# Project_2

*Laney Huang*

*April 28, 2017*

Reading Data:

```
bike.train <- read.csv("train.csv", header=TRUE)
bike.train$season <- factor(bike.train$season, levels=c(1,2,3,4),
                            labels = c("spring","summer","fall","winter"))
bike.train$holiday <- factor(bike.train$holiday, levels=c(0,1),
                             labels=c("No","Yes"))
bike.train$workingday <- factor(bike.train$workingday, levels=c(0,1),
                             labels=c("No","Yes"))
bike.train$weather <- factor(bike.train$weather, levels=c(1,2,3,4),
                             labels=c(1,2,3,3)) # 1 is most pleasant
times <- as.numeric(str_sub(bike.train$datetime, -8, -7))
bike.train$time <- factor(sapply(times,
                           function(x) {
                             if(1<x & x<6) {return("Early Morn")}
                             else if(5<x & x<10) {return("Morning")}
                             else if(9<x & x<14) {return("Noon")}
                             else if(13<x & x<18) {return("Afternoon")}
                             else if(17<x & x<22) {return("Evening")}
                             else if(x>21 | x<2) {return("Night")}
                           }), levels=c("Early Morn","Morning","Noon","Afternoon",
                                        "Evening","Night"))
```

*I have converted the datetime variable to 6 factors, corresponding to the hour of the day of the rental. From 2:00 to 6:00 is night, from 6:00 to 10:00 is morning, from 10:00 to 14:00 is noon, from 14:00 to 18:00 is afternoon, from 18:00 to 22:00 is evening, and from 22:00 to 2:00 is night. I have also combined weather levels 3 and 4, because below, I have noted in the brief summary analysis, that category 4 has only 1 data entry, and thus may be difficult to analyze.

```
bike.test <- read.csv("test.csv", header = TRUE)
bike.test$season <- factor(bike.test$season, levels=c(1,2,3,4),
                            labels = c("spring","summer","fall","winter"))
bike.test$holiday <- factor(bike.test$holiday, levels=c(0,1),
                             labels=c("No","Yes"))
bike.test$workingday <- factor(bike.test$workingday, levels=c(0,1),
                             labels=c("No","Yes"))
bike.test$weather <- factor(bike.test$weather, levels=c(1,2,3,4),
                             labels=c(1,2,3,3)) # 1 is most pleasant
times2 <- as.numeric(str_sub(bike.test$datetime, -8, -7))
bike.test$time <- factor(sapply(times2,
                           function(x) {
                             if(1<x & x<6) {return("Early Morn")}
                             else if(5<x & x<10) {return("Morning")}
                             else if(9<x & x<14) {return("Noon")}
                             else if(13<x & x<18) {return("Afternoon")}
                             else if(17<x & x<22) {return("Evening")}
                             else if(x>21 | x<2) {return("Night")}
                           }), levels=c("Early Morn","Morning","Noon","Afternoon",
```

```
                                        "Evening","Night"))
```

*Reading in the test dataset for use later in the prediction accuracy.

