Extra Project

Yunxuan

2022-12-09

```
Cars <- read.table("cars.txt", head = T)
nrow(Cars)
```

```
## [1] 32
```

```
Cars
```

```
##                   name  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1            Mazda RX4 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2        Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3           Datsun 710 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 4       Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 5    Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 6              Valiant 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## 7           Duster 360 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## 8            Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## 9             Merc 230 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## 10            Merc 280 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 11           Merc 280C 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## 12          Merc 450SE 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## 13          Merc 450SL 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## 14         Merc 450SLC 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## 15  Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## 16 Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## 17   Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## 18            Fiat 128 32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## 19         Honda Civic 30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## 20      Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## 21       Toyota Corona 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 22    Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## 23         AMC Javelin 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## 24          Camaro Z28 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## 25    Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## 26           Fiat X1-9 27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## 27       Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 28        Lotus Europa 30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## 29      Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 30        Ferrari Dino 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## 31       Maserati Bora 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## 32          Volvo 142E 21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

1. Partition the data set into two sets a training data and a test data. Remove every fifth observation from the data for use as a test sample.

There are 32 observations. So the set test will contain 6 observations and the set train will contain 32-6=26 observations.

```
c<- 1:nrow(Cars)
train <- Cars[!c%%5==0,]
test <- Cars[c%%5==0,]
nrow(train)
```

```
## [1] 26
```

```
nrow(test)
```

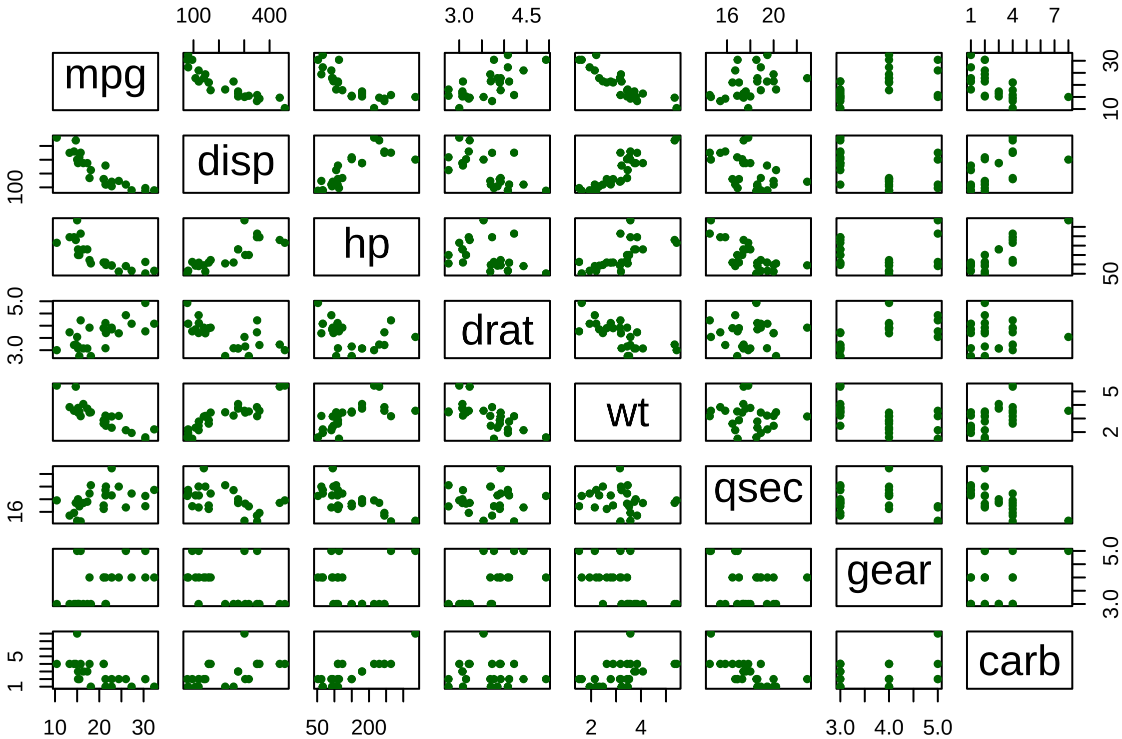
```
## [1] 6
```

2. Perform an exploratory analysis. Comment on your findings.

```
library(ggplot2)
library(cowplot)
library(ISLR)

cars <- train[, -c(1,3,9,10)]
summary(cars)
```

```
##       mpg             disp             hp             drat      
##  Min.   :10.40   Min.   : 75.7   Min.   : 52.0   Min.   :2.760  
##  1st Qu.:15.28   1st Qu.:120.5   1st Qu.: 95.5   1st Qu.:3.098  
##  Median :19.55   Median :196.3   Median :111.5   Median :3.715  
##  Mean   :20.07   Mean   :221.8   Mean   :145.2   Mean   :3.622  
##  3rd Qu.:22.80   3rd Qu.:303.2   3rd Qu.:180.0   3rd Qu.:3.920  
##  Max.   :32.40   Max.   :460.0   Max.   :335.0   Max.   :4.930  
##        wt             qsec            gear            carb      
##  Min.   :1.513   Min.   :14.50   Min.   :3.000   Min.   :1.000  
##  1st Qu.:2.504   1st Qu.:16.88   1st Qu.:3.000   1st Qu.:2.000  
##  Median :3.203   Median :17.71   Median :4.000   Median :2.000  
##  Mean   :3.168   Mean   :17.90   Mean   :3.692   Mean   :2.731  
##  3rd Qu.:3.570   3rd Qu.:18.90   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :5.424   Max.   :22.90   Max.   :5.000   Max.   :8.000
```

```
pairs(cars[,c("mpg","disp","hp","drat","wt","qsec","gear","carb")],col="darkgreen",pch=20)
```




```
cor(cars[,c("mpg","disp","hp","drat","wt","qsec","gear","carb")])
```

```
##             mpg       disp          hp        drat         wt        qsec
## mpg   1.0000000 -0.8879788 -0.77751430  0.66124336 -0.8592776  0.40733515
## disp -0.8879788  1.0000000  0.82413517 -0.65054746  0.8889453 -0.48080537
## hp   -0.7775143  0.8241352  1.00000000 -0.38749899  0.6489729 -0.70529394
## drat  0.6612434 -0.6505475 -0.38749899  1.00000000 -0.6853894  0.02358719
## wt   -0.8592776  0.8889453  0.64897286 -0.68538941  1.0000000 -0.16302267
## qsec  0.4073352 -0.4808054 -0.70529394  0.02358719 -0.1630227  1.00000000
## gear  0.5113112 -0.4801054 -0.09115853  0.70791536 -0.5747590 -0.21560113
## carb -0.5623724  0.5332080  0.80794216 -0.07828427  0.4678306 -0.66441318
##             gear        carb
## mpg   0.51131118 -0.56237236
## disp -0.48010537  0.53320804
## hp   -0.09115853  0.80794216
## drat  0.70791536 -0.07828427
## wt   -0.57475897  0.46783065
## qsec -0.21560113 -0.66441318
## gear  1.00000000  0.19999446
## carb  0.19999446  1.00000000
```

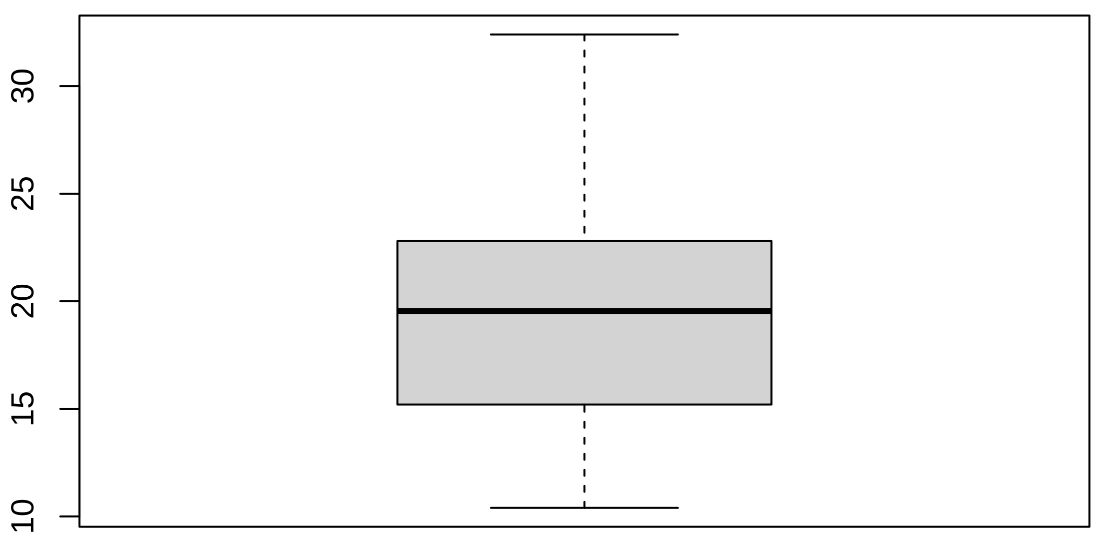
We see that predictors such as mpg, disp and hp has a relationship between each other, as the graph shows.

