AD 699
Assignment 1
Tiange Chang
Zip code: 78745

1. File downloaded.

```
# AD 699
# Assignment 1
# Tiange Chang
# Zip code: 78745
library(tidyverse)
library(leaflet)
library(tidyr)
library(dplyr)
# 1
# File downloaded
```

2. a

```
Console  Terminal  Jobs
R 4.1.1 · ~/Desktop/AD 699/Assignment 1/
> # 2. a
> setwd('/Users/t/Desktop/AD 699/Assignment 1')
> Aus <- read.csv('Austin_311_Public_Data.csv')
> str(Aus)
'data.frame':   1203460 obs. of  22 variables:
 $ Service.Request..SR..Number: chr  "19-00090956" "20-00135805" "20-00052942" "19-00467964" ...
 $ SR.Description             : chr  "Animal - Proper Care" "Tree Issue ROW" "Pothole Repair" "ARR Dead Animal Collection"
 ...
 $ Method.Received            : chr  "Phone" "Mobile Device" "Web" "Phone" ...
 $ SR.Status                  : chr  "Closed" "Closed" "Closed" "Closed" ...
 $ Status.Change.Date         : chr  "11/23/2020 01:41:21 PM" "11/23/2020 12:02:05 AM" "02/12/2020 12:57:35 PM" "12/11/2019
 09:57:54 AM" ...
 $ Created.Date               : chr  "03/14/2019 03:02:15 PM" "04/07/2020 07:06:32 PM" "02/12/2020 01:55:04 AM" "12/10/2019
 02:17:04 PM" ...
 $ Last.Update.Date           : chr  "11/23/2020 01:41:22 PM" "11/23/2020 12:02:05 AM" "08/11/2020 01:03:41 AM" "12/11/2019
 09:57:54 AM" ...
 $ Close.Date                 : chr  "11/23/2020 01:41:21 PM" "11/23/2020 12:02:05 AM" "02/12/2020 12:57:35 PM" "12/11/2019
 09:57:54 AM" ...
 $ SR.Location                : chr  "4609 RIBBECKE AVE, AUSTIN, TX 78721" "3521 WEST AVE, AUSTIN, TX 78705" "7900 E BEN WH
ITE BLVD WB, AUSTIN, TX" "7100 METROPOLIS DR, AUSTIN, TX 78744" ...
 $ Street.Number              : chr  "4609" "3521" "7900" "7100" ...
 $ Street.Name                : chr  "RIBBECKE" "WEST" "BEN WHITE BLVD WB" "METROPOLIS" ...
 $ City                       : chr  "AUSTIN" "AUSTIN" "AUSTIN" "AUSTIN" ...
 $ Zip.Code                   : int  78721 78705 78741 78744 78741 78723 78712 78741 78745 78745 ...
 $ County                     : chr  "TRAVIS" "TRAVIS" "TRAVIS" "TRAVIS" ...
 $ State.Plane.X.Coordinate   : num  3132348 3114174 3131888 3127538 3125956 ...
 $ State.Plane.Y.Coordinate   : num  10075731 10083407 10053622 10048498 10054146 ...
 $ Latitude.Coordinate        : num  30.3 30.3 30.2 30.2 30.2 ...
 $ Longitude.Coordinate       : num  -97.7 -97.7 -97.7 -97.7 -97.7 ...
 $ X.Latitude.Longitude.      : chr  "(30.28058076, -97.68531793)" "(30.30285681, -97.74231249)" "(30.21983764, -97.6884401
1)" "(30.20603574, -97.70259216)" ...
 $ Council.District           : int  1 9 3 2 3 9 9 3 5 3 ...
 $ Map.Page                   : chr  "586L" "555S" "646F" "646N" ...
 $ Map.Tile                   : chr  "MM23" "MJ25" "ML18" "ML17" ...
```

2. b

```
# 2. b
?str
# This function compactly display the structure of an arbitrary R object.
# My data frame contained 1,203,460 rows and 22 columns.
```

3. a

```
22  # 3. a
23  Aus1 <- filter(Aus, Zip.Code == '78745')
24  dim(Aus1)
25  # The new data frame contained 78,489 rows and 22 columns.
26:1  (Top Level)

Console  Terminal  Jobs
R 4.1.1 · ~/Desktop/AD 699/Assignment 1/
> # 3. a
> Aus1 <- filter(Aus, Zip.Code == '78745')
> dim(Aus1)
[1] 78489    22
```

