R Assignment 1- Neighbourhood Crime Rate

##The data describes crimerate in Toronto region in 2020. It shows the number and type of crimes prevelant in the neighbourhoods of Toronto.

Load data - CSV

This is a Neighbourhood Crime Rate dataset and downloaded Open data

```
crimedata <- readr::read_csv(file ="C:/Users/Dell/Downloads/Neighbourhood_Crime_Rates (1).csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   OBJECTID = col_double(),
##   Neighbourhood = col_character(),
##   F2020_Population_Projection = col_double(),
##   Assault_2020 = col_double(),
##   Assault_Rate2020 = col_double(),
##   AutoTheft_2020 = col_double(),
##   AutoTheft_Rate2020 = col_double(),
##   BreakAndEnter_2020 = col_double(),
##   BreakAndEnter_Rate2020 = col_double(),
##   Robbery_2020 = col_double(),
##   Robbery_Rate2020 = col_double(),
##   Shootings_2020 = col_double(),
##   Shootings_Rate2020 = col_double(),
##   Shape__Area = col_double(),
##   Shape__Length = col_double()
## )
```

2.Print the structure of your dataset.

```
str(crimedata)
```

```
## spec_tbl_df[,15] [140 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ OBJECTID                   : num [1:140] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Neighbourhood              : chr [1:140] "Yonge-St.Clair" "York University Heights" "Lansing-Westg
##  $ F2020_Population_Projection: num [1:140] 14083 30277 18146 17560 27410 ...
##  $ Assault_2020               : num [1:140] 23 341 97 156 104 131 84 56 94 56 ...
##  $ Assault_Rate2020           : num [1:140] 163 1126 535 888 379 ...
##  $ AutoTheft_2020             : num [1:140] 9 184 44 87 45 37 23 43 20 16 ...
##  $ AutoTheft_Rate2020         : num [1:140] 63.9 607.7 242.5 495.4 164.2 ...
```

```
##  $ BreakAndEnter_2020         : num [1:140] 26 105 39 81 35 54 66 10 41 33 ...
##  $ BreakAndEnter_Rate2020     : num [1:140] 185 347 215 461 128 ...
##  $ Robbery_2020               : num [1:140] 5 47 11 40 20 18 6 9 5 9 ...
##  $ Robbery_Rate2020           : num [1:140] 35.5 155.2 60.6 227.8 73 ...
##  $ Shootings_2020             : num [1:140] 0 12 1 14 0 2 0 1 9 0 ...
##  $ Shootings_Rate2020         : num [1:140] 0 3.963 0.551 7.973 0 ...
##  $ Shape__Area                : num [1:140] 1161315 13246656 5346186 6038326 7946202 ...
##  $ Shape__Length              : num [1:140] 5873 18505 11112 10079 11853 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   OBJECTID = col_double(),
##   ..   Neighbourhood = col_character(),
##   ..   F2020_Population_Projection = col_double(),
##   ..   Assault_2020 = col_double(),
##   ..   Assault_Rate2020 = col_double(),
##   ..   AutoTheft_2020 = col_double(),
##   ..   AutoTheft_Rate2020 = col_double(),
##   ..   BreakAndEnter_2020 = col_double(),
##   ..   BreakAndEnter_Rate2020 = col_double(),
##   ..   Robbery_2020 = col_double(),
##   ..   Robbery_Rate2020 = col_double(),
##   ..   Shootings_2020 = col_double(),
##   ..   Shootings_Rate2020 = col_double(),
##   ..   Shape__Area = col_double(),
##   ..   Shape__Length = col_double()
##   .. )
```

##3.List the variables in your dataset.

```
names(crimedata)
```

```
##  [1] "OBJECTID"                    "Neighbourhood"
##  [3] "F2020_Population_Projection" "Assault_2020"
##  [5] "Assault_Rate2020"            "AutoTheft_2020"
##  [7] "AutoTheft_Rate2020"          "BreakAndEnter_2020"
##  [9] "BreakAndEnter_Rate2020"      "Robbery_2020"
## [11] "Robbery_Rate2020"            "Shootings_2020"
## [13] "Shootings_Rate2020"          "Shape__Area"
## [15] "Shape__Length"
```

##4.Print the top 15 rows of your dataset.

```
head(crimedata, n=15)
```

```
## # A tibble: 15 x 15
##    OBJECTID Neighbourhood      F2020_Population_P~ Assault_2020 Assault_Rate2020
##       <dbl> <chr>                            <dbl>        <dbl>            <dbl>
##  1        1 Yonge-St.Clair                   14083           23             163.
##  2        2 York University H~               30277          341            1126.
##  3        3 Lansing-Westgate                 18146           97             535.
##  4        4 Yorkdale-Glen Park               17560          156             888.
##  5        5 Stonegate-Queensw~               27410          104             379.
##  6        6 Tam O'Shanter-Sul~               29970          131             437.
```

```
##  7        7 The Beaches                      23364          84             360.
##  8        8 Thistletown-Beaum~               10948          56             512.
##  9        9 Thorncliffe Park                 23518          94             400.
## 10       10 Danforth East York               18427          56             304.
## 11       11 Humewood-Cedarvale               15854          42             265.
## 12       12 Islington-City Ce~               51481         222             431.
## 13       13 Danforth                         10353          65             628.
## 14       14 Rustic                           10756          60             558.
## 15       15 Scarborough Villa~               18116         178             983.
## # ... with 10 more variables: AutoTheft_2020 <dbl>, AutoTheft_Rate2020 <dbl>,
## #   BreakAndEnter_2020 <dbl>, BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>,
## #   Robbery_Rate2020 <dbl>, Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>,
## #   Shape__Area <dbl>, Shape__Length <dbl>
```

