Individual Assignment #1

AD699: Data Mining for Business Analytics

Boston University

Spring 2020

Yuqi Zheng

1, 2.

By using code below to read the file and find the dimension of the dataset

```
Boston<-read.csv("boston311.csv")
dim(Boston)
```

We find there are *259496* rows in this dataframe.

3.

Creating a new variable called “Boston1” and filter “1207” as my precinct by the following code

```
Boston1<-filter(Boston,precinct=="1207")
dim(Boston1)
```

We find there are *1239* rows in this dataset.

4.

In order to change the closed_dt data type, we use the code

```
str(Boston1)
Boston1$closed_dt <- as.Date(Boston1$closed_dt)
```

and recheck it by using

```
str(Boston1)
```

and then create a new variable called month and extracted the data from closed_dt, and just keep the month

```
Boston1 <-mutate(Boston1, month = closed_dt)
```

It is more visualize to change the month as numeric

```
Boston1$month <- as.numeric(format(Boston1$closed_dt, "%m"))
```

```
View(Boston1)
```

and we want to remove the NA value in the month column.

Firstly, we check if the dataset contains the NA value

```
anyNA(Boston1)
```

create a new variable called Boston2 to store the new variable month from Boston1 dataset.

Remove the NA value in the month column.

```
Boston2 <-mutate(Boston1, month = months(Boston1$closed_dt)) %>%
na.omit(Boston1$month)
```

Set the data type as numeric

```
Boston2$month <- as.numeric(format(Boston2$closed_dt, "%m"))
View(Boston2)
```

Recheck if the dataset still contains NA value

```
anyNA(Boston2)
dim(Boston2)
```

Finally, we get there are *677* rows right now.

5.

We remove the open_dt column by using the following code

```
Boston2 <- Boston2[,-2]
```

We can view and check the dimension of the dataset again.

```
View(Boston2)
dim(Boston2)
```

6.

To calculate the percentage of the cases in my precinct have a case_status of closed, we could use the code

```
dfsum <- summary(Boston2$case_status)
Perct <- dfsum["Closed"]/(dfsum["Closed"]+dfsum["Open"])
percent(Perct)
```

we find the percentage of closed status in my precinct is *100%*

Use the same way to calculate the overall percentage of closed status

```
dfsum1 <- summary(Boston1$case_status)
Perct1 <- dfsum1["Closed"]/(dfsum1["Closed"]+dfsum1["Open"])
percent(Perct1)
```

we find the percentage of closed status overall is *82.2%*

7.

Identify the six most common reasons in my precinct by using code

```
reason_kind <- group_by(Boston2, reason)
reason_n <-count(Boston2, reason)
reason_6 <- arrange(reason_n, desc(n))
head(reason_6,6)
```

The six most common reasons in my precinct are *Sanitation, Enforcement & Abandoned Vehicles, Street Cleaning, Code Enforcement, Park Maintenance & Safety and Highway Maintenance*.

8.

Create a new dataframe that only contains the six most common reasons in my precinct by using the code as below:

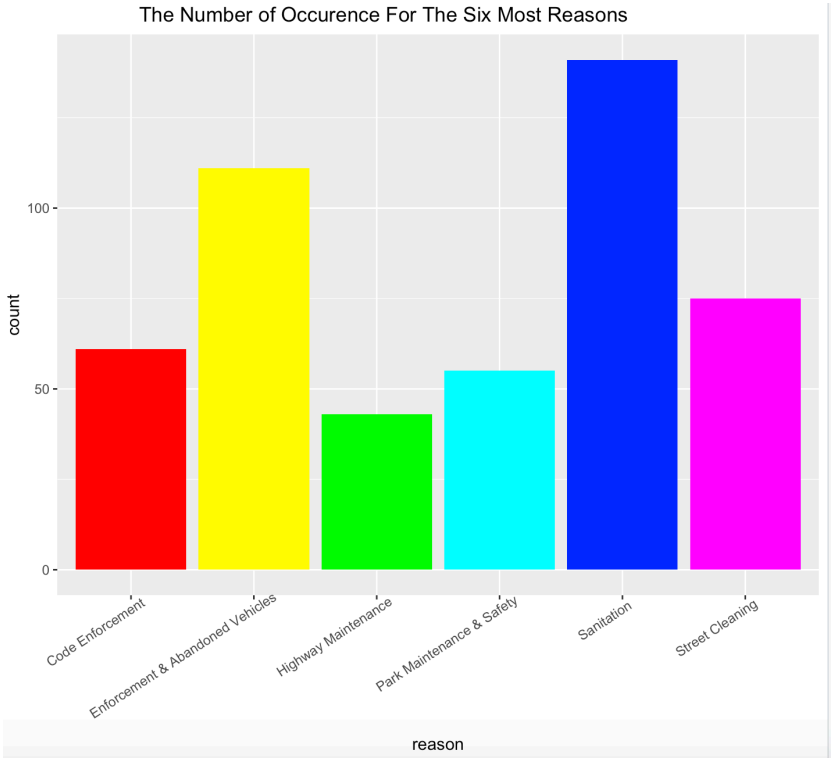
```
Boston3 <- filter(Boston2, reason == "Enforcement & Abandoned Vehicles"| reason == "Code Enforcement"|reason == "Street Cleaning"|reason == "Sanitation"|reason == "Highway Maintenance"|reason =="Park Maintenance & Safety")
View(Boston3)
str(Boston3)
```

