Exercise2 Output

> datapply<-cbind(datstu[,5:16],datstu[,2],datstu[,18],datstu[,1],datstu[,17])

> dim(datapply)

[1] 340823 16

> #[1] 340823 16

> #Create a dataframe which includes shoolcodes, programs, scores and admission information.

> datapply<-datapply[complete.cases(datapply[,14]),]

> dim(datapply)

[1] 160935 16

> #[1] 160935 16

> # Omit the rows whose rank places are NA

> datapply<-datapply[datapply[,14]!=99,]

> #Omit the rows whose rank places are 99

> colnames(datapply)=c("schoolcode1","schoolcode2","schoolcode3","schoolcode4","schoolcode5","schoolcode6","choicepgm1","choicepgm2","choicepgm3","choicepgm4","choicepgm5","choicepgm6","score","rank","id","jssdistrict")

>

> rank1<-subset.data.frame(datapply,datapply$rank==1,select = c(schoolcode1,choicepgm1,score,id,jssdistrict))

> colnames(rank1)<-c("schoolcode","choicepgm","score","id","jssdistrict")

> rank2<-subset.data.frame(datapply,datapply$rank==2,select = c(schoolcode2,choicepgm2,score,id,jssdistrict))

> colnames(rank2)<-c("schoolcode","choicepgm","score","id","jssdistrict")

> rank3<-subset.data.frame(datapply,datapply$rank==3,select = c(schoolcode3,choicepgm3,score,id,jssdistrict))

> colnames(rank3)<-c("schoolcode","choicepgm","score","id","jssdistrict")

> rank4<-subset.data.frame(datapply,datapply$rank==4,select = c(schoolcode4,choicepgm4,score,id,jssdistrict))

> colnames(rank4)<-c("schoolcode","choicepgm","score","id","jssdistrict")

> rank5<-subset.data.frame(datapply,datapply$rank==5,select = c(schoolcode5,choicepgm5,score,id,jssdistrict))

> colnames(rank5)<-c("schoolcode","choicepgm","score","id","jssdistrict")

> rank6<-subset.data.frame(datapply,datapply$rank==6,select = c(schoolcode6,choicepgm6,score,id,jssdistrict))

> colnames(rank6)<-c("schoolcode","choicepgm","score","id","jssdistrict")

>

> rank<-rbind(rank1[1:nrow(rank1),],rank2[1:nrow(rank2),],rank3[1:nrow(rank3),],rank4[1:nrow(rank4),],rank5[1:nrow(rank5),],rank6[1:nrow(rank6),])

> # Generate the admission information

> unique(datsss[,3])

[1] 30107 30103 21003 10111 30104 20301 50110 10110 40103 21103 50102 10102 30301 50108

[15] 60106 20102 20402 50107 30101 30102 50201 30106 10112 70101 20104 50105 10121 10117

[29] 50502 50104 30401 70102 10201 50103 60104 30601 50113 50101 51701 70601 61201 40104

[43] 21501 10103 51002 21302 20303 30905 60107 40105 50111 20103 50114 30501 20101 21303

[57] 50301 10105 10501 80101 90505 70602 10114 10106 10401 50106 30105 70502 100201 50109

[71] 40107 50202 40102 50115 20302 51202 80107 50203 30108 10115 10202 31001 60201 100104

[85] 60301 10101 50112 20601 61001 50119 90403 40904 21002 21306 9050101 30402 20803 20403

[99] 20501 20201 20401 80102 30903 51802 30801 40106 40101 51602 10116 70112 10502 50204

[113] 30902 90501 40201 10104 50801 20410 21102 51603 50605 10213 50503 60701 30502 31201

[127] 10107 51801 9010101 50602 10118 20105 60601 50601 50704 100302 51401 51601 60101 30304

[141] 10301 20305 10108 100102 70901 20602 70104 9020101 9021301 10205 70106 71005 51201 60303

[155] 20304 10302 20106 60501 21006 40903 30109 9071001 21502 71006 71201 70501 90602 10142

[169] 61102 51203 10128 70504 10109 21004 40108 10504 80104 60203 30602 100503 50307 10210

[183] 21001 9010201 31101 50701 50302 20603 31002 10119 90101 40301 51001 50702 90401 21101

[197] 30201 20902 60206 9021001 70114 21201 70703 9030401 30701 21301 31202 30504 71002 40701

[211] 10120 60105 9030101 51604 10503 50205 40902 100202 20202 21305 20503 40601 51803 40501

[225] 61203 80103 70806 60702 9020901 70603 50501 70701 50139 71007 21401 50606 51301 31003

[239] 100401 40111 90402 100105 50703 60502 50401 80301 9040101 30901 61202 60801 80105 30303

[253] 10203 60505 21304 20408 30603 20406 50504 80108 20801 90201 60302 20405 90301 50802

[267] 70704 71010 30302 40901 61101 61003 60202 80106 21504 71101 40202 40112 50603 100106

[281] 20502 9070901 40702 10403 90303 61301 80801 70604 40302 70401 70801 50304 21005 30110

[295] 80201 50303 71004 60103 51204 9021002 40110 9070601 51102 30305 30702 51302 50705 40401

[309] 60901 70301 41002 30503 9051801 70202 51605 70302 90405 20901 21007 20607 51205 50707

[323] 51003 70103 50803 60602 70805 80501 70108 70505 40502 21202 71009 9070501 50604 51804

[337] 100501 20606 90601 70111 70802 10204 20407 60402 70605 31204 41001 50305 50206 70902

[351] 9100101 9010301 9040301 51103 20802 51501 80302 50901 60403 9080103 80701 81201 70705 70110

[365] 60404 41101 31205 9010102 71103 71102 71008 9090401 61005 20604 70201 50402 71202 20704

