Regression

Jimmy Harvin, Yixin Sun

9-25-2022

Linear regression is a supervised machine learning algorithm that can be used to predict numerical targets. This is done with a function of inputs, the predictors in a data set, which are applied a series of weight coefficients and added to a general intercept. The function does not need to result in a straight line, as the inputs can be polynomial, and both numeric and categorical predictors can be used, with the latter having unique coefficients for each factor in the data set. For this data set, regression will be used to predict the hourly bike rentals based on data from a service in Washington D.C.

Data Import

```
bike_sharing <- read.csv("C:/Users/Yixin Sun/Downloads/hour.csv")
```

The columns in this data set are: instant (index) dteday (date string) season yr (year starting from 2011) month hr (hour of the day) holiday (whether or not the day was a holiday) weekday (numerical day of the week) workingday (whether or not the day was not a holiday or on a weekend) weathersit (numerical representation of how bad the weather was outside) temp (temperature) atemp (adjusted feeling temperature) hum (humidity) windspeed casual (count of casual bike users) registered (count of registered bike users) cnt (count of bike users)

Cleanup on Factors

Columns converted to factors if they are categorical in nature

```
attach(bike_sharing)

bike_sharing <- subset(bike_sharing, select = -c(instant, dteday))

bike_sharing$holiday <- factor(holiday, labels=c(TRUE, FALSE))
bike_sharing$season <- factor(season, labels=c("Spring", "Summer", "Autumn", "Winter"))
bike_sharing$workingday <- factor(workingday, labels=c("Not Working", "Working"))
bike_sharing$weathersit <- factor(weathersit, labels=c("Clear", "Overcast", "Light Prec.", "Heav
y Prec."))

detach(bike_sharing)
```

Train and test sets for linear regression

Divide into 80% train and 20% test

```
set.seed(1477)
i <- sample(1:nrow(bike_sharing), 0.8*nrow(bike_sharing), replace=FALSE)
train <- bike_sharing[i,]
test <- bike_sharing[-i,]
```

Data Exploration

Numerical exploration using various functions, including finding correlation between numerical fields

```
attach(train)

wind_median <- median(windspeed)
print(paste("Median Windspeed: ", wind_median))
```

```
## [1] "Median Windspeed:  0.1642"
```

```
wind_range <- range(windspeed)
print(paste("Range of Windspeed: ", wind_range))
```

```
## [1] "Range of Windspeed:  0"      "Range of Windspeed:  0.8507"
```

```
temp_median <- median(temp)
print(paste("Median Temperature: ", temp_median))
```

```
## [1] "Median Temperature:  0.5"
```

```
temp_range <- range(temp)
print(paste("Range of Temperature: ", temp_range))
```

```
## [1] "Range of Temperature:  0.02" "Range of Temperature:  1"
```

```
hum_mean <- mean(hum)
print(paste("Mean Humidity: ", hum_mean))
```

```
## [1] "Mean Humidity:  0.627989642523196"
```

```
hum_range <- range(hum)
print(paste("Range of Humidity: ", hum_range))
```

```
## [1] "Range of Humidity:  0" "Range of Humidity:  1"
```

```
total_cas <- sum(casual)
print(paste("Total Casual Sharing: ", total_cas))
```

```
## [1] "Total Casual Sharing:  495691"
```

```
total_reg <- sum(registered)
print(paste("Total Registered Sharing: ", total_reg))
```

```
## [1] "Total Registered Sharing:  2138449"
```

```
summary(registered)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    34.0   115.0   153.8   220.0   886.0
```

```
summary(cnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    40.0   142.0   189.5   281.0   977.0
```

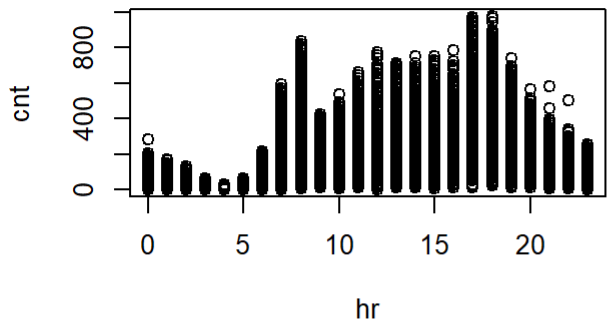
```
cor(train[, unlist(lapply(train, is.numeric))], use="complete")
```

