First Assignment

```r
library(ggplot2)
library(readr)
library(dotwhisker)
```

```
## Loading required package: gtable
```

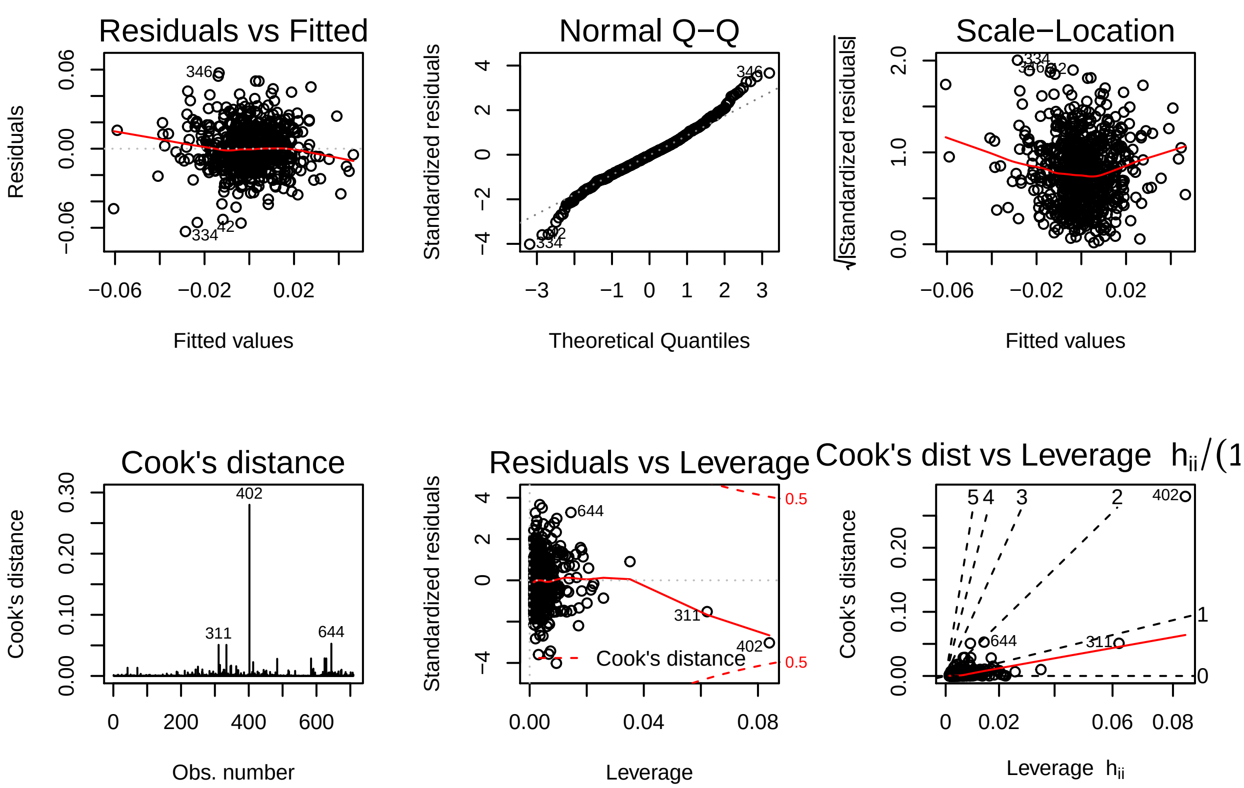
```r
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-5
```

```r
set.seed(100L)
myData=read_csv("w_logret_3automanu.csv",col_names=FALSE)
names(myData)=c("Toyota","Ford","GM")
```

