IST652: Scripting for Data Analysis

New York State Bridge Conditions

Final Report

Brian Hogan & Katie Poole

Professor Landowski

Syracuse University

June 7, 2019

Table of Contents

Introduction

This following will assess New York State (NYS) bridge data focusing on bridge condition, repair status, and traffic assessment with respect to geographic location. Python will be used to assemble and preprocess data producing data summaries, list, and other structures. Text mining of Twitter feeds to generate traffic word chatter will be explored to see how it may relate to heavy inbound/outbound traffic congestion days and/or short-extended periods, e.g. a weekend or designated repair period. Python will be used for spatial analysis using QGIS. The team feels this effort is reasonable given our class learnings to date.

Describe the data, its sources, and any preprocessing (8 points)

The main source of data for this project is the New York State Bridge Dataset. The dataset is managed by the New York State Department of Transportation. The download is a shapefile, which can be transferred into a csv file using a GIS system. The dataset includes approximately 20,000 bridges. To name a few, fields include latitude, longitude, AADT (average annualized daily traffic), condition rating, structure material, inspection date, year built.

To preprocess the data for the QGIS analysis, the geodatabase file was converted to a csv. Within the dataset 42 fields were analyzed and it was decided to drop 24 fields, leaving 18 fields. All records with NAs were removed. Also, inspection date was changed to an date data type. Year built was changed to an integer data type.

Another source of data used is the New York State Population by County. The dataset simply lists of the 62 counties within New York State and their associated populations as of 2017. This data is managed by the US Census Bureau.

Before the QGIS analysis, the New York State Population by County table had to be converted to a csv and cleaned. This data was used for a join, and the join was dependent upon the county name. Therefore, “County” had to be removed from each record. For example, “Erie County” was changed to “Erie”. And as a final step, the columns were renamed so they were all lower case.

In order bring the New York State Population by County table into QGIS, it was joined with a New York State Counties shapefile (which is included in the source_data_files.zip). This data came from Katie’s work database. However, a New York State Counties shapefile can be accessed here. There was no cleanup necessary for the New York State Counties shapefile.

The AADT data is included in the New York State Bridge Dataset, however it is also its own shapefile. This dataset is of interest because of the AADT value. Since we are looking to see which bridges need the most attention, the number of cars traveling on the bridges is important to know. The data was not preprocessed before using for analysis because it is strictly for visualizations purposes. After the tables were analyzed in QGIS, it was deemed necessary to clean the bridge data (joinresult.csv) before the final analysis. Unnecessary columns were dropped, and NAs were removed. The downside of a shapefile is that it has a 10-limit character

for field names. Therefore, when the shapefile was transferred to a csv the field names were cut short. Therefore, columns were renamed so they are legible.

For the prediction work additional preprocessing including transforming the nominal field categories for bridge "material" from alpha/numeric to numeric where 3= steel and so on. This was also performed for structure type composing 13 categories. Normalization from 0 to 1 wasn't required as was not performing any k-means or kNN predictive analysis. However, population was normalized as ranged from <10,000 to >2.5MM by applying z-score standardization for logistic regression. AADT (annualized traffic) was also scaled from 0 to 276476. Region and country were also made nominal from an alpha/numeric field. Primary owner and maintenance descriptions fields were excluded.

Initially the project wanted to "identify" good, bad, and ugly traffic period time periods and collect tweets for analysis for days by those types based on new reports. The level-1 developer account users are not able to query for specific days, so the project decided to collect the tweets and perform and analysis of tweets to create good, bad, and ugly days based on words counts and sentiment analysis. Brian experienced issues with his mongoDB and installation of Anaconda when tweepy was installed requiring time away from the project to learn more advanced techniques for tweet data mining and data binning so a more basic analysis is detailed.

The team queried a series of accounts across multiple days during the end of May and start of June pulling feeds with the goal of understanding "traffic chatter." The team tokenized the data, worked on cleaning it for bad characters and learned quickly how valuable asynchronous lessons were for this task. Twitter "tweets" are absolutely full of various characters, emojis, and miscellaneous text detracting from word frequency generation and sentiment analysis. Brian did his best to parse data and it was a super exercise to continue development work with. Twitter traffic data was collected from the following accounts:

@ThruwayTraffic	@NYTrafficBureau	DriveSafe_NYS	NYSThruway	LaceyTVNews
Trafficmanmatt	NYTrafficAlert	Fox5NYTraffic	511nyAlbany	Traffic4NY
@TotalTrafficROC	@511NY	@TotalTrafficNYC		

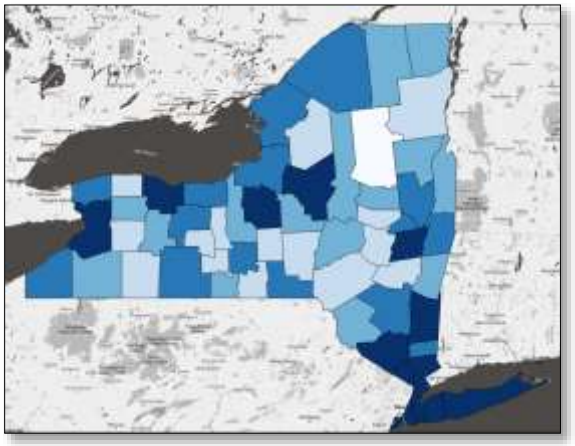
Twitter preprocessing turned into be quite a learning situation as removal of characters of detracting value was difficult. These speaks to using techniques, such as regedit & regular expressions, to help address unique situations. For example: a prominent "tweeter" is "@Trafficmanmatt." This individual is very active on traffic states providing good volume and quality texts. However, one of his approaches is to include his handle "@Trafficmanmatt" in his tweet to build his brand. Such learnings confirm need to scrub and learn data pull nuisances to result in more significant text and sentiment data mining.

