**Police Incident Response: Detailed Description**

Task: Report on the statistics of each data set to include: type, unique values, missing values, quantile statistics, descriptive statistics, most frequent values, and histogram. Include analysis statements based on results.

Feel free to skip to Pg 3 where the analysis statements based on results begin. The Histograms are contained within the referenced .html files.

**Original Raw Data Summary: call\_data\_report\_original.html & beats\_data\_report.html**

The raw call data is summarized in call\_data\_original.html. There are 642,998 rows overall. There are two variables for Time (Arrived Time and Original Time Queued) which are categorical, due to their format as unrecognized strings. These will require careful formatting to cast into DateTime objects, in order to do math on the pair of them.

There are 5 string (ie “object” or categorical) variables:

Beat, which is a 2-3 character alphanumeric for the Seattle Police Department “beat” area, which is on the scale of a neighborhood or half of one (if the neighborhood is dense). Sector and Precinct, other categorical variables, are supersets of Beat. There are 18 Sectors and 9 Precincts. Beat is the most specific information about location; Precinct is the least specific.

Call Type, which is the method by which the incident was reported to SPD: ONVIEW means an officer saw it, 911 means a citizen called it in via 911, TELEPHONE, OTHER means a citizen called it in via the (now deprecated) non-emergency line. ALARM CALL (NOT POLICE ALARM) originate from residential and commercial property alarms, such as when a sensor is tripped – the city charges $200 for these if they are false alarms, by the way. There are several others that make up a vanishingly small percentage of the data, only 28 rows overall. There are 9 unique values.

Event Clearance Description is the final determination of how the event was handled, such as ASSISTANCE RENDERED. There are 23 unique values.

FINAL CALL TYPE and INITIAL CALL TYPE are summaries of the topic of the call, when the report is first received and when it is ultimately recorded (once the response has ascertained the true situation).

There are 2 numeric variables.

CAD Event Number, which is a unique identifier; these are roughly linearly increasing with time with numbers in the form YYYY<9 digit number>. They are not strictly linked to time however. All 642998 rows are unique.

Priority, which is a 1-9 ranking of incident importance assigned by whoever received the report. The mean is 3.82

The beats data is summarized in beats\_data\_report.html. It is worth noting that this dataset represents the output of a manual merging process between neighborhood names/locations and SPD beat boundaries. So this dataset is quite clean to start with. It has 55 rows each with an OBJECTID, which is a numeric. The Neighborhood is a string (object) with 39 unique values. The Beat has 55 unique values. Each beat is matched to a neighborhood; each neighborhood can cover one or more beats. The data is set up to join on the Beat column with the call data, such that the call data will end up with a Neighborhood column.

**Pre-Processing / Cleaning Steps: call\_data\_report.html & result\_report.html**

Several pre-processing steps were necessary to clean the call data:

1. Replace ‘nan’ and ‘UNKNOWN’, which are the two built-in missing value signfiers, with np.nan. Luckily there are only a dozen or so of them.

2. Convert the two time variables to Pandas DateTime objects. Original Time Queued had a recognizable format, but Arrived Time had a messier time format that had to be precisely specified for Pandas.

a. DateTime objects are wonderful because you can instantly extract attributes like DayOfWeek, Month, Hour, etc. I did this on the Original Time Queued to add columns related to when the call was made, to analyze how this relates to response time.

b. You can also do time-based math. In this instance, I took the difference of Arrived Time and Original Time Queued to calculate, and converted it to seconds

c. The DateTime objects can also be used to do filtering based on when a report was received. In this instance the analysis was restricted to 2017, which reduced the number of rows to 394031– quite a lot easier to work with.

3. Natural Language Processing via the NLTK toolkit was used to extract stemmed tokens of the Initial Call Type, which is summary of the initial report.

At this point, prior to joining the data with the Neighborhoods\_to\_Beats dataset, and prior to removing outliers, dealing with missing data, the call data we need are all there, but it is still messy.

• For example, there are 2 instances of negative response time, which appear to be the arrival and call times being swapped. I felt comfortable just dropping these rather than make an assumption about why they were wrong

• Response time also has half a dozen large outliers, which are handled with Windsorization on the 95% percentile. I allowed the lowest values to remain in the data because instant responses are neither unusual nor uncommon, especially when a cop is already on the scene seeing something happen.

