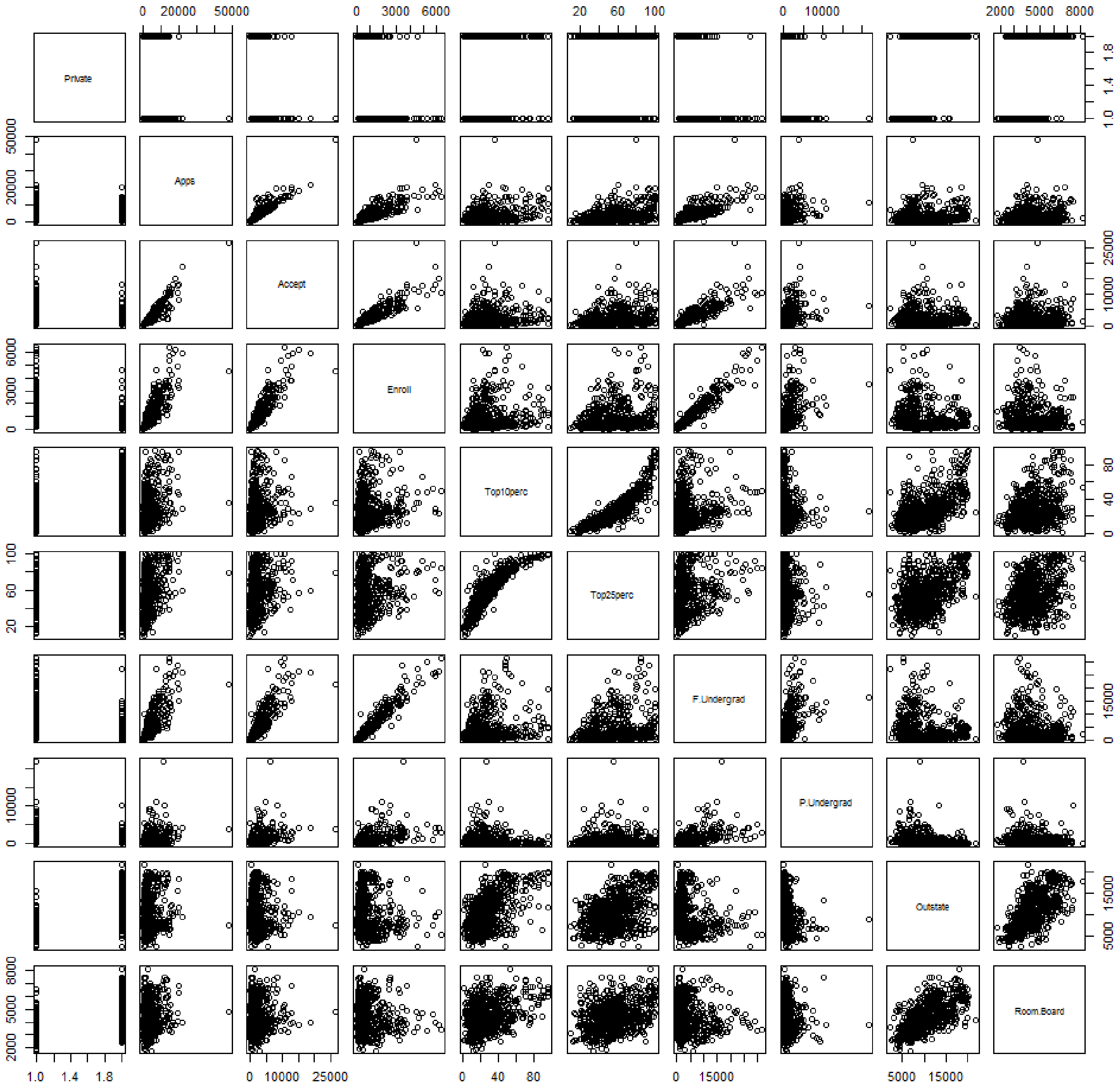
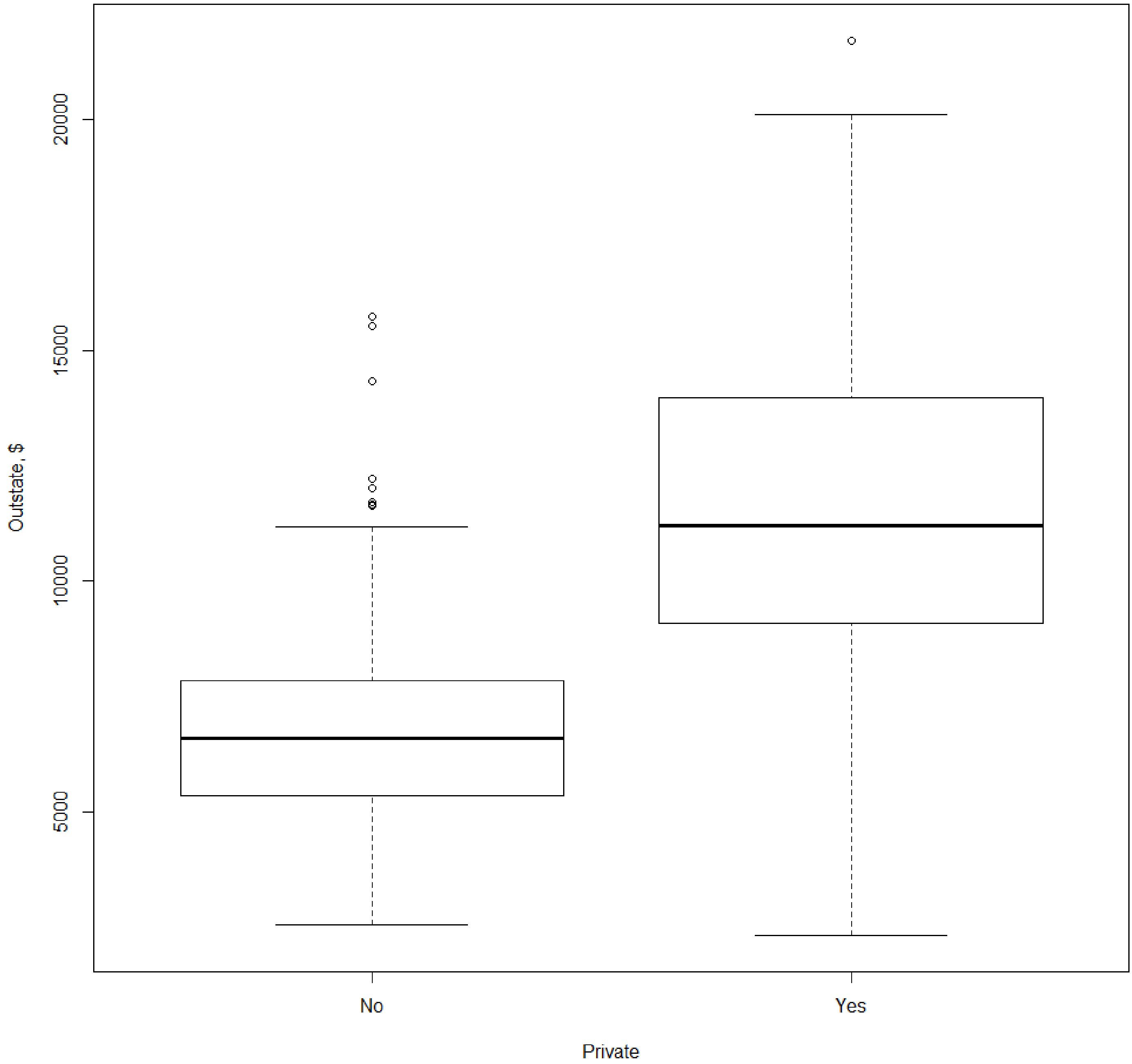
```r
# Ryan Torelli
# CptS 483-04
# Assignment 2
# September 10, 2017


# 1.(a)
# Read College.csv, a table of 777 tuples of 18 attributes and name
college <- read.csv("College.csv")

# 1.(b)
# Set names and drop
rownames(college)=college[,1]
fix(college)
college=college[,-1]
fix(college)

# 1.(c)i.
# Print summary statistics of variables
summary(college)
# Summary results include Grad.Rate Mean : 65.46
# Remaining output not shown.

# ii.
# Plot first 10 variables by scatterplot (see Figure1(c)ii)
pairs(college[,1:10], main="Figure 1(c)ii")
```