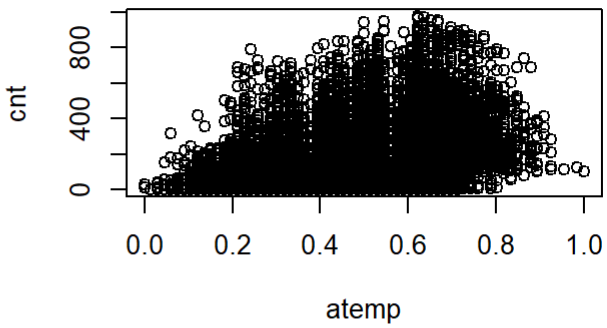
```
##                     yr         mnth           hr       weekday          temp
## yr          1.000000000 -0.008670787 -0.005456282 -0.0023655035  0.0400806380
## mnth       -0.008670787  1.000000000 -0.009195440  0.0169941685  0.1996333905
## hr         -0.005456282 -0.009195440  1.000000000  0.0018469791  0.1351608888
## weekday    -0.002365504  0.016994169  0.001846979  1.0000000000 -0.0008183826
## temp        0.040080638  0.199633391  0.135160889 -0.0008183826  1.0000000000
## atemp       0.038281151  0.206198020  0.131114042 -0.0082433970  0.9875824073
## hum        -0.082764378  0.161115578 -0.283033758 -0.0342871387 -0.0762893162
## windspeed  -0.010804718 -0.137341165  0.137766524  0.0120391542 -0.0228987647
## casual      0.140046700  0.072761844  0.302397024  0.0376629845  0.4622764318
## registered  0.254728700  0.123889943  0.370364219  0.0229816780  0.3312901412
## cnt         0.250811586  0.123246611  0.391479616  0.0294216386  0.4022336650
##                   atemp         hum   windspeed      casual  registered
## yr          0.038281151 -0.08276438 -0.01080472  0.14004670  0.25472870
## mnth        0.206198020  0.16111558 -0.13734117  0.07276184  0.12388994
## hr          0.131114042 -0.28303376  0.13776652  0.30239702  0.37036422
## weekday    -0.008243397 -0.03428714  0.01203915  0.03766298  0.02298168
## temp        0.987582407 -0.07628932 -0.02289876  0.46227643  0.33129014
## atemp       1.000000000 -0.05791937 -0.06179955  0.45682452  0.32890355
## hum        -0.057919374  1.00000000 -0.28857296 -0.34864781 -0.27198096
## windspeed  -0.061799547 -0.28857296  1.00000000  0.08623362  0.07814086
## casual      0.456824521 -0.34864781  0.08623362  1.00000000  0.50373130
## registered  0.328903554 -0.27198096  0.07814086  0.50373130  1.00000000
## cnt         0.398760109 -0.32184485  0.08868482  0.69224472  0.97212127
##                    cnt
## yr          0.25081159
## mnth        0.12324661
## hr          0.39147962
## weekday     0.02942164
## temp        0.40223366
## atemp       0.39876011
## hum        -0.32184485
## windspeed   0.08868482
## casual      0.69224472
## registered  0.97212127
## cnt         1.00000000
```

