University of St.Gallen

Penalized Regression

University of St. Gallen
School of Management, Economics, Law,
Social Sciences, International Affairs
and Computer Science

Assignment 2

Data Analytics I: Predictive Econometrics
Prof. Jana Mareckova

submitted by

Cyril Janak, 16-611-287
Jonas Husmann, 16-610-917
Niklas Kampe, 16-611-618
Robin Scherrer, 18-617-969

08.12.2021

Contents

Requirements

To solve the following tasks, the required libraries and the data sets are loaded first.

```
library(glmnet)
library(corrplot)
library(ggplot2)
library(dplyr)

load("GHA/student-mat-train.RData")
load("GHA/student-mat-test.RData")
```

Exercise 1

There are 214 observations in the training data set and 143 observations in the test data set.

```
(n_obs_train <- nrow(train))
```

```
## [1] 214
```

```
(n_obs_test <- nrow(test))
```

```
## [1] 143
```

Exercise 2

The average grade is ~11.64, the minimum grade is 4 and the maximum grade is 19. All numbers were calculated using the training data.

```
(avg_grade <- mean(train$G3))
```

```
## [1] 11.64019
```

```
(min_grade <- min(train$G3))
```

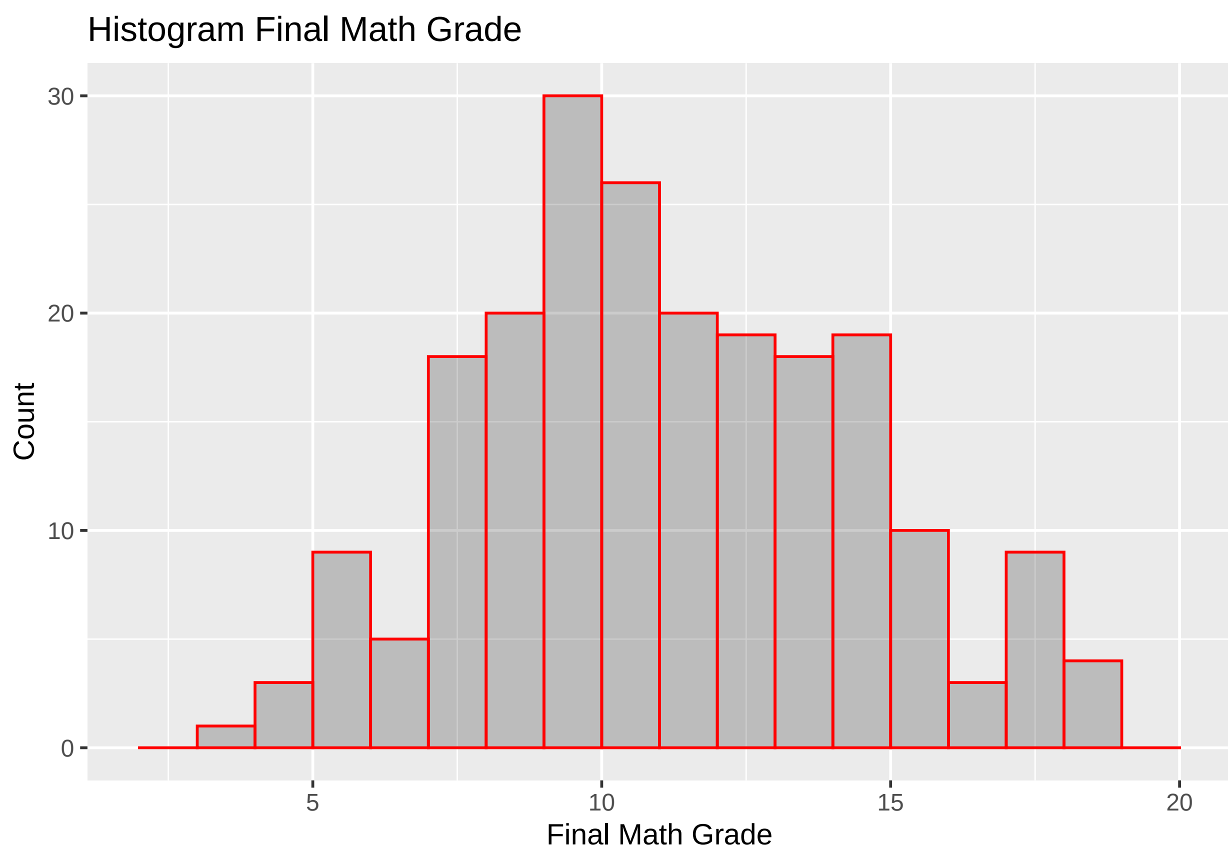
```
## [1] 4
```

```
(max_grade <- max(train$G3))
```

```
## [1] 19
```

