Jin Kweon (3032235207)

Jin Kweon

9/5/2017

"The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes." (Quinlan, 1993)

Number of Instances: 398

Number of Attributes: 9 including the class attribute

Missing Attribute Values: horsepower has 6 missing values

(Source for the data)Source

Import

```
setwd("/Users/yjkweon24/Desktop/Cal/2017 Fall/Stat 151a/HW/HW1")
data <- read.table("auto-mpg.data.txt", stringsAsFactors = F)
colnames(data) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration",
                    "model_year", "origin", "car_name")
dim(data)
```

```
## [1] 398   9
```

```
#There are 6 missing values (marked as ? in the data set), so I would need to modify/change/clean it.
#I will replace to -999.
#Another problem of horsepower column is, its class is factor, which is hard to modify.
#So, I will change it to numeric.
for (i in 1:nrow(data)){
  if (data[i,4] == "?"){
  data[i,4] <- NA
  }
}

data <- data %>% mutate(horsepower = as.numeric(horsepower))
data <- na.omit(data)
```

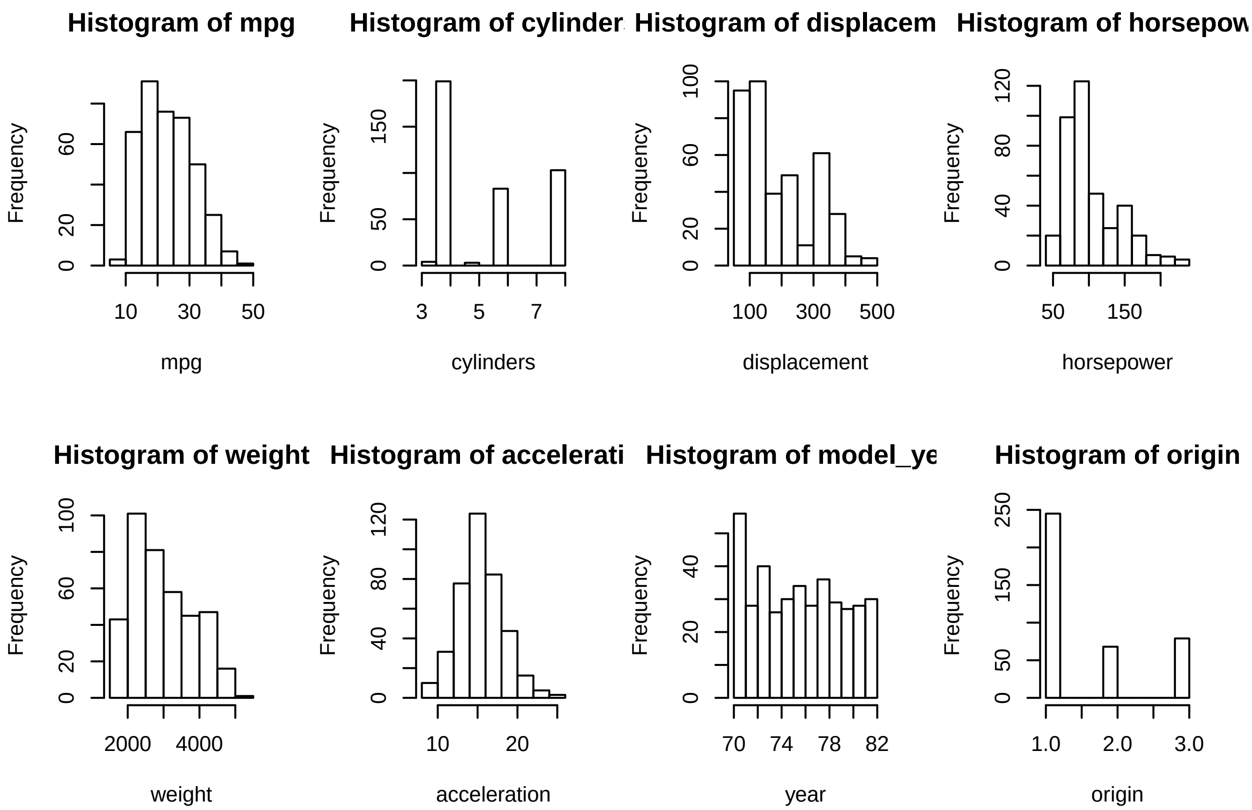
Part A

Generate questions about my data Search for answers by visualising, transforming, and modelling your data Use what you learn to refine your questions and/or generate new questions Calculate main characteristics Understand the data and find possible new hypothesis

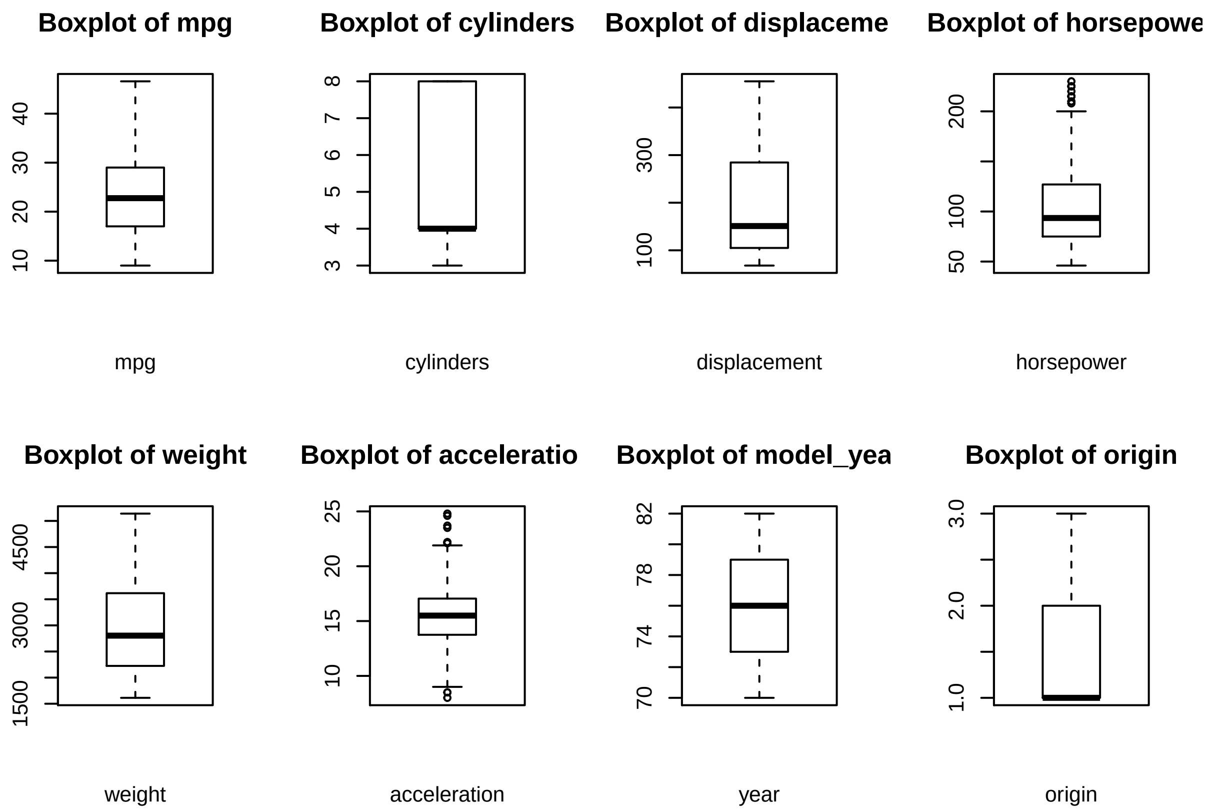
```r
summary(data) #All, but car_name are quantitative variables.
```

```
##       mpg          cylinders      displacement     horsepower   
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0  
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0  
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5  
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5  
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0  
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0  
##      weight      acceleration     model_year        origin     
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000  
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000  
##  Median :2804   Median :15.50   Median :76.00   Median :1.000  
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577  
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000  
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000  
##    car_name        
##  Length:392        
##  Class :character  
##  Mode  :character  
##                    
##                    
## 
```

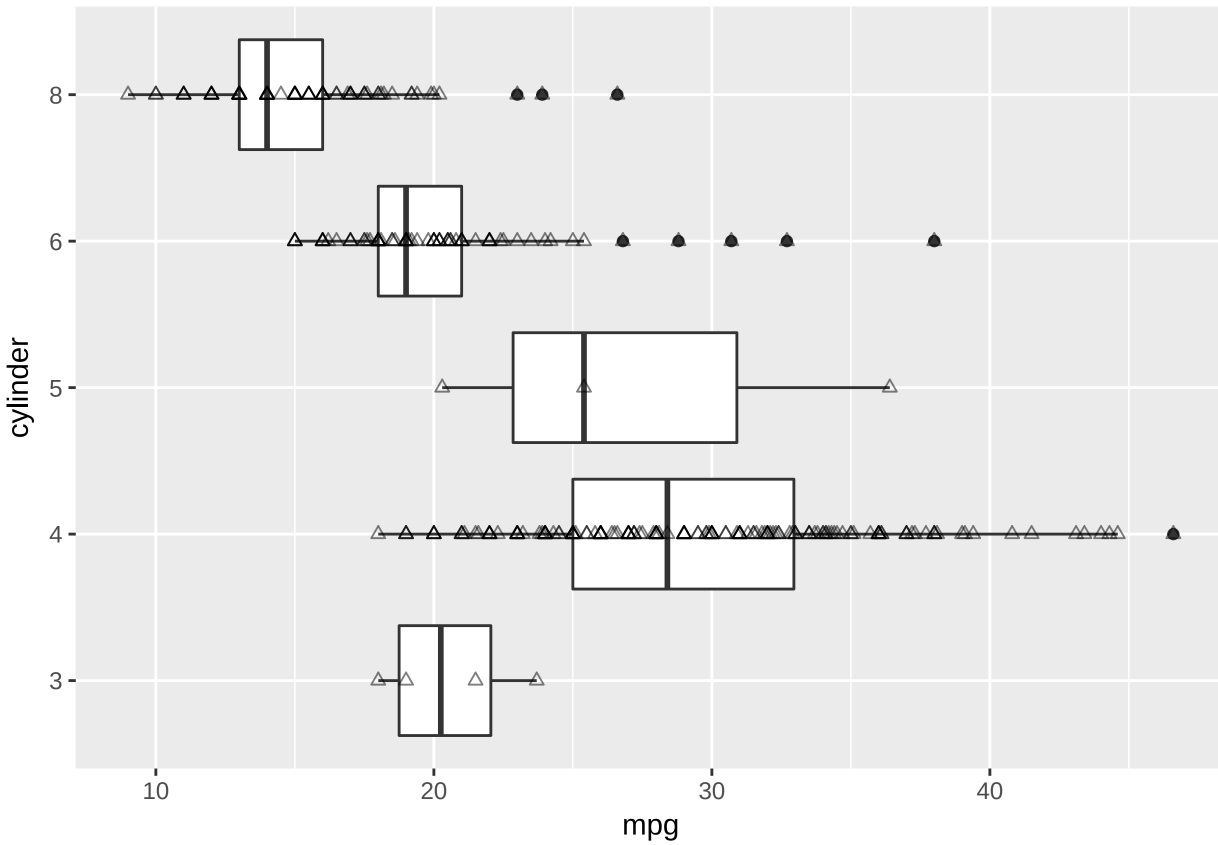
```r
#histogram
par(mfrow=c(2,4))
hist(data$mpg, main = "Histogram of mpg", xlab = "mpg")
#ggplot(data) + geom_bar(aes(x = displacement))
hist(data$cylinders, main = "Histogram of cylinders", xlab = "cylinders")
hist(data$displacement, main = "Histogram of displacement", xlab = "displacement")
hist(na.omit(data$horsepower), main = "Histogram of horsepower", xlab = "horsepower")
hist(data$weight, main = "Histogram of weight", xlab = "weight")
hist(data$acceleration, main = "Histogram of acceleration", xlab = "acceleration")
hist(data$model_year, main = "Histogram of model_year", xlab = "year")
hist(data$origin, main = "Histogram of origin", xlab = "origin")
```



