**DATA 590 | Homework 1 | Praveen Kumar Neelappa**

#Read the data file into R

> af <- read.csv("Asian\_Fusion\_Data.csv");

#Use sqldf to select record,Catkey,sex,NAR,BRR,VRR,LAR,OSR,BAR,SBA

> mydata <- sqldf("select record,Catkey,sex,NAR,BRR,VRR,LAR,OSR,BAR,SBA

from af")

#Test if the variable in data set are normally distributed by using

shapiro test

> shapiro.test(mydata$NAR)

Shapiro-Wilk normality test

data: mydata$NAR

W = 0.99374, p-value = 0.0159

> shapiro.test(mydata$BRR)

Shapiro-Wilk normality test

data: mydata$BRR

W = 0.99176, p-value = 0.00247

> shapiro.test(mydata$VRR)

Shapiro-Wilk normality test

data: mydata$VRR

W = 0.99282, p-value = 0.006623

> shapiro.test(mydata$LAR)

Shapiro-Wilk normality test

data: mydata$LAR

W = 0.99501, p-value = 0.05499

> shapiro.test(mydata$OSR)

Shapiro-Wilk normality test

data: mydata$OSR

W = 0.98966, p-value = 0.0003955

> shapiro.test(mydata$BAR)

Shapiro-Wilk normality test

data: mydata$BAR

W = 0.98888, p-value = 0.0002059

> shapiro.test(mydata$SBA)

Shapiro-Wilk normality test

data: mydata$SBA

W = 0.98717, p-value = 0.0002193

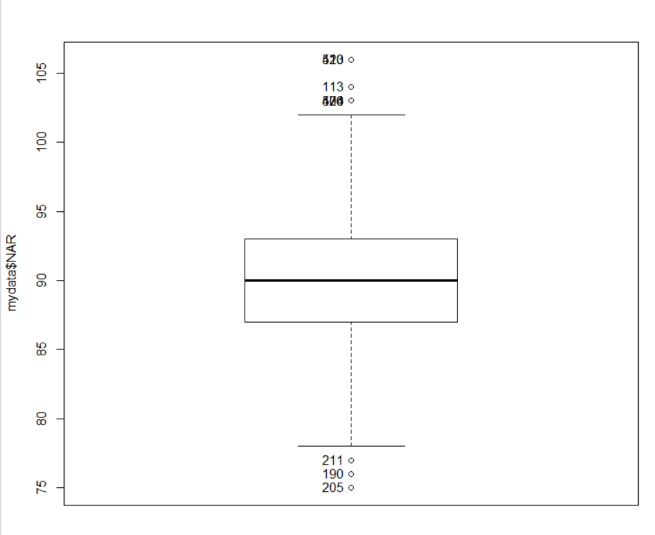
#Use boxplot to identify the outliers

> OUTNAR <- Boxplot(mydata$NAR)

> OUTNAR

[1] 190 205 211 113 404 423 476 510

[9] 520



#Save the outlier in list

> listNAR <- cbind("NAR",mydata[OUTNAR,c("Catkey","NAR")])

> names(listNAR)[1] <- "Var"

> names(listNAR)[3] <- "Val"

