol=1)

> schoolcode2 = matrix(datstu[,6], ncol=1)

> schoolcode3 = matrix(datstu[,7], ncol=1)

> schoolcode4 = matrix(datstu[,8], ncol=1)

> schoolcode5 = matrix(datstu[,9], ncol=1)

> schoolcode6 = matrix(datstu[,10], ncol=1)

> choicepgm1 = matrix(datstu[,11], ncol=1)

> choicepgm2 = matrix(datstu[,12], ncol=1)

> choicepgm3 = matrix(datstu[,13], ncol=1)

> choicepgm4 = matrix(datstu[,14], ncol=1)

> choicepgm5 = matrix(datstu[,15], ncol=1)

> choicepgm6 = matrix(datstu[,16], ncol=1)

> testscore = matrix(datstu[,2],ncol=1)

>

> #Exercise 1 Missing data

> #Number of students

> numstudent = nrow(studentcode)

>

> #Number of school

> school = rbind(schoolcode1,schoolcode2,schoolcode3,schoolcode4,schoolcode5,schoolcode6) #stack the data

> school = na.omit(school) #delete NA value

> schoolnumber = unique(school) #delete duplications

> dim(schoolnumber)

[1] 640 1

>

> #Number of program

> program = rbind(choicepgm1,choicepgm2,choicepgm3,choicepgm4,choicepgm5,choicepgm6) #stack the data

> program = na.omit(program) #delete NA value

> programnumber = unique(program) #delete duplications

> dim(programnumber) #calculate # of rows

[1] 33 1

>

> #Number of choices

> choice1 = cbind(schoolcode1,choicepgm1) #combine info

> choice2 = cbind(schoolcode2,choicepgm2)

> choice3 = cbind(schoolcode3,choicepgm3)

> choice4 = cbind(schoolcode4,choicepgm4)

> choice5 = cbind(schoolcode5,choicepgm5)

> choice6 = cbind(schoolcode6,choicepgm6)

> choice = rbind(choice1,choice2,choice3,choice4,choice5,choice6) #stack info

> colnames(choice) = c("schoolcode","program")

> choice[choice==""] = NA #save empty value as NA value

> choice = na.omit(choice) #omit NA value

> choice = unique.data.frame(choice) #omit duplications

> num\_choice = nrow(choice) #num of choice

> num\_choice

[1] 2773

>

> #Number of missing test score

> allscore = dim(testscore)

> score = na.omit(testscore) #delete replications

> score = dim(score) #number of rows

> missingscore = allscore - score

> missingscore

[1] 179887 0

>

> #Number of applying to the same school

> apply1 = cbind(schoolcode1,studentcode) #combine info

> apply2 = cbind(schoolcode2,studentcode)

> apply3 = cbind(schoolcode3,studentcode)

> apply4 = cbind(schoolcode4,studentcode)

> apply5 = cbind(schoolcode5,studentcode)

> apply6 = cbind(schoolcode6,studentcode)

> apply = rbind(apply1,apply2,apply3,apply4,apply5,apply6) #stack the above vectors

> apply = na.omit(apply) #omit NA

> apply = unique.data.frame(apply,fromlast=F) #delete duplicate values

> applyfactor = factor(apply[,1])

> applyfactor = as.data.frame(table(applyfactor))

> colnames(applyfactor)=c("schoolcode","num\_apply")

>

> #Number of students apply to less than 6 choices

> sixchoices = cbind(choicepgm1,choicepgm2,choicepgm3,choicepgm4,choicepgm5,choicepgm6)

> sixchoices[sixchoices==""] = NA

> lesssixchoicesna = na.omit(sixchoices) #delete replications

> lesssixchoicesna = nrow(lesssixchoicesna)

