Assignment - 1

AD699 A3 Data Mining (Spring 2024)

Submitted by: Aishwarya Malhotra (U17161095)

Solution - 1

In [26]:
```
library(tidyverse)
library(dplyr)
library(lubridate)
install.packages("naniar")
library(naniar)
```

```
 There is a binary version available but the source version is later:
       binary source needs_compilation
naniar  0.6.0  1.0.0             FALSE
```

```
installing the source package 'naniar'
```

Solution - 2 a]

In [9]:
```
df <- read.csv("chicago_violations.csv")
```

In [10]:
```
str(df)
```

```
'data.frame':   784225 obs. of  22 variables:
 $ ID                    : Factor w/ 784225 levels "00000da0351f130050dcff1d86eb41d73066e930",..: 486179 560000 680109 782916 78
0628 604896 217983 442604 595273 293554 ...
 $ DOCKET.NUMBER         : Factor w/ 122012 levels "","08BN00001A",..: 118771 76470 117663 114704 118585 118771 117101 56196 118
780 118140 ...
 $ NOV.NUMBER            : Factor w/ 122054 levels ".14CO421880",..: 116777 79974 117840 114588 117379 116777 117006 51994 11798
0 116738 ...
 $ ADDRESS               : Factor w/ 80639 levels "0  UNKNOWN ST",..: 48848 32030 37675 41871 11803 48848 64819 65932 74492 4445
5 ...
 $ STREET.NUMBER         : int  5006 3241 3811 4259 1614 5006 6729 6926 8155 4520 ...
 $ STREET.DIRECTION      : Factor w/ 5 levels "","E","N","S",..: 5 5 5 5 5 5 4 3 4 4 ...
 $ STREET.NAME           : Factor w/ 1408 levels "`81ST","0GDEN",..: 623 117 116 1374 125 623 533 1354 762 470 ...
 $ STREET.TYPE           : Factor w/ 30 levels "","AVE","BLVD",..: 25 25 25 25 25 25 2 2 2 3 ...
 $ WARD                  : int  45 23 23 28 15 45 20 49 8 4 ...
 $ ISSUING.DEPARTMENT    : Factor w/ 1 level "Buildings": 1 1 1 1 1 1 1 1 1 1 ...
 $ HEARING.DATE          : Factor w/ 4046 levels "","01/02/2013",..: 500 2237 500 799 500 500 112 3611 500 500 ...
 $ CASE.DISPOSITION      : Factor w/ 9 levels "","Adjudication Performance / Other",..: 4 7 8 4 8 4 4 7 4 8 ...
 $ IMPOSED.FINE          : num  2500 300 0 500 0 2500 1000 200 500 0 ...
 $ ADMIN.COSTS           : int  40 75 0 40 0 40 40 75 40 0 ...
 $ LAST.MODIFIED.DATE    : Factor w/ 78645 levels "01/01/2017 01:17:19 PM",..: 8768 53125 8765 8775 8750 8768 8767 8776 8753 875
0 ...
 $ VIOLATION.DATE        : Factor w/ 43509 levels "","01/01/1991 09:00:00 AM",..: 41472 18723 27736 34106 40656 41472 23145 7221
40324 34108 ...
 $ VIOLATION.CODE        : Factor w/ 1198 levels "000001","0006014",..: 852 223 53 1068 541 426 336 1175 49 906 ...
 $ VIOLATION.DESCRIPTION: Factor w/ 1190 levels "- Every building or part thereof which is in an unsanitary condition by reason
of the basement or cellar being "| __truncated__,..: 725 645 241 437 352 907 1073 106 1151 802 ...
 $ RESPONDENTS           : Factor w/ 108348 levels "'D' BROTHERS INVESTMENTS INC C/O RICHARD ASH | THE BERNADETTE CORPORATION C/
O MICHAEL DIFOGGIO | TR# 10-2564 NO"| __truncated__,..: 25873 66707 90146 82549 25377 25873 77333 107459 43432 26626 ...
 $ LATITUDE              : num  42 41.8 41.8 41.9 41.8 ...
 $ LONGITUDE             : num  -87.8 -87.7 -87.7 -87.7 -87.7 ...
 $ LOCATION              : Factor w/ 80190 levels "","(41.64470194112152, -87.61515119956638)",..: 73713 23781 24362 44939 21124
73713 20327 79196 10681 32651 ...
```

Solution - 2 b]

> **💡 Interpretation of Code 💡 -**
>
> Calling the 'str()' function would display the structure of the dataset, including information about the variables (columns) and observations (rows).
>
> It provides a concise summary of the structure of the object.
>
> It displays the data type and first few values of each variable.

> From the provided output of 'str(cv)' we can see that dataframe contains 784225 observations and 22 varaibles.

Solution - 3

In [11]:
```
## Filter the data with Ward 13 records
assigned_ward <- 13
df2 <- df[df$WARD == assigned_ward,]
str(df2)
```

```
'data.frame':   8234 obs. of  22 variables:
 $ ID                  : Factor w/ 784225 levels "00000da0351f130050dcff1d86eb41d73066e930",..: NA NA NA NA NA NA NA 595590 27
12 NA ...
 $ DOCKET.NUMBER       : Factor w/ 122012 levels "","08BN00001A",..: NA NA NA NA NA NA NA 86124 86124 NA ...
 $ NOV.NUMBER          : Factor w/ 122054 levels ".14CO421880",..: NA NA NA NA NA NA NA 83174 83174 NA ...
 $ ADDRESS             : Factor w/ 80639 levels "0  UNKNOWN ST",..: NA NA NA NA NA NA NA 42165 42165 NA ...
 $ STREET.NUMBER       : int  NA NA NA NA NA NA NA 4311 4311 NA ...
 $ STREET.DIRECTION    : Factor w/ 5 levels "","E","N","S",..: NA NA NA NA NA NA NA 5 5 NA ...
 $ STREET.NAME         : Factor w/ 1408 levels "`81ST","0GDEN",..: NA NA NA NA NA NA NA 929 929 NA ...
 $ STREET.TYPE         : Factor w/ 30 levels "","AVE","BLVD",..: NA NA NA NA NA NA NA 22 22 NA ...
 $ WARD                : int  NA NA NA NA NA NA NA 13 13 NA ...
 $ ISSUING.DEPARTMENT  : Factor w/ 1 level "Buildings": NA NA NA NA NA NA NA 1 1 NA ...
 $ HEARING.DATE        : Factor w/ 4046 levels "","01/02/2013",..: NA NA NA NA NA NA NA 518 518 NA ...
 $ CASE.DISPOSITION    : Factor w/ 9 levels "","Adjudication Performance / Other",..: NA NA NA NA NA NA NA 4 4 NA ...
 $ IMPOSED.FINE        : num  NA NA NA NA NA NA NA 1000 1000 NA ...
 $ ADMIN.COSTS         : int  NA NA NA NA NA NA NA 40 40 NA ...
 $ LAST.MODIFIED.DATE  : Factor w/ 78645 levels "01/01/2017 01:17:19 PM",..: NA NA NA NA NA NA NA 10125 10125 NA ...
 $ VIOLATION.DATE      : Factor w/ 43509 levels "","01/01/1991 09:00:00 AM",..: NA NA NA NA NA NA NA 21101 21101 NA ...
 $ VIOLATION.CODE      : Factor w/ 1198 levels "000001","0006014",..: NA NA NA NA NA NA NA 656 720 NA ...
 $ VIOLATION.DESCRIPTION: Factor w/ 1190 levels "- Every building or part thereof which is in an unsanitary condition by reason
of the basement or cellar being "| __truncated__,..: NA NA NA NA NA NA NA 1016 1176 NA ...
 $ RESPONDENTS         : Factor w/ 108348 levels "'D' BROTHERS INVESTMENTS INC C/O RICHARD ASH | THE BERNADETTE CORPORATION C/
O MICHAEL DIFOGGIO | TR# 10-2564 NO"| __truncated__,..: NA NA NA NA NA NA NA 69010 69010 NA ...
 $ LATITUDE            : num  NA NA NA NA NA ...
 $ LONGITUDE           : num  NA NA NA NA NA ...
 $ LOCATION            : Factor w/ 80190 levels "","(41.64470194112152, -87.61515119956638)",..: NA NA NA NA NA NA NA 19896 19
896 NA ...
```

> 💡 **Interpretation of Results** 💡 -
>
> After filtering the dataframe for Ward = 13, resulting dataset contains 8234 observations and 22 variables. The three interesting facts about Ward 13 in Chicago are -
>
> Ward 13, also known as the 'Near West Side', is home to the historic University of Illinois at Chicago (UIC) campus, which plays a significant role in shaping the cultural and educational landscape of the area.
>
> Ward 13 is known for its diverse population and vibrant neighborhoods, including the bustling Greek Town, Little Italy, and the developing West Loop area, which has seen rapid growth in recent years with the emergence of trendy restaurants, art galleries, and tech startups.
>
> Ward 13 is also home to important cultural landmarks such as the United Center, where the Chicago Bulls and Chicago Blasckhawks play, and the historic Maxwell Street Market, known for its rich history as a vibrant marketplace for food, music, and commerce.

