Exercise 2

Nikolaus Czernin

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("knitr")
# install.packages("cvTools")
library("cvTools")
```

```
## Loading required package: lattice
```

```
## Loading required package: robustbase
```

```
# install.packages("leaps", type = "binary")
library("leaps")

myseed <- 11721138
set.seed(myseed)
```

```
load("building.RData")
```

```
N <- nrow(df)
train_ids <- sample(1:N, (N %/% 3) * 2)
train <- df[train_ids, ]
test <- df[-train_ids, ]
```

1: Full model

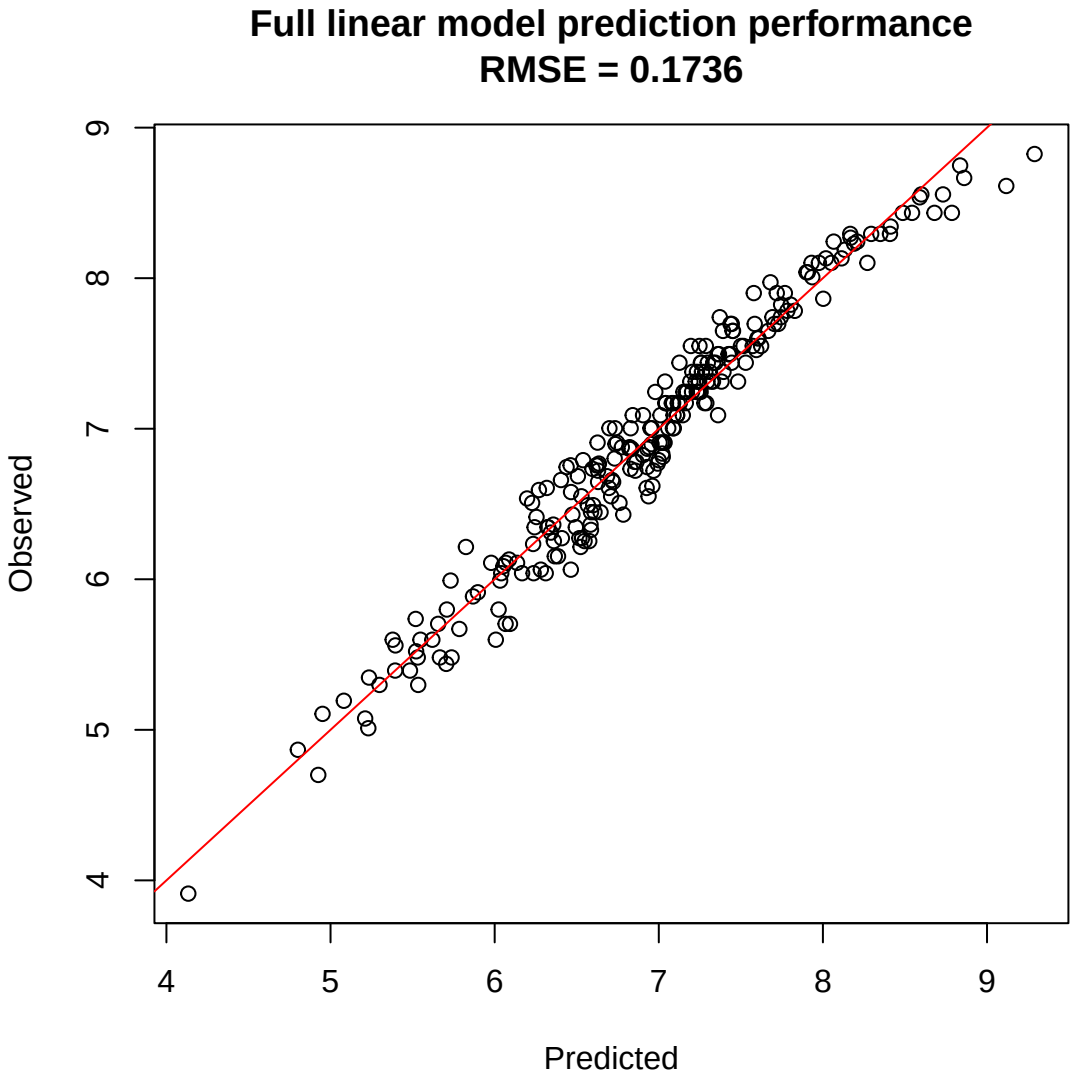
```r
rmse <- function(residuals, r=4){
  residuals^2 %>%
    mean() %>%
    sqrt() %>%
    round(r)
}

lm.full <- lm(y ~ ., train)

plot(predict(lm.full, train), train$y,
     main="Full linear model prediction performance\nRMSE =" %>% paste(rmse(lm.full$residuals)),
     xlab="Predicted",
     ylab="Observed"
     )
```

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