```
hist(train$mpg)
```


```
summary(train$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.28   19.55   20.07   22.80   32.40
```

```
boxplot(train$mpg)
```



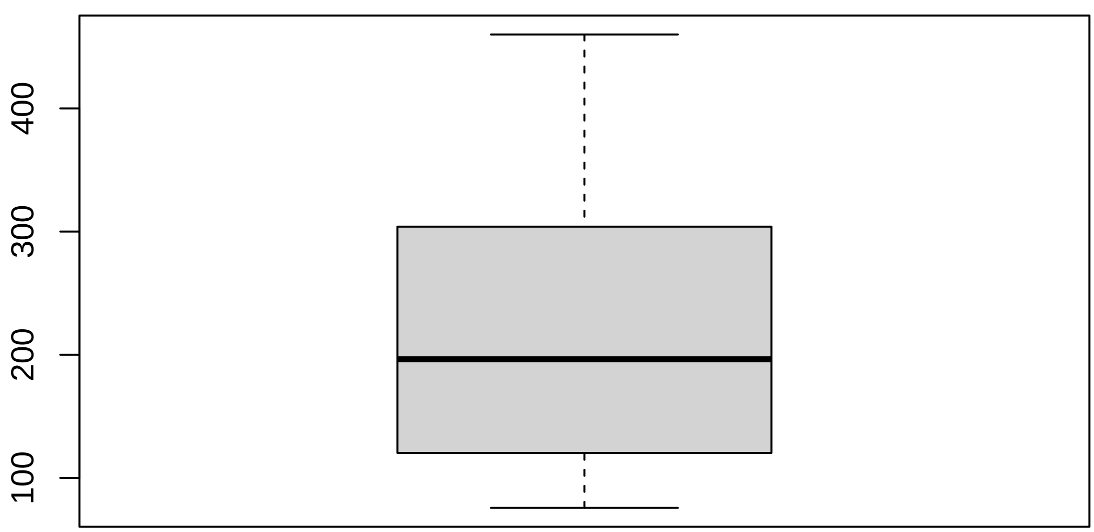
For mpg, most values are in the interval [15,28,22,80]. There is no outlier. The values are comparatively concentrated.

```
hist(train$disp)
```


```
summary(train$disp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    75.7   120.5   196.3   221.8   303.2   460.0
```

```
boxplot(train$disp)
```



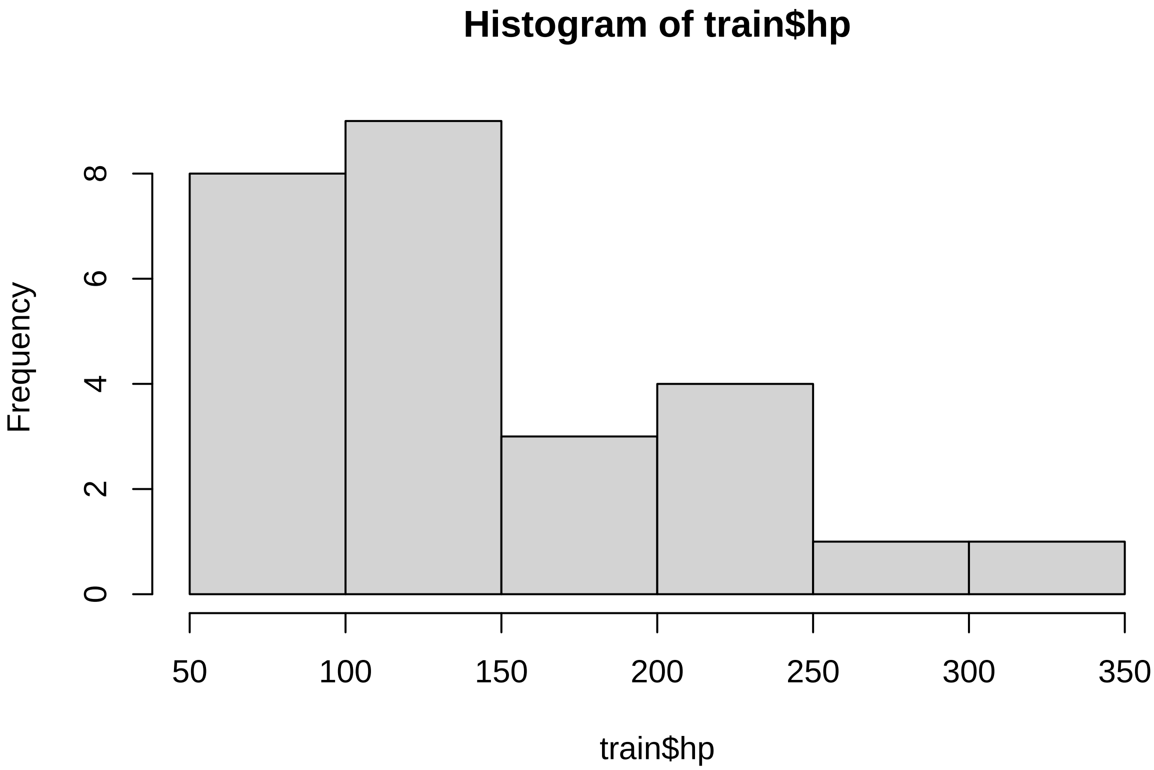
For disp, most values are in the interval [120.5,303.2]. There is no outlier. The values are comparatively scattered.

```
summary(train$cyl)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.000   4.000   6.000   6.077   8.000   8.000
```

For cyl, the number of getting 6 cyl is the least while the number of getting 8 cyl is the most.

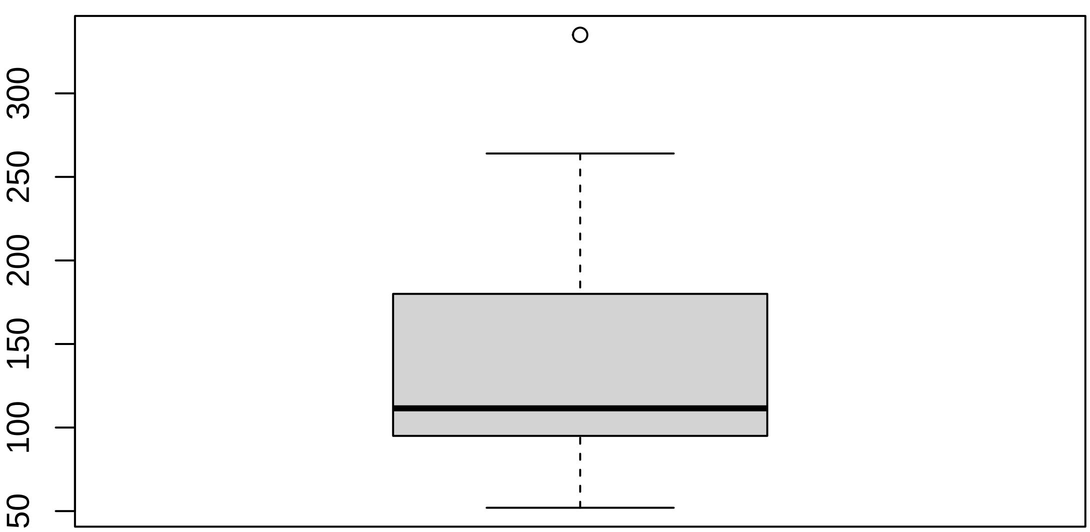
```
hist(train$hp)
```