4. a

```
27  # 4. a
28  anyNA(Aus1)
29  sum(is.na(Aus1))
30  # There are NA values in my data frame.
31
32  # I used 'anyNA' function to know this, the returned value was true, so it means
33  # there is NA values in my data frame.
34
35  # There are 3556 NAs in the data frame.
30:1  (Top Level)

Console  Terminal  Jobs
R 4.1.1 · ~/Desktop/AD 699/Assignment 1/
> # 4. a
> anyNA(Aus1)
[1] TRUE
> sum(is.na(Aus1))
[1] 3556
```

4. b

```
37  # 4. b
38  # complete. cases() function in R Language is used to return a logical vector
39  # with cases which are complete, i.e., no missing value.
40
41  (78489-sum(is.na(Aus1)))/78489 # percentage of rows in the data frame
42                                 # are complete cases.
43  # The percentage of the rows in the data frame are complete cases is 95.47%.
42:34  (Top Level)

Console  Terminal  Jobs
R 4.1.1 · ~/Desktop/AD 699/Assignment 1/
> (78489-sum(is.na(Aus1)))/78489 # percentage of rows in the data frame
[1] 0.9546943
```

4. c

```
# 4. c
Aus1 <- Aus1 %>% mutate_all(na_if, "")
```

4. d

```
48  # 4. d
49  NumberOfMissingValues <- sum(is.na(Aus1))
50  NumberOfMissingValues
51  # Now, there are 19,042 NAs in the data frame.
```

55:1 (Top Level)

Console Terminal Jobs

R 4.1.1 · ~/Desktop/AD 699/Assignment 1/

```
> # 4. d
> NumberOfMissingValues <- sum(is.na(Aus1))
> NumberOfMissingValues
[1] 19042
```

4. e

```
53  # 4. e
54  PercentageOfMissingValues <- (78489-sum(is.na(Aus1)))/78489
55  PercentageOfMissingValues
56  # The percentage of the rows in the data frame are complete cases is 75.74%.
57  # The answer was different because in part b, I did not include blank cells.
```

56:1 (Top Level)

Console Terminal Jobs

R 4.1.1 · ~/Desktop/AD 699/Assignment 1/

```
> # 4. e
> PercentageOfMissingValues <- (78489-sum(is.na(Aus1)))/78489
> PercentageOfMissingValues
[1] 0.7573928
```

4. f

```
# 4. f
df <- data.frame(NumberOfMissingValues, PercentageOfMissingValues)
View(df)

df1 <- filter(Aus1, Close.Date == 'NA')
View(df1)
# There is no NA in Close.Date, but I looked inside the data and found that some
# of the create date and the close date are the same.

df2 <- filter(Aus1, Close.Date == Created.Date)
View(df2)
# This means the duration is zero. There are 1,439 data have duration of zero.
```

4. g

```
# 4. g
Aus1 <- Aus1 %>% drop_na(City)
```

4. h

```
75 # 4. h
76 dim(Aus1)
77 # By removing the rows that have NA values for the column 'City', I now have
78 # 78,289 rows of data.
```

74:1 (Top Level)

Console Terminal Jobs

R 4.1.1 · ~/Desktop/AD 699/Assignment 1/

```
> # 4. h
> dim(Aus1)
[1] 78289    22
> # By removing the rows that have NA values for the column 'City', I now have
> # 78,289 rows of data.
```

5. a

```
80 # 5. a
81 str(Aus1)
82 # Created.Date and Close.Date is chr. In R, a piece of text is represented as a
83 # sequence of characters (letters, numbers, and symbols).
```

