Statistical Simulation and Computerintensive Methods

Exercise 6

Markus Kiesel | 1228952

27.11.2021

Contents

Task 1

```
# My Seed
SEED <- 1228952
set.seed(SEED)
```

We will work with the dataset Auto in the ISLR package. Obtain information on the data set, its structure and the real world meaning of its variables from the help page.

```
# load data
data(Auto, package = "ISLR")
str(Auto)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 1
```

1.1)

Fit the following models

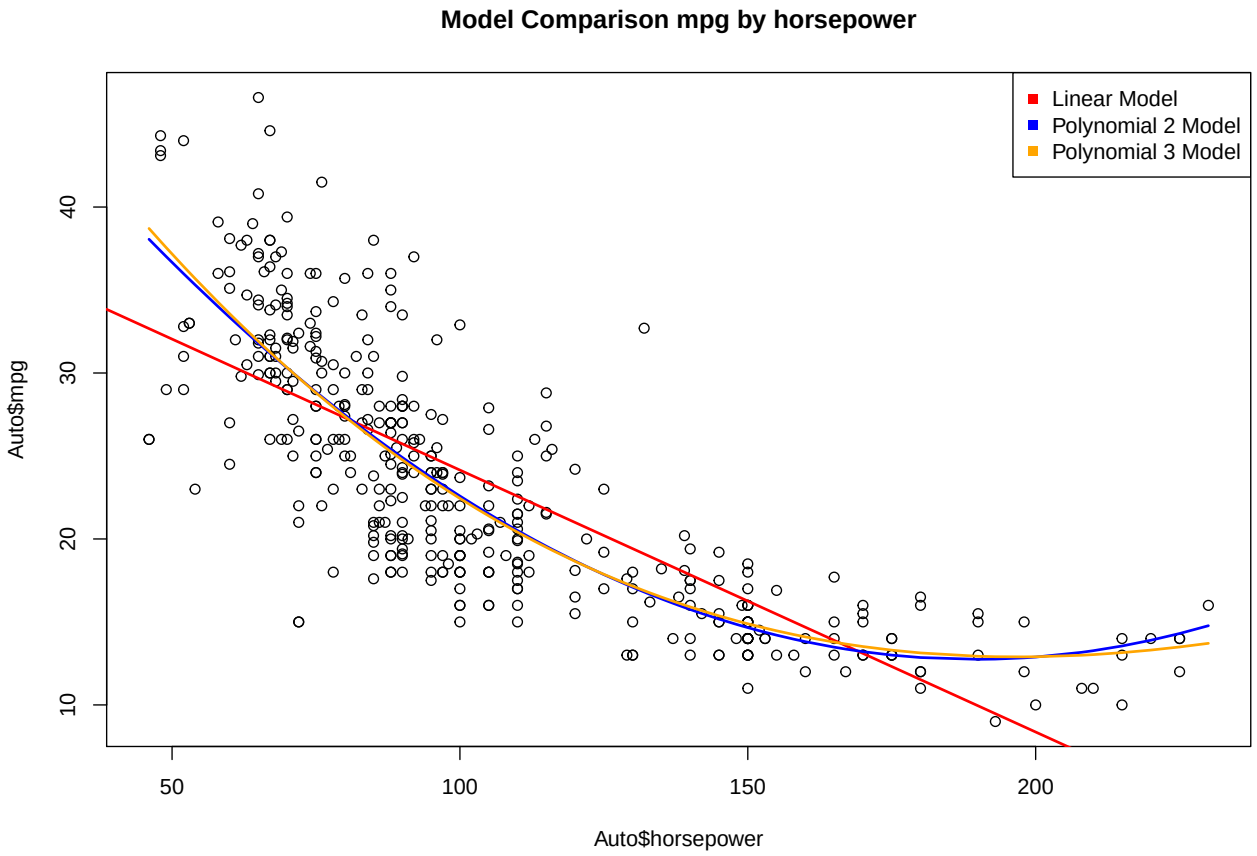
mpg ~ horsepower

mpg ~ poly(horsepower,2)

mpg ~ poly(horsepower,3)

