Addressing the Issue of Potholes in San Diego

Andrew Kim, Grigor Tashchyan, Jonathan Yoon

Shiley-Marcos School of Engineering, University of San Diego

Time Series Analysis

December 5, 2022

Abstract

Potholes are an everyday issue that affect Americans on a daily basis. These potholes can cause tire blowouts and in the extreme case, injuries to commuters. Our project focused on evaluating the pothole repair rates over time segmented by district type and determining the best method to forecast pothole repair rates. The city of San Diego generated a website where citizens can report potholes in the city and track the progress of its repairs. In our analysis, we analyzed 26,620 observations across 23 features in the dataset retrieved from the City of San Diego Datasets. After exploring the key geographical characteristics of pothole cases, a time series was created by grouping the number of cases by months spanning from December 2018 to October 2022. Once the pre-processed data was partitioned into training and validation sets, six forecasting methods were evaluated based on the metric RMSE (root mean square error). Through this process, the neural network method was found to be the optimal method for San Diego's department of City Management to evaluate their progress towards repairing potholes within the city's districts and determining the amount of funds to allocate to preserve the road quality's effectiveness and keep people safe.

Keywords: ARIMA, Linear Forecast, Mean Forecast, Naïve Forecast, Neural Network, RMSE, Smoothing, Time Series Analysis.

Introduction

Potholes are depressions or holes in the pavement surface of roads that vary in shape and size. They are caused when the top layer of pavement and the material beneath the underlying soil structure, mainly water, weakens and cannot support the weight of traffic, which can result with the road surface cracking. This issue presents a constant threat to people commuting on the road on a daily basis. Among the primary factors the pothole issue presents to commuters include straining the vehicle's interior aspects that include suspension, shocks, tires, and steering alignment, all critical aspects that can leave drivers and passengers with serious injuries in the event of an incident. The threat of potholes is also an issue that takes a strain towards drivers' wallets. According to the American Association of Retired Persons, or AARP, potholes cause $3 billion dollars in damage to vehicles each year in the United States. The American Automobile Association, or AAA, estimates that the average repair bill for a pothole-related incident is $306, but can reach over $1,000 in severe cases (McCandless, 2022). Each city and county in the United States is trying its own best efforts to resolve, minimize, and prevent problems associated with potholes. According to the City of San Diego, the City repairs more than 30,000 potholes per year, using materials such as hot patch compounds and bagged asphalt (2022). It is not surprising that accurate predictions and resource estimation based on these predictions would minimize any unexpected problems and would save considerable amounts of resources and the environment. The City of San Diego uses a reporting system in its website sandiego.gov to collect information on pothole issues on roads. This system would allow one to analyze the overall trend of the pothole problems, and accurate forecasting could be made based on the information from this analysis.

Problem Statement

The purpose of our analysis is to evaluate the repair rates over time segmented by district type. This analytical approach is key to sharing with the city of San Diego to evaluate how the city can approach and repair the potholes along with providing them with insights on where they are doing well versus where they might need improvement. This study is also necessary to understand the problem and to determine whether or not more funds were necessary to be allocated to the department for maintaining roads effectively.

Literature Review

Potholes have and remain a major threat towards the city of San Diego that have also affected many lives and resulted with the city piled with multitude of injury lawsuits. There are an estimated 55 million potholes across America today spanning more than 4 million miles of roads. Since 2010, the city of San Diego has spent $220 million on lawsuit payouts related to pothole incidents that tally to more than 20,000 (Garrick, 2022). The City of San Diego also reported that in 2017, the estimated cost for repairing a single pothole was $495.65 (August, 2022). Such incidents call for the city to take proactive measures to fix roads that are damaged and concrete that are rough or have loose gravel. By taking a deep dive into the data, we can determine just how effective the city of San Diego is at repairing potholes. Fixing these potholes is not just a simple service the city must provide, but also a liability. Since the city is responsible for building and maintaining these roads, they are also responsible for ensuring the roads are safe for drivers with proper signage, well-lit roads, and repairs of hazardous holes and dips in the road. Looking at the classification zones will also give us some good insight. For example, we

can look at different population zones and determine whether the population has an effect on the repair rates.

This study evaluates the repair rates over time segmented by district type that is essential to share with the city of San Diego while evaluating the insights on what the city is doing well versus what the city needs to improve. This study is tested by utilizing a dataset of 26,620 observations across 23 features retrieved from the City of San Diego Datasets.

