Lab 2

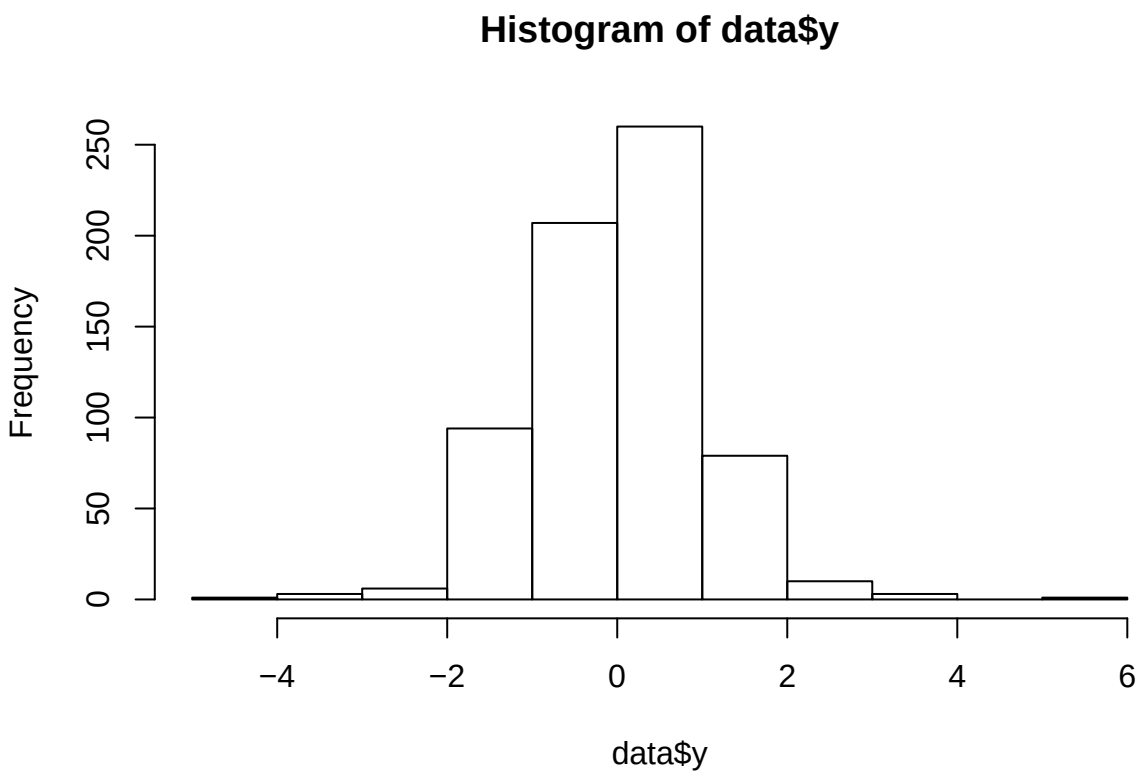
Attila Lazar

21.10.2020

Data

We load the data and extract the varibles y and X20-X61

```
load("data/dat.RData")
#str(d)
data <- d[, names(d) %in% c('y', paste0('X', 20:65))]
hist(data$y)
```




```
set.seed(123)
n <- nrow(data)
train <- sample(1:n, round(n*2/3))
test <- (1:n) [-train]
```

The Histogram of the respose variable 'y' looks normally distributed

1. Full model

```
#full_formula <- as.formula(paste('y~', paste(paste0('X', 20:65), collapse="+")))
model1 <- lm(y~., data, subset=train)
summary (model1)
```

```
##
## Call:
## lm(formula = y ~ ., data = data, subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.21975 -0.27220 -0.03225  0.27112  2.16927
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.01156    0.02695   0.429  0.66810
## X20           0.04890    0.07841   0.624  0.53322
## X21           0.18245    1.16776   0.156  0.87593
## X22           0.03899    0.06927   0.563  0.57385
## X23          -0.99924    1.36039  -0.735  0.46306
## X24          -1.02367    1.15948  -0.883  0.37784
## X25          -0.30079    0.56521  -0.532  0.59490
## X26          -0.79628    0.35469  -2.245  0.02532 *
## X27           2.28642    1.14929   1.989  0.04734 *
## X28           0.37627    0.40471   0.930  0.35308
## X29         -11.56281   15.17184  -0.762  0.44644
## X30          -1.84651    1.44085  -1.282  0.20075
## X31           0.79186    1.72246   0.460  0.64597
## X32          -1.53890    1.27309  -1.209  0.22746
## X33           0.11954    0.16458   0.726  0.46808
## X34           3.01746    0.74671   4.041 6.39e-05 ***
## X35          -0.19646    0.04677  -4.201 3.29e-05 ***
## X36           0.70653    0.10119   6.982 1.23e-11 ***
## X37           0.20653    0.04449   4.642 4.70e-06 ***
## X38          -0.03215    0.04405  -0.730  0.46592
## X39          -0.16915    0.12470  -1.356  0.17572
## X40           0.03149    0.08224   0.383  0.70197
## X41          -0.01142    0.10025  -0.114  0.90939
## X42          -0.04666    0.04539  -1.028  0.30455
## X43          -1.35034    1.51997  -0.888  0.37486
## X44          -0.02812    0.99191  -0.028  0.97740
## X45          -0.25559    0.32690  -0.782  0.43476
## X46           0.03600    1.14239   0.032  0.97488
## X47          -0.67810    1.34663  -0.504  0.61485
## X48           1.02809    0.24730   4.157 3.95e-05 ***
## X49          -0.44484    0.52625  -0.845  0.39845
## X50           0.52129    0.48155   1.083  0.27968
## X51           1.66433    1.69504   0.982  0.32675
## X52          -1.77484    1.12856  -1.573  0.11659
## X53          -0.61736    1.73932  -0.355  0.72282
## X54          -0.01578    0.65804  -0.024  0.98088
## X55           0.04775    0.07000   0.682  0.49553
## X56          -0.39152    1.23496  -0.317  0.75138
## X57           0.57076    0.38797   1.471  0.14204
## X58           0.37110    0.50378   0.737  0.46178
## X59          11.99787   15.34847   0.782  0.43486
## X60          -1.64988    2.23328  -0.739  0.46048
## X61                NA         NA      NA       NA
## X62           1.14733    0.71795   1.598  0.11082
```

```
## X63           0.56568    0.19342   2.925  0.00365 **
## X64          -0.67518    0.13702  -4.928 1.22e-06 ***
## X65          -2.87475    0.63777  -4.507 8.64e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5577 on 397 degrees of freedom
## Multiple R-squared:  0.6966, Adjusted R-squared:  0.6622
## F-statistic: 20.26 on 45 and 397 DF,  p-value: < 2.2e-16
```

