Analyzing the data, “Diabetes”

Arpan Dutta
Soumyajit Roy
Sourav Biswas
MStat. I : 2023

October 29, 2023

Contents

Chapter 1

Introduction.

The **"diabetes.csv"** dataset consists of data related to relative weight and results of different tests to diagonise diabetes of 144 persons.

The dataset consists of the following columns :

- **relwt** : Relative weight.
- **glufast** : Fasting Plasma Glucose (FPG). This test is the simplest and fastest way to measure blood glucose and diagonise diabetes. Fasting means one has nothing to eat or drink (except water) for 8 to 12 hours before the test. One will be diagonised with diabetes if blood glucose level is 126 mg/dL or greater on two separate tests.
- **glutest** : Test Plasma Glucose.
- **sspg** : Steady State Plasma Glucose. It's a period of time where the coefficients of variations for blood glucose, plasma insulin and GIR are less than 5%. This period is usually defined as being greater than 30 minutes, and at least 1 hour after the initiation of insulin infusion. The expected values for normal fasting blood glucose concentration are between 70 mg/dL and 100 mg/dL.
- **instest** : Plasma Insulin during Test. A blood test to measure insuline levels produced in our body.
- **group** : Clinical group. A Categorical Variate with 3 levels, 1 means Over Diabetic, 2 means chem. diabetic, 3 means normal.

Now we perform the analysis.

Chapter 2

R Packages used.

```
library(ggplot2)
library(tibble)
library(PerformanceAnalytics)

Loading required package:  xts
Loading required package:  zoo

Attaching package:  ’zoo’
The following objects are masked from ’package:base’:
      as.Date, as.Date.numeric

Attaching package:  ’PerformanceAnalytics’
The following object is masked from ’package:graphics’:
      legend

library(wesanderson)
library(gridExtra)
library(car)

Loading required package:  carData

library(glmnet)

Loading required package:  Matrix
Loaded glmnet 4.1-7

library(pls)


Attaching package:  ’pls’
The following object is masked from ’package:stats’:
      loadings
```

```r
library(lmtest)
library(nlme)
library(MASS)
library(lattice)
library(leaps)
library(splines)
library(plyr)
```

Chapter 3

The Dataset.

3.1 Importing the Datset.

We load the dataset in R and save it as a dataframe named "**dbts**".

```r
dbts=read.csv("E:/RFiles/diabetes.csv")
```

Now we convert the variable "**group**" into a factor covariate, having 3 levels. After that, for the sake of convenience, the levels are relabeled. From now, we shall denote 0 by 'Normal', 1 by 'Chem. Diabetic', 2 by 'Over Diabetic'. Our reference level will be the Normal group, i.e. our whole analysis will be with respect to a person who is not diabetic.

```r
#---Converting appropriate variables into factors---
dbts$group=as.factor(dbts$group)
#-changing levels---
levels(dbts$group)=c("2","1","0")
#--Releveling---
dbts$group=relevel(dbts$group,ref="0")
```

3.2 The structure of the dataset,

```r
str(dbts)

'data.frame': 144 obs. of  6 variables:
 $ relwt  : num  0.81 0.95 0.94 1.04 1 0.76 0.91 1.1 0.99 0.78 ...
 $ glufast: int  80 97 105 90 90 86 100 85 97 97 ...
 $ glutest: int  356 289 319 356 323 381 350 301 379 296 ...
 $ sspg   : int  124 117 143 199 240 157 221 186 142 131 ...
 $ instest: int  55 76 105 108 143 165 119 105 98 94 ...
 $ group  : Factor w/ 3 levels "0","2","1": 1 1 1 1 1 1 1 1 1 1 ...
```

As we can see, the dataset has 144 observations of **6** variables. We would like to infer the relationship of “**relwt**” with the other variables. We are showing first few rows of the dataset as follows,

```
head(dbts,5)

  relwt glufast glutest sspg instest group
1  0.81      80     356  124      55     0
2  0.95      97     289  117      76     0
3  0.94     105     319  143     105     0
4  1.04      90     356  199     108     0
5  1.00      90     323  240     143     0
```

3.3 Statistical summary of ’dbts’.

```
summary(dbts)

     relwt           glufast         glutest            sspg      
 Min.   :0.7100   Min.   : 70.0   Min.   : 269.0   Min.   : 10.0  
 1st Qu.:0.8875   1st Qu.: 90.0   1st Qu.: 352.0   1st Qu.:119.5  
 Median :0.9800   Median : 97.0   Median : 408.0   Median :156.5  
 Mean   :0.9790   Mean   :120.4   Mean   : 536.5   Mean   :187.3  
 3rd Qu.:1.0800   3rd Qu.:112.0   3rd Qu.: 557.2   3rd Qu.:221.2  
 Max.   :1.2000   Max.   :353.0   Max.   :1520.0   Max.   :748.0  
    instest       group 
 Min.   : 29.00   0:76  
 1st Qu.: 99.75   2:32  
 Median :158.50   1:36  
 Mean   :183.73         
 3rd Qu.:257.50         
 Max.   :480.00         
```

3.3.1 Groupwise Summary.

```
dlply(dbts,.(group),summary)

$`0`
     relwt           glufast           glutest            sspg      
 Min.   :0.7100   Min.   : 70.00   Min.   :269.0   Min.   : 73.0  
 1st Qu.:0.8400   1st Qu.: 86.00   1st Qu.:322.5   1st Qu.:129.5
 Median :0.9500   Median : 90.00   Median :353.0   Median :157.0
 Mean   :0.9372   Mean   : 91.18   Mean   :350.0   Mean   :172.6
 3rd Qu.:1.0100   3rd Qu.: 97.00   3rd Qu.:378.0   3rd Qu.:200.5
 Max.   :1.2000   Max.   :112.00   Max.   :426.0   Max.   :490.0
    instest       group
 Min.   : 29.00   0:76
 1st Qu.: 73.75   2: 0
 Median :105.00   1: 0
 Mean   :114.00
 3rd Qu.:142.25
 Max.   :273.00

$`2`
     relwt          glufast         glutest           sspg
 Min.   :0.8100   Min.   :120.0   Min.   : 538.0   Min.   : 10.0
 1st Qu.:0.9000   1st Qu.:148.2   1st Qu.: 843.2   1st Qu.: 43.5
 Median :0.9950   Median :199.0   Median : 969.5   Median : 85.0
 Mean   :0.9916   Mean   :213.7   Mean   :1027.4   Mean   :108.8
 3rd Qu.:1.0625   3rd Qu.:276.2   3rd Qu.:1285.8   3rd Qu.:132.8
 Max.   :1.2000   Max.   :353.0   Max.   :1520.0   Max.   :460.0
    instest      group
 Min.   :150.0   0: 0
 1st Qu.:274.2   2:32
 Median :322.0   1: 0
 Mean   :320.9
 3rd Qu.:375.0
 Max.   :480.0

$`1`
     relwt         glufast          glutest          sspg
 Min.   :0.830   Min.   : 75.00   Min.   :413.0   Min.   :109.0
 1st Qu.:0.975   1st Qu.: 92.00   1st Qu.:455.8   1st Qu.:160.2
 Median :1.065   Median : 99.50   Median :476.5   Median :251.5
 Mean   :1.056   Mean   : 99.31   Mean   :493.9   Mean   :288.0
 3rd Qu.:1.123   3rd Qu.:107.25   3rd Qu.:534.0   3rd Qu.:330.5
 Max.   :1.200   Max.   :114.00   Max.   :643.0   Max.   :748.0
    instest      group
 Min.   : 60.0   0: 0
 1st Qu.:164.5   2: 0
 Median :223.0   1:36
 Mean   :209.0
 3rd Qu.:258.8
 Max.   :300.0

attr(,"split_type")
```

```
[1] "data.frame"
attr(,"split_labels")
  group
1     0
2     2
3     1
```

Some Observations :

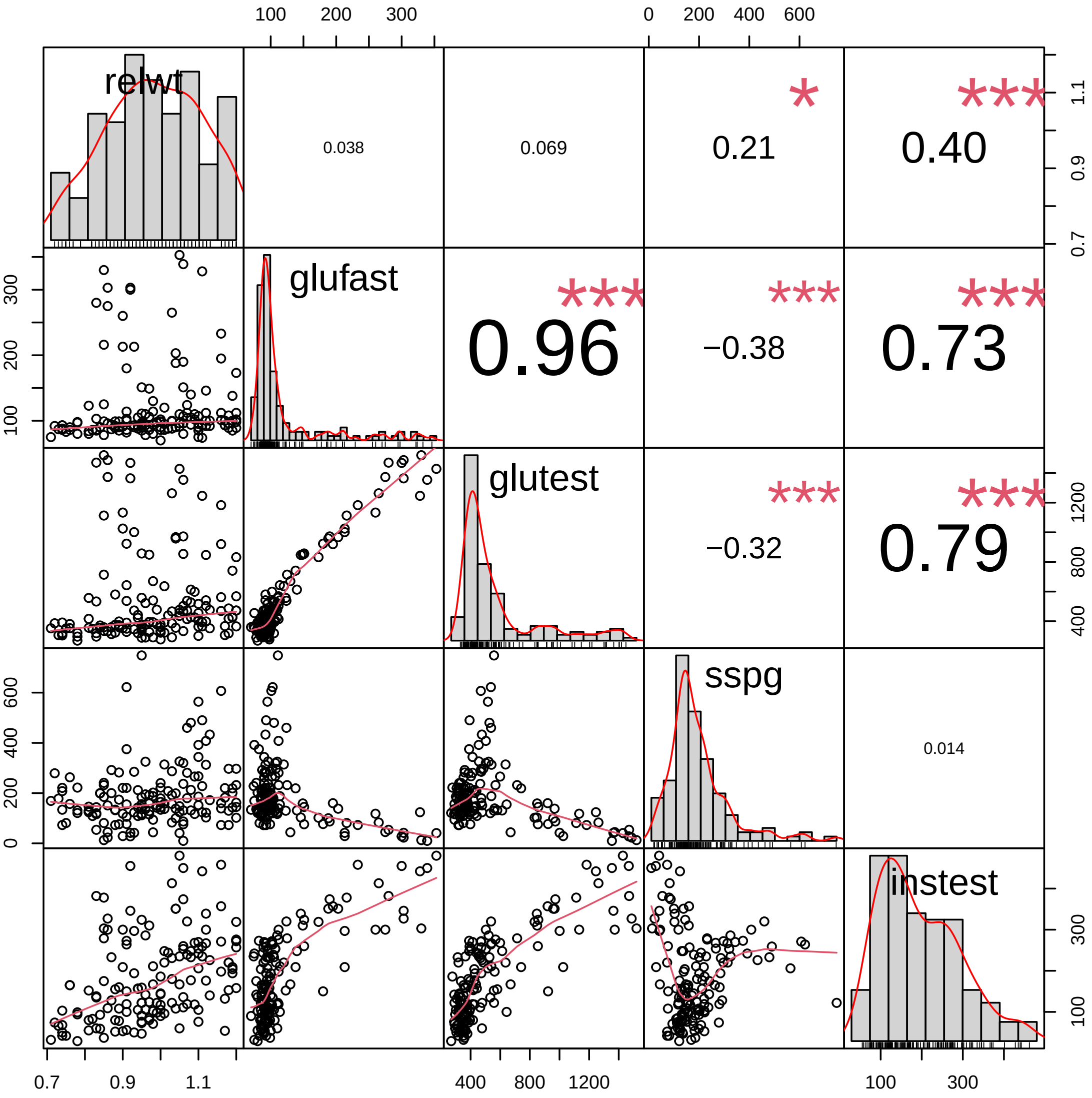
- On an average patients had a large amount of Test Plasma Glucose as compared to Fasting Plasma Glucose and Steady State Plasma Glucose.
- Insulin levels and Steady State Plasma Glucose were more or less same.
- There were 32 over diabetic, 36 chem diabetic and 76 normal people, Among them, mean relative weight was highest for chem. diabetic people.

Chapter 4

Dependency among the response and covariates and Modelling.

4.1 Effects of Covariates on Response.

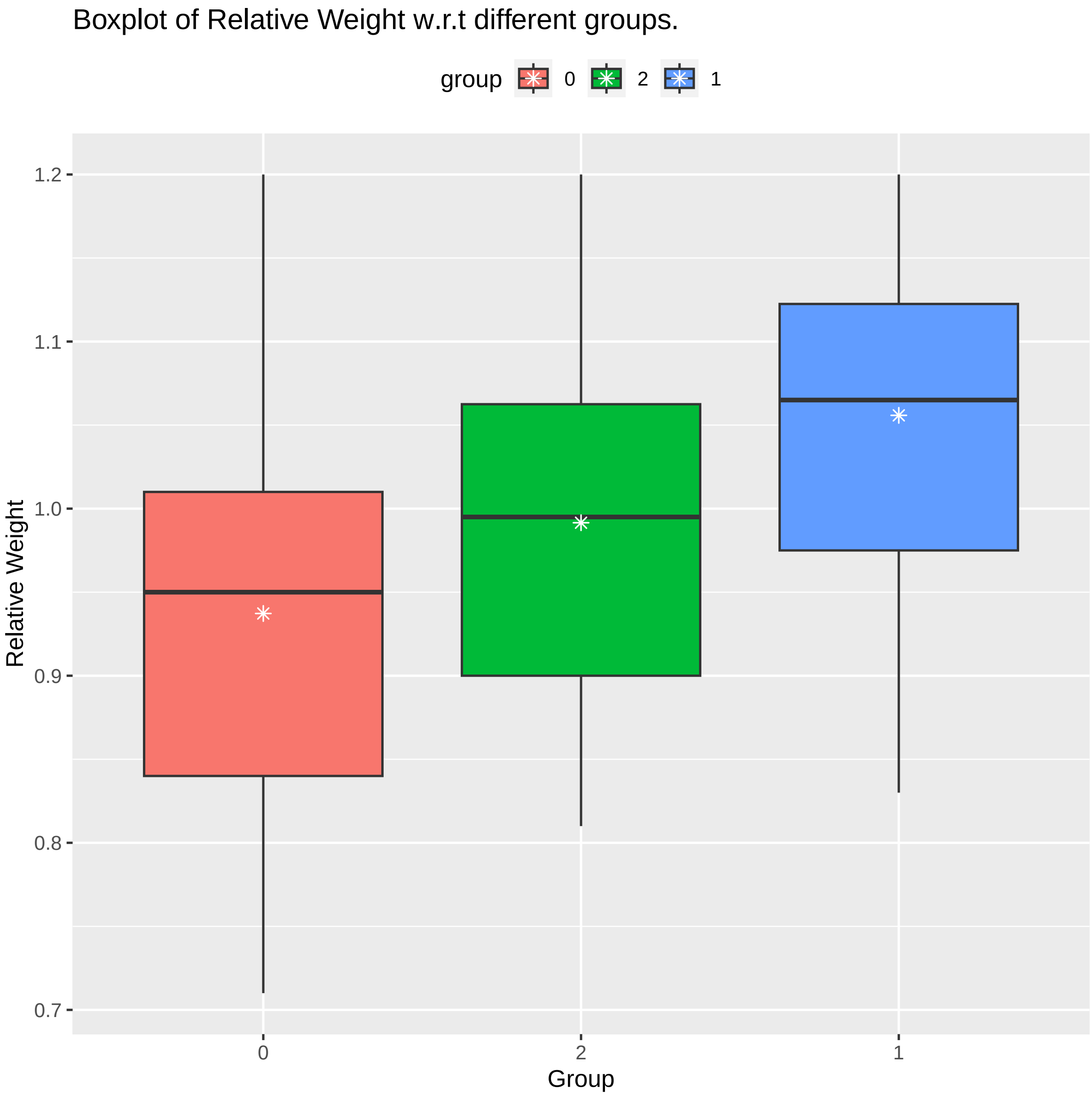
```
#---For Quantitative Responses---
chart.Correlation(dbts[,-6],main="Relationship among the Quantitative Covariates")
```



This plot reveals some interesting features, the response is much afffected by the quantitative responses so the regression may be useful. Also, there are covariates which are highly correlated among themselves like glutest and glufast, glutest and instest, glufast and instest etc. which can affect our estimates.

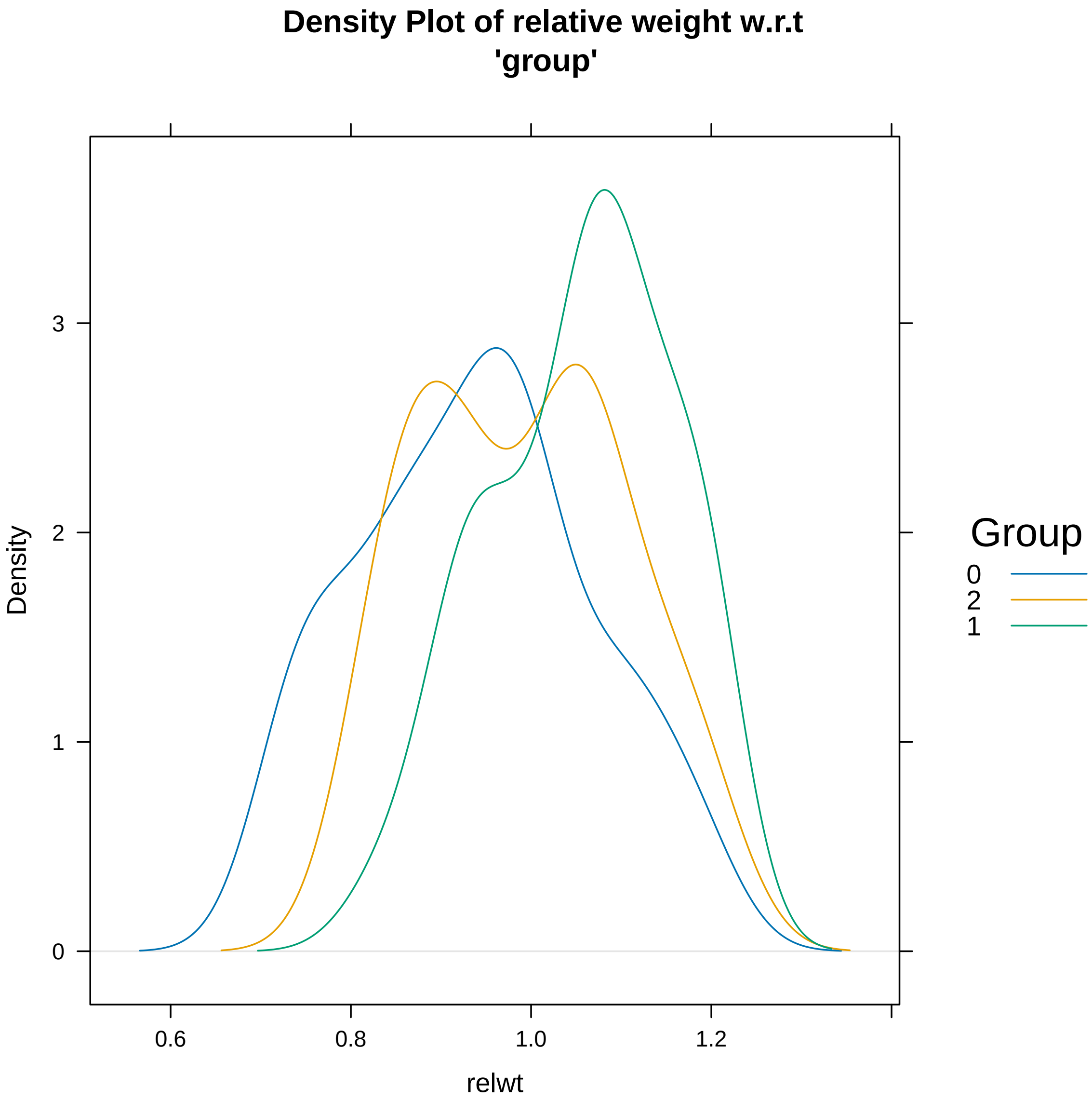
4.2 Boxplot of response w.r.t. 'group'.

```
ggplot(dbts,mapping=aes(x=group,y=relwt,fill=group))+
geom_boxplot()+stat_summary(fun="mean",geom="point",
shape=8,size=2,col="white")+
labs(title="Boxplot of Relative Weight w.r.t different groups.",
y="Relative Weight",x="Group")+theme(legend.position="top")
```



4.3 Densityplot of response w.r.t. 'group'.

```
dp=densityplot(~relwt,data=dbts,groups=group,plot.points=F,
ref=T,main="Density Plot of relative weight w.r.t
'group'",auto.key=list("right",title="Group"))
plot(dp)
```



The response is also greatly affected by the factor 'group'. It will be useful if we use the given variables as our covariates.

4.4 Modelling.

First we consider the full model, i.e. regressing with all the covariates available. As we have one categorical variable and other covariates are quantitative, the model will be an ANCOVA model.

The Model will be as follows,

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i} + \beta_5 z_{5i} + \beta_6 z_{6i} + \epsilon_i \; i = 1\,(1)\,n.$$

assuming, ϵ_i's are iid $\mathcal{N}\left(0, \sigma^2\right)$,σ^2 is unknown.
In Matrix form,

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \sigma^2 I_n\right)$$

where,

$$X = \left(\mathbf{1}_{144}, \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}\right), \boldsymbol{\beta}_{7\times 1} = \left(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\right)'$$

No. of Covariates $p = 7$. (Including the Intercept)

- $\boldsymbol{y}$ stands for Relative Weight.
- $\boldsymbol{x_1}$‘is Glutest.
- $\boldsymbol{x_2}$ is Glufast.
- $\boldsymbol{x_3}$ is sspg.
- $\boldsymbol{x}_4$ is Instest.
- $\boldsymbol{z_5}$ is the indicator whether the person is chem. diabetic.
- $\boldsymbol{z_6}$ is the indicator whether the person is over diabetic.

We name the full model as **'lmodel1'** which will be used further.

```
p=7      #--#Covariates
n=nrow(dbts) #--#Observations.
lmodel1=lm(relwt~.,data=dbts)
summary(lmodel1)
```

```
Call:
lm(formula = relwt ~ ., data = dbts)

Residuals:
      Min        1Q    Median        3Q       Max
-0.214399 -0.065387  0.004717  0.072101  0.293801

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.398e-01  3.288e-02  28.585  < 2e-16 ***
glufast      7.723e-04  6.331e-04   1.220  0.22460
glutest     -4.704e-04  1.541e-04  -3.053  0.00272 **
sspg        -1.156e-04  9.384e-05  -1.232  0.22024
instest      9.793e-04  1.559e-04   6.280 4.19e-09 ***
group2       6.838e-02  5.188e-02   1.318  0.18969
group1       1.004e-01  3.075e-02   3.264  0.00139 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1041 on 137 degrees of freedom
Multiple R-squared:  0.3673,Adjusted R-squared:  0.3396
F-statistic: 13.26 on 6 and 137 DF,  p-value: 8.367e-12
```

The full model depicts that the variales glutest, instest can be dropped frrom the model, but this model stands on several assumptions, so we need to check for validity first.

