Naeem - Nowrouzi - 707 - Final Project - Spring 2018

The dataset for this brief analysis is a two-year hourly log for years 2011 and 2012 from the Capital bike sharing system in Washington D.C.. The aim is to use (count) regressions to predict hourly bike rental count based on environmental and seasonal factors. The dates re disregarded, however the weekday, hours, months, and seasons are included as factors. We will begin with a brief data exploration, and then start from a simple linear regressions which we will attemp to improve, followed by some count models and other regressions. The dataset is available at UCI Machine Learning repository. The variables except "casual"(rental counts for non-subscribers) and "registered"(counts for subsribers) counts are normalized in the original data.

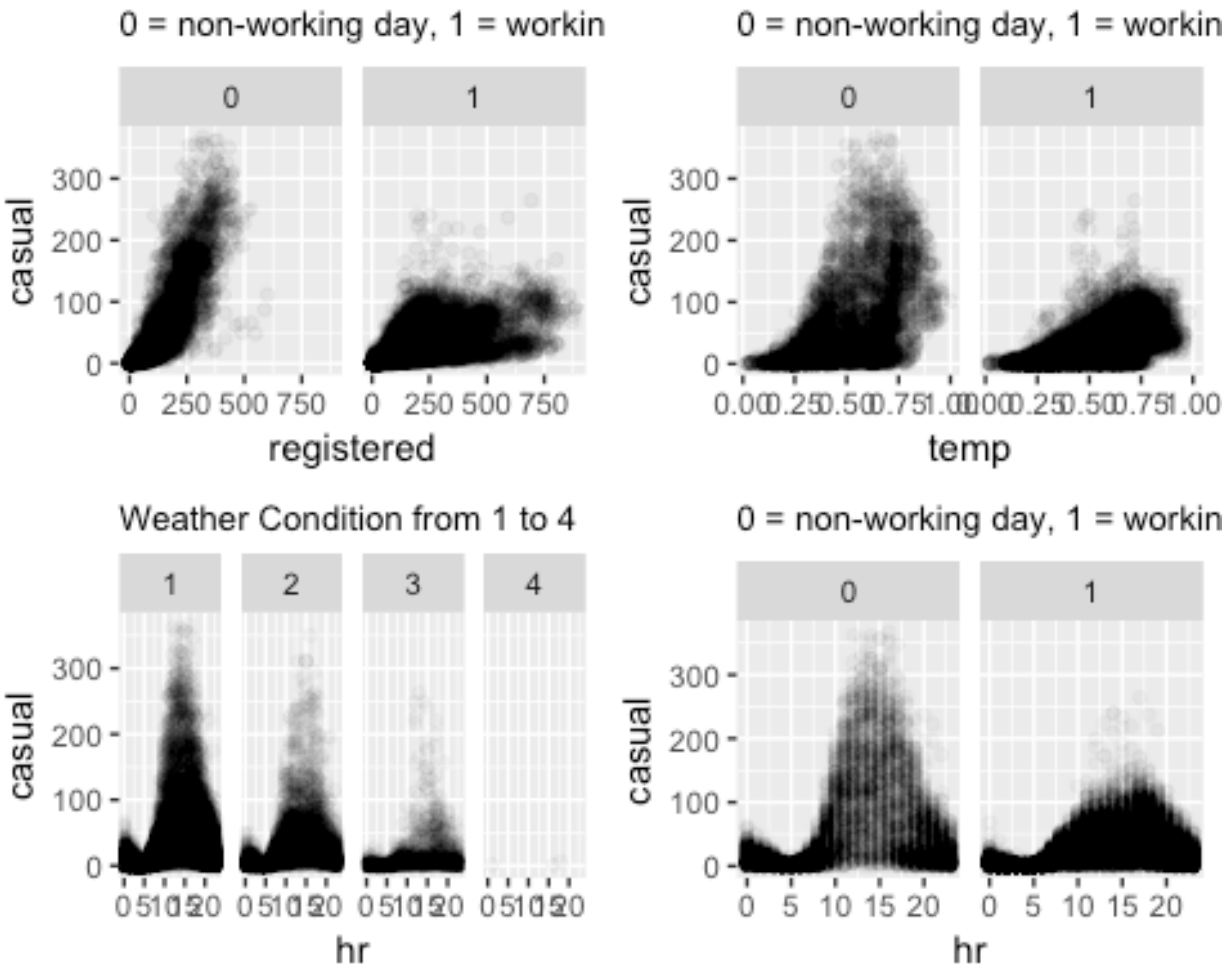
```
setwd("/Users/naeemnowrouzi/Desktop")
bike.data0 <- read.csv("~/Desktop/NAEEM/hour2.csv")
dim(bike.data0)

## [1] 17379    15

bike.data <- bike.data0[,-c(2,3,15)] # remove the total count, date, and id v
ariables.
dim(bike.data) # Check data dimension.

## [1] 17379    12

# sum(is.na(bike.data))
#str(bike.data)
#summary(bike.data)
#pairs(casual~.,bike.data)
# Check the correlation matrix
#cor(bike.data)
```

We checked the correlation matrix (not included here) and there are some mild and a few strong correlations between the variables. The cor() function uses the Pearson correlation coefficient which is a measure of linear dependence. For instance, in the case of our response variable "casual", the first three largest correlation coefficients correspond to the variables "registered", and the two measurements of temperature(atemp for the "feels like" temperature). These coefficients are,

```
col.names <- names(bike.data)[order(cor(bike.data)["casual",], decreasing = T
)][2:4]
cor(bike.data)["casual", c(col.names)]

## registered       temp      atemp
##  0.5066177  0.4596156  0.4540801
```

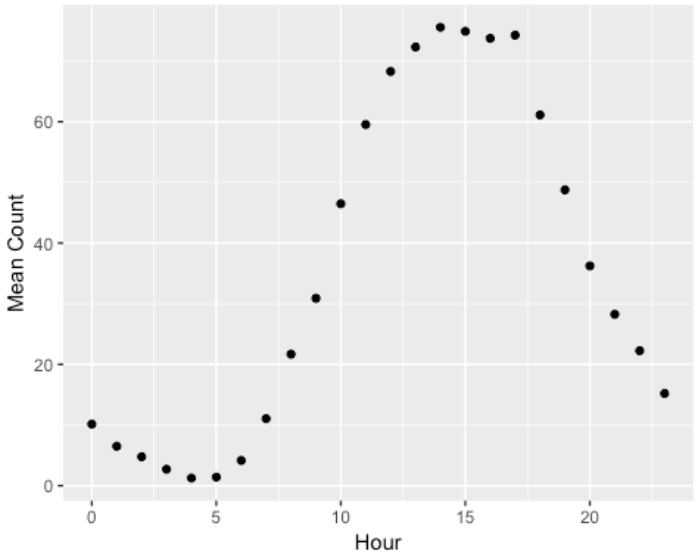
All three coefficients are positive. Intuitively this makes sense. As the number of bikes being used by registered customers increases, we expect the number of non-registered customers to increase as well. Though this may not necessarily be the case. Same for temperature, as the weather gets warmer, we expect to see more bikes being used. An example of negative correlation is with humidity in which case the number of bikes used

decreases when humidity increases. It is also natural to believe that the response should be strongly correlated to the day and hour, but this does not seem to be properly reflected in the correlation matrix perhaps due to presence of a non-linear relationship. We check some of the more revealing plots of our response against some of the seemingly important variables



We particularly observe the importance of hour, weather condition, and whether or not it's a working day on the rental counts. From the top left plot it seems that on a non-working day there is a more pronounced linear relationship between rental counts for subscribers and for non-subscribers. On a working day, no clear-cut relationsihp is visible.

