Project_1

Anandu Muralidharan; Nehal UR Rahman; Karanjeet Singh Sihra

2023-02-13

Anandu Muralidharan - 991713040

Nehal UR Rahman - 991691259

Karanjeet Singh Sihra – 991705289

```
library(sets)
library(ggplot2)
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✓ tibble  3.1.8     ✓ dplyr   1.1.0
## ✓ tidyr   1.3.0     ✓ stringr 1.5.0
## ✓ readr   2.1.3     ✓ forcats 1.0.0
## ✓ purrr   1.0.1
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ forcats::%>%()  masks stringr::%>%(), dplyr::%>%(), purrr::%>%(),
tidyr::%>%(), tibble::%>%(), sets::%>%()
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()

library(ISLR)
library(moments)
library(dplyr)
```

Introduction

In this data set we are trying to analyze the Killed or Survived Data set which tells us about deaths or survivors in road accidents over the span of years while giving us information of the accident type, location, district and much more. We have four such data over the years 2015, 2016, 2017, 2018. The data is majorly categorical with lot of “” values.

```
df1 <- read.csv("2015_KSI.csv")
df2 <- read.csv("2016_KSI.csv")
df3 <- read.csv("2017_KSI.csv")
df4 <- read.csv("2018_KSI.csv")
```

```r
colnames1 <- colnames(df1)
colnames2 <- colnames(df2)
colnames3 <- colnames(df3)
colnames4 <- colnames(df4)
```

Q1

This is a simple function which takes four dataframes as parameters and then prints out the similarity index of each dataframe with each other. This function has made our life much simpler in the aspect that this set of code needs to be repeated to check the similarity index later after doing our task of merging the data.

```r
check_similarity <- function(df1, df2, df3, df4) {
  colnames1 <- colnames(df1)
  colnames2 <- colnames(df2)
  colnames3 <- colnames(df3)
  colnames4 <- colnames(df4)
  identical_colnames1_2 <- all(colnames1 == colnames2)
  identical_colnames1_3 <- all(colnames1 == colnames3)
  identical_colnames1_4 <- all(colnames1 == colnames4)
  identical_colnames2_3 <- all(colnames1 == colnames4)
  identical_colnames2_4 <- all(colnames1 == colnames4)
  identical_colnames3_4 <- all(colnames3 == colnames4)
  cat("Are the columns in df1 and df2 identical? ", identical_colnames1_2,
"\n")
  cat("Are the columns in df1 and df3 identical? ", identical_colnames1_3,
"\n")
  cat("Are the columns in df1 and df4 identical? ", identical_colnames1_4,
"\n")
  cat("Are the columns in df2 and df3 identical? ", identical_colnames2_3,
"\n")
  cat("Are the columns in df2 and df3 identical? ", identical_colnames2_4,
"\n")
  cat("Are the columns in df3 and df4 identical? ", identical_colnames3_4,
"\n")
}
```

Here you can see that the dataframes df1, df2, and df3 have a few columns that are not identical, but df3 and df4 and identical among themselves. Since we have to merge the data of the four dataframes, we can simply work towards making the columns of df1, df2 and df3 similar.

```r
check_similarity(df1, df2, df3, df4)

## Are the columns in df1 and df2 identical?  FALSE
## Are the columns in df1 and df3 identical?  FALSE
## Are the columns in df1 and df4 identical?  FALSE
## Are the columns in df2 and df3 identical?  FALSE
## Are the columns in df2 and df3 identical?  FALSE
## Are the columns in df3 and df4 identical?  TRUE
```

In this chunk we are trying to find out what are the columns which are different from each other, in the dataframes df1 - df3.

```
diff_colnames1_2 <- setdiff(colnames1, colnames2)
diff_colnames2_1 <- setdiff(colnames2, colnames1)

diff_colnames1_3 <- setdiff(colnames1, colnames3)
diff_colnames3_1 <- setdiff(colnames3, colnames1)

diff_colnames2_3 <- setdiff(colnames2, colnames3)
diff_colnames3_2 <- setdiff(colnames3, colnames2)

# Print the differences
cat("Columns in df1 but not in df2: ", diff_colnames1_2, "\n")
```

```
## Columns in df1 but not in df2:  NEIGHBOURHOOD VEHICLES_IN_STREET
```

```
cat("Columns in df2 but not in df1: ", diff_colnames2_1, "\n")
```

```
## Columns in df2 but not in df1:  NEIGHBOUR VEHICLE_STREET
```

```
cat("cols df1 ", ncol(df1), "\n")
```

```
## cols df1  58
```

```
cat("cols df2", ncol(df2), "\n")
```

```
## cols df2 58
```

```
print("-----------------")
```

```
## [1] "-----------------"
```

```
cat("Columns in df1 but not in df3: ", diff_colnames1_3, "\n")
```

```
## Columns in df1 but not in df3:  VEHICLES_IN_STREET
```

```
cat("Columns in df3 but not in df1: ", diff_colnames3_1, "\n")
```

```
## Columns in df3 but not in df1:  VEHICLE_IN_STREET
```

```
cat("cols df1", ncol(df1), "\n")
```

```
## cols df1 58
```

```
cat("cols df3", ncol(df3), "\n")
```

```
## cols df3 58
```

```
print("-----------------")
```

```
## [1] "-----------------"
```

```
cat("Columns in df2 but not in df3: ", diff_colnames2_3, "\n")
```

```
## Columns in df2 but not in df3:  NEIGHBOUR VEHICLE_STREET
```

```
cat("Columns in df3 but not in df2: ", diff_colnames3_2, "\n")
```

```
## Columns in df3 but not in df2:  NEIGHBOURHOOD VEHICLE_IN_STREET
```

```
cat("cols df2", ncol(df2), "\n")
```

```
## cols df2 58
```

```
cat("cols df3", ncol(df3), "\n")
```

```
## cols df3 58
```