Basic Analysis:

```
summary(bike.train)
```

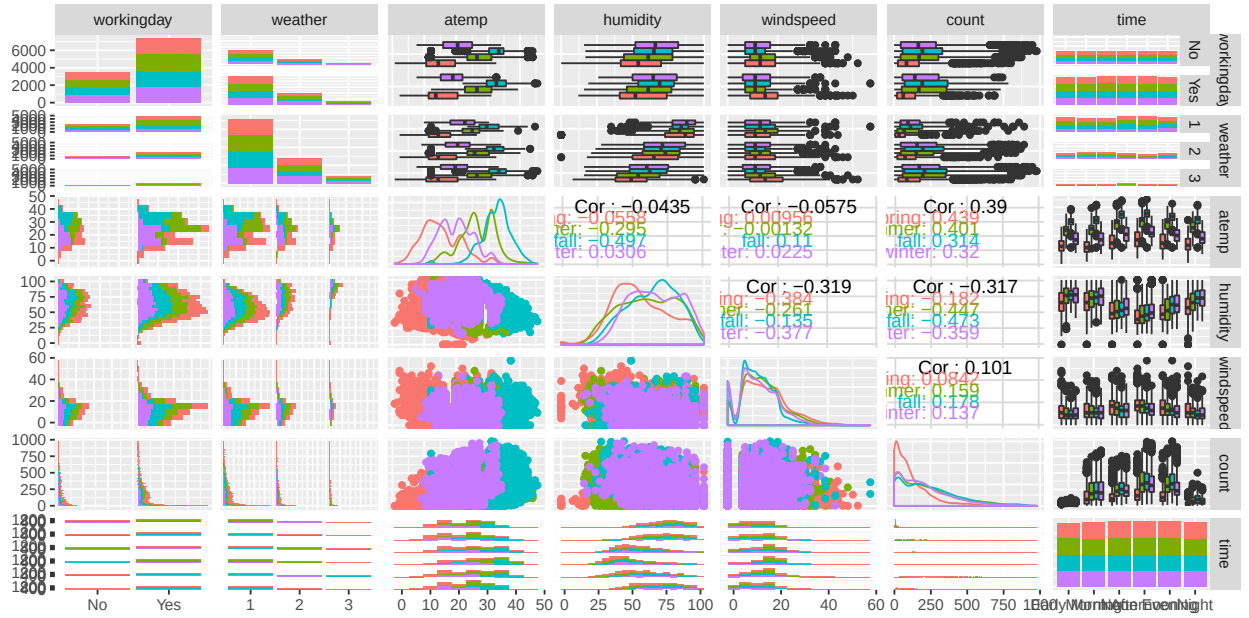
```
##                 datetime        season     holiday     workingday weather
##  2011-01-01 00:00:00:    1   spring:2686   No :10575   No :3474   1:7192
##  2011-01-01 01:00:00:    1   summer:2733   Yes:  311   Yes:7412   2:2834
##  2011-01-01 02:00:00:    1   fall  :2733                          3: 860
##  2011-01-01 03:00:00:    1   winter:2734                          3:   0
##  2011-01-01 04:00:00:    1
##  2011-01-01 05:00:00:    1
##  (Other)            :10880
##       temp           atemp          humidity        windspeed
##  Min.   : 0.82   Min.   : 0.76   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:13.94   1st Qu.:16.66   1st Qu.: 47.00   1st Qu.: 7.002
##  Median :20.50   Median :24.24   Median : 62.00   Median :12.998
##  Mean   :20.23   Mean   :23.66   Mean   : 61.89   Mean   :12.799
##  3rd Qu.:26.24   3rd Qu.:31.06   3rd Qu.: 77.00   3rd Qu.:16.998
##  Max.   :41.00   Max.   :45.45   Max.   :100.00   Max.   :56.997
##
##      casual         registered        count               time
##  Min.   :  0.00   Min.   :  0.0   Min.   :  1.0   Early Morn:1775
##  1st Qu.:  4.00   1st Qu.: 36.0   1st Qu.: 42.0   Morning   :1820
##  Median : 17.00   Median :118.0   Median :145.0   Noon      :1822
##  Mean   : 36.02   Mean   :155.6   Mean   :191.6   Afternoon :1824
##  3rd Qu.: 49.00   3rd Qu.:222.0   3rd Qu.:284.0   Evening   :1824
##  Max.   :367.00   Max.   :886.0   Max.   :977.0   Night     :1821
##
```

Bike rentals appear to be distributed evenly through the seasons, even during winter which is somewhat unexpected. Most of the rentals occur on nonholidays. Not surprisingly, there are also more rentals during the week, by mere number of days counted. Also as expected, days in which there were rentals are greater during conditions in which weather is pleasant, with large decreases as the weather conditions worsen. That one entry during which the weather is measured as "4", the most extreme, will likely be an outlier or cause issues with the regression analysis. Temperatures seem to be feel higher than the true temperature on average. The counts of rentals per time period is cannot be judged yet because the data is provided per hour entry, and the actual counts have not yet been taken into consideration.

Rather than analyze separate histograms for the numeric variables and barplots for the factor variables, it may be better to just analyze pairwise comparisons of the variables in a pairs plot. Below, I have excluded some variables, based on what I have observed in the summaries. Holiday is overwhelmingly weighted on the nonholiday dates, and it is difficult to even see how the distribution of the holiday dates, because of the relatively low counts. Temp and atemp explain similar things, and so I chose to keep atemp because it is closer to what the renter would likely feel. Finally, to simplify counts of rentals, I took only the total count, rather than separate casual and registered rentals.

Below are two pairs plots, categorized by season. As only entries that have nonzero rental counts are recorded, grouping by season may give some preliminary indication of any difference in rental habits between the seasons.

```
ggpairs(data= bike.train,
        aes(color = season),
        columns = c(4, 5, 7, 8, 9, 12, 13),
        lower=list(combo=wrap("facethist", binwidth=5)),
        diag=list(continuous=no_fill))
```




(Spring = pink, Summer = green, Fall = blue, Winter = purple)

In the diagonal are either the bar plots or density plots of the variables themselves, factors or numeric, respectively. The atemp variable shows the most distinct difference among the seasons, with the plot of fall rentals furthest right, followed then by summer, winter, and then spring. Humidity has a wide range, with a longer left tail. In contrast, windspeed has a narrower range, with a long right tail. For count of rentals, spring is the only season that shows a difference from the rest, heavily shifted left. Nothing new is really gleaned from the bar plots, which are almost equally divide between the seasons.

Somewhat unexpectedly, spring counts are lower than all of the other seasons. This could be attributed to the fact that this plot is counting the number of entries, rather than the sum of rentals. It also appears that fall is considerably warmer than spring, which is not that expected, since both seasons are between the extreme seasons of summer and winter, and would make more sense to have similar weather conditions. As for the overlaid plots, there is a lot of noise that makes it hard to make any informed observations, but for some, it is apparent that spring and fall have the greatest difference in values, whereas summer and winter have great overlap. For weather related histograms of atemp, humidity, and windspeed, it appers that winter has the widest distributions, whereas spring has the most narrow ones.

