Kruskal-Wallis Analysis of Emergency Communications Data

Tony R. Dunsworth

Western Governors University

Author Note

Abstract

[The abstract should be one paragraph of between 150 and 250 words. It is not indented. Section titles, such as the word Abstract above, are not considered headings so they don’t use bold heading format. Instead, use the Section Title style. This style automatically starts your section on a new page, so you don’t have to add page breaks. Note that all of the styles for this template are available on the Home tab of the ribbon, in the Styles gallery.]

Keywords: [Click here to add keywords.]

Kruskal-Wallis Analysis of Emergency Communications Data

As SARS-COV2 (COVID-19) has impacted every aspect of society in the United States and around the world, emergency communications centers across the country have faced significant challenges and have, from the closure of the 911 center in Puerto Rico (“Puerto Rico shutters 911 call centers amid coronavirus cases,” 2020) to the City of Alexandria, Virginia deploying 911 call takers to work from home while isolating other personnel (Stone 2020), addressed the impact in different ways to ensure the continuity of operations to serve their respective communities. To understand the impact of the decisions made by the City of Alexandria, this paper will employ non-parametric analytical techniques to compare data from 2019 and 2020 and within 2020 to view the changes in the operational times for key metrics in the 911 call process: the time from call pick-up to available to dispatch, the time from available to dispatch to the assignment of the first unit, and the time from call pick-up to release of call. Through the analysis, the impact of operational changes can be viewed and recommendations given to address future major events to the benefit of the community served by the public safety answering point (PSAP).

# Research Question

[The first two heading levels get their own paragraph, as shown here. Headings 3, 4, and 5 are run-in headings used at the beginning of the paragraph.]

## Data Collection1

The data needed for this analysis exists in a SQL Server database owned by the City of Alexandria, Virginia and maintained in cooperation between the Department of Emergency and Customer Communication (DECC) which is responsible for the 911 and 311 call centers for the city, and the Information Technology Services (ITS) department which is responsible for the city’s technical services. The data is generated through the CentralSquare Enterprise Computer Aided Dispatch software package and is stored, due to age, on an archive server in a database named Reporting\_System and in one table; Response\_Master\_Incident. (“Computer-Aided Dispatch | CAD Dispatch Software | CentralSquare,” 2020) This table consists of 119 columns, of which only nine were determined to be necessary for this analysis. Those columns are Response\_Date: the datetime stamp the software determines to be the start time for the incident. Priority\_Number: the numeric value assigned to the call based on definitions given by the agencies served by the Public Safety Answering Point (PSAP) and indicating the relative importance of the call on a scale from one to ten; one being the most important and ten being the least important. Problem: the descriptor of the reason for the service call. (e.g., Traffic Stop, Cardiac Arrest) Agency: the responding agency as defined by DECC in concert with the agencies they serve. MethodOfCallRcvd: the way the call for service was received by the PSAP. Fixed\_Time\_PhonePickUp: the datetime stamp recording the moment the call was officially started per the CAD software. Fixed\_Time\_CallEnteredQueue: the datetime stamp recording the moment the call taker makes the call available for the radio dispatcher to assign responding units. Time\_First\_Unit\_Assigned: the datetime stamp recording the moment the radio dispatcher assigned the first unit to the service call. Fixed\_Time\_CallTakingComplete: the datetime stamp recording the moment the call taker stops contact with the reporter and stops actively working the call for service.

After the appropriate columns were identified, an additional ten columns were created for the dataset. These columns were created in order to present additional analytical opportunities and identify additional significant differences in parts of the PSAP operations. The first six of these columns were created from the Response\_Date column. They are: Year: this column indicates the year portion of the datetime value for the call start. In this study the possible values for this column are 2019 and 2020. Month: this column indicates the month portion of the datetime value for the call start. WeekNo: this column indicates the week number as calculated by SQL Server 2016 from the datetime value of the call start. DOW: this column indicates the day of the week as calculated by SQL Server 2016 from the datetime value of the call start. Shift: this column shows the group which received the call as based upon the hour of the start of the call. In this study, the possible values for this column are “Day” and “Night”. “Day” comprises all calls between 6 a.m. and 6 p.m. with “Night” comprising the opposite. The final four columns in this dataset are calculated from the other datetime columns and reflect elapsed times for different stages in DECC’s handling process. The first is QueueTime which is the time elapsed between the Fixed\_Time\_PhonePickUp and Fixed\_Time\_CallEnteredQueue. This is the time the call taker uses to start processing the call and collect enough information to send the service call to a radio dispatcher for assignment. The second is DispatchTime which is the time elapsed between the Fixed\_Time\_CallEnteredQueue and Time\_First\_Unit\_Assigned. The third is Process time which is the elapsed time between the Fixed\_Time\_PhonePickUp and the Time\_First\_Unit\_Assigned. This is the time the radio dispatcher uses to find the appropriate unit(s) and commit the assignment. The third column is CallTime which is the time elapsed between the Fixed\_Time\_PhonePickUp and Fixed\_Time\_CallTakingComplete. This is the time the call taker uses to process the call from pickup to release. The final additional computed column is ProcessTime which is the time elapsed, in seconds, between the time the phone was picked up and the time the first unit was assigned to the call. While this should be an aggregate of QueueTime and DispatchTime, this can vary in the measures of central tendency slightly from a simple addition of the QueueTime and DispatchTime columns.