Solution - 4 a]

In [13]:
```
## Dealing with NA data
sum(is.na(df2))
```

61822

> 💡 **Interpretation of Results** 💡 -
>
> Yes, there are NA values in my data df2. I checked this by "View(df2)" function in the RStudio. From the output that was delivered we can see that, there are 61822 NAs in the dataframe.

Solution - 4 b]

In [14]:
```
complete_case <- sum(complete.cases(df2))
total_rows <- nrow(df2)
percentage_complete <- (complete_case / total_rows) * 100
percentage_complete
```

65.8610638814671

> 💡 **Interpretation of Results** 💡 -

> 65.86106% of data is complete case. A complete case in a data frame refers to a row that contains no missing values across all its variables. In other words, a complete case is a row where every variable has a non-missing (non-NA) value.

Solution - 4 c]

In [16]:
```
#Converting any blank cells in the data frame into NA
df2[df2 == ""] <- NA
```

Solution - 4 d]

In [17]:
```
sum(is.na(df2))
```

61946

> 💡**Interpretation of Results**💡 -
>
> The total number of NA values in my dataset are 61946

Solution - 4 e]

In [18]:
```
complete_case2 <- sum(complete.cases(df2))
total_rows2 <- nrow(df2)
percentage_complete2 <- (complete_case2 / total_rows2) * 100
percentage_complete2
```

64.3672577119262

> 💡**Interpretation of Results**💡 -
>
> 64.37% percentage of rows in the dataframe are complete cases

Solution - 4 f]

In [27]:
```
# Generate a table that shows the number of missing values and the percentage of missing values for each variable.
missing_summary <- miss_var_summary(df2)
print(missing_summary)
```

```
# A tibble: 22 x 3
   variable         n_miss pct_miss
   <chr>             <int>    <dbl>
 1 CASE.DISPOSITION   2911     35.4
 2 STREET.TYPE        2830     34.4
 3 VIOLATION.DATE     2812     34.2
 4 LATITUDE           2811     34.1
 5 LONGITUDE          2811     34.1
 6 LOCATION           2811     34.1
 7 ID                 2810     34.1
 8 DOCKET.NUMBER      2810     34.1
 9 NOV.NUMBER         2810     34.1
10 ADDRESS            2810     34.1
# ... with 12 more rows
```

Solution - 5 a]

In [33]:
```
### We have HEARING.DATE, LAST.MODIFIED.DATE & VIOLATION.DATE as date variables. They are currently seen in chr - text format.
str(df2)
```

```
'data.frame':   8234 obs. of  22 variables:
 $ ID                  : Factor w/ 784225 levels "00000da0351f130050dcff1d86eb41d73066e930",..: NA NA NA NA NA NA NA 595590 27
12 NA ...
 $ DOCKET.NUMBER       : Factor w/ 122012 levels "","08BN00001A",..: NA NA NA NA NA NA NA 86124 86124 NA ...
 $ NOV.NUMBER          : Factor w/ 122054 levels ".14CO421880",..: NA NA NA NA NA NA NA 83174 83174 NA ...
 $ ADDRESS             : Factor w/ 80639 levels "0  UNKNOWN ST",..: NA NA NA NA NA NA NA 42165 42165 NA ...
 $ STREET.NUMBER       : int  NA NA NA NA NA NA NA 4311 4311 NA ...
 $ STREET.DIRECTION    : Factor w/ 5 levels "","E","N","S",..: NA NA NA NA NA NA NA 5 5 NA ...
 $ STREET.NAME         : Factor w/ 1408 levels "`81ST","0GDEN",..: NA NA NA NA NA NA NA 929 929 NA ...
 $ STREET.TYPE         : Factor w/ 30 levels "","AVE","BLVD",..: NA NA NA NA NA NA NA 22 22 NA ...
 $ WARD                : int  NA NA NA NA NA NA NA 13 13 NA ...
 $ ISSUING.DEPARTMENT  : Factor w/ 1 level "Buildings": NA NA NA NA NA NA NA 1 1 NA ...
 $ HEARING.DATE        : Date, format: NA NA ...
 $ CASE.DISPOSITION    : Factor w/ 9 levels "","Adjudication Performance / Other",..: NA NA NA NA NA NA NA 4 4 NA ...
 $ IMPOSED.FINE        : num  NA NA NA NA NA NA NA 1000 1000 NA ...
 $ ADMIN.COSTS         : int  NA NA NA NA NA NA NA 40 40 NA ...
 $ LAST.MODIFIED.DATE  : POSIXct, format: NA NA ...
 $ VIOLATION.DATE      : Date, format: NA NA ...
 $ VIOLATION.CODE      : Factor w/ 1198 levels "000001","0006014",..: NA NA NA NA NA NA NA 656 720 NA ...
 $ VIOLATION.DESCRIPTION: Factor w/ 1190 levels "- Every building or part thereof which is in an unsanitary condition by reason
of the basement or cellar being "| __truncated__,..: NA NA NA NA NA NA NA 1016 1176 NA ...
 $ RESPONDENTS         : Factor w/ 108348 levels "'D' BROTHERS INVESTMENTS INC C/O RICHARD ASH | THE BERNADETTE CORPORATION C/
O MICHAEL DIFOGGIO | TR# 10-2564 NO"| __truncated__,..: NA NA NA NA NA NA NA 69010 69010 NA ...
 $ LATITUDE            : num  NA NA NA NA NA ...
 $ LONGITUDE           : num  NA NA NA NA NA ...
 $ LOCATION            : Factor w/ 80190 levels "","(41.64470194112152, -87.61515119956638)",..: NA NA NA NA NA NA NA 19896 19
896 NA ...
```

Solution - 5 b]

