

# Appendix: Pedestrians on the Brooklyn Bridge

## Brief Description of Dataset:

Our dataset is called the **Brooklyn Bridge Automated Pedestrian Counts Demonstration Project**. It is publicly available through **NYC Open Data** and can be found at this link: [https://data.cityofnewyork.us/Transportation/Brooklyn-Bridge-Automated-Pedestrian-Counts-Demonstration/6fi9-q3ta/about\\_data](https://data.cityofnewyork.us/Transportation/Brooklyn-Bridge-Automated-Pedestrian-Counts-Demonstration/6fi9-q3ta/about_data). The New York City Department of Transportation is testing automated technology to count pedestrians, and this dataset keeps track of all records. The records are taken hourly from a counter located on the Manhattan approach of the Brooklyn Bridge. It contains **16,100 entries** with **12 variables**; however, we did not utilize all of the variables.

The columns included in the dataset are hour\_beginning, location, Pedestrians (sum of towards Manhattan and towards Brooklyn), Pedestrians Toward Brooklyn, weather\_summary (a categorical variable), temperature, precipitation, latitude, longitude, events, and Location1. We dropped the following columns: Towards Manhattan, Towards Brooklyn, location, latitude, longitude, and Location1. Events is a column that indicates whether there is a holiday that may affect the number of people walking on the bridge. We changed this column to has\_event, a categorical column with a value of 1 for an event and 0 for no event.

**Our Final Variables:** Pedestrians (y), weather\_summary, temperature, precipitation, has\_event, hour, weekday, and month

---

## Our Code:

```
#imported the file as pedestrians, include headeres, do NOT select as factor  
df = pedestrians
```

```
#dropping useless cols  
df$Towards.Brooklyn = NULL  
df$Towards.Manhattan = NULL  
df$lat = NULL
```

```

df$long = NULL
df$location = NULL
df$Location1 = NULL
#-----
#this is checking the unique values for each col so we can see the NAs. I ran these lines after to
see if there were still NAs
unique_values <- unique(df$weather_summary)
unique_values
unique_valuestemp <- unique(df$temperature)

#-----
#changing event column to be boolean
df$has_event <- ifelse(df$events == "" | is.na(df$events), 0, 1)
df$events <- NULL

#dropping NAs for specified cols.
#this is every column except events because events
df_clean <- df[complete.cases(df$hour_beginning, df$Pedestrians, df$weather_summary,
df$temperature), ]
unique_valuestemp <- unique(df_clean$temperature)
df <- df[df$weather_summary != "", ]

#setting our working dataset to df_clean now
df = df_clean

#-----

#Converting Ped. Count to numeric had to remove commas because they were being treated as
chars
df$Pedestrians <- as.numeric(gsub(",", "", df$Pedestrians))

#quick check
uniques <- unique(df$Pedestrians)
uniques

#weather needs to be factor
df$weather_summary <- as.factor(df$weather_summary)
#-----

#checking what type of events there are

```

```

uniques <- unique(df$events)

#df$events<-as.factor(df$events)
df$hour_beginning <- as.POSIXct(df$hour_beginning,format = "%Y %b %d %I:%M:%S %p")
df$hour <- lubridate::hour(df$hour_beginning)
df$weekday <- lubridate::wday(df$hour_beginning, label = TRUE)
df$month <- lubridate::month(df$hour_beginning, label = TRUE)
df$hour_beginning <- NULL

#-----
#make base model to see what the R2 is and run variable selection.
base_model <- lm(Pedestrians ~ ., data=df)
summary(base_model)
#our F score shows our model is significant enough to use! p-val < 2.2e-16

#variable selection:
library(olsrr) #variable screening package
ols_step_both_p(base_model, penter=0.05, prem=0.1, details=TRUE)
#variable selection confirms keeping all variables. R^2 adjusted is .59 now

#-----
#PLOTTING to determine transformations on x and higher order terms
library(ggplot2)
ggplot(df, aes(x = temperature, y = Pedestrians)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Temperature vs Pedestrians", x = "Temperature", y = "Pedestrians")
#shows a linear trend

ggplot(df, aes(x = weather_summary, y = Pedestrians)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Weather Summary vs Pedestrians", x = "Weather Summary", y = "Pedestrians")

ggplot(df, aes(x = precipitation, y = Pedestrians)) +
  geom_point(color = "blue") +
  labs(title = "Precipitation vs Pedestrians", x = "Precipitation", y = "Pedestrians")

#applying this transformation to make it a more linear trend
#tries to linearize relationship a bit. does spread it out more. Not fully.

```

#In the model there should be a higher order term on precipitation.

```
ggplot(df, aes(x = sqrt(precipitation), y = Pedestrians)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm", se = FALSE, color = "red") +  
  labs(title = "Square Root of Precipitation vs Pedestrians", x = "Sqrt(Precipitation)", y =  
"Pedestrians")
```

#Conclusion here: Looks like a higher order term would help fit model better. Hour<sup>3</sup>/ Hour<sup>2</sup>

```
ggplot(df, aes(x = hour, y = Pedestrians)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm", se = FALSE, color = "red") +  
  labs(title = "Hour vs Pedestrians", x = "Hour", y = "Pedestrians")
```

```
ggplot(df, aes(x = month, y = Pedestrians)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm", se = FALSE, color = "red") +  
  labs(title = "Month vs Pedestrians", x = "Month", y = "Pedestrians")
```

```
ggplot(df, aes(x = weekday, y = Pedestrians)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm", se = FALSE, color = "red") +  
  labs(title = "Weekday vs Pedestrians", x = "Weekday", y = "Pedestrians")
```

```
ggplot(df, aes(x = has_event, y = Pedestrians)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm", se = FALSE, color = "red") +  
  labs(title = "Has event vs pedestrians", x = "Has Event", y = "Pedestrians")
```

#

#Now we have to test correlation between all our variables. Using Variable Inflation Scores

#i have to use the base model here because higher order terms were in my other models

```
library(car)
```

```
vif(base_model)
```

#No signification VIF scores. all less than 10.