Simple regression and it's plots

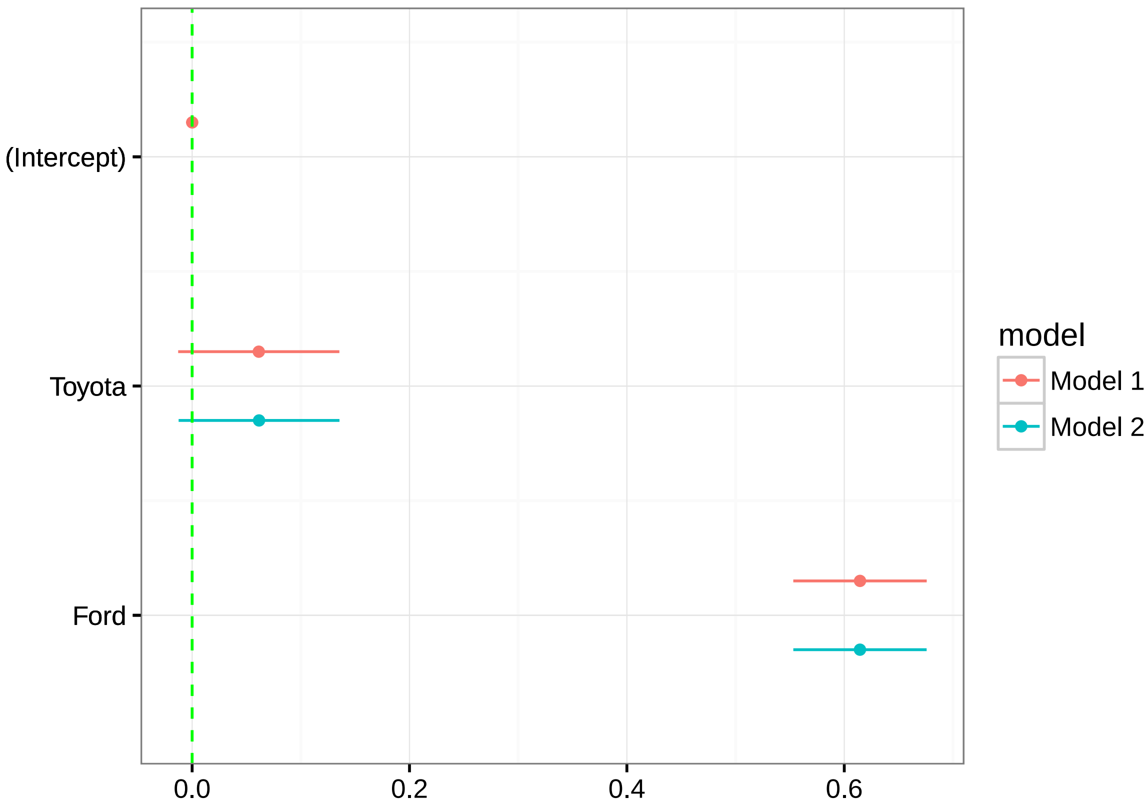
```r
myFit1=lm(GM~.,data=myData)
par(mfrow=c(2,3))
plot(myFit1,which=1:6,ask=FALSE,id.n=3)
```



- the labelled point of the first three and the last three are different
- symbol : and * is different (see ?formula)

The influence of intercept

```
myFit2=update(myFit1,.~.-1)

dwplot(list(myFit1,myFit2))+
    theme_bw()+
    geom_vline(xintercept = 0,colour="green",linetype=2)
```