82:1 (Top Level) R Script

Console Terminal Jobs

R 4.1.1 · ~/Desktop/AD 699/Assignment 1/

```
> # 5. a
> str(Aus1)
'data.frame':   78289 obs. of  22 variables:
 $ Service.Request..SR..Number: chr  "19-00351755" "19-00354934" "16-00033203" "16-00112695" ...
 $ SR.Description             : chr  "Austin Code - Request Code Officer" "Austin Code - Request Code Officer" "Animal Contro
l - Assistance Request" "Animal Control - Assistance Request" ...
 $ Method.Received            : chr  "Phone" "Phone" "Phone" "Phone" ...
 $ SR.Status                  : chr  "Closed" "Closed" "Closed" "Closed" ...
 $ Status.Change.Date         : chr  "09/13/2019 12:31:39 PM" "09/16/2019 10:01:32 AM" "10/24/2019 12:08:44 PM" "10/24/2019 1
2:11:34 PM" ...
 $ Created.Date               : chr  "09/13/2019 12:22:30 PM" "09/16/2019 09:51:55 AM" "02/14/2016 12:01:22 PM" "05/11/2016 0
2:34:25 PM" ...
 $ Last.Update.Date           : chr  "11/21/2019 10:53:17 AM" "11/21/2019 10:56:18 AM" "10/24/2019 12:08:43 PM" "10/24/2019 1
2:11:34 PM" ...
 $ Close.Date                 : chr  "09/13/2019 12:31:39 PM" "09/16/2019 10:01:32 AM" "10/24/2019 12:08:44 PM" "10/24/2019 1
2:11:34 PM" ...
 $ SR.Location                : chr  "8401 SEMINARY RIDGE DR, AUSTIN, TX 78745" "4720 S CONGRESS AVE, AUSTIN, TX 78745" "4713
 CLAWSON RD, AUSTIN, TX 78745" "4713 CLAWSON RD, AUSTIN, TX 78745" ...
 $ Street.Number              : chr  "8401" "4720" "4713" "4713" ...
 $ Street.Name                : chr  "SEMINARY RIDGE" "CONGRESS" "CLAWSON" "CLAWSON" ...
 $ City                       : chr  "AUSTIN" "AUSTIN" "AUSTIN" "AUSTIN" ...
 $ Zip.Code                   : int  78745 78745 78745 78745 78745 78745 78745 78745 78745 78745 ...
 $ County                     : chr  "TRAVIS" "TRAVIS" "TRAVIS" "TRAVIS" ...
 $ State.Plane.X.Coordinate   : num  3089629 3106172 3100924 3100924 3092160 ...
 $ State.Plane.Y.Coordinate   : num  10042961 10050938 10053516 10053516 10048055 ...
 $ Latitude.Coordinate        : num  30.2 30.2 30.2 30.2 30.2 ...
 $ Longitude.Coordinate       : num  -97.8 -97.8 -97.8 -97.8 -97.8 ...
 $ X.Latitude.Longitude.      : chr  "(30.1932236, -97.82296358)" "(30.21411526, -97.77003531)" "(30.22153337, -97.78645728)"
"(30.22153337, -97.78645728)" ...
 $ Council.District           : int  5 3 5 5 5 2 5 5 3 5 ...
 $ Map.Page                   : chr  "643X" "644L" "644A" "644A" ...
 $ Map.Tile                   : chr  "ME16" "MH17" "MG18" "MG18" ...
```

5. b

```
# 5. b
install.packages('anytime')
library(anytime)
Aus1$Created.Date <- anydate(Aus1$Created.Date)
Aus1$Close.Date <- anydate(Aus1$Close.Date)
str(Aus1)
# By running the str function, the data type has been successfully converted.
```

```
> library(anytime)
> Aus1$Created.Date <- anydate(Aus1$Created.Date)
> Aus1$Close.Date <- anydate(Aus1$Close.Date)
> str(Aus1)
'data.frame':   78289 obs. of  22 variables:
 $ Service.Request..SR..Number: chr  "19-00351755" "19-00354934" "16-00033203" "16-00112695" ...
 $ SR.Description             : chr  "Austin Code - Request Code Officer" "Austin Code - Request Code Officer" "Animal Cont
rol - Assistance Request" "Animal Control - Assistance Request" ...
 $ Method.Received            : chr  "Phone" "Phone" "Phone" "Phone" ...
 $ SR.Status                  : chr  "Closed" "Closed" "Closed" "Closed" ...
 $ Status.Change.Date         : chr  "09/13/2019 12:31:39 PM" "09/16/2019 10:01:32 AM" "10/24/2019 12:08:44 PM" "10/24/2019
 12:11:34 PM" ...
 $ Created.Date               : Date, format: "2019-09-13" "2019-09-16" "2016-02-14" "2016-05-11" ...
 $ Last.Update.Date           : chr  "11/21/2019 10:53:17 AM" "11/21/2019 10:56:18 AM" "10/24/2019 12:08:43 PM" "10/24/2019
 12:11:34 PM" ...
 $ Close.Date                 : Date, format: "2019-09-13" "2019-09-16" "2019-10-24" "2019-10-24" ...
 $ SR.Location                : chr  "8401 SEMINARY RIDGE DR, AUSTIN, TX 78745" "4720 S CONGRESS AVE, AUSTIN, TX 78745" "47
13 CLAWSON RD, AUSTIN, TX 78745" "4713 CLAWSON RD, AUSTIN, TX 78745" ...
 $ Street.Number              : chr  "8401" "4720" "4713" "4713" ...
 $ Street.Name                : chr  "SEMINARY RIDGE" "CONGRESS" "CLAWSON" "CLAWSON" ...
 $ City                       : chr  "AUSTIN" "AUSTIN" "AUSTIN" "AUSTIN" ...
 $ Zip.Code                   : int  78745 78745 78745 78745 78745 78745 78745 78745 78745 78745 ...
 $ County                     : chr  "TRAVIS" "TRAVIS" "TRAVIS" "TRAVIS" ...
 $ State.Plane.X.Coordinate   : num  3089629 3106172 3100924 3100924 3092160 ...
 $ State.Plane.Y.Coordinate   : num  10042961 10050938 10053516 10053516 10048055 ...
 $ Latitude.Coordinate        : num  30.2 30.2 30.2 30.2 30.2 ...
 $ Longitude.Coordinate       : num  -97.8 -97.8 -97.8 -97.8 -97.8 ...
 $ X.Latitude.Longitude.      : chr  "(30.1932236, -97.82296358)" "(30.21411526, -97.77003531)" "(30.22153337, -97.7864572
8)" "(30.22153337, -97.78645728)" ...
 $ Council.District           : int  5 3 5 5 5 2 5 5 3 5 ...
 $ Map.Page                   : chr  "643X" "644L" "644A" "644A" ...
 $ Map.Tile                   : chr  "ME16" "MH17" "MG18" "MG18" ...
```

