Bike Share Model

Imogen Cleaver-Stigum

SETUP

Load the Data

```
fullDataset = read.csv("day.csv")
```

Split the Data Randomly into Test and Training Data Sets

Note: The exact values for correlation coefficients, the exact points on the scatter plots, etc will be slightly different each time this is run because the randomly selected training data set used during the exploration phase will consist of slightly different data each time.

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
splitVector = caret::createDataPartition(fullDataset[,1], p=.8, list=F, times=1)
train = fullDataset[splitVector,] # the training data set
test = fullDataset[!row.names(fullDataset) %in% row.names(train),] # the test data set
```

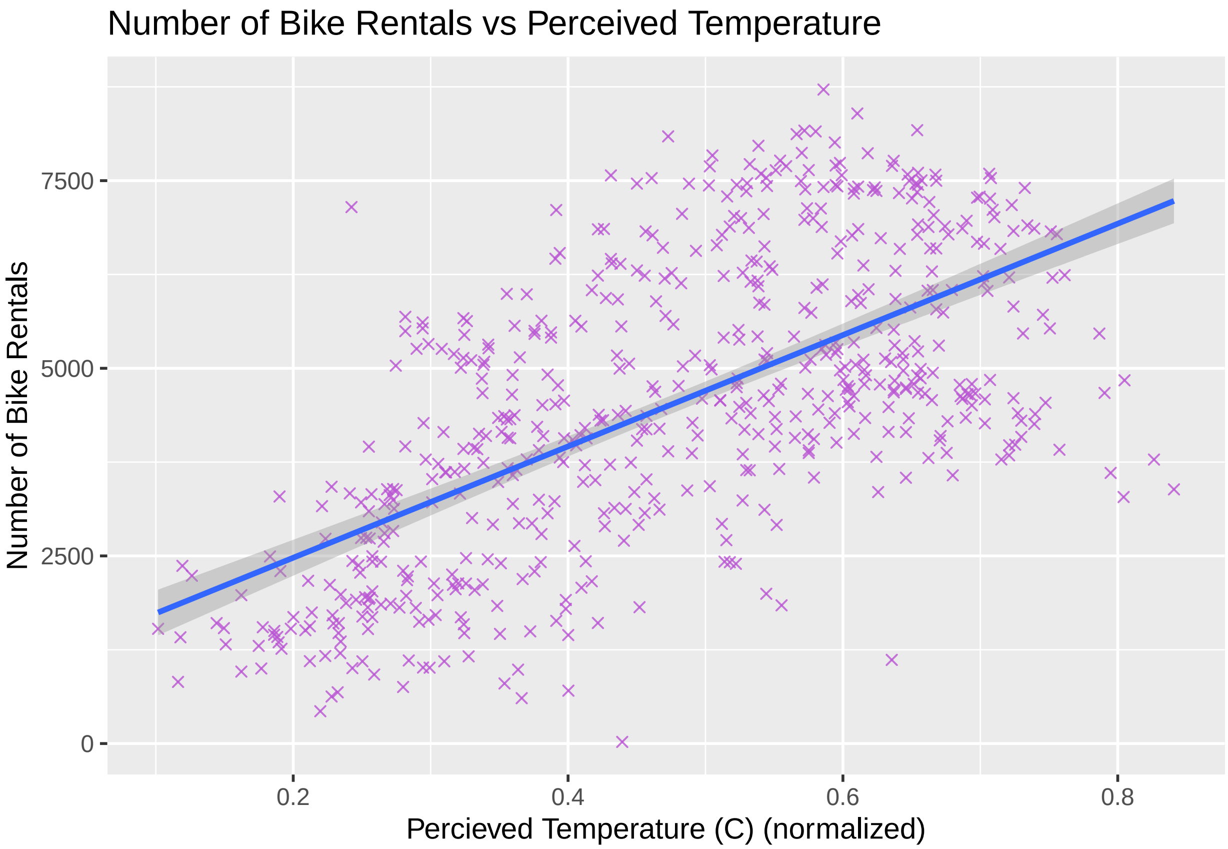
EXPLORING THE DATA BY INDIVIDUAL VARIABLES

Is there a correlation between bike rentals and actual or perceived temperature?

Note: The temperature was already normalized when I downloaded the data set.

```
library(ggplot2)
#first with perceived temperature
ggplot(train, aes(x=atemp, y=cnt)) +
  geom_point(colour = alpha("mediumorchid", .8), shape = 4)+
  geom_smooth(method=lm)+
  labs(title = "Number of Bike Rentals vs Perceived Temperature") +
  xlab("Percieved Temperature (C) (normalized)") +
  ylab("Number of Bike Rentals")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

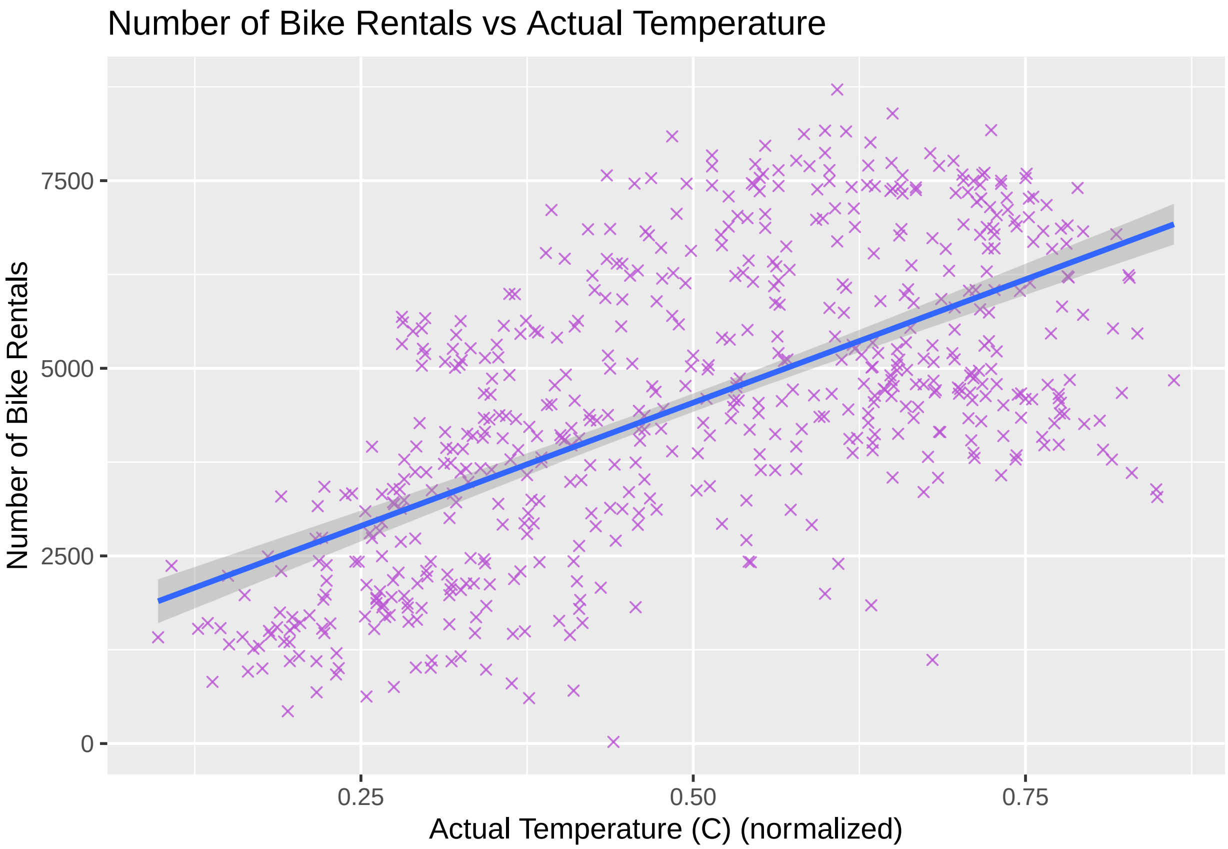


```
cor(train$atemp, train$cnt)
```

```
## [1] 0.6290006
```

```
# then with actual temperature
ggplot(train, aes(x=temp, y=cnt)) +
  geom_point(colour = alpha("mediumorchid", .8), shape = 4)+
  geom_smooth(method=lm)+
  labs(title = "Number of Bike Rentals vs Actual Temperature") +
  xlab("Actual Temperature (C) (normalized)") +
  ylab("Number of Bike Rentals")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
cor(train$temp, train$cnt)
```