Visualize all 3 models in comparison added to a scatterplot of the input data.

```
# define models
lm_formula <- formula(mpg ~ horsepower)
lm <- glm(lm_formula, data = Auto)
poly2_formula <- formula(mpg ~ poly(horsepower, 2))
poly2 <- glm(poly2_formula, data = Auto)
poly3_formula <- formula(mpg ~ poly(horsepower, 3))
poly3 <- glm(poly3_formula, data = Auto)

# plot models + data
plot(Auto$mpg ~ Auto$horsepower, main = "Model Comparison mpg by horsepower")
colors <- c("red", "blue", "orange")
abline(lm$coeff[1], lm$coeff[2], col = colors[1], lw = 2)
lines(sort(Auto$horsepower), fitted(poly2)[order(Auto$horsepower)], col = colors[2],
    type = "l", lw = 2)
lines(sort(Auto$horsepower), fitted(poly3)[order(Auto$horsepower)], col = colors[3],
    type = "l", lw = 2)
legend("topright", legend = c("Linear Model", "Polynomial 2 Model", "Polynomial 3 Model"),
    pch = c(15, 15), col = colors)
```



1.2)

Use the validation set approach to compare the models. Use once a train/test split of 50%/50% and once 70%/30%. Choose the best model based on Root Mean Squared Error, Mean Squared Error and Median Absolute Deviation.

```
# function to evaluate a model on train tests split using RMSE, MSE and MAD
eval_tts <- function(model_formula, train_index, test_index) {
    results <- rep(0, 3)
    # evaluate on test data
    model <- glm(model_formula, data = Auto, subset = train_index)
    y <- Auto[test_index, "mpg"]
    X <- Auto[test_index, ]
    y_hat <- predict(model, X)

    results[1] <- rmse(y, y_hat)
    results[2] <- mse(y, y_hat)
    results[3] <- mad(y, y_hat)

    return(results)
}
```

```
results_tts <- matrix(0, 6, 3)
rownames(results_tts) <- c("Linear TTS 50/50", "Poly 2 TTS 50/50", "Poly 3 TTS 50/50",
    "Linear TTS 70/30", "Poly 2 TTS 70/30", "Poly 3 TTS 70/30")

# Train Test Split 50/50
```

```r
set.seed(SEED)
n <- nrow(Auto)
train_index <- sample(1:n, round(n * 0.5))
test_index <- c(1:n)[-train_index]
# linear model
results_tts[1, ] <- eval_tts(lm_formula, train_index, test_index)
# poly model second order
results_tts[2, ] <- eval_tts(poly2_formula, train_index, test_index)
# poly model third order
results_tts[3, ] <- eval_tts(poly3_formula, train_index, test_index)

# Train Test Split 70/30
set.seed(SEED)
n <- nrow(Auto)
train_index <- sample(1:n, round(n * 0.7))
test_index <- c(1:n)[-train_index]
# linear model
results_tts[4, ] <- eval_tts(lm_formula, train_index, test_index)
# poly model second order
results_tts[5, ] <- eval_tts(poly2_formula, train_index, test_index)
# poly model third order
results_tts[6, ] <- eval_tts(poly3_formula, train_index, test_index)
```

```r
table <- results_tts
colnames(table) <- c("RMSE", "MSE", "MAD")

kable(table, caption = "Comparison of Model Performance Train Test Split")
```

Table 1: Comparison of Model Performance Train Test Split

	RMSE	MSE	MAD
Linear TTS 50/50	4.944726	24.45031	4.745990
Poly 2 TTS 50/50	4.368509	19.08387	3.724000
Poly 3 TTS 50/50	4.365312	19.05595	3.746513
Linear TTS 70/30	4.960305	24.60462	4.305612
Poly 2 TTS 70/30	4.324618	18.70232	3.316794
Poly 3 TTS 70/30	4.316400	18.63131	3.337487

Based on the train-test split validation the linear model performs worst for all three measures. Both polynomial models perform better over all three measures whereas the polynomial model of third degree performs slightly better on RMSE and MSE but not on MAD. We further notice that the models trained on 70% of the data performs better than the model only trained on 50% of the data which we would expect.

