Analysis

In the section,we first generate training and test sets of data, randomization seed establishment, parameter choosing via cross-validation and model fitting for each of the methods in order to study the accuracy of fit for each model

Exploratory Data Analysis

To look at the basic statistical distributions and summaries of all the variables, the folder `Data` contains all the `.Rdata` and `.txt` files created from the R scripts. Furthermore, relevant plots such as histograms, boxplots, barplots, and conditional boxplots are created that can be accessed in the `Images` folder. Finally, the ANOVA tests were performed using the `aov()` function.

Pre-modeling Data Processing

Two major processing steps are done before the models are created:

1. Converting the factors into dummy variables:

   Transforming the categorizal variables into dummy variables in order to use the built-in function *glmnet* to fit the data.

2. Mean centering and standardization

   By doing so, we make sure that the coefficients would function properly and in an equivalent manner

Ordinary Least Squares Regression (OLS)

```
## [1] 0.04468304
```

We first fit the Ordinary Least Squares Regression model to the data set, with no parameters to be tuned, using hte `lm()` function. The mean square error of the OLS model on the test set is shown above: 0.04468304. We use this as a reference point when comparing with other prediction models.

Ridge Regression (RR)

We fit the Ridge Regression model to the training set. Firstly, we need to train the model for the best parameter λ via cross-validation (`cv.glmet`). Using `predict()` and calling the library package `*glment*` to perform the regression, we get the following:

```
## [1] 0.01
```

```
## [1] 0.001027031
```

From the graph above, we choose the best λ given this traning set, which is 0.01 and applied it to the entire data set

The resulting mean square error for the same test set is given above. The #####0.04611172 does show an improvment compared to the OLS Regression's 0.04468304.

Lasso Regression (LR)

We fit the Lasso Regression model to the training set using the same method, but the difference is that the alpha variable in the function `cv.glmnet()` is changed from 0 to 1.

```
## [1] 0.01
```

```
## [1] 0.05406075
```

From the graph above, we choose the best λ given this traning set, which is also 0.01 and applied it to the entire data set.

The resulting mean square error for the same test set is given above. The 0.05406075 does show a slight deterioration compared to the OLS Regression's 0.04468304.

Principal Components Regression (PCR)

We fit the Principal Components Regression to the training set by installing the library package `pls` and using the function `pcr()` to run the cross-validation using the opitimal lambda.

From the validation plot (displayed in the next section), we may find that the lowest cross-validation error occurs when $M = 11$ component are used. Therefore, we compute the test MSE:

```
## [1] 0.05143916
```

The resulting mean square error for the same test set is given above. The 0.05198999 is slightly higher than the the OLS Regression's 0.04468304, but lower than Lasso. We also fit the model with the best choice of M, the number of components to the entire data set for future purpose.

Partial Least Squares Regression (PLSR)

We fit the Partial Least Squares Regression to the training set by installing the same library package `pcr` and using the function `plsr()` to run the cross-validation using the opitimal lambda or tuning parameter.

From the validation plot (displayed in the next section), we may find that the lowest cross-validation error occurs when $M = 4$ component are used. Therefore, we compute the test MSE:

```
## [1] 0.05173802
```

The resulting mean square error for the same test set is given above: 0.05173802 This too is larger than the OLS Regression's 0.04468304. We also fit the model with the best choice of M, the number of components to the entire data set for future purpose.

So, apart from the OLS Regression's MSE, the PLSR regression's MSE is the smallest from the 4 other methods.