In [44]:
```r
df2$HEARING.DATE <- as.Date(df2$HEARING.DATE, format = "%Y/%m/%d")
str(df2)
df2$LAST.MODIFIED.DATE <- as.Date(df2$LAST.MODIFIED.DATE, format = "%Y/%m/%d")
df2$VIOLATION.DATE <- as.Date(df2$VIOLATION.DATE, format = "%Y/%m/%d")

# Sucessfully converted all the date data format
```


```
'data.frame':   8234 obs. of  22 variables:
 $ ID                  : Factor w/ 784225 levels "00000da0351f130050dcff1d86eb41d73066e930",..: NA NA NA NA NA NA NA 595590 27
12 NA ...
 $ DOCKET.NUMBER       : Factor w/ 122012 levels "","08BN00001A",..: NA NA NA NA NA NA NA 86124 86124 NA ...
 $ NOV.NUMBER          : Factor w/ 122054 levels ".14CO421880",..: NA NA NA NA NA NA NA 83174 83174 NA ...
 $ ADDRESS             : Factor w/ 80639 levels "0  UNKNOWN ST",..: NA NA NA NA NA NA NA 42165 42165 NA ...
 $ STREET.NUMBER       : int  NA NA NA NA NA NA NA 4311 4311 NA ...
 $ STREET.DIRECTION    : Factor w/ 5 levels "","E","N","S",..: NA NA NA NA NA NA NA 5 5 NA ...
 $ STREET.NAME         : Factor w/ 1408 levels "`81ST","0GDEN",..: NA NA NA NA NA NA NA 929 929 NA ...
 $ STREET.TYPE         : Factor w/ 30 levels "","AVE","BLVD",..: NA NA NA NA NA NA NA 22 22 NA ...
 $ WARD                : int  NA NA NA NA NA NA NA 13 13 NA ...
 $ ISSUING.DEPARTMENT  : Factor w/ 1 level "Buildings": NA NA NA NA NA NA NA 1 1 NA ...
 $ HEARING.DATE        : Date, format: NA NA ...
 $ CASE.DISPOSITION    : Factor w/ 9 levels "","Adjudication Performance / Other",..: NA NA NA NA NA NA NA 4 4 NA ...
 $ IMPOSED.FINE        : num  NA NA NA NA NA NA NA 1000 1000 NA ...
 $ ADMIN.COSTS         : int  NA NA NA NA NA NA NA 40 40 NA ...
 $ LAST.MODIFIED.DATE  : Date, format: NA NA ...
 $ VIOLATION.DATE      : Date, format: NA NA ...
 $ VIOLATION.CODE      : Factor w/ 1198 levels "000001","0006014",..: NA NA NA NA NA NA NA 656 720 NA ...
 $ VIOLATION.DESCRIPTION: Factor w/ 1190 levels "- Every building or part thereof which is in an unsanitary condition by reason
of the basement or cellar being "| __truncated__,..: NA NA NA NA NA NA NA 1016 1176 NA ...
 $ RESPONDENTS         : Factor w/ 108348 levels "'D' BROTHERS INVESTMENTS INC C/O RICHARD ASH | THE BERNADETTE CORPORATION C/
O MICHAEL DIFOGGIO | TR# 10-2564 NO"| __truncated__,..: NA NA NA NA NA NA NA 69010 69010 NA ...
 $ LATITUDE            : num  NA NA NA NA NA ...
 $ LONGITUDE           : num  NA NA NA NA NA ...
 $ LOCATION            : Factor w/ 80190 levels "","(41.64470194112152, -87.61515119956638)",..: NA NA NA NA NA NA NA 19896 19
896 NA ...
```


Solution - 5 c]

In [46]:
```r
# Added CityDelay as a variable
df3 <- mutate(df2, CityDelay = HEARING.DATE - VIOLATION.DATE)
```

Solution - 5 d]

In [48]:
```r
birthday_violations <- filter(df3, format(HEARING.DATE, "%m-%d") == "02-25")
violation_count <- nrow(birthday_violations)
most_common_disposition <- birthday_violations %>%
  count(CASE.DISPOSITION) %>%
  arrange(desc(n)) %>%
  slice(1) %>%
  pull(CASE.DISPOSITION)
violation_count
most_common_disposition
```

32
Non-Suit
▶ **Levels**:

> 💡**Interpretation of Results**💡 -
>
> Ordinance violations were issued on my birthday 32; Most common Case Disposition for those ordinance violations = "Non-Suit"

Solution - 6 a]

Ward represent different sections of Chicago, which suggests that it is categorical rather than numeric. Each ward is a distinct category or group, and there is no inherent order or numerical relationship between the wards.

Solution - 6 b]

In [49]:
```
df4 <- na.omit(df3[, c("IMPOSED.FINE", "ADMIN.COSTS")])

correlation <- cor(df4$IMPOSED.FINE, df4$ADMIN.COSTS)
print(correlation)
```

```
[1] 0.1589375
```

> 💡**Interpretation of Results**💡 -
>
> Correlation of 0.1589375 suggests a weak positive correlation between "imposed fine" & "admin costs". This means that there is some tendency for them to increase together, but the relationship is not strong. Other factos may have a more significant influence on admin costs than imposed fines alone.

Solution - 6 c]

In [50]:
```
street_type <- df3 %>%
count(STREET.TYPE)
street_type

most_common_street_type <- street_type %>%
  arrange(desc(n)) %>%
  slice(1) %>%
  pull(STREET.TYPE)
print(paste("The most common street type in Ward 13 is:", most_common_street_type))
```

```
Warning message:
"Factor `STREET.TYPE` contains implicit NA, consider using `forcats::fct_explicit_na`"
```

STREET.TYPE	n
AVE	3250
PL	753
RD	189
ST	1212
NA	2830

```
[1] "The most common street type in Ward 13 is: AVE"
```

> 💡**Interpretation of Results**💡 -
>
> The most common street type is Ave. Yes, it is the same street type as the one that I live on.

Solution - 6 d]

In [51]:
```
vc <- length(unique(df3$VIOLATION.CODE))
vd <- length(unique(df3$VIOLATION.DESCRIPTION))
print(paste("Number of unique Violation Code values are:", vc))
print(paste("Number of unique violation description are:", vd))
```

```
[1] "Number of unique Violation Code values are: 341"
[1] "Number of unique violation description are: 339"
```

Solution - 7 a]

In [52]:
```
df3$Year <- lubridate::year(df3$HEARING.DATE)
average_imposed_fine <- df3 %>%
```

```
  filter(IMPOSED.FINE > 0) %>% #Filter out cases where the fine is greater than zero
  group_by(Year) %>%
  summarise(average_imposed_fine = mean(IMPOSED.FINE, na.rm = TRUE))
average_imposed_fine
```

Year	average_imposed_fine
2008	2219.2529
2009	1312.6402
2010	746.6667
2011	3939.6176
2012	1239.3720
2013	1298.2843
2014	507.8431
2015	1989.6907
2016	872.6415
2017	488.7850
2018	9182.3529
2019	3992.8571
2020	1633.3333
2021	1532.2581
2022	1482.7586
2023	1685.7143
2024	876.4706

> 💡 **Interpretation of Results** 💡 -
>
> One reason for numbers of 2024 is lower than other is because we only have 2 month data available with us.