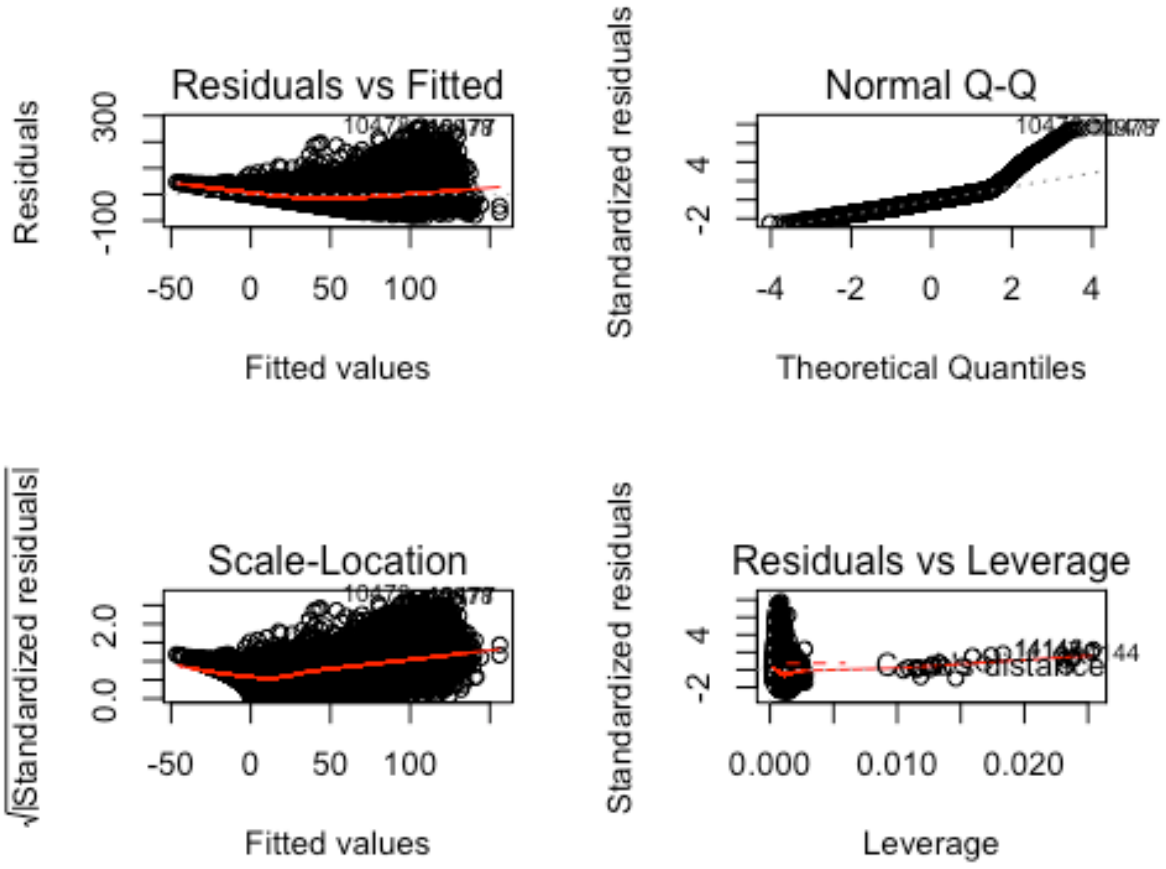
It seems in general the rental count for non-subscribers is higher on non-working days, and for subscribers it is higher on working days, which makes sense. The top right plot displays the relationship between non-subscriber rental count and temperature on working and non-working days seperately. On a non-working day we see about two thirds of a bell shape, while on a working day we see only about half of a bell shape that's also shorter, implying that the combination of working day and unfavarobale temprature specially reduces the number of bikes rented by non-subscribers. These two variables, i.e., hour and temperature, exhibit cyclical patterns. We will shortly try to derive a few new features by transforming these predictors using sine and cosine functions. Lastly, since the size of observations is large, to confirm this cyclic pattern we look at the mean rental count against hour for instance.



We begin by performing an ordinary least squares regression that includes all of the variables.

```
lm.fit.00 <- lm(casual ~ ., data = bike.data)
summary(lm.fit.00)

##
## Call:
## lm(formula = casual ~ ., data = bike.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.295 -19.782  -3.215  13.242 260.349
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.141749   1.613842  14.959  < 2e-16 ***
## season       -0.447104   0.429975  -1.040    0.298
## mnth         -0.069542   0.133610  -0.520    0.603
## hr            0.428950   0.040804  10.512  < 2e-16 ***
## workingday  -39.029109   0.552915 -70.588  < 2e-16 ***
## weekday       0.785064   0.126886   6.187 6.26e-10 ***
## weathersit    2.703895   0.448982   6.022 1.75e-09 ***
## temp         48.274362   8.711275   5.542 3.04e-08 ***
## atemp        44.461182   9.791217   4.541 5.64e-06 ***
## hum         -55.091514   1.644001 -33.511  < 2e-16 ***
## windspeed    -1.784034   2.269603  -0.786    0.432
## registered    0.119240   0.001978  60.298  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.45 on 17367 degrees of freedom
## Multiple R-squared:   0.54,  Adjusted R-squared:  0.5397
## F-statistic:  1854 on 11 and 17367 DF,  p-value: < 2.2e-16
```

There are four significantly large coefficients, two positive and two negative, these are workingday, temp, atemp, and humidity. atemp has a particularly large SE of about 9.8. The coefficient for hour is about 0.43. temp, atemp, and humidity are normalized so their value is small. The RSE is at 33.45, and the Adjusted-r^2 is about 0.54. Three variables, season, month, and windspeed are not statistically significant. We check the diagnostic plots as well.