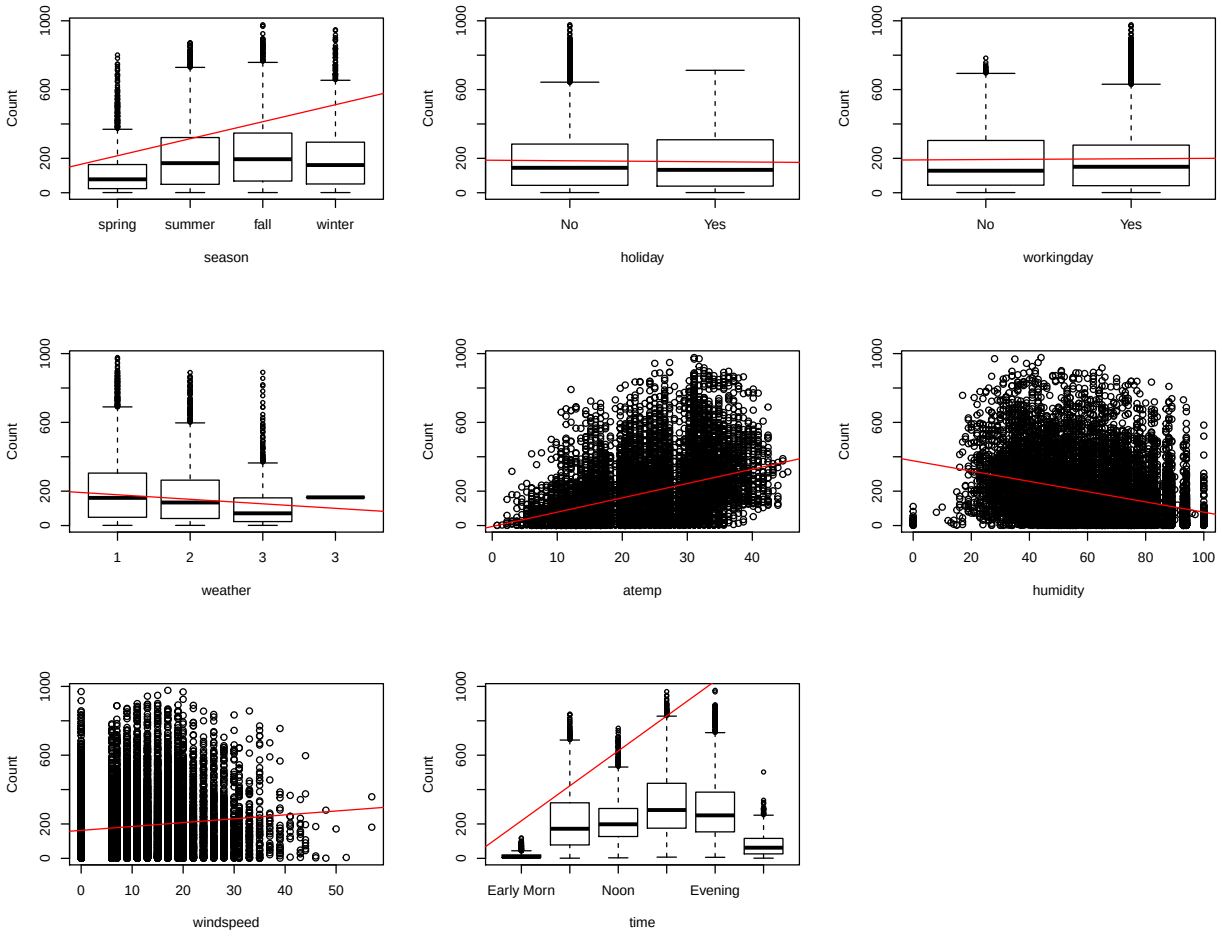
Regression Choice:

It seems like it would make more sense to just perform regression on the total count. From prelimary exploration of the variables casual and registered, they appear to have relatively similar distributions, so accuracy of the regression may be retained with the count variable as a summary of both. As counts is a numeric variable, rather than a factor, it does not make sense to perform logistic regression. Multiple regression is the better choice here. Furthermore, based on preliminary analysis, it is necessary to work with the log(count), or else the regression model gives negative values.

