Regression

Fernando Colman and Linus Fackler

CS 4375.003 Linear Models Project

Linear regression is a method that utilizes x variables (predictors) and y values (targets) such that their relationship can be modeled using the linear equation y = mx * b. The goal of linear regression is to find a line of "best-fit" which can be used to predict future y values given new x values. An advantage of linear regression is that you can also do polynomial linear regression to make models that are not necessarily linear equations. A disadvantage of linear regression is that it's a method with a lot of bias, meaning it wants to find a linear model even if it doesn't really fit the model.

Dataset Citation: Fanaee-T, Hadi. (2013). Bike Sharing Dataset. UCI Machine Learning Repository.

A. Loading Data and Splitting into Test/Train

```
bikes <- read.csv("hour.csv", header=TRUE)
set.seed(1234)
i <- sample(1:nrow(bikes), .8*nrow(bikes), replace=FALSE)
train <- bikes[i,]
test <- bikes[-i,]
```

B. Use R Functions for Data Exploration on Training Data

```
str(train)
```

```
## 'data.frame':    13903 obs. of  17 variables:
##  $ instant   : int  7452 8016 7162 8086 9196 623 15241 10885 934 12688 ...
##  $ dteday    : chr  "2011-11-12" "2011-12-05" "2011-10-31" "2011-12-08" ...
##  $ season    : int  4 4 4 4 1 1 4 2 1 2 ...
##  $ yr        : int  0 0 0 0 1 0 1 1 0 1 ...
##  $ mnth      : int  11 12 10 12 1 1 10 4 2 6 ...
##  $ hr        : int  2 15 0 13 1 4 5 16 12 20 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 1 1 4 2 6 2 2 5 0 ...
##  $ workingday: int  0 1 1 1 1 0 1 1 1 0 ...
##  $ weathersit: int  1 2 1 1 2 1 2 1 1 1 ...
##  $ temp      : num  0.24 0.46 0.26 0.3 0.32 0.16 0.56 0.62 0.22 0.62 ...
##  $ atemp     : num  0.258 0.455 0.303 0.273 0.303 ...
##  $ hum       : num  0.65 0.72 0.87 0.49 0.93 0.69 0.83 0.21 0.47 0.57 ...
##  $ windspeed : num  0.0896 0.0896 0 0.3582 0.2537 ...
##  $ casual    : int  7 16 3 9 2 1 1 145 7 101 ...
##  $ registered: int  39 132 20 115 7 2 42 340 64 201 ...
##  $ cnt       : int  46 148 23 124 9 3 43 485 71 302 ...
```

```
names(train)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"
```

```
##  [6] "hr"         "holiday"    "weekday"    "workingday" "weathersit"
## [11] "temp"       "atemp"      "hum"        "windspeed"  "casual"
## [16] "registered" "cnt"
```

```
nrow(train)
```

```
## [1] 13903
```

```
ncol(train)
```

```
## [1] 17
```

```
colSums(is.na(train))
```

```
##    instant     dteday     season         yr       mnth         hr    holiday
##          0          0          0          0          0          0          0
##    weekday workingday weathersit       temp      atemp        hum  windspeed
##          0          0          0          0          0          0          0
##     casual registered        cnt
##          0          0          0
```

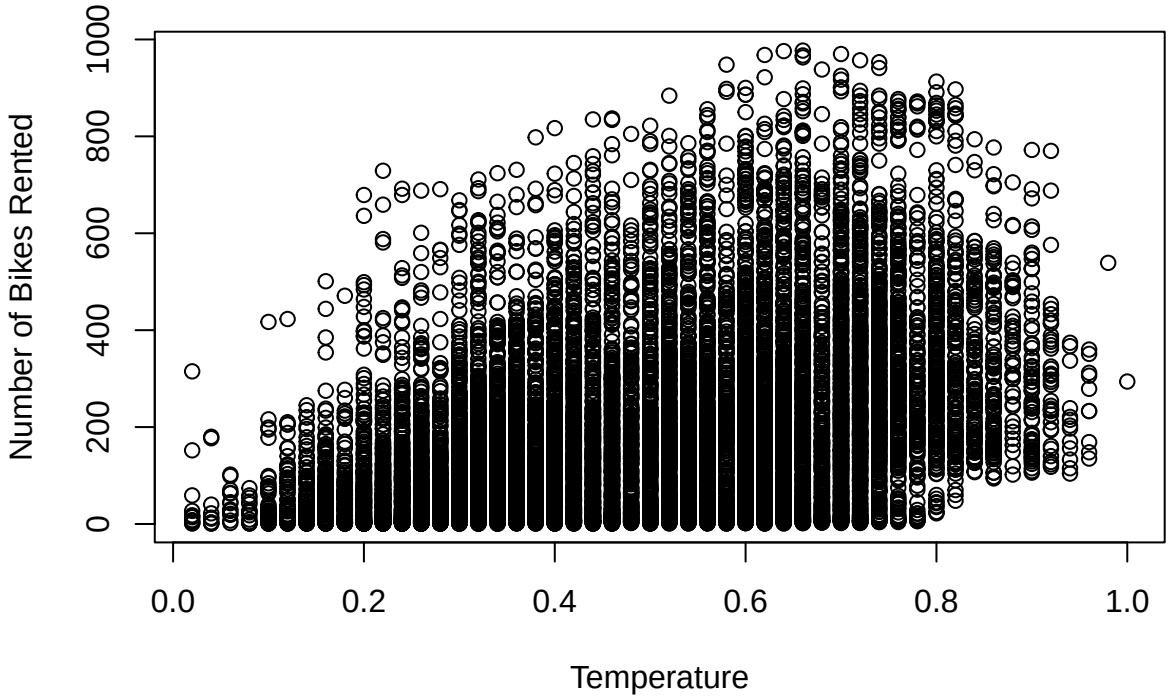
```
summary(train)
```

```
##     instant         dteday              season           yr
##  Min.   :    1   Length:13903       Min.   :1.000   Min.   :0.0000
##  1st Qu.: 4386   Class :character   1st Qu.:2.000   1st Qu.:0.0000
##  Median : 8768   Mode  :character   Median :3.000   Median :1.0000
##  Mean   : 8730                      Mean   :2.504   Mean   :0.5064
##  3rd Qu.:13054                      3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :17379                      Max.   :4.000   Max.   :1.0000
##       mnth              hr           holiday           weekday
##  Min.   : 1.000   Min.   : 0.00   Min.   :0.00000   Min.   :0.000
##  1st Qu.: 4.000   1st Qu.: 6.00   1st Qu.:0.00000   1st Qu.:1.000
##  Median : 7.000   Median :12.00   Median :0.00000   Median :3.000
##  Mean   : 6.549   Mean   :11.55   Mean   :0.02877   Mean   :3.011
##  3rd Qu.:10.000   3rd Qu.:18.00   3rd Qu.:0.00000   3rd Qu.:5.000
##  Max.   :12.000   Max.   :23.00   Max.   :1.00000   Max.   :6.000
##    workingday       weathersit         temp            atemp
##  Min.   :0.0000   Min.   :1.000   Min.   :0.0200   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.3400   1st Qu.:0.3333
##  Median :1.0000   Median :1.000   Median :0.5000   Median :0.4848
##  Mean   :0.6821   Mean   :1.421   Mean   :0.4972   Mean   :0.4760
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:0.6600   3rd Qu.:0.6212
##  Max.   :1.0000   Max.   :4.000   Max.   :1.0000   Max.   :0.9848
##       hum           windspeed          casual         registered
##  Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   :  0.0
##  1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.:  4.00   1st Qu.: 35.0
##  Median :0.6300   Median :0.1940   Median : 17.00   Median :117.0
##  Mean   :0.6265   Mean   :0.1896   Mean   : 35.89   Mean   :154.9
##  3rd Qu.:0.7800   3rd Qu.:0.2537   3rd Qu.: 49.00   3rd Qu.:221.0
##  Max.   :1.0000   Max.   :0.8507   Max.   :367.00   Max.   :886.0
##       cnt
##  Min.   :  1.0
##  1st Qu.: 41.0
##  Median :144.0
##  Mean   :190.8
##  3rd Qu.:282.0
```