Figure 1(c)ii


Private
Apps
Accept
Enroll
Top10perc
Top25perc
F.Undergrad
P.Undergrad
Outstate
Room.Board
0 20000 50000
0 3000 6000
20 60 100
0 10000
2000 5000 8000
1.0 1.4 1.8
0 10000 25000
0 40 80
0 15000
5000 15000

```
# Ryan Torelli
# CptS 483-04
# Assignment 2
# September 10, 2017


# iii.
# Plot Outstate v Private by boxplot (see Figure1(c)iii)
plot(y=college$Outstate, ylab="Outstate, $",
     x=college$Private, xlab="Private",
     main="Figure 1(c)iii. Outstate v Private")
```

Figure 1(c)iii. Outstate v Private



```
# Ryan Torelli
# CptS 483-04
# Assignment 2
# September 10, 2017


# iv.
# Add variable Top that reports isTop25 > 50%
Top=rep("No",nrow(college))
Top[college$Top25perc > 50]="Yes"
Top=as.factor(Top)
college=data.frame(college, Top)
summary(college)
# Top No: 328 Yes: 449

# Plot Outstate v Top by boxplot (see Figure1(c)iv)
plot(y=college$Outstate, ylab="Outstate, $",
     x=college$Top, xlab="isTop25 > 50%",
     main="Figure 1(c)iv. Outstate v Top")
```