```r
abline(coef = c(0,1), col="red")
```

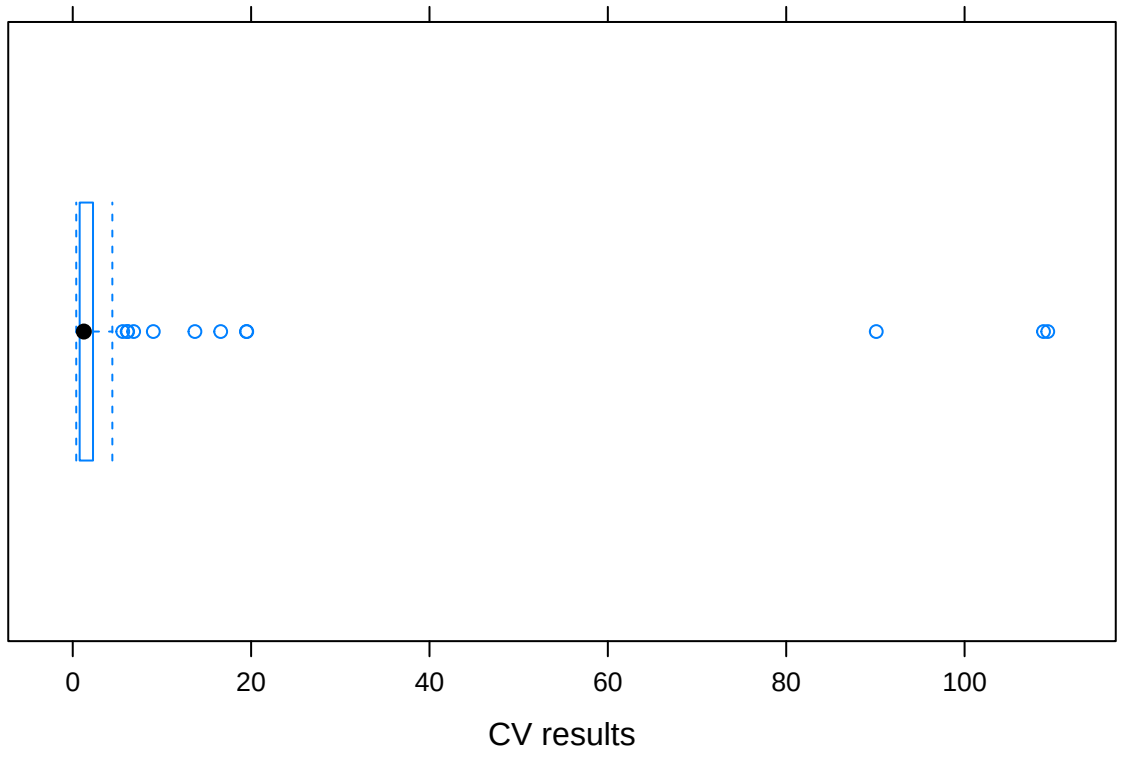


The training error is very small here, close to 1 even. This indicates a fine fit, but we may be overfitting here I think.

```
suppressWarnings(
  cv_results <- cvFit(lm.full, data = df, y = df$y,
                      cost = rmspe,
                      K = 5, ,
                      R = 100,
                      seed=myseed)
)
cv_results %>%  print()
```

```
## 5-fold CV results:
##       CV
## 5.319878
```

```
cv_results %>%   plot()
```

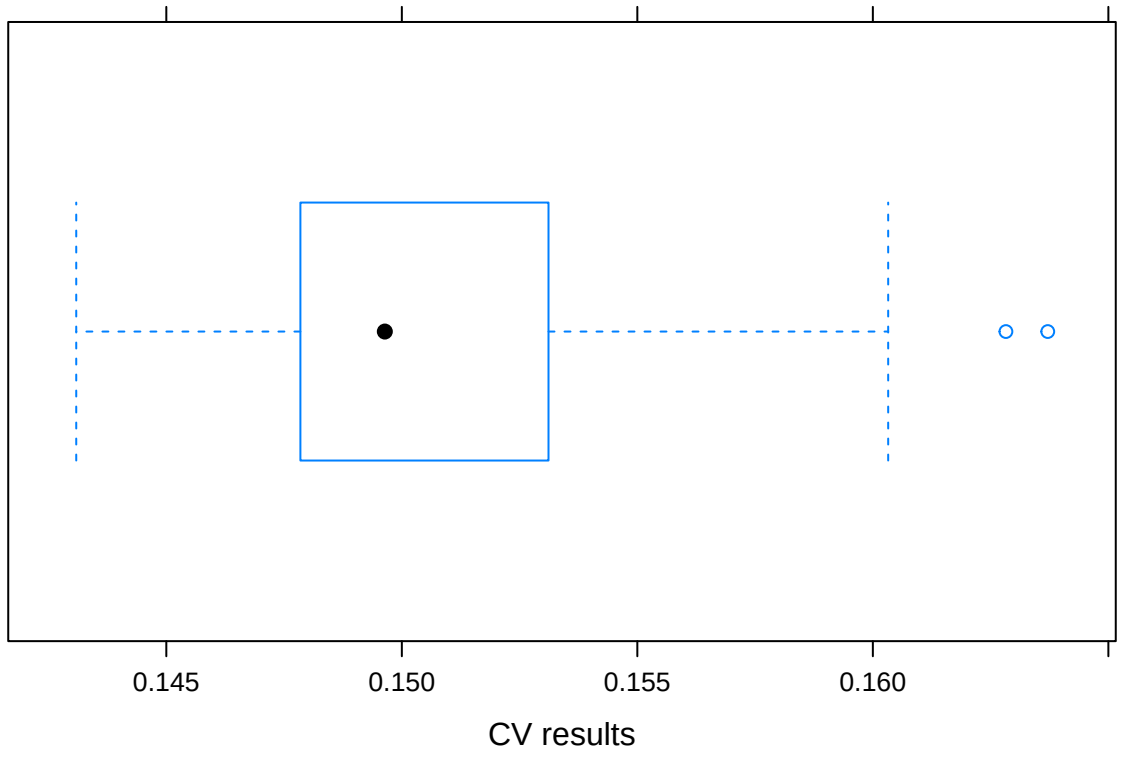


Using 100 rounds of 5-fold cross validation (CV) on the whole dataset, we get an average RMSE of ~5.3, which is a little higher than when using a single split like before.
When looking at the resulting boxplot of the CV, we see the meat of the results clustering near zero, which is a good thing and in line with our previous result. There are some statistical outliers, i.e. rounds where the error measure was really high. So many outliers may indicate that the model overfits on some splits. There may also be some observations that really leverage the model, wrecking its performance when being used in training or testing.