#

#lets rerun base model again

```
base_model <- lm(Pedestrians ~ weather_summary + temperature + precipitation + has_event  
+ hour + weekday + month, data=df)
```

```
summary(base_model) #R^2 adjusted is 0.5878
```

```
#apply the precipitation transformation from graph and add a higher order term for precipitation
```

```
df$precipitation = sqrt(df$precipitation)
```

```
second_model <- lm(Pedestrians ~ . + I(precipitation^2), data=df)
```

```
summary(second_model) #R^2 adjusted is 0.5901, not much of a change
```

```
#apply the higher order terms on hour
```

```
third_model <- lm(Pedestrians ~ . + I(precipitation^2) + I(hour^2) + I(hour^3), data=df)
```

```
summary(third_model) #much better R^2 adjusted. R^2 adjusted is 0.7116
```

```
#going to try adding some interaction terms
```

```
#temperature and hour probably interact....
```

```
#does the effect of temperature depends on the time of day?
```

```
fourth_model <- lm(Pedestrians ~ . + I(precipitation^2) + I(hour^2) + I(hour^3) +
```

```
temperature:hour, data=df)
```

```
summary(fourth_model) #adjusted R^2 only went up a bit. Now it's 0.7191
```

```
#interaction of temperature and added event. R^2 adjusted is more or less the same.
```

```
bad_model <- lm(Pedestrians ~ . + I(precipitation^2) + I(hour^2) + I(hour^3) + temperature:hour
```

```
+ temperature:has_event, data=df)
```

```
summary(bad_model) #adjusted R^2 is 0.7195
```

```
#adding an interaction term for weather summary and hour.
```

```
#rain does not have the same effect at all hours. same for every other weather summary
```

```
possibility
```

```
fifth_model <- lm(Pedestrians ~ . + I(precipitation^2) + I(hour^2) + I(hour^3) +
```

```
temperature:hour + weather_summary:hour, data=df)
```

```
summary(fifth_model) #R^2 adjusted here is 0.7437
```

```
#adding an interaction term for temperature and has event.
```

```
#temperature doesn't necessarily have the same effect on an event vs no event.
```

```
sixth_model <- lm(Pedestrians ~ . + I(precipitation^2) + I(hour^2) + I(hour^3) +
```

```
temperature:hour + weather_summary:hour + temperature:has_event, data=df)
```

```
summary(sixth_model) #R^2 adjusted here is 0.744. Not worth adding the interaction term
```

```
#at this point we have tested all interaction terms we think may exist. These are the ones that we kept
```

```
final_model <- lm(Pedestrians ~ . + I(precipitation^2) + I(hour^2) + I(hour^3) +
```

```
temperature:hour + weather_summary:hour, data=df)
```

```
summary(final_model) #final R^2 adjust here is 0.7437
```

```
#-----
```

```
install.packages("lmtest")
```

```
library(lmtest)
```

```
#running the Durbin-Watson d-Test for correlation since this deals with time.
```

```
#d ~ 2, residuals are uncorrelated
```

```
dwtest(final_model)
```

```
#-----
```

```
#RESIDUAL ANALYSIS
```

```
# plotting the residuals vs the variable "temperature":
```

```
plot(df$temperature, resid(final_model),
```

```
  xlab = "Temperature",
```

```
  ylab = "Regression Residuals",
```

```
  main = "Regression Residuals vs Temperature")
```

```
# adding a horizontal reference line at residual = 0 to the plot
```

```
abline(h = 0, col = "red")
```

```
#Residuals seem to be more or less randomly distributed. This is good.
```

```
# plotting the residuals vs the variable "hour" for the second model:
```

```
plot(df$hour, resid(final_model),
```

```
  xlab = "hour",
```

```
  ylab = "Regression Residuals",
```

```
  main = "Regression Residuals vs Hour")
```

```
# adding a horizontal reference line at residual = 0 to the plot
```

```
abline(h = 0, col = "red")
```

```
#This shows an issue with our model. Residuals are not random around hour.
```

```
#This may be because its not treating the hour as a cycle. hour 23 is only 1 hour away from hour  
0
```

```
#Instead of applying a transformation, we will make the hour a factor that way they are treated  
separately.
```

```
dfHour_asFactor = df
```

```
dfHour_asFactor$hour <- as.factor(df$hour)
```

```
seventh_model <- lm(Pedestrians ~ . + I(precipitation^2) + temperature:hour,
```

```
data=dfHour_asFactor) #note: got rid of the last interaction term because it was giving NAs
```

```
summary(seventh_model) #this model performs the best. It has an R^2 adjusted of 0.8181
#we got rid of the higher order terms on hour in this situation
#Now, every hour is technically allowed to have a different residual behavior
```

```
#lets keep moving with seventh_model and hour being treated as a factor
df = dfHour_asFactor
final_model <- lm(Pedestrians ~ . + I(precipitation^2) + temperature:hour,
data=dfHour_asFactor)
summary(final_model)
```

```
#not fair to judge based off of just precipitation, there are higher order terms
# plotting the residuals vs the precipitation:
plot(df$precipitation, resid(final_model),
     xlab = "1/Precipitation",
     ylab = "Regression Residuals",
     main = "Regression Residuals vs 1 over precipitation")
# adding a horizontal reference line at residual = 0 to the plot
abline(h = 0, col = "red")
```

```
#code from notes
plot(df$Pedestrians, resid(final_model),
     xlab = "Pedestrians",
     ylab = "Residuals",
     main = "Residuals vs Pedestrians")
abline(h = 0, col = "red", lwd = 2)
#very skewed increasing residuals as pedestrians increase
#multiplicative error?
#applying log transformation
```

```
plot(log(df$Pedestrians), resid(final_model),
     xlab = "Pedestrians",
     ylab = "Residuals",
     main = "Pedestrians vs Residuals")
abline(h = 0, col = "red", lwd = 2)
#log transformation definitely helped randomize our residuals.
```