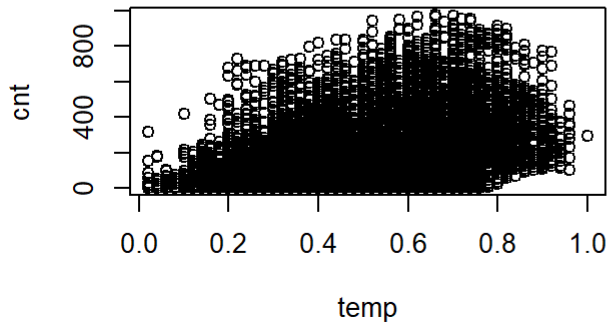
```
detach(train)
```

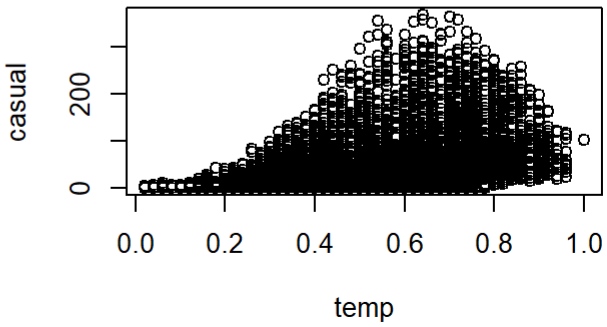
Plotting pairs with promising correlations Plots between casual/registered/cnt avoided because cnt is merely the sum of casual and registered

```
attach(train)
par(mfrow=c(2,2))
plot(hr, cnt)
plot(atemp, cnt)
plot(temp, cnt)
plot(temp, casual)
```










```
detach(train)
```

Simple Linear Model

Create the model with count as a function of temperature

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

```
##
## Call:
## lm(formula = cnt ~ temp, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -290.54 -110.04  -32.85   76.68  744.78
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.595      3.890    0.41    0.682
## temp         378.134      7.300   51.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 165.8 on 13901 degrees of freedom
## Multiple R-squared:  0.1618, Adjusted R-squared:  0.1617
## F-statistic:  2683 on 1 and 13901 DF,  p-value: < 2.2e-16
```

The resulting simple regression indicates that for every degree in Celsius there are about 378 more bike rentals for that hour. Based on the residual standard error, the residuals are about 165 bike rentals off from the prediction on average. The adjusted r-squared value is found by dividing the RSE by the sample variance of the target field, and this is a scaled goodness of fit metric. The r-squared is close to 0, so the model does not fit the data very accurately. The p-value is incredibly small, though, so the null hypothesis can be rejected; that is, the variation in bike rentals can be attributed somewhat to temperature rather than just chance.

Predict with test data and evaluate

```
attach(test)
pred <- predict(slm, newdata=test)

correlation <- cor(pred, cnt)
print(paste("Correlation: ", correlation))
```

```
## [1] "Correlation:  0.414859907116087"
```

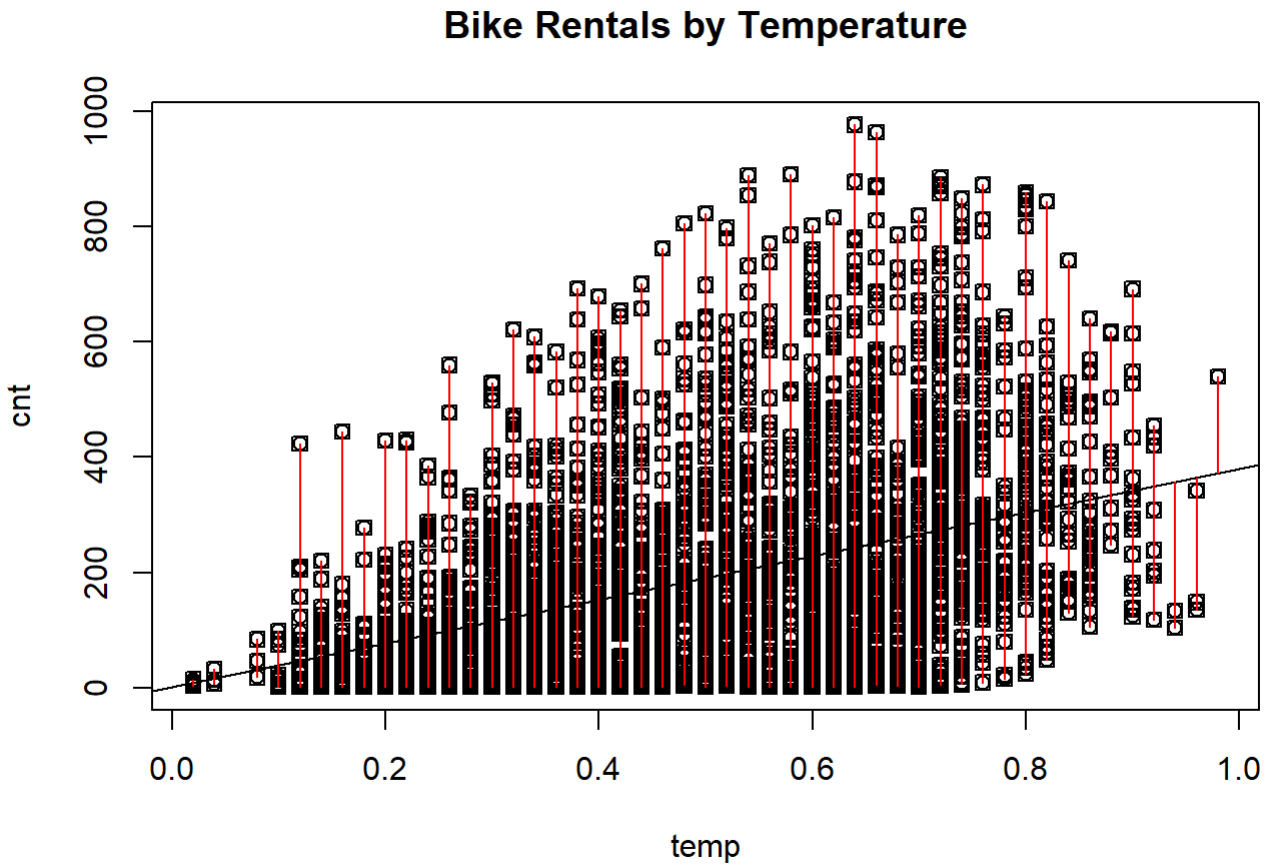
```
mse <- mean((pred - cnt)^2)
rmse <- sqrt(mse)
print(paste("RMSE: ", rmse))
```

```
## [1] "RMSE:  166.291519450613"
```

