Problem Set 8

Jadyn Gonzalez

10/25/2019

What Predicts Bike Share Usage?

- a) We downloaded the Bike Sharing folder from the UCI repository. Let's set our working directory and import the day.csv file. Let's also take a look at the data.

```
setwd("~/Documents/MGSC310/Bike-Sharing-Dataset")
Bike_DF = read.csv("day.csv")
summary(Bike_DF)
```

```
##     instant          dteday        season          yr
##  Min.   :  1.0   2011-01-01:  1   Min.   :1.000   Min.   :0.0000
##  1st Qu.:183.5   2011-01-02:  1   1st Qu.:2.000   1st Qu.:0.0000
##  Median :366.0   2011-01-03:  1   Median :3.000   Median :1.0000
##  Mean   :366.0   2011-01-04:  1   Mean   :2.497   Mean   :0.5007
##  3rd Qu.:548.5   2011-01-05:  1   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :731.0   2011-01-06:  1   Max.   :4.000   Max.   :1.0000
##                  (Other)   :725
##       mnth          holiday          weekday        workingday
##  Min.   : 1.00   Min.   :0.00000   Min.   :0.000   Min.   :0.000
##  1st Qu.: 4.00   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000
##  Median : 7.00   Median :0.00000   Median :3.000   Median :1.000
##  Mean   : 6.52   Mean   :0.02873   Mean   :2.997   Mean   :0.684
##  3rd Qu.:10.00   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000
##  Max.   :12.00   Max.   :1.00000   Max.   :6.000   Max.   :1.000
##
##    weathersit         temp              atemp              hum
##  Min.   :1.000   Min.   :0.05913   Min.   :0.07907   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200
##  Median :1.000   Median :0.49833   Median :0.48673   Median :0.6267
##  Mean   :1.395   Mean   :0.49538   Mean   :0.47435   Mean   :0.6279
##  3rd Qu.:2.000   3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302
##  Max.   :3.000   Max.   :0.86167   Max.   :0.84090   Max.   :0.9725
##
##    windspeed          casual          registered        cnt
##  Min.   :0.02239   Min.   :   2.0   Min.   :  20   Min.   :  22
##  1st Qu.:0.13495   1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152
##  Median :0.18097   Median : 713.0   Median :3662   Median :4548
##  Mean   :0.19049   Mean   : 848.2   Mean   :3656   Mean   :4504
##  3rd Qu.:0.23321   3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956
##  Max.   :0.50746   Max.   :3410.0   Max.   :6946   Max.   :8714
##
```

```
str(Bike_DF)
```

```
## 'data.frame':    731 obs. of  16 variables:
```

```
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
##  $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
##  $ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
##  $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
##  $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

- b) Let's do some basic data cleaning and convert factor variables.

```r
table(is.na(Bike_DF))
```

```
##
## FALSE
## 11696
```

```r
# no NaN looks good

# Taking care of the season attribute
Bike_DF$season = factor(format(Bike_DF$season, format = "%A"),
                          levels = c("1", "2","3","4"),
                          labels = c("Spring", "Summer", "Fall", "Winter"))
# Next convert holiday into a factor
Bike_DF$holiday = factor(format(Bike_DF$holiday, format = "%A"),
                           levels = c("0", "1"),
                           labels = c("No", "Yes"))
# Convert weatersit to a factor
Bike_DF$weathersit = factor(format(Bike_DF$weathersit, format = "%A"),
                              levels = c("1", "2","3","4"),
                              labels = c("Clear", "Cloudy", "Light Storms", "Heavy Storms"))

# Fix the yr variable
Bike_DF$yr = factor(format(Bike_DF$yr, format = "%A"),
                      levels = c("0", "1"),
                      labels = c("2011","2012"))
# Convert workingday
Bike_DF$workingday = factor(format(Bike_DF$workingday, format = "%A"),
                    levels = c("0", "1"),
                    labels = c("No","Yes"))

# Convert weekday
Bike_DF$weekday = factor(format(Bike_DF$weekday, format = "%A"),
                         levels = c("0","1","2","3","4","5","6"),
                         labels = c("0","1","2","3","4","5","6"))
```

- c) Running sapply() to make sure all our factor variables are actually of factor type.

```r
sapply(Bike_DF, is.factor)
```

```
##    instant     dteday     season         yr       mnth    holiday 
##      FALSE       TRUE       TRUE       TRUE      FALSE       TRUE 
##    weekday workingday weathersit       temp      atemp        hum 
##       TRUE       TRUE       TRUE      FALSE      FALSE      FALSE 
##  windspeed     casual registered        cnt 
##      FALSE      FALSE      FALSE      FALSE
```

```r
str(Bike_DF)
```

```
## 'data.frame':    731 obs. of  16 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ season    : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : Factor w/ 2 levels "2011","2012": 1 1 1 1 1 1 1 1 1 1 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ weekday   : Factor w/ 7 levels "0","1","2","3",..: 7 1 2 3 4 5 6 7 1 2 ...
##  $ workingday: Factor w/ 2 levels "No","Yes": 1 1 2 2 2 2 2 1 1 2 ...
##  $ weathersit: Factor w/ 4 levels "Clear","Cloudy",..: 2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
##  $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
##  $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