The summar of the model reveals NA values probably becouse variables are not lin. independent. We use alias to determine these variables

```
alias(model1,  partial = FALSE)
```

```
## Model :
## y ~ X20 + X21 + X22 + X23 + X24 + X25 + X26 + X27 + X28 + X29 +
##     X30 + X31 + X32 + X33 + X34 + X35 + X36 + X37 + X38 + X39 +
##     X40 + X41 + X42 + X43 + X44 + X45 + X46 + X47 + X48 + X49 +
##     X50 + X51 + X52 + X53 + X54 + X55 + X56 + X57 + X58 + X59 +
##     X60 + X61 + X62 + X63 + X64 + X65
##
## Complete :
##     (Intercept) X20 X21 X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33
## X61 0           0   0   1   0   0   0   0   0   0   0   0   0   0   0
##     X34 X35 X36 X37 X38 X39 X40 X41 X42 X43 X44 X45 X46 X47 X48 X49 X50
## X61 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##     X51 X52 X53 X54 X55 X56 X57 X58 X59 X60 X62 X63 X64 X65
## X61 0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

We drop X61 from our model

```
model2 <- update(model1, .~.-X61)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ X20 + X21 + X22 + X23 + X24 + X25 + X26 + X27 +
##     X28 + X29 + X30 + X31 + X32 + X33 + X34 + X35 + X36 + X37 +
##     X38 + X39 + X40 + X41 + X42 + X43 + X44 + X45 + X46 + X47 +
##     X48 + X49 + X50 + X51 + X52 + X53 + X54 + X55 + X56 + X57 +
##     X58 + X59 + X60 + X62 + X63 + X64 + X65, data = data, subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.21975 -0.27220 -0.03225  0.27112  2.16927
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.01156    0.02695   0.429  0.66810
## X20           0.04890    0.07841   0.624  0.53322
## X21           0.18245    1.16776   0.156  0.87593
## X22           0.03899    0.06927   0.563  0.57385
## X23          -0.99924    1.36039  -0.735  0.46306
## X24          -1.02367    1.15948  -0.883  0.37784
## X25          -0.30079    0.56521  -0.532  0.59490
```

```
## X26          -0.79628    0.35469  -2.245  0.02532 *
## X27           2.28642    1.14929   1.989  0.04734 *
## X28           0.37627    0.40471   0.930  0.35308
## X29         -11.56281   15.17184  -0.762  0.44644
## X30          -1.84651    1.44085  -1.282  0.20075
## X31           0.79186    1.72246   0.460  0.64597
## X32          -1.53890    1.27309  -1.209  0.22746
## X33           0.11954    0.16458   0.726  0.46808
## X34           3.01746    0.74671   4.041 6.39e-05 ***
## X35          -0.19646    0.04677  -4.201 3.29e-05 ***
## X36           0.70653    0.10119   6.982 1.23e-11 ***
## X37           0.20653    0.04449   4.642 4.70e-06 ***
## X38          -0.03215    0.04405  -0.730  0.46592
## X39          -0.16915    0.12470  -1.356  0.17572
## X40           0.03149    0.08224   0.383  0.70197
## X41          -0.01142    0.10025  -0.114  0.90939
## X42          -0.04666    0.04539  -1.028  0.30455
## X43          -1.35034    1.51997  -0.888  0.37486
## X44          -0.02812    0.99191  -0.028  0.97740
## X45          -0.25559    0.32690  -0.782  0.43476
## X46           0.03600    1.14239   0.032  0.97488
## X47          -0.67810    1.34663  -0.504  0.61485
## X48           1.02809    0.24730   4.157 3.95e-05 ***
## X49          -0.44484    0.52625  -0.845  0.39845
## X50           0.52129    0.48155   1.083  0.27968
## X51           1.66433    1.69504   0.982  0.32675
## X52          -1.77484    1.12856  -1.573  0.11659
## X53          -0.61736    1.73932  -0.355  0.72282
## X54          -0.01578    0.65804  -0.024  0.98088
## X55           0.04775    0.07000   0.682  0.49553
## X56          -0.39152    1.23496  -0.317  0.75138
## X57           0.57076    0.38797   1.471  0.14204
## X58           0.37110    0.50378   0.737  0.46178
## X59          11.99787   15.34847   0.782  0.43486
## X60          -1.64988    2.23328  -0.739  0.46048
## X62           1.14733    0.71795   1.598  0.11082
## X63           0.56568    0.19342   2.925  0.00365 **
## X64          -0.67518    0.13702  -4.928 1.22e-06 ***
## X65          -2.87475    0.63777  -4.507 8.64e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5577 on 397 degrees of freedom
## Multiple R-squared:  0.6966, Adjusted R-squared:  0.6622
## F-statistic: 20.26 on 45 and 397 DF,  p-value: < 2.2e-16
```

There are no NA values any more

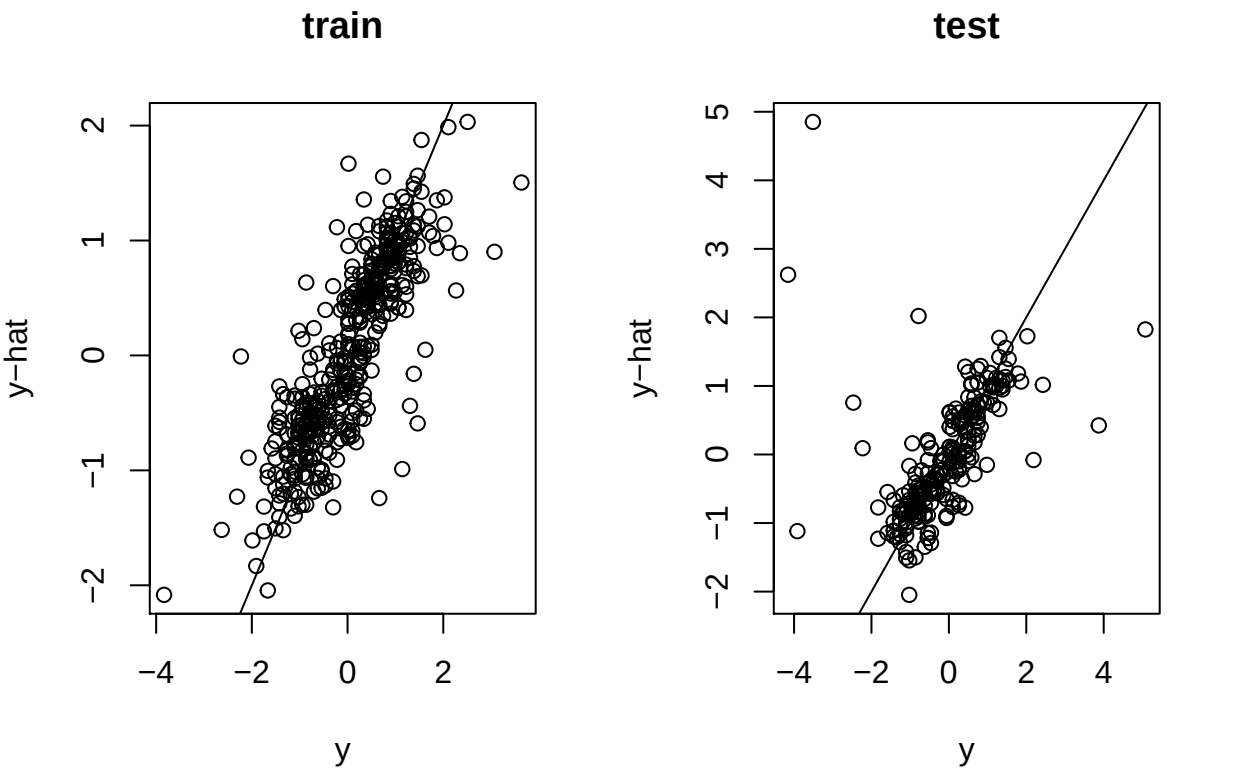
```
alias(model2,  partial = FALSE)
```