Regression Analysis:

First, the response variable will be plotted against each of the explanatory variables in order to identify general trends and potential outliers. As temp and atemp have a high correlation, I chose to only keep atemp in any further analysis, to simplify modeling. A very basic regression line is overlaid on each plot for each individual variable.

```
bike.trim <- bike.train[,c(12, 2:5, 7:9, 13)]
cols <- colnames(bike.trim)
par(mfrow = c(3, 3))
for (i in 2:9) {
  plot(bike.trim[, i], bike.trim[, 1], xlab = cols[i], ylab = "Count")
  abline(lm(paste("count", "~", cols[i]), data=bike.trim), col = "red")
}
```



Season appears to have distinct differences in its distributions of counts, so it likely has some importance in determining rental count. Holiday and workingday do not seem to really differ between the two categories. Weather and time also seems to important in count distribution, with the bulk of the data varying between each category, but the line is not very informative in the exact relation, because they are factor variables. For the numerical variables, atemp seems to have similar distributions, with a slight positive correlation. Humidity seems to have somewhat of a negative correlation. Windspeed is the odd case here. Though from the initial distribution, one would assume that the correlation would be clearly negative. However, this is not so, and perhaps there may be outliers that need to be addressed.

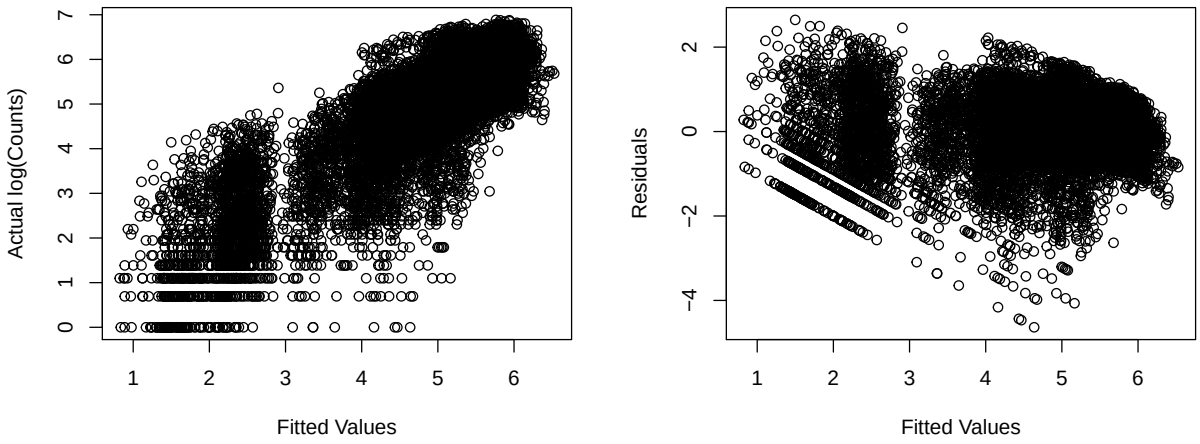
Basic fit of all the variables

```
bike_fit <- lm(log(count)~., data=bike.trim)
summary(bike_fit)
```

```
##
## Call:
## lm(formula = log(count) ~ ., data = bike.trim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6363 -0.4341  0.0242  0.5137  2.6444
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.6301139  0.0508639  32.049  < 2e-16 ***
## seasonsummer   0.3129925  0.0278781  11.227  < 2e-16 ***
## seasonfall     0.2430274  0.0344257   7.059 1.77e-12 ***
## seasonwinter   0.5667620  0.0234038  24.217  < 2e-16 ***
## holidayYes    -0.0778616  0.0473414  -1.645   0.1001
## workingdayYes -0.0838197  0.0169313  -4.951 7.51e-07 ***
## weather2      -0.0041062  0.0187795  -0.219   0.8269
## weather3      -0.4641348  0.0317275 -14.629  < 2e-16 ***
## weather3       0.6547984  0.7963418   0.822   0.4109
## atemp          0.0410230  0.0015240  26.918  < 2e-16 ***
## humidity      -0.0068957  0.0005274 -13.075  < 2e-16 ***
## windspeed     -0.0020536  0.0010126  -2.028   0.0426 *
## timeMorning    2.7219608  0.0266086 102.296  < 2e-16 ***
## timeNoon       2.7821706  0.0286593  97.077  < 2e-16 ***
## timeAfternoon  3.0366682  0.0303215 100.149  < 2e-16 ***
## timeEvening    3.0379022  0.0280371 108.353  < 2e-16 ***
## timeNight      1.6829254  0.0267433  62.929  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7957 on 10869 degrees of freedom
## Multiple R-squared:  0.7149, Adjusted R-squared:  0.7145
## F-statistic:  1703 on 16 and 10869 DF,  p-value: < 2.2e-16
```

There is not much that can be determined by this basic fit for now. The current R-squared value of this model is 0.7149. Below is the plot of the fitted values against the actual values, and then the residuals

```
par(mfrow=c(1, 2))
plot(bike_fit$fitted.values, log(bike.trim$count), xlab="Fitted Values", ylab="Actual log(Counts)")
plot(bike_fit$fitted.values, bike_fit$residuals, xlab="Fitted Values", ylab="Residuals")
```