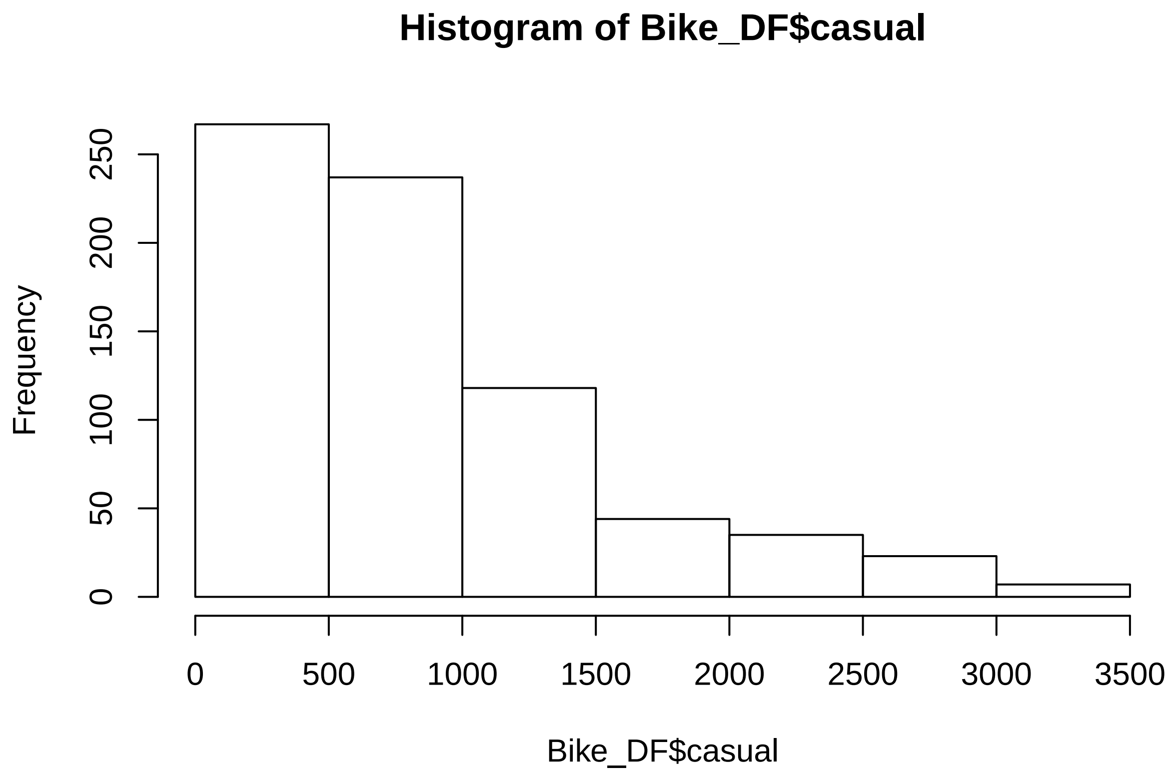
And it looks like we are good to go.

- d) Let's add some feature transformation

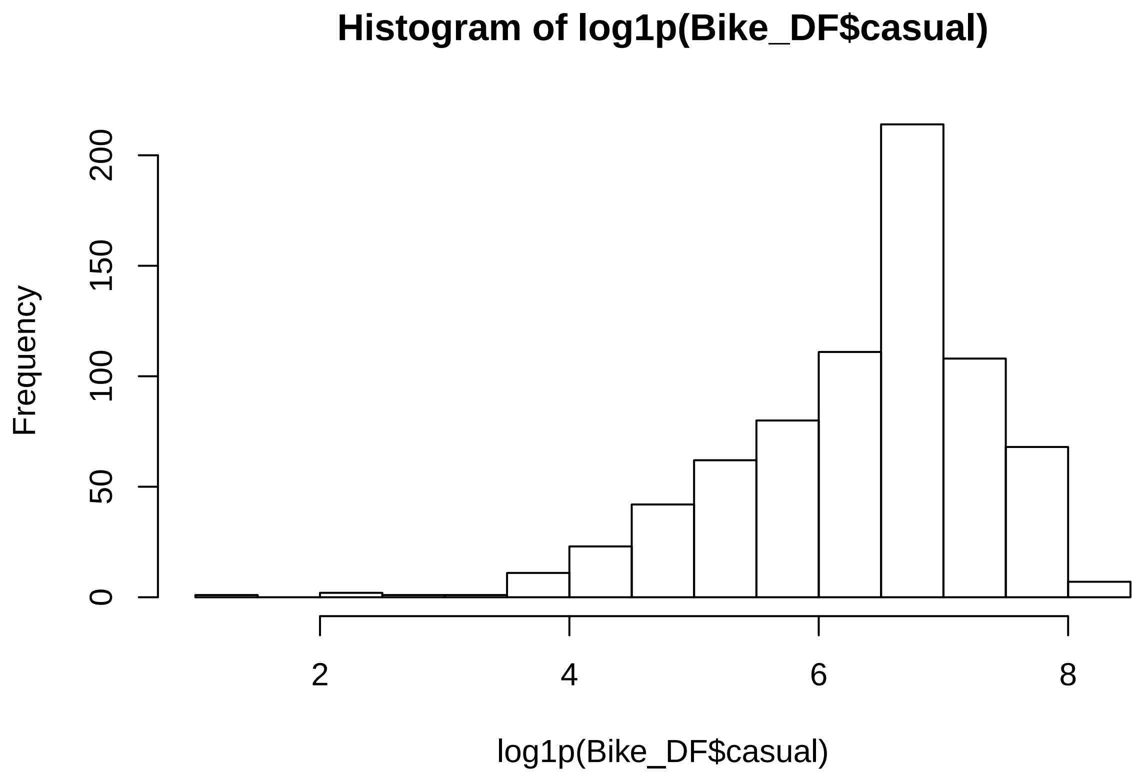
```r
# temp: need to multiply by 41
# atemp: (feeling) need to multiply by 50
# hum: need to multiply by 100
# windspeed: need to multiply by 67
Bike_DF$actual_temp = Bike_DF$temp*41 # in celsius
Bike_DF$feel_temp = Bike_DF$atemp*50 # in celsius
Bike_DF$actual_windspeed = Bike_DF$windspeed*67
Bike_DF$actual_humidity = Bike_DF$hum*100 # percent
```