##5. Write a user defined function using any of the variables from the data set.

```
Crime_2019_2020 <-function(a,b,c){a*b+c}
Crime_2019_2020(crimedata$Assault_2020,crimedata$AutoTheft_2020,crimedata$BreakAndEnter_2020)
```

```
##   [1]   233 62849  4307 13653  4715  4901  1998  2418  1921   929  1015 33399
##  [13]   495  1274  4122  3880  8198  1600  1628 11026  6175   468 23071  9076
##  [25]  2145  1871  1287   531  3650   459   492   437   643  1685 34645  3347
##  [37]   388  1451  1842 51070 18201 98315  4383  5457  5697   528  3641  1783
##  [49]  2385   247  1134 23863   237   608   958  1797 10378   410   538  1868
##  [61]   662  1193  1726  9230  1207 16283 11073  4914   735  1544  3009   542
##  [73]  6324  3333  7449  2882 16262  4919  1934  4019  8398  3124  2112  6417
##  [85]  4108  2695  1079  1508 12343  3028  1712 54709   354 13127 22787 10150
##  [97]  3592  9488 17272  1891   816  3806   420  2178  3884  1781   864  6648
## [109]  3155   589  6643 33473   136  1962  4093  7844   750  3294 15936   898
## [121]  4785  1575 14578  1478   521  1188  1201  1008  1654  3242 30441  2214
## [133]   782  2073  6080  1208  2255  1257  3228 14925
```

##6.Use data manipulation techniques and filter rows based on any logical criteria that exist in your dataset.

```
over_25000 <- filter(crimedata,crimedata$F2020_Population_Projection >25000)
print(over_25000)
```

```
## # A tibble: 39 x 15
##    OBJECTID Neighbourhood       F2020_Population_~ Assault_2020 Assault_Rate2020
##       <dbl> <chr>                            <dbl>        <dbl>            <dbl>
##  1        2 York University He~              30277          341            1126.
##  2        5 Stonegate-Queensway              27410          104             379.
##  3        6 Tam O'Shanter-Sull~              29970          131             437.
##  4       12 Islington-City Cen~              51481          222             431.
##  5       17 South Riverdale                  30225          212             701.
##  6       23 Church-Yonge Corri~              39279          761            1937.
##  7       24 Clairlea-Birchmount              29302          184             628.
##  8       35 Glenfield-Jane Hei~              33031          279             845.
##  9       40 Waterfront Communi~              87808          757             862.
## 10       41 West Hill                        29669          336            1132.
## # ... with 29 more rows, and 10 more variables: AutoTheft_2020 <dbl>,
## #   AutoTheft_Rate2020 <dbl>, BreakAndEnter_2020 <dbl>,
```

```
## #   BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>, Robbery_Rate2020 <dbl>,
## #   Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>, Shape__Area <dbl>,
## #   Shape__Length <dbl>
```