> num\_lesssixchoices = numstudent - lesssixchoicesna

> num\_lesssixchoices

[1] 20988

>

> #Exercise 2: Data

> school=datsss[,2:6] #subset school info

> school=na.omit(school) #omit na value

> school=unique.data.frame(school) #omit duplications

> schoollist=school[!duplicated(school$schoolcode),] #list of school

> rankchoice = as.data.frame(datstu[c(1,2,5:16,18,17)])

> rank1 = subset(rankchoice, rankchoice[,15] == 1, select = c(X,score,schoolcode1,choicepgm1)) #subset admitted data

> colnames(rank1)=c("X","score","schoolcode","program")

> rank2 = subset(rankchoice, rankchoice[,15] == 2, select = c(X,score,schoolcode2,choicepgm2))

> colnames(rank2)=c("X","score","schoolcode","program")

> rank3 = subset(rankchoice, rankchoice[,15] == 3, select = c(X,score,schoolcode3,choicepgm3))

> colnames(rank3)=c("X","score","schoolcode","program")

> rank4 = subset(rankchoice, rankchoice[,15] == 4, select = c(X,score,schoolcode4,choicepgm4))

> colnames(rank4)=c("X","score","schoolcode","program")

> rank5 = subset(rankchoice, rankchoice[,15] == 5, select = c(X,score,schoolcode5,choicepgm5))

> colnames(rank5)=c("X","score","schoolcode","program")

> rank6 = subset(rankchoice, rankchoice[,15] == 6, select = c(X,score,schoolcode6,choicepgm6))

> colnames(rank6)=c("X","score","schoolcode","program")

> rankall = rbind(rank1,rank2,rank3,rank4,rank5,rank6)

> colnames(rankall)=c("X","score","schoolcode","program")

> min\_score = as.data.frame(aggregate(score~schoolcode+program, data = rankall, FUN = "min")) #calculate cutoff for each choice

> colnames(min\_score) = c("schoolcode","program","cutoff")

> mean\_score = as.data.frame(aggregate(score~schoolcode+program, data = rankall, FUN = "mean")) #calculate quality for each choice

> colnames(mean\_score) = c("schoolcode","program","quality")

> num\_score = as.data.frame(table(rankall[,3:4])) #calculate # of admitted for each choice

> colnames(num\_score) = c("schoolcode","program","size")

> admitted = cbind(min\_score, mean\_score = mean\_score[,3]) #combine info

> admitted = merge(admitted, num\_score, by.admitted = c(1,2), by.num\_score = c(1,2), all = FALSE, sort=TRUE) #vlookup to add "num\_admitted"

> admitted = merge(admitted, schoollist, by.admitted = schoolcode, by.schoollist = schoolcode, all.x = TRUE, sort=TRUE) #vlookup to add geometrical info

>

> #Exercise 3: Distance

> sdis=datsss[,4:6] #subset sssdistrict info

> sdis=unique.data.frame(sdis) #omit duplications

> sdis=na.omit(sdis) #omit na value

> jdis=datjss[,2:4] #subset jssdistrict info

> colnames(jdis)=c("jssdistrict","jsslong","jsslat")

> sdis=sdis[rep(1:nrow(sdis),each=139),] #replicate each row by 139 times

> jdis=jdis[rep(1:nrow(jdis),time=111),] #replicate each row wholly by 111 times

> distance=cbind(sdis[,1],jdis[,1],sdis[,2:3],jdis[,2:3]) #combine geometrical info

> colnames(distance)=c("sssdistrict","jssdistrict","ssslong","ssslat","jsslong","jsslat")

> ddistance=as.data.frame(sqrt((69.172\*(distance[,3]-distance[,5])\*cos(distance[,6]/57.3))^2+(69.172\*(distance[,4]-distance[,6]))^2)) #calculate distance

> colnames(ddistance)=c("distance")

> distance=cbind(distance,ddistance) #combine info

>

> #Exercise 4 Descriptive Characteristics

> rank11 = as.data.frame(datstu[c(1,2,5,11,17)])#combine info

> colnames(rank11)=c("X","score","schoolcode","program","jssdistrict")