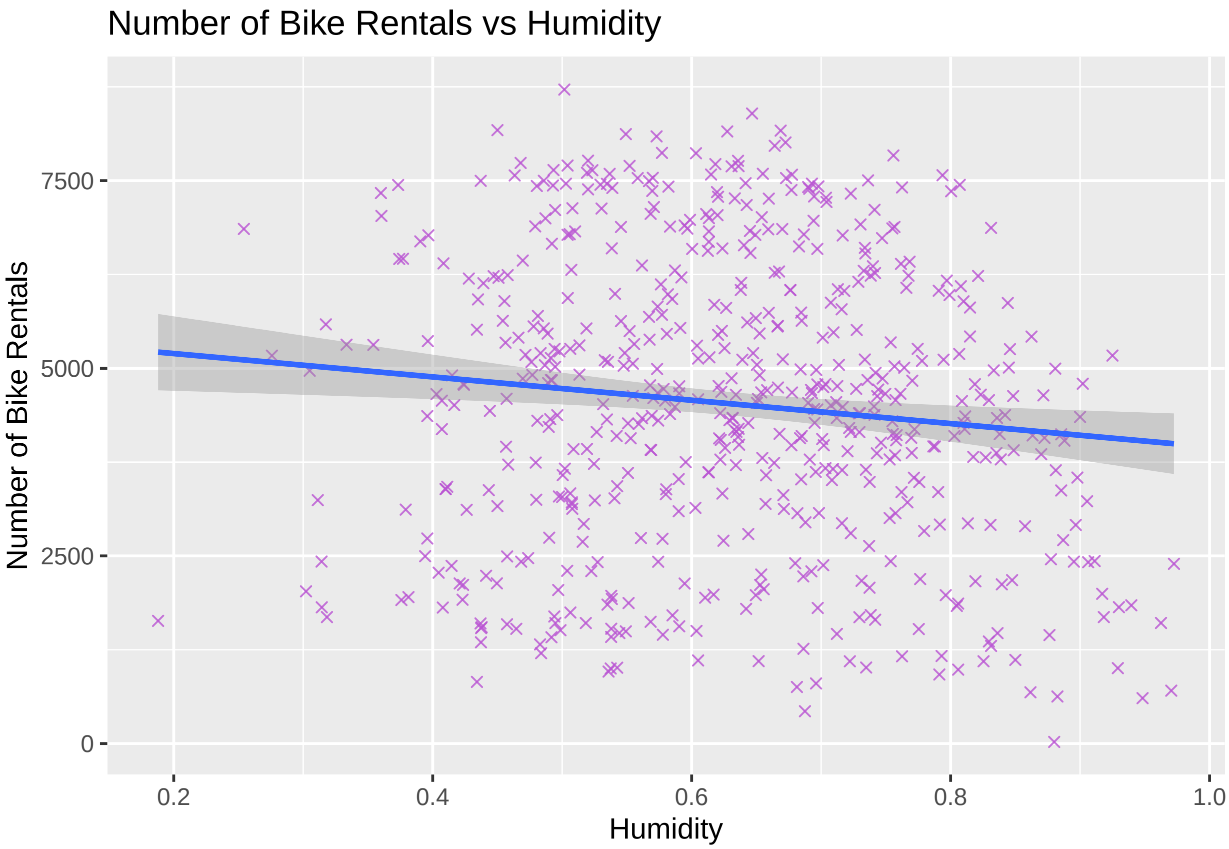
```
## [1] 0.62698
```

Is there a correlation between bike rentals and humidity?