By looking over the data it looks like casual and registered can use some transformations, lets look at casual first.

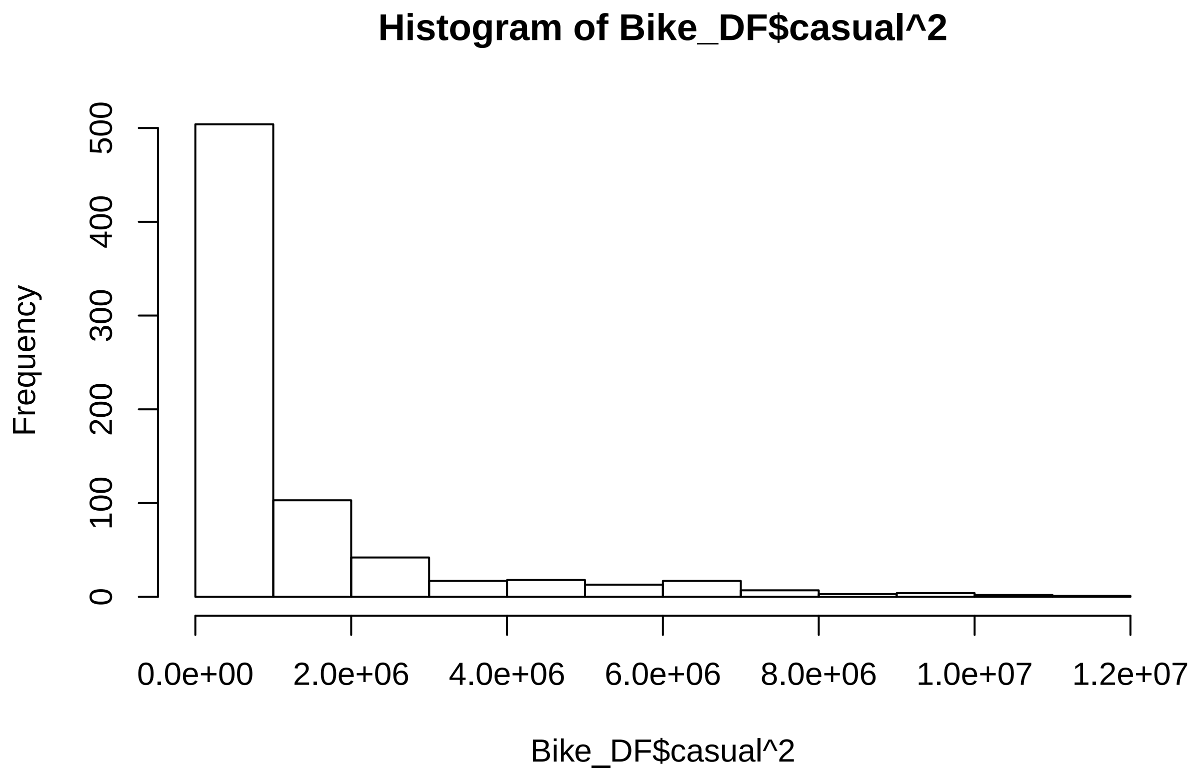
```r
hist(Bike_DF$casual)
```




```r
hist(log1p(Bike_DF$casual))
```