5. c

```
# 5. c
Aus1 <- Aus1 %>% mutate(Aus1, Duration = Close.Date - Created.Date)
View(Aus1)
```

5. d

```
> # 5. d
> # My birthday is 09/11.
> library(lubridate)
> count(Aus1 %>% filter(month(Created.Date) == '9' & mday(Created.Date) == '11'))
    n
1 211
> head(Aus1 %>% filter(month(Created.Date) == '9' & mday(Created.Date) == '11')
+           %>% count(SR.Description)
+           %>% arrange(desc(n)))
                      SR.Description  n
1 Austin Code - Request Code Officer 42
2                     Tree Issue ROW 20
3                          Loose Dog 12
4  ARR Missed Yard Trimmings/Compost 10
5              Injured / Sick Animal 10
6         ARR Dead Animal Collection  9
> # There were 211 service requests in Austin, TX were initiated on my birthday.
> # The most common SR.Description is Austin Code - Request Code Officer.
```

6. a

```
# 6. a
# ZIP Code should be considered as a categorical variable. Because it has no
# numeric meaning, 78744 does not mean it has more data or larger than 78704.
```

6. b

```
> # 6. b
> nrow(Aus1 %>% filter(Method.Received =='Spot311 Interface'))
[1] 7641
> nrow(Aus1)
[1] 78289
> 7641/78289
[1] 0.09759992
> # 9.76% of all the 311 city service requests in my data frame came in through
> # the spot 311 interface.
```

6. c

```
> # 6. c
> nrow(Aus1 %>% filter(SR.Description =='Loose Dog'))
[1] 4531
> 4531/78289
[1] 0.05787531
> # 5.79% of all the 311 city service requests in my data frame were made because
> # of loose dogs.
```

6. d

```
> # 6. d
> unique(Aus1$Method.Received)
 [1] "Phone"             "Spot311 Interface"  "Mobile Device"      "Web"                "Mobile Created"
 [6] "E-Mail"            "Other"              "CSR - Follow On SR" "Field Request"      "Open311"
[11] "Mass Entry"        "Interface"          "External Interface"
> # There 13 unique types of methods Austin received.
```

7.

```
> # 7
> Aus1 = subset(Aus1, select = -c(Map.Page))
> View(Aus1)
> dim(Aus1)
[1] 78289    22
```

8.

```
# 8
Aus1 <- mutate(Aus1, Season = quarter(Created.Date, with_year = FALSE,
                                      fiscal_start = 10))

Aus1 <- Aus1 %>% mutate(Season = recode(Season, '1' = 'Winter',
                                                '2' = 'Spring',
                                                '3' = 'Summer',
                                                '4' = 'Fall' ))
```

9.

```
# 9
ggplot(Aus1, aes(x=Season)) + geom_bar(fill = 'pink', color = 'black')
# From the plot, we can tell the service is higher in Fall and Summer, lower in
# Spring and Winter. I think it is because the weather condition is bad in
# winters, people are staying home or service needed were covered by snow i.e.
# potholes, so the service requests are low. However, it does not mean the
# actual service needed are low. As a result, in the next Fall, people are
# seeing those potholes or some other services need to be done. The service
# requested are low in Spring is because many parts were being replaced in
# Fall i.e. signal lights. Since they are new, so they can run a while before
# service needed. That also explained why the service requested in Summer were
# high again.
```