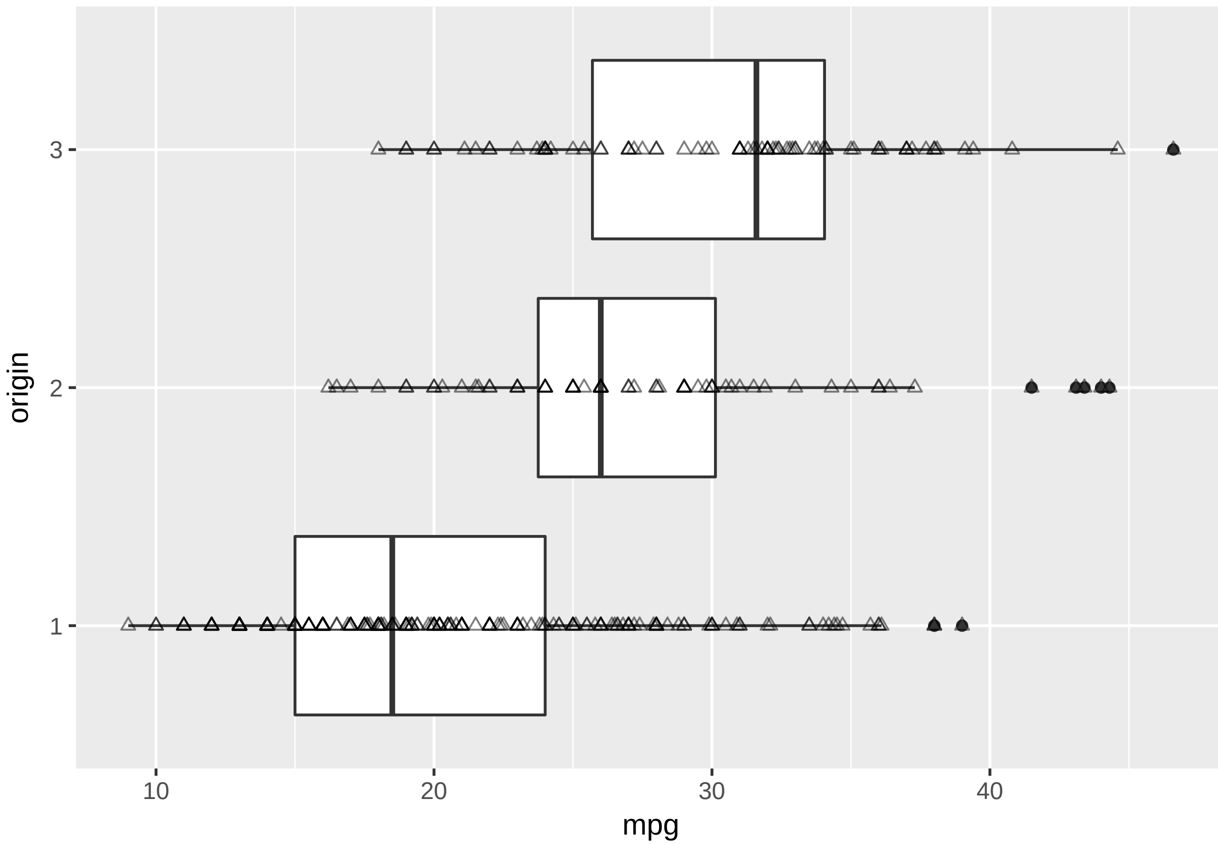

```r
#Boxplots
boxplot(data$mpg, main = "Boxplot of mpg", xlab = "mpg")
boxplot(data$cylinders, main = "Boxplot of cylinders", xlab = "cylinders")
boxplot(data$displacement, main = "Boxplot of displacement", xlab = "displacement")
boxplot(na.omit(data$horsepower), main = "Boxplot of horsepower", xlab = "horsepower")
boxplot(data$weight, main = "Boxplot of weight", xlab = "weight")
boxplot(data$acceleration, main = "Boxplot of acceleration", xlab = "acceleration")
boxplot(data$model_year, main = "Boxplot of model_year", xlab = "year")
boxplot(data$origin, main = "Boxplot of origin", xlab = "origin")
```




```
par(mfrow=c(2,1))
ggplot(data, mapping = aes(x = factor(cylinders), y = mpg)) + geom_boxplot() +
  coord_flip() + labs(x = "cylinder") + geom_point(shape=2, alpha = 0.5)
```




```r
ggplot(data, mapping = aes(x = factor(origin), y = mpg)) + geom_boxplot() +
  coord_flip() + labs(x = "origin") + geom_point(shape=2, alpha = 0.5)
```




```
#Kernel-desnsity plots (bumpy)
par(mfrow=c(2,3))
den_acc <- density(data$acceleration, adjust = 0.4)
plot(den_acc, main = "acceleration")
polygon(den_acc, col = "red", border = "blue")
density(data$acc)
```

```
##
## Call:
##  density.default(x = data$acc)
##
## Data: data$acc (392 obs.);   Bandwidth 'bw' = 0.6612
##
##        x                 y
##  Min.   : 6.016   Min.   :2.016e-05
##  1st Qu.:11.208   1st Qu.:4.431e-03
##  Median :16.400   Median :2.176e-02
##  Mean   :16.400   Mean   :4.810e-02
##  3rd Qu.:21.592   3rd Qu.:8.327e-02
##  Max.   :26.784   Max.   :1.564e-01
```

```
den_year <- density(data$model_year, adjust = 0.4)
plot(den_year, main = "model_year")
polygon(den_year, col = "red", border = "blue")
density(data$model_year)
```

```
##
## Call:
```

```
##  density.default(x = data$model_year)
##
## Data: data$model_year (392 obs.);    Bandwidth 'bw' = 1.004
##
##        x                 y
##  Min.   :66.99   Min.   :0.00034
##  1st Qu.:71.49   1st Qu.:0.02880
##  Median :76.00   Median :0.07124
##  Mean   :76.00   Mean   :0.05541
##  3rd Qu.:80.51   3rd Qu.:0.07941
##  Max.   :85.01   Max.   :0.08263
```

```r
den_pow <- density(data$horsepower, adjust = 0.4)
plot(den_pow, main = "horsepower")
polygon(den_pow, col = "red", border = "blue")
density(data$horsepower)
```

```
##
## Call:
##  density.default(x = data$horsepower)
##
## Data: data$horsepower (392 obs.);    Bandwidth 'bw' = 10.38
##
##        x                  y
##  Min.   : 14.87   Min.   :1.863e-06
##  1st Qu.: 76.44   1st Qu.:7.996e-04
##  Median :138.00   Median :2.452e-03
##  Mean   :138.00   Mean   :4.057e-03
##  3rd Qu.:199.56   3rd Qu.:5.320e-03
##  Max.   :261.13   Max.   :1.397e-02
```

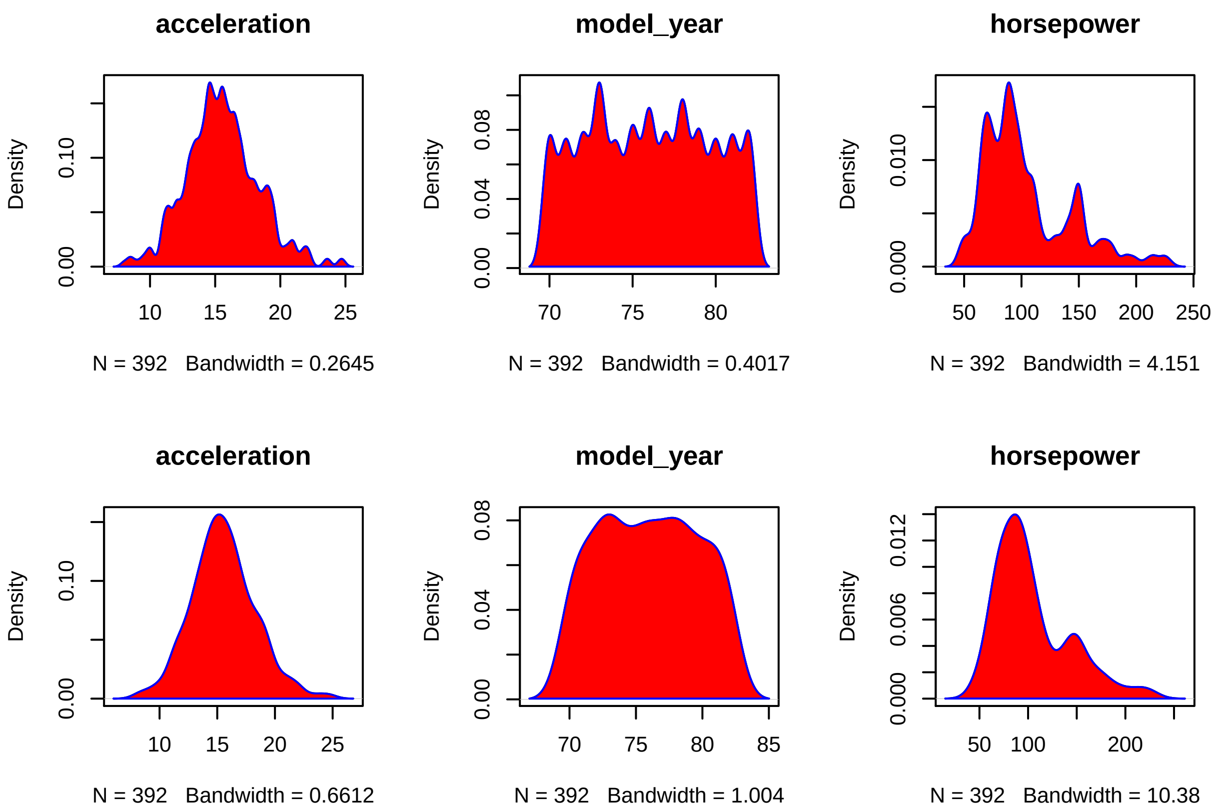
```r
#Kernel-desnsity plots (smooth)
den_acc <- density(data$acceleration, adjust = 1)
plot(den_acc, main = "acceleration")
polygon(den_acc, col = "red", border = "blue")
density(data$acc)
```

```
##
## Call:
##  density.default(x = data$acc)
##
## Data: data$acc (392 obs.);   Bandwidth 'bw' = 0.6612
##
##        x                  y
##  Min.   : 6.016   Min.   :2.016e-05
##  1st Qu.:11.208   1st Qu.:4.431e-03
##  Median :16.400   Median :2.176e-02
##  Mean   :16.400   Mean   :4.810e-02
##  3rd Qu.:21.592   3rd Qu.:8.327e-02
##  Max.   :26.784   Max.   :1.564e-01
```

```r
den_year <- density(data$"model_year", adjust = 1)
plot(den_year, main = "model_year")
polygon(den_year, col = "red", border = "blue")
density(data$"model_year")
```

```
##
## Call:
##  density.default(x = data$model_year)
##
## Data: data$model_year (392 obs.);    Bandwidth 'bw' = 1.004
##
##        x               y
##  Min.   :66.99   Min.   :0.00034
##  1st Qu.:71.49   1st Qu.:0.02880
##  Median :76.00   Median :0.07124
##  Mean   :76.00   Mean   :0.05541
##  3rd Qu.:80.51   3rd Qu.:0.07941
##  Max.   :85.01   Max.   :0.08263
```

```r
den_pow <- density(data$horsepower, adjust = 1)
plot(den_pow, main = "horsepower")
polygon(den_pow, col = "red", border = "blue")
```