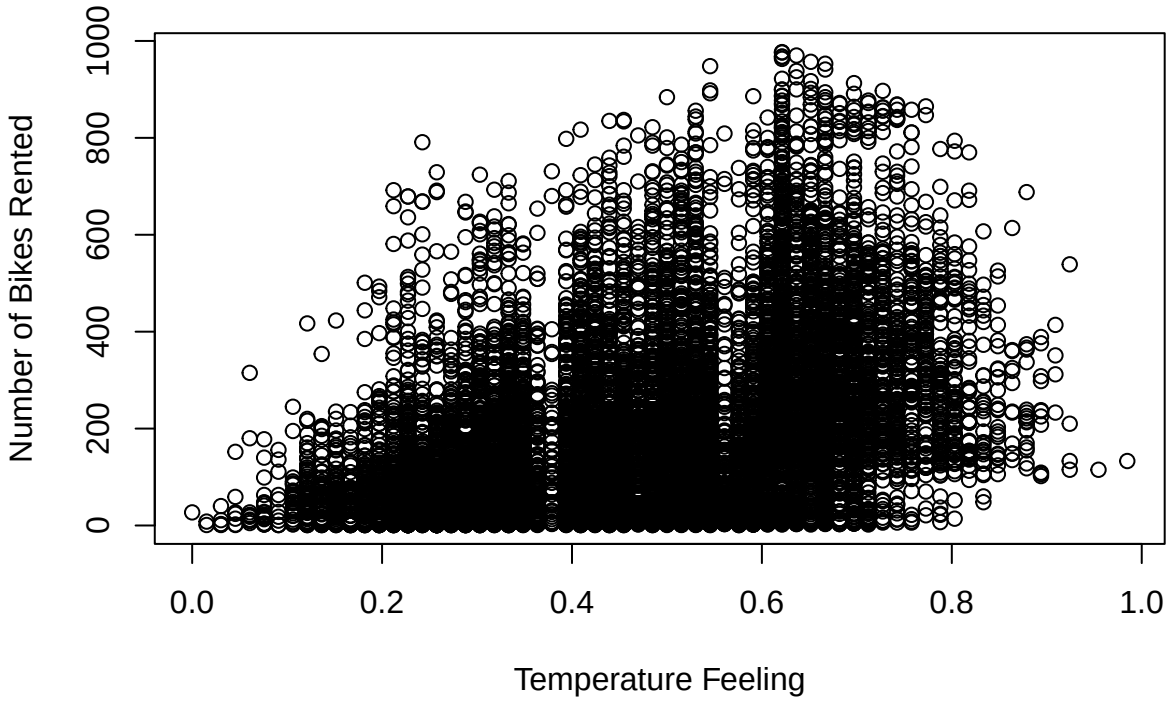
```
##  Max.   :977.0
```

C. Use Training Data to Build Informative Graphs

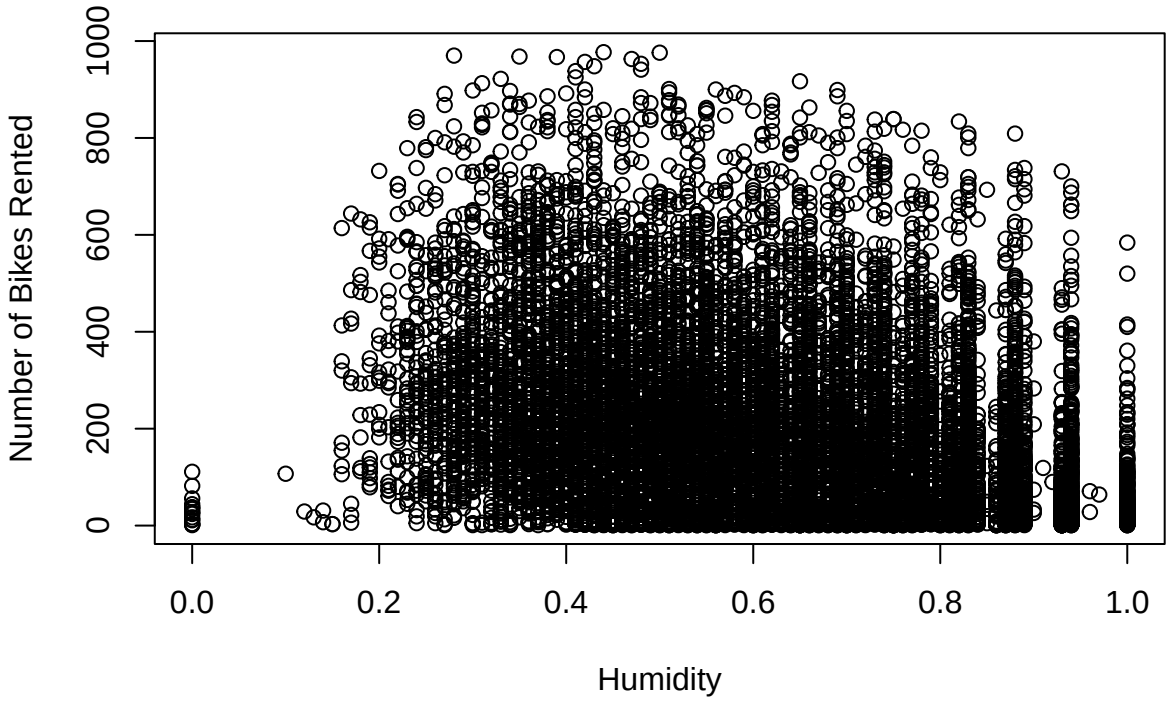
```
plot(train$temp, train$cnt, xlab = "Temperature", ylab = "Number of Bikes Rented")
```