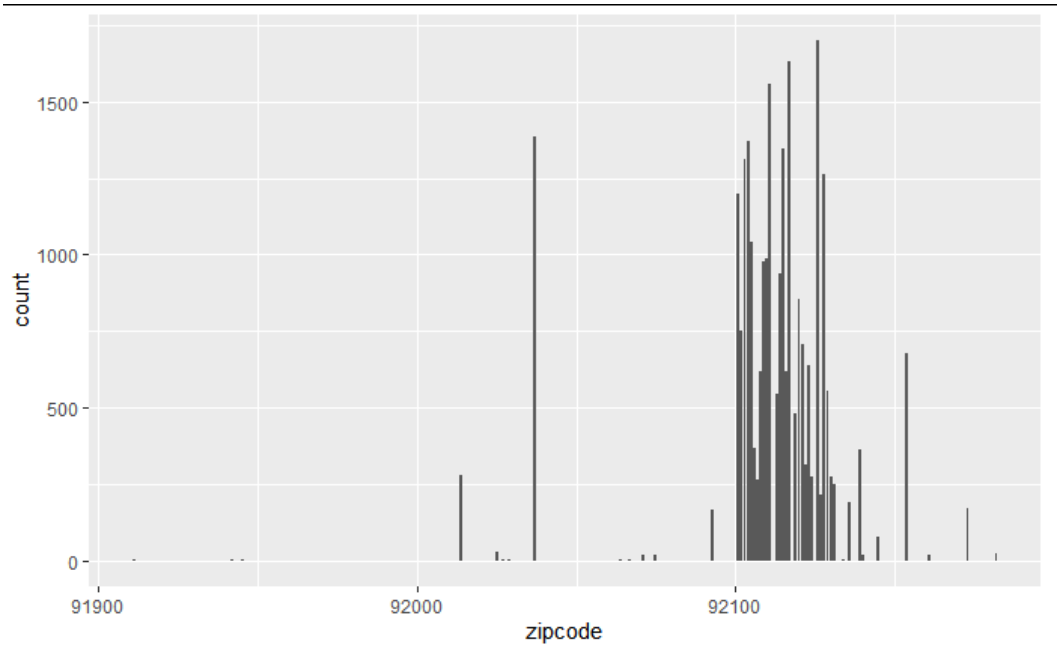
Methodology

The total number of observations and variables within the dataset describe each of the pothole repair requests in the City of San Diego from January 2017 to November 2022. For the purpose of exploratory data analysis, the geographical characteristics of the cases in the city were analyzed. One of our preliminary steps was to check if the data was complete. It appears that the dataset contains some missing entries and duplicate entries that needed to be dropped to provide insightful conclusions. We explored the dataset to understand which predictors were the most important ones, and use those for our analysis. We ultimately decided that the best predictors to use were based on geolocation. The variables that describe the areas in the city are zip codes, council districts, and community plan codes. We felt these three predictors were essentially the ones that would help us understand the problem and create possible solutions.

The first analytical approach made towards the dataset was analyzing the distribution of pothole cases based on zip codes. Generating a histogram that provided a visual analysis of the distribution process between the cases in relation to zip codes, it was found that the 10 most common areas that require repair are zip codes 92126, 92117, 92111, 92037, 92104, 92115, 92103, 92128, 92101, and 92105. The histogram of this distribution is in Fig 1.

Figure 1

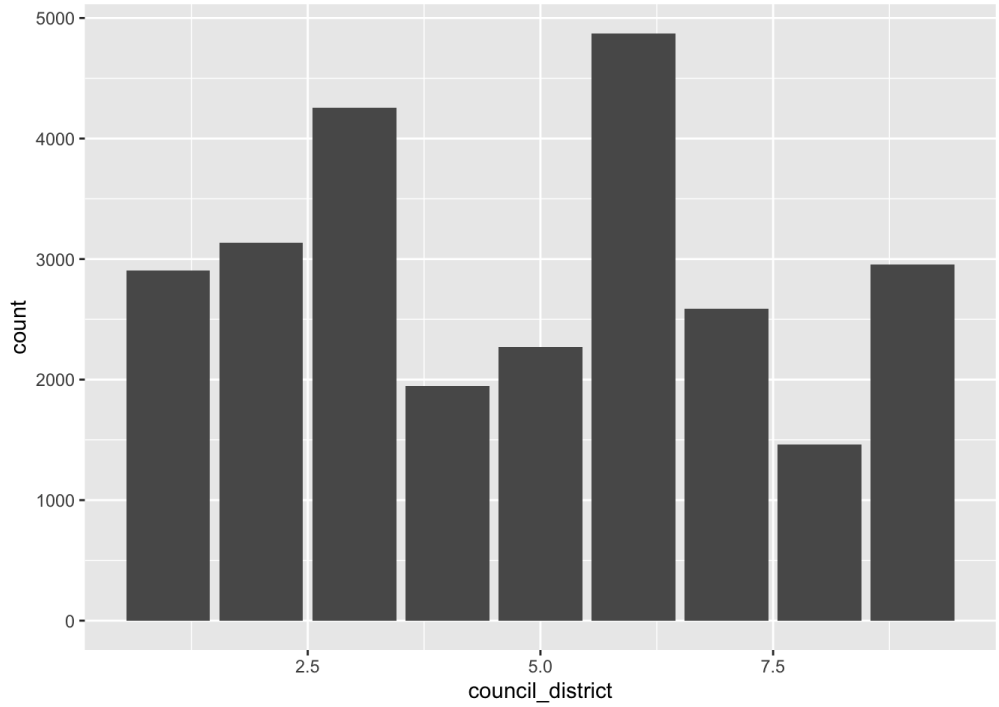
Distribution of Cases based on Zip Codes



The second analytical approach was looking at the distribution of pothole cases based on council districts. From the visualization, the most common council districts that require repairs are 6, 3, 2, 9, 1, 7, 5, 4, and 8, in decreasing order. The histogram of case distribution based on council districts is in Fig. 2.