> listNAR

"NAR" Catkey NAR

190 NAR JI-5022 76

205 NAR JIKEI-47F 75

211 NAR KU-1116 77

113 NAR B5SC297 104

404 NAR TU-1367 103

423 NAR TU-1471 106

476 NAR TU-2218 103

510 NAR TU-49 106

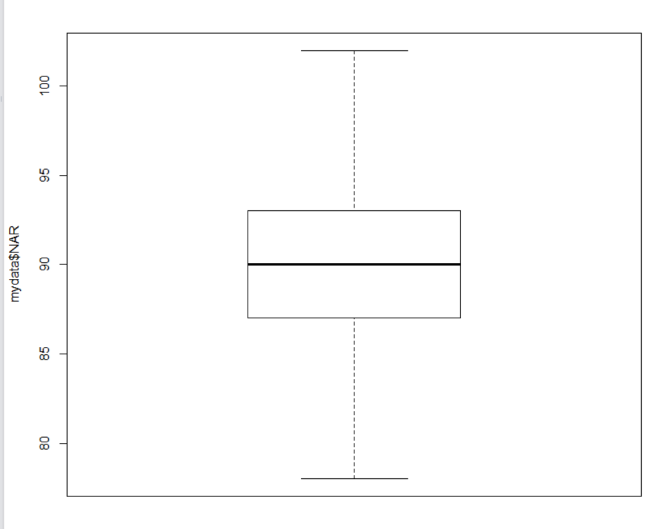
520 NAR Tohoku-1199 103

#assign NA value to these outliers

> mydata[Boxplot(mydata$NAR),"NAR"] <- NA

#check if there is any other outlier after removing the previous outlier

> Boxplot(mydata$NAR)

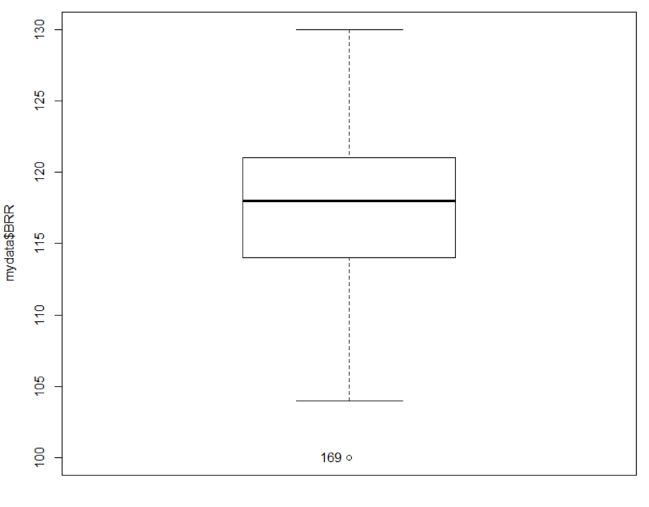


#Use boxplot to identify the outliers

> OUTBRR <- Boxplot(mydata$BRR)

> OUTBRR

[1] 169



#Save the outlier in list

> listBRR <- cbind("BRR",mydata[OUTBRR,c("Catkey","BRR")])

> names(listBRR)[1] <- "Var"

> names(listBRR)[3] <- "Val"

> listBRR

"BRR" Catkey BRR

169 BRR JI-133 100

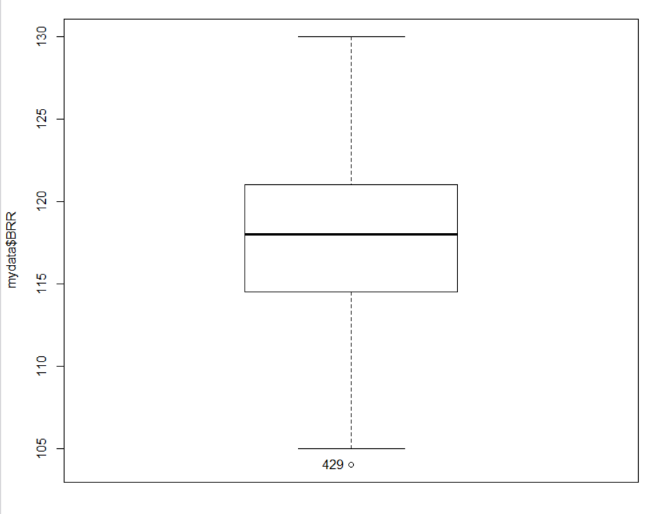
#assign NA value to these outliers

> mydata[Boxplot(mydata$BRR),"BRR"] <- NA

#Use boxplot to identify the outliers

> Boxplot(mydata$BRR)

[1] 429



#Repeat the step to BRR to remove the outlier

> OUTBRR1 <- Boxplot(mydata$BRR)

> listBRR1 <- cbind("BRR",mydata[OUTBRR1,c("Catkey","BRR")])

> mydata[Boxplot(mydata$BRR),"BRR"] <- NA

#Use boxplot to identify the outliers

> Boxplot(mydata$BRR)

[1] 237

#Repeat the step to BRR to remove the outlier

> OUTBRR2 <- Boxplot(mydata$BRR)