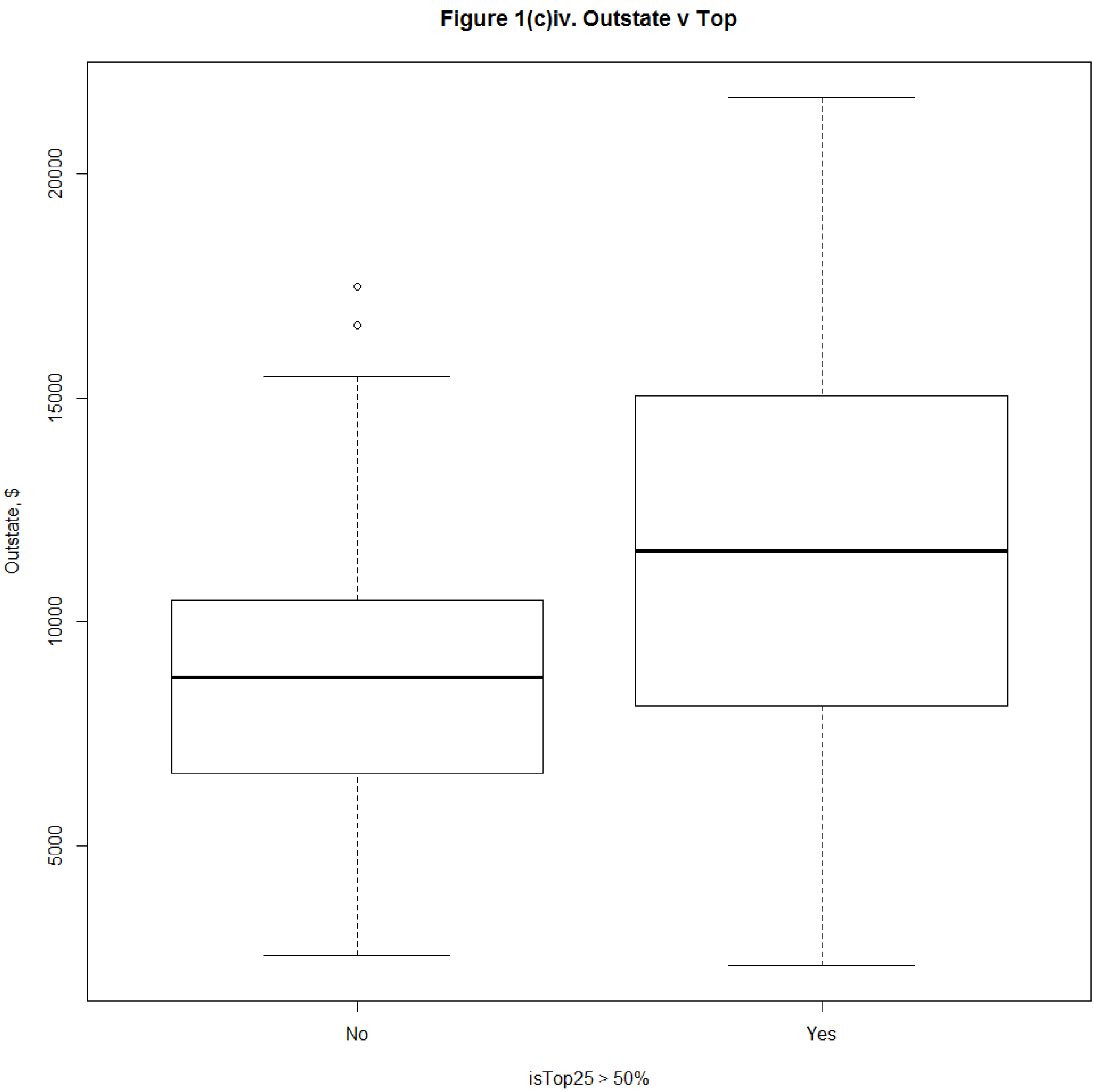
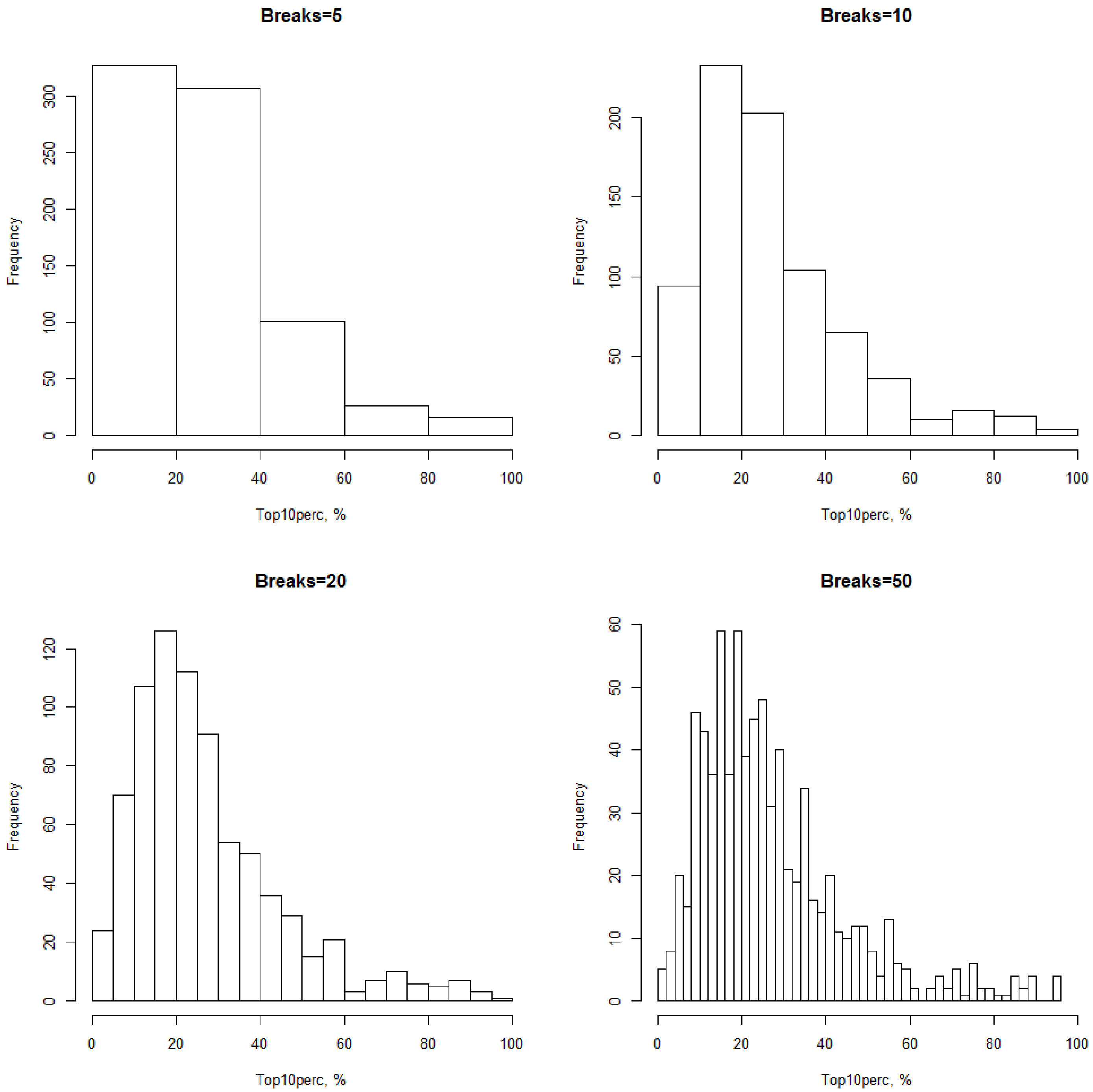
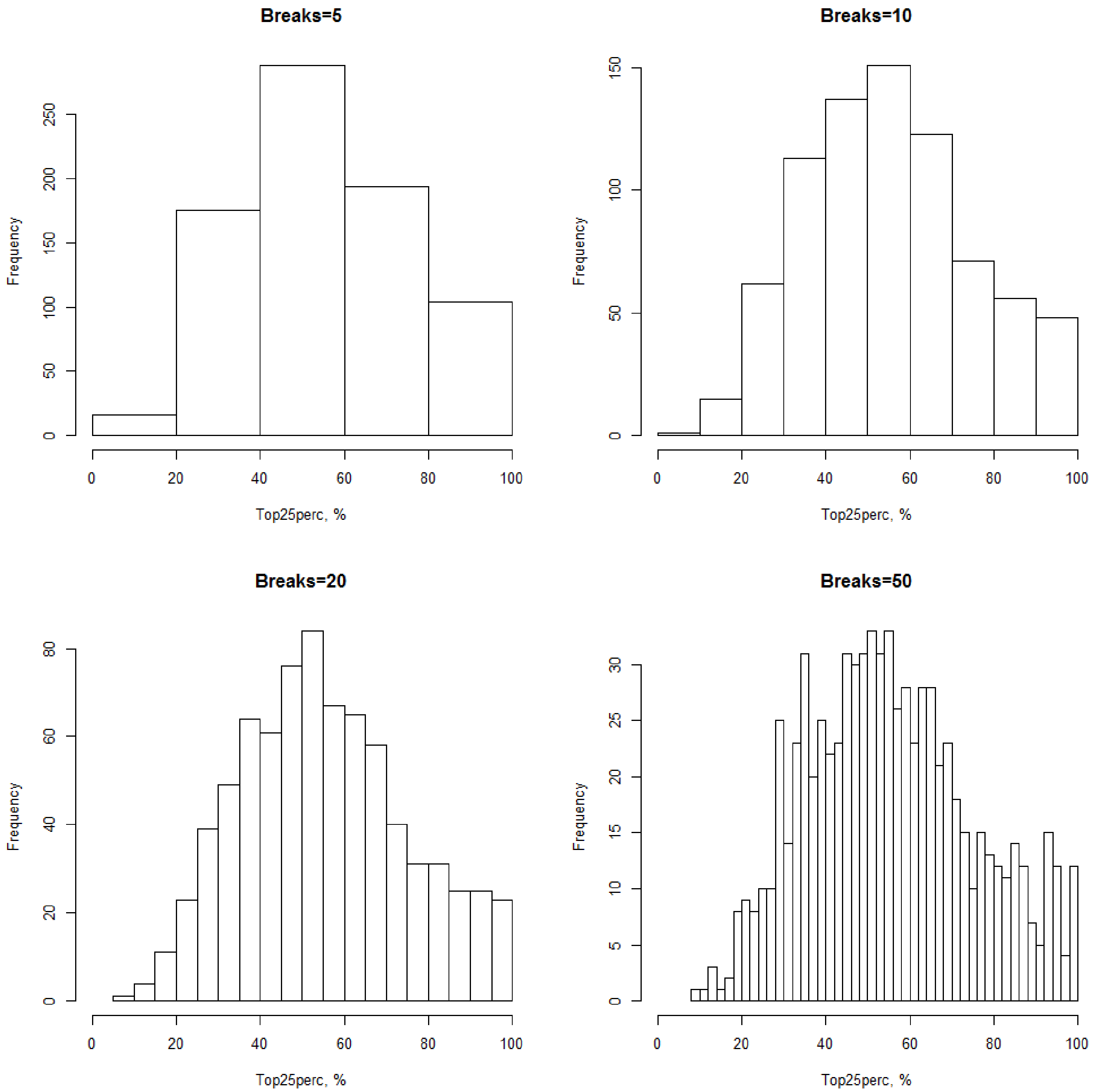