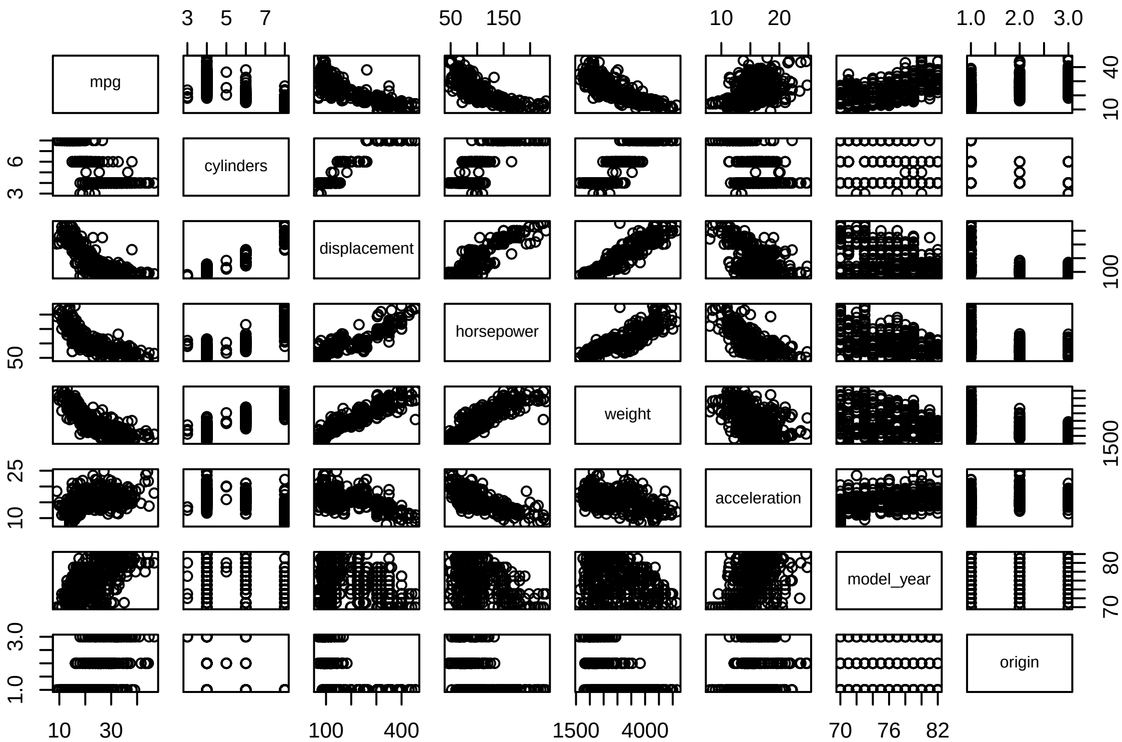



```r
density(data$horsepower)
```

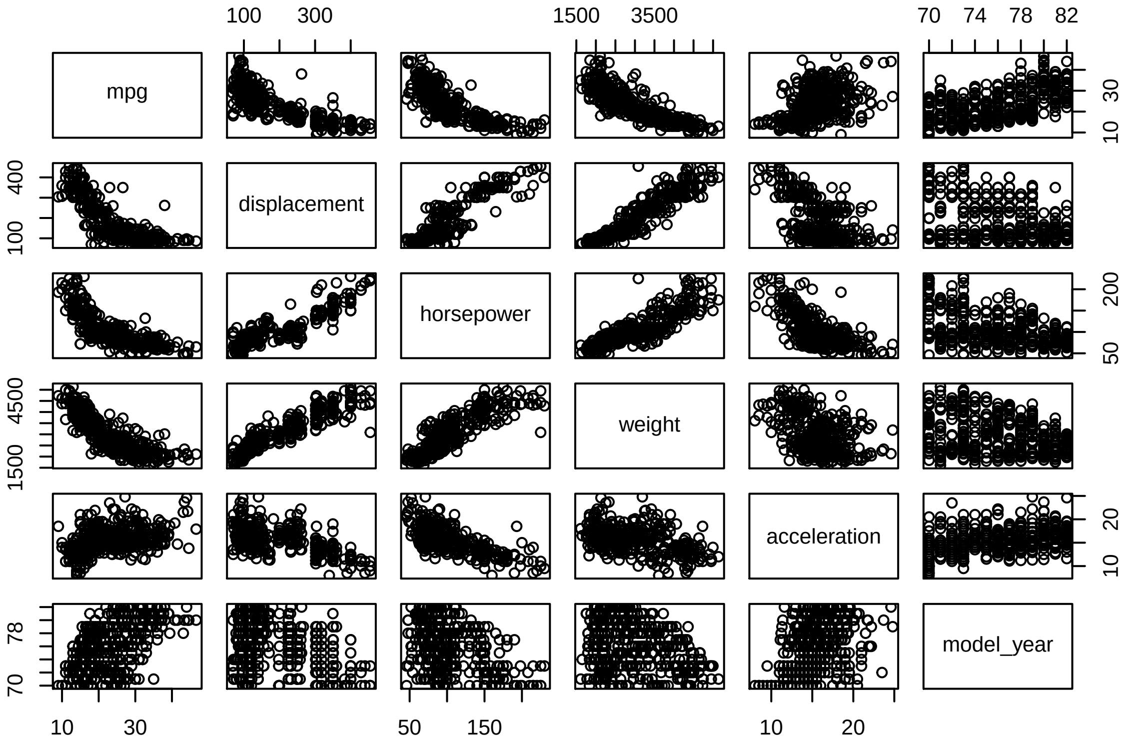
```
##
## Call:
##  density.default(x = data$horsepower)
##
## Data: data$horsepower (392 obs.);    Bandwidth 'bw' = 10.38
##
##        x                 y
##  Min.   : 14.87   Min.   :1.863e-06
##  1st Qu.: 76.44   1st Qu.:7.996e-04
```

```
##   Median :138.00   Median :2.452e-03
##   Mean   :138.00   Mean   :4.057e-03
##   3rd Qu.:199.56   3rd Qu.:5.320e-03
##   Max.   :261.13   Max.   :1.397e-02
```

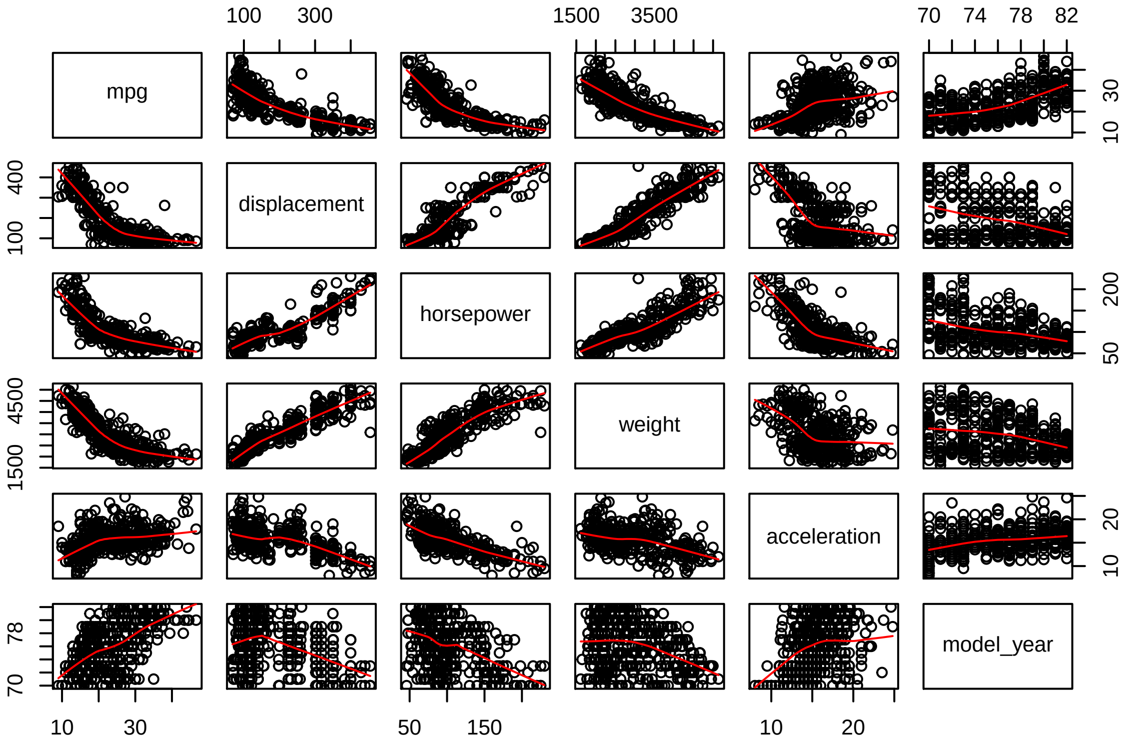
```r
#scatter plots matrices
pairs(data[,-ncol(data)])
```




```r
#scatter plots matrices (modified)
pairs(data[-c(2, 8, 9)])
```




```r
pairs(data[-c(2, 8, 9)], panel = panel.smooth)
```