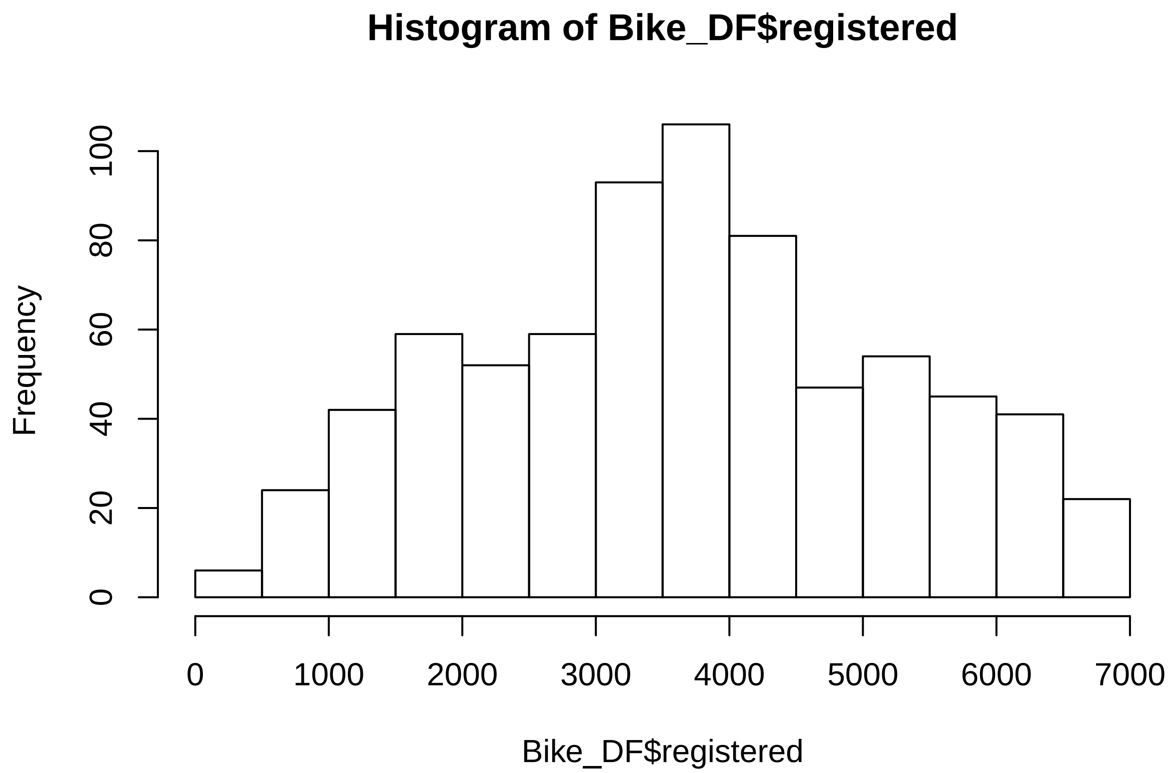



```
hist(Bike_DF$casual^2)
```

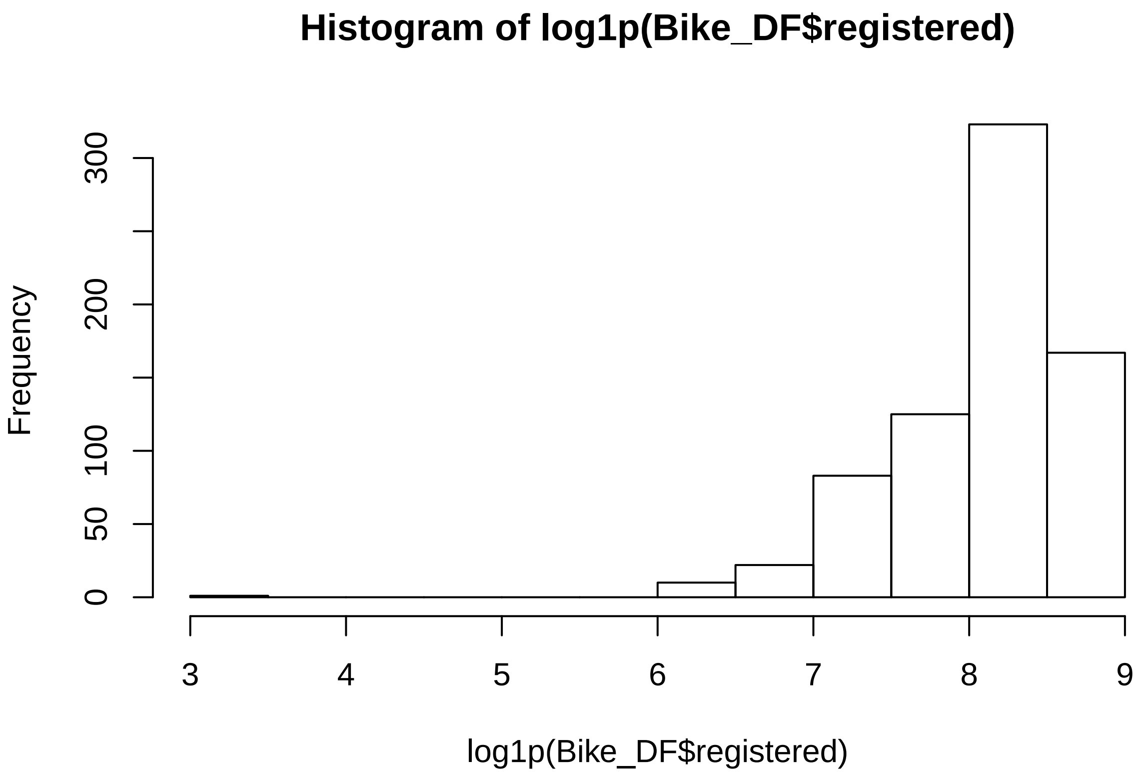


It appears casual benefits from the log transformation, but we'll add the square to our data anyway. Now let's look at registered.