Exercise 3

```r
(final_grade_hist <- ggplot(data=train, aes(G3)) +
    geom_histogram(breaks=seq(2,20, by=1),
                   col="red",
                   fill="black",
                   alpha = 0.2)+
    labs(title="Histogram Final Math Grade", x="Final Math Grade", y="Count"))
```



Exercise 4

Predictive modeling is used to predict an object of interest (e.g. forecasting or nowcasting) by using predictors (covariates). The goal is to get as good out-of-sample predictions as possible (e.g. predicting unemployment). The goal of causal modeling, in contrast, is to establish a causal relationship between the explanatory variables (covariates) and the object of interest (e.g. causal effect of inflation, GDP per capita, . . . on unemployment). While for predictive modeling only the full rank condition is mandatory, for causal modeling both the full rank condition and the exclusion restriction must be fulfilled.

Exercise 5

```r
OLS1 <- lm(G3 ~ . ,
            data=select(train, G3, Medu, Fedu, studytime, schoolsup, higher))
(summary(OLS1))
```

```
##
## Call:
## lm(formula = G3 ~ ., data = select(train, G3, Medu, Fedu, studytime,
##     schoolsup, higher))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4668 -2.1690 -0.1981  2.0630  7.0630
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.38701    1.05127   8.929 2.29e-16 ***
## Medu         0.36742    0.24753   1.484   0.1392
## Fedu         0.07675    0.24727   0.310   0.7566
## studytime    0.60662    0.24803   2.446   0.0153 *
## schoolsup   -3.36832    0.67412  -4.997 1.24e-06 ***
## higher       0.77327    1.02224   0.756   0.4502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.041 on 208 degrees of freedom
## Multiple R-squared:  0.1501, Adjusted R-squared:  0.1297
## F-statistic: 7.346 on 5 and 208 DF,  p-value: 2.312e-06
```

```r
OLS2 <- lm(G3 ~ . + .^2,
            data=select(train, G3, Medu, Fedu, studytime, schoolsup, higher))
(summary(OLS2))
```

```
##
## Call:
## lm(formula = G3 ~ . + .^2, data = select(train, G3, Medu, Fedu,
##     studytime, schoolsup, higher))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6603 -2.0887 -0.0921  1.8277  7.8154
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          13.172132   5.437629   2.422  0.01632 *
## Medu                  0.145263   1.064137   0.137  0.89156
## Fedu                 -1.427466   2.050517  -0.696  0.48715
## studytime            -0.464677   2.785748  -0.167  0.86769
## schoolsup             1.920704   4.596735   0.418  0.67652
## higher               -4.432522   5.533668  -0.801  0.42409
## Medu:Fedu            -0.001922   0.217956  -0.009  0.99297
## Medu:studytime        0.105788   0.312408   0.339  0.73525
## Medu:schoolsup       -2.611720   0.899135  -2.905  0.00409 **
## Medu:higher           0.322940   1.040274   0.310  0.75656
## Fedu:studytime       -0.499887   0.298323  -1.676  0.09538 .
## Fedu:schoolsup        1.271388   0.844657   1.505  0.13386
## Fedu:higher           1.939871   2.119975   0.915  0.36128
## studytime:schoolsup -0.210424   0.851999  -0.247  0.80518
## studytime:higher      2.074920   2.739937   0.757  0.44978
## schoolsup:higher     -1.165641   4.726445  -0.247  0.80546
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.01 on 198 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.1476
## F-statistic: 3.459 on 15 and 198 DF,  p-value: 3.007e-05
```

```r
MSE_IS_OLS1 <- mean((train$G3 - OLS1$fitted.values)^2)
MSE_IS_OLS2 <- mean((train$G3 - OLS2$fitted.values)^2)
(MSE_IS <- data.frame(model = c("OLS1_IS", "OLS2_IS"),
                      MSE = c(MSE_IS_OLS1, MSE_IS_OLS2)))
```

```
##     model      MSE
## 1 OLS1_IS 8.988840
## 2 OLS2_IS 8.380403
```