9.

Using ggplot to create a bar plot of the six most common reasons in my precinct by using code

```
counts <- table(Boston3$reason)
ggplot(Boston3,aes(x=reason))+geom_bar(fill=rainbow(n=6))+ggtitle("The Number of Occurence For The Six Most Reasons")+theme(plot.title = element_text(hjust=0.3), axis.text.x = element_text(angle=33, hjust=0.9, vjust=1))
```

As the bar plot shows, the most common reason among these six reasons is sanitation. It is more than twice than the fifth reason, park maintenance & safety. The rank 2 common reason is enforcement & abandoned vehicles. This reason is also more than twice than the sixth reason highway maintenance. The third one is street cleaning and the fourth one is code enforcement.



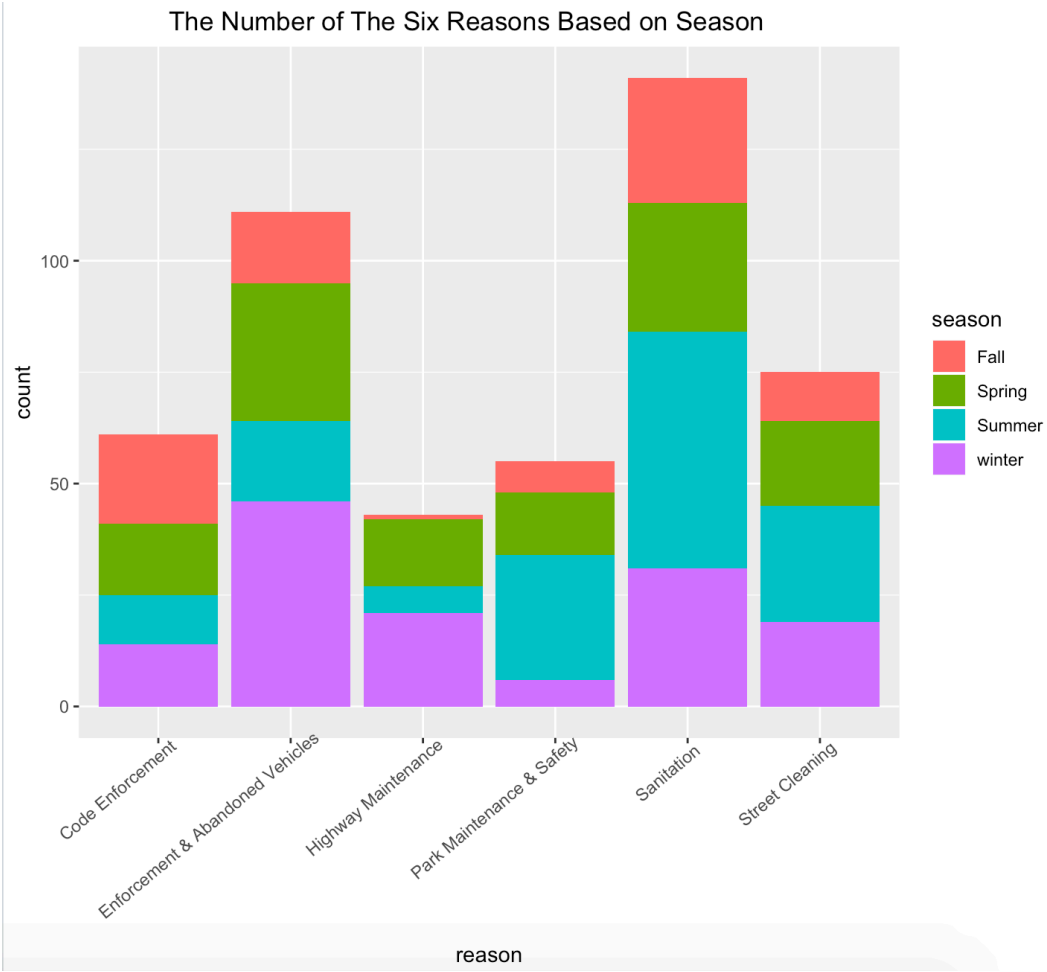
10.

