# Homework 2: Data Partition and Backward Selection

Thomas Zwiller

There are six questions (30 total points) in this assignment. The minimum increment is 1 point. Please type in your answers directly in the R Markdown file. After completion, **successfully** knit Rr as an html file. Submit **both** the html file and the R Markdown file via Canvas. Please name the R Markdown file in the following format: LastName_FirstName_HW2.Rmd, e.g. Zhao_Zifeng_HW2.Rmd.

## Used Car Dataset [12 points]

The used car dataset is the one we analyzed in class. Let's read in the data stored in `UsedCar.csv` and further partition the data into training and test data. Note that we use the same random seed `set.seed(7)` as in class to ensure reproducibility.

```
total_data <- read.csv("/Users/TomTheIntern/Desktop/Mendoza/Mod 2/Advanced Stats/Hmwk 2/UsedCar.csv", he
set.seed(7)
total_obs <- dim(total_data)[1]
# Data partition / Sample splitting
train_data_indices <- sample(1:total_obs, 0.8*total_obs)
train_data <- total_data[train_data_indices,]
test_data <- total_data[-train_data_indices,]
# Record the size of training data and test data
train_obs <- dim(train_data)[1]
```

### Q1 [3 points] Model Estimation

Instead of building linear regression models on the log-scale Price, let's build linear regression models for the original scale of Price, i.e. without log transformation to correct the right-skewness of Price.

**Q1(a)** [**2 points**] Fit a linear regression model of **original scale** Price w.r.t. all 10 predictors using the **training data**, name it `lm_full`.

```
lm_full <- lm(Price ~ . , data = train_data)
summary(lm_full)$r.squared
```

```
## [1] 0.8746515
```

**Q1(b)** [**1 points**] Check the estimated coefficient for `Mileage`, how do we interpret it?

```
summary(lm_full)$coefficients
```

```
##                     Estimate   Std. Error     t value      Pr(>|t|)
## (Intercept)     -8.664909e+03 1.392626e+03  -6.2219910  6.890394e-10
## Age             -1.221492e+02 2.862940e+00 -42.6656436 3.500996e-238
```

```
## Mileage          -1.669933e-02 1.411596e-03 -11.8301001  1.556575e-30
## Fuel_TypeDiesel   4.251779e+02 4.099777e+02   1.0370757  2.999213e-01
## Fuel_TypePetrol   2.050609e+03 3.905400e+02   5.2507024  1.807575e-07
## HP                2.140461e+01 3.985901e+00   5.3700817  9.539183e-08
## Metallic         -8.593943e+00 8.168572e+01  -0.1052074  9.162298e-01
## Automatic         1.208227e+02 1.764780e+02   0.6846334  4.937149e-01
## CC               -2.884598e-02 8.879009e-02  -0.3248783  7.453330e-01
## Doors            -9.750384e+01 4.374474e+01  -2.2289271  2.601377e-02
## Quarterly_Tax     8.550168e+00 1.783249e+00   4.7947148  1.845267e-06
## Weight            2.147624e+01 1.425264e+00  15.0682527  6.505069e-47
```

Answer: The Mileage variable has a coefficient of -1.669933e-02 meaning that for each increase of 1 mile the price goes down roughly $0.0016699. This makes sense because we know that the value of a car is negatively correlated with price: cars tend to be considered less valuable the more they are drive.

