Exercise 2

Advanced Methods for Regression and Classification

Martina Adamuscinova

2025-10-21

Load and split the data:

The data contains information about construction costs `df$y` of real estate single-family residential apartments in Tehran, Iran.

```
setwd('/Users/martinaadamuscinova')
load("building.RData")
```

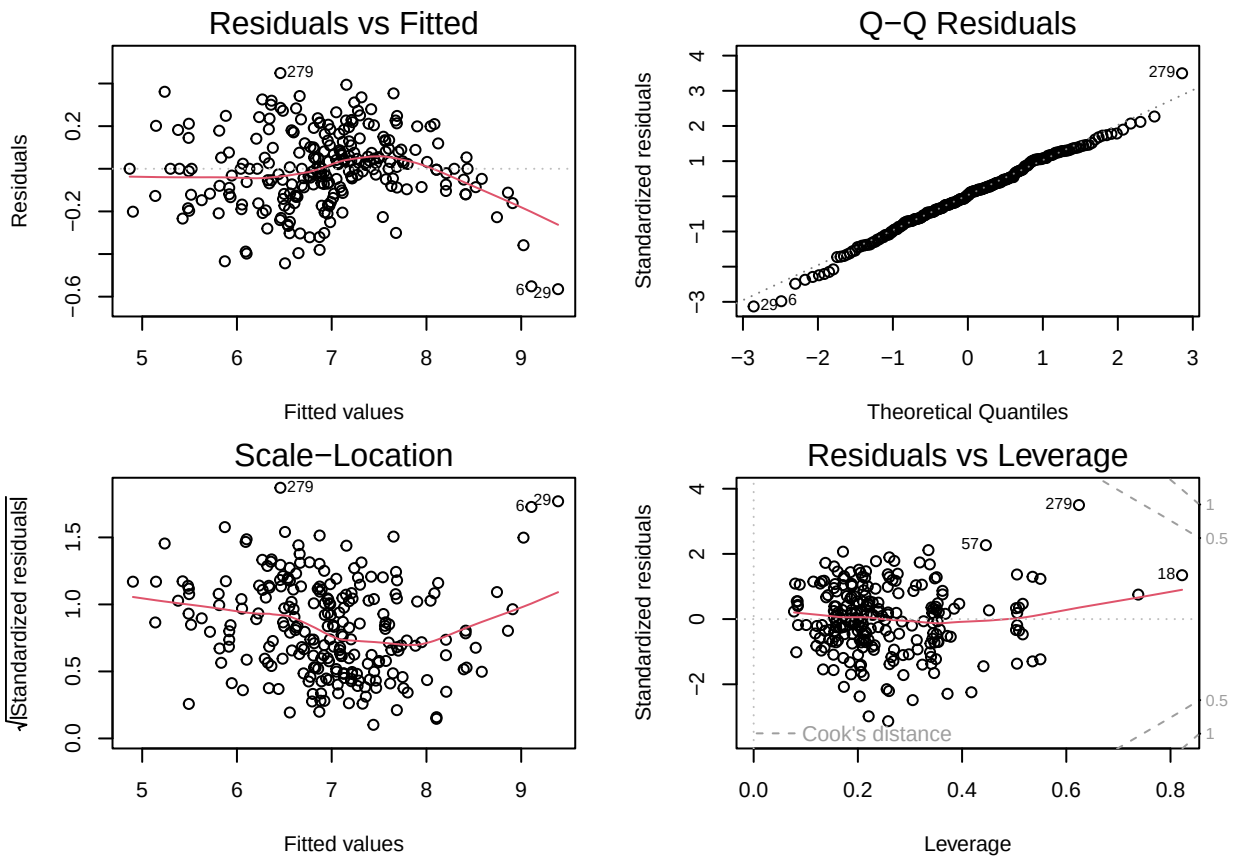
```
set.seed(24)

n <- nrow(df)
train <- sample(1:n, size = round(2/3 * n))
test <- (1:n)[-train]
```

Exercise 1

```
m1<- lm(y~., data = df, subset = train)
m1_pred_test <- predict(m1, newdata = df[test,])
```

a)

```
par(mfrow = c(2, 2), cex = 0.7, mar = c(4, 4, 2, 2))
plot(m1)
```



From the 'Residuals vs Fitted' and 'Scale-Location' plots, it seems that the homoscedasticity assumption is satisfied. Residuals are fairly evenly scattered with no clear pattern. However, slight curvature can be detected in the plots which could indicate mild non-linearity. From the 'Q-Q Residuals' plot we see that most residuals follow the normal line closely, though deviations occur at both tails, pointing to the potential presence of outliers. The 'Residuals vs Leverage' plot indicates a few observations that could influence the regression coefficients, but overall influence seems moderate.

