# Regression Analysis Project 2

Gabriel Afriyie

23/03/2020

## Introduction of Dataset and Purpose of Project

We seek to perform multiple regression analysis on 'Salaries' dataset. This dataset which consists of 397 observations on 6 variables shows the nine-month academic salary for Assistant Professors, Associate Professors and Professors in a college in the United States. The dataset comprises the follwoing 6 variables.

- rank : a factor with levels *AssocProf, AsstProf, Prof.*
- discipline : a factor with levels *A ("theoretical departments") or B ("applied departments").*
- yrs.since.phd : years since PhD was obtained.
- yrs.service : years of service.
- sex :a factor with levels *Female, Male.*
- salary : nine-month salary, in dollars.

The data were collected as part of the on-going effort of the college's administration to monitor salary differences between male and female faculty members. In this project, we seek to perform multiple linear regression to predict the years of service of faculty members based on the other variables.

First, we explore the data we wish to use for this analysis. The *head()* function gives us the first six rows of the data.

```
library(carData)
attach(Salaries)
head(Salaries)
```

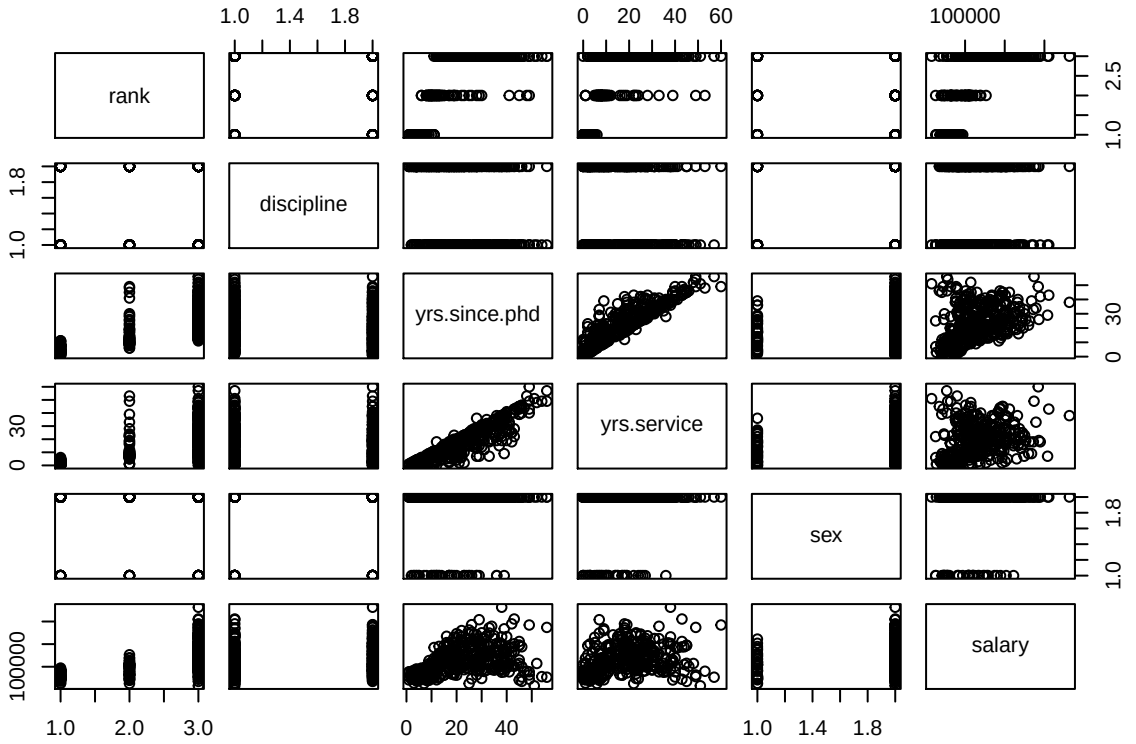
```
##        rank discipline yrs.since.phd yrs.service  sex salary
## 1      Prof          B            19          18 Male 139750
## 2      Prof          B            20          16 Male 173200
## 3  AsstProf          B             4           3 Male  79750
## 4      Prof          B            45          39 Male 115000
## 5      Prof          B            40          41 Male 141500
## 6 AssocProf          B             6           6 Male  97000
```

