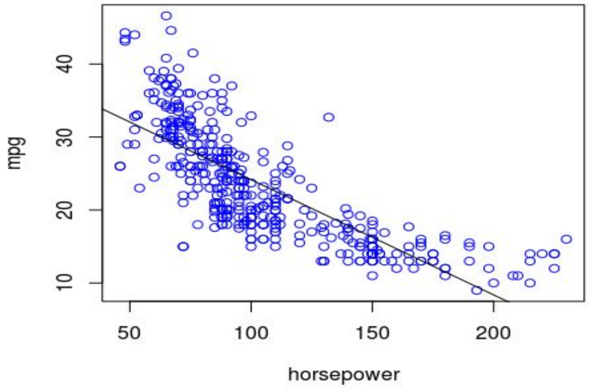
Q1. In hand written pages.
Q2.

Q3
(a)

```
Call:
lm(formula = mpg ~ horsepower)

Residuals:
     Min       1Q   Median       3Q      Max
-13.5710  -3.2592  -0.3435   2.7630  16.9240

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

(i) Yes, there is a relationship between mpg and horsepower since both t-test and F-test are significant.
(ii) The relationship is very strong, t-test on horsepower have a three star significant code. And the p-value is almost zero.
(iii) It is negative base on the sign of estimate of horsepower and the plot.
(iv) Predict mpg is 24.47. And the two interval are shown below:

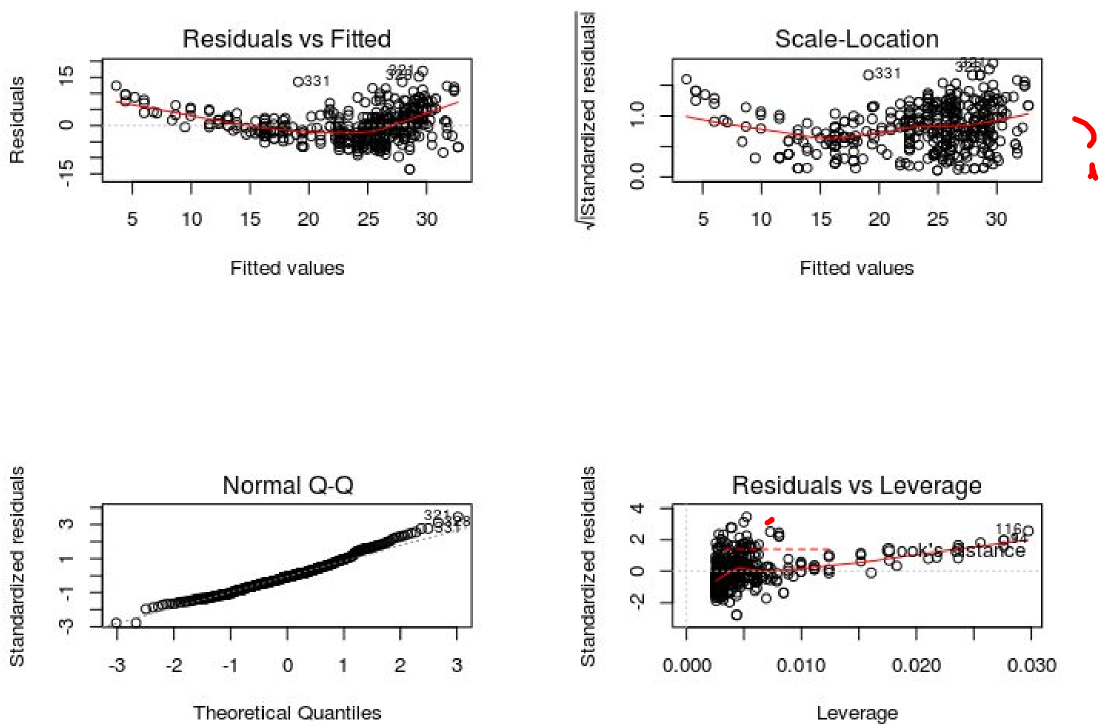
```
> predict(autofit,new.df)
       1
24.46708
> predict(autofit,new.df,interval = "confidence")
       fit      lwr      upr
1 24.46708 23.97308 24.96108
> predict(autofit,new.df,interval = "prediction")
       fit     lwr      upr
1 24.46708 14.8094 34.12476
>
```



(b)

Q3

(c)In the first plot, the residual vs fitted plot is not quit a straight line, so they may violate the assumption of linearity. Normality plot is good just few points.
And left-bottom plot doesn’t seem has extreme leverage.



Q4

(a)