In order to elaborate the in-sample fit of the two models, we define the coefficient of determination R^2 as well as the in-sample MSE as the key fit determinants. From the results, we can observe that the first linear model with five covariates has a R^2 of 0.13 and an in-sample MSE of 8.99, whereas the second linear model including the first order interactions has an R^2 of 0.15 and an in-sample MSE of 8.38. From these results, we can conclude that the second model performns better in both fit coefficients, which is in accordance with the general result that an increased number of covariates often leads to better in-sample fits (or delivers the same model fit). Nevertheless, the both R^2s aand MSEs are relatively low/high, latter compared to the level of the dependent variable, which concludes an overall weak model fit.

Exercise 6

```
OLS3 <- lm(G3 ~ . ,
            data=select(train, G3, Medu, Fedu, studytime, schoolsup, higher, Pstatus,
                        famrel, failures, famsup,internet))
(summary(OLS3))
```

```
##
## Call:
## lm(formula = G3 ~ ., data = select(train, G3, Medu, Fedu, studytime,
##     schoolsup, higher, Pstatus, famrel, failures, famsup, internet))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0295 -2.1703 -0.0742  1.9681  7.1631
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.649486   1.345550   7.171 1.36e-11 ***
## Medu         0.350090   0.248736   1.407   0.1608
## Fedu         0.007509   0.242057   0.031   0.9753
## studytime    0.597455   0.247898   2.410   0.0168 *
## schoolsup   -3.151785   0.657969  -4.790 3.21e-06 ***
## higher       0.284839   1.002100   0.284   0.7765
## Pstatus      0.022675   0.610754   0.037   0.9704
## famrel       0.272672   0.230688   1.182   0.2386
## failures    -1.016545   0.317847  -3.198   0.0016 **
## famsup      -0.891842   0.449153  -1.986   0.0484 *
## internet     0.562597   0.542256   1.038   0.3007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.943 on 203 degrees of freedom
## Multiple R-squared:  0.2233, Adjusted R-squared:  0.185
## F-statistic: 5.836 on 10 and 203 DF,  p-value: 1.013e-07
```

```
OLS4 <- lm(G3 ~ . + .^2,
            data=select(train, G3, Medu, Fedu, studytime, schoolsup, higher, Pstatus,
                        famrel, failures, famsup,internet))
(summary(OLS4))
```

```
##
## Call:
## lm(formula = G3 ~ . + .^2, data = select(train, G3, Medu, Fedu,
##     studytime, schoolsup, higher, Pstatus, famrel, failures,
```

```
##     famsup, internet))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.302 -1.638  0.000  1.569  7.129
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.00486   19.71735   0.406  0.68530