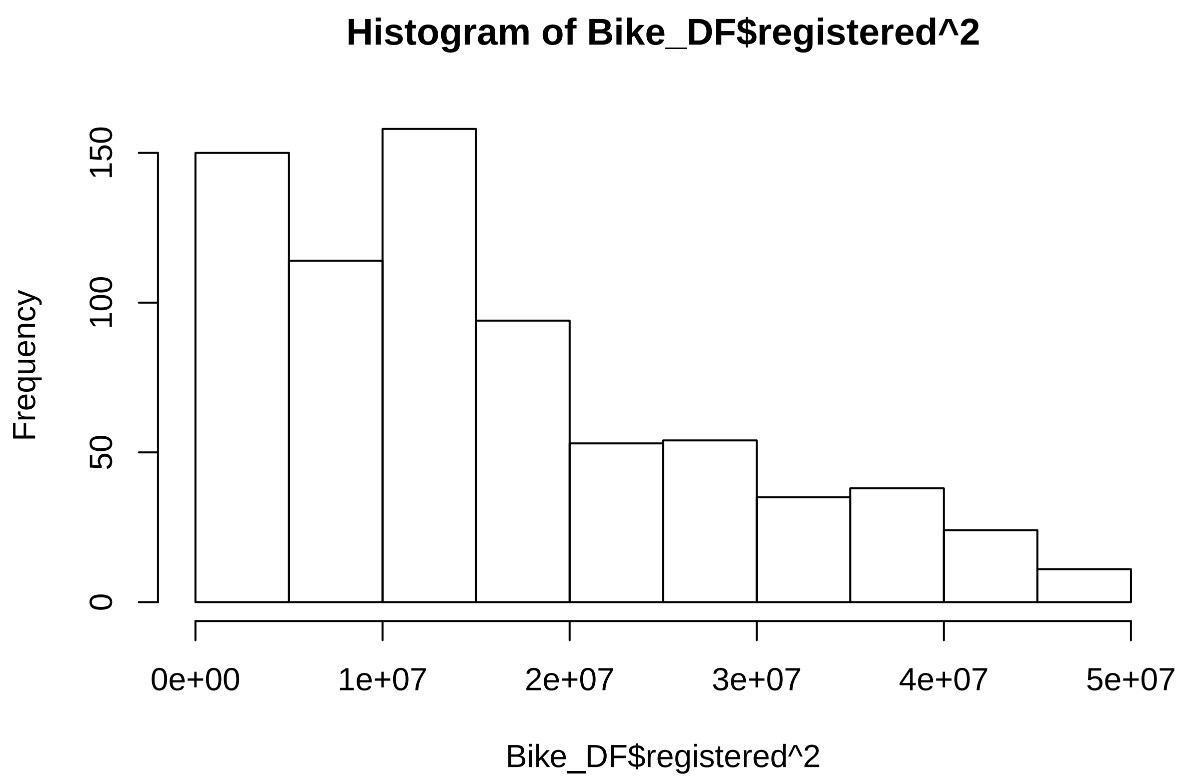
```
hist(Bike_DF$registered)
```




```
hist(log1p(Bike_DF$registered))
```




```r
hist(Bike_DF$registered^2)
```



It looks like registered does not need a transformation but we'll add the square and log anyway.

```
Bike_DF$sq_casual = Bike_DF$casual^2
Bike_DF$sq_registered = Bike_DF$registered^2
Bike_DF$log_casual = log1p(Bike_DF$casual)
Bike_DF$log_registered = log1p(Bike_DF$registered)
```

- e) Now we need to split our data into test and train sets.

```
set.seed(310)
train_idx = sample(1:nrow(Bike_DF),size = floor(0.70*nrow(Bike_DF)))
Bike_train = Bike_DF[train_idx,]
Bike_test = Bike_DF[-train_idx,]
```

- f) Run a forward stepwise model on the train set

```
library(leaps)
fwd_fit = regsubsets(cnt ~ season + holiday + mnth + weathersit + workingday + temp + hum + windspeed,
                     data = Bike_train,
                     nvmax = 10,
                     method = "forward")
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