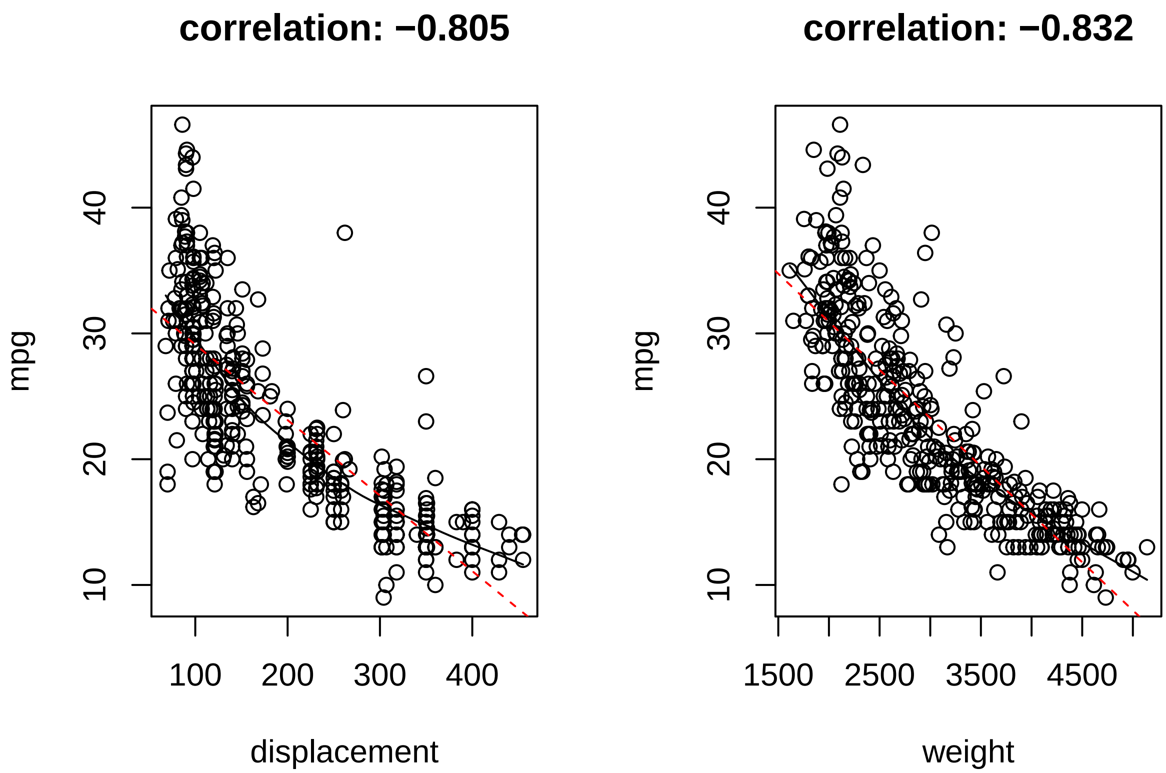



```r
#scatter plot - invidiual (with loess to a scatter plot)
par(mfrow=c(1,2))

beta1 <- coef(lm(data$mpg ~ data$displacement))
scatter.smooth(data$displacement, data$mpg, xlab = "displacement", ylab = "mpg")
```

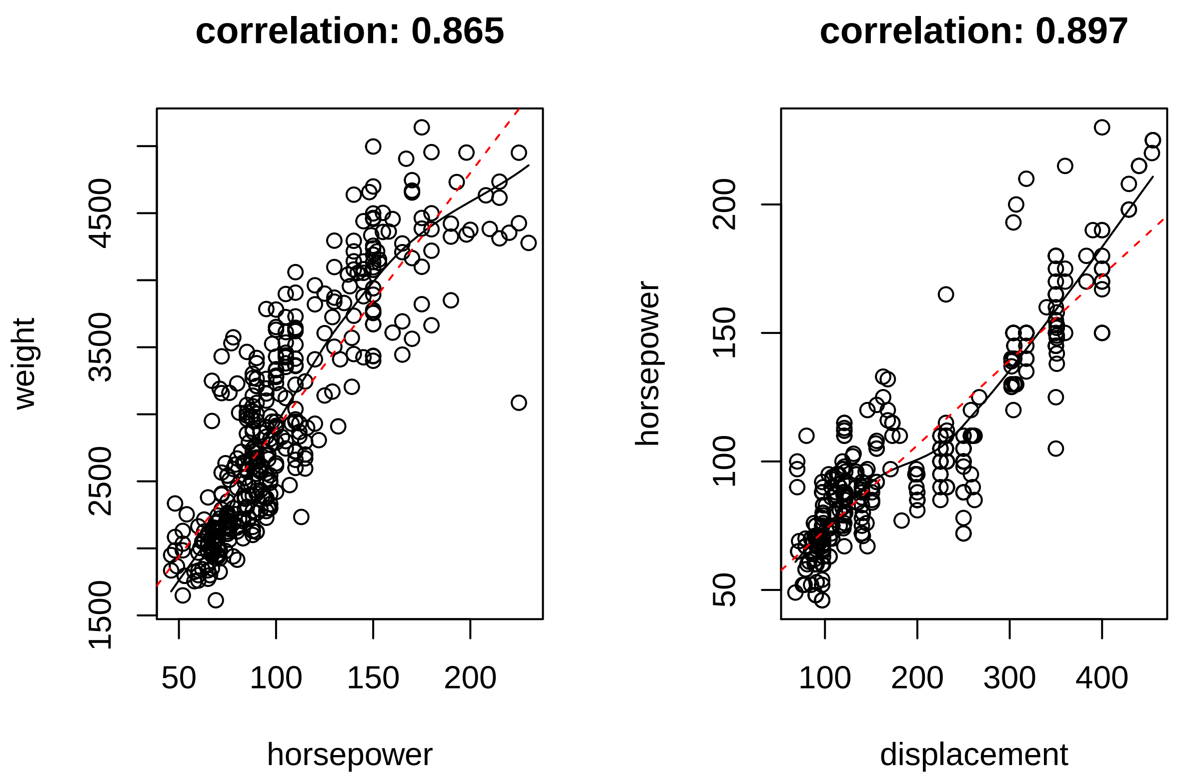
```r
abline(beta1[1], beta1[2], col = "red", lty = 2)
title(paste("correlation:", round(cor(data$displacement, data$mpg), 3)))

beta2 <- coef(lm(data$mpg ~ data$weight))
scatter.smooth(data$weight, data$mpg, xlab = "weight", ylab = "mpg")
abline(beta2[1], beta2[2], col = "red", lty = 2)
title(paste("correlation:", round(cor(data$weight, data$mpg), 3)))
```

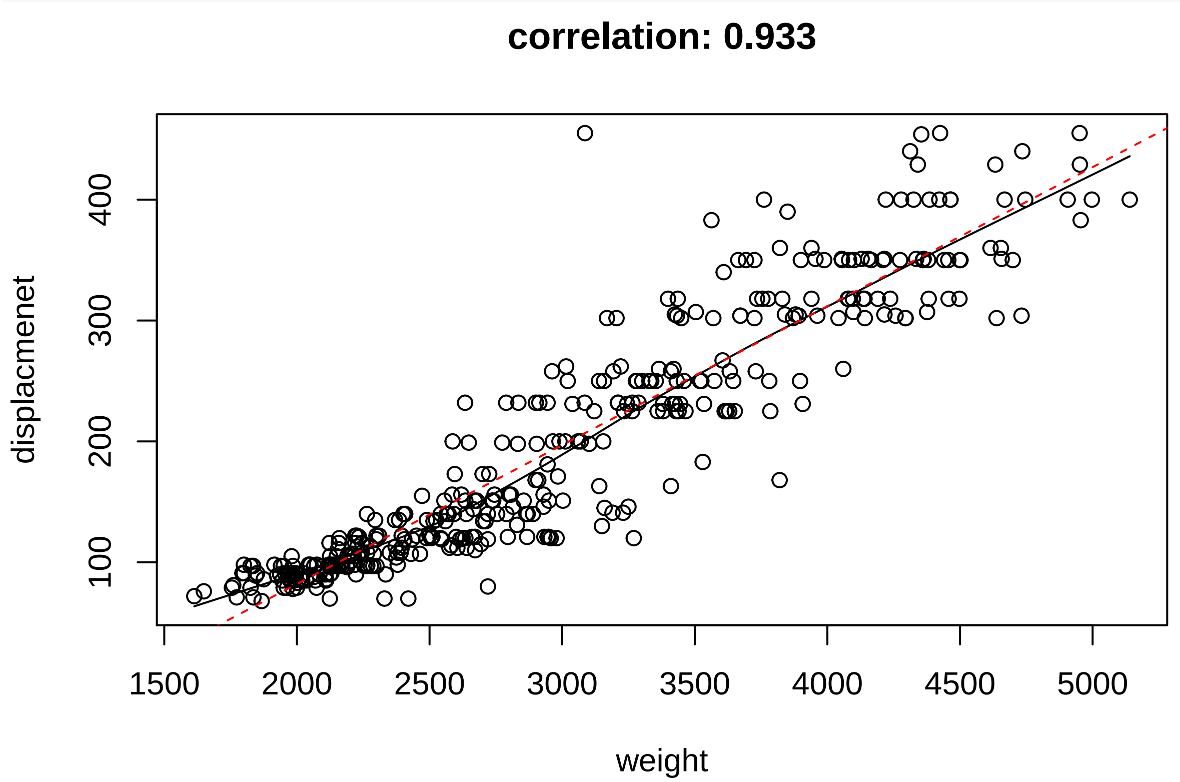



```r
beta3 <- coef(lm(data$weight ~ data$horsepower))
scatter.smooth(data$horsepower, data$weight, xlab = "horsepower", ylab = "weight")
abline(beta3[1], beta3[2], col = "red", lty = 2)
title(paste("correlation:", round(cor(data$horsepower, data$weight), 3)))

beta4 <- coef(lm(data$horsepower ~ data$displacement))
scatter.smooth(data$displacement, data$horsepower, xlab = "displacement", ylab = "horsepower")
abline(beta4[1], beta4[2], col = "red", lty = 2)
title(paste("correlation:", round(cor(data$displacement, data$horsepower), 3)))
```



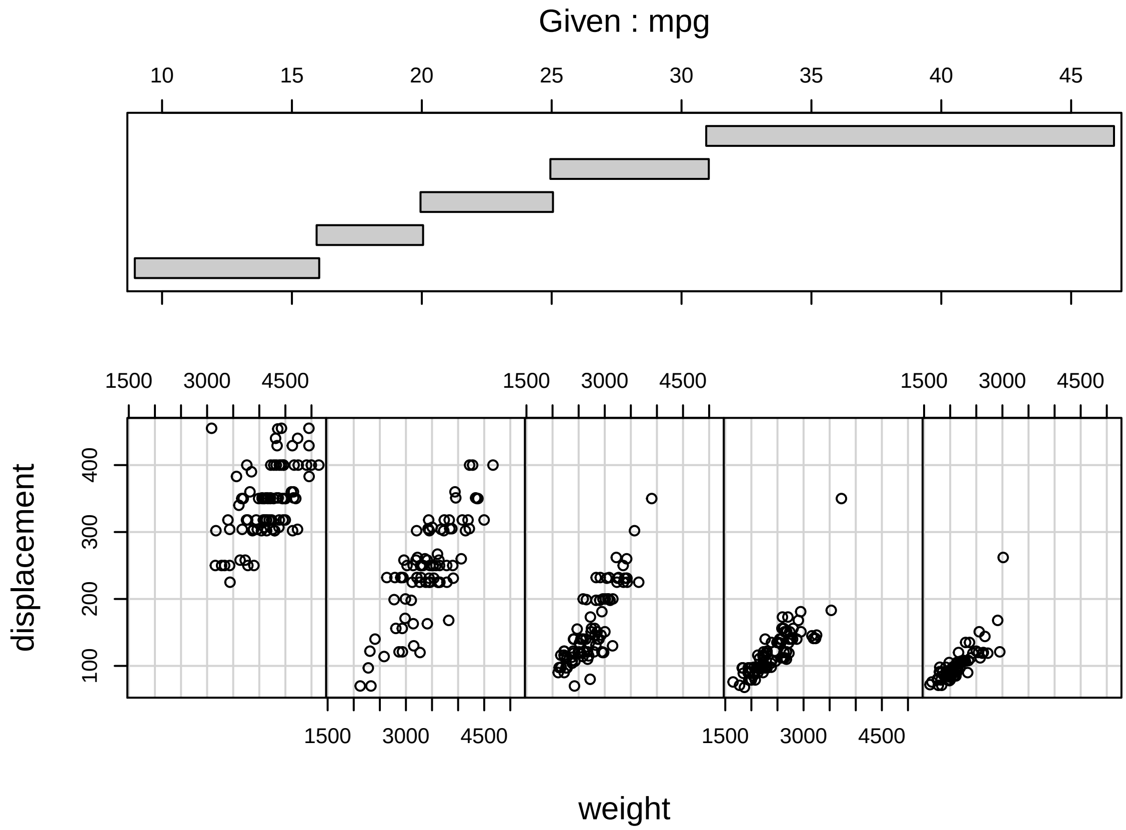

```r
par(mfrow=c(1,1))
beta5 <- coef(lm(data$displacement ~ data$weight))
scatter.smooth(data$weight, data$displacement, xlab = "weight", ylab = "displacmenet")
abline(beta5[1], beta5[2], col = "red", lty = 2)
title(paste("correlation:", round(cor(data$weight, data$displacement), 3)))
```