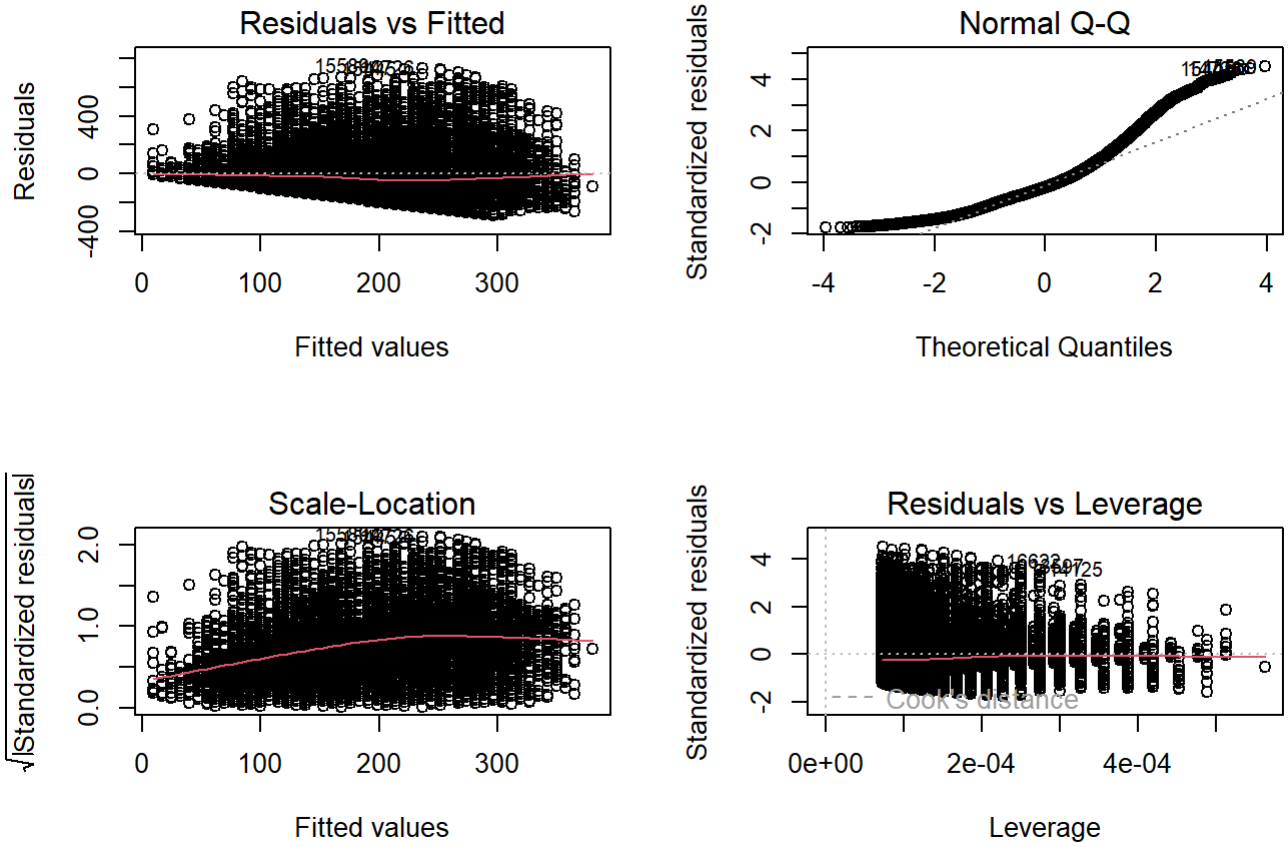
```
detach(test)
```

Plot residuals in red

```
attach(test)

plot(temp, cnt, main="Bike Rentals by Temperature")
abline(slm)
points(temp, cnt, pch=0)
segments(temp, cnt, temp, pred, col="red")
```




```
par(mfrow=c(2,2))
plot(slm)
```




```
detach(test)
```

The residuals do not seem to be following a certain pattern, though the Normal Q-Q graph indicates that residuals are deviating more with very high or low counts. The scale location plot indicates that deviation is increasing with higher temperatures, which can also be seen with the custom residual plot comparing with test data. The leverage graph indicates one potential leverage point that is skewing the model down, but the effect seems to be minor. It is clear that there is too much variation in the data to form a suitable linear regression with just temperature.

Multiple Linear Model

Create the model with count as a function of temperature, wind speed, and hour of the day

```
mlm <- lm(cnt~temp+windspeed+hr, data=train)
summary(mlm)
```

```
##
## Call:
## lm(formula = cnt ~ temp + windspeed + hr, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -321.48 -102.19  -31.23   58.22  710.08
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -93.6235     4.4182 -21.191  < 2e-16 ***
## temp        336.5130     6.8347  49.236  < 2e-16 ***
## windspeed    74.9228    10.7744   6.954 3.71e-12 ***
## hr            8.8103     0.1923  45.805  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 153.6 on 13899 degrees of freedom
## Multiple R-squared:  0.2801, Adjusted R-squared:  0.2799
## F-statistic:  1802 on 3 and 13899 DF,  p-value: < 2.2e-16
```

```
anova(slm, mlm)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	13901	381968958	*NA*	*NA*	*NA*	*NA*
2	13899	328075996	2	53892963	1141.593	0

2 rows

Predict with test data and evaluate

```
attach(test)
predm <- predict(mlm, newdata=test)