```
## Model :
## y ~ X20 + X21 + X22 + X23 + X24 + X25 + X26 + X27 + X28 + X29 +
##     X30 + X31 + X32 + X33 + X34 + X35 + X36 + X37 + X38 + X39 +
##     X40 + X41 + X42 + X43 + X44 + X45 + X46 + X47 + X48 + X49 +
##     X50 + X51 + X52 + X53 + X54 + X55 + X56 + X57 + X58 + X59 +
```

```
##     X60 + X62 + X63 + X64 + X65
```

```
par(mfrow=c(1,2))

plot(data[train, 'y'], predict(model2, data[train,]), xlab='y' ,ylab='y-hat', main='train')
abline(c(0,1))
plot(data[test, 'y'], predict(model2, data[test,]), xlab='y' ,ylab='y-hat', main='test')
abline(c(0,1))
```



According to the plots the predictions look promising. The MSE is as expected much bigger for the test data.

```
#mse_train
mean((data[train, 'y'] - predict(model2, data[train,]))^2)
```

```
## [1] 0.2787313
```

```
#mse_test
mean((data[test, 'y'] - predict(model2, data[test,]))^2)
```

```
## [1] 0.9699528
```

2. Stepwise regression

We train a model using 'forward', 'backward' and 'both' options. for the 'forward' model we use the formula from model2 as scope.

```
reducedf1 <-as.formula(paste('y~', paste(paste0('X', setdiff(20:65, 61)), collapse="+")))

model3 <- step(lm(y~1,data, train), scope=reducedf1, direction='forward')
model4 <- step(lm(reducedf1, data, train), direction='backward')
model5 <- step(lm(reducedf1, data, train))
```

```r
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ X36 + X64 + X54 + X21 + X37 + X63 + X32 + X45 +
##     X26 + X35 + X22 + X29 + X65 + X34 + X23 + X48 + X28 + X53 +
##     X57 + X62 + X55 + X58, data = data, subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28953 -0.27453 -0.01628  0.28641  2.26626
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01301    0.02653   0.490 0.624090
## X36          0.70663    0.08033   8.796  < 2e-16 ***
## X64         -0.69812    0.10315  -6.768 4.41e-11 ***
## X54         -0.34452    0.09753  -3.532 0.000458 ***
## X21         -0.50388    0.14431  -3.492 0.000531 ***
## X37          0.22056    0.03942   5.595 3.99e-08 ***
## X63          0.69468    0.13947   4.981 9.27e-07 ***
## X32         -0.66914    0.12011  -5.571 4.53e-08 ***
## X45         -0.22862    0.10056  -2.273 0.023508 *
## X26         -0.40910    0.12139  -3.370 0.000821 ***
## X35         -0.16997    0.03714  -4.577 6.23e-06 ***
## X22          0.09211    0.04982   1.849 0.065218 .
## X29          0.38724    0.09718   3.985 7.96e-05 ***
## X65         -2.66865    0.52419  -5.091 5.39e-07 ***
## X34          2.58672    0.59262   4.365 1.60e-05 ***
## X23         -0.55169    0.11319  -4.874 1.55e-06 ***
## X48          0.89747    0.17220   5.212 2.94e-07 ***
## X28          0.01142    0.06697   0.170 0.864713
## X53         -1.30677    0.34534  -3.784 0.000177 ***
## X57          0.66624    0.16930   3.935 9.72e-05 ***
## X62          0.72779    0.26177   2.780 0.005676 **
## X55          0.06545    0.03477   1.882 0.060492 .
## X58          0.16030    0.10508   1.525 0.127890
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5523 on 420 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6687
## F-statistic: 41.55 on 22 and 420 DF,  p-value: < 2.2e-16
```