The residuals normality assumption is clearly violated as seen in the summary results and the diagnostic plots. There seems to be right-skewness. The funnel shape in the residuals vs fitted values plots indicate heteroscedasticity, suggesting the use of a quasi model. Further, there are a few points that have large cook's distances. They all have values for residual that is not concerning, but they do have large leveregase. We identify these points.

```
#Get a list of the high leveraged points
#filter(bike.data0, hatvalues(lm.fit.00)>0.008)
```

There are exactly 24 of them corresponding to 24 hours of a single day, August 17th, 2012. The total rental count for subscribers and non-subscribes on this day is actually lower than the previous day and apparently than the average of the month of August. The temperature is also similar to the previous day. However, the atemp variable, which is the "feels like" temperature has suddenly dropped to about a third of the previous day and is constant for all hours. We remove this single day from our data, instead of computing new values for them and check what takes place.

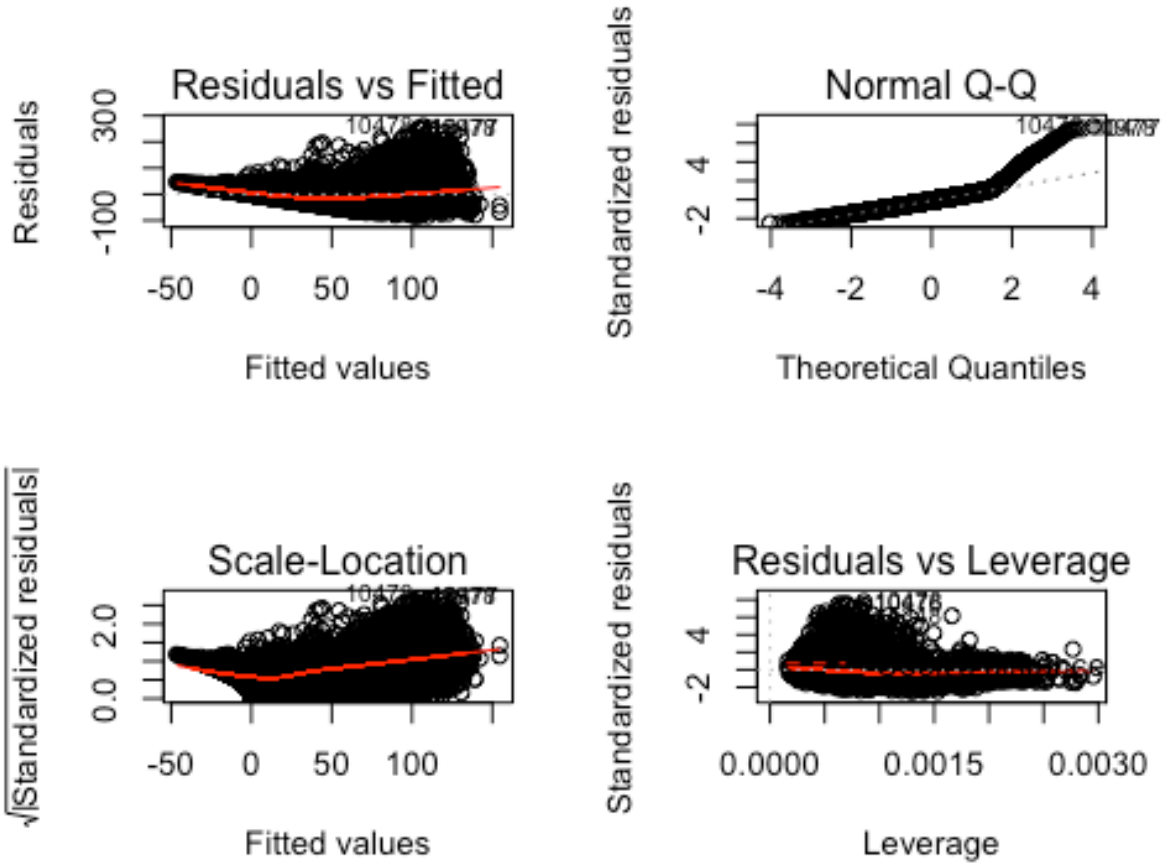
```r
library("dplyr")
#filter(bike.data0, hatvalues(lm.fit.00)>0.008) %>% select(instant)
new.data <- bike.data0[-c(14132:14155), -c(2,3,15)] # Remove the points.
dim(new.data) # Check
```

```
## [1] 17355    12
```

```r
#new.data[14132:14155,] # Check

final.data00 <- read.csv("~/Desktop/bike.data02.csv")
final.data <- final.data00[,-c(2,3,15)]


lm.fit.01 <- lm(casual ~ ., data = final.data)
summary(lm.fit.01)
```

```
##
## Call:
## lm(formula = casual ~ ., data = final.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.706 -19.861  -3.177  13.278 259.961
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.588017   1.638315  13.787  < 2e-16 ***
## season       -0.486816   0.429723  -1.133    0.257
## mnth         -0.067940   0.133508  -0.509    0.611
## hr            0.426607   0.040775  10.463  < 2e-16 ***
## workingday  -39.078960   0.552536 -70.727  < 2e-16 ***
## weekday       0.783122   0.126727   6.180 6.57e-10 ***
## weathersit    2.773158   0.448836   6.179 6.61e-10 ***
## temp         12.670649  11.112417   1.140    0.254
## atemp        84.708328  12.508553   6.772 1.31e-11 ***
## hum         -55.411572   1.643755 -33.710  < 2e-16 ***
## windspeed     0.088892   2.295205   0.039    0.969
## registered    0.118929   0.001977  60.155  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.43 on 17367 degrees of freedom
## Multiple R-squared:  0.5407, Adjusted R-squared:  0.5404
## F-statistic:  1859 on 11 and 17367 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(lm.fit.01)
```



The only important change is in the significane of variables since now the “temp” variable is statistically insignificant as it has a large p-value. “atemp” now has a much larger coefficient(84.7), but also a large standard error of about 12.5, but a very small p-value. Thus it plays an important role but the estimation for its coefficient is relatively instable. I was curious to see if this was the effect of removing the high leverage points, and thus undid that and placed new values for these 24 points based on the values of “temp” and the observed trend of “atemp” being generally less than “temp”. The result is almost identical to when we remove these points. We proceed with using this final version of the data throught the rest of the paper. Lastly, plot of the residuals vs leverages was checked and it was no longer problematic.

```
lm.01.preds <- predict(lm.fit.01, newdata = final.data, type = "response")
lm.01.train.err <- mean((final.data$casual - lm.01.preds)^2)
lm.01.train.err

## [1] 1116.511
```

The training error on the final.data is 1116.5.

We now apply the sine and cosine transformations to the hour and temp variables since they exhibited cyclical patterns in the previous plots. Since the working day appeared to affect the amplitude of the shape by a factor, we tried to incorporate that into the model and it was unsuccesful as it did not make any tangible improvements.