1.3)

Use the cv.glm function in the boot package for the following steps.

```r
# function to evaluate models using CV (use adjusted cross-validation estimate)
eval_cv <- function(model, data, K) {
    results <- rep(0, 3)

    # extract CV results
    results[1] <- cv.glm(data, model, K = K, cost = rmse)$delta[2]
```

```
    results[2] <- cv.glm(data, model, K = K, cost = mse)$delta[2]
    results[3] <- cv.glm(data, model, K = K, cost = mad)$delta[2]

    return(results)
}
```

1.3.1)

Use cv.glm for Leave-one-out Cross Validation to compare the models above.

```
# results for Leave-one-out Cross Validation
results_loocv <- matrix(0, 3, 3)
rownames(results_loocv) <- c("Linear LOOCV", "Poly 2 LOOCV", "Poly 3 LOOCV")

# Leave One Out Cross Validation (k is number of observations)
K <- nrow(Auto)
# linear model
results_loocv[1, ] <- eval_cv(lm, Auto, K)
# poly model second order
results_loocv[2, ] <- eval_cv(poly2, Auto, K)
# poly model third order
results_loocv[3, ] <- eval_cv(poly3, Auto, K)
```

1.3.2)

Use cv.glm for 5-fold and 10-fold Cross Validation to compare the models above.

```
# results for 5-fold and 10-fold Cross Validation´
results_fcv <- matrix(0, 6, 3)
rownames(results_fcv) <- c("Linear CV K=5", "Poly 2 CV K=5", "Poly 3 CV K=5", "Linear CV K=10",
    "Poly 2 K=10", "Poly 3 K=10")
# 5-Fold Cross Validation
K <- 5
# linear model
results_fcv[1, ] <- eval_cv(lm, Auto, K)
# poly model second order
results_fcv[2, ] <- eval_cv(poly2, Auto, K)
# poly model third order
results_fcv[3, ] <- eval_cv(poly3, Auto, K)

# 10-Fold Cross Validation
K <- 10
# linear model
results_fcv[4, ] <- eval_cv(lm, Auto, K)
# poly model second order
results_fcv[5, ] <- eval_cv(poly2, Auto, K)
# poly model third order
results_fcv[6, ] <- eval_cv(poly3, Auto, K)
```

1.4)

Compare all results from 2 and 3. in a table and draw your conclusions.

```
table <- rbind(results_tts, results_loocv, results_fcv)
colnames(table) <- c("RMSE", "MSE", "MAD")
```

```
kable(table, caption = "Comparison of Model Performance TTS and CV")
```

Table 2: Comparison of Model Performance TTS and CV

	RMSE	MSE	MAD
Linear TTS 50/50	4.944726	24.45031	4.745990
Poly 2 TTS 50/50	4.368509	19.08387	3.724000
Poly 3 TTS 50/50	4.365312	19.05595	3.746513
Linear TTS 70/30	4.960305	24.60462	4.305612
Poly 2 TTS 70/30	4.324618	18.70232	3.316794
Poly 3 TTS 70/30	4.316400	18.63131	3.337487
Linear LOOCV	3.848711	24.23114	5.703339
Poly 2 LOOCV	3.272002	19.24788	4.851604
Poly 3 LOOCV	3.276749	19.33448	4.857694
Linear CV K=5	4.916584	24.11982	4.558327
Poly 2 CV K=5	4.394567	19.15379	3.603384
Poly 3 CV K=5	4.387401	19.31408	3.694432
Linear CV K=10	4.900777	24.26306	4.741957
Poly 2 K=10	4.289369	19.26153	3.693192
Poly 3 K=10	4.341485	19.27639	3.559753

When we compare the measures of all evaluation methods and all models we notice that using Cross Validation leads to better performing models in general which we would expect because the model can be trained on more data. In general a larger K leads to slightly better performing models. Here it is important to know that LOOCV can be a too optimistic estimation of the error.

The exception is the MAD for the linear LOOCV.

The same arguments can be made about the difference in performance of the models as was done for the train-test split part.