10. a

```
> # 10. a
> head(Aus1 %>% count(SR.Description) %>% arrange(desc(n)))
                    SR.Description     n
1 Austin Code - Request Code Officer 14769
2                         Loose Dog  4531
3              ARR Missed Recycling  4140
4       Traffic Signal - Maintenance  4028
5         ARR Dead Animal Collection  3492
6  ARR Missed Yard Trimmings/Compost  3159
>
> Aus2 <- filter(Aus1, SR.Description == 'Austin Code - Request Code Officer'|
+                      SR.Description == 'Loose Dog'|
+                      SR.Description == 'ARR Missed Recycling'|
+                      SR.Description == 'Traffic Signal - Maintenance'|
+                      SR.Description == 'ARR Dead Animal Collection'|
+                      SR.Description == 'ARR Missed Yard Trimmings/Compost')
> View(Aus2)
> dim(Aus2)
[1] 34119    23
> nrow(Aus2)
[1] 34119
> # By filtering the data set with the six most common SR.Description types, my
> # data frame contains 34,119 rows.
```

10. b

```
> # 10. b
> ggplot(Aus2, aes(x = SR.Description)) + geom_bar(fill = 'pink',
+                                                  color = 'black') +
+   theme(axis.text.x = element_text(angle = 90))
```

10. c

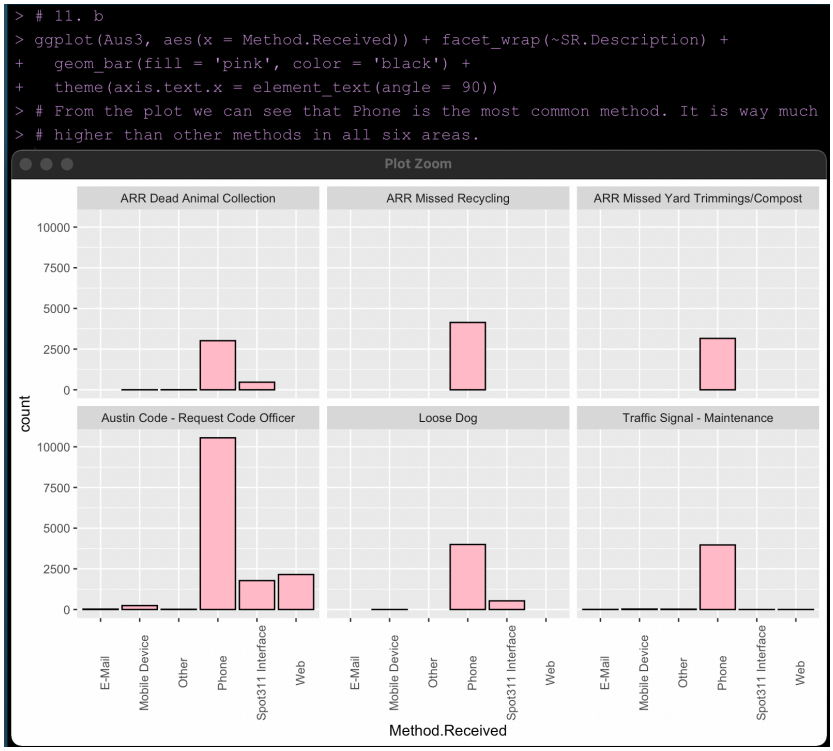
```
> # 10. c
> # From the plot we can know that the most highest service requested was
> # Austin Code - Request Code Officer (14,769). It is three times higher than
> # the second most service requested (Loose Dog, 4,531).
```

11. a

```
> # 11. a
> head(Aus2 %>% count(Method.Received) %>% arrange(desc(n)))
    Method.Received     n
1            Phone 28829
2 Spot311 Interface  2772
3              Web  2152
4     Mobile Device   272
5            Other    36
6           E-Mail    31
>
> Aus3 <- filter(Aus2, Method.Received == 'Phone'|
+                      Method.Received == 'Spot311 Interface'|
+                      Method.Received == 'Web'|
+                      Method.Received == 'Mobile Device'|
+                      Method.Received == 'Other'|
+                      Method.Received == 'E-Mail')
> dim(Aus3)
[1] 34092    23
> nrow(Aus3)
[1] 34092
> # By filtering the data set with the six most common Method.Received types, my
> # data frame contains 34,092 rows.
```