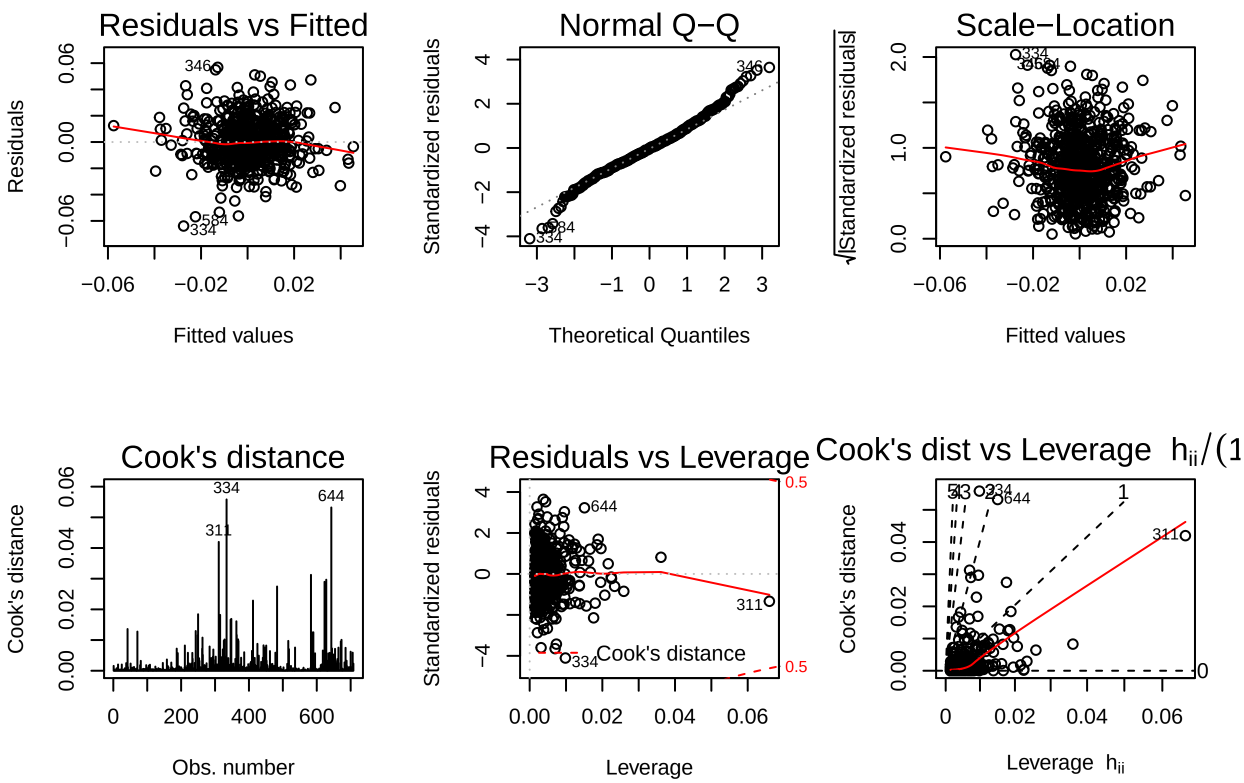



```
confint(myFit1)
```

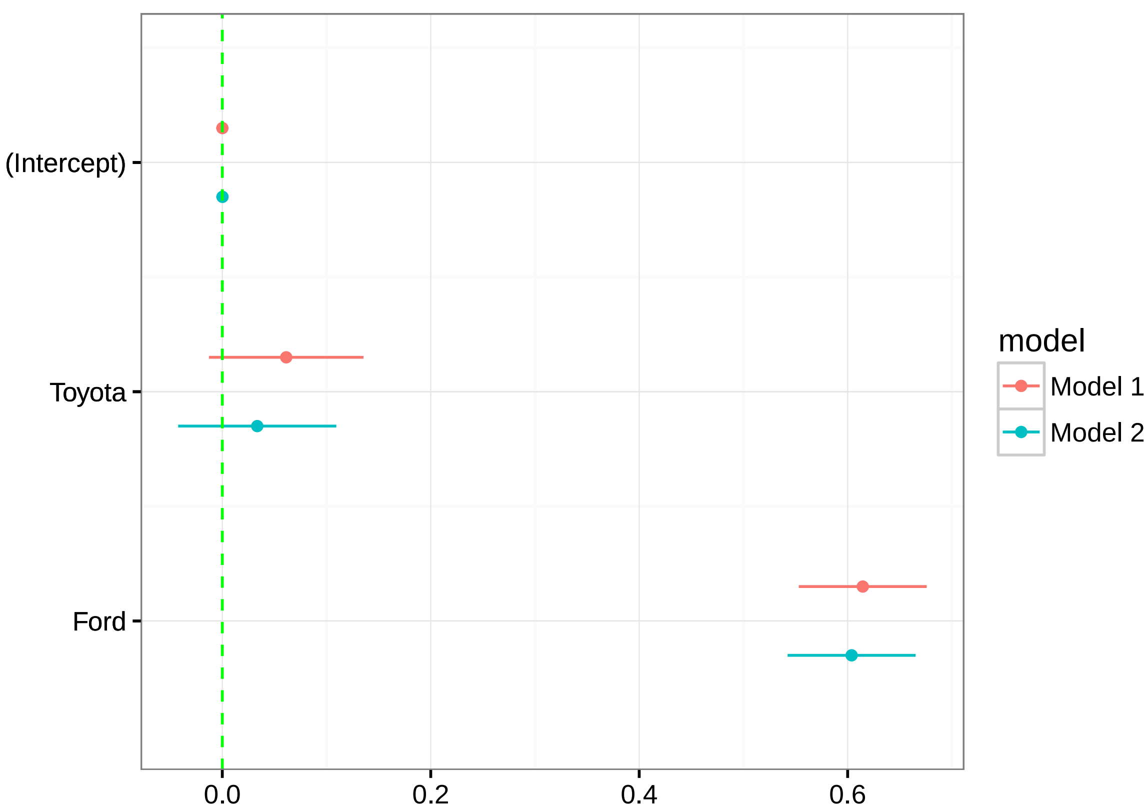
```
##                    2.5 %      97.5 %
## (Intercept) -0.001090582 0.001231555
## Toyota      -0.012971455 0.135614228
## Ford         0.553001107 0.675991088
```

remove influential points

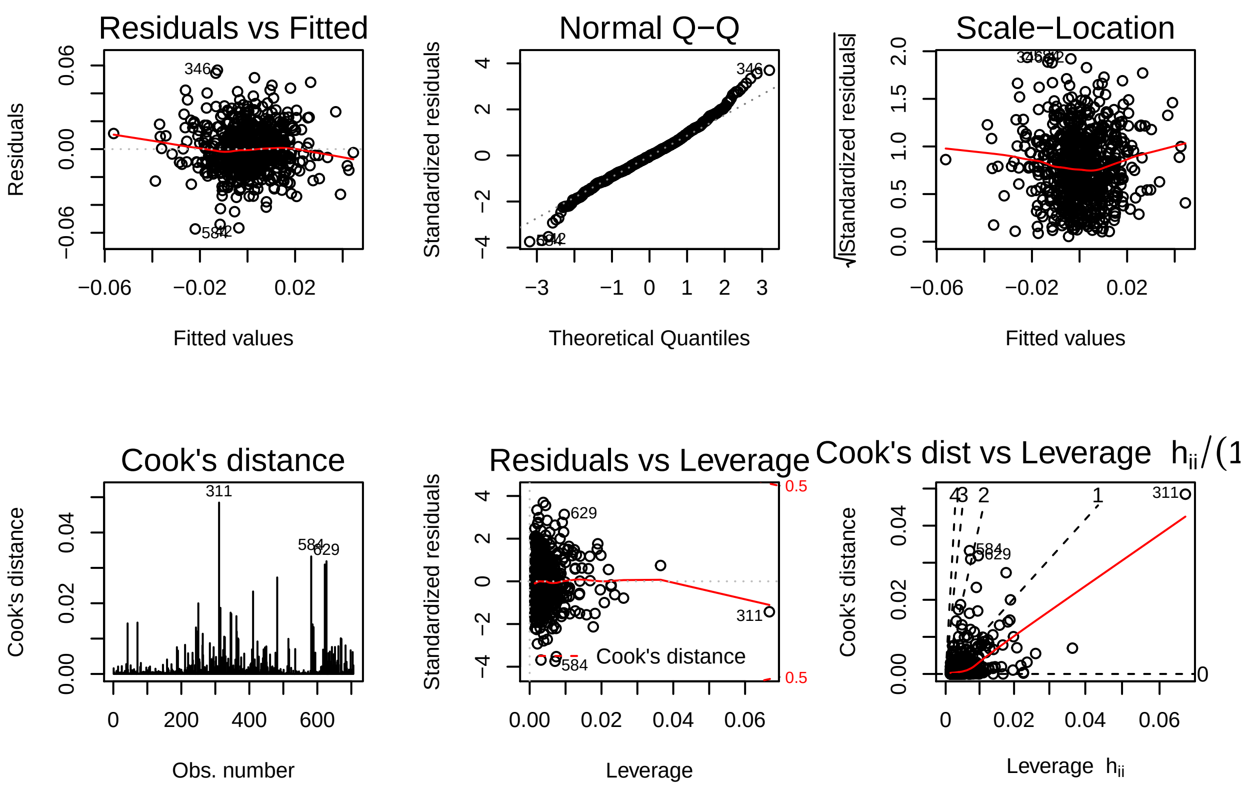
```
tmp= 1:709 %in% 402
myFit3=update(myFit1,subset=!tmp)
par(mfrow=c(2,3))
plot(myFit3,which=1:6,ask=FALSE,id.n=3)
```