Chapter 5

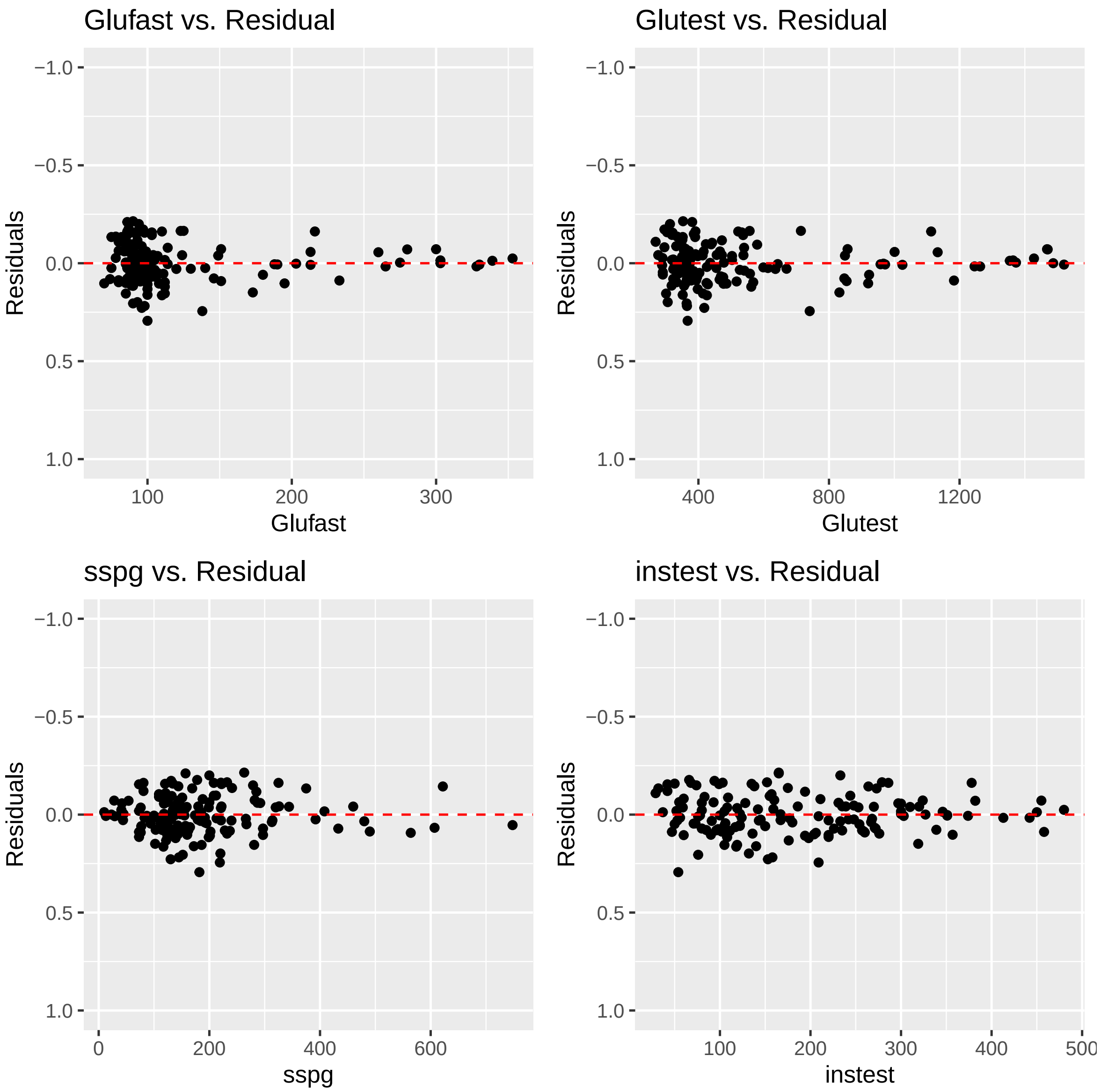
Exploratory Diagonstics.

5.1 Checking for Homoscedasticity.

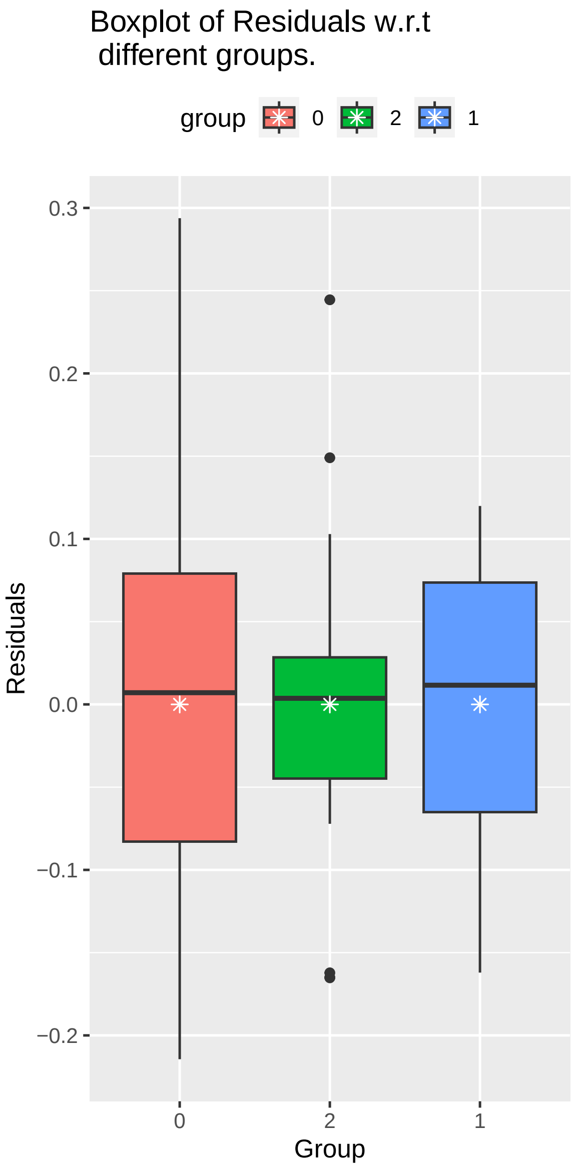
Here we check for the homoscedasticity of the errors by some exploratory diagonstics. We plot the residuals against the different predictors and fitted values. A scatter-plot with the residuals on the vertical axis and the predictor variable on the horizontal axis should ideally look like a constant-width blur of points around a straight, flat line at $y = 0$. Deviations from this like changing width, curvature, substancial regions of the x-axis where the average residuals are either positive or negative are all signs that the model is mis-specified at the beginning.

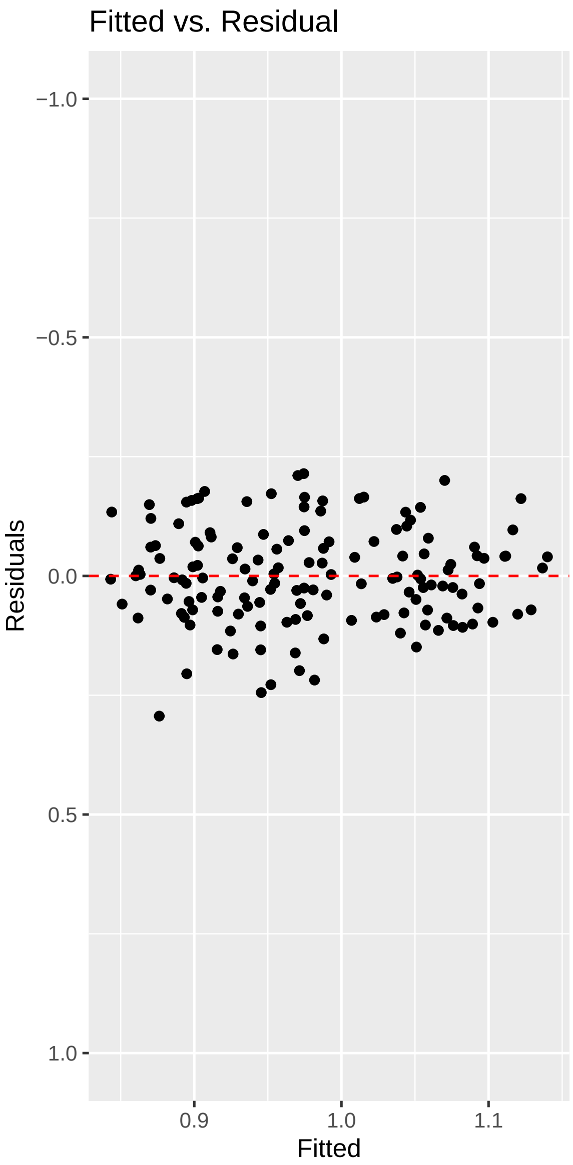
Plotting the residuals against predictors, and fitted values.

```
ggobj1=ggplot(data=dbts,mapping=aes(x=glufast,
y=residuals(lmodel1)))+geom_point()+
geom_hline(yintercept=0,linetype="dashed",col="red")+
ylim(1,-1)+xlab("Glufast")+ylab("Residuals")+
labs(title="Glufast vs. Residual")

ggobj2=ggplot(data=dbts,mapping=aes(x=glutest,
y=residuals(lmodel1)))+geom_point()+
geom_hline(yintercept=0,linetype="dashed",col="red")+
ylim(1,-1)+xlab("Glutest")+ylab("Residuals")+
labs(title="Glutest vs. Residual")

ggobj3=ggplot(data=dbts,mapping=aes(x=sspg,
y=residuals(lmodel1)))+geom_point()+
geom_hline(yintercept=0,linetype="dashed",col="red")+
ylim(1,-1)+xlab("sspg")+ylab("Residuals")+
labs(title="sspg vs. Residual")

ggobj4=ggplot(data=dbts,mapping=aes(x=instest,
y=residuals(lmodel1)))+geom_point()+
geom_hline(yintercept=0,linetype="dashed",col="red")+
ylim(1,-1)+xlab("instest")+ylab("Residuals")+
labs(title="instest vs. Residual")

ggobj5=ggplot(dbts,mapping=aes(x=group,
y=residuals(lmodel1),fill=group))+geom_boxplot()+
stat_summary(fun="mean",geom="point",shape=8,size=2,col="white")+
labs(title="Boxplot of Residuals w.r.t \n different groups.",
y="Residuals",x="Group")+theme(legend.position="top")
grid.arrange(ggobj1,ggobj2,ggobj3,ggobj4,ncol=2,nrow=2)
```



```
ggobj=ggplot(data=dbts,mapping=aes(x=fitted(lmodel1),y=residuals(lmodel1)))
ggobj6=ggobj+geom_point()+geom_hline(yintercept=0,linetype="dashed",col="red")+
ylim(1,-1)+xlab("Fitted")+
ylab("Residuals")+labs(title="Fitted vs. Residual")

grid.arrange(ggobj5,ggobj6,nrow=1,ncol=2)
```





These plots give indication that the errors in the model may be more or less homoscedastic. In spite of that we perform some tests to verify whether the homoscedastic assumptions are valid or not.

5.2 Tests for Heteroscedasticity.

5.2.1 Breusch-Pagan Test.

Here from the full model we calculate the residuals, say $\boldsymbol{e} = (I - P)\,\boldsymbol{y}$, where $P = X\,(X'X)^{-1}\,X'$ and obtain $RSS = \boldsymbol{e}'\,(I - P)\,\boldsymbol{e}$.

We, would test the null hypothesis H_0 : The errors are homoscedastic, against, $H_1 : H_0$ isn't true. Regressing say $p_i = \frac{e_I}{rss}$ on the available covariates. So, we would have a model like,

$$p_i = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_3 x_{3i} + \alpha_4 x_{4i} + \alpha_5 z_{5i} + \alpha_6 z_{6i} + \nu_i \quad i = 1\,(1)\,n$$

The ESS is the sample variance of fitted values got from the above model. Assuming errors are normal, we have,

$$\chi^2 = \frac{1}{2} ESS \sim \chi^2_6 \text{ for sufficiently large } n.$$

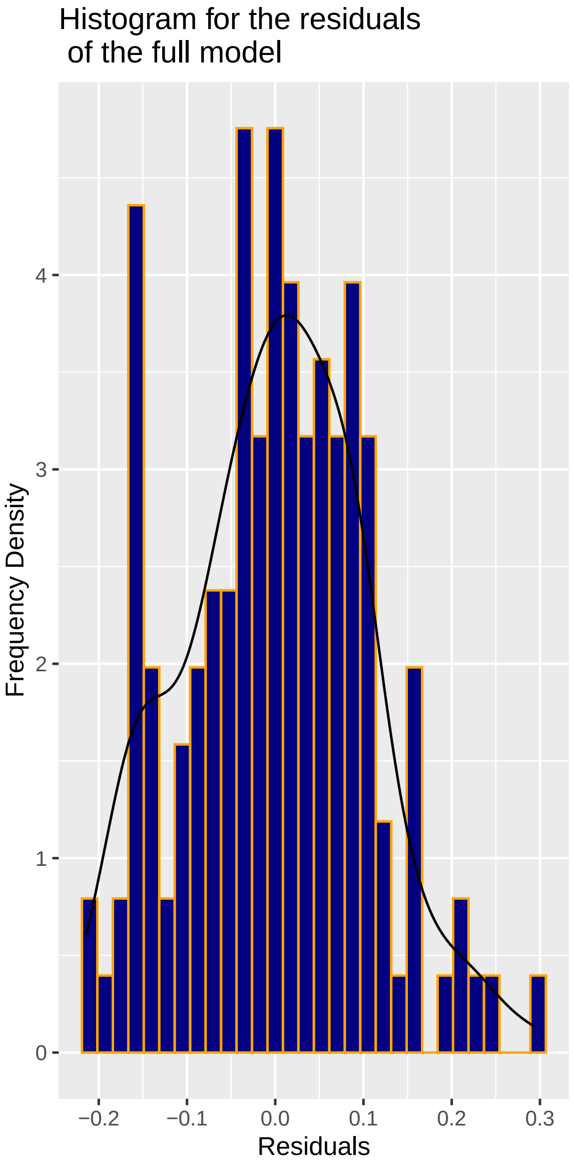
```
bptest(lmodel1)


studentized Breusch-Pagan test

data:  lmodel1
BP = 10.084, df = 6, p-value = 0.1212
```

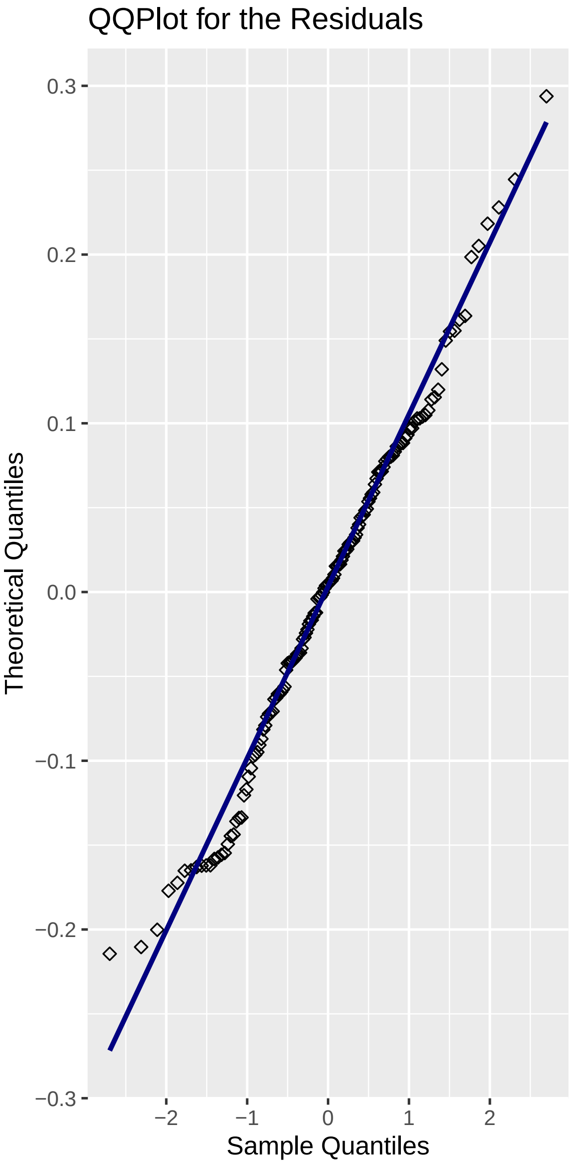
The $p-$value for the test is 0.1212. At level $\alpha = 0.05$ we have to accept the null hypothesis, i.e. the errors may be homoscedastic.

5.3 Checking for Normality.

Now we check for the Normality.

```
df=data.frame(y=residuals(lmodel1))
ggn.obj=ggplot(df,aes(x=df$y,y=after_stat(density)))+
geom_histogram(color="orange",fill="navyblue")+
labs(title="Histogram for the residuals \n of the full model",x="Residuals"
,y="Frequency Density")+geom_density(alpha=0.2)
ggn.obj2=ggplot(df,aes(sample=y))+
stat_qq(shape=5)+stat_qq_line(lwd=1,col="navyblue")+
labs(y="Theoretical Quantiles",x="Sample Quantiles",
title="QQPlot for the Residuals")
grid.arrange(ggn.obj,ggn.obj2,nrow=1,ncol=2)

‘stat_bin()‘ using ‘bins = 30‘.  Pick better value with ‘binwidth‘.
```





Normality assumption slightly violated.

5.4 Testing for Normality.

5.4.1 Kolmogorov-Smirnov Test.

To check for whether a sample $\boldsymbol{x} \sim F$ (in our case $\boldsymbol{e}$) comes from a normal distribution, i.e. the null hypothesis, $H_0 : F = \Phi$ against $H_1 : F \neq \Phi$, we calculate the empirical CDF F_n, defined as,

$$F_n\left(x\right) = \frac{1}{n}\sum_{i=1}^{n} 1_{(-\infty,x]}\left(X_i\right)$$

The Kolmogorov-Smirnov Statistic for a given CDF $F\left(x\right)$ (in our case $\Phi\left(x\right)$) is,

$$D_n = \sup_x \left|F_n\left(x\right) - \Phi\left(x\right)\right|$$

By Glivanko-Cantelli Lemma, if the sample comes from the distribution $\Phi\left(x\right)$, then D_n converges to 0 almost surely as $n \to \infty$.

In practice, we reject H_0 at level α iff $D_n\left(\text{obs.}\right) > D^+_{\alpha/2}$. $D^+_{\alpha/2}$ for different (n, α) is given in DB Owen's Table.

```
ks.test(residuals(lmodel1),'pnorm')


One-sample Kolmogorov-Smirnov test

data:  residuals(lmodel1)
D = 0.41512, p-value < 2.2e-16
alternative hypothesis: two-sided
```

The p-value for the test is very small. At level $\alpha = 0.05$ we reject the null hypothesis stated above.

But, for inferential purposes, we would stick to normality.

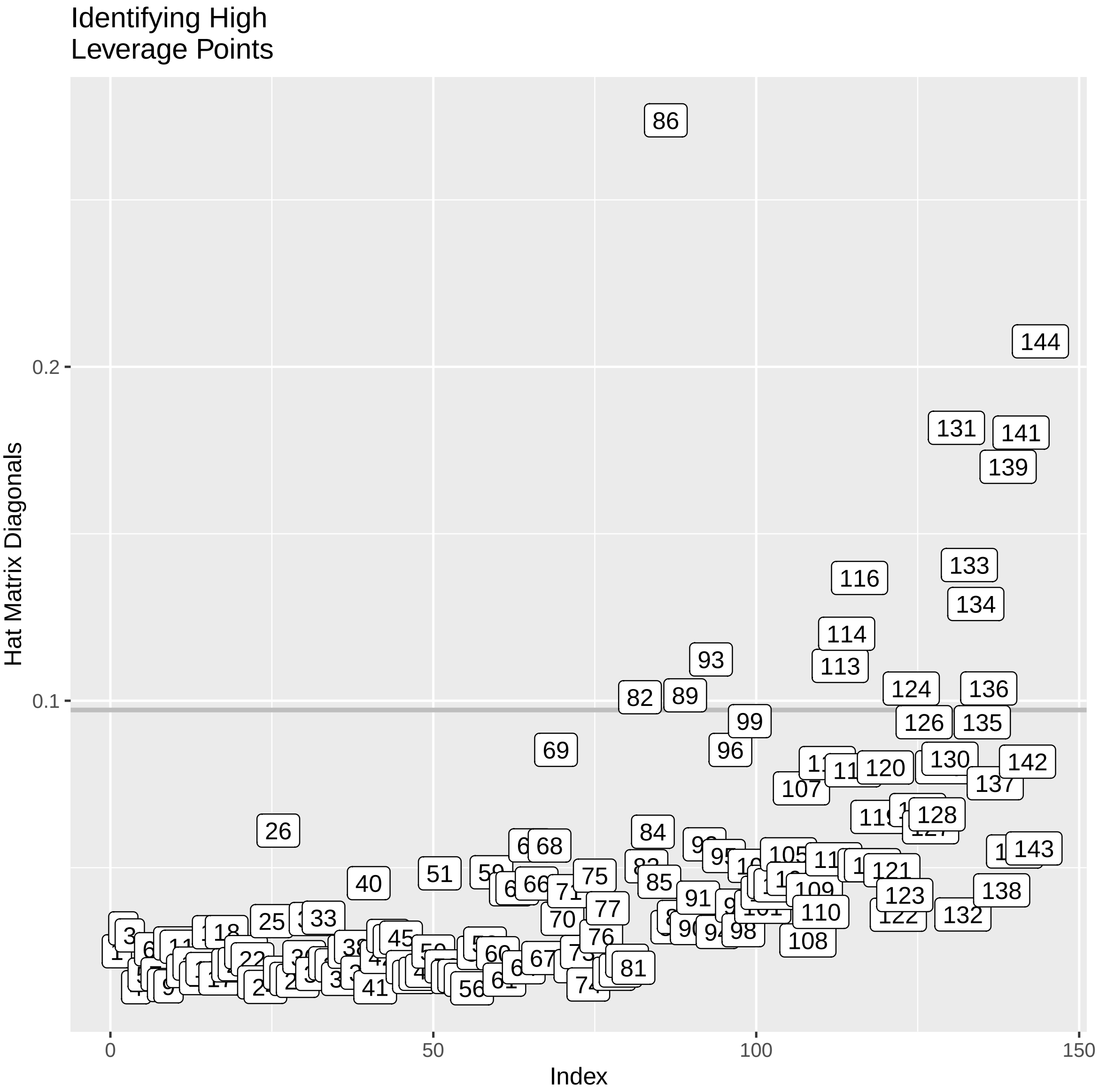
Chapter 6

Outlier Detection.

6.1 Leverages.

Calculating the hat matrix diagonals h_{ii}. We use the cutoff ${}^{2p}/_{n}$.

```
gghat=ggplot(data.frame(y=hatvalues(lmodel1)),
mapping=aes(y=y,x=1:length(y)))+labs(title="Identifying High
Leverage Points",x="Index",y="Hat Matrix Diagonals")
plot2=gghat+geom_point()+geom_hline(yintercept=2*p/n,col="grey",lwd=1)
gghat+geom_point()+geom_hline(yintercept=2*p/n,col="grey",
lwd=1)+geom_label(aes(label=1:length(y)))
```



From the plot we see that there are many high leverage points. But not all the high leverage points influences the regression line, hence we go for some influential measures.

6.2 Influential Observations.

To check for influential observations we use various measures like $DFFITS, DFBETAS, COVRATIO$, Cook's Distance.