[379] 70109 30802 60605 90103 31206 61002 90202 100304 30403 70119 21009 60604 40604 50706

[393] 20404 80601 60802 60205 70804 70506 9080901 61302 100203 60401 51101 70107 100402 80602

[407] 71003 51402 40801 71105 70402 80109 21203 80303 50902 90302 80401 40802 50207 30203

[421] 40109 90504 9090101 60703 70803 51606 81102 20409 51702 61004 60603 70607 51206 9021101

[435] 70507 70508 70608 70606 70403 61204 60304 70113 21008 61303 30904 51004 50804 70105

[449] 21307 20805 20703 50403 60102 60504 60902 20203 80202 20702 71001 60204 40603 20701

[463] 81101 60503 71104 40602 21104 21601 30202 60704 70509 21010 10402 60705 20608 20204

[477] 80902 40304 90404 70303 70702 70503 70203 21503 40303 80402 40402 40906 80901 41102

[491] 40503 80503 90502 80802 80502 80110 100502 30703 81202 81301 70204 90102 81302 81001

[505] 31102 90503 31203 21402 100303 21505 40905 20804 80702 90506 100204 100301 100101 10144

[519] 70708 30406 60707 9010124 50155 10506 31408 50156 10126 9010119 10140 50164 31207 9010108

[533] 51304 30604 21311 10211 9060901 9010120 50125 9010114 40306 50150 70709 10148 40703 90507

[547] 50133 10163 70706 9050105 9010103 9100102 9010115 51607 10124 20109 50144 9040103 81401 21403

[561] 9060102 10125 9070401 9010118 9030801 80117 50145 10145 61304 50136 20306 30907 20108 100403

[575] 30404 9060114 9060101 9100201 50158 9050106 60607 100107 10123 20505 10131 10173 70122 51209

[589] 9050107 9100501 9020103 9020102 10168 50208 9021501 9010107 51208 10141 60608 40803 9010203 9010112

[603] 50607 9010116 50153 9010113 20113 9071002 70121 9080603 70118 9050104 60109 10158 10510 80111

[617] 10149 9010105 30506 50805 9060201 21012 20107 9020301 10216 51805 51608 60108 10404 21408

[631] 9020501 60708 40504 21310 9021401 9010111 9021601 50163 9010104 9080703 71203 9011001 10146 9060202

[645] 80604 10175 10505 50148 10512 51704 50162 71106 10154 9081501 30607 9010202 10135 40305

[659] 50116 10212 90508 60111 70125 61006 9050103 9010204 10129 60904 60611 20307 9100401 10220

[673] 10511 51005 40114 70510 10164 30405 9060103 100504 40307 9000201 50128 40704 9010117 9030501

[687] 9030301 30306 70907 50903 50120 20609 9000301 10214 9021502 10509 80203 60609 30908 10215

[701] 9070602 9010106 80204 9080102 90509 9050102 50123 9010109 90603 30505 50126 50154 20110 60903

[715] 21406 30605 10138 9010501 50161 50140 9070701 100205 80504 30906 10153 100103 70117 50146

[729] 80114 10514 50165 9040102 50129 50135 50310 80116 31103 50141 50149 9060502 50117 60507

[743] 10516 50309 21404 50166 50121 70807 50172 70205 70899 21308 9060104 10147 60208 10133

[757] 30606 10601 10151 9060701 50160 21508 50143 50168 10159 9051001 80112 10508 70206 50507

[771] 30704 20112 60610 31299 70120 50505 60113 10162 50138 9071003 40113 10155 60706 21312

[785] 60709 60699 50142 10139 60207 21309 10137 80115 10221 10161 30910 51703 50131 61406

[799] 10166 30114 9070101 10143 50508 50137 10218 10208 10156 10171 10170 50169 90510 10217

[813] 31209 40605 51207 9021201 10507 20605 50173 10219 10157 10160 31208 10517 90406 20504

[827] 80205 50118 100206 50311 9060501 70123 10122 30608 10513 9050192 9010123 9010110 20806 10405

[841] 50157 70707 50132 71204 40115 50159 10206 50171 31104 10152 10165 50167 21507 20111

[855] 60112 30909 71299 10174 51411 71011 21013 31004 50708 20610 50170 10172 10167 50147

[869] 50134 60307 10132 90407 80113 10150 50124 70116 10599 60110 10134 60506 20807 21405

[883] 50122 61007 21014 50130 9080101 50209 21506 30111 30112 50151 50152 60306 10169 71107

[897] 30204 80703

> datsss\_new<-datsss[!duplicated(datsss[,3]),]

> datsss\_new<-datsss\_new[complete.cases(datsss\_new),]

> # Make the schoolcode in datsss be unique

>

> datadmit<-cbind(as.data.frame(paste(rank[,1],rank[,2])),rank[,3],rank[,4])

> #Combine the schoolcode and program to prepare for sorting data

>

> quality<-as.data.frame(tapply(as.numeric(datadmit[,2]),datadmit[,1],mean))

> cutoff<-as.data.frame(tapply(as.numeric(datadmit[,2]),datadmit[,1],min))

> size<-as.matrix(table(datadmit[,1]))

>

> aa<-as.vector(row.names(size))

> schoolcode<-as.data.frame(as.numeric(gsub("\\D","",aa)))

> program<-as.data.frame(gsub("\\d","",aa))

> size<-as.data.frame(size)

> new2<-cbind(schoolcode,program,cutoff,quality,size)

> colnames(new2)=c("schoolcode","program","cutoff","quality","size")

>

> datsch<-merge(new2,datsss\_new,by="schoolcode",all.x=TRUE)

> # Complete exercise 2