## Medu                 2.65008    2.18285   1.214  0.22653
## Fedu                -2.33596    5.62321  -0.415  0.67840
## studytime           -0.74497    5.53425  -0.135  0.89309
## schoolsup           -1.03745   10.53794  -0.098  0.92170
## higher               0.15984   19.57437   0.008  0.99350
## Pstatus             -3.81423    6.70935  -0.568  0.57050
## famrel               1.04447    4.90832   0.213  0.83176
## failures             0.04181    2.33857   0.018  0.98576
## famsup               3.05249    5.01243   0.609  0.54340
## internet            -1.90817    2.90639  -0.657  0.51242
## Medu:Fedu           -0.03842    0.25236  -0.152  0.87919
## Medu:studytime      -0.15811    0.34987  -0.452  0.65196
## Medu:schoolsup      -2.85143    1.06173  -2.686  0.00801 **
## Medu:higher         -0.54863    1.68739  -0.325  0.74551
## Medu:Pstatus        -0.91279    0.98786  -0.924  0.35689
## Medu:famrel         -0.36014    0.40866  -0.881  0.37951
## Medu:failures       -1.01657    0.63945  -1.590  0.11388
## Medu:famsup          0.74973    0.61857   1.212  0.22730
## Medu:internet       -0.59083    0.67187  -0.879  0.38052
## Fedu:studytime      -0.32178    0.33980  -0.947  0.34510
## Fedu:schoolsup       1.41207    0.96285   1.467  0.14448
## Fedu:higher          1.45072    5.53069   0.262  0.79343
## Fedu:Pstatus         0.10424    0.80597   0.129  0.89725
## Fedu:famrel          0.22857    0.38702   0.591  0.55563
## Fedu:failures        0.34034    0.67323   0.506  0.61389
## Fedu:famsup         -0.49855    0.63597  -0.784  0.43426
## Fedu:internet        1.11916    0.71180   1.572  0.11787
## studytime:schoolsup -0.43875    0.94950  -0.462  0.64465
## studytime:higher     1.74282    5.29531   0.329  0.74249
## studytime:Pstatus    1.28299    1.13311   1.132  0.25923
## studytime:famrel     0.09156    0.28279   0.324  0.74654
## studytime:failures  -0.61083    0.81245  -0.752  0.45326
## studytime:famsup     0.79983    0.63934   1.251  0.21277
## studytime:internet   0.10897    0.81962   0.133  0.89440
## schoolsup:higher     0.67617    9.41038   0.072  0.94281
## schoolsup:Pstatus   -0.95843    2.60822  -0.367  0.71376
## schoolsup:famrel     0.16037    1.43567   0.112  0.91120
## schoolsup:failures   1.93873    1.64480   1.179  0.24028
## schoolsup:famsup     1.18057    2.35659   0.501  0.61709
```

```
## schoolsup:internet   0.27075    2.11487   0.128  0.89830
## higher:Pstatus       2.66107    4.74288   0.561  0.57554
## higher:famrel       -0.14551    4.88194  -0.030  0.97626
## higher:failures      1.51356    1.98140   0.764  0.44607
## higher:famsup       -4.68077    4.62750  -1.012  0.31331
## higher:internet           NA         NA      NA       NA
## Pstatus:famrel       1.04521    0.69232   1.510  0.13310
## Pstatus:failures    -0.19850    1.71396  -0.116  0.90795
## Pstatus:famsup      -0.71384    1.84148  -0.388  0.69880
## Pstatus:internet     0.89706    1.95701   0.458  0.64730
## famrel:failures     -0.26244    0.39874  -0.658  0.51138
## famrel:famsup       -0.72776    0.67389  -1.080  0.28180
## famrel:internet      0.21125    0.72523   0.291  0.77121
## failures:famsup      0.67703    0.98499   0.687  0.49286
## failures:internet   -0.39309    1.02403  -0.384  0.70159
## famsup:internet      0.95906    1.41469   0.678  0.49880
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.919 on 159 degrees of freedom
## Multiple R-squared:  0.4014, Adjusted R-squared:  0.1981
## F-statistic: 1.974 on 54 and 159 DF,  p-value: 0.0006072
```

