Exercise 8

Cross Validation of Models

Olesia Galynskaia 12321492

2025

Fix the random seed for reproducibility

```
set.seed(12321492)
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE, fig.width = 6, fig.height = 4, dpi
```

Task 1

Load Auto data, inspect it

I load the Auto dataset from the ISLR package and check its structure and variable meaning using the help page.

```
# load package and dataset
library(ISLR)

# inspect dataset structure and variable types
?Auto
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
```

```
summary(Auto)
```

```
##       mpg          cylinders      displacement     horsepower        weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##
##   acceleration        year           origin                      name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5
##  Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5
##  Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4
##  Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
##                                                  (Other)           :365
```

```r
# check missing values
colSums(is.na(Auto))
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##            0            0            0            0            0            0
##         year       origin         name
##            0            0            0
```

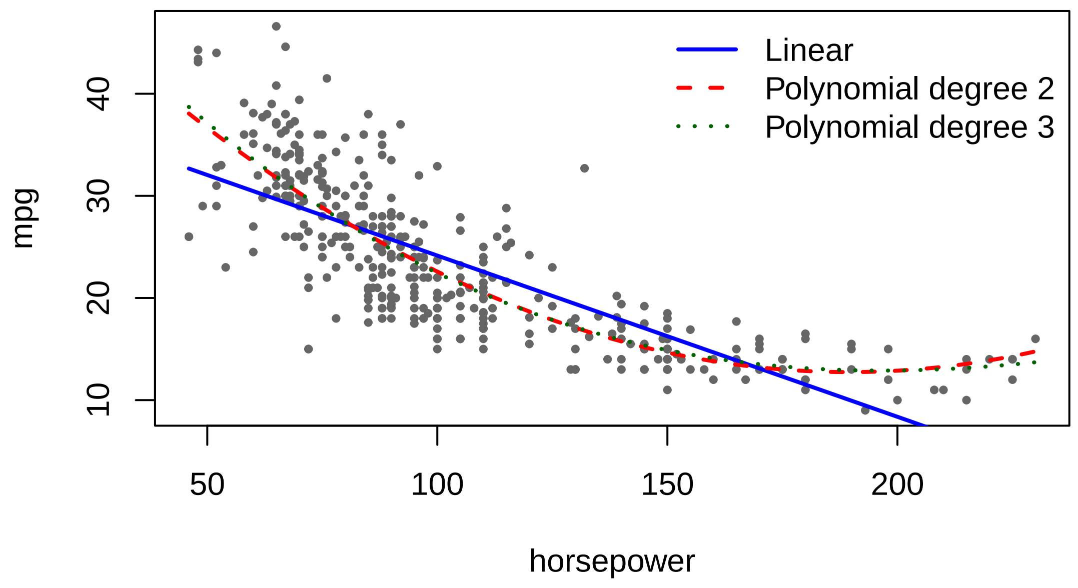
The Auto dataset contains 392 observations and 9 variables describing technical characteristics of cars and their fuel efficiency.
The response variable mpg measures fuel consumption in miles per gallon, while the explanatory variables include engine characteristics such as horsepower, displacement, weight, and cylinders, as well as year and origin of the car.

All variables are numeric except for name, which is a factor identifying the car model.
There are no missing values in the dataset, as observations with missing entries were removed beforehand.
Overall, the dataset is well structured and suitable for regression analysis of fuel efficiency.

1. Fit the models

I fit three regression models for mpg as a function of horsepower: a linear model and polynomial models of degree 2 and 3, and visualize them on top of the scatterplot to compare their shapes.

```r
m1 <- lm(mpg ~ horsepower, data = Auto)
m2 <- lm(mpg ~ poly(horsepower, 2), data = Auto)
m3 <- lm(mpg ~ poly(horsepower, 3), data = Auto)

# a grid of horsepower values
hp_grid <- seq(min(Auto$horsepower, na.rm = TRUE),
               max(Auto$horsepower, na.rm = TRUE),
               length.out = 300)

# predictions
pred1 <- predict(m1, newdata = data.frame(horsepower = hp_grid))
pred2 <- predict(m2, newdata = data.frame(horsepower = hp_grid))
pred3 <- predict(m3, newdata = data.frame(horsepower = hp_grid))

# scatterplot
plot(Auto$horsepower, Auto$mpg,
     xlab = "horsepower", ylab = "mpg",
     pch = 16, cex = 0.6, col = "grey40")

lines(hp_grid, pred1, col = "blue", lwd = 2)
lines(hp_grid, pred2, col = "red", lwd = 2, lty = 2)
lines(hp_grid, pred3, col = "darkgreen", lwd = 2, lty = 3)

legend("topright",
       legend = c("Linear", "Polynomial degree 2", "Polynomial degree 3"),
       col = c("blue", "red", "darkgreen"),
       lwd = 2, lty = c(1, 2, 3), bty = "n")
```