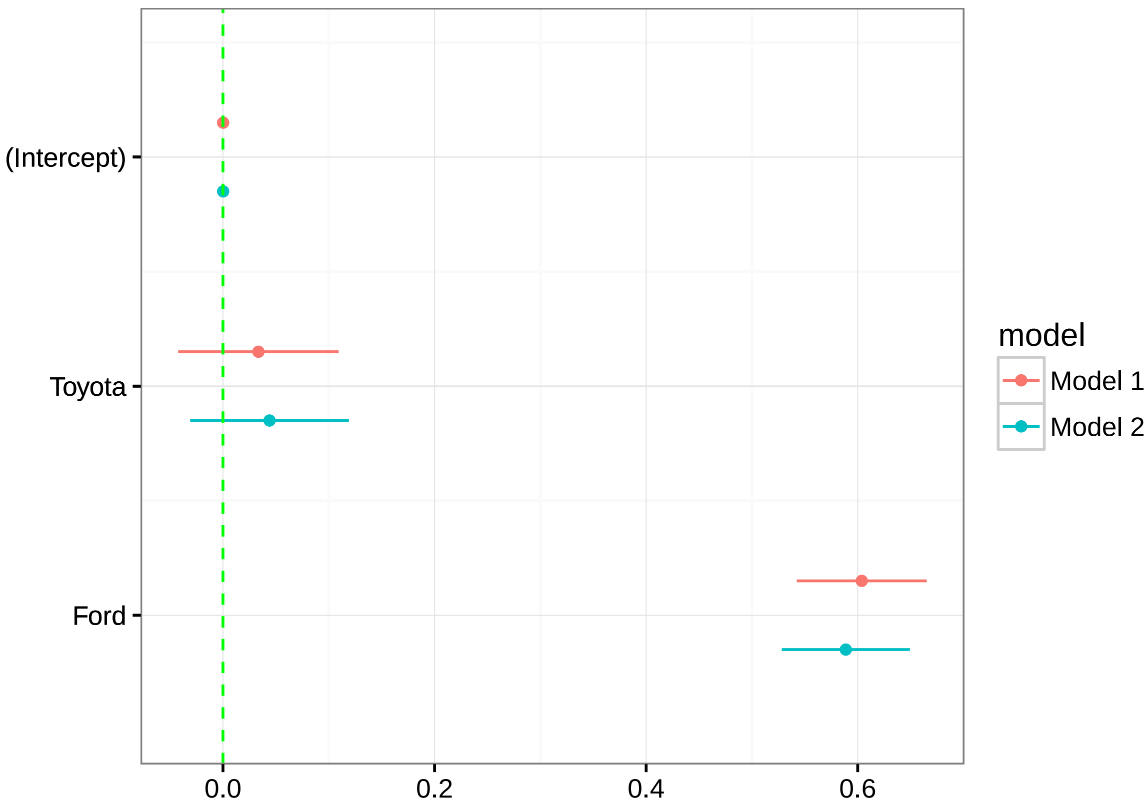

```r
dwplot(list(myFit1,myFit3))+
    theme_bw()+
    geom_vline(xintercept = 0,colour="green",linetype=2)
```



```r
tmp=1:709 %in% c(402,
                 334,644)
myFit4=update(myFit1,subset=!tmp)
par(mfrow=c(2,3))
plot(myFit4,which=1:6,ask=FALSE,id.n=3)
```