b)

```
coef(m1)
```

```
##         (Intercept)          START.YEAR       START.QUARTER     COMPLETION.YEAR
##        3.060275e+02       -4.583304e+00       -6.491958e-01        7.747832e-02
## COMPLETION.QUARTER            PhysFin1            PhysFin2            PhysFin3
##        2.575343e-02       -2.957669e-02        3.202121e-05        7.620888e-05
##            PhysFin4            PhysFin5            PhysFin6            PhysFin7
##       -1.062381e-04       -3.112674e-03        6.660530e-04                  NA
##            PhysFin8               Econ1               Econ2               Econ3
##        4.867836e-04        6.459288e-04        3.510751e-01        4.535328e-01
##               Econ4               Econ5               Econ6               Econ7
##       -2.739181e-02        2.865240e-05       -5.126585e-04       -1.653414e-01
##               Econ8               Econ9              Econ10              Econ11
##        1.145921e-02       -3.146889e-04        9.555793e-01       -2.710021e-03
##              Econ12              Econ13              Econ14              Econ15
```

```
##       4.087347e-03      -1.540769e-04       1.200412e-03      -3.096889e-01
##             Econ16             Econ17             Econ18             Econ19
##       9.643135e-01       5.505590e-05       1.578076e-04      -7.579301e-06
##         Econ1.lag1         Econ2.lag1         Econ3.lag1         Econ4.lag1
##      -5.100883e-04      -9.855374e-01      -1.692274e-01       6.684749e-01
##         Econ5.lag1         Econ6.lag1         Econ7.lag1         Econ8.lag1
##       1.299865e-05       8.645901e-04      -5.323517e-02      -6.956040e-03
##         Econ9.lag1        Econ10.lag1        Econ11.lag1        Econ12.lag1
##       5.252813e-05      -5.369477e-01      -5.840291e-03       9.687316e-04
##        Econ13.lag1        Econ14.lag1        Econ15.lag1        Econ16.lag1
##       1.120091e-04       3.085272e-04       9.695643e-01      -1.211819e+00
##        Econ17.lag1        Econ18.lag1        Econ19.lag1         Econ1.lag2
##      -6.210047e-04       8.009613e-05      -9.024392e-06      -2.390565e-04
##         Econ2.lag2         Econ3.lag2         Econ4.lag2         Econ5.lag2
##       1.049877e+00      -5.013525e-02      -4.123327e-01      -1.770217e-05
##         Econ6.lag2         Econ7.lag2         Econ8.lag2         Econ9.lag2
##      -5.325516e-04      -8.112318e-02      -1.355449e-02      -1.145171e-04
##        Econ10.lag2        Econ11.lag2        Econ12.lag2        Econ13.lag2
##       4.462067e-01      -5.264031e-03       1.068559e-02       1.070018e-04
##        Econ14.lag2        Econ15.lag2        Econ16.lag2        Econ17.lag2
##      -6.555360e-04      -3.294641e-01       7.179874e-01      -1.993108e-04
##        Econ18.lag2        Econ19.lag2         Econ1.lag3         Econ2.lag3
##       1.555121e-04       5.096244e-06       8.471132e-04      -7.834777e-01
##         Econ3.lag3         Econ4.lag3         Econ5.lag3         Econ6.lag3
##       1.134360e-01      -2.288266e-01                 NA                 NA
##         Econ7.lag3         Econ8.lag3         Econ9.lag3        Econ10.lag3
##                 NA                 NA                 NA                 NA
##        Econ11.lag3        Econ12.lag3        Econ13.lag3        Econ14.lag3
##                 NA                 NA                 NA                 NA
##        Econ15.lag3        Econ16.lag3        Econ17.lag3        Econ18.lag3
##                 NA                 NA                 NA                 NA
##        Econ19.lag3         Econ1.lag4         Econ2.lag4         Econ3.lag4
##                 NA                 NA                 NA                 NA
##         Econ4.lag4         Econ5.lag4         Econ6.lag4         Econ7.lag4
##                 NA                 NA                 NA                 NA
##         Econ8.lag4         Econ9.lag4        Econ10.lag4        Econ11.lag4
##                 NA                 NA                 NA                 NA
##        Econ12.lag4        Econ13.lag4        Econ14.lag4        Econ15.lag4
##                 NA                 NA                 NA                 NA
##        Econ16.lag4        Econ17.lag4        Econ18.lag4        Econ19.lag4
##                 NA                 NA                 NA                 NA
```

The dataset includes of 19 economic variables in 5 time lags, where each time lag represents the number of quarters before the start of the specific project. These lagged versions of the same economic variable are highly correlated, and therefore having all lags in the model together leads to multicollinearity. This causes that the model cannot solve for unique coefficients and produced NAs.

c)

```r
m1_pred_train <- predict(m1, newdata = df[train,])
rmse_train <- sqrt(mean((df[train,"y"] - m1_pred_train)^2))
```

```r
rmse_test <- sqrt(mean((df[test,"y"] - m1_pred_test)^2))
cat('RMSE of training set:', rmse_train, '\n')
```

```
## RMSE of training set: 0.1759113
```