The scatterplot shows a clear negative relationship between horsepower and mpg, with higher horsepower associated with lower fuel efficiency.
The linear model captures the overall downward trend but deviates systematically at low and high horsepower values.
Both polynomial models follow the curvature in the data more closely, especially for small and large horsepower.
The degree 3 model provides only a slight additional adjustment compared to the quadratic model, suggesting that a second-degree polynomial may already be sufficient.

2. Validation set approach

I use the validation set approach to compare the three models.
The data is randomly split into training and test sets 50/50 and 70/30.
Each model is fitted on the training set and evaluated on the test set using MSE, RMSE and MAD.

Function to compute error measures

```
compute_errors <- function(y_true, y_pred) {
  mse  <- mean((y_true - y_pred)^2)
  rmse <- sqrt(mse)
  mad  <- median(abs(y_true - y_pred))
  c(MSE = mse, RMSE = rmse, MAD = mad)
}
```

50/50 split

```
n <- nrow(Auto)
train_id_50 <- sample(1:n, size = n / 2)

train_50 <- Auto[train_id_50, ]
test_50  <- Auto[-train_id_50, ]

# fit models on training data
m1_50 <- lm(mpg ~ horsepower, data = train_50)
m2_50 <- lm(mpg ~ poly(horsepower, 2), data = train_50)
m3_50 <- lm(mpg ~ poly(horsepower, 3), data = train_50)

# predictions on test data
pred1_50 <- predict(m1_50, newdata = test_50)
pred2_50 <- predict(m2_50, newdata = test_50)
pred3_50 <- predict(m3_50, newdata = test_50)

# error measures
err_50 <- rbind(
  Linear = compute_errors(test_50$mpg, pred1_50),
  Poly2  = compute_errors(test_50$mpg, pred2_50),
  Poly3  = compute_errors(test_50$mpg, pred3_50)
)

err_50
```

```
##             MSE     RMSE      MAD
## Linear 19.92186 4.463391 2.921380
## Poly2  19.67291 4.435415 2.534157
## Poly3  20.28708 4.504118 2.437058
```

For the 50/50 split, all three models show comparable prediction errors, which indicates that the general relationship between horsepower and mpg is already captured reasonably well.
The linear model reflects the overall decreasing trend, but its higher errors suggest that it is too restrictive and cannot fully capture the curvature present in the data.
The quadratic model slightly improves MSE and RMSE, indicating that introducing a simple nonlinear term helps to better describe the relationship between horsepower and fuel efficiency.
The cubic model does not substantially improve overall prediction accuracy, suggesting that additional flexibility mainly fits local variations rather than the underlying structure.

70/30 split

```r
train_id_70 <- sample(1:n, size = 0.7 * n)

train_70 <- Auto[train_id_70, ]
test_70  <- Auto[-train_id_70, ]

# fit models on training data
m1_70 <- lm(mpg ~ horsepower, data = train_70)
m2_70 <- lm(mpg ~ poly(horsepower, 2), data = train_70)
m3_70 <- lm(mpg ~ poly(horsepower, 3), data = train_70)

# predictions on test data
pred1_70 <- predict(m1_70, newdata = test_70)
pred2_70 <- predict(m2_70, newdata = test_70)
pred3_70 <- predict(m3_70, newdata = test_70)

# error measures
err_70 <- rbind(
  Linear = compute_errors(test_70$mpg, pred1_70),
  Poly2  = compute_errors(test_70$mpg, pred2_70),
  Poly3  = compute_errors(test_70$mpg, pred3_70)
)

err_70
```

```
##             MSE     RMSE      MAD
## Linear 25.20639 5.020597 3.472426
## Poly2  20.31709 4.507448 2.623270
## Poly3  20.85199 4.566398 2.814605
```

For the 70/30 split, the limitations of the linear model become more pronounced, as it yields clearly higher prediction errors.
Both polynomial models perform better, which confirms that the relationship between horsepower and mpg is nonlinear.
The quadratic model achieves the lowest errors, indicating that a smooth curved relationship is sufficient to describe the data.
The cubic model adds extra flexibility, but this does not translate into better predictive performance and may reflect mild overfitting.