```r
summary(model4)
```

```
##
## Call:
## lm(formula = y ~ X26 + X27 + X28 + X29 + X30 + X32 + X34 + X35 +
##     X36 + X37 + X39 + X45 + X48 + X52 + X53 + X55 + X57 + X58 +
##     X59 + X60 + X62 + X63 + X64 + X65, data = data, subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.0957 -0.2795 -0.0410  0.2789  2.2227
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01355    0.02646   0.512 0.608908
## X26         -0.38622    0.12185  -3.170 0.001639 **
## X27          1.34837    0.88993   1.515 0.130494
## X28          0.60253    0.32887   1.832 0.067644 .
## X29         -7.66543    2.39249  -3.204 0.001459 **
## X30         -2.61813    1.01926  -2.569 0.010555 *
## X32         -1.20650    0.17650  -6.836 2.90e-11 ***
## X34          2.96720    0.66140   4.486 9.38e-06 ***
## X35         -0.16604    0.03654  -4.544 7.25e-06 ***
## X36          0.75932    0.08473   8.961  < 2e-16 ***
## X37          0.21078    0.04072   5.176 3.53e-07 ***
## X39         -0.14500    0.10460  -1.386 0.166417
## X45         -0.31098    0.09898  -3.142 0.001798 **
## X48          1.00924    0.19262   5.239 2.56e-07 ***
## X52         -0.53224    0.12333  -4.315 1.99e-05 ***
## X53         -1.20612    0.58106  -2.076 0.038529 *
## X55          0.07987    0.03411   2.341 0.019684 *
## X57          0.94846    0.16972   5.588 4.14e-08 ***
## X58          0.61619    0.22236   2.771 0.005834 **
## X59          8.07963    2.40274   3.363 0.000843 ***
## X60         -0.97887    0.51584  -1.898 0.058436 .
## X62          1.15908    0.47979   2.416 0.016128 *
## X63          0.50372    0.14691   3.429 0.000667 ***
## X64         -0.64632    0.10919  -5.919 6.75e-09 ***
## X65         -2.89423    0.57242  -5.056 6.41e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5503 on 418 degrees of freedom
## Multiple R-squared:  0.689,  Adjusted R-squared:  0.6712
## F-statistic: 38.59 on 24 and 418 DF,  p-value: < 2.2e-16
```

```
summary(model5)
```

```
##
## Call:
## lm(formula = y ~ X26 + X27 + X28 + X29 + X30 + X32 + X34 + X35 +
##     X36 + X37 + X39 + X45 + X48 + X52 + X53 + X55 + X57 + X58 +
##     X59 + X60 + X62 + X63 + X64 + X65, data = data, subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0957 -0.2795 -0.0410  0.2789  2.2227
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01355    0.02646   0.512 0.608908
## X26         -0.38622    0.12185  -3.170 0.001639 **
## X27          1.34837    0.88993   1.515 0.130494
## X28          0.60253    0.32887   1.832 0.067644 .
## X29         -7.66543    2.39249  -3.204 0.001459 **
```

```
## X30         -2.61813    1.01926  -2.569 0.010555 *
## X32         -1.20650    0.17650  -6.836 2.90e-11 ***
## X34          2.96720    0.66140   4.486 9.38e-06 ***
## X35         -0.16604    0.03654  -4.544 7.25e-06 ***
## X36          0.75932    0.08473   8.961  < 2e-16 ***
## X37          0.21078    0.04072   5.176 3.53e-07 ***
## X39         -0.14500    0.10460  -1.386 0.166417
## X45         -0.31098    0.09898  -3.142 0.001798 **
## X48          1.00924    0.19262   5.239 2.56e-07 ***
## X52         -0.53224    0.12333  -4.315 1.99e-05 ***
## X53         -1.20612    0.58106  -2.076 0.038529 *
## X55          0.07987    0.03411   2.341 0.019684 *
## X57          0.94846    0.16972   5.588 4.14e-08 ***
## X58          0.61619    0.22236   2.771 0.005834 **
## X59          8.07963    2.40274   3.363 0.000843 ***
## X60         -0.97887    0.51584  -1.898 0.058436 .
## X62          1.15908    0.47979   2.416 0.016128 *
## X63          0.50372    0.14691   3.429 0.000667 ***
## X64         -0.64632    0.10919  -5.919 6.75e-09 ***
## X65         -2.89423    0.57242  -5.056 6.41e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5503 on 418 degrees of freedom
## Multiple R-squared:  0.689,  Adjusted R-squared:  0.6712
## F-statistic: 38.59 on 24 and 418 DF,  p-value: < 2.2e-16
```

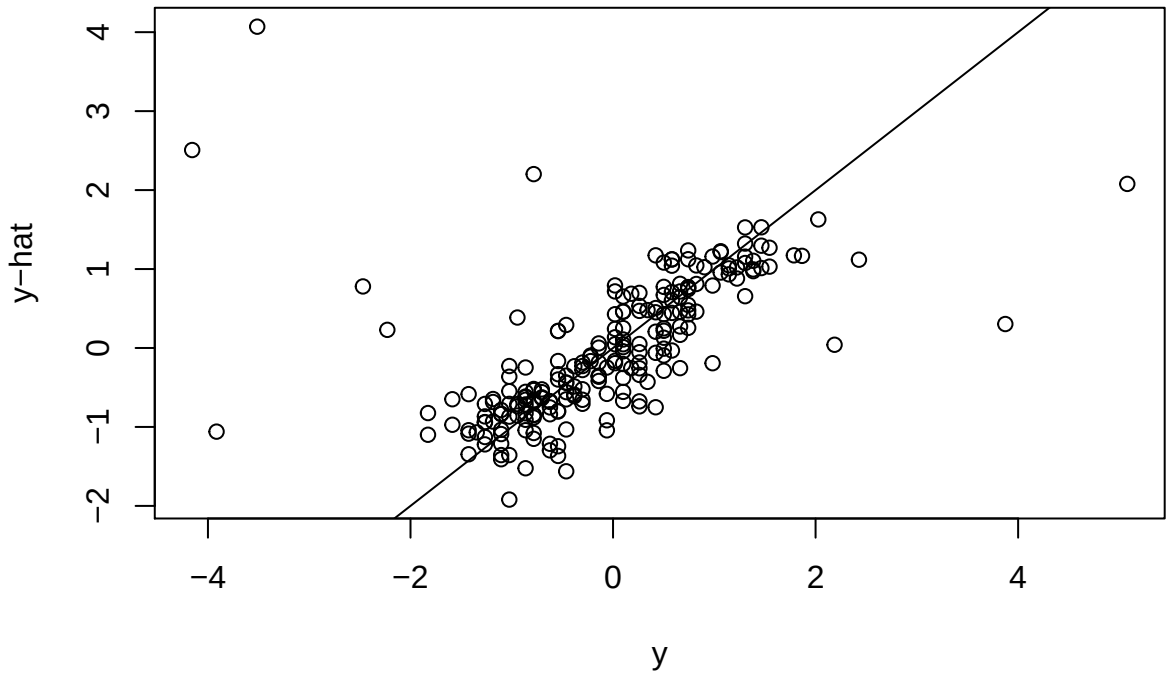
model4 (with backward selection) and model5 (with 'both') are the same. model 4 has smaller Adjusted RS