```
> summary(carfit)

Call:
lm(formula = Sales ~ Price + Urban + US)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.9206 -1.6220 -0.0564  1.5786  7.0581 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
Price       -0.054459   0.005242 -10.389  < 2e-16 ***
UrbanYes    -0.021916   0.271650  -0.081    0.936    
USYes        1.200573   0.259042   4.635 4.86e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.472 on 396 degrees of freedom
Multiple R-squared:  0.2393,	Adjusted R-squared:  0.2335 
F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

(b)

For the coefficient Price, when the price increase one unit , the sales will decrease 0.05 unit. And its t-test show that price is significant for predicting sales.

Urban, this coefficient take one for YES, and zero for no. since the t-test is not significant, there is no relation between urban and sales.

US, is significant, so there are some relation in US and sales, if the location is US, fix other coefficient, sales will increase 1.2 as answer YES in US.

(c)

$$Sales = 13.04 - 0.054 * Price - 0.022 * UrbanYes + 1.2 * USYes$$

(d)

Intercept, price and USYes. Since they are significant.

(e)

```
Call:
lm(formula = Sales ~ Price + Urban)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.5324 -1.8441 -0.1443  1.6662  7.5000 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.621458   0.655230  20.789   <2e-16 ***
Price       -0.053104   0.005367  -9.895   <2e-16 ***
UrbanYes     0.034095   0.278293   0.123    0.903    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.535 on 397 degrees of freedom
Multiple R-squared:  0.198,	Adjusted R-squared:  0.194 
F-statistic: 49.01 on 2 and 397 DF,  p-value: < 2.2e-16
```

R^2

(f)

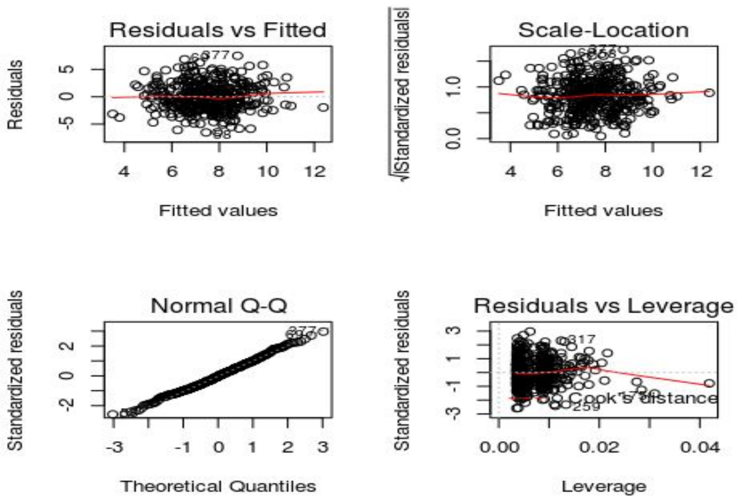
In part a the F-test is 41.52, and in part e is 49.01. So both model are similar. The relation in the overall model is statistically significant. However in both (a) and (e) they have one predictor UrbanYes not statistically significant.

(g)

```
> #CI for the coefficient
> confint(carfit2,level=0.95)
                  2.5 %      97.5 %
(Intercept) 12.33330469 14.90961133
Price       -0.06365522 -0.04255265
UrbanYes    -0.51301769  0.58120758
```

(h)

The Diagnostic Plots show there are no outliers, but the leverage indicate there might have some high leverage points.



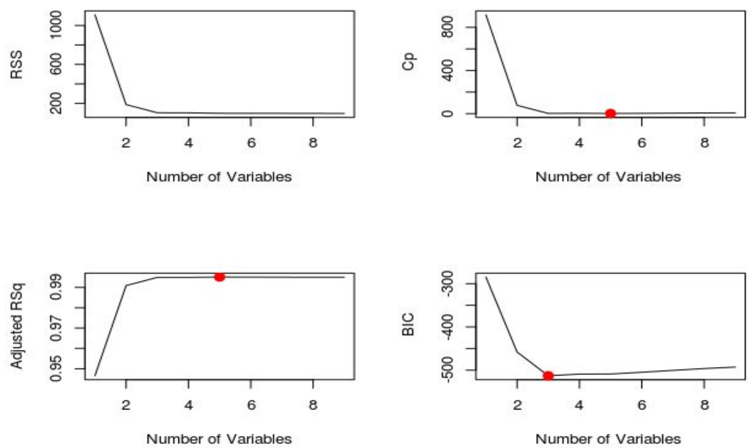
Q5.