Figure 1(c)iv. Outstate v Top
Outstate, $
5000
10000
15000
20000
No
Yes
isTop25 > 50%

```
# Ryan Torelli
# CptS 483-04
# Assignment 2
# September 10, 2017


# v.
# Plot histograms of Top10perc and Top25perc with breaks 5, 10, 20, 50 (see
Break...)
par(mfrow=c(2,2))
hist(college$Top10perc, xlab="Top10perc, %", xlim=range(0,100),
     breaks=5, ylab="Frequency", main="Breaks=5")
hist(college$Top10perc, xlab="Top10perc, %", xlim=range(0,100),
     breaks=10, ylab="Frequency", main="Breaks=10")
hist(college$Top10perc, xlab="Top10perc, %", xlim=range(0,100),
     breaks=20, ylab="Frequency", main="Breaks=20")
hist(college$Top10perc, xlab="Top10perc, %", xlim=range(0,100),
     breaks=50, ylab="Frequency", main="Breaks=50")
hist(college$Top25perc, xlab="Top25perc, %", xlim=range(0,100),
     breaks=5, ylab="Frequency", main="Breaks=5")
hist(college$Top25perc, xlab="Top25perc, %", xlim=range(0,100),
     breaks=10, ylab="Frequency", main="Breaks=10")
hist(college$Top25perc, xlab="Top25perc, %", xlim=range(0,100),
     breaks=20, ylab="Frequency", main="Breaks=20")
hist(college$Top25perc, xlab="Top25perc, %", xlim=range(0,100),
     breaks=50, ylab="Frequency", main="Breaks=50")
```