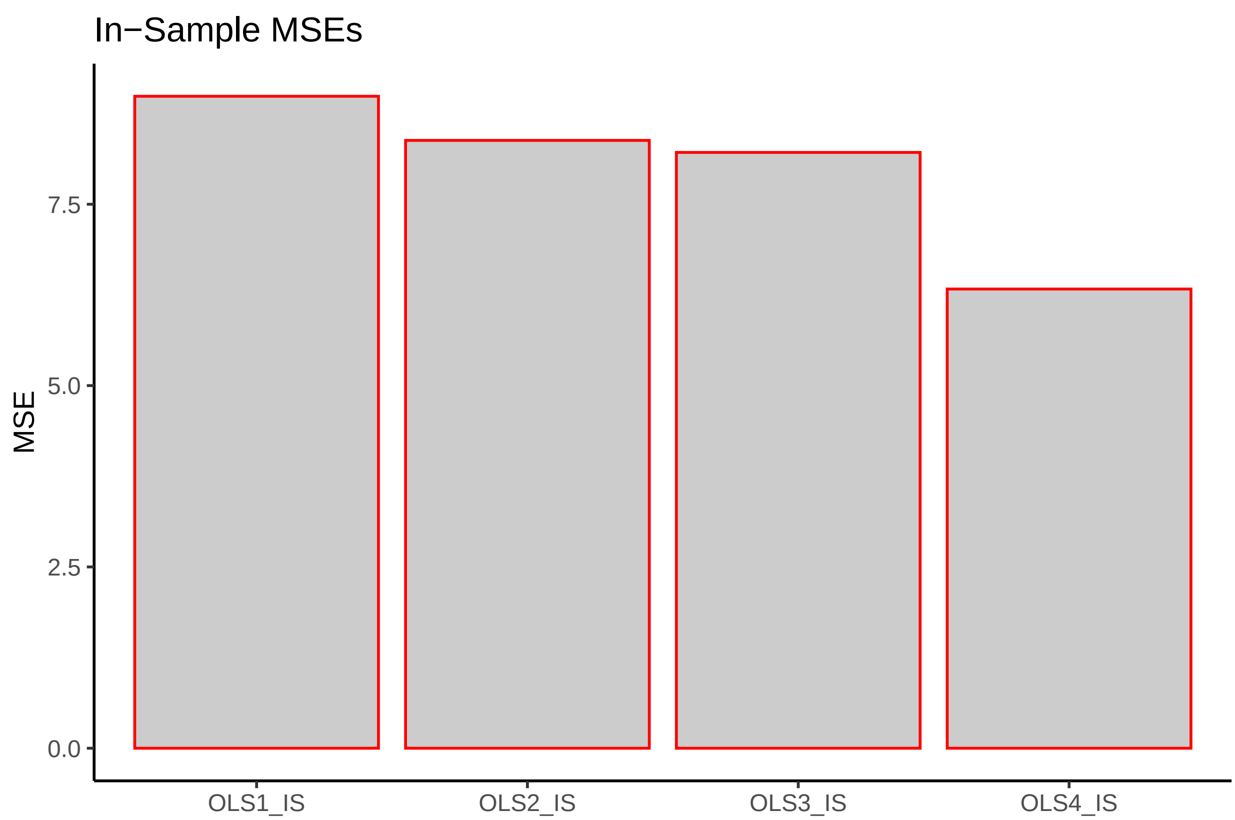
```r
MSE_IS_OLS3 <- mean((train$G3 - OLS3$fitted.values)^2)
MSE_IS_OLS4 <- mean((train$G3 - OLS4$fitted.values)^2)

fit_OLS1 <- predict(OLS1, newdata = test)
MSE_OOS_OLS1 <- mean((test$G3 - fit_OLS1)^2)

fit_OLS2 <- predict(OLS2, newdata = test)
MSE_OOS_OLS2 <- mean((test$G3 - fit_OLS2)^2)

fit_OLS3 <- predict(OLS3, newdata = test)
MSE_OOS_OLS3 <- mean((test$G3 - fit_OLS3)^2)

fit_OLS4 <- predict(OLS4, newdata = test)
MSE_OOS_OLS4 <- mean((test$G3 - fit_OLS4)^2)

(MSE_IS <- data.frame(model = c("OLS1_IS", "OLS2_IS", "OLS3_IS", "OLS4_IS"),
                      MSE = c(MSE_IS_OLS1, MSE_IS_OLS2,
                              MSE_IS_OLS3, MSE_IS_OLS4)))
```

```
##     model      MSE
## 1 OLS1_IS 8.988840
## 2 OLS2_IS 8.380403
## 3 OLS3_IS 8.214626
## 4 OLS4_IS 6.330987
```