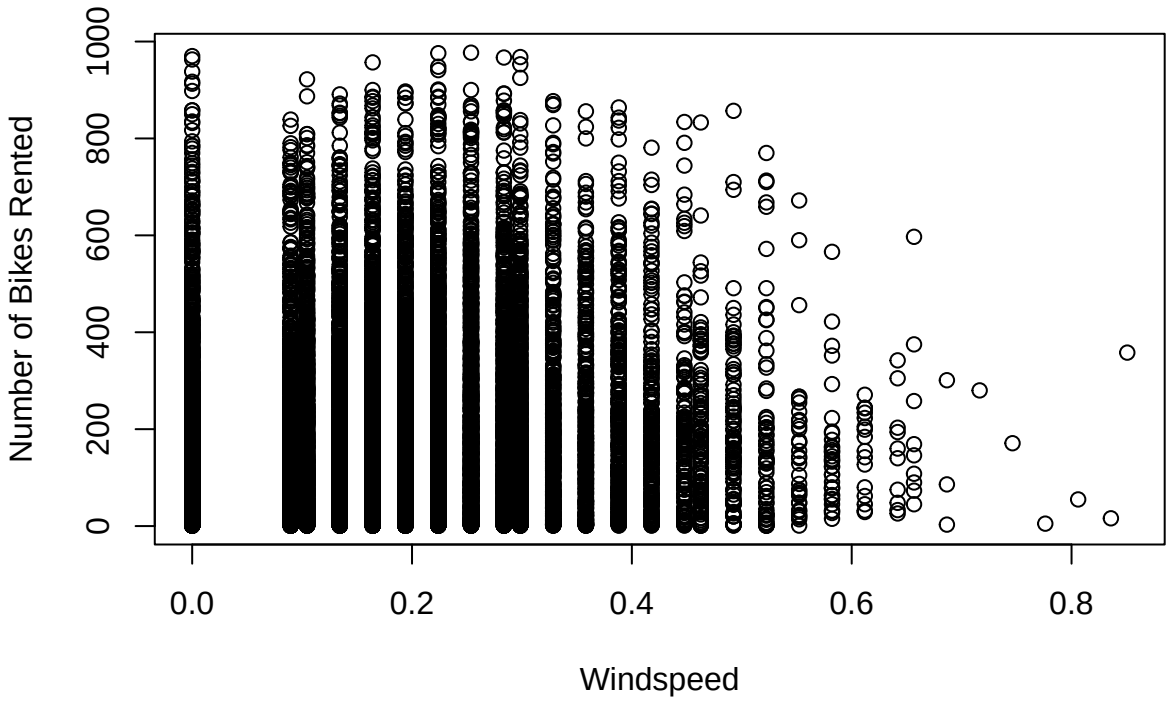

```
plot(train$atemp, train$cnt, xlab = "Temperature Feeling", ylab = "Number of Bikes Rented")
```



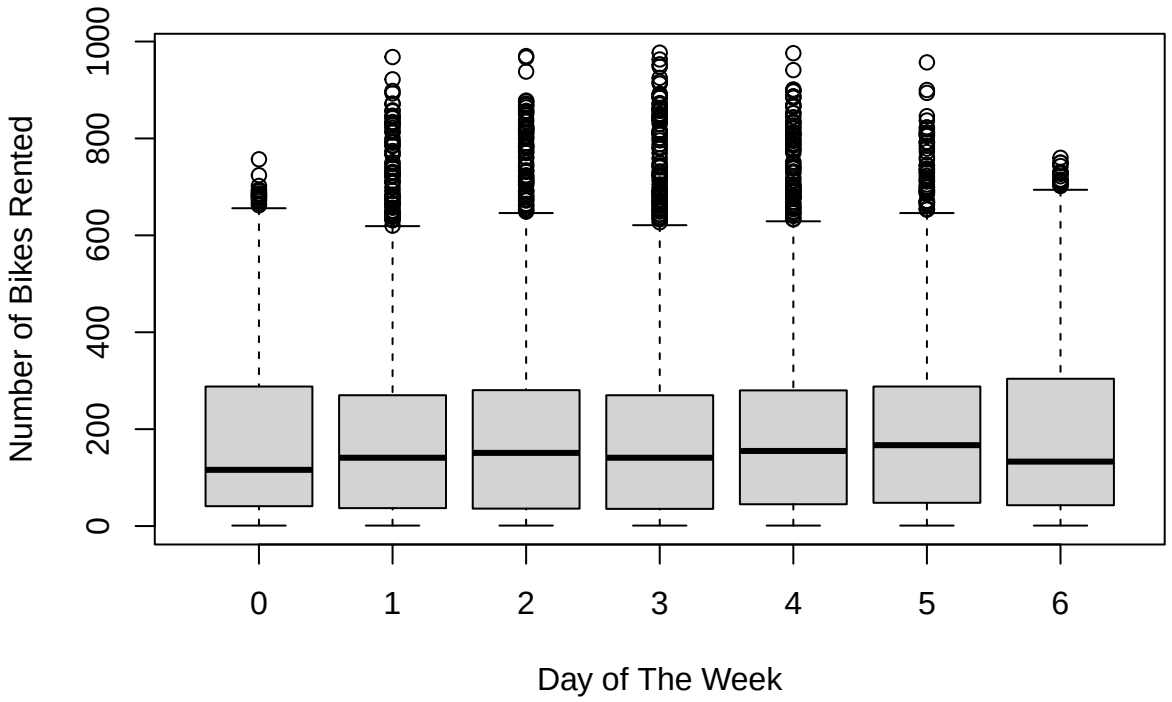
```
plot(train$hum, train$cnt, xlab = "Humidity", ylab = "Number of Bikes Rented")
```




```
plot(train$windspeed, train$cnt, xlab = "Windspeed", ylab = "Number of Bikes Rented")
```




```
boxplot(cnt~weekday, data=train, xlab = "Day of The Week", ylab = "Number of Bikes Rented")
```