In the hand written pages.

Q6

(a)-(b)

```
> #q6
> set.seed(1)
> n=100
> X = rnorm(n)
> error = rnorm(n)
> beta0 = 1
> beta1 =2
> beta2=3
> beta3=4
> Y = beta0 + beta1*X + beta2*X^2 + beta3*X^3 + error
> Y
  [1]  -0.67933427   1.53535052  -1.82134821  28.22280631   1.47326273   0.93660215   3.86755421   6.63210596   4.29386228   2.23726090
 [11]  24.06486920   2.01096498   1.38857465 -32.81689620  12.53315319   0.52301905   0.64839600   8.64421783   7.37518374   3.90654909
 [21]   7.96993242   7.65637047   0.95288867 -22.77729726   4.24452106   1.60915281   0.67253528  -8.21541312  -0.38935112   2.32765431
 [31]  19.34809474   0.23288207   2.99075880  -0.61794227  -6.20391590  -1.13565975   0.13164520   0.36281286  11.50249183   5.99477424
 [41]  -1.18001588   1.79738106   3.54045015   3.26939937  -1.37721812  -1.08070622   4.40893144   6.14210396  -0.47879993   6.18685724
 [51]   2.97424422  -0.03588604   1.87203330  -4.12353968  20.31044089  46.71994372   1.47205906  -2.99221837   2.46843319   2.64404721
 [61]  78.93953450   0.68725054   6.17773043   1.94486716  -1.09092447   3.71753080 -16.61264041  21.54134244   1.24696442  60.73449291
 [71]   5.36739392  -0.23333866   4.70858273  -1.58787720  -5.00730201   1.90201200   1.14213787   3.07745938   2.19429841   1.25195482
 [81]  -1.13410227   1.75847750  14.27998151 -10.69696930   4.60534185   1.98735211  12.78731285  -0.20944822   1.92321053   0.89833457
 [91]  -0.01787538  15.24343033  12.87874712   6.07495457  26.50254843   2.70149320  -3.54475002  -0.93005486  -3.88431122  -0.07992462
>
```



(c)

For Cp, the best model is when number of variables equal 5

For Adj Rsq, best model is when number of variables equal 5

```
> coef(regfit.full,5)
 (Intercept)            X          X.2          X.3          X.6          X.8
 1.113909391  2.053217562  2.778847934  3.973332603 -0.013122156  0.003936642
>
```

For BIC, best model is when number of variables equal 3

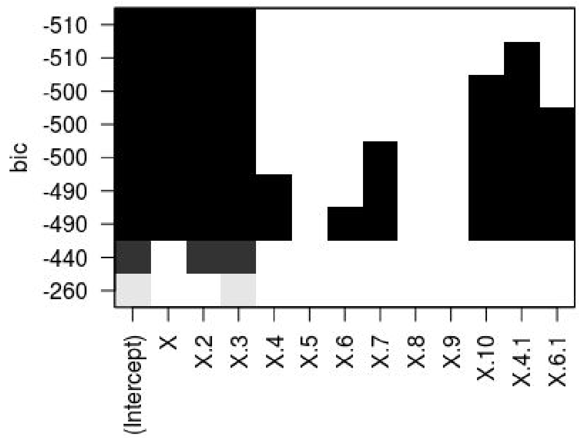
```
> coef(regfit.full,3)
(Intercept)           X         X.2         X.3
   1.061507    1.975280    2.876209    4.017639
>
```

(d)

```
> coef(regfit.full,9)
 (Intercept)            X          X.2          X.5          X.6          X.7          X.9         X.10        X.4.1        X.6.1
 0.994781947  3.257911484  2.892595851  2.987659650  0.000000000 -0.801105343  0.070505356 -0.001311704  0.083258244  0.019696886
> coef(regfit.fwd,9)
(Intercept)           X         X.2         X.3         X.4         X.6         X.7         X.9        X.10       X.4.1
 1.03322759  2.31590351  1.96068406  3.20655021  1.54794750 -0.53253104  0.25070294 -0.05446710  0.01507367  0.00000000
> coef(regfit.bwd,9)
(Intercept)           X         X.2         X.4         X.6         X.7         X.9        X.10       X.4.1       X.6.1
 1.37157251  4.54498602 -0.22284275  4.70547316 -1.74326168  1.02526025 -0.21350935  0.05333355  0.00000000  0.00000000
>
```