Figure 2.

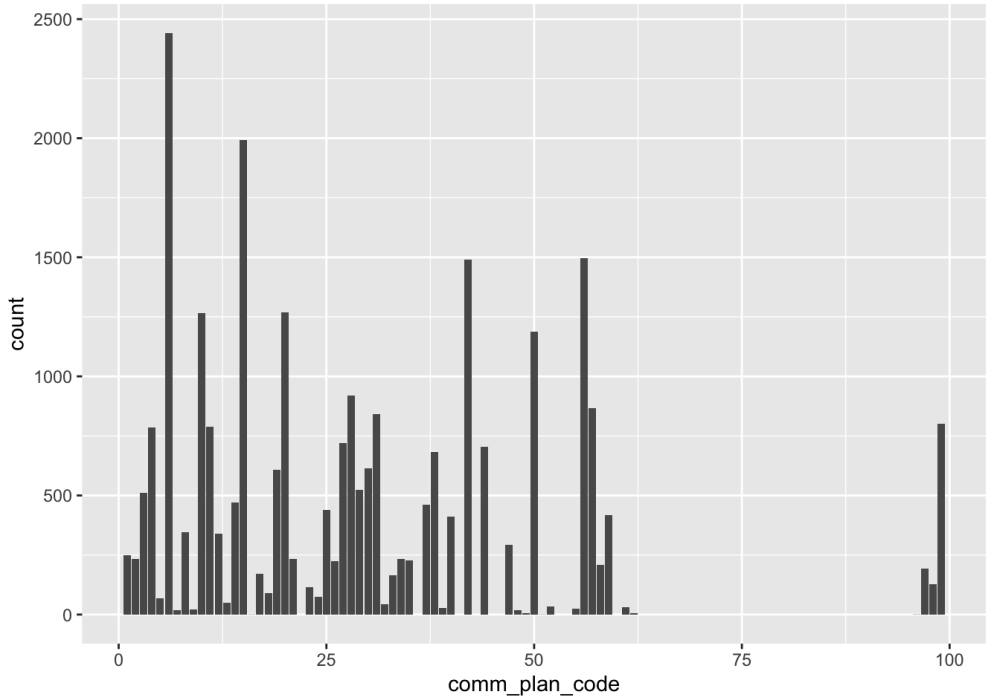
Case Distribution based on Council Districts



Lastly, we looked at the distribution of pothole cases based on the community plan codes. We found that the 10 most common community plan codes that require repairs are 6, 15, 56, 42, 20, 10, 50, 28, 57, and 31. This histogram that describes the case distribution based on community plan codes is in Fig. 3.

Figure 3.

Case Distribution Based on Community Plan Codes

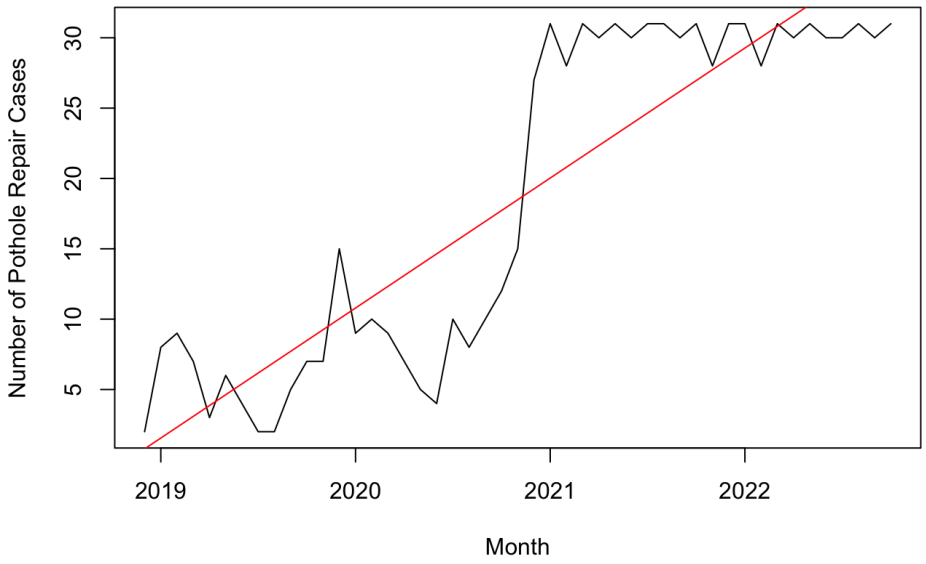


After the analysis, for the purpose of forecasting the number of cases to allocate resources for repair more efficiently, it has been concluded that these describing factors do not contribute to the main purpose of this project. It has been realized, however, that purely grouping the number of cases by month and creating a time series based on the number of cases per month would be the most appropriate approach to forecast the number of cases by month. To meet this criteria, the dataset is preprocessed by grouping cases by requested date, counting the number of cases. The requested dates are then grouped by the month with the adding up all the counts, or the number of cases. After grouping and adding the number of cases by month, it is realized that the series should begin from December 2018 because omissions are noted before this month.