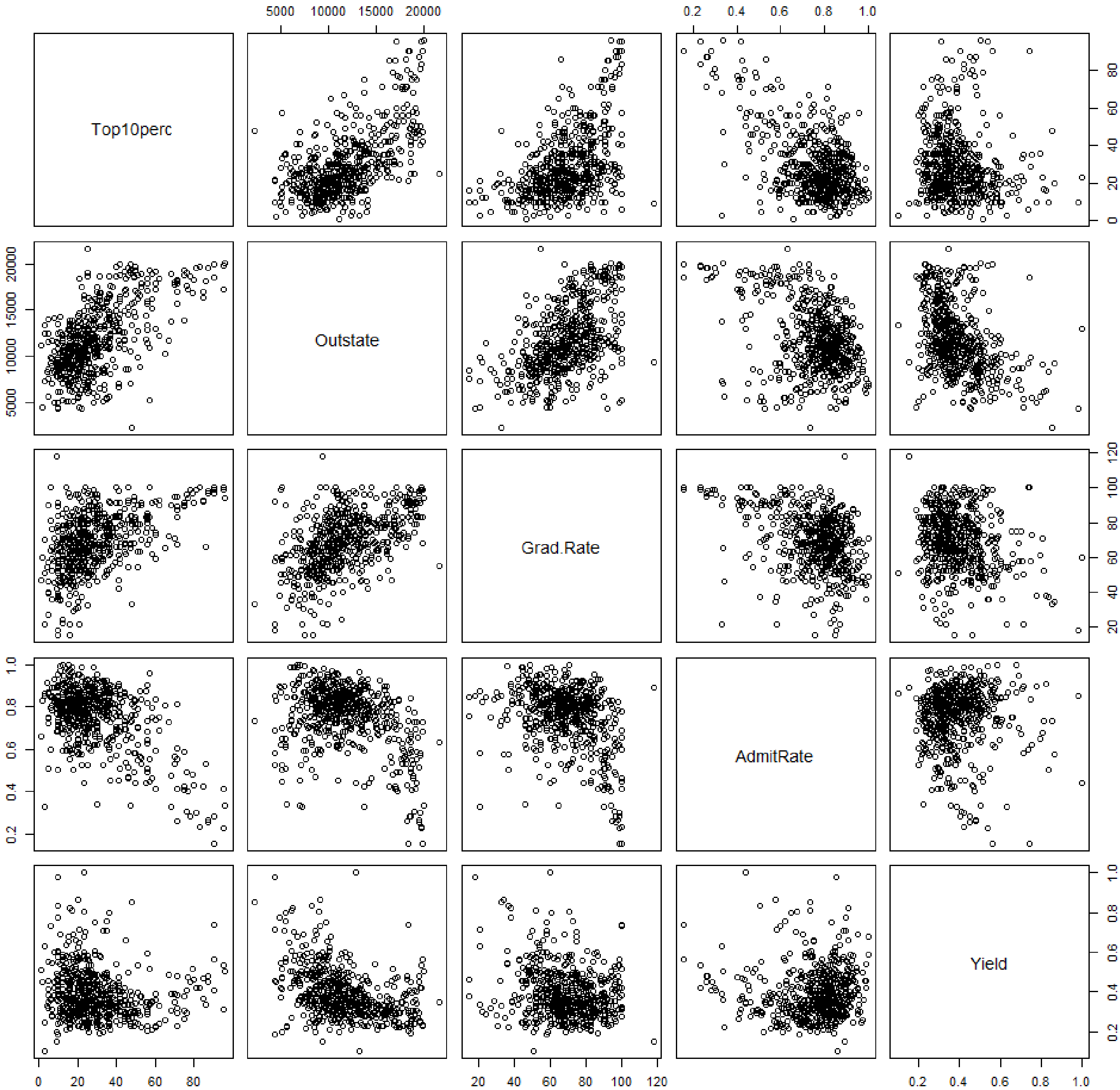

Breaks=5
Breaks=10
Breaks=20
Breaks=50
Frequency
Top10perc, %



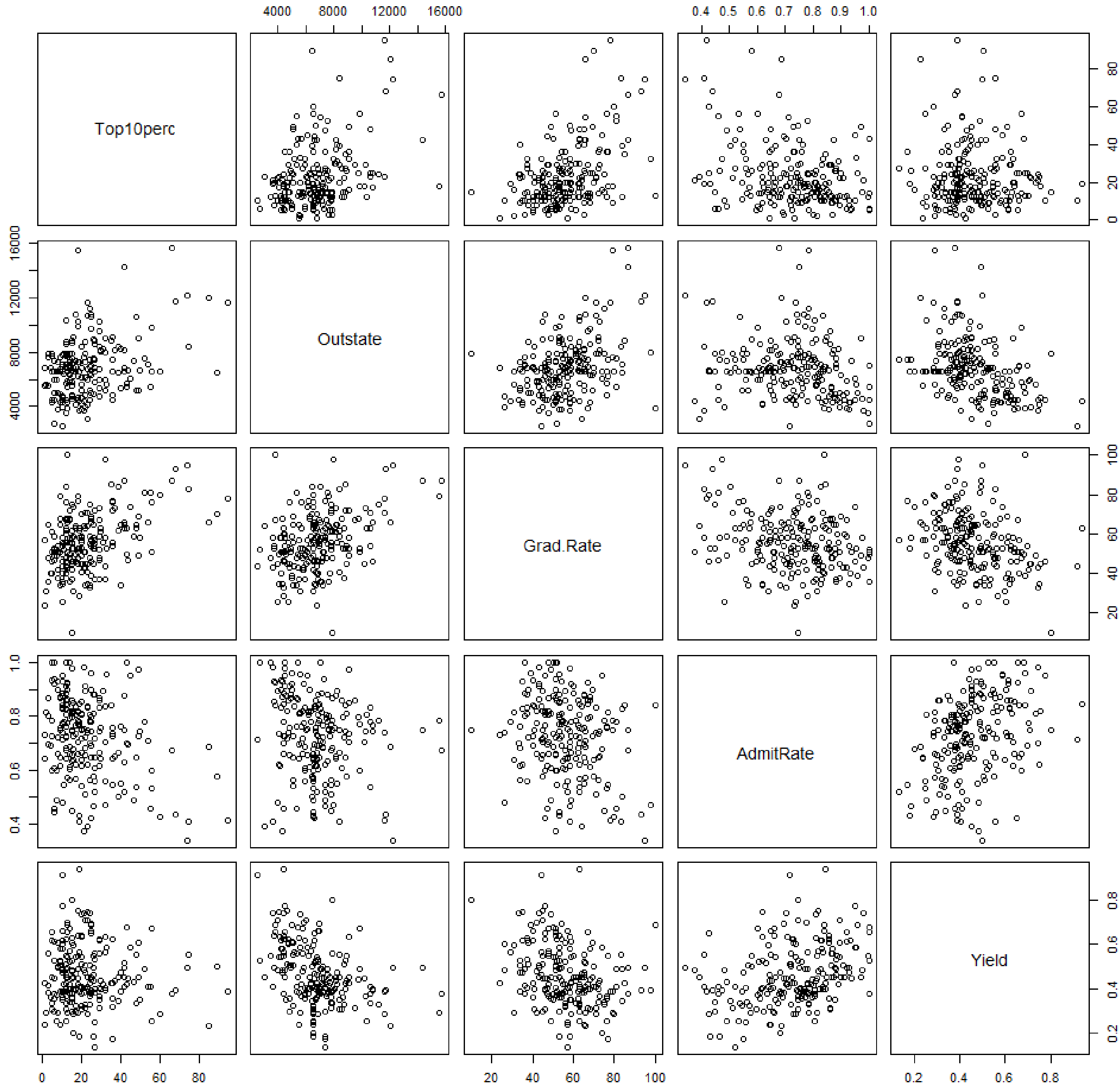
Breaks=5
Breaks=10
Breaks=20
Breaks=50
Frequency
Top25perc, %

```
# Ryan Torelli
# CptS 483-04
# Assignment 2
# September 10, 2017


# vi.
# Add ratios for admission rate and admission yield
AdmitRate <- college$Accept / college$Apps
Yield <- college$Enroll / college$Accept
college=data.frame(college,AdmitRate,Yield)

# Subset Public or Private
collegePublic=college[college$Private=="No",]
collegePrivate=college[college$Private=="Yes",]

# Plot Public+Private, Public only, Private only (see Variable Pairs)
pairs(college[,c(5,9,18,20,21)], main="Variable Pairs: Public and Private")
pairs(collegePublic[,c(5,9,18,20,21)], main="Variable Pairs: Public")
pairs(collegePrivate[,c(5,9,18,20,21)], main="Variable Pairs: Private")
# Less interesting than expected but...
# High Top10perc enrollment guarantees superior Grad.Rates
# The association between Yield and AdmitRate is stronger for private than
public
```