• Sector has 3126 missing values. These appear to be a mix of miscellaneous police work (like assisting other agencies, testifying at court, writing reports, other administrative work) – which aren’t relevant to the problems we’re analyzing -- and missing information. The “missing” cases appeared to have a Beat even if they didn’t have a Sector; 780 were beat 99 which are the outskirts of the city. I decided that if the Beat was defined, I would keep it in the analysis, otherwise, it probably wasn’t relevant (since I’m not interested in administrative work, etc). This would be handled by the join with the call data.

After the data join, there are 392,056 rows which corresponds to losing the rows with no beat or bad beat information. The “bad beat” information by and large was again not relevant to the questions at hand, so I am comfortable losing that data.

At this point, I work on the call type column. There are 9 unique values to start with, but only 4 have any real amount of data. I go ahead and ditch the 16 rows with other entries, and re-label the categories to be more meaningful. Now we have 4 meaningfully-named categories.

The two numeric variables, priority and response time, are scaled. I do not replace the variables with their scaled versions because I’m interested in retaining that information. Instead I append columns for the scaled versions. We will NOT want to include the non-scaled version in any machine learning model!

Finally, I drop ten columns that are extraneous to the analysis. tokens, nopunc\_tokens, and nostop\_tokens were intermediates of the NLP on Initial call Type; we just want to keep the stemmed tokens. Beat, Sector are not necessary now that we have Neighborhood. I keep Precinct because although it is duplicative, I’m specifically interested in the North Precinct and this is the easiest categorical variable related to area to work with to analyze it (due to only 9 categories). I drop the two “ID” type columns; the row index is sufficient. I don’t need the intermediate call\_type column I used to handle that variable. Finally, I am most interested in questions around the original call (it’s time, content, neighborhood, and response time), not in the final post-hoc determination, so I dropped the columns related to how it was finally entered into the data (Final Call Type and Event Clearance Description).

**Statements Based On Results: result\_report\_final.html**

The final data set has 8 numeric variables: call\_day (day of week), call\_month, call\_hour, priority, response time, index (can be ignored), scaled priority, scaled response time (which are obviously entirely correlated with their non-scaled versions). There are 2 Date variables, Arrived Time and Original Call Time. There are 4 Categorical variables: Neighborhood, Call Type, Precinct, Initial Call Type. Stemmed tokens, derived from Initial Call Type, is composed of lists of strings; this is not supported by the profiler.

1. ¬, unique values, missing values, quantile statistics, descriptive statistics, most frequent values, and histogram. Include analysis statements based on results.

Arrived time ranges from Jan 1, 2017 10min after midnight through the last minute of 2017. 97.%% of the values are unique, and no data are missing. The call frequency remains above 35,000 every month, with a broad peak during the dry summer months (May – Sept) with a peak above 40,000 in July. I would not be at all surprised if this was driven by the general tendency of Seattlites to emerge from hibernation when the rains stop. There were a lot of sirens on July 4 specifically, which may turn up in the data.

Original Time Queued, understandably, looks extremely similar to Arrived Time.

Call Type has 4 distinct values. “visible” where an officer saw something & did something, makes up 42.4% of the calls, which is surprising given how rarely one actually sees cops in Seattle (relative to places like NYC). 34.4% are emergency calls, and 20% are non-emergency calls. I’m curious to see if the word-tokens associated with non-emergency calls are closely related to or significantly different than emergency calls; similarity would suggest callers are bad at self-triaging, therefore everyone should just call 911 (which is the new, current policy). To round out the numbers, 3.2% are due to building alarms, which helps explain why the city charges $200 for a false alarm, simply to discourage them! Though I’ve wondered how fast the cops even respond to those alarms, especially when I’m writing a $200 check to the city. There are no missing values.

Call\_Day is a numeric ranging from 0-6, where 0 = Sunday and 6 = Saturday. [Min, Q1, Median, Q3, Max] = [0,1,3,5,6] with a mean = 2.95. The center metrics reflect a fairly flat distribution (14-15%) although there are somewhat fewer calls on Friday (13.7%) and Saturday (12.8%). I’m eager to split this by call-type. Given that a plurality of responses are due to cops noticing things, I wonder if the Fri/Sat dip is true across all call-types. If there are more calls but fewer “on-view” reports, that would suggest a need for more manpower over the weekend. There are no missing values.