Comparison

Across both train-test splits, the results consistently suggest that the relationship between horsepower and mpg is nonlinear but relatively smooth.
The quadratic model provides enough flexibility to capture this curvature while remaining stable across different splits.
In contrast, the linear model is too simple, and the cubic model appears unnecessarily complex for this problem.
Overall, the quadratic model represents the best compromise between model simplicity and predictive accuracy for this dataset.

3. CV for models comparison

1. Leave-One-Out Cross-Validation

```r
library(boot)

glm1 <- glm(mpg ~ horsepower, data = Auto)
glm2 <- glm(mpg ~ poly(horsepower, 2), data = Auto)
glm3 <- glm(mpg ~ poly(horsepower, 3), data = Auto)

cv1_loocv <- cv.glm(Auto, glm1)$delta[1]
cv2_loocv <- cv.glm(Auto, glm2)$delta[1]
cv3_loocv <- cv.glm(Auto, glm3)$delta[1]

loocv_results <- data.frame(
  Model = c("Linear", "Polynomial degree 2", "Polynomial degree 3"),
  LOOCV_MSE = c(cv1_loocv, cv2_loocv, cv3_loocv)
)

loocv_results
```

```
##                 Model LOOCV_MSE
## 1              Linear  24.23151
## 2 Polynomial degree 2  19.24821
## 3 Polynomial degree 3  19.33498
```

The LOOCV results show a clear difference between the linear and polynomial models.
The linear model has the highest cross-validated MSE (24.23), indicating that it does not capture the structure of the data well.
Both polynomial models substantially reduce the error, with the quadratic model achieving an MSE of 19.25 and the cubic model 19.33.
The improvement from degree 2 to degree 3 is very small, suggesting that increasing model complexity beyond a quadratic term brings little benefit.

2. 5-fold and 10-fold Cross-Validation

```r
# 5-fold CV
cv1_5 <- cv.glm(Auto, glm1, K = 5)$delta[1]
cv2_5 <- cv.glm(Auto, glm2, K = 5)$delta[1]
cv3_5 <- cv.glm(Auto, glm3, K = 5)$delta[1]

# 10-fold CV
cv1_10 <- cv.glm(Auto, glm1, K = 10)$delta[1]
cv2_10 <- cv.glm(Auto, glm2, K = 10)$delta[1]
cv3_10 <- cv.glm(Auto, glm3, K = 10)$delta[1]

cv_kfold_results <- data.frame(
  Model = c("Linear", "Polynomial degree 2", "Polynomial degree 3"),
  CV5_MSE  = c(cv1_5, cv2_5, cv3_5),
  CV10_MSE = c(cv1_10, cv2_10, cv3_10)
)

cv_kfold_results
```

```
##                  Model  CV5_MSE CV10_MSE
## 1               Linear 24.17327 24.32538
## 2 Polynomial degree 2 19.33195 19.24021
## 3 Polynomial degree 3 19.47831 19.31513
```

For 5-fold cross-validation, the linear model again performs worst with an MSE of 24.17.
The quadratic model achieves the lowest MSE (19.33), while the cubic model shows a slightly higher error (19.48).
This indicates that a moderate level of nonlinearity improves predictive performance, whereas additional complexity does not lead to further improvement.

The 10-fold cross-validation results are very similar to the 5-fold results.
The linear model has the highest error (24.33), while both polynomial models perform better.
The quadratic model achieves the lowest MSE (19.24), and the cubic model is only marginally worse (19.32).
Overall, the results are stable across different numbers of folds.

4. Comparison

Both the validation set approach and cross-validation lead to consistent conclusions.
Across all methods, the linear model shows the highest prediction errors, indicating that it is too simple for this problem.
The polynomial models perform better, confirming the presence of a nonlinear relationship between horsepower and mpg.

Among the polynomial models, the quadratic model consistently achieves the lowest or near-lowest error across validation splits, LOOCV, and k-fold cross-validation.
The cubic model does not provide a meaningful improvement and sometimes performs slightly worse, suggesting unnecessary model complexity.

Overall, the quadratic model represents the best compromise between predictive accuracy and model simplicity for this dataset.