The table on the right illustrates a both a good (left hand) and poor (right hand side) word frequency data pull. The good example has routes and words like "blocked & accident." The right hand side, based on ~400 tweets, has a slew of bad characters not removed and twitter "call signs" further branding their chatter.

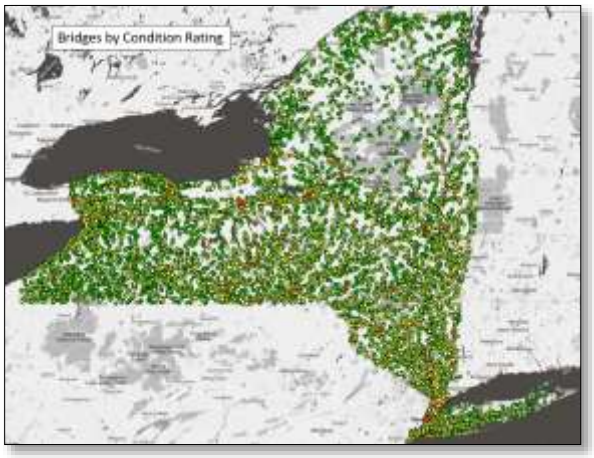
Example of a Good Data Pull (of a Bad day)		Example of What Happens When Expressions Not Addressed	
Word	Count	Word	Count
hudson	9	@	1950
valley	9	:	75
lower	8	https	249
to	7	the	226
nb	7	.	217
i-87	7	trafficmanmatt	214
traffic	7	,	195
sb	7	!	170
blocked	6	rt	159
#	5	suzan916	116
a	5	to	110
accident	5	a	104
slow	5	frecklequeen45	102
service	5	dizzymom64	100

Describe your methods of analysis, including the questions that will be answered, in what fields the data will be used, and what the resulting output will be (10 points)

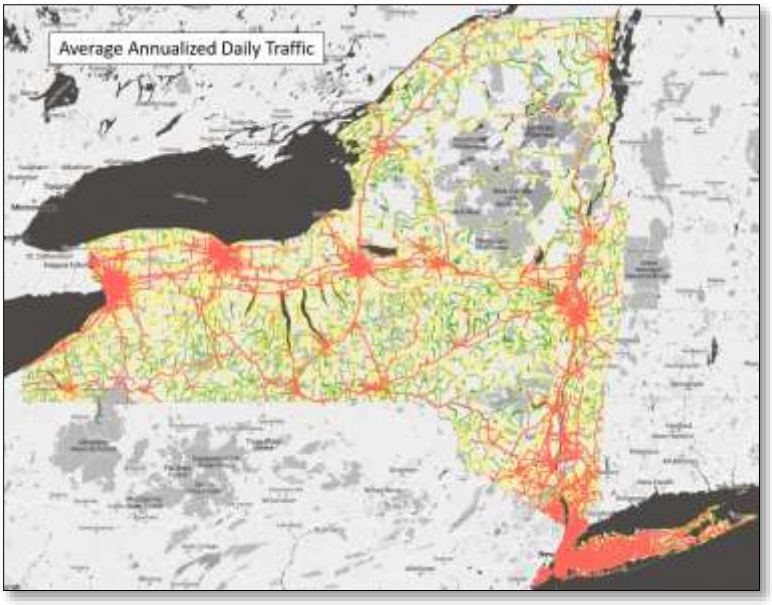
At the beginning of the project, it was deemed important to understand population by county compared to the location of the bridges. We wanted to know, is there a correlation that can be seen visually between the county population and the number of low-quality bridges? To answer this question, first a shapefile of New York State Counties was joined with the New York State Population by County csv. Using PyQGIS (python in QGIS), the tables were joined by county name. As an output, a visualization was created to see the highest populated counties. A graduated symbology style was applied to the counties shapefile, the darkest blue representing the highest populations and white being the lowest populations.



The QGIS-related analysis was running from the perspective of New York State Governments. They would be interested to know which bridges and areas need the most attention. Therefore, the next set of analysis simply focuses on the condition rating per bridge. The dataset was broken into categories by condition rating. This data was used to produce a map visualization. The condition rating is a number from 0 to 7, 0 being the worst score, 7 being the best score. Therefore, bridges within the highest buckets are represented by green points and the lowest are red points. For this analysis the important fields of data are latitude, longitude and condition rating. As a result, a shapefile was created that shows the condition rating scores via color scale.



Another important spatial aspect of this project was looking at traffic data. The number of cars that travel over the bridges helps us to prioritize which bridges need the most attention. As mentioned before, the AADT data is already in the bridge dataset, however to gain more insight, a visualization was created. For this analysis, the shapefile was imported, and a style applied. The style created four categories to show High Traffic, Medium-High Traffic, Medium-Low Traffic and Low Traffic roadways.



The next analysis was very important for potential future python processing and analysis. If we are looking to understand what variables impact a bridges condition rating the most, population should be included. So far in the analysis the bridge dataset has a record for each bridge, condition rating and AADT. Also, there is a county dataset with the population. A dataset needed to be created that included the bridge, condition rating, AADT and population. Therefore, within QGIS, a spatial join was ran. A spatial join merges two datasets together based upon their location. For example, as an output, bridge number 4021480 in Lockport NY now shows a condition rating of 5.4, an AADT of 6616 and the population of Niagara County which is 212,675. The results were exported to a single csv file for further python analysis.

Given the resulting table from QGIS, the python analysis could commence. Questions we wanted to answer were “What are the top 10 worst ranked bridges?”, “Which counties can we identify as our Good, Bad and Ugly counties (using percent high, medium and low priority bridges)?”, “Overall, how many bridges are ranked high medium and low priority?”, “Which county has the highest count of high priority bridges?”