Create a bar plot based on the season of the year by using ggplot. Firstly, we are using ifelse function to distinguish the seasons by month.

```
Boston3$season <-rep("winter", nrow(Boston3))
Boston3$season <-ifelse(Boston3$month > 3, "Spring", Boston3$season)
Boston3$season <-ifelse(Boston3$month > 6, "Summer", Boston3$season)
Boston3$season <-ifelse(Boston3$month > 9, "Fall", Boston3$season)
View(Boston3)
ggplot(Boston3, aes(x=reason,fill=season))+geom_bar()+ggtitle('The Number of The Six Reasons Based on Season')+theme(plot.title = element_text(hjust=0.4), axis.text.x = element_text(angle=40, hjust=0.9, vjust=1))
```

In the bar plot, we could see the reason Street Cleaning occurs more in the summer and performs quite equally in the rest seasons. For the Code Enforement reason, it quite occurs more in Fall,

and then Spring and Winter. For the Fall and Summer quarter, Sanitation counts the most. In Spring and Winter Enforcement & Abandoned Vehicles counts the most.



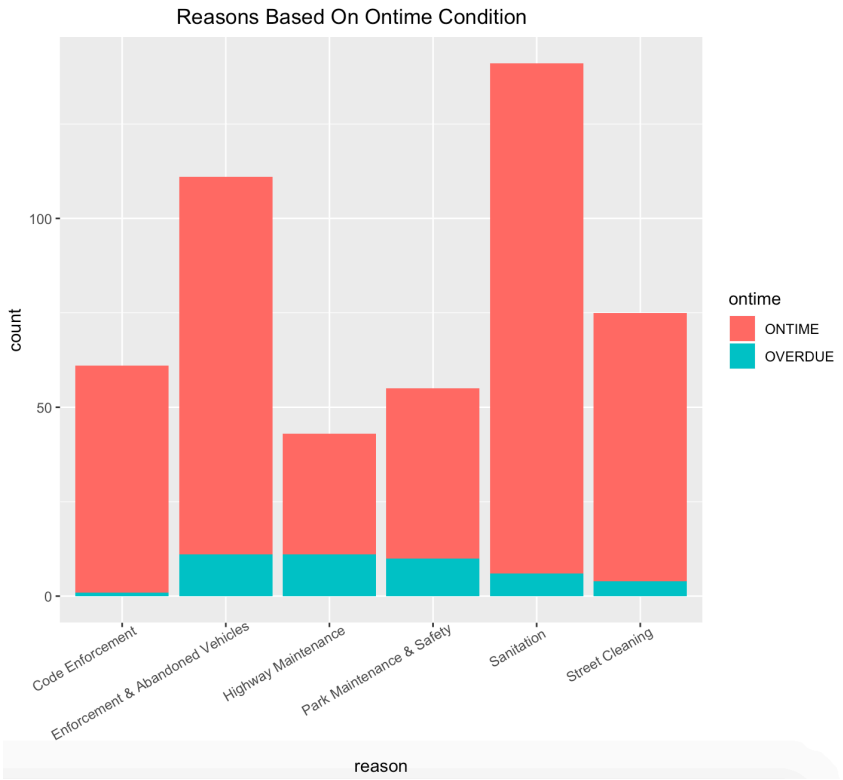
11.