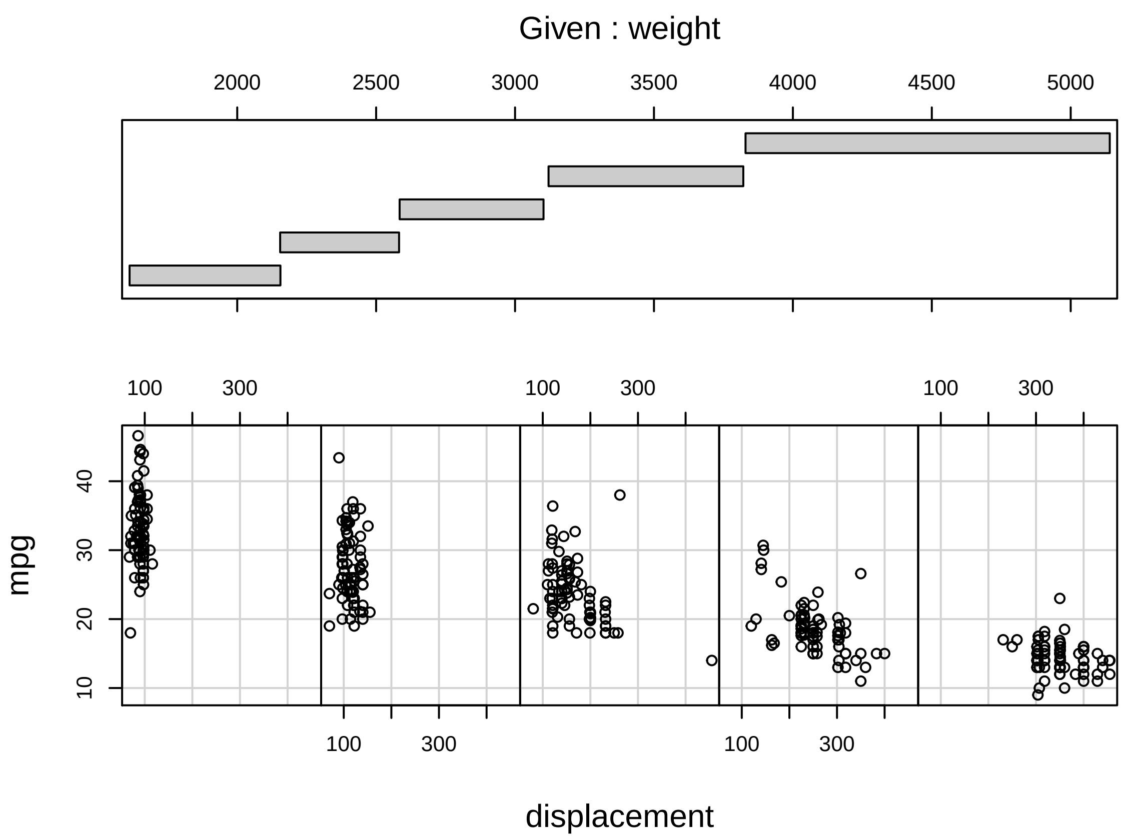
```r
#Find the correlation first for the selected variables
cor_mat <- as.matrix(cor(data[c(1, 3, 4, 5, 6)]))
cor_table <- arrange(melt(cor_mat), -abs(value))
cor_table_mod <- dplyr::filter(cor_table, value < 1)
cor_table_mod
```

```
##            Var1         Var2      value
## 1        weight displacement  0.9329944
## 2  displacement       weight  0.9329944
## 3    horsepower displacement  0.8972570
## 4  displacement   horsepower  0.8972570
## 5        weight   horsepower  0.8645377
## 6    horsepower       weight  0.8645377
## 7        weight          mpg -0.8322442
## 8           mpg       weight -0.8322442
## 9  displacement          mpg -0.8051269
## 10          mpg displacement -0.8051269
## 11   horsepower          mpg -0.7784268
## 12          mpg   horsepower -0.7784268
## 13 acceleration   horsepower -0.6891955
## 14   horsepower acceleration -0.6891955
## 15 acceleration displacement -0.5438005
## 16 displacement acceleration -0.5438005
## 17 acceleration          mpg  0.4233285
## 18          mpg acceleration  0.4233285
## 19 acceleration       weight -0.4168392
## 20       weight acceleration -0.4168392
```

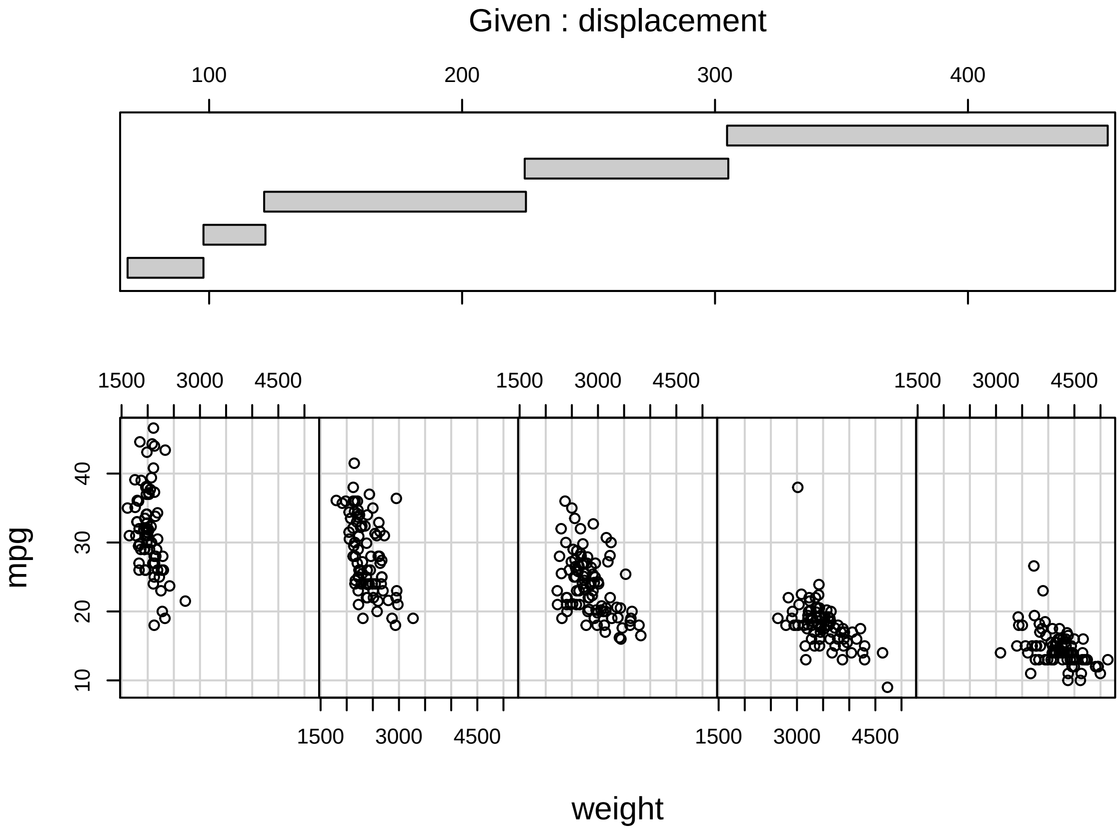
```r
#coplots
coplot(displacement ~ weight | mpg, data = data, rows = 1, overlap = 0, number = 5)
```



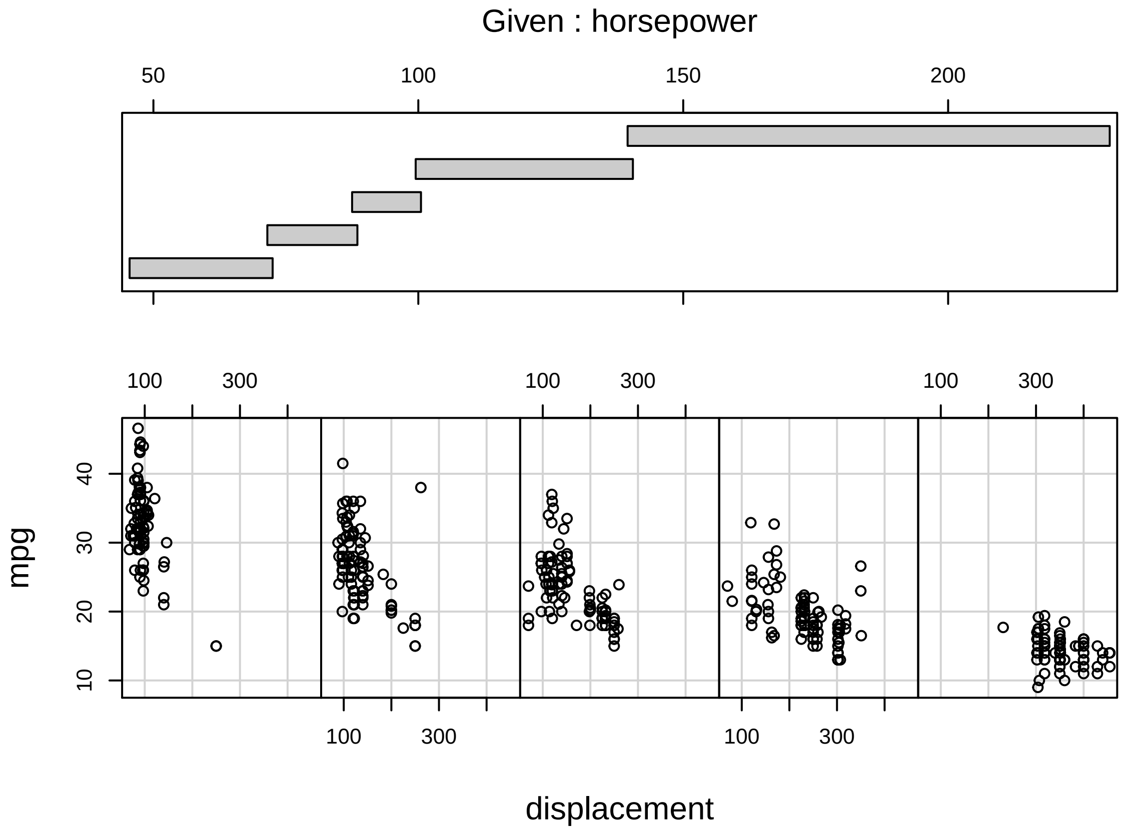

```
coplot(mpg ~ displacement | weight, data = data, rows = 1, overlap = 0, number = 5)
```