To answer these questions, the condition rating and AADT were considered. The dataset was normalized by ranking them on a scale of 0 to 1 and summing the results together. A loop was created to create buckets to indicate “high”, “medium” and “low” priority bridges. As an output a couple of tables were exported that show the ranking per bridge and the average ranking per county. The following table summarizes questions and query results.

Summary of Bridge Data Analysis Questions	
Question	**Python Result**
Q1: overall, how many bridges are ranked high. medium and low?	hmlpc = jnybd1['priority'].value_counts() hmlpc Out[11]: low 10812 medium 5543 high 264
Q2: which county has the highest count of "high priority" bridges?	cnttbl=pd.crosstab(jnybd1['county_name'],jnybd1['priority']) chp = cnttbl[cnttbl['high']==cnttbl['high'].max()] chp Out[12]: priority high low medium county_name WESTCHESTER 42 208 417
Q3: which county has the highest percentage of "high priority" bridges?	php = prcnttbl[prcnttbl['high']==prcnttbl['high'].max()] php Out[13]: priority high low medium county_name NEW YORK 0.104396 0.263736 0.631868
Q4: which political unit has the highest count of "high priority" bridges?	cnttbl1=pd.crosstab(jnybd1['political_unit'],jnybd1['priority']) chp1 = cnttbl1[cnttbl1['high']==cnttbl1['high'].max()] chp1 Out[15]: priority high low medium political_unit 2034 - City of NEW YORK 82 355 752
Q5: which political unit has the highest percentage of "high priority" bridges?	prcnttbl1=pd.crosstab(jnybd1['political_unit'],jnybd1['priority']).apply(lambda r: r/r.sum(), axis=1) php1 = prcnttbl1[prcnttbl1['high']==prcnttbl1['high'].max()] php1 Out[14]: priority high low medium political_unit 0514 - Town of MARLBOROUGH 1.0 0.0 0.0
Q6: if we were to provide reporting to NYS, which bridges are the top ten highest priority?	jnybd2=jnybd1[['bin','county_name','political_unit','total_rank']] jnybd2 print(jnybd2.sort_values('total_rank',ascending=0).head(10)) bin county_name political_unit total_rank 53 2231439 KINGS 2034 - City of NEW YORK 1.971147 15 1065318 KINGS 2034 - City of NEW YORK 1.969974 70 106531B KINGS 2034 - City of NEW YORK 1.966604 80 5516340 ROCKLAND 0615 - Town of ORANGETOWN 1.963867 162 2229289 NEW YORK 2034 - City of NEW YORK 1.962122 153 2268650 NEW YORK 2034 - City of NEW YORK 1.961550 25 5521218 RICHMOND 2034 - City of NEW YORK 1.959444 230 2065629 BRONX 2034 - City of NEW YORK 1.959263 22 5521217 RICHMOND 2034 - City of NEW YORK 1.952614 23 552121E RICHMOND 2034 - City of NEW YORK 1.952494

The following are a series of Twitter questions generated when the program "Twitter_Data_Pull_Project_ist652_bphogan.py" is run. It is an incomplete program as the "top word frequencies" illustrates a series of characters still require removal and still working on.

Twitter Questions

Tweet columns in the csv output reports include	Tweet columns in the csv output reports include: Index(['id', 'language', 'location', 'screen_name', 'followers', 'tweet'], dtype='object')
What are unique total counts, unique values, top values of tweets? Top is English language with "trafficmanmatt" big presence	tweet count 639 639 639 unique 4 136 222 top en cab867 RT @Trafficmanmatt: … freq 614 121 24
What are the average total tweet followers?	What are the average total tweet followers : id followers count 6.390000e+02 639.000000 mean 1.134483e+18 5058.250391 std 9.218587e+14 17897.514534 min 1.132271e+18 0.000000
What are top word frequencies? Note: table illustrates some improvements in text parsing code but clearly ellipses, twitter name handles, and single letters such as "s" illustrate unique challenges to text mining and future programming work to optimize this effort.	

	Word	Freq	Word	Freq
Word:	with	29	nbamlb	24
Word:	my	29	near	22
Word:	s	29	that	21
Word:	traffic	28	your	21
Word:	just	28	me	20
Word:	jasonnym	28	…	20
Word:	lisalgm1	28	&	20
Word:	thruwaytraffic	25	amp	20
Word:	(	24	he	20
Word:	wnyt	24	tractor	20
Word:	trafficmanmatt	244	trailer	20

Message "sentiment" is valuable but need to learn how group together tweets for an entire data pull and combine as individual tweet sentiment is less useful and meaningful.