The base and most of the derived columns were collected via a T-SQL query from a SQL Server 2016 archive database. The final three columns were created through R Studio after the preliminary dataset was imported for analysis. To create these variables, the timestamp variables are aligned for each variable and then subtracted where the output is given as a numeric value. The advantage to this option is the simplicity of derivation through one mathematical operation between the columns. This ensures that there are fewer opportunities for miscalculations. The disadvantage of this method is that it is done after importation into an R tool. If there are problems identified in the dataset, those problems could be magnified through the alteration of the existing columns. The other option considered would have been to use the CAST and CONVERT functions in T-SQL to turn the datetime columns into big integers and then subtract one value from another. The biggest advantage to using dynamic SQL to grab the columns is the ease of collection at the data source. The biggest disadvantage to this method of collection is the impact on the database from which the data is pulled. If the columns needed for the computation are not indexed, performance can be degraded and the calculations can and results can become unreliable.

Due to the nature of this data, prior to collection, an agreement was reached with DECC for access to the data, provided no identifiable nor restricted data was collected. The data fields used for this analysis were approved by an assistant director and the director prior to collection.

### Data Extraction and Preparation

The data was extracted from a SQL Server database through a query which will be included in the exhibits. After the query was prepared and executed, the results were exported to a comma-separated values (csv) file in Microsoft Azure Data Studio. The data was then inspected for NULL values and other anomalies which could be addressed and ameliorated prior to importation into RStudio. After the first query, adjustments were made to the query to account for the additional data fields which needed to be included as columns in the final csv file. The details of the created columns included in the final query have been discussed earlier in the analysis. To provide additional details and address missing values in the csv file, additional changes were made to the SQL query. To give a better breakdown on calls assigned to the Fire Department between fire calls and medical calls, the SQL query uses a case statement to make a separation based upon the problem nature assigned to the service call. In the first view of the csv file there were numerous NULL values in the field MethodOfCallRcvd; a string indicating the origin of the service call. There were definite patterns in the NULL values keyed to certain problem types. The SQL query was then updated to address those discovered patterns, for example, any calls arriving from Mutual Aid partners are then updated in the query to have a value of ‘MUTUAL AID’. For service calls where there is no discernable pattern to be found, the NULL value was changed to “Not Reported’ to eliminate NULL values from that column. The final column which needs updating to address additional NULL values is the start time for the service call. To address this through the SQL query, there are three clauses to address this. If the problem type assigned to the service call is any one of the Mutual Aid problem types, then we use the ClockStartTime as the start of the service call. If there is no entry in the Fixed\_Time\_PhonePickup column, then we use the Time\_PhonePickup field. Otherwise, we will use the Fixed\_Time\_PhonePickup column.

After identifying the columns needed and the computed columns required, restrictive WHERE clauses are added to ensure the data collected fits the parameters needed for the final analytical data set. The first restrictive clause limited the data returned to the last two full years; 2019 and 2020. The next clause restricts the returned data to those where the call taking personnel are part of DECC. The table retains the name of the call takers and is joined against the Personnel table where DECC personnel are identified with a four-digit serial number in the four thousand range. This restriction eliminates most of the Mutual Aid calls from the final data set. These calls are restricted from the data set since the operational procedures for these calls bypass the call taker under most circumstances. The next restrictive clause is designed to ensure the call was dispatched and assigned for service. The final restrictive clause is designed to prevent the inclusion of service calls which were never run. After the query was run and exported to a csv file. The csv file was then inspected to look for missing rows with missing data points. With the construction of the query, there were only 25 rows with a missing data column. Because of the volume of rows collected in the extraction, deleting these rows is an appropriate method of data cleansing.

