Statistical Data Anlysis of Traffic Accidents in Calgary

Mary Sarafraz, Ryan Leeson, Shora Dehkordi, Guarav Kumar

October 2019

```
#Call all the libraries
library(data.table)
library(stringi)
library(dplyr)
library(ggplot2)
library(mosaic)
library(binom)
library(mdsr)
library(tinytex)
#To be able to preview HTML
library(tidyverse)
library(tidyr)
library(stringr)
library(readr)
library(rmarkdown)
library(Matrix)
library(purrr)
library(markdown)
library(knitr)
options(scipen = 999)
```

#Introduction:

#Purpose:

#Data #Part 1: Data Wrangling

Reading Data:

```
trafficAccidentDF =
read.csv("C:\\Users\\shora\\Desktop\\Calgary_Traffic_Accident.csv")

head(trafficAccidentDF,10)

##    X                                                     INCIDENT.INFO
## 1  0 Westbound McKnight Boulevard approaching John Laurie Boulevard NW
## 2  1                                          20 Avenue at 8 Street NW
## 3  2                                      Sunridge Way at 36 Street NE
## 4  3                 Westbound Stoney approaching Shaganappi Trail NW.
## 5  4         Southbound Nose Hill Drive approaching Crowchild Trail NW
## 6  5                          Southbound Macleod Trail at 94 Avenue SE
## 7  6                                 Anderson Road at Acadia Drive SE.
## 8  7                                      Centre Street at 7 Street NE
## 9  8                  Eastbound Anderson Road approaching 14 Street SW
## 10 9                                        130 Avenue at 48 Street SE
##                             DESCRIPTION            START_DT
## 1                  2 vehicle incident. 2016-12-09 16:46:32
## 2                  2 vehicle incident. 2016-12-09 16:58:23
## 3  There is an incident involving LRT. 2016-12-09 17:14:08
## 4              Multi vehicle incident. 2016-12-09 17:16:08
## 5              Multi vehicle incident. 2016-12-09 17:38:05
## 6                  2 vehicle incident. 2016-12-09 17:49:59
## 7              Multi vehicle incident. 2016-12-09 17:55:04
## 8             Single vehicle incident. 2016-12-09 18:08:09
## 9                  2 vehicle incident. 2016-12-09 18:20:14
## 10                 2 vehicle incident. 2016-12-09 18:36:21
##               MODIFIED_DT QUADRANT Longitude Latitude
## 1  12/09/2016 05:16:54 PM       NW -114.0833 51.09732
## 2  12/09/2016 05:16:54 PM       NW -114.0814 51.07054
## 3  12/09/2016 05:16:54 PM       NE -113.9849 51.06730
## 4  12/09/2016 05:16:53 PM       NW -114.1479 51.15274
## 5  12/09/2016 05:55:52 PM       NW -114.2032 51.11968
## 6  12/09/2016 05:55:52 PM       SW -114.0717 50.96863
## 7  12/09/2016 05:55:52 PM       SE -114.0441 50.94833
## 8  12/09/2016 06:21:31 PM       NW -114.0625 51.05888
## 9  12/09/2016 06:21:31 PM       SW -114.0973 50.95059
## 10 12/09/2016 06:56:02 PM       SE -113.9657 50.93209
##                                 location Count
## 1   (51.09731625733, -114.083317961464)     1
## 2  (51.070538552637, -114.081377719156)     1
## 3   (51.067298691023, -113.98493374196)     1
## 4  (51.152736445625, -114.147933369876)     1
## 5   (51.11968378497, -114.203240843777)     1
## 6  (50.968632228523, -114.071706940396)     1
## 7  (50.948331405788, -114.044139639421)     1
## 8   (51.058877561027, -114.06253407232)     1
## 9  (50.950590701047, -114.097344384191)     1
## 10 (50.932090880143, -113.965739722774)     1
##                                                       id DAY MONTH YEAR
## 1  2016-12-09T16:46:3251.0973162573297-114.083317961464   9    12 2016
## 2  2016-12-09T16:58:2351.0705385526371-114.081377719156   9    12 2016
## 3   2016-12-09T17:14:0851.0672986910231-113.98493374196   9    12 2016
## 4  2016-12-09T17:16:0851.1527364456253-114.147933369876   9    12 2016
## 5  2016-12-09T17:38:0551.1196837849704-114.203240843777   9    12 2016
## 6  2016-12-09T17:49:5950.9686322285233-114.071706940396   9    12 2016
## 7  2016-12-09T17:55:0450.9483314057881-114.044139639421   9    12 2016
## 8   2016-12-09T18:08:0951.0588775610272-114.06253407232   9    12 2016
## 9  2016-12-09T18:20:1450.9505907010468-114.097344384191   9    12 2016
## 10 2016-12-09T18:36:2150.9320908801432-113.965739722774   9    12 2016
##    HOUR MINUTE SECOND PEDESTRIAN SINGLE_VEHICLE TWO_VEHICLE MULTI_VEHICLE
## 1    16     46     32      False          False        True         False
## 2    16     58     23      False          False        True         False
## 3    17     14      8      False          False       False         False
## 4    17     16       8      False          False       False          True
## 5    17     38       5      False          False       False          True
## 6    17     49      59      False          False        True         False
## 7    17     55       4      False          False       False          True
## 8    18      8       9      False           True       False         False
## 9    18     20      14      False          False        True         False
## 10   18     36      21      False          False        True         False
```

```
tail(trafficAccidentDF,10)
```

```
##           X
## 14824 14823
## 14825 14824
## 14826 14825
## 14827 14826
## 14828 14827
## 14829 14828
## 14830 14829
## 14831 14830
## 14832 14831
## 14833 14832
## 
INCIDENT.INFO
## 14824  Stoney Trail between Country Hills Boulevard and McKnight Boulevard 
NE
## 14825                       Northbound Deerfoot Trail at McKnight Boulevard 
NE
## 14826                                                11 Street and 9 Avenue 
SE
## 14827  Stoney Trail between Country Hills Boulevard and McKnight Boulevard 
NE
## 14828                             Nose Hill Drive and John Laurie Boulevard 
NW
## 14829                            Eastbound McKnight Boulevard and 52 Street 
NE
## 14830                 Northbound MacLeod Trail approaching Lake Fraser Gate 
SE
## 14831                                     Heritage Drive at Blackfoot Trail 
SE
## 14832                                    Heritage Drive and Blackfoot Trail 
SE
## 14833                                                5 Street and 57 Avenue 
SW
## 
DESCRIPTION
## 14824 Multi-vehicle incident.      The road is closed northbound, 
southbound has reopened.
## 14825                                                              Multi-
vehicle incident.
## 14826                                                                Two
```

```
vehicle incident.
## 14827                                      Multi-vehicle incident.
Northbound has reopened
## 14828                                                    Traffic signals
are flashing red.
## 14829                                        Two vehicle incident.
Blocking the left lane.
## 14830                                     Two vehicle incident.   Blocking
the right lane.
## 14831                         Traffic signals are flashing red. Crews have
been dispatched
## 14832                                                                Two
vehicle incident.
## 14833                                                                Two
vehicle incident.
##                 START_DT MODIFIED_DT QUADRANT Longitude Latitude
## 14824 2019-03-23 10:59:41                      -113.9209 51.14071
## 14825 2019-03-23 16:57:52                      -114.0394 51.09270
## 14826 2019-03-23 18:50:27                      -114.0368 51.04219
## 14827 2019-03-23 10:59:41                      -113.9209 51.14056
## 14828 2019-03-24 09:23:04                      -114.1949 51.12742
## 14829 2019-03-24 09:52:42                      -113.9589 51.09598
## 14830 2019-03-24 11:49:37                      -114.0690 50.93826
## 14831 2019-03-24 13:38:18                      -114.0502 50.98082
## 14832 2019-03-24 15:28:32                      -114.0501 50.98068
## 14833 2019-03-24 15:51:54                      -114.0764 51.00246
##                                  location Count
## 14824  (51.140706179316, -113.92094503013)     1
## 14825 (51.092698562796, -114.039432010784)     1
## 14826  (51.042194586533, -114.03676844764)     1
## 14827 (51.140556046825, -113.920935569254)     1
## 14828 (51.127420345846, -114.194895145175)     1
## 14829 (51.095980099367, -113.958863881354)     1
## 14830 (50.938255268697, -114.069003267344)     1
## 14831 (50.980824068937, -114.050172754793)     1
## 14832 (50.980681880897, -114.050100257277)     1
## 14833 (51.002463427192, -114.076439295549)     1
##                                                    id DAY MONTH YEAR HOUR
## 14824  2019032310594151.1407061793158-113.92094503013  23     3 2019   10
## 14825 2019032316575251.0926985627956-114.039432010784  23     3 2019   16
## 14826  2019032318502751.0421945865328-114.03676844764  23     3 2019   18
## 14827 2019032310594151.1405560468253-113.920935569254  23     3 2019   10
## 14828 2019032409230451.1274203458462-114.194895145175  24     3 2019    9
## 14829  2019032409524251.095980099367-113.958863881354  24     3 2019    9
## 14830 2019032411493750.9382552686966-114.069003267344  24     3 2019   11
## 14831 2019032413381850.9808240689374-114.050172754793  24     3 2019   13
## 14832  2019032415283250.980681880897-114.050100257277  24     3 2019   15
## 14833 2019032415515451.0024634271915-114.076439295549  24     3 2019   15
##       MINUTE SECOND PEDESTRIAN SINGLE_VEHICLE TWO_VEHICLE MULTI_VEHICLE
## 14824     59     41      False          False       False          True
## 14825     57     52       False          False       False          True
## 14826     50     27       False          False        True         False
## 14827     59     41       False          False       False          True
## 14828     23      4       False          False       False         False
## 14829     52     42       False          False        True         False
## 14830     49     37       False          False        True         False
## 14831     38     18       False          False       False         False
## 14832     28     32       False          False        True         False
## 14833     51     54       False          False        True         False
```