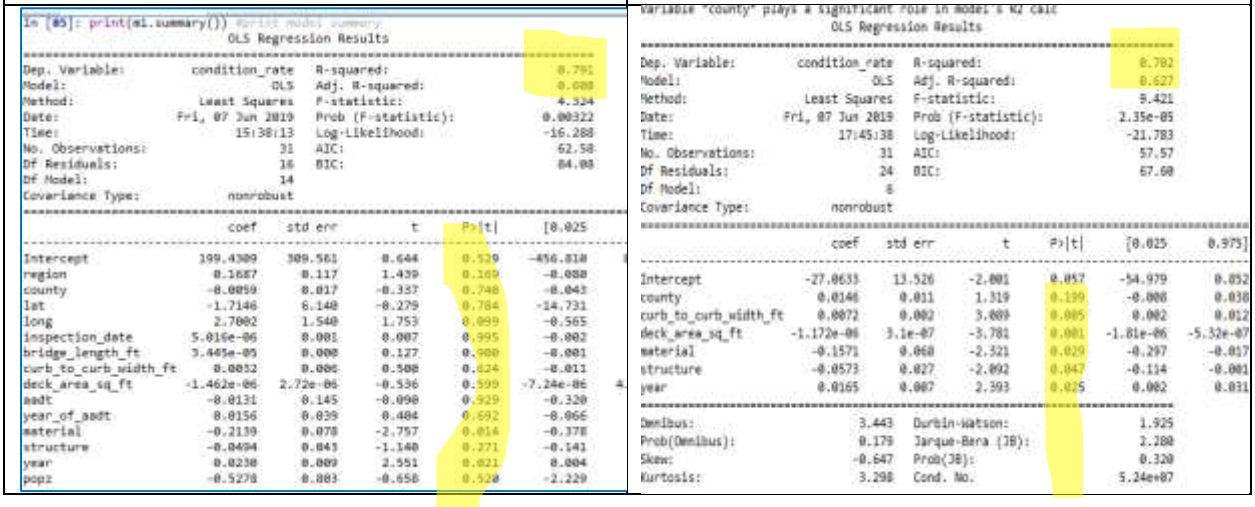
Sentiment Analysis – What is the Overall "Mood" of Individual Tweets Example? (sample)	
When approaching or in a work zone: ✔ Eliminate Distractions ✔ Slow Down & Move Over,… RT @NYSThruway: Drive like YOU work here. 🚧👷👷	compound:0.128, neg:0.078, neu:0.824, pos:0.098,
compound:0.5837, neg:0.0, neu:0.649, pos:0.351, RT @jonjy36: @RonDarlingJr @Trafficmanmatt Welcome back. We all missed you and where happy your on your way to being healthy. Mets broadca…	
compound:0.802, neg:0.074, neu:0.608, pos:0.318, @Trafficmanmatt Oh no!!!!!	

The prediction process was initially interested in understanding how the core bridge factors in the following table contributed to the bridges overall condition rating using methods built in Python. The team will focus on predicting condition rating and understanding what factors are contributing meaningfully. The following are the summarized data elements used for this process and the sample code for fitting a linear model and performing an analysis of variance:

Variable	Sample Value	Variable	Sample Value
ID	0	curb_to_curb_width_ft	88
region	10	deck_area_sq_ft	613314
county	31	aadt	12.2
lat	40.84975399	year_of_aadt	2014
long	-73.94321502	material	4
condition_rate	3.95	structure	14
inspection_date	43014	year	1931
bridge_length_ft	5188	popz	2.2

Manual iteration was performed amongst the variables to assess model influence and P <0.05 evaluations as the initial model had a high degree of variability explained at an R^2 of 0.791. Variables didn't change significantly R^2 until the variable "county" was dropped. The "county" variable has the highest significance in terms of a linear model suggesting "location" of a bridge plays a critical role in both usage, value, work performed, and rating. Effort was spent on the graphing of the linear model but had difficulty achieving the visualization in Python.

All Variables	***Final Variable Reduction***
m1 = ols("condition_rate ~ region+county+lat+long+inspection_date+bridge_length_ft+curb_to_curb_width_ft+deck_area_sq_ft+aadt+year_of_aadt+material+structure+year+popz",df1).fit()	*m2 = ols("condition_rate ~ county+curb_to_curb_width_ft+deck_area_sq_ft+material+structure+year+popz",df1).fit()* ******COUNTY MOST SIGNIFICANT VARIABLE******



```
In [85]: print(m1.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:         condition_rate   R-squared:                       0.791
Model:                            OLS   Adj. R-squared:                  0.600
Method:                 Least Squares   F-statistic:                     4.124
Date:                Fri, 07 Jun 2019   Prob (F-statistic):            0.00322
Time:                        15:38:13   Log-Likelihood:                -16.288
No. Observations:                  31   AIC:                             62.58
Df Residuals:                      16   BIC:                             84.08
Df Model:                          14
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025
-------------------------------------------------------------------------------------
Intercept           199.4309    309.561      0.644      0.529    -456.818
region                0.1687      0.117      1.439      0.169      -0.080
county               -0.0059      0.017     -0.337      0.740      -0.043
lat                  -1.7146      6.148     -0.279      0.784     -14.731
long                  2.7002      1.540      1.753      0.099      -0.565
inspection_date    5.016e-06      0.001      0.007      0.995      -0.002
bridge_length_ft   3.445e-05      0.000      0.127      0.900      -0.001
curb_to_curb_width_ft     0.0032      0.006      0.508      0.624      -0.011
deck_area_sq_ft   -1.462e-06   2.72e-06     -0.536      0.599   -7.24e-06
aadt                 -0.0131      0.145     -0.090      0.929      -0.320
year_of_aadt          0.0156      0.039      0.404      0.692      -0.066
material             -0.2139      0.078     -2.757      0.014      -0.378
structure            -0.0494      0.043     -1.140      0.271      -0.141
year                  0.0230      0.009      2.551      0.021       0.004
popz                 -0.5278      0.803     -0.658      0.520      -2.229
```

```
Variable "county" plays a significant role in model 6 NZ calc
                            OLS Regression Results
==============================================================================
Dep. Variable:         condition_rate   R-squared:                       0.782
Model:                            OLS   Adj. R-squared:                  0.627
Method:                 Least Squares   F-statistic:                     5.421
Date:                Fri, 07 Jun 2019   Prob (F-statistic):           2.35e-05
Time:                        17:45:38   Log-Likelihood:                -21.783
No. Observations:                  31   AIC:                             57.57
Df Residuals:                      24   BIC:                             67.60
Df Model:                           6
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
Intercept          -27.0633     13.526     -2.001      0.057     -54.979       0.852
county               0.0146      0.011      1.319      0.199      -0.008       0.038
curb_to_curb_width_ft     0.0072      0.002      3.609      0.005       0.002       0.012
deck_area_sq_ft  -1.172e-06   3.1e-07     -3.781      0.001   -1.81e-06   -5.32e-07
material            -0.1571      0.068     -2.321      0.029      -0.297      -0.017
structure           -0.0573      0.027     -2.092      0.047      -0.114      -0.001
year                 0.0165      0.007      2.393      0.025       0.002       0.031
==============================================================================
Omnibus:                        3.443   Durbin-Watson:                   1.925
Prob(Omnibus):                  0.179   Jarque-Bera (JB):                2.288
Skew:                          -0.647   Prob(JB):                        0.320
Kurtosis:                       3.298   Cond. No.                     5.24e+07
```

In our final model, although the "county" variable isn't significant we wouldn't have a prediction without it. "County" captures bridge flow traffic (aadt) and population dynamics impacting bridge

condition. Predicting existing bridge future conditions becomes a function of its (year built + structure + material & square footage). Bridge inspection dates and rating are by products of the structure itself.