D. Build a Simple Linear Regression Model using Training Data

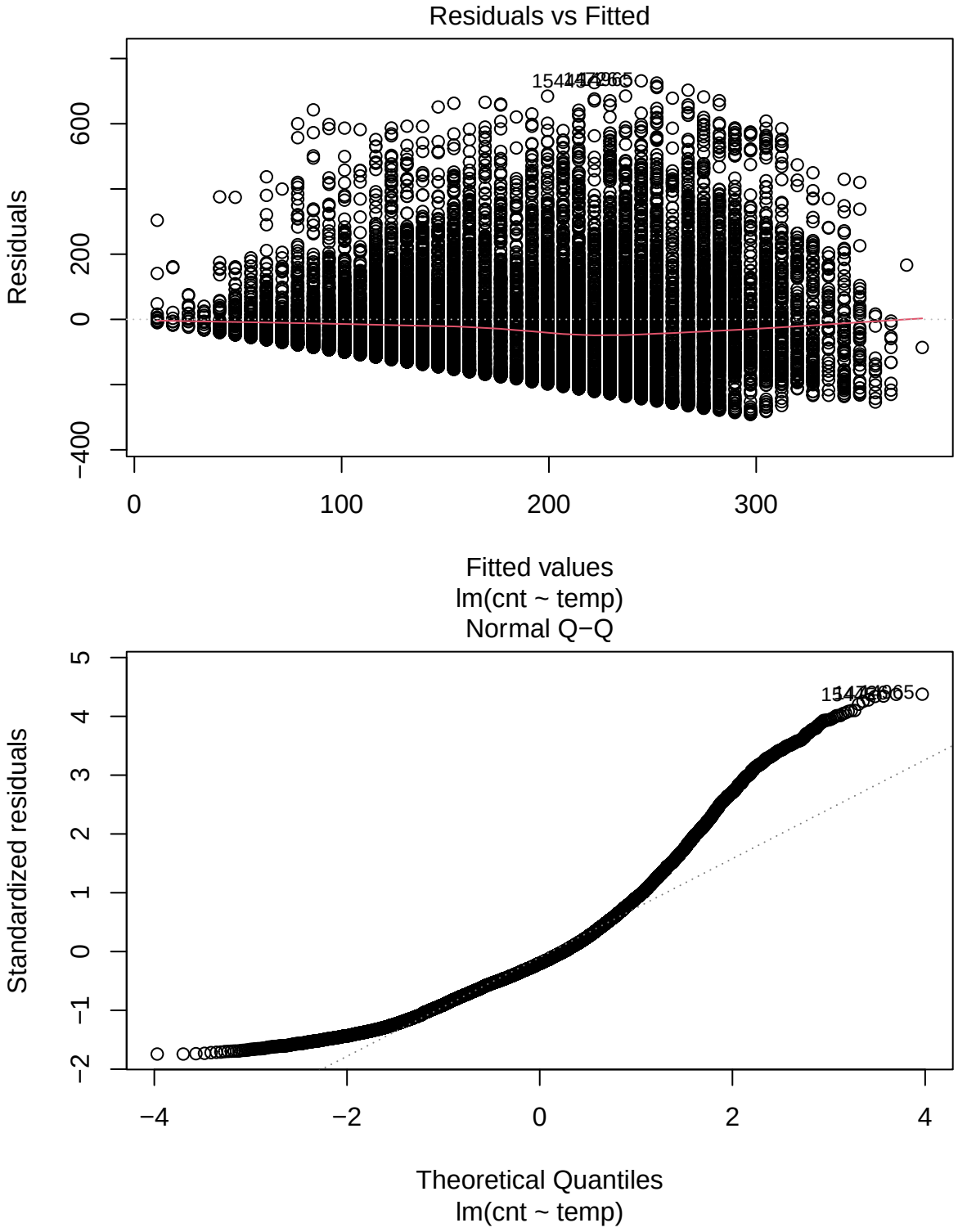
```
lm1 <- lm(cnt~temp, data=train)
summary(lm1)
```

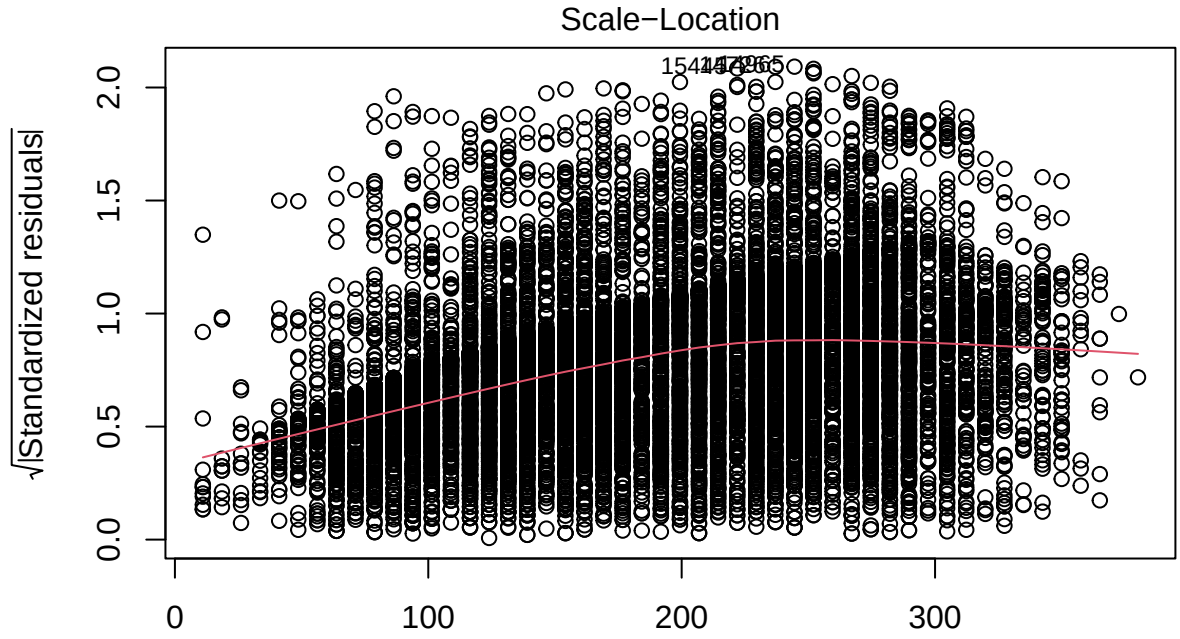
```
##
## Call:
## lm(formula = cnt ~ temp, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -291.25 -111.50  -33.73   77.94  731.48
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.495      3.934   0.888    0.374
## temp         376.607      7.381  51.023   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 167.1 on 13901 degrees of freedom
## Multiple R-squared:  0.1577, Adjusted R-squared:  0.1577
## F-statistic:  2603 on 1 and 13901 DF,  p-value: < 2.2e-16
```

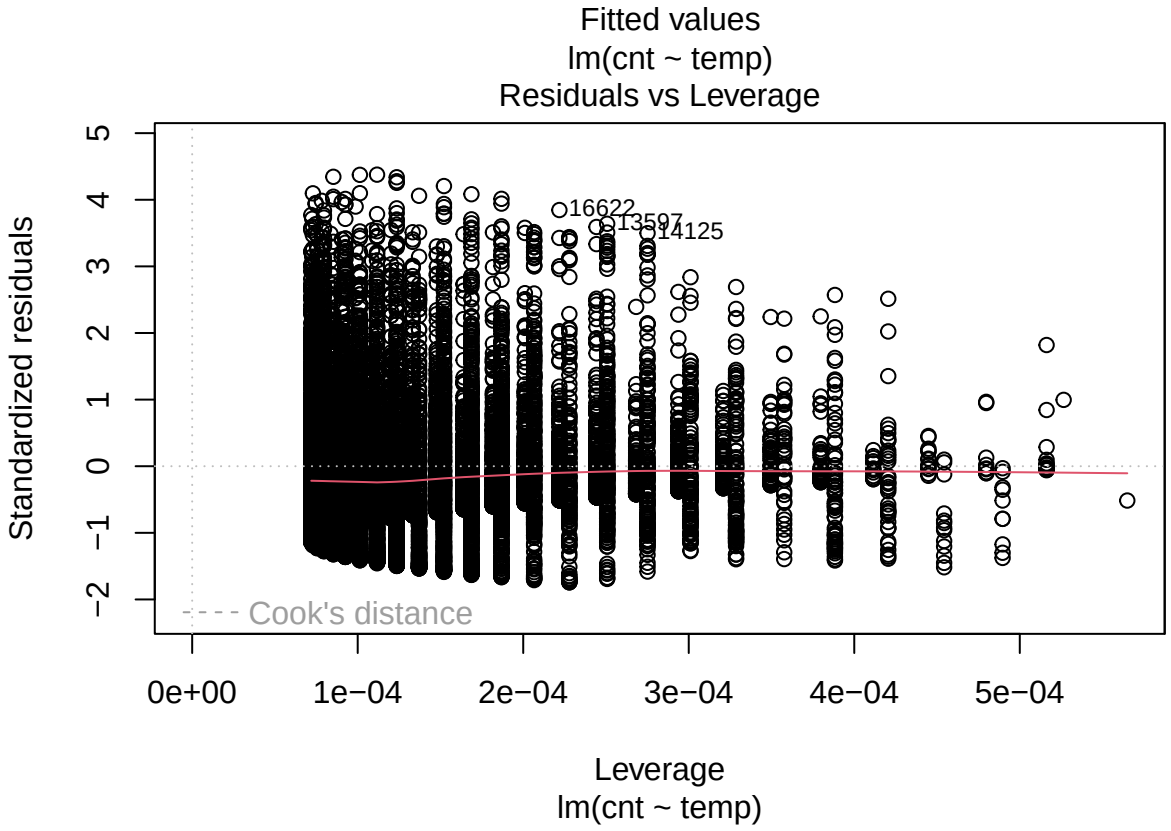
We created a linear regression model to see the effect that temperature of a particular day might have on the amount of bikes rented on that day. The model shows that the Residual Standard Error is 167.1 and the R-Squared value is at 0.1577, meaning there is not a lot of correlation between temperature and the amount of bikes rented. Ideally, we would want to see an R-Squared value closer to 1.0 which would indicate a strong correlation between temperature and bikes rented.