### Q2 [4 points] Backward Selection with BIC

**Q2(a)** [**2 points**] Perform backward selection for `lm_full` with **BIC** using the function `step()` and name the selected model `lm_bwd`. Make sure you use the correct `k` argument in the `step()` function.

```r
lm_bwd <- step(lm_full, direction = 'backward', k = log(nrow(train_data)))
```

```
## Start:  AIC=16503.03
## Price ~ Age + Mileage + Fuel_Type + HP + Metallic + Automatic +
##     CC + Doors + Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS   AIC
## - Metallic       1      18191 1866998122 16496
## - CC             1     173461 1867153392 16496
## - Automatic      1     770331 1867750262 16496
## - Doors          1    8164941 1875144872 16501
## <none>                        1866979931 16503
## - Quarterly_Tax  1   37782168 1904762099 16519
## - HP             1   47393972 1914373903 16525
## - Fuel_Type      2   68499200 1935479131 16530
## - Mileage        1  230005464 2096985395 16629
## - Weight         1  373153144 2240133075 16705
## - Age            1 2991699172 4858679103 17594
##
## Step:  AIC=16495.99
## Price ~ Age + Mileage + Fuel_Type + HP + Automatic + CC + Doors +
##     Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS   AIC
## - CC             1     175928 1867174050 16489
## - Automatic      1     776351 1867774472 16489
## - Doors          1    8245518 1875243640 16494
## <none>                        1866998122 16496
## - Quarterly_Tax  1   37802180 1904800302 16512
## - HP             1   47396574 1914394696 16518
## - Fuel_Type      2   68548708 1935546830 16523
## - Mileage        1  230253765 2097251887 16622
## - Weight         1  373141986 2240140108 16698
```

```
## - Age            1 2997877140 4864875261 17588
##
## Step:  AIC=16489.06
## Price ~ Age + Mileage + Fuel_Type + HP + Automatic + Doors +
##     Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS   AIC
## - Automatic      1     698620 1867872670 16482
## - Doors          1    8337015 1875511065 16487
## <none>                        1867174050 16489
## - Quarterly_Tax  1   37769256 1904943306 16505
## - HP             1   47830799 1915004849 16511
## - Fuel_Type      2   69996282 1937170332 16517
## - Mileage        1  230648701 2097822751 16616
## - Weight         1  372972418 2240146468 16691
## - Age            1 2997732933 4864906984 17581
##
## Step:  AIC=16482.44
## Price ~ Age + Mileage + Fuel_Type + HP + Doors + Quarterly_Tax +
##     Weight
##
##                 Df  Sum of Sq        RSS   AIC
## - Doors          1    8984001 1876856672 16481
## <none>                        1867872670 16482
## - Quarterly_Tax  1   37391488 1905264158 16498
## - HP             1   47204197 1915076868 16504
## - Fuel_Type      2   71640001 1939512671 16512
## - Mileage        1  233546136 2101418807 16611
## - Weight         1  402977323 2270849993 16700
## - Age            1 3034975091 4902847761 17583
##
## Step:  AIC=16480.9
## Price ~ Age + Mileage + Fuel_Type + HP + Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS   AIC
## <none>                        1876856672 16481
## - Quarterly_Tax  1   36227462 1913084133 16496
## - Fuel_Type      2   64768058 1941624730 16506
## - HP             1   53928656 1930785328 16506
## - Mileage        1  242944944 2119801616 16614
## - Weight         1  413275152 2290131824 16702
## - Age            1 3051930827 4928787499 17582
```

**Q2(b)** [**2 points**] Examine the selected model in `lm_bwd`, list all the predictors that are eliminated during the backward selection process.

```r
summary(lm_bwd)$coefficients
```

```
##                     Estimate   Std. Error    t value      Pr(>|t|)
## (Intercept)     -8067.1404363 1.326023e+03  -6.083710  1.600873e-09
## Age              -122.1459393 2.836970e+00 -43.055062 2.958827e-241
## Mileage            -0.0170415 1.402869e-03 -12.147609  5.109204e-32
## Fuel_TypeDiesel   482.4162071 4.008245e+02   1.203560  2.290097e-01
```

```
## Fuel_TypePetrol  1989.2546759 3.899850e+02   5.100850   3.957534e-07
## HP                 21.9987254 3.843712e+00   5.723302   1.334607e-08
## Quarterly_Tax       8.3591439 1.781993e+00   4.690896   3.049288e-06
## Weight             20.5368825 1.296218e+00  15.843697   3.011272e-51
```

Answer: lm_bwd only uses Age, Mileage, Fuel_TypeDiesel, Fuel_TypePetrol, HP, Quarterly_Tax and Weight.

This means that the backward selection process discarded the Doors, Automatic, CC and Metallic variables.

### Q3 [5 points] Model Evaluation (Prediction)

**Q3(a) [2 points]** Use `lm_full` and `lm_bwd` to generate predictions for Price on the test data and store the prediction in `lm_full_pred` and `lm_bwd_pred` respectively.

```
lm_full_pred <- predict(lm_full, newdata = test_data)
lm_bwd_pred <- predict(lm_bwd, newdata = test_data)
```

**Q3(b) [2 points]** Use the `R` package `forecast` to evaluate the prediction performance of `lm_full_pred` and `lm_bwd_pred`. What are the MAE for `lm_full` and `lm_bwd`?

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
accuracy(lm_full_pred, test_data$Price)
```

```
##                 ME     RMSE      MAE       MPE     MAPE
## Test set 55.3254 1465.521 1058.757 -1.026606 10.59039
```