Include an overall description of the program (5 points)

After cleanup in Python the tables were ready for QGIS Analysis. The counties shapefile and county population csv were imported. From there the tables were joined by county name. The population field needed to be converted to an integer to apply the graduated symbol style. After the integer field was created and populated, the style was created. Categories were created with a color applied to each.

The bridge dataset was imported; however, it was a csv file and needed to be converted to a shapefile, so it could be viewed on the map. The key components of this conversion are the latitude and longitude fields. After the projection was defined, the csv was converted to a shapefile as point features. After the shapefile was created, it was added into QGIS. Next, new numerical fields had to be created for the AADT and condition rating fields. The data was transferred from the old fields (in a string data type format) to the new fields (numeric). After those fields became numeric, the graduated symbol style could be applied. In this case, a style was created that shows the condition rating into four categories.

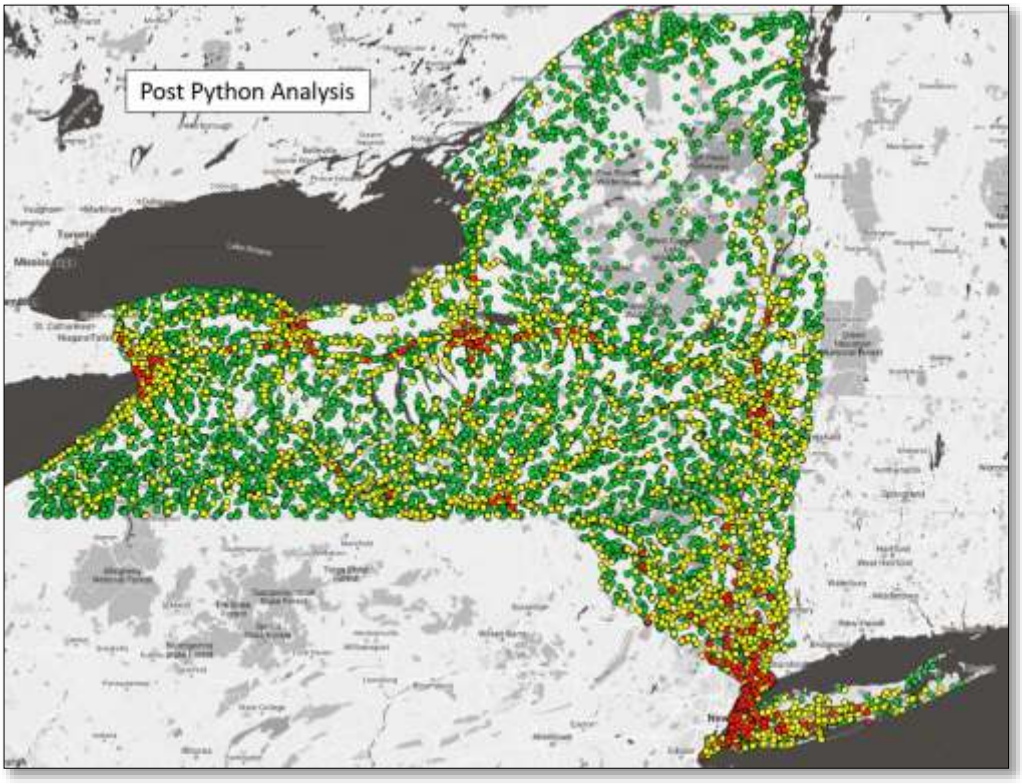
Next, the AADT visualization was created. This included importing the shapefile, and using the AADT column to create a graduated symbol style. This was broken into four categories to show Low to High Traffic areas.

In order to administer the python analysis, the important data fields needed to be within a single table. This is where the Spatial Join was ran. Given each bridge record had its own point geometry on the map and each county had its own polygon geometry, assuming the geometries overlap, a spatial join could be used to migrate the data into one table. The join attributes by location command was executed to run the analysis. The outputs were a shapefile and csv.

The csv file was cleaned (see beginning of report for cleaning description) and analysis began. The data needed to be normalized because the condition rating was based on a scale from 0 to 7. AADT scaled from 0 to 276476. Therefore, both AADT and condition ranting were transformed to a scale of 0 to 1. The newly transformed fields were then summed together, hence our true ranking column was created. Next, to categorize the data, three categories were created to see the high priority, medium priority and low priority bridges. An if/else statement loop was used to create a column that indicated the level of priority. Next, a table was created that shows each county and the percentage of high priority, medium priority and low priority bridges. These results were exported to a spreadsheet.

A final bridge visualization was created. Again, a lot like the first bridge visualization, the data is imported as an csv and converted to a shapefile. The latitude, longitude and projection were provided. This time, instead of a graduated symbol style, the style was dependent upon the priority field which either had high, medium or low indicated. A color and legend name were provided. And finally, we can see the results of which bridges and which counties the highest priority is. In the next section of the program, important data questions are

answered to understand which areas need the most attention and which bridges have the highest priority.