correlation <- cor(predm, cnt)
print(paste("Correlation: ", correlation))
```

```
## [1] "Correlation:  0.545324295460418"
```

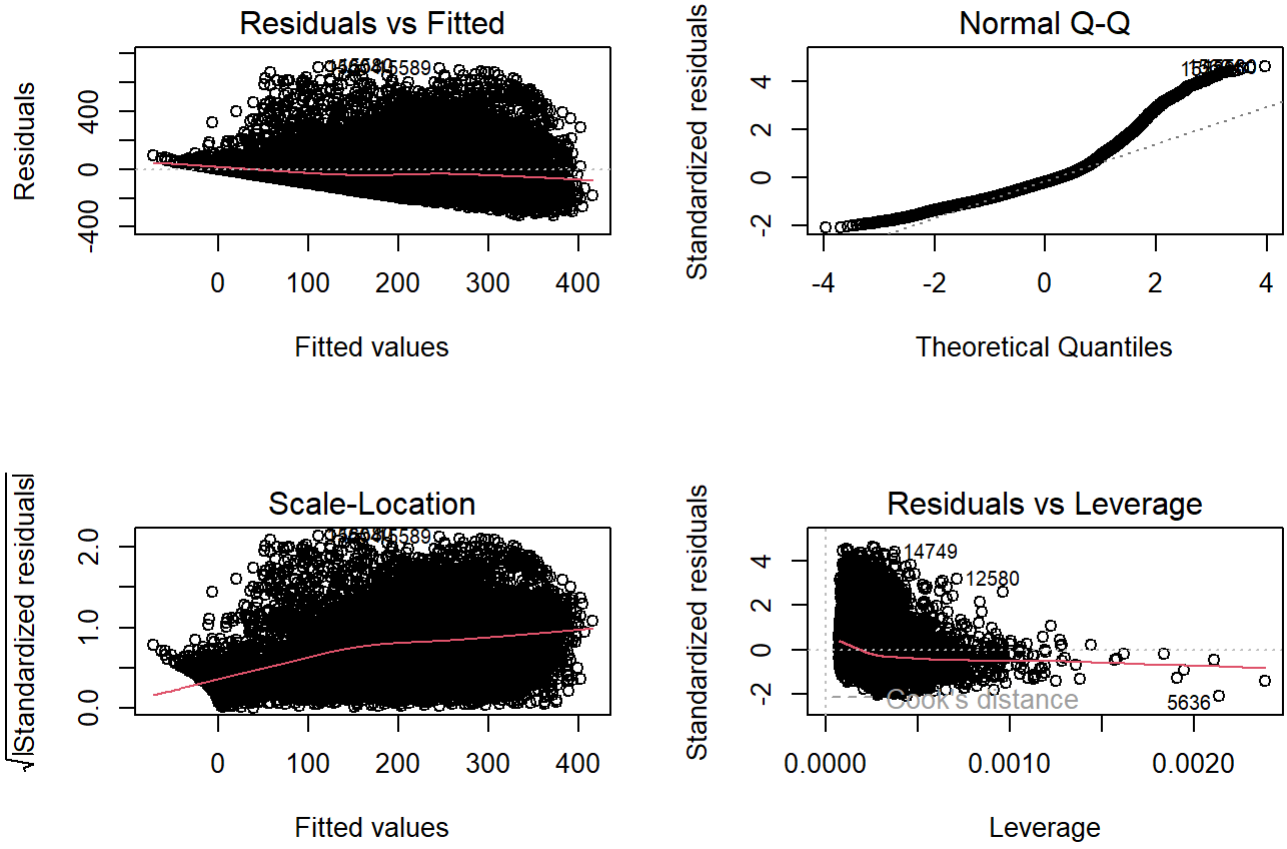
```
mse <- mean((predm - cnt)^2)
rmse <- sqrt(mse)
print(paste("RMSE: ", rmse))
```

```
## [1] "RMSE:  153.202937202661"
```

```
detach(test)
```

Plot residuals of multiple regression

```
par(mfrow=c(2,2))
plot(mlm)
```

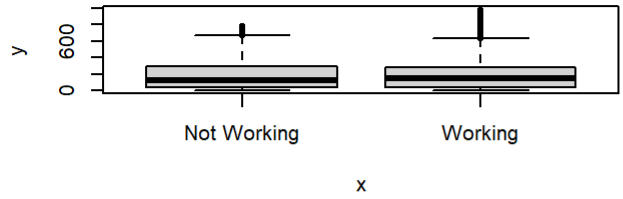


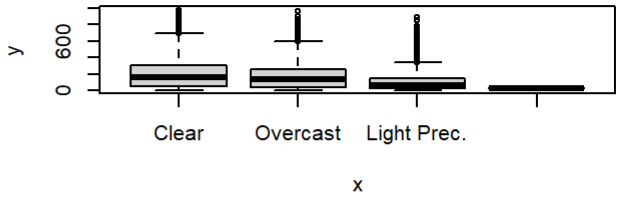