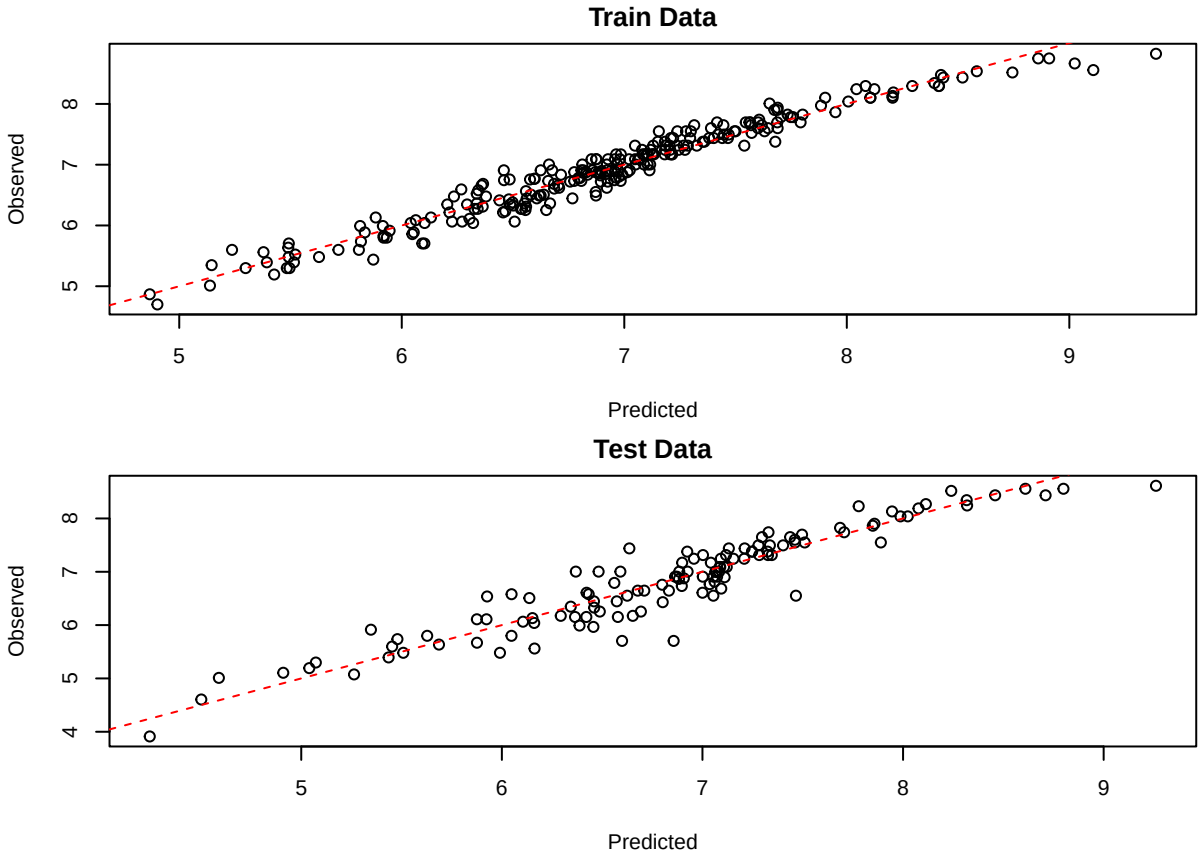
```r
cat('RMSE of test set:', rmse_test)
```

```
## RMSE of test set: 0.3063756
```

The full model fits the training data closely but performs worse on the test data, which is to be expected. However, the relative increase of $\frac{0.306}{0.176} \approx 1.74$ may be suggesting limited generalization and potential overfitting due to the large number of predictors and multicollinearity.

d)

```r
par(mfrow = c(2, 1), cex = 0.7, mar = c(4, 4, 2, 2))
plot(m1_pred_train, df[train,'y'], ylab = 'Observed', xlab = 'Predicted',
     main = 'Train Data')
abline(0,1,lty=2, col = 'red')
plot(m1_pred_test, df[test,'y'], ylab = 'Observed', xlab = 'Predicted',
     main = 'Test Data')
abline(0,1,lty=2, col = 'red')
```



Model fits training data relatively well, points lie close to the 45° line. The model, however, shows moderate generalization loss as the test data appears more scattered around the line, which is consistent with earlier RMSE gap. In both plots we can see that at extremes, points shrink toward the line center, indicating slight underprediction of very high values ($Observed > Predicted$) and overprediction of very low values ($Observed < Predicted$).