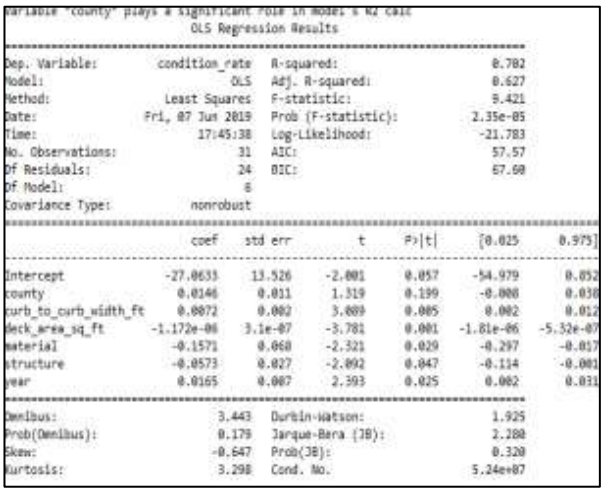
Tweet results were gathered across ten New York Twitter handles (detailed prior) and brought into a mongo database. They were brought in at different times but about 600 total tweets were gathered and regathered at the start of June. Initially the tweets were going to be organized and associated with good, bad, and ugly traffic days by key bridge locations. The analysis and learning time for this twitter binning by geocode and type was comprised as Brian had experienced mechanical issues with both Anaconda and mongodb.

This application uses the paper author's Twitter Developer authentication keys to connect to Twitter and download tweets from a Twitter-User of interest. The program has functions to: save & load to a database; connect to Twitter; & search for tweets. The program has a "main" function that organizes the functions and asks the users for 4 variables including: # of tweets, twitter handles, mongodb name, and mongodb file.

The program generates descriptive statistics for the total tweets, detailed stats on total followers, and performs tweet tokenization resulting in frequency distributions for top words.

One issue in the program is tokenization. As more tweets are brought in the tokenization begins to perform more poorly. The parsing of stop-words and the continued inclusion of unnecessary characters such as "@" illustrate more work will need to be performed to keep learning and improving on these new techniques. The current state of the tokenization will focus work on "regular expressions" to address and clean up characters negatively impacting results.

The linear prediction was built in Python working with the ***scipy*** and the ***statsmodel*** packages. The model sought to understand what variables contribute significantly to prediction bridge condition rate. The "county" variable has the "most" significance on the model suggesting "location" of a bridge plays a critical role in both usage, value, work performed, and rating. The prediction model runs separate of the bridge component. Effort was spent on "graphing" the multiple linear regression equation but proved to be challenging. Upon completion of the prediction program the resulting model is exported to "prediction.txt" generating the following table.



Variable "county" plays a significant role in model's R2 calc

OLS Regression Results

Dep. Variable:	condition_rate	R-squared:	0.782
Model:	OLS	Adj. R-squared:	0.627
Method:	Least Squares	F-statistic:	5.421
Date:	Fri, 07 Jun 2019	Prob (F-statistic):	2.35e-05
Time:	17:45:38	Log-Likelihood:	-21.783
No. Observations:	31	AIC:	57.57
Df Residuals:	24	BIC:	67.60
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-27.0633	13.526	-2.001	0.057	-54.979	0.852
county	0.0146	0.011	1.319	0.199	-0.008	0.038
curb_to_curb_width_ft	0.0072	0.002	3.089	0.005	0.002	0.012
deck_area_sq_ft	-1.172e-06	3.1e-07	-3.781	0.001	-1.81e-06	-5.32e-07
material	-0.1571	0.068	-2.321	0.029	-0.297	-0.017
structure	-0.0573	0.027	-2.092	0.047	-0.114	-0.001
year	0.0165	0.007	2.393	0.025	0.002	0.031

Omnibus:	3.443	Durbin-Watson:	1.925
Prob(Omnibus):	0.179	Jarque-Bera (JB):	2.280
Skew:	-0.647	Prob(JB):	0.320
Kurtosis:	3.298	Cond. No.	5.24e+07

For a group project, describe the tasks and roles of each member of the group (2 points)

Katie took on the task of doing the QGIS Analysis. This included preparing the datasets for analysis, joining tables, categorizing the data and creating visualizations. Once the QGIS analysis was complete, analysis was ran to distinguish which bridges and counties were the highest priority or needed the most attention. From those results QGIS was used for visualizations.

Brian performed gathering Twitter data from ~15 prominent NY daily traffic Twitter handles at different time points. Given restrictions from Twitter the data received was daily chatter and not for a specific time period. An initial objective was to organize words into good, bad, and ugly traffic days and match busy traffic period days, but this task became complex so reporting focused on top word frequencies. Brian also performed the prediction analysis work building the code in Python with learnings from the web versus performing in language **R**.

Both Katie & Brian both organized data, program files, output files, presentation, and the final report for submission.

Draw conclusions from your results about your data (15 points)

In general, there are no distinct spatial patterns for the bridges with the lowest condition ratings in New York State. Outside of New York City, it doesn't appear population and bridges with low condition ratings are correlated. In the overall analysis, which considered both AADT and Condition Rating data we were able to define the top 10 highest priority bridges. Nine out of ten of the bridges were in the City of New York. The other bridge, in Orangetown, NY is within

15 miles of City of New York. There are some visual patterns to the high priority bridge locations. Most are along major highways such as NY-90, NY-390, NY-87 or within cities where there tends to be many overpasses. Very few seem to be over waterways.

Regarding identifying the "Good, Bad and Ugly" counties, a table was produced that shows the percentage of high, medium and low ranked bridges by county (see Analysis Results, Priority per County). The "Ugly" county is New York which has 10% of their bridges as a high priority and only 26% were low priority. As for the "Bad" county, it was chosen because it is the worst county outside of the New York County area. Technically Kings, Bronx, Westchester and Rockland counties rank lower, but they all border New York County. Therefore, Onondaga county is our "Bad" county with 5% of its bridges being high priority and 43% being low priority. And the "Good" county was the best ranked county with high priority bridges being 0%, and low priority bridges being 96%. The county with the highest count of high priority bridges was Westchester County. Finally, there are only 264 high priority bridges in New York State which is only 2% of the bridges. There are 5543 medium priority (33%) and 10812 low priority (65%) bridges.