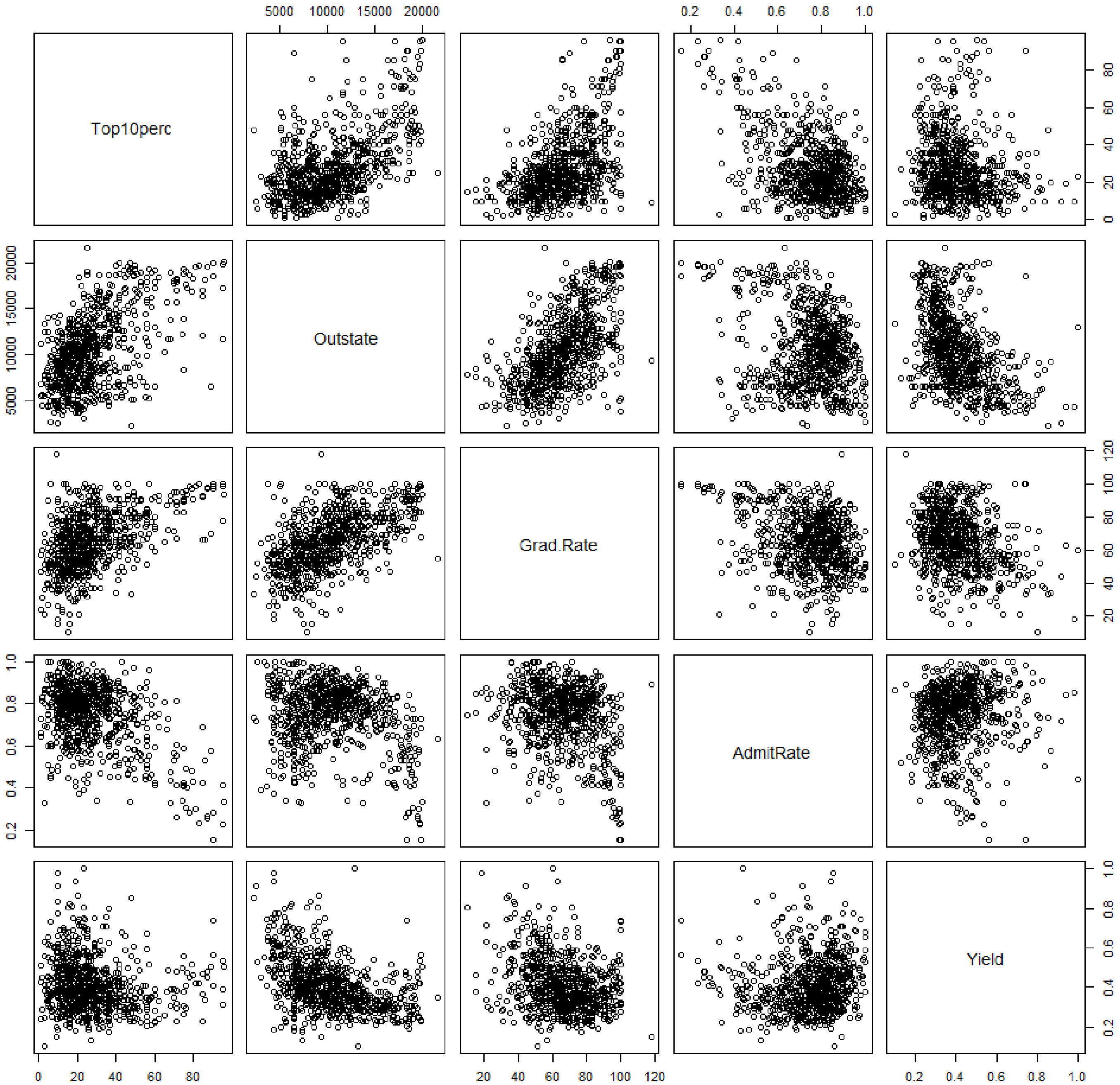
Variable Pairs: Private



Variable Pairs: Public



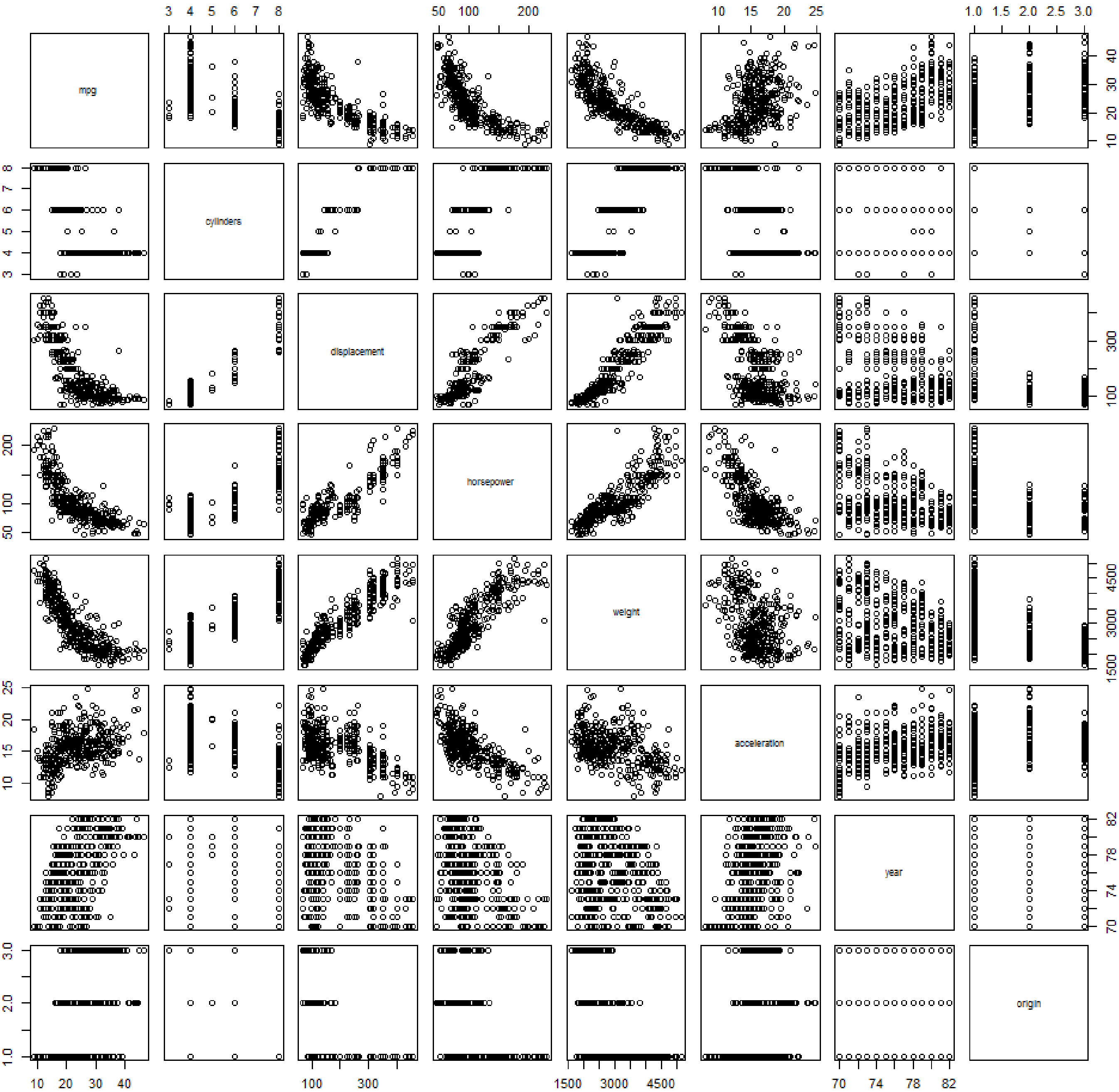
Variable Pairs: Public and Private



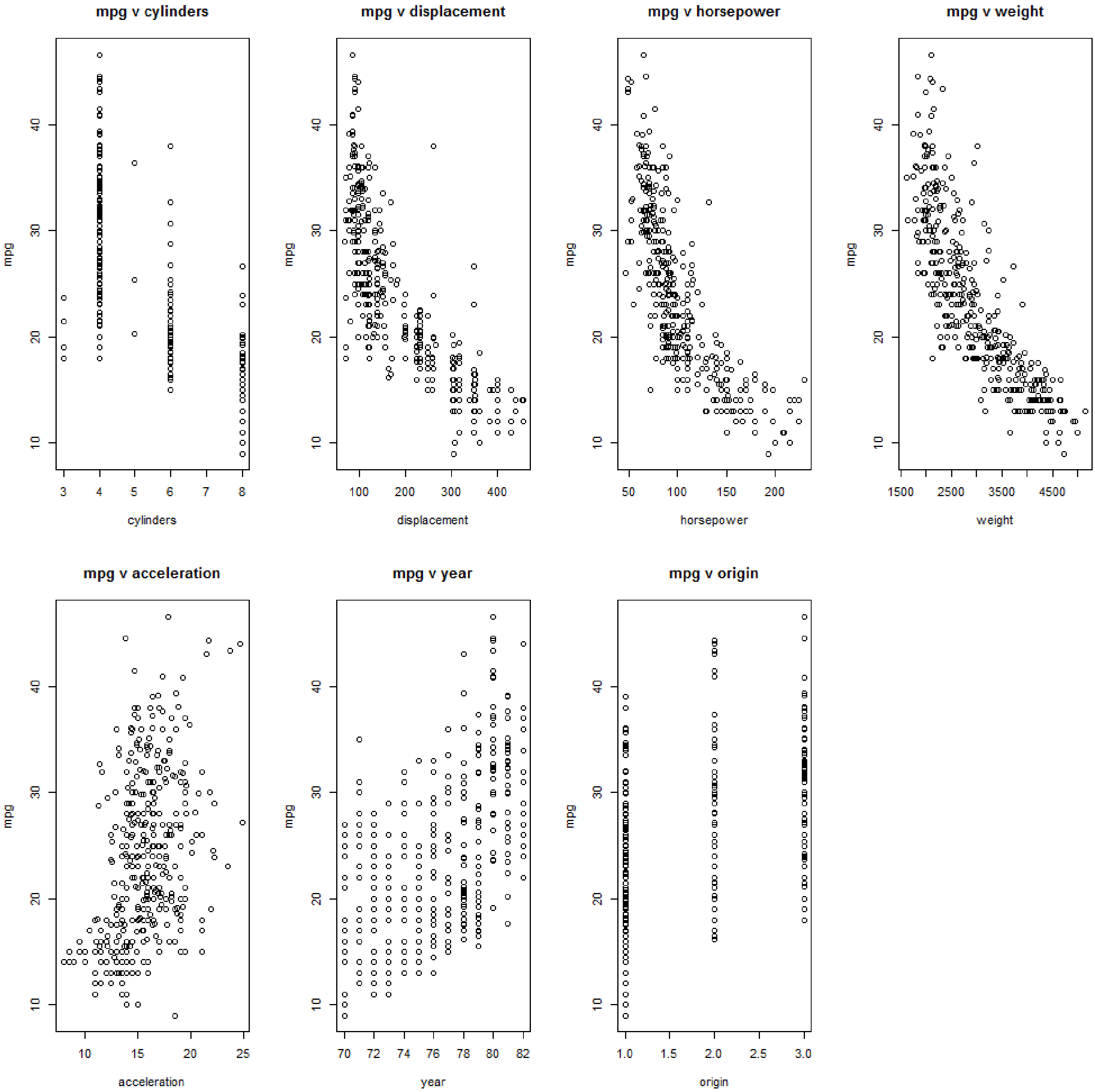
```
# Ryan Torelli
# CptS 483-04
# Assignment 2
# September 10, 2017


# 2 Auto.csv
# (a)
# Read Auto.csv, a table of 397 tuples of 8 attributes and name
auto <- read.csv("Auto.csv", na.strings="?")
# The quantitative predictors are mpg, displacement, horsepower, weight, and
acceleration.
# The qualitative predictors are cyclinders, year, and origin.
# The number of cyclinders has range [4,8]; so it's classed as qualitative.

# (b)
# Subset quantitative predictors and omit N/A
autoQuant <-
subset(auto,select=c(mpg,displacement,horsepower,weight,acceleration))
na.omit(autoQuant)

# Apply range to subset
sapply(autoQuant,range)
#       mpg displacement horsepower weight acceleration
# [1,]  9.0           68          NA          1613          8.0
# [2,] 46.6          455         NA          5140         24.8

# (c)
# Apply mean and stdev to subset
sapply(autoQuant,mean)
#         mpg displacement   horsepower       weight acceleration
#    23.51587    193.53275           NA   2970.26196     15.55567

sapply(autoQuant,sd)
#         mpg displacement   horsepower       weight acceleration
#    7.825804   104.379583           NA   847.904119     2.749995

summary(autoQuant)
# horsepower
# Min.   : 46.0
# 1st Qu.: 75.0
# Median : 93.5
# Mean   :104.5
# 3rd Qu.:126.0
# Max.   :230.0
# NA's   :5

# (d)
# Drop 25th - 75th observations
autoQuantDrop <- rbind(autoQuant[1:24,],autoQuant[76:397,])

# Apply range, mean, and stdev
sapply(autoQuantDrop,range)
#       mpg displacement horsepower weight acceleration
# [1,] 11.0           68         NA   1649          8.0
# [2,] 46.6          455         NA   4997         24.8

sapply(autoQuantDrop,mean)
#         mpg displacement   horsepower       weight acceleration
#    24.23353    186.89884           NA   2919.23410     15.65058

sapply(autoQuantDrop,sd)
#        mpg displacement   horsepower       weight acceleration
#    7.758210   100.924616           NA   799.058624     2.740002
```