It can be seen that this preliminary regression line is not too accurate. The accuracy appears to be greater for higher count values, but the variance between the two greatly increases as log(count) decreases. The residuals also show some correlation when it would be expected to be a massless cloud of points around 0 had the line fit well. Also, because of the skew of more extreme negative residuals, my current model must be predicting values considerably too low at a much greater rate.

```
r.sq.dif <- c()
base.r.sq <- summary(bike_fit)$r.squared
for(i in 2:9) {
  for(i2 in 2:9) {
    if(i < i2) {
      f <- paste("log(",cols[1],")", "~ . +", cols[i], ":", cols[i2])
      fit <- lm(f, data=bike.trim)
      if(summary(fit)$r.squared > base.r.sq + 0.002) {
        r.sq.dif = c(r.sq.dif, paste(cols[i], ":", cols[i2]))
      }
    }
  }
}
r.sq.dif
```

```
## [1] "season : atemp"    "workingday : time" "atemp : time"
```

Continuing to test the variables using log(count) as the response, I iterated through all possible pairs of the explanatory variables, and compared the multiple r squared values to the baseline from the original log(count) model. As it is likely that all additions of an interaction term will increase the r squared value to some degree, I set the threshold to be at least 0.002 greater than that found in the base model. As a result, 3 pairs are filtered out, which seems like a reasonable number out of the total possible 56 pairs.

```
f <- paste("log(",cols[1], ")", "~ . +")
for(pair in r.sq.dif) {
  f <- paste(f, pair, "+")
}
f <- str_sub(f, end=-3)
bike.fit.inter <- lm(f, data=bike.trim)
summary(bike.fit.inter)
```

```
##
## Call:
## lm(formula = f, data = bike.trim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6447 -0.4022  0.0150  0.4230  2.4039
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.8252794  0.0687697  26.542  < 2e-16 ***
## seasonsummer                0.5594102  0.0768550   7.279 3.60e-13 ***
## seasonfall                  1.3943289  0.1089864  12.794  < 2e-16 ***
## seasonwinter                0.9655091  0.0642567  15.026  < 2e-16 ***
## holidayYes                 -0.0446810  0.0415847  -1.074 0.282641
## workingdayYes              -0.7499006  0.0356227 -21.051  < 2e-16 ***
## weather2                   -0.0287182  0.0164856  -1.742 0.081534 .
## weather3                   -0.4728975  0.0278500 -16.980  < 2e-16 ***
## weather3                    0.7121731  0.6978182   1.021 0.307480
## atemp                       0.0596327  0.0029656  20.108  < 2e-16 ***
## humidity                   -0.0073247  0.0004700 -15.583  < 2e-16 ***
## windspeed                  -0.0030683  0.0008886  -3.453 0.000557 ***
## timeMorning                 1.3627158  0.0740067  18.413  < 2e-16 ***
## timeNoon                    2.9898982  0.0774138  38.622  < 2e-16 ***
```

```
## timeAfternoon                2.9412202  0.0809792  36.321  < 2e-16 ***
## timeEvening                  2.0722353  0.0781028  26.532  < 2e-16 ***
## timeNight                    1.3383075  0.0766894  17.451  < 2e-16 ***
## seasonsummer:atemp          -0.0168526  0.0033272  -5.065 4.15e-07 ***
## seasonfall:atemp            -0.0448450  0.0038348 -11.694  < 2e-16 ***
## seasonwinter:atemp          -0.0239060  0.0033829  -7.067 1.68e-12 ***
## workingdayYes:timeMorning    2.2316426  0.0498481  44.769  < 2e-16 ***
## workingdayYes:timeNoon       0.0798455  0.0498664   1.601 0.109364
## workingdayYes:timeAfternoon  0.4323272  0.0498418   8.674  < 2e-16 ***
## workingdayYes:timeEvening    1.1096778  0.0498368  22.266  < 2e-16 ***
## workingdayYes:timeNight      0.1369761  0.0498161   2.750 0.005976 **
## atemp:timeMorning           -0.0067361  0.0029180  -2.308 0.020993 *
## atemp:timeNoon              -0.0094318  0.0029042  -3.248 0.001167 **
## atemp:timeAfternoon         -0.0068815  0.0029766  -2.312 0.020804 *
## atemp:timeEvening            0.0088889  0.0029552   3.008 0.002637 **
## atemp:timeNight              0.0113436  0.0029899   3.794 0.000149 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6967 on 10856 degrees of freedom
## Multiple R-squared:  0.7817, Adjusted R-squared:  0.7811
## F-statistic:  1340 on 29 and 10856 DF,  p-value: < 2.2e-16
```

Though many additional variables are now included, this model has a higher multiple R-squared value, 0.7817, which is greater by approximately .07, indicating that the accuracy of this model is superior to the base model.

*A note: I tested several threshold values (0.001, 0.002, 0.003), and resulted in a total of 45, 29, and 26 variables respectively. Though a threshold of 0.002 does, unfortunately, result in a whole 13 extra variables, the jump in r-squared compared to a threshold of 0.003 was large enough to consider including them, nonetheless.