E. Plot the Residuals of the Linear Regression Model

```
plot(lm1)
```







1. The first plot of "Residuals vs Fitted Value" is supposed to indicate whether the predictors and target potentially have a non-linear relationship. For example, a horizontal line with equally spread residuals means that there's not likely to be a non-linear relationship. However, our graph shows that while there is a fairly horizontal line there is a lot of variety on the spread of the residuals around that line. This likely indicates that while there isn't a non-linear relationship that isn't being displayed in the model, the strength of that linear relationship that is there probably isn't very strong. I would argue that this

plot shows that when temperatures are at their extremes then temperature is a great predictor for bikes rented, but when temperatures are in a normal range then the relationship is a lot weaker.

2. The second plot of "Normal Q-Q" shows whether residuals are normally distributed as shows by normal line which is dashed and the actual values which are in black. This plot actually backs up my hypothesis from the previous plot. It shows that when temperatures are in a normal range, meaning the middle of the data, then the residuals are normally distributed which is good. However, when the temperatures start reaching their extremes like in the start and end of this plot then the residuals are not so normally distributed and therefore being influenced by other factors.

3. The "Scale Location" plot shows if residuals are spread evenly among the ranges of the predictors, which in this case is just temperature. To show that the residuals are in fact spread equally then we would want to see a horizontal line with randomly spread points around it. However, this is not the case for our graph which shows a clear incline and later decline slope with a lot more spread out points in the middle of the data as opposed to the extremes. This plot continues to provide evidence for the hypothesis that I put forth since the first graph that temperature becomes a worst predictor once it enters the really hot or really cold ranges.

4. The last plot of "Residuals vs Leverage" is meant to show how impactful specific outliers are to our linear regression model. This graph isn't read by looking at patterns but rather by looking at what are known as "Cook Distances" which are shown in the graph by a red dashed line. If there are any data points which are outside of the Cook's line then it means that not only is that point an outlier but it also having a very large impact on the model. Thankfully, our model doesn't even show the Cook's line because our data points have such low Cook's distances. This is probably due to the sheer amount of observations that the data has but it is also good news since that means that no single outlier is having a large negative influence on our model.

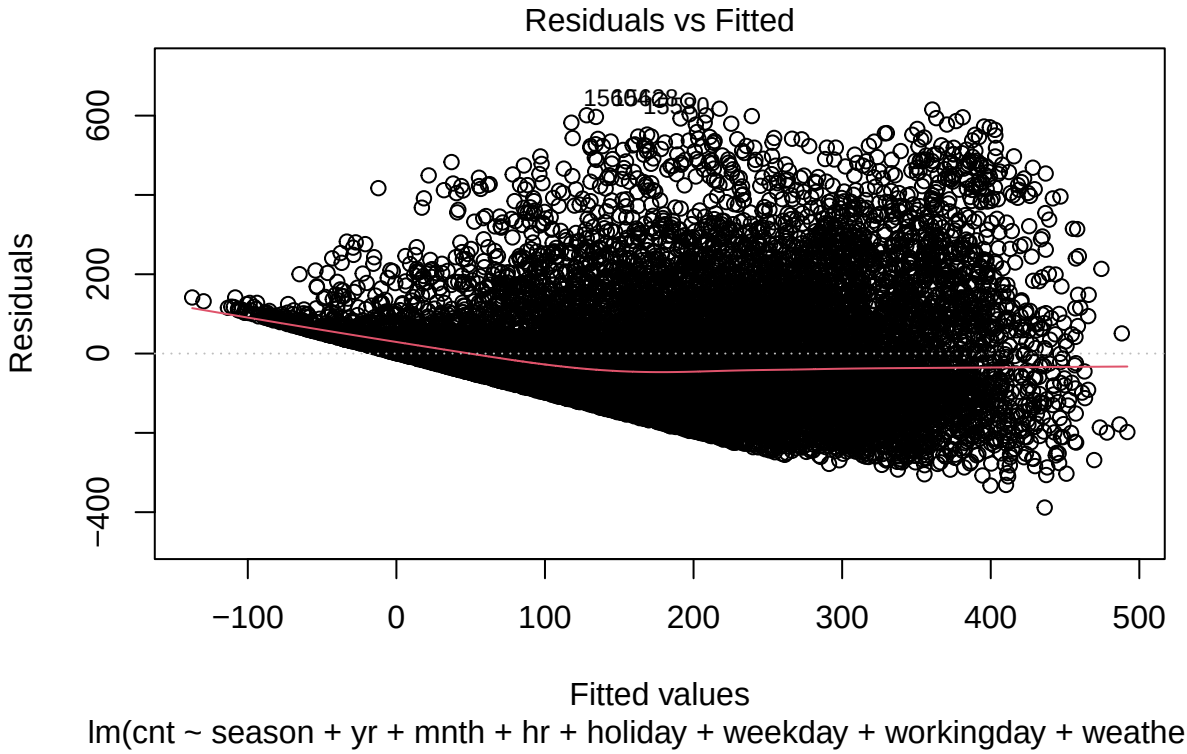
F. Build a Multiple Linear Regression Model