Note: The humidity was already normalized when I downloaded the data set.

```
ggplot(train, aes(x=hum, y=cnt)) +
  geom_point(colour = alpha("mediumorchid", .8), shape = 4)+
  geom_smooth(method=lm)+
  labs(title = "Number of Bike Rentals vs Humidity") +
  xlab("Humidity") +
  ylab("Number of Bike Rentals")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



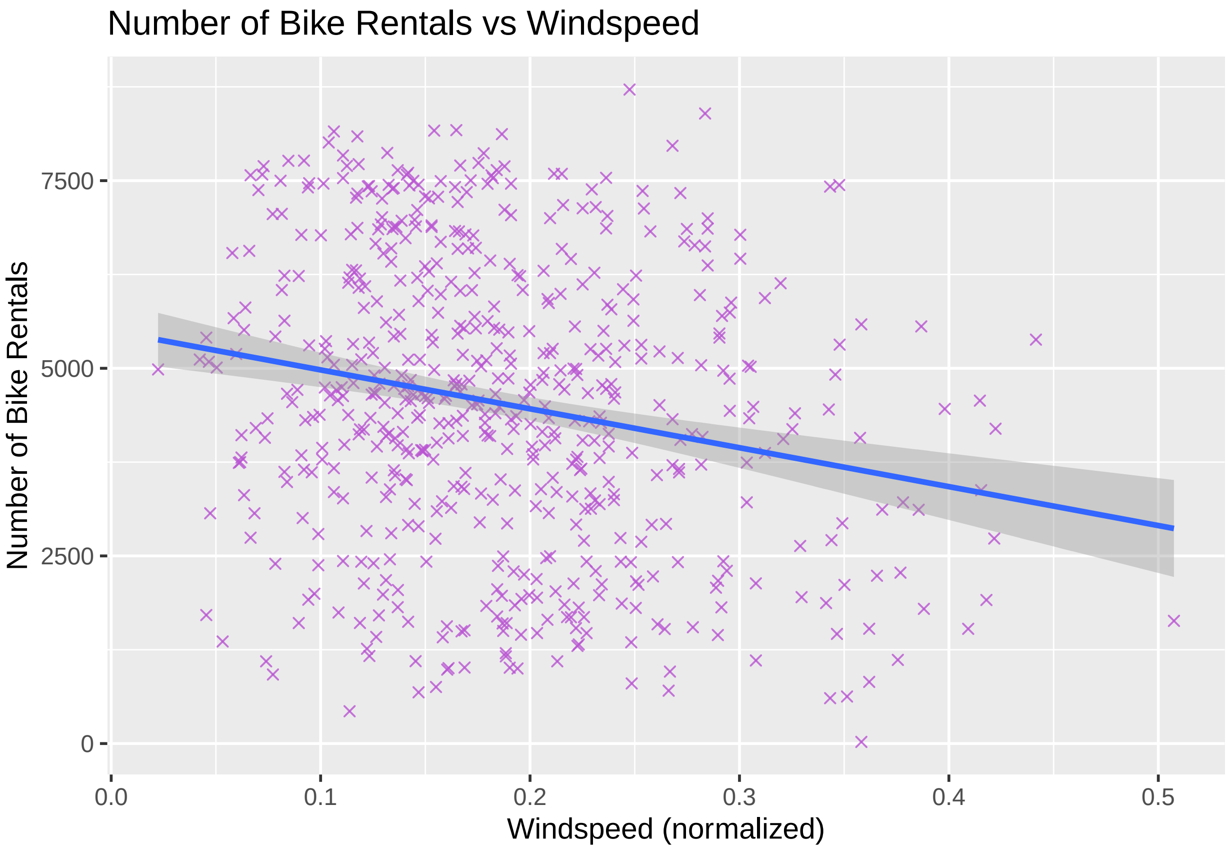
```
cor(train$hum, train$cnt)
```

```
## [1] -0.1148358
```

Is there a correlation between bike rentals and windspeed?

```
ggplot(train, aes(x=windspeed, y=cnt)) +
  geom_point(colour = alpha("mediumorchid", .8), shape = 4)+
  geom_smooth(method=lm)+
  labs(title = "Number of Bike Rentals vs Windspeed") +
  xlab("Windspeed (normalized)") +
  ylab("Number of Bike Rentals")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

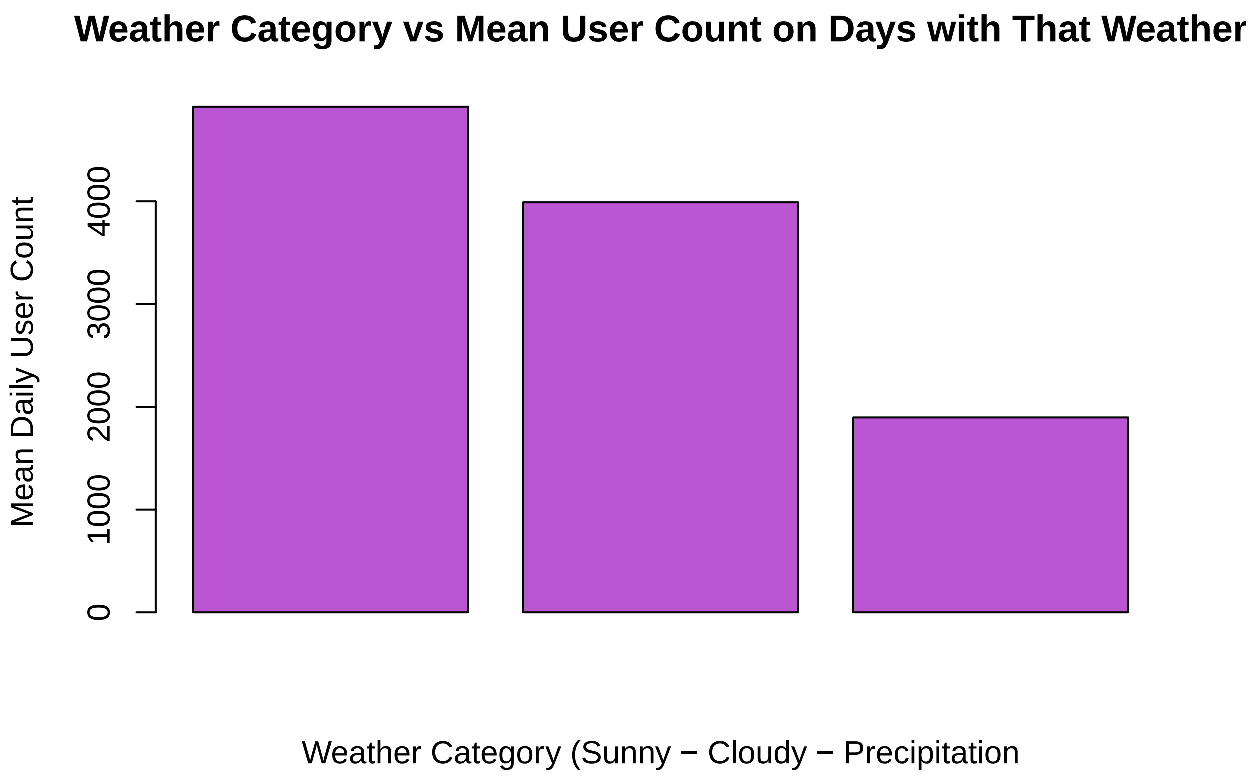


```r
cor(train$windspeed, train$cnt)
```

```
## [1] -0.2099737
```