We further added a few interaction terms that appeared to be statistically significant and slightly improve the adj-r^2 to 0.63. All of the derived features and interaction terms appear to be statistically significant with low SEs and very small p-values.

Lastly we transformed the response variable by take its square root. This appears to significantly improve the model. The RSE has dropped from around 30 to about 1.7, the adjusted-r^2 has raised to 0.78, and the residuals are much closer to being normally distributed now. Coefficients and their SEs also seem much more normalized and less wild. The training error has also dropped from 1116.5 to 897.

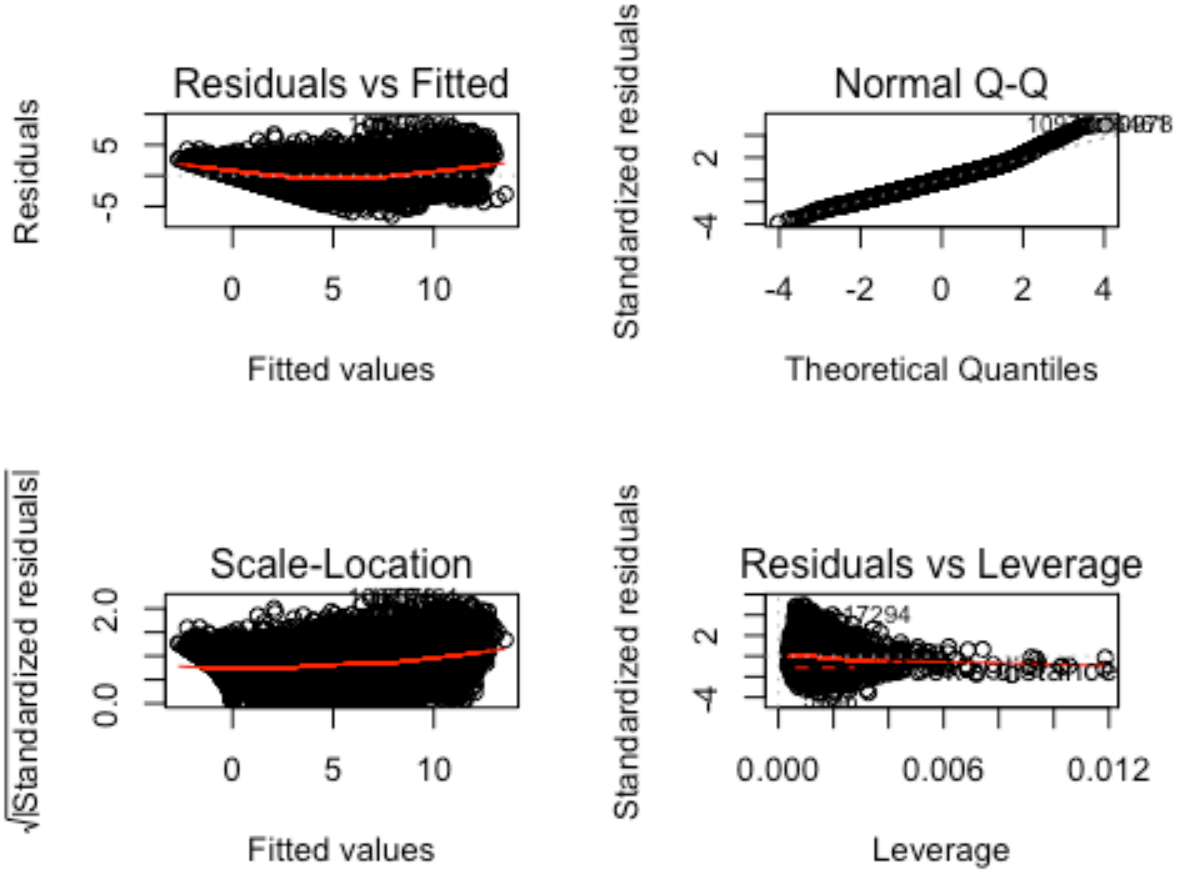
```
lm.fit.11 <- lm(sqrt(casual) ~ . + hr:workingday:atemp + weathersit:registere
d +
                  temp:weekday + sin(2*pi*hr/24) + cos(2*pi*hr/24) +
                  sin(2*pi*atemp/max(atemp)) + cos(2*pi*atemp/max(atemp)),
                  data = final.data)


summary(lm.fit.11)

##
## Call:
## lm(formula = sqrt(casual) ~ . + hr:workingday:atemp + weathersit:registere
d +
##     temp:weekday + sin(2 * pi * hr/24) + cos(2 * pi * hr/24) +
##     sin(2 * pi * atemp/max(atemp)) + cos(2 * pi * atemp/max(atemp)),
##     data = final.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5010 -1.1246 -0.0434  1.0181  8.2374
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.534e+00  1.610e-01  28.160  < 2e-16 ***
## season                          8.848e-02  2.163e-02   4.090 4.33e-05 ***
## mnth                           -1.636e-02  6.736e-03  -2.428  0.01518 *
## hr                              3.477e-02  3.520e-03   9.879  < 2e-16 ***
## workingday                     -1.729e+00  4.372e-02 -39.546  < 2e-16 ***
## weekday                         2.316e-02  1.716e-02   1.350  0.17714
## weathersit                     -2.899e-01  2.952e-02  -9.821  < 2e-16 ***
## temp                           -1.946e+00  6.035e-01  -3.224  0.00127 **
## atemp                           5.592e+00  6.513e-01   8.587  < 2e-16 ***
## hum                            -1.924e+00  8.977e-02 -21.437  < 2e-16 ***
## windspeed                      -7.586e-01  1.158e-01  -6.553 5.81e-11 ***
## registered                      6.749e-03  2.256e-04  29.912  < 2e-16 ***
## sin(2 * pi * hr/24)            -1.451e+00  3.114e-02 -46.604  < 2e-16 ***
## cos(2 * pi * hr/24)            -1.624e+00  2.122e-02 -76.500  < 2e-16 ***
## sin(2 * pi * atemp/max(atemp)) -1.120e+00  6.460e-02 -17.331  < 2e-16 ***
## cos(2 * pi * atemp/max(atemp)) -5.272e-01  3.263e-02 -16.156  < 2e-16 ***
## weathersit:registered           7.334e-06  1.482e-04   0.049  0.96054
## weekday:temp                    6.957e-02  3.266e-02   2.130  0.03317 *
## hr:workingday:atemp            -1.200e-01  6.149e-03 -19.516  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.675 on 17360 degrees of freedom
## Multiple R-squared:  0.7845, Adjusted R-squared:  0.7842
## F-statistic:  3510 on 18 and 17360 DF,  p-value: < 2.2e-16
```

```
lm.11.train.preds <- predict(lm.fit.11, newdata = final.data, type = "respons
e")
lm.11.train.err <- mean((final.data$casual - (lm.11.train.preds)^2)^2)
lm.11.train.err
```

```
## [1] 681.5309
```

```
par(mfrow=c(2,2))
plot(lm.fit.11)
```




```
#drop1(lm.fit.11, test = "F")
```

The MSE on the original data is is 681.5, which is improved compared to 1116.5 for the simple model, but still quite large! We display the diagnostic plots and it seems they are improved as well. The peculiar lines on the residuals plot are likely to be due to high frequency counts.