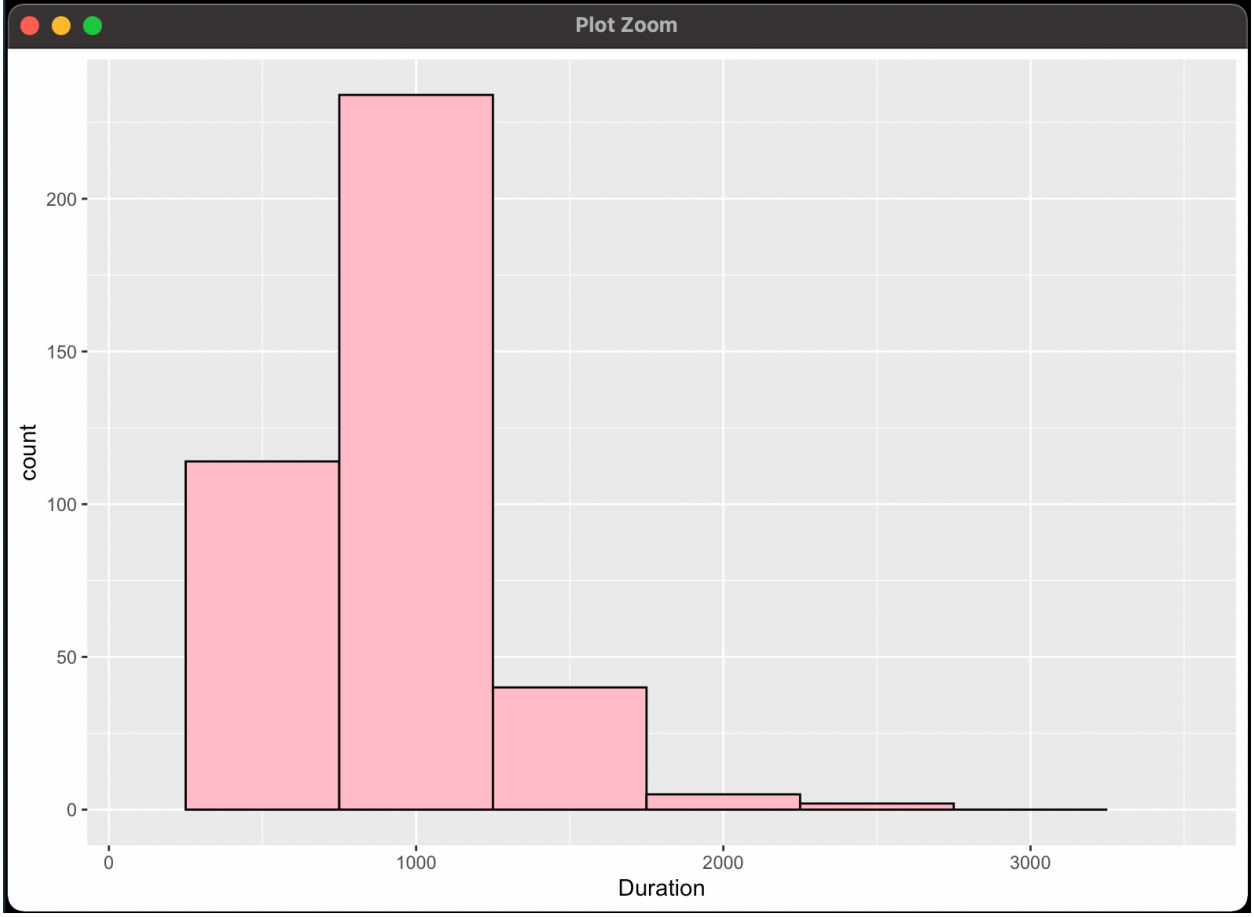
11. b

```
> # 11. b
> ggplot(Aus3, aes(x = Method.Received)) + facet_wrap(~SR.Description) +
+   geom_bar(fill = 'pink', color = 'black') +
+   theme(axis.text.x = element_text(angle = 90))
> # From the plot we can see that Phone is the most common method. It is way much
> # higher than other methods in all six areas.
```