Create a bar plot by fill the data “ontime”

```
ggplot(Boston3, aes(x=reason,fill=ontime))+geom_bar()+ggtitle('Reasons Based On Ontime Condition')+theme(plot.title = element_text(hjust=0.4), axis.text.x = element_text(angle=30, hjust=0.9, vjust=1))
```

As the bar plot shows, the Enforcement & Abandoned Vehicles and Highway Maintenance have the most overdue counts. The Code Enforcement and the fewest overdue counts, it might as the

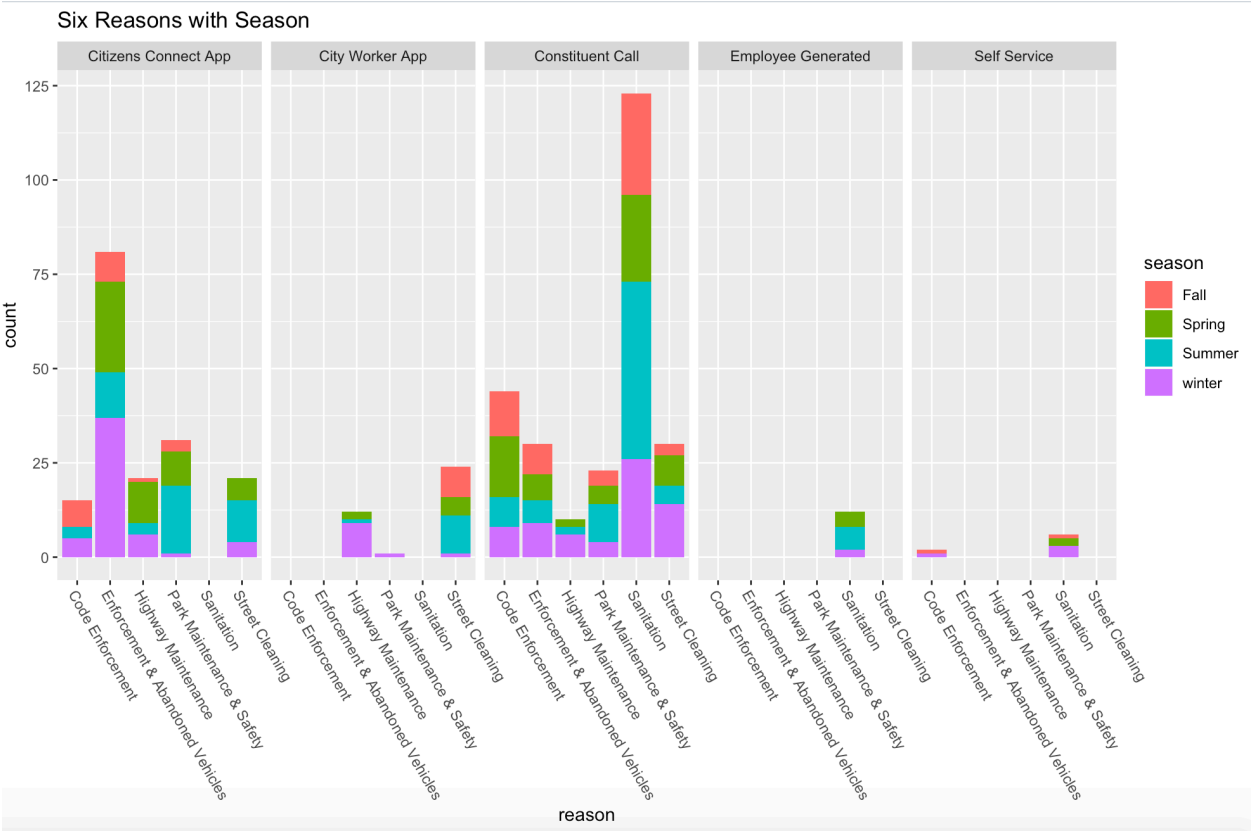
same overdue percentage as the Sanitation reason. Overall, as the percentage of this dataset, the Highway Maintenance and Park Maintenance & Safety should improve the overdue problem in the future.



12.

In this question, we use the following code and get the bar plot as attached below

```
ggplot(Boston3, aes(x=reason,fill=season)) + geom_bar()+ theme(axis.text.x=element_text(angle = -60, hjust = 0))+ggtitle("Six Reasons with Season")+ facet_grid(~source)
```



As the bar plot shows, there are five cources in my precinct and the Constituent Call is the most popular source among those five sources. People like to report the sanitation problem by using constituent call mostly. The second popular source is the Citizens Connect App. People always like to report Enforcement & Abandoned Vehicles problem by using this source. In addition, people rarely use self-service and employee generated source to solve problems. Only few people will report Sanitation and Code Enforcement by using self-service.

13.