1.Adding TYPE column for Type of accident.

```
#converting values to logical values
trafficAccidentDF[,"PEDESTRIAN"] <-
as.logical(trafficAccidentDF[,"PEDESTRIAN"] )
trafficAccidentDF[,"SINGLE_VEHICLE"] <-
as.logical(trafficAccidentDF[,"SINGLE_VEHICLE"] )
trafficAccidentDF[,"TWO_VEHICLE"] <-
as.logical(trafficAccidentDF[,"TWO_VEHICLE"] )
trafficAccidentDF[,"MULTI_VEHICLE"] <-
as.logical(trafficAccidentDF[,"MULTI_VEHICLE"] )


for (i in 1: nrow(trafficAccidentDF)) {

 if (trafficAccidentDF[i,"MULTI_VEHICLE"]==TRUE)  {
trafficAccidentDF[i,"TYPE"] = "MULTI_VEHICLE"}
  else if (trafficAccidentDF[i,"TWO_VEHICLE"]==TRUE)  {
trafficAccidentDF[i,"TYPE"] = "TWO_VEHICLE"}
  else if (trafficAccidentDF[i,"SINGLE_VEHICLE"]==TRUE)  {
trafficAccidentDF[i,"TYPE"] = "SINGLE_VEHICLE"}
  else if (trafficAccidentDF[i,"PEDESTRIAN"]==TRUE)  {
trafficAccidentDF[i,"TYPE"] = "PEDESTRIAN"}
  else {  trafficAccidentDF[i,"TYPE"] = "OTHERS"}

}
```

2. Adding Season column based on date of accident

```
#Adding season
for (i in 1: nrow(trafficAccidentDF)) {

 if (trafficAccidentDF[i,"MONTH"] %in% c(1,2,3))  {
trafficAccidentDF[i,"SEASON"] = "WINTER"}
  else if (trafficAccidentDF[i,"MONTH"] %in% c(4,5,6))  {
trafficAccidentDF[i,"SEASON"] = "SPRING"}
  else if (trafficAccidentDF[i,"MONTH"] %in% c(7,8,9))  {
trafficAccidentDF[i,"SEASON"] = "SUMMER"}
```

```
  else {  trafficAccidentDF[i,"SEASON"] = "FALL"}
}
```

3. Adding HOURTYPE column for type of hour(Rush-hour, NotRush-hour) based on time of accident

```
#Adding rushhour
for (i in 1: nrow(trafficAccidentDF)) {

 if (trafficAccidentDF[i,"HOUR"] %in% c(7,8))  {
trafficAccidentDF[i,"HOURTYPE"] = "RUSHHOUR"}
  else if (trafficAccidentDF[i,"HOUR"] %in% c(15,16,17))  {
trafficAccidentDF[i,"HOURTYPE"] = "RUSHHOUR"}
  else {  trafficAccidentDF[i,"HOURTYPE"] = "NOTRUSHHOUR"}
}
```

4. Replace empty QUADRANT

```
## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated
## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated

## Warning in `[<-.factor`(`*tmp*`, iseq, value = " N"): invalid factor
level,
## NA generated
```

Filtering 24 datapoints that don't have true quadrant:

```
unique(trafficAccidentDF$QUADRANT)

## [1] NW   NE   SW   SE   <NA>
## Levels:  NE NW SE SW

trafficAccidentDF = filter(trafficAccidentDF, (!is.na(QUADRANT )
|trimws(QUADRANT) != ""))
head(trafficAccidentDF,4)

##   X                                                      INCIDENT.INFO
## 1 0 Westbound McKnight Boulevard approaching John Laurie Boulevard NW
## 2 1                                           20 Avenue at 8 Street NW
## 3 2                                       Sunridge Way at 36 Street NE
## 4 3                   Westbound Stoney approaching Shaganappi Trail NW.
##                           DESCRIPTION            START_DT
## 1                 2 vehicle incident. 2016-12-09 16:46:32
## 2                 2 vehicle incident. 2016-12-09 16:58:23
## 3 There is an incident involving LRT. 2016-12-09 17:14:08
## 4             Multi vehicle incident. 2016-12-09 17:16:08
##              MODIFIED_DT QUADRANT Longitude Latitude
## 1 12/09/2016 05:16:54 PM       NW -114.0833 51.09732
## 2 12/09/2016 05:16:54 PM       NW -114.0814 51.07054
## 3 12/09/2016 05:16:54 PM       NE -113.9849 51.06730
## 4 12/09/2016 05:16:53 PM       NW -114.1479 51.15274
##                                location Count
## 1  (51.09731625733, -114.083317961464)     1
## 2 (51.070538552637, -114.081377719156)     1
## 3  (51.067298691023, -113.98493374196)     1
## 4 (51.152736445625, -114.147933369876)     1
##                                                      id DAY MONTH YEAR HOUR
## 1 2016-12-09T16:46:3251.0973162573297-114.083317961464   9    12 2016   16
## 2 2016-12-09T16:58:2351.0705385526371-114.081377719156   9    12 2016   16
## 3  2016-12-09T17:14:0851.0672986910231-113.98493374196   9    12 2016   17
## 4 2016-12-09T17:16:0851.1527364456253-114.147933369876   9    12 2016   17
##   MINUTE SECOND PEDESTRIAN SINGLE_VEHICLE TWO_VEHICLE MULTI_VEHICLE
## 1     46     32      FALSE          FALSE        TRUE         FALSE
## 2     58     23      FALSE          FALSE        TRUE         FALSE
## 3     14      8      FALSE          FALSE       FALSE         FALSE
## 4     16      8      FALSE          FALSE       FALSE          TRUE
##            TYPE SEASON HOURTYPE QUADRANT1
## 1   TWO_VEHICLE   FALL RUSHHOUR        NW
## 2   TWO_VEHICLE   FALL RUSHHOUR        NW
## 3        OTHERS   FALL RUSHHOUR        NE
## 4 MULTI_VEHICLE   FALL RUSHHOUR        NW
```

```
unique(trafficAccidentDF$QUADRANT)
```

```
## [1] NW NE SW SE
## Levels:  NE NW SE SW
```