> listBRR2 <- cbind("BRR",mydata[OUTBRR2,c("Catkey","BRR")])

> listBRR2

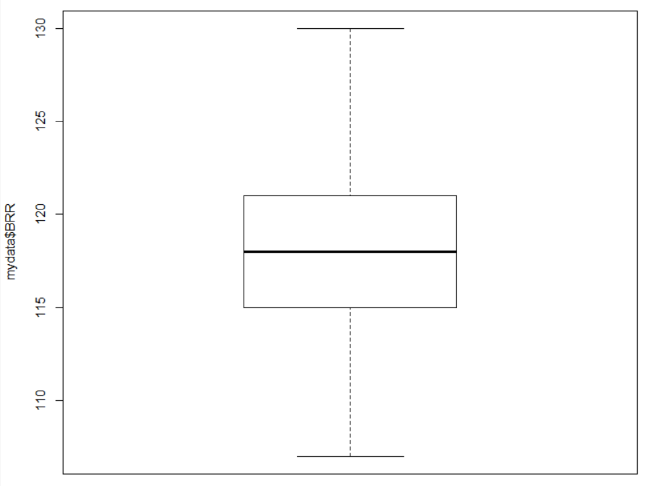
"BRR" Catkey BRR

237 BRR KU-2181 105

> mydata[Boxplot(mydata$BRR),"BRR"] <- NA

#Use boxplot to identify the outliers

> Boxplot(mydata$BRR)

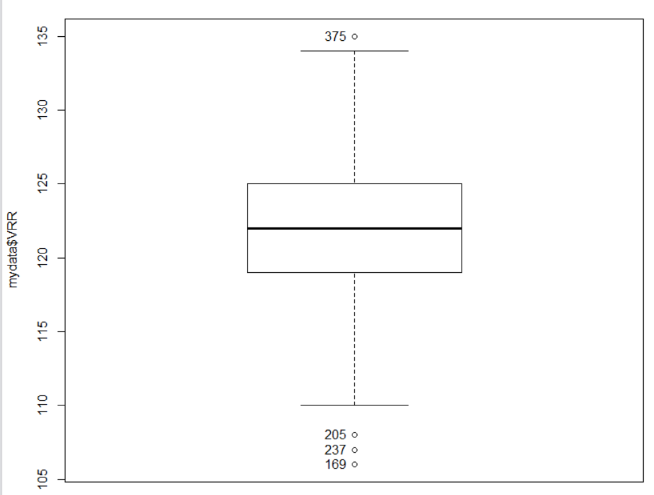


#Use boxplot to identify the outliers

> OUTVRR <- Boxplot(mydata$VRR)

> OUTVRR

[1] 169 205 237 375



#Save the outlier in list

> listVRR <- cbind("VRR",mydata[OUTVRR,c("Catkey","VRR")])

> names(listVRR)[1] <- "Var"

> names(listVRR)[3] <- "Val"

"VRR" Catkey VRR

169 VRR JI-133 106

205 VRR JIKEI-47F 108

237 VRR KU-2181 107

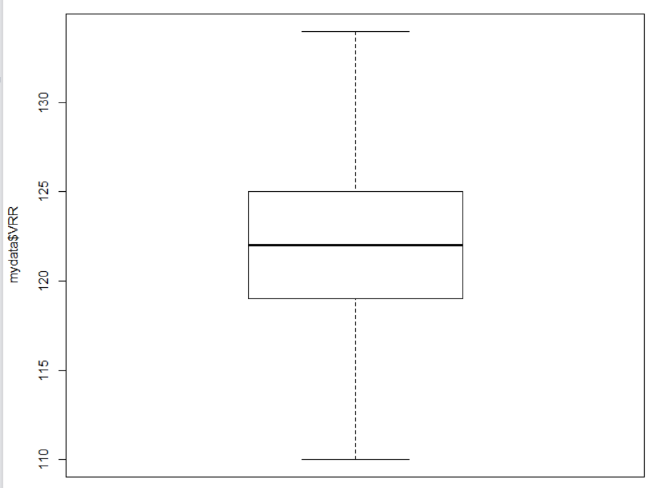
375 VRR T79-288 135

#assign NA value to these outliers

> mydata[Boxplot(mydata$VRR),"VRR"] <- NA

#check if there is any other outlier after removing the previous outlier

> Boxplot(mydata$VRR)