In this question, we use the following code to show the 5 most common types in my precinct

```
typea<-group_by(Boston3,type)
typeb<-count(Boston3,type)
type5<-arrange(typeb,desc(n))
```

```
head(type5,5)
```

We can find the 5 most common types in my precinct are *Parking Enforcement, Schedule a Bulk Item Pickup, Missed Trash/Recycling/Yard Waste/Bulk Item, Request for Pothole Repair and Requests for Street Cleaning.*

And then, we use the following code to create a new dataset called "type" which only contains data for the 5 most common type.

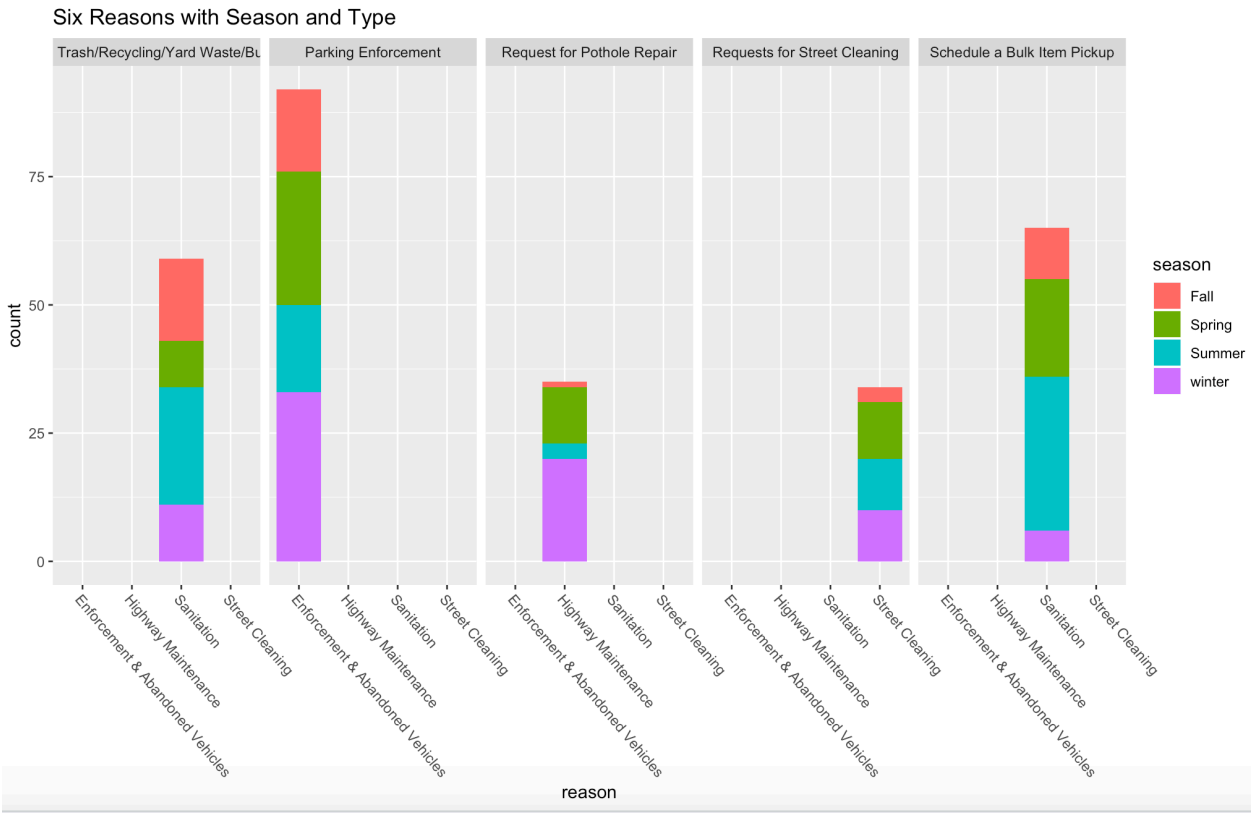
```
five<-c("Parking Enforcement","Schedule a Bulk Item Pickup","Missed Trash/Recycling/Yard Waste/Bulk Item","Request for Pothole Repair","Requests for Street Cleaning")
type<-filter(Boston3,type %in% five)
View(type)
```

14.