Variable Selection:

```
step(bike.fit.inter, direction="both")
```

```
## Start:  AIC=-7838.6
## log(count) ~ season + holiday + workingday + weather + atemp +
##     humidity + windspeed + time + season:atemp + workingday:time +
##     atemp:time
##
##                    Df Sum of Sq    RSS     AIC
## - holiday           1      0.56 5269.9 -7839.4
## <none>                          5269.3 -7838.6
## - windspeed         1      5.79 5275.1 -7828.6
## - atemp:time        5     47.69 5317.0 -7750.5
## - season:atemp      3     70.74 5340.1 -7699.4
## - humidity          1    117.87 5387.2 -7599.8
## - weather           3    145.79 5415.1 -7547.5
## - workingday:time   5   1484.15 6753.5 -5147.2
##
## Step:  AIC=-7839.44
## log(count) ~ season + workingday + weather + atemp + humidity +
##     windspeed + time + season:atemp + workingday:time + atemp:time
##
##                    Df Sum of Sq    RSS     AIC
## <none>                          5269.9 -7839.4
## + holiday           1      0.56 5269.3 -7838.6
## - windspeed         1      5.85 5275.7 -7829.4
## - atemp:time        5     47.62 5317.5 -7751.5
## - season:atemp      3     71.89 5341.8 -7697.9
## - humidity          1    117.82 5387.7 -7600.7
## - weather           3    145.62 5415.5 -7548.7
## - workingday:time   5   1484.25 6754.1 -5148.1
##
## Call:
## lm(formula = log(count) ~ season + workingday + weather + atemp +
##     humidity + windspeed + time + season:atemp + workingday:time +
##     atemp:time, data = bike.trim)
##
## Coefficients:
##                  (Intercept)                 seasonsummer
##                     1.819512                     0.562065
##                   seasonfall                 seasonwinter
##                     1.398602                     0.969812
##                workingdayYes                     weather2
##                    -0.745874                    -0.028989
##                     weather3                     weather3
##                    -0.472677                     0.712407
##                        atemp                     humidity
##                     0.059766                    -0.007323
##                    windspeed                  timeMorning
##                    -0.003085                     1.362564
##                     timeNoon                timeAfternoon
##                     2.989611                     2.940651
##                  timeEvening                    timeNight
```

```
##                     2.072070                     1.338301
##          seasonsummer:atemp             seasonfall:atemp
##                    -0.017003                    -0.045068
##          seasonwinter:atemp    workingdayYes:timeMorning
##                    -0.024177                     2.231704
##      workingdayYes:timeNoon  workingdayYes:timeAfternoon
##                     0.079820                     0.432265
##   workingdayYes:timeEvening      workingdayYes:timeNight
##                     1.109634                     0.136962
##           atemp:timeMorning               atemp:timeNoon
##                    -0.006728                    -0.009409
##         atemp:timeAfternoon            atemp:timeEvening
##                    -0.006848                     0.008903
##             atemp:timeNight
##                     0.011346
```

I used the step() function to reduce the variables in the model. Only the variable holiday was taken out in the result.