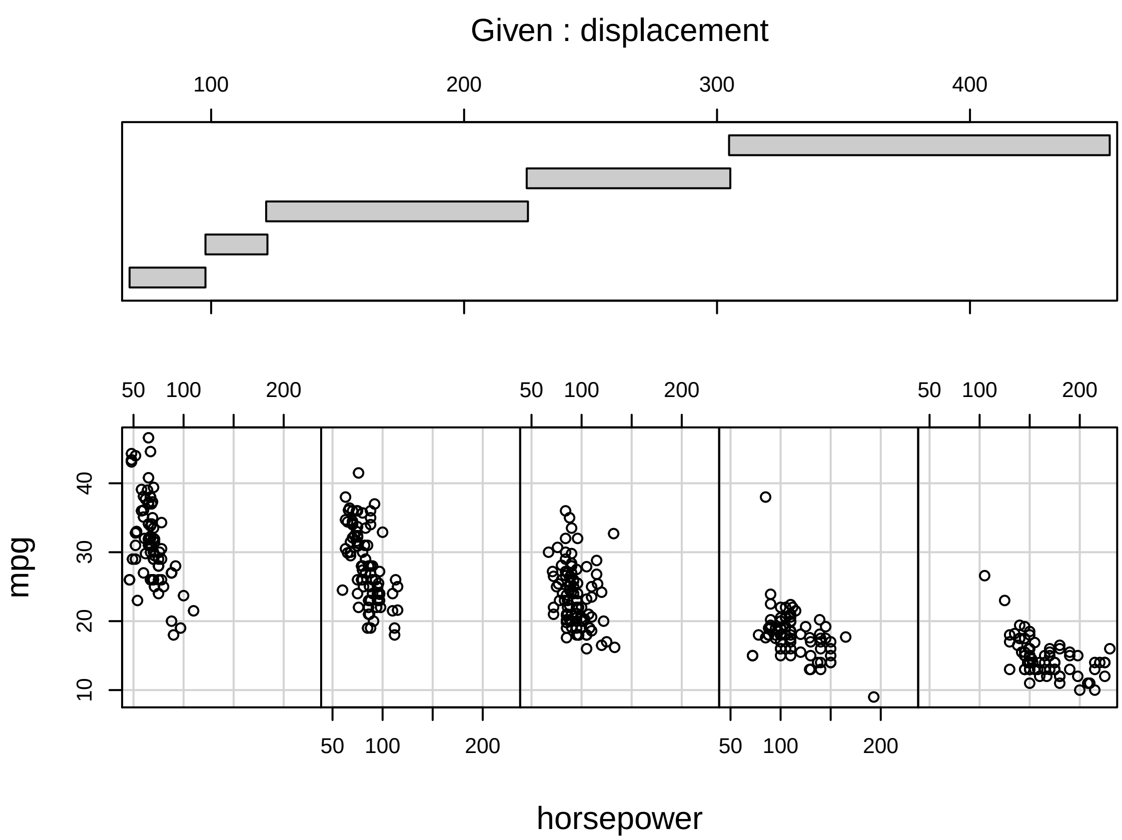
```r
coplot(mpg ~ weight | displacement, data = data, rows = 1, overlap = 0, number = 5)
```



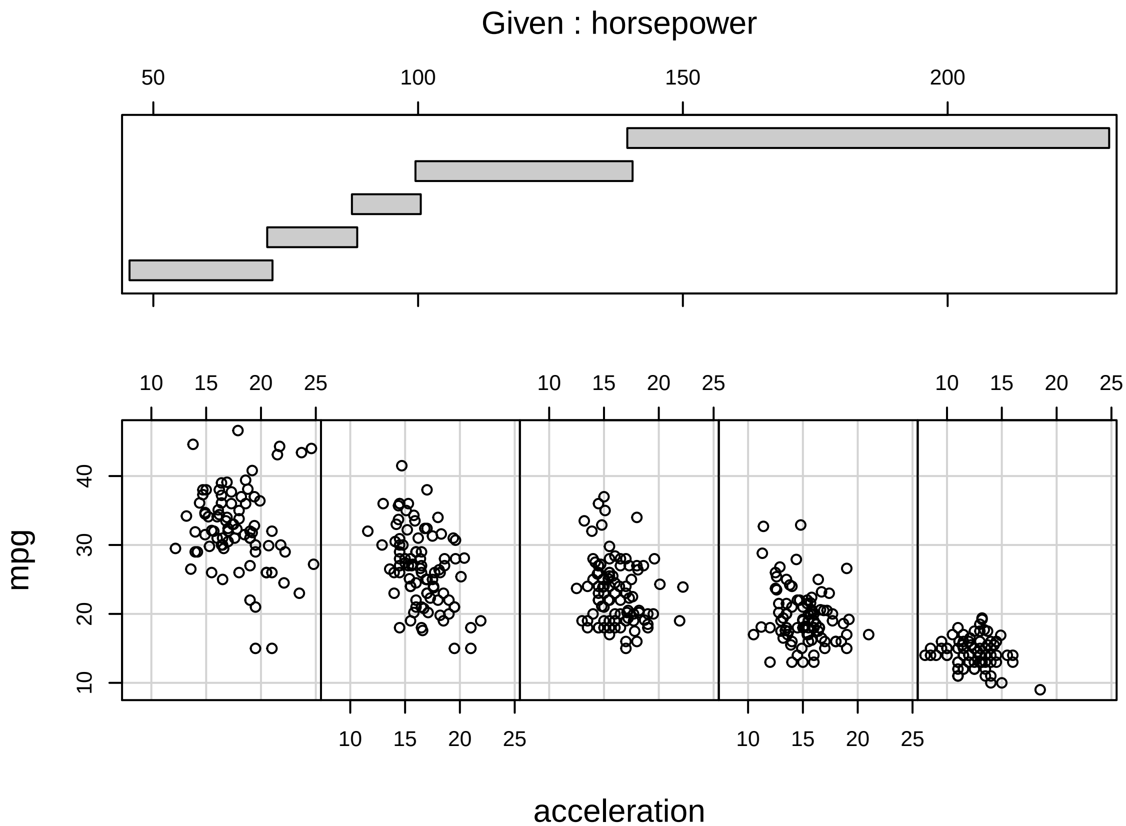

```r
coplot(mpg ~ displacement | horsepower, data = data, rows = 1, overlap = 0, number = 5)
```




```r
coplot(mpg ~ horsepower | displacement, data = data, rows = 1, overlap = 0, number = 5)
```



```r
coplot(mpg ~ acceleration | horsepower, data = data, rows = 1, overlap = 0, number = 5)
```




```r
#ecdf
#https://en.wikipedia.org/wiki/Empirical_distribution_function
#https://stat.ethz.ch/R-manual/R-devel/library/stats/html/ecdf.html
#http://r4ds.had.co.nz/exploratory-data-analysis.html


#quantile-quantile plots
#https://en.wikipedia.org/wiki/Q-Q_plot
#http://www.astrostatistics.psu.edu/su07/R/html/stats/html/qqnorm.html
```

Summary:

From the histogram, I found the four interesting things. First, acceleration have approximate normal distribution. Second, model_year diagram shows approximately-close uniform distribution (excluding the first column). Third, all the continuous variables, but acceleration shows positive skewed distribution. Fourth, Most of vehicles come from cylinder = 4 and origin = 1. To take a close look at those diagrams I mentioned, I decided to draw kernel density plots.

After, I made some boxplots. According to what says on the Fox textbook, boxplot shows "ONLY summary information on center, spread, skewness, and outliers." So, using the boxplots, I was wishing to get some of the information I could not find from other diagrams. I found out that there are not many outliers in most of variables (some in mpg and acceleration). As I mentioned in the histogram, Model-year is the one that makes it the most balanced shaped diagram. (uniform) Acceleration variable is also balanced. Origin and cylinders show huge positive skeness, and these are even obvious from the histograms (they have only a few bars, and some data are concentrated on left-sides.)

After I draw two versions of kernel-density estimate plots (and, kernel density estimation), I could conclude that acceleratoin is almost normal and model_year is uniform (bu a lot of bumps).

Next, I drew the scatterplot matrices, and as we have learned in the class, origin and cylinders can be considered as cateogorical/qualitative (although they are numerical variables, and by definition, they are both qualitative variabels. However, I would group them as categorical, just for me to do EDA easily.) Also, car_name is categorical, so I would exclude them in the scatterplot matrices. After, I came up with modified scatterplot matrice, it looks way better and organized. I want to take a closer look at scatterplots of "mpg and displacement," "mpg and weight," "weight and horsepower," "mpg and horsepower," "horsepower and displacement," and "displacement and weight." Since question b is asking to put mpg as response variable and others as explanatory variabels, I would make a scatterplot following those directions, so I could take advantage of my diagrams when making analysis.

After taking scatterplots, I made a table to compare the correlations amongst those variables I selected. So, I could have better summary of correlations. As I can see from the result, some of them are highly correlated: i.e. mpg and weight.

Furthermore, I implemented coplots. I will explain the reason why I used coplots for my EDA. I studied the relationship between weight, displacement, and mpg, as they have high correlation. As the textbook says on pg 48, plot focuses on the marginal relationships between the pairs of variables. And, we know that the EDA also requires to study partial relationships. Also, it is possible to have marginal relationship with no partial relationship, or vice versa. So, it is important for me to make coplots to find out more patterns and relationships amongst variables. For example, mpg and acceleration do not seem to be related at all, but when I see the plots conditioning on horsepower, there seem to be.

Personally, most of the results make sense to me, since I have been interested into mechanics and vehicles, but other people might think some of the results are weird.

Ordinary least square coefficient estimates:

$\hat{\beta} = (X^TX)^{-1}X^Ty$

Residual sum of squares (RSS):

Say that $y = X\beta + e$. As $\hat{e} = y - \hat{y} = y - X\hat{\beta} = y - X(X^TX)^{-1}X^Ty$, I can say that RSS $= \sum \hat{e}_i^2 = \hat{e}^T\hat{e} = ||\hat{e}||^2 = y^Ty - y^TX(X^TX)^{-1}X^Ty = y^T(I - X(X^TX)^{-1}X^T)y$.