In this question, we build a bar plot to show the six reasons with filling date season in type.

```
ggplot(type, aes(x=reason,fill=season)) + geom_bar()+ theme(axis.text.x=element_text(angle = -60, hjust = 0))+ggtitle("Six Reasons with Season and Type")+facet_grid(~type)
```

As the bar plot shows, Enforcement & Abandoned Vehicle problem mostly occurs because the type of Parking Enforcement and mainly occurs in Winter, as well as in Spring. Sanitation problem occurs due to Trash/Recycling/Yard Waste/Bulk Item and Schedule a Bulk Item Pickup and it mostly happens in Summer. Highway Maintenance problem only occurs by the type of Request for Pothole Repair and it usually happens in Winter, then Spring, rarely in Fall. Street Cleaning problem only occurs by type of Requests for Street Cleaning and it often happens in Spring, then Summer and Winter.



15.

We use the following code to see the number of levels in my dataframe for the type variable, including the 0 level types by simply using the table function.

```
table(type$type)
```

Then we drop the types with 0 levels by using droplevels function

```
type<-droplevels(type)
table(type$type)
```

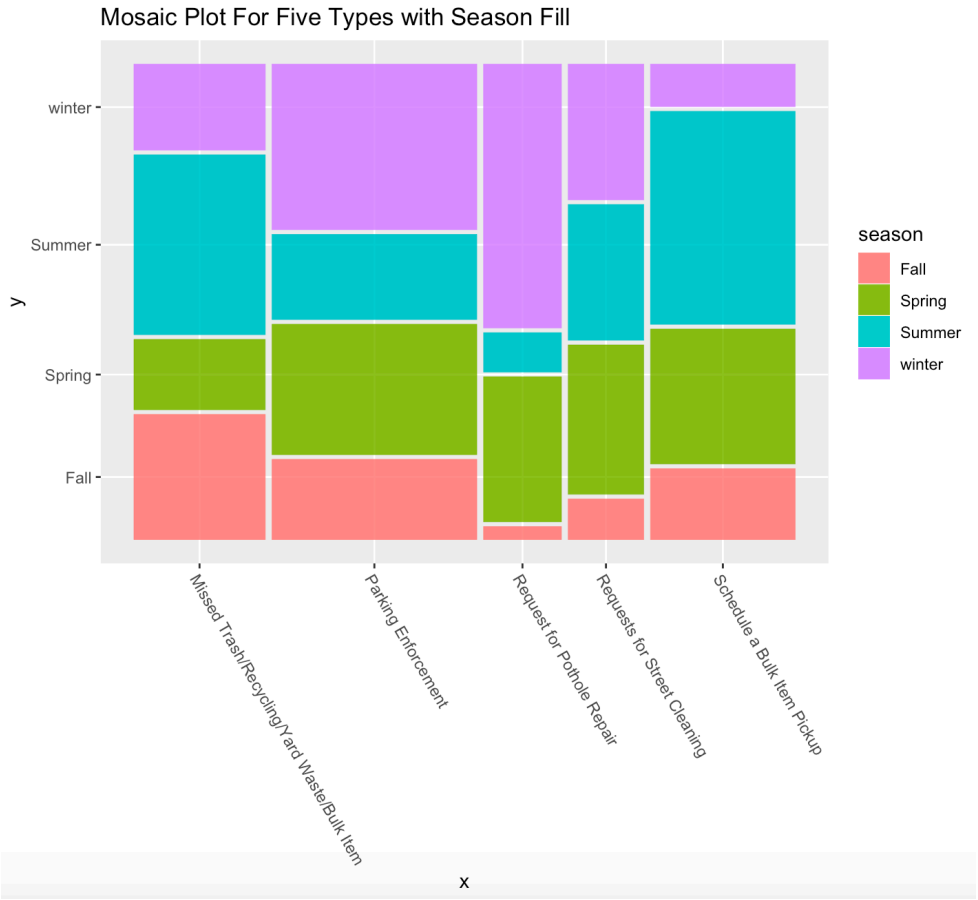
Finally, we get the table only contains the type without 0 levels.

16.

Using ggplot to create a mosaic plot for five types with season fill.

```
ggplot(data=type) + geom_mosaic(aes( x=product(type), fill=season))+
theme(axis.text.x=element_text(angle = -60, hjust = 0))+ggtitle("Mosaic Plot For Five Types
with Season Fill")
```

In this graph, we can see the order from the number of occurrences for the five most common types from big to small is “Parking Enforcement”, “Schedule a Bulk Item Pickup”, “Missed Trash/Recycling/Yard Waste/Bulk Item”, “Request for Pothole Repair” and “Requests for Street Cleaning”. Compare to other types, season distributed quite evenly in Parking Enforcement. Type Schedule a Bulk Item Pickup is frequently happened in Summer. Type Requests for Street Cleaning is also evenly distributed in Summer, Spring and Winter, sometimes happened in Fall. Type Request for Pothole Repair rarely happened in Fall but mostly happened in Winter. Type Missed Trash/Recycling/Yard Waste/Bulk Item usually happened in Summer and Fall.