We now explore numerical and graphical summaries of all the variables.

```
str(Salaries)
```

```
## 'data.frame':    397 obs. of  6 variables:
##  $ rank         : Factor w/ 3 levels "AsstProf","AssocProf",..: 3 3 1 3 3 2 3 3 3 3 ...
##  $ discipline   : Factor w/ 2 levels "A","B": 2 2 2 2 2 2 2 2 2 2 ...
##  $ yrs.since.phd: int  19 20 4 45 40 6 30 45 21 18 ...
##  $ yrs.service  : int  18 16 3 39 41 6 23 45 20 18 ...
##  $ sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 1 ...
##  $ salary       : int  139750 173200 79750 115000 141500 97000 175000 147765 119250 129000 ...
```

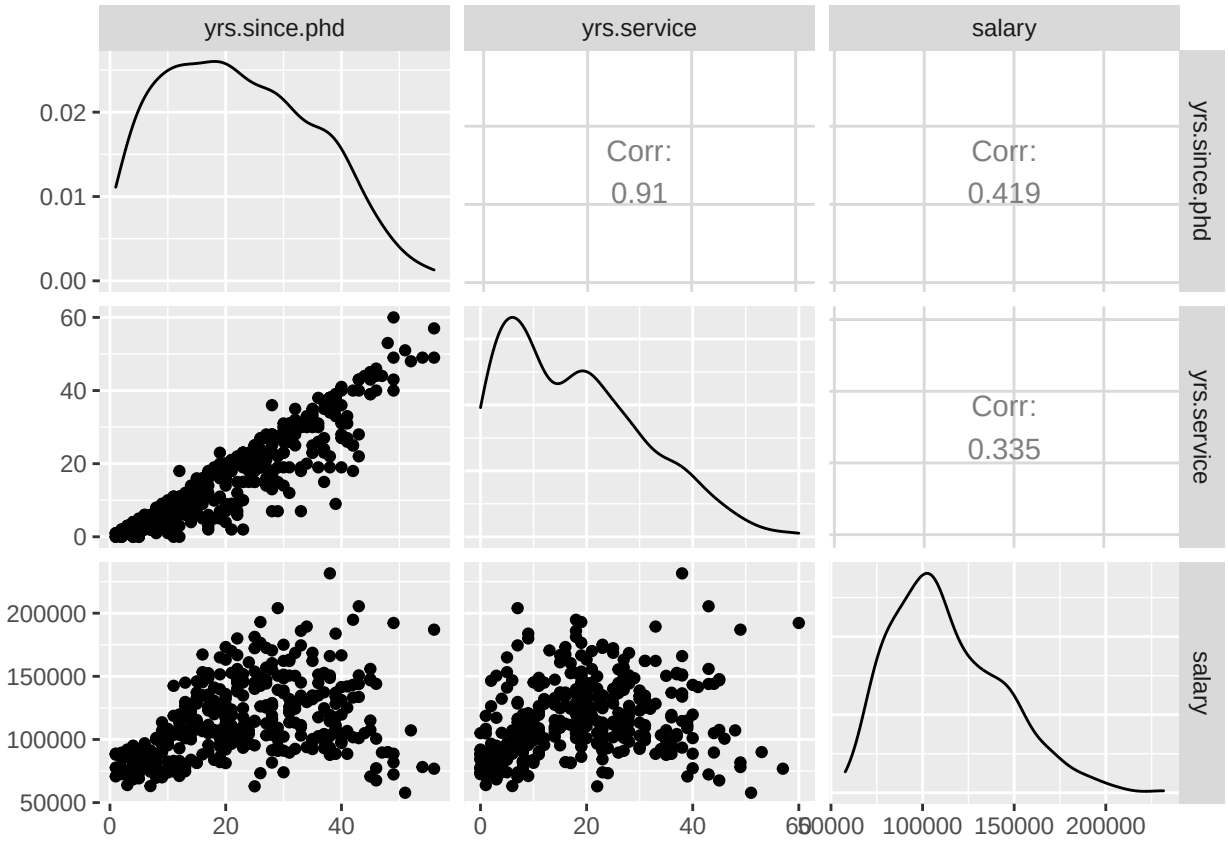
```
set.seed(100)
data("Salaries")
x <- Salaries[sample(1:nrow(Salaries)),]
plot(x)
```




```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(Salaries[,c(3,4,6)])
```




```
summary(x)
```

```
##         rank     discipline yrs.since.phd    yrs.service        sex
##  AsstProf : 67   A:181      Min.   : 1.00   Min.   : 0.00   Female: 39
##  AssocProf: 64   B:216      1st Qu.:12.00   1st Qu.: 7.00   Male  :358
##  Prof     :266              Median :21.00   Median :16.00
##                             Mean   :22.31   Mean   :17.61
##                             3rd Qu.:32.00   3rd Qu.:27.00
##                             Max.   :56.00   Max.   :60.00
##      salary
##  Min.   : 57800
##  1st Qu.: 91000
##  Median :107300
##  Mean   :113706
##  3rd Qu.:134185
##  Max.   :231545
```