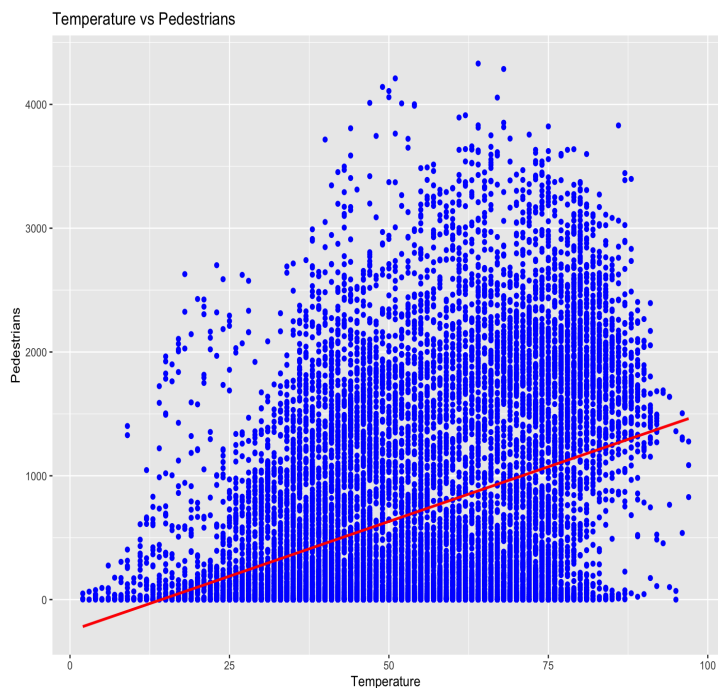
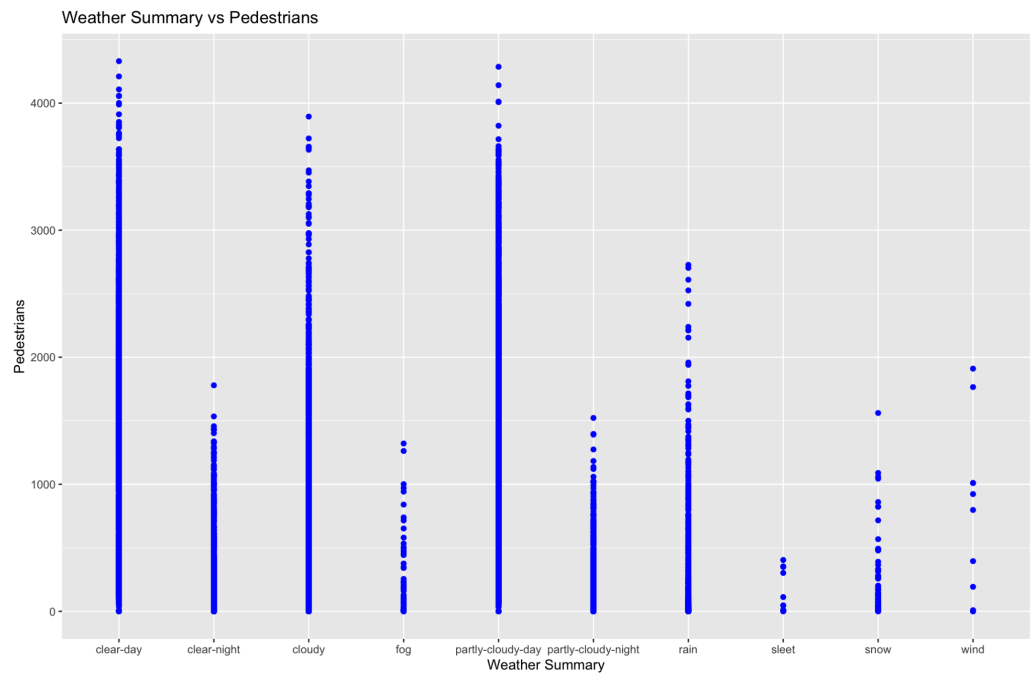
```
dfcopy = df
dfcopy$Pedestrians = log(dfcopy$Pedestrians + 1 ) #have to add 1 because log doesnt take in 0
final_model <- lm(Pedestrians ~ . + I(precipitation^2) + temperature:hour, data=dfcopy)
summary(final_model) #adjusted R-squared is now 0.853 because of transformation.
```

---

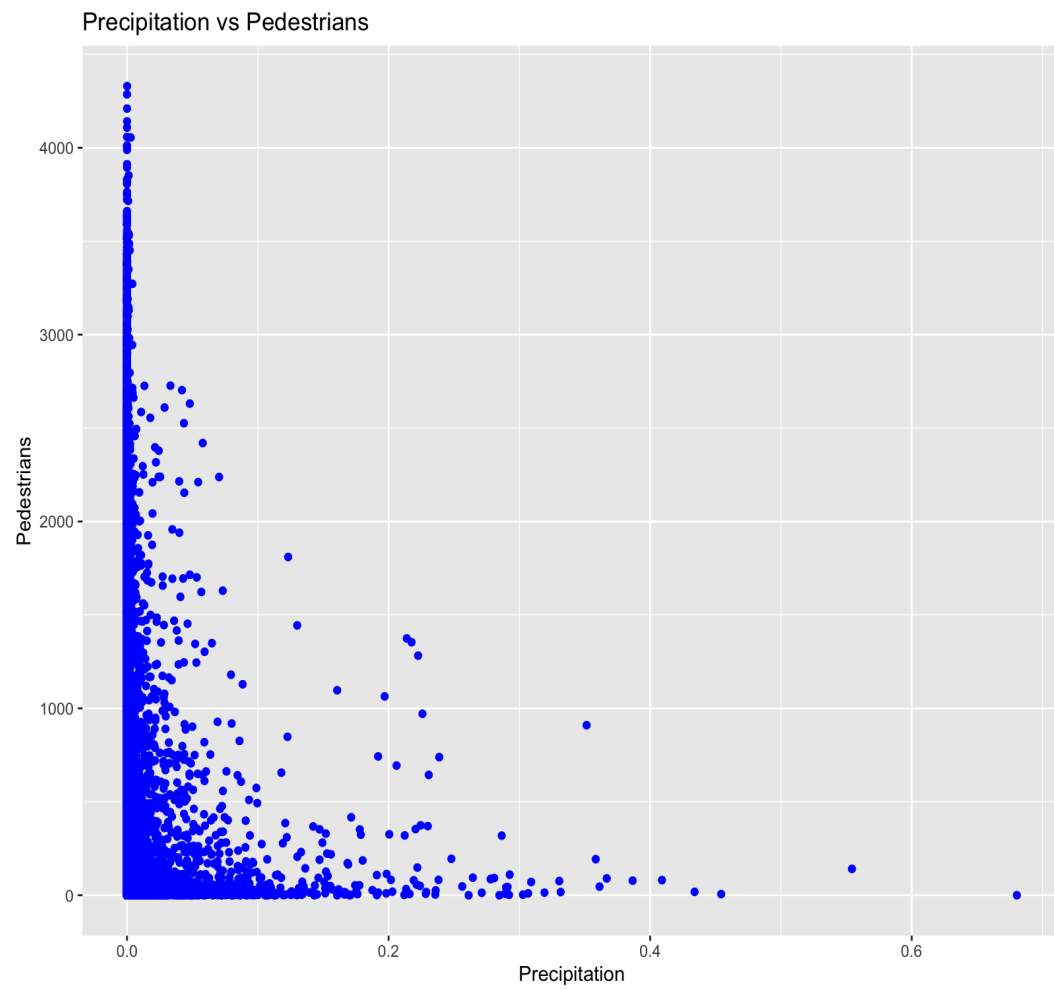
Note: The goal of our project is to make predictions, not to interpret the model. We chose to use a simpler base model in our presentation for interpretation. The code above shows models with higher order terms, interaction terms, and transformations on our x's and Pedestrian count variable (y)