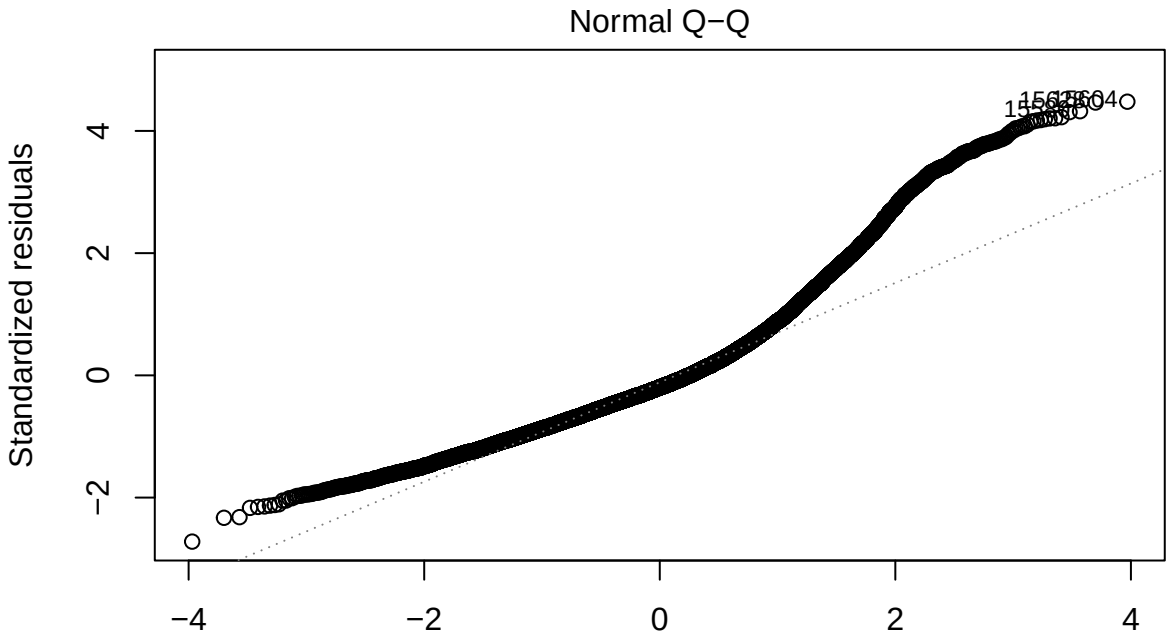
```
lm2 <- lm(cnt~season+yr+mnth+hr+holiday+weekday+workingday+weathersit+temp+atemp+hum+windspeed, data=tra
summary(lm2)
```

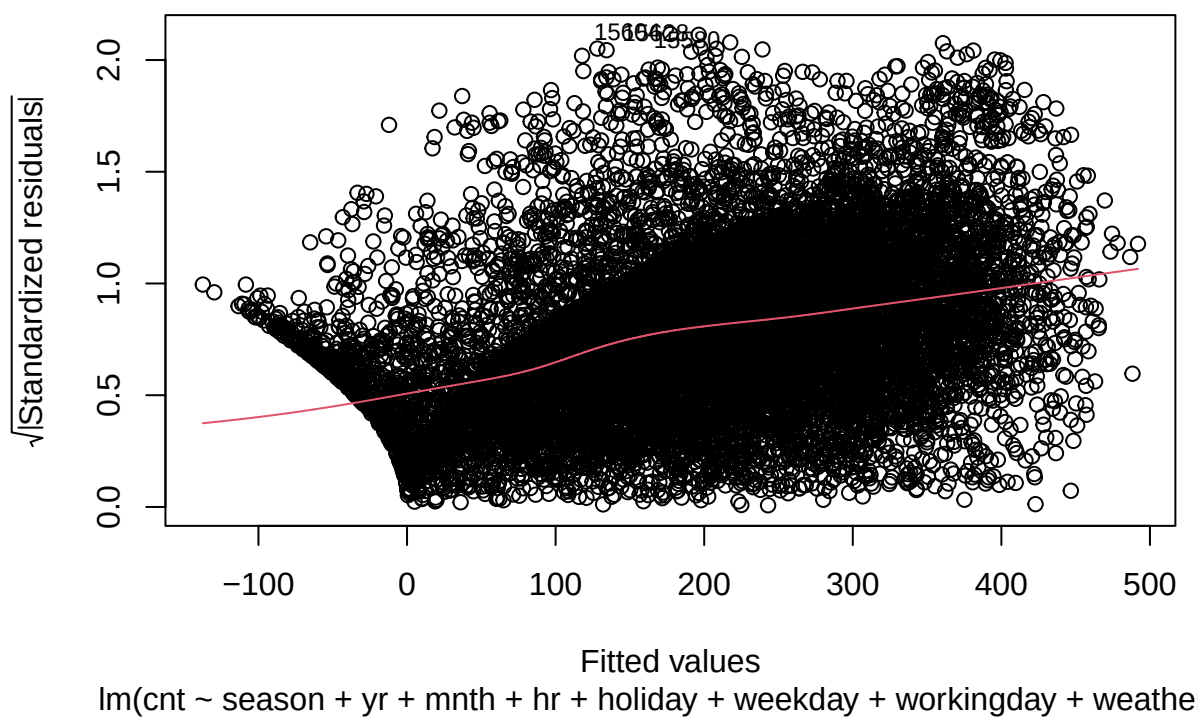
```
##
## Call:
## lm(formula = cnt ~ season + yr + mnth + hr + holiday + weekday +
##     workingday + weathersit + temp + atemp + hum + windspeed,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -388.32  -94.57  -28.22   62.11  639.74
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -23.3452     7.9469  -2.938 0.003312 **
## season        19.7149     2.0474   9.629  < 2e-16 ***
## yr            82.7079     2.4354  33.960  < 2e-16 ***
## mnth           0.1842     0.6393   0.288 0.773258
## hr             7.6861     0.1861  41.312  < 2e-16 ***
## holiday      -20.4087     7.5383  -2.707 0.006791 **
## weekday        1.6390     0.6084   2.694 0.007067 **
## workingday     5.2080     2.6968   1.931 0.053476 .
## weathersit    -3.9226     2.1585  -1.817 0.069200 .
## temp          58.1022    41.1164   1.413 0.157645
## atemp        250.4947    46.1990   5.422 5.99e-08 ***
```