We found out here that the Neighbourhood and Vehicles in street are the two columns which have been spelt wrong and hence causing us the issue. The evidence for the fact that issue is happening only because column names are spelt wrong is because the number of column are pretty much the same. On further analysis the data that the columns are containing is also the same.

So here we just need to rename them

```
colnames(df1)[which(colnames(df1) == "VEHICLES_IN_STREET")] <-
"VEHICLE_IN_STREET"
colnames(df2)[which(colnames(df2) == "VEHICLE_STREET")] <-
"VEHICLE_IN_STREET"
colnames(df2)[which(colnames(df2) == "NEIGHBOUR")] <- "NEIGHBOURHOOD"
```

Calling the function to check similarity.

```
check_similarity(df1, df2, df3, df4)
```

```
## Are the columns in df1 and df2 identical?  TRUE
## Are the columns in df1 and df3 identical?  TRUE
## Are the columns in df1 and df4 identical?  TRUE
## Are the columns in df2 and df3 identical?  TRUE
## Are the columns in df2 and df3 identical?  TRUE
## Are the columns in df3 and df4 identical?  TRUE
```

Combining the four dataframes

```
df_combined <- rbind(df1, df2, df3, df4)
cat("rows ",nrow(df_combined),"\n")
```

```
## rows  3989
```

```
cat("cols", ncol(df_combined))
```

```
## cols 58
```

Distinct column names

```
colnames(df_combined)

##  [1] "X"                 "Y"                 "INDEX_"           
##  [4] "ACCNUM"            "YEAR"              "DATE"             
##  [7] "TIME"              "HOUR"              "STREET1"          
## [10] "STREET2"           "OFFSET"            "ROAD_CLASS"       
## [13] "DISTRICT"          "WARDNUM"           "DIVISION"         
## [16] "LATITUDE"          "LONGITUDE"         "LOCCOORD"         
## [19] "ACCLOC"            "TRAFFCTL"          "VISIBILITY"       
## [22] "LIGHT"             "RDSFCOND"          "ACCLASS"          
## [25] "IMPACTYPE"         "INVTYPE"           "INVAGE"           
## [28] "INJURY"            "FATAL_NO"          "INITDIR"          
## [31] "VEHTYPE"           "MANOEUVER"         "DRIVACT"          
## [34] "DRIVCOND"          "PEDTYPE"           "PEDACT"           
## [37] "PEDCOND"           "CYCLISTYPE"        "CYCACT"           
## [40] "CYCCOND"           "PEDESTRIAN"        "CYCLIST"          
## [43] "AUTOMOBILE"        "MOTORCYCLE"        "TRUCK"            
## [46] "TRSN_CITY_VEH"     "EMERG_VEH"         "PASSENGER"        
## [49] "SPEEDING"          "AG_DRIV"           "REDLIGHT"         
## [52] "ALCOHOL"           "DISABILITY"        "POLICE_DIVISION"  
## [55] "HOOD_ID"           "NEIGHBOURHOOD"     "ObjectId"         
## [58] "VEHICLE_IN_STREET"
```

Taking out the columns YEAR, VEHICLE_IN_STREET, DISTRICT, NEIGHBOURHOOD to make dataframe for first question.

```
df_q1 <- df_combined[c("YEAR", "VEHICLE_IN_STREET", "DISTRICT",
"NEIGHBOURHOOD")]
str(df_q1)

## 'data.frame':    3989 obs. of  4 variables:
##  $ YEAR             : int  2015 2015 2015 2015 2015 2015 2015 2015 2015
2015 ...
##  $ VEHICLE_IN_STREET: int  46 23 17 14 14 14 52 52 52 9 ...
##  $ DISTRICT         : chr  "North York" "Etobicoke York" "Etobicoke York"
"Etobicoke York" ...
##  $ NEIGHBOURHOOD    : chr  "Pleasant View (46)" "Pelmo Park-Humberlea
(23)" "Mimico (17)" "Islington-City Centre West (14)" ...
```