```
summary(train$hp)
```

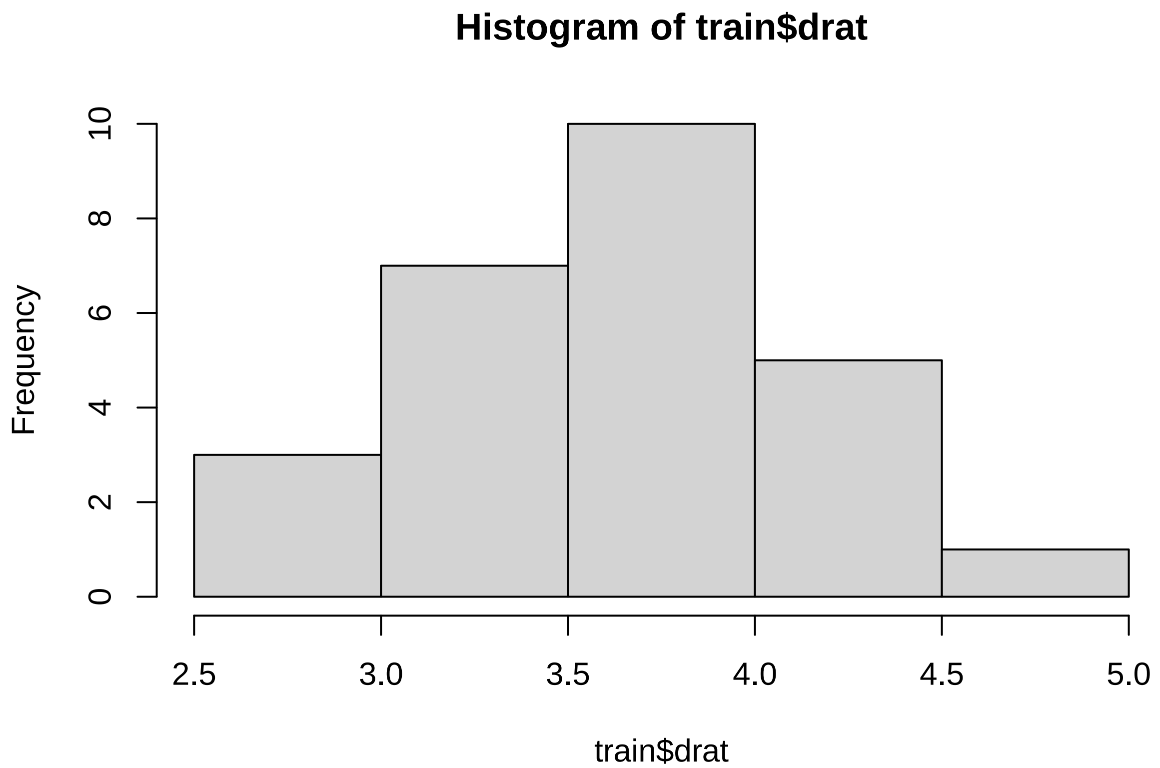
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    52.0    95.5   111.5   145.2   180.0   335.0
```

```
boxplot(train$hp)
```



For hp, most values are in the interval [95.5,180]. There is an outlier, but we should keep it. The values are comparatively scattered.

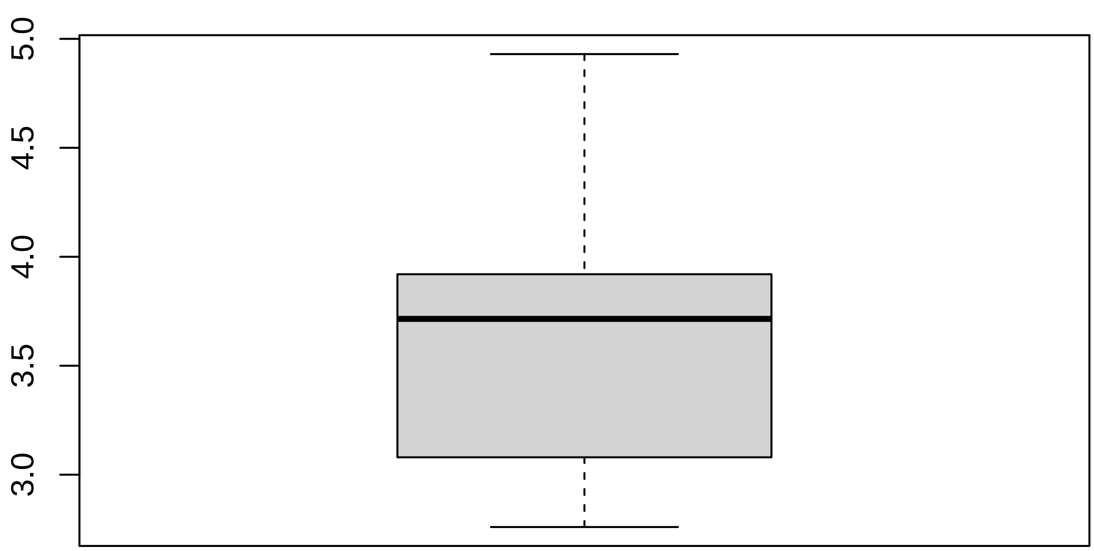
```
hist(train$drat)
```




```
summary(train$drat)
```

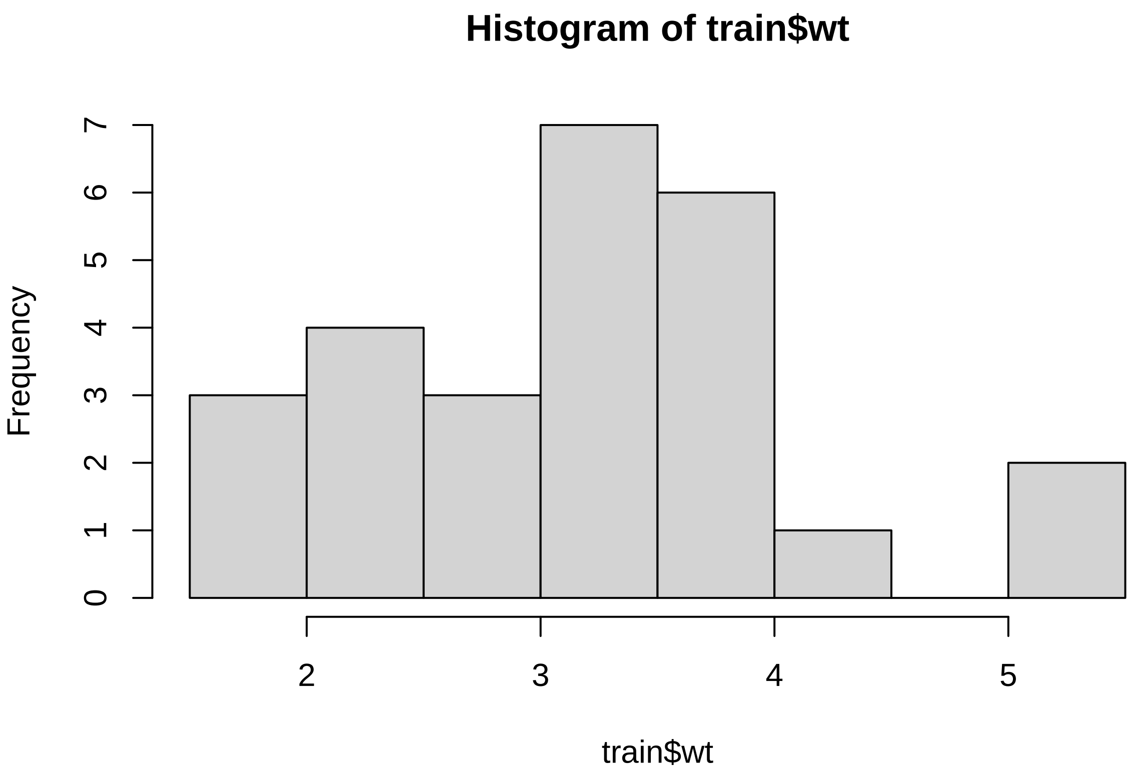
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.760   3.098   3.715   3.622   3.920   4.930
```

```
boxplot(train$drat)
```



For drat, most values are in the interval [3.098,3.920]. There is no outlier. The values are comparatively concentrated.

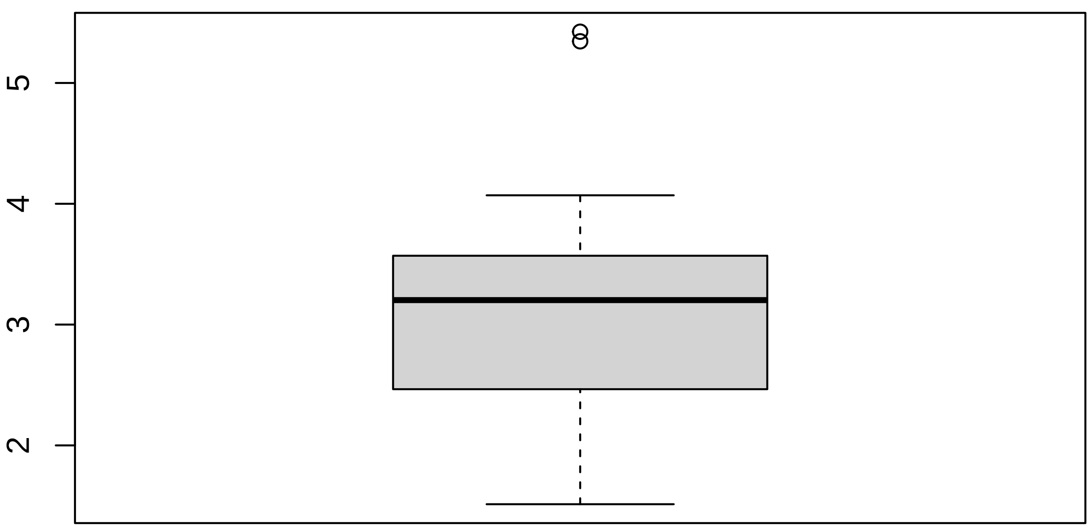
```
hist(train$wt)
```