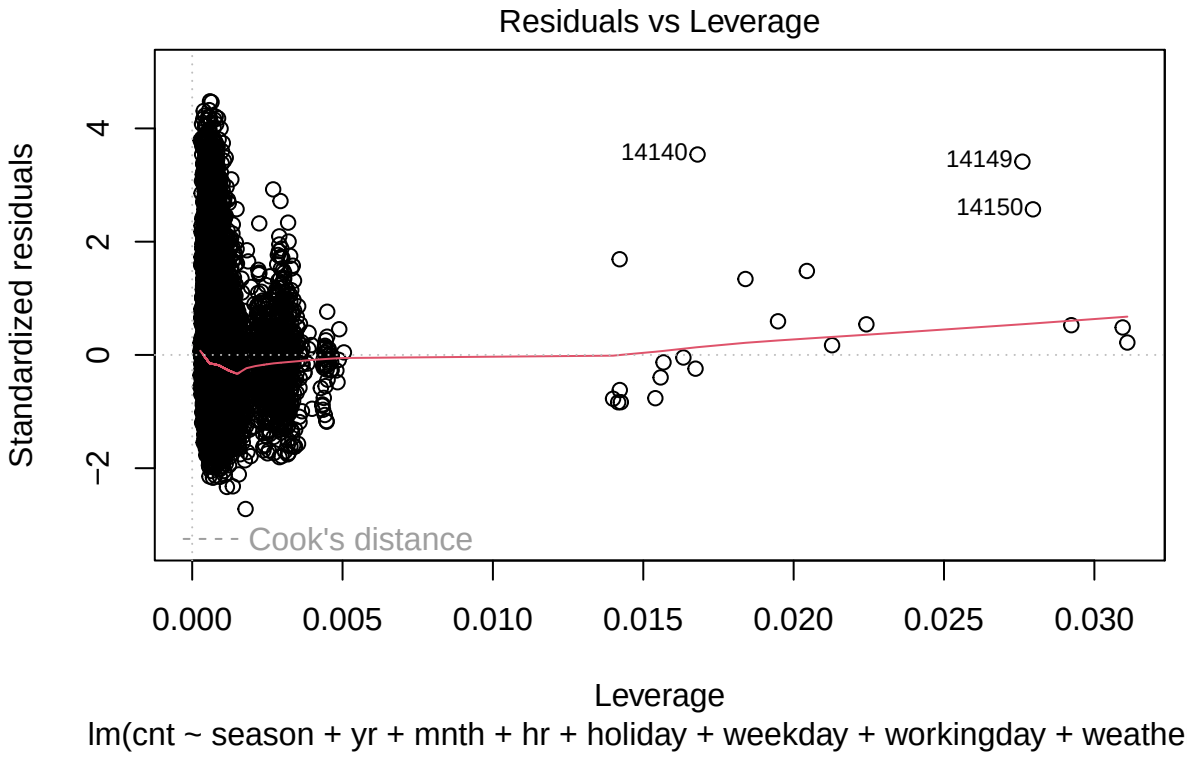
```
## hum         -200.3781     7.7889 -25.726  < 2e-16 ***
## windspeed     41.9978    10.8877   3.857 0.000115 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.9 on 13890 degrees of freedom
## Multiple R-squared:  0.3849, Adjusted R-squared:  0.3843
## F-statistic: 724.2 on 12 and 13890 DF,  p-value: < 2.2e-16
```

```
plot(lm2)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(cnt ~ season + yr + mnth + hr + holiday + weekday + workingday + weather ...


Scale−Location
√|Standardized residuals|
Fitted values
lm(cnt ~ season + yr + mnth + hr + holiday + weekday + workingday + weather ...

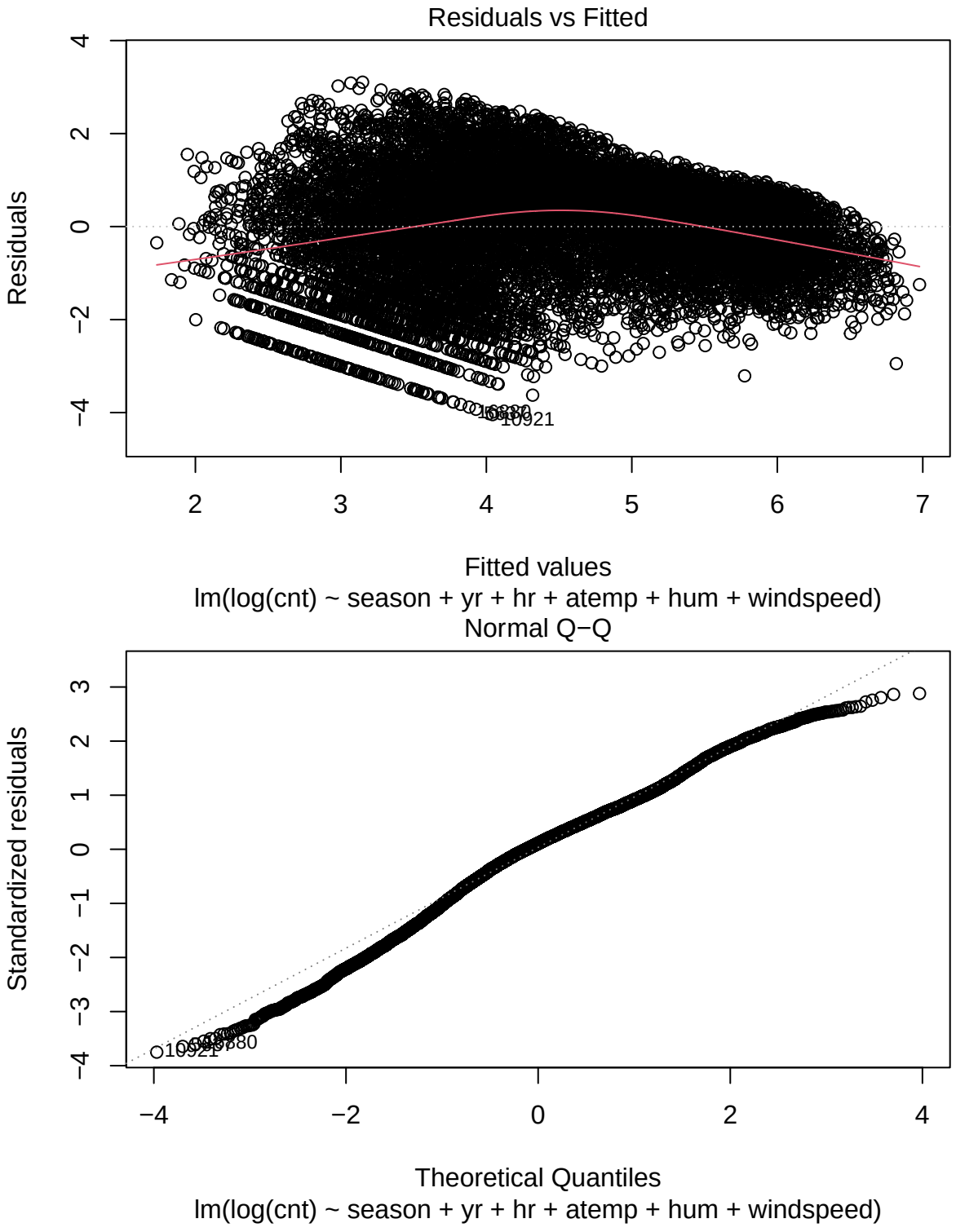


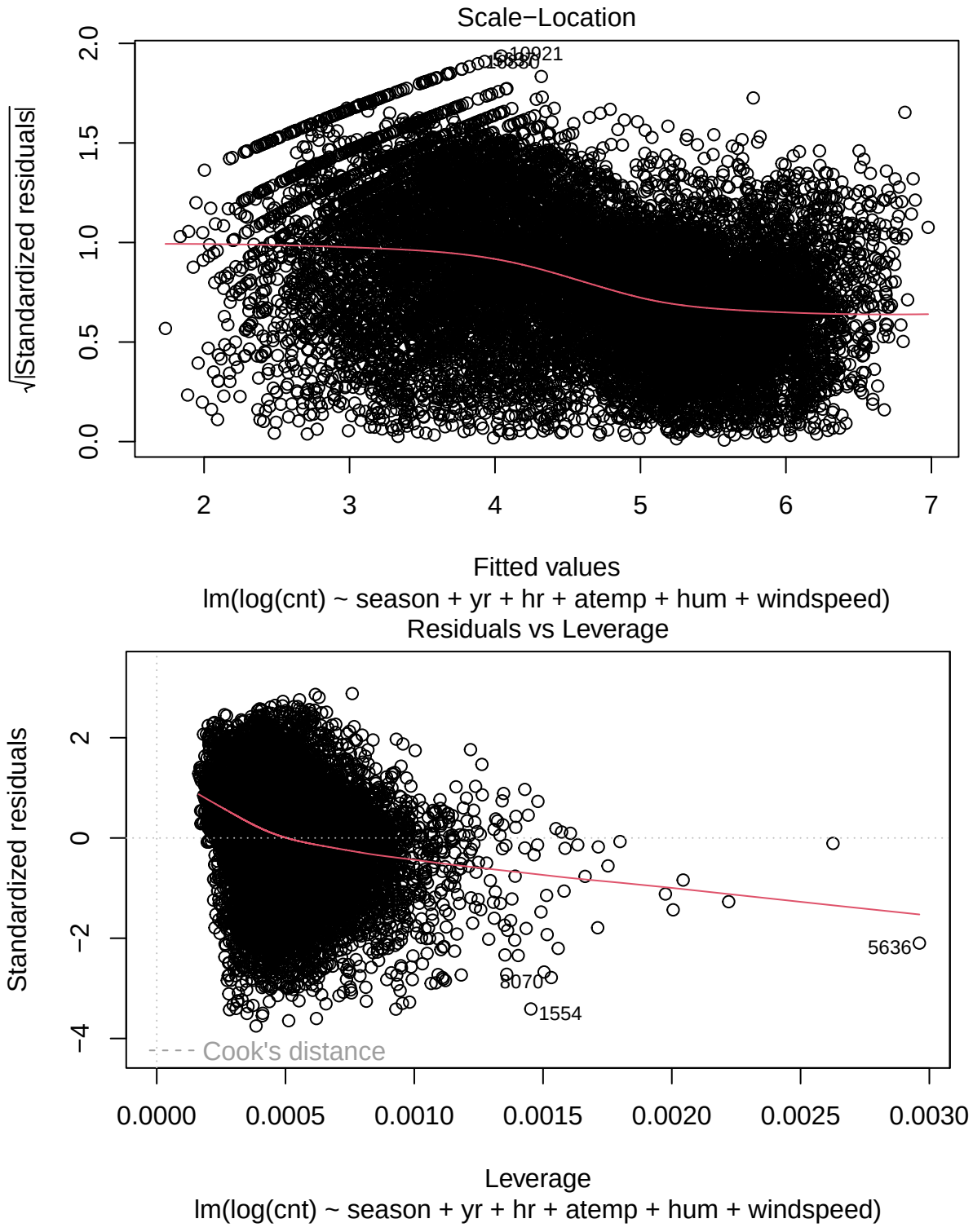
G. Build a Third Linear Regression Model using any Combination of Methods