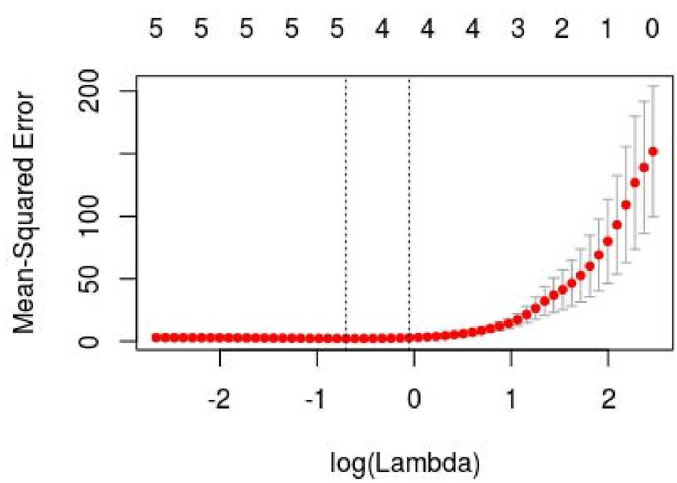
As in the output, if use fwd, the best subset contain 8 variables

And if use bwd, best subset will be 7 variables.



(e)

The plot of c-v error as function of lambda.



The best lambda value is 0.496.

The coefficients estimate are:

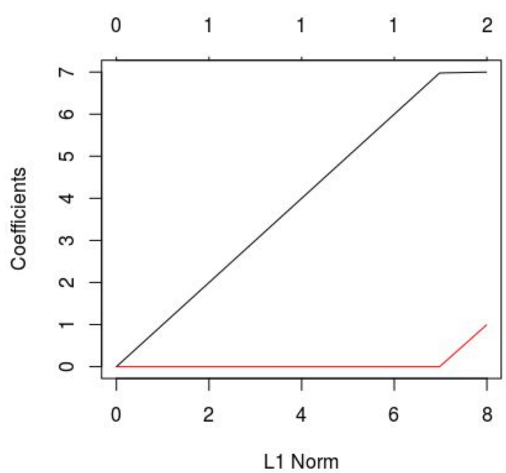
```
> predict(out,type="coefficients",s=bestlam)[1:13,]
(Intercept)           X         X.2         X.3         X.4       X.4.1         X.5         X.6       X.6.1         X.7         X.8
   1.189041    1.474105    2.800501    4.007895    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
        X.9        X.10
   0.000000    0.000000
> 
```

The best lambda yield the model has only X X^2 and X^3 three variables, and other terms are shrink to zero.

(f)
Best subset selection output:

```
> #best subset selection by cv
> reg.best=regsubsets(newY~.,data=newYX)
> coef(reg.best,2)
(Intercept)         X.7       error
          1           7           1
>
```

Lasso output:



Explain: the best subset selection obtain a model y=1+7*X^7+1(error).
And in lasso plot, as lambda is close to zero lasso regression close to OLS, as lambda increase Some coefficients shrink to zero which performs variable selection.

Q7.
(a)
Split test and training data base on column Apps, if Apps is odd set it as test data, otherwise set it as train data.

(b)
The plot output is shown as below,
The mean square error of the test data is 1248850.

(c)
Fit a ridge regression on training data, Value of the best lambda is 441.7765.
And the error is 1234065

(d)
Fit lasso on training data, and the error is 1221711,   best lambda is 11.46

(e)
Overall these three method are not significantly different in terms of the error we calculate.
And in lasso coefficient output, it penalize terms like F.undergrad, Books, Personal.
These predictors are also not significant in t-test in least square method.

```
> lasso.coef
 [1] -683.55052371 -154.50629405    1.61964013   -0.67304513   47.95536451  -14.81943119    0.00000000    0.11255607
 [9]   -0.09705219    0.12096751    0.15636387    0.00000000   -3.06767999   -1.45410828   14.60236742    3.54188855
[17]    0.05339605    6.36637492            NA            NA
```