Generation of a word heat map based on social media Twitter activity has great potential to summarize and communication to people a current state of "chatter." To achieve such an outcome, it requires significant cleaning "bots" to account for all kinds of keyboard communication of symbols, emojis, and any repeated user "handles" that may be repeated for self-promotion. Such self-promotion items are difficult to predict but perhaps can be found by using complex "regular expression" programmatic routines. Clearly such tools are being engineered and sold as products in the market place but can be developed with class IST652 learnings.

It is also fascinating to learn "population" was not an overall predictor of bridge condition even though one can deduce that population & proximity lead to automobiles which leads to bridge traffic. The R^2 of condition rating doesn't seem to reflect nor be concerned with population and traffic variables, such as "aadt" (aka average annualized traffic flow). While a linear model was helpful in skimming the surface of such relationships it suggests more complex data analysis approaches are required to build an understanding of traffic causes bridge wear and tear. The team speculates a k-means nearest neighbor model might be able to use hyperplane distance measures to shed some light on this situation.

Output Files

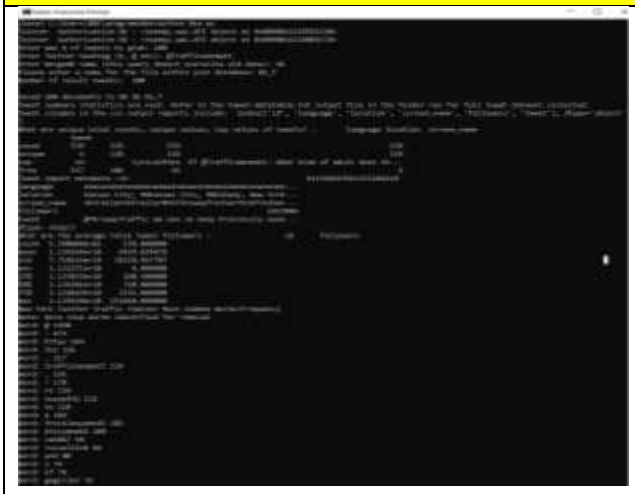
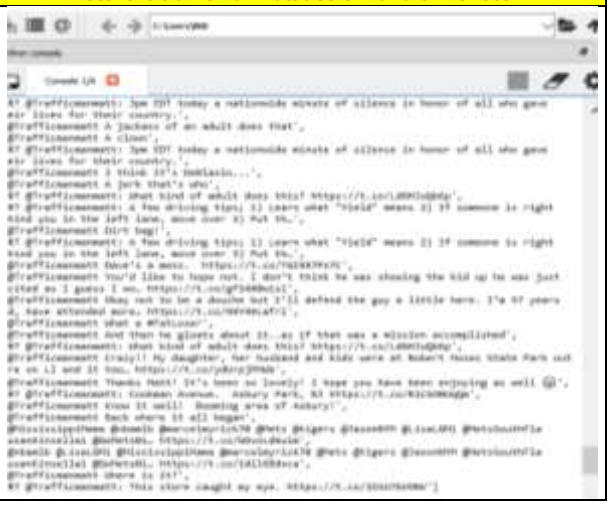
The following are the output files generated from the models located in "Brian+Kat+Ouputs."

Output File Name	Description
nys_bridges.csv	Bridge final data
nysb.shp	Bridge shape file
nysb_1.csv	Bridge combined shape file
rank_per_bridge.shp	Bridge ranking
BBE_HW2_tweet_dataframe_describe_data.txt	Tweet data frame results-descriptive
nysb.csv	Bridge descriptive statistics
joinresult.shp	Join results shape
nysc_pop.csv	Population data
Analysis_Results.xlsx	Final analysis results from QGIS
BBE_HW2_tweet_datatable_0.txt (2 pulls)	Raw tweet data table
Prediction_results.txt	Linear model results

Twitter Program Output:

```
IPython 7.4.0 -- An enhanced Interactive Python.
runfile('C:/Users/BBE/programsbbe/bbe.py', wdir='C:/Users/BBE/programsbbe')
Twitter  Authorization OK : <tweepy.api.API object at 0x0000018B195C1F28>
Twitter  Authorization OK : <tweepy.api.API object at 0x0000018B195C1FD0>

Enter max # of tweets to grab: 100
Enter Twitter hashtag (#, @ etc): @ThruwayTraffic
Enter mongodb name (this query doesnt overwrite old data): mytraffic
Please enter a name for the file within your database: mytraffic_file
Number of result tweets:  28
Saved 28 documents to DB mytraffic mytraffic_file
Tweet summary statistics are next. Refer to the tweet-datatable.txt output file in the folder run for full tweet dataset collected.
Tweet columns in the csv output reports include:  Index(['id', 'language', 'location', 'screen_name', 'followers', 'tweet'], dtype='object')
What are unique total counts, unique values, top values of tweets? :       language  ...                                              tweet
count      250  ...                                               250
unique       1  ...                                                32
top         en  ...  @ThruwayTraffic We now have a huge amount of P...
freq       250  ...                                                24
[4 rows x 4 columns]
Tweet import metadata :id                                  28345956462003390504
language       enenenenenenenenenenenenenenenenenenenenenenen...
location       Kansas City, MOMasturbation StationAlbany, New...
screen_name    USTrailerLittleMrObviousNYSThruwayNYSThruwayNY...
followers                                        1504003
tweet         @ThruwayTraffic We certainly have a bunch of P...
dtype: object
What are the average total tweet followers :                id     followers
count  2.500000e+02    250.000000
mean   1.133838e+18   6016.012000
std    1.240628e+15   5469.827225
min    1.131298e+18      0.000000
max    1.135646e+18  12528.000000
New York Twitter Traffic Chatter Most Common Words/Frequency
Word: : 604             Word: @ 277          Word: thruwaytraffic 250
Word: rt 169                Word: exit 161              Word: on 138              Word: . 110          Word: at 84
Word: accident 84           Word: and 82                Word: blocked 76          Word: to 74          Word: hudson 72
Word: valley 72             Word: a 71                  Word: nb 70               Word: lane 67        Word: i-87 67
Word: lower 61              Word: # 56
```