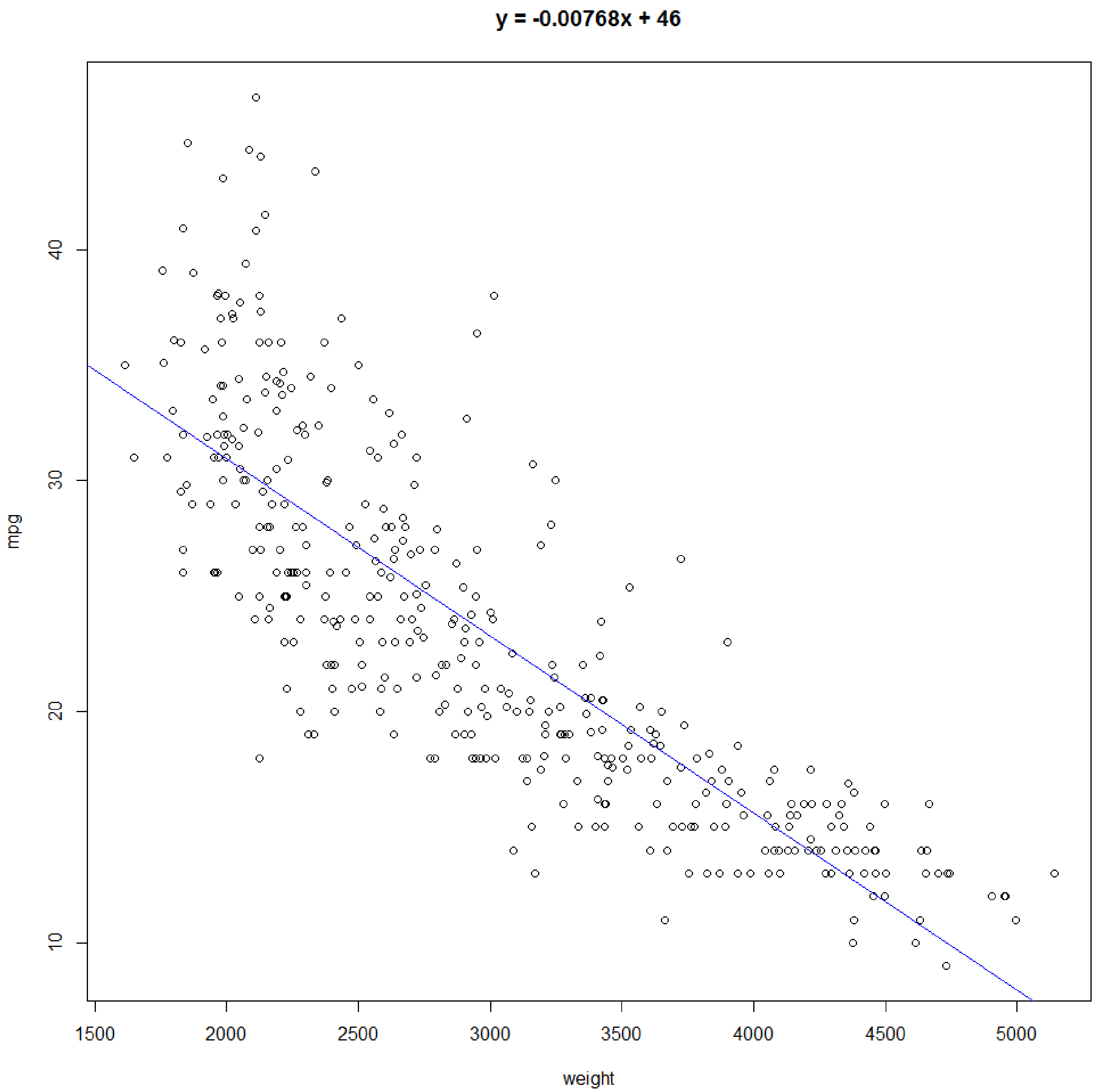
```
summary(autoQuantDrop)
# horsepower
# Min.   : 46.0
# 1st Qu.: 75.0
# Median : 91.5
# Mean   :101.5
# 3rd Qu.:115.0
# Max.   :230.0
# NA's   :4

# (e)
# Subset predictors and omit N/A
autoPredictors <- subset(auto,select=-name)
na.omit(autoPredictors)

# Plot predictors by scatterplot (see predictors)
pairs(autoPredictors[,1:8], main="Predictor Pairs")
# mpg has an observable association with hp and weight.
# cylinders is a proxy for hp.
# displacement is a proxy for {cyclinders, hp}.
# horsepower associates with weight.
# weight associates with horsepower.
# acceleration is unclear.
# year trends with mpg.
# cars with certain origin weigh less and have better mpg.
```

Predictor Pairs



```r
# Ryan Torelli
# CptS 483-04
# Assignment 2
# September 10, 2017


# (f)
# Plot mpg against each predictor (see mpg)
par(mfrow=c(2,4))
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$cylinders, xlab="cylinders",
     main="mpg v cylinders")
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$displacement, xlab="displacement",
     main="mpg v displacement")
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$horsepower, xlab="horsepower",
     main="mpg v horsepower")
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$weight, xlab="weight",
     main="mpg v weight")
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$acceleration, xlab="acceleration",
     main="mpg v acceleration")
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$year, xlab="year",
     main="mpg v year")
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$origin, xlab="origin",
     main="mpg v origin")
# The predictors of displacement, horsepower, and weight plot similar patterns.
# The predictors of year and origin associate positively with mpg.
# The predictor cylinders associates negatively with mpg.
# The predictor acceleration has an unclear association with mpg.

# Plot mpg v weight with regression (see Regression)
regression <- lm(autoPredictors$mpg~autoPredictors$weight)
coefficient=coefficients(regression)
equation=paste0("y = ", round(coefficient[2],5), "x + ",
round(coefficient[1],0))
plot(y=autoPredictors$mpg, ylab="mpg",
     x=autoPredictors$weight, xlab="weight",
     main=equation)
abline(regression, col="blue")
```


mpg v cylinders
mpg v displacement
mpg v horsepower
mpg v weight
mpg v acceleration
mpg v year
mpg v origin
mpg
10
20
30
40
3
4
5
6
7
8
cylinders
100
200
300
400
displacement
50
100
150
200
horsepower
1500
2500
3500
4500
weight
10
15
20
25
acceleration
70
72
74
76
78
80
82
year
1.0
1.5
2.0
2.5
3.0
origin



y = -0.00768x + 46
mpg
10
20
30
40
weight
1500
2000
2500
3000
3500
4000
4500
5000