Here one thing you observe is that the neighborhood column has name of some neighbourhood and a number inside the parentheses. On further investigation it can be seen that the number matches the value in column Vehicle_In_Street hence we will try and removing that portion from Nehighbourhood column

```
df_q1$NEIGHBOURHOOD <- str_trim(sub("\\(.*", "", df_q1$NEIGHBOURHOOD))
#Here we are making use of sub function to take anything a "space followed by
an open parentheses and any number followed after", this portion is replaced
by empty space, which can then be trimmed using the str_trim function.
```

```
nrow(df_q1)

## [1] 3989
```

Unique year -> shows we have merged all four years

```
unique(df_q1$YEAR)

## [1] 2015 2016 2017 2018
```

Unique Vehicle in street

```
unique(df_q1$VEHICLE_IN_STREET)

##   [1]  46  23  17  14  52   9   2  82 103 108 121  61  95  50  87 136  75
51
##  [19] 118  78   1   6 133 117  45 126  29  26   3 131   7  90 130 119  63
30
##  [37] 124  36  58  94  85  71  28  33 116  47 139  68 111  44 101  31  62
16
##  [55]  22  77  70  83 110  76  21 137 107  41 120  35  48 132  34  18 127
73
##  [73]  25  88  49 138  99 122  80  81  89 104  64   8 106  43  39  15 129
93
##  [91]  98 112  56  57  79  53  72  32  96  65  59  27 128  42  54 113  69
19
## [109]  24  84  10  20  60 102  86   4  38 123  74  92 109  11 140 100 134
135
## [127]  37  97  55  91  40  13  66  67 115 105 114 125
```

Unique Neighbourhood

```
unique(df_q1$NEIGHBOURHOOD)

##   [1] "Pleasant View"
##   [2] "Pelmo Park-Humberlea"
##   [3] "Mimico"
##   [4] "Islington-City Centre West"
##   [5] "Bayview Village"
##   [6] "Edenbridge-Humber Valley"
##   [7] "Mount Olive-Silverstone-Jamestown"
##   [8] "Niagara"
##   [9] "Lawrence Park South"
##  [10] "Briar Hill-Belgravia"
##  [11] "Oakridge"
##  [12] "Taylor-Massey"
##  [13] "Annex"
##  [14] "Newtonbrook East"
##  [15] "High Park-Swansea"
##  [16] "West Hill"
##  [17] "Church-Yonge Corridor"
##  [18] "Willowdale East"
##  [19] "Tam O'Shanter-Sullivan"
##  [20] "Kensington-Chinatown"
##  [21] "West Humber-Clairville"
##  [22] "Kingsview Village-The Westway"
##  [23] "Centennial Scarborough"
##  [24] "L'Amoreaux"
##  [25] "Parkwoods-Donalda"
##  [26] "Dorset Park"
##  [27] "Maple Leaf"
##  [28] "Downsview-Roding-CFB"
##  [29] "Thistletown-Beaumond Heights"
##  [30] "Rouge"
##  [31] "Willowridge-Martingrove-Richview"
##  [32] "Junction Area"
##  [33] "Milliken"
##  [34] "Wexford/Maryvale"
##  [35] "The Beaches"
##  [36] "Brookhaven-Amesbury"
##  [37] "Kennedy Park"
##  [38] "Newtonbrook West"
##  [39] "Old East York"
##  [40] "Wychwood"
##  [41] "South Parkdale"
##  [42] "Cabbagetown-South St.James Town"
##  [43] "Rustic"
##  [44] "Clanton Park"
##  [45] "Steeles"
##  [46] "Don Valley Village"
##  [47] "Scarborough Village"
##  [48] "North Riverdale"
##  [49] "Rockcliffe-Smythe"
##  [50] "Flemingdon Park"
##  [51] "Forest Hill South"
##  [52] "Yorkdale-Glen Park"
##  [53] "East End-Danforth"
##  [54] "Stonegate-Queensway"
##  [55] "Humbermede"
##  [56] "Waterfront Communities-The Island"
##  [57] "South Riverdale"
##  [58] "Dufferin Grove"
##  [59] "Keelesdale-Eglinton West"
##  [60] "Bay Street Corridor"
##  [61] "Humber Summit"
##  [62] "Woburn"
##  [63] "Oakwood Village"
##  [64] "Bridle Path-Sunnybrook-York Mills"
##  [65] "Clairlea-Birchmount"
##  [66] "Westminster-Branson"
##  [67] "Hillcrest Village"
##  [68] "Malvern"
##  [69] "Bathurst Manor"
##  [70] "New Toronto"
##  [71] "Bendale"
##  [72] "Moss Park"
##  [73] "Glenfield-Jane Heights"
##  [74] "High Park North"
##  [75] "Bayview Woods-Steeles"
##  [76] "Eglinton East"
##  [77] "Mount Pleasant East"
##  [78] "Birchcliffe-Cliffside"
##  [79] "Palmerston-Little Italy"
##  [80] "Trinity-Bellwoods"
##  [81] "Runnymede-Bloor West Village"
##  [82] "Mount Pleasant West"
##  [83] "Woodbine Corridor"
##  [84] "Humber Heights-Westmount"
##  [85] "Humewood-Cedarvale"
##  [86] "Victoria Village"
##  [87] "Bedford Park-Nortown"
##  [88] "Kingsway South"
##  [89] "Agincourt North"
##  [90] "Dovercourt-Wallace Emerson-Junction"
##  [91] "Rosedale-Moore Park"
##  [92] "Beechborough-Greenbrook"
##  [93] "Leaside-Bennington"
##  [94] "Broadview North"
##  [95] "University"
##  [96] "Henry Farm"
##  [97] "Regent Park"
##  [98] "Englemount-Lawrence"
##  [99] "Casa Loma"
## [100] "Greenwood-Coxwell"
## [101] "Danforth East York"
## [102] "York University Heights"
## [103] "Agincourt South-Malvern West"
## [104] "Banbury-Don Mills"
## [105] "O'Connor-Parkview"
## [106] "Weston"
## [107] "Blake-Jones"
## [108] "Long Branch"
## [109] "Black Creek"
## [110] "Little Portugal"
## [111] "Princess-Rosethorn"
## [112] "Alderwood"
## [113] "Woodbine-Lumsden"
## [114] "Forest Hill North"
## [115] "Roncesvalles"
## [116] "Rexdale-Kipling"
## [117] "Lansing-Westgate"
## [118] "Cliffcrest"
## [119] "North St.James Town"
## [120] "Corso Italia-Davenport"
## [121] "Caledonia-Fairbank"
## [122] "Eringate-Centennial-West Deane"
## [123] "Guildwood"
## [124] "Yonge-Eglinton"
## [125] "Highland Creek"
## [126] "Morningside"
## [127] "Willowdale West"
## [128] "Yonge-St.Clair"
## [129] "Thorncliffe Park"
## [130] "Weston-Pellam Park"
## [131] "St.Andrew-Windfields"
## [132] "Etobicoke West Mall"
## [133] "Danforth"
## [134] "Playter Estates-Danforth"
## [135] "Mount Dennis"
## [136] "Lawrence Park North"
## [137] "Lambton Baby Point"
## [138] "Ionview"
```