```r
summary(fwd_fit)
```

```
## Subset selection object
## Call: regsubsets.formula(cnt ~ season + holiday + mnth + weathersit +
##     workingday + temp + hum + windspeed, data = Bike_train, nvmax = 10,
##     method = "forward")
## 12 Variables  (and intercept)
##                          Forced in Forced out
## seasonSummer                 FALSE      FALSE
## seasonFall                   FALSE      FALSE
## seasonWinter                 FALSE      FALSE
## holidayYes                   FALSE      FALSE
## mnth                         FALSE      FALSE
## weathersitCloudy             FALSE      FALSE
## weathersitLight Storms       FALSE      FALSE
## workingdayYes                FALSE      FALSE
## temp                         FALSE      FALSE
## hum                          FALSE      FALSE
## windspeed                    FALSE      FALSE
## weathersitHeavy Storms       FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: forward
##           seasonSummer seasonFall seasonWinter holidayYes mnth
## 1  ( 1 )  " "          " "        " "          " "        " "
## 2  ( 1 )  " "          " "        "*"          " "        " "
## 3  ( 1 )  " "          " "        "*"          " "        " "
## 4  ( 1 )  " "          " "        "*"          " "        " "
## 5  ( 1 )  "*"          " "        "*"          " "        " "
## 6  ( 1 )  "*"          " "        "*"          " "        " "
## 7  ( 1 )  "*"          " "        "*"          " "        " "
## 8  ( 1 )  "*"          " "        "*"          " "        " "
## 9  ( 1 )  "*"          " "        "*"          "*"        " "
## 10  ( 1 ) "*"          "*"        "*"          "*"        " "
## 11  ( 1 ) "*"          "*"        "*"          "*"        "*"
##           weathersitCloudy weathersitLight Storms weathersitHeavy Storms
## 1  ( 1 )  " "              " "                    " "
## 2  ( 1 )  " "              " "                    " "
## 3  ( 1 )  " "              " "                    " "
## 4  ( 1 )  " "              " "                    " "
## 5  ( 1 )  " "              " "                    " "
## 6  ( 1 )  " "              "*"                    " "
## 7  ( 1 )  " "              "*"                    " "
## 8  ( 1 )  "*"              "*"                    " "
## 9  ( 1 )  "*"              "*"                    " "
## 10  ( 1 ) "*"              "*"                    " "
## 11  ( 1 ) "*"              "*"                    " "
##           workingdayYes temp hum windspeed
## 1  ( 1 )  " "           "*"  " " " "
## 2  ( 1 )  " "           "*"  " " " "
## 3  ( 1 )  " "           "*"  "*" " "
## 4  ( 1 )  " "           "*"  "*" "*"
## 5  ( 1 )  " "           "*"  "*" "*"
```

```
## 6  ( 1 )  " "           "*"  "*" "*"
## 7  ( 1 )  "*"           "*"  "*" "*"
## 8  ( 1 )  "*"           "*"  "*" "*"
## 9  ( 1 )  "*"           "*"  "*" "*"
## 10  ( 1 ) "*"           "*"  "*" "*"
## 11  ( 1 ) "*"           "*"  "*" "*"
```

The first five variables selected are 'temp', 'seasonWinter', 'hum', 'windspeed', 'seasonSummer'.

- g) Now a backwards stepwise model.

```r
back_fit = regsubsets(cnt ~ season + holiday + mnth + weathersit + workingday + temp + hum + windspeed,
                     data = Bike_train,
                     nvmax = 10,
                     method = "backward")
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:
```

```r
summary(back_fit)
```

```
## Subset selection object
## Call: regsubsets.formula(cnt ~ season + holiday + mnth + weathersit +
##     workingday + temp + hum + windspeed, data = Bike_train, nvmax = 10,
##     method = "backward")
## 12 Variables  (and intercept)
##                        Forced in Forced out
## seasonSummer               FALSE      FALSE
## seasonFall                 FALSE      FALSE
## seasonWinter               FALSE      FALSE
## holidayYes                 FALSE      FALSE
## mnth                       FALSE      FALSE
## weathersitCloudy           FALSE      FALSE
## weathersitLight Storms     FALSE      FALSE
## workingdayYes              FALSE      FALSE
## temp                       FALSE      FALSE
## hum                        FALSE      FALSE
## windspeed                  FALSE      FALSE
## weathersitHeavy Storms     FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: backward
##           seasonSummer seasonFall seasonWinter holidayYes mnth
## 1  ( 1 )  " "          " "        " "          " "        " "
## 2  ( 1 )  " "          " "        "*"          " "        " "
## 3  ( 1 )  " "          " "        "*"          " "        " "
## 4  ( 1 )  " "          " "        "*"          " "        " "
## 5  ( 1 )  "*"          " "        "*"          " "        " "
## 6  ( 1 )  "*"          " "        "*"          " "        " "
## 7  ( 1 )  "*"          " "        "*"          " "        " "
## 8  ( 1 )  "*"          " "        "*"          " "        " "
```