The best lambda lasso coefficient is shown in the table.

```
Call:
lm(formula = Apps ~ ., data = data.train)

Residuals:
    Min      1Q  Median      3Q     Max 
-5366.0  -386.9   -20.0   283.6  7484.7 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -584.46385  531.92002  -1.099 0.272555    
PrivateYes  -189.43910  190.92499  -0.992 0.321720    
Accept         1.68262    0.04734  35.542  < 2e-16 ***
Enroll        -1.01894    0.26850  -3.795 0.000172 ***
Top10perc     59.65232    8.26679   7.216 2.91e-12 ***
Top25perc    -23.52803    6.48771  -3.627 0.000326 ***
F.Undergrad    0.02951    0.04876   0.605 0.545361    
P.Undergrad    0.13375    0.04066   3.290 0.001096 ** 
Outstate      -0.11831    0.02710  -4.365 1.64e-05 ***
Room.Board     0.14238    0.06719   2.119 0.034734 *  
Books          0.17842    0.34666   0.515 0.607080    
Personal      -0.01054    0.08079  -0.130 0.896271    
PhD           -4.47474    6.66766  -0.671 0.502556    
Terminal      -1.56018    7.28034  -0.214 0.830427    
S.F.Ratio     22.52834   19.61340   1.149 0.251431    
perc.alumni    7.37181    5.84233   1.262 0.207794    
Expend         0.05627    0.01687   3.336 0.000933 ***
Grad.Rate      8.23082    4.28739   1.920 0.055631 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1005 on 382 degrees of freedom
Multiple R-squared:  0.9454,	Adjusted R-squared:  0.943 
F-statistic: 389.1 on 17 and 382 DF,  p-value: < 2.2e-16

> lm.pred = predict(lm.fit, data.test)
> mean((data.test[, "Apps"] - lm.pred)^2)
[1] 1248850
```

```
> ridge.pred=predict(ridge.mod,s=bestlamdba,newx =xtest )
> mean((ridge.pred-ytest)^2)
[1] 1234065
> bestlamdba
[1] 441.7765
```

```
> lasso.pred=predict(lasso.mod,s=bestlamdba,newx=xtest)
> bestlamdba
[1] 11.46421
> mean((lasso.pred-ytest)^2)
[1] 1221711
```

Q8.

(a)

P(x) = $e^{-6+0.05*40+3.5}/1+e^{-6+0.05*40+3.5}$ = **37.75%**

(b)

0.5 = $e^{-6+0.05*HOUR+3.5}/1+e^{-6+0.05*HOUR+3.5}$

Solve this eq get hour = 50 hours

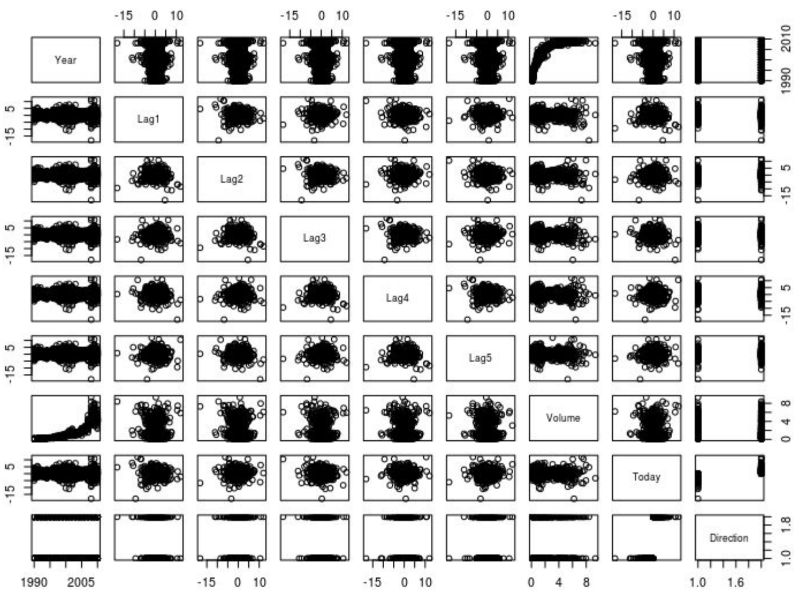
Q9

(a)

```
> summary(Weekly)
      Year           Lag1               Lag2               Lag3               Lag4               Lag5              Volume
 Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747
 1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580   1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
 Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410   Median :  0.2380   Median :  0.2340   Median :1.00268
 Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472   Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
 3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
 Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
     Today          Direction
 Min.   :-18.1950   Down:484
 1st Qu.: -1.1540   Up  :605
 Median :  0.2410
 Mean   :  0.1499
 3rd Qu.:  1.4050
 Max.   : 12.0260
```