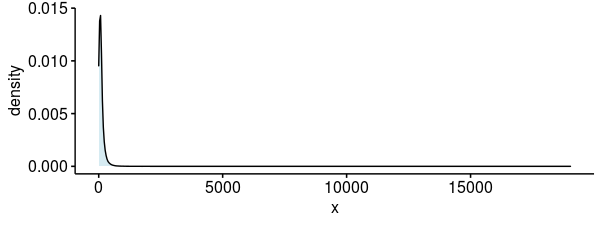
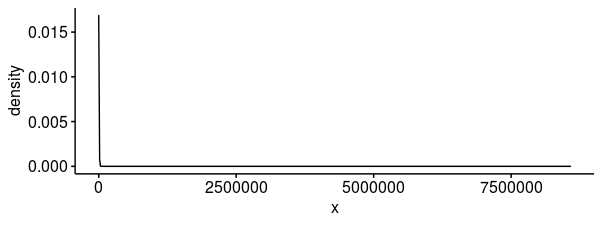
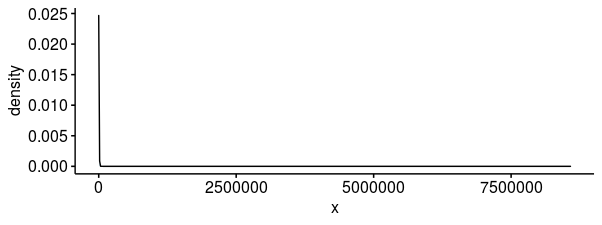
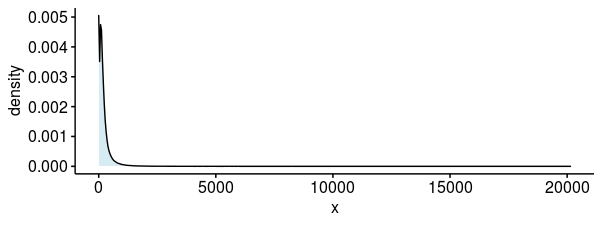
After the data was extracted from the database, it was imported into RStudio for creation of the computed columns and the start of data cleaning. Upon importation into RStudio, the dataset was 15 columns and 150,281 rows. To ensure all data points were correctly imported, instead of using the read.csv function from base R, the readr package was loaded and its read\_csv function was used instead. The advantage to using that specific function is that the datetime columns are properly imported as POSIXct fields. This allows for easier creation of the computed columns discussed above. The character columns are converted into factors using the as.factor function for better univariate analysis. The computed columns are created via subtracting the values of the correct originating columns and the difference converted to a numeric value through the as.numeric function. Now the data frame is a tibble of 19 columns and 150,281 rows. Next the summary function was run against the computed columns. The minimum values for all of these columns were negative integers. Since these columns are meant to be elapsed times, negative values represent problematic outliers which must be addressed. As there are multiple options using the software as intended which could contribute to a negative value for the difference between two time points, the next step was to count the number of rows where any of the computed columns were a negative value. 571 rows were found to have a computed column with a value less than zero. This represents 0.38% of the rows in the data frame. Since the number of rows is so small, the simplest and safest way to address the negative outliers was to remove the rows from the data frame. Since large positive values for the computed columns could be legitimate values, no rows with large positive values were removed from the data frame. This decision may lead to an increase in the mean of these columns and possibly contribute to an increased right tailed skew in the overall distribution of the remaining data. Comparative normality tests have been run against the data frame as adjusted by the removal of the negative value rows from the data frame and the data frame prior to the negative values removal. Comparative means and medians were also run against both data frames to gauge the impact that a one-sided removal could have on the analyses.

Prior to analysis, after the data has been imported and cleaned, the categorical variables have been converted to factors in order to make univariate and bivariate analyses much easier to perform. Some ordinal numeric variables also had a factor converted version created in R as they are treated, in this analysis, more like categorical variables than numeric variables.

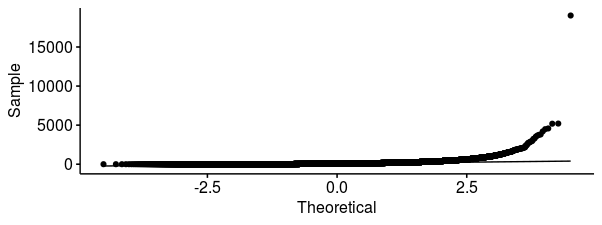
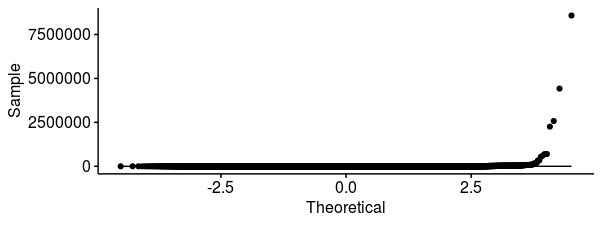
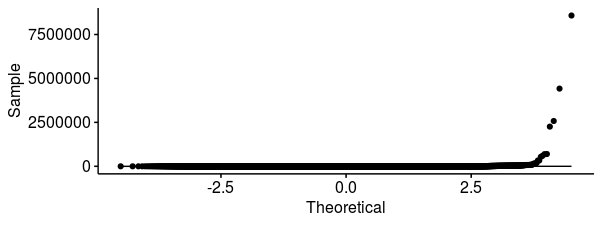
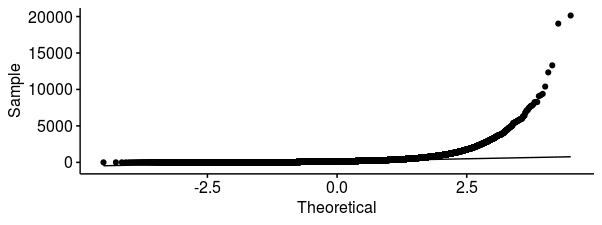
The analyses were performed in RStudio Cloud using the R programming language. R was chosen for the analysis because it can handle the analytical work with fewer add in packages and per Professor Norm Matloff, “R is written by statisticians, for statisticians,” giving it an advantage over Python for detailed statistical analysis (Matloff, 2019).