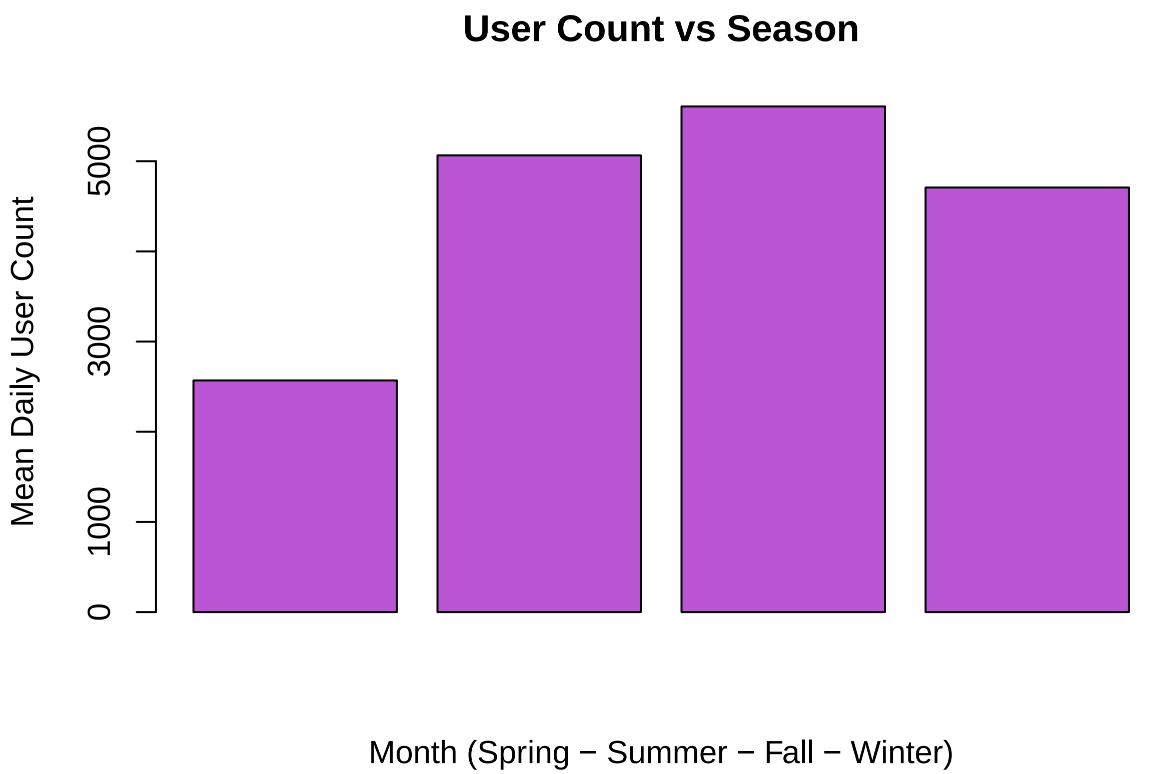
Does the weather category affect bike rentals?

```r
weatherMeans = c(
  mean(train$cnt[train$weathersit == 1]), # mean number of rentals on clear / partly cloudy days
  mean(train$cnt[train$weathersit == 2]), # mean number of rentals on misty / cloudy days
  mean(train$cnt[train$weathersit == 3])) # mean number of rentals on days with precipitation
barplot(weatherMeans,
        main = "Weather Category vs Mean User Count on Days with That Weather",
        xlab = "Weather Category (Sunny - Cloudy - Precipitation",
        ylab = "Mean Daily User Count",
        col = "mediumorchid")
```

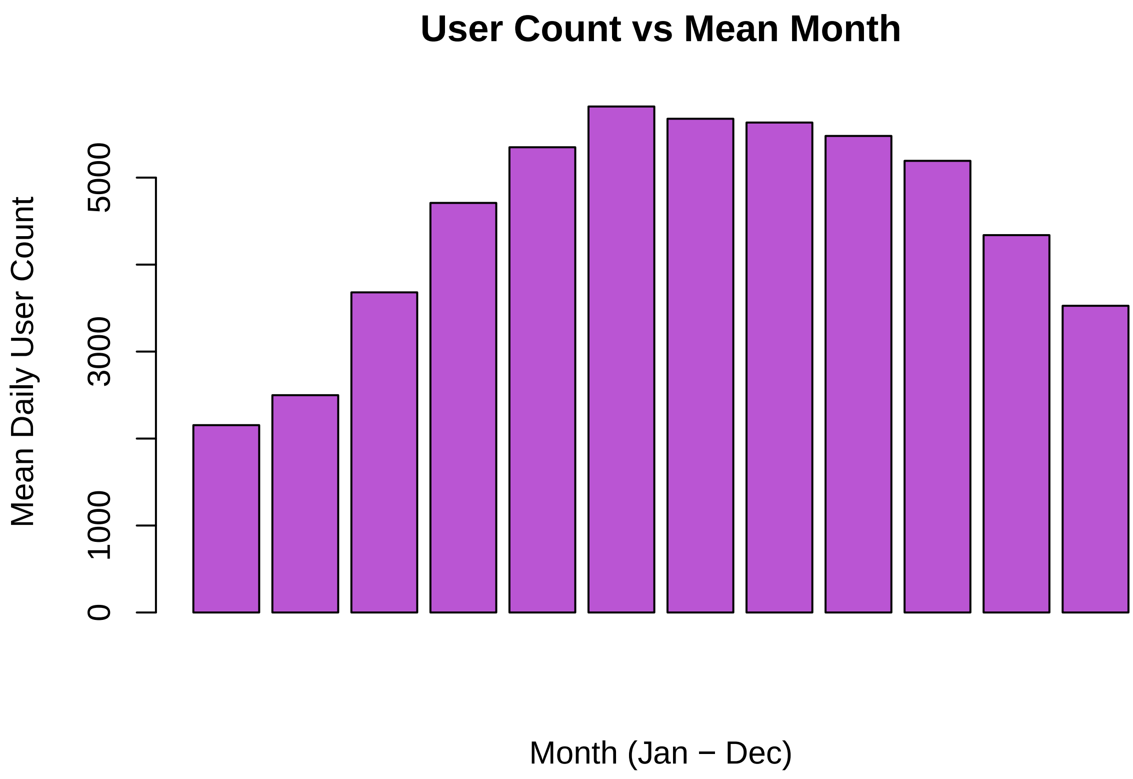


### Does the season or month affect bike rentals?

```r
seasonMeans = c(
  mean(train$cnt[train$season == 1]), # mean number of rentals in spring
  mean(train$cnt[train$season == 2]), # mean number of rentals in summer
  mean(train$cnt[train$season == 3]), # mean number of rentals in fall
  mean(train$cnt[train$season == 4])) # mean number of rentals in winter
barplot(seasonMeans,
        main = "User Count vs Season",
        xlab = "Month (Spring - Summer - Fall - Winter)",
        ylab = "Mean Daily User Count",
        col = "mediumorchid")
```



```
monthMeans = c(
  mean(train$cnt[train$mnth == 1]), # mean number of rentals in Jan
  mean(train$cnt[train$mnth == 2]), # mean number of rentals in Feb
  mean(train$cnt[train$mnth == 3]), # mean number of rentals in Mar
  mean(train$cnt[train$mnth == 4]), # mean number of rentals in Apr
  mean(train$cnt[train$mnth == 5]), # mean number of rentals in May
  mean(train$cnt[train$mnth == 6]), # mean number of rentals in Jun
  mean(train$cnt[train$mnth == 7]), # mean number of rentals in Jul
  mean(train$cnt[train$mnth == 8]), # mean number of rentals in Aug
  mean(train$cnt[train$mnth == 9]), # mean number of rentals in Sept
  mean(train$cnt[train$mnth == 10]), # mean number of rentals in Oct
  mean(train$cnt[train$mnth == 11]), # mean number of rentals in Nov
  mean(train$cnt[train$mnth == 12])) # mean number of rentals in Dec
barplot(monthMeans,
        main = "User Count vs Mean Month",
        xlab = "Month (Jan - Dec)",
        ylab = "Mean Daily User Count",
        col = "mediumorchid")
```