```
suppressWarnings(
  cv_results.t <- cvFit(lm.full, data = df, y = df$y,
                        cost = rtmspe,
                        K = 5, ,
                        R = 100,
                        seed=myseed)
)
cv_results.t %>%  print()
```

```
## 5-fold CV results:
##        CV
## 0.1507312
```

```
cv_results.t %>%   plot()
```

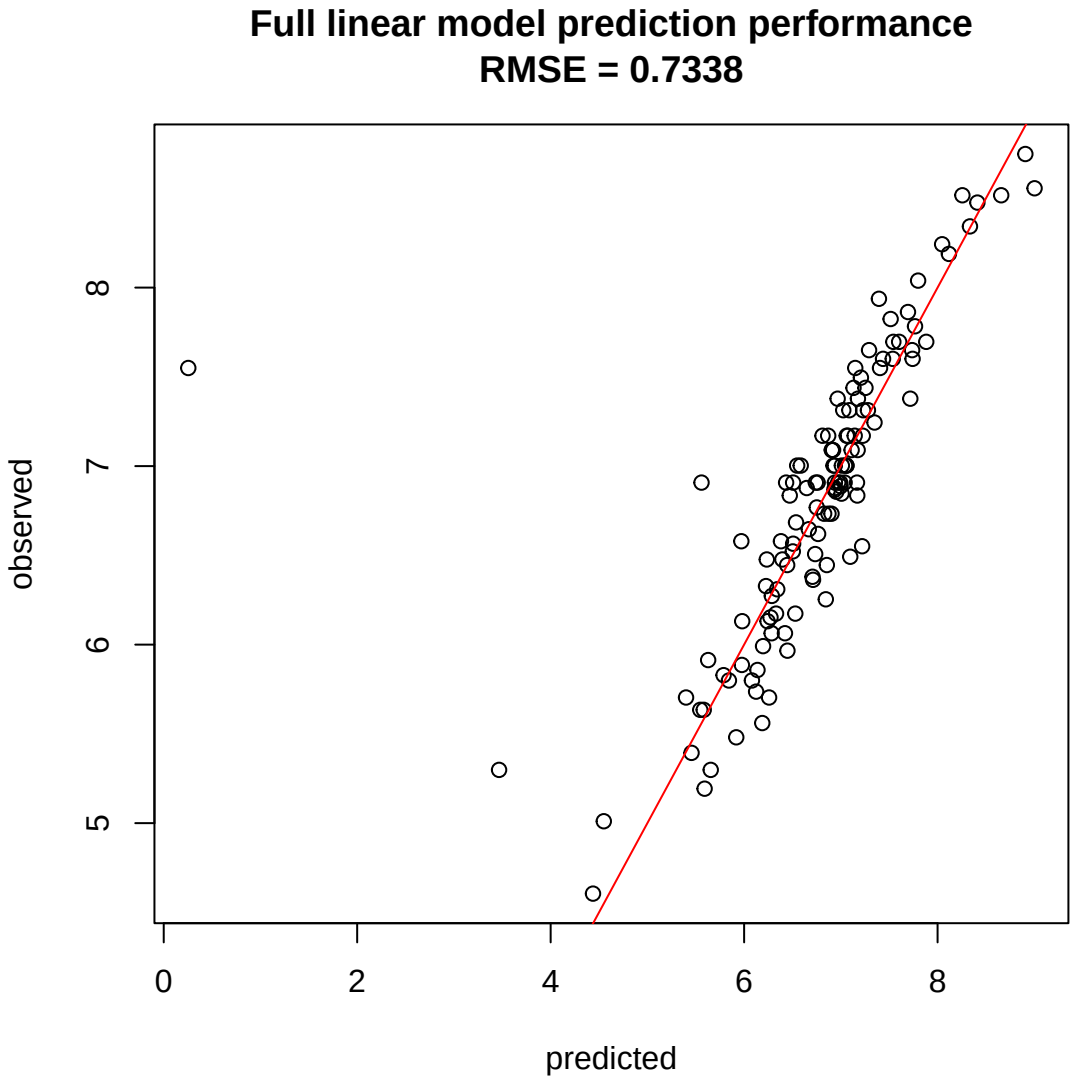


When using the RTMSPE, the Root Trimmed Mean Squared Error, we prune the outliers in terms of errors to get a less pessimistic view on the results. The resulting mean loss is now ~0.1507, a vast improvement. The plot also looks more promising and legible. Only two outliers remain, and well below 1 too.