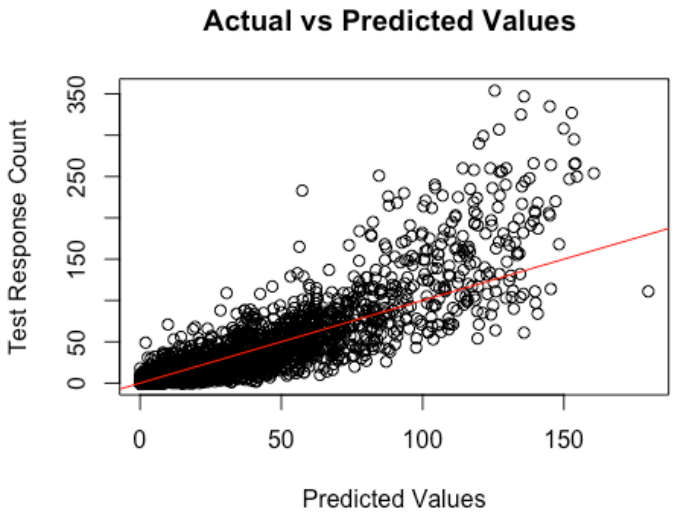
We also performed a variable selection by the drop1() function and noted that it suggests a 12-variable model (including the intercept) compared the the original 19-variable model. However, we tested the model and it was inferioir to the original model. So we keep all of the variables except for one of the interaction terms that was not significant, and proceed to fit this model to a training set and testing it on a validation set.

```
# Create the test and training sets using 80% of the data for train and 20% f
or test.
set.seed(1)
train <- sample(1:nrow(final.data), round(0.85*nrow(final.data),0))
train.set <- final.data[train,]
dim(train.set)

## [1] 14772    12

test.set <- final.data[-train,]
dim(test.set)

## [1] 2607   12

lm.fit.12 <- lm(sqrt(casual) ~ . + hr:workingday:atemp + weathersit:registere
d +
                  temp:weekday + sin(2*pi*hr/24) + cos(2*pi*hr/24) +
                  sin(2*pi*atemp/max(atemp)) + cos(2*pi*atemp/max(atemp)),
                  data = train.set)

# Compute the training MSE.
lm.12.train.preds <- predict(lm.fit.12, type = "response")
lm.12.train.err <- mean((train.set$casual - (lm.12.train.preds)^2)^2)
lm.12.train.err

## [1] 674.6287

# Compute the test MSE
lm.12.preds <- predict(lm.fit.12, newdata = test.set, type = "response")
lm.12.test.err <- mean((test.set$casual - (lm.12.preds)^2)^2)
lm.12.test.err

## [1] 720.7059
```

The train MSE is 678.5 and the test MSE is 718.75, not a large departure from the training error.

We look at the actual vs fitted values plot. Although the model seems to fit the data, it appears that it is not really predicting any large counts, which are available in the data.

```
par(mfrow=c(1,1))
lp <- (lm.12.preds)^2
plot(test.set$casual~lp, ylab = "Test Response Count", xlab = "Predicted Valu
es",
     main = "Actual vs Predicted Values ")
abline(0,1,col="red")
```

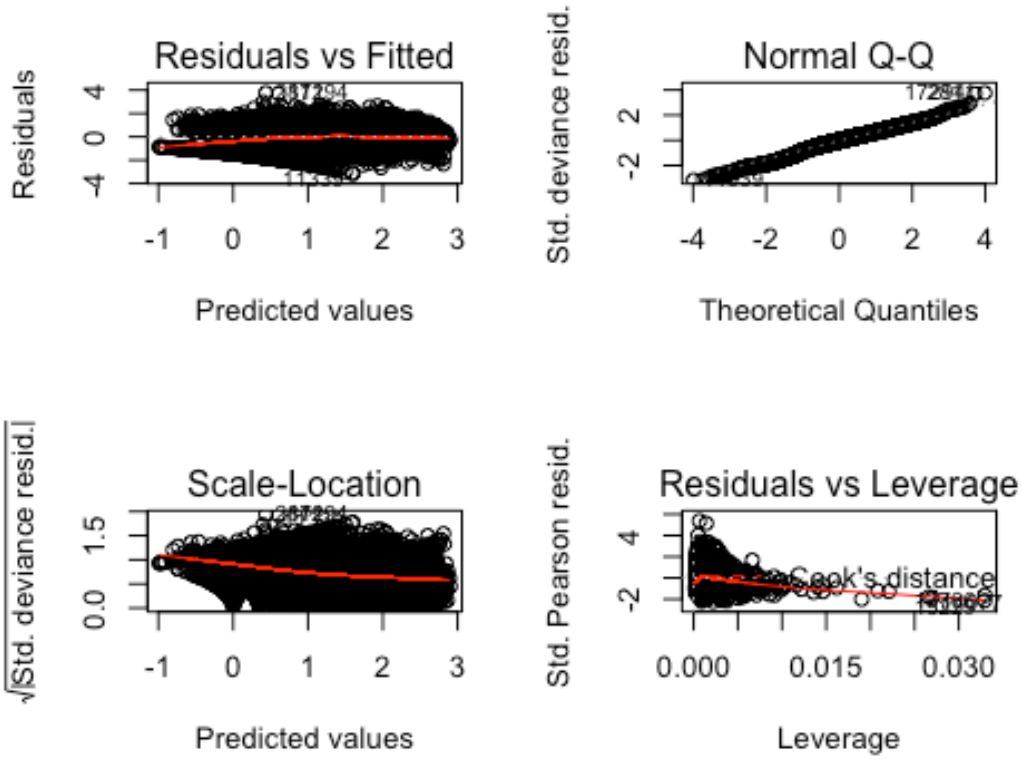


We now proceed to some count regression models. We begin by fitting a poisson regression model on the transfomed variable along with the interaction terms and derived feature, as this model was superior to the the simple poisson on the original variables (summary tables excluded).

```
library("dplyr")
library("faraway")

poisson.fit <- glm(sqrt(casual) ~ . + hr:workingday:atemp + weathersit:regist
ered +
                   temp:weekday + sin(2*pi*hr/24) + cos(2*pi*hr/24) +
                   sin(2*pi*atemp/max(atemp)) + cos(2*pi*atemp/max(atemp)),
                   family = poisson, train.set) #response could not be transfo
rmed
summary(poisson.fit)

##
## Call:
## glm(formula = sqrt(casual) ~ . + hr:workingday:atemp + weathersit:register
ed +
##     temp:weekday + sin(2 * pi * hr/24) + cos(2 * pi * hr/24) +
##     sin(2 * pi * atemp/max(atemp)) + cos(2 * pi * atemp/max(atemp)),
##     family = poisson, data = train.set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1528  -0.5378  -0.0232   0.4457   3.7642
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    7.931e-01  5.714e-02  13.880  < 2e-16 ***
## season                         2.712e-02  7.683e-03   3.529 0.000417 ***
## mnth                          -8.232e-03  2.514e-03  -3.275 0.001057 **
## hr                             2.274e-02  1.307e-03  17.399  < 2e-16 ***
## workingday                    -5.414e-01  1.678e-02 -32.273  < 2e-16 ***
```