The graphs show a positive and high linear correlation between *yrs.since.phd* and *yrs.service*. The salaries of the faculty members seem to increase with their years of service and years since PhD to a point and then show a decrease.

## Data Analysis

We first split the dataset into two sets: *train set* and *test set*. We fit our model on the *train set* and test our model on the *test set*.

```
train <- x[1:300,]
test <- x[301:397,]
```

To determine the best predictive model for years of service, we fit a multiple regression model with all the 5 predictors.

$$yrs.service = \beta_0 + \beta_1(rank) + \beta_2(discipline) + \beta_2(yrs.since.phd) + \beta_3(yrs.service) + \beta_4(sex) + \beta_5(salary) + \varepsilon$$

```
model1 <- lm(yrs.service ~ .,
             data = train)
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = yrs.service ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.245  -2.594   1.169   3.744  17.802
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.667e+00  1.573e+00  -1.695   0.0911 .
## rankAssocProf  2.032e-01  1.174e+00   0.173   0.8627
## rankProf      -1.360e+00  1.351e+00  -1.006   0.3152
## disciplineB    1.326e+00  6.681e-01   1.985   0.0480 *
## yrs.since.phd  9.824e-01  3.522e-02  27.896   <2e-16 ***
## sexMale        1.239e+00  1.101e+00   1.125   0.2615
## salary        -2.328e-05  1.336e-05  -1.743   0.0824 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.436 on 293 degrees of freedom
## Multiple R-squared:  0.824,  Adjusted R-squared:  0.8204
## F-statistic: 228.7 on 6 and 293 DF,  p-value: < 2.2e-16
```

## Analysis of Results

The intercept, which is $-2.667$ is the estimated years of service when all other explanatory variables take on the value 0. This value is not feasible because it should not be negative. The estimated expected years of service of Associate Professors is 0.2032 higher than that of Assistant Professors, with all other variables held constant. The estimated expected change in years of service with a unit increase in salary, with all other variables held constant is $-2.328e-05$. $R^2 = 82.4\%$ means that 82.4% of the variability in yrs.service is explained by the model. Adjusted $R^2 = 82\%$. We test the hypothesis:

$H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$ versus $H_1 : \exists i$ such that $\beta_i \neq 0, i = 1, \ldots, 5$

At $\alpha = 0.05$, we found only discipline and years since PhD to be significant.We reject $H_0$ and then fit a simpler model by omitting sex.

### Choosing the Best Model.

```
model2 <- lm(yrs.service ~ .,
             data = train[,-5])
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = yrs.service ~ ., data = train[, -5])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.249  -2.660   1.367   3.795  17.763
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.739e+00  1.340e+00  -1.297   0.1955
## rankAssocProf  1.987e-01  1.174e+00   0.169   0.8658
## rankProf      -1.309e+00  1.351e+00  -0.969   0.3333
## disciplineB    1.327e+00  6.684e-01   1.985   0.0480 *
## yrs.since.phd  9.851e-01  3.515e-02  28.024   <2e-16 ***
## salary        -2.242e-05  1.334e-05  -1.681   0.0939 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.438 on 294 degrees of freedom
## Multiple R-squared:  0.8233, Adjusted R-squared:  0.8203
## F-statistic: 273.9 on 5 and 294 DF,  p-value: < 2.2e-16
```