7.Identify the dependent & independent variables and use reshaping techniques and create a new data frame by joining those variables from your dataset.

```
rate1 <- as.data.frame (crimedata %>% select(6))
rate2 <-as.data.frame (crimedata%>% select(7))
r1_r2 = data.frame(rate1,rate2)
print(r1_r2)
```

```
##     AutoTheft_2020 AutoTheft_Rate2020
## 1                9           63.90684
## 2              184          607.72200
## 3               44          242.47770
## 4               87          495.44420
## 5               45          164.17370
## 6               37          123.45680
## 7               23           98.44205
## 8               43          392.76580
## 9               20           85.04124
## 10              16           86.82911
## 11              23          145.07380
## 12             150          291.36960
## 13               7           67.61325
## 14              21          195.23990
## 15              23          126.95960
## 16              16           66.08293
## 17              38          125.72370
## 18              31          154.45940
## 19              14           84.07903
## 20             110          817.35770
## 21              74          440.63350
## 22              10           68.15704
## 23              30           76.37669
## 24              49          167.22410
## 25              19          107.25370
## 26              15           63.81621
## 27              14           92.47639
## 28               6           41.64642
## 29              36          230.42950
## 30               7           56.57937
## 31              20          162.72070
## 32              13          103.29760
## 33              10           76.49939
## 34              40          294.37740
## 35             124          375.40490
## 36              29          187.45960
## 37              11          103.00590
## 38              13           71.15490
## 39              16           84.18837
```

```
## 40              67           76.30284
## 41              54          182.00820
## 42             396         1066.43700
## 43              48          169.92350
## 44              31          168.41420
## 45              58          248.73490
## 46              15          106.99000
## 47              44          234.76680
## 48              22          182.57260
## 49              20           99.22604
## 50               8           81.49129
## 51              18          145.93810
## 52              56          264.20080
## 53              18          174.30040
## 54              17          157.52410
## 55              37          297.18880
## 56              21           93.14704
## 57              37          106.68970
## 58               8           74.56426
## 59              23          186.61260
## 60              15           93.32421
## 61               7           83.00723
## 62              25          136.24720
## 63              24          163.23200
## 64              71          262.46720
## 65              37          210.00060
## 66              63          156.11450
## 67              61          233.58220
## 68              27           71.18188
## 69              14          196.35340
## 70              18           73.29886
## 71              40          150.97760
## 72              16          114.75290
## 73              27          131.09980
## 74              21          133.46040
## 75              50          211.03280
## 76              25          110.67820
## 77              93          311.41170
## 78              32          128.59150
## 79              40          398.20810
## 80              58          183.43980
## 81              63          229.87670
## 82              30          119.57430
## 83              50          250.95360
## 84              33           68.01460
## 85              49          157.12180
## 86              46          260.94850
## 87              28          211.44840
## 88              10           76.15566
## 89              57          173.93960
## 90              21           86.84145
## 91              17          140.79840
## 92             153          392.49890
## 93              11          130.43990
```

```
## 94              25           76.24276
## 95              81          349.04770
## 96              73          126.17530
## 97              43          219.71280
## 98              63          124.04020
## 99              85          237.61600
## 100             13          102.72620
## 101             19          105.72000
## 102             24           66.05565
## 103              7           84.46965
## 104             45          405.58810
## 105             26          112.49570
## 106             20          159.50230
## 107             16          106.72360
## 108             44          115.01160
## 109             57          464.24500
## 110              7           84.77655
## 111             69          285.61970
## 112             96          165.20680
## 113              4           48.14057
## 114             28          112.90780
## 115             79          303.67100
## 116             61          256.83130
## 117             16          146.22560
## 118             34          115.62270
## 119             78          389.27980
## 120             30          180.88630
## 121             54          285.83530
## 122             23          198.89310
## 123             63          132.49770
## 124             18          139.99070
## 125             17          155.29370
## 126             28          251.21120
## 127             28          104.04670
## 128             23          146.02250
## 129             18          136.53950
## 130             31          164.37770
## 131             40          167.32900
## 132             27          152.18980
## 133             11           81.57816
## 134             37          208.54470
## 135             70          240.00550
## 136             29          172.04560
## 137             26          162.12510
## 138             42          225.47910
## 139             37          239.51320
## 140             52          134.23170
```

#8. Remove missing values in your dataset. •

```
crimedata1 = na.omit(`crimedata`)
```

#9. Identify and remove duplicated data in your dataset.

#Identify -

```
crimedata[duplicated('crimedata')]
```

```
## # A tibble: 140 x 0
```

#Remove -

```
crimedata1[!duplicated(crimedata1$Neighbourhood),]
```

```
## # A tibble: 140 x 15
##    OBJECTID Neighbourhood      F2020_Population_P~ Assault_2020 Assault_Rate2020
##       <dbl> <chr>                            <dbl>        <dbl>            <dbl>
##  1        1 Yonge-St.Clair                   14083           23             163.
##  2        2 York University H~               30277          341            1126.
##  3        3 Lansing-Westgate                 18146           97             535.
##  4        4 Yorkdale-Glen Park               17560          156             888.
##  5        5 Stonegate-Queensw~               27410          104             379.
##  6        6 Tam O'Shanter-Sul~               29970          131             437.
##  7        7 The Beaches                      23364           84             360.
##  8        8 Thistletown-Beaum~               10948           56             512.
##  9        9 Thorncliffe Park                 23518           94             400.
## 10       10 Danforth East York               18427           56             304.
## # ... with 130 more rows, and 10 more variables: AutoTheft_2020 <dbl>,
## #   AutoTheft_Rate2020 <dbl>, BreakAndEnter_2020 <dbl>,
## #   BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>, Robbery_Rate2020 <dbl>,
## #   Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>, Shape__Area <dbl>,
## #   Shape__Length <dbl>
```

#10.Reorder multiple rows in descending order # ex) due to double dataset, using first dataset for descending order.

```
crimedata %>% arrange(desc(F2020_Population_Projection))
```

```
## # A tibble: 140 x 15
##    OBJECTID Neighbourhood       F2020_Population_~ Assault_2020 Assault_Rate2020
##       <dbl> <chr>                            <dbl>        <dbl>            <dbl>
##  1       40 Waterfront Communi~              87808          757             862.
##  2      112 Woburn                           58109          348             599.
##  3       96 Willowdale East                  57856          138             239.
##  4       12 Islington-City Cen~              51481          222             431.
##  5       98 Rouge                            50790          150             295.
##  6       84 L'Amoreaux                       48519          193             398.
##  7      123 Malvern                          47548          231             486.
##  8       66 Dovercourt-Wallace~              40355          256             634.
##  9       23 Church-Yonge Corri~              39279          761            1937.
## 10       92 Downsview-Roding-C~              38981          357             916.
## # ... with 130 more rows, and 10 more variables: AutoTheft_2020 <dbl>,
## #   AutoTheft_Rate2020 <dbl>, BreakAndEnter_2020 <dbl>,
## #   BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>, Robbery_Rate2020 <dbl>,
## #   Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>, Shape__Area <dbl>,
## #   Shape__Length <dbl>
```

#11.Rename some of the column names in your dataset

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.0.5
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:here':
##
##     here
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
## The following object is masked from 'package:ggpubr':
##
##     mutate
```