Task 2

Load the data set 'economics' from the package 'ggplot2'.

```
# load data
data(economics, package = "ggplot2")
str(economics)
```

```
## spec_tbl_df [574 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ date    : Date[1:574], format: "1967-07-01" "1967-08-01" ...
##  $ pce     : num [1:574] 507 510 516 512 517 ...
##  $ pop     : num [1:574] 198712 198911 199113 199311 199498 ...
##  $ psavert : num [1:574] 12.6 12.6 11.9 12.9 12.8 11.8 11.7 12.3 11.7 12.3 ...
##  $ uempmed : num [1:574] 4.5 4.7 4.6 4.9 4.7 4.8 5.1 4.5 4.1 4.6 ...
##  $ unemploy: num [1:574] 2944 2945 2958 3143 3066 ...
```

2.1)

Fit the following models to explain the number of unemployed persons 'unemploy' by the median number of days unemployed 'uempmed' and vice versa:

- linear model

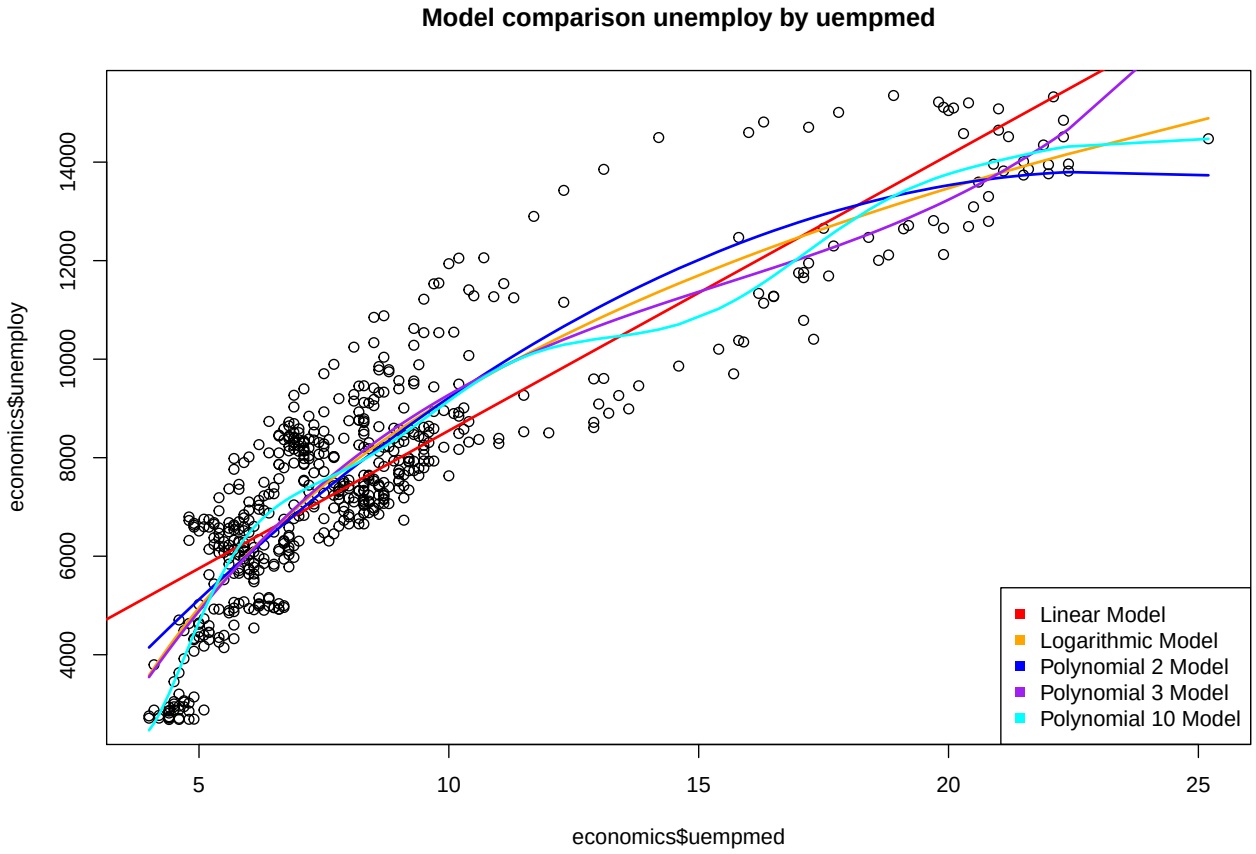
- an appropriate exponential or logarithmic model (which one is appropriate depends on which is the dependent or independent variable)
- polynomial model of 2nd, 3rd and 10th degree

```
# number of unemployed persons by the median number of days unemployed

# linear model
lm_unemploy <- glm(unemploy ~ uempmed, data = economics)
# logarithmic model
log_unemploy <- glm(unemploy ~ log(uempmed), data = economics)
# polynomial models
poly_unemploy_2 <- glm(unemploy ~ poly(uempmed, 2), data = economics)
poly_unemploy_3 <- glm(unemploy ~ poly(uempmed, 3), data = economics)
poly_unemploy_10 <- glm(unemploy ~ poly(uempmed, 10), data = economics)

# median number of days unemployed by number of unemployed persons

# linear model
lm_uempmed <- glm(uempmed ~ unemploy, data = economics)
# exponential model
expm_uempmed <- glm(uempmed ~ exp(scale(unemploy)), data = economics)
# polynomial models
poly_uempmed_2 <- glm(uempmed ~ poly(unemploy, 2), data = economics)
poly_uempmed_3 <- glm(uempmed ~ poly(unemploy, 3), data = economics)
poly_uempmed_10 <- glm(uempmed ~ poly(unemploy, 10), data = economics)
```