```
yhat <- predict(lm.full, test)
```

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

```
plot(yhat, test$y, xlab="predicted", ylab="observed",
      main="Full linear model prediction performance\nRMSE =" %>% paste(rmse(yhat - test$y))
     )
abline(coef = c(0,1), col="red")
```



The RMSE is a little higher in the testing than in training altogether, so there may have been overfitting after all. There is a likely leveraging point in the test data, which the model highly underestimated. Perhaps this point alone influenced the model performance enough to get it such poor results compared to training.

2. Best subset regression

```
summary(lm.full)
```

```
##
## Call:
## lm(formula = y ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.50472 -0.10788  0.00561  0.11410  0.38962
##
## Coefficients: (37 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.391e+01  1.854e+02   0.291 0.771594
## START.YEAR          -7.305e-01  2.867e+00  -0.255 0.799166
## START.QUARTER       -1.174e+00  8.138e-01  -1.443 0.150741
## COMPLETION.YEAR      5.016e-02  3.212e-02   1.562 0.120095
## COMPLETION.QUARTER   1.822e-02  1.722e-02   1.058 0.291541
## PhysFin1            -3.714e-02  4.359e-03  -8.520 6.78e-15 ***
## PhysFin2             3.749e-05  3.754e-05   0.999 0.319321
## PhysFin3            -1.881e-04  1.056e-04  -1.782 0.076509 .
## PhysFin4            -1.156e-05  8.396e-05  -0.138 0.890647
## PhysFin5            -3.380e-03  6.008e-04  -5.626 7.08e-08 ***
## PhysFin6             7.484e-04  1.612e-04   4.642 6.69e-06 ***
## PhysFin7                    NA         NA      NA       NA
## PhysFin8             4.380e-04  2.808e-05  15.596  < 2e-16 ***
## Econ1                8.038e-05  3.633e-04   0.221 0.825172
## Econ2               -3.495e-01  2.188e-01  -1.597 0.112018
## Econ3                4.631e-02  1.284e-01   0.361 0.718879
## Econ4               -3.367e-02  3.277e-01  -0.103 0.918266
## Econ5               -1.637e-05  2.076e-05  -0.788 0.431603
## Econ6                4.372e-04  4.988e-04   0.876 0.381960
## Econ7               -6.570e-02  4.982e-02  -1.319 0.188948
## Econ8                1.474e-03  6.011e-03   0.245 0.806604
## Econ9               -1.161e-04  1.046e-04  -1.110 0.268612
## Econ10               1.504e-01  5.904e-01   0.255 0.799157
## Econ11              -3.024e-03  1.480e-03  -2.043 0.042549 *
## Econ12              -1.088e-03  2.002e-03  -0.544 0.587352
## Econ13              -2.500e-04  1.830e-04  -1.366 0.173671
## Econ14               7.251e-04  5.955e-04   1.218 0.224979
## Econ15              -1.874e-01  2.171e-01  -0.864 0.388993
## Econ16               8.183e-01  2.352e-01   3.478 0.000635 ***
## Econ17               4.237e-04  2.528e-04   1.676 0.095541 .
## Econ18              -6.407e-05  1.656e-04  -0.387 0.699340
## Econ19              -3.929e-06  3.767e-06  -1.043 0.298412
## Econ1.lag1          -2.823e-04  2.941e-04  -0.960 0.338401
## Econ2.lag1          -1.717e-01  3.471e-01  -0.495 0.621433
## Econ3.lag1          -8.135e-02  8.253e-02  -0.986 0.325662
## Econ4.lag1           3.392e-01  3.699e-01   0.917 0.360468
## Econ5.lag1          -1.262e-05  2.841e-05  -0.444 0.657297
## Econ6.lag1           8.414e-05  2.147e-04   0.392 0.695566
## Econ7.lag1           1.545e-02  5.086e-02   0.304 0.761573
## Econ8.lag1          -8.491e-03  4.311e-03  -1.970 0.050416 .
## Econ9.lag1           1.721e-04  1.300e-04   1.323 0.187379
## Econ10.lag1         -6.428e-01  7.819e-01  -0.822 0.412157
## Econ11.lag1         -3.705e-03  3.457e-03  -1.072 0.285247
## Econ12.lag1          1.358e-03  2.600e-03   0.522 0.602193
## Econ13.lag1          3.990e-04  1.247e-04   3.201 0.001626 **
## Econ14.lag1          9.353e-04  3.713e-04   2.519 0.012644 *
## Econ15.lag1          1.940e-01  4.008e-01   0.484 0.629052
## Econ16.lag1         -5.326e-01  6.696e-01  -0.795 0.427498
## Econ17.lag1          1.127e-04  2.539e-04   0.444 0.657568
## Econ18.lag1         -9.340e-05  1.250e-04  -0.747 0.456091
```