5.Filtering only data for Year 2017 and 2018

6.Select only required columns

```
trafficAccident_wantedColumn_DF = select(trafficAccidentDF,"Count","MONTH"
,"YEAR", "QUADRANT", "TYPE","DAY","SEASON","HOURTYPE")

trafficAccident_wantedColumn_DF = filter(trafficAccident_wantedColumn_DF ,
(YEAR == "2017" |YEAR =="2018"   ))
```

Part 2: Data visualization and Preliminary Observations

```
nrow(trafficAccident_wantedColumn_DF)
```

```
## [1] 11840
```

```
head(trafficAccident_wantedColumn_DF,4)
```

```
##   Count MONTH YEAR QUADRANT           TYPE DAY SEASON    HOURTYPE
## 1     1     2 2017       SE SINGLE_VEHICLE   8 WINTER    RUSHHOUR
## 2     1     2 2017       SE  MULTI_VEHICLE   8 WINTER NOTRUSHHOUR
## 3     1     2 2017       NE    TWO_VEHICLE   8 WINTER NOTRUSHHOUR
## 4     1     2 2017       SE    TWO_VEHICLE   8 WINTER NOTRUSHHOUR
```

```
tail(trafficAccident_wantedColumn_DF,4)
```

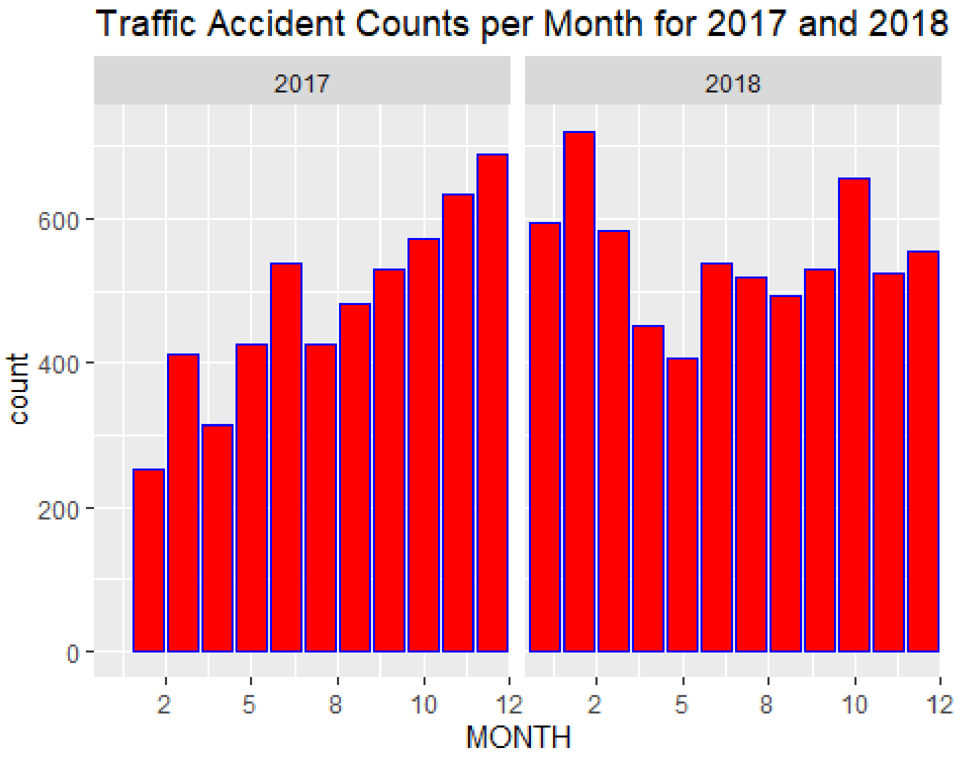
```
##       Count MONTH YEAR QUADRANT          TYPE DAY SEASON    HOURTYPE
## 11837     1    12 2018       NW   TWO_VEHICLE  31   FALL    RUSHHOUR
## 11838     1    12 2018       SE MULTI_VEHICLE  31   FALL NOTRUSHHOUR
## 11839     1    12 2018       NW   TWO_VEHICLE  31   FALL NOTRUSHHOUR
## 11840     1    12 2018       SW   TWO_VEHICLE  31   FALL NOTRUSHHOUR
```

```
require(scales)
```

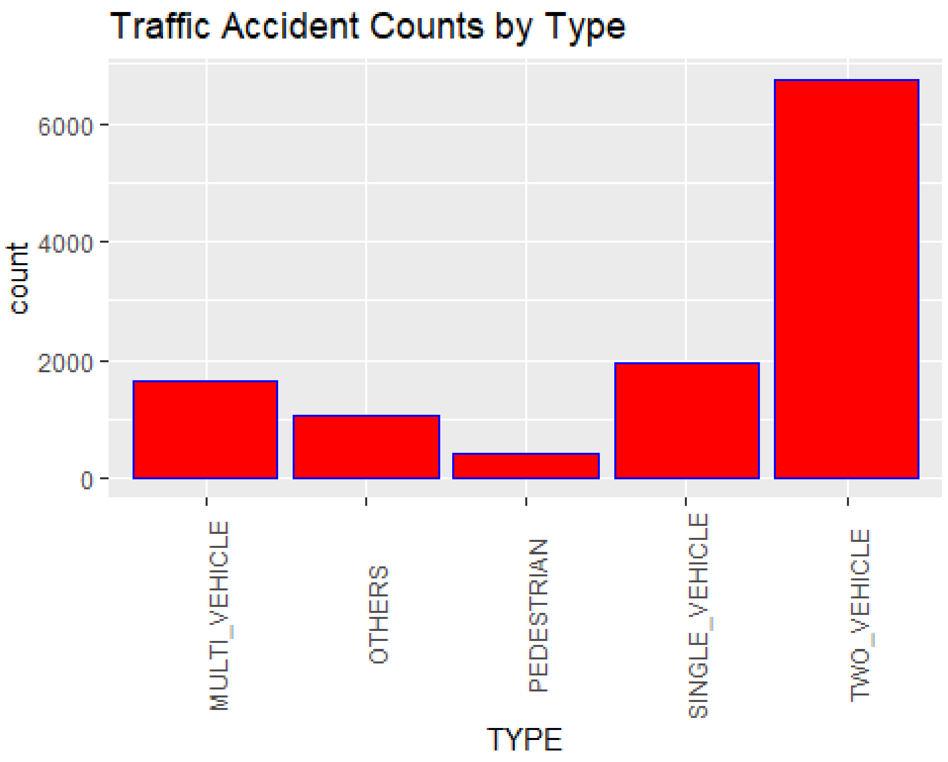
```
## Loading required package: scales
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

## The following object is masked from 'package:mosaic':
##
##     rescale
```