SSreg:

SSreg = SStotal - SSres So, it is $\sum(\hat{y_i} - \bar{y})^2$

R^2:

$R^2 = \frac{SSreg}{SS_{total}} = \frac{SS_{total} - SS_{res}}{SS_{total}}$, where $SS_{res} = RSS$ and $SS_{total} = \sum(y_i - \bar{y})^2$

Part B

```r
y <- as.matrix(data$mpg)
X1 <- data[c(3:6)] #Continuous data set.
```

```r
intercept <- rep(1, nrow(data))

#Make dummy variable matrix for cylinders.
X_cylinders_catg <- length(unique(data$cylinders)) #gives different output with unique(data[2])
X_cylinders <- intercept
for (i in 2:X_cylinders_catg){
  X_cylinders <- cbind(X_cylinders, as.numeric(data$cylinders == sort(unique(data$cylinders))[i]))
}
colnames(X_cylinders)[1] <- c("cylinder intercept")
for (i in 2:X_cylinders_catg){
  colnames(X_cylinders)[i] <- paste(c("cylinder: "), unique(data$cylinders)[i])
}


#Make dummy variable matrix for model_year.
X_year_catg <- length(unique(data$model_year)) #gives different output with unique(data[27)
X_year <- intercept
for (i in 2:X_year_catg){
  X_year <- cbind(X_year, as.numeric(data$model_year == sort(unique(data$model_year))[i]))
}
colnames(X_year)[1] <- c("year intercept")
for (i in 2:X_year_catg){
  colnames(X_year)[i] <- paste(c("year: "), unique(data$year)[i])
}


#Make dummy variable matrix for origin.
X_origin_catg <- length(unique(data$origin)) #gives different output with unique(data[8])
X_origin <- intercept
for (i in 2:X_origin_catg){
  X_origin <- cbind(X_origin, as.numeric(data$origin == sort(unique(data$origin))[i]))
}
colnames(X_origin)[1] <- c("origin")
for (i in 2:X_origin_catg){
  colnames(X_origin)[i] <- paste(c("origin: "), unique(data$origin)[i])
}

X <- cbind(intercept, X_cylinders, X1, X_year, X_origin) #1 + 5 + 4 + 13 + 3 = 26 columns
X <- as.matrix(X)

colnames(X)[1] <- "intercept"

X <- X[,-c(2, 11, 24)]



#coefficient estimates
  #Method 1
beta <- round(solve(t(X) %*% X) %*% t(X) %*% y, 5)
#as it says on the lecture handout, it is not the most efficient way...

  #Method 2
Q <- qr.Q(qr(X))
```

```r
R <- qr.R(qr(X))

beta2 <- round(solve(R) %*% t(Q) %*% y, 5)

beta
```

```
##                   [,1]
## intercept    30.91684
## cylinder:  4  6.93992
## cylinder:  6  6.63773
## cylinder:  3  4.29731
## cylinder:  5  6.36681
## displacement  0.01182
## horsepower   -0.03923
## weight       -0.00518
## acceleration  0.00361
## year:         0.91043
## year:        -0.49031
## year:        -0.55289
## year:         1.24200
## year:         0.87040
## year:         1.49666
## year:         2.99870
## year:         2.97378
## year:         4.89618
## year:         9.05893
## year:         6.45816
## year:         7.83758
## origin:  3    1.69329
## origin:  2    2.29293
```

```r
#coefficient estimates using lm():
a <- lm(mpg ~ displacement + horsepower + weight + acceleration + factor(cylinders) +
          factor(model_year) + factor(origin), data = data[,-9])

a #I got the same coefficients estimates.
```

```
##
## Call:
## lm(formula = mpg ~ displacement + horsepower + weight + acceleration +
##     factor(cylinders) + factor(model_year) + factor(origin),
##     data = data[, -9])
##
## Coefficients:
##          (Intercept)          displacement            horsepower
##            30.916841              0.011825             -0.039232
##               weight          acceleration    factor(cylinders)4
##            -0.005180              0.003608              6.939922
##   factor(cylinders)5    factor(cylinders)6    factor(cylinders)8
##             6.637731              4.297314              6.366813
## factor(model_year)71  factor(model_year)72  factor(model_year)73
##             0.910429             -0.490306             -0.552893
## factor(model_year)74  factor(model_year)75  factor(model_year)76
##             1.241998              0.870402              1.496660
## factor(model_year)77  factor(model_year)78  factor(model_year)79
```

```
##             2.998697              2.973778              4.896176
## factor(model_year)80  factor(model_year)81  factor(model_year)82
##             9.058932              6.458158              7.837585
##      factor(origin)2       factor(origin)3
##             1.693285              2.292927
```

```r
#residual sum of squares
y_hat <- X %*% beta
residual <- y - y_hat
RSS <- as.numeric(t(residual) %*% residual)




#SSreg
y_bar <- rep(mean(y), nrow(y))
SSreg <- as.numeric(t(y_hat - y_bar) %*% (y_hat - y_bar))




#SStotal
SStotal <- as.numeric(t(y - y_bar) %*% (y - y_bar))


#R^2
cor_sq <- SSreg / SStotal
cor_sq2 <- 1 - (RSS / SStotal)


summary(a)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + horsepower + weight + acceleration +
##     factor(cylinders) + factor(model_year) + factor(origin),
##     data = data[, -9])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9267 -1.6678 -0.0506  1.4493 11.6002
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          30.9168415  2.3608985  13.095  < 2e-16 ***
## displacement          0.0118246  0.0067755   1.745 0.081785 .
## horsepower           -0.0392323  0.0130356  -3.010 0.002795 **
## weight               -0.0051802  0.0006241  -8.300 1.99e-15 ***
## acceleration          0.0036080  0.0868925   0.042 0.966902
## factor(cylinders)4    6.9399216  1.5365961   4.516 8.48e-06 ***
## factor(cylinders)5    6.6377310  2.3372687   2.840 0.004762 **
## factor(cylinders)6    4.2973139  1.7057848   2.519 0.012182 *
## factor(cylinders)8    6.3668129  1.9687277   3.234 0.001331 **
## factor(model_year)71  0.9104285  0.8155744   1.116 0.265019
## factor(model_year)72 -0.4903062  0.8038193  -0.610 0.542257
```

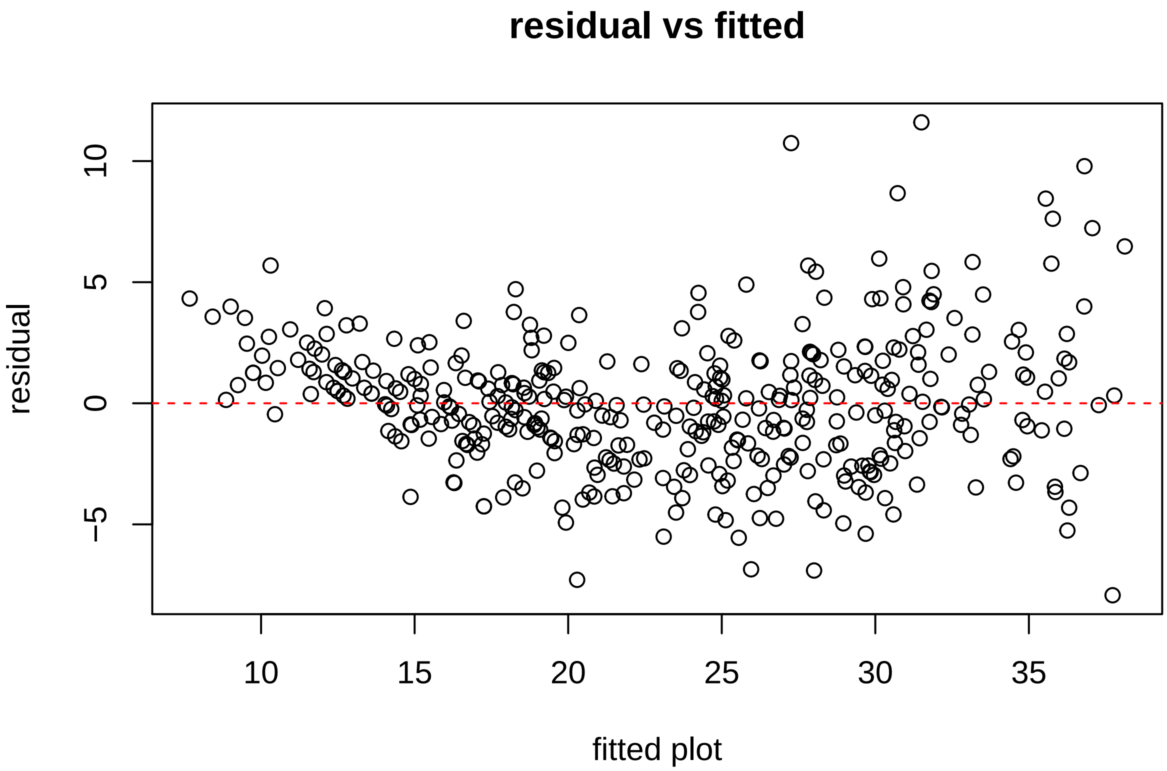
```
## factor(model_year)73 -0.5528934  0.7214463  -0.766 0.443947
## factor(model_year)74  1.2419976  0.8547434   1.453 0.147056
## factor(model_year)75  0.8704016  0.8374036   1.039 0.299297
## factor(model_year)76  1.4966598  0.8019080   1.866 0.062782 .
## factor(model_year)77  2.9986967  0.8198949   3.657 0.000292 ***
## factor(model_year)78  2.9737783  0.7792185   3.816 0.000159 ***
## factor(model_year)79  4.8961763  0.8248124   5.936 6.74e-09 ***
## factor(model_year)80  9.0589316  0.8751948  10.351  < 2e-16 ***
## factor(model_year)81  6.4581580  0.8637018   7.477 5.58e-13 ***
## factor(model_year)82  7.8375850  0.8493560   9.228  < 2e-16 ***
## factor(origin)2       1.6932853  0.5162117   3.280 0.001136 **
## factor(origin)3       2.2929268  0.4967645   4.616 5.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.848 on 369 degrees of freedom
## Multiple R-squared:  0.8744, Adjusted R-squared:  0.8669
## F-statistic: 116.8 on 22 and 369 DF,  p-value: < 2.2e-16
```

Let the mpg as response variable. Since the 9th variable, car_name is string, I would exclude it in my model/design matrix X. Also, I would consider cylinders, model_year, and origin as dummy variables. I always exclude the first column of dummy variables, as I include the intercept.

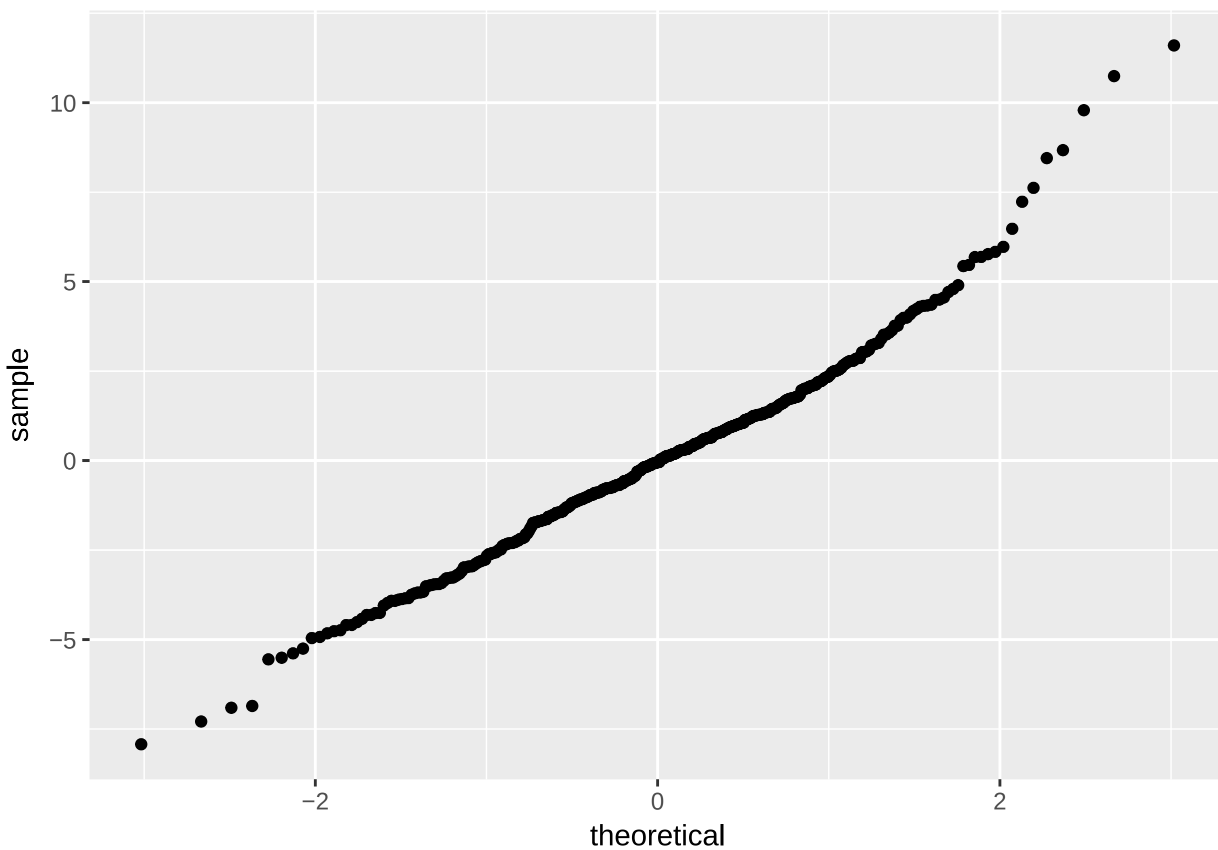
The coefficient estimates, residual sum of squares, SSreg, and coefficient of determination(R^2) are all similar with what I got from lm() function. RSS is around 2992, SSreg is around 20828, and SStotal is around 23819. So, SStotal is equal to the sum of SSreg and RSS, which makes sense. Also, coefficient of determination (= R^2) of the model is around 0.8744, and this is pretty high. It means that the "full" model is better than the "small" model, and this backs up why we are using this model. So, RSS is smaller compared to TSS, so explanatory variables are useful in predicting the response. So, I can conclude that the response variable, mpg is can be well predicted by the model we have. (regression line approximates the outcomes good enough.) Since we have high SStotal and SSreg compared to RSS, I could have high coefficient of determination. One drawback of this conclusoin is that RSS decreases as I add more explanatory variables. I know I have 7 explanatory variables, and it is pretty a lot. (even though it is considered small in real industry.)