```
accuracy(lm_bwd_pred, test_data$Price)
```

```
##                  ME     RMSE      MAE       MPE     MAPE
## Test set 51.65676 1464.133 1069.966 -1.033851 10.70136
```

Answer:

lm_full has an MAE of 1058.757 while lm_bwd has an MAE of 1069.966.

**Q3(c) [1 points]** Recall from the in-class exercise that the MAE made by `lm_full` with log-transformation are `950.0841`. Compare with the MAE made by `lm_full` in Q3(b) without log-transformation. Answer:

Based on the MAE of lm_full from the in-class work and the lm_full I developed from the homework, I would say that the log transformation was able to help create a more accurate model. Because the log model has an MAE of 950.0841, it is more accurate than the non-log model, which had an MAE of 1058.757.

## Car Seat Sales Dataset [18 points]

The car seat sales dataset is the one we analyzed in HW1. It contains sales of child car seats at 400 different stores and the data is stored in `Carseats.csv`. It contains 9 variables, `Sales`, `CompPrice`, `Income`, `Advertising`, `Population`, `Price`, `ShelveLoc`, `Age` and `Urban`. We would like to build a linear regression model to predict `Sales` at a planned new store. The data description is as follows.

- `Sales`: Unit sales (in thousands) at each location
- `CompPrice`: Price charged by competitor at each location
- `Income`: Community income level (in thousands of dollars)
- `Advertising`: Local advertising budget for company at each location (in thousands of dollars)
- `Population`: Population size in region (in thousands)
- `Price`: Price company charges for car seats at each site
- `ShelveLoc`: A factor with levels Bad, Good and Medium indicating the quality of the shelving location for the car seats at each site
- `Age`: Average age of the local population
- `Urban`: A factor with levels No and Yes to indicate whether the store is in an urban or rural location

### Q4 [5 points] Data Partition

**Q4(a) [2 points]** Let's correctly read in the data in `Carseats.csv` and name it as `total_data`.

```
total_data <- read.csv( "/Users/TomTheIntern/Desktop/Mendoza/Mod 2/Advanced Stats/Hmwk 2/Carseats.csv",
```

**Q4(b) [3 points]** Let's partition the data in `total_data` into training **(80%)** and test data **(20%)** and store them as `R` objects `train_data` and `test_data` respectively. Use random seed **`set.seed(7)`**!

```
set.seed(7)
total_obs <- dim(total_data)[1]
train_data_indices <- sample(1:total_obs, 0.8*total_obs)
train_data <- total_data[train_data_indices,]
test_data <- total_data[-train_data_indices,]
```

### Q5 [8 points] Model Estimation and Backward Selection

**Q5(a) [2 points]** Fit a linear regression model of **original scale** Sales w.r.t. all 8 predictors using the **training data**, name it `lm_full`.

```
lm_full <- lm(Sales ~ . , data = train_data)
summary(lm_full)
```

```
##
## Call:
## lm(formula = Sales ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7665 -0.7358  0.0641  0.6279  3.2428
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      5.2159117  0.5931485   8.794  < 2e-16 ***
## CompPrice        0.0955360  0.0047278  20.207  < 2e-16 ***
## Income           0.0136980  0.0020307   6.745 7.48e-11 ***
## Advertising      0.1235861  0.0091724  13.474  < 2e-16 ***
## Population       0.0000621  0.0004160   0.149    0.881
## Price           -0.0963762  0.0030284 -31.825  < 2e-16 ***
## ShelveLocGood    4.8093429  0.1761479  27.303  < 2e-16 ***
## ShelveLocMedium  2.0786701  0.1414990  14.690  < 2e-16 ***
## Age             -0.0469240  0.0036214 -12.957  < 2e-16 ***
## UrbanYes         0.1290656  0.1291231   1.000    0.318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.024 on 310 degrees of freedom
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.8736
## F-statistic:   246 on 9 and 310 DF,  p-value: < 2.2e-16
```

**Q5(b)** [**2 points**] Perform backward selection for `lm_full` with **BIC** using the function `step()` and name the selected model `lm_bwd`. Make sure you use the correct **`k`** argument in the `step()` function.

```
lm_bwd <- step(lm_full, direction = 'backward', k = log(nrow(train_data)))
```

```
## Start:  AIC=62.8
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Urban
##
##               Df Sum of Sq     RSS    AIC
## - Population   1      0.02  325.18  57.06
## - Urban        1      1.05  326.21  58.06
## <none>                      325.16  62.80
## - Income       1     47.73  372.88 100.86
## - Age          1    176.10  501.26 195.53
## - Advertising  1    190.42  515.58 204.54
## - CompPrice    1    428.30  753.45 325.95
## - ShelveLoc    2    782.95 1108.11 443.61
## - Price        1   1062.33 1387.48 521.33
##
## Step:  AIC=57.06
## Sales ~ CompPrice + Income + Advertising + Price + ShelveLoc +
##     Age + Urban
##
##               Df Sum of Sq     RSS    AIC
## - Urban        1      1.03  326.21  52.29
## <none>                      325.18  57.06
## - Income       1     47.72  372.91  95.11
## - Age          1    178.36  503.55 191.22
## - Advertising  1    209.85  535.03 210.63
## - CompPrice    1    431.69  756.87 321.63
## - ShelveLoc    2    784.40 1109.58 438.27
## - Price        1   1062.53 1387.71 515.62
##
## Step:  AIC=52.29
## Sales ~ CompPrice + Income + Advertising + Price + ShelveLoc +
```