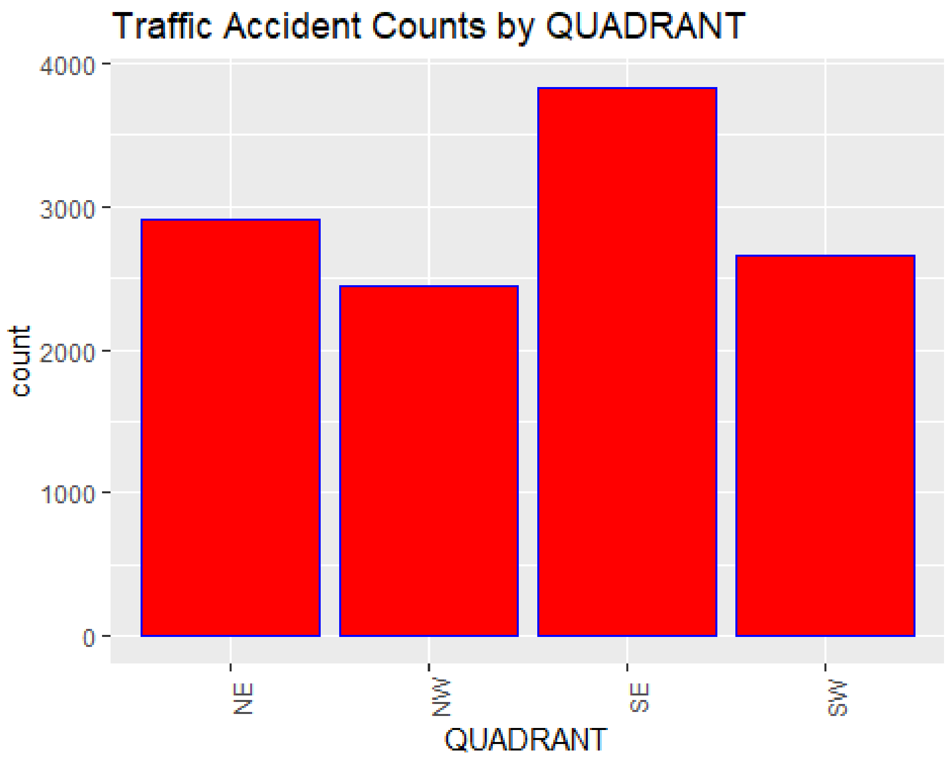
```
ggplot(data= trafficAccident_wantedColumn_DF , aes(x= MONTH ) )+
geom_bar(col= 'blue' , fill='red')+ coord_cartesian(xlim = c(1,
12))+scale_x_continuous(labels = comma) + facet_wrap(~YEAR) +
ggtitle("Traffic Accident Counts per Month for 2017 and 2018")
```



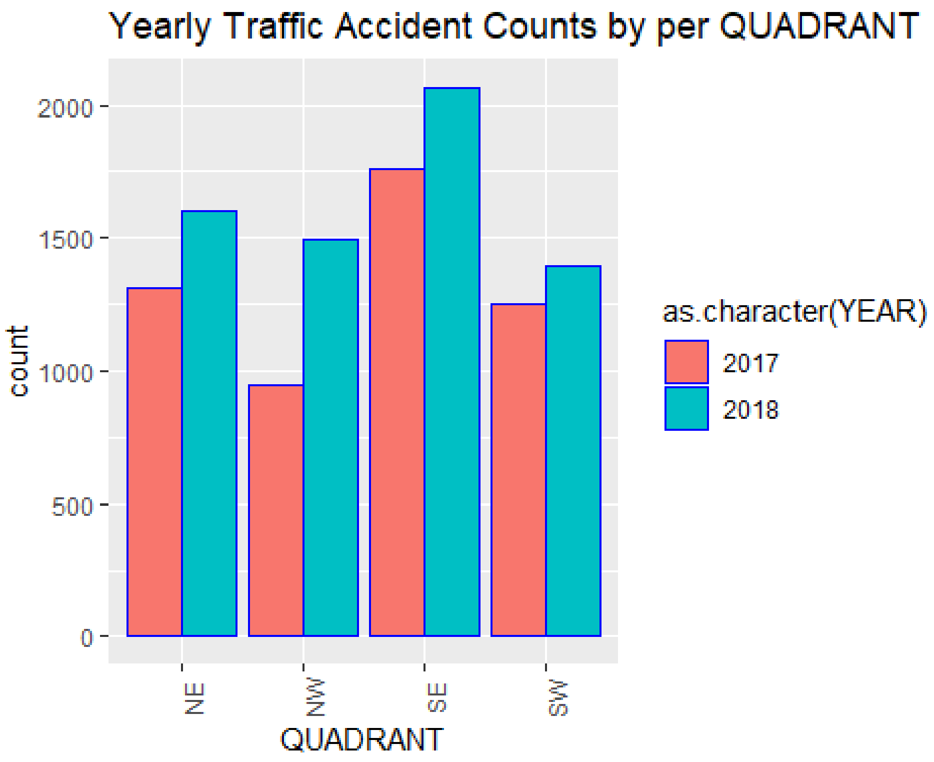
```
# First we try to see if there is any pattern between two years, we see there
is no pattern in two years

ggplot(data= trafficAccident_wantedColumn_DF , aes(x= TYPE ))+ geom_bar(col=
'blue',fill='red' ,position="dodge")+ ggtitle("Traffic Accident Counts by
Type")+ theme(axis.text.x = element_text(angle = 90))
```



```r
# Two vehicle accident is the most common accident type following by Multi
vehicle

ggplot(data= trafficAccident_wantedColumn_DF , aes(x= QUADRANT ))+
geom_bar(col= 'blue',fill='red' ,position="dodge") + theme(axis.text.x =
element_text(angle = 90)) + ggtitle("Traffic Accident Counts by QUADRANT")
```