```
## Econ19.lag1       -7.154e-06  3.490e-06  -2.050 0.041853 *
## Econ1.lag2        -6.582e-06  2.823e-04  -0.023 0.981423
## Econ2.lag2         6.476e-01  6.600e-01   0.981 0.327856
## Econ3.lag2        -2.952e-02  6.226e-02  -0.474 0.635949
## Econ4.lag2        -8.962e-01  4.474e-01  -2.003 0.046683 *
## Econ5.lag2         6.338e-05  2.596e-05   2.442 0.015607 *
## Econ6.lag2         8.687e-05  9.330e-04   0.093 0.925920
## Econ7.lag2        -2.784e-03  7.552e-02  -0.037 0.970633
## Econ8.lag2        -1.135e-02  5.911e-03  -1.920 0.056417 .
## Econ9.lag2         1.120e-04  1.581e-04   0.709 0.479555
## Econ10.lag2        1.075e+00  9.190e-01   1.170 0.243763
## Econ11.lag2       -3.214e-03  4.494e-03  -0.715 0.475477
## Econ12.lag2        2.859e-03  4.210e-03   0.679 0.497967
## Econ13.lag2        4.689e-05  1.565e-04   0.300 0.764833
## Econ14.lag2       -3.016e-04  5.473e-04  -0.551 0.582264
## Econ15.lag2        1.446e-01  4.487e-01   0.322 0.747548
## Econ16.lag2       -4.128e-01  5.801e-01  -0.712 0.477706
## Econ17.lag2        5.639e-06  6.522e-04   0.009 0.993112
## Econ18.lag2       -3.535e-05  1.434e-04  -0.247 0.805573
## Econ19.lag2        5.425e-06  4.505e-06   1.204 0.230135
## Econ1.lag3         6.997e-04  3.644e-04   1.920 0.056413 .
## Econ2.lag3        -1.795e-01  6.337e-01  -0.283 0.777260
## Econ3.lag3                NA         NA      NA       NA
## Econ4.lag3                NA         NA      NA       NA
## Econ5.lag3                NA         NA      NA       NA
## Econ6.lag3                NA         NA      NA       NA
## Econ7.lag3                NA         NA      NA       NA
## Econ8.lag3                NA         NA      NA       NA
## Econ9.lag3                NA         NA      NA       NA
## Econ10.lag3               NA         NA      NA       NA
## Econ11.lag3               NA         NA      NA       NA
## Econ12.lag3               NA         NA      NA       NA
## Econ13.lag3               NA         NA      NA       NA
## Econ14.lag3               NA         NA      NA       NA
## Econ15.lag3               NA         NA      NA       NA
## Econ16.lag3               NA         NA      NA       NA
## Econ17.lag3               NA         NA      NA       NA
## Econ18.lag3               NA         NA      NA       NA
## Econ19.lag3               NA         NA      NA       NA
## Econ1.lag4                NA         NA      NA       NA
## Econ2.lag4                NA         NA      NA       NA
## Econ3.lag4                NA         NA      NA       NA
## Econ4.lag4                NA         NA      NA       NA
## Econ5.lag4                NA         NA      NA       NA
## Econ6.lag4                NA         NA      NA       NA
## Econ7.lag4                NA         NA      NA       NA
## Econ8.lag4                NA         NA      NA       NA
## Econ9.lag4                NA         NA      NA       NA
## Econ10.lag4               NA         NA      NA       NA
## Econ11.lag4               NA         NA      NA       NA
## Econ12.lag4               NA         NA      NA       NA
## Econ13.lag4               NA         NA      NA       NA
## Econ14.lag4               NA         NA      NA       NA
## Econ15.lag4               NA         NA      NA       NA
```

```
## Econ16.lag4                NA         NA      NA       NA
## Econ17.lag4                NA         NA      NA       NA
## Econ18.lag4                NA         NA      NA       NA
## Econ19.lag4                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2054 on 177 degrees of freedom
## Multiple R-squared:  0.9596, Adjusted R-squared:  0.9436
## F-statistic: 60.02 on 70 and 177 DF,  p-value: < 2.2e-16
```

I chose to preselect only the variables whose coefficients were marked as significantly unequal to 0 in the lm() function, which were the following 11:
- PhysFin1
- PhysFin5
- PhysFin6
- PhysFin8
- Econ11
- Econ16
- Econ13.lag1
- Econ14.lag1
- Econ19.lag1
- Econ4.lag2
- Econ5.lag2

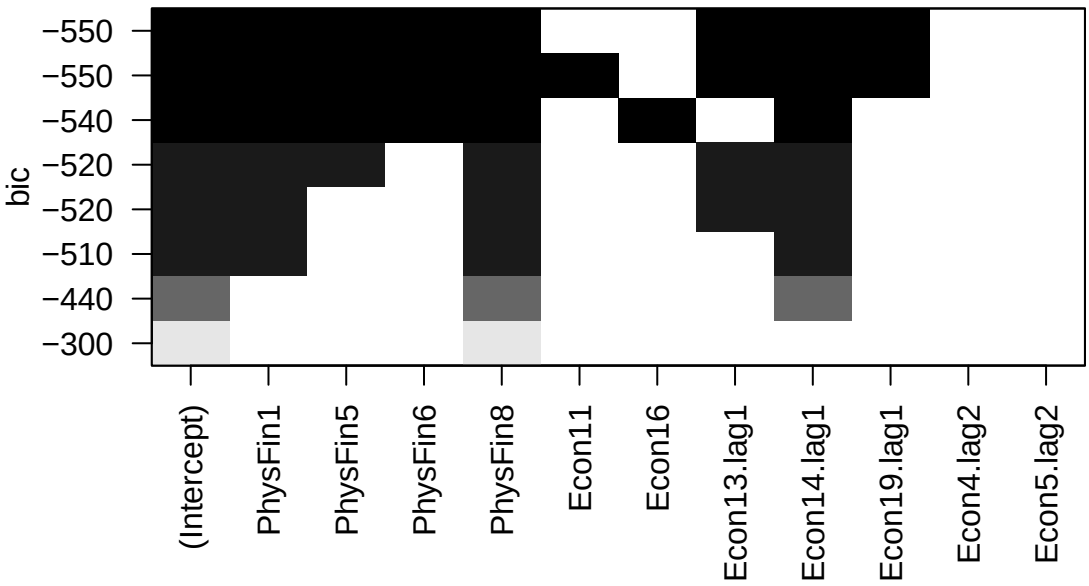
```
results_subsets <- regsubsets(y~ PhysFin1 + PhysFin5 + PhysFin6 + PhysFin8 + Econ11 + Econ16 +
             Econ13.lag1 + Econ14.lag1 + Econ19.lag1 + Econ4.lag2 + Econ5.lag2
           , data=train)