Part C

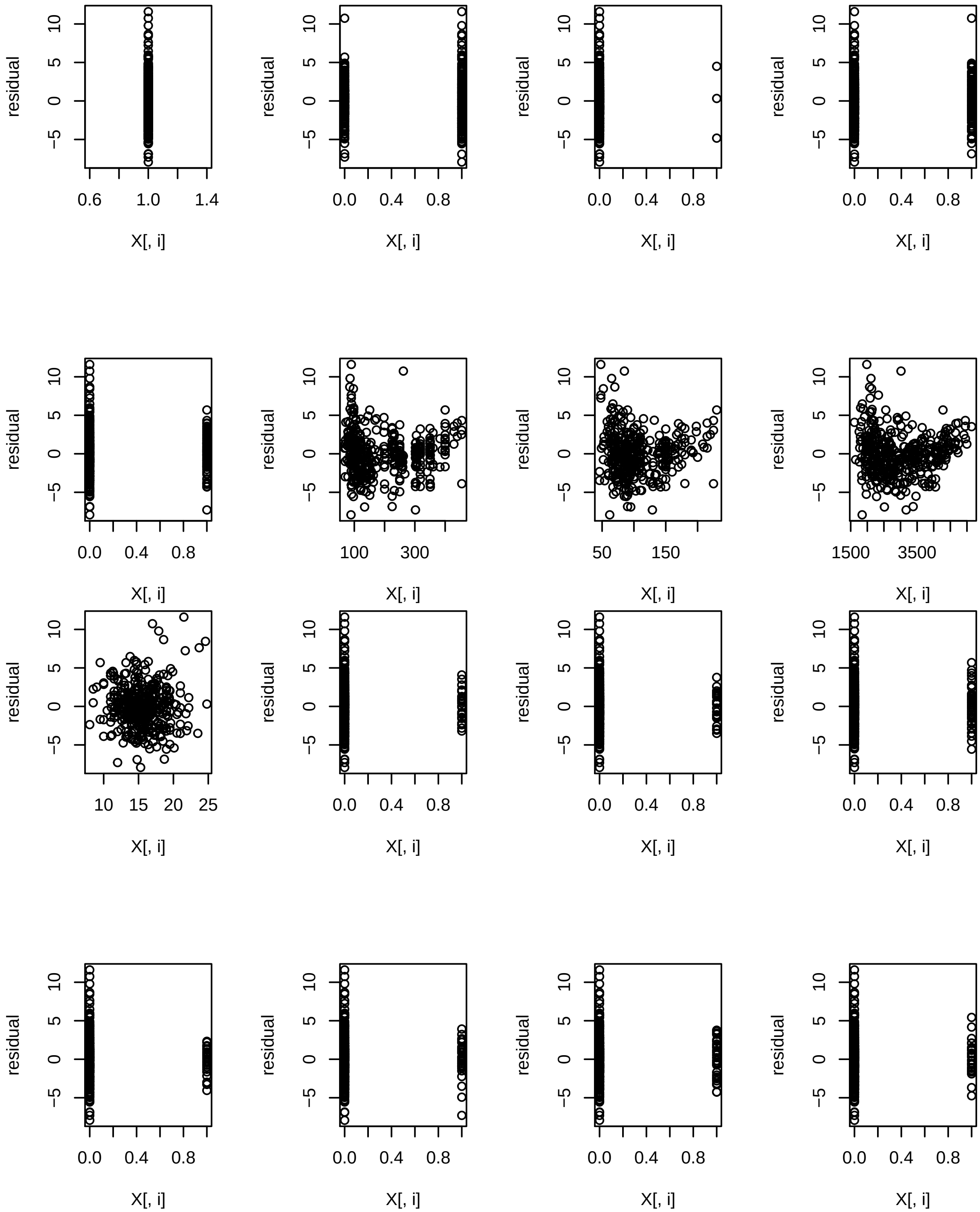
```r
plot(y_hat ,residual, ylab = "residual", xlab = "fitted plot", main = "residual vs fitted", type = "p")
abline(0, 0, col = "red", lty = 2)
```

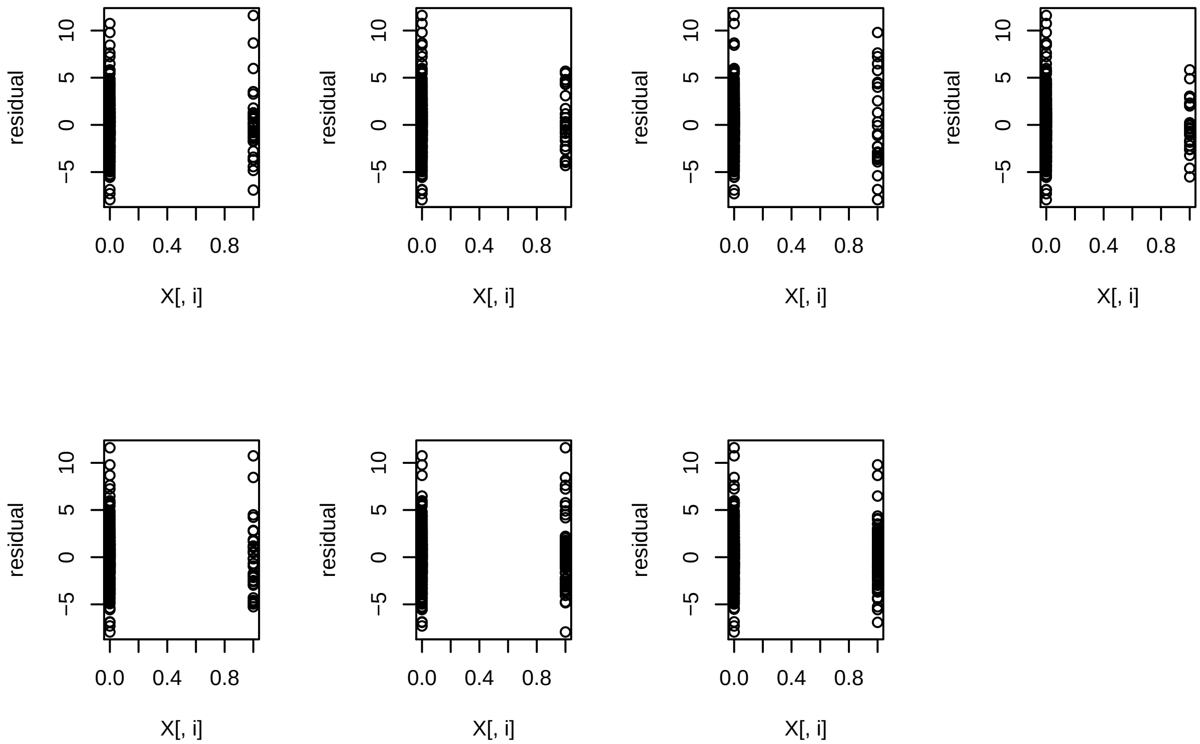



```r
saving_data <- data.frame(residual)
ggplot (saving_data) + geom_qq(aes(sample = residual))
```



```r
#Residual vs predictor
par(mfrow=c(2,4))
for (i in 1:ncol(X)){
  plot(X[,i], residual)
}
```





To tell the truth, there seems to be a small pattern on this graph; however, I will not say there is too much pattenr here. Variance of residual seems to increase (but pretty constant… but it is still heteroscedasticitic) as fitted plot (y hat) increases. Also, I tried to check whether reisduals follow normal probability plots, and I found out that light-tailed from my second quantile-quantile plots. From what I have learned from the last lecture (09/07 Thurs), since variance is not constant and has heavy-tailed distribution, it means that response variable does not follow normal distribution.

So, my conclusion is since the residuals roughly form a horizontal band around “residual = 0,” the linear relationship is pretty reasonable. (but, I think there is a better relationship than linear, since they show a little bit of pattern.) Also, since there are only three standing out residuals where fitted plot is around 27 ~ 30 from the plot, this implies that I can conlude there is **no significant outlier.**

By the way, an alternative to residual v.s. fitted plot is residual v.s. predictor plot. Those two interpretations are the same.

I found one really good explnation of how residual v.s. predictor plot can be used, following: Click here

This “residuals versus weight” plot can be used to determine whether we should add the predictor weight to the model that already contains the predictor age. In general, if there is some non-random pattern to the plot, it indicates that it would be worthwhile adding the predictor to the model. In essence, you can think of the residuals on the y axis as a “new response,” namely the individual’s diastolic blood pressure adjusted for their age. If a plot of the “new response” against a predictor shows a non-random pattern, it indicates that the predictor explains some of the remaining variability in the new (adjusted) response. Here, there is a pattern in the plot. It appears that adding the predictor weight to the model already containing age would help to explain some of the remaining variability in the response.

After I plotted residual vs predictor (all of 24 predictors), most of them show pattern and not bounded to “residual = 0.” And, this is realistic as most of the data in real industry shows pattern, which means that there is usually a better way to model the data than linear.

Part D

Conclusion:

As I have proved in part b), I have a good linear model (with high coefficient of determination) where mpg can be calculated with 7 predicted variables. And, this makes sense that mpg has a linear relationship with other varriables I have chosen from part a) where different coplots (conditioning on a/multiple variable(s)) and pair plots (marginal, meaning when I assume other variables are ignored) show decent linear relationships. However, part d) shows a possibility that the model can be improved. There is no significant outlier I could find; however, variance of residuals seem not constant. This is the real data comes from the real world, and I believe it is reasonable to conclude that linear modeling can always be improved.