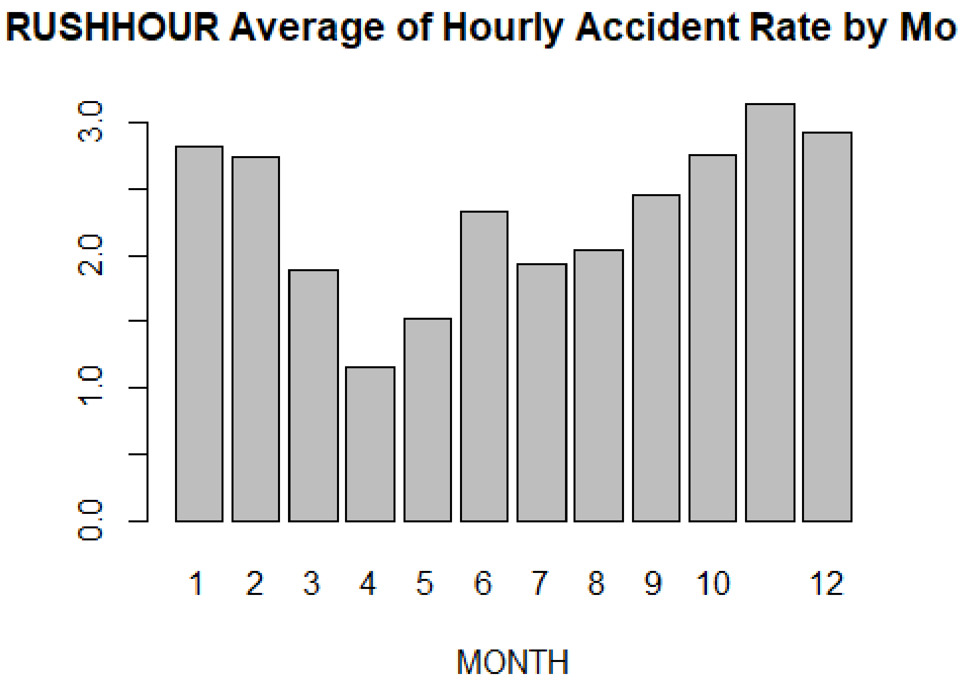


```r
# SE has highest number of accidents

ggplot(data= trafficAccident_wantedColumn_DF , aes(x= QUADRANT ,
fill=as.character(YEAR)))+ geom_bar(col= 'blue' ,position="dodge")+
theme(axis.text.x = element_text(angle = 90))+ ggtitle("Yearly Traffic
Accident Counts by per QUADRANT")
```

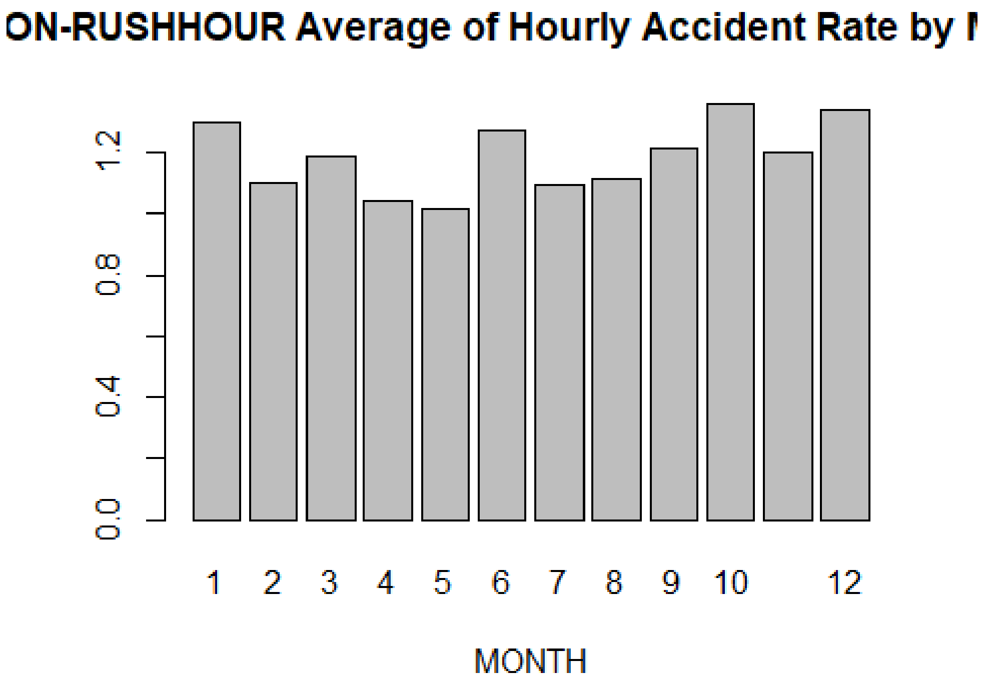


```
# 2018 has more accidents in every QUADRANT, NW has the most increase in
accident

#this is our aggregation ,level(Month,day,HourType), to find number of
accident per hour for rush hour and not rush hour
trafficAccident_Perhour_agg =
aggregate(trafficAccident_wantedColumn_DF$Count, by= list(
MONTH=trafficAccident_wantedColumn_DF$MONTH,DAY=trafficAccident_wantedColumn_
DF$DAY, HOURTYPE=trafficAccident_wantedColumn_DF$HOURTYPE  ), FUN=sum,
na.rm=T)

# Adding HourlyRate column
for ( i in 1:nrow(trafficAccident_Perhour_agg)){
trafficAccident_Perhour_agg[i,"HourlyRate"] = if
(trafficAccident_Perhour_agg[i,"HOURTYPE"] =="RUSHHOUR")
{trafficAccident_Perhour_agg[i,"x"] / 5 } # "5" number of rushhour hours
else {trafficAccident_Perhour_agg[i,"x"] / 19 } # "19" number of not rushhour
hours
}
# Caclulating traffic accident hourly average houly rate per month
trafficAccident_AvgRate = aggregate(trafficAccident_Perhour_agg$HourlyRate,
by= list( MONTH=trafficAccident_Perhour_agg$MONTH,
HOURTYPE=trafficAccident_Perhour_agg$HOURTYPE  ), FUN=mean, na.rm=T)
nrow(trafficAccident_AvgRate)

## [1] 24
```

```
#To show the graph nice, I just assumed that Jan 2017 has same rate as Jan
2018 because we are missing Jan 2017 data
for (i in 1: nrow(trafficAccident_AvgRate)){
  if (trafficAccident_AvgRate[i,"MONTH"] == 1)  {trafficAccident_AvgRate
[i,3] = trafficAccident_AvgRate [i,3]*2}
}

# Sorting data by month to show it in the bar graph
trafficAccident_AvgRate =
trafficAccident_AvgRate[order(trafficAccident_AvgRate$MONTH),]

barplot(filter(trafficAccident_AvgRate,HOURTYPE=="RUSHHOUR")$x, xlab =
"MONTH", main ="RUSHHOUR Average of Hourly Accident Rate by Month" ,names.arg
=c(seq(1,12) ))
```



```
barplot(filter(trafficAccident_AvgRate,HOURTYPE=="NOTRUSHHOUR")$x, xlab =
"MONTH",main ="NON-RUSHHOUR Average of Hourly Accident Rate by Month" ,
names.arg =c(seq(1,12)))
```



```
# Rush hour accidents show a more pronounced dependents to months, in April
we have less than 2 accident per hour and in November we have 4 accident per
hour
# Average hourly accident rate is more consistent for not rush hour and we
have 6 accident each 5 hours (1.2  per hour)
```

Part 3: Statistical Analysis

1. Do more traffic accidents occur during rush hour time or non-rush hour times?

#Avrage monthly accident per hour for rush hour vs non rush hour

```
trafficAccident_Perhour_agg1 =
aggregate(trafficAccident_wantedColumn_DF$Count, by= list(
TYPE=trafficAccident_wantedColumn_DF$TYPE,
HOURTYPE=trafficAccident_wantedColumn_DF$HOURTYPE  ), FUN=sum, na.rm=T)

# Adding HourlyRate column
# for ( i in 1:nrow(trafficAccident_Perhour_agg1)){
# trafficAccident_Perhour_agg1[i,"HourlyRate"] = if
(trafficAccident_Perhour_agg1[i,"HOURTYPE"] =="RUSHHOUR")
{trafficAccident_Perhour_agg1[i,"x"] / 5 } # "5" number of rushhour hours
# else {trafficAccident_Perhour_agg1[i,"x"] / 19 } # "19" number of not
rushhour hours
# }
```

```
totalRushHour = sum(filter(trafficAccident_Perhour_agg1,HOURTYPE=="RUSHHOUR"
)[,3])
totalRushHour
```

```
## [1] 3983
```