```
anova(model3, model4)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ X36 + X64 + X54 + X21 + X37 + X63 + X32 + X45 + X26 + X35 +
##     X22 + X29 + X65 + X34 + X23 + X48 + X28 + X53 + X57 + X62 +
##     X55 + X58
## Model 2: y ~ X26 + X27 + X28 + X29 + X30 + X32 + X34 + X35 + X36 + X37 +
##     X39 + X45 + X48 + X52 + X53 + X55 + X57 + X58 + X59 + X60 +
##     X62 + X63 + X64 + X65
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    420 128.14
## 2    418 126.58  2    1.5589 2.5739 0.07745 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anova select the second model, however we achive better MSE on the Test data with our first model

```
#plot(data[train, 'y'], predict(model3, data[train, ]), xlab='y' ,ylab='y-hat')
#abline(c(0,1))
plot(data[test, 'y'], predict(model3, data[test, ]), xlab='y' ,ylab='y-hat')
abline(c(0,1))
```



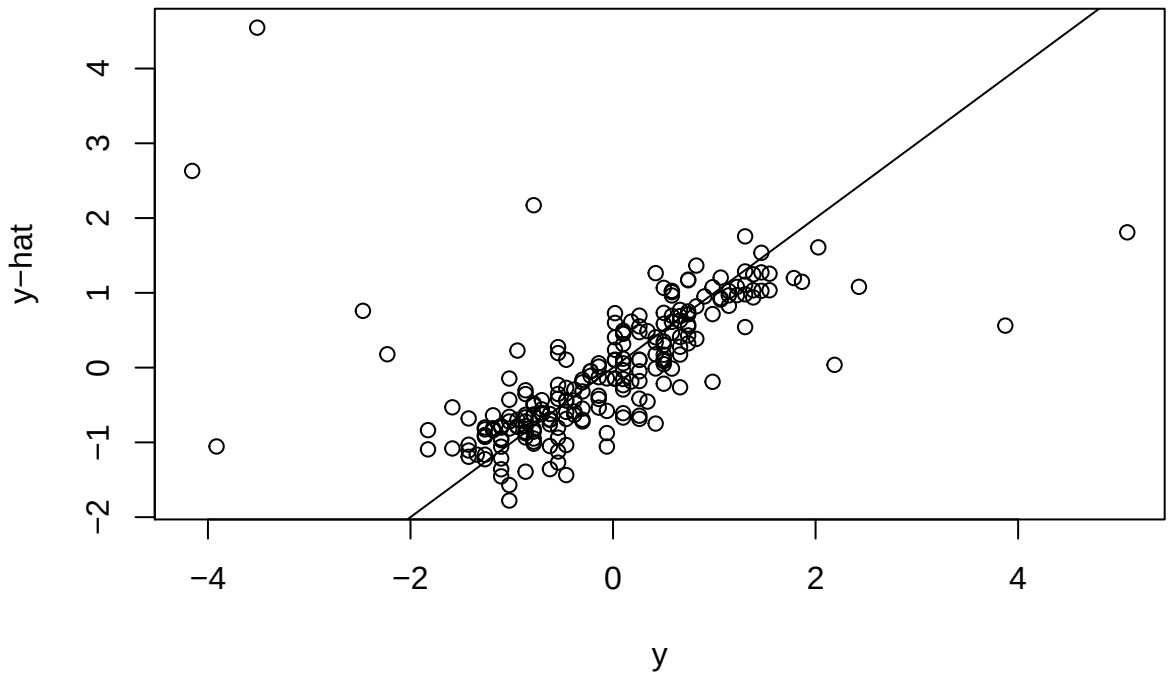

```
#mse_train
mean((data[train, 'y'] - predict(model3, data[train,]))^2)
```

```
## [1] 0.2892473
```

```
#mse_test
mean((data[test, 'y'] - predict(model3, data[test,]))^2)
```

```
## [1] 0.9190005
```

```
plot(data[test, 'y'], predict(model4, data[test, ]), xlab='y' ,ylab='y-hat')
abline(c(0,1))
```



```r
#mse_train
mean((data[train, 'y'] - predict(model4, data[train,]))^2)
```

```
## [1] 0.2857285
```

```r
#mse_test
mean((data[test, 'y'] - predict(model4, data[test,]))^2)
```

```
## [1] 0.9478807
```