```
## 9  ( 1 )  "*"          " "         "*"          "*"         " "
## 10  ( 1 ) "*"          "*"         "*"          "*"         " "
## 11  ( 1 ) "*"          "*"         "*"          "*"         "*"
##           weathersitCloudy weathersitLight Storms weathersitHeavy Storms
## 1  ( 1 )  " "              " "                    " "
## 2  ( 1 )  " "              " "                    " "
## 3  ( 1 )  " "              " "                    " "
## 4  ( 1 )  " "              " "                    " "
## 5  ( 1 )  " "              " "                    " "
## 6  ( 1 )  " "              "*"                    " "
## 7  ( 1 )  " "              "*"                    " "
## 8  ( 1 )  "*"              "*"                    " "
## 9  ( 1 )  "*"              "*"                    " "
## 10  ( 1 ) "*"              "*"                    " "
## 11  ( 1 ) "*"              "*"                    " "
##           workingdayYes temp hum windspeed
## 1  ( 1 )  " "           "*"  " " " "
## 2  ( 1 )  " "           "*"  " " " "
## 3  ( 1 )  " "           "*"  "*" " "
## 4  ( 1 )  " "           "*"  "*" "*"
## 5  ( 1 )  " "           "*"  "*" "*"
## 6  ( 1 )  " "           "*"  "*" "*"
## 7  ( 1 )  "*"           "*"  "*" "*"
## 8  ( 1 )  "*"           "*"  "*" "*"
## 9  ( 1 )  "*"           "*"  "*" "*"
## 10  ( 1 ) "*"           "*"  "*" "*"
## 11  ( 1 ) "*"           "*"  "*" "*"
```

The five variables are the same as the ones chosen in the forward model. We are not guaranteed the same variables in both models because in when using the forward method we add variables one at a time. The addition of another variable may cause one of the other variables that we have already added to become insignificant. In backwards we start with a full model and drop only the least significant variables.

- h) Now we'll run a Ridge model and plot it.

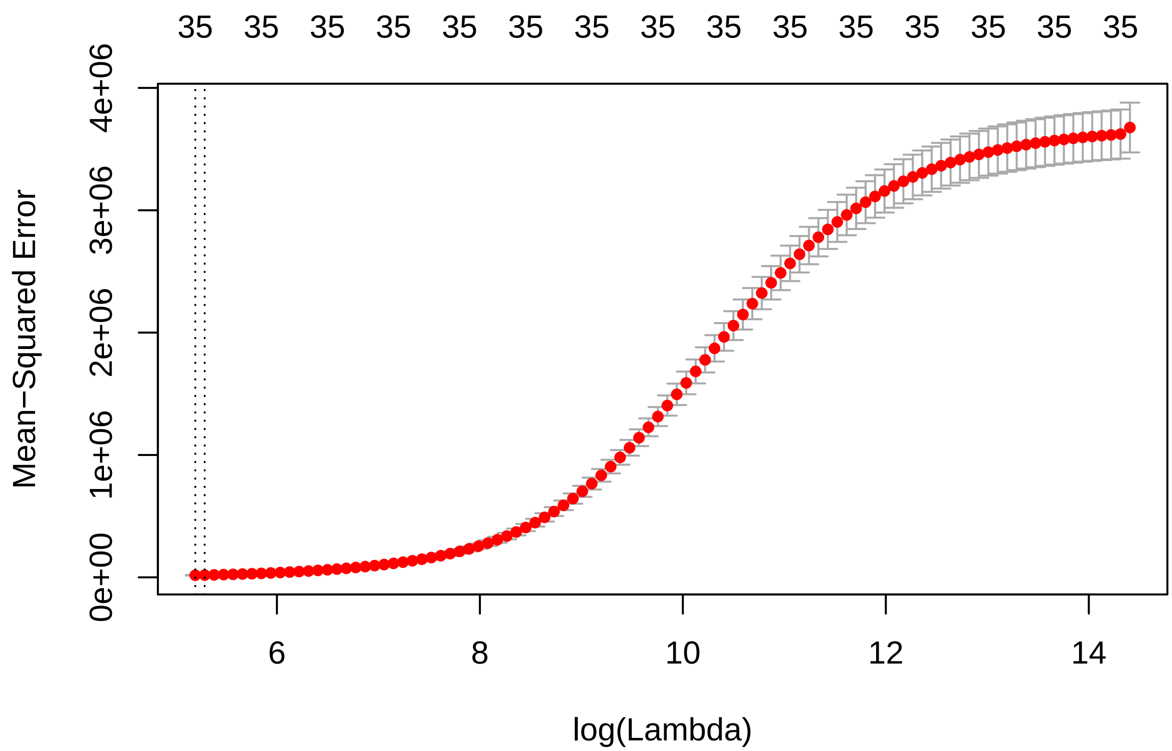
```r
library(ggplot2)
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-18
```

```r
library(glmnetUtils)
```

```
##
## Attaching package: 'glmnetUtils'

## The following objects are masked from 'package:glmnet':
##
##     cv.glmnet, glmnet
```

```r
train_subset = subset(Bike_train, select = -c(dteday,instant))
ridge_fit = cv.glmnet(cnt ~ .,
                      data = train_subset,
                      alpha = 0)
plot(ridge_fit)
```



- i) Output the values of lambda.min and lambda.1se

```r
ridge_fit$lambda.min
```

```
## [1] 180.4364
```

```r
ridge_fit$lambda.1se
```

```
## [1] 198.0288
```

Lambda.min gives us the lambda that produces the model with the minimized mean cross-validated error, and Lambda.1se gives us the lambda where the most regularized model is within 1 standard error of the minimum.

- j) Coefficients of the ridge model.