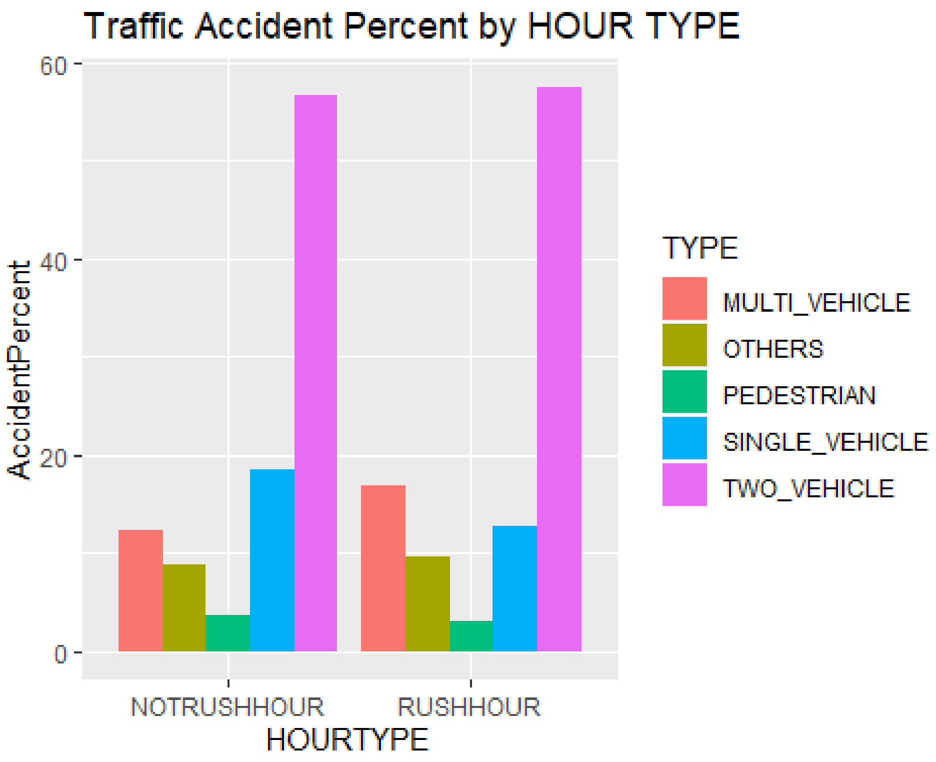
```
totalnotRushHour =
sum(filter(trafficAccident_Perhour_agg1,HOURTYPE=="NOTRUSHHOUR" )[,3])
totalnotRushHour
```

```
## [1] 7857
```

```
# Adding HourlyRate percent column
for ( i in 1:nrow(trafficAccident_Perhour_agg1)){
trafficAccident_Perhour_agg1[i,"AccidentPercent"] = if
(trafficAccident_Perhour_agg1[i,"HOURTYPE"] =="RUSHHOUR")
{round(trafficAccident_Perhour_agg1[i,"x"] / totalRushHour *100,2)}
else {round(trafficAccident_Perhour_agg1[i,"x"] / totalnotRushHour *100,2) }
}

head(trafficAccident_Perhour_agg1,3)
```

```
##             TYPE    HOURTYPE   x AccidentPercent
## 1 MULTI_VEHICLE NOTRUSHHOUR 971           12.36
## 2        OTHERS NOTRUSHHOUR 689            8.77
## 3    PEDESTRIAN NOTRUSHHOUR 292            3.72
```

```
ggplot(data= trafficAccident_Perhour_agg1 , aes(x= HOURTYPE,fill=TYPE ))+
geom_col(aes(y=AccidentPercent),position="dodge")+ ggtitle("Traffic Accident
Percent by HOUR TYPE")
```

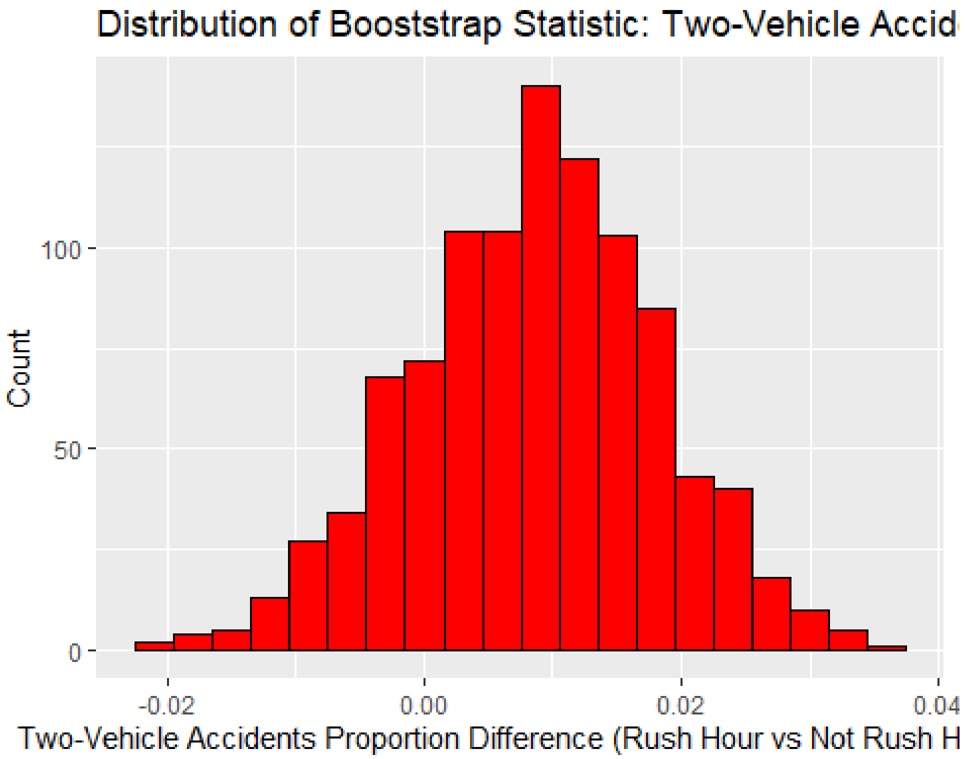


1) Do more two vehicle traffic incidents occur during rush hour time or non-rush hour times?