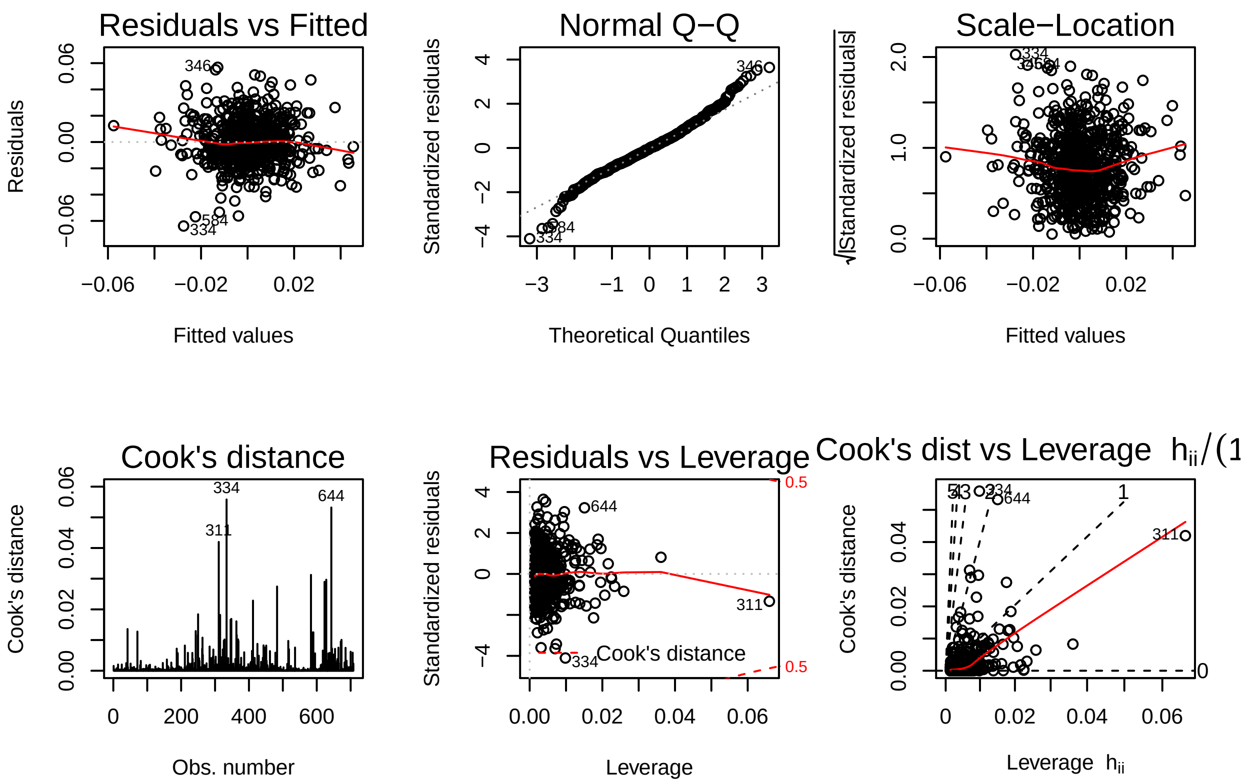



```r
dwplot(list(myFit3,myFit4))+
    theme_bw()+
    geom_vline(xintercept = 0,colour="green",linetype=2)
```



the wrong way:

```
myData=read.csv("w_logret_3automanu.csv",header = FALSE)
names(myData)=c("Toyota","Ford","GM")
myData1=myData[-402,]
wrong1=lm(GM~.,data=myData1)
par(mfrow=c(2,3))
plot(wrong1,which=1:6,ask=FALSE)
```



now the 3 biggest cook's D is 334, 643, 311. But

```r
wrong1$residuals[c(334,643,311)]
```

```
##         334         644         311
## -0.06387989  0.05009237 -0.02017410
```

```r
wrong1$residuals[c(335,644,311)]
```

```
##         335         645         311
## -0.01246845 -0.02110989 -0.02017410
```

it is because of the behavior of rownames of myData1. So how to solve it?

First method: rename the row

```r
rownames(myData1)=1:nrow(myData1)
```

Second method: use read_csv as we did (recommand)