Unique District <- even though this data has value in it, since the coulmn doesn't particularly affect the data we are trying to analyze we can leave it be

```
unique(df_q1$DISTRICT)
```

```
## [1] "North York"              "Etobicoke York"         "Toronto and East 
York"
## [4] "Scarborough"             "<Null>"                 "Toronto East York"
```

The exploration at this point which we are trying to do is to have a count per neighbourhood of people who were either killed or injured. Since there is no particular column in the whole dataset which has the "number" of people either killed or injured what we do is we make use of Injury column to determine who were not particulary injured or the None class and remove that.

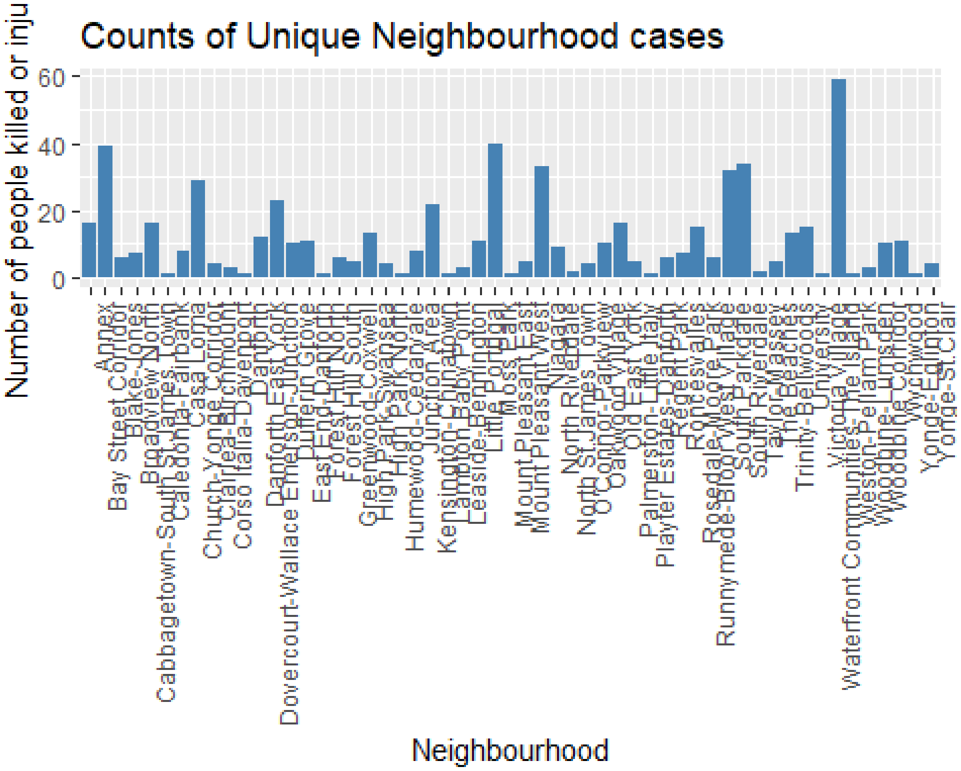
In order to reach that point however we had to get rid of Districts which did not have "Tor" in it.

```
tor_rows <- df_combined[grepl("tor", df_combined$DISTRICT, ignore.case = 
TRUE), ]

tor_rows_no_none <- filter(tor_rows, tor_rows$INJURY != "None")

tor_rows_no_none$NEIGHBOURHOOD <- str_trim(sub("\\(.*", "", 
tor_rows_no_none$NEIGHBOURHOOD))

neighbourhood_count <- table(tor_rows_no_none$NEIGHBOURHOOD)

neighbourhood_count_df <- as.data.frame(neighbourhood_count, responseName = 
"count")
names(neighbourhood_count_df) <- c("Neighbourhood", "count")
neighbourhood_count_df
```

```
##                          Neighbourhood count
## 1                                Annex    16
## 2                  Bay Street Corridor    39
## 3                          Blake-Jones     6
## 4                      Broadview North     7
## 5      Cabbagetown-South St.James Town    16
## 6                   Caledonia-Fairbank     1
## 7                            Casa Loma     8
## 8                Church-Yonge Corridor    29
## 9                  Clairlea-Birchmount     4
## 10              Corso Italia-Davenport     3
## 11                            Danforth     1
## 12                  Danforth East York    12
## 13 Dovercourt-Wallace Emerson-Junction    23
## 14                      Dufferin Grove    10
## 15                   East End-Danforth    11
## 16                   Forest Hill North     1
## 17                   Forest Hill South     6
## 18                   Greenwood-Coxwell     5
## 19                   High Park-Swansea    13
## 20                     High Park North     4
## 21                  Humewood-Cedarvale     1
## 22                       Junction Area     8
## 23                Kensington-Chinatown    22
## 24                  Lambton Baby Point     1
## 25                  Leaside-Bennington     3
## 26                     Little Portugal    11
## 27                           Moss Park    40
## 28                 