Exercise 2

Forward Selection

```
m0<- lm(y~1, data = df, subset = train)
m_fs<- step(m0, scope = list(lower = formula(m0), upper = formula(m1)),
            direction = "forward", trace = 0)
summary(m_fs)
```

```
##
## Call:
## lm(formula = y ~ PhysFin8 + Econ14.lag4 + PhysFin1 + COMPLETION.YEAR +
##     Econ7.lag3 + PhysFin3 + PhysFin4 + PhysFin6 + PhysFin5 +
##     Econ12.lag4 + COMPLETION.QUARTER, data = df, subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60256 -0.11499  0.00195  0.14862  0.56762
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -3.944e+00  1.119e+00  -3.526 0.000507 ***
## PhysFin8            4.798e-04  3.095e-05  15.502  < 2e-16 ***
## Econ14.lag4         6.405e-05  1.779e-05   3.599 0.000389 ***
## PhysFin1           -3.291e-02  3.525e-03  -9.336  < 2e-16 ***
## COMPLETION.YEAR     1.225e-01  1.530e-02   8.002 5.51e-14 ***
## Econ7.lag3         -4.444e-03  1.192e-03  -3.728 0.000242 ***
## PhysFin3            1.438e-04  5.761e-05   2.496 0.013258 *
## PhysFin4           -5.650e-05  4.553e-05  -1.241 0.215812
## PhysFin6            5.292e-04  8.879e-05   5.960 9.12e-09 ***
## PhysFin5           -3.033e-03  5.337e-04  -5.682 3.90e-08 ***
## Econ12.lag4         3.788e-04  1.145e-04   3.308 0.001087 **
## COMPLETION.QUARTER  3.598e-02  1.305e-02   2.757 0.006288 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2216 on 236 degrees of freedom
## Multiple R-squared:  0.9286, Adjusted R-squared:  0.9252
## F-statistic: 278.9 on 11 and 236 DF,  p-value: < 2.2e-16
```

Backward Selection

```
m_bs<- step(m1, direction = "backward", trace = 0)
summary(m_bs)
```

```
##
## Call:
## lm(formula = y ~ START.YEAR + COMPLETION.YEAR + COMPLETION.QUARTER +
##     PhysFin1 + PhysFin2 + PhysFin4 + PhysFin5 + PhysFin6 + PhysFin8 +
##     Econ1 + Econ2 + Econ3 + Econ5 + Econ6 + Econ7 + Econ8 + Econ9 +
##     Econ10 + Econ11 + Econ12 + Econ13 + Econ14 + Econ15 + Econ16 +
##     Econ18 + Econ19 + Econ1.lag1 + Econ2.lag1 + Econ3.lag1 +
##     Econ4.lag1 + Econ5.lag1 + Econ6.lag1 + Econ7.lag1 + Econ8.lag1 +
##     Econ10.lag1 + Econ11.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 + Econ14.lag2 +
##     Econ15.lag2 + Econ16.lag2 + Econ17.lag2 + Econ18.lag2 + Econ19.lag2 +
##     Econ1.lag3 + Econ2.lag3 + Econ3.lag3 + Econ4.lag3, data = df,
##     subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56748 -0.10845  0.00134  0.10975  0.48532
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.441e+02  8.088e+01   4.254 3.34e-05 ***
## START.YEAR         -5.205e+00  1.230e+00  -4.233 3.64e-05 ***
## COMPLETION.YEAR     7.513e-02  3.053e-02   2.461 0.014764 *
## COMPLETION.QUARTER  2.626e-02  1.541e-02   1.704 0.090061 .
## PhysFin1           -2.938e-02  3.690e-03  -7.962 1.70e-13 ***
## PhysFin2            5.576e-05  1.848e-05   3.017 0.002914 **
## PhysFin4           -1.270e-04  6.016e-05  -2.111 0.036149 *
## PhysFin5           -3.016e-03  6.357e-04  -4.744 4.19e-06 ***
## PhysFin6            6.479e-04  1.160e-04   5.585 8.30e-08 ***
## PhysFin8            4.856e-04  3.137e-05  15.480  < 2e-16 ***
## Econ1               6.132e-04  1.476e-04   4.155 4.98e-05 ***
## Econ2               5.610e-01  1.255e-01   4.469 1.37e-05 ***
## Econ3               4.369e-01  9.736e-02   4.488 1.26e-05 ***
## Econ5               3.399e-05  1.152e-05   2.951 0.003581 **
## Econ6              -4.457e-04  2.467e-04  -1.807 0.072392 .