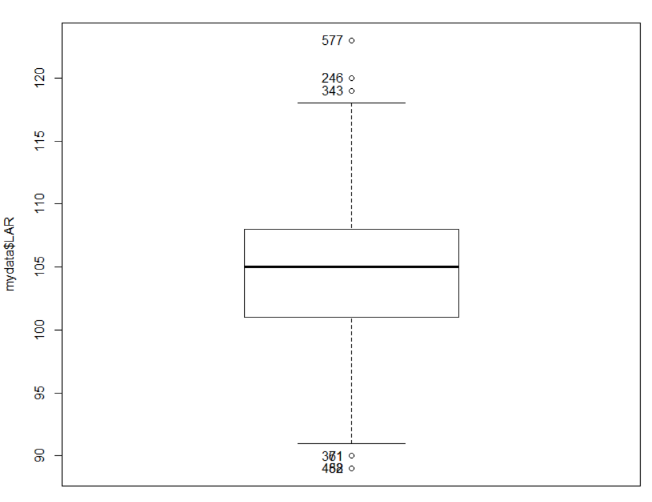


#Use boxplot to identify the outliers

> OUTLAR <- Boxplot(mydata$LAR)

> OUTLAR

[1] 71 88 361 452 246 343 577



#Save the outlier in list

> listLAR <- cbind("LAR",mydata[OUTLAR,c("Catkey","LAR")])

> names(listLAR)[1] <- "Var"

> names(listLAR)[3] <- "Val"

> listLAR

"LAR" Catkey LAR

71 LAR A4SC097 90

88 LAR B2SC182 89

361 LAR T79-107 90

452 LAR TU-2101 89

246 LAR KU-2594 120

343 LAR T78-220 119

577 LAR Tohoku-481 123

#assign NA value to these outliers

> mydata[Boxplot(mydata$LAR),"LAR"] <- NA

#check if there is any other outlier after removing the previous outlier

> Boxplot(mydata$LAR)

#Use boxplot to identify the outliers

> OUTOSR <- Boxplot(mydata$OSR)

#Save the outlier in list

> listOSR <- cbind("OSR",mydata[OUTOSR,c("Catkey","OSR")])

> names(listOSR)[1] <- "Var"

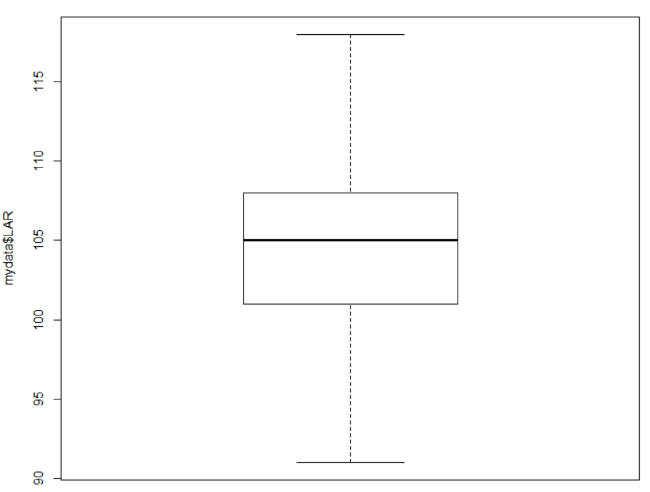
> names(listOSR)[3] <- "Val"

#assign NA value to these outliers

> mydata[Boxplot(mydata$OSR),"OSR"] <- NA

#check if there is any other outlier after removing the previous outlier

> Boxplot(mydata$OSR)