> rank11[rank11==""] = NA #save empty value as NA value

> rankchoice1=merge(rank11,admitted, by=c("schoolcode","program"), all.x= TRUE,sort=TRUE) #vlookup to add cutoff, quality and sssdistrict

> rankchoice11=merge(rankchoice1,distance, by=c("sssdistrict","jssdistrict"), all.x=TRUE, sort=TRUE) #vlookup to add distance

> rankchoice1=rankchoice11[,c(7,8,17)] #subset "cutoff", "quality", "distance" info

> averank1=colMeans(rankchoice1,na.rm=TRUE) #calculate columns average

> averank1

cutoff mean\_score distance

294.23718 316.68978 28.24322

> sd\_cutoffrank1=sd(rankchoice1[,1],na.rm=TRUE) #calculate sd

> sd\_cutoffrank1

[1] 54.12556

> sd\_qualityrank1=sd(rankchoice1[,2],na.rm=TRUE)

> sd\_qualityrank1

[1] 48.5386

> sd\_distancerank1=sd(rankchoice1[,3],na.rm=TRUE)

> sd\_distancerank1

[1] 44.22767

> rank22 = as.data.frame(datstu[c(1,2,6,12,17)]) #repeat above steps for rank2

> colnames(rank22)=c("X","score","schoolcode","program","jssdistrict")

> rank22[rank22==""] = NA

> rankchoice2=merge(rank22,admitted, by=c("schoolcode","program"), all.x= TRUE,sort=TRUE)

> rankchoice22=merge(rankchoice2,distance, by=c("sssdistrict","jssdistrict"), all.x=TRUE, sort=TRUE)

> rankchoice2=rankchoice22[,c(7,8,17)]

> averank2=colMeans(rankchoice2,na.rm=TRUE)

> averank2

cutoff mean\_score distance

281.35659 304.58320 28.16671

> sd\_cutoffrank2=sd(rankchoice2[,1],na.rm=TRUE)

> sd\_cutoffrank2

[1] 49.57892

> sd\_qualityrank2=sd(rankchoice2[,2],na.rm=TRUE)

> sd\_qualityrank2

[1] 43.82404

> sd\_distancerank2=sd(rankchoice2[,3],na.rm=TRUE)

> sd\_distancerank2

[1] 42.54828

> rank33 = as.data.frame(datstu[c(1,2,7,13,17)]) #repeat above steps for rank3

> colnames(rank33)=c("X","score","schoolcode","program","jssdistrict")

> rank33[rank33==""] = NA

> rankchoice3=merge(rank33,admitted, by=c("schoolcode","program"), all.x= TRUE,sort=TRUE)

> rankchoice33=merge(rankchoice3,distance, by=c("sssdistrict","jssdistrict"), all.x=TRUE, sort=TRUE)

> rankchoice3=rankchoice33[,c(7,8,17)]

> averank3=colMeans(rankchoice3,na.rm=TRUE)

> averank3

cutoff mean\_score distance

272.68599 296.78969 27.31817

> sd\_cutoffrank3=sd(rankchoice3[,1],na.rm=TRUE)

> sd\_cutoffrank3

[1] 46.96337

> sd\_qualityrank3=sd(rankchoice3[,2],na.rm=TRUE)

> sd\_qualityrank3

[1] 41.16625

> sd\_distancerank3=sd(rankchoice3[,3],na.rm=TRUE)

> sd\_distancerank3

[1] 41.09999

> rank44 = as.data.frame(datstu[c(1,2,8,14,17)]) #repeat above steps for rank4

> colnames(rank44)=c("X","score","schoolcode","program","jssdistrict")

> rank44[rank44==""] = NA

> rankchoice4=merge(rank44,admitted, by=c("schoolcode","program"), all.x= TRUE,sort=TRUE)

> rankchoice44=merge(rankchoice4,distance, by=c("sssdistrict","jssdistrict"), all.x=TRUE, sort=TRUE)