Call\_Hour is a numeric ranging from 0-23, midnight to 11pm. [Min, Q1, Median, Q3, Max] = [0,8,13,18,23] with a mean of 12.9. The most common values are 12, 13, 20, 16, and 15 – that is, middle of the day, late at night, mid-afternoon. Again, I want to split this by call-type, and also by neighborhood. The histogram is varied, with the fewest incidents from 1am-6am, the most from 12pm – 3pm, and another peak around midnight. There are no missing values.

Call\_Month is a numeric ranging from 1-12, Jan to Dec. [Min, Q1, Median, Q3, Max] = [1,4,7,9,12] with a mean = 6.54. This reflects the pattern in arrived time where we saw a broad peak in the summertime.

Initial Call Type is a categorical with 258 distinct entries. The most common, 10%, is “premise check, officer initiated onview only” followed by “traffic stop - officer initiated onview” and “suspicious person, vehicle or incident” at 5.7% and 5.6% respectively. The top 2 are purely officer initiated, which corresponds to “visible” or “ONVIEW” being the plurality call-type. I wonder what this distribution looks like for different neighborhoods or different call types! The stemmed-tokens may offer a better avenue for performing the analysis, because 55.6% of initial call types do not fit into the top 10 – so recoding to categories isn’t really an option.

Neighborhood has 37 distinct entries. It is categorical. The most frequent neighborhood for incidents is the Central Business District, ie downtown, at 11%, followed by Capitol Hill and SODO at 7.4% and 7.1% respectively. Notably, the North Precinct is represented in the dataset – Northgate and Lake City are 4th and 5th most frequent. We will need to be concerned about police presence as a confounding variable – to the extent that ‘visible’ is the most common call type, it’s entirely possible that high police presence would cause a neighborhood to be overrepresented regardless of other factors. Slicing along this axis will be very important.

Precinct is a categorical value with 5 values. The highest frequency precincts are WEST and NORTH (29% and 27.5% respectively) followed by SOUTH (16.2%), EAST (15.2%), and SOUTHWEST (11.9%). There are 1130 missing values, which I believe are mostly Beat 99, which is a catch-all for the outskirts of the city. I’m ambivalent on whether or not to include it in the final analysis, but have retained it for now. The Beat column has no missing data.

Priority is a numeric ranging from 1-9. [Min, Q1, Median, Q3, Max] = [1,2,3,7,9] with a mean of 3.88. There are no missing values. The distribution is strongly bimodal, with a peak at 3 and another peak at 7. The distribution is right-skewed in the sense that most calls are priority 1-3. I’m eager to analyze the initial call tokens vs the priority assigned, and also the priority assigned vs the response time. For call-type NOT visible – that is, where a dispatcher had to assign a priority – you’d expect high priority incidents to get a fast response time. You’d also expect certain words or phrases to correspond to high prioritization. We will test if that is true.

Finally, the response time. First I noticed the 2 rows with negative response time were still in the data, which prompted me to go back and drop them. That was the cost of not dealing with low-end outliers. Obviously it is better to lose 2 rows than compromise 5% of the data that are likely legitimate. Beyond that: a plurality, 40.4%, of response times are 0.0. At first this confused me; was it real? Was it an issue of precision on the TimeDelta type? I switched from minutes to seconds, and spot-checked individual records. I convinced myself they were real, but it still seemed strange.

However, in light of the call-type (recall that 42.4% were “visible”), the 40.4% 0.0s response times make much more sense – these are the incidents where a cop saw something and responded immediately. Luckily for our analysis, there is a distribution stretching out to 8000s (the original 95th percentile) where we can analyze how long it took cops to respond, if the cops were not already on-the-scene. I’d also be interested to look at the 2.2% of cases where a cop WAS on the scene but had non-zero response time; if the response time is still <5 seconds, that’s indistinguishable, but if it is much longer, why? What factors associate with that? Of course I’m interested in the very slow responses; I’d guess those are property crimes. I wonder how many came in on the non-emergency line.

Even though we haven’t done any modeling yet, the process of cleaning and visualizing the data has already raised several interesting avenues of analysis. It has revealed officer-on-the-scene as a significant driver of rapid responses and responses in general. It has also put me on notice to be aware of officer-on-scene as a confounding variable because it drives BOTH responses and response-time.