The time series was generated beginning with December 2018, and the plot of this time series is generated in Fig 4, with a fitted line.

Figure 4.

Time Series of Number of Pothole Repair Cases per Month



Results

To evaluate the performance of each of the forecasting models, the time series generated and pre-processed was partitioned into training and validation sets. The training set is composed of the number of repair cases from December 2018 to May 2022, which is a total of 42 months, and the validation set contains the number of repair cases from June 2022 to October 2022, which spans a total of five months. The months before December 2018 are not included in the training set due to lack of cases perhaps from improper implementation of the reporting system at the time. In addition, November 2022 was not included in the validation set due to its lack of completeness in its reporting. A total of six forecasting models were generated using the training set, which

included Naïve, mean, linear, ARIMA with 1 lag, exponential smoothing, and neural network. Each of these models were then applied to forecast the five months following May 2022, and the forecasted data was then compared to the validation set. Several metrics were calculated to measure the difference between the forecasted values and the validation set to evaluate the model: mean error (ME), root mean squared error (RMSE), mean absolute error (MAE), mean percentage error (MPE), and mean absolute percentage error (MAPE). Among these, RMSE was chosen as the metric to compare the models' performances. Table 1 summarizes the results of the performance measurements.

Table 1.

Comparison of the Six Forecasting Models

Model	ME	RMSE	MAE	MPE	MAPE
Naive Forecast	-0.6	0.7745967	0.6	-2	2
Mean Forecast	13.32857	13.33757	13.32857	43.82949	43.82949
Linear Forecast	-6.69432	6.769359	6.69432	-22.01563	22.011563
ARIMA with 1-lag	0.8485574	1.326189	1.048313	2.746315	3.412168
Smoothing	-0.5549385	0.740241	0.5909877	-1.851733	1.968021
Neural Network	0.2267282	0.5236672	0.4155503	0.7201501	1.351507

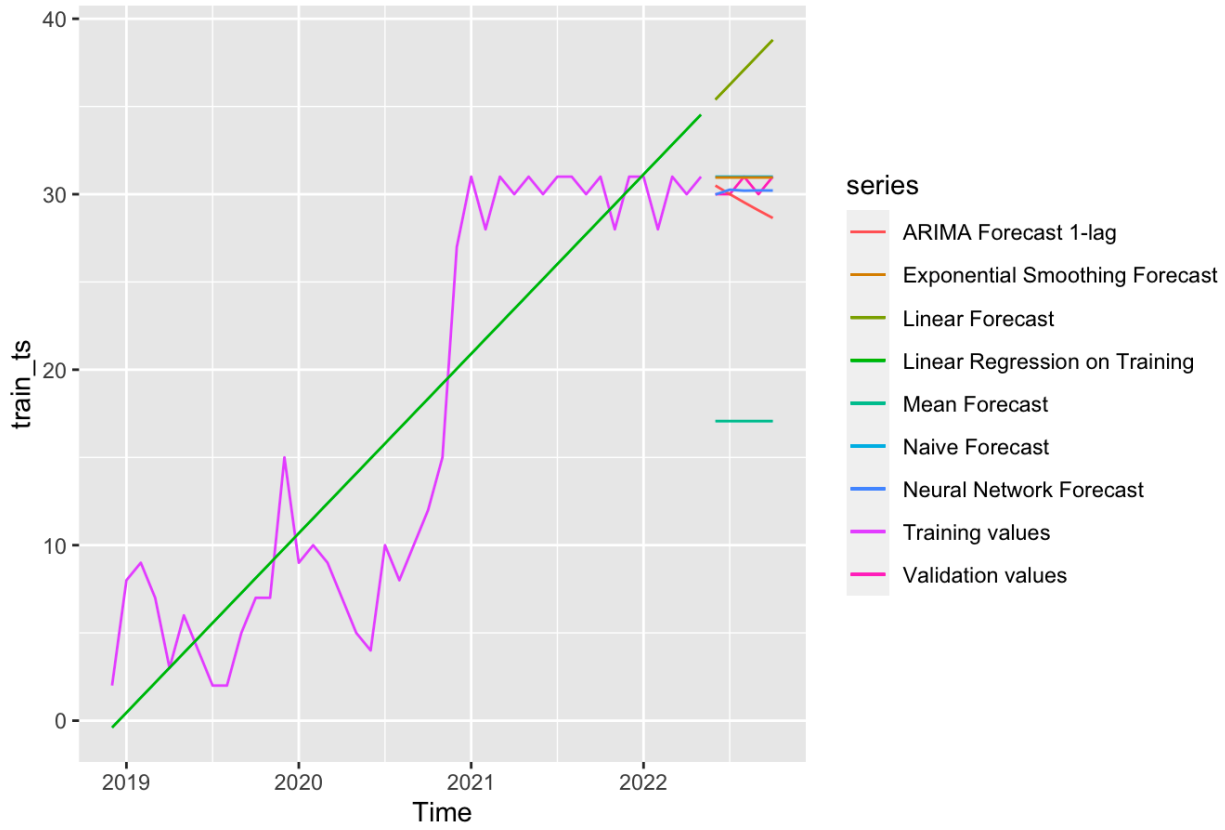
Based on the RMSE values, mean forecast shows the highest error, whereas neural network forecasting shows the lowest. It can be concluded that the neural network method would be the

most appropriate method for forecasting due to its lowest RMSE score (0.5236672) among the six models.