## Econ7              -1.519e-01  3.778e-02  -4.021 8.44e-05 ***
## Econ8               1.039e-02  2.917e-03   3.563 0.000468 ***
## Econ9              -3.002e-04  6.968e-05  -4.308 2.68e-05 ***
## Econ10              1.005e+00  2.602e-01   3.862 0.000156 ***
## Econ11             -1.674e-03  4.766e-04  -3.512 0.000560 ***
## Econ12              3.505e-03  1.141e-03   3.074 0.002437 **
## Econ13             -7.891e-05  4.799e-05  -1.644 0.101821
## Econ14              1.047e-03  2.595e-04   4.033 8.06e-05 ***
## Econ15             -3.051e-01  9.325e-02  -3.272 0.001276 **
## Econ16              7.753e-01  1.961e-01   3.953 0.000110 ***
## Econ18              2.445e-04  5.446e-05   4.489 1.26e-05 ***
## Econ19             -4.134e-06  2.468e-06  -1.675 0.095646 .
## Econ1.lag1         -3.709e-04  1.345e-04  -2.757 0.006424 **
## Econ2.lag1         -1.043e+00  2.434e-01  -4.287 2.92e-05 ***
## Econ3.lag1         -1.384e-01  4.282e-02  -3.233 0.001452 **
## Econ4.lag1          4.999e-01  1.863e-01   2.684 0.007945 **
## Econ5.lag1          1.306e-05  8.387e-06   1.558 0.121035
```

```
## Econ6.lag1          9.744e-04  2.066e-04   4.717 4.72e-06 ***
## Econ7.lag1         -8.318e-02  1.951e-02  -4.263 3.22e-05 ***
## Econ8.lag1         -4.198e-03  1.394e-03  -3.010 0.002975 **
## Econ10.lag1        -4.502e-01  1.608e-01  -2.799 0.005672 **
## Econ11.lag1        -4.871e-03  1.138e-03  -4.279 3.02e-05 ***
## Econ15.lag1         1.027e+00  2.366e-01   4.340 2.35e-05 ***
## Econ16.lag1        -1.090e+00  2.464e-01  -4.424 1.65e-05 ***
## Econ17.lag1        -5.813e-04  1.917e-04  -3.033 0.002772 **
## Econ18.lag1         1.701e-04  3.887e-05   4.376 2.02e-05 ***
## Econ19.lag1        -9.994e-06  2.355e-06  -4.243 3.49e-05 ***
## Econ1.lag2         -2.018e-04  1.177e-04  -1.715 0.087981 .
## Econ2.lag2          8.928e-01  1.988e-01   4.492 1.24e-05 ***
## Econ3.lag2         -7.571e-02  2.741e-02  -2.763 0.006318 **
## Econ4.lag2         -2.422e-01  1.361e-01  -1.779 0.076847 .
## Econ5.lag2         -3.441e-05  1.667e-05  -2.064 0.040437 *
## Econ6.lag2         -4.755e-04  2.212e-04  -2.150 0.032889 *
## Econ7.lag2         -9.823e-02  2.998e-02  -3.277 0.001256 **
## Econ8.lag2         -1.017e-02  3.573e-03  -2.848 0.004905 **
## Econ9.lag2         -1.561e-04  5.048e-05  -3.093 0.002292 **
## Econ10.lag2         2.591e-01  1.369e-01   1.893 0.059920 .
## Econ11.lag2        -4.081e-03  1.011e-03  -4.035 7.99e-05 ***
## Econ12.lag2         1.061e-02  2.572e-03   4.123 5.66e-05 ***
## Econ14.lag2        -5.948e-04  1.656e-04  -3.593 0.000420 ***
## Econ15.lag2        -2.559e-01  1.102e-01  -2.321 0.021385 *
## Econ16.lag2         8.003e-01  2.489e-01   3.215 0.001539 **
## Econ17.lag2        -3.046e-04  9.671e-05  -3.150 0.001907 **
## Econ18.lag2         2.225e-04  5.154e-05   4.318 2.57e-05 ***
## Econ19.lag2         4.430e-06  1.806e-06   2.453 0.015106 *
## Econ1.lag3          7.778e-04  1.319e-04   5.895 1.76e-08 ***
## Econ2.lag3         -7.609e-01  1.787e-01  -4.257 3.30e-05 ***
## Econ3.lag3          1.231e-01  3.434e-02   3.586 0.000431 ***
## Econ4.lag3         -3.490e-01  1.037e-01  -3.364 0.000933 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2054 on 184 degrees of freedom
## Multiple R-squared:  0.9522, Adjusted R-squared:  0.9358
## F-statistic: 58.14 on 63 and 184 DF,  p-value: < 2.2e-16
```