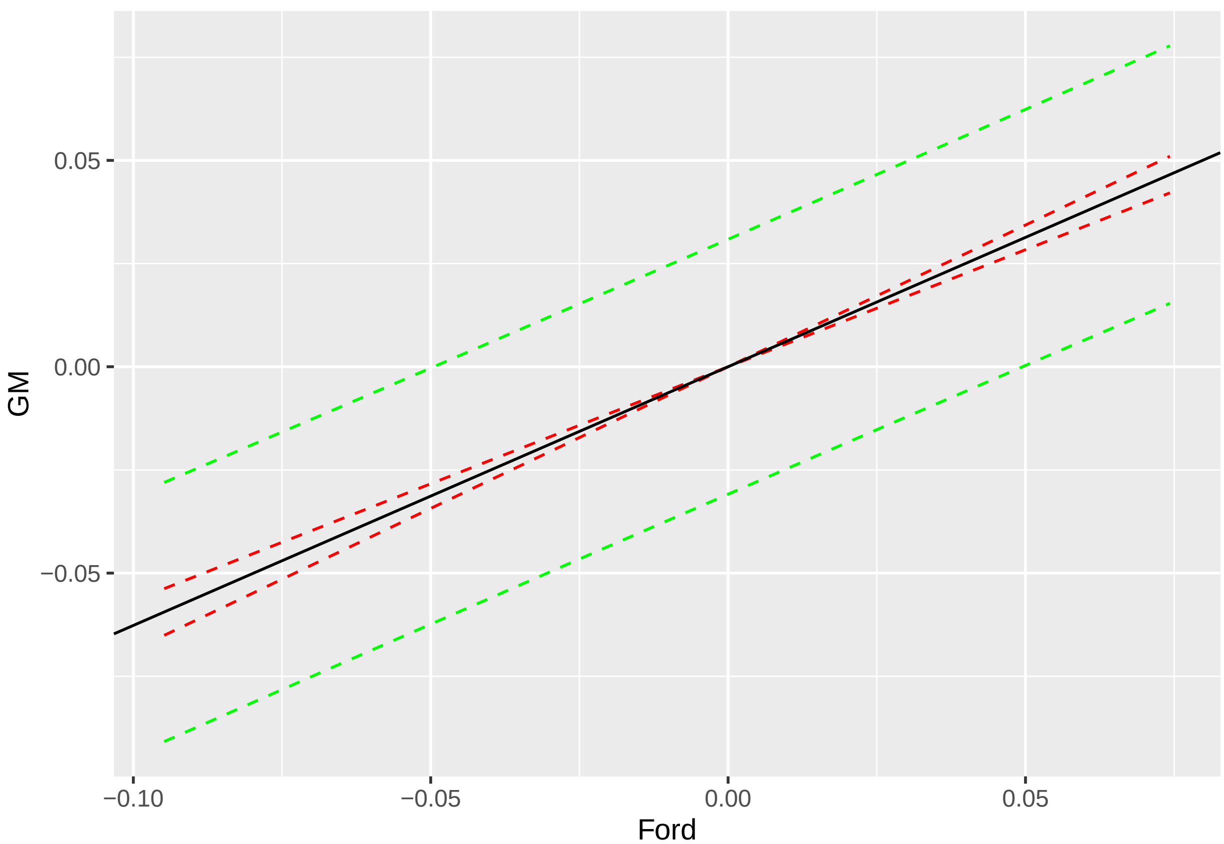
```r
myData=read_csv("w_logret_3automanu.csv",col_names=FALSE)
names(myData)=c("Toyota","Ford","GM")
myData1=myData[-402,]
right1=lm(GM~.,data=myData1)
right1$residuals[c(334,643,311)]
```

```
##         334         643         311
## -0.06387989  0.05009237 -0.02017410
```

Prediction

First we plot the confident interval of myFit1

```r
myFit5=update(myFit1,.~Ford-1)
tmpData=myData
tmpData[,c("lc","uc")]=predict(myFit5,myData,level=0.95,interval="confidence")[,c(2,3)]
tmpData[,c("lp","up")]=predict(myFit5,myData,level=0.95,interval="prediction")[,c(2,3)]

ggplot(data=tmpData)+
  geom_line(aes(x=Ford,y=lc),color="red",linetype=2)+
  geom_line(aes(x=Ford,y=uc),color="red",linetype=2)+
  geom_line(aes(x=Ford,y=lp),color="green",linetype=2)+
  geom_line(aes(x=Ford,y=up),color="green",linetype=2)+
  geom_abline(intercept = 0,slope = myFit5$coefficients[["Ford"]])+
  ylab("GM")
```



Next, we want to do better

Feature engineering

```
for(i in 1:4){
  myData[,paste("featureT",i,sep="")]=factor(1*(myData$Toyota>quantile(abs(myData$Toyota),i/5)))
}

for(i in 1:4){
  myData[,paste("featureF",i,sep="")]=factor(1*(myData$Ford>quantile(abs(myData$Ford),i/5)))
}
```

Split the data to obtain training set and testing set

```
inT=sample(1:nrow(myData),600)
training=myData[inT,]
testing=myData[-inT,]
```

The MSE of univariate regression and new regression

```
pFit1=lm(GM~(.)^2,training)
sum((predict(pFit1,testing)-testing$GM)^2)
```

```
## Warning in predict.lm(pFit1, testing): prediction from a rank-deficient fit
## may be misleading
```

```
## [1] 0.02212196
```

```
pFit2=lm(GM~Ford,training)
sum((predict(pFit2,testing)-testing$GM)^2)
```

```
## [1] 0.02134008
```

The new model behave even worse! It overfits!! To overcome overfitting, we regularize the regeression by the LASSO.