---

## Our Plots:

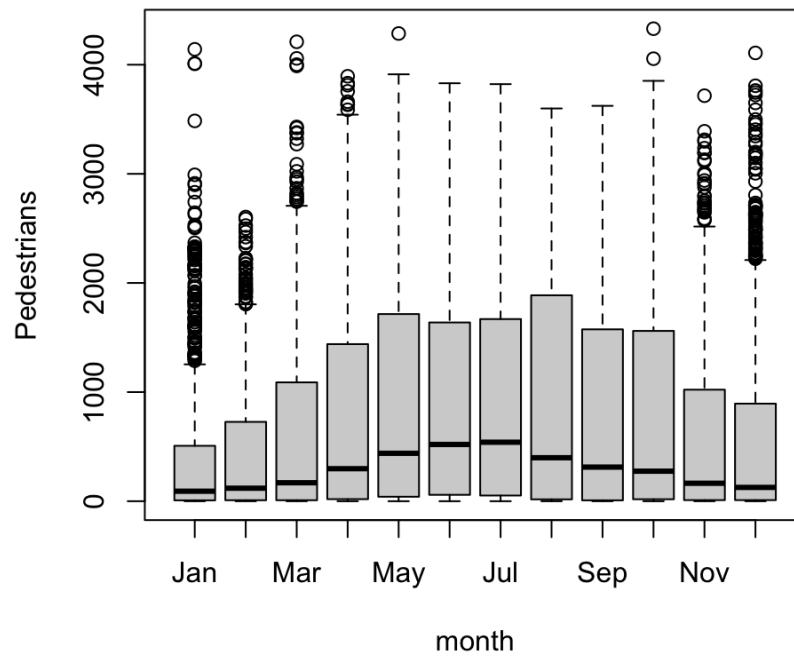




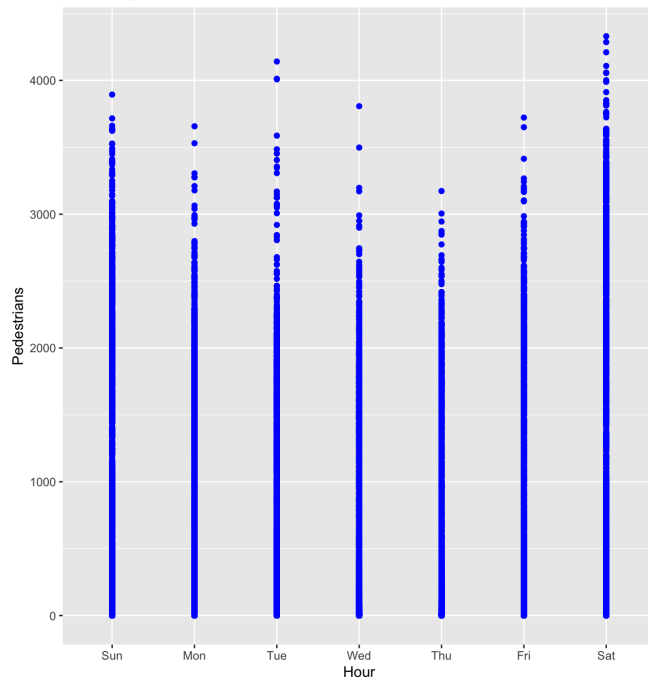


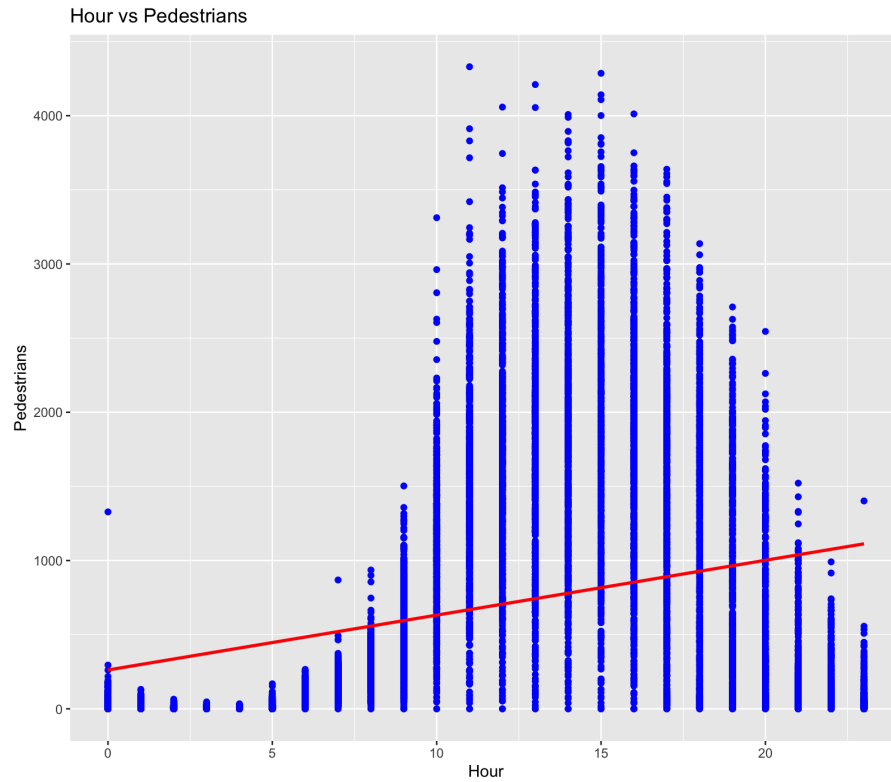
- The plot above led us to transform the Precipitation variable. We transformed it to be  $\text{precipitation} = 1/\text{precipitation}$