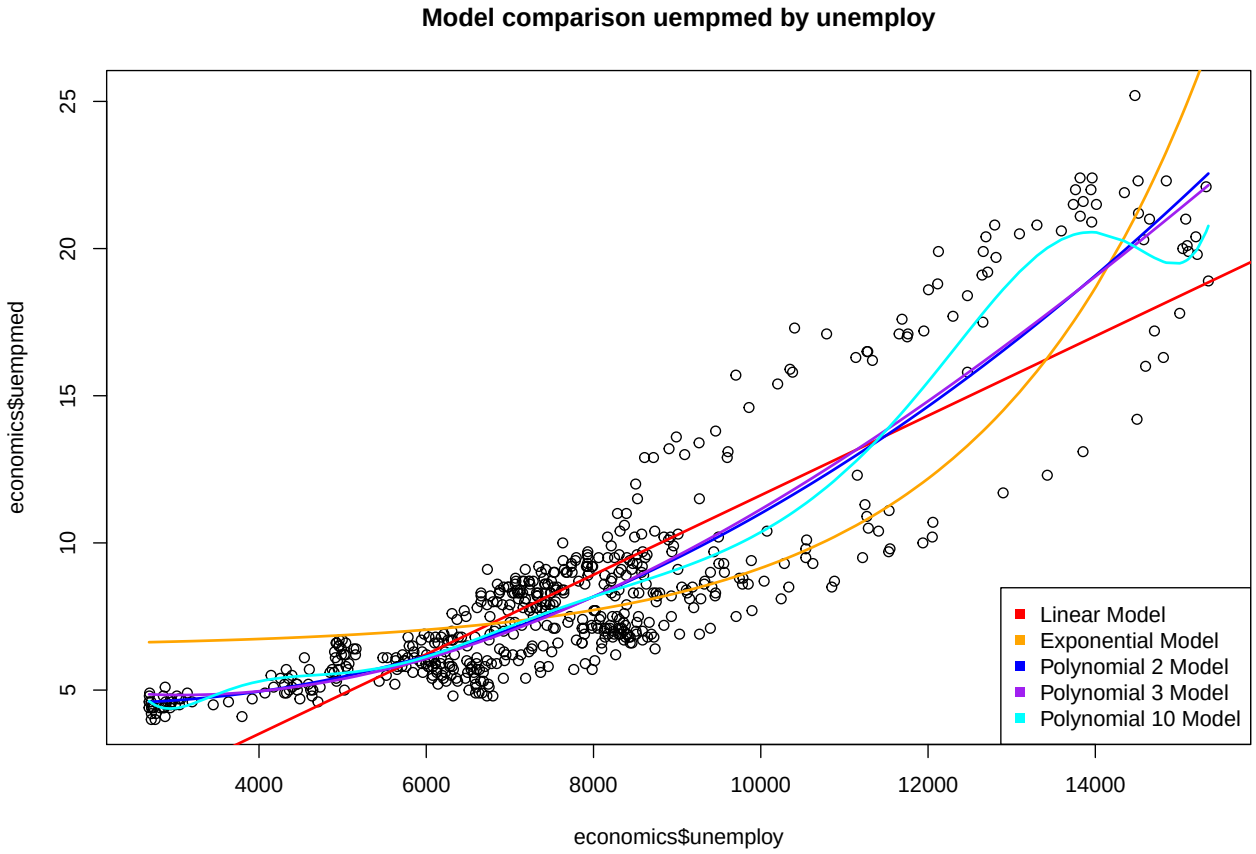
2.2)