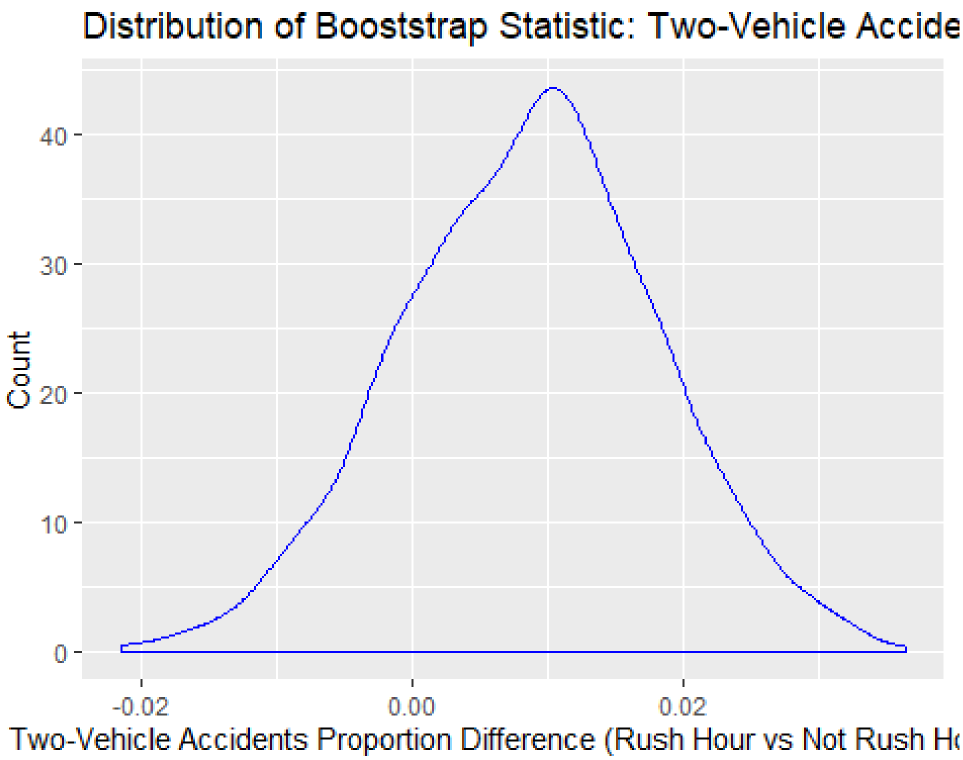
```
###

# AS discussed we investigate our hypothesis based on the proportion of
TWO_VEHICLE incidents to the total number of incidents for rush hour and not
rush hour. We use a prop.test
# Also, rush hour has been considered 7-9am and 3-6 pm for Calgary

# H0:  p_2v_RH = p_2v_NRH
# Ha:  p_2v_RH > p_2v_NRH

N_total_RH = nrow(filter(trafficAccident_wantedColumn_DF, HOURTYPE==
'RUSHHOUR'))  # Compute number of incidents happened during rush hour
N_total_notRH = nrow(filter(trafficAccident_wantedColumn_DF, HOURTYPE==
'NOTRUSHHOUR')) # Compute number of incidents happened out of  rush hour

N_2v_RH = nrow(filter(trafficAccident_wantedColumn_DF, HOURTYPE== 'RUSHHOUR',
TYPE == 'TWO_VEHICLE')) # Compute number of two-vehicle incidents happened
during rush hour
N_2v_notRH = nrow(filter(trafficAccident_wantedColumn_DF, HOURTYPE==
'NOTRUSHHOUR', TYPE == 'TWO_VEHICLE')) # Compute number of two-vehicle
incidents happened out of rush hour
```

```r
# Perform a prop-test

prop.test(c(N_2v_RH,N_2v_notRH),c(N_total_RH,N_total_notRH),alternative =
"greater", correct= FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(N_2v_RH, N_2v_notRH) out of c(N_total_RH, N_total_notRH)
## X-squared = 0.79213, df = 1, p-value = 0.1867
## alternative hypothesis: greater
## 95 percent confidence interval:
##  -0.007257613  1.000000000
## sample estimates:
##    prop 1    prop 2
## 0.5746924 0.5661194

# Inference
# As the Pvalue, P(Zobs > sqrt(8.0255) = 0.002316 which is less than 0.05
confidence level, we can reject H0 in favor of the alternative. In other
words, the proportion of two-vehicle incidents that happen during rush hour
is more than the proportion of two-vehicle incidents that take place outside
of rush hour period.

Inc_total_RH = filter(trafficAccident_wantedColumn_DF, HOURTYPE== 'RUSHHOUR')
Inc_total_notRH = filter(trafficAccident_wantedColumn_DF, HOURTYPE==
'NOTRUSHHOUR')

# Bootstrap method to compare two populations

nsamples = 1000
p_hat_RH_2v = numeric(nsamples)
p_hat_notRH_2v = numeric(nsamples)
p_hat_diff_2v = numeric(nsamples)


for(i in 1:nsamples){

  # Computing the bootstrap statistic for two-vehicle accidents
  sample_RH_2v = resample(Inc_total_RH$TYPE, n =N_2v_RH)
  p_hat_RH_2v[i] = table(sample_RH_2v)[5]/N_total_RH  # Creating two-vehicle
proportion vector for  rush hours
  sample_notRH_2v = resample(Inc_total_notRH$TYPE, n =N_2v_notRH)
  p_hat_notRH_2v[i] = table(sample_notRH_2v)[5]/N_total_notRH  # Creating
two-vehicle proportion vector for  not rush hours
  p_hat_diff_2v[i] = p_hat_RH_2v[i] - p_hat_notRH_2v[i]

 }
```

```
boot_difference = data.frame(p_hat_RH_2v,p_hat_notRH_2v, p_hat_diff_2v)

head(boot_difference)

##   p_hat_RH_2v p_hat_notRH_2v p_hat_diff_2v
## 1   0.5679136      0.5703195  -0.002405827
## 2   0.5633944      0.5550465   0.008347971
## 3   0.5862415      0.5616648   0.024576769
## 4   0.5885011      0.5701922   0.018308944
## 5   0.5807181      0.5679012   0.012816817
## 6   0.5774542      0.5665012   0.010952971

tail(boot_difference)

##      p_hat_RH_2v p_hat_notRH_2v p_hat_diff_2v
## 995    0.5709264      0.5761741 -0.0052476749
## 996    0.5910118      0.5724831  0.0185286641
## 997    0.5779563      0.5591193  0.0188370576
## 998    0.5794627      0.5761741  0.0032886043
## 999    0.5782074      0.5690467  0.0091606714
## 1000   0.5638966      0.5645921 -0.0006955231

favstats(p_hat_diff_2v, data =boot_difference)

##          min          Q1      median         Q3        max       mean
##  -0.02144672 0.002109425 0.009157188 0.01457389 0.03633324 0.00856246
##           sd    n missing
##  0.009446102 1000       0

# Confidence interval for two-vehicle accidents
qdata(p_hat_diff_2v,c(.025,.975), data =boot_difference )

##          quantile     p
## 2.5%  -0.01010291 0.025
## 97.5%  0.02670910 0.975

# The 95% confidence interval using bootstarpping :  0.00797 < p_hat_diff_2v
< 0.04379 which is always positive and doesn't include zero. We can infer
that 95% of the times the proportion of all two-vehicle incidents that happen
during rush hour is " more" than the proportion of all two-vehicle incidents
that happen out of rush hour period.

# Density plot and Histogram of Boostrap Statistic: Population Difference for
two-vehicle


ggplot(data=boot_difference, aes(x = p_hat_diff_2v)) +
geom_histogram(fill='red', col='black',binwidth = 0.003) + xlab("Two-Vehicle
Accidents Proportion Difference (Rush Hour vs Not Rush Hour )") +
ylab("Count") + ggtitle("Distribution of Booststrap Statistic: Two-Vehicle
Accidents Proportion Difference ")
```