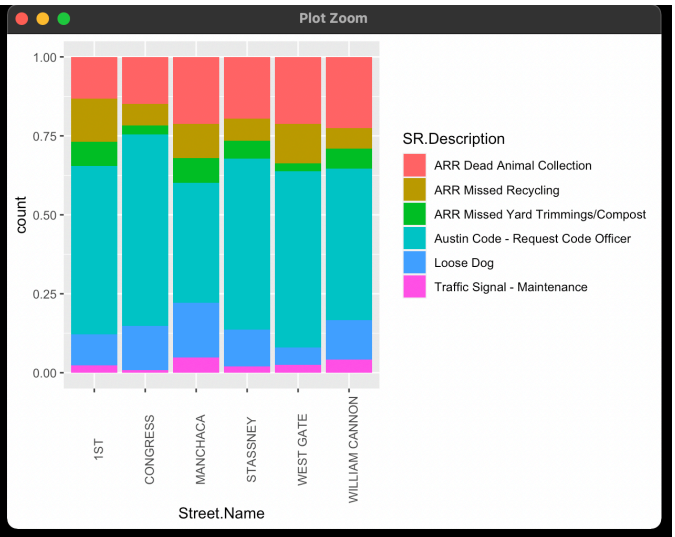
12.

```
> # 12
> ggplot(Aus3, aes(x=Duration)) +
+   geom_histogram(binwidth = 500, fill="pink", color="black") + xlim(100,3500)
Warning messages:
1: Removed 33483 rows containing non-finite values (stat_bin).
2: Removed 2 rows containing missing values (geom_bar).
> # From the plot we can tell the most of the service requested were done by
> # around 1000 days.
```



13. a

```
> # 13. a
> View(Aus3)
> head(Aus3 %>% count(Street.Name) %>% arrange(desc(n)))
     Street.Name    n
1 WILLIAM CANNON 1070
2       CONGRESS  796
3            1ST  757
4       MANCHACA  572
5      WEST GATE  439
6       STASSNEY  415
>
> Aus4 <- filter(Aus3, Street.Name == 'WILLIAM CANNON'|
+                      Street.Name == 'CONGRESS'|
+                      Street.Name == '1ST'|
+                      Street.Name == 'MANCHACA'|
+                      Street.Name == 'WEST GATE'|
+                      Street.Name == 'STASSNEY')
> dim(Aus4)
[1] 4049   23
> nrow(Aus4)
[1] 4049
>
> ggplot(Aus4, aes(x = Street.Name, fill = SR.Description)) +
+   geom_bar(position = 'fill') + theme(axis.text.x = element_text(angle = 90))
```



13. b

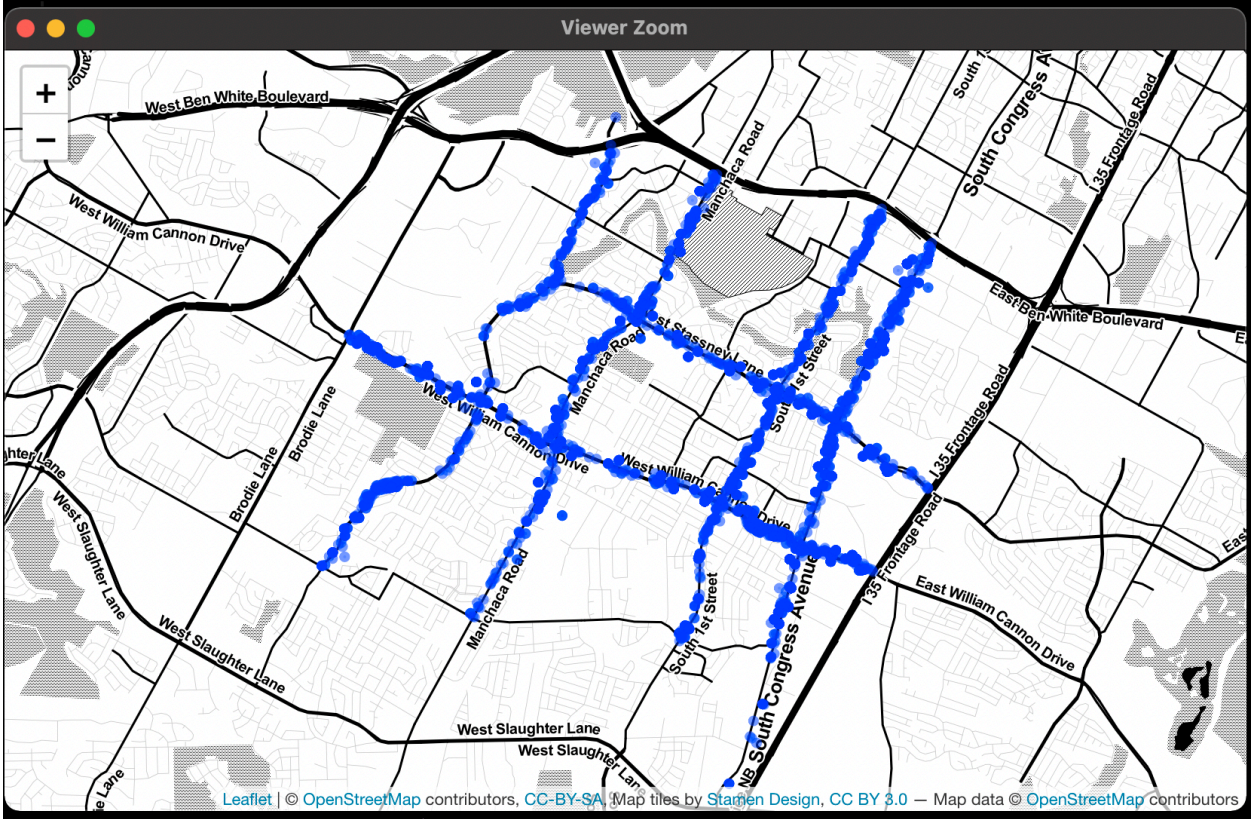
```
> # 13. b
> # From the plot, we can tell the light blue color has the most portion which is
> # the Austin Code - Request Code Officer. We can also tell that the William
> # Cannon street has the most dead animal collection, and the Congress street
> # has the best quality of traffic signal.
```