```
summary(train$wt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.513   2.504   3.203   3.168   3.570   5.424
```

```
boxplot(train$wt)
```



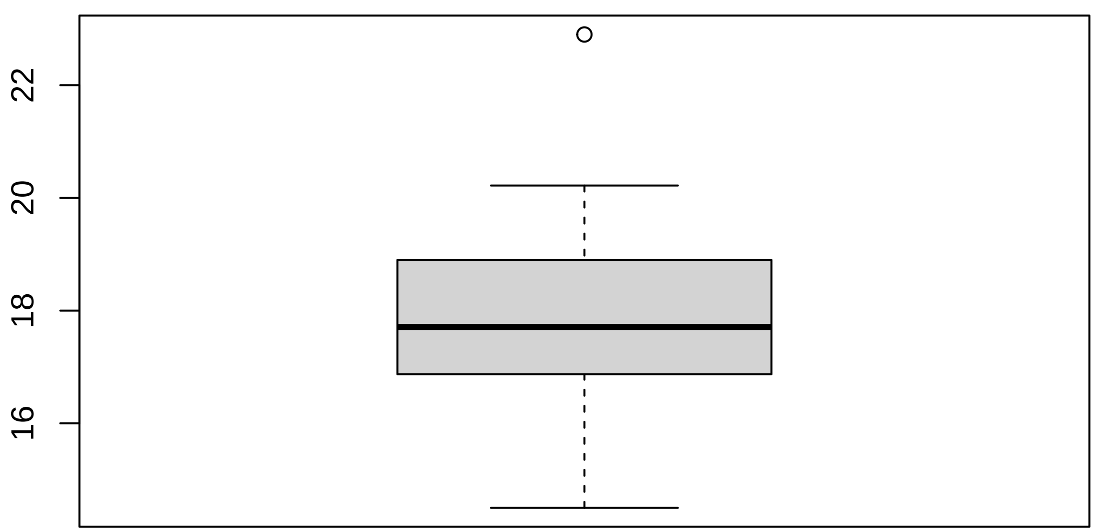
For wt, most values are in the interval [2.5,3.6]. There are two outliers, but we should keep them.

```
hist(train$qsec)
```


```
summary(train$qsec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.50   16.88   17.71   17.90   18.90   22.90
```

```
boxplot(train$qsec)
```



For qsec, most values are in the interval [17,19]. There is an outlier, but we should keep it. The values are comparatively concentrated.

```
summary(train$vs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4615  1.0000  1.0000
```

The number of being 0 is a little bit more than number being 1 for vs.

```
summary(train$am)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4231  1.0000  1.0000
```

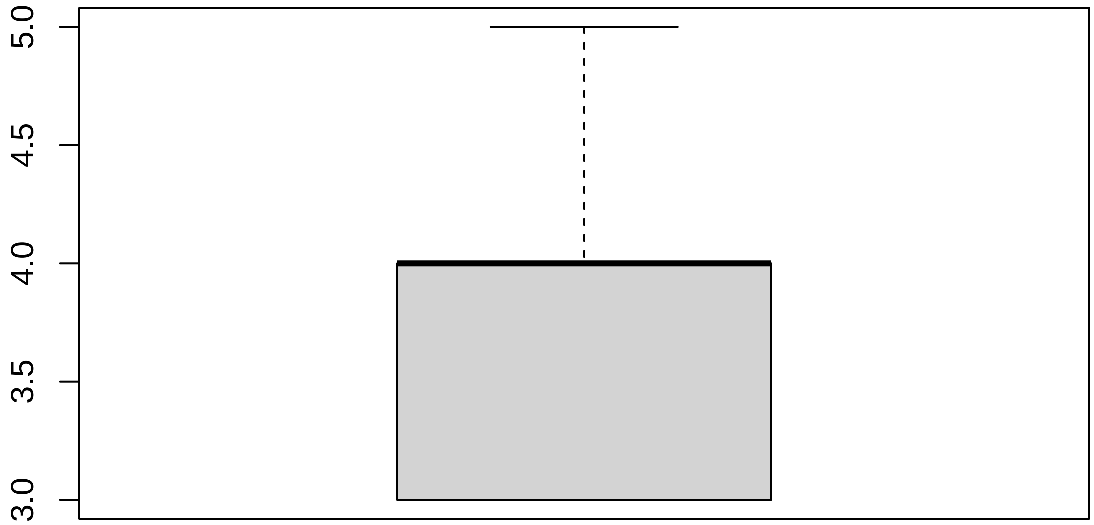
The number of being 0 is a little bit more than number being 1 for am.

```
hist(train$gear)
```


```
summary(train$gear)
```

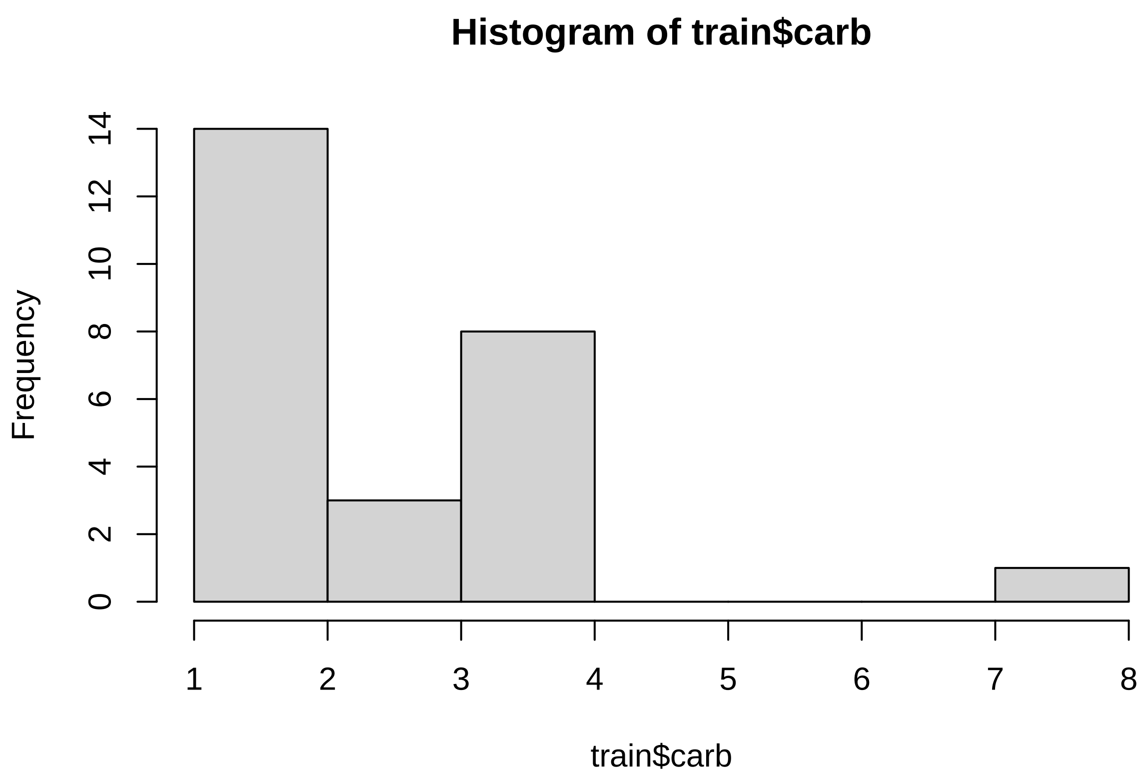
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   3.000   4.000   3.692   4.000   5.000
```

```
boxplot(train$gear)
```



For gear, most values are in the interval [3,4]. There is no outlier.