Plot the corresponding data and add all the models for comparison.

```
# plot models unemploy ~ uempmed
plot(economics$unemploy ~ economics$uempmed, main = "Model comparison unemploy by uempmed")
colors <- c("red", "orange", "blue", "purple", "cyan")
abline(lm_unemploy$coeff[1], lm_unemploy$coeff[2], col = colors[1], lw = 2)
lines(sort(economics$uempmed), fitted(log_unemploy)[order(economics$uempmed)], col = colors[2],
    type = "l", lw = 2)
lines(sort(economics$uempmed), fitted(poly_unemploy_2)[order(economics$uempmed)],
    col = colors[3], type = "l", lw = 2)
lines(sort(economics$uempmed), fitted(poly_unemploy_3)[order(economics$uempmed)],
    col = colors[4], type = "l", lw = 2)
lines(sort(economics$uempmed), fitted(poly_unemploy_10)[order(economics$uempmed)],
    col = colors[5], type = "l", lw = 2)
legend("bottomright", legend = c("Linear Model", "Logarithmic Model", "Polynomial 2 Model",
    "Polynomial 3 Model", "Polynomial 10 Model"), pch = c(15, 15), col = colors)
```




```r
# plot models uempmed ~ unemploy
plot(economics$uempmed ~ economics$unemploy, main = "Model comparison uempmed by unemploy")
colors <- c("red", "orange", "blue", "purple", "cyan")
abline(lm_uempmed$coeff[1], lm_uempmed$coeff[2], col = colors[1], lw = 2)
lines(sort(economics$unemploy), fitted(expm_uempmed)[order(economics$unemploy)],
    col = colors[2], type = "l", lw = 2)
lines(sort(economics$unemploy), fitted(poly_uempmed_2)[order(economics$unemploy)],
    col = colors[3], type = "l", lw = 2)
lines(sort(economics$unemploy), fitted(poly_uempmed_3)[order(economics$unemploy)],
    col = colors[4], type = "l", lw = 2)
lines(sort(economics$unemploy), fitted(poly_uempmed_10)[order(economics$unemploy)],
    col = colors[5], type = "l", lw = 2)
legend("bottomright", legend = c("Linear Model", "Exponential Model", "Polynomial 2 Model",
    "Polynomial 3 Model", "Polynomial 10 Model"), pch = c(15, 15), col = colors)
```



2.3)

Use the cv.glm function in the boot package for the following steps.

2.3.1)

Use cv.glm for Leave-one-out Cross Validation to compare the models above.

```
# Results for unemploy ~ uempmed
results_unemploy_loocv <- matrix(0, 5, 3)
rownames(results_unemploy_loocv) <- c("Linear LOOCV", "Log LOOCV", "Poly 2 LOOCV",
    "Poly 3 LOOCV", "Poly 10 LOOCV")

# Leave One Out Cross Validation (k is number of observations)
K <- nrow(economics)
# linear model
results_unemploy_loocv[1, ] <- eval_cv(lm_unemploy, economics, K)
# log model
results_unemploy_loocv[2, ] <- eval_cv(log_unemploy, economics, K)
# poly model second order
results_unemploy_loocv[3, ] <- eval_cv(poly_unemploy_2, economics, K)
# poly model third order
results_unemploy_loocv[4, ] <- eval_cv(poly_unemploy_3, economics, K)
# poly model tenth order
results_unemploy_loocv[5, ] <- eval_cv(poly_unemploy_10, economics, K)
```

```r
# Results for uempmed ~ unemploy
results_uempmed_loocv <- matrix(0, 5, 3)
rownames(results_uempmed_loocv) <- c("Linear LOOCV", "Exp LOOCV", "Poly 2 LOOCV",
    "Poly 3 LOOCV", "Poly 10 LOOCV")

# Leave One Out Cross Validation (k is number of observations)
K <- nrow(economics)
# linear model
results_uempmed_loocv[1, ] <- eval_cv(lm_uempmed, economics, K)
# exp model
results_uempmed_loocv[2, ] <- eval_cv(expm_uempmed, economics, K)
# poly model second order
results_uempmed_loocv[3, ] <- eval_cv(poly_uempmed_2, economics, K)
# poly model third order
results_uempmed_loocv[4, ] <- eval_cv(poly_uempmed_3, economics, K)
# poly model tenth order
results_uempmed_loocv[5, ] <- eval_cv(poly_uempmed_10, economics, K)
```