For better understanding of the evaluation comparisons, a time plot describing the actual values and the forecasted values is generated. Based on the plot, it can be concluded that the neural network has forecasted values that are closest to the values in the validation series.

Figure 5.

Time Plot of San Diego Pothole Repair Cases in Training, Validation, and Forecasted series.



Conclusion

The purpose of our analysis was to evaluate the city of San Diego's pothole repair rates over time segmented by district type. After analyzing the geographical characteristics of pothole cases in the city through exploratory data analysis, which included zip codes, council district, and community plan codes, we created a time series by grouping the number of cases by requested date, later changed to month, through the pre-processing phase. Due to omissions present within the dataset's date variable, our forecasting series started in December 2018 and spanned through October 2022 to avoid any skewness within the data. The pre-processed dataset was partitioned into training and validation sets before six different forecasting methods were performed on the trained dataset to compare towards the validated dataset. Using the metric RMSE to evaluate the models' performances, the neural network method serves as the best forecasting method for forecasting pothole repair rates because it generated the lowest RMSE score among the other five models and it forecasted values that were nearly similar towards the validation series. The city of San Diego should take this into account if using the neural network method to forecast, analyze, and highlight aspects on where the city is doing well versus where they need improvement towards repairing potholes within the city's districts and within a reasonable time frame. The approach would also be essential for the city's department to allocate the required funds to maintain effective road quality and keep commuters safe.

References

August, J. W. (2022, February 26). San Diego getting little rain this year, but the potholes are still sprouting. Times of San Diego. https://timesofsandiego.com/life/2022/02/25/san-diego-getting-little-rain-this-year-but-the-potholes-are-still-sprouting/

Garrick, D. (2022, June 30). *San Diego paying out more than $300K to settle two pothole injury lawsuits.* The San Diego Union Tribune. https://www.sandiegouniontribune.com/news/politics/story/2022-06-30/san-diego-paying-out-more-than-300k-to-settle-two-pothole-injury-lawsuits

McCandless, J. (2022, February 23). *10 states and cities with the biggest pothole problems*. Newsweek.https://www.newsweek.com/10-states-cities-biggest-pothole-problems-1679840#:~:text=According%20to%20AARP%2C%20potholes%20cause,over%20%241%2C000%20in%20severe%20cases

Street resurfacing and Pothole Repair. The City of San Diego. (2022). https://www.sandiego.gov/street-div/services/street-resurfacing-pothole-repair

Yoon, J., Kim, A., Tashchyan, G. (2022). ADS 506 – Team 1. Github. https://github.com/hjyoon16/ADS506-Team1

Appendix

```
# Import packages

library(dplyr)

library(tidyverse)

library(fpp2)

library(caret)

set.seed(506)
# Data Import
ph <- read.csv("/Users/yhjnthn/Documents/USD_MS-
ADS/ADS506/project/get_it_done_pothole_requests_datasd_v1.csv")
# Changing date_requested format to Date format
ph$date <- as.Date(ph$date_requested)
# Number of rows
count(ph) # There are 26590 cases in the data set.

##       n
## 1 26590

# Data Summary
summary(ph)

##  service_request_id service_request_parent_id sap_notification_number
##  Min.   : 131444    Min.   :  93344           Min.   :4.03e+10
##  1st Qu.:3346050    1st Qu.:3293984           1st Qu.:4.03e+10
##  Median :3608736    Median :3496503           Median :4.03e+10
##  Mean   :3558615    Mean   :3493908           Mean   :4.03e+10
##  3rd Qu.:3738606    3rd Qu.:3701681           3rd Qu.:4.03e+10
##  Max.   :3981658    Max.   :3981196           Max.   :4.03e+10
##                     NA's   :19268             NA's   :8288
##  date_requested     case_age_days     case_record_type   service_name
##  Length:26590       Min.   : -45.00   Length:26590       Length:26590
##  Class :character   1st Qu.:   2.00   Class :character   Class :character
##  Mode  :character   Median :   5.00   Mode  :character   Mode  :character
##                     Mean   :  20.73
##                     3rd Qu.:  15.00
##                     Max.   :1818.00
##
##  service_name_detail date_closed           status              lat
##  Mode:logical        Length:26590       Length:26590       Min.   :32.54
##  NA's:26590          Class :character   Class :character   1st Qu.:32.75
##                      Mode  :character   Mode  :character   Median :32.79
##                                                            Mean   :32.80
##                                                            3rd Qu.:32.84
##                                                            Max.   :33.10
##
##       lng         street_address        zipcode      council_district
##  Min.   :-117.3   Length:26590       Min.   :91911   Min.   :1.000
##  1st Qu.:-117.2   Class :character   1st Qu.:92105   1st Qu.:3.000
##  Median :-117.1   Mode  :character   Median :92114   Median :5.000
##  Mean   :-117.1                      Mean   :92111   Mean   :4.802
##  3rd Qu.:-117.1                      3rd Qu.:92123   3rd Qu.:7.000
##  Max.   :-116.9                      Max.   :92182   Max.   :9.000
```

```
##                                      NA's   :134     NA's   :205
##  comm_plan_code  comm_plan_name      park_name         case_origin
##  Min.   : 1.00   Length:26590       Length:26590       Length:26590
##  1st Qu.:11.00   Class :character   Class :character   Class :character
##  Median :27.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :29.93
##  3rd Qu.:42.00
##  Max.   :99.00
##  NA's   :7
##    referred           iamfloc              floc           public_description
##  Length:26590       Length:26590       Length:26590       Length:26590
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       date
##  Min.   :2017-01-09
##  1st Qu.:2021-05-07
##  Median :2021-12-19
##  Mean   :2021-11-08
##  3rd Qu.:2022-04-15
##  Max.   :2022-11-05
##
```