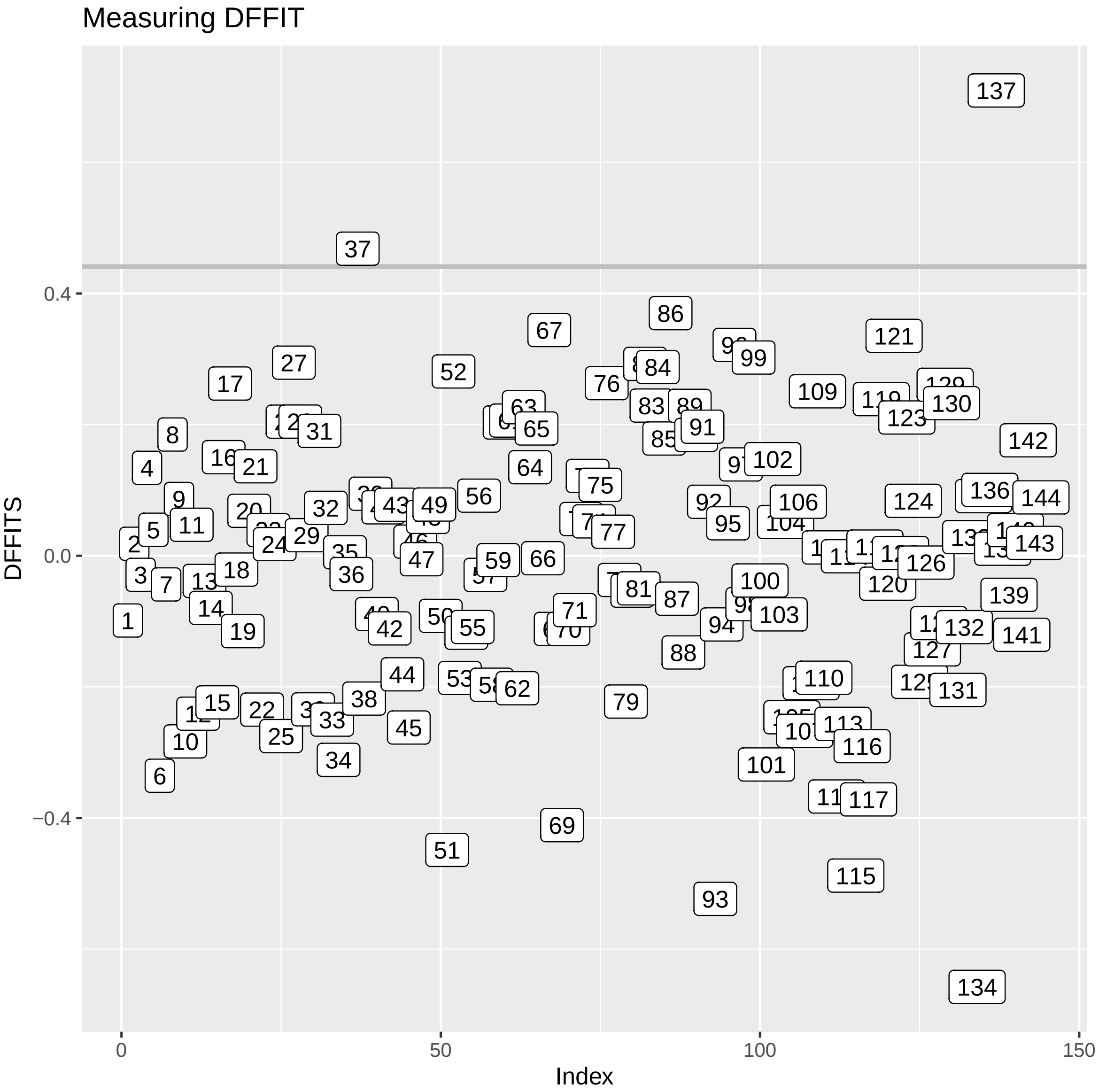
6.2.1 DFFITS.

Standardizing by estimated standard deviation $S\left(i\right)h_i^{1/2}$ of $\boldsymbol{x_i}'\hat{\boldsymbol{\beta}}$, the working formula for $DFFITSS_i$ is,

$$DFFITSS_i = \frac{h_i^{1/2} e_i}{S\left(i\right)\left(1 - h_i\right)}.$$

We use the cutoff $2\sqrt{\frac{p}{n}}$ for finding influential observations.
Plotting DFFITS.

```
ggdffits=ggplot(data.frame(y=dffits(lmodel1)),
mapping=aes(y=y,x=1:length(y)))+
labs(title="Measuring DFFIT",x="Index",y="DFFITS")
plot1=ggdffits+geom_point()+geom_hline(yintercept=2*sqrt(p/n),col="grey",lwd=1)
ggdffits+geom_point()+geom_hline(yintercept=2*sqrt(p/n)
,col="grey",lwd=1)+geom_label(aes(label=1:length(y)))
```

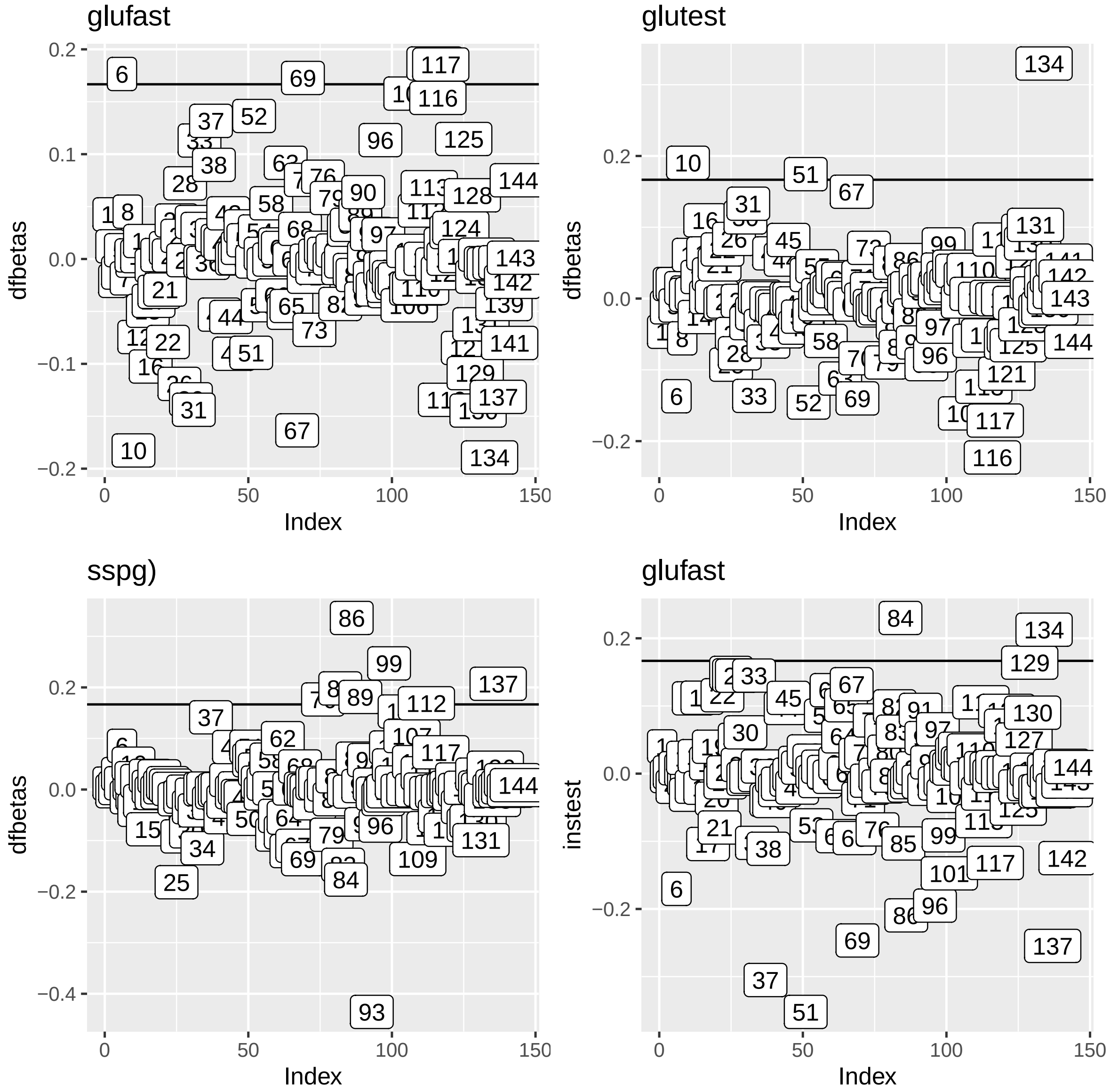


6.2.2 DFBETAS.

$$DFBETAS_{ij} = \frac{\hat{\beta}_j - \hat{\beta}\left(i\right)_j}{S\left(i\right)\left[\left(X'X\right)^{-1}\right]^{1/2}_{j+1,j+1}}.$$

A suitable cutoff will be $2/\sqrt{n}$. We plot for different covariates.

```
lmodel1.dfbetas=data.frame(dfbetas(lmodel1))
db1=ggplot(data.frame(y=lmodel1.dfbetas[,2])
,mapping=aes(y=y,x=1:length(y)))+geom_point()+
geom_hline(yintercept=2/sqrt(n))+labs(title="glufast",x="Index",y="dfbetas")

db2=ggplot(data.frame(y=lmodel1.dfbetas[,3])
,mapping=aes(y=y,x=1:length(y)))+geom_point()+
geom_hline(yintercept=2/sqrt(n))+labs(title="glutest",x="Index",y="dfbetas")

db3=ggplot(data.frame(y=lmodel1.dfbetas[,4])
,mapping=aes(y=y,x=1:length(y)))+geom_point()+
geom_hline(yintercept=2/sqrt(n))+labs(title="sspg)",x="Index",y="dfbetas")

db4=ggplot(data.frame(y=lmodel1.dfbetas[,5])
,mapping=aes(y=y,x=1:length(y)))+geom_point()+
geom_hline(yintercept=2/sqrt(n))+labs(title="glufast",x="Index",y="instest")

grid.arrange(db1+geom_label(aes(label=1:length(y))),
db2+geom_label(aes(label=1:length(y))),db3+geom_label(aes(label=1:length(y)))
,db4+geom_label(aes(label=1:length(y))),ncol=2,nrow=2)
```



```
#---grid.arrange(db1,db2,db3,db4,ncol=2,nrow=2)
```

6.2.3 COVRATIO.

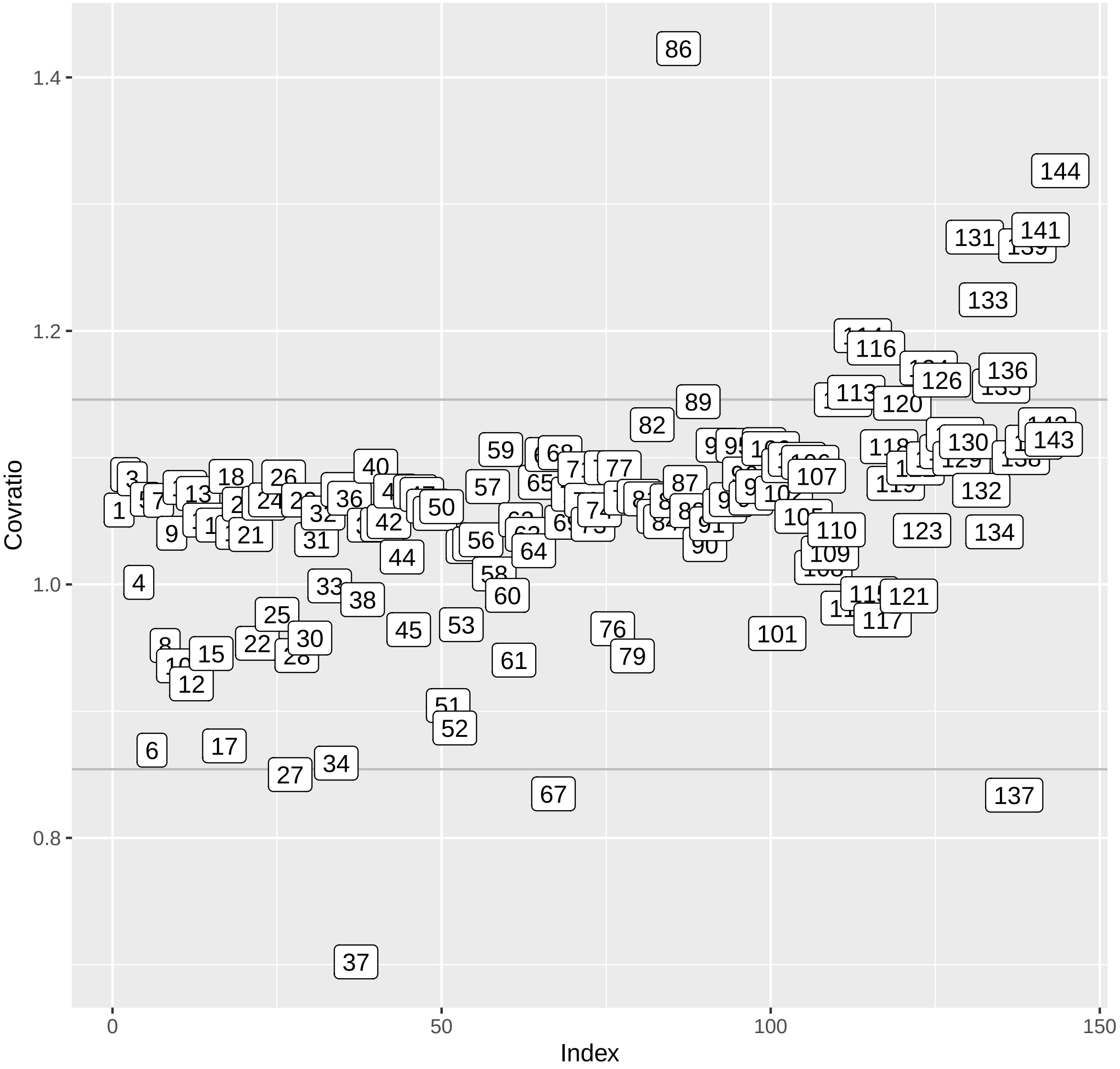
The expression for $COVRATIO$ is,

$$\frac{\det\left\{S\left(i\right)^{2}\left[X\left(i\right)'X\left(i\right)\right]^{-1}\right\}}{\det\left[S^{2}\left(X'X\right)^{-1}\right]}$$

The cases having $|COVRATIO - 1| > \frac{3p}{n}$ are considered to have high influence. Plotting $COVRATIO$,

```
cratio=data.frame(y=covratio(lmodel1))
ggplot(cratio,mapping=aes(x=1:length(y),y=y))+geom_point()+
geom_hline(yintercept=1+(3*p/n),col="grey")+geom_hline(yintercept=1-(3*p/n)
,col="grey")+labs(title="COVRATIO of the full model."
,x="Index",y="Covratio")+geom_label(aes(label=1:length(y)))
```

COVRATIO of the full model.



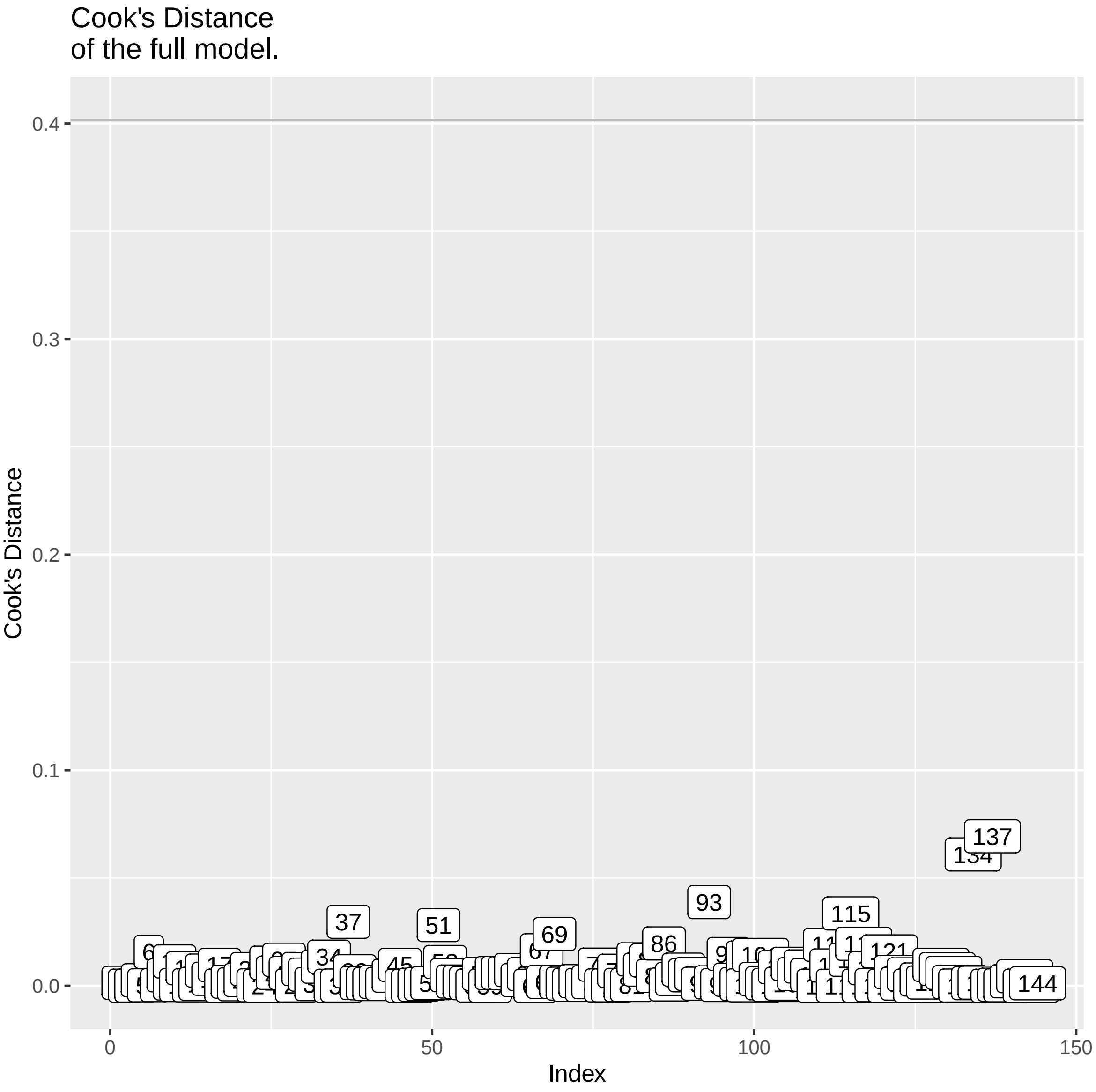
6.2.4 Cook's Distance.

Cook [1977] suggested measuring the distance of $\hat{\boldsymbol{\beta}}\,(\boldsymbol{i})$ from $\hat{\boldsymbol{\beta}}$ by using the measure,

$$D_i = \frac{\left(\hat{\boldsymbol{\beta}}\,(\boldsymbol{i}) - \hat{\boldsymbol{\beta}}\right)' X'X \left(\hat{\boldsymbol{\beta}}\,(\boldsymbol{i}) - \hat{\boldsymbol{\beta}}\right)}{pS^2}$$

He suggested flagging as suspicious those points for which $D_i > F^{0.10}_{p,n-p}$.

```
cd=data.frame(y=cooks.distance(lmodel1))
ggplot(cd,mapping=aes(x=1:length(y),y=y))+geom_point()+
geom_hline(yintercept=qf(p=0.10,df1=p,df2=n-p,
lower.tail=T),col="grey")+labs(title="Cook's Distance
of the full model.",x="Index",y="Cook's Distance")+
geom_label(aes(label=1:length(y)))
```