> rankchoice4=rankchoice44[,c(7,8,17)]

> averank4=colMeans(rankchoice4,na.rm=TRUE)

> averank4

cutoff mean\_score distance

263.07662 288.57669 24.36015

> sd\_cutoffrank4=sd(rankchoice4[,1],na.rm=TRUE)

> sd\_cutoffrank4

[1] 45.1399

> sd\_qualityrank4=sd(rankchoice4[,2],na.rm=TRUE)

> sd\_qualityrank4

[1] 39.2662

> sd\_distancerank4=sd(rankchoice4[,3],na.rm=TRUE)

> sd\_distancerank4

[1] 39.02596

> rank55 = as.data.frame(datstu[c(1,2,9,15,17)]) #repeat above steps for rank5

> colnames(rank55)=c("X","score","schoolcode","program","jssdistrict")

> rank55[rank55==""] = NA

> rankchoice5=merge(rank55,admitted, by=c("schoolcode","program"), all.x= TRUE,sort=TRUE)

> rankchoice55=merge(rankchoice5,distance, by=c("sssdistrict","jssdistrict"), all.x=TRUE, sort=TRUE)

> rankchoice5=rankchoice55[,c(7,8,17)]

> averank5=colMeans(rankchoice5,na.rm=TRUE)

> averank5

cutoff mean\_score distance

250.08489 277.50407 28.67916

> sd\_cutoffrank5=sd(rankchoice5[,1],na.rm=TRUE)

> sd\_cutoffrank5

[1] 32.06955

> sd\_qualityrank5=sd(rankchoice5[,2],na.rm=TRUE)

> sd\_qualityrank5

[1] 26.73123

> sd\_distancerank5=sd(rankchoice5[,3],na.rm=TRUE)

> sd\_distancerank5

[1] 28.32131

> rank66 = as.data.frame(datstu[c(1,2,10,16,17)]) #repeat above steps for rank6

> colnames(rank66)=c("X","score","schoolcode","program","jssdistrict")

> rank66[rank66==""] = NA

> rankchoice6=merge(rank66,admitted, by=c("schoolcode","program"), all.x= TRUE,sort=TRUE)

> rankchoice66=merge(rankchoice6,distance, by=c("sssdistrict","jssdistrict"), all.x=TRUE, sort=TRUE)

> rankchoice6=rankchoice66[,c(7,8,17)]

> averank6=colMeans(rankchoice6,na.rm=TRUE)

> averank6

cutoff mean\_score distance

246.18031 274.07874 29.48621

> sd\_cutoffrank6=sd(rankchoice6[,1],na.rm=TRUE)

> sd\_cutoffrank6

[1] 31.44553

> sd\_qualityrank6=sd(rankchoice6[,2],na.rm=TRUE)

> sd\_qualityrank6

[1] 26.23097

> sd\_distancerank6=sd(rankchoice6[,3],na.rm=TRUE)

> sd\_distancerank6

[1] 28.42286

>

> #Redo: by student test score quartiles

> rankall\_quartile=merge(rankall,admitted,by=c("schoolcode","program"), all.x=TRUE, sort=TRUE) #merge to add sssdistrict info

> rankall\_quartile=merge(rankall\_quartile, datstu,by=c("X"), all.x=TRUE, sort=TRUE) #merge to add jssdistrict info

> rankall\_quartile=merge(rankall\_quartile, distance, by=c("jssdistrict","sssdistrict"), all.x=TRUE, sort=TRUE) #merge to add distance info

> rankall\_quartile=rankall\_quartile[,c(3,6,4,5,7,8,33)] #subset neccessary info

> colnames(rankall\_quartile)=c("X","score","schoolcode","program","cutoff","quality","distance")

> rankall\_quartile=rankall\_quartile[order(rankall\_quartile$score),] #order by score

> q1rank=rankall\_quartile[1:34806,c(5,6,7)] #make quartile by ordering in to 4 groups

> q2rank=rankall\_quartile[34807:69612,c(5,6,7)]