Investigating most common areas that require repairs, we will look at zip codes, council_district, and comm_plan_code for the investigation.

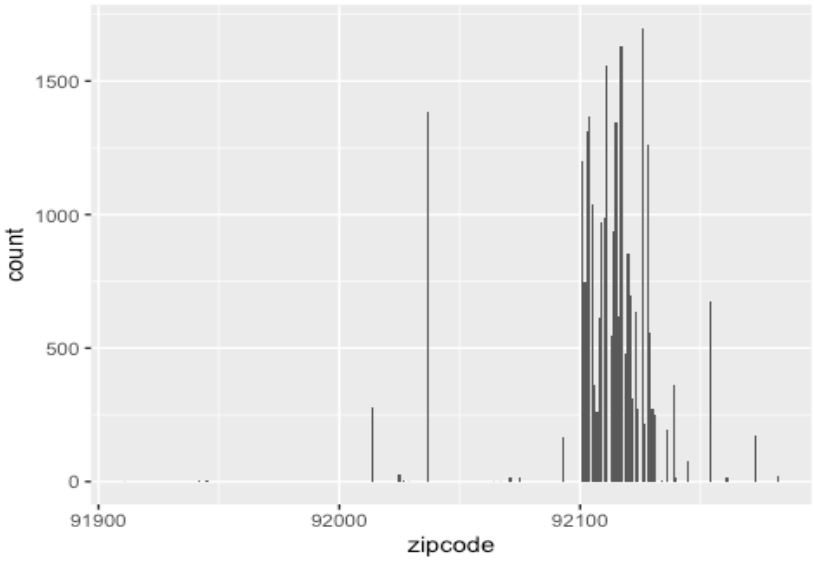
```
# Zip Code Count
ph %>% count(zipcode, sort = TRUE)
```

```
##    zipcode    n
## 1    92126 1700
## 2    92117 1629
## 3    92111 1556
## 4    92037 1387
## 5    92104 1368
## 6    92115 1344
## 7    92103 1314
## 8    92128 1264
## 9    92101 1201
## 10   92105 1040
## 11   92110  987
## 12   92109  971
## 13   92114  938
## 14   92120  854
## 15   92102  748
## 16   92121  700
## 17   92154  676
## 18   92123  636
## 19   92116  617
## 20   92108  615
## 21   92129  555
## 22   92113  546
## 23   92119  478
## 24   92106  362
## 25   92139  362
## 26   92122  313
## 27   92014  281
## 28   92130  274
```

```
## 29   92124  272
## 30   92107  264
## 31   92131  248
## 32   92127  217
## 33   92136  192
## 34   92173  172
## 35   92093  167
## 36      NA  134
## 37   92145   75
## 38   92025   26
## 39   92182   22
## 40   92071   18
## 41   92075   18
## 42   92161   18
## 43   92140   16
## 44   92134    4
## 45   91945    3
## 46   91942    2
## 47   92027    2
## 48   91911    1
## 49   92029    1
## 50   92064    1
## 51   92067    1
```

```
# Histogram
library(ggplot2)
ggplot(ph) +
  geom_bar(mapping = aes(zipcode))
```

```
## Warning: Removed 134 rows containing non-finite values (stat_count).
```



The 10 most common areas that require repair are zipcodes 92126, 92117, 92111, 92037, 92104, 92115, 92103, 92128, 92101, and 92105.