#### Analysis.

After the importation of data and the creation of the computed columns in RStudio, the first step was to address and verify the normality of the continuous computed columns. Three tests were performed on the four computed columns to visually and statistically ascertain normality. The first test performed was the creation of a density plot for each of the four variables. The screenshots of these density plots are as below:

****

As can be seen from these density plots from the ggplot2 package, none of the four continuous variables display a normal distribution. With the length of the tails, all four variables show a significant right skew pattern along with a lack of a normal distribution curve. After the density curves, QQ plots were performed in order to verify the results of the density plots. Those QQ plots are as follows:



Again, the QQ plots show distributions which are not normal. Together with the density plots, it becomes apparent this data will not conform, with statistical adjustments, to a normal distribution. Finally, as further confirmation an Anderson-Darling normality test was run on each of the computed continuous variables. Because there are ties in the data ranks, the standard Kolmogorov-Smirnov test cannot be run against these variables without throwing errors and warnings. The results of all four variables’ exhibited p values from the Anderson-Darling tests were reported by RStudio as p < 2.2e-16, showing the null hypothesis that the variables are normally distributed can be safely rejected. Since all three tests confirm the variables are not normally distributed, further analysis will use nonparametric methods with comparisons of the population medians rather than using the means which are more impacted by the heavy right skew exhibited by the data.

##### [Heading 5].

[Like all sections of your paper, references start on their own page. The references page that follows is created using the Citations & Bibliography feature, available on the References tab. This feature includes a style option that formats your references for APA 6th Edition. You can also use this feature to add in-text citations that are linked to your source, such as those shown at the end of this paragraph and the preceding paragraph. To customize a citation, right-click it and then click Edit Citation.] (Last Name, Year)

# References

Puerto Rico shutters 911 call centers amid coronavirus cases. (2020, October 22). Retrieved April 15, 2021, from AP NEWS website: https://apnews.com/article/virus-outbreak-puerto-rico-latin-america-c939e01656d5c0d2ce490ae9dbe8db46

Stone, L. (2020, May 5). City of Alexandria, VA Utilizes FirstNet for 911 Remote Call Taking During Pandemic | First Responder Network Authority. Retrieved April 15, 2021, from www.firstnet.gov website: <https://www.firstnet.gov/newsroom/blog/city-alexandria-va-utilizes-firstnet-911-remote-call-taking-during-pandemic>

Computer-Aided Dispatch | CAD Dispatch Software | CentralSquare. (2020). Retrieved March 10, 2021, from www.centralsquare.com website: https://www.centralsquare.com/public-safety/cad

Matloff, N. (2019, October 14). *R-vs.-Python-for-Data-Science*. GitHub

Footnotes

1[Add footnotes, if any, on their own page following references. For APA formatting requirements, it’s easy to just type your own footnote references and notes. To format a footnote reference, select the number and then, on the Home tab, in the Styles gallery, click Footnote Reference. The body of a footnote, such as this example, uses the Normal text style. (Note: If you delete this sample footnote, don’t forget to delete its in-text reference as well. That’s at the end of the sample Heading 2 paragraph on the first page of body content in this template.)]

Tables

Table 1

[Table Title]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Head | Column Head | Column Head | Column Head | Column Head |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |

Note: [Place all tables for your paper in a tables section, following references (and, if applicable, footnotes). Start a new page for each table, include a table number and table title for each, as shown on this page. All explanatory text appears in a table note that follows the table, such as this one. Use the Table/Figure style, available on the Home tab, in the Styles gallery, to get the spacing between table and note. Tables in APA format can use single or 1.5 line spacing. Include a heading for every row and column, even if the content seems obvious. A default table style has been setup for this template that fits APA guidelines. To insert a table, on the Insert tab, click Table.]

Figures title:

Figure 1. [Include all figures in their own section, following references (and footnotes and tables, if applicable). Include a numbered caption for each figure. Use the Table/Figure style for easy spacing between figure and caption.]

For more information about all elements of APA formatting, please consult the APA Style Manual, 6th Edition.