We get similar results from this model. We the fit an even simpler model without sex and rank.

```
model3 <- lm(yrs.service ~.,
             data = train[,-c(1,5)])
```

```
summary(model3)
```

```
##
## Call:
## lm(formula = yrs.service ~ ., data = train[, -c(1, 5)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.656  -2.557   1.384   3.570  18.577
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.084e+00  1.219e+00  -0.889  0.37461
## disciplineB    1.420e+00  6.648e-01   2.135  0.03356 *
## yrs.since.phd  9.611e-01  2.859e-02  33.614  < 2e-16 ***
## salary        -3.124e-05  1.152e-05  -2.711  0.00709 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.438 on 296 degrees of freedom
```

```
## Multiple R-squared:  0.8221, Adjusted R-squared:  0.8203
## F-statistic: 455.9 on 3 and 296 DF,  p-value: < 2.2e-16
```

The same results from the first and second model are obtained in the third. The anova table below also confirms that a simpler model gives the same results. *model1* is not better than *model2* and *model2* is not better than *model3*.

```
anova(model1,model2,model3)
```

```
## Analysis of Variance Table
##
## Model 1: yrs.service ~ rank + discipline + yrs.since.phd + sex + salary
## Model 2: yrs.service ~ rank + discipline + yrs.since.phd + salary
## Model 3: yrs.service ~ discipline + yrs.since.phd + salary
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    293 8657.9
## 2    294 8695.3 -1   -37.407 1.2659 0.2615
## 3    296 8754.6 -2   -59.273 1.0029 0.3681
```

We can also select a model by AIC in a Stepwise Algorithm. We use the *step()* function.

```
s1 <- step(lm(yrs.service ~ . , data=train))
```

```
## Start:  AIC=1022.73
## yrs.service ~ rank + discipline + yrs.since.phd + sex + salary
##
##                 Df Sum of Sq   RSS    AIC
## - rank           2      63.6  8721 1020.9
## - sex            1      37.4  8695 1022.0
## <none>                       8658 1022.7
## - salary         1      89.8  8748 1023.8
## - discipline     1     116.5  8774 1024.7
## - yrs.since.phd  1   22995.1 31653 1409.6
##
## Step:  AIC=1020.93
## yrs.service ~ discipline + yrs.since.phd + sex + salary
##
##                 Df Sum of Sq   RSS    AIC
## - sex            1        33  8755 1020.1
## <none>                       8721 1020.9
## - discipline     1       135  8857 1023.5
## - salary         1       231  8953 1026.8
## - yrs.since.phd  1     32776 41497 1486.9
##
## Step:  AIC=1020.07
## yrs.service ~ discipline + yrs.since.phd + salary
##
##                 Df Sum of Sq   RSS    AIC
## <none>                       8755 1020.1
## - discipline     1       135  8889 1022.6
## - salary         1       217  8972 1025.4
## - yrs.since.phd  1     33418 42173 1489.7
```

```
summary(s1)
```

```
##
## Call:
```

