Exercise 7

Tobias Famos

Preliminaries Task 1

Load library boot and load the Cars data

```
library(boot)
cars <- read.table("/home/tobias/unibe/statistical methods in R/Exercise 7/Cars.txt", header = T)
cars <- subset(cars, select = -c(name))
cars <- na.omit(cars)
summary(cars)
```

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

Task 1

Build the multiple linear regression model from exercise 5 and 6

```
model.mulitple_linear <- glm(mpg~., data=cars)
summary(model.mulitple_linear)
```

```
## 
## Call:
## glm(formula = mpg ~ ., data = cars)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -9.5903  -2.1565  -0.1169   1.8690  13.0604  
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780    
## displacement   0.019896   0.007515   2.647  0.00844 ** 
## horsepower    -0.016951   0.013787  -1.230  0.21963    
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548    
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 11.07347)
## 
##     Null deviance: 23819.0  on 391  degrees of freedom
## Residual deviance:  4252.2  on 384  degrees of freedom
## AIC: 2064.9
## 
## Number of Fisher Scoring iterations: 2
```

We see that cylinders, horsepower, and acceleration are not significant. Thus we drop them and run a linear model again

```
cars_clean <- subset(cars, select=-c(cylinders, horsepower,acceleration ))
model.mulitple_linear_clean <- glm(mpg~., data=cars_clean)
summary(model.mulitple_linear_clean)
```

```
## 
## Call:
## glm(formula = mpg ~ ., data = cars_clean)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -9.8102  -2.1129  -0.0388   1.7725  13.2085  
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -1.861e+01  4.028e+00  -4.620 5.25e-06 ***
## displacement  5.588e-03  4.768e-03   1.172    0.242    
## weight       -6.575e-03  5.571e-04 -11.802  < 2e-16 ***
## year          7.714e-01  4.981e-02  15.486  < 2e-16 ***
## origin        1.226e+00  2.670e-01   4.593 5.92e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 11.19568)
## 
##     Null deviance: 23819.0  on 391  degrees of freedom
## Residual deviance:  4332.7  on 387  degrees of freedom
## AIC: 2066.3
## 
## Number of Fisher Scoring iterations: 2
```

From the output we can derive again that the displacement is not significant in this model, thus we drop it as well,

```
cars_clean2 <- subset(cars_clean, select=-c(displacement))
model.mulitple_linear_clean2 <- glm(mpg~., data=cars_clean2)
summary(model.mulitple_linear_clean2)
```

```
##
## Call:
## glm(formula = mpg ~ ., data = cars_clean2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -9.9440  -2.0948  -0.0389   1.7255  13.2722
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.805e+01  4.001e+00  -4.510 8.60e-06 ***
## weight      -5.994e-03  2.541e-04 -23.588  < 2e-16 ***
## year         7.571e-01  4.832e-02  15.668  < 2e-16 ***
## origin       1.150e+00  2.591e-01   4.439 1.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 11.20646)
##
##     Null deviance: 23819.0  on 391  degrees of freedom
## Residual deviance:  4348.1  on 388  degrees of freedom
## AIC: 2065.7
##
## Number of Fisher Scoring iterations: 2
```

Now we have arrived at multiple linear regression model with only significant Coefficients. As we have three linear models now that should get better with each tweaking we did, lets compare them with 10 fold cross validation.

```r
set.seed(123)
cv.initial <- cv.glm(cars, model.mulitple_linear, K=10)
cv.clean <- cv.glm(cars_clean, model.mulitple_linear_clean, K=10)
cv.clean_2 <- cv.glm(cars_clean2, model.mulitple_linear_clean2, K=10)

# Print the MSE for each
cv.initial$delta[1]
```

```
## [1] 11.36738
```

```r
cv.clean$delta[1]
```

```
## [1] 11.24438
```

```r
cv.clean_2$delta[1]
```

```
## [1] 11.34917
```

The mean squared error does not change, which makes sense in a linear model, as the omitted predictors are just set to 0 if they are left in.

```r
t.test(formual=model.mulitple_linear$formula, cars)
```

```
##
##  One Sample t-test
##
## data:  cars
## t = 23.48, df = 3135, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  389.3356 460.2846
## sample estimates:
## mean of x
##  424.8101
```

```
t.test(formual=model.mulitple_linear_clean$formula, cars)
```

```
##
##  One Sample t-test
##
## data:  cars
## t = 23.48, df = 3135, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  389.3356 460.2846
## sample estimates:
## mean of x
##  424.8101
```

```
t.test(formual=model.mulitple_linear_clean2$formula, cars)
```

```
##
##  One Sample t-test
##
## data:  cars
## t = 23.48, df = 3135, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  389.3356 460.2846
## sample estimates:
## mean of x
##  424.8101
```

Somehow my t test tells we can accept the alternative hypothesis for all the hypothesis and we have the same confidence interval. I think I have made a mistake here.