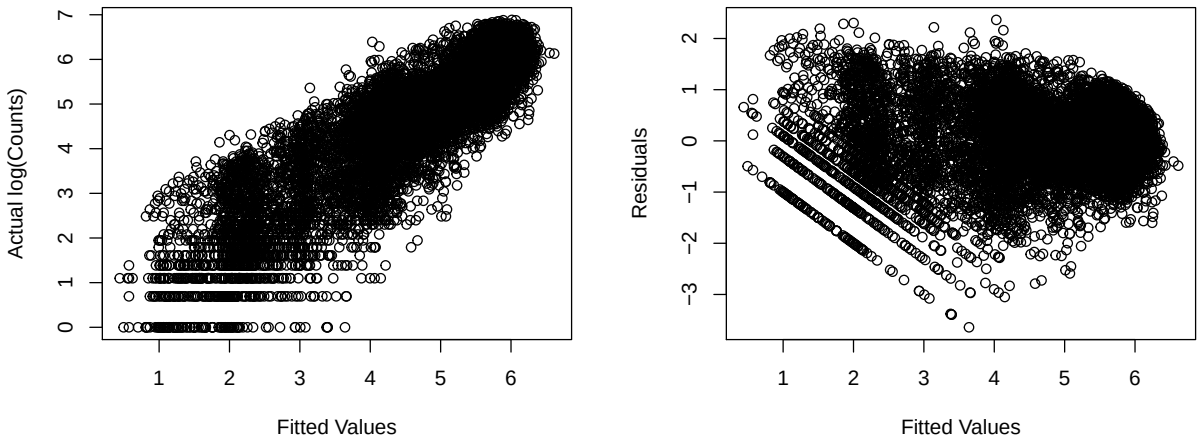
The final model is log(count) ~ atemp + time + humidity + weather + atemp:season + time:workingday + time:atemp, a moderately sized model with 7 variables(including the interaction terms).

```
bike.step.fit <- lm(formula = log(count) ~ season + workingday + weather + atemp +
    humidity + windspeed + time + season:atemp + workingday:time +
    atemp:time, data = bike.trim)
summary(bike.step.fit)
```

```
##
## Call:
## lm(formula = log(count) ~ season + workingday + weather + atemp +
##     humidity + windspeed + time + season:atemp + workingday:time +
##     atemp:time, data = bike.trim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6413 -0.4028  0.0148  0.4235  2.3630
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.8195121  0.0685604  26.539  < 2e-16 ***
## seasonsummer                 0.5620651  0.0768159   7.317 2.71e-13 ***
## seasonfall                   1.3986023  0.1089146  12.841  < 2e-16 ***
## seasonwinter                 0.9698125  0.0641323  15.122  < 2e-16 ***
## workingdayYes               -0.7458735  0.0354252 -21.055  < 2e-16 ***
## weather2                    -0.0289889  0.0164838  -1.759 0.078668 .
## weather3                    -0.4726768  0.0278494 -16.973  < 2e-16 ***
## weather3                     0.7124071  0.6978231   1.021 0.307325
## atemp                        0.0597659  0.0029630  20.171  < 2e-16 ***
## humidity                    -0.0073230  0.0004700 -15.580  < 2e-16 ***
## windspeed                   -0.0030846  0.0008885  -3.472 0.000519 ***
## timeMorning                  1.3625644  0.0740071  18.411  < 2e-16 ***
## timeNoon                     2.9896109  0.0774139  38.619  < 2e-16 ***
## timeAfternoon                2.9406515  0.0809780  36.314  < 2e-16 ***
## timeEvening                  2.0720699  0.0781032  26.530  < 2e-16 ***
## timeNight                    1.3383005  0.0766900  17.451  < 2e-16 ***
## seasonsummer:atemp          -0.0170033  0.0033243  -5.115 3.19e-07 ***
```

```
## seasonfall:atemp            -0.0450681  0.0038292 -11.770  < 2e-16 ***
## seasonwinter:atemp          -0.0241771  0.0033735  -7.167 8.18e-13 ***
## workingdayYes:timeMorning    2.2317036  0.0498484  44.770  < 2e-16 ***
## workingdayYes:timeNoon       0.0798201  0.0498667   1.601 0.109479
## workingdayYes:timeAfternoon  0.4322653  0.0498421   8.673  < 2e-16 ***
## workingdayYes:timeEvening    1.1096341  0.0498371  22.265  < 2e-16 ***
## workingdayYes:timeNight      0.1369619  0.0498164   2.749 0.005982 **
## atemp:timeMorning           -0.0067277  0.0029180  -2.306 0.021155 *
## atemp:timeNoon              -0.0094086  0.0029041  -3.240 0.001200 **
## atemp:timeAfternoon         -0.0068483  0.0029765  -2.301 0.021420 *
## atemp:timeEvening            0.0089025  0.0029552   3.012 0.002597 **
## atemp:timeNight              0.0113460  0.0029900   3.795 0.000149 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6967 on 10857 degrees of freedom
## Multiple R-squared:  0.7816, Adjusted R-squared:  0.7811
## F-statistic:  1388 on 28 and 10857 DF,  p-value: < 2.2e-16
```