**Pedestrians by Month**

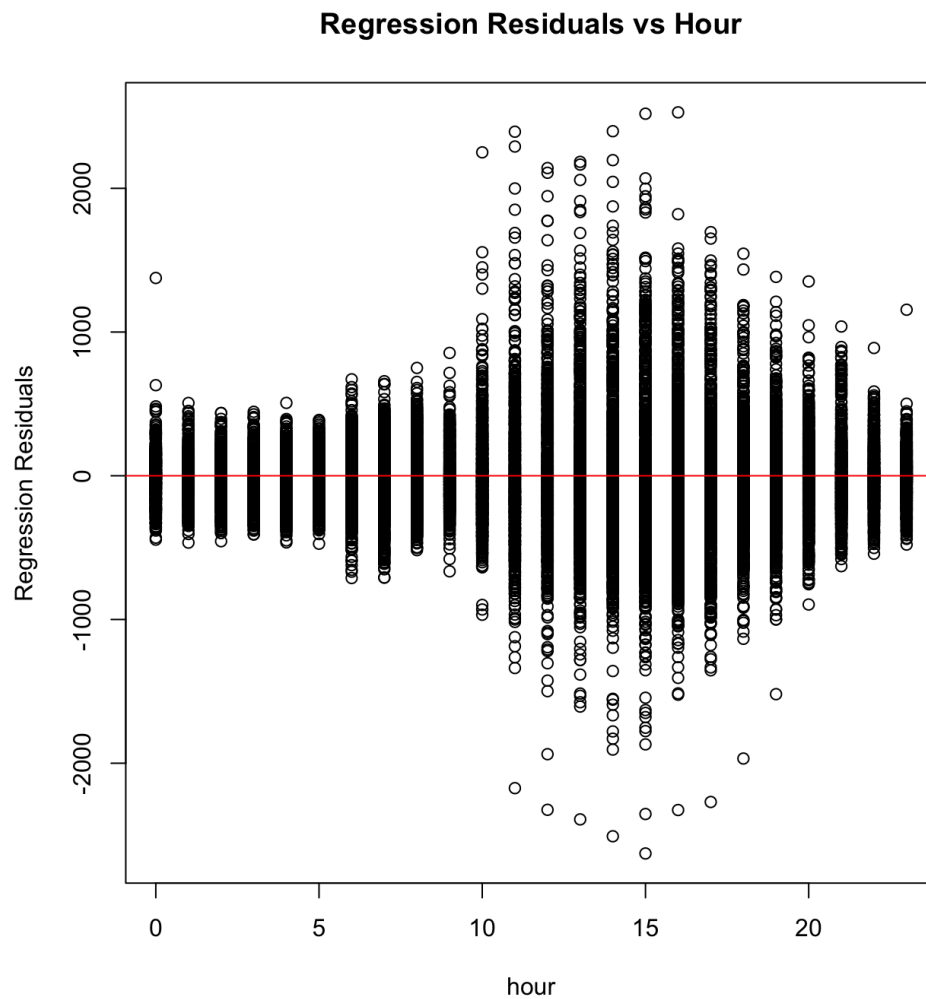


**Weekday vs Pedestrians**

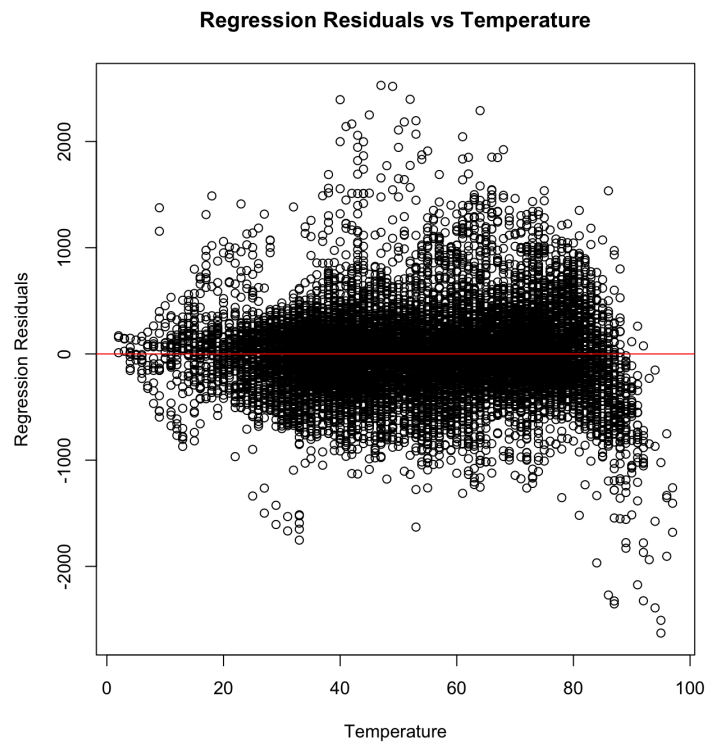




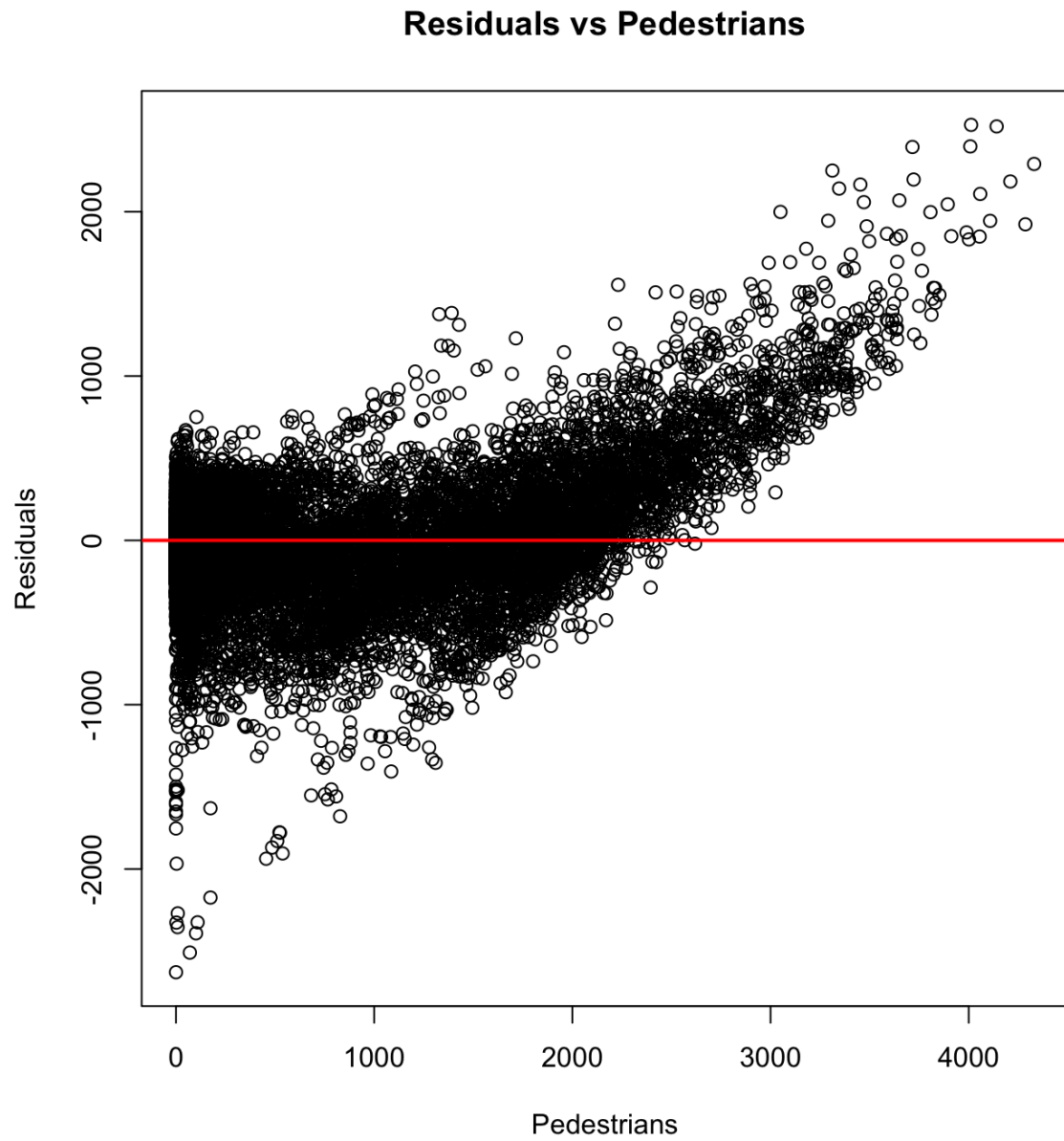
- The hour variable seemed to be cyclic. At first, we introduced higher-order terms such as  $x^2$  and  $x^3$  to help with this trend, while the  $R^2$  adjusted showed an increase in performance, our residuals relative to the hour variable were not showing a random distribution.



- Regression residuals are not randomly distributed. This led us to change the hour variable into a categorical variable where each hour can be treated separately. This increased our  $R^2$  adjusted to about 0.81, but this was not our final model.