Task 2

Load Auto data, inspect it

I load the dataset economics from the ggplot2 package and study the relationship between the number of unemployed persons (unemploy) and the median duration of unemployment (uempmed).

```r
library(ggplot2)

data(economics)

str(economics)
```

```
## spc_tbl_ [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 ...
```

```r
summary(economics)
```

```
##       date                 pce               pop            psavert
##  Min.   :1967-07-01   Min.   :  506.7   Min.   :198712   Min.   : 2.200
##  1st Qu.:1979-06-08   1st Qu.: 1578.3   1st Qu.:224896   1st Qu.: 6.400
##  Median :1991-05-16   Median : 3936.8   Median :253060   Median : 8.400
##  Mean   :1991-05-17   Mean   : 4820.1   Mean   :257160   Mean   : 8.567
##  3rd Qu.:2003-04-23   3rd Qu.: 7626.3   3rd Qu.:290291   3rd Qu.:11.100
##  Max.   :2015-04-01   Max.   :12193.8   Max.   :320402   Max.   :17.300
##     uempmed          unemploy
##  Min.   : 4.000   Min.   : 2685
##  1st Qu.: 6.000   1st Qu.: 6284
##  Median : 7.500   Median : 7494
##  Mean   : 8.609   Mean   : 7771
##  3rd Qu.: 9.100   3rd Qu.: 8686
##  Max.   :25.200   Max.   :15352
```

```r
dat_econ <- economics[, c("unemploy", "uempmed")]
```

The dataset contains 574 monthly observations covering the period from 1967 to 2015. The variable unemploy represents the total number of unemployed persons, while uempmed measures the median duration of unemployment in weeks.

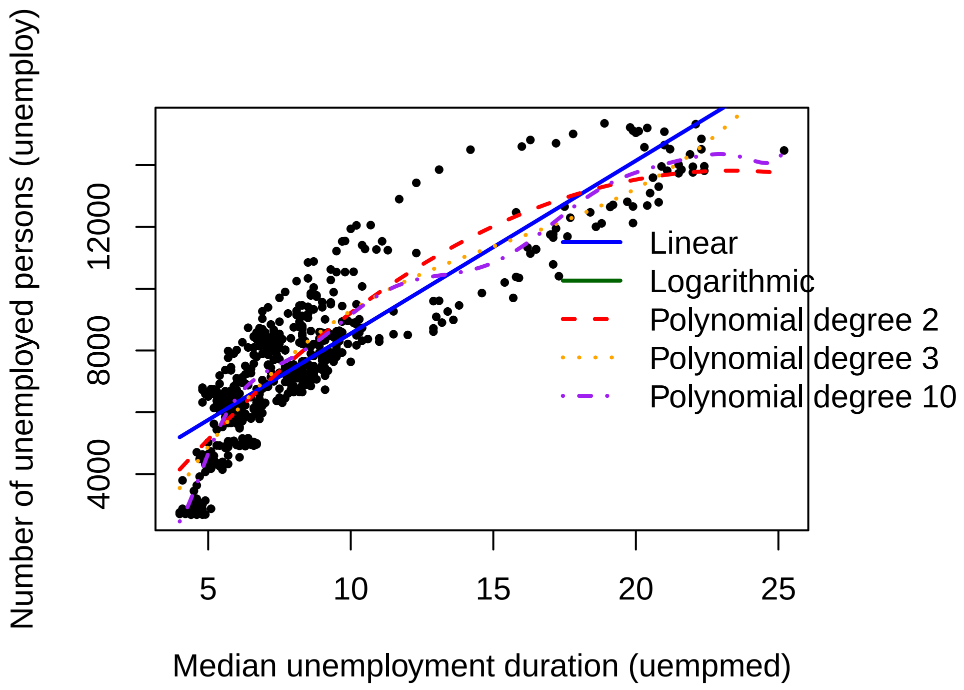
1. Fit the models

```r
# linear model
m_lin <- lm(unemploy ~ uempmed, data = dat_econ)

# logarithmic model
m_log <- lm(log(unemploy) ~ uempmed, data = dat_econ)

# polynomial models
m_poly2 <- lm(unemploy ~ poly(uempmed, 2), data = dat_econ)
m_poly3 <- lm(unemploy ~ poly(uempmed, 3), data = dat_econ)
m_poly10 <- lm(unemploy ~ poly(uempmed, 10), data = dat_econ)
```