And finally, the potential influential observations are,

```
influence.measures(lmodel1)
```

```
Influence measures of
 lm(formula = relwt ~ ., data = dbts) :

       dfb.1_  dfb.glfs  dfb.glts  dfb.sspg  dfb.inst  dfb.grp2  dfb.grp1
1   -5.20e-02  0.042638 -0.046618  1.16e-02  4.02e-02  0.022870  3.09e-02
2    6.78e-03  0.012117 -0.012286 -3.42e-03 -3.60e-03  0.005742  5.21e-03
3   -6.28e-03 -0.020874  0.020419  2.88e-03  1.85e-03 -0.006245 -5.63e-03
4    3.20e-02 -0.012966  0.020287  3.03e-02 -1.95e-02 -0.035553 -6.39e-02
5    3.31e-03  0.008398 -0.009792  1.23e-02  7.93e-03 -0.004187 -1.57e-02
6   -1.23e-01  0.176477 -0.137012  8.55e-02 -1.70e-01  0.183067  2.15e-01
7   -3.07e-04 -0.018486  0.014601 -1.50e-02  4.97e-03  0.002015  9.94e-03
8    7.43e-02  0.045181 -0.055814  1.13e-02 -8.34e-03  0.011194 -3.48e-02
9    3.24e-02 -0.001633  0.008468 -8.84e-03 -1.67e-02 -0.024239 -2.68e-02
10  -1.01e-01 -0.182625  0.190489  5.06e-02  2.33e-02 -0.073692 -5.96e-02
11   3.57e-03  0.003417  0.005189  2.22e-02 -3.12e-02 -0.001268 -1.01e-02
12  -8.30e-02 -0.074439  0.061164 -3.97e-02  1.11e-01 -0.039460  6.52e-03
13  -2.39e-02  0.002729  0.006371  1.65e-02 -1.93e-02  0.003963  1.02e-02
14  -1.57e-02  0.017154 -0.025965 -2.29e-02  2.58e-02  0.024029  3.87e-02
15  -5.17e-02 -0.049389  0.030248 -7.78e-02  1.12e-01 -0.023693  3.09e-02
16   3.40e-02 -0.102568  0.109796  2.66e-02 -2.09e-02 -0.074203 -9.96e-02
17   1.04e-01 -0.039927  0.063371  4.98e-03 -1.04e-01 -0.056028 -8.09e-02
18  -2.05e-03 -0.006321  0.002007 -6.96e-03  1.53e-02 -0.001726  5.73e-05
19  -6.80e-02 -0.031348  0.033897  2.58e-02  3.97e-02 -0.021999 -1.19e-02
20   3.20e-02  0.003758  0.002354 -5.69e-03 -3.70e-02 -0.000493 -3.57e-03
21   4.98e-02 -0.029179  0.047865  4.37e-03 -7.96e-02 -0.025593 -3.15e-02
22  -1.08e-01 -0.079131  0.068891  1.25e-02  1.16e-01 -0.049670 -2.71e-02
23   8.85e-03  0.010295 -0.006532  7.66e-03 -1.28e-02 -0.001685 -7.52e-03
24   5.33e-03  0.003372 -0.004120  1.31e-03  1.95e-03 -0.001739 -5.61e-03
25   3.44e-02  0.036752 -0.092758 -1.82e-01  1.49e-01  0.057519  1.19e-01
26   3.73e-02 -0.119127  0.083641 -1.91e-02  1.45e-01 -0.104635 -1.41e-01
27   9.99e-02  0.022085 -0.049678 -9.18e-02  1.45e-01 -0.090803 -1.02e-01
28   4.51e-02  0.072172 -0.076786 -1.40e-02  4.99e-02 -0.027921 -4.64e-02
29   1.26e-02 -0.001104 -0.003228 -4.13e-03  1.29e-02 -0.006896 -1.33e-02
30  -8.48e-03 -0.133842  0.113866 -8.19e-02  6.10e-02 -0.035819  8.24e-03
31   9.81e-02 -0.143801  0.133083 -3.46e-02 -7.63e-03 -0.079068 -1.02e-01
32   2.47e-02  0.035242 -0.034465 -1.02e-02 -4.59e-03  0.004027  6.17e-04
33  -8.85e-02  0.113036 -0.136507 -4.10e-02  1.45e-01  0.046079  8.37e-02
34   6.56e-03  0.028709 -0.020592 -1.15e-01 -1.02e-01  0.105145  1.95e-01
35   1.74e-03 -0.002230  0.002691  7.10e-04 -1.94e-03 -0.001766 -2.53e-03
36  -9.80e-03 -0.003628  0.000301  1.96e-03  1.05e-02  0.004188  4.25e-03
37   5.08e-02  0.131764 -0.038468  1.41e-01 -3.05e-01  0.008002 -2.96e-02
38  -4.19e-02  0.089794 -0.060260 -2.62e-02 -1.11e-01  0.088574  1.51e-01
39   2.43e-02  0.022587 -0.006956 -5.51e-03 -3.86e-02 -0.014715 -9.16e-03
```

```
40  -5.97e-03 -0.053033  0.063415 -8.50e-04 -4.05e-02 -0.012196  1.53e-03
41   1.56e-02  0.017121 -0.011100  7.95e-04 -3.79e-03 -0.017180 -2.09e-02
42   1.04e-02  0.014488 -0.016039 -5.41e-02 -2.43e-02  0.038665  7.23e-02
43   3.76e-02  0.043588 -0.045977 -2.40e-02 -1.49e-02  0.021917  2.14e-02
44  -9.93e-02 -0.055307  0.054623  1.89e-02  9.70e-02 -0.056840 -3.53e-02
45  -1.42e-01 -0.090396  0.082173  8.32e-02  1.12e-01 -0.049638 -5.89e-02
46   1.30e-02  0.000489 -0.000096 -5.29e-03 -8.80e-03  0.000322 -6.66e-04
47  -3.36e-03  0.000930 -0.000192  2.37e-03 -1.48e-03  0.001252  1.65e-03
48   2.80e-02  0.010663 -0.006710 -1.28e-02 -2.06e-02 -0.002713 -2.07e-03
49   3.48e-02  0.031113 -0.040843 -1.53e-02  1.59e-02  0.009082 -3.20e-03
50   9.56e-03  0.017879 -0.025403 -5.74e-02  8.82e-03  0.027631  5.72e-02
51  -6.51e-02 -0.089469  0.174513  8.38e-02 -3.52e-01  0.065264  1.20e-01
52   3.56e-02  0.136412 -0.145868  5.74e-02  3.28e-02  0.023845 -4.12e-02
53  -9.95e-02 -0.002409  0.033115  6.42e-02 -7.64e-02  0.029381  5.39e-02
54  -7.39e-02  0.026234 -0.023625  3.91e-02  1.92e-02  0.024304  2.91e-02
55  -3.98e-02 -0.044101  0.045285  1.36e-02  1.66e-03 -0.002241  6.95e-03
56   4.67e-02  0.004893 -0.008003 -7.68e-03 -1.09e-02 -0.005848 -2.17e-02
57  -1.85e-02 -0.004834  0.004844  1.14e-02  1.03e-02 -0.002687 -3.36e-03
58  -1.21e-01  0.053157 -0.059456  5.86e-02  8.56e-02  0.025124  2.52e-02
59  -2.21e-03 -0.000556  0.000400  2.40e-03  2.50e-03 -0.001250 -5.07e-03
60   6.91e-02 -0.034821  0.014077 -8.82e-02  1.23e-01 -0.097565 -9.07e-02
61   5.36e-02  0.013927  0.010308 -5.17e-02 -4.64e-03 -0.087063 -6.09e-02
62  -6.73e-02  0.010546  0.029981  1.01e-01 -9.24e-02 -0.007017 -1.05e-01
63   5.59e-02  0.091741 -0.112180 -8.38e-02  1.54e-02  0.069302  1.89e-01
64   8.15e-02 -0.051277  0.027710 -5.51e-02  5.51e-02 -0.044876 -6.10e-02
65   7.36e-02 -0.044816  0.005990 -1.21e-01  1.01e-01 -0.021827  7.36e-02
66  -1.64e-03 -0.000381  0.000782  2.03e-03  4.93e-05 -0.000800 -2.89e-03
67   1.19e-01 -0.163453  0.149857 -1.10e-01  1.32e-01 -0.212112 -2.07e-01
68  -2.71e-02  0.028765 -0.008405  4.52e-02 -9.52e-02  0.050734  5.21e-02
69   6.83e-02  0.172693 -0.140032 -1.36e-01 -2.47e-01  0.198371  3.02e-01
70  -3.60e-02  0.075464 -0.083699  1.66e-03  2.76e-02  0.054453  6.06e-02
71   1.55e-03 -0.016947  0.029847 -1.27e-03 -3.72e-02 -0.011714 -3.80e-02
72   6.45e-03 -0.001637 -0.001618 -3.23e-03  3.14e-02 -0.024534 -3.00e-02
73   5.88e-02 -0.067583  0.071135 -2.69e-02 -2.09e-02 -0.052668 -5.50e-02
74   1.01e-02 -0.008349  0.013448  1.22e-02 -1.04e-02 -0.017974 -2.66e-02
75  -1.68e-02  0.003138 -0.010386  1.31e-02  7.86e-02 -0.044297 -5.76e-02
76  -8.89e-02  0.078494 -0.016673  1.76e-01 -8.25e-02 -0.052667 -9.19e-02
77   1.03e-02  0.010703 -0.013638 -1.16e-02 -1.82e-03  0.011228  3.07e-02
78  -1.33e-02  0.004943 -0.000867  9.89e-03 -1.89e-02  0.014469  1.76e-02
79  -1.26e-02  0.058195 -0.089647 -8.86e-02  6.32e-02  0.086207  1.27e-01
80  -9.27e-03  0.000538 -0.009033 -1.90e-02  3.25e-02  0.004444  1.34e-02
81  -2.68e-02  0.008100 -0.008082  2.59e-02 -2.96e-03  0.018263  1.24e-02
82  -1.20e-01 -0.042921  0.050314  1.98e-01  9.94e-02 -0.100372 -1.88e-01
83   1.11e-01 -0.008567 -0.034810 -1.48e-01  6.20e-02  0.027238  1.45e-01
84   1.26e-01 -0.002219 -0.067750 -1.76e-01  2.30e-01 -0.054679 -5.15e-02
85   2.59e-02  0.027899 -0.011805  6.26e-04 -1.03e-01  0.046299  1.32e-01
```

```
86  -1.98e-01  0.032576  0.054045  3.36e-01 -2.09e-01  0.011653  4.70e-03
87   1.05e-02 -0.010502  0.011831 -4.08e-03 -1.75e-02  0.002331 -3.13e-02
88  -2.00e-02 -0.015250  0.014511  6.05e-02 -1.27e-02  0.008456 -9.70e-02
89  -1.22e-01  0.042380 -0.023640  1.81e-01 -4.91e-03  0.008191  1.60e-02
90  -1.86e-02  0.063780 -0.061556  6.83e-03  7.81e-03  0.024436  1.25e-01
91   5.40e-03 -0.037779  0.026496 -6.79e-02  9.42e-02 -0.063293  5.72e-02
92  -3.99e-02  0.001219  0.009654  5.78e-02 -1.20e-02 -0.007090  1.07e-02
93   2.95e-01  0.023940 -0.091991 -4.35e-01  5.25e-02  0.066332  4.35e-02
94  -2.93e-03 -0.018076  0.025680  9.54e-03 -2.25e-02 -0.009568 -6.42e-02
95   7.49e-05 -0.029028  0.027950 -8.03e-03  1.67e-02 -0.024135 -3.86e-04
96   7.17e-02  0.113420 -0.079662 -7.06e-02 -1.95e-01  0.103884  2.61e-01
97  -2.82e-03  0.023481 -0.039435 -1.86e-02  6.55e-02  0.001339  6.62e-02
98   4.21e-03 -0.011624  0.016469 -5.88e-03 -1.46e-02 -0.007327 -4.01e-02
99  -1.43e-01 -0.030629  0.075984  2.47e-01 -9.12e-02 -0.021852  1.04e-04
100  1.38e-02 -0.009639  0.010095 -1.15e-02 -1.44e-02  0.001758 -1.04e-02
101  5.97e-02 -0.025588  0.040647 -3.48e-03 -1.47e-01  0.047730 -1.06e-01
102 -4.58e-02 -0.008066  0.014657  8.23e-02 -8.70e-03 -0.002513  3.33e-02
103 -2.28e-02 -0.017357  0.029155  4.72e-02 -3.29e-02 -0.006780 -5.82e-02
104 -2.54e-03 -0.022873  0.018497  1.39e-02  1.02e-02 -0.007791  3.57e-03
105 -1.19e-01 -0.019477  0.044766  1.53e-01  7.44e-03 -0.047941 -1.86e-01
106 -1.12e-03 -0.044215  0.039030 -2.80e-03  2.98e-02 -0.031303  1.77e-03
107 -8.44e-02  0.157794 -0.161086  1.05e-01  3.68e-02  0.077429 -5.73e-02
108 -1.56e-02  0.007761 -0.004134  1.26e-03  1.51e-02 -0.012521 -1.15e-01
109  6.34e-02 -0.028205  0.028171 -1.36e-01  1.30e-02 -0.038295  1.39e-01
110 -6.24e-02 -0.027646  0.040196  5.37e-02  3.43e-02 -0.054681 -1.51e-01
111  1.23e-03 -0.004285  0.006589 -1.61e-03 -6.88e-03 -0.002755  4.03e-03
112 -1.24e-01  0.046210 -0.059293  1.69e-01  1.05e-01  0.004104 -2.26e-01
113  1.05e-01  0.068463 -0.123684  3.19e-02 -7.01e-02  0.145395  1.00e-01
114  3.29e-04  0.000156 -0.000427 -5.86e-05  2.34e-04  0.000333  1.90e-04
115 -1.89e-01  0.186474 -0.051713 -7.57e-02 -2.94e-02 -0.273881  4.69e-02
116  9.48e-02  0.153671 -0.222867 -7.07e-03  8.15e-03  0.163507  1.34e-01
117 -9.72e-03  0.185518 -0.171100  7.29e-02 -1.32e-01  0.053901  1.30e-01
118  3.70e-03 -0.005216  0.002270 -4.26e-03  7.10e-03  0.002225 -2.68e-03
119  1.27e-01 -0.134344  0.082789 -5.15e-02 -3.34e-03  0.089059 -1.75e-02
120  2.17e-02 -0.013505 -0.001093 -2.88e-03  9.66e-03  0.006596 -5.92e-04
121  1.37e-01  0.014114 -0.105371 -7.88e-02  9.28e-02  0.218865  5.67e-02
122  2.04e-04  0.000240 -0.000745 -6.66e-05  1.16e-03  0.001776 -3.92e-05
123  1.61e-02  0.025683 -0.061744  3.46e-03  7.09e-02  0.106831  4.88e-03
124  3.83e-02  0.029568 -0.052851 -8.68e-03  1.30e-03  0.076637  3.56e-02
125 -7.53e-02  0.114470 -0.065477  1.97e-02 -5.17e-02 -0.051420  4.61e-02
126  3.48e-03  0.002815 -0.006086 -1.03e-03  3.82e-03  0.003407  2.44e-03
127  5.37e-02 -0.084826  0.051711 -3.41e-02  5.05e-02 -0.048518 -3.54e-02
128 -3.90e-02  0.060777 -0.036129  3.44e-03 -1.89e-02 -0.031114  2.38e-02
129 -1.91e-02 -0.108854  0.095040 -6.96e-02  1.64e-01 -0.080768 -1.05e-01
130  1.09e-01 -0.144548  0.077247 -6.20e-02  9.01e-02  0.048241 -5.31e-02
131 -2.99e-03 -0.062282  0.103350 -9.92e-02 -1.53e-02 -0.146861 -2.19e-02
```

```
132 -2.22e-02 -0.016955  0.021007  2.92e-02  5.80e-03 -0.053145 -2.54e-02
133 -1.45e-02  0.004174  0.007323  3.80e-03 -1.25e-02 -0.008022 -1.31e-03
134 -3.23e-01 -0.189288  0.329725 -8.83e-06  2.13e-01 -0.627542 -3.09e-01
135  6.00e-02 -0.003970 -0.017524 -2.03e-02 -2.63e-02  0.073870  3.31e-02
136  2.42e-02 -0.004997 -0.013783  4.27e-02 -2.56e-02  0.075623  8.08e-03
137  2.29e-01 -0.131530  0.023173  2.07e-01 -2.55e-01  0.494459  5.50e-02
138  2.88e-03 -0.003699  0.001591 -2.44e-03  4.08e-03  0.002740 -1.64e-03
139  2.61e-02 -0.042963  0.031683  6.39e-03 -1.06e-02 -0.001257 -1.28e-02
140 -1.50e-02 -0.003108  0.009351 -2.35e-03  1.16e-02 -0.011159 -1.10e-02
141  6.43e-02 -0.080092  0.053322  4.34e-03 -2.44e-02  0.010315 -1.48e-02
142  4.60e-02 -0.021786  0.030931  1.83e-02 -1.25e-01  0.088746  3.50e-02
143  3.24e-03  0.001235  0.000909 -1.10e-03 -1.18e-02  0.008732  5.19e-03
144 -4.18e-02  0.075091 -0.060682  8.54e-03  9.72e-03  0.017035  2.35e-02
        dffit cov.r   cook.d    hat inf
1   -0.098757 1.059 1.40e-03 0.0250
2    0.018281 1.087 4.81e-05 0.0318
3   -0.029052 1.083 1.21e-04 0.0297
4    0.134091 1.002 2.56e-03 0.0142
5    0.039666 1.067 2.26e-04 0.0180
6   -0.335546 0.869 1.57e-02 0.0256
7   -0.043730 1.066 2.75e-04 0.0181
8    0.184905 0.952 4.84e-03 0.0149
9    0.086661 1.040 1.08e-03 0.0145
10  -0.283143 0.936 1.13e-02 0.0273
11   0.047288 1.076 3.22e-04 0.0263
12  -0.240915 0.922 8.17e-03 0.0190
13  -0.038400 1.072 2.12e-04 0.0213
14  -0.079521 1.051 9.08e-04 0.0169
15  -0.223583 0.945 7.06e-03 0.0196
16   0.150406 1.047 3.24e-03 0.0307
17   0.262700 0.873 9.65e-03 0.0168
18  -0.021101 1.085 6.41e-05 0.0307
19  -0.114976 1.041 1.89e-03 0.0207
20   0.068593 1.063 6.76e-04 0.0211
21   0.136460 1.039 2.67e-03 0.0247
22  -0.234630 0.954 7.79e-03 0.0226
23   0.039251 1.064 2.22e-04 0.0157
24   0.017768 1.067 4.54e-05 0.0143
25  -0.275044 0.976 1.07e-02 0.0340
26   0.204538 1.085 5.99e-03 0.0611
27   0.294532 0.850 1.21e-02 0.0185
28   0.204652 0.944 5.92e-03 0.0167
29   0.031356 1.067 1.41e-04 0.0161
30  -0.234283 0.958 7.77e-03 0.0232
31   0.190428 1.035 5.18e-03 0.0346
32   0.073051 1.056 7.66e-04 0.0182
```

```
33  -0.249965 0.999 8.88e-03 0.0350
34  -0.311297 0.859 1.35e-02 0.0214
35   0.005323 1.075 4.08e-06 0.0208
36  -0.027923 1.068 1.12e-04 0.0167
37   0.468356 0.702 2.97e-02 0.0248   *
38  -0.217853 0.988 6.74e-03 0.0263
39   0.094590 1.047 1.28e-03 0.0186
40  -0.088920 1.093 1.14e-03 0.0454
41   0.073844 1.047 7.83e-04 0.0142
42  -0.110826 1.050 1.76e-03 0.0233
43   0.077798 1.074 8.70e-04 0.0296
44  -0.180850 1.022 4.67e-03 0.0280
45  -0.261962 0.964 9.71e-03 0.0290
46   0.021557 1.073 6.69e-05 0.0202
47  -0.005276 1.071 4.01e-06 0.0173
48   0.059243 1.062 5.04e-04 0.0182
49   0.078095 1.056 8.76e-04 0.0191
50  -0.091753 1.061 1.21e-03 0.0249
51  -0.448683 0.904 2.82e-02 0.0483
52   0.281088 0.887 1.11e-02 0.0204
53  -0.186409 0.968 4.93e-03 0.0173
54  -0.117464 1.030 1.97e-03 0.0176
55  -0.108758 1.032 1.69e-03 0.0164
56   0.091654 1.035 1.20e-03 0.0139
57  -0.029222 1.077 1.23e-04 0.0245
58  -0.196723 1.008 5.51e-03 0.0273
59  -0.006631 1.106 6.33e-06 0.0486
60   0.203300 0.991 5.88e-03 0.0244
61   0.206171 0.940 6.00e-03 0.0165
62  -0.202045 1.051 5.84e-03 0.0435
63   0.226565 1.040 7.33e-03 0.0439
64   0.135059 1.027 2.61e-03 0.0202
65   0.194112 1.080 5.40e-03 0.0567
66  -0.003741 1.102 2.01e-06 0.0452
67   0.344286 0.835 1.64e-02 0.0229   *
68  -0.111644 1.104 1.79e-03 0.0566
69  -0.410923 1.049 2.40e-02 0.0853
70  -0.111682 1.071 1.79e-03 0.0347
71  -0.082964 1.091 9.89e-04 0.0430
72   0.056030 1.066 4.51e-04 0.0205
73   0.121749 1.047 2.12e-03 0.0247
74   0.052436 1.059 3.95e-04 0.0150
75   0.108171 1.092 1.68e-03 0.0476
76   0.263293 0.965 9.81e-03 0.0294
77   0.036650 1.092 1.93e-04 0.0378
78  -0.036889 1.068 1.96e-04 0.0182
```

```
79  -0.222343 0.944 6.98e-03 0.0192
80  -0.053253 1.070 4.08e-04 0.0221
81  -0.049737 1.067 3.56e-04 0.0200
82   0.292727 1.126 1.23e-02 0.1010
83   0.229103 1.054 7.50e-03 0.0504
84   0.287819 1.050 1.18e-02 0.0608
85   0.178833 1.066 4.58e-03 0.0458
86   0.370002 1.423 1.96e-02 0.2738   *
87  -0.065519 1.080 6.17e-04 0.0322
88  -0.147504 1.058 3.12e-03 0.0352
89   0.228672 1.144 7.50e-03 0.1016
90   0.184267 1.031 4.85e-03 0.0319
91   0.196842 1.048 5.54e-03 0.0410
92   0.082332 1.110 9.75e-04 0.0570
93  -0.523366 1.062 3.88e-02 0.1124
94  -0.104739 1.067 1.57e-03 0.0308
95   0.049475 1.110 3.52e-04 0.0535
96   0.321572 1.087 1.48e-02 0.0853
97   0.139300 1.068 2.78e-03 0.0386
98  -0.073807 1.077 7.83e-04 0.0313
99   0.302320 1.110 1.31e-02 0.0939
100 -0.037724 1.107 2.05e-04 0.0506
101 -0.318349 0.961 1.43e-02 0.0383
102  0.147387 1.072 3.11e-03 0.0425
103 -0.089421 1.094 1.15e-03 0.0458
104  0.051455 1.099 3.81e-04 0.0447
105 -0.246339 1.054 8.67e-03 0.0540
106  0.082443 1.096 9.77e-04 0.0467
107 -0.267046 1.085 1.02e-02 0.0738
108 -0.194358 1.013 5.38e-03 0.0282
109  0.250828 1.025 8.96e-03 0.0433
110 -0.186071 1.043 4.95e-03 0.0368
111  0.012704 1.146 2.32e-05 0.0813
112 -0.367214 0.981 1.91e-02 0.0525
113 -0.256381 1.152 9.42e-03 0.1104
114 -0.000776 1.196 8.67e-08 0.1201   *
115 -0.487851 0.993 3.36e-02 0.0792
116 -0.290419 1.186 1.21e-02 0.1368   *
117 -0.371635 0.972 1.95e-02 0.0507
118  0.014011 1.109 2.82e-05 0.0507
119  0.238913 1.080 8.17e-03 0.0652
120 -0.042591 1.143 2.61e-04 0.0800
121  0.335410 0.991 1.59e-02 0.0492
122  0.004131 1.092 2.46e-06 0.0359
123  0.210910 1.043 6.35e-03 0.0418
124  0.083794 1.171 1.01e-03 0.1036   *
```

```
125 -0.192263 1.099 5.30e-03 0.0673
126 -0.010417 1.161 1.56e-05 0.0936   *
127 -0.142793 1.105 2.93e-03 0.0621
128 -0.102747 1.118 1.52e-03 0.0660
129  0.260756 1.099 9.73e-03 0.0801
130  0.232762 1.113 7.76e-03 0.0826
131 -0.204646 1.274 6.02e-03 0.1817   *
132 -0.108854 1.074 1.70e-03 0.0360
133  0.028442 1.225 1.16e-04 0.1406   *
134 -0.657575 1.041 6.09e-02 0.1290
135  0.091005 1.156 1.19e-03 0.0936   *
136  0.100438 1.169 1.45e-03 0.1037   *
137  0.709716 0.834 6.93e-02 0.0753   *
138  0.010717 1.100 1.65e-05 0.0432
139 -0.059546 1.267 5.10e-04 0.1701   *
140  0.039086 1.112 2.20e-04 0.0548
141 -0.120572 1.280 2.09e-03 0.1803   *
142  0.176127 1.126 4.45e-03 0.0818
143  0.019659 1.114 5.56e-05 0.0558
144  0.088997 1.326 1.14e-03 0.2076   *
```

From all the and the table above, we the index of the observations which may affect the regression the most are $37, 86, 134, 137, 144$. So we discard them.

```
#--Old data---
old.dbts=dbts
#--outlier discarded data---
dbts=dbts[-c(37,86,134,137,144),]
```

Chapter 7

Autocorrelation detection.

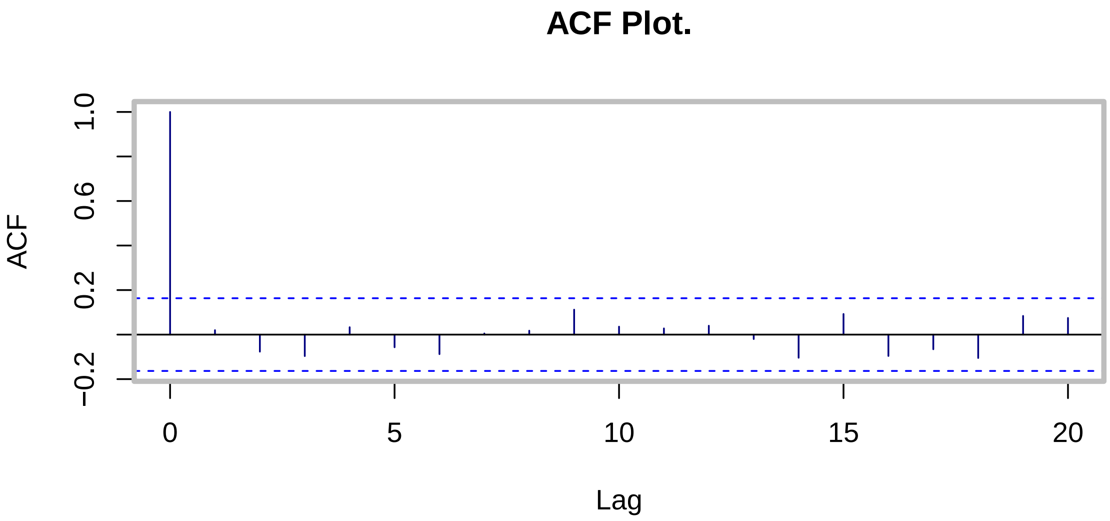
7.1 ACF and PACF Plots.