2.3.2)

Use cv.glm for 5-fold and 10-fold Cross Validation to compare the models above. Compare the Root Mean Squared Error, Mean Squared Error and Median Absolute Deviation.

```r
# Results for unemploy ~ uempmed
results_unemploy_fcv <- matrix(0, 10, 3)
rownames(results_unemploy_fcv) <- c("Linear CV K=5", "Log CV K=5", "Poly 2 CV K=5",
    "Poly 3 CV K=5", "Poly 10 CV K=5", "Linear CV K=10", "Log CV K=10", "Poly 2 CV K=10",
    "Poly 3 CV K=10", "Poly 10 CV K=10")
# 5-Fold Cross Validation
K <- 5
# linear model
results_unemploy_fcv[1, ] <- eval_cv(lm_unemploy, economics, K)
# log model
results_unemploy_fcv[2, ] <- eval_cv(log_unemploy, economics, K)
# poly model second order
results_unemploy_fcv[3, ] <- eval_cv(poly_unemploy_2, economics, K)
# poly model third order
results_unemploy_fcv[4, ] <- eval_cv(poly_unemploy_3, economics, K)
# poly model tenth order
results_unemploy_fcv[5, ] <- eval_cv(poly_unemploy_10, economics, K)

# 10-Fold Cross Validation
K <- 10
# linear model
results_unemploy_fcv[6, ] <- eval_cv(lm_unemploy, economics, K)
# log model
results_unemploy_fcv[7, ] <- eval_cv(log_unemploy, economics, K)
# poly model second order
results_unemploy_fcv[8, ] <- eval_cv(poly_unemploy_2, economics, K)
# poly model third order
results_unemploy_fcv[9, ] <- eval_cv(poly_unemploy_3, economics, K)
# poly model tenth order
results_unemploy_fcv[10, ] <- eval_cv(poly_unemploy_10, economics, K)
```

```r
# Results for uempmed ~ unemploy
results_uempmed_fcv <- matrix(0, 10, 3)
rownames(results_uempmed_fcv) <- c("Linear CV K=5", "Exp CV K=5", "Poly 2 CV K=5",
    "Poly 3 CV K=5", "Poly 10 CV K=5", "Linear CV K=10", "Exp CV K=10", "Poly 2 CV K=10",
    "Poly 3 CV K=10", "Poly 10 CV K=10")
# 5-Fold Cross Validation
K <- 5
# linear model
results_uempmed_fcv[1, ] <- eval_cv(lm_uempmed, economics, K)
# exp model
results_uempmed_fcv[2, ] <- eval_cv(expm_uempmed, economics, K)
# poly model second order
results_uempmed_fcv[3, ] <- eval_cv(poly_uempmed_2, economics, K)
# poly model third order
results_uempmed_fcv[4, ] <- eval_cv(poly_uempmed_3, economics, K)
# poly model tenth order
results_uempmed_fcv[5, ] <- eval_cv(poly_uempmed_10, economics, K)

# 10-Fold Cross Validation
K <- 10
# linear model
results_uempmed_fcv[6, ] <- eval_cv(lm_uempmed, economics, K)
# exp model
results_uempmed_fcv[7, ] <- eval_cv(expm_uempmed, economics, K)
# poly model second order
results_uempmed_fcv[8, ] <- eval_cv(poly_uempmed_2, economics, K)
# poly model third order
results_uempmed_fcv[9, ] <- eval_cv(poly_uempmed_3, economics, K)
# poly model tenth order
results_uempmed_fcv[10, ] <- eval_cv(poly_uempmed_10, economics, K)
```

```r
table <- rbind(results_unemploy_loocv, results_unemploy_fcv)
colnames(table) <- c("RMSE", "MSE", "MAD")

kable(table, caption = "Comparison of Model uempmed ~ unemploy CV Performance")
```