Python Output of Twitter Run	Spyder Output of "Chatter" ***need to learn to parse and bucket words better! *Note: the trafficmanmatt use of handle in tweet…*
	

Prediction Model Output

m1 = ols("condition_rate ~ region+county+lat+long+inspection_date+bridge_length_ft+curb_to_curb_width_ft+deck_area_sq_ft+aadt+year_of_aadt+material+structure+year+popz",df1).fit()

print(m1.summary()) #print model summary

OLS Regression Results

Dep. Variable:	condition_rate	R-squared:	0.791
Model:	OLS	Adj. R-squared:	0.608
Method:	Least Squares	F-statistic:	4.324
Date:	Fri, 07 Jun 2019	Prob (F-statistic):	0.00322
Time:	18:49:16	Log-Likelihood:	-16.288
No. Observations:	31	AIC:	62.58
Df Residuals:	16	BIC:	84.08
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	199.4309	309.561	0.644	0.529	-456.810	855.672
region	0.1687	0.117	1.439	0.169	-0.080	0.417
county	-0.0059	0.017	-0.337	0.740	-0.043	0.031
lat	-1.7146	6.140	-0.279	0.784	-14.731	11.302
long	2.7002	1.540	1.753	0.099	-0.565	5.965
inspection_date	5.016e-06	0.001	0.007	0.995	-0.002	0.002
bridge_length_ft	3.445e-05	0.000	0.127	0.900	-0.001	0.001
curb_to_curb_width_ft	0.0032	0.006	0.500	0.624	-0.011	0.017
deck_area_sq_ft	-1.462e-06	2.72e-06	-0.536	0.599	-7.24e-06	4.31e-06
aadt	-0.0131	0.145	-0.090	0.929	-0.320	0.294
year_of_aadt	0.0156	0.039	0.404	0.692	-0.066	0.098
material	-0.2139	0.078	-2.757	0.014	-0.378	-0.049
structure	-0.0494	0.043	-1.140	0.271	-0.141	0.042
year	0.0230	0.009	2.551	0.021	0.004	0.042

```
popz                  -0.5278      0.803     -0.658      0.520      -2.229       1.174
==============================================================================
Omnibus:                        1.117   Durbin-Watson:                   2.306
Prob(Omnibus):                  0.572   Jarque-Bera (JB):                0.433
Skew:                          -0.268   Prob(JB):                        0.805
Kurtosis:                       3.217   Cond. No.                     1.17e+09
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.17e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
m2 = ols("condition_rate ~ county+curb_to_curb_width_ft+deck_area_sq_ft+material+structure+year",df1).fit()

print("Variable *county* plays a significant role in model's R2 calc")
```

Variable *county* plays a significant role in model's R2 calc

```
print(m2.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:         condition_rate   R-squared:                       0.702
Model:                            OLS   Adj. R-squared:                  0.627
Method:                 Least Squares   F-statistic:                     9.421
Date:                Fri, 07 Jun 2019   Prob (F-statistic):           2.35e-05
Time:                        18:49:25   Log-Likelihood:                -21.783
No. Observations:                  31   AIC:                             57.57
Df Residuals:                      24   BIC:                             67.60
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -27.0633     13.526     -2.001      0.057     -54.979       0.852
county         0.0146      0.011      1.319      0.199      -0.008       0.038
curb_to_curb_width_ft     0.0072      0.002      3.089      0.005       0.002       0.012
deck_area_sq_ft  -1.172e-06   3.1e-07     -3.781      0.001   -1.81e-06   -5.32e-07
material      -0.1571      0.068     -2.321      0.029      -0.297      -0.017
structure     -0.0573      0.027     -2.092      0.047      -0.114      -0.001
year           0.0165      0.007      2.393      0.025       0.002       0.031
==============================================================================
Omnibus:                        3.443   Durbin-Watson:                   1.925
Prob(Omnibus):                  0.179   Jarque-Bera (JB):                2.280
Skew:                          -0.647   Prob(JB):                        0.320
Kurtosis:                       3.298   Cond. No.                     5.24e+07
```

Data References

New York State Bridge Data (QGIS) (main data set)

- Main source of bridge statistics. Data library 23 fields: ID, lat, long, status, priority rating, etc
- https://www.dot.ny.gov/divisions/engineering/structures/repository/manuals/inventory/rc01_june06.pdf

New York State Average Annualized Daily Traffic (AADT)

- Total volume of vehicle traffic by road by year divided by 365 days
- https://www.dot.ny.gov/divisions/engineering/applications/traffic-data-viewer/tdv-definitions#AADT

New York State Population by County (and/or Department of Transportation Vehicle Ownership Stats)

- Population by county metrics may be useful for building new variables assessing ratios of populations, vehicle ownership counts, traffic throughput statistics, and similar.
- https://www.newyork-demographics.com/counties_by_population

New York State County shapefile

- Boundaries of Counties within the State of New York.
- http://gis.ny.gov/gisdata/inventories/details.cfm?DSID=927

Twitter

- collected chatter across multiple periods across up to 15 different Twitter accounts