```
library(dplyr)
renamecrimedata <- crimedata
colnames(renamecrimedata)
```

```
##  [1] "OBJECTID"                    "Neighbourhood"
##  [3] "F2020_Population_Projection" "Assault_2020"
##  [5] "Assault_Rate2020"            "AutoTheft_2020"
##  [7] "AutoTheft_Rate2020"          "BreakAndEnter_2020"
##  [9] "BreakAndEnter_Rate2020"      "Robbery_2020"
## [11] "Robbery_Rate2020"            "Shootings_2020"
## [13] "Shootings_Rate2020"          "Shape__Area"
## [15] "Shape__Length"
```

```
names(renamecrimedata)[names(renamecrimedata) == "F2020_Population_Projection"] <-"Pop_Proj_2020"
names(renamecrimedata)[names(renamecrimedata) == "Neighbourhood"] <- "Region"
view(renamecrimedata)
print(renamecrimedata)
```

```
## # A tibble: 140 x 15
##    OBJECTID Region    Pop_Proj_2020 Assault_2020 Assault_Rate2020 AutoTheft_2020
##       <dbl> <chr>             <dbl>        <dbl>            <dbl>          <dbl>
##  1        1 Yonge-St~         14083           23             163.              9
##  2        2 York Uni~         30277          341            1126.            184
##  3        3 Lansing-~         18146           97             535.             44
##  4        4 Yorkdale~         17560          156             888.             87
##  5        5 Stonegat~         27410          104             379.             45
##  6        6 Tam O'Sh~         29970          131             437.             37
##  7        7 The Beac~         23364           84             360.             23
##  8        8 Thistlet~         10948           56             512.             43
##  9        9 Thorncli~         23518           94             400.             20
## 10       10 Danforth~         18427           56             304.             16
## # ... with 130 more rows, and 9 more variables: AutoTheft_Rate2020 <dbl>,
## #   BreakAndEnter_2020 <dbl>, BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>,
## #   Robbery_Rate2020 <dbl>, Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>,
## #   Shape__Area <dbl>, Shape__Length <dbl>
```

#12.Add new variables in your data frame by using a mathematical function # (for e.g. –multiply an existing column by 2 and add it as a new variable to your data frame) #Top 10 Neighbourhood with highest crimerate

```
Crimerate1 <- crimedata %>% mutate(Total_Crimerate= round(((Assault_2020+AutoTheft_2020
+BreakAndEnter_2020+Robbery_2020+Shootings_2020)/F2020_Population_Projection
)*100000,digits=2))%>% arrange(desc(Total_Crimerate))
head(Crimerate1, 10)
```

```
## # A tibble: 10 x 16
##    OBJECTID Neighbourhood      F2020_Population_P~ Assault_2020 Assault_Rate2020
##       <dbl> <chr>                            <dbl>        <dbl>            <dbl>
##  1      131 Moss Park                        23905          756            3163.
##  2       52 Kensington-Chinat~               21196          423            1996.
##  3       23 Church-Yonge Corr~               39279          761            1937.
##  4       94 Bay Street Corrid~               32790          516            1574.
##  5        2 York University H~               30277          341            1126.
##  6       42 West Humber-Clair~               37133          248             668.
##  7        4 Yorkdale-Glen Park               17560          156             888.
##  8       20 Humber Summit                    13458          100             743.
##  9       61 University                        8433           86            1020.
## 10      100 Cabbagetown-South~               12655          140            1106.
## # ... with 11 more variables: AutoTheft_2020 <dbl>, AutoTheft_Rate2020 <dbl>,
## #   BreakAndEnter_2020 <dbl>, BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>,
## #   Robbery_Rate2020 <dbl>, Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>,
## #   Shape__Area <dbl>, Shape__Length <dbl>, Total_Crimerate <dbl>
```