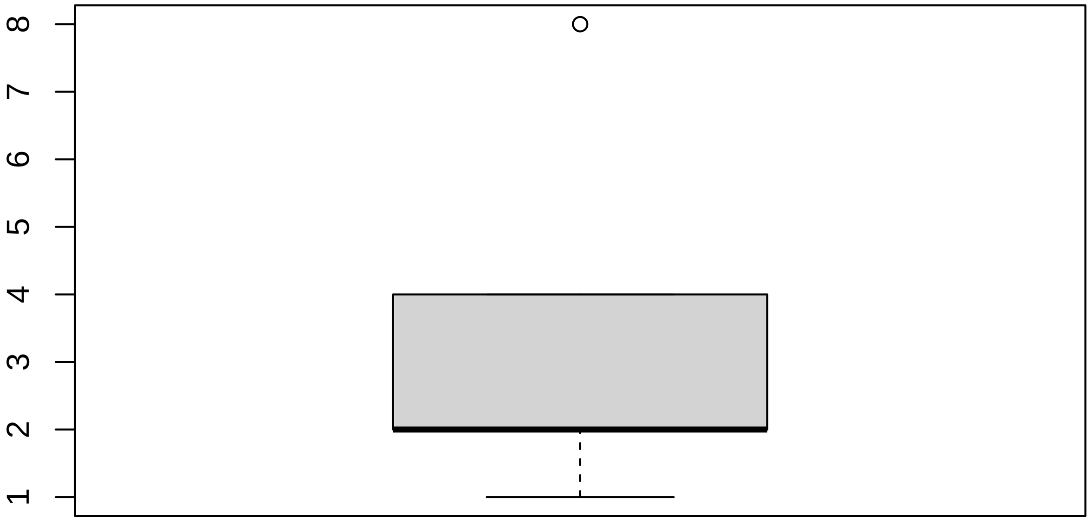
```
hist(train$carb)
```




```
summary(train$carb)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.731   4.000   8.000
```

```
boxplot(train$carb)
```



For carb, most values are in the interval [2,4]. There is an outlier, but we should keep it. The distribution is comparatively concentrated.

3. Perform a regression analysis and come up with the best multiple linear regression model that explains the response mpg in terms of the rest (except name). Comment on your findings and explain the methods and strategies that you employed in order to select the model you picked. Things you have to include in this part:

- Model diagnostics
- Justification on whether it is necessary or not to do any transformation on the response or the predictors
- Variable selection

```
lmod1 <- lm(mpg ~ factor(cyl)+disp+hp+drat+wt+qsec+factor(vs)+factor(am)+gear+carb, data=train)
summary(lmod1)
```

```
##
## Call:
## lm(formula = mpg ~ factor(cyl) + disp + hp + drat + wt + qsec +
##     factor(vs) + factor(am) + gear + carb, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1964 -1.2457 -0.2034  0.7388  5.0684
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.33640   17.38473   1.457    0.167
## factor(cyl)6 -2.85871    2.48654  -1.150    0.270
## factor(cyl)8 -0.66735    4.66762  -0.143    0.888
## disp          0.01117    0.02283   0.489    0.632
## hp           -0.04869    0.03021  -1.611    0.129
## drat         -0.48406    1.86041  -0.260    0.799
## wt           -2.71321    2.16587  -1.253    0.231
## qsec          0.22852    0.82846   0.276    0.787
## factor(vs)1   1.94056    2.50451   0.775    0.451
## factor(am)1   2.25309    2.30330   0.978    0.345
## gear          0.73942    1.51024   0.490    0.632
## carb          0.67277    1.03493   0.650    0.526
##
```

```
## Residual standard error: 2.592 on 14 degrees of freedom
## Multiple R-squared:  0.8877, Adjusted R-squared:  0.7995
## F-statistic: 10.06 on 11 and 14 DF,  p-value: 7.337e-05
```

Diagnostic Test: 1. Check Error Assumptions 1i. Check Constant Variance.

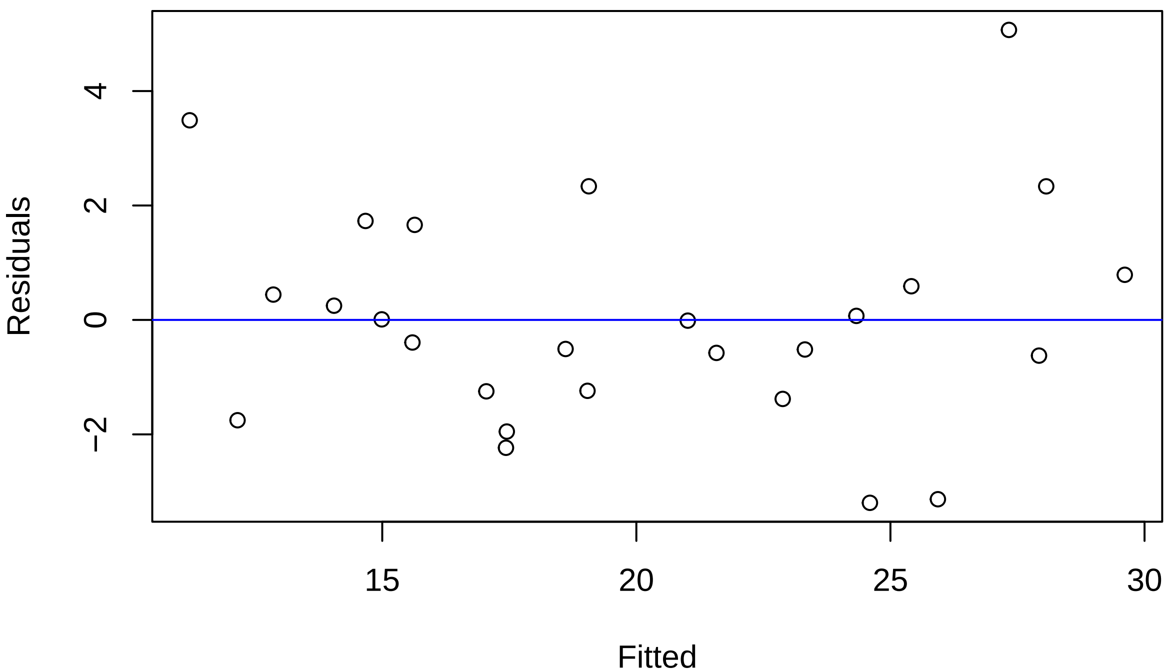
```
require(lmtest)
```

```
## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
require(MASS)
```

```
## Loading required package: MASS
```

```
require(ggplot2)
plot(fitted(lmod1),residuals(lmod1),xlab='Fitted',ylab='Residuals')
abline(h=0, col="blue")
```



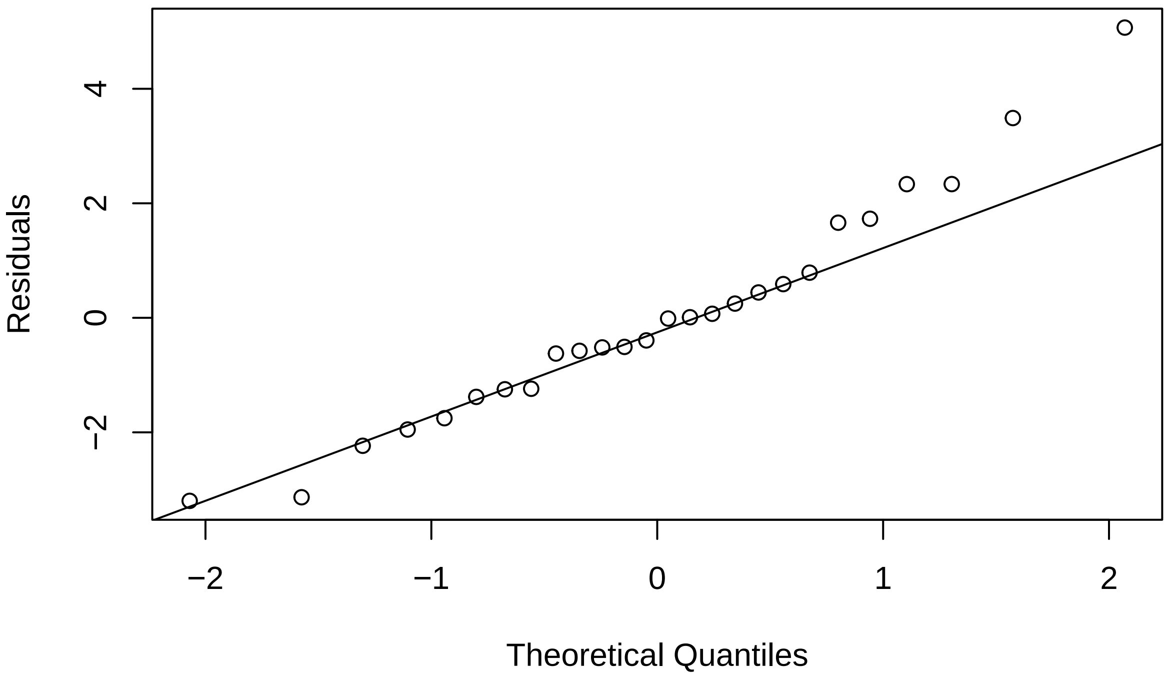

```
car::ncvTest(lmod1) # Null hypothesis = constant error variance
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.206952, Df = 1, p = 0.27194
```

By using the graph we find that there is no clear pattern of variance change along observations, so we then use a heteroscedasticity test. Thus, it is homoscedasticity (constant symmetrical variation).