$$min\frac{1}{2n}\|y-X\beta\|^2+\lambda\sum_{i=1}^{p}|\beta_i|$$

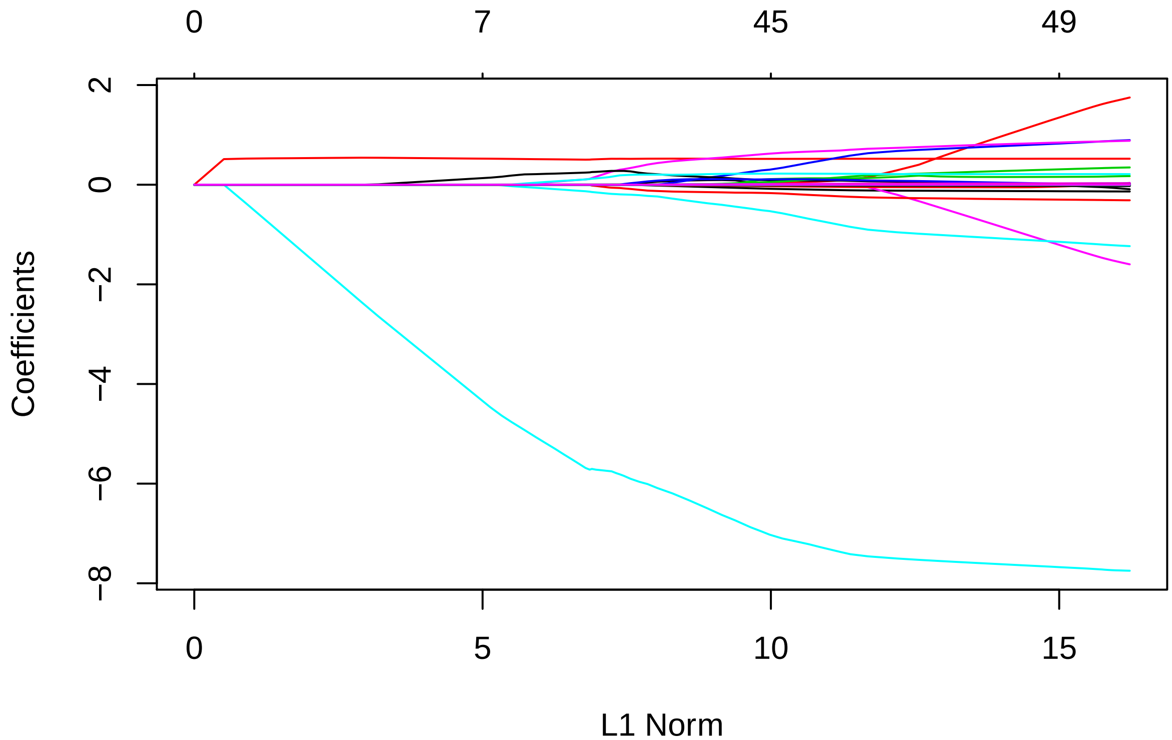
```
tmp=model.matrix(GM~(.)^2,training)
tmp=as.data.frame(tmp)
glmFit=glmnet(as.matrix(tmp),as.matrix(training$GM),family="gaussian")
coef(glmFit,s=0.01)
```

```
## 57 x 1 sparse Matrix of class "dgCMatrix"
##                                     1
## (Intercept)           -0.0001126704
```

```
## (Intercept)            .
## Toyota                 .
## Ford                   0.1419696369
## featureT11             .
## featureT21             .
## featureT31             .
## featureT41             .
## featureF11             .
## featureF21             .
## featureF31             .
## featureF41             .
## Toyota:Ford            .
## Toyota:featureT11      .
## Toyota:featureT21      .
## Toyota:featureT31      .
## Toyota:featureT41      .
## Toyota:featureF11      .
## Toyota:featureF21      .
## Toyota:featureF31      .
## Toyota:featureF41      .
## Ford:featureT11        .
## Ford:featureT21        .
## Ford:featureT31        .
## Ford:featureT41        .
## Ford:featureF11        .
## Ford:featureF21        .
## Ford:featureF31        .
## Ford:featureF41        .
## featureT11:featureT21  .
## featureT11:featureT31  .
## featureT11:featureT41  .
## featureT11:featureF11  .
## featureT11:featureF21  .
## featureT11:featureF31  .
## featureT11:featureF41  .
## featureT21:featureT31  .
## featureT21:featureT41  .
## featureT21:featureF11  .
## featureT21:featureF21  .
## featureT21:featureF31  .
## featureT21:featureF41  .
## featureT31:featureT41  .
## featureT31:featureF11  .
## featureT31:featureF21  .
## featureT31:featureF31  .
## featureT31:featureF41  .
## featureT41:featureF11  .
## featureT41:featureF21  .
## featureT41:featureF31  .
## featureT41:featureF41  .
## featureF11:featureF21  .
## featureF11:featureF31  .
## featureF11:featureF41  .
## featureF21:featureF31  .
```