results_subsets %>% summary()
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ PhysFin1 + PhysFin5 + PhysFin6 + PhysFin8 +
##     Econ11 + Econ16 + Econ13.lag1 + Econ14.lag1 + Econ19.lag1 +
##     Econ4.lag2 + Econ5.lag2, data = train)
## 11 Variables  (and intercept)
##             Forced in Forced out
## PhysFin1        FALSE      FALSE
## PhysFin5        FALSE      FALSE
## PhysFin6        FALSE      FALSE
## PhysFin8        FALSE      FALSE
## Econ11          FALSE      FALSE
## Econ16          FALSE      FALSE
## Econ13.lag1     FALSE      FALSE
## Econ14.lag1     FALSE      FALSE
## Econ19.lag1     FALSE      FALSE
## Econ4.lag2      FALSE      FALSE
## Econ5.lag2      FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          PhysFin1 PhysFin5 PhysFin6 PhysFin8 Econ11 Econ16 Econ13.lag1
## 1  ( 1 ) " "      " "      " "      "*"      " "    " "    " "
## 2  ( 1 ) " "      " "      " "      "*"      " "    " "    " "
```

```
## 3  ( 1 ) "*"      " "      " "      "*"      " "    " "    " "
## 4  ( 1 ) "*"      " "      " "      "*"      " "    " "    "*"
## 5  ( 1 ) "*"      "*"      " "      "*"      " "    " "    "*"
## 6  ( 1 ) "*"      "*"      "*"      "*"      " "    "*"    " "
## 7  ( 1 ) "*"      "*"      "*"      "*"      " "    " "    "*"
## 8  ( 1 ) "*"      "*"      "*"      "*"      "*"    " "    "*"
##          Econ14.lag1 Econ19.lag1 Econ4.lag2 Econ5.lag2
## 1  ( 1 ) " "         " "         " "        " "
## 2  ( 1 ) "*"         " "         " "        " "
## 3  ( 1 ) "*"         " "         " "        " "
## 4  ( 1 ) "*"         " "         " "        " "
## 5  ( 1 ) "*"         " "         " "        " "
## 6  ( 1 ) "*"         " "         " "        " "
## 7  ( 1 ) "*"         "*"         " "        " "
## 8  ( 1 ) "*"         "*"         " "        " "
```

```
results_subsets %>% plot()
```



Judging from the resulting plot, the model performance saturates at a BIC of ~520 or 540, in which case it would require either 6 or 8 variables (including the intercept which is no variable of course). Let's go with the latter.
- intercept - PhysFin1
- PhysFin5
- PhysFin6
- PhysFin8
- Econ16

- Econ14.lag1

Now on to use these variables again in another shot at the `lm()` function:

```
lm.slim <- lm(y ~PhysFin1 + PhysFin5 + PhysFin6 + PhysFin8 + Econ16 + Econ14.lag1,
              data=train)
lm.slim %>% summary()
```

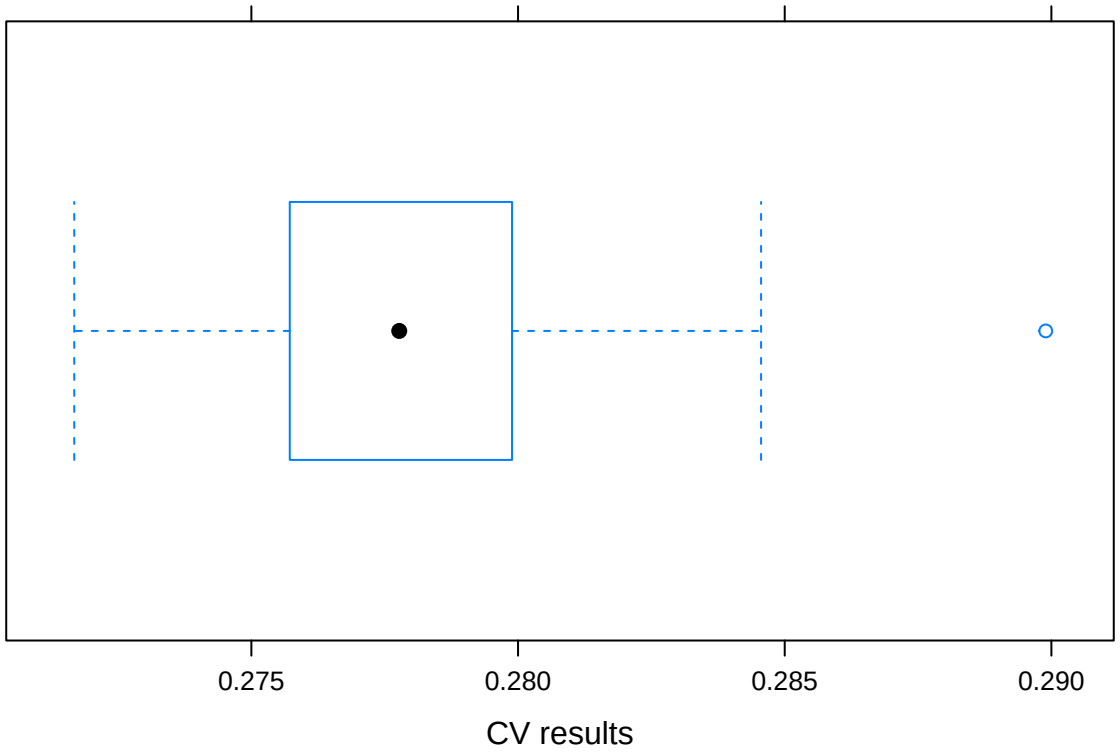
```
##
## Call:
## lm(formula = y ~ PhysFin1 + PhysFin5 + PhysFin6 + PhysFin8 +
##     Econ16 + Econ14.lag1, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93764 -0.18155  0.03297  0.16561  0.68042
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.873e+00  1.156e-01  42.145  < 2e-16 ***
## PhysFin1    -3.486e-02  4.458e-03  -7.821 1.64e-13 ***
## PhysFin5    -3.100e-03  5.278e-04  -5.873 1.41e-08 ***
## PhysFin6     6.289e-04  1.168e-04   5.383 1.73e-07 ***
## PhysFin8     4.469e-04  3.365e-05  13.283  < 2e-16 ***
## Econ16       6.869e-03  1.082e-03   6.346 1.09e-09 ***
## Econ14.lag1  1.868e-04  1.402e-05  13.324  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2744 on 241 degrees of freedom
## Multiple R-squared:  0.9018, Adjusted R-squared:  0.8993
## F-statistic: 368.8 on 6 and 241 DF,  p-value: < 2.2e-16
```

```
# lm.slim %>% plot()
```

```
suppressWarnings(
  cv_results.slim <- cvFit(lm.slim, data = df, y = df$y,
                       cost = rmspe,
                       K = 5, ,
                       R = 100,
                       seed=myseed)
)
cv_results.slim %>%  print()
```

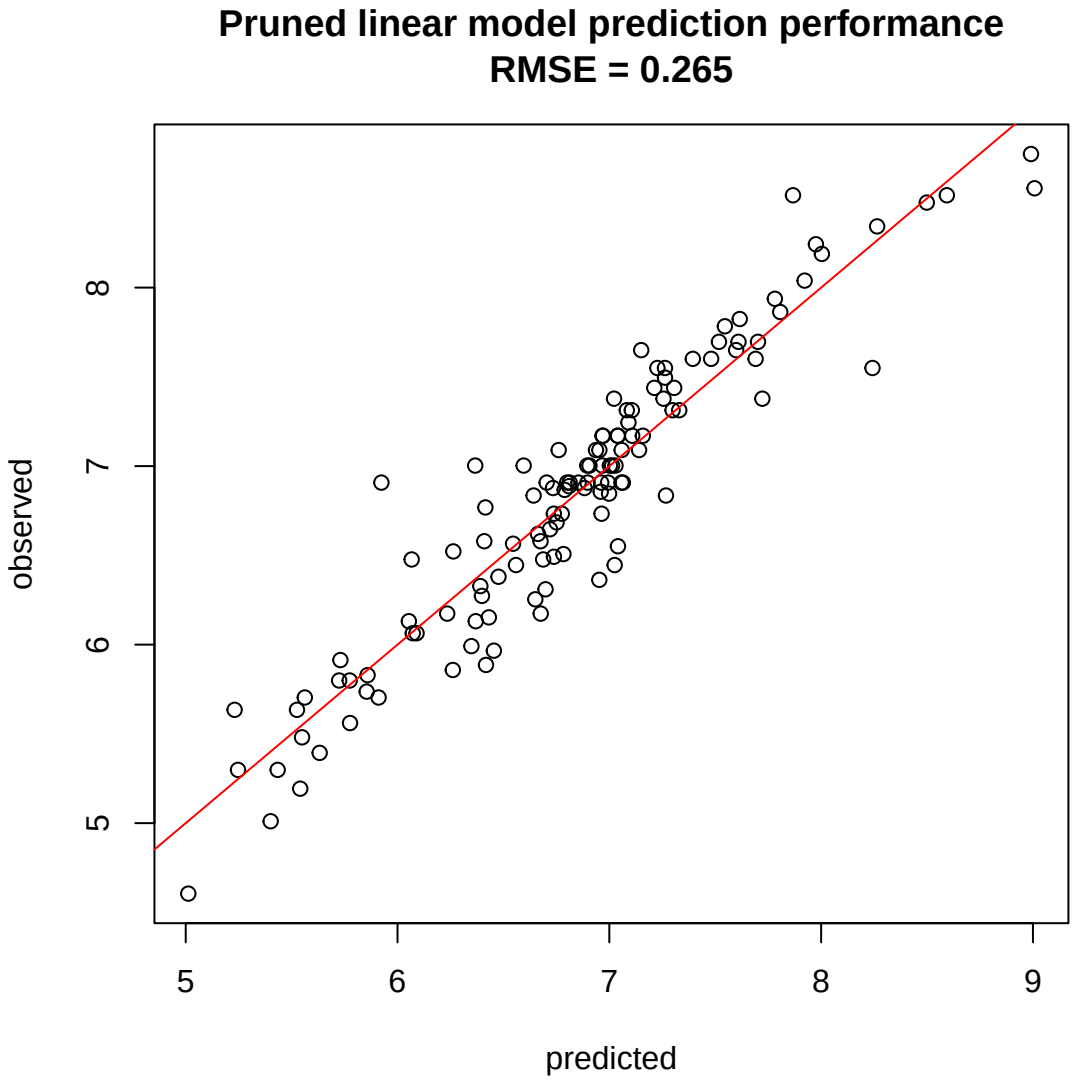
```
## 5-fold CV results:
##        CV
## 0.2779278
```