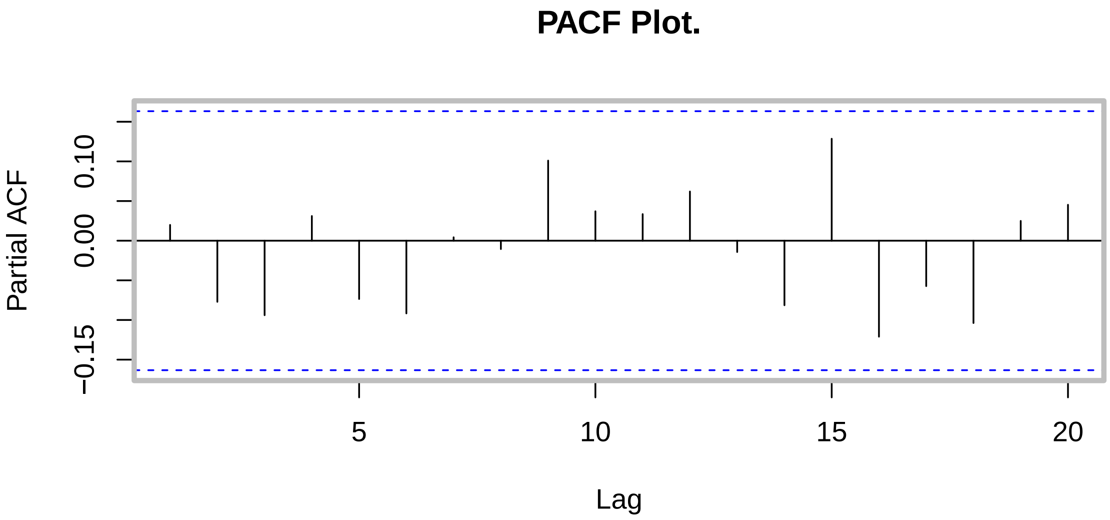
To check whether the residuals are autocorrelated or not, we have the autocorrelation function at lag h as,

$$\rho_e\left(h\right)=\frac{\gamma_e\left(h\right)}{\gamma_e\left(0\right)}=\mathrm{Corr}\left(e_{t+h},e_t\right).$$

We plot this for different lags. Also we plot the partial autocorrelations.

```
par(mfrow=c(2,1))
acf(residuals(lmodel1),lag.max = 20,main="ACF Plot.",col="navyblue")
box(lwd=3,col="grey")
pacf(residuals(lmodel1),lag.max = 20,main="PACF Plot.")
box(lwd=3,col="grey")
```





The ACF and PACF plots behave abruptly, neither it would be an AR or MA process. However, we perform the Durbin-Watson test for presence of autocorrelation.

```
dw=durbinWatsonTest(lmodel1,max.lag = 1)
dw

 lag Autocorrelation D-W Statistic p-value
   1      0.01991714      1.957266   0.682
```

```
Alternative hypothesis: rho != 0
```

The $p-$value for this test is 0.63. At level $\alpha = 0.05$, we have to accept the null hypothesis that $H_0 : \rho = 0$. Autocorrelation mayn't be present here.

Chapter 8

Shrinkage Methods.

8.1 Ridge Regression.

The ridge estimator for $\boldsymbol{\beta}$ will be as follows,

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{\boldsymbol{ridge}} = \left(X'X + \lambda I_p\right)^{-1} X'\boldsymbol{y}$$

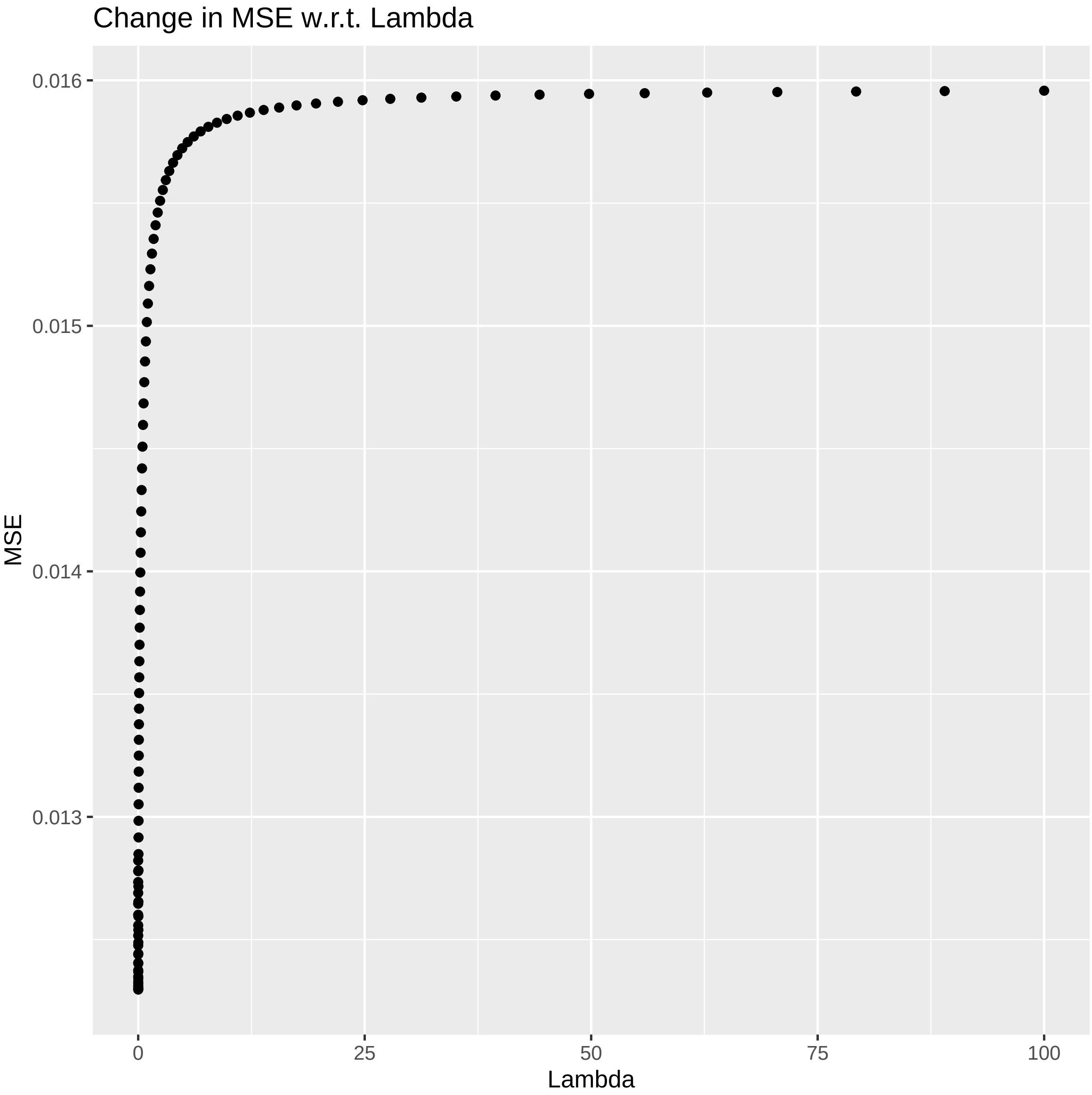
Firstly, we see how does MSE behaves w.r.t. λ.

```
X=dbts[-1]
lambda=10^seq(2,-3,length=100) #--seq. of auxiliary values---
#--Ridge Regression Model---
ridge.mod=glmnet(X,dbts$relwt,alpha=0,lambda = lambda)
summary(ridge.mod)

          Length Class     Mode
a0        100    -none-    numeric
beta      500    dgCMatrix S4
df        100    -none-    numeric
dim         2    -none-    numeric
lambda    100    -none-    numeric
dev.ratio 100    -none-    numeric
nulldev     1    -none-    numeric
npasses     1    -none-    numeric
jerr        1    -none-    numeric
offset      1    -none-    logical
call        5    -none-    call
nobs        1    -none-    numeric

newx=as.matrix(X,nc=5)
newx=apply(newx,2,as.numeric)
```

```
#---Calculating MSE for different values of lambda---
mse=NULL
pred=predict(ridge.mod,s=lambda,newx = newx)
mean((pred-dbts$relwt)**2)
```

```
[1] 0.01420525
```

```
for(l in 1:length(lambda))
{
        mse[l]=mean((pred[,l]-dbts$relwt)^2)
}
ggplot(data.frame(x=lambda,y=mse),aes(x=x,y=y))+
geom_point()+labs(x="Lambda",y="MSE",title="Change in MSE w.r.t. Lambda")
```

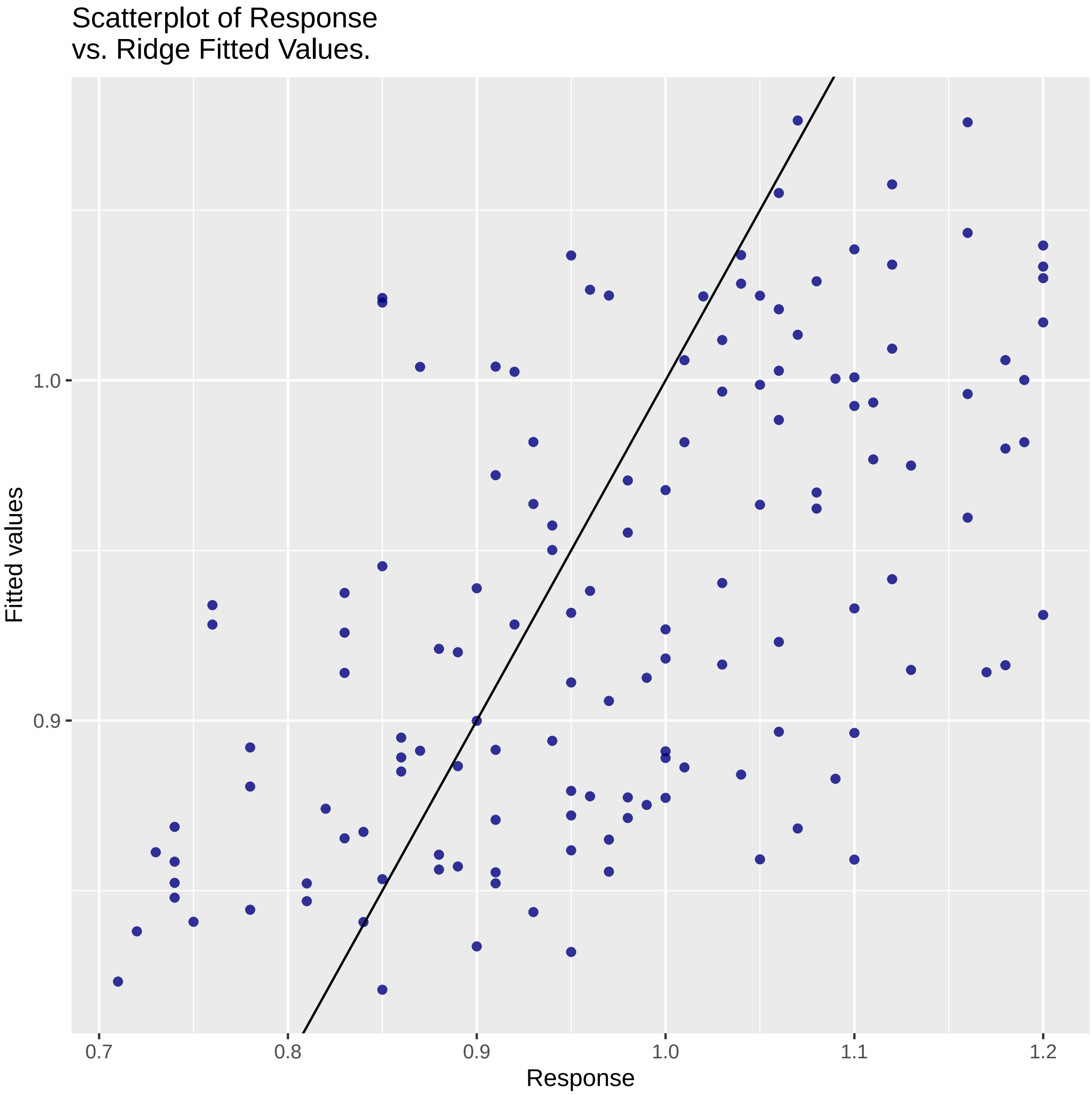


From the curve we see that the MSE more or less increases as λ does. To find the optimum value of λ, we apply Cross Validation,

```
ridge.cv=cv.glmnet(newx,dbts$relwt,alpha=0)
#---Optimum value of Lambda---
cv.lam=ridge.cv$lambda.min
#---Performing Ridge with that optimum value--
ridge.min=glmnet(X,dbts$relwt,alpha=0,lambda=cv.lam)
ridge.min


Call:  glmnet(x = X, y = dbts$relwt, alpha = 0, lambda = cv.lam)

  Df  %Dev   Lambda
1  5 39.42 0.005172
```

Via Cross Validation, the optimum value of λ, say λ_{opt} is about 0.005. Now, we plot $\boldsymbol{y}$ vs. $X\hat{\boldsymbol{\beta}}^{\boldsymbol{ridge}}_{\boldsymbol{\lambda_{opt}}}$ to see how close they are.

```
pred.cv=predict(ridge.min,s=cv.lam,newx=newx)
cv=ggplot()+geom_point(aes(x=dbts$relwt,y=pred.cv),
col="navyblue",alpha=0.8)+geom_abline(slope = 1,intercept = 0)+
labs(x="Response",y="Fitted values",title="Scatterplot of Response
vs. Ridge Fitted Values.")
cv
```



So, squared correlation between responses and fitted values for Ridge Regression will be,

```
ridge=cor(pred.cv,dbts$relwt)
ridge^2

           [,1]
s1 0.3425501
```

Very close to R^2 of the full model.

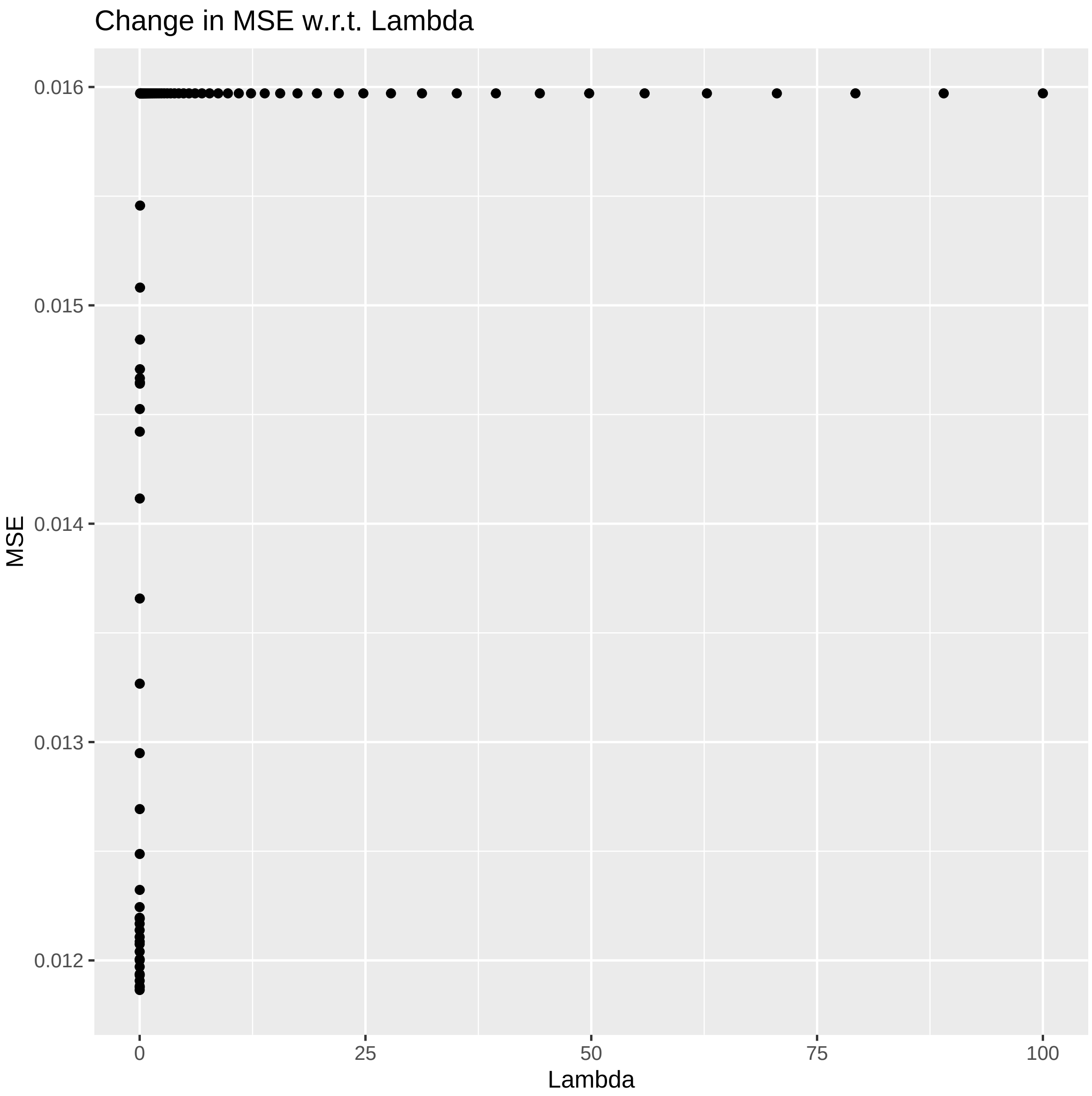
8.2 LASSO.

The LASSO estimates are defined as,

$$\hat{\boldsymbol{\beta}}^{\boldsymbol{lasso}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \underbrace{\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\boldsymbol{\beta}\|_1}_{\text{Penalty}} \right\}$$

Firstly, we see how does MSE behaves w.r.t. λ.

```
X=dbts[-1]
lambda=10^seq(2,-3,length=100) #--seq. of auxiliary values---
#-- LASSO Model---
lasso.mod=glmnet(X,dbts$relwt,alpha=1,lambda = lambda)
summary(lasso.mod)

          Length Class     Mode
a0        100    -none-    numeric
beta      500    dgCMatrix S4
df        100    -none-    numeric
dim         2    -none-    numeric
lambda    100    -none-    numeric
dev.ratio 100    -none-    numeric
nulldev     1    -none-    numeric
npasses     1    -none-    numeric
jerr        1    -none-    numeric
offset      1    -none-    logical
call        5    -none-    call
nobs        1    -none-    numeric

newx=as.matrix(X,nc=5)
newx=apply(newx,2,as.numeric)

#---Calculating MSE for different values of lambda---
mse=NULL
pred=predict(lasso.mod,s=lambda,newx = newx)
mean((pred-dbts$relwt)**2)

[1] 0.01495179

for(l in 1:length(lambda))
{
        mse[l]=mean((pred[,l]-dbts$relwt)^2)
}
ggplot(data.frame(x=lambda,y=mse),aes(x=x,y=y))+
geom_point()+labs(x="Lambda",y="MSE",title="Change in MSE w.r.t. Lambda")
```

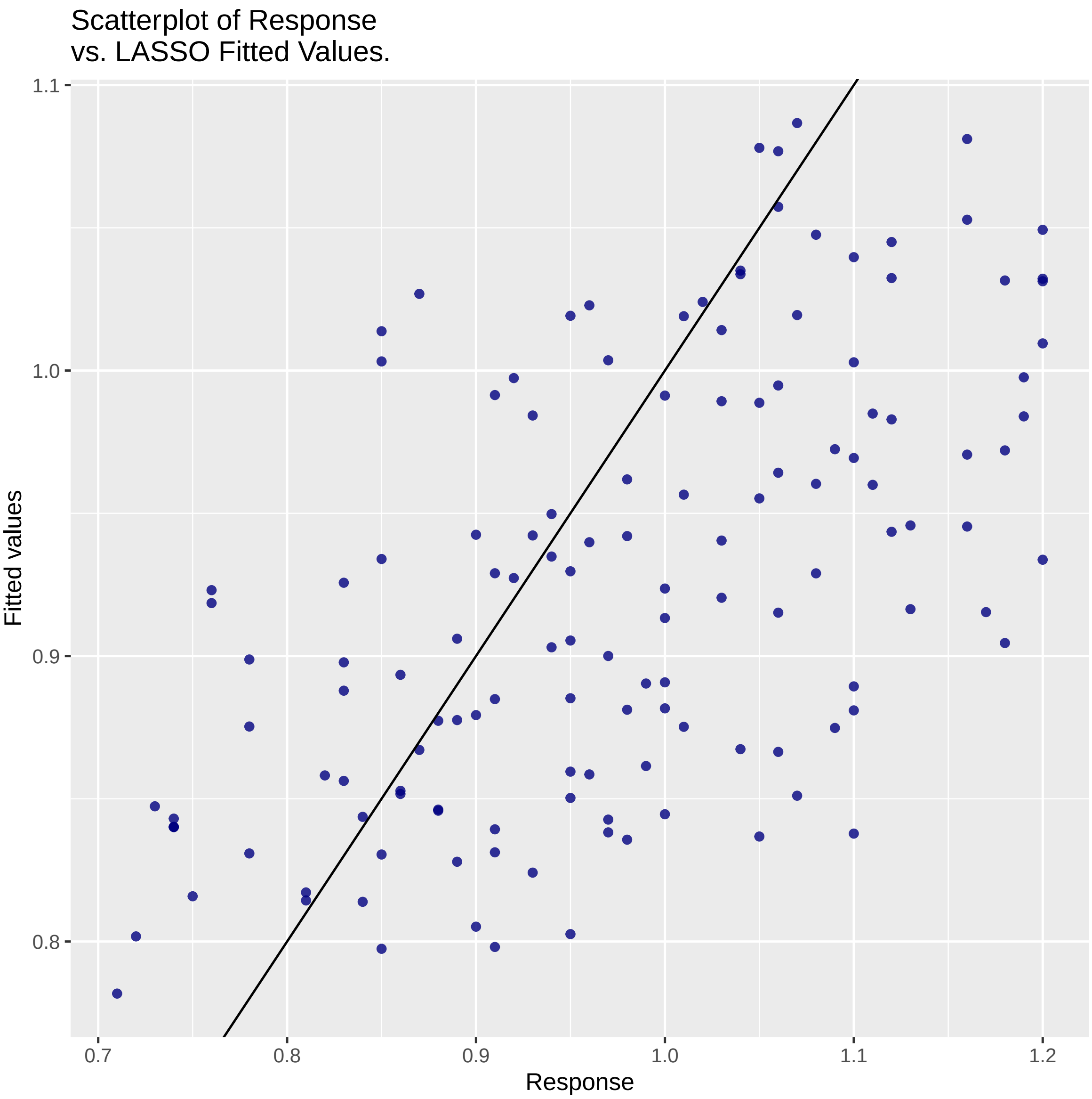


From the curve we see that the MSE more or less increases as λ does. To find the optimum value of λ, we apply Cross Validation,

```
lasso.cv=cv.glmnet(newx,dbts$relwt,alpha=1)
#---Optimum value of Lambda---
cv.lam=lasso.cv$lambda.min
#---Performing LASSO with that optimum value--
lasso.min=glmnet(X,dbts$relwt,alpha=1,lambda=cv.lam)
lasso.min


Call:  glmnet(x = X, y = dbts$relwt, alpha = 1, lambda = cv.lam)

  Df  %Dev    Lambda
1  5 41.34 5.294e-05
```

Via Cross Validation, the optimum value of λ, say λ_{opt} is about 0.0011. Now, we plot $\boldsymbol{y}$ vs. $X\hat{\boldsymbol{\beta}}^{\boldsymbol{lasso}}_{\boldsymbol{\lambda_{opt}}}$ to see how close they are.

```
pred.cv=predict(lasso.min,s=cv.lam,newx=newx)
cv=ggplot()+geom_point(aes(x=dbts$relwt,y=pred.cv),
col="navyblue",alpha=0.8)+geom_abline(slope = 1,intercept = 0)+
labs(x="Response",y="Fitted values",title="Scatterplot of Response
vs. LASSO Fitted Values.")
cv
```



So, squared correlation between responses and fitted values for LASSO Regression will be,

```
lasso=cor(pred.cv,dbts$relwt)
lasso^2

        [,1]
s1 0.343535
```

Chapter 9

Presence of Multicollinearity.