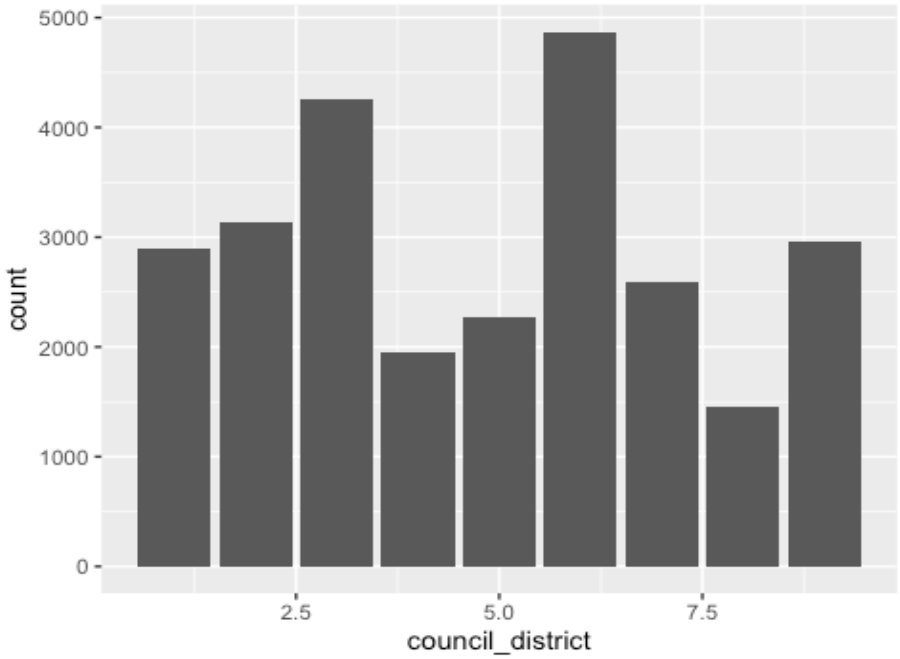
```
ph %>% count(council_district, sort = TRUE)
```

```
##    council_district    n
## 1                 6 4873
## 2                 3 4257
## 3                 2 3134
```

```
## 4                 9 2954
## 5                 1 2903
## 6                 7 2586
## 7                 5 2269
## 8                 4 1947
## 9                 8 1462
## 10               NA  205

# Histogram
ggplot(ph) +
  geom_bar(mapping = aes(council_district))

## Warning: Removed 205 rows containing non-finite values (stat_count).
```



The most common council districts that require repairs are 6, 3, 2, 9, 1, 7, 5, 4, and 8, in decreasing order.

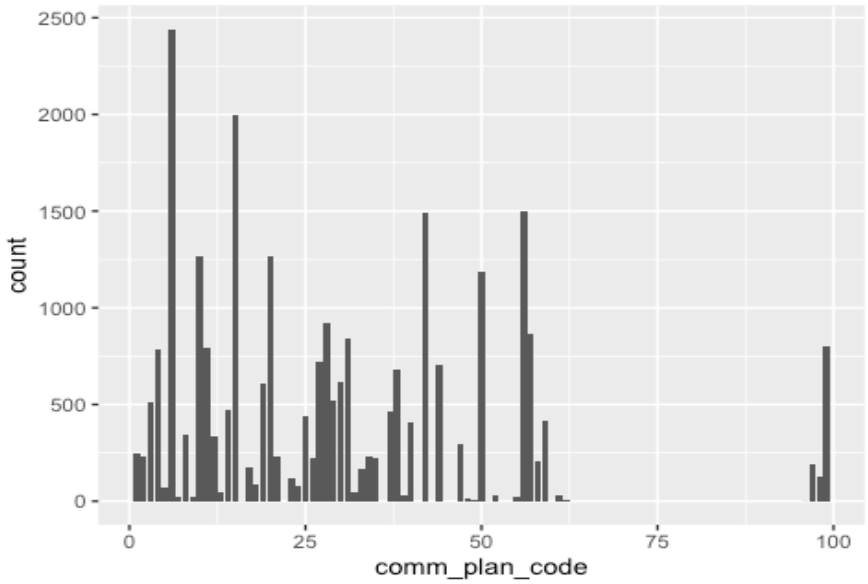
```
ph %>% count(comm_plan_code, sort = TRUE)

##    comm_plan_code    n
## 1               6 2442
## 2              15 1993
## 3              56 1496
## 4              42 1491
## 5              20 1270
## 6              10 1265
## 7              50 1188
## 8              28  919
## 9              57  867
## 10             31  841
## 11             99  800
## 12             11  789
## 13              4  786
## 14             27  720
## 15             44  703
## 16             38  683
## 17             30  614
```

```
## 18             19  607
## 19             29  523
## 20              3  512
## 21             14  472
## 22             37  460
## 23             25  439
## 24             59  417
## 25             40  410
## 26              8  346
## 27             12  339
## 28             47  292
## 29              1  250
## 30             34  235
## 31              2  232
## 32             21  232
## 33             35  227
## 34             26  223
## 35             58  208
## 36             97  194
## 37             17  172
## 38             33  165
## 39             98  129
## 40             23  116
## 41             18   89
## 42             24   75
## 43              5   68
## 44             13   50
## 45             32   44
## 46             52   33
## 47             61   31
## 48             39   28
## 49             55   26
## 50              9   20
## 51              7   19
## 52             48   17
## 53             62    7
## 54             NA    7
## 55             49    6
## 56             16    1
## 57             51    1
## 58             96    1
```

```
# Histogram
ggplot(ph) +
  geom_bar(mapping = aes(comm_plan_code))
```

```
## Warning: Removed 7 rows containing non-finite values (stat_count).
```