2. Plots for comparison

I visualize the relationship between the median duration of unemployment (uempmed) and the number of unemployed persons (unemploy) using a scatter plot. On top of the observed data, I add the fitted values from several regression models, including linear, logarithmic, and polynomial models of different degrees. This graphical comparison helps to assess how well each model captures the structure of the data and to visually inspect potential underfitting or overfitting.

```r
# larger plotting window (works in RMarkdown and RStudio)
par(mar = c(5, 5, 4, 8))  # extra space on the right for legend

plot(dat_econ$uempmed, dat_econ$unemploy,
     xlab = "Median unemployment duration (uempmed)",
     ylab = "Number of unemployed persons (unemploy)",
     pch = 16, cex = 0.6)

x_grid <- seq(min(dat_econ$uempmed),
              max(dat_econ$uempmed),
              length.out = 300)

lines(x_grid, predict(m_lin, newdata = data.frame(uempmed = x_grid)),
      col = "blue", lwd = 2)

lines(x_grid, predict(m_log, newdata = data.frame(uempmed = x_grid)),
      col = "darkgreen", lwd = 2)

lines(x_grid, predict(m_poly2, newdata = data.frame(uempmed = x_grid)),
      col = "red", lwd = 2, lty = 2)

lines(x_grid, predict(m_poly3, newdata = data.frame(uempmed = x_grid)),
      col = "orange", lwd = 2, lty = 3)

lines(x_grid, predict(m_poly10, newdata = data.frame(uempmed = x_grid)),
      col = "purple", lwd = 2, lty = 4)

legend("right",
       inset = -0.25,
       legend = c("Linear","Logarithmic","Polynomial degree 2",
                  "Polynomial degree 3","Polynomial degree 10"),
       col = c("blue","darkgreen","red","orange","purple"),
       lty = c(1,1,2,3,4),
       lwd = 2,
       bty = "n",
       xpd = TRUE)
```



The plot shows a clear positive relationship between the median duration of unemployment and the number of unemployed persons.

The linear model captures the general upward trend but fails to account for the curvature present in the data.

The logarithmic model improves the fit at lower values of unemployment duration, but still underestimates the trend for higher values.

The polynomial models of degree 2 and 3 follow the data more closely, providing a good balance between flexibility and smoothness and therefore appear to give the most reasonable fit.
In contrast, the polynomial model of degree 10 adapts very tightly to the observed points and exhibits strong local fluctuations, which indicates overfitting rather than a meaningful underlying relationship.

3. CV for models comparison

1. Leave-One-Out Cross-Validation

```
g_lin   <- glm(unemploy ~ uempmed, data = dat_econ)
g_log   <- glm(unemploy ~ log(uempmed), data = dat_econ)
g_poly2 <- glm(unemploy ~ poly(uempmed, 2), data = dat_econ)
g_poly3 <- glm(unemploy ~ poly(uempmed, 3), data = dat_econ)
g_poly10 <- glm(unemploy ~ poly(uempmed, 10), data = dat_econ)

loocv_mse <- c(
  Linear      = cv.glm(dat_econ, g_lin)$delta[1],
  Logarithmic = cv.glm(dat_econ, g_log)$delta[1],
  Poly2       = cv.glm(dat_econ, g_poly2)$delta[1],
  Poly3       = cv.glm(dat_econ, g_poly3)$delta[1],
```

```r
  Poly10      = cv.glm(dat_econ, g_poly10)$delta[1]
)

loocv_tbl <- data.frame(
  Model = names(loocv_mse),
  LOOCV_MSE = as.numeric(loocv_mse),
  LOOCV_RMSE = sqrt(as.numeric(loocv_mse))
)

loocv_tbl
```

```
##         Model LOOCV_MSE LOOCV_RMSE
## 1      Linear   1715211   1309.661
## 2 Logarithmic   1333997   1154.988
## 3       Poly2   1432531   1196.884
## 4       Poly3   1366405   1168.933
## 5      Poly10   4530738   2128.553
```

The linear model shows the weakest performance under LOOCV, which indicates that a simple linear relationship is not sufficient to describe the data.
The logarithmic model performs best and clearly improves the fit by capturing the nonlinear structure of the relationship.
The polynomial models of degree 2 and 3 also improve over the linear model, with the degree 3 polynomial giving a slightly better and more flexible fit.
The polynomial model of degree 10 performs very poorly under LOOCV, which is a clear sign of overfitting and poor generalization.