14.

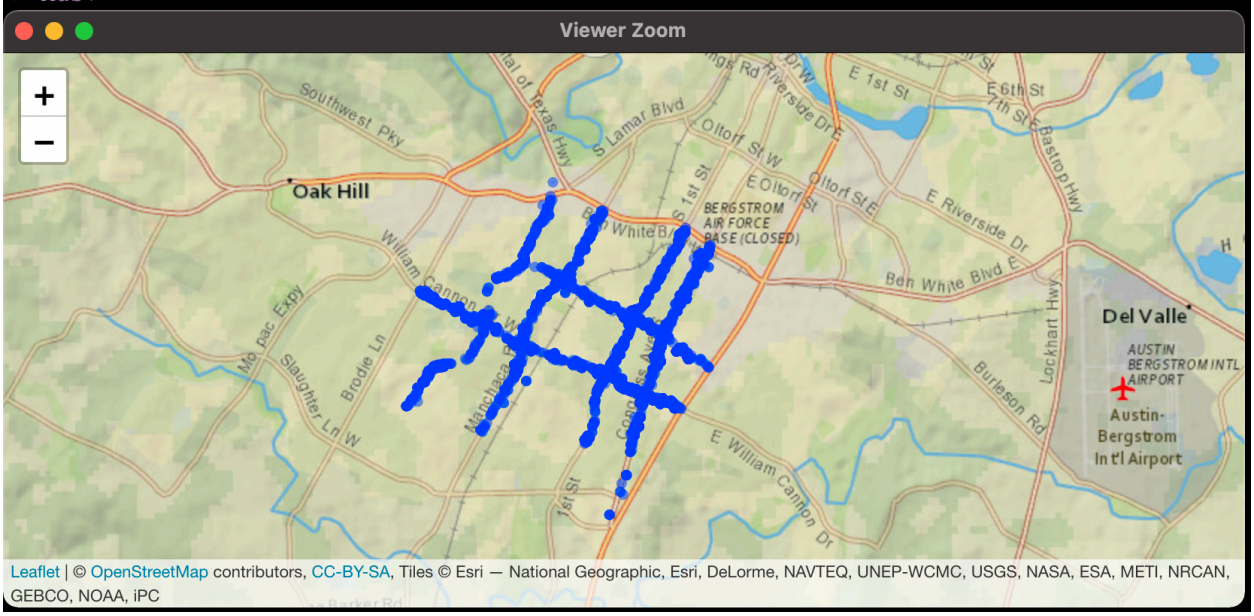
```
# 14
summary(Aus4)
Aus5 <- leaflet() %>% addTiles() %>%  addCircles(lng=Aus4$Longitude.Coordinate,
                                                 lat=Aus4$Latitude.Coordinate)
Aus5  # Print the map
```

15.

```
> Aus6 <- leaflet() %>% addTiles() %>%
+   addCircles(lng=Aus4$Longitude.Coordinate, lat=Aus4$Latitude.Coordinate) %>%
+   addProviderTiles(providers$Stamen.Toner)
> Aus6  # Print the map
```




```
> Aus7 <- leaflet() %>% addTiles() %>%
+   addCircles(lng=Aus4$Longitude.Coordinate, lat=Aus4$Latitude.Coordinate) %>%
+   addProviderTiles(providers$Esri.NatGeoWorldMap)
> Aus7
```



```
> Aus8 <- leaflet() %>% addTiles() %>%
+   addCircles(lng=Aus4$Longitude.Coordinate, lat=Aus4$Latitude.Coordinate) %>%
+   addProviderTiles(providers$Stamen.TonerLabels)
> Aus8
```