BACS HW16
Student ID: 110065508

Due: 2022/06/05
Special thanks to 108071001

Question 1. First, split the data into train and test sets (70:30).

```r
# Question 1
set.seed(48763)
train_indices <- sample(1:nrow(cars_log), size=0.70*nrow(cars_log)) # Split the dataset
test_indices <- setdiff(1:nrow(cars_log), train_indices)
train_set <- cars_log[train_indices,]
test_set <- cars_log[test_indices,]
```

(a) This is a regular regression problem

```r
# Question 1 (a)
lm_trained <- lm(log.mpg. ~ log.weight. + log.acceleration. +
                   model_year + factor(origin),
                 data=train_set) # train the model
model_report <- summary(lm_trained)
write.table(model_report$coefficients, file="1a.csv", sep = ",", col.names=NA)
```

	Estimate	Std. Error	t value	$Pr(>\|t\|)$
(Intercept)	7.68571305514322	0.381446482860176	20.1488633412318	1.34317494264466e-55
log.weight.	-0.892922806578692	0.0353182162549613	-25.282216976438	6.32840618595712e-73
log.acceleration.	0.0742013254761747	0.0446141749715566	1.66317825048835	0.0974457001142396
model_year	0.0303228848528229	0.00208984158869215	14.5096571036274	1.33646119837473e-35
factor(origin)2	0.0477657895161414	0.021722121774988	2.19894677006836	0.0287362555776429
factor(origin)3	0.0182650327019749	0.0226373246965084	0.806854738660532	0.420465631824332

(b) By the following codes,

```r
# Question 1 (b)
mpg_actual_train <- train_set$log.mpg. # true label of train set
mpg_actual_test <- test_set$log.mpg. # true label of test set
mpg_predicted_train <- predict(lm_trained, train_set) # predict on train set
mpg_predicted_test <- predict(lm_trained, test_set) # predict on test set
pred_err_train <- mpg_actual_train - mpg_predicted_train # error on training set
pred_err_test <- mpg_actual_test - mpg_predicted_test # error on testing set
mse_is <- mean((mpg_predicted_train - mpg_actual_train)^2) # MSE_IS
mse_oos <- mean((mpg_predicted_test - mpg_actual_test)^2) # MSE_OOS
```

we have $\mathrm{MSE}_{\mathrm{IS}} \approx 0.01316246$, and $\mathrm{MSE}_{\mathrm{OOS}} \approx 0.01401028$.

(c) I save the dataframe to a `.csv` file, then convert into LaTeXtable by online tools.

```r
# Question 1 (c)
result_dataframe <- cbind(mpg_actual_test, mpg_predicted_test, pred_err_test)
names(result_dataframe) <- c("Actual log.mpg.", "Predict log.mpg.", "error")
write.table(result_dataframe[1:5, 1:3], file="1c.csv", sep = ",", col.names=NA)
```

	mpg_actual_test	mpg_predicted_test	pred_err_test
1	2.83321334405622	2.72063665910161	0.112576684954602
3	3.67376581630389	3.46832859503583	0.205437221268057
4	2.77258872223978	2.79377976849132	-0.0211910462515355
9	2.70805020110221	2.98543919467625	-0.277388993574044
11	3.19867311755068	3.36255680850516	-0.163883690954478

Question 2. (a) I wrote a function to compute MSE:

```
# Question 2 (a)
MSE <- function(model, dataset, actual) {
  predicted <- predict(model, dataset) # predict
  pred_err <- actual - predicted # error
  mse <- mean((predicted - actual)^2) # MSE
  return(mse)
}

# Split the origin dataset without log-transformed
train_set_org <- cars[train_indices,]

# Compute the MSE_IS'
MSE_cars_lm <- MSE(cars_lm, train_set_org, train_set_org$mpg)
MSE_cars_log_lm <- MSE(cars_log_lm, train_set, train_set$log.mpg.)
MSE_cars_log_full_lm <- MSE(cars_log_full_lm, train_set, train_set$log.mpg.)
```

We have that

- `cars_lm`: $\mathrm{MSE}_{\mathrm{IS}} \approx 11.5206$
- `cars_log_lm`: $\mathrm{MSE}_{\mathrm{IS}} \approx 0.01325627$
- `cars_log_full_lm`: $\mathrm{MSE}_{\mathrm{IS}} \approx 0.01255269$

(b) The implementation of k-fold is in the following codes:

```
# Question 2 (b)
# Calculates mse_oos across all folds
k_fold_mse <- function(model, dataset, k=10) { # model should be a string
  fold_pred_errors <- sapply(1:k, \(i) {
    fold_i_pe(model, i, k, dataset)
  })
  pred_errors <- unlist(fold_pred_errors)
  mean(pred_errors^2)
}

# Calculates prediction error for fold i out of k
fold_i_pe <- function(model, i, k, dataset) {
  folds <- cut(1:nrow(dataset), k, labels = FALSE) # cut into 10 folds

  # Split the dataset
  test_indices <- which(folds == i)
  train_indices <- setdiff(1:nrow(dataset), train_indices)
  train_set <- dataset[train_indices,]
  test_set <- dataset[test_indices,]

  # train
  if (model == "cars_lm") {
    trained_model <- lm(mpg ~ weight + acceleration +
                          model_year + factor(origin), data=train_set)
    actual <- test_set$mpg
  }
  else if (model == "cars_log_lm") {
    trained_model <- lm(log.mpg. ~ log.weight. + log.acceleration. +
                          model_year + factor(origin), data=train_set)
```

```
    actual <- test_set$log.mpg.
  }
  else if (model == "cars_log_full_lm") {
    trained_model <- lm(log.mpg. ~ log.cylinders. + log.displacement. +
                          log.horsepower. + log.weight. + log.acceleration. +
                          model_year + factor(origin), data=train_set)
    actual <- test_set$log.mpg.
  }

  # predict
  predictions <- predict(trained_model, test_set)
  return(actual - predictions)
}

cars_lm_mse <- k_fold_mse("cars_lm", cars, 10)
cars_log_lm_mse <- k_fold_mse("cars_log_lm", cars_log, 10)
cars_log_full_lm_mse <- k_fold_mse("cars_log_full_lm", cars_log, 10)
```

We have that

- `cars_lm`: $\text{MSE}_{\text{OOS}} \approx 11.20695$
- `cars_log_lm`: $\text{MSE}_{\text{OOS}} \approx 0.01382051$
- `cars_log_full_lm`: $\text{MSE}_{\text{OOS}} \approx 0.01361829$

(ii) Since MSE_{OOS} drops from 11 to 0.01, so the non-linearity seem to harm the predictions.

(iii) As (ii), the multicollinearity seem to harm the predictions as well.

(c) Use the function I write in (b),

```
# Question 2 (c)
cars_log_lm_392folds_mse <- k_fold_mse("cars_log_lm", cars_log, 392)
```

In this case, $\text{MSE}_{\text{OOS}} \approx 0.01382051$. ∎