The issues and patterns with residuals actually became more pronounced with more data. Deviation is even more dependent on the individual predictors, increasing with higher wind speeds, temperatures, and hours, the Normal Q-Q indicates a more pronounced pattern off of the desired residual values, and there are more issues with leverage from high-value inputs.

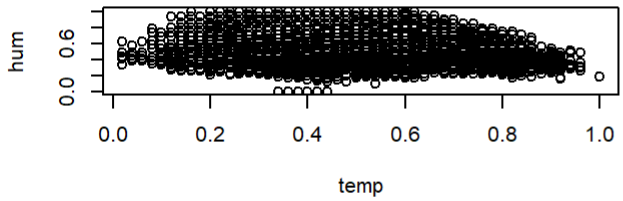
Data Exploration on Factors

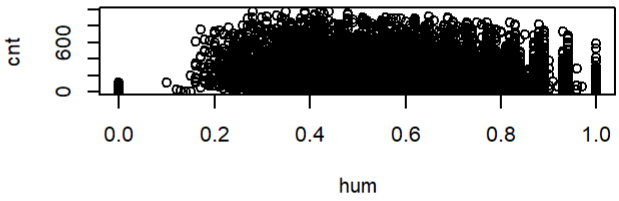
Trying to find other suitable predictors

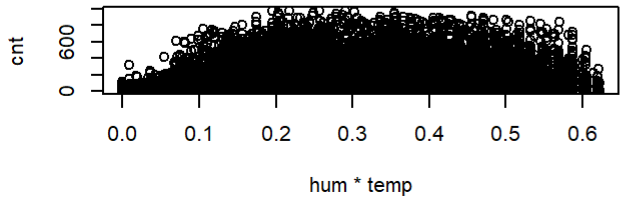
```
attach(train)