###

Are there more/less bike rentals on holidays?

```
mean(train$cnt[train$holiday == 0]) # mean number of rentals on non-holidays
```

```
## [1] 4543.492
```

```
mean(train$cnt[train$holiday == 1]) # mean number of rentals on holidays
```

```
## [1] 3962.5
```

Are there more/less bike rentals on working days?

```
mean(train$cnt[train$workingday == 0]) # mean number of rentals on non-working days
```

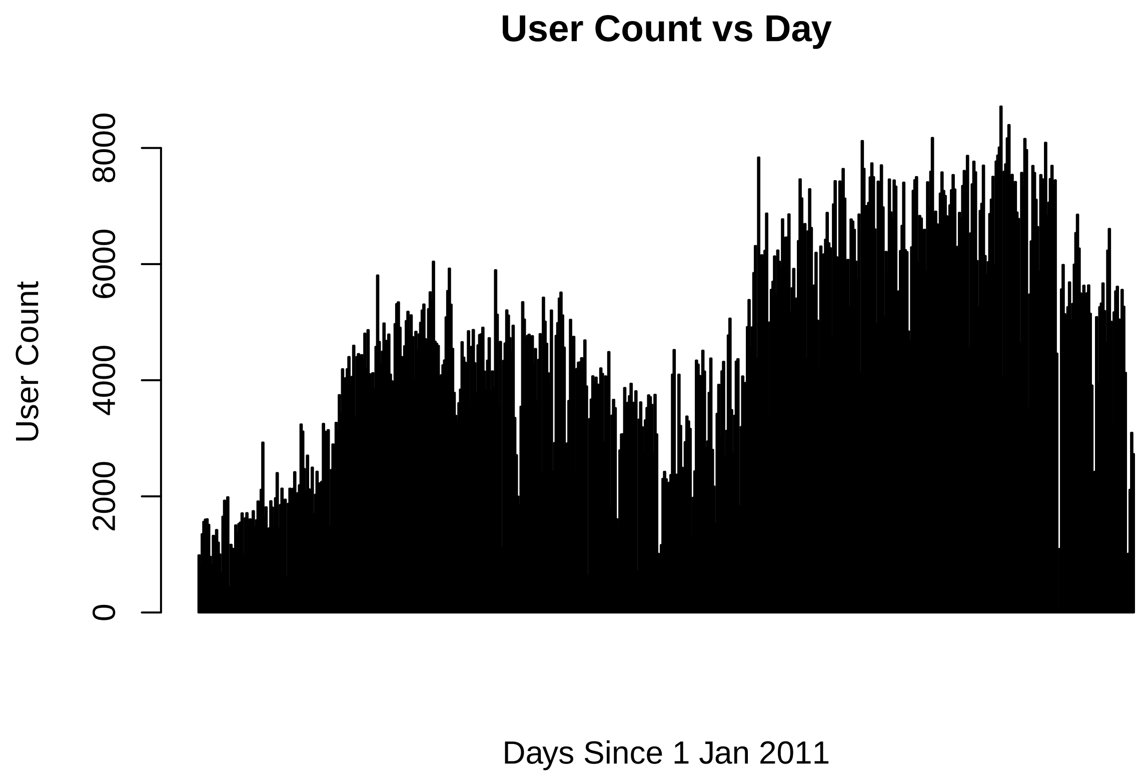
```
## [1] 4356.848
```

```
mean(train$cnt[train$workingday == 1]) # mean number of rentals on working days
```

```
## [1] 4602.759
```

Does the number of bike rentals change over the long term?

```
barplot(train$cnt,
        main = "User Count vs Day",
        xlab = "Days Since 1 Jan 2011",
        ylab = "User Count") # this works because the data is already in order by date
```



```r
# the "instant" attribute is equivalent to the number of days since 1 Jan 2011
mean(train$cnt[train$workingday == 0]) # mean number of rentals in 2011
```