```r
coefmat = data.frame(
  ridge_min = as.matrix(round(coef(ridge_fit, s = ridge_fit$lambda.min), 3)),
  ridge_1se = as.matrix(round(coef(ridge_fit, s = ridge_fit$lambda.1se), 3))
)

colnames(coefmat) = c("Ridge Min", "Ridge 1se")

coefmat
```

```
##                        Ridge Min Ridge 1se
## (Intercept)            -4538.710 -4509.987
## seasonSpring            -104.729  -109.761
## seasonSummer              18.874    19.925
## seasonFall                 9.893     9.207
## seasonWinter              83.958    87.767
## yr2011                  -130.845  -139.388
## yr2012                   142.747   149.374
## mnth                      -0.704    -0.556
## holidayNo                 46.746    49.846
## holidayYes               -51.128   -53.477
## weekday0                 -34.744   -37.364
## weekday1                   1.715     1.359
## weekday2                  22.277    23.059
## weekday3                   8.210     8.979
## weekday4                   5.658     6.982
## weekday5                   5.190     5.356
## weekday6                  -6.724    -7.264
## workingdayNo             -33.925   -35.581
## workingdayYes             36.589    38.032
## weathersitClear           35.527    37.611
## weathersitCloudy         -29.265   -30.149
## weathersitLight Storms   -91.529   -98.995
## weathersitHeavy Storms     0.000     0.000
## temp                     160.575   166.152
## atemp                    176.496   184.153
## hum                      -77.228   -81.324
## windspeed               -146.750  -153.767
## casual                     0.429     0.422
## registered                 0.313     0.308
## actual_temp                3.391     3.701
## feel_temp                  3.034     3.257
## actual_windspeed          -1.929    -2.098
## actual_humidity           -0.661    -0.737
## sq_casual                  0.000     0.000
## sq_registered              0.000     0.000
## log_casual               175.978   178.344
## log_registered           665.174   662.199
```

All the coefficients increase as lambda increases except for casual and registered; they decreased with a higher lambda. This is interesting as I would think that they would go up as the number of casual and registered users should increase along with count (showing a strong correlation). When we look at the log transformed coefficients we get better looking, increasing values.

- k) Now we'll estimate a Lasso model.

```r
lasso_fit = cv.glmnet(cnt ~ .,
                      data = train_subset,
                      alpha = 1)
```

- l) Our lasso model chose three variables, registered, casual, and log_casual. Now we'll outout lambda.min/1se and the coefficients.

```r
lasso_fit$lambda.min
```

```
## [1] 47.9255
```

```r
lasso_fit$lambda.1se
```

```
## [1] 47.9255
```

```r
coefmat = data.frame(
  lasso_min = as.matrix(round(coef(lasso_fit, s = lasso_fit$lambda.min), 3)),
  lasso_1se = as.matrix(round(coef(lasso_fit, s = lasso_fit$lambda.1se), 3))
)

colnames(coefmat) = c("Lasso Min", "Lasso_1se")

coefmat
```

```
##                        Lasso Min Lasso_1se
## (Intercept)               75.897    75.897
## seasonSpring               0.000     0.000
## seasonSummer               0.000     0.000
## seasonFall                 0.000     0.000
## seasonWinter               0.000     0.000
## yr2011                     0.000     0.000
## yr2012                     0.000     0.000
## mnth                       0.000     0.000
## holidayNo                  0.000     0.000
## holidayYes                 0.000     0.000
## weekday0                   0.000     0.000
## weekday1                   0.000     0.000
## weekday2                   0.000     0.000
## weekday3                   0.000     0.000
## weekday4                   0.000     0.000
## weekday5                   0.000     0.000
## weekday6                   0.000     0.000
## workingdayNo               0.000     0.000
## workingdayYes              0.000     0.000
## weathersitClear            0.000     0.000
## weathersitCloudy           0.000     0.000
## weathersitLight Storms     0.000     0.000
## weathersitHeavy Storms     0.000     0.000
## temp                       0.000     0.000
## atemp                      0.000     0.000
## hum                        0.000     0.000
```

```
## windspeed                 0.000     0.000
## casual                    0.935     0.935
## registered                0.975     0.975
## actual_temp               0.000     0.000
## feel_temp                 0.000     0.000
## actual_windspeed          0.000     0.000
## actual_humidity           0.000     0.000
## sq_casual                 0.000     0.000
## sq_registered             0.000     0.000
## log_casual               10.780    10.780
## log_registered            0.000     0.000
```

- m) While Lasso retained the most significant variables in the model, count of casual users (normal count and log transformed), and count of registered users, it completely dropped any other variable that may have been affecting bike share usage. It's obvious that usage will go up with the number of registered users and casual users, so in this case I would choose the Ridge model. The ridge model may be giving us the variables that have an impact on the number of bike shares outside the number of users that are within the bike share.