9.1 Detecting dependency among Covariates.

Variance covariance matrix of quantitative covariates. Also, the correlation matrix and eigenvalues of $(X'X)$ is as follows.

```
#--mat of quantitative covariates---
dbts.qcov=dbts[,-c(1,ncol(dbts))]
cor(dbts.qcov)

           glufast    glutest       sspg   instest
glufast  1.0000000  0.9647115 -0.4141934 0.7208502
glutest  0.9647115  1.0000000 -0.3500347 0.7801762
sspg    -0.4141934 -0.3500347  1.0000000 0.0452164
instest  0.7208502  0.7801762  0.0452164 1.0000000

eigen(crossprod(as.matrix(dbts.qcov)))

eigen() decomposition
$values
[1] 63043262.13  3626802.54   401921.80    40479.52

$vectors
           [,1]        [,2]       [,3]        [,4]
[1,] -0.1956474  0.03797932 -0.1196253  0.97260961
[2,] -0.9054798  0.28160473 -0.2278025 -0.22115850
[3,] -0.2252126 -0.94453606 -0.2360678 -0.03745502
[4,] -0.3018404 -0.16464498  0.9370527  0.06096379
```

9.2 VIF.

The formula for calculating VIF for j^{th} covariate is as follows,

$$VIF_j = \frac{1}{1 - R_j^2}.$$

R_j^2 is the Multiple R squared when j^{th}covariate is regressed on other covariates in case of scaled and centered model.

```
vif(lmodel1)

             GVIF Df GVIF^(1/(2*Df))
glufast 19.882225  1        4.458949
glutest 29.346064  1        5.417201
sspg     1.686218  1        1.298544
instest  3.619047  1        1.902379
group    8.693669  2        1.717121
```

It indicates that there is a linear relationship between glutest and glufast which supports our previous diagnostics. Later we'll deal with collinearity.

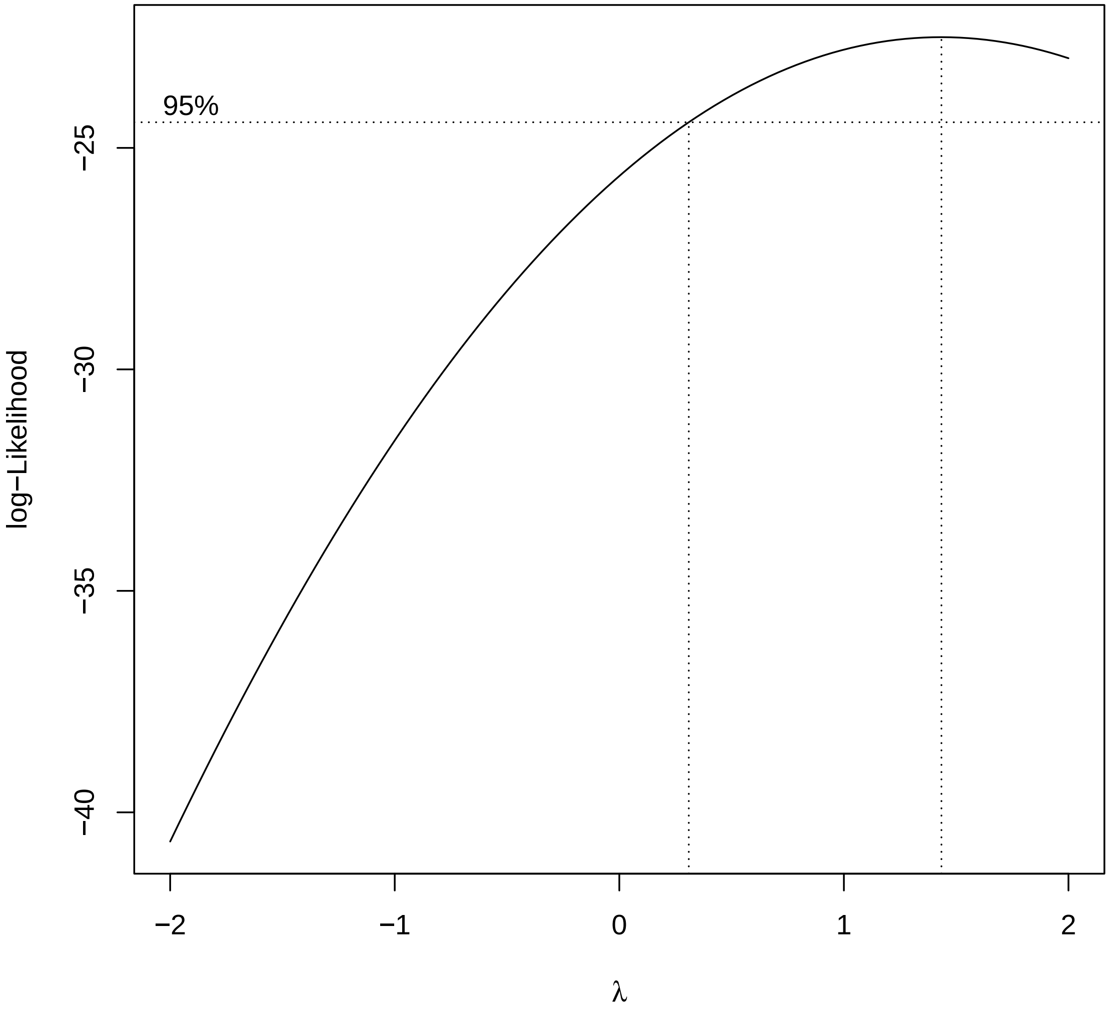
Chapter 10

Transforming the Covariates.

10.1 Box-Cox Transformation.

We perform the Box-Cox transformation on response and storing the model into the object '**model.bc**'.

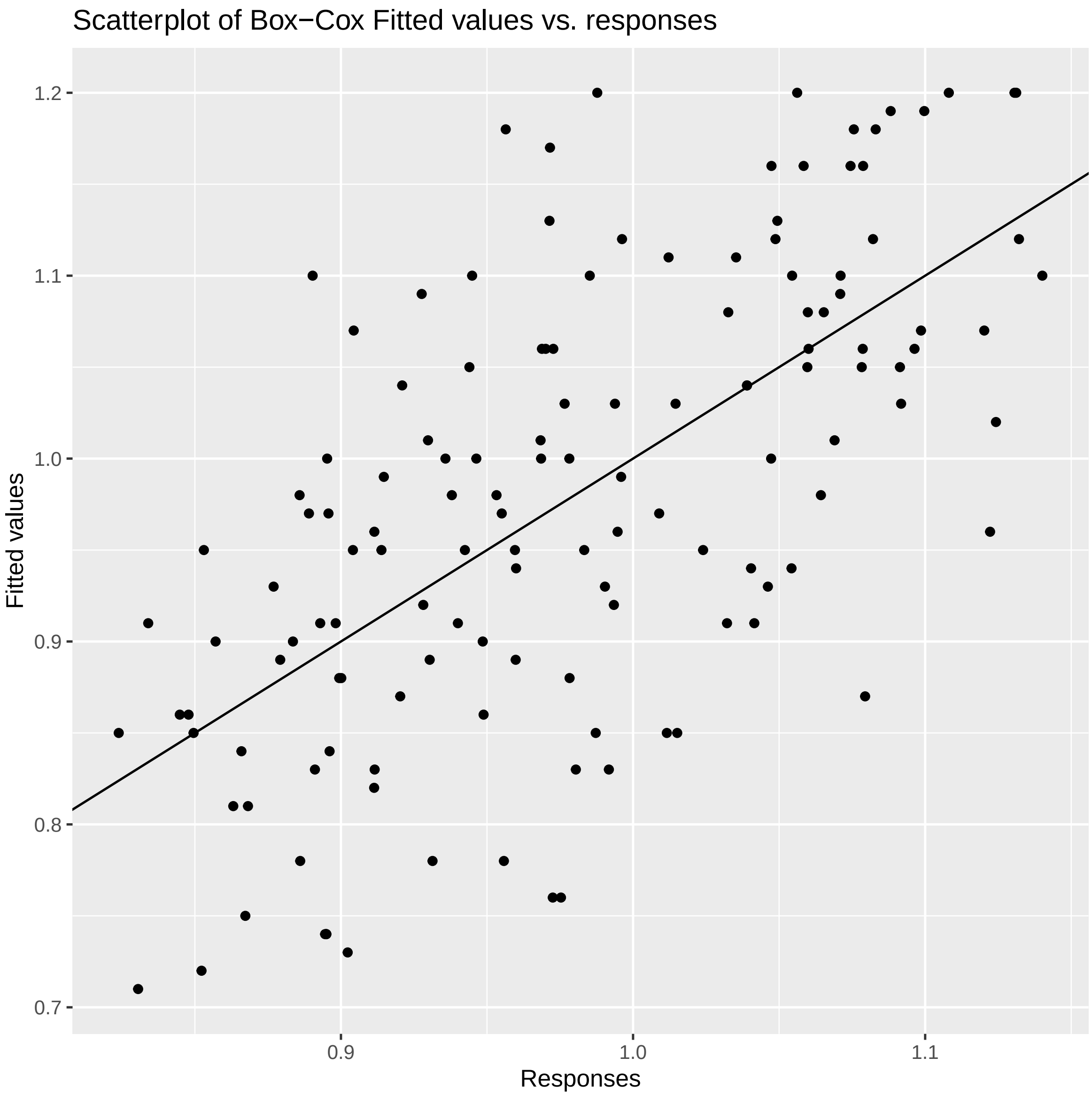
```
bc=boxcox(lmodel1)
```



```
#--Optimum value of lambda for which loglikelihood is maximum.--
lambda.bc=bc$x[which.max(bc$y)]
res.bc=(dbts$relwt**lambda.bc-1)/lambda.bc
#--BoxCox transformed model---
model.bc=lm(res.bc~.,data=dbts[-1])
summary(model.bc)
```

```
Call:
lm(formula = res.bc ~ ., data = dbts[-1])

Residuals:
      Min        1Q    Median        3Q       Max 
-0.207032 -0.061681  0.007383  0.073757  0.229983 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.0454541  0.0339180  -1.340 0.182509    
glufast      0.0007849  0.0006534   1.201 0.231768    
glutest     -0.0005073  0.0001563  -3.244 0.001491 ** 
sspg        -0.0001898  0.0001016  -1.868 0.063954 .  
instest      0.0010695  0.0001583   6.755 4.15e-10 ***
group2       0.0667485  0.0533365   1.251 0.212980    
group1       0.1051031  0.0296118   3.549 0.000536 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09769 on 132 degrees of freedom
Multiple R-squared:  0.4157,Adjusted R-squared:  0.3891 
F-statistic: 15.65 on 6 and 132 DF,  p-value: 1.63e-13
```

```
pred.bc=(predict(model.bc)*lambda.bc+1)**(1/lambda.bc)
ggbc=ggplot()+geom_point(aes(pred.bc,dbts$relwt))+geom_abline(intercept=0,slope=1)+
labs(title="Scatterplot of Box-Cox Fitted values vs. responses",x="Responses",y="Fitted values")
ggbc
```



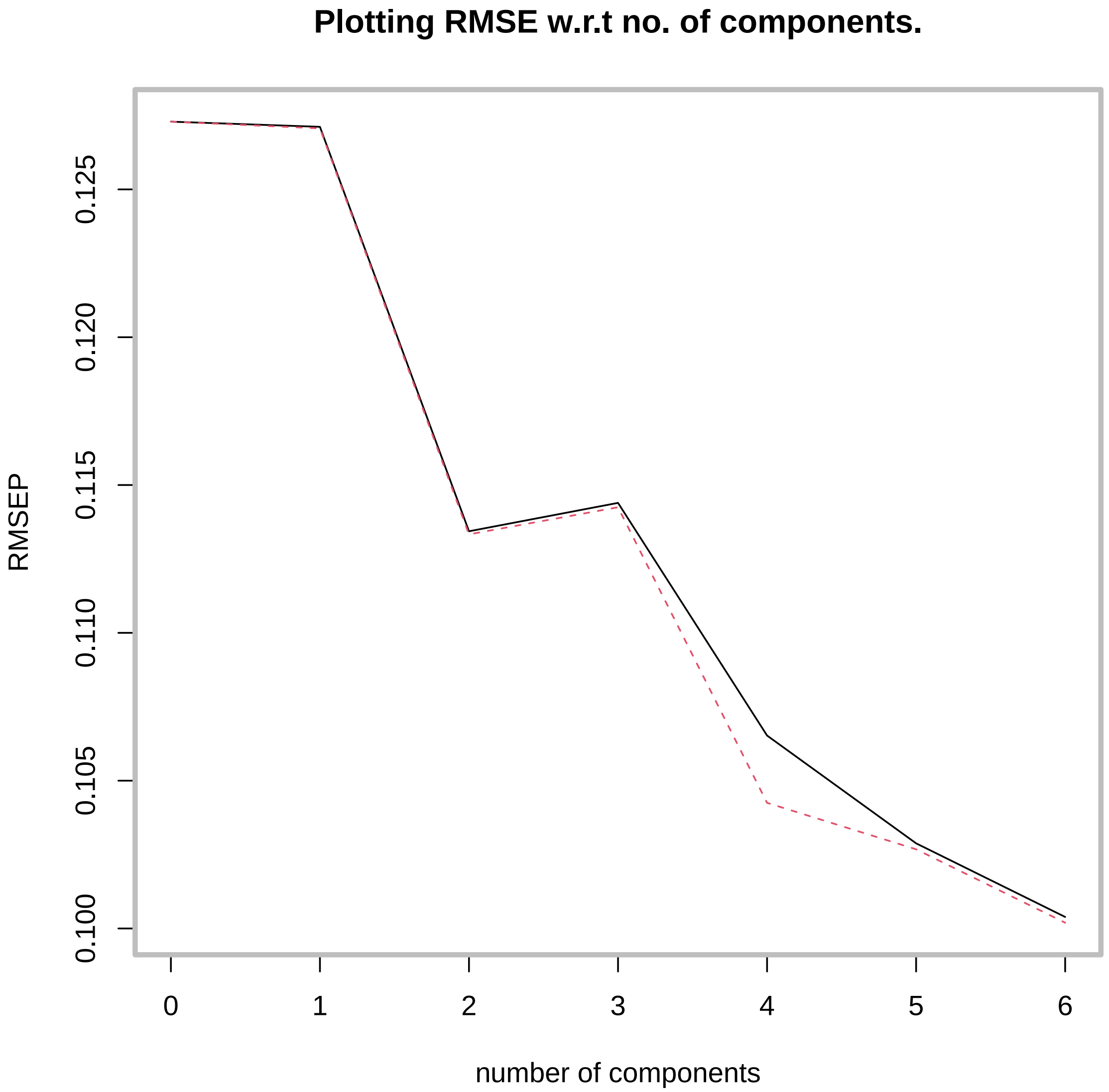
```
bc.cor=cor(pred.bc,dbts$relwt)^2
bc.cor

[1] 0.414665
```

This performed well w.r.t the full model.

10.2 Principal Component Regression (PCR).

The Principal Components Regression approach involves constructing the first M principal components, $Z_1, Z_2, \ldots, Z_M$ and then using these components as the predictors in a linear regression model that is fit using least squares. We fit a PCR model storing into the object **'model.pc'** .

```
model.pc=pcr(relwt~.,data=dbts,validation="CV",scale=T,centre=T)
summary(model.pc)

Data:  X dimension: 139 6
Y dimension: 139 1
Fit method: svdpc
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV          0.1273   0.1271   0.1134   0.1144   0.1065   0.1029   0.1004
adjCV       0.1273   0.1271   0.1133   0.1142   0.1042   0.1027   0.1002

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X      60.0859    85.18    94.24    97.11     99.7    100.0
relwt   0.4119    22.18    22.26    36.16     38.5     41.4

validationplot(model.pc,main="Plotting RMSE w.r.t no. of components.")
box(lwd=3,col="grey")
```



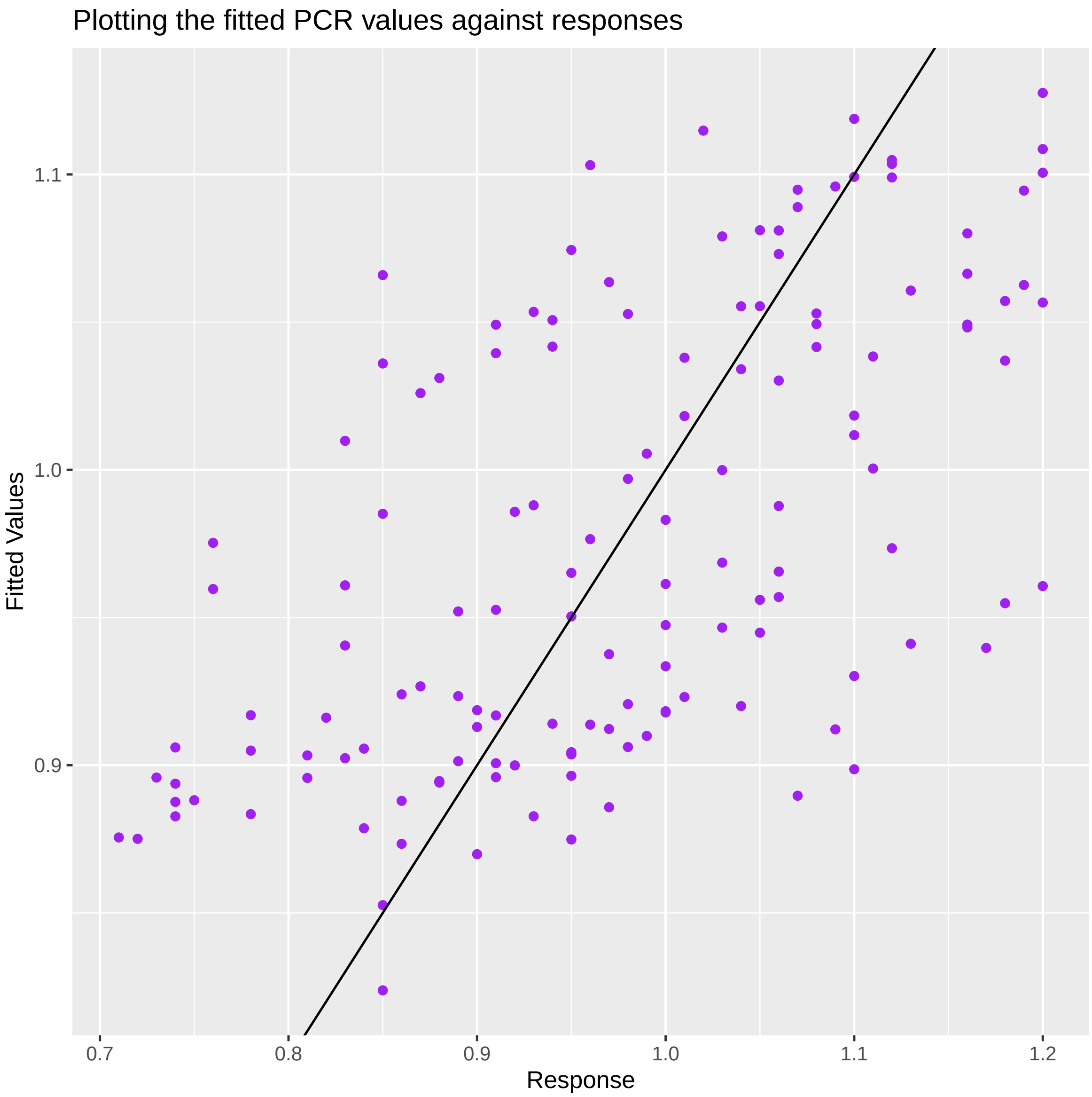
Here the RMSE's for different components are obtained by 10-fold Cross Validation. From above plot we see that the RMSE is least when $M = 6$, which is barely fewer than $M = 7$, which amounts to performing least squares., because when all of the components are used in PCR no diemnsion reduction occurs. However from the plot we also see that the cv error is roughly same only one component is included in the model. This suggests that a model that uses just a small number of components might suffice.

Also from the above output, only 4 principal components captures about 97.11% of the variation. Hence we might attempt to do the same analysis using 4 components.

```
#---Another PCR Model with 4 components---
model.pc2=pcr(relwt~.,data=dbts,validation="CV",scale=T,centre=T,ncomp=4)
summary(model.pc2)

Data:   X dimension: 139 6
Y dimension: 139 1
Fit method: svdpc
Number of components considered: 4

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps
CV          0.1273   0.1278   0.1132   0.1137   0.1030
adjCV       0.1273   0.1277   0.1132   0.1136   0.1019

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps
X      60.0859    85.18    94.24    97.11
relwt   0.4119    22.18    22.26    36.16
```

Now, we plot the fitted values against the response and check squared correlation among them.

```
pred.pc=predict(model.pc2,ncomp=4)
ggpcr=ggplot()+geom_point(col="purple",aes(x=dbts$relwt,y=pred.pc))+
geom_abline(intercept=0,slope=1)+labs(y="Fitted Values",x="Response"
,title="Plotting the fitted PCR values against responses")
ggpcr
```



```
#--Estimated MSE---
mean((pred.pc-dbts$relwt)**2)

[1] 0.01019545

pcrcor=cor(pred.pc,dbts$relwt)^2
pcrcor

[1] 0.3616197
```

The squared correlation between Fitted values and responses here is very close to as we got Multiple R^2 from the full model.

Chapter 11

Model Selection.

We apply various procedures like Forward, Backward and Stepwise selection to select which variables to include in order to get a good model, and how to use the available variables to construct a good predictor.

11.1 Forward Selection.

```
stepAIC(lm(relwt~.,data=old.dbts),direction='forward')

Start:  AIC=-644.64
relwt ~ glufast + glutest + sspg + instest + group

Call:
lm(formula = relwt ~ glufast + glutest + sspg + instest + group, 
    data = old.dbts)

Coefficients:
(Intercept)      glufast      glutest         sspg      instest       group2  
  0.9397650    0.0007723   -0.0004704   -0.0001156    0.0009793    0.0683806  
     group1  
  0.1003792  
```

11.2 Backward Selection.

```
stepAIC(lm(relwt~.,data=dbts),direction='backward')

Start:  AIC=-635.32
relwt ~ glufast + glutest + sspg + instest + group
```

```
          Df Sum of Sq    RSS     AIC
- glufast  1   0.01461 1.3156 -635.77
<none>                 1.3010 -635.32
- sspg     1   0.03401 1.3350 -633.73
- glutest  1   0.10390 1.4048 -626.64
- group    2   0.13542 1.4364 -625.56
- instest  1   0.44452 1.7455 -596.47

Step:  AIC=-635.77
relwt ~ glutest + sspg + instest + group

          Df Sum of Sq    RSS     AIC
<none>                 1.3156 -635.77
- sspg     1   0.04020 1.3558 -633.59
- group    2   0.12444 1.4400 -627.21
- glutest  1   0.22261 1.5382 -616.04
- instest  1   0.47219 1.7877 -595.14

Call:
lm(formula = relwt ~ glutest + sspg + instest + group, data = dbts)

Coefficients:
(Intercept)      glutest         sspg      instest       group2       group1
  0.9631595   -0.0003436   -0.0002070    0.0011051    0.0438685    0.0880131
```

11.3 Stepwise Selection.

```
stepAIC(lm(relwt~.,data=dbts),direction='both')

Start:  AIC=-635.32
relwt ~ glufast + glutest + sspg + instest + group

          Df Sum of Sq    RSS     AIC
- glufast  1   0.01461 1.3156 -635.77
<none>                 1.3010 -635.32
- sspg     1   0.03401 1.3350 -633.73
- glutest  1   0.10390 1.4048 -626.64
- group    2   0.13542 1.4364 -625.56
- instest  1   0.44452 1.7455 -596.47

Step:  AIC=-635.77
relwt ~ glutest + sspg + instest + group
```

```
          Df Sum of Sq    RSS     AIC
<none>                 1.3156 -635.77
+ glufast  1   0.01461 1.3010 -635.32
- sspg     1   0.04020 1.3558 -633.59
- group    2   0.12444 1.4400 -627.21
- glutest  1   0.22261 1.5382 -616.04
- instest  1   0.47219 1.7877 -595.14

Call:
lm(formula = relwt ~ glutest + sspg + instest + group, data = dbts)

Coefficients:
(Intercept)      glutest         sspg      instest       group2       group1
  0.9631595   -0.0003436   -0.0002070    0.0011051    0.0438685    0.0880131
```

In all the three methods, full model is selected. But earlier we found the covariates are highly collinear. Hence, we compare between different models as follows.