.

```
> cor(Weekly[,-9])
              Year         Lag1        Lag2        Lag3         Lag4         Lag5      Volume         Today
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923 -0.030519101  0.84194162 -0.032459894
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876 -0.008183096 -0.06495131 -0.075031842
Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535 -0.072499482 -0.08551314  0.059166717
Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865  0.060657175 -0.06928771 -0.071243639
Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000 -0.075675027 -0.06107462 -0.007825873
Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027  1.000000000 -0.05851741  0.011012698
Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617 -0.058517414  1.00000000 -0.033077783
Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873  0.011012698 -0.03307778  1.000000000
> 
```



Summary of the plot: in cor(Weekly) output, year and volume have value of 0.8419, may indicate some relation. But no others variables have strong relation in the plot.

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Weekly)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b)

Base on the summary output, Lag2 and intercept are significant.

(c)

```
> table(glm.pred,Direction)
        Direction
glm.pred Down  Up
    Down   54  48
    Up    430 557
```

False positive rate: the fraction of down example classified as up = 430/(54+430) = 88.84%

False negative: up example classified as down = 48/557+48 = 7.93%

(d)

```
> table(glm.pred,Direction.test)
        Direction.test
glm.pred Down Up
    Down    9  5
    Up     34 56
```

Overall correct prediction: 9+56/(9+5+34+56)= 62.5%.

(e)

```
> table(lda.class,Direction.test)
         Direction.test
lda.class Down Up
     Down    9  5
     Up     34 56
> 
```

Overall correct prediction rate = 62.5%.

(f)

```
> table(qda.class,Direction.test)
         Direction.test
qda.class Down Up
     Down    0  0
     Up     43 61
```

Overall correct prediction rate = 58.65%.

(g)

Logistic and Linear discriminant analysis provide and correct rate 62.5%, better than qda.

(h)

Fit a model Direction~Lag1+Lag2+Lag3*Lag4 in logistic method, the matrix is

```
> table(glm.pred,Direction.test)
        Direction.test
glm.pred Down Up
    Down    9  7
    Up     34 54
> 
```