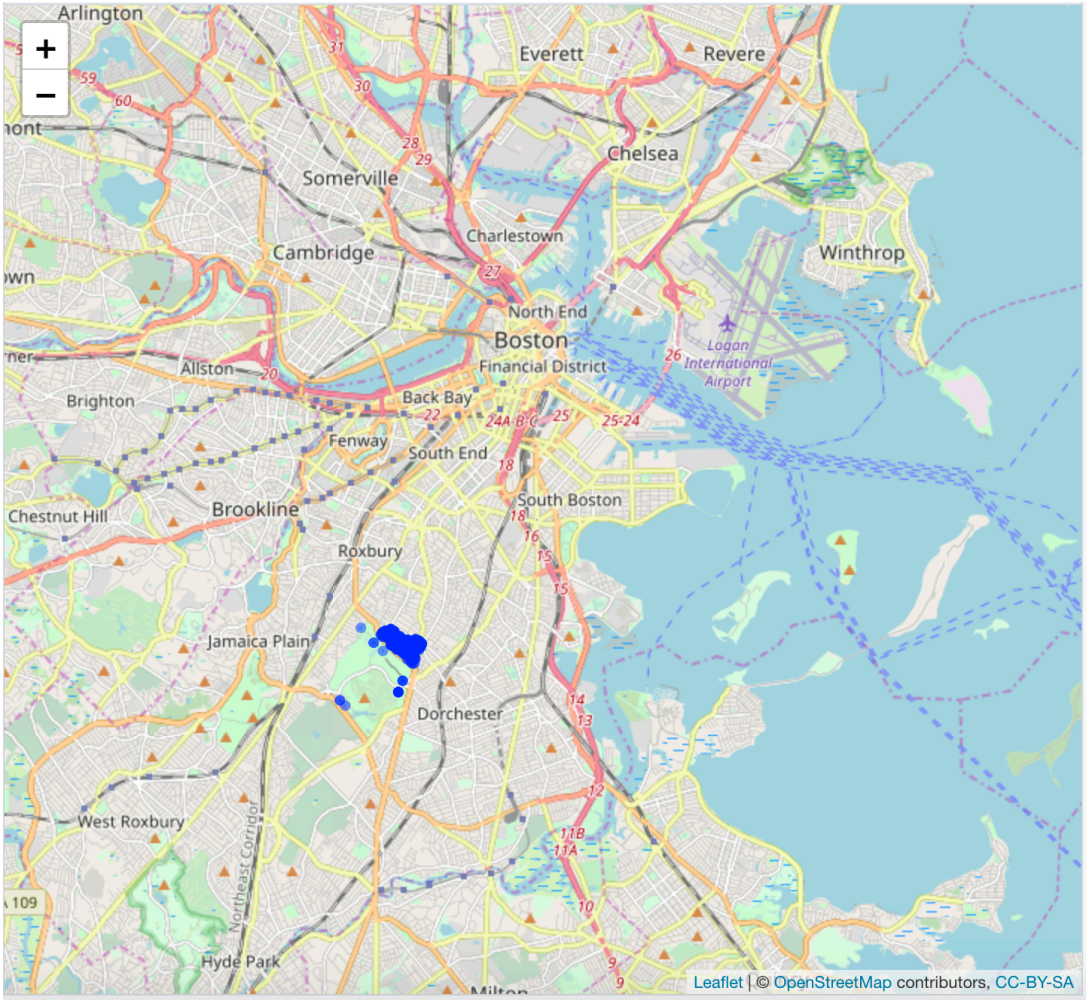
17.

```
install.packages("leaflet")

library(leaflet)
```

18.