par(mfrow=c(3,2))
plot(workingday, cnt)
plot(weathersit, cnt)

plot(temp, hum)
plot(hum, cnt)
plot(hum*temp, cnt)

detach(train)
```











Humidity and working day seem unhelpful, weather has a trend

Polynomial Regression

Finding a better degree for temperature, the predictor with the highest correlation with count

```
attach(train)

for(degree in 1:4) {
  fm <- lm(cnt ~ poly(temp, degree), data = train)
  assign(paste("train", degree, sep = "."), fm)
}
anova(train.1, train.2, train.3, train.4)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	13901	381968958	*NA*	*NA*	*NA*	*NA*
2	13900	381874960	1	93998.32	3.422967	0.06431658
3	13899	381813409	1	61551.04	2.241393	0.13438262
4	13898	381653918	1	159491.33	5.807907	0.01596718

4 rows

```
detach(train)
```

Final Linear Regression

Create the model with count as a function of polynomial temperature, wind speed, precipitation, and hour of the day

```
flm <- lm(cnt~poly(temp, 3)+windspeed+weathersit+hr, data=train)
summary(flm)
```

```
##
## Call:
## lm(formula = cnt ~ poly(temp, 3) + windspeed + weathersit + hr,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -317.87 -101.24  -30.34   57.92  707.97
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              79.4312     3.2278  24.608  < 2e-16 ***
## poly(temp, 3)1         7435.6309   154.5580  48.109  < 2e-16 ***
## poly(temp, 3)2         -227.5578   153.6829  -1.481  0.13871
## poly(temp, 3)3         -363.8822   152.7554  -2.382  0.01723 *
## windspeed                86.6215    10.7084   8.089 6.51e-16 ***
## weathersitOvercast       -8.7429     3.0098  -2.905  0.00368 **
## weathersitLight Prec.   -80.4493     4.8574 -16.562  < 2e-16 ***
## weathersitHeavy Prec.   -44.3543   107.6453  -0.412  0.68032
## hr                        8.8736     0.1912  46.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 152.1 on 13894 degrees of freedom
## Multiple R-squared:  0.2943, Adjusted R-squared:  0.2939
## F-statistic: 724.2 on 8 and 13894 DF,  p-value: < 2.2e-16
```

```
anova(mlm, flm)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	13899	328075996	*NA*	*NA*	*NA*	*NA*
2	13894	321591174	5	6484821	56.03394	7.159688e-58

2 rows

Predict with test data and evaluate

```
attach(test)
predf <- predict(flm, newdata=test)