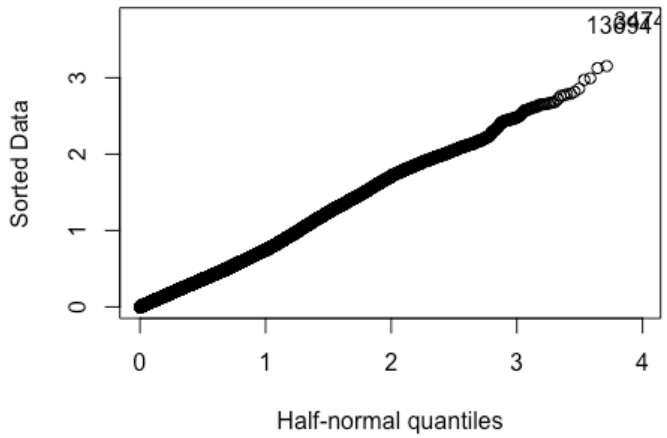
```
## weekday                             1.709e-02  5.860e-03   2.915 0.003551 **
## weathersit                         -1.722e-01  1.083e-02 -15.907  < 2e-16 ***
## temp                               -6.026e-02  1.685e-01  -0.358 0.720568
## atemp                               1.422e+00  1.836e-01   7.745 9.57e-15 ***
## hum                                -2.839e-01  2.724e-02 -10.425  < 2e-16 ***
## windspeed                          -1.397e-01  3.262e-02  -4.284 1.84e-05 ***
## registered                          4.086e-04  5.923e-05   6.898 5.26e-12 ***
## sin(2 * pi * hr/24)                -1.948e-01  1.040e-02 -18.731  < 2e-16 ***
## cos(2 * pi * hr/24)                -4.463e-01  6.705e-03 -66.563  < 2e-16 ***
## sin(2 * pi * atemp/max(atemp)) -7.732e-02  2.098e-02  -3.685 0.000228 ***
## cos(2 * pi * atemp/max(atemp)) -3.246e-01  1.011e-02 -32.101  < 2e-16 ***
## weathersit:registered               4.606e-04  3.962e-05  11.624  < 2e-16 ***
## weekday:temp                       -9.269e-03  9.817e-03  -0.944 0.345096
## hr:workingday:atemp                 9.337e-03  1.975e-03   4.726 2.29e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 42334.8  on 14771  degrees of freedom
## Residual deviance:  9523.5  on 14753  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 5
```

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



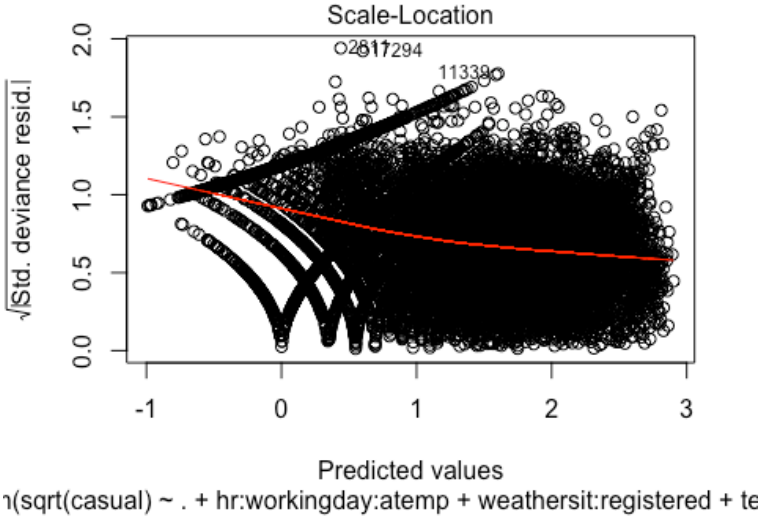
The residual deviance is significantly smaller than the null deviance, and smaller than the number of degrees of freedom. The coefficients are mild with small standard errors. The largest coefficient with a value of around 5 belongs to the “feels like” temperature. All of the variables except month are significant. We now look for outliers and then shortly move to use the quasi-Poisson model.

```
halfnorm(abs(residuals(poisson.fit)))
```



There are two points on the top right corner that are furthur away from the rest, not not in an extreme manner, so we disregard them. We observe in the following plot that that general that the variance has a mild decreasing trend indicative of small amoun of underdispersion. This is confirmed by the estimation of dispersion parameter which is slightly less than 1, at 0.56.

```
par(mfrow=c(1,1))
plot(poisson.fit, which=3)
```



η(sqrt(casual) ~ . + hr:workingday:atemp + weathersit:registered + te

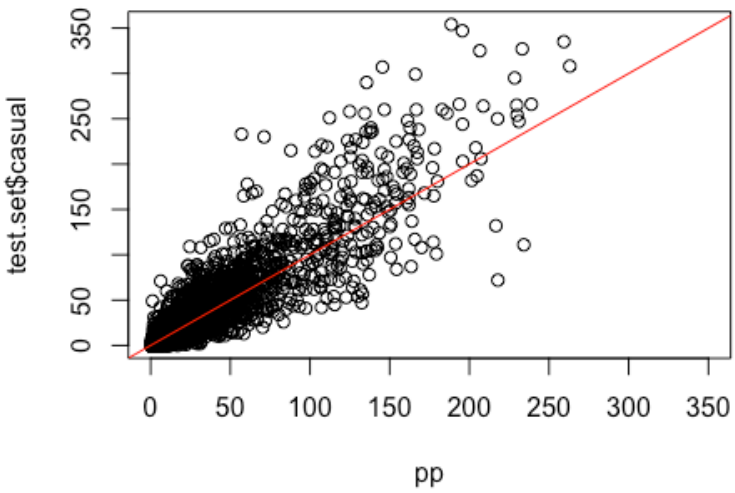
```
# Estimate the dispersion parameter
(dp <- sum(residuals(poisson.fit,type="pearson")^2)/poisson.fit$df.res)

## [1] 0.5541936
```