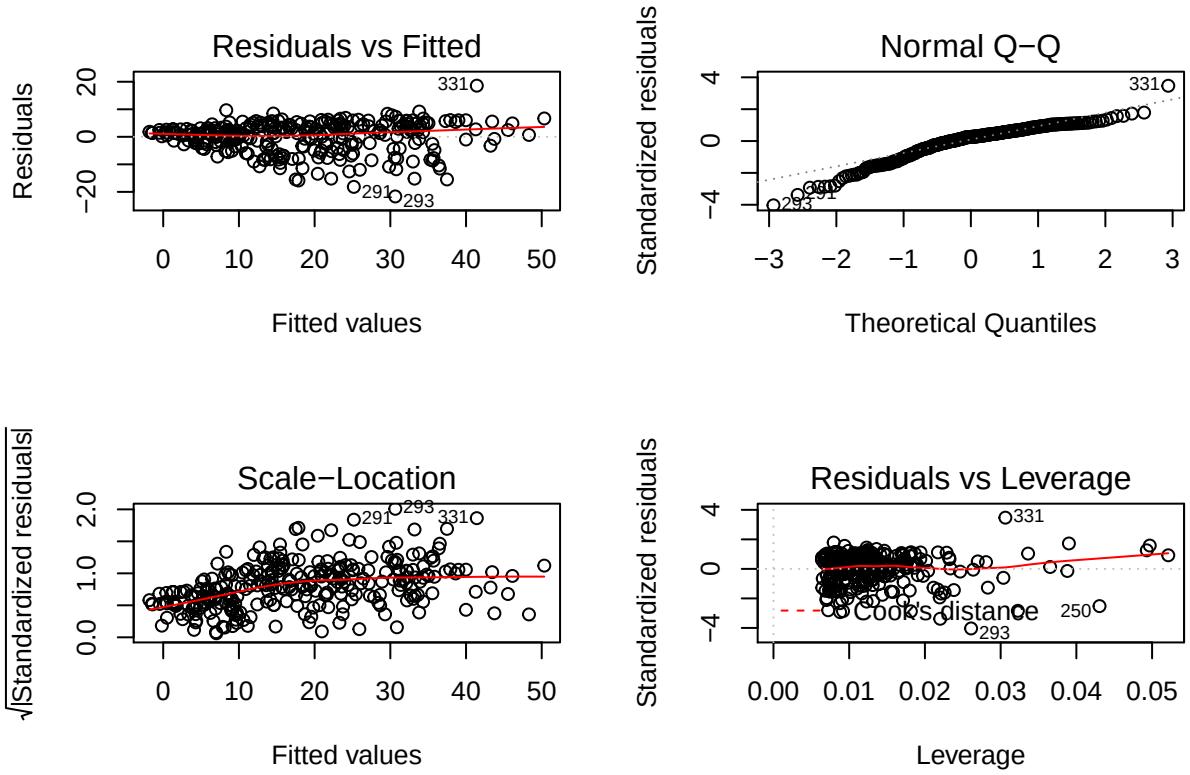
```
## lm(formula = yrs.service ~ discipline + yrs.since.phd + salary,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.656  -2.557   1.384   3.570  18.577
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.084e+00  1.219e+00  -0.889  0.37461
## disciplineB    1.420e+00  6.648e-01   2.135  0.03356 *
## yrs.since.phd  9.611e-01  2.859e-02  33.614  < 2e-16 ***
## salary        -3.124e-05  1.152e-05  -2.711  0.00709 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.438 on 296 degrees of freedom
## Multiple R-squared:  0.8221, Adjusted R-squared:  0.8203
## F-statistic: 455.9 on 3 and 296 DF,  p-value: < 2.2e-16
```

This method chooses the model with the least AIC. From this we obtain our best model.

## Model Diagnostics

After choosing the model, we check the adequacy of the fitted model by testing that assumptions of normality and constant variance have been satisfied.

```
par(mfrow=c(2,2))
plot(model3)
```




These plots show that the model violates the assumptions of normality and constant variance. We recommend log or Box Cox transformations to enhance the adequacy of the model.

## Making predictions

We test the quality of the chosen model by making predictions, using our *test set*. We also calculate the mean squared error (MSE) to check the accuracy of the model.

```
pred.yrs.service =predict(model3,test)
MSE = sum((test$yrs.service-pred.yrs.service)^2)/(length(test))
MSE
```

```
## [1] 408.7943
```

Unfortunately, the $MSE = 408.7943$ is extremely large. We introduce another regression method that forces some coefficients in the model to go to 0.

## Least Absolute Shrinkage and Selection Operator (LASSO)

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of muticollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients

can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) doesn't result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge. In Lasso, we seek to minimize:

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Which is equivalent to minimizing the sum of squares with constraint $\sum |\beta_j| \leq s$. Some of the $\beta$s are shrunk to exactly zero, resulting in a regression model that's easier to interpret. A tuning parameter, $\lambda$ controls the strength of the L1 penalty. $\lambda$ is basically the amount of shrinkage:

- When $\lambda = 0$, no parameters are eliminated. The estimate is equal to the one found with linear regression.
- As $\lambda$ increases, more and more coefficients are set to zero and eliminated (theoretically, when $\lambda = \infty$, all coefficients are eliminated).
- As $\lambda$ increases, bias increases.
- As $\lambda$ decreases, variance increases.