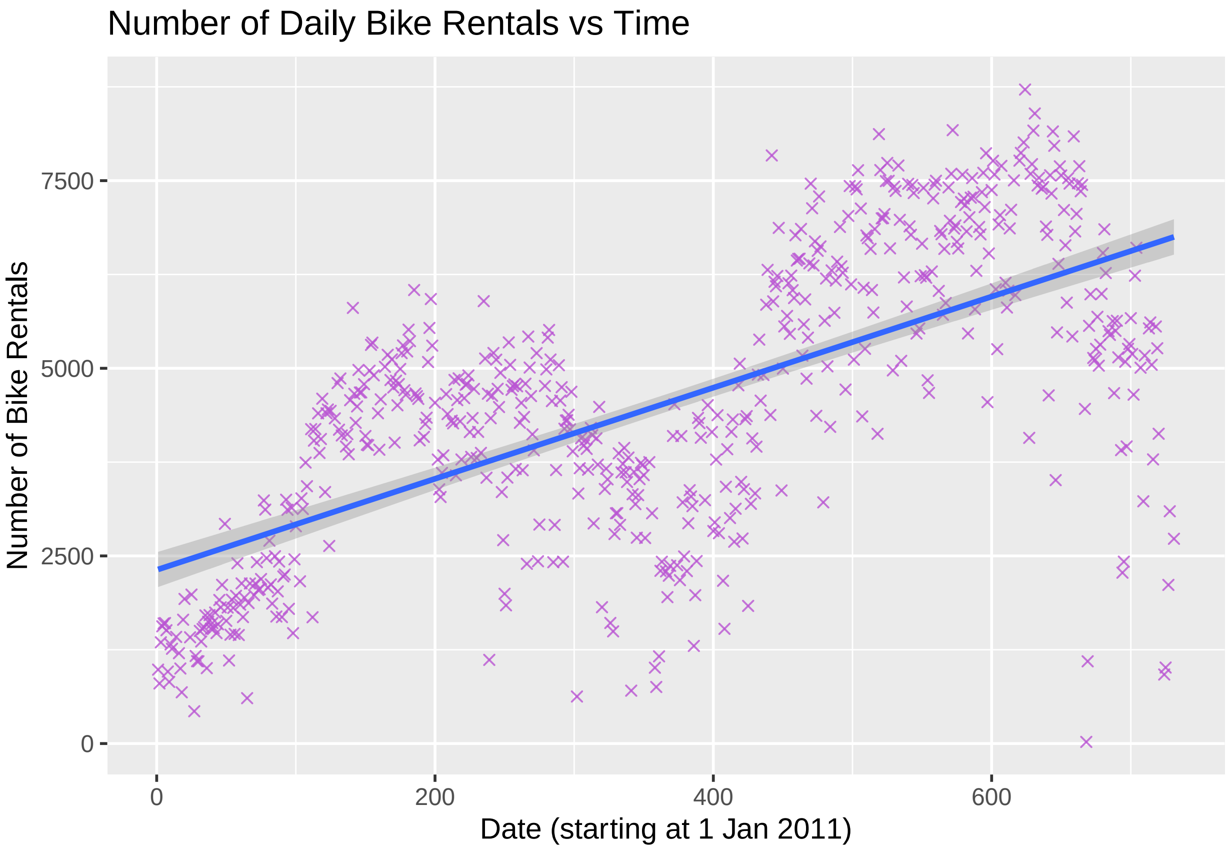
```
## [1] 4356.848
```

```r
mean(train$cnt[train$workingday == 1]) # mean number of rentals in 2012
```

```
## [1] 4602.759
```

```r
ggplot(train, aes(x=instant, y=cnt)) +
  geom_point(colour = alpha("mediumorchid", .8), shape = 4)+
  geom_smooth(method=lm)+
  labs(title = "Number of Daily Bike Rentals vs Time") +
  xlab("Date (starting at 1 Jan 2011)") +
  ylab("Number of Bike Rentals")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

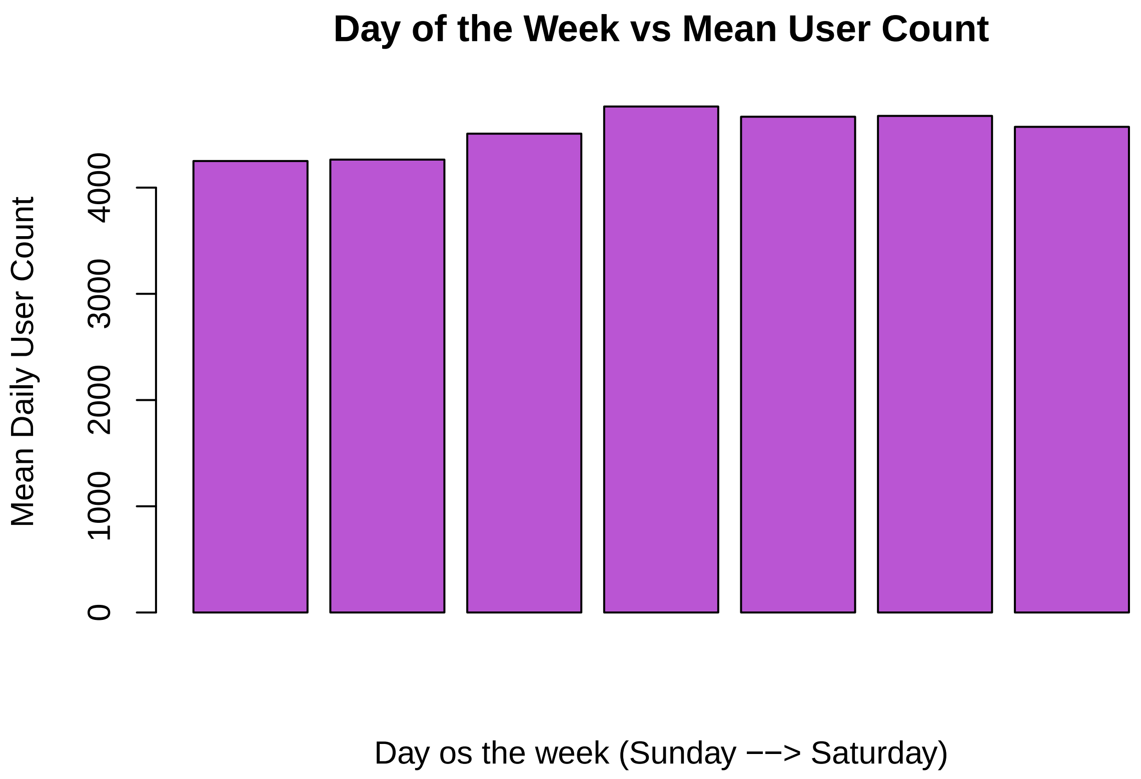


```
cor(train$instant, train$cnt)
```

```
## [1] 0.6618153
```

How do the days of the week affect number of bike rentals?

```
weekdayMeans = c(
  mean(train$cnt[train$weekday == 0]), # mean number of rentals on Sundays
  mean(train$cnt[train$weekday == 1]), # mean number of rentals on Mondays
  mean(train$cnt[train$weekday == 2]), # mean number of rentals on Tuesdays
  mean(train$cnt[train$weekday == 3]), # mean number of rentals on Wednesdays
  mean(train$cnt[train$weekday == 4]), # mean number of rentals on Thursdays
  mean(train$cnt[train$weekday == 5]), # mean number of rentals on Fridays
  mean(train$cnt[train$weekday == 6])) # mean number of rentals on Saturdays
barplot(weekdayMeans,
        main = "Day of the Week vs Mean User Count",
        xlab = "Day os the week (Sunday --> Saturday)",
        ylab = "Mean Daily User Count",
        col = "mediumorchid")
```



DATA PREPARATION

For both the training data and the testing data.

Normalize number of days since 1 Jan 2011

```
train$normDays = (train$instant-1)/730
test$normDays = (test$instant-1)/730
```

Create binary indicators for the categorical variable of weather

```
train$goodWeather = ifelse(train$weathersit == 1, 1, 0)
train$mediumWeather = ifelse(train$weathersit == 2, 1, 0)
train$badWeather = ifelse(train$weathersit == 3, 1, 0)

test$goodWeather = ifelse(test$weathersit == 1, 1, 0)
test$mediumWeather = ifelse(test$weathersit == 2, 1, 0)
test$badWeather = ifelse(test$weathersit == 3, 1, 0)
```

Create binary indicators for the categorical variable of month

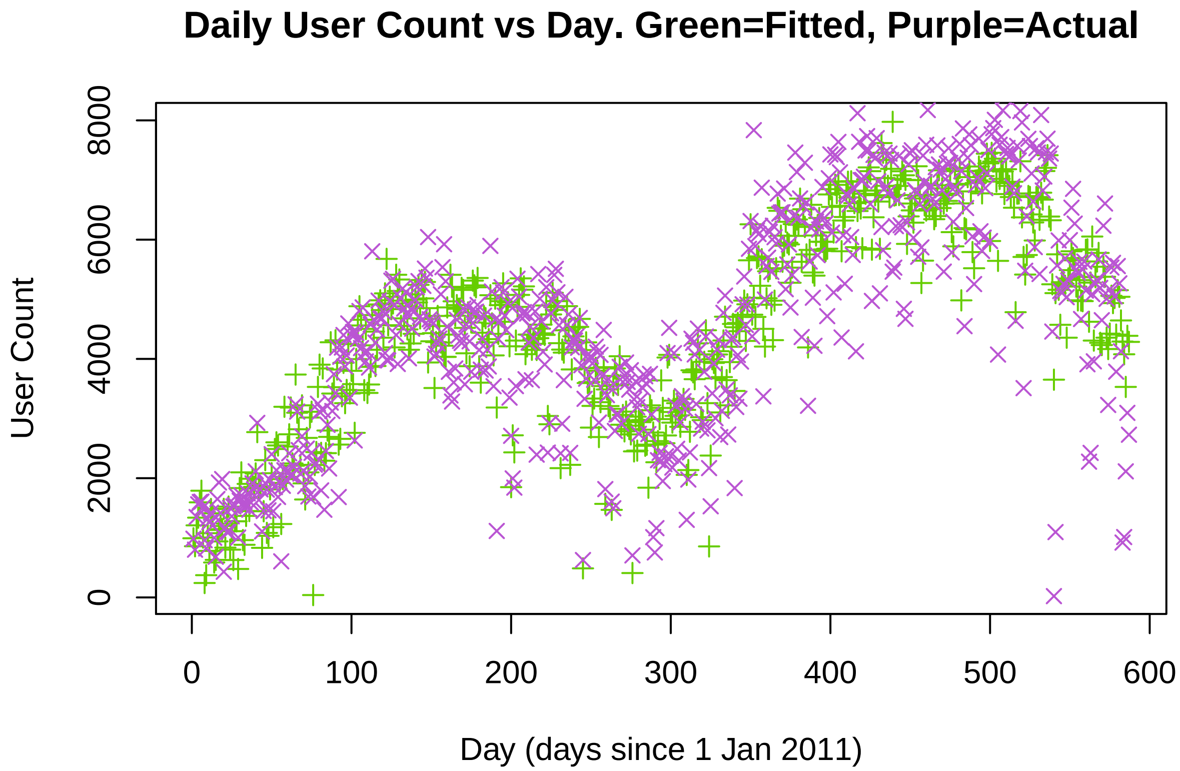
None is needed for December because December is just the case where all the other values are 0.