Mount Pleasant East     1
## 29                 Mount Pleasant West     5
## 30                             Niagara    33
## 31                     North Riverdale     9
## 32                 North St.James Town     2
## 33                   O'Connor-Parkview     4
## 34                     Oakwood Village    10
## 35                       Old East York    16
## 36             Palmerston-Little Italy     5
## 37            Playter Estates-Danforth     1
## 38                         Regent Park     6
## 39                        Roncesvalles     7
## 40                 Rosedale-Moore Park    15
## 41        Runnymede-Bloor West Village     6
## 42                      South Parkdale    32
## 43                     South Riverdale    34
## 44                       Taylor-Massey     2
## 45                         The Beaches     5
## 46                   Trinity-Bellwoods    13
## 47                          University    15
## 48                    Victoria Village     1
## 49   Waterfront Communities-The Island    59
## 50                  Weston-Pellam Park     1
## 51                    Woodbine-Lumsden     3
## 52                   Woodbine Corridor    10
## 53                            Wychwood    11
## 54                      Yonge-Eglinton     1
## 55                      Yonge-St.Clair     4
```

In this plot we understand that “Waterfront Communities - The Island” is a neighbourhood which is most prone to accidents.

```
ggplot(neighbourhood_count_df, aes(x = Neighbourhood, y = count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Neighbourhood", y = "Number of people killed or injured", title = 
"Counts of Unique Neighbourhood cases")+theme(axis.text.x = 
element_text(angle = 90, hjust = 1))
```



Q2

The sequence of code below provides us the total number(sum) of vehicles in each district during the accident.

```
unique(df_q1$DISTRICT)
```

```
## [1] "North York"            "Etobicoke York"        "Toronto and East
York"
## [4] "Scarborough"           "<Null>"                "Toronto East York"
```

As we see above there are 6 unique neighborhoods in the given dataset.

```
#We have used the subset function to get the data for only those districts
which has Null values
subset(df_q1, DISTRICT == "<Null>")
```

```
##      YEAR VEHICLE_IN_STREET DISTRICT          NEIGHBOURHOOD
## 100  2015               130   <Null>               Milliken
## 2450 2017                25   <Null> Glenfield-Jane Heights
## 2694 2017                37   <Null>        Willowdale West
## 2695 2017                37   <Null>        Willowdale West
## 2696 2017                37   <Null>        Willowdale West
## 2697 2017                37   <Null>        Willowdale West
```