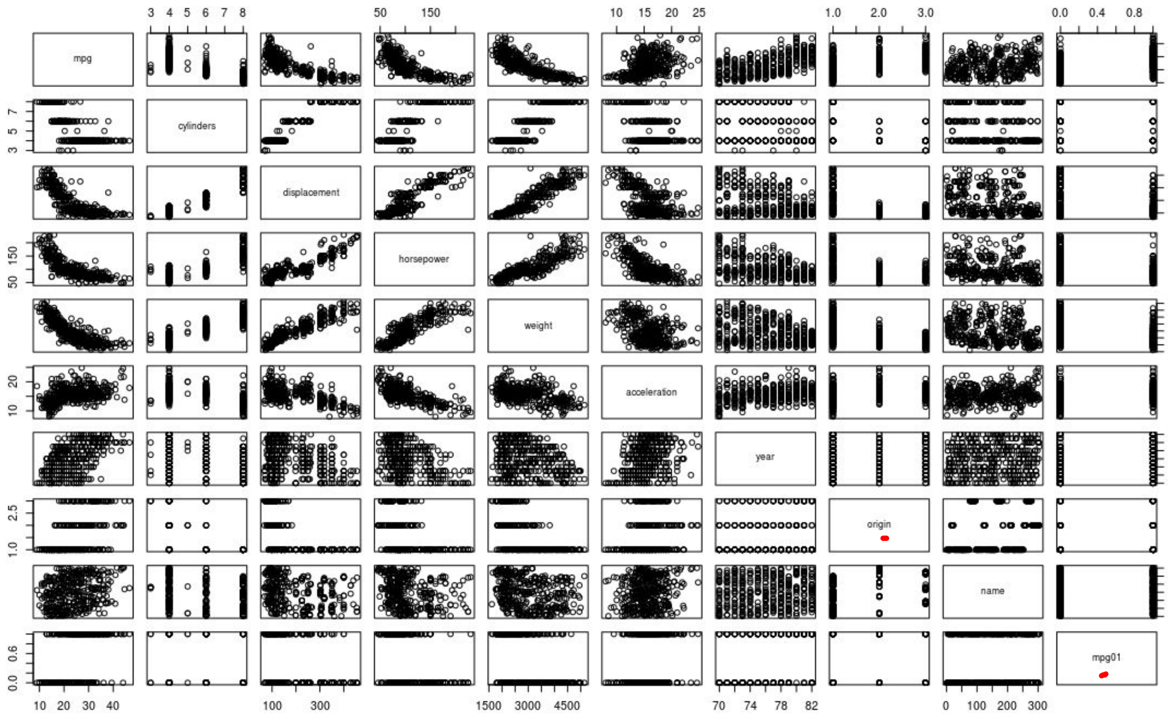
correct rate = 60.57%

Fit a model Direction~Lag1+Lag2+Lag3*Lag4+Lag5+Lag6,

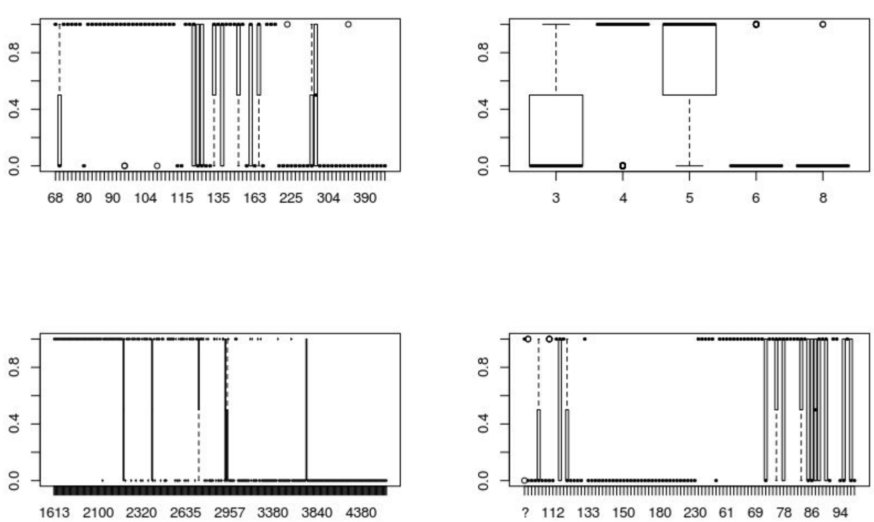
Produce same as last one, correct rate=60.57%

Fit a model Direction~Lag6+Lag3*Lag2*Lag4, correct rate =56.73%.

Q10
(a)-(b)



Boxplot of some variables seem useful to predict mpg01



Therefore, mpg, horsepower, weight, cylinders and displacement may be useful to predict mpg0.

(c) by observing the data,one of the easy way to separate data is set year=even number as train data, and the rest as test data since both cases odd and even have fair amount of data.

(d)
LDA
If put horsepower as one of the predictor, the R will output error that “variables appear to be constant within groups “. So omit the horsepower and run the lda.
Test error in LDA is 0.0934

```
Call:
lda(mpg01 ~ mpg + weight + displacement + cylinders, data = newAuto,
    subset = train)

Prior probabilities of groups:
        0         1 
0.4647887 0.5352113 

Group means:
       mpg   weight displacement cylinders
0 17.09293 3572.707     267.7172  6.747475
1 30.65526 2314.588     111.7325  4.070175

Coefficients of linear discriminants:
                       LD1
mpg           0.1463274641
weight       -0.0000942717
displacement  0.0042922656
cylinders    -0.7275368570
```

(e)

```
Call:
qda(mpg01 ~ mpg + weight + displacement + cylinders, data = newAuto,
    subset = train)

Prior probabilities of groups:
        0         1 
0.4647887 0.5352113 

Group means:
       mpg   weight displacement cylinders
0 17.09293 3572.707     267.7172  6.747475
1 30.65526 2314.588     111.7325  4.070175
```

The error in QDA is 0.0909

(f)
Logistic regression, predictors mpg and cylinders are significant for predicting mpg01.
And the error is 0.2141.

```
Call:
glm(formula = mpg01 ~ mpg + weight + displacement + cylinders,
    data = newAuto, subset = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.64926  -0.17857   0.06497   0.19424   0.57294

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.882e-01  2.146e-01   2.275   0.0239 *
mpg           3.210e-02  3.844e-03   8.350 9.53e-15 ***
weight       -2.068e-05  6.146e-05  -0.336   0.7369
displacement  9.415e-04  6.305e-04   1.493   0.1369
cylinders    -1.596e-01  3.455e-02  -4.619 6.76e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```