```
print(head (Crimerate1,10))
```

```
## # A tibble: 10 x 16
##    OBJECTID Neighbourhood      F2020_Population_P~ Assault_2020 Assault_Rate2020
##       <dbl> <chr>                            <dbl>        <dbl>            <dbl>
##  1      131 Moss Park                        23905          756            3163.
##  2       52 Kensington-Chinat~               21196          423            1996.
##  3       23 Church-Yonge Corr~               39279          761            1937.
```

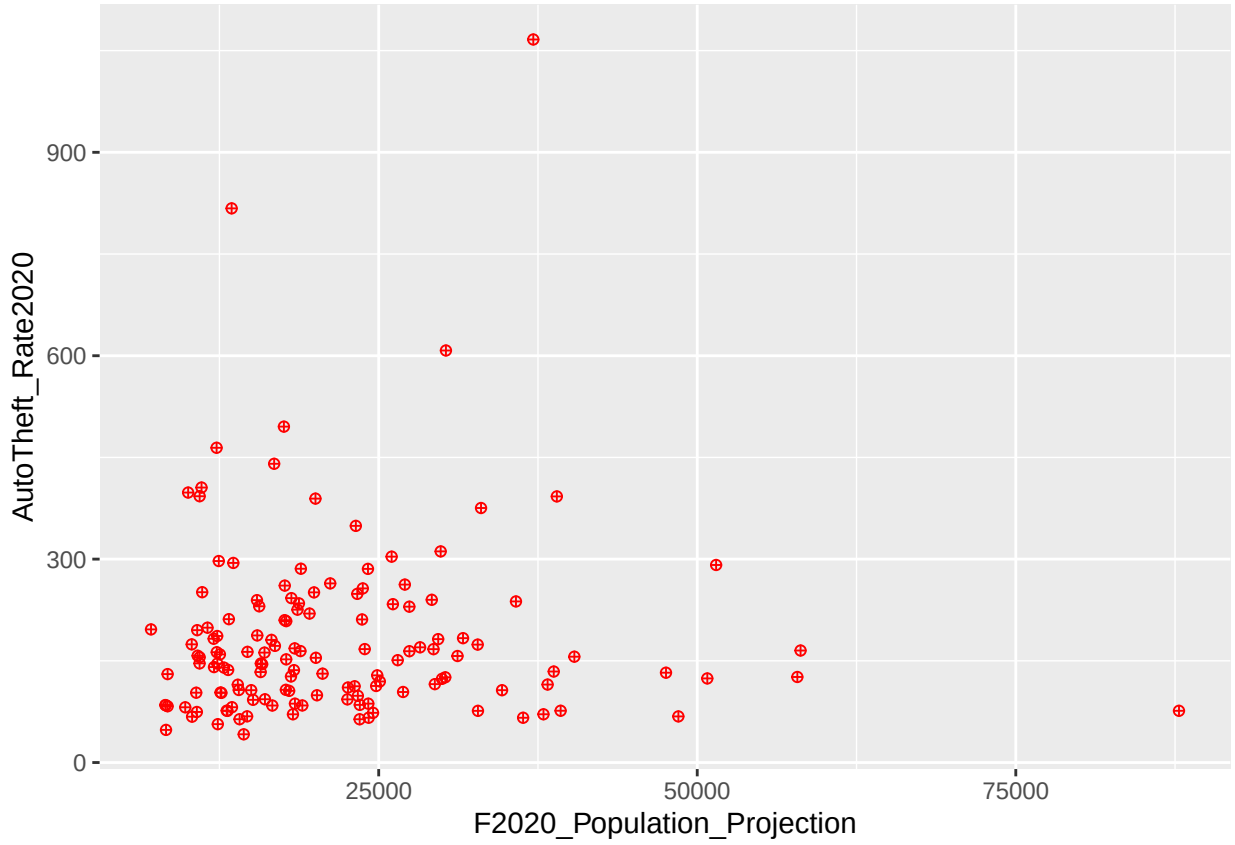
```
##  4       94 Bay Street Corrid~                32790          516            1574.
##  5        2 York University H~                30277          341            1126.
##  6       42 West Humber-Clair~                37133          248             668.