We now compute the test MSE.

```
#Get the predicted values and transform back by squaring when getting the tes
t MSE.
poisson.preds <- predict(poisson.fit, newdata = test.set, type = "response")
poiss.test.err <- mean((test.set$casual-(poisson.preds)^2)^2)
poiss.test.err

## [1] 551.4294
```

The test error of 450.9 is much smaller than the error from the linear regression model. We also look at the actual vs predicted values plot for the test set. It appears that the model does fit the data well.

```
# Plot the actual test values vs the predicted values on the test set.
pp<-(poisson.preds)^2
plot(test.set$casual~pp, xlim=c(0,350))
abline(0,1, col = "red")
```



In order to confirm our conclusion regarding the dispersion parameter, we fit a quasi-poisson model and check the results.

```
poiss.quasi.fit <- glm(sqrt(casual) ~ . + hr:workingday:atemp + weathersit:re
gistered +
                  temp:weekday + sin(2*pi*hr/24) + cos(2*pi*hr/24) +
                  sin(2*pi*atemp/max(atemp)) + cos(2*pi*atemp/max(atemp)),
                  family = "quasipoisson", train.set)
#summary(poiss.quasi.fit)
quasi.poiss.preds <- predict(poiss.quasi.fit, newdata = test.set, type = "res
ponse")
```

```
quasi.test.err <- mean((test.set$casual-(quasi.poiss.preds)^2)^2)
quasi.test.err

## [1] 551.4294
```

The results are identical so we disregard this model and proceed to fitting a negative binomial model.

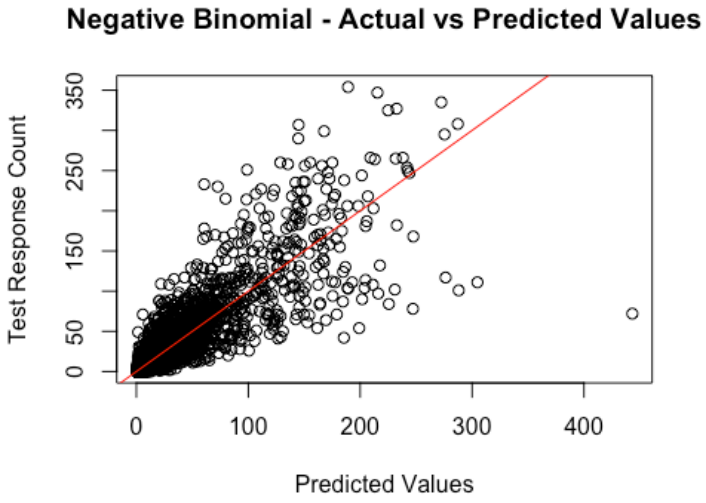
```
library("MASS")
neg.binom.fit <- glm(sqrt(casual) ~ . + hr:workingday:atemp + weathersit:regi
stered +
                   temp:weekday + sin(2*pi*hr/24) + cos(2*pi*hr/24) +
                   sin(2*pi*atemp/max(atemp)) + cos(2*pi*atemp/max(atemp)),
                 negative.binomial(1),train.set) # Theta = 1 for geometric di
stribution.
summary(neg.binom.fit)

##
## Call:
## glm(formula = sqrt(casual) ~ . + hr:workingday:atemp + weathersit:register
ed +
##     temp:weekday + sin(2 * pi * hr/24) + cos(2 * pi * hr/24) +
##     sin(2 * pi * atemp/max(atemp)) + cos(2 * pi * atemp/max(atemp)),
##     family = negative.binomial(1), data = train.set)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.87696  -0.24710  -0.01226   0.20091   1.89816
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     7.411e-01  5.003e-02  14.814  < 2e-16 ***
## season                          2.904e-02  6.719e-03   4.322 1.56e-05 ***
## mnth                           -6.815e-03  2.122e-03  -3.212  0.00132 **
## hr                              1.396e-02  1.073e-03  13.003  < 2e-16 ***
## workingday                     -5.400e-01  1.363e-02 -39.612  < 2e-16 ***
## weekday                         9.806e-03  5.275e-03   1.859  0.06305 .
## weathersit                     -2.001e-01  9.290e-03 -21.543  < 2e-16 ***
## temp                            5.398e-01  1.766e-01   3.057  0.00224 **
## atemp                           1.026e+00  1.899e-01   5.403 6.67e-08 ***
## hum                            -2.635e-01  2.661e-02  -9.905  < 2e-16 ***
## windspeed                      -1.473e-01  3.358e-02  -4.388 1.15e-05 ***
## registered                      4.585e-04  6.446e-05   7.113 1.18e-12 ***
## sin(2 * pi * hr/24)            -2.951e-01  9.361e-03 -31.520  < 2e-16 ***
## cos(2 * pi * hr/24)            -4.421e-01  6.251e-03 -70.727  < 2e-16 ***
## sin(2 * pi * atemp/max(atemp))  6.984e-03  1.973e-02   0.354  0.72342
## cos(2 * pi * atemp/max(atemp)) -3.235e-01  9.536e-03 -33.928  < 2e-16 ***
## weathersit:registered           6.331e-04  4.245e-05  14.913  < 2e-16 ***
## weekday:temp                   -2.198e-03  9.714e-03  -0.226  0.82098
## hr:workingday:atemp             1.184e-02  1.821e-03   6.502 8.19e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.157899
6)
##
##     Null deviance: 9830.1  on 14771  degrees of freedom
## Residual deviance: 3474.1  on 14753  degrees of freedom
## AIC: 72252
##
## Number of Fisher Scoring iterations: 6
```

We observe that the dispersion parameter is very small, the coefficients are all small, with largest being about 1.02 corresponding to the "feels like" temperature. The residual deviande is significantly lower than the previous two models and is about 3496. We now test the model.

```
#drop1(neg.binom.fit, test = "F")
# The model suggested by drop1() does not improve the results.

neg.binom.preds <- predict(neg.binom.fit, newdata = test.set, type = "respons
e")
sp <- (neg.binom.preds)^2
neg.bin.test.err <- mean((test.set$casual-sp)^2)
neg.bin.test.err

## [1] 742.6524
```

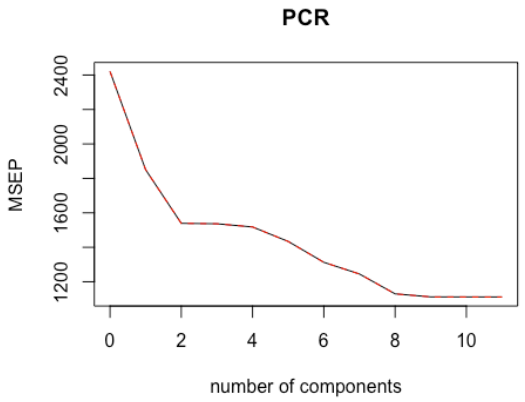
The test error is approximately 699, comparable to the linear regression model. We look at the actual vs predicted values plot and it looks quite similar to the one for the Poisson model. Again, it seems that the model does not predict any larger counts, which are available in the data.