3. Best subset regression

```r
library(leaps)
model.rs <- regsubsets(reducedf1, data=data, subset=train, really.big=TRUE, nvmax=10)
summary(model.rs)
```

```
## Subset selection object
## Call: regsubsets.formula(reducedf1, data = data, subset = train, really.big = TRUE,
##     nvmax = 10)
## 45 Variables  (and intercept)
##     Forced in Forced out
## X20     FALSE      FALSE
## X21     FALSE      FALSE
## X22     FALSE      FALSE
## X23     FALSE      FALSE
## X24     FALSE      FALSE
## X25     FALSE      FALSE
## X26     FALSE      FALSE
## X27     FALSE      FALSE
## X28     FALSE      FALSE
## X29     FALSE      FALSE
## X30     FALSE      FALSE
## X31     FALSE      FALSE
## X32     FALSE      FALSE
## X33     FALSE      FALSE
## X34     FALSE      FALSE
## X35     FALSE      FALSE
## X36     FALSE      FALSE
## X37     FALSE      FALSE
## X38     FALSE      FALSE
## X39     FALSE      FALSE
## X40     FALSE      FALSE
## X41     FALSE      FALSE
## X42     FALSE      FALSE
## X43     FALSE      FALSE
## X44     FALSE      FALSE
## X45     FALSE      FALSE
## X46     FALSE      FALSE
## X47     FALSE      FALSE
## X48     FALSE      FALSE
## X49     FALSE      FALSE
## X50     FALSE      FALSE
## X51     FALSE      FALSE
```

```
## X52     FALSE      FALSE
## X53     FALSE      FALSE
## X54     FALSE      FALSE
## X55     FALSE      FALSE
## X56     FALSE      FALSE
## X57     FALSE      FALSE
## X58     FALSE      FALSE
## X59     FALSE      FALSE
## X60     FALSE      FALSE
## X62     FALSE      FALSE
## X63     FALSE      FALSE
## X64     FALSE      FALSE
## X65     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           X20 X21 X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35
## 1  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 5  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 6  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "*"
## 7  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "*"
## 8  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "*"
## 9  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "*"
## 10  ( 1 ) " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "*"
##           X36 X37 X38 X39 X40 X41 X42 X43 X44 X45 X46 X47 X48 X49 X50 X51
## 1  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 5  ( 1 )  "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 6  ( 1 )  "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 7  ( 1 )  "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 8  ( 1 )  "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 9  ( 1 )  "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 10  ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
##           X52 X53 X54 X55 X56 X57 X58 X59 X60 X62 X63 X64 X65
## 1  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 1 )  " " " " "*" " " " " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 5  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 6  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 7  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 8  ( 1 )  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 9  ( 1 )  "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 10  ( 1 ) " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
```