##  7        4 Yorkdale-Glen Park                17560          156             888.
##  8       20 Humber Summit                     13458          100             743.
##  9       61 University                         8433           86            1020.
## 10      100 Cabbagetown-South~                12655          140            1106.
## # ... with 11 more variables: AutoTheft_2020 <dbl>, AutoTheft_Rate2020 <dbl>,
## #   BreakAndEnter_2020 <dbl>, BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>,
## #   Robbery_Rate2020 <dbl>, Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>,
## #   Shape__Area <dbl>, Shape__Length <dbl>, Total_Crimerate <dbl>
```

```
view(head (Crimerate1,10))
```

#13. Plot a scatter plot for any 2 variables in your dataset.

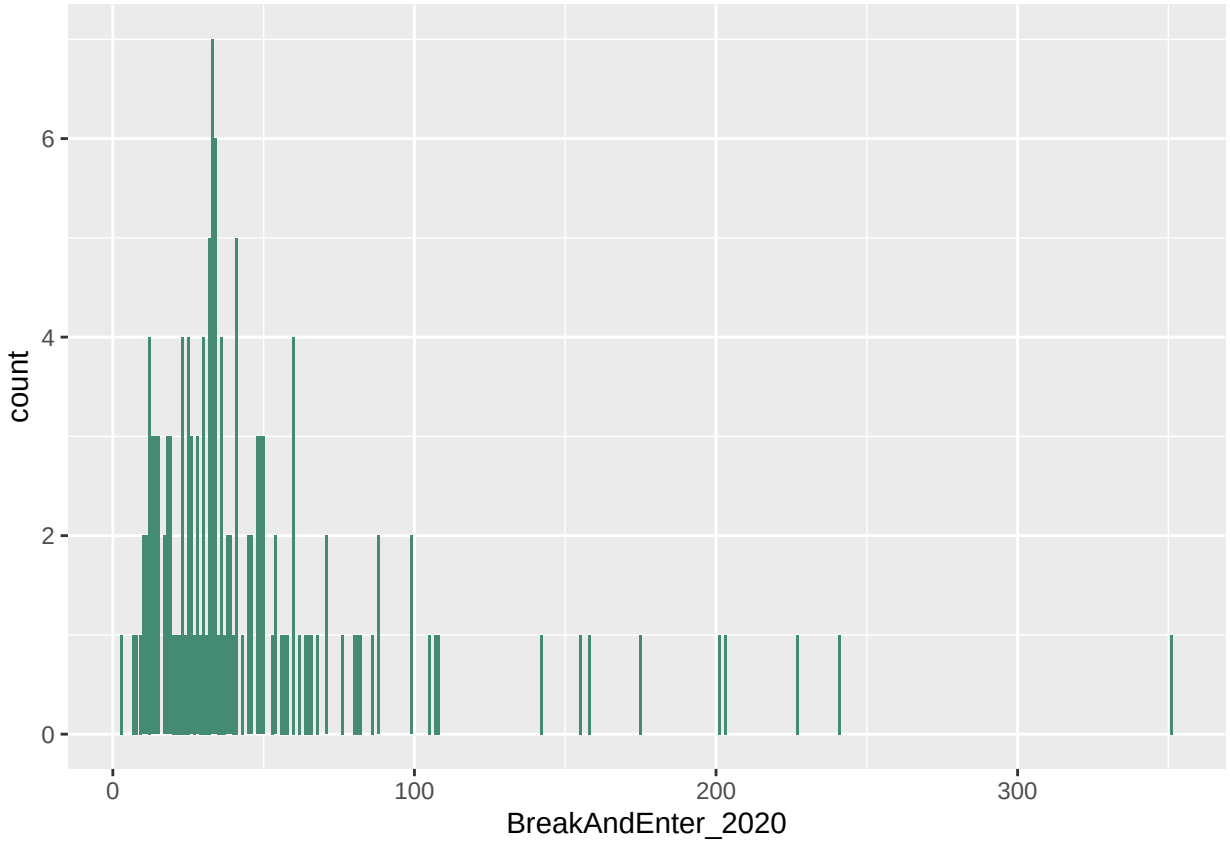
```
library(ggplot2)