> q3rank=rankall\_quartile[69613:104418,c(5,6,7)]

> q4rank=rankall\_quartile[104419:139224,c(5,6,7)]

> ave\_q1rank=colMeans(q1rank, na.rm = TRUE) #take average of cutoff, quality and distance of group 1

> ave\_q1rank

cutoff quality distance

217.64426 252.22711 25.81073

> sd\_q1rank\_cutoff=sd(q1rank[,1],na.rm = TRUE) #take sd of cutoff of group 1

> sd\_q1rank\_cutoff

[1] 14.7006

> sd\_q1rank\_quality=sd(q1rank[,2],na.rm = TRUE) #take sd of quality of group 1

> sd\_q1rank\_quality

[1] 12.58252

> sd\_q1rank\_distance=sd(q1rank[,3],na.rm = TRUE) #take sd of distance of group 1

> sd\_q1rank\_distance

[1] 43.21944

> ave\_q2rank=colMeans(q2rank, na.rm = TRUE) #repeat above steps for group 2

> ave\_q2rank

cutoff quality distance

242.7788 272.8719 28.4397

> sd\_q2rank\_cutoff=sd(q2rank[,1],na.rm = TRUE)

> sd\_q2rank\_cutoff

[1] 23.52717

> sd\_q2rank\_quality=sd(q2rank[,2],na.rm = TRUE)

> sd\_q2rank\_quality

[1] 17.55403

> sd\_q2rank\_distance=sd(q2rank[,3],na.rm = TRUE)

> sd\_q2rank\_distance

[1] 48.12314

> ave\_q3rank=colMeans(q3rank, na.rm = TRUE) #repeat above steps for group 3

> ave\_q3rank

cutoff quality distance

275.80923 301.68146 31.24321

> sd\_q3rank\_cutoff=sd(q3rank[,1],na.rm = TRUE)

> sd\_q3rank\_cutoff

[1] 29.0557

> sd\_q3rank\_quality=sd(q3rank[,2],na.rm = TRUE)

> sd\_q3rank\_quality

[1] 23.08222

> sd\_q3rank\_distance=sd(q3rank[,3],na.rm = TRUE)

> sd\_q3rank\_distance

[1] 47.22777

> ave\_q4rank=colMeans(q4rank, na.rm = TRUE) #repeat above steps for group 4

> ave\_q4rank

cutoff quality distance

337.06686 357.25908 38.54292

> sd\_q4rank\_cutoff=sd(q4rank[,1],na.rm = TRUE)

> sd\_q4rank\_cutoff

[1] 39.38961

> sd\_q4rank\_quality=sd(q4rank[,2],na.rm = TRUE)

> sd\_q4rank\_quality

[1] 35.20401

> sd\_q4rank\_distance=sd(q4rank[,3],na.rm = TRUE)

> sd\_q4rank\_distance

[1] 46.35148

>

> #Exercise 5

> #group choice by cutoffs

> schoolgroups=admitted[,c(1,2,3)]

> schoolgroups=schoolgroups[order(schoolgroups$cutoff),]

> d1school=as.data.frame(schoolgroups[1:230,]) #make decile by ordering in to 10 groups

> a = data.frame(groupnumber=1)

> a = as.vector(a[rep(1:nrow(a),each=230),]) #create a (1\*230) vetor of 1

> d1school=cbind(d1school,a) #combine group 1 with group number(1)

> d2school=schoolgroups[231:460,] #repeat above steps for group 2

> a = data.frame(groupnumber=2)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d2school=cbind(d2school,a)

> d3school=schoolgroups[461:690,] #repeat above steps for group 3

> a = data.frame(groupnumber=3)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d3school=cbind(d3school,a)

> d4school=schoolgroups[691:920,] #repeat above steps for group 4

> a = data.frame(groupnumber=4)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d4school=cbind(d4school,a)