As shown in the table above, there are 6 rows that contains Null values in the district column. For now We will be keeping these values as it is and then cleaning them, when we will be visualizing the data.

```
subset(df_q1, DISTRICT == "<Null>")
```

```
##      YEAR VEHICLE_IN_STREET DISTRICT          NEIGHBOURHOOD
## 100  2015               130   <Null>               Milliken
## 2450 2017                25   <Null> Glenfield-Jane Heights
## 2694 2017                37   <Null>        Willowdale West
## 2695 2017                37   <Null>        Willowdale West
## 2696 2017                37   <Null>        Willowdale West
## 2697 2017                37   <Null>        Willowdale West
```

Now we use the function tapply to show the sum of vehicles in street during the accident in each district by grouping them based on district.

```
#This is done by creating another data frame called df_q2. To get the total
number of vehicles in street with respect to the neighborhood we use the
tapply function.
df_q2 <- as.data.frame(tapply(df_q1$VEHICLE_IN_STREET, df_q1$DISTRICT, FUN =
sum))
#We name the total number of vehicles in the street during the accident as
"count"
names(df_q2) <- c( "count")
df_q2
```

```
##                         count
## <Null>                    303
## Etobicoke York          30572
## North York              50188
## Scarborough            117032
## Toronto and East York   92208
## Toronto East York         837
```

We can see that the total number of vehicle in the street during the accident is the highest in Scarborough followed by Toronto and East York.

##Q3 Here, we are Calculating the average mean of vehicles in street in each district during accidents.To find the top 5 neighborhoods with the highest number of vehicles in the street we need first to see the average.

```
#The central tendency measure "MEAN" is the average value for a given set of
data.
# Using tapply we can group the "VEHICLES_IN_STREET" based on neighborhood
values, after which we can find the mean of each of these values per
group.This data will further be used to make the table.
avg_per_hood <- tapply(df_q1$VEHICLE_IN_STREET, df_q1$NEIGHBOURHOOD, FUN =
mean)
# using sort function to arrange top Average vehicles in order.
avg_per_hood_sort <- sort(avg_per_hood, decreasing = TRUE)
#Selecting the top five rows of the sorted data
df_q3 <- as.data.frame(avg_per_hood_sort[1:5])
# Here we are Changing column name to count
names(df_q3) <- c( "count")
df_q3
```

```
##                      count
## Guildwood              140
## Scarborough Village    139
## Eglinton East          138
## Woburn                 137
## West Hill              136
```

Here we can realize that the highest number of average vehicle on street is for Guildwood Neighbourhood. Followed by Scarborough Village and Eglington East.

##DATA ANALYSIS To begin our data exploration and data visulization, we create another dataframe called df_analyze. In this dataset we only include those atrributes or columns that we feel are important to analyze the data. We do this to get the required visualizations.

```
colnames(df_combined)
```

```
##  [1] "X"                 "Y"                 "INDEX_"
##  [4] "ACCNUM"            "YEAR"              "DATE"
##  [7] "TIME"              "HOUR"              "STREET1"
## [10] "STREET2"           "OFFSET"            "ROAD_CLASS"
## [13] "DISTRICT"          "WARDNUM"           "DIVISION"
## [16] "LATITUDE"          "LONGITUDE"         "LOCCOORD"
## [19] "ACCLOC"            "TRAFFCTL"          "VISIBILITY"
## [22] "LIGHT"             "RDSFCOND"          "ACCLASS"
## [25] "IMPACTYPE"         "INVTYPE"           "INVAGE"
## [28] "INJURY"            "FATAL_NO"          "INITDIR"
## [31] "VEHTYPE"           "MANOEUVER"         "DRIVACT"
## [34] "DRIVCOND"          "PEDTYPE"           "PEDACT"
## [37] "PEDCOND"           "CYCLISTYPE"        "CYCACT"
## [40] "CYCCOND"            "PEDESTRIAN"         "CYCLIST"
## [43] "AUTOMOBILE"         "MOTORCYCLE"         "TRUCK"
## [46] "TRSN_CITY_VEH"      "EMERG_VEH"          "PASSENGER"
## [49] "SPEEDING"           "AG_DRIV"            "REDLIGHT"
## [52] "ALCOHOL"            "DISABILITY"         "POLICE_DIVISION"
## [55] "HOOD_ID"            "NEIGHBOURHOOD"      "ObjectId"
## [58] "VEHICLE_IN_STREET"
```

```
df_analyze <- df_combined[c("HOUR","ROAD_CLASS", "DISTRICT", "LOCCOORD",
"ACCLOC", "TRAFFCTL", "VISIBILITY", "LIGHT", "RDSFCOND", "ACCLASS", "INJURY",
"SPEEDING", "REDLIGHT", "ALCOHOL", "NEIGHBOURHOOD", "VEHICLE_IN_STREET")]
str(df_analyze)
```

```
## 'data.frame':    3989 obs. of  16 variables:
##  $ HOUR             : int  13 15 1 0 0 0 14 14 14 18 ...
##  $ ROAD_CLASS       : chr  "Major Arterial" "Major Arterial" "Major
Arterial" "Major Arterial" ...
##  $ DISTRICT         : chr  "North York" "Etobicoke York" "Etobicoke York"
"Etobicoke York" ...
##  $ LOCCOORD         : chr  "Intersection" "Intersection" "Mid-Block"
"Intersection" ...
##  $ ACCLOC           : chr  "At Intersection" "At Intersection" "Non
Intersection" "Intersection Related" ...
##  $ TRAFFCTL         : chr  "Traffic Signal" "Stop Sign" "No Control" "No
Control" ...
##  $ VISIBILITY       : chr  "Clear" "Clear" "Clear" "Clear" ...
##  $ LIGHT            : chr  "Daylight" "Daylight" "Dark, artificial" "Dark,
artificial" ...
##  $ RDSFCOND         : chr  "Dry" "Dry" "Dry" "Dry" ...
##  $ ACCLASS          : chr  "Non-Fatal Injury" "Non-Fatal Injury" "Non-
Fatal Injury" "Fatal" ...
##  $ INJURY           : chr  "Major" "None" "Major" "Minimal" ...
##  $ SPEEDING         : chr  "<Null>" "<Null>" "Yes" "Yes" ...
##  $ REDLIGHT         : chr  "<Null>" "<Null>" "<Null>" "<Null>" ...
##  $ ALCOHOL          : chr  "<Null>" "<Null>" "<Null>" "<Null>" ...
##  $ NEIGHBOURHOOD    : chr  "Pleasant View (46)" "Pelmo Park-Humberlea
(23)" "Mimico (17)" "Islington-City Centre West (14)" ...
##  $ VEHICLE_IN_STREET: int  46 23 17 14 14 14 52 52 52 9 ...
```