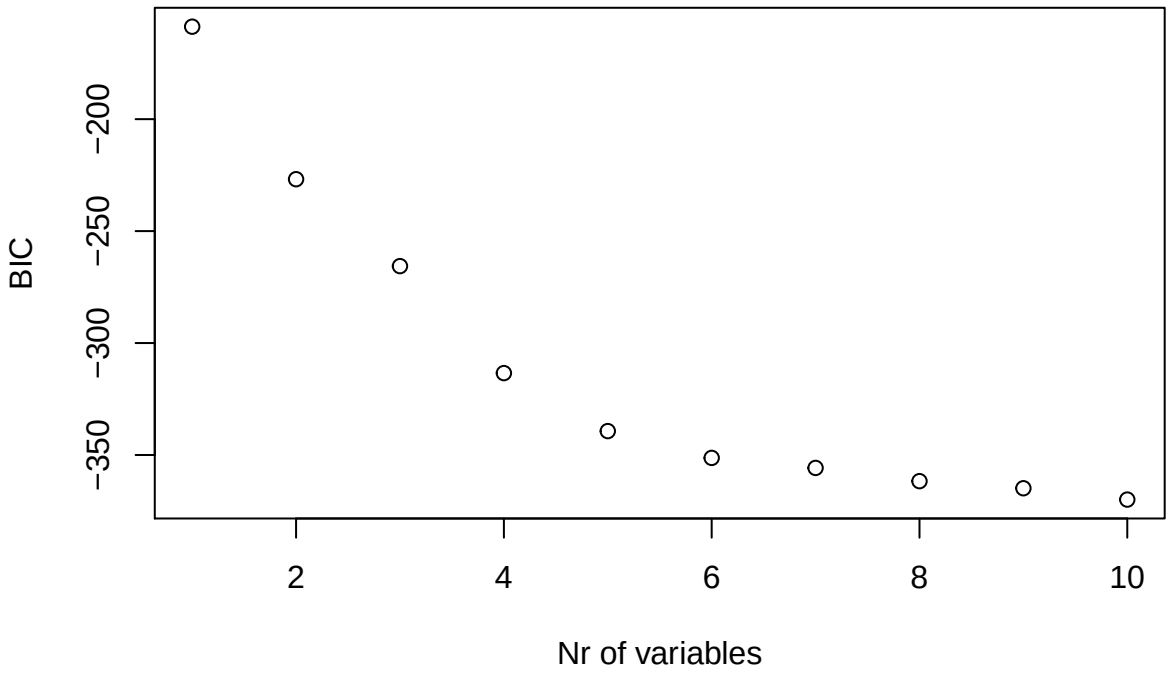
```
s <- summary(model.rs)
str(s)
```

```
## List of 8
##  $ which : logi [1:10, 1:46] TRUE TRUE TRUE TRUE TRUE TRUE ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : chr [1:10] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:46] "(Intercept)" "X20" "X21" "X22" ...
##  $ rsq   : num [1:10] 0.32 0.425 0.48 0.54 0.572 ...
##  $ rss   : num [1:10] 277 234 211 187 174 ...
##  $ adjr2 : num [1:10] 0.319 0.422 0.477 0.536 0.567 ...
##  $ cp    : num [1:10] 451 316 245 169 129 ...
##  $ bic   : num [1:10] -159 -227 -266 -313 -339 ...
##  $ outmat: chr [1:10, 1:45] " " " " " " " " " " " " ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:10] "1  ( 1 )" "2  ( 1 )" "3  ( 1 )" "4  ( 1 )" ...
##   .. ..$ : chr [1:45] "X20" "X21" "X22" "X23" ...
##  $ obj   :List of 28
##   ..$ np       : int 46
##   ..$ nrbar    : int 1035
##   ..$ d        : num [1:46] 443 450 421 109 426 ...
##   ..$ rbar     : num [1:1035] 0.0372 0.0121 0.0293 -0.0228 -0.0025 ...
##   ..$ thetab   : num [1:46] 0.0403 0.4116 0.0195 -0.4756 0.1053 ...
##   ..$ first    : int 2
##   ..$ last     : int 46
##   ..$ vorder   : int [1:46] 1 46 4 41 27 23 31 7 28 3 ...
##   ..$ tol      : num [1:46] 1.05e-08 3.11e-08 1.66e-08 2.68e-08 3.81e-08 ...
##   ..$ rss      : num [1:46] 407 331 331 306 301 ...
##   ..$ bound    : num [1:46] 407 277 234 211 187 ...
##   ..$ nvmax    : int 11
##   ..$ ress     : num [1:11, 1] 407 277 234 211 187 ...
##   ..$ ir       : int 11
##   ..$ nbest    : int 1
##   ..$ lopt     : int [1:66, 1] 1 1 18 1 18 45 1 18 45 36 ...
##   ..$ il       : int 66
##   ..$ ier      : int 0
##   ..$ xnames   : chr [1:46] "(Intercept)" "X20" "X21" "X22" ...
##   ..$ method   : chr "exhaustive"
##   ..$ force.in : Named logi [1:46] TRUE FALSE FALSE FALSE FALSE FALSE ...
##   .. ..- attr(*, "names")= chr [1:46] "" "X20" "X21" "X22" ...
##   ..$ force.out: Named logi [1:46] FALSE FALSE FALSE FALSE FALSE FALSE ...
##   .. ..- attr(*, "names")= chr [1:46] "" "X20" "X21" "X22" ...
##   ..$ sserr    : num 123
##   ..$ intercept: logi TRUE
##   ..$ lindep   : logi [1:46] FALSE FALSE FALSE FALSE FALSE FALSE ...
##   ..$ nullrss  : num 407
##   ..$ nn       : int 443
##   ..$ call     : language regsubsets.formula(reducedf1, data = data, subset = train, really.big = TRU
##   ..- attr(*, "class")= chr "regsubsets"
##  - attr(*, "class")= chr "summary.regsubsets"
```

We plot the BIC of the models

```
plot(1:10, s$bic, xlab='Nr of variables', ylab='BIC')
```



The bigest model with 10 regressors has the best BIC Value, but is not much better than the model with 6 regressors. We train this models and look at the MSE values

first with 10 regressors

```
bestformula10 <- paste0("y~", paste(setdiff(names(which(s$which[10,])), '(Intercept)'), collapse = '+'))
bestformula10
```

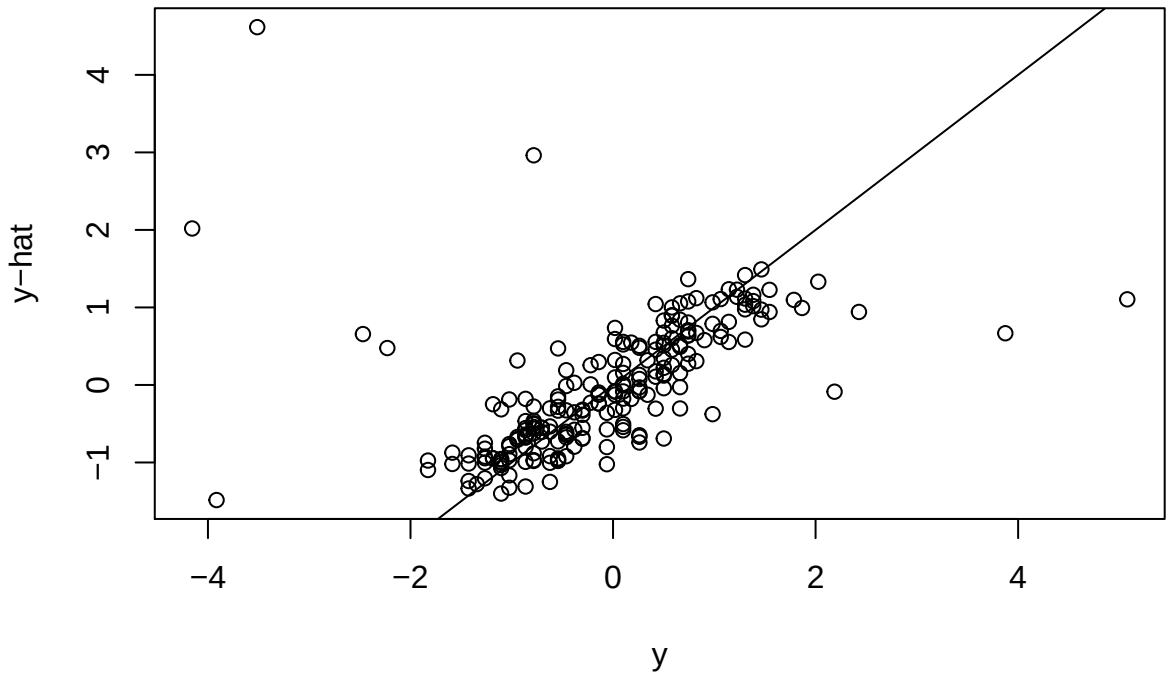
```
## [1] "y~X29+X32+X34+X36+X37+X48+X49+X63+X64+X65"
```

```
model.best10 <- lm(bestformula10, data, train)
summary(model.best10)
```

```
##
## Call:
## lm(formula = bestformula10, data = data, subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.06978 -0.32957 -0.00982  0.29868  3.11980
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01264    0.02837   0.445    0.656
## X29          0.49270    0.07686   6.410 3.82e-10 ***
## X32         -0.28542    0.03877  -7.362 9.25e-13 ***
## X34          2.37413    0.55357   4.289 2.22e-05 ***
## X36          0.56194    0.07132   7.879 2.70e-14 ***
## X37          0.28537    0.03995   7.143 3.90e-12 ***
## X48          0.68948    0.10801   6.383 4.48e-10 ***
## X49         -0.70309    0.13052  -5.387 1.18e-07 ***
## X63          0.47793    0.08926   5.354 1.40e-07 ***
## X64         -0.77379    0.06716 -11.522  < 2e-16 ***
## X65         -2.81726    0.53140  -5.302 1.84e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5928 on 432 degrees of freedom
## Multiple R-squared:  0.627,  Adjusted R-squared:  0.6184
## F-statistic: 72.63 on 10 and 432 DF,  p-value: < 2.2e-16
```

```
plot(data[test, 'y'], predict(model.best10, data[test, ]), xlab='y' ,ylab='y-hat')
abline(c(0,1))
```




```
#mse_test
mean((data[test, 'y'] - predict(model.best10, data[test,]))^2)
```

```
## [1] 0.9573281
```