ggplot(data = crimedata, aes(x = F2020_Population_Projection, y=  AutoTheft_Rate2020 ))+ geom_point(siz
```



#14. Plot a bar plot for any 2 variables in your dataset. #geom_bar() or geom_col()

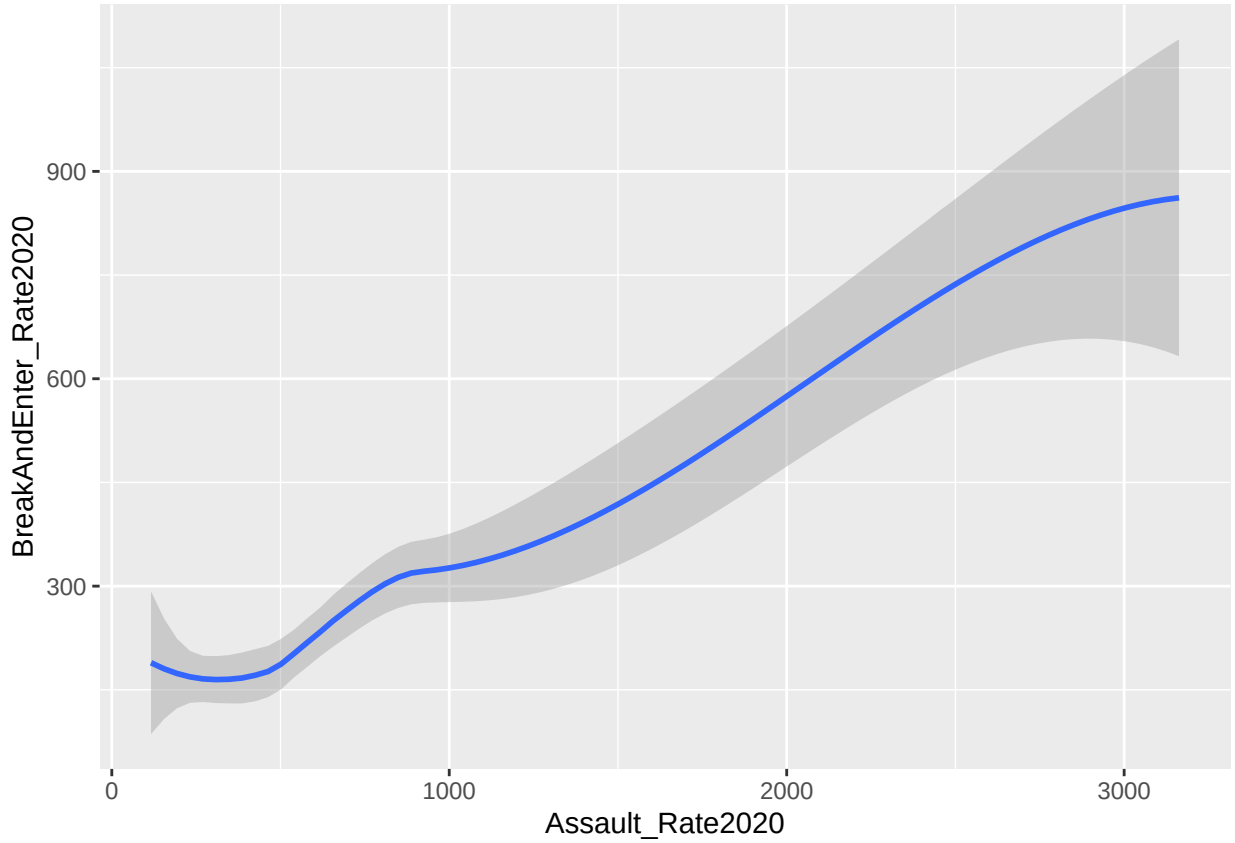
```
library(ggplot2)
ggplot(data = crimedata, aes(x= BreakAndEnter_2020))+geom_bar(fill = "aquamarine4")
```



#15. Find the correlation between any 2 variables by applying least square linear regression model.

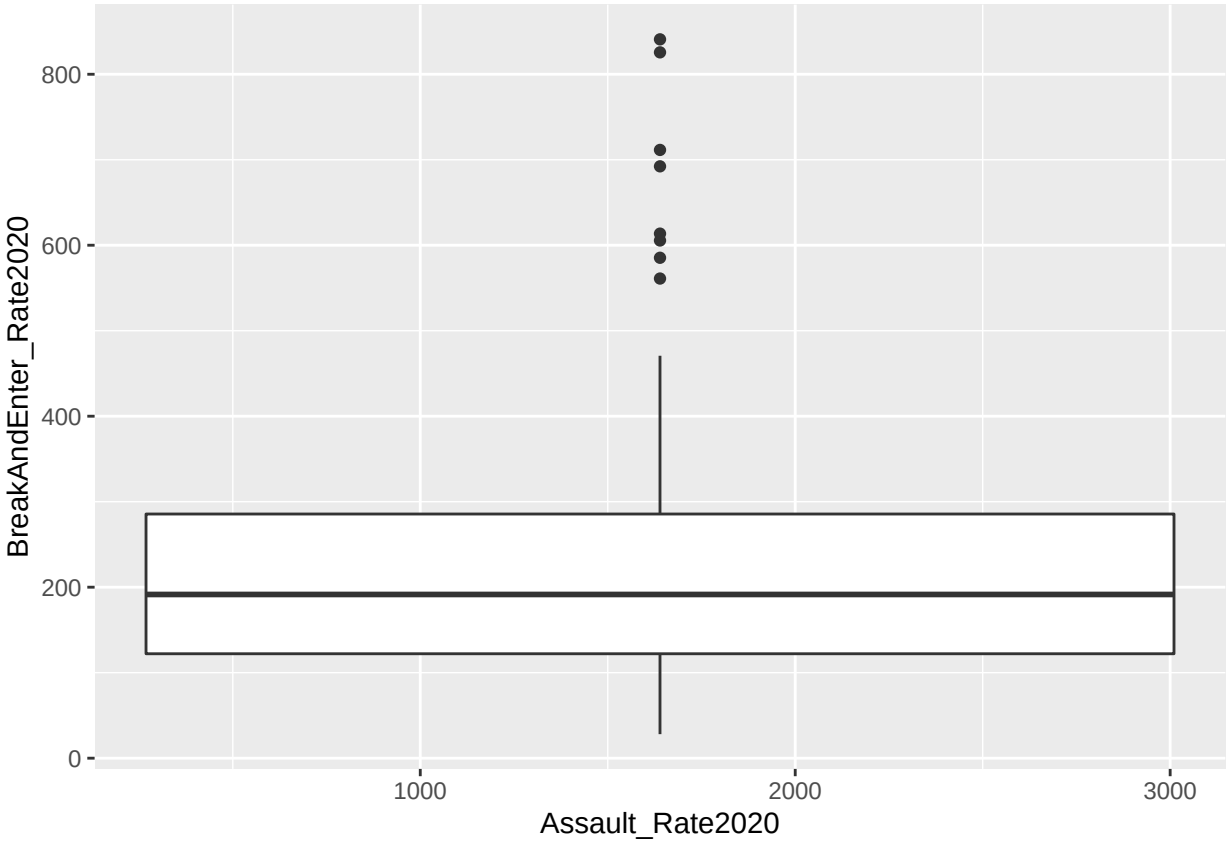
```
ggplot(data = crimedata, aes(x = Assault_Rate2020, y= BreakAndEnter_Rate2020))+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data = crimedata, aes(x = Assault_Rate2020, y= BreakAndEnter_Rate2020))+ geom_boxplot()
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