Preliminaries Task 2

Load the cancer data set. Import it as is so the Diagnostic gets converted to a factor.

```
cancer <- read.table("/home/tobias/unibe/statistical methods in R/Exercise 7/Cancer.txt", header = T, as
str(cancer)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ ID          : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 8450100
##  $ Diagnostic  : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Radius      : num  18 20.6 19.7 11.4 20.3 ...
##  $ Texture     : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ Perimeter   : num  122.8 132.9 130 77.6 135.1 ...
##  $ Area        : num  1001 1326 1203 386 1297 ...
##  $ Smooth      : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ Compact     : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ Concavity   : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ Concave     : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ Symmetry    : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ Fractal     : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ RadiusSE    : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ TextureSE   : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ PerimeterSE : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ AreaSE      : num  153.4 74.1 94 27.2 94.4 ...
##  $ SmoothSE    : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ CompactSE   : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ ConcavitySE : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ ConcaveSE   : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ SymmetrySE  : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ FractalSE   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ RadiusMax   : num  25.4 25 23.6 14.9 22.5 ...
##  $ TextureMax  : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ PerimeterMax: num  184.6 158.8 152.5 98.9 152.2 ...
##  $ AreaMax     : num  2019 1956 1709 568 1575 ...
##  $ SmoothMax   : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ CompactMax  : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ ConcavityMax: num  0.712 0.242 0.45 0.687 0.4 ...
##  $ ConcaveMax  : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ SymmetryMax : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ FractalMax  : num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

Check for NA

```
sum(is.na(cancer))
```

```
## [1] 0
```

Drop the ID as we don't need it for predcitions

```
cancer <- subset(cancer, select = -c(ID))
```

Task 2: Apply a general logistic regression to estimate the `Diagnostic`

Using the `glm`with `family=binomial` we can build a model using all the predictors.

```
model.logistic <- glm(cancer$Diagnostic ~ ., data = cancer, family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model.logistic)
```

```
## 
## Call:
## glm(formula = cancer$Diagnostic ~ ., family = binomial, data = cancer)
## 
## Deviance Residuals: 
##    Min      1Q  Median      3Q     Max  
##  -8.49   -8.49   -8.49    8.49    8.49  
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  -2.881e+06  2.816e+05 -10.233  < 2e-16 ***
## Radius        2.427e+06  2.693e+05   9.014  < 2e-16 ***
## Texture       1.958e+05  1.471e+04  13.313  < 2e-16 ***
## Perimeter     1.473e+06  2.464e+04  59.791  < 2e-16 ***
## Area         -1.301e+05  3.907e+03 -33.301  < 2e-16 ***
## Smooth       -1.525e+08  8.361e+06 -18.234  < 2e-16 ***
## Compact      -6.428e+06  3.213e+06  -2.001  0.04539 *
## Concavity     1.042e+06  1.408e+06   0.740  0.45959
## Concave      -1.716e+07  5.382e+06  -3.188  0.00143 **
## Symmetry      4.049e+07  7.772e+05  52.093  < 2e-16 ***
## Fractal      -4.233e+07  2.169e+06 -19.519  < 2e-16 ***
## RadiusSE      3.328e+07  1.169e+06  28.478  < 2e-16 ***
## TextureSE     6.368e+06  2.005e+05  31.763  < 2e-16 ***
## PerimeterSE   1.701e+06  4.720e+04  36.032  < 2e-16 ***
## AreaSE       -6.393e+05  1.835e+04 -34.840  < 2e-16 ***
## SmoothSE      7.492e+08  1.224e+07  61.213  < 2e-16 ***
## CompactSE    -1.773e+08  5.732e+06 -30.931  < 2e-16 ***
## ConcavitySE   1.529e+08  5.340e+06  28.624  < 2e-16 ***
## ConcaveSE    -1.260e+09  4.012e+07 -31.398  < 2e-16 ***
## SymmetrySE    2.890e+08  4.126e+06  70.055  < 2e-16 ***
## FractalSE     1.512e+09  6.597e+07  22.921  < 2e-16 ***
## RadiusMax    -6.130e+06  2.143e+05 -28.606  < 2e-16 ***
## TextureMax   -5.832e+05  2.437e+04 -23.935  < 2e-16 ***
## PerimeterMax -3.538e+05  1.219e+04 -29.023  < 2e-16 ***
## AreaMax       8.950e+04  2.741e+03  32.658  < 2e-16 ***
## SmoothMax    -2.161e+07  3.298e+06  -6.553 5.66e-11 ***
## CompactMax    8.986e+06  3.999e+05  22.470  < 2e-16 ***
## ConcavityMax -3.028e+07  1.523e+06 -19.875  < 2e-16 ***
## ConcaveMax    1.431e+08  5.471e+06  26.162  < 2e-16 ***
## SymmetryMax  -2.474e+07  3.392e+05 -72.923  < 2e-16 ***
## FractalMax   -3.698e+07  5.340e+06  -6.926 4.33e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:   751.44  on 568  degrees of freedom
## Residual deviance: 32006.76  on 538  degrees of freedom
## AIC: 32069
##
## Number of Fisher Scoring iterations: 25
```

Most important values of the model

Lets start with the **Coefficients**: Most of the Coefficients have a high `z-value` and thus a small probability of being bigger than `|z|`. There are a few exceptions:

- **Concavity** has a low `z-value` of `0.74` and thus is not significant. To simplify the model it can be omitted (although it is practically already omitted by setting a coefficient approximately equal 0)
- **Compact** and **Concave** have both a probability `>1%` and `<5%`. As we do not have a hughe dataset, we could also omit them if we want to be strict

The **AIC** score is quite high. This could be a warning sign, but as we only have one model there is nothing to compare it with.

Also the **Deviance Residual** is quite high.

Estimating error Rate

We use the resubstitution approach to get an optimistic view on the correctly and falsely classified instances

```
labels <- cancer$Diagnostic
without_label <- subset(cancer, select = -c(Diagnostic))
prediction <- predict(model.logistic, data = without_label, type = "response")
factor_prediction <- cut(prediction, labels = c("M", "B"), breaks = 2)
table(factor_prediction == labels)
```

```
##
## FALSE  TRUE
##   125   444
```

Calculate the Error rate

```
444 / length(labels)
```

```
## [1] 0.7803163
```

We arrive at an accuracy of **78.03163**. This isn't too much but also not too bad. With a few tweaks we may be able to get it higher