1ii. Check Normality.

```
qqnorm(residuals(lmod1), ylab = 'Residuals', main = '')
qqline(residuals(lmod1))
```




```
shapiro.test(residuals(lmod1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmod1)
## W = 0.96313, p-value = 0.457
```

Since p-value is 0.457 which is greater than 0.05 and 0.1, we say it fails to reject the null hypothesis that the random errors are normally distributed. Thus, we conclude that the random errors follow a normal distribution.

1iii. Uncorrelated Errors

```
dwtest(mpg ~ factor(cyl)+disp+hp+drat+wt+qsec+factor(vs)+factor(am)+gear+carb, data=train)
```

```
##
##  Durbin-Watson test
##
## data:  mpg ~ factor(cyl) + disp + hp + drat + wt + qsec + factor(vs) +     factor(am) + gear + carb
## DW = 1.6514, p-value = 0.05224
## alternative hypothesis: true autocorrelation is greater than 0
```

Since the p-value is 0.05224 which is greater than 0.05, we fail to reject the hypothesis of uncorrelated errors.

2. Check Unusual Observations 2i. High Leverage Points

```
lev=hatvalues(lmod1)
n<-length(lev)
p<-dim(model.matrix(lmod1))[2]
dat=data.frame(index=seq(n),leverage=lev)
high.lev<-dat[which(dat$lev>2*(p)/n),"index"];high.lev
```

```
## integer(0)
```

2ii. Outliers

```
r=rstandard(lmod1)
r.a<- abs(r)
outliersm<-which(abs(r)>=3); outliersm
```

```
## named integer(0)
```

2iii. Influential Observations.

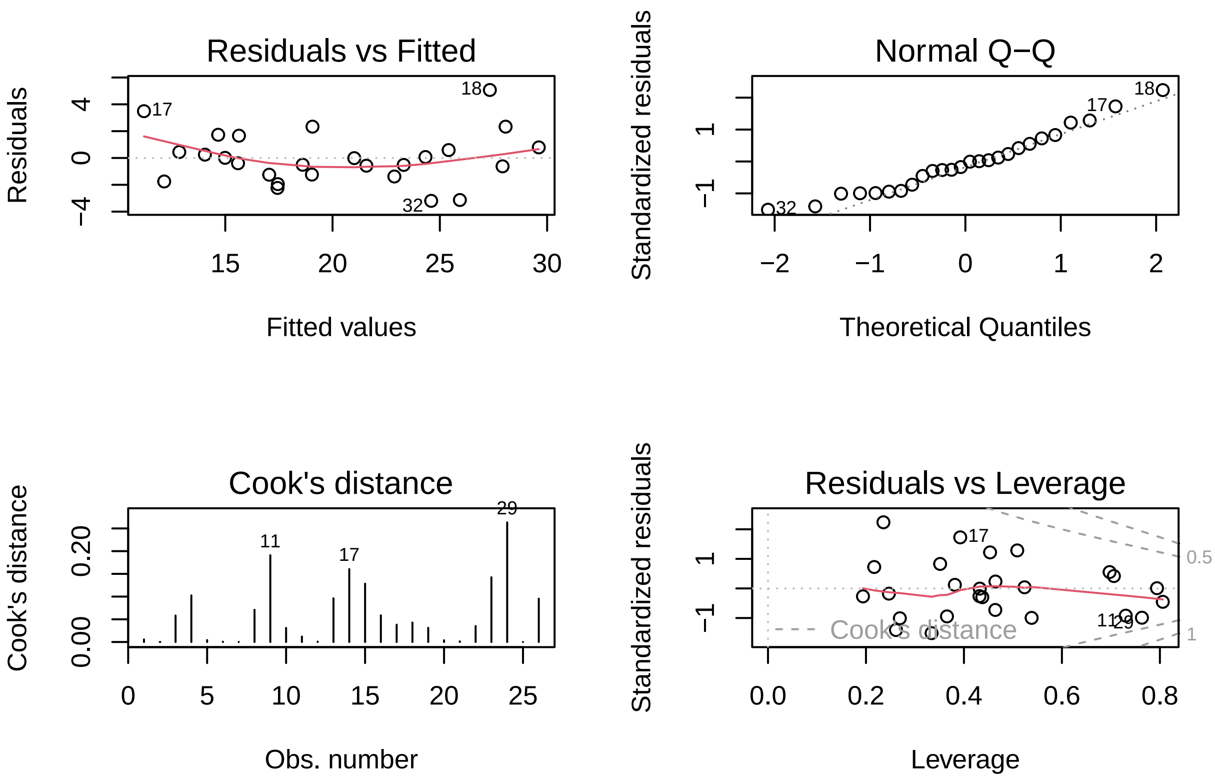
```
d=cooks.distance(lmod1)
dat3=data.frame(index=seq(length(r)),distance=d)
influ<-dat3[which(dat3$distance>4/n),"index"];influ
```

```
## [1]  9 14 24
```

Therefore, there are three influential observations. There is no outlier or high leverage point.

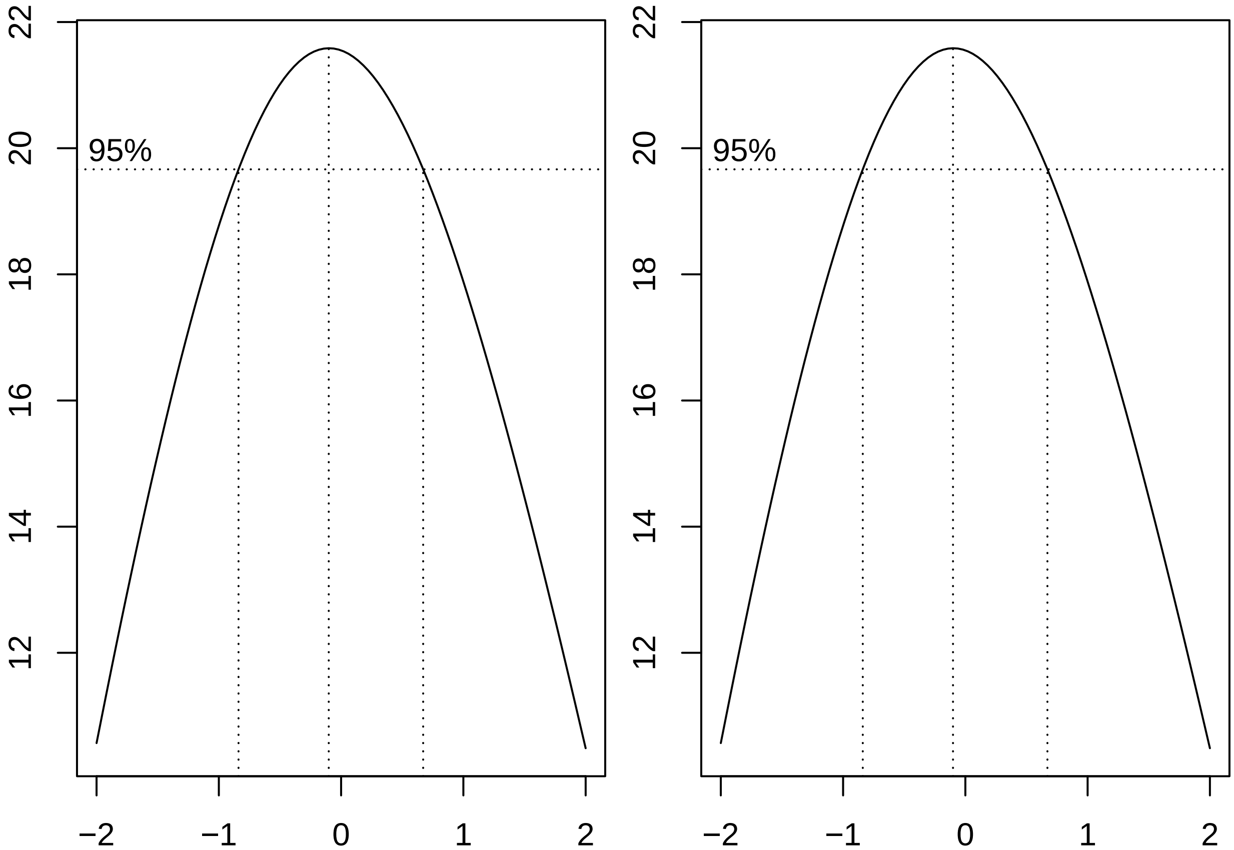
Diagnostic Summary(Unusual observations in a single plot)

```
par(mfrow = c(2,2))
plot(lmod1,c(1,2,4,5))
```