correlation <- cor(predf, cnt)
print(paste("Correlation: ", correlation))
```

```
## [1] "Correlation:  0.558535476626803"
```

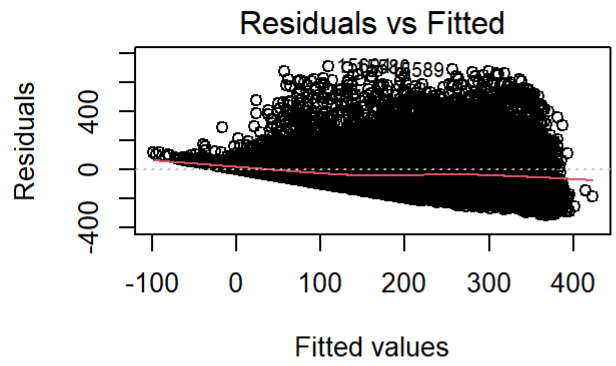
```
mse <- mean((predf - cnt)^2)
rmse <- sqrt(mse)
print(paste("RMSE: ", rmse))
```

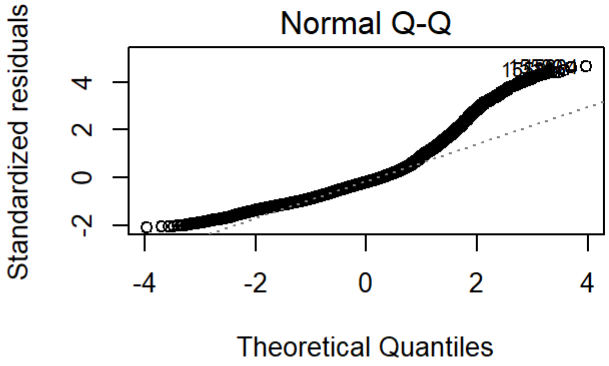
```
## [1] "RMSE:  151.612485394234"
```

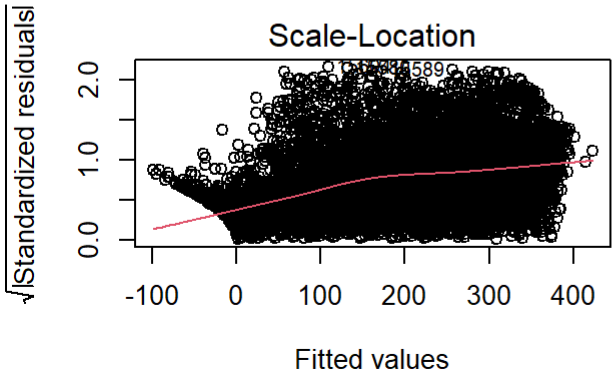
```
detach(test)
```

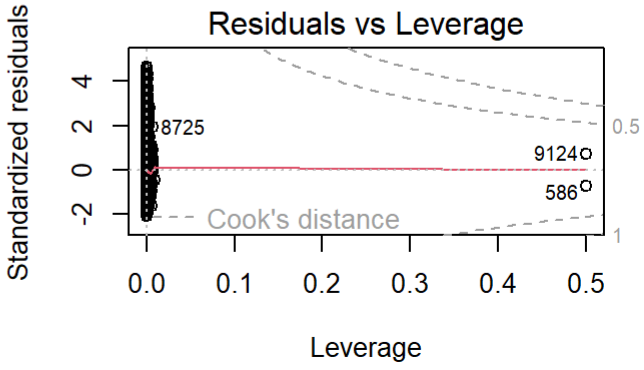
Plot residuals of multiple regression

```
par(mfrow=c(2,2))
plot(flm)
```









Based on the summaries of each model and the correlations and root mean squared error between each prediction and test data, it does seem that the changes and additional parameters slightly improved each linear regression. Residual standard error decreased within the linear models by about 14 hourly bike rentals, and the R-squared increased from 0.1617 to 0.2939, which is significantly better but still not suitable for highly accurate predictions. There seemed to be a greater difference between the simple regression and the multiple regression than the multiple regression and the final regression, as the differences in RSE and R-squared are apparent but small. Despite this, the final model is still the best of the three in a vacuum. The evaluations on test data indicate that the model improvements were not the result of over-fitting training data, for correlation increased from 0.4148 to 0.5585 across the models, and RMSE decreased from about 166 bike rentals to about 151 bike rentals. These metrics similarly indicate that there was a greater improvement from adding wind speed and hour as predictors

than from adding weather and polynomial temperature in the final model, but there still was some improvement based on the test evaluations. Polynomial inputs allowed a more fine-grained model for temperature without forcing a straight line, and the addition of a significant input with several factors also helped. The improvements were not as strong as they could have been because some of the variables are confounding, with high wind speed correlating with the weather situation and temperature correlating with the hour. Much higher correlations and root mean squared errors could have been achieved by using the number of registered rentals as a predictor, but that clearly goes against the intent of the data set.