The MSE Value is slightly worse then in our model with stepwise selection. On the other hand is this model much smaller (10 vs 24 variables).

then we look at the model with 6 regressors

```
bestformula6 <- paste0("y~", paste(setdiff(names(which(s$which[6,])), '(Intercept)'), collapse = '+'))
bestformula6
```

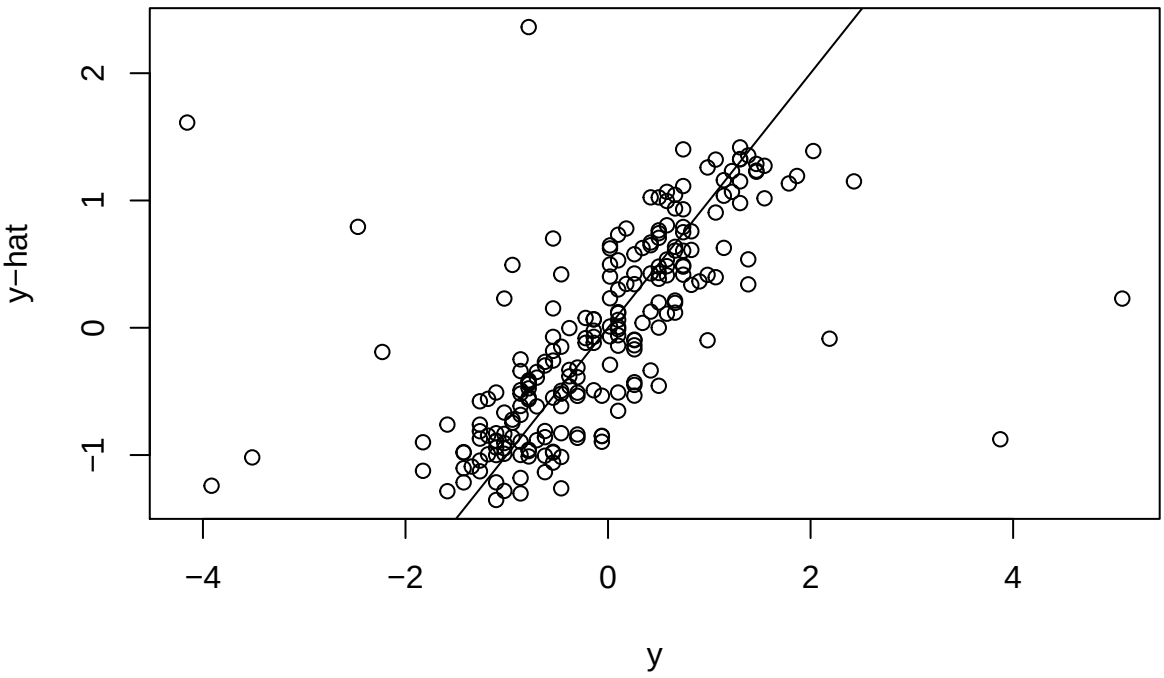
```
## [1] "y~X35+X36+X37+X47+X48+X52"
```

```
model.best6 <- lm(bestformula6, data, train)
summary(model.best6)
```

```
##
## Call:
## lm(formula = bestformula6, data = data, subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7036 -0.3151 -0.0180  0.2749  3.9611
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01171    0.02954   0.397    0.692
## X35         -0.13041    0.03062  -4.259 2.51e-05 ***
## X36          0.51850    0.03686  14.065  < 2e-16 ***
## X37          0.21377    0.03389   6.307 6.98e-10 ***
## X47         -1.15266    0.07920 -14.553  < 2e-16 ***
## X48          1.26049    0.08921  14.129  < 2e-16 ***
## X52         -0.42557    0.04515  -9.425  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6194 on 436 degrees of freedom
## Multiple R-squared:  0.589,  Adjusted R-squared:  0.5834
## F-statistic: 104.2 on 6 and 436 DF,  p-value: < 2.2e-16
```

```
plot(data[test, 'y'], predict(model.best6, data[test, ]), xlab='y' ,ylab='y-hat')
abline(c(0,1))
```




```
#mse_test
mean((data[test, 'y'] - predict(model.best6, data[test,]))^2)
```

```
## [1] 0.7366839
```

This model predicts test data with the best MSE performance by far. with only 6 regressors