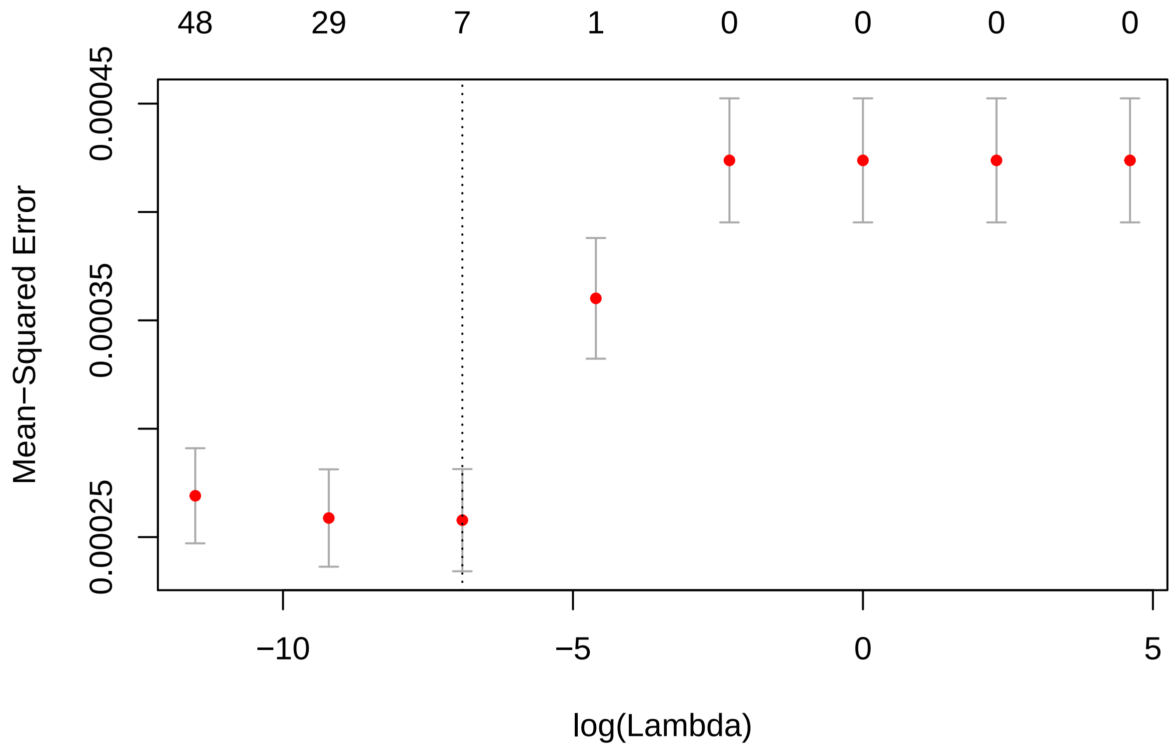
```
## featureF21:featureF41   .
## featureF31:featureF41   .
```

```r
plot(glmFit)
```



Determine λ by cross validation

```r
glmFit2=cv.glmnet(as.matrix(tmp),
                  as.matrix(training$GM),
                  family="gaussian",
                  nfolds=10,
                  lambda=c(1e-5,0.0001,0.001,0.01,0.1,1,10,100))

plot(glmFit2)
```




```r
coef(glmFit,s=1e-4)
```

```
## 57 x 1 sparse Matrix of class "dgCMatrix"
##                                 1
## (Intercept)          -2.649150e-03
## (Intercept)           .
## Toyota                .
## Ford                  5.200818e-01
## featureT11            1.751511e-03
## featureT21            2.268234e-03
## featureT31           -6.881640e-04
## featureT41           -2.293043e-04
## featureF11            4.504605e-03
## featureF21            .
## featureF31            2.443505e-03
## featureF41           -3.557123e-04
## Toyota:Ford          -5.817757e+00
## Toyota:featureT11     .
## Toyota:featureT21     .
## Toyota:featureT31     .
## Toyota:featureT41     .
## Toyota:featureF11     .
## Toyota:featureF21    -1.940659e-01
## Toyota:featureF31     3.015835e-01
## Toyota:featureF41     2.764767e-01
## Ford:featureT11      -6.742353e-02
## Ford:featureT21       .
## Ford:featureT31       2.898905e-03
## Ford:featureT41       1.878310e-01
## Ford:featureF11       .
```

```
## Ford:featureF21        .
## Ford:featureF31        .
## Ford:featureF41        .
## featureT11:featureT21  7.932750e-05
## featureT11:featureT31 -1.656907e-03
## featureT11:featureT41  .
## featureT11:featureF11 -5.147582e-04
## featureT11:featureF21  .
## featureT11:featureF31  .
## featureT11:featureF41  .
## featureT21:featureT31  .
## featureT21:featureT41  .
## featureT21:featureF11  .
## featureT21:featureF21 -1.346622e-03
## featureT21:featureF31  .
## featureT21:featureF41  .
## featureT31:featureT41  .
## featureT31:featureF11 -7.141401e-04
## featureT31:featureF21  .
## featureT31:featureF31 -3.611941e-03
## featureT31:featureF41  .
## featureT41:featureF11  .
## featureT41:featureF21  .
## featureT41:featureF31 -1.912871e-03
## featureT41:featureF41  .
## featureF11:featureF21  .
## featureF11:featureF31  3.157505e-04
## featureF11:featureF41 -4.979810e-07
## featureF21:featureF31  1.377886e-05
## featureF21:featureF41 -4.937184e-07
## featureF31:featureF41  .
```

```r
myPre=predict.glmnet(glmFit,model.matrix(GM~(.)^2,testing),s=1e-4)
sum((myPre-testing$GM)^2)
```

```
## [1] 0.02096205
```