The 10 most common community plan codes that require repairs are 6, 15, 56, 42, 20, 10, 50, 28, 57, and 31.

```
# Aggregating by date
ph2 <- aggregate(ph$service_request_id, by=list(ph$date), FUN=length)
names(ph2)[1] = "date"
names(ph2)[2] = "count"

# Grouping by month
ph3 <- mutate(ph2, month = format(date, "%Y-%m"))

# Aggregating by date
ph4 <- aggregate.data.frame(ph3, by = list(ph3$month), FUN = length)

# subsetting into relevant months and training/validation sets
ph_train <- ph4[11:52,]
ph_valid <- ph4[53:57,]
train_ts <- ts(ph_train$count, start = c(2018, 12), frequency = 12)
valid_ts <- ts(ph_valid$count, start = c(2022, 06), frequency = 12)
train_ts

##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2018                                                2
## 2019   8   9   7   3   6   4   2   2   5   7   7  15
## 2020   9  10   9   7   5   4  10   8  10  12  15  27
## 2021  31  28  31  30  31  30  31  31  30  31  28  31
## 2022  31  28  31  30  31

valid_ts

##      Jun Jul Aug Sep Oct
## 2022  30  30  31  30  31
```

It is noted that, beginning December 2020, the data is more valid than the previous months. Thus, the data from December 2020 will be used for model construction.

Naive Forecast

```
ph_naive <- naive(train_ts, 5)
naive_pred <- forecast(ph_naive, 5)
accuracy(naive_pred$mean, valid_ts)
```

```
##            ME      RMSE MAE MPE MAPE       ACF1 Theil's U
## Test set -0.6 0.7745967 0.6  -2    2 -0.4666667 0.8120723
```

Mean forecast

```
ph_mean <- meanf(train_ts)
mean_pred <- forecast(ph_mean, h= 5)
accuracy(mean_pred$mean, valid_ts)
```

```
##                ME     RMSE      MAE      MPE     MAPE       ACF1 Theil's U
## Test set 13.32857 13.33757 13.32857 43.82949 43.82949 -0.4666667  15.56792
```

Linear Regression

```
ph_lm <- tslm(train_ts ~ trend)
lm_pred <- forecast(ph_lm, h=5)
accuracy(lm_pred$mean, valid_ts)
```

```
##                ME     RMSE     MAE       MPE     MAPE      ACF1 Theil's U
## Test set -6.69432 6.769359 6.69432 -22.01563 22.01563 0.2953234   8.17101
```

ARIMA with 1 lag

```
ph_arima <- arima(train_ts, order = c(1,0,0))
arima_pred <- forecast(ph_arima, 5)
accuracy(arima_pred$mean, valid_ts)
```

```
##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.8485574 1.326189 1.048313 2.746315 3.412168 0.1483457  1.700984
```

Exponential Smoothing

```
ph_ets <- ets(train_ts)
ets_pred <- forecast(ph_ets, 5)
summary(ets_pred)
```

```
##
## Forecast method: ETS(A,N,N)
##
## Model Information:
## ETS(A,N,N)
##
## Call:
##  ets(y = train_ts)
##
##   Smoothing parameters:
##     alpha = 0.953
##
##   Initial states:
##     l = 2.2859
##
##   sigma:  3.4491
##
##      AIC     AICc      BIC
## 264.9356 265.5672 270.1486
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.7162394 3.366024 2.446651 -1.560644 25.63784 0.2422427
##                      ACF1
## Training set -0.05145034
##
```

```
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jun 2022       30.95494 26.53468 35.37520 24.19473 37.71514
## Jul 2022       30.95494 24.84879 37.06108 21.61640 40.29348
## Aug 2022       30.95494 23.53663 38.37325 19.60962 42.30026
## Sep 2022       30.95494 22.42396 39.48592 17.90793 44.00194
## Oct 2022       30.95494 21.44053 40.46934 16.40391 45.50596
```

ANN model was fit.

```
accuracy(ets_pred$mean, valid_ts)

##                    ME      RMSE       MAE       MPE     MAPE       ACF1 Theil's U
## Test set -0.5549385 0.740241 0.5909877 -1.851733 1.968021 -0.4666667 0.7763702
```

Neural Network

```
ph_nn <- nnetar(train_ts)
nn_pred <- forecast(ph_nn, 5)
accuracy(nn_pred$mean, valid_ts)

##                   ME      RMSE       MAE       MPE     MAPE       ACF1 Theil's U
## Test set 0.2262782 0.5236672 0.4155503 0.7201501 1.351057 -0.6068247 0.6824996
```

Visualization

```
autoplot(train_ts, series = "Training values") +
  autolayer(valid_ts, series = "Validation values") +
  autolayer(naive_pred$mean, series = "Naive Forecast") +
  autolayer(mean_pred$mean, series = "Mean Forecast") +
  autolayer(lm_pred$mean, series = "Linear Forecast") +
  autolayer(arima_pred$mean, series = "ARIMA Forecast 1-lag") +
  autolayer(ets_pred$mean, series = "Exponential Smoothing Forecast") +
  autolayer(nn_pred$mean, series = "Neural Network Forecast") +
  autolayer(ph_lm$fitted.values, series = "Linear Regression on Training")
```