If an intercept is included in the model, it is usually left unchanged.

We use the *glmnet()* function for Lasso regression in R. Firstly, we create a design matrix on the *train set*

```
library(glmnet)
```

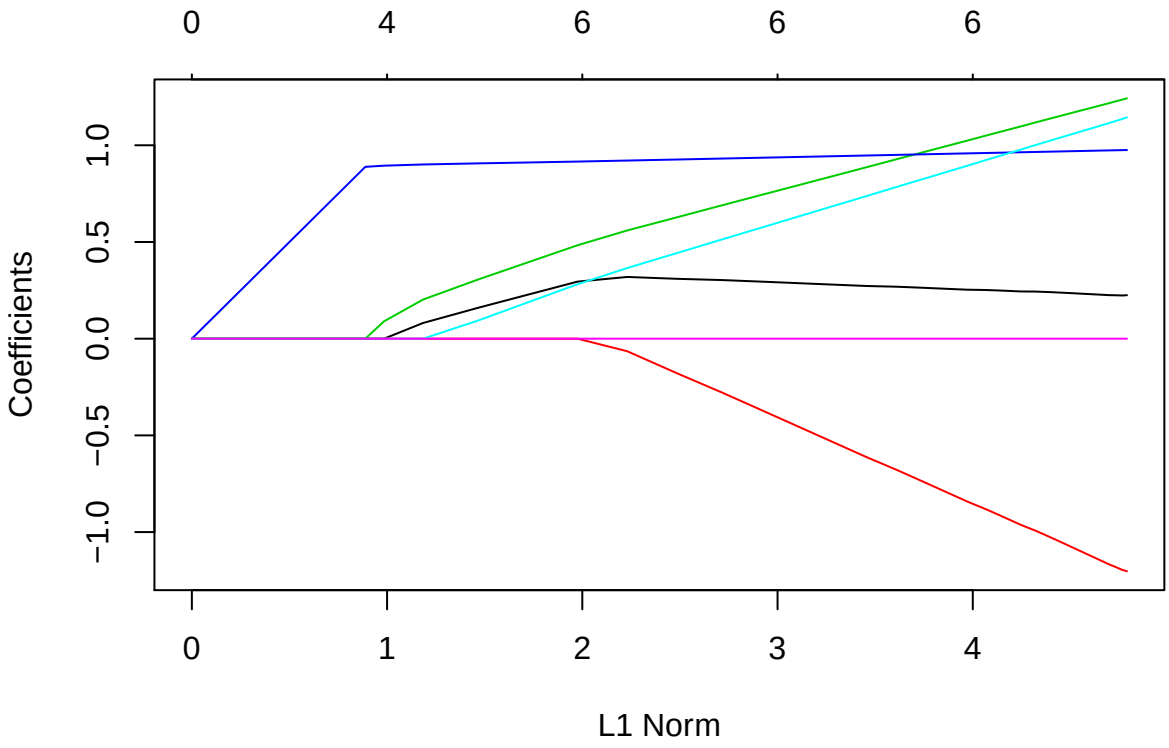
```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-2
```

```
X = model.matrix(yrs.service~.,data = train)
```