RSME Comparison

```
pred_test_fs  <- predict(m_fs, newdata = df[test,])
pred_test_bs  <- predict(m_bs, newdata = df[test,])

rmse_test_fs  <- sqrt(mean((df[test,"y"]  - pred_test_fs)^2))
rmse_test_bs  <- sqrt(mean((df[test,"y"]  - pred_test_bs)^2))

cat("RMSE test (forward):", rmse_test_fs,  "\n")
```

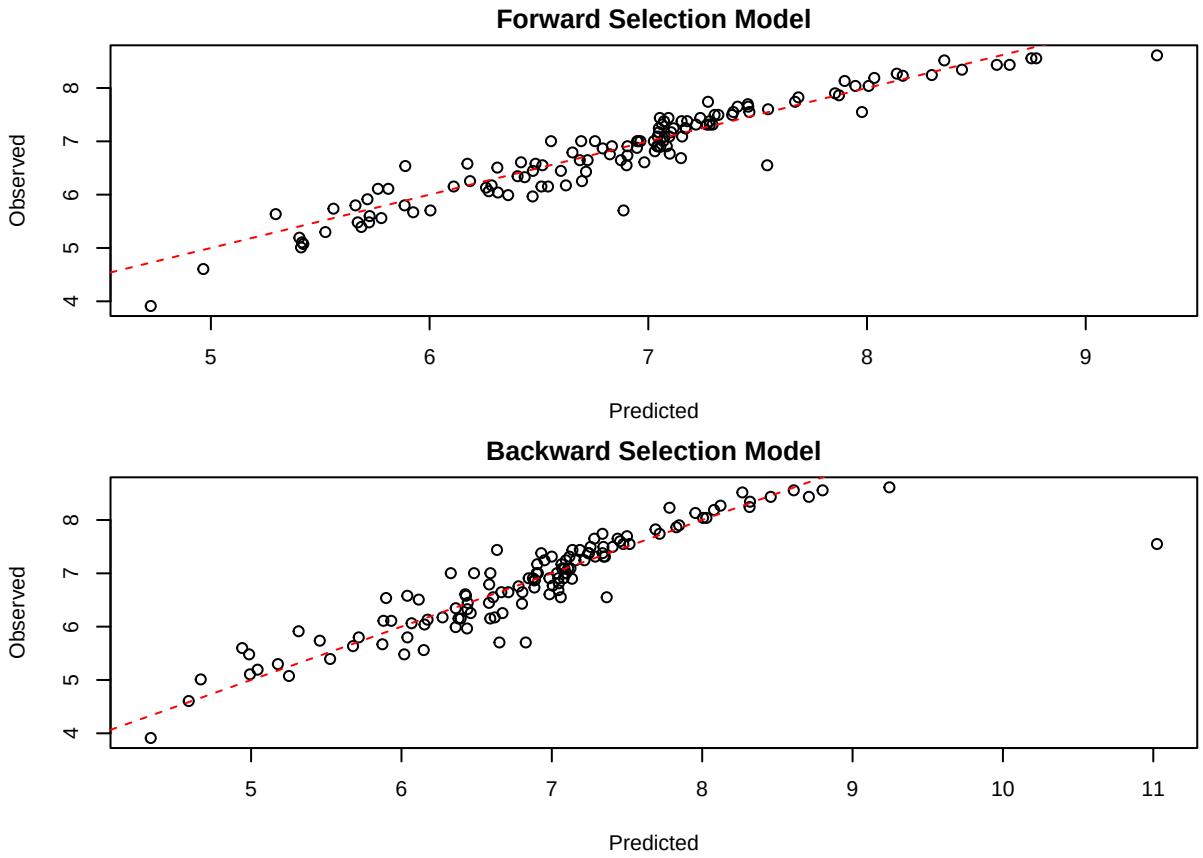
```
## RMSE test (forward): 0.2823906
```

```r
cat("RMSE test (backward):", rmse_test_bs)
```

```
## RMSE test (backward): 0.439674
```

Plots Comparison

```r
par(mfrow = c(2, 1), cex = 0.7, mar = c(4, 4, 2, 2))
plot(pred_test_fs, df[test,'y'], ylab = 'Observed', xlab = 'Predicted',
     main = 'Forward Selection Model')
abline(0,1,lty=2, col = 'red')
plot(pred_test_bs, df[test,'y'], ylab = 'Observed', xlab = 'Predicted',
     main = 'Backward Selection Model')
abline(0,1,lty=2, col = 'red')
```



The forward selection model has a lower test RMSE (0.28) compared to the backward selection model (0.44), meaning it predicts unseen data more accurately. Compared to the test RMSE of the full model (0.306), the forward selection model shows only slight improvement, while the backward selection model preforms worse. In the plots, the forward model's points are closer to the red reference line, while the backward model shows more scatter and a few larger deviations at high predicted values. The difference suggests that backward selection may have retained too many correlated variables, causing higher variance and poorer generalization.

Exercise 3

The forward selection model would be preferred due to its lower test error and simpler structure. ANOVA makes sense only if the models are nested, and as the backward and forward selection models are not nested, we can only compare them individually with the full model. In this way, we can test whether the simpler model explains the data just as well as the full model.

```
#Forward Selection Model vs Full Model Variance Table:
print(anova(m_fs, m1)[, c("Res.Df", "RSS", "Df", "Sum of Sq", "F", "Pr(>F)")])
```

```
##   Res.Df     RSS Df Sum of Sq     F  Pr(>F)
## 1    236 11.5905
## 2    175  7.6743 61    3.9162 1.464 0.02906 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Backward Selection Model vs Full Model Variance Table:
print(anova(m_bs, m1)[, c("Res.Df", "RSS", "Df", "Sum of Sq", "F", "Pr(>F)")])
```

```
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    184 7.7613
## 2    175 7.6743  9  0.086951 0.2203 0.9913
```