Solution - 8

In [53]:
```
# Remove ID and Docket Number

df5 <- df3[,-c(1,2)]
```

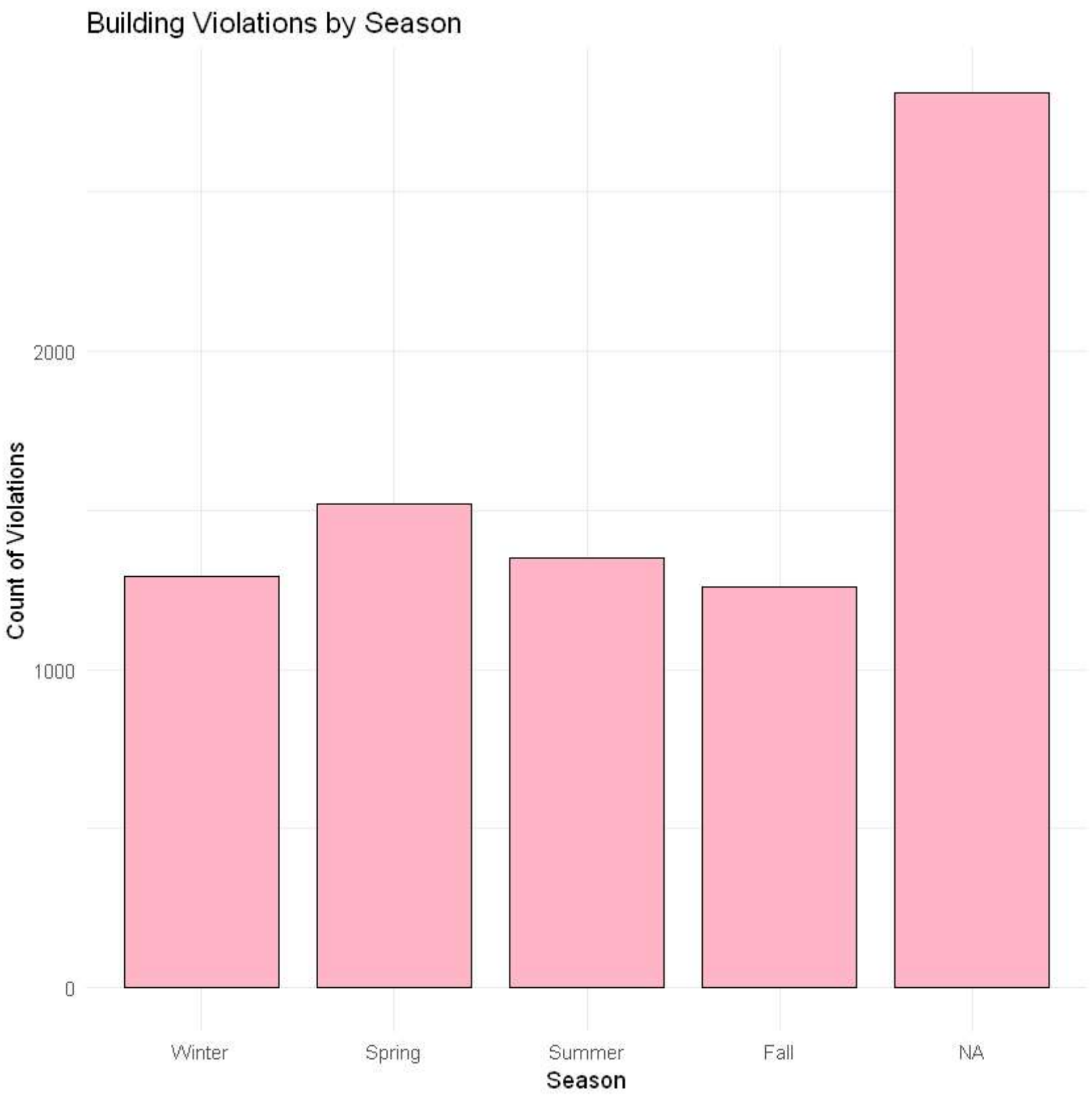
Solution - 9

In [54]:
```
# Create a new column called season based on the quarter of Violation Date
df5$season <- quarter(df5$VIOLATION.DATE)
df5$season <- factor(df5$season, levels = 1:4, labels = c("Winter", "Spring", "Summer", "Fall"))
head(df5)
```

NOV.NUMBER	ADDRESS	STREET.NUMBER	STREET.DIRECTION	STREET.NAME	STREET.TYPE	WARD	ISSUING.DEPARTMENT	HEARING.DATE
NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA

Solution - 10

In [55]:
```
library(ggplot2)

ggplot(df5, aes(x = season)) +
  geom_bar(fill = "pink1", color = "black", width = 0.8) +
  labs(x = "Season", y = "Count of Violations", title = "Building Violations by Season") +
  theme_minimal()
```



> 💡 **Interpretation of Results** 💡 -
>
> Spring has the highest count of Violations that maybe due to increase in constructional activities during that season.

Solution - 11 a]

In [57]:
```
## Step: 1 - Calculate the frequency of each Case Disposition
case_disposition_freq <- df5 %>%
  count(CASE.DISPOSITION)
## Step: 2 - Select the top 5 most common Case Dispositions
top_5_disposition <- case_disposition_freq %>%
  arrange(desc(n)) %>%
  slice(1:5) %>%
  pull(CASE.DISPOSITION)

## Step: 3 Filter the dataset to keep only rows with the top 5 Case Disposition
df5_filtered <- df5 %>%
  filter(CASE.DISPOSITION %in% top_5_disposition)

## Step: 4
str(df5_filtered)
### a] The dataframe has 8041 rows.
```

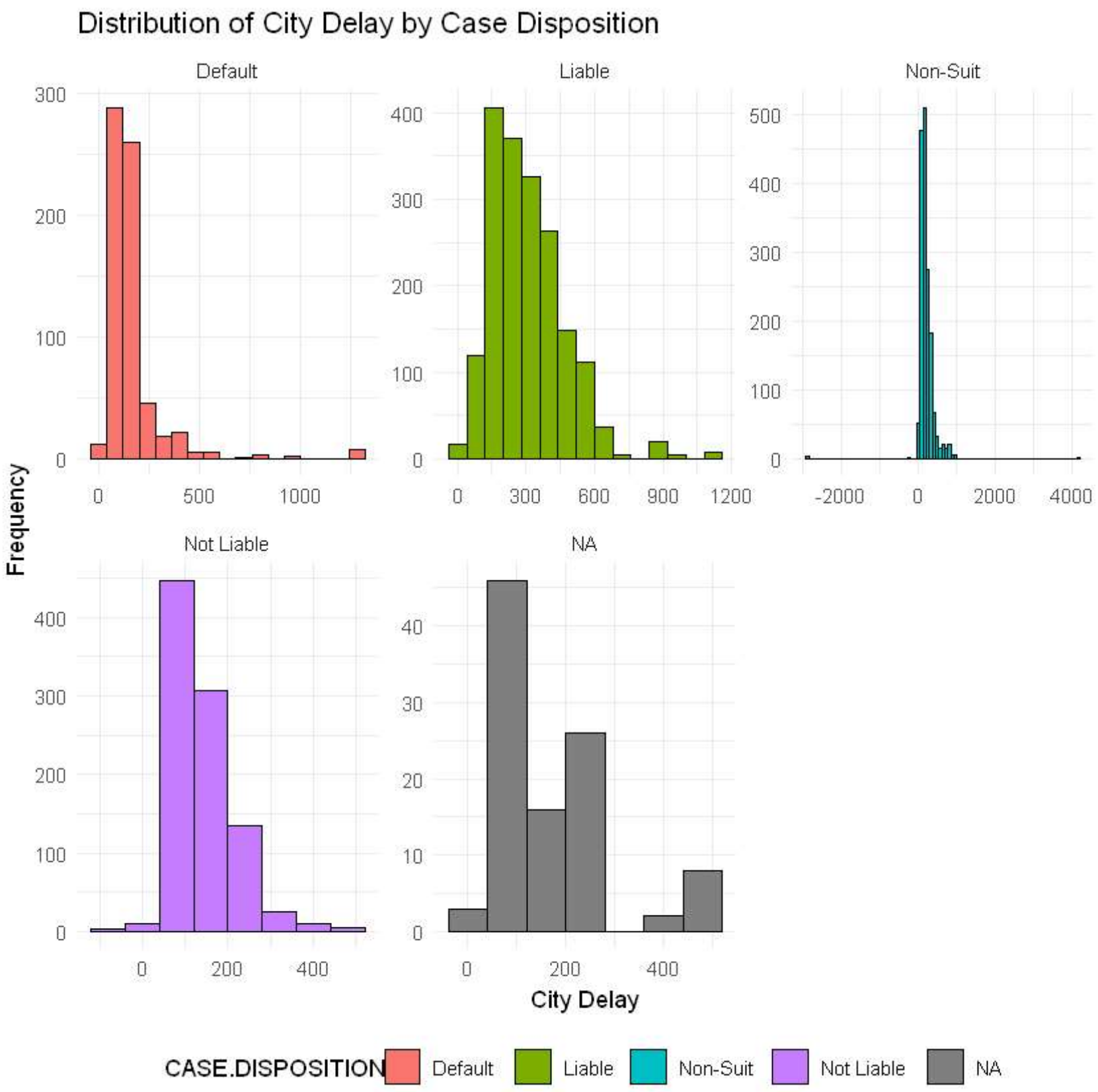
```
Warning message:
"Factor `CASE.DISPOSITION` contains implicit NA, consider using `forcats::fct_explicit_na`"
```

```
'data.frame':   8041 obs. of  23 variables:
 $ NOV.NUMBER           : Factor w/ 122054 levels ".14CO421880",..: NA NA NA NA NA NA NA 83174 83174 NA ...
 $ ADDRESS              : Factor w/ 80639 levels "0  UNKNOWN ST",..: NA NA NA NA NA NA NA 42165 42165 NA ...
 $ STREET.NUMBER        : int  NA NA NA NA NA NA NA 4311 4311 NA ...
 $ STREET.DIRECTION     : Factor w/ 5 levels "","E","N","S",..: NA NA NA NA NA NA NA 5 5 NA ...
 $ STREET.NAME          : Factor w/ 1408 levels "`81ST","0GDEN",..: NA NA NA NA NA NA NA 929 929 NA ...
 $ STREET.TYPE          : Factor w/ 30 levels "","AVE","BLVD",..: NA NA NA NA NA NA NA 22 22 NA ...
 $ WARD                 : int  NA NA NA NA NA NA NA 13 13 NA ...
 $ ISSUING.DEPARTMENT   : Factor w/ 1 level "Buildings": NA NA NA NA NA NA NA 1 1 NA ...
 $ HEARING.DATE         : Date, format: NA NA ...
 $ CASE.DISPOSITION     : Factor w/ 9 levels "","Adjudication Performance / Other",..: NA NA NA NA NA NA NA 4 4 NA ...
 $ IMPOSED.FINE         : num  NA NA NA NA NA NA NA 1000 1000 NA ...
 $ ADMIN.COSTS          : int  NA NA NA NA NA NA NA 40 40 NA ...
 $ LAST.MODIFIED.DATE   : Date, format: NA NA ...
 $ VIOLATION.DATE       : Date, format: NA NA ...
 $ VIOLATION.CODE       : Factor w/ 1198 levels "000001","0006014",..: NA NA NA NA NA NA NA 656 720 NA ...
 $ VIOLATION.DESCRIPTION: Factor w/ 1190 levels "- Every building or part thereof which is in an unsanitary condition by reason
of the basement or cellar being "| __truncated__,..: NA NA NA NA NA NA NA 1016 1176 NA ...
 $ RESPONDENTS          : Factor w/ 108348 levels "'D' BROTHERS INVESTMENTS INC C/O RICHARD ASH | THE BERNADETTE CORPORATION C/
O MICHAEL DIFOGGIO | TR# 10-2564 NO"| __truncated__,..: NA NA NA NA NA NA NA 69010 69010 NA ...
 $ LATITUDE             : num  NA NA NA NA NA ...
 $ LONGITUDE            : num  NA NA NA NA NA ...
 $ LOCATION             : Factor w/ 80190 levels "","(41.64470194112152, -87.61515119956638)",..: NA NA NA NA NA NA NA 19896 19
896 NA ...
 $ CityDelay            : 'difftime' num  NA NA NA NA ...
  ..- attr(*, "units")= chr "days"
 $ Year                 : num  NA NA NA NA NA ...
 $ season               : Factor w/ 4 levels "Winter","Spring",..: NA NA NA NA NA NA NA 2 2 NA ...
```