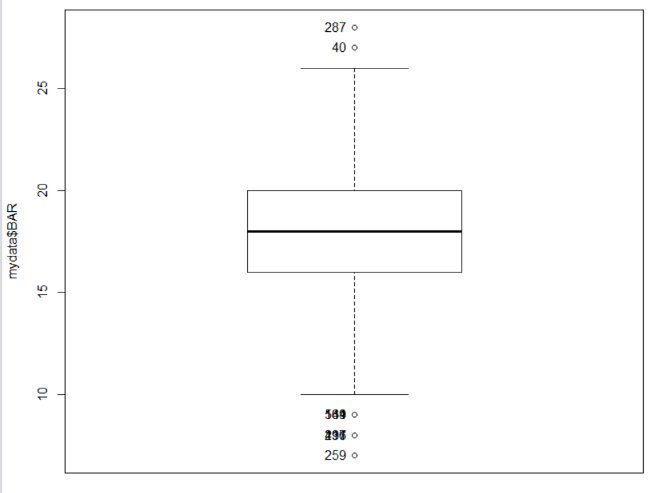
#Use boxplot to identify the outliers

> OUTBAR <- Boxplot(mydata$BAR)

> OUTBAR

[1] 134 136 161 211 259 497 549

[8] 40 287



#Save the outlier in list

> listBAR <- cbind("BAR",mydata[OUTBAR,c("Catkey","BAR")])

> names(listBAR)[1] <- "Var"

> names(listBAR)[3] <- "Val"

> listBAR

"BAR" Catkey BAR

134 BAR C5SC423 9

136 BAR C5SC427 8

161 BAR E2SC652 9

211 BAR KU-1116 8

259 BAR KU-2939 7

497 BAR TU-261 8

549 BAR Tohoku-320 9

40 BAR 09-040 27

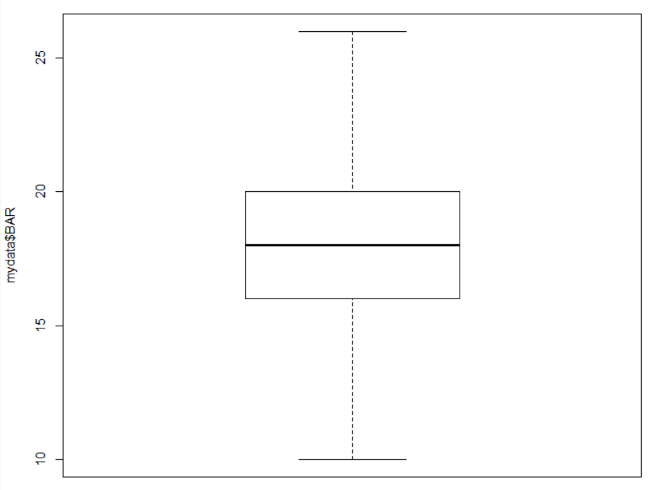
287 BAR T74-249 28

#assign NA value to these outliers

> mydata[Boxplot(mydata$BAR),"BAR"] <- NA

#check if there is any other outlier after removing the previous outlier

> Boxplot(mydata$BAR)



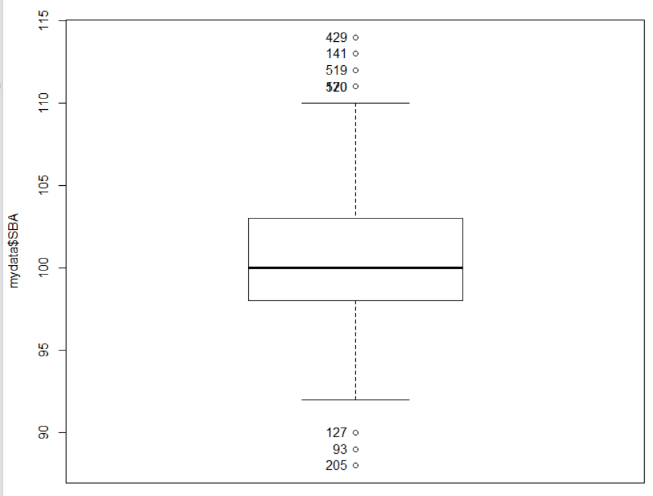
#Use boxplot to identify the outliers

> OUTSBA <- Boxplot(mydata$SBA)

> OUTSBA

[1] 93 127 205 141 170 429 519

[8] 520



#Save the outlier in list

> listSBA <- cbind("SBA",mydata[OUTSBA,c("Catkey","SBA")])

> names(listSBA)[1] <- "Var"

> names(listSBA)[3] <- "Val"

> names(listNAR)