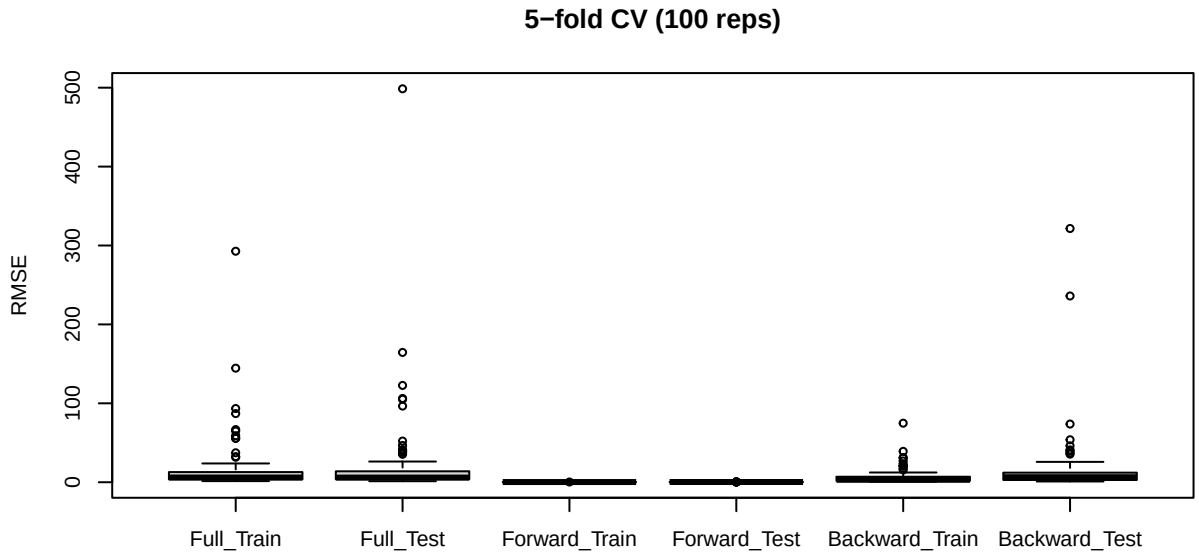
The full model explains more variation than the forward model. However, the improvement seems small relative to the complexity added (61 extra variables). The RSS improves for the full model by $11.5905 - 7.6743 \approx 3.92$ that is $\frac{3.9162}{61} = 0.0642$ RSS improvement per added variable. For this, it seems reasonable to continue with the forward selection model. The full model does not improve fit over the backward model. Both perform almost identically in terms of explained variance.

Exercise 4

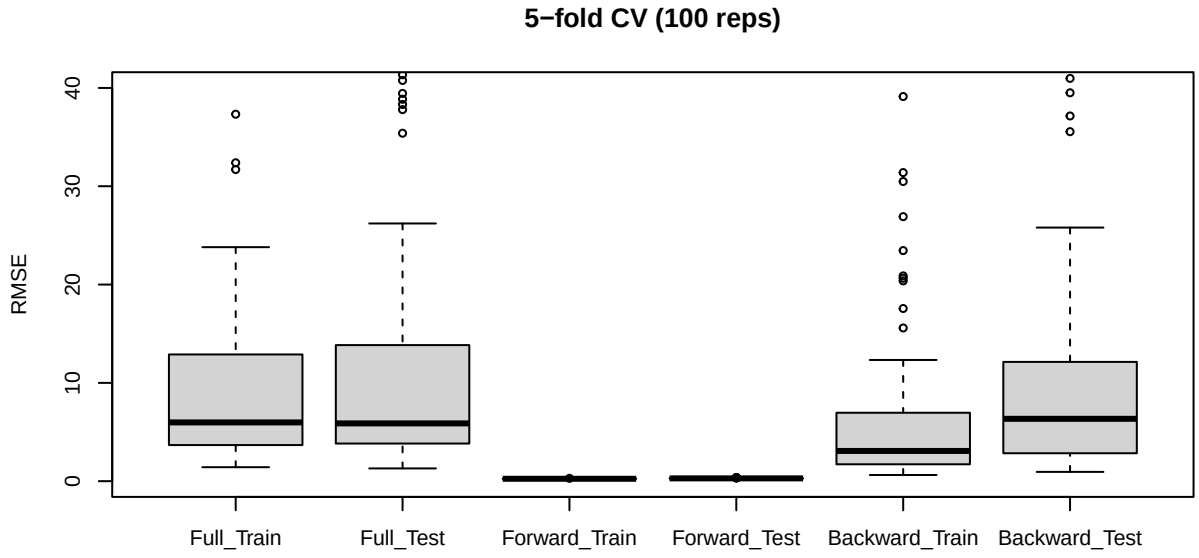
```
library(cvTools)

cv_m1_train <- cvFit(m1, y = df[train, "y"], data = df[train,],
                     cost = rmspe, K = 5, R = 100)
cv_m1_test  <- cvFit(m1, y = df[test,  "y"], data = df[test, ],
                     cost = rmspe, K = 5, R = 100)
cv_fs_train <- cvFit(m_fs, y = df[train, "y"], data = df[train,],
                     cost = rmspe, K = 5, R = 100)
cv_fs_test  <- cvFit(m_fs, y = df[test,  "y"], data = df[test, ],
                     cost = rmspe, K = 5, R = 100)
cv_bs_train <- cvFit(m_bs, y = df[train, "y"], data = df[train,],
                     cost = rmspe, K = 5, R = 100)
cv_bs_test  <- cvFit(m_bs, y = df[test,  "y"], data = df[test, ],
                     cost = rmspe, K = 5, R = 100)
```

```
par(cex = 0.7)

boxplot(list(
  Full_Train=cv_m1_train$reps, Full_Test=cv_m1_test$reps,
```

```
  Forward_Train=cv_fs_train$reps, Forward_Test=cv_fs_test$reps,
  Backward_Train=cv_bs_train$reps, Backward_Test=cv_bs_test$reps
), ylab="RMSE", main="5-fold CV (100 reps)")
```