#16.using any of the numerical variables from the dataset and perform the following statistical functions mean, median, mode and range

#Mean

```
mean(crimedata$Assault_2020)
```

```
## [1] 127.4571
```

#Median

```
median(crimedata$Assault_2020)
```

```
## [1] 87.5
```

#Mode

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

```
v <- c(crimedata$Assault_2020)


result <- getmode(v)
print(result)
```

```
## [1] 42
```

#Range

```
range(crimedata$Assault_2020)
```

```
## [1]  12 761
```

#17.Create a training set using random number generator engine

```
head(crimedata)
```

```
## # A tibble: 6 x 15
##   OBJECTID Neighbourhood      F2020_Population_Pr~ Assault_2020 Assault_Rate2020
##      <dbl> <chr>                             <dbl>        <dbl>            <dbl>
## 1        1 Yonge-St.Clair                    14083           23             163.
## 2        2 York University H~                30277          341            1126.
## 3        3 Lansing-Westgate                  18146           97             535.
## 4        4 Yorkdale-Glen Park                17560          156             888.
## 5        5 Stonegate-Queensw~                27410          104             379.
## 6        6 Tam O'Shanter-Sul~                29970          131             437.
## # ... with 10 more variables: AutoTheft_2020 <dbl>, AutoTheft_Rate2020 <dbl>,
## #   BreakAndEnter_2020 <dbl>, BreakAndEnter_Rate2020 <dbl>, Robbery_2020 <dbl>,
## #   Robbery_Rate2020 <dbl>, Shootings_2020 <dbl>, Shootings_Rate2020 <dbl>,
## #   Shape__Area <dbl>, Shape__Length <dbl>
```

```
h <-runif(nrow(crimedata))
crime<-crimedata[order(h), ]
str(crime)
```

```
## tibble[,15] [140 x 15] (S3: tbl_df/tbl/data.frame)
##  $ OBJECTID                   : num [1:140] 71 30 102 40 43 37 76 114 72 85 ...
##  $ Neighbourhood              : chr [1:140] "High Park-Swansea" "Broadview North" "Mount Pleasant We
##  $ F2020_Population_Projection: num [1:140] 26494 12372 36333 87808 28248 ...
##  $ Assault_2020               : num [1:140] 74 62 152 757 91 35 114 69 33 83 ...
##  $ Assault_Rate2020           : num [1:140] 279 501 418 862 322 ...
##  $ AutoTheft_2020             : num [1:140] 40 7 24 67 48 11 25 28 16 49 ...
##  $ AutoTheft_Rate2020         : num [1:140] 151 56.6 66.1 76.3 169.9 ...
##  $ BreakAndEnter_2020         : num [1:140] 49 25 158 351 15 3 32 30 14 41 ...
##  $ BreakAndEnter_Rate2020     : num [1:140] 184.9 202.1 434.9 399.7 53.1 ...
##  $ Robbery_2020               : num [1:140] 10 3 18 83 17 3 23 18 2 7 ...
##  $ Robbery_Rate2020           : num [1:140] 37.7 24.2 49.5 94.5 60.2 ...
##  $ Shootings_2020             : num [1:140] 2 0 0 9 1 1 7 1 1 2 ...
##  $ Shootings_Rate2020         : num [1:140] 0.755 0 0 1.025 0.354 ...
##  $ Shape__Area                : num [1:140] 5358123 1745969 1343312 13416295 3667762 ...
##  $ Shape__Length              : num [1:140] 10614 7178 5182 18367 11495 ...
```

```
train <- crime[1:50, ]
crime.data1 = na.omit(crimedata)
set.seed(1234)
trainingcrimedata = as.data.frame(crime.data1 %>% sample_frac(0.75,replace = FALSE))
```

#18.Print the summary statistics of your dataset

```
summary(crimedata)
```

```
##     OBJECTID      Neighbourhood      F2020_Population_Projection
##  Min.   :  1.00   Length:140         Min.   : 7130
##  1st Qu.: 35.75   Class :character   1st Qu.:13227
##  Median : 70.50   Mode  :character   Median :18378
##  Mean   : 70.50                      Mean   :21729
##  3rd Qu.:105.25                      3rd Qu.:26598
##  Max.   :140.00                      Max.   :87808
##   Assault_2020    Assault_Rate2020 AutoTheft_2020   AutoTheft_Rate2020
##  Min.   : 12.00   Min.   : 116.2   Min.   :  4.00   Min.   :  41.65
##  1st Qu.: 51.75   1st Qu.: 320.4   1st Qu.: 18.00   1st Qu.: 102.94
##  Median : 87.50   Median : 484.4   Median : 28.50   Median : 151.58
##  Mean   :127.46   Mean   : 563.3   Mean   : 40.17   Mean   : 183.20
##  3rd Qu.:150.00   3rd Qu.: 664.9   3rd Qu.: 49.25   3rd Qu.: 226.58
##  Max.   :761.00   Max.   :3162.5   Max.   :396.00   Max.   :1066.44
##  BreakAndEnter_2020 BreakAndEnter_Rate2020  Robbery_2020    Robbery_Rate2020
##  Min.   :  3.00     Min.   : 28.09         Min.   :  1.00   Min.   :  7.133
##  1st Qu.: 23.00     1st Qu.:122.14         1st Qu.:  8.00   1st Qu.: 50.666
##  Median : 34.00     Median :191.50         Median : 16.00   Median : 76.242
##  Mean   : 49.31     Mean   :227.14         Mean   : 19.71   Mean   : 87.842
##  3rd Qu.: 54.50     3rd Qu.:285.58         3rd Qu.: 24.00   3rd Qu.:112.382
##  Max.   :351.00     Max.   :840.83         Max.   :138.00   Max.   :543.819
##  Shootings_2020 Shootings_Rate2020  Shape__Area       Shape__Length
##  Min.   : 0.0   Min.   : 0.0000    Min.   :  424197   Min.   : 2574
##  1st Qu.: 0.0   1st Qu.: 0.0000    1st Qu.: 1861853   1st Qu.: 6372
##  Median : 2.0   Median : 0.7678    Median : 3290879   Median : 8962
##  Mean   : 3.3   Mean   : 1.4427    Mean   : 4589400   Mean   :10138
##  3rd Qu.: 5.0   3rd Qu.: 2.1048    3rd Qu.: 5402335   3rd Qu.:11900
##  Max.   :34.0   Max.   :10.2934    Max.   :37534495   Max.   :43081
```

#19.Conclusion: We can Conclude that The top crime prone neighbourhood in Toronto in 2020 was Moss Part followed by Kensington-Chinatown and Church-Yonge Corridor. Breaking and entering was much less compared to assaults.

tinytex::install_tinytex()