```
train$jan = ifelse(train$mnth == 1, 1, 0)
train$feb = ifelse(train$mnth == 2, 1, 0)
train$mar = ifelse(train$mnth == 3, 1, 0)
train$apr = ifelse(train$mnth == 4, 1, 0)
train$may = ifelse(train$mnth == 5, 1, 0)
train$jun = ifelse(train$mnth == 6, 1, 0)
train$jul = ifelse(train$mnth == 7, 1, 0)
train$aug = ifelse(train$mnth == 8, 1, 0)
train$sep = ifelse(train$mnth == 9, 1, 0)
train$oct = ifelse(train$mnth == 10, 1, 0)
train$nov = ifelse(train$mnth == 11, 1, 0)

test$jan = ifelse(test$mnth == 1, 1, 0)
test$feb = ifelse(test$mnth == 2, 1, 0)
test$mar = ifelse(test$mnth == 3, 1, 0)
test$apr = ifelse(test$mnth == 4, 1, 0)
test$may = ifelse(test$mnth == 5, 1, 0)
test$jun = ifelse(test$mnth == 6, 1, 0)
test$jul = ifelse(test$mnth == 7, 1, 0)
test$aug = ifelse(test$mnth == 8, 1, 0)
test$sep = ifelse(test$mnth == 9, 1, 0)
test$oct = ifelse(test$mnth == 10, 1, 0)
test$nov = ifelse(test$mnth == 11, 1, 0)
```

MULTIVARIATE LINEAR REGRESSION

```
# define the coefficients and constant
coeff = c(1:19)

# the hypothesis function
hypothesis <- function(coeffi, i, df) {
  return (coeffi[1] +
    coeffi[2] * df$windspeed[i] +
    coeffi[3] * df$atemp[i] +
    coeffi[4] * df$normDays[i] +
    coeffi[5] * df$hum[i] +
    coeffi[6] * df$goodWeather[i] +
    coeffi[7] * df$mediumWeather[i] +
    coeffi[8] * df$workingday[i] +
    coeffi[9] * df$jan[i] +
    coeffi[10] * df$feb[i]+
    coeffi[11] * df$mar[i]+
    coeffi[12] * df$apr[i]+
    coeffi[13] * df$may[i]+
    coeffi[14] * df$jun[i]+
    coeffi[15] * df$jul[i]+
    coeffi[16] * df$aug[i]+
    coeffi[17] * df$sep[i]+
    coeffi[18] * df$oct[i]+
    coeffi[19] * df$nov[i])
}
```

```
# find the coefficients and record them
model = lm(cnt ~ windspeed + atemp + normDays + hum + goodWeather + mediumWeather
           + workingday + jan + feb + mar + apr + may + jun + jul + aug + sep
           + oct + nov, data=train)
for (i in 1:length(model$coefficients)){
  coeff[i] = model$coefficients[i]
}

# visually compare the fitted vs actual values
plot(fitted(model),
     col="chartreuse3",
     pch=3,
     main = "Daily User Count vs Day. Green=Fitted, Purple=Actual",
     xlab = "Day (days since 1 Jan 2011)",
     ylab = "User Count")
points(train$cnt,
       col="mediumorchid",
       pch=4)
```



```
# animate
# the animations do not display properly in the pdf report
library(gganimate)
library(gifski)
anim = data.frame(cnt=c(train$cnt,fitted(model)),
                  instant=c(train$instant,train$instant),
                  indicator = c(rep("Actual Counts (purple)", 587),
                                rep("Predicted Counts (green)", 587)))
colors = c(rep("mediumorchid", 587), rep("chartreuse3", 587))
p = ggplot(anim, aes(x=instant,y=cnt)) +
  geom_point(color=colors, size=1, pch=4) +
  transition_states(indicator) + ease_aes('cubic-in-out') +
  ggtitle('Now Showing {closest_state}')
```

TESTING AND EVALUATING THE MODEL

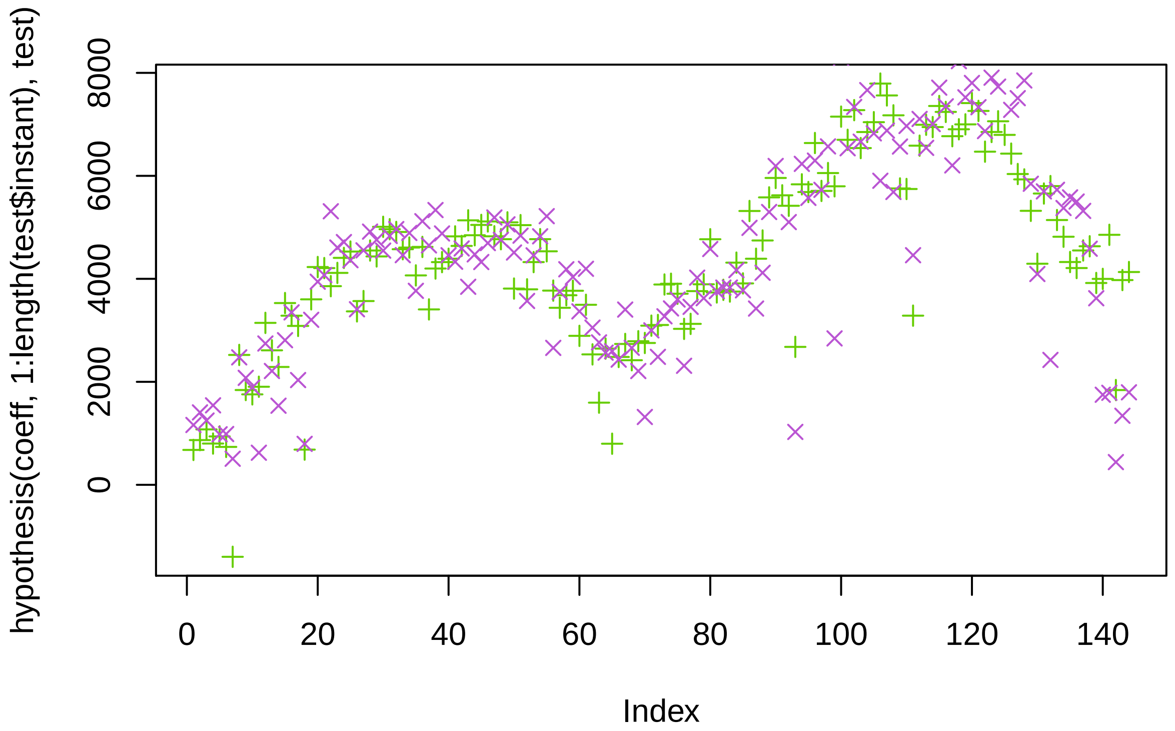
```
# the PRESS function
press = function(coeffi, df) {
  sumOfResidualsSquared = 0
  for (i in 1:length(df$instant)) {
    residual = df$cnt[i] - hypothesis(coeffi, i, df)
    sumOfResidualsSquared = sumOfResidualsSquared + residual ^ 2
  }
  return (sumOfResidualsSquared)
}