[1] "Var" "Catkey" "Val"

> listSBA

"SBA" Catkey SBA

93 SBA B2SC189 89

127 SBA C3SC390 90

205 SBA JIKEI-47F 88

141 SBA CAST 113

170 SBA JI-178 111

429 SBA TU-1831 114

519 SBA Tohoku-1139 112

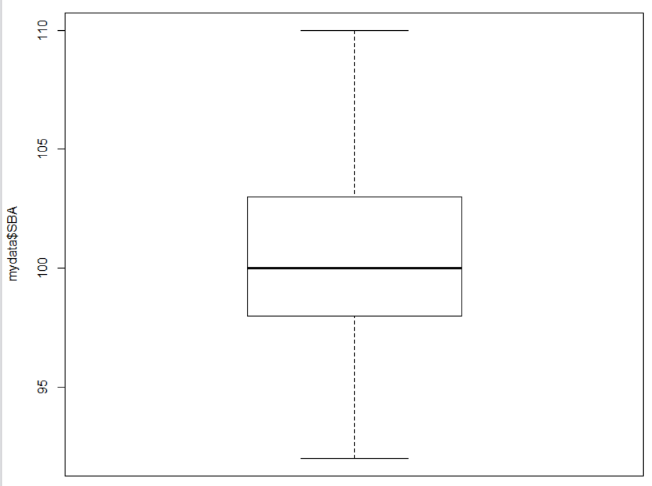
520 SBA Tohoku-1199 111

#assign NA value to these outliers

> mydata[Boxplot(mydata$SBA),"SBA"] <- NA

#check if there is any other outlier after removing the previous outlier

> Boxplot(mydata$SBA)



#Append all the outlier into a final list

> Finallist <- rbind(listNAR,listBRR, listVRR, listLAR, listOSR, listBAR, listSBA)

#Print the final list

> Finallist

Var Catkey Val

190 NAR JI-5022 76

205 NAR JIKEI-47F 75

211 NAR KU-1116 77

113 NAR B5SC297 104

404 NAR TU-1367 103

423 NAR TU-1471 106

476 NAR TU-2218 103

510 NAR TU-49 106

520 NAR Tohoku-1199 103

169 BRR JI-133 100

1691 VRR JI-133 106

2051 VRR JIKEI-47F 108

237 VRR KU-2181 107

375 VRR T79-288 135

71 LAR A4SC097 90

88 LAR B2SC182 89

361 LAR T79-107 90

452 LAR TU-2101 89

246 LAR KU-2594 120

343 LAR T78-220 119

577 LAR Tohoku-481 123

2111 OSR KU-1116 33

570 OSR Tohoku-395 30

324 OSR T77-408 52

337 OSR T78-150 51

435 OSR TU-1917 52

492 OSR TU-2556 53

134 BAR C5SC423 9

136 BAR C5SC427 8

161 BAR E2SC652 9

2112 BAR KU-1116 8

259 BAR KU-2939 7

497 BAR TU-261 8

549 BAR Tohoku-320 9

40 BAR 09-040 27

287 BAR T74-249 28

93 SBA B2SC189 89

127 SBA C3SC390 90

2052 SBA JIKEI-47F 88

141 SBA CAST 113

170 SBA JI-178 111

429 SBA TU-1831 114

519 SBA Tohoku-1139 112

5201 SBA Tohoku-1199 111

#Test if the variable in data set after removing outliers if they are

normally distributed by using shapiro test

> shapiro.test(mydata$NAR)

Shapiro-Wilk normality test

data: mydata$NAR

W = 0.99229, p-value = 0.004453

> shapiro.test(mydata$BRR)

Shapiro-Wilk normality test

data: mydata$BRR

W = 0.99046, p-value = 0.0008029

> shapiro.test(mydata$VRR)

Shapiro-Wilk normality test

data: mydata$VRR

W = 0.99144, p-value = 0.001949

> shapiro.test(mydata$LAR)

Shapiro-Wilk normality test

data: mydata$LAR

W = 0.99328, p-value = 0.01097

> shapiro.test(mydata$OSR)

Shapiro-Wilk normality test

data: mydata$OSR

W = 0.98744, p-value = 7.162e-05

> shapiro.test(mydata$BAR)