Solution - 11 b]

In [59]:
```
ggplot(data = df5_filtered, aes(x= CityDelay, fill = CASE.DISPOSITION)) +
  geom_histogram(binwidth = 80, color = "black") +
  facet_wrap(~ CASE.DISPOSITION, scales = "free") +
  labs(x = 'City Delay', y = "Frequency", title = "Distribution of City Delay by Case Disposition") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
Warning message:
"Removed 2812 rows containing non-finite values (stat_bin)."
```



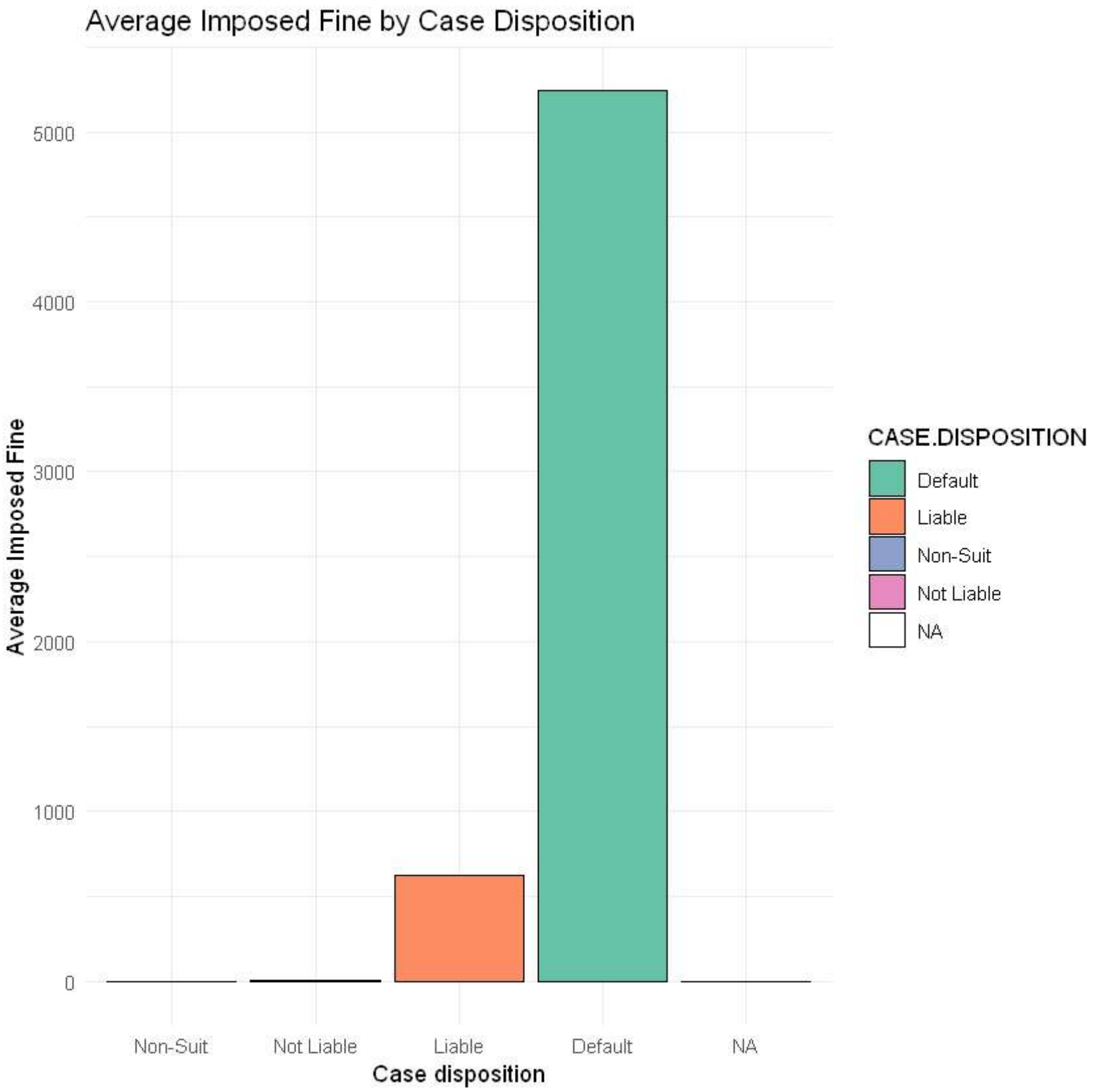
> 💡 **Interpretation of Graphs** 💡 -
>
> In 'Default' CASE.DISTRIBUTION most city delays fall within the range of approximately 0 to 500. This suggests that for cases with a 'Default' disposition, the time between the violation data and the hearing date tends to be relatively short, with most cases being resolved or addressed within this timeframe.
>
> 'Liable' CASE.DISTRIBUTION is bell-shaped with the peak occuring around a city delay of approximately 600. This indicates that for cases where the disposition is 'Liable', there is a broader range of city delays, with a signifuicant number of cases having longer delays before the hearing date compared to other dispositions.
>
> 'Non-Suit' CASE.DISTRIBUTION exhibit unusual pattern. Most frequencies are extreme negative values, which could imply an outlier.
>
> 'Not Liable' CASE.DISTRIBUTION, frequencies are concentrated between 0 and about 400 with a peak around 200. This indicates that for cases with a 'Not Liable' disposition, the city delays are generally shorter compared to 'Liable' cases, with a peak occurring around a delay of 200. This suggests that these cases tend to be resolved or adderessed relatively quickly.