```
cv_results.slim %>%   plot()
```



The resulting RMSE and boxplot look a lot more like the time we used the trimmed RMSE instead of keeping all rounds. This indicates that pruning the non-significant variables alone improved the general performance of a linear model with the given data on average by a good amount. This way we could hopefully deal with some of the overfitting we were running into earlier.

```
yhat <- predict(lm.slim, test)
plot(yhat, test$y, xlab="predicted", ylab="observed",
      main="Pruned linear model prediction performance\nRMSE =" %>% paste(rmse(yhat - test$y))
     )
abline(coef = c(0,1), col="red")
```



Not only is the RMSE of this test lower than in the previous model, the plot also shows how the pruning of variables allowed the model to avoid the crass underestimation it made in the previous test.

```
anova(lm.full, lm.slim)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ START.YEAR + START.QUARTER + COMPLETION.YEAR + COMPLETION.QUARTER +
##     PhysFin1 + PhysFin2 + PhysFin3 + PhysFin4 + PhysFin5 + PhysFin6 +
##     PhysFin7 + PhysFin8 + Econ1 + Econ2 + Econ3 + Econ4 + Econ5 +
##     Econ6 + Econ7 + Econ8 + Econ9 + Econ10 + Econ11 + Econ12 +
##     Econ13 + Econ14 + Econ15 + Econ16 + Econ17 + Econ18 + Econ19 +
##     Econ1.lag1 + Econ2.lag1 + Econ3.lag1 + Econ4.lag1 + Econ5.lag1 +
##     Econ6.lag1 + Econ7.lag1 + Econ8.lag1 + Econ9.lag1 + Econ10.lag1 +
##     Econ11.lag1 + Econ12.lag1 + Econ13.lag1 + Econ14.lag1 + Econ15.lag1 +
##     Econ16.lag1 + Econ17.lag1 + Econ18.lag1 + Econ19.lag1 + Econ1.lag2 +
##     Econ2.lag2 + Econ3.lag2 + Econ4.lag2 + Econ5.lag2 + Econ6.lag2 +
```

```
##     Econ7.lag2 + Econ8.lag2 + Econ9.lag2 + Econ10.lag2 + Econ11.lag2 +
##     Econ12.lag2 + Econ13.lag2 + Econ14.lag2 + Econ15.lag2 + Econ16.lag2 +
##     Econ17.lag2 + Econ18.lag2 + Econ19.lag2 + Econ1.lag3 + Econ2.lag3 +
##     Econ3.lag3 + Econ4.lag3 + Econ5.lag3 + Econ6.lag3 + Econ7.lag3 +
##     Econ8.lag3 + Econ9.lag3 + Econ10.lag3 + Econ11.lag3 + Econ12.lag3 +
##     Econ13.lag3 + Econ14.lag3 + Econ15.lag3 + Econ16.lag3 + Econ17.lag3 +
##     Econ18.lag3 + Econ19.lag3 + Econ1.lag4 + Econ2.lag4 + Econ3.lag4 +
##     Econ4.lag4 + Econ5.lag4 + Econ6.lag4 + Econ7.lag4 + Econ8.lag4 +
##     Econ9.lag4 + Econ10.lag4 + Econ11.lag4 + Econ12.lag4 + Econ13.lag4 +
##     Econ14.lag4 + Econ15.lag4 + Econ16.lag4 + Econ17.lag4 + Econ18.lag4 +
##     Econ19.lag4
## Model 2: y ~ PhysFin1 + PhysFin5 + PhysFin6 + PhysFin8 + Econ16 + Econ14.lag1
##   Res.Df     RSS  Df Sum of Sq      F    Pr(>F)
## 1    177  7.4706
## 2    241 18.1497 -64   -10.679 3.9535 3.076e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova test shows that the the smaller model makes a significant improvement. s