```
lm3 <- lm(log(cnt)~season+yr+hr+atemp+hum+windspeed, data=train)
summary(lm3)
```

```
##
## Call:
## lm(formula = log(cnt) ~ season + yr + hr + atemp + hum + windspeed,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0431 -0.6396  0.1183  0.7149  3.1046
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.492829   0.055065  45.271  < 2e-16 ***
## season       0.163210   0.008884  18.371  < 2e-16 ***
## yr           0.442447   0.018371  24.085  < 2e-16 ***
## hr           0.101739   0.001393  73.054  < 2e-16 ***
## atemp        2.279388   0.057127  39.900  < 2e-16 ***
## hum         -1.453824   0.052029 -27.943  < 2e-16 ***
## windspeed    0.358776   0.079433   4.517 6.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.078 on 13896 degrees of freedom
## Multiple R-squared:  0.4761, Adjusted R-squared:  0.4758
## F-statistic:  2104 on 6 and 13896 DF,  p-value: < 2.2e-16
```

```
plot(lm3)
```


Residuals vs Fitted
Residuals
Fitted values
lm(log(cnt) ~ season + yr + hr + atemp + hum + windspeed)
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(cnt) ~ season + yr + hr + atemp + hum + windspeed)



H. Compare Results of Models

After building the first linear model, I realized that temperature alone was not going to be the best predictor for the number of bikes rented simply because once temperatures reach a normal range then there are a lot of other predictors that have a stronger correlation and therefore predictive value with bikes rented. In my second linear model, I did a multiple linear regression model using all of the other meaningful predictors from

the dataset. This alone drastically improved the effectiveness of the model, allowing the r-squared value to reach a value of 0.385. Once I created this model I could see which of the predictors actually had the biggest impact on the coefficient of the line of best fit. This is how I created my third linear model which is both a multiple regression model as well as a logarithmic regression model since it's trying to predict the log of the number of bikes registers. This gave us the highest r-squared value yet of 0.476. The third model is clearly the best one since it uses only the predictors which are of most consequence and also a logarithmic value of the target variable, however it's important to note that the third model wouldn't do great at predicting the values of rented bikes since it's using the squared values of rented bikes instead.

I. Predict and Evaluate on Test Data

```
pred1 <- predict(lm1, newdata=test)
pred2 <- predict(lm2, newdata=test)
pred3 <- predict(lm3, newdata=test)
cor1 <- cor(pred1, test$cnt)
cor2 <- cor(pred2, test$cnt)
cor3 <- cor(pred3, test$cnt)
mse1 <- mean((pred1-test$cnt)^2)
mse2 <- mean((pred2-test$cnt)^2)
mse3 <- mean((pred3-test$cnt)^2)
rmse1 <- sqrt(mse1)
rmse2 <- sqrt(mse2)
rmse3 <- sqrt(mse3)
print(paste('correlation of 1st model:' , cor1))
```

```
## [1] "correlation of 1st model: 0.435648148621893"
```

```
print(paste('correlation of 2nd model:' , cor2))
```

```
## [1] "correlation of 2nd model: 0.63622062658053"
```

```
print(paste('correlation of 3rd model:' , cor3))
```

```
## [1] "correlation of 3rd model: 0.612653531839747"
```

```
print(paste('mse of 1st model: ' ,mse1))
```

```
## [1] "mse of 1st model:  25846.3516689335"
```

```
print(paste('mse of 2nd model: ' ,mse2))
```

```
## [1] "mse of 2nd model:  18960.0206982109"
```

```
print(paste('mse of 3rd model: ' ,mse3))
```

```
## [1] "mse of 3rd model:  63916.5595696046"
```

```
print(paste('rmse of 1st model: ' ,rmse1))
```

```
## [1] "rmse of 1st model:  160.768005737875"
```

```
print(paste('rmse of 2nd model: ' ,rmse2))
```

```
## [1] "rmse of 2nd model:  137.695390983907"
```

```
print(paste('rmse of 3rd model: ' ,rmse3))
```

```
## [1] "rmse of 3rd model:  252.817245395967"
```

Looking at the correlation, mean squared error, and r-mean squared error, it's clear that it is the second model which is the best predictor of the number of rented bikes. It makes sense that the 2nd model and 3rd model are better than the 1st in terms of correlation simply because we used many more predicors to create those models, however, it's important to note that while the 3rd model has a high correlation it also the largest mse and rmse. This discrepancy is likely due to the fact that we used the relationship between the predictors and the log of the target variable as oppossed dot the actual values of the target. In conclusion, the multiple regression model with all of the predictors included was the overall best model of rented bikes, making it clear that when so many significant predictors are available to use then it's better to use them as opposed to sticking to just one predictor like we did for the first linear model.