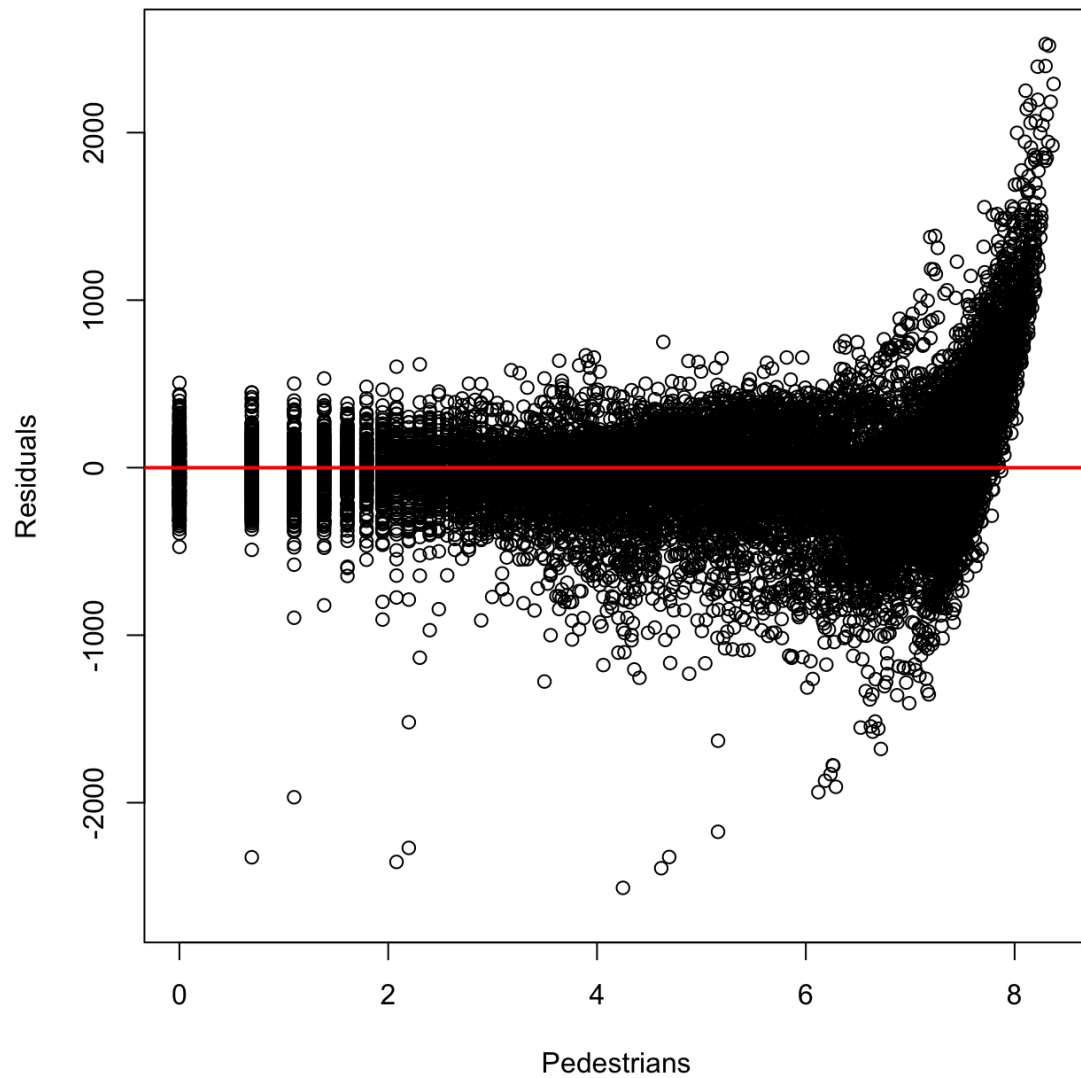


- This plot shows a more or less random distribution of residuals around the temperature variable.



- This is a plot of the Residuals relative to the number of pedestrians. The closest type of response we saw in this was the multiplicative response. So, we decided to apply the log transformation on the pedestrians variable to see if that would help.

## Pedestrians vs Residuals



- This plot is from after we applied the log transformation to pedestrians. Residuals are more randomly distributed.
-