> d5school=schoolgroups[921:1150,] #repeat above steps for group 5

> a = data.frame(groupnumber=5)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d5school=cbind(d5school,a)

> d6school=schoolgroups[1151:1380,] #repeat above steps for group 6

> a = data.frame(groupnumber=6)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d6school=cbind(d6school,a)

> d7school=schoolgroups[1381:1610,] #repeat above steps for group 7

> a = data.frame(groupnumber=7)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d7school=cbind(d7school,a)

> d8school=schoolgroups[1611:1840,] #repeat above steps for group 8

> a = data.frame(groupnumber=8)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d8school=cbind(d8school,a)

> d9school=schoolgroups[1841:2070,] #repeat above steps for group 9

> a = data.frame(groupnumber=9)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d9school=cbind(d9school,a)

> d10school=schoolgroups[2071:2300,] #repeat above steps for group 10

> a = data.frame(groupnumber=10)

> a = as.vector(a[rep(1:nrow(a),each=230),])

> d10school=cbind(d10school,a)

> dschool=rbind(d1school,d2school,d3school,d4school,d5school,d6school,d7school,d8school,d9school,d10school) #create database of schoolcode, program, cutoff and groupnumber

> colnames(dschool)=c("schoolcode","program","cutoff","groupnumber")

> rankalll=rbind(rank11,rank22,rank33,rank44,rank55,rank66) #create database of choices

> rankalll=rankalll[,c(1,3,4)]

> rankalll[rankalll==""] = NA

> rankalll=na.omit(rankalll) #delete NA value

> drankalll=merge(rankalll,dschool,by=c("schoolcode","program"),all.x=TRUE, sort=TRUE) #merge to add groupnumber for each choice

> drankalll=drankalll[,c(3,5)]

> drankalll=unique(drankalll) #delete duplication of same group for each student

> num\_groups\_cutoff = as.data.frame(table(drankalll[,1])) #calculate num\_groups for each student

> colnames(num\_groups\_cutoff)=c("X","num\_groups\_cutoff")

>

> #redo: group choice by qualities

> schoolgroups2=admitted[,c(1,2,4)]

> schoolgroups2=schoolgroups2[order(schoolgroups2$quality),]

Error in order(schoolgroups2$quality) : argument 1 is not a vector

> q1school=as.data.frame(schoolgroups2[1:575,]) #make quartile by ordering in to 4 groups

> a = data.frame(groupnumber=1)

> a = as.vector(a[rep(1:nrow(a),each=575),]) #create a (1\*575) vetor of 1

> q1school=cbind(q1school,a) #combine group 1 with group number(1)

> q2school=as.data.frame(schoolgroups2[576:1150,]) #repeat above steps for group 2

> a = data.frame(groupnumber=2)

> a = as.vector(a[rep(1:nrow(a),each=575),])

> q2school=cbind(q2school,a)

> q3school=as.data.frame(schoolgroups2[1151:1725,]) #repeat above steps for group 3

> a = data.frame(groupnumber=3)

> a = as.vector(a[rep(1:nrow(a),each=575),])

> q3school=cbind(q3school,a)

> q4school=as.data.frame(schoolgroups2[1726:2300,]) #repeat above steps for group 4

> a = data.frame(groupnumber=4)

> a = as.vector(a[rep(1:nrow(a),each=575),])

> q4school=cbind(q4school,a)

> qschool=rbind(q1school,q2school,q3school,q4school) #create database of schoolcode, program, quality and groupnumber

> colnames(qschool)=c("schoolcode","program","quality","groupnumber")

> qrankalll=merge(rankalll,qschool,by=c("schoolcode","program"),all.x=TRUE, sort=TRUE) #merge to add groupnumber for each choice

> qrankalll=qrankalll[,c(3,5)]

> qrankalll=unique(qrankalll)

> num\_groups\_quality = as.data.frame(table(qrankalll[,1])) #calculate num\_groups fro each student

> colnames(num\_groups\_quality)=c("X","num\_groups\_quality")