Solution - 12

In [60]:
```
#Calculate average imposed fine for each type of case disposition
average_imposed_fine <- df5_filtered %>%
  group_by(CASE.DISPOSITION) %>%
  summarise(average_imposed_fine = mean(IMPOSED.FINE, na.rm = TRUE)) %>%
  arrange(average_imposed_fine)
```

```
Warning message:
"Factor `CASE.DISPOSITION` contains implicit NA, consider using `forcats::fct_explicit_na`"
```

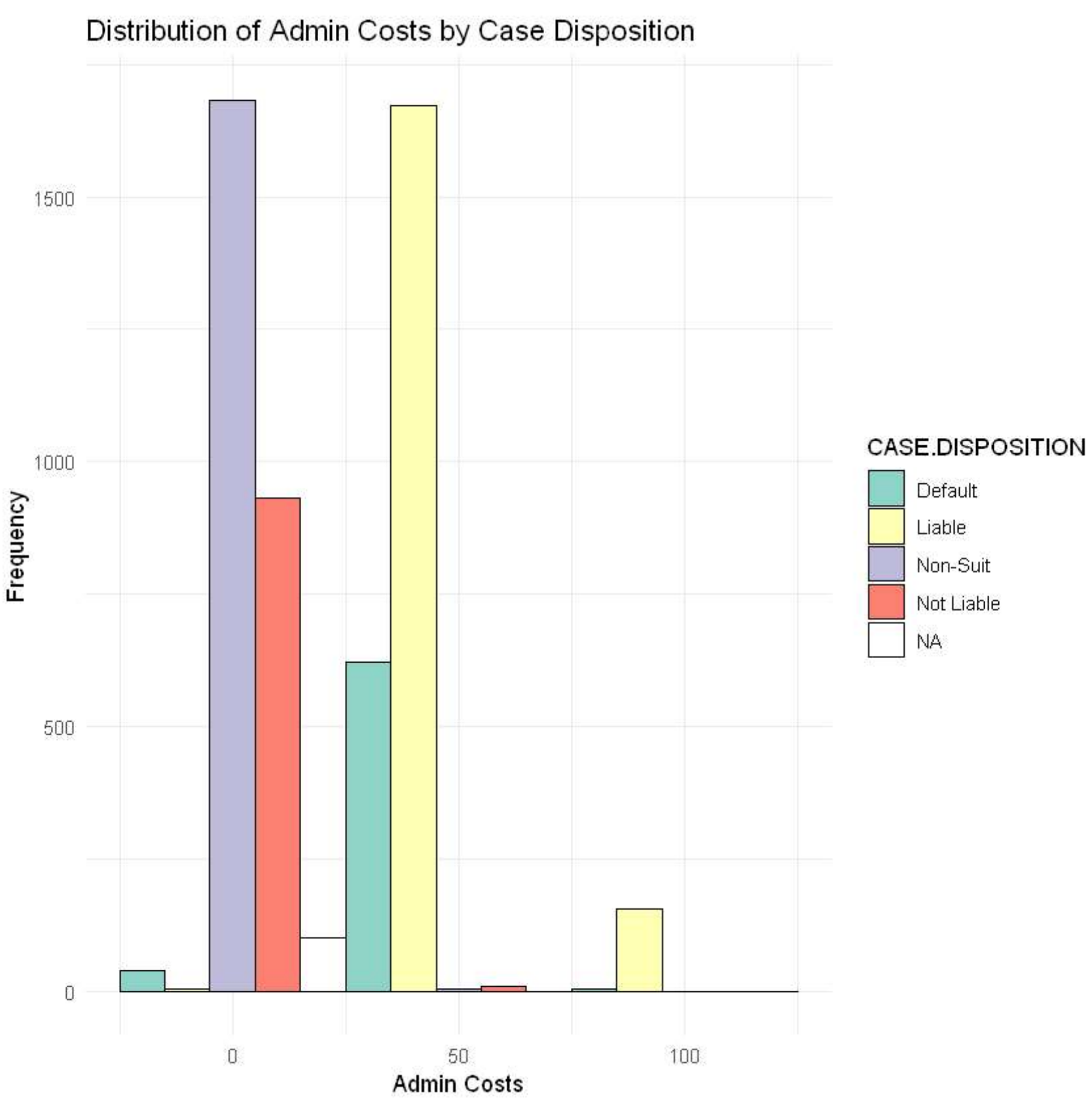
In [61]:
```
# Barplot
ggplot(average_imposed_fine, aes(x = reorder(CASE.DISPOSITION, average_imposed_fine), y = average_imposed_fine, fill = CASE.DI
  geom_bar(stat = 'identity', color = 'black') +
  labs(x = "Case disposition", y = "Average Imposed Fine", title = "Average Imposed Fine by Case Disposition") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



Solution - 13

In [62]:
```
# Histogram
ggplot(df5_filtered, aes(x = ADMIN.COSTS, fill = CASE.DISPOSITION)) +
  geom_histogram(binwidth = 50, color = "black", position = "dodge") +
  labs(x = "Admin Costs", y = "Frequency", title = "Distribution of Admin Costs by Case Disposition") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```

```
Warning message:
"Removed 2810 rows containing non-finite values (stat_bin)."
```



Solution - 14

In [96]:
```
library(dplyr)
library(forcats)

# Count occurrences of each Violation Description and identify the top 5
top_5_violation_desc <- df5_filtered %>%
  count(VIOLATION.DESCRIPTION) %>%
  arrange(desc(n)) %>%
  head(5) %>%
  pull(VIOLATION.DESCRIPTION)

# Filter the dataset to keep only rows with the top 5 Violation Descriptions
df7 <- df5_filtered %>%
  filter(VIOLATION.DESCRIPTION %in% top_5_violation_desc)

# Convert VIOLATION.DESCRIPTION to factor with explicit NA levels
df7$VIOLATION.DESCRIPTION <- fct_explicit_na(df7$VIOLATION.DESCRIPTION)

# Print out the unique values for Violation Description
unique_violation_desc <- unique(df7$VIOLATION.DESCRIPTION)
print(unique_violation_desc)
```

```
Warning message:
"Factor `VIOLATION.DESCRIPTION` contains implicit NA, consider using `forcats::fct_explicit_na`"
```

```
[1] (Missing)
[2] Description of work:
[3] Submit plans prepared, signed, and sealed by a licensed architect or registered structural engineer for approval and obtain
permit. (13-32-010, 13-32-040, 13-40-010, 13-40-020)
[4] Arrange for inspection of premises. (13-12-100)
[5] Obtain permit before performing work. (13-32-010)
1191 Levels: - Every building or part thereof which is in an unsanitary condition by reason of the basement or cellar being cov
ered with stagnant water, or by reason of the presence of sewer gas, or by reason of any portion of a building being infected w
ith disease or being unfit for human habitation, or which by reason of any other unsanitary condition, is a source of sickness,
or which endangers the public health, is hereby declared to be a public nuisance. (7-28-060, 18-29-102.3) ...
```

> 💡 **Interpretation of Code** 💡 -
>
> In the enviornment of R programming language, the code aims to filter the dataset `df5_filtered` to retain only rows with the top 5 most common Violation Descriptionss.
>
> We first pull up the library `dplyr` & `forcats`. Then we count the occurance of each Violation Description in the dataset and identify the top 5 most common description using `count`, `arrange`, `head` and `pull` functions.
>
> Next, we filter the dataset `df5_filtered` to keep only rows with top 5 Violation Descriptions using `filter` functionn from `dplyr`. After filtering, we attempt to convert the `VIOLATION.DESCRIPTION` coloumn into a factor with explicit NA using `forcats::fct_explicit_na`. Howerver, still a warning messege is generated indicating that the it still contains NA values.