## Our final model before the transformation on Pedestrians:

```
lm(formula = Pedestrians ~ . + I(precipitation^2) + temperature:hour,
    data = dfHour_asFactor)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2627.89	-183.63	-5.36	157.95	2528.55

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	551.6780	50.5618	10.911	< 2e-16 ***
weather_summaryclear-night	-490.5684	13.5337	-36.248	< 2e-16 ***
weather_summarycloudy	-402.9876	11.4774	-35.111	< 2e-16 ***
weather_summaryfog	-512.5531	43.3851	-11.814	< 2e-16 ***
weather_summarypartly-cloudy-day	-94.0359	9.5340	-9.863	< 2e-16 ***
weather_summarypartly-cloudy-night	-492.4687	14.0197	-35.127	< 2e-16 ***
weather_summaryrain	-453.8230	24.2382	-18.723	< 2e-16 ***
weather_summarysleet	-645.2121	100.8237	-6.399	1.60e-10 ***
weather_summarysnow	-444.7883	41.9040	-10.614	< 2e-16 ***
weather_summarywind	-510.1055	106.7736	-4.777	1.79e-06 ***
temperature	-0.3691	0.9101	-0.406	0.685090
precipitation	-1957.4437	141.1402	-13.869	< 2e-16 ***
has_event	60.4785	11.7247	5.158	2.52e-07 ***
hour1	-16.9620	63.8237	-0.266	0.790425
hour2	-10.1042	63.7241	-0.159	0.874016
hour3	-13.0998	63.4532	-0.206	0.836444
hour4	-14.9907	63.3150	-0.237	0.812843
hour5	-41.8586	63.1792	-0.663	0.507636
hour6	133.1337	63.3359	2.102	0.035567 *
hour7	25.7790	62.7357	0.411	0.681141
hour8	-338.4045	63.6766	-5.314	1.08e-07 ***
hour9	-238.1254	64.0953	-3.715	0.000204 ***
hour10	35.4831	64.4830	0.550	0.582141
hour11	305.2628	64.9508	4.700	2.62e-06 ***
hour12	423.9641	65.3810	6.485	9.16e-11 ***
hour13	523.3348	65.7057	7.965	1.77e-15 ***
hour14	536.5153	66.0356	8.125	4.81e-16 ***
hour15	597.8051	65.9941	9.058	< 2e-16 ***
hour16	399.6273	65.7432	6.079	1.24e-09 ***
hour17	-238.0244	64.6096	-3.684	0.000230 ***
hour18	-531.0593	64.5494	-8.227	< 2e-16 ***
hour19	-598.6229	64.4127	-9.294	< 2e-16 ***
hour20	-350.5651	64.5610	-5.430	5.72e-08 ***
hour21	-154.8276	64.5449	-2.399	0.016462 *
hour22	-70.9150	64.3552	-1.102	0.270508
hour23	-17.0655	64.1719	-0.266	0.790294
weekday.L	109.9624	7.8120	14.076	< 2e-16 ***

weekday.Q	265.8775	7.7397	34.352	< 2e-16 ***
weekday.C	30.7435	7.7085	3.988	6.69e-05 ***
weekday^4	39.5565	7.7061	5.133	2.88e-07 ***
weekday^5	0.3295	7.6985	0.043	0.965867
weekday^6	1.0018	7.7009	0.130	0.896500
month.L	75.9686	11.7377	6.472	9.94e-11 ***
month.Q	-84.2274	20.9060	-4.029	5.63e-05 ***
month.C	140.4231	11.8681	11.832	< 2e-16 ***
month^4	-17.7675	10.5777	-1.680	0.093033 .
month^5	-61.3865	10.6933	-5.741	9.60e-09 ***
month^6	121.8473	10.4807	11.626	< 2e-16 ***
month^7	13.4279	10.4251	1.288	0.197752
month^8	-9.7015	10.3389	-0.938	0.348079
month^9	79.1956	10.8380	7.307	2.86e-13 ***
month^10	53.1778	11.4320	4.652	3.32e-06 ***
month^11	21.4195	10.5336	2.033	0.042025 *
I(precipitation^2)	2214.7203	290.1699	7.632	2.43e-14 ***
temperature:hour1	-0.0964	1.1876	-0.081	0.935302
temperature:hour2	-0.3566	1.1911	-0.299	0.764660
temperature:hour3	-0.3675	1.1922	-0.308	0.757873
temperature:hour4	-0.3254	1.1953	-0.272	0.785449
temperature:hour5	0.5739	1.1976	0.479	0.631788
temperature:hour6	-3.4837	1.2107	-2.877	0.004014 **
temperature:hour7	-1.8257	1.1951	-1.528	0.126620
temperature:hour8	5.4106	1.1779	4.593	4.40e-06 ***
temperature:hour9	7.8634	1.1676	6.735	1.70e-11 ***
temperature:hour10	12.0122	1.1581	10.373	< 2e-16 ***
temperature:hour11	14.6553	1.1511	12.732	< 2e-16 ***
temperature:hour12	15.1625	1.1456	13.235	< 2e-16 ***
temperature:hour13	14.4153	1.1416	12.627	< 2e-16 ***
temperature:hour14	15.0381	1.1405	13.186	< 2e-16 ***
temperature:hour15	14.9150	1.1377	13.109	< 2e-16 ***
temperature:hour16	16.8045	1.1350	14.806	< 2e-16 ***
temperature:hour17	23.8409	1.1396	20.920	< 2e-16 ***
temperature:hour18	24.2168	1.1522	21.017	< 2e-16 ***
temperature:hour19	20.4796	1.1579	17.688	< 2e-16 ***
temperature:hour20	11.9812	1.1660	10.276	< 2e-16 ***
temperature:hour21	5.6998	1.1716	4.865	1.15e-06 ***
temperature:hour22	2.8030	1.1767	2.382	0.017226 *
temperature:hour23	0.9686	1.1804	0.821	0.411913
---				
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.' 0.1 ' ' 1
Residual standard error:	367.8 on 15964 degrees of freedom			
Multiple R-squared:	0.819,	Adjusted R-squared:	0.8181	