By running the following code, we can get a map as below

```
type %>% leaflet() %>% setView(lng=-71.105, lat=42.35, zoom=10) %>%

  addCircles %>% addTiles()
```

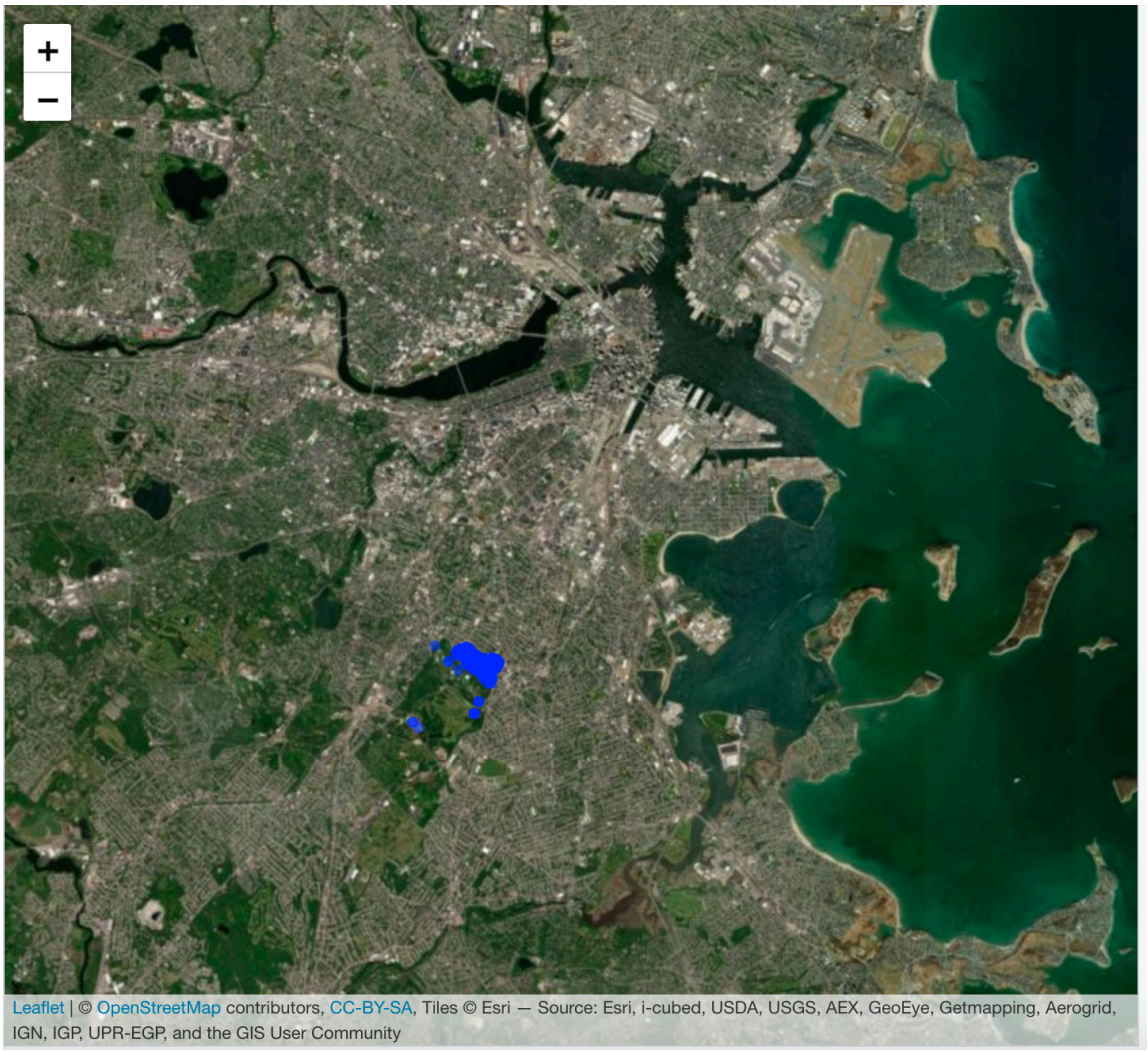


19.

Through code

```
type %>% leaflet() %>% setView(lng=-71.105, lat=42.35, zoom=10) %>%
  addCircles %>% addTiles() %>% addProviderTiles(providers$Esri.WorldImagery)
```

We get a map



20.

Through code,

```
type %>% leaflet() %>% setView(lng=-71.105, lat=42.35, zoom=10) %>% addCircles %>%
  addTiles() %>%
  addProviderTiles(providers$Esri.WorldImagery) %>% addMarkers(lng=-71.120996,
lat=42.351444, popup="BU MET")
```

We get