Now, we check whether the response needs a Box-Cox transformation.

```
par(mfrow=c(1,2),mar=c(2,2,0.8,0.5))
boxcox(lmod1,plotit=TRUE)
bc = boxcox(lmod1,plotit=TRUE)
```




```
lambda <- bc$x[which.max(bc$y)]
lambda1 <- round(lambda, 1)
lambda1
```

```
## [1] -0.1
```

Since 1 is not in the confidence interval, we need to do a Box-Cox transformation. Take log transformation on the response

```
fit <- lm(log(mpg) ~ factor(cyl)+disp+hp+drat+wt+qsec+factor(vs)+factor(am)+gear+carb, data=train)
summary(fit)
```

```
##
## Call:
## lm(formula = log(mpg) ~ factor(cyl) + disp + hp + drat + wt +
##     qsec + factor(vs) + factor(am) + gear + carb, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15498 -0.05515 -0.01540  0.04231  0.20882
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1572565  0.7824988   4.035  0.00123 **
## factor(cyl)6 -0.0681248  0.1119206  -0.609  0.55248
## factor(cyl)8 -0.0323755  0.2100929  -0.154  0.87973
## disp         -0.0002144  0.0010278  -0.209  0.83776
## hp           -0.0014273  0.0013600  -1.050  0.31171
```

```
## drat         -0.0169624  0.0837382  -0.203  0.84239
## wt           -0.1075789  0.0974874  -1.104  0.28841
## qsec          0.0130669  0.0372894   0.350  0.73124
## factor(vs)1   0.0393093  0.1127298   0.349  0.73250
## factor(am)1   0.0631129  0.1036733   0.609  0.55243
## gear          0.0520620  0.0679768   0.766  0.45647
## carb          0.0056894  0.0465830   0.122  0.90453
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1167 on 14 degrees of freedom
## Multiple R-squared:  0.9069, Adjusted R-squared:  0.8338
## F-statistic:  12.4 on 11 and 14 DF,  p-value: 2.141e-05
```

AIC and BIC:

```
require(leaps)
```

```
## Loading required package: leaps
```

```
models<- regsubsets(log(mpg) ~ cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb, data=train, nvmax= NULL)
rs<- summary(models)
rs$which
```

```
##    (Intercept)   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
## 1         TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2         TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 3         TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 4         TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
## 5         TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
## 6         TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
## 7         TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
## 8         TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 9         TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 10        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

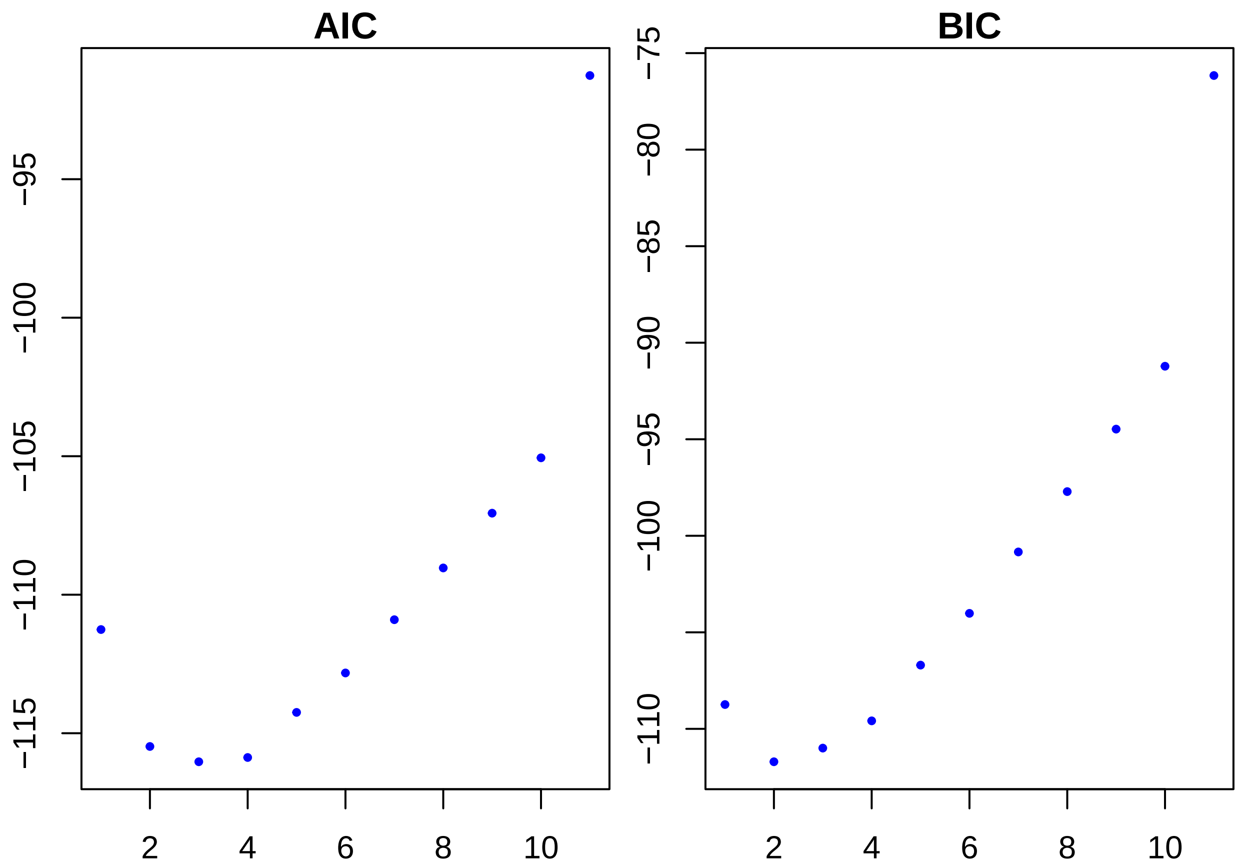
```
n <- dim(train)[1]
AIC <- n*log(rs$rss/n)+2*seq(2,12,1)
```

```
## Warning in n * log(rs$rss/n) + 2 * seq(2, 12, 1): longer object length is not a
## multiple of shorter object length
```

```
BIC <- n*log(rs$rss/n)+log(n)*seq(2,12,1)
```

```
## Warning in n * log(rs$rss/n) + log(n) * seq(2, 12, 1): longer object length is
## not a multiple of shorter object length
```

```
par(mar = c(2,2,1.2,0.5),mfrow=c(1,2))
plot(AIC~I(1:11),main="AIC",xlab = "# predictors", pch = 20, col = "blue",cex=0.7)
plot(BIC~I(1:11),main="BIC",xlab = "# predictors", pch = 20, col = "blue",cex=0.7)
```




```r
which.min(AIC)
```

```
## [1] 3
```

```r
which.min(BIC)
```

```
## [1] 2
```

Thus, we attain minimum AIC at 3 and BIC at 2 predictors.

```r
### best model according to AIC
AICnames<-names(which(rs$which[which.min(AIC),])=="TRUE" )[-1]
train.formulaAIC <- as.formula(paste("log(mpg) ~", paste(AICnames, collapse = " + ")))
AIC.model<-lm(train.formulaAIC, data =train)
summary(AIC.model)
```

```
##
## Call:
## lm(formula = train.formulaAIC, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.146383 -0.062130 -0.005557  0.056563  0.207179
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.8633348  0.0782008  49.403  < 2e-16 ***
## cyl         -0.0579078  0.0236393  -2.450 0.022725 *
## hp          -0.0007432  0.0004935  -1.506 0.146258
```

```
## wt          -0.1400308  0.0331313  -4.227 0.000347 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1001 on 22 degrees of freedom
## Multiple R-squared:  0.8924, Adjusted R-squared:  0.8777
## F-statistic:  60.8 on 3 and 22 DF,  p-value: 8.243e-11
```

```
### best model according to BIC
BICnames<-names(which(rs$which[which.min(BIC),])=="TRUE" )[-1]
train.formulaBIC <- as.formula(paste("log(mpg) ~", paste(BICnames, collapse = " + ")))
BIC.model<-lm(train.formulaBIC, data =train)
summary(BIC.model)
```