2. 5-fold and 10-fold Cross-Validation

```r
# helper to compute CV MSE via cv.glm for a given K
cv_mse_k <- function(glm_fit, K) {
  cv.glm(dat_econ, glm_fit, K = K)$delta[1]
}

# 5-fold CV MSE
cv5_mse <- c(
  Linear      = cv_mse_k(g_lin, 5),
  Logarithmic = cv_mse_k(g_log, 5),
  Poly2       = cv_mse_k(g_poly2, 5),
  Poly3       = cv_mse_k(g_poly3, 5),
  Poly10      = cv_mse_k(g_poly10, 5)
)

# 10-fold CV MSE
cv10_mse <- c(
  Linear      = cv_mse_k(g_lin, 10),
  Logarithmic = cv_mse_k(g_log, 10),
  Poly2       = cv_mse_k(g_poly2, 10),
  Poly3       = cv_mse_k(g_poly3, 10),
  Poly10      = cv_mse_k(g_poly10, 10)
)
```

```r
cv_tbl <- data.frame(
  Model = names(cv5_mse),
  CV5_MSE = as.numeric(cv5_mse),
  CV5_RMSE = sqrt(as.numeric(cv5_mse)),
  CV10_MSE = as.numeric(cv10_mse),
  CV10_RMSE = sqrt(as.numeric(cv10_mse))
)

cv_tbl
```

```
##         Model CV5_MSE CV5_RMSE CV10_MSE CV10_RMSE
## 1      Linear 1715701 1309.848  1712540  1308.641
## 2 Logarithmic 1332916 1154.520  1331419  1153.871
## 3       Poly2 1421749 1192.371  1424683  1193.601
## 4       Poly3 1377930 1173.853  1376102  1173.074
## 5      Poly10 1772149 1331.221  7162991  2676.377
```

Using 5-fold cross-validation, the linear model again shows the weakest performance, which indicates underfitting.
The logarithmic model achieves the best results and provides a good description of the nonlinear relationship in the data.
The polynomial models of degree 2 and 3 perform better than the linear model, with degree 3 giving a slightly more flexible fit.
The polynomial model of degree 10 performs worse than the lower-degree models, which suggests that it is already overfitting the data.

The results from 10-fold cross-validation confirm the conclusions from 5-fold CV.
The linear model remains too restrictive and does not capture the curvature in the data.
The logarithmic model again shows the most stable and best performance, indicating strong generalization ability.
The polynomial model of degree 3 performs reasonably well, but does not improve over the logarithmic model.
The polynomial model of degree 10 shows unstable behavior and a clear loss in predictive performance, which is a strong indication of overfitting.

Across all cross-validation methods, the logarithmic model provides the best balance between model complexity and predictive performance.
The linear model underfits the data, while the high-degree polynomial clearly overfits.
Cross-validation helps to identify this trade-off and shows that moderately flexible models generalize better than both overly simple and overly complex models.

4. Underfitting, Overfitting and Model Selection using Cross-Validation

Underfitting occurs when a model is too simple to capture the structure of the data.
In this analysis, the linear model is an example of underfitting: it captures only a general increasing trend but fails to describe the curvature visible in the data, which leads to consistently higher prediction errors across all cross-validation methods.

Overfitting occurs when a model is overly complex and starts fitting noise instead of the underlying relationship.
This behavior is clearly observed for the polynomial model of degree 10. Although it follows the data points very closely in the plot, its cross-validation errors are high and unstable, especially for higher-fold CV, indicating poor generalization to unseen data.

Models with moderate complexity, such as the logarithmic model and the polynomial model of degree 3, achieve a better balance between bias and variance.
These models capture the nonlinear relationship in the data while maintaining stable and relatively low cross-validation errors, which indicates good predictive performance.

Cross-validation is used to determine the appropriate model fit by estimating how well a model generalizes to new data.
Leave-one-out cross-validation uses almost the entire dataset for training and provides an accurate but computationally expensive estimate of prediction error.
K-fold cross-validation (such as 5-fold and 10-fold CV) splits the data into larger validation sets, which reduces computation time and often gives more stable estimates.
Comparing results across different cross-validation schemes helps confirm whether a model's performance is robust and not dependent on a specific data split.

Overall, cross-validation clearly shows that neither the simplest nor the most complex model is optimal.
Instead, models with moderate flexibility provide the best trade-off between underfitting and overfitting in this context.