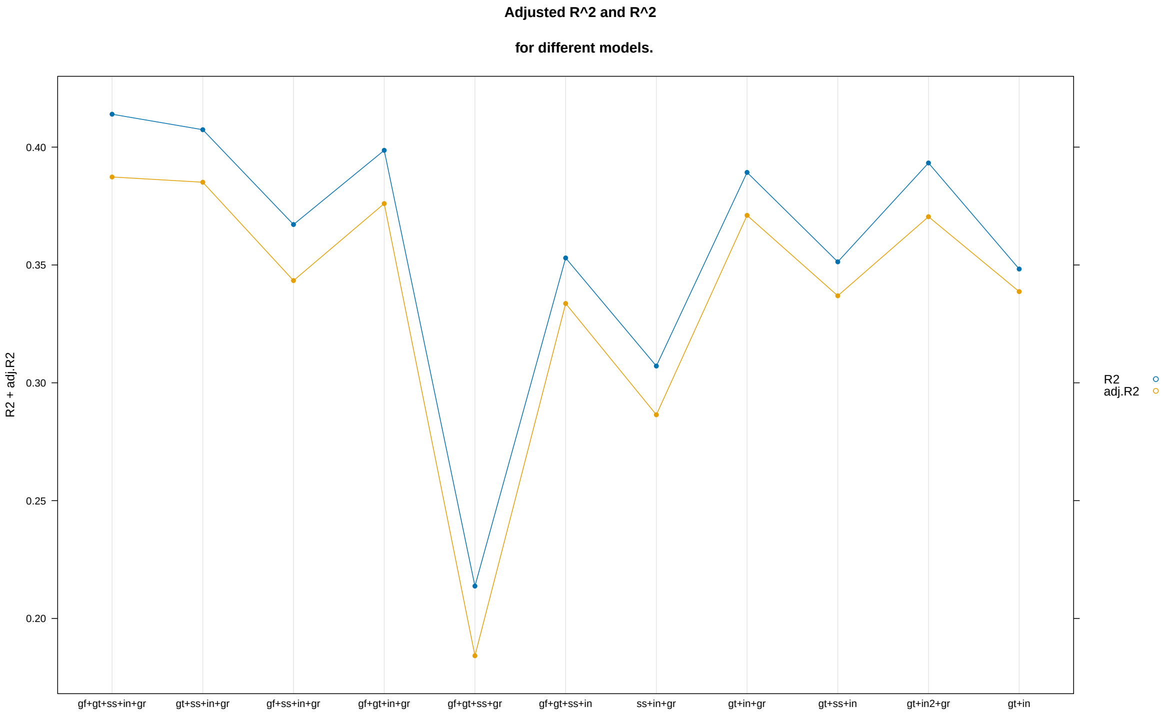
Next we consider all possible combination of covariates for constructing the model.

```
## models
fm<-list()
fm[['gf+gt+ss+in+gr']]<-lm(relwt~.,data=dbts)

fm[['gt+ss+in+gr']]<-lm(relwt~.-glufast,data=dbts)
fm[['gf+ss+in+gr']]<-lm(relwt~.-glutest,data=dbts)
fm[['gf+gt+in+gr']]<-lm(relwt~.-sspg,data=dbts)
fm[['gf+gt+ss+gr']]<-lm(relwt~.-instest,data=dbts)
fm[['gf+gt+ss+in']]<-lm(relwt~.-group,data=dbts)

fm[['ss+in+gr']]<-lm(relwt~sspg+instest+group,data=dbts)
fm[['gt+in+gr']]<-lm(relwt~glutest+instest+group,data=dbts)
fm[['gt+ss+in']]<-lm(relwt~glutest+sspg+instest,data=dbts)
fm[['gt+in2+gr']]<-lm(relwt~glutest+poly(instest,2,raw=T)+group,data=dbts)

fm[['gt+in']]<-lm(relwt~glutest+instest,data=dbts)

models<-factor(names(fm),levels=names(fm))
```

11.4 R^2 and Adjusted R^2.

```
## R2 and adjusted R2
R2<-sapply(fm,function(model)summary(model)$r.squared)
adj.R2<-sapply(fm,function(model)summary(model)$adj.r.squared)
```

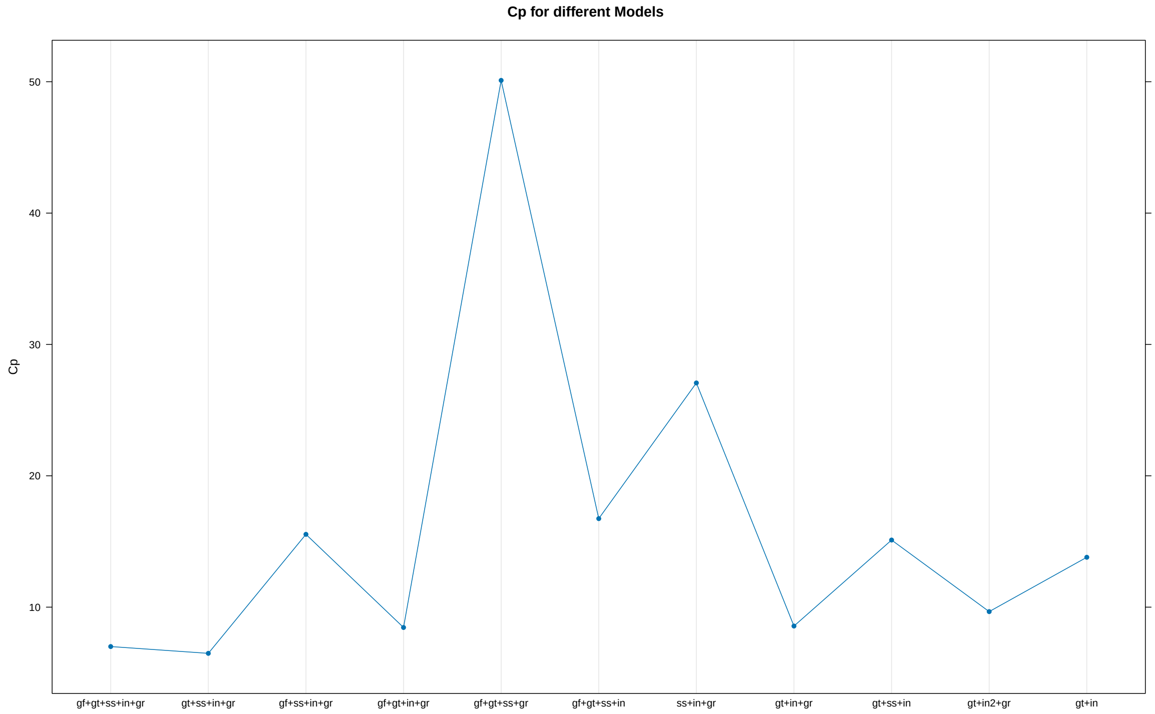
```
dotplot(R2+adj.R2~models,type='o',pch=16,main="Adjusted R^2 and R^2
\n for different models.",auto.key=list(space="right"))
```



From this plot we select the model “relwt ~ glutest + instest + group” .

11.5 Mallow’s C_p.

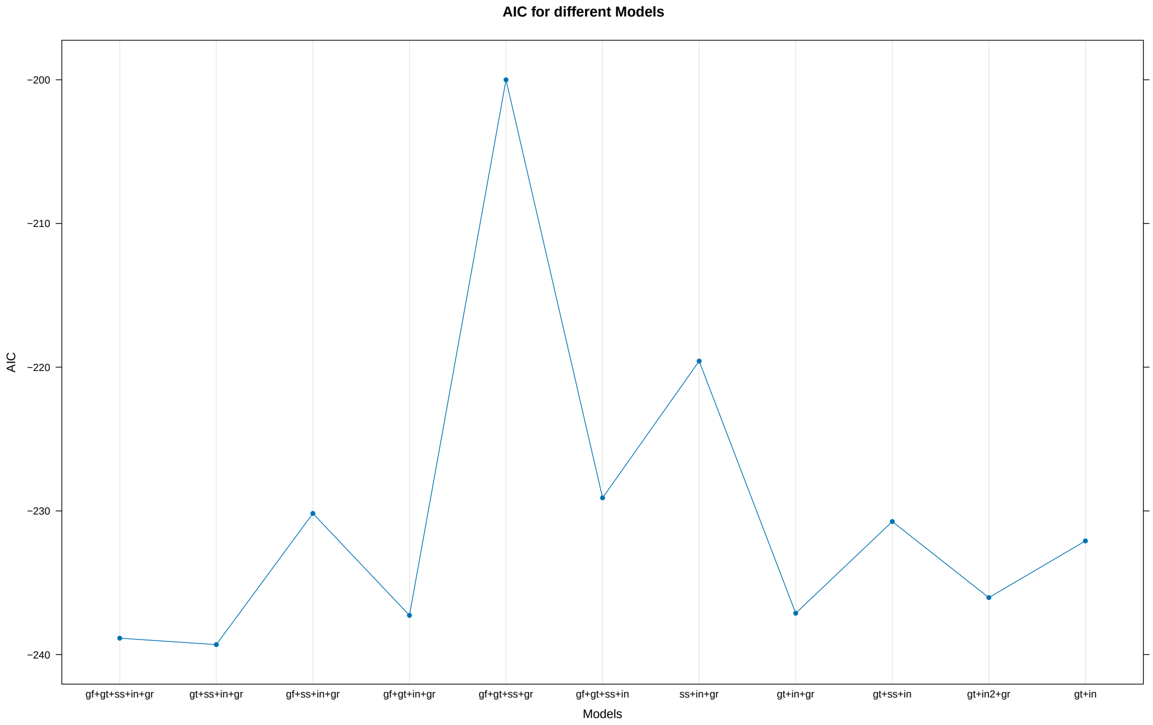
```
## Mallows Cp
sigma.sq<-summary(fm[['gf+gt+ss+in+gr']])$sigma**2 #for the big model
Cp <- sapply(fm, function(fit) extractAIC(fit, scale = sigma.sq)[2])
dotplot(Cp~models,type='o',pch=16,main="Cp for different Models")
```



From this plot we select the model “relwt ~ glutest + instest + sspg+group” .

11.6 AIC.

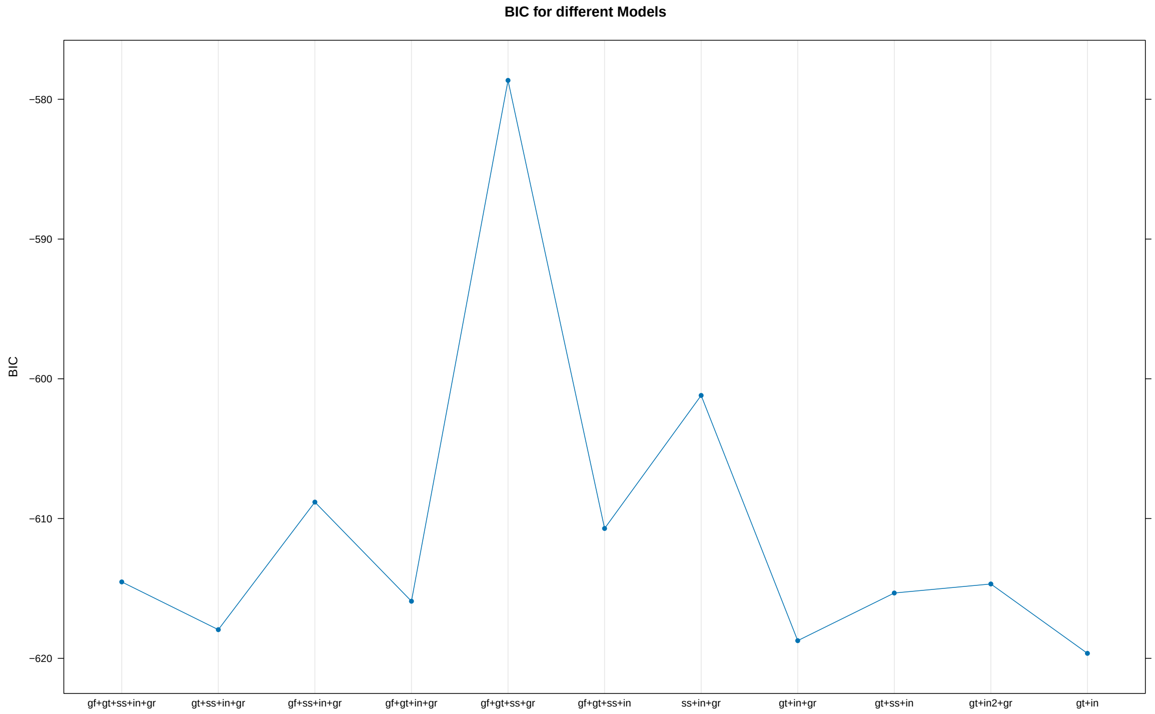
```
##AIC
AIC <- sapply(fm, function(fit) AIC(fit))
dotplot(AIC ~ models, type = "o", pch = 16,xlab="Models",main="AIC for different Models")
```



From this plot we select the model “relwt ~ glutest + instest +sspg+ group” .

11.7 BIC.

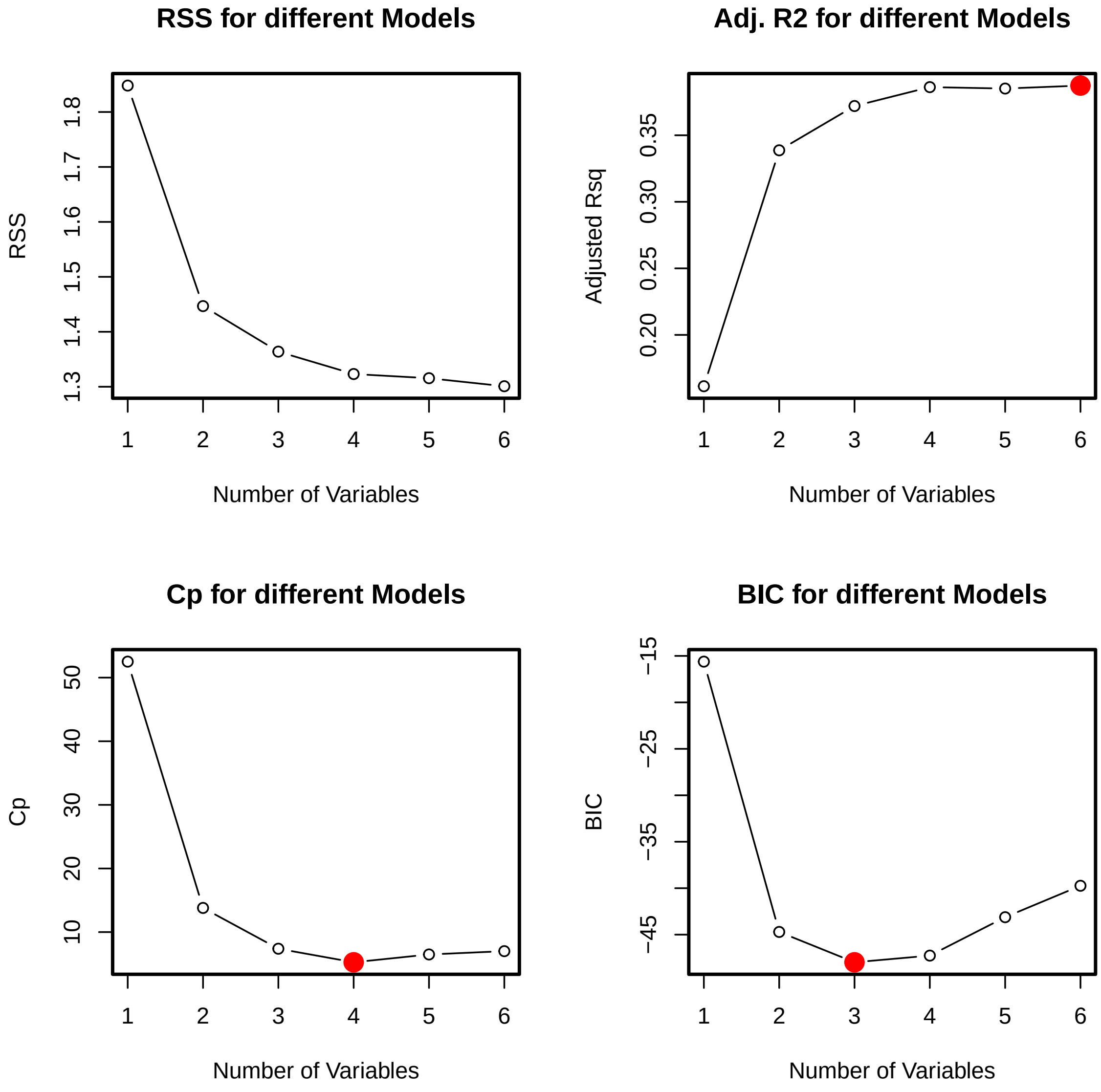
```
## BIC
BIC <- sapply(fm, function(fit) extractAIC(fit, k = log(n))[2])
dotplot(BIC ~ models, type = "o", pch = 16,main="BIC for different Models")
```



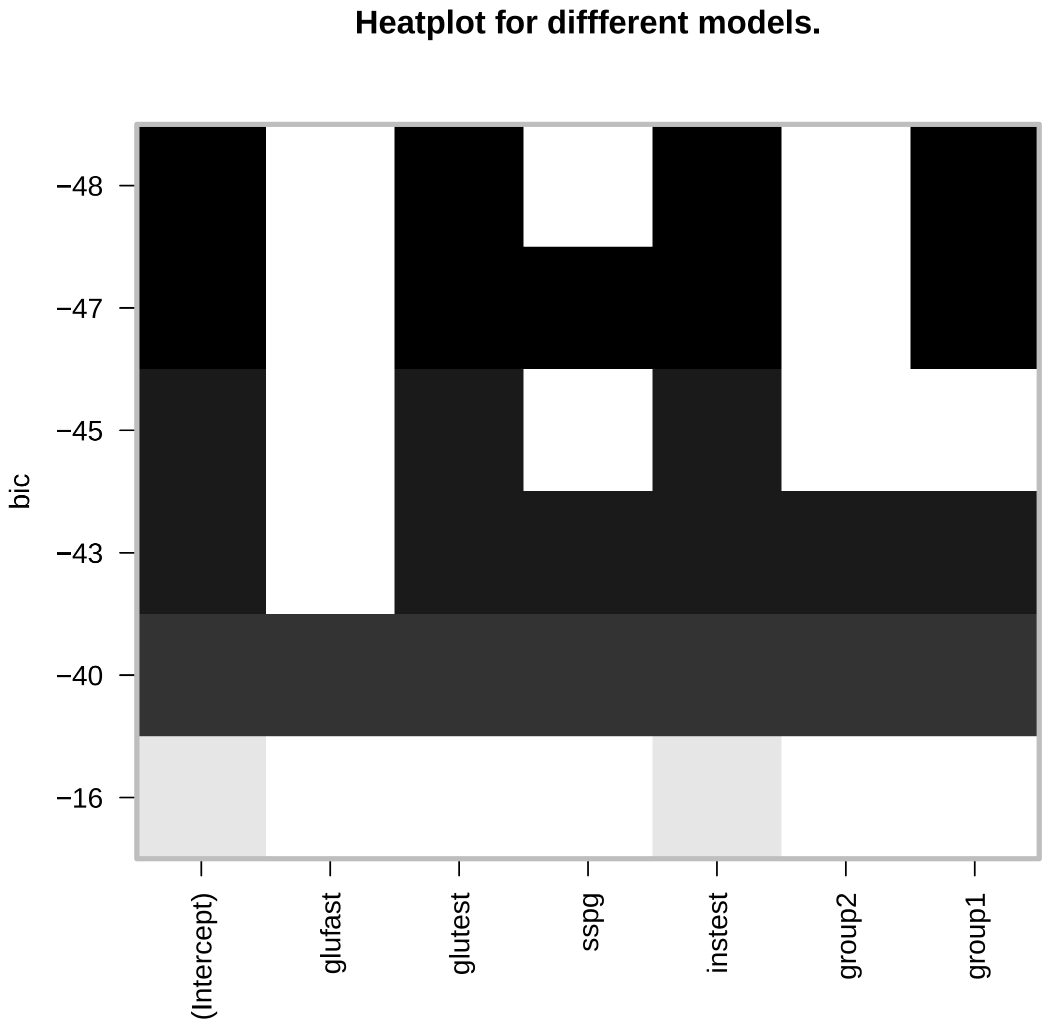
11.8 Some Additional Plot.

```
##----additional plots
reg.sub<-regsubsets(relwt~.,data=dbts)
reg.sum<-summary(reg.sub)

par(mfrow=c(2,2))
# rss
plot(reg.sum$rss,type='b',ylab='RSS',xlab='Number of Variables',
main="RSS for different Models");box(lwd=2)
# adj R2
plot(reg.sum$adjr2,type='b',ylab='Adjusted Rsq',xlab='Number of Variables',
main="Adj. R2 for different Models");box(lwd=2)
max<-which.max(reg.sum$adjr2)
points(max,reg.sum$adjr2[max],col='red',cex=2,pch=16)
# Cp
plot(reg.sum$cp,type='b',ylab='Cp',xlab='Number of Variables',
main="Cp for different Models");box(lwd=2)
min<-which.min(reg.sum$cp)
```

```
points(min,reg.sum$cp[min],col='red',cex=2,pch=16)
# bic
plot(reg.sum$bic,type='b',ylab='BIC',xlab='Number of Variables',
main="BIC for different Models");box(lwd=2)
min<-which.min(reg.sum$bic)
points(min,reg.sum$bic[min],col='red',cex=2,pch=16)
```



```
par(mfrow=c(1,1))
plot(reg.sub,scale='bic',main="Heatplot for diffferent models.");box(lwd=3,col="grey")
```



These plots gives us some interesting features,

- As the # covariates increases RSS always decreases and adjusted R^2 always increases. So, these criterions will always give full model as the best though we have seen there are highly collinear covariates.

- Mallow's C_p or BIC gives model with 3 covariates will be better.
- Also, plots of R^2, adjusted R^2 AIC, BIC gives the model with glutest, instest, group will be better for us.
- In the Heatplot, the top row of each plot contains a black square for each variable selected according to the optimal model associated with the statistic. It left 'glutest', 'instest' and 'group' in the model.