```
##
## Call:
## lm(formula = train.formulaBIC, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.161424 -0.047284 -0.008892  0.061733  0.211381
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.90510    0.07511  51.992  < 2e-16 ***
## cyl         -0.08216    0.01778  -4.622 0.000119 ***
## wt          -0.14076    0.03403  -4.136 0.000401 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1028 on 23 degrees of freedom
## Multiple R-squared:  0.8813, Adjusted R-squared:  0.871
## F-statistic: 85.36 on 2 and 23 DF,  p-value: 2.276e-11
```

Ridge Regression

```
require(glmnet)
```

```
## Loading required package: glmnet
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-6
```

```
library(glmnet)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

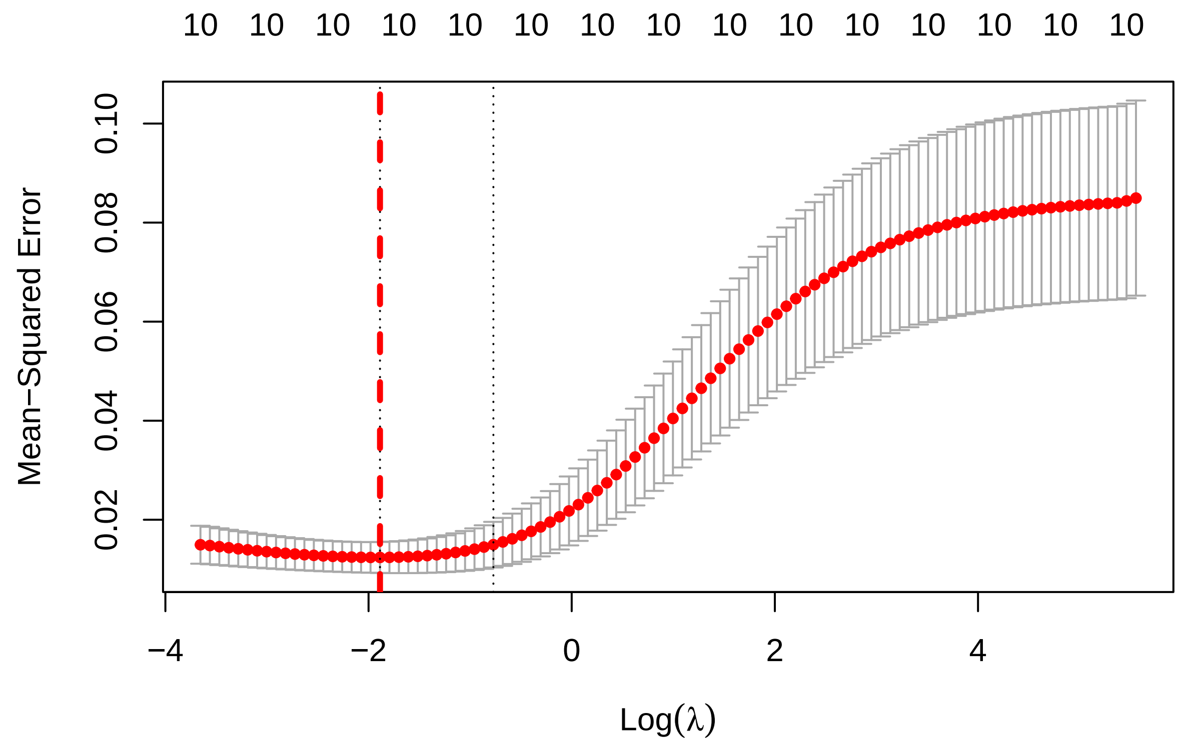
```
##
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
Train = na.omit(train)
x = scale(model.matrix(log(mpg)~ cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb, train)[,-1])
y = log(Train$mpg)
grid = 10^seq(10, -2, length = 100)
ridge_mod = glmnet(x, y, alpha = 0, lambda = grid)
```

```
set.seed(1) #we set a random seed first so our results will be reproducible.
cv.out.ridge=cv.glmnet(x, y, alpha = 0)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
plot(cv.out.ridge)
abline(v = log(cv.out.ridge$lambda.min), col="red", lwd=3, lty=2)
```




```
bestlam = cv.out.ridge$lambda.min
bestlam
```

```
## [1] 0.151475
```

```
out = glmnet(x,y,alpha=0)
predict(out,type="coefficients",s=bestlam)[1:11,]
```

```
## (Intercept)         cyl        disp          hp        drat          wt
##  2.95985680 -0.03693273 -0.04750091 -0.04108515  0.02034380 -0.05099436
##        qsec          vs          am        gear        carb
##  0.01331621  0.02179306  0.02860022  0.02619382 -0.02742018
```

```
best_model <- glmnet(x,y, alpha = 0, lambda = bestlam)
```

Now we have log full model, AIC model, BIC model, and ridge model.

4. point estimation:

```
test1_point <- predict(fit, newdata = test, interval = "confidence")
test1_point
```

```
##         fit      lwr      upr
## 5  2.764375 2.607370 2.921380
## 10 2.950512 2.715737 3.185287
## 15 2.530480 2.304874 2.756086
## 20 3.356639 3.208632 3.504645
## 25 2.713809 2.544550 2.883068
## 30 3.008960 2.788467 3.229453
```

```
test2_point <- predict(AIC.model, newdata = test, interval = "confidence")
test2_point
```

```
##         fit      lwr      upr
## 5  2.788304 2.711303 2.865305
## 10 2.942767 2.892647 2.992886
## 15 2.512552 2.404855 2.620248
## 20 3.326438 3.254696 3.398180
## 25 2.731591 2.662361 2.800822
## 30 2.997940 2.939512 3.056368
```

```
test3_point <- predict(BIC.model, newdata = test, interval = "confidence")
test3_point
```

```
##         fit      lwr      upr
## 5  2.763587 2.692815 2.834359
## 10 2.927913 2.881030 2.974795
## 15 2.508816 2.398593 2.619038
## 20 3.318154 3.245581 3.390727
## 25 2.706580 2.645045 2.768116
## 30 3.022220 2.973123 3.071317
```

```
test4_point <- predict(best_model, s = bestlam, newx = scale(as.matrix(test[,2:11])))
test4_point
```

```
##          s1
## 5  2.925668
## 10 2.973764
## 15 3.006795
## 20 2.987386
## 25 2.925207
## 30 2.940320
```

Since models are log value, we need to turn them back to y.

```
t1pt <- exp(test1_point)
t1pt
```

```
##         fit      lwr      upr
## 5  15.86912 13.56334 18.56690
## 10 19.11574 15.11575 24.17424
## 15 12.55953 10.02292 15.73812
## 20 28.69258 24.74521 33.26965
## 25 15.08664 12.73750 17.86901
## 30 20.26631 16.25607 25.26584
```

```
t2pt <- exp(test2_point)
t2pt
```

```
##         fit      lwr      upr
## 5  16.25343 15.04887 17.55440
## 10 18.96825 18.04100 19.94316
## 15 12.33637 11.07682 13.73914
## 20 27.83901 25.91174 29.90962
## 25 15.35731 14.33008 16.45817
## 30 20.04420 18.90663 21.25023
```

```
t3pt <- exp(test3_point)
t3pt
```

```
##         fit      lwr      upr
## 5  15.85662 14.77321 17.01949
## 10 18.68858 17.83263 19.58562
## 15 12.29037 11.00768 13.72252
## 20 27.60934 25.67662 29.68754
## 25 14.97797 14.08407 15.92859
## 30 20.53684 19.55290 21.57029
```

```
t4pt <- exp(test4_point)
t4pt
```

```
##          s1
## 5  18.64668
## 10 19.56543
## 15 20.22248
## 20 19.83377
## 25 18.63809
## 30 18.92190
```

```
SSR_t1 = sum((t1pt[,1]-test$mpg)^2)
SSR_t2 = sum((t2pt[,1]-test$mpg)^2)
SSR_t3 = sum((t3pt[,1]-test$mpg)^2)
SSR_t4 = sum((t4pt[,1]-test$mpg)^2)
MSE_t1 = SSR_t1/(dim(test)[1]-length(coef(fit)))
MSE_t2 = SSR_t2/(dim(test)[1]-length(coef(AIC.model)))
MSE_t3 = SSR_t3/(dim(test)[1]-length(coef(BIC.model)))
MSE_t4 = SSR_t4/(dim(test)[1]-length(coef(best_model)))
```

```
abs(MSE_t1)
```

```
## [1] 9.507035
```

```
abs(MSE_t2)
```

```
## [1] 30.70467
```

```r
abs(MSE_t3)
```

```
## [1] 23.33937
```

```r
abs(MSE_t4)
```

```
## [1] 59.07952
```

Full model has the least MSE, so full model is the best model.