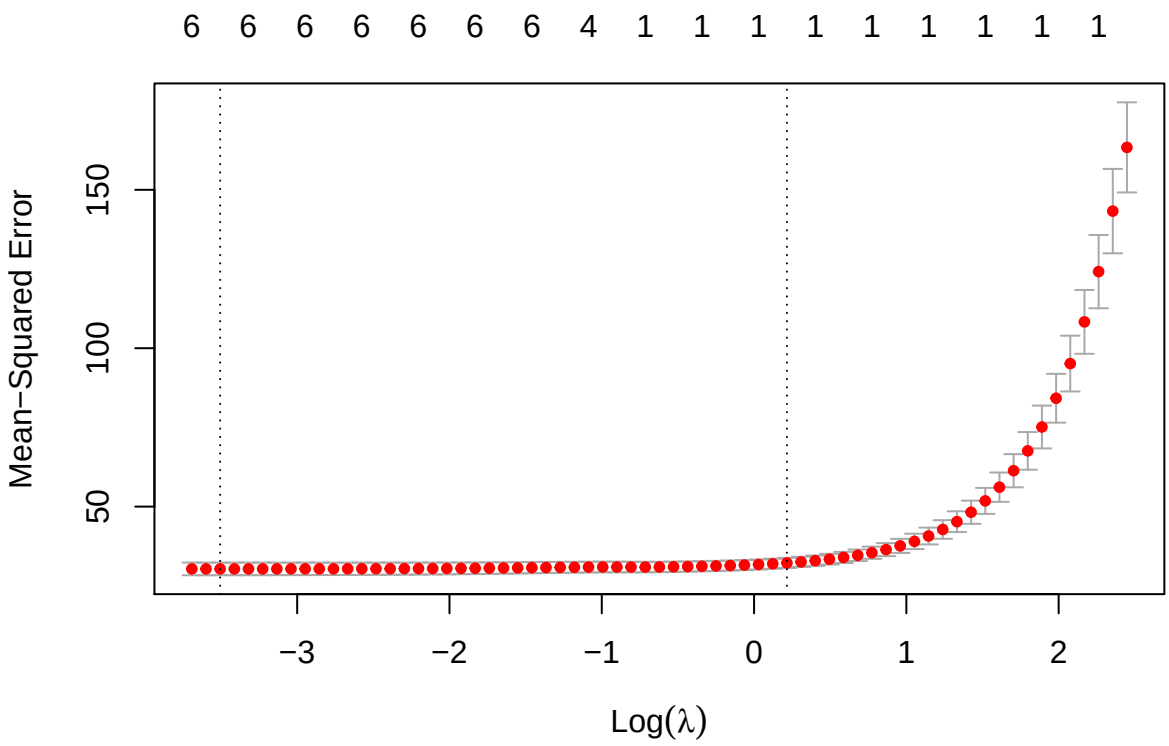
We also fit and plot the Lasso model.

```
Y = train$yrs.service
lasso.model = glmnet(X,Y,alpha = 1)
plot(lasso.model)
```




The graph shows the coefficients against the L1 norm. A decrease in the L1 norm results in the coefficients shrinking to 0. This plot can be made with coefficients against $\lambda$. We perform a k-fold cross-validation on the Lasso model to choose the $\lambda$ that produces the best model.

```
CV.error = cv.glmnet(X,Y,alpha=1)
plot(CV.error)
```




```
best.lambda = CV.error$lambda.min
coef(CV.error)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept)   -0.7719420
## (Intercept)    .
## rankAssocProf  .
## rankProf       .
## disciplineB    .
## yrs.since.phd  0.8201566
## sexMale        .
## salary         .
```

Finally, we obtain a model with the intercept and one coefficient (yrs.since.phd).

$$\hat{Y} = -0.7719420 + 0.8201566$$

### Making Predictions with Lasso Model.

We test the accuracy of our Lasso model by making some predictions and calculating the mean squared error.

```
DesignMatTest = model.matrix(yrs.service~.,data = test)
lasso.pred= predict(lasso.model,s=best.lambda,DesignMatTest)
MSE2 = mean((test$yrs.service-lasso.pred)^2)
MSE2
```

```
## [1] 25.9517
```

We observe a smaller $MSE = 25.9517$ as compared to that of the linear regression model.

## Conclusion

We began by fitting a multiple linear regression model of years of service against all other variables in the Salaries data. Reducing the number of variables in the model showed that simpler models could also provide the same results. Making predictions with the model obtained, produced a high MSE. Lasso was used to force some of the coefficients to go to 0. Finally, we were only left with one predictor variable which is yr.since.phd. This is a very good predictor because most faculty members start teaching right after obtaining their PhD. The MSE obtained after making predictions with the Lasso model is relatively lower. This makes the Lasso model desirable in our case. It is most appropriate to choose simpler models for our analysis if they would give us better results.