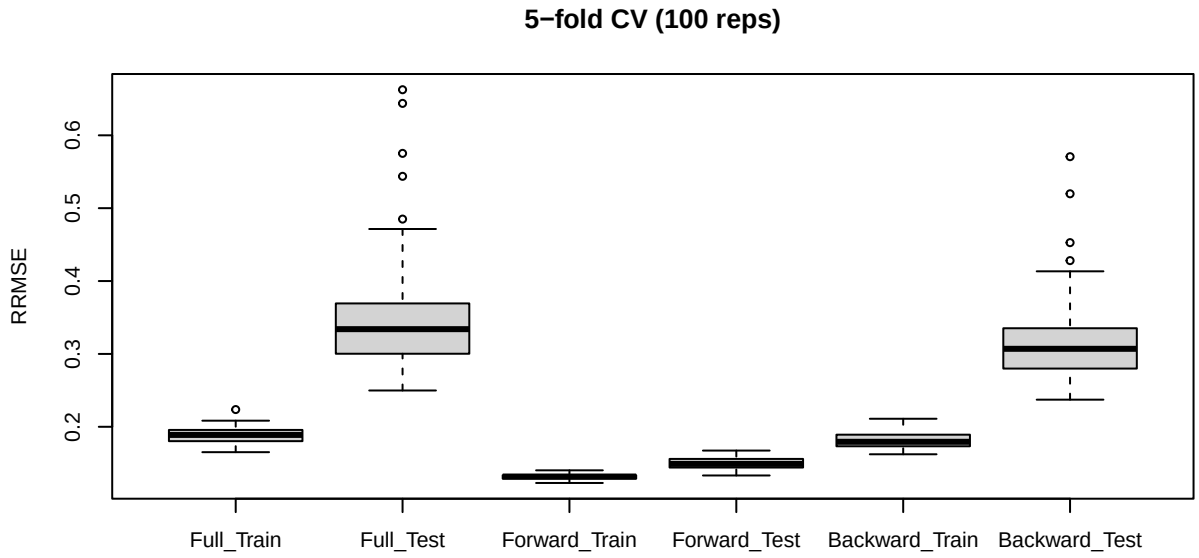

```
boxplot(list(
  Full_Train=cv_m1_train$reps, Full_Test=cv_m1_test$reps,
  Forward_Train=cv_fs_train$reps, Forward_Test=cv_fs_test$reps,
  Backward_Train=cv_bs_train$reps, Backward_Test=cv_bs_test$reps
), ylab="RMSE", main="5-fold CV (100 reps)", ylim = c(0, 40))
```



From the plots it is clear that the forward selection model shows the lowest and most stable RMSE for both training and test sets. Its boxes are compact and centered around the median values, while the full and backward models show extreme variance and instability with higher median RSMEs.

Exercise 5

```
par(cex = 0.7)

cv_m1_train <- cvFit(m1, y = df[train, "y"], data = df[train,],
                     cost = rtmspe, K = 5, R = 100)
cv_m1_test  <- cvFit(m1, y = df[test,  "y"], data = df[test, ],
                     cost = rtmspe, K = 5, R = 100)
cv_fs_train <- cvFit(m_fs, y = df[train, "y"], data = df[train,],
                     cost = rtmspe, K = 5, R = 100)
cv_fs_test  <- cvFit(m_fs, y = df[test,  "y"], data = df[test, ],
                     cost = rtmspe, K = 5, R = 100)
cv_bs_train <- cvFit(m_bs, y = df[train, "y"], data = df[train,],
                     cost = rtmspe, K = 5, R = 100)
cv_bs_test  <- cvFit(m_bs, y = df[test,  "y"], data = df[test, ],
                     cost = rtmspe, K = 5, R = 100)

boxplot(list(
  Full_Train=cv_m1_train$reps, Full_Test=cv_m1_test$reps,
  Forward_Train=cv_fs_train$reps, Forward_Test=cv_fs_test$reps,
  Backward_Train=cv_bs_train$reps, Backward_Test=cv_bs_test$reps
), ylab="RRMSE", main="5-fold CV (100 reps)")
```



Here instead of measuring root mean squared prediction error (RMSE) in absolute units, we measure relative root mean squared prediction error (RRMSE) where each error is scaled by the observed value. That means we observe percentage deviation, which makes comparisons less sensitive to the response's scale. Using

relative RRMSE confirms the same result as before, the forward selection model remains the preferred model.