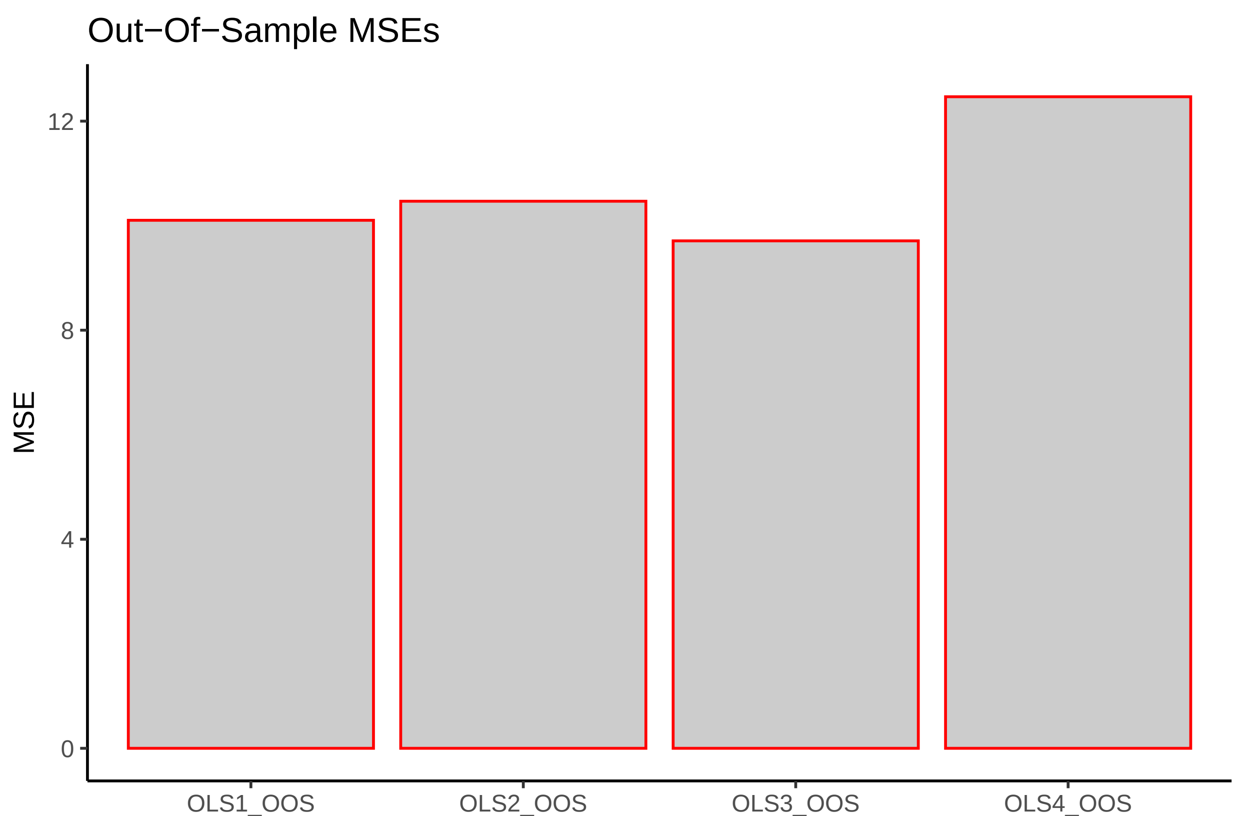
```r
(MSE_OOS <- data.frame(model = c("OLS1_OOS", "OLS2_OOS", "OLS3_OOS", "OLS4_OOS"),
                       MSE = c(MSE_OOS_OLS1, MSE_OOS_OLS2,
                               MSE_OOS_OLS3, MSE_OOS_OLS4)))
```

```
##      model       MSE
## 1 OLS1_OOS 10.103001
## 2 OLS2_OOS 10.467642
## 3 OLS3_OOS  9.709007
## 4 OLS4_OOS 12.466627
```

```r
(ggplot(MSE_IS, aes(model, MSE)) +
  geom_col(color = "red", fill = 'black', alpha = 0.2) +
  ggtitle("In-Sample MSEs") +
  xlab("") +
  theme_classic())
```




```r
(ggplot(MSE_OOS, aes(model, MSE)) +
  geom_col(color = "red", fill = 'black', alpha = 0.2) +
  ggtitle("Out-Of-Sample MSEs") +
  xlab("") +
  theme_classic())
```



In order to determine the best-performing model, we define the out-of-sample MSE as the main determinant, based on the fact that the purpose of a prediction model is to perform best out of the training sample. From the results, we can observe that the four OLS models, as described in the formulas above, have out-of-sample MSEs of 10.30, 10.47, 9.71 and 12.47, respectively. Hence, we can conclude that the third model, namely the linear regression based on OLS with ten different covariates, performs best in the out-of-sample/test data. Thus, a better prediction performance on new data is expected compared to the three other models.