Solution - 14 a]

In [98]:
```r
library(dplyr)
library(forcats)

# Count occurrences of each Violation Description and identify the top 5
top_5_violation_desc <- df7 %>%
  count(VIOLATION.DESCRIPTION) %>%
  arrange(desc(n)) %>%
  head(5) %>%
  pull(VIOLATION.DESCRIPTION)

# Filter the dataset to keep only rows with the top 5 Violation Descriptions
df7 <- df7 %>%
  filter(VIOLATION.DESCRIPTION %in% top_5_violation_desc)

# Define shortened labels for the top 5 Violation Descriptions
short_labels <- c("Missing", "Work Description", "Inspection Arrangement", "Obtain Permit", "Public Nuisance")

# Create a new column with the shortened descriptions
df7 <- df7 %>%
  mutate(shortened_description = case_when(
    VIOLATION.DESCRIPTION == top_5_violation_desc[1] ~ short_labels[1],
    VIOLATION.DESCRIPTION == top_5_violation_desc[2] ~ short_labels[2],
    VIOLATION.DESCRIPTION == top_5_violation_desc[3] ~ short_labels[3],
    VIOLATION.DESCRIPTION == top_5_violation_desc[4] ~ short_labels[4],
    VIOLATION.DESCRIPTION == top_5_violation_desc[5] ~ short_labels[5]
  ))

# Convert the shortened descriptions into a factor and explicitly define NA levels
df7$shortened_description <- fct_explicit_na(factor(df7$shortened_description))


# Print out the unique values for Violation Description
unique_violation_desc <- unique(df7$shortened_description)
print(unique_violation_desc)
```

```
[1] Missing               Obtain Permit         Work Description
[4] Inspection Arrangement Public Nuisance
5 Levels: Inspection Arrangement Missing Obtain Permit ... Work Description
```

> 💡 **Interpretation of Results** 💡 -
>
> The top 5 Most Common Violation Descriptions are- Missing, Obtain Permit, Work Description, Inspection Arrangement, Public Nuisance.

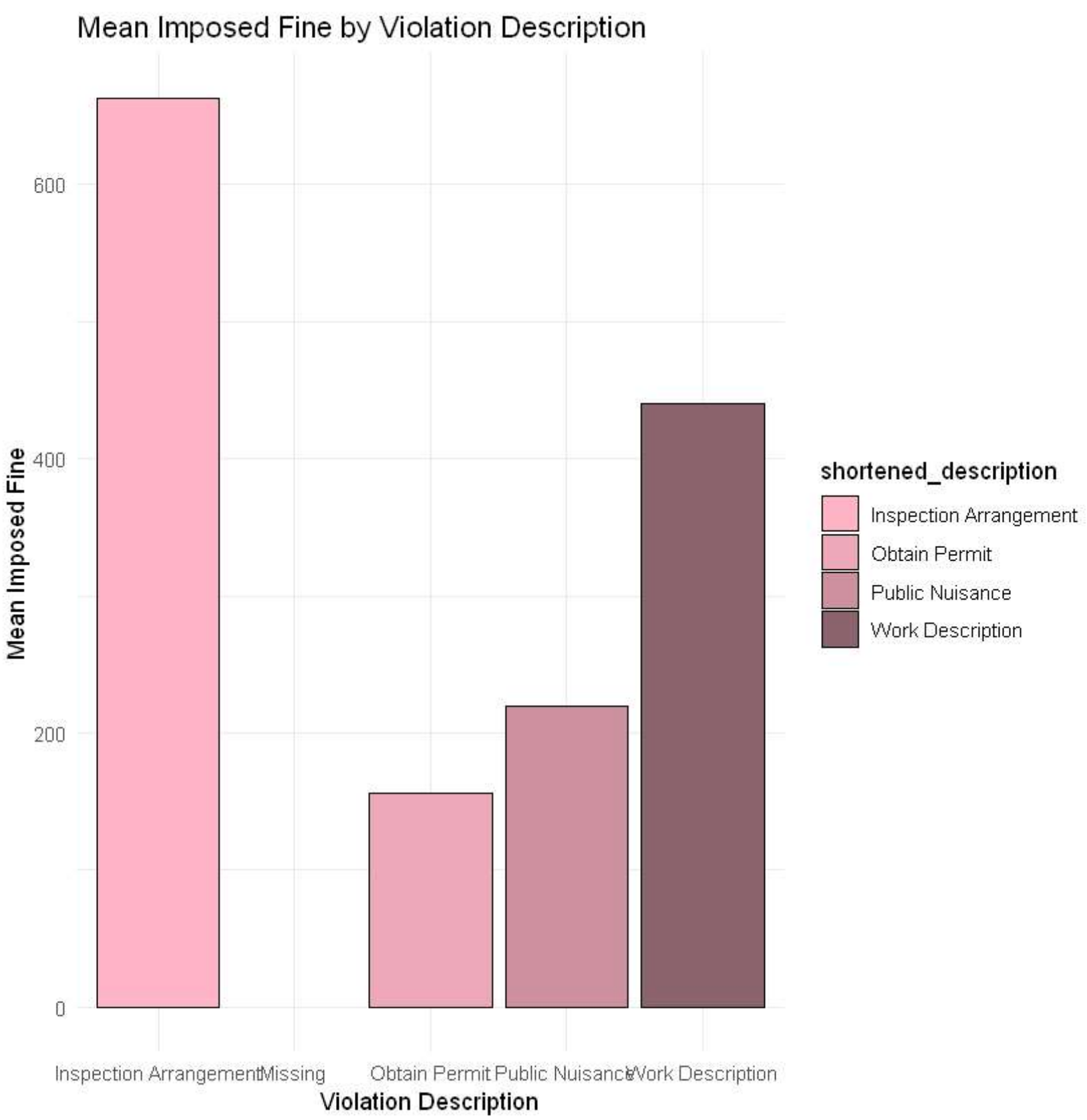
Solution - 15

In [76]:
```r
# Install and load the leaflet package
install.packages("leaflet")
library(leaflet)

# Load necessary libraries
library(dplyr)
library(ggplot2)
```

```
also installing the dependencies 'rlang', 'Rcpp', 'htmltools', 'terra', 'htmlwidgets', 'raster'

  There are binary versions available but the source versions are later:
             binary source needs_compilation
rlang        0.4.11  1.1.3              TRUE
Rcpp          1.0.6 1.0.12              TRUE
htmltools   0.5.1.1  0.5.7              TRUE
terra         1.2-5 1.7-71              TRUE
htmlwidgets   1.5.3  1.6.4             FALSE
raster       3.4-10 3.6-26              TRUE
leaflet     2.0.4.1  2.2.1             FALSE

  Binaries will be installed
package 'rlang' successfully unpacked and MD5 sums checked
package 'Rcpp' successfully unpacked and MD5 sums checked
package 'htmltools' successfully unpacked and MD5 sums checked
package 'terra' successfully unpacked and MD5 sums checked
package 'raster' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asusw\AppData\Local\Temp\RtmpiWbHnW\downloaded_packages
installing the source packages 'htmlwidgets', 'leaflet'

Warning message in install.packages("leaflet"):
"installation of package 'htmlwidgets' had non-zero exit status"Warning message in install.packages("leaflet"):
"installation of package 'leaflet' had non-zero exit status"
Error in library(leaflet): there is no package called 'leaflet'
Traceback:

1. library(leaflet)
```

In [99]:
```r
# Create the barplot
mean_imposed_fine <- df7 %>%
  group_by(shortened_description) %>%
  summarise(mean_imposed_fine = mean(IMPOSED.FINE, na.rm = TRUE))

# Arrange bars in decreasing order of mean imposed fine
mean_imposed_fine <- mean_imposed_fine %>%
  arrange(desc(mean_imposed_fine))

# Create the barplot
ggplot(mean_imposed_fine, aes(x = shortened_description, y = mean_imposed_fine, fill = shortened_description)) +
  geom_bar(stat = "identity", color = "black") +
  labs(x = "Violation Description", y = "Mean Imposed Fine", title = "Mean Imposed Fine by Violation Description") +
  scale_fill_manual(values = c("pink1", "pink2", "pink3", "pink4", "red")) +
  theme_minimal()
```