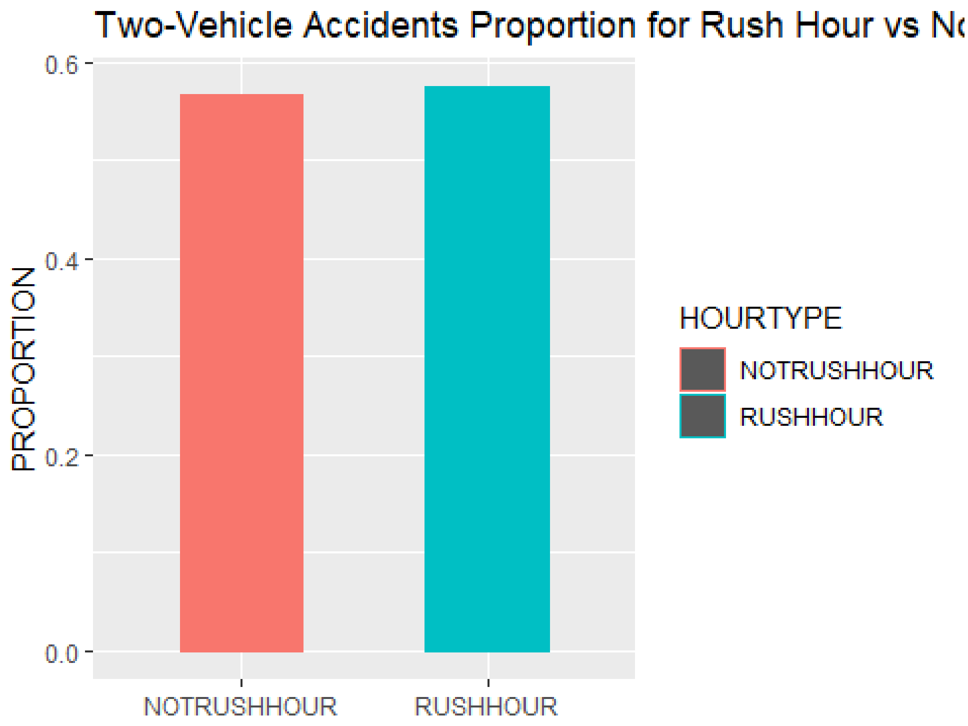


```
ggplot(data=boot_difference, aes(x = p_hat_diff_2v)) + geom_density(
col='blue') + xlab("Two-Vehicle Accidents Proportion Difference (Rush Hour vs
Not Rush Hour )") + ylab("Count") + ggtitle("Distribution of Booststrap
Statistic: Two-Vehicle Accidents Proportion Difference ")
```



```
for (i in 1: nrow(trafficAccident_wantedColumn_DF)) {

 if (trafficAccident_wantedColumn_DF[i,"HOURTYPE"] == "RUSHHOUR")
   {
   trafficAccident_wantedColumn_DF[i,"PROPORTION"] =
trafficAccident_wantedColumn_DF[i,"Count"]/nrow(filter (
trafficAccident_wantedColumn_DF, HOURTYPE == "RUSHHOUR"))
   }
  else
  {
    trafficAccident_wantedColumn_DF[i,"PROPORTION"] =
trafficAccident_wantedColumn_DF[i,"Count"]/nrow(filter (
trafficAccident_wantedColumn_DF, HOURTYPE == "NOTRUSHHOUR"))
  }

}

head(trafficAccident_wantedColumn_DF,5)

##   Count MONTH YEAR QUADRANT           TYPE DAY SEASON    HOURTYPE
## 1     1     2 2017       SE SINGLE_VEHICLE   8 WINTER    RUSHHOUR
## 2     1     2 2017       SE  MULTI_VEHICLE   8 WINTER NOTRUSHHOUR
## 3     1     2 2017       NE    TWO_VEHICLE   8 WINTER NOTRUSHHOUR
## 4     1     2 2017       SE    TWO_VEHICLE   8 WINTER NOTRUSHHOUR
## 5     1     2 2017       NW  MULTI_VEHICLE   8 WINTER NOTRUSHHOUR
##    PROPORTION
## 1 0.000251067
## 2 0.000127275
## 3 0.000127275
## 4 0.000127275
## 5 0.000127275
```

```
ggplot(data= filter(trafficAccident_wantedColumn_DF, TYPE == "TWO_VEHICLE") ,
aes(x= HOURTYPE ,y = PROPORTION, color = HOURTYPE))+ geom_bar(stat =
"identity", width = 0.5) + ggtitle("Two-Vehicle Accidents Proportion for Rush
Hour vs Not Rush Hour")+ xlab(" ")
```



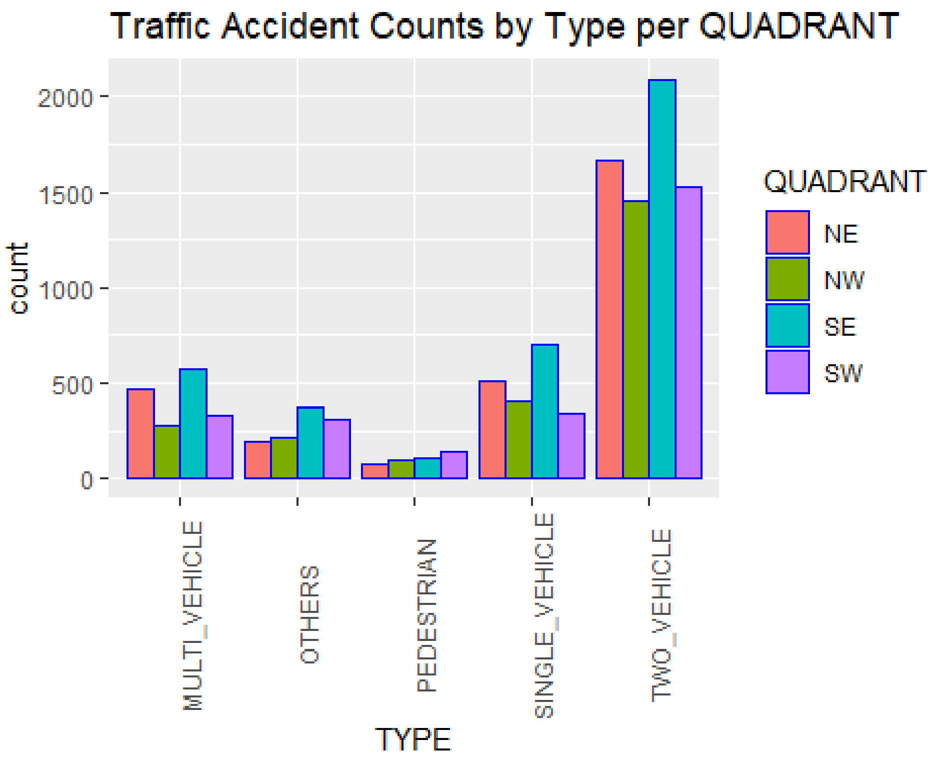
2) Does the type of accident depend on the quadrant of the city?

In this part we investigated if accident types and quadrants are related. In other words, do some accident types happen more frequently in some quadrants? Since both variables are categorical, we used Chi-Squared Test of independence and a test of two proportions.

The dependency between the accident types and quadrants can be in different forms. We choose to visualize the data in a few different forms to investigate this dependency further.

Traffic accident counts by type per quadrant shows that SW has the highest number of pedestrian accident and SE has the highest number of two-vehicle accidents.

```
ggplot(data= trafficAccident_wantedColumn_DF , aes(x= TYPE ,fill=QUADRANT))+
geom_bar(col= 'blue' ,position="dodge")+ theme(axis.text.x =
element_