Table 3: Comparison of Model uempmed ~ unemploy CV Performance

	RMSE	MSE	MAD
Linear LOOCV	1040.0126	1715199	1543.270
Log LOOCV	980.4149	1333988	1453.795
Poly 2 LOOCV	1012.2492	1432517	1501.059
Poly 3 LOOCV	984.5519	1366371	1459.499
Poly 10 LOOCV	994.8375	4524996	1474.028
Linear CV K=5	1303.5725	1727740	1261.230
Log CV K=5	1151.5245	1337800	1338.206
Poly 2 CV K=5	1193.4690	1426084	1369.817
Poly 3 CV K=5	1162.8218	1378126	1329.038
Poly 10 CV K=5	1743.6093	1278282	1239.385
Linear CV K=10	1304.0201	1718612	1324.673
Log CV K=10	1154.0975	1330354	1319.889
Poly 2 CV K=10	1192.0500	1432607	1396.953
Poly 3 CV K=10	1168.3437	1371132	1334.406

	RMSE	MSE	MAD
Poly 10 CV K=10	1752.2842	4014066	1190.101

```
table <- rbind(results_uempmed_loocv[-2, ], results_uempmed_fcv)
colnames(table) <- c("RMSE", "MSE", "MAD")

kable(table, caption = "Comparison of Model unemploy ~ uempmed CV Performance")
```

Table 4: Comparison of Model unemploy ~ uempmed CV Performance

	RMSE	MSE	MAD
Linear LOOCV	1.602072	4.159756	2.374407
Poly 2 LOOCV	1.296974	3.005072	1.923278
Poly 3 LOOCV	1.309643	3.009880	1.942484
Poly 10 LOOCV	1.211396	2.832152	1.798069
Linear CV K=5	2.040128	4.151191	1.946248
Exp CV K=5	2.207081	5.225557	1.788029
Poly 2 CV K=5	1.729666	3.034949	1.513578
Poly 3 CV K=5	1.737669	3.021622	1.545242
Poly 10 CV K=5	1.659361	2.804697	1.437626
Linear CV K=10	2.030770	4.163645	1.951563
Exp CV K=10	2.575442	7.803139	1.806892
Poly 2 CV K=10	1.720266	2.989153	1.509884
Poly 3 CV K=10	1.716875	3.016139	1.505347
Poly 10 CV K=10	1.667640	2.800632	1.408551

Elvauation of Moedl

All measures clearly indicate that the liner model performs worst which we also clearly see in the model fit graph. The polynomial models of second and third degree and the logarithmic model all perform very similar where the polynomial model of third degree is best performing model in general. For this models we can not detect any overfitting. The polynomial model of tenth degree performs very good when we evaluate it with LOOCV at least for RMSE and MAD but not so well with 5-fold and 10-fold CV. We also notice that the polynomial model of tenth degree clearly overfitts.

The exponential model did not return LOOCV errors unfortunately.

2.4)

Explain based on the CV and graphical model fits the concepts of Underfitting, Overfitting and how to apply cross-validation to determine the appropriate model fit. Also, describe the different variants of cross validation in this context.

In our experiments we underfitt when we use a linear model. Both plots show us a clear non linear relationship between unemploy and uempmed. When we use a linear model in this case we ignore this fact which leads to a model which is too simple in its strcture. On the other hand we overfitt when we use a polynomial model of tenth degree. In the plots we clearly see that the model tries to fit the training data too much. This model would not perform very well on new unseen data in some regions. So the model is to complex and does not generalise well. The challenge is to find a model which balances these factors and generalizes well.

The different variants of CV are Leave-one-out CV and K-Fold CV. LOOCV has the advantage of having less bias because all n-1 observations are used in the model estimation and we do not need random splitting. The

main disadvantage is the computational cost of this evaluation method. In K-Fold CV we split the data k times and use k-1 parts of the data for our model estimation. This method is more effective with usual values of 5 and 10 for k.