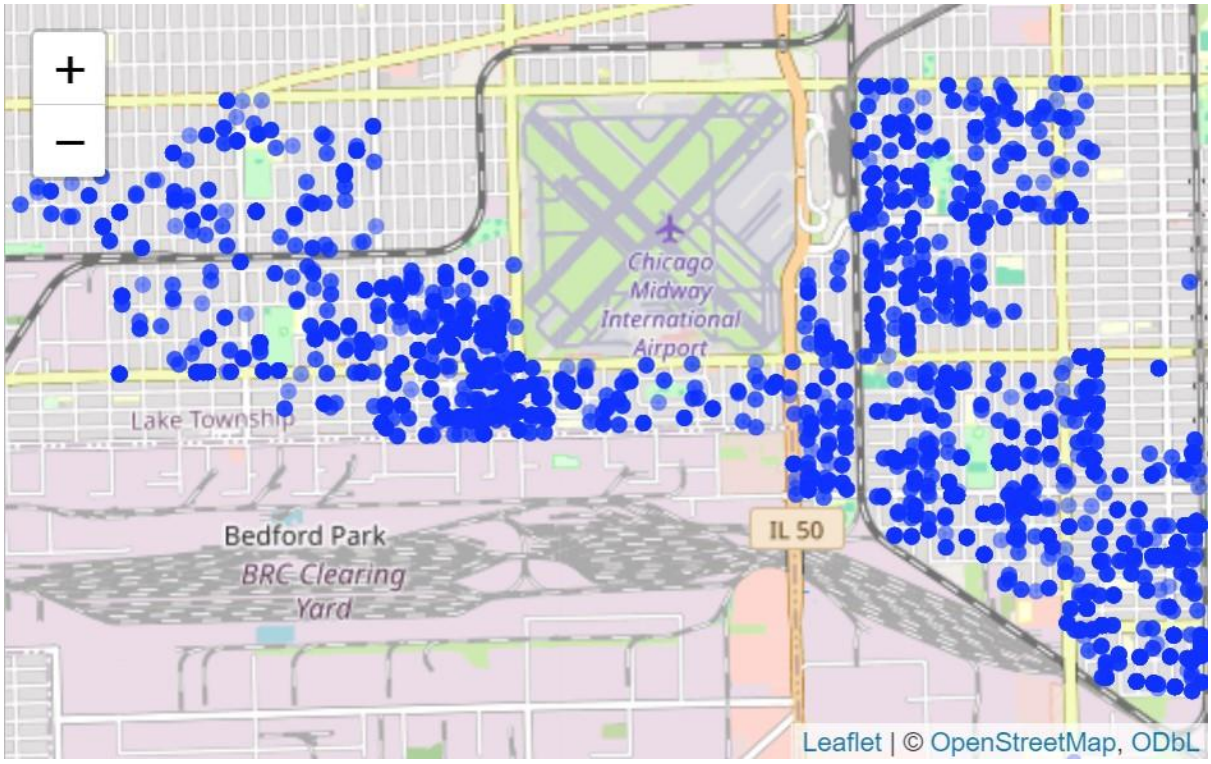
```
Warning message:
"Removed 1 rows containing missing values (position_stack)."
```



> **💡Interpretation of Code & Graph💡 -**
>
> Using `leaflet`, `dplyr` and `ggplot2` this graph was generated.
>
> We calculate the mean imposed fine for wach if the top 5 violaiton descriptions (`shortened_description`) from the `df` dataset. Thisis achieved by grouping the data by `shortened_desccription` and summarizing the mean imposed fine for rach group using `summarise()` function.
>
> The resulting barplot visually represents the mean imposed fine for each of the top five violation descriptions. It helps in understanding the variations in fines imposed for different types of violations.
>
> We noticed that in the graph 'Inspection Arrangement' & 'Work Descrition' have higher fines ccompared to others. This could be because of multiple reasons such as:-
>
> 1. They tend to have more severe infractions or violations of regulations compared to other type of violations.
> 2. They occur more frequently or are more commonly reported, leading to increased scrutiny and enforcement by regulatory athorities.
>
> NOTE - The warning message "Removed 1 rows containing missing values (position_stack)" indicates that during the creation of the barplot, one row was removed due to missing values. This typically occurs when there are NA (missing) values present in the dataset used for plotting. As a result the ggplot2 removed this row from the plot to avoid errors or inconsistencies in the visualization.

Solution - 16

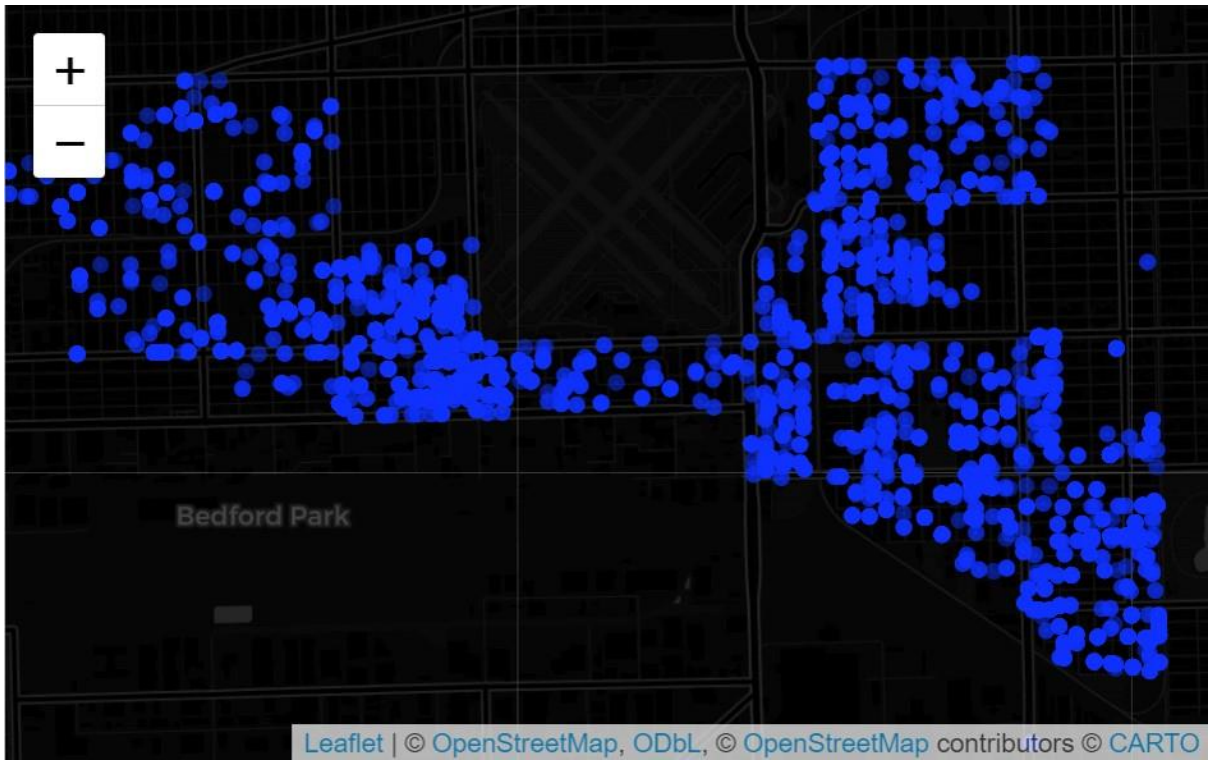
```r
# Install leaflet package if not already installed
install.packages("leaflet")

# Load the leaflet package
library(leaflet)

m <- leaflet() %>%
  addTiles() %>%
  addCircles(lng = df5_filtered$LONGITUDE, lat = df5_filtered$LATITUDE)
# Print the map
m
```



Interpretation of Results:-

An interactive map is generated that displays circular markers at specific geographical locations based on the provided longitude and latitude coordinates from df5_filtered dataframe. This can help visualize the spatial distribution of data points and identify patterns or clusters on the map.

```r
#Solution 17

# Create the leaflet map and add tiles and circles
m <- leaflet() %>%
  addTiles() %>%
  addCircles(lng = df5_filtered$LONGITUDE, lat = df5_filtered$LATITUDE) %>%
  addProviderTiles(providers$CartoDB.DarkMatter)
# Print the map
m
```



Interpretation of Results:-

A dark themed map is generated.

Solution 18

My assignment submission includes the following:

a) 2 files containing code (IPYNB AND .R) As I was unable to generate the maps on Jupyter notebook where I completed most of my work.
b) a write-up in a PDF that clearly includes all of my code, results, and interpretation statements, together in a single document.