11.9 Final Model.

Hence in our final model, we'll take glutest, instest , sspg and group only. Let's see how the final model, **'fmodel'** works,

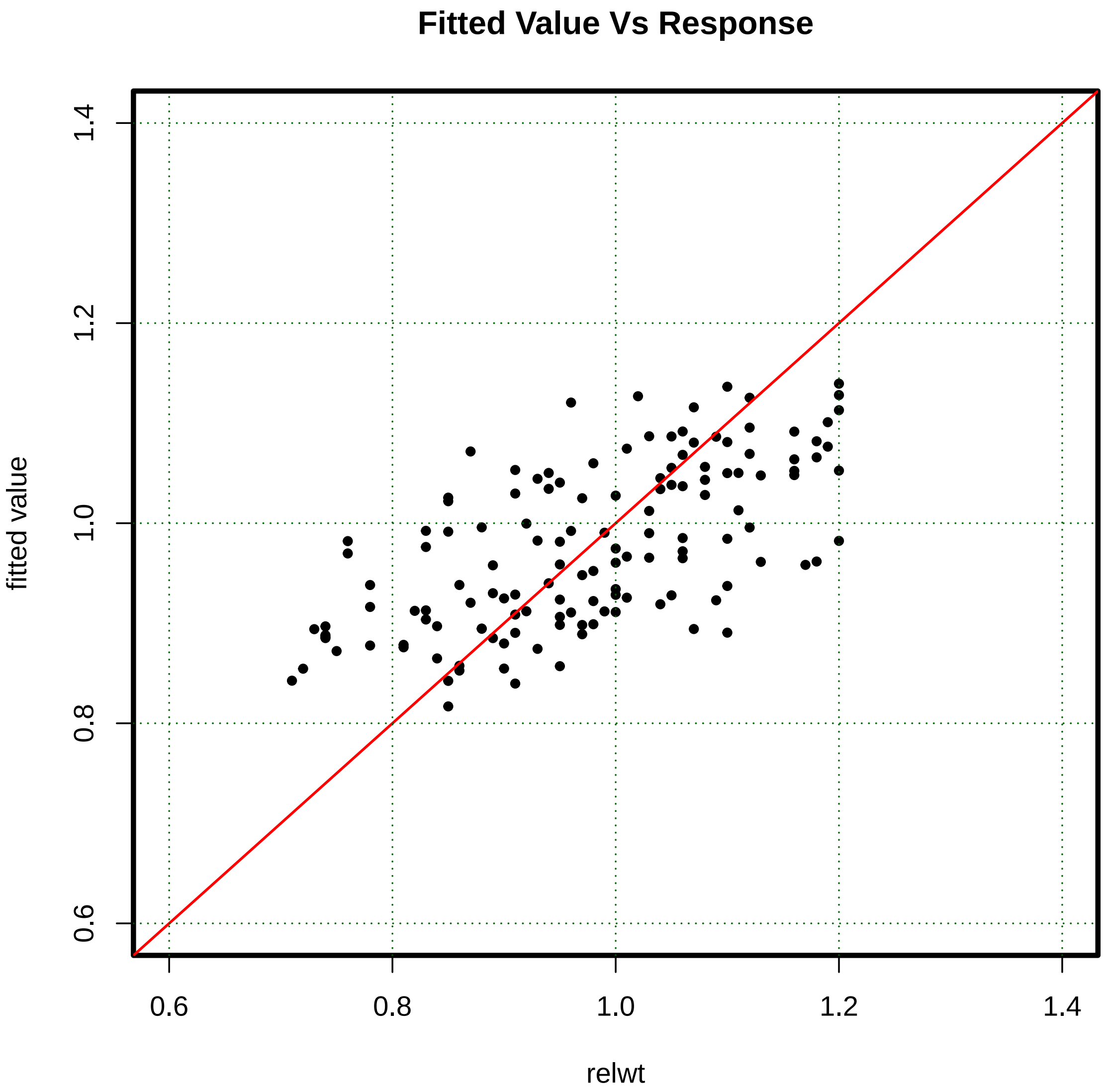
```
#final model
fmodel<-lm(relwt~glutest+instest+group+sspg,data=dbts)
summary(fmodel)


Call:
lm(formula = relwt ~ glutest + instest + group + sspg, data = dbts)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.222082 -0.068149  0.007328  0.070946  0.218304 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.632e-01  3.278e-02  29.382  < 2e-16 ***
glutest     -3.436e-04  7.243e-05  -4.744 5.33e-06 ***
instest      1.105e-03  1.599e-04   6.909 1.83e-10 ***
group2       4.387e-02  5.005e-02   0.876 0.382383    
group1       8.801e-02  2.581e-02   3.411 0.000859 ***
sspg        -2.070e-04  1.027e-04  -2.016 0.045824 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09946 on 133 degrees of freedom
Multiple R-squared:  0.4074,Adjusted R-squared:  0.3851 
F-statistic: 18.29 on 5 and 133 DF,  p-value: 8.725e-14
```

- Multiple R Squared is slightly better than the full model. In spite of that, this model would be better as it's free from collinearity. Now let's see whether our assumptions are valid or not.

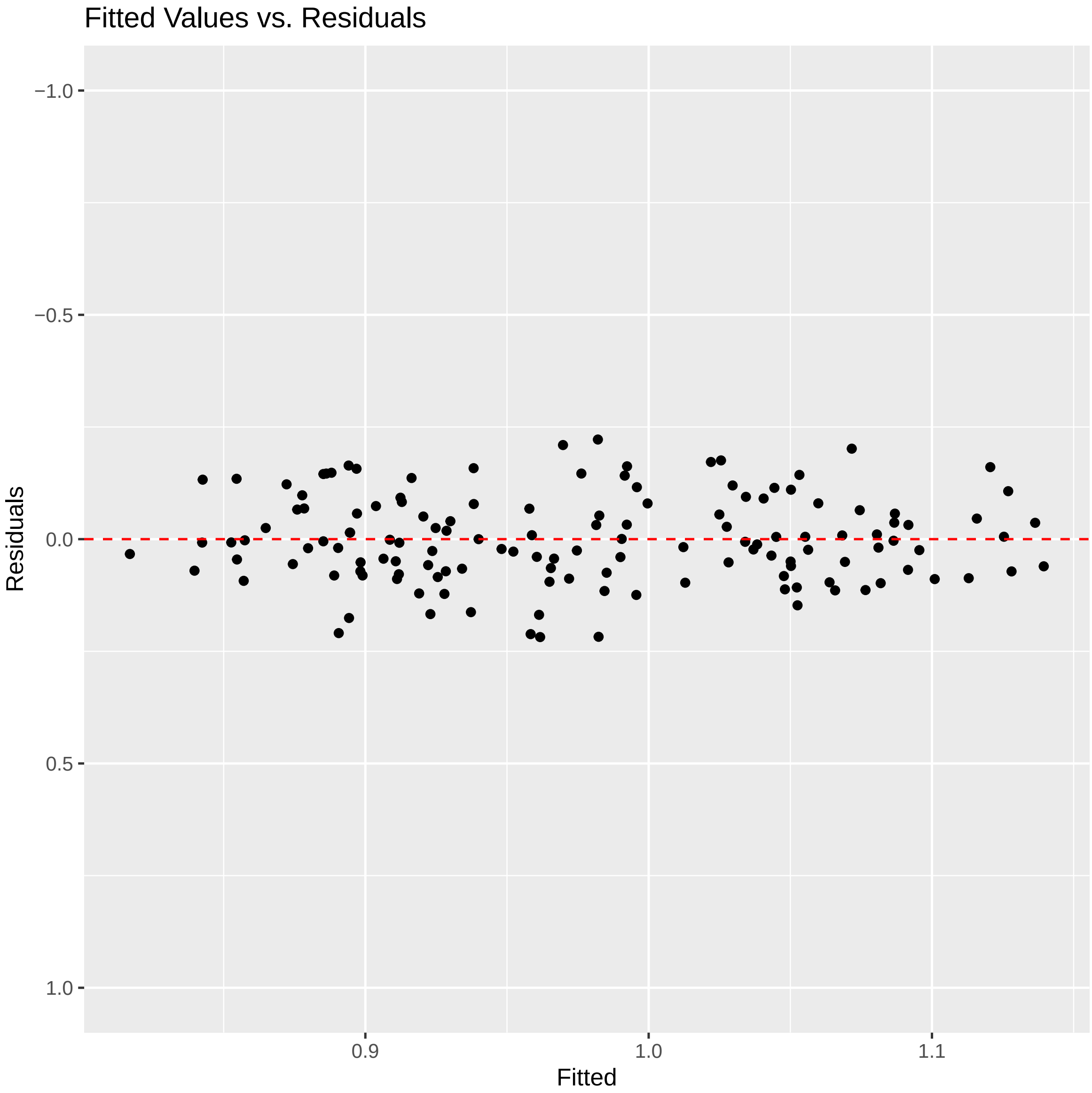
- SE of some estimates dropped significantly.

EDA for the final model.

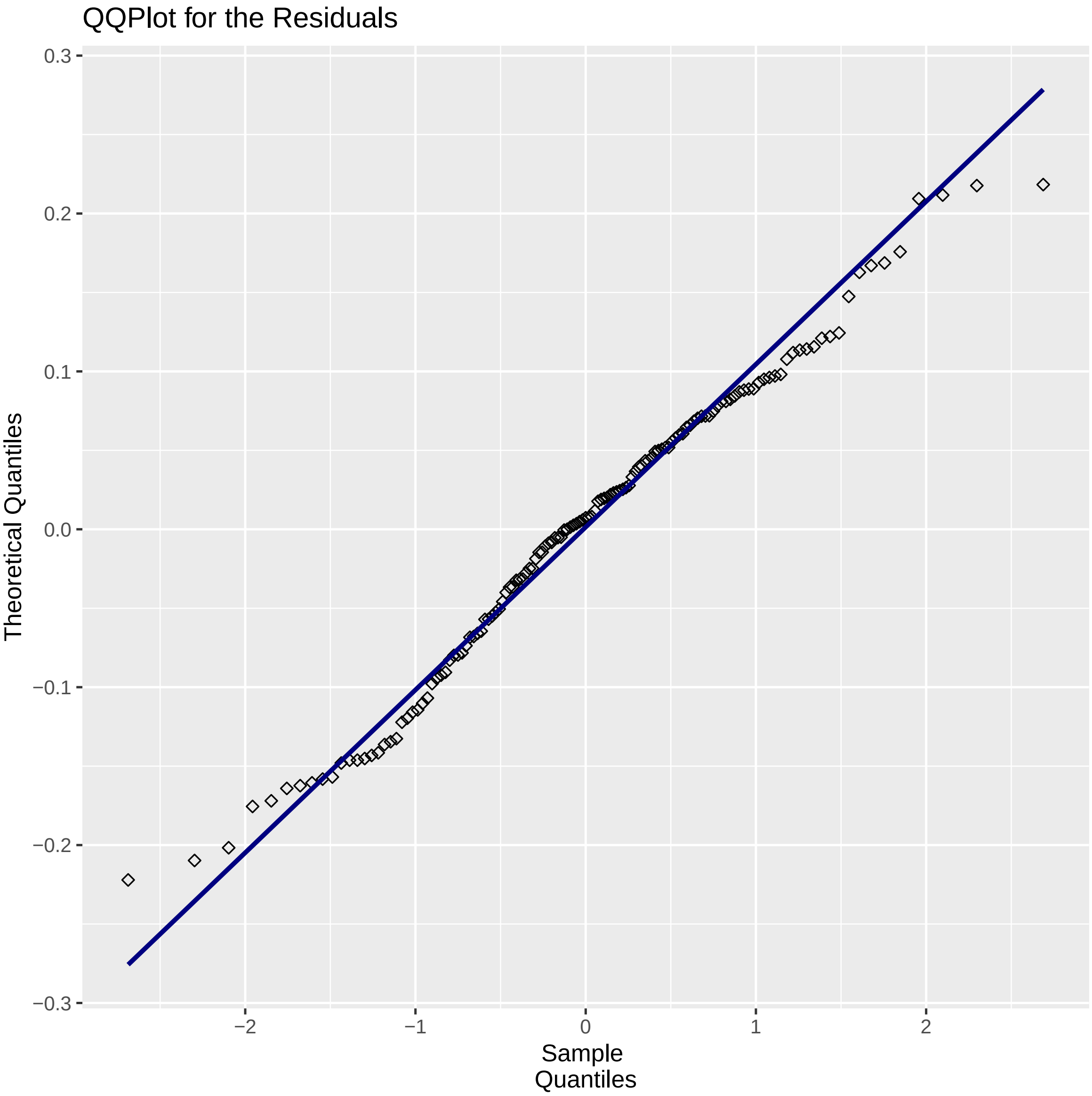
```
##Exploratory data analysis of final model
#fitted model vs response
plot(dbts$relwt,fmodel$fitted.values,xlim=c(0.6,1.4),ylim=c(0.6,1.4),
     main='Fitted Value Vs Response',xlab='relwt',ylab='fitted value',
     pch=20
     );box(lwd=3)
abline(a=0,b=1,col='red',lwd=1.5)
grid(col="darkgreen")
```



```
#--Diagonstics---
ggobj=ggplot(data=dbts,mapping=aes(x=fitted(fmodel)
,y=residuals(fmodel)))
ggobj+geom_point()+geom_hline(yintercept=0,
linetype="dashed",col="red")+ylim(1,-1)+
  xlab("Fitted")+ylab("Residuals")+
labs(title="Fitted Values vs. Residuals")
```



```
#---Checking for Normality---
df=data.frame(y=residuals(fmodel))
ggplot(df,aes(sample=y))+stat_qq(shape=5)+
stat_qq_line(lwd=1,col="navyblue")+labs(y="Theoretical Quantiles",x="Sample
Quantiles",title="QQPlot for the Residuals")
```



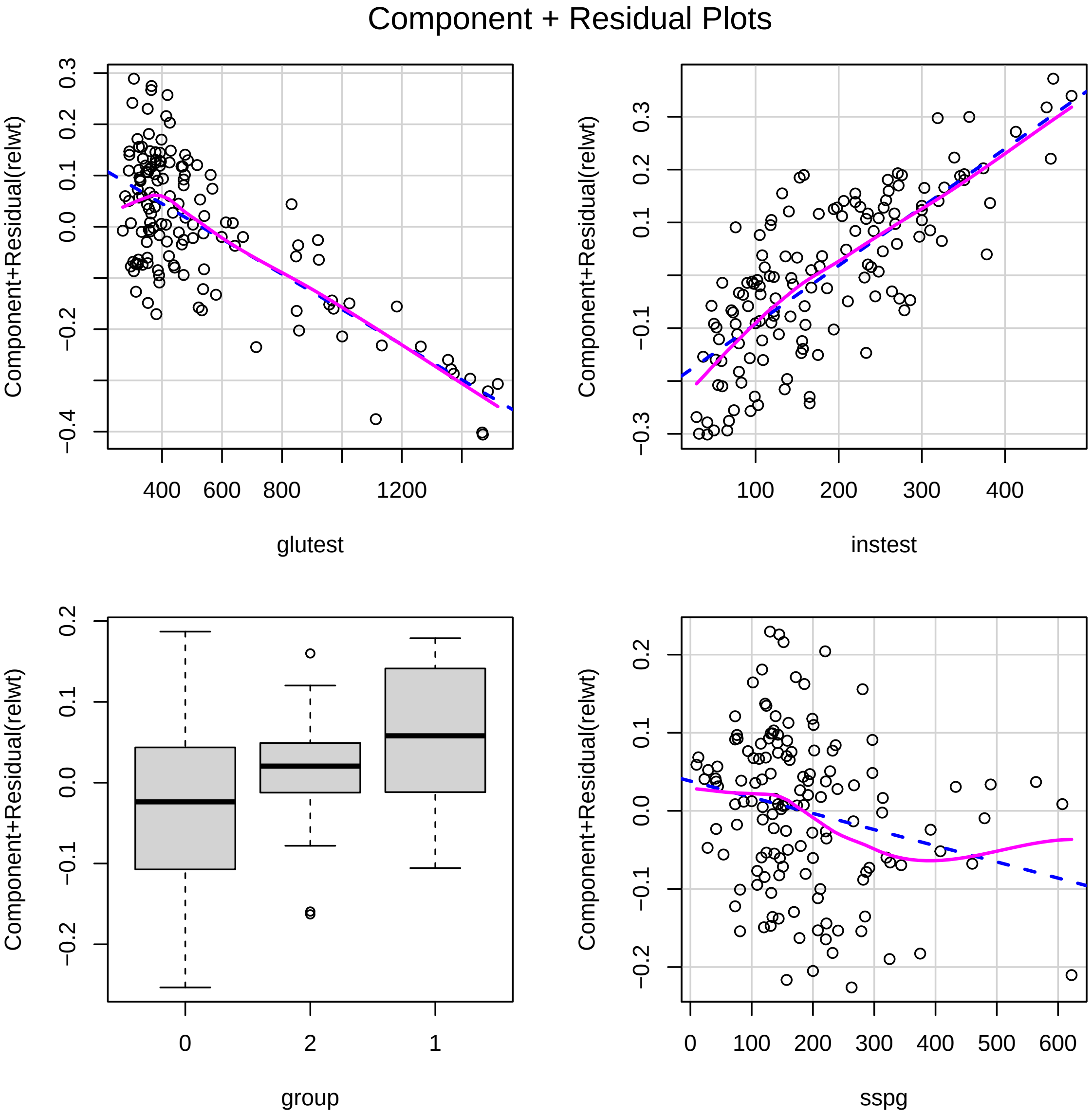
```
#---Normality Accepted---
```

11.10 Partial and Added Variable Plot for the Final Model.

11.10.1 Partial Residual Plot.

Partial residual plots are useful to verify one fo the key assumptions of multiple linear regression that there is a linear relationship b/w each predictor and response variable. If this assumptions is violated then the results of the regression model can be unreliable. One way to check this assumption is to create a partial residual plot, which displays $e_i^* = e_i + \hat{\beta}_j x_{ij}$ against x'_{ij}s.

```
crPlots(fmodel)
```



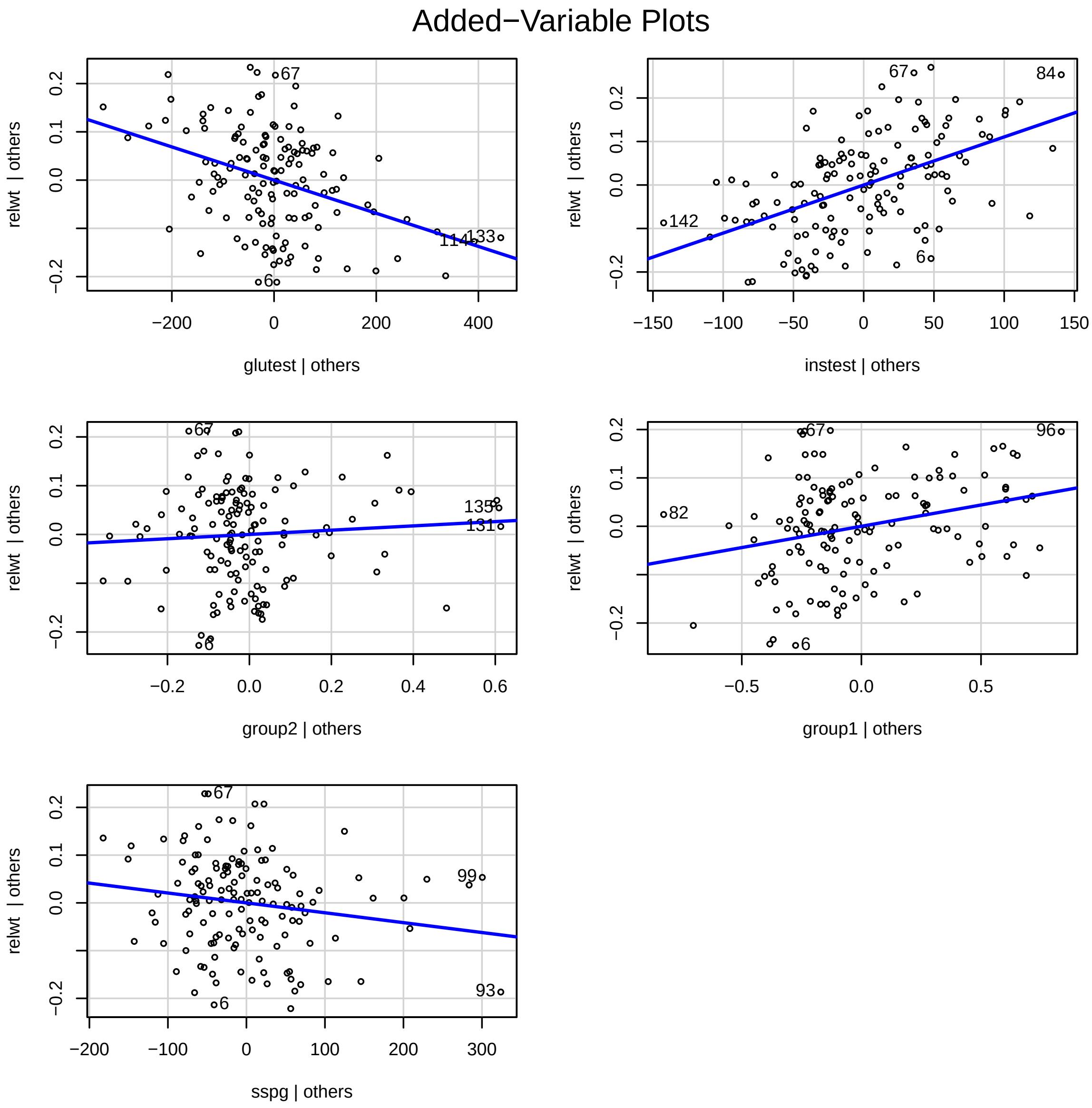
The blue line shows the expected residuals if the relationship b/w the predictor and response variable are linear. The pink line shows the actual residuals. If the two lines are significantly different, then this is evidence of a nonlinear relationship.

In our plots the graphs two lines in glutest and instest are very much close, but sspg mayn't be linearly related to $\boldsymbol{y}$. So, our multiple linear assumption is quite useful.

11.10.2 Added Variable Plot.

Added Variable Plots gives a better indication of the contribution that each explanatory variable makes to the fit. Here we plot the 'x_j residuals' and j^{th} covariate deleted residuals for each covariate we have.

```
avPlots(fmodel)
```



Each of the variables selected contributes significantly.

Chapter 12

Robust Regression

12.1 M-estimation

We will now approach the modeling with a different approach without deleting the outlying observations.

```
m=rlm(relwt~.,data=old.dbts)
summary(m)


Call: rlm(formula = relwt ~ ., data = old.dbts)
Residuals:
      Min        1Q    Median        3Q       Max
-0.213342 -0.064421  0.006464  0.075124  0.299906

Coefficients:
            Value   Std. Error t value
(Intercept)  0.9364  0.0361    25.9688
glufast      0.0008  0.0007     1.0905
glutest     -0.0005  0.0002    -2.8161
sspg        -0.0001  0.0001    -1.2300
instest      0.0010  0.0002     6.0267
group2       0.0634  0.0569     1.1146
group1       0.1014  0.0337     3.0076

Residual standard error: 0.1059 on 137 degrees of freedom
```

```
cor(predict(m),old.dbts$relwt)**2

[1] 0.3671875
```

The squared correlation between Fitted values and responses here is very close to as we got Multiple R^2 from the full model.