Shapiro-Wilk normality test

data: mydata$BAR

W = 0.98741, p-value = 7.329e-05

> shapiro.test(mydata$SBA)

Shapiro-Wilk normality test

data: mydata$SBA

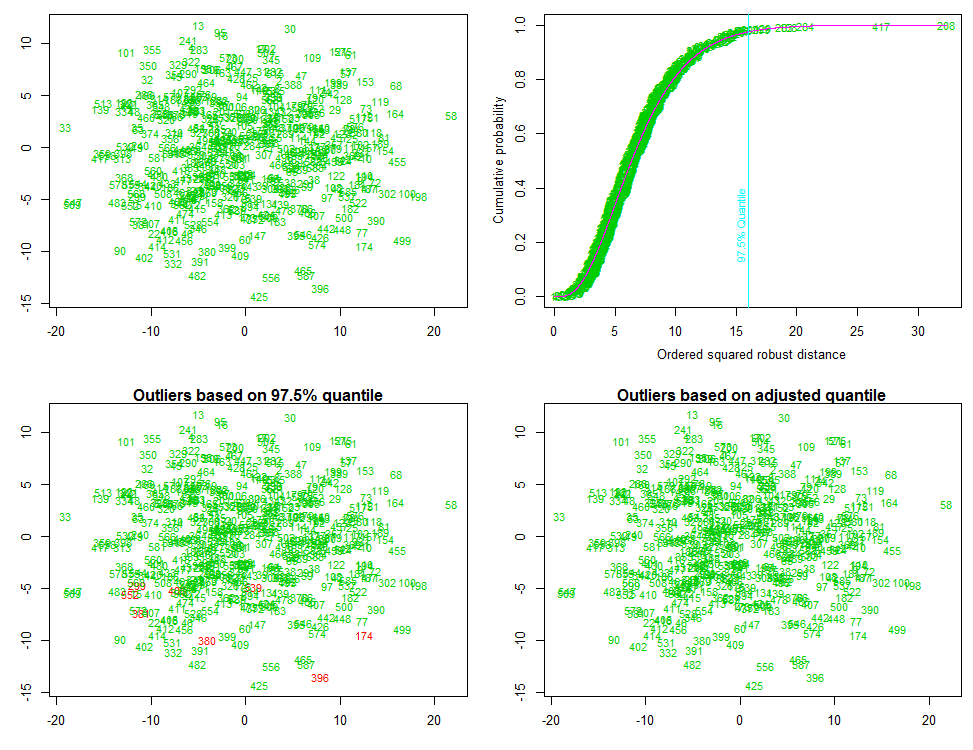
W = 0.98388, p-value = 2.831e-05

**Multivariate Outliers**

> outliers <- aq.plot(na.omit(mydata[c(4:10)]), alpha = 0.01)

Projection to the first and second robust principal components.

Proportion of total variation (explained variance): 0.5045211



> mydata[outliers[[1]],]

[1] Record Catkey Sex NAR BRR VRR

[7] LAR OSR BAR SBA

<0 rows> (or 0-length row.names)