F-statistic:	950.2 on 76 and 15964 DF,	p-value:	< 2.2e-16	



## Our final model statistics after the transformation on Pedestrians:

Call:					weekday.Q					0.2247940 0.0202750 11.087 < 2e-16 ***				
lm(formula = Pedestrians ~ . + I(precipitation^2) + temperature:hour,					weekday.C					-0.0322853 0.0201932 -1.599 0.109880				
data = dfcopy)					weekday^4					0.0691049 0.0201870 3.423 0.000620 ***				
					weekday^5					-0.0136228 0.0201671 -0.675 0.499370				
Residuals:					weekday^6					0.0001959 0.0201733 0.010 0.992254				
Min 1Q Median 3Q Max					month.L					-0.0742967 0.0307482 -2.416 0.015691 *				
-8.0002 -0.4279 0.0719 0.5516 5.2584					month.Q					-0.3758522 0.0547653 -6.863 6.99e-12 ***				
					month.C					0.4968623 0.0310898 15.982 < 2e-16 ***				
Coefficients:					month^4					0.2015036 0.0277093 7.272 3.70e-13 ***				
					month^5					-0.3311949 0.0280122 -11.823 < 2e-16 ***				
(Intercept)					month^6					-0.0928273 0.0274553 -3.381 0.000724 ***				
weather_summaryclear-night					month^7					0.1725475 0.0273095 6.318 2.72e-10 ***				
weather_summarycloudy					month^8					0.1831114 0.0270837 6.761 1.42e-11 ***				
weather_summaryfog					month^9					0.1852161 0.0283912 6.524 7.06e-11 ***				
weather_summarypartly-cloudy-day					month^10					0.0281975 0.0299472 0.942 0.346425				
weather_summarypartly-cloudy-night					month^11					-0.0429430 0.0275938 -1.556 0.119668				
weather_summaryrain					I(precipitation^2)					-1.6400883 0.7601297 -2.158 0.030970 *				
weather_summarysleet					temperature:hour1					-0.0002356 0.0031109 -0.076 0.939628				
weather_summarysnow					temperature:hour2					-0.0068233 0.0031203 -2.187 0.028775 *				
weather_summarywind					temperature:hour3					-0.0075717 0.0031231 -2.424 0.015343 *				
temperature					temperature:hour4					-0.0072165 0.0031311 -2.305 0.021192 *				
precipitation					temperature:hour5					0.0327313 0.0031373 10.433 < 2e-16 ***				
has_event					temperature:hour6					0.0172500 0.0031715 5.439 5.44e-08 ***				
hour1					temperature:hour7					0.0014242 0.0031306 0.455 0.649180				
hour2					temperature:hour8					0.0036516 0.0030857 1.183 0.236672				
hour3					temperature:hour9					0.0022101 0.0030587 0.723 0.469967				
hour4					temperature:hour10					-0.0001659 0.0030337 -0.055 0.956395				
hour5					temperature:hour11					-0.0028749 0.0030154 -0.953 0.340402				
hour6					temperature:hour12					-0.0038162 0.0030011 -1.272 0.203532				
hour7					temperature:hour13					-0.0050914 0.0029906 -1.702 0.088694 .				
hour8					temperature:hour14					-0.0053969 0.0029877 -1.806 0.070876 .				
hour9					temperature:hour15					-0.0063573 0.0029804 -2.133 0.032936 *				
hour10					temperature:hour16					-0.0027941 0.0029733 -0.940 0.347372				
hour11					temperature:hour17					0.0135749 0.0029854 4.547 5.48e-06 ***				
hour12					temperature:hour18					0.0280102 0.0030184 9.280 < 2e-16 ***				
hour13					temperature:hour19					0.0364447 0.0030331 12.016 < 2e-16 ***				
hour14					temperature:hour20					0.0337221 0.0030544 11.040 < 2e-16 ***				
hour15					temperature:hour21					0.0182127 0.0030690 5.934 3.01e-09 ***				
hour16					temperature:hour22					0.0114620 0.0030824 3.719 0.000201 ***				
hour17					temperature:hour23					0.0054567 0.0030921 1.765 0.077632 .				
hour18					---									
hour19					Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1									
hour20														
hour21														
hour22														
hour23														

- Our R<sup>2</sup> adjusted went up to 0.853 and our residuals are more random around our Pedestrians variable.
- Note: Although we have many variables, our dataset is extremely large, comprising 16,000 entries. So, we do not think our model overfits our data.

## Sources:

- **Dataset:**  
[https://data.cityofnewyork.us/Transportation/Brooklyn-Bridge-Automated-Pedestrian-Counts-Demons/6fi9-q3ta/data\\_preview](https://data.cityofnewyork.us/Transportation/Brooklyn-Bridge-Automated-Pedestrian-Counts-Demons/6fi9-q3ta/data_preview)
- **Textbook:** A Second Course in Statistics: Regression Analysis, Authors: William Mendenhall & Terry Sincich, Publisher: Pearson, Eighth Edition.
- **Class Notes** (used for code)