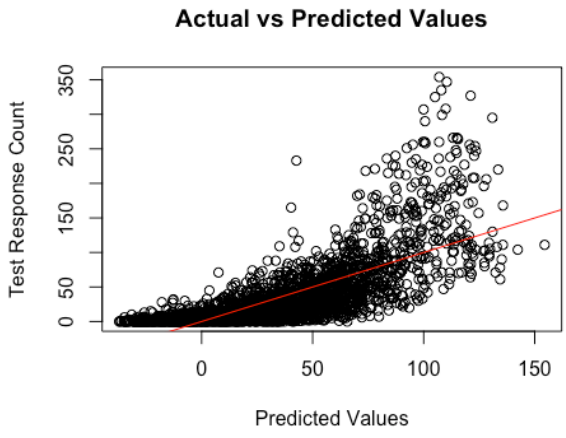
In the first line of code below we checked the number of zeroes in the dataset to determine whether or not we should fit a zero-inflated model. There are 1,581 zero counts in a total of 17,379, not very small, but not very large either, so we skip this model. We now try fitting a cross-validated Principal Component regression, followed by the two penalized regression

methods, Ridge and Lasso regression, and end the analysis with a non-parametric regression model, KNN, just to compare the test results.

```
# Check the number of zeroes in the daset
sum(final.data$casual==0)

## [1] 1581

# Fit a PCR

library("pls")
set.seed(2)
pcr.fit <- pcr(casual ~., data=train.set ,scale=TRUE, validation ="CV")
#summary(pcr.fit)
validationplot(pcr.fit, val.type="MSEP", main = "PCR")
```




```
pcr.pred <- predict(pcr.fit,newdata = test.set, ncomp=11)
mean((pcr.pred-test.set$casual)^2)

## [1] 1157.842

plot(test.set$casual ~ pcr.pred, main = "Actual vs Predicted Values", xlab =
"Predicted Values",
     ylab = "Test Response Count")
abline(0,1, lwd = 1, col="red")
```



Even with 11 components the model's performance is poor. We check the results for cross-validated Ridge and Lasso regressions with the original variables.

```
# Fit a Ridge Regression
library("glmnet")

train.x <- as.matrix(train.set[,-11])
train.y <- as.matrix(train.set[,11])
test.x <- as.matrix(test.set[,-11])
test.y <- as.matrix(test.set[,11])

ridge.mod.cv <- cv.glmnet(x=train.x, y=train.y, alpha=0) # Perform cross-vali
dation
best.lam <- ridge.mod.cv$lambda.min # Get the best lambda

grid <- 10^seq(10,-2,length=100) # Create a grid
ridge.mod <- glmnet(x=train.x, y=train.y, family = "poisson", alpha=0, lambda
= grid) # Fit
ridge.pred <- predict(ridge.mod, s = best.lam, newx = test.x, type = "respons
e") # Predicted values
ridge.test.err <- mean((test.set$casual-ridge.pred)^2) # Test error
ridge.test.err

## [1] 1053.629
```

Test error is 1053.628728.

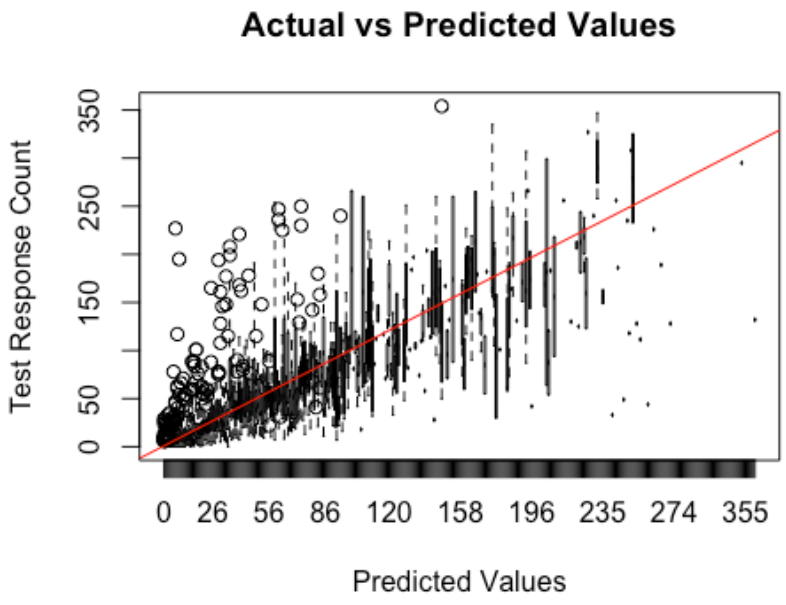
```
# Fit Lasso

lasso.mod.cv <- cv.glmnet(x=train.x, y=train.y, alpha=1)
best.lam.lass <- lasso.mod.cv$lambda.min

grid1 = 10^seq(10,-2,length=100)
lasso.mod <- glmnet(x=train.x, y=train.y, family = "poisson", alpha=1, lambda
= grid1)

lasso.pred <- predict(lasso.mod, s = best.lam.lass, newx = test.x, type = "re
sponse")
lasso.test.err <- mean((test.set$casual-lasso.pred)^2)
lasso.test.err

## [1] 1046.515
```

The test error is very similar to that of the Ridge regression. Lastly, we try the non-parametric KNN regression on the test set and plot the actual vs predicted values for this model.

```
library("class")

# Fit a KNN with k=1 as the best k.
knn.fit.preds <- knn(train.x, test.x, train.y , k=1)
knn.test.error <- mean((test.set$casual -as.numeric(knn.fit.preds))^2)
knn.test.error
```

```
## [1] 973.415
```

```
plot(test.set$casual ~ knn.fit.preds, main = "Actual vs Predicted Values", xl
ab = "Predicted Values",
     ylab = "Test Response Count")
abline(0,1, lwd = 1, col="red")
```



The test error for the KNN model is somewhere in between the previous models, closer to that of Ridge and Lasso. K=1 gave the best results for this model.

We conclude our analysis by observing that all of the models that were fitted gave large test errors. A more in-depth analysis and exploration is required to find a more appropriate model. We note that the simple least squares, Ridge and Lasso regressions gave similar errors. However, the derived features and added interactions could significantly improve the simple linear regression, bringing the test error to lower than that of the negative binomial model, and closest to the Poisson model test error, which was the lowest.