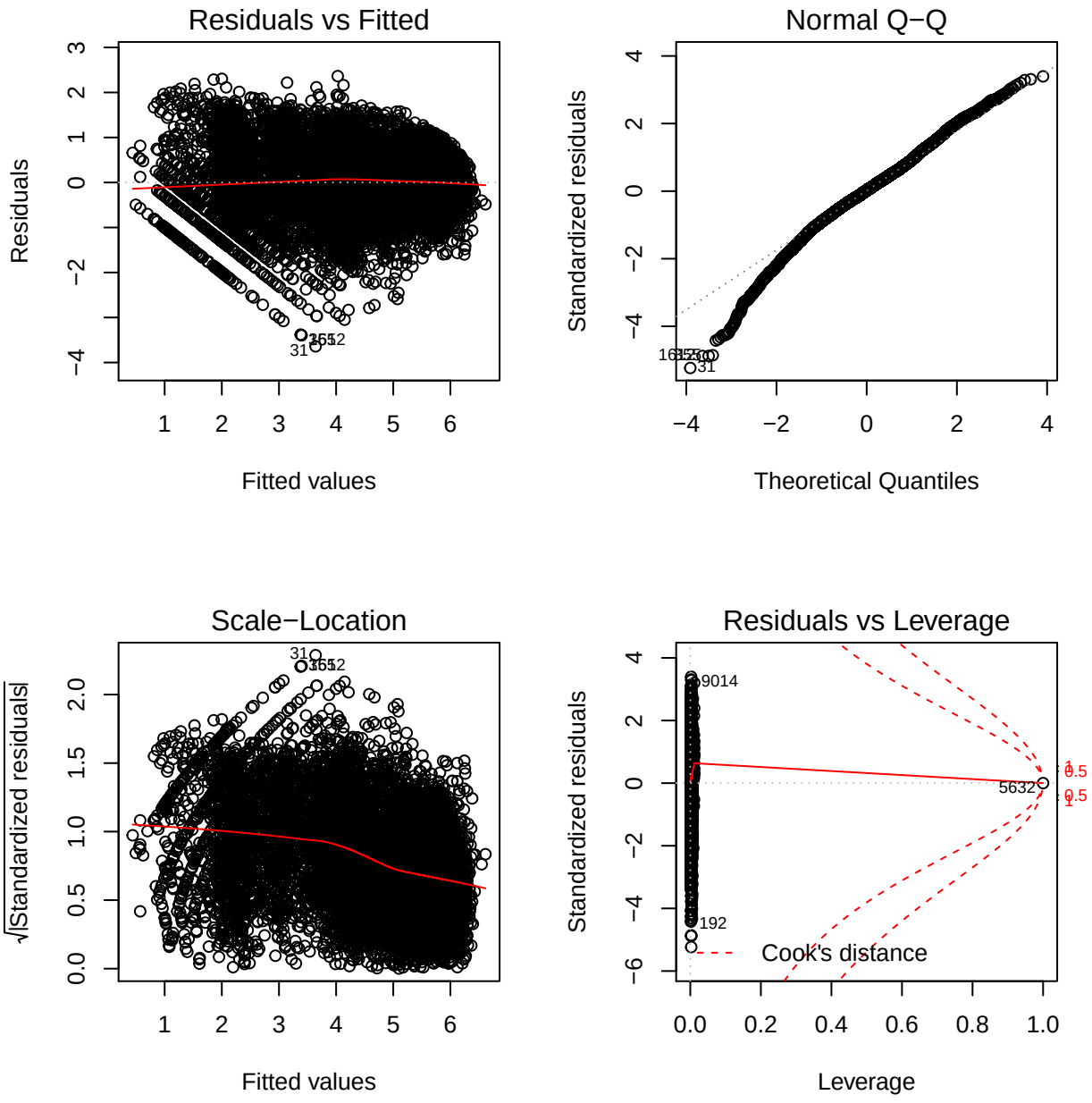
```
par(mfrow=c(1, 2))
plot(bike.step.fit$fitted.values, log(bike.trim$count), xlab="Fitted Values", ylab="Actual log(Counts)")
plot(bike.step.fit$fitted.values, bike.step.fit$residuals, xlab="Fitted Values", ylab="Residuals")
```




Once again plotting the fitted values against the actual values and the residuals, we can see that there isn't much change in the patterns seen in the fitted versus the actual values. However, the range of negative residuals has decreased and it appears they may be a bit more balanced around 0, which is an improvement over the previous model.

Regression Diagnostics:

```
par(mfrow = c(2, 2))
plot(bike.step.fit)
```




```
# bike.trim[5632, ]
```

From the first plot of the residuals against the fitted, it is somewhat obvious that the relation between the variables is not precisely linear for smller values, but is reasonably modeled for larger counts. Rather than a shapeless cloud, the variance of the points around the 0 line seems to decrease as the values increase. This is also seen somewhat in the Q-Q plot, whose points are relatively close to the line except for the left end of the graph, where the normal assumption may not hold. THe scale-location plot, since it is representative of

over 10000 values, may not necessarily show a visible pattern among the points. However, there is a slight decrease in the range of points as the value of the fitted numbers increase, and the red overlaid line slopes downward. This is also likely a sign of heteroscedascity. Finally, for the residuals vs leverage plot, not many outliers are flagged in the plot, perhaps due to the density of the data points. However, it can be seen that there is a greater range in the lesser values of the plot relative to the more extreme ones.

*The one extreme outlier flagged here, entry 5632, is a special case. After observing this outlier separately, and noting the weather category as "4", something I noted earlier, I attempted to remove this point and rerun regression analysis. However, this did not ultimately work out, because in subsequent prediction with my model, I require at least one data point of this factor level to even predict other entries in the test set. Therefore, I kept this outlier in my data.

Predict:

```
pred = exp(predict(bike.step.fit, bike.test)) # retransformation
subm = data.frame(datetime = bike.test$datetime, count = pred)
write.csv(subm, file = "Subm.csv", row.names = FALSE)
```

Score given by Kaggle: 0.66162

Comparison:

My score was still significantly worse than those in the Kaggle public leaderboards. Perhaps ways to increase this score would be to increase the number of interaction terms between the variables, or to not do that initial trimming of variables. Also, perhaps I could take the datetime variable and also create a new variable corresponding to specific months as well as make the corresponding hour time periods different factors for each hour, rather than grouping them together. If I had more time, I would also look for some way to deal with that outlier in the weather group 4 entry, and also remove some other outliers within the data to obtain a clearer relation between the explanatory variables and the count. Finally, I don't believe the relation between the variables is exactly linear, because of the lack of accuracy in the lower end of the scale. I believe it may be similar to an exponential distribution, or quadratic, but these are just hypotheses that require more testing.