As we can see above, we have included 16 variables from the 58 variables of the original merged dataset. Using these 16 variables we will be providing the data visualizations.

Finding count of null in district and Neighbourhood

```
cat("Number of null in district ", sum(df_analyze$DISTRICT == "<Null>"),
"\n")

## Number of null in district  6

cat("Number of null in Neighbourhood ", sum(df_analyze$NEIGHBOURHOOD ==
"<Null>"), "\n")
```

```
## Number of null in Neighbourhood  0
```

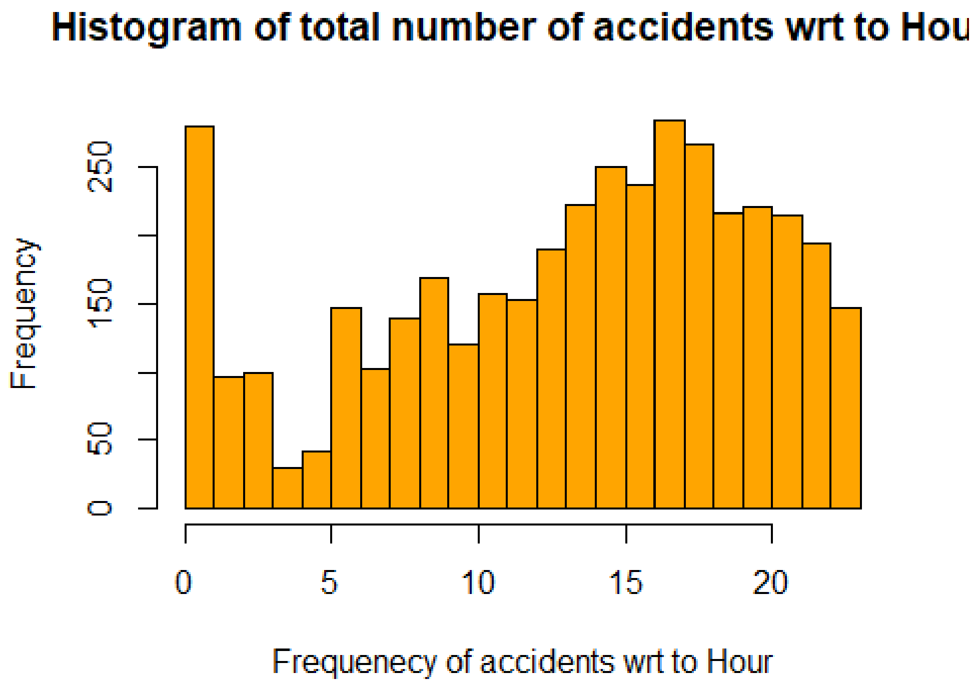
#As seen in the above output, we can see that there are 6 null values present in the district column and there are 0 null values in the neighborhood column. #Therefore, We can either find out the values of these 6 positions by analyzing the supporting columns from df_combine or we can simple drop the 6 rows. #Hence, for time being we will drop the rows so as to refine the data.

```
df_analyze <- subset(df_analyze, DISTRICT != "<Null>")
#Using this line of code we drop the rows containing null in the district
column
cat("Number of null in district ", sum(df_analyze$DISTRICT == "<Null>"),
"\n")

## Number of null in district  0

#We check the number of null values in district again to make sure that the
null value rows are dropped
```

#We can see now that 6 rows have been removed as there are 0 null values in the district column.