# compare the PRESS for train data vs test data to see if the model is over/under fitted
press(coeff,train) / 587
```

```
## [1] 636290.3
```

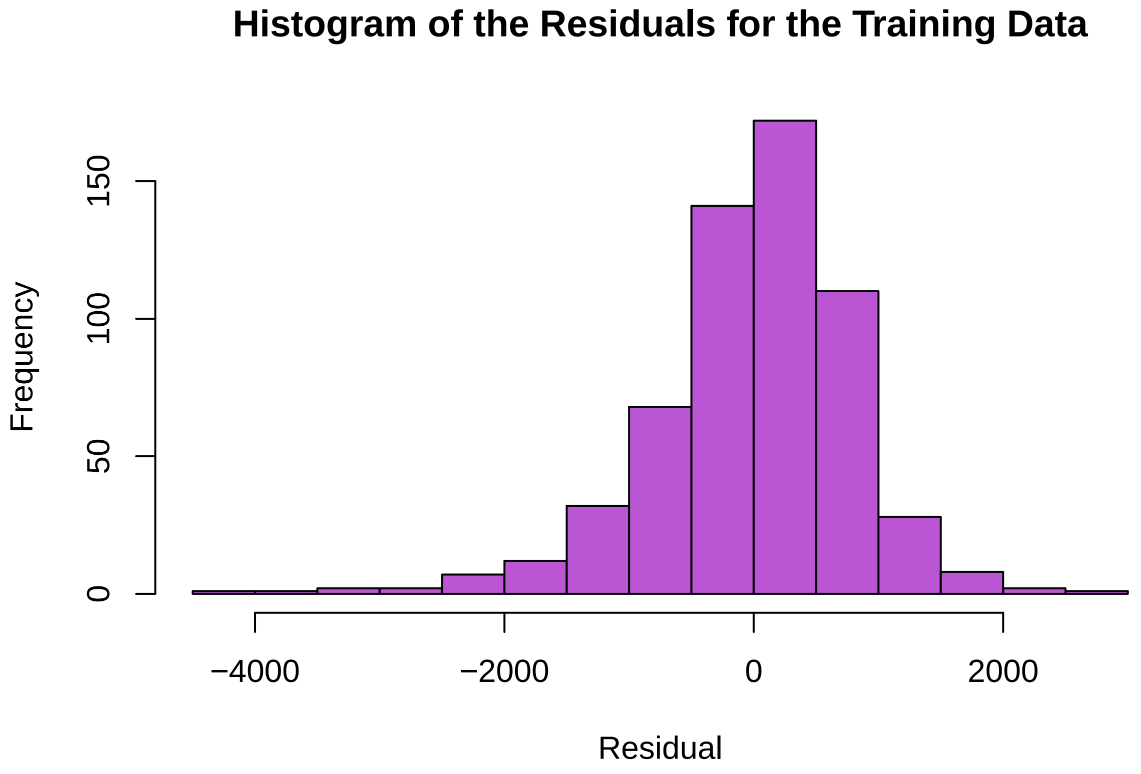
```
press(coeff,test) / 144
```

```
## [1] 871231.4
```

```
# plot actual vs fitted values for test data
plot(hypothesis(coeff, 1:length(test$instant), test),
     col="chartreuse3",
     pch=3)
points(test$cnt,
       col="mediumorchid",
       pch=4)
```



```
# plot the residuals to make sure they are relatively low and centered around 0
hist(resid(model),
     main = "Histogram of the Residuals for the Training Data",
     xlab = "Residual",
     ylab = "Frequency",
     col = "mediumorchid")
```



```r
# plot and animate the fitted vs actual values for test data
# the animations do not display properly in the pdf report
testAnim = data.frame(cnt=c(test$cnt,hypothesis(coeff, 1:length(test$instant), test)),
                      instant=c(test$instant,test$instant),
                      indicator = c(rep("Actual Counts (purple)", 144),
                                    rep("Predicted Counts (green)", 144)))
testColors = c(rep("mediumorchid", 144), rep("chartreuse3", 144))
testp = ggplot(testAnim, aes(x=instant,y=cnt)) +
  geom_point(color=testColors, size=1, pch=4) +
  transition_states(indicator) + ease_aes('cubic-in-out') +
  ggtitle('Now Showing {closest_state}')

# check the R^2 value
coeffOfDetermination = function(coeffi, df) {
  numerator = 0
  denominator = 0
  for (i in 1:length(df$instant)){
    numerator = numerator + ( hypothesis(coeffi, i, df) - df$cnt[i] )^2
    denominator = denominator + ( df$cnt[i] - mean(df$cnt) )^2
  }
  return (1 - (numerator / denominator))
}
coeffOfDetermination(coeff, train)
```

```
## [1] 0.8290872
```

```r
coeffOfDetermination(coeff, test)
```

```
## [1] 0.7730678
```

```r
# Root mean squares error of prediction
rmsep = function(actual, predicted) {
  return (sqrt( (1/length(actual)) * sum((actual - predicted)^2)))
}
rmsep(train$cnt, hypothesis(coeff, 1:length(train$cnt), train))
```

```
## [1] 797.678
```

```r
rmsep(test$cnt, hypothesis(coeff, 1:length(test$cnt), test))
```

```
## [1] 933.3978
```