```
##     Age
##
##                Df Sum of Sq     RSS    AIC
## <none>                       326.21  52.29
## - Income       1     49.19  375.40  91.47
## - Age          1    179.22  505.43 186.64
## - Advertising  1    212.03  538.24 206.77
## - CompPrice    1    436.86  763.06 318.46
## - ShelveLoc    2    787.27 1113.48 433.63
## - Price        1   1068.20 1394.41 511.39
```

```r
summary(lm_bwd)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Price +
##     ShelveLoc + Age, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7467 -0.7008  0.0113  0.6360  3.2837
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.313287   0.558189   9.519  < 2e-16 ***
## CompPrice        0.095797   0.004687  20.441  < 2e-16 ***
## Income           0.013848   0.002019   6.859 3.73e-11 ***
## Advertising      0.124453   0.008739  14.241  < 2e-16 ***
## Price           -0.096512   0.003019 -31.964  < 2e-16 ***
## ShelveLocGood    4.789134   0.174614  27.427  < 2e-16 ***
## ShelveLocMedium  2.061061   0.140170  14.704  < 2e-16 ***
## Age             -0.047075   0.003596 -13.093  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.023 on 312 degrees of freedom
## Multiple R-squared:  0.8768, Adjusted R-squared:  0.874
## F-statistic: 317.1 on 7 and 312 DF,  p-value: < 2.2e-16
```

**Q5(c) [2 points]** Examine the printout of the `step()` function in Q5(b), what is the first predictor removed in the backward selection?

Answer:

Population is the first predictor removed by the backward selection process.

**Q5(d) [2 points]** Examine the selected model in `lm_bwd`, list all the predictors that are eliminated during the backward selection process.

```r
summary(lm_bwd)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Price +
```

```
##     ShelveLoc + Age, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7467 -0.7008  0.0113  0.6360  3.2837
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.313287   0.558189   9.519  < 2e-16 ***
## CompPrice        0.095797   0.004687  20.441  < 2e-16 ***
## Income           0.013848   0.002019   6.859 3.73e-11 ***
## Advertising      0.124453   0.008739  14.241  < 2e-16 ***
## Price           -0.096512   0.003019 -31.964  < 2e-16 ***
## ShelveLocGood    4.789134   0.174614  27.427  < 2e-16 ***
## ShelveLocMedium  2.061061   0.140170  14.704  < 2e-16 ***
## Age             -0.047075   0.003596 -13.093  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.023 on 312 degrees of freedom
## Multiple R-squared:  0.8768, Adjusted R-squared:  0.874
## F-statistic: 317.1 on 7 and 312 DF,  p-value: < 2.2e-16
```

Answer: The backward selection process removed the population predictor and the UrbanYes predictor.

### Q6 [5 points] Model Evaluation (Prediction)

**Q6(a)** [**2 points**] Use `lm_full` and `lm_bwd` to generate predictions for Sales on the test data and store the prediction in `lm_full_pred` and `lm_bwd_pred` respectively.

```r
lm_full_pred <- predict(lm_full, newdata = test_data)
lm_bwd_pred <- predict(lm_bwd, newdata = test_data)
```

**Q6(b)** [**2 points**] Use the `R` package `forecast` to evaluate the prediction performance of `lm_full_pred` and `lm_bwd_pred`. What are the MAE for `lm_full` and `lm_bwd`?

```r
library(forecast)
accuracy(lm_full_pred, test_data$Sales)
```

```
##                  ME     RMSE       MAE       MPE     MAPE
## Test set -0.1258756 1.036768 0.8582053 -4.271043 13.92952
```

```r
accuracy(lm_bwd_pred, test_data$Sales)
```

```
##                 ME     RMSE       MAE       MPE     MAPE
## Test set -0.130388 1.038782 0.8597975 -4.260752 13.90842
```

Answer: lm_full has an MAE of 0.8582053 while lm_bwd has an MAE of 0.8597975.

**Q6(c)** [**1 points**] Which statistical model do you prefer, `lm_full` or `lm_bwd`? Give reasons.

Answer:

There are two ways to look at which model is better. The first is accuracy: which model has the higher $R^2$ value and which model has the lower MAE?

lm_full had the higher $R^2$ value at .8772, beating out lm_bwd narrowly (0.8768). lm_full also had a slightly lower MAE (0.8582053) thabn lm_bwd (0.8597975).

But we also want our model to have a parsimonious structure, or we want our model to use fewer predictors when possible. Predictors should only be included when they have great predictive power. So, because lm_bwd has fewer predictors, it seems like the better model. We can test this with a BIC test.

```r
BIC(lm_full)
```

```
## [1] 976.6896
```

```r
BIC(lm_bwd)
```

```
## [1] 966.1835
```

Based on the BIC test, we can select the lm_bwd model.