> outliers

$outliers

1 2 3 4 5 6 7 8

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

9 10 11 12 13 14 15 16

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

17 18 19 20 21 22 23 24

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

25 26 27 28 29 30 31 32

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

33 34 35 36 37 38 39 41

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

42 43 44 45 46 47 48 49

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

50 51 52 54 55 56 57 58

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

59 60 61 62 63 64 65 66

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

68 70 72 73 74 75 76 77

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

78 79 80 81 82 83 84 85

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

86 87 89 90 91 92 94 95

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

96 97 98 99 100 101 102 103

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

104 105 106 107 108 109 110 111

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

112 114 115 117 118 119 120 121

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

122 123 124 125 126 128 129 130

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

131 132 133 135 137 138 139 140

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

142 143 144 145 146 147 148 149

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

150 151 152 153 154 155 156 157

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

158 159 160 162 163 164 165 166

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

167 168 171 172 173 174 175 176

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

177 178 179 180 181 182 183 184

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

185 186 187 188 189 191 192 193

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

194 195 196 197 198 199 200 201

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

202 203 204 232 240 241 273 282

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

283 284 285 286 288 289 290 291

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

292 293 294 295 296 297 298 299

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

300 301 302 303 304 305 306 307

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

308 309 310 311 312 313 314 315

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

316 317 318 319 320 321 322 323

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

325 326 327 328 329 330 331 332

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

333 334 335 338 339 340 341 342

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

344 345 346 347 348 349 350 351

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

352 353 354 355 356 357 358 359

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

360 362 363 364 365 366 367 368

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

369 370 371 372 373 374 376 377

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

378 379 380 381 382 383 384 385

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

386 387 388 389 390 391 392 393

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

394 395 396 397 398 399 401 402

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

403 405 406 407 408 409 410 411

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

412 413 414 415 416 417 418 419

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

420 421 422 424 425 426 427 428

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

430 431 432 433 434 436 437 439

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

440 442 443 445 446 447 448 449

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

450 451 453 454 455 456 457 458

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

459 460 461 462 463 464 465 466

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

467 468 471 472 474 475 477 478

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

479 482 483 485 486 487 488 489

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

490 494 495 496 499 500 501 502

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

504 505 506 507 508 509 511 512

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

513 514 515 516 517 518 521 522

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

523 524 525 526 527 528 529 530

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

531 532 533 534 535 536 537 538

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

539 540 541 542 543 544 545 546

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

547 548 550 551 552 553 554 555

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

556 557 558 559 560 561 562 563

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

564 565 566 567 568 569 571 572

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

573 574 575 576 578 579 580 581

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

582 583 584 585 586 587 588 589

FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

#There are no outliers

#This is the penny method for identifying the multivariate outliers

> af <- read.csv("Asian\_Fusion\_Data.csv")

> mydata <- sqldf("select record,Catkey,sex,NAR,BRR,VRR,LAR,OSR,BAR,SBA from af")

>

> x <- vector(mode="numeric", length=0)

> y<- c(1:length(mydata$Catkey))

> for (i in y)

+ {

+ mdist <- mahalanobis(na.omit(mydata[i,c(4:10)]),center=colMeans(na.omit(mydata[-(i),c(4:10)])),cov(na.omit(mydata[-(i),c(4:10)])) )

+

+ if (length(mdist) > 0)

+ {

+ val2 <- 1-pchisq(mdist,7)

+ if(val2 < 0.01)

+ {

+ x <- c(x, val2)

+ }

+ }

+

+ else

+ {

+ #Do Nothin

+ }

+ }

#Printing the x to get the list

> x

93 141 161 169 170 205 259 287 299 381

0.0014927000 0.0014615645 0.0034621486 0.0043717021 0.0051587345 0.0000000000 0.0053782376 0.0091229762 0.0005312978 0.0091655049

429 519 539 570

0.0016818204 0.0015985878 0.0071210562 0.0089918990

**Part 2: Creating a Macros**

> af <- read.csv("Asian\_Fusion\_Data.csv");

>

> mydata <- sqldf("select record,Catkey,sex,NAR,BRR,VRR,LAR,OSR,BAR,SBA from af")

> myfunction <- function(ads)

+ {

+ x <- vector(mode="numeric", length=0)

+ y<- c(1:length(ads$Catkey))

+ for (i in y)

+ {

+ mdist <- mahalanobis(na.omit(ads[i,c(4:10)]),center=colMeans(na.omit(ads[-(i),c(4:10)])),cov(na.omit(ads[-(i),c(4:10)])) )

+

+ if (length(mdist) > 0)

+ {

+ val2 <- 1-pchisq(mdist,7)

+ if(val2 < 0.01)

+ {

+ x <- c(x, val2)

+ }

+ val2 <- NA

+ }

+

+ else

+ {

+ #Do Nothin

+ }

+ }

+ print(x)

+ }

>

#Running the function with database name

> myfunction(mydata)

93 141 161 169 170 205 259 287 299 381

0.0014927000 0.0014615645 0.0034621486 0.0043717021 0.0051587345 0.0000000000 0.0053782376 0.0091229762 0.0005312978 0.0091655049

429 519 539 570

0.0016818204 0.0015985878 0.0071210562 0.0089918990