#Now we will try to understand the spread of accidents over time in a duration of 24 hours.

```
hist(df_analyze$HOUR, xlab = "Frequenecy of accidents wrt to Hour",
     main="Histogram of total number of accidents wrt to Hour",
     col="orange", breaks = 20)
```



Here we can observe most cases have been registered between 1500 or 3PM in the afternoon to around 8PM in the evening. Also Early morning 12AM accidents have peaked.

```
tapply(df_analyze$HOUR, df_analyze$DISTRICT, FUN = mean)
```

```
##        Etobicoke York            North York           Scarborough 
##              13.71955              14.12336              13.25926 
## Toronto and East York     Toronto East York 
##              12.75731               9.00000
```

This here we understand that in most districts that are present in the dataset, most accidents happen around the afternoon time, except for in Toronto East York where most accidents tend to happen in the morning.

```
cat("Before: \n unique values in speeding", unique(df_analyze$SPEEDING),
"\n")
```

```
## Before: 
##  unique values in speeding <Null> Yes
```

```
#converting the null values to no
df_analyze$SPEEDING <- gsub("<Null>", "No", df_analyze$SPEEDING)
cat("After: \n unique values in speeding", unique(df_analyze$SPEEDING))
```

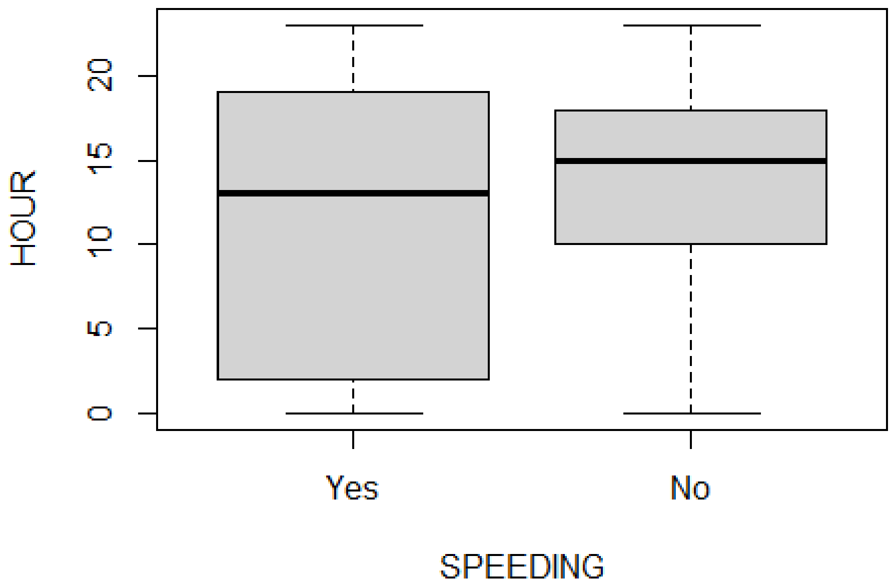
```
## After: 
##  unique values in speeding No Yes
```

```r
#using factor function to convert it into levels "Yes" and "No"
#This function will help give "yes" and "no" levels which eventually will
make R Realize that the data is ordinal in the speeding column
df_analyze$SPEEDING <- factor(df_analyze$SPEEDING, levels = c("Yes", "No"),
ordered= TRUE)
summary(df_analyze$SPEEDING)

##  Yes   No
##  650 3333
```

Here we can understand that majority of the accident cases have not been due to speeding, the other suspects now will be Influence of Alcohol, or Disability can be the reason for accidents.

#Here, we are drawing Side by Side Boxplot of Hour and Speeding

```r
boxplot(HOUR ~ SPEEDING, data = df_analyze)
```



People on average when speeding tend to get into accidents at around 1PM in the afternoon, how not speeding 4PM.

Conclusion

After merging the data of the four years we have concluded the following things: . Waterfront Communities - The Island is one of the most accident-prone neighborhoods in Toronto. . The total number of vehicles in the street during the accident is the highest in Scarborough followed by Toronto and East York. . Guildwood Neighbourhood has the highest average number of vehicles in street. . The frequency of Accidents is high between 3 PM - 8 PM and then again spikes towards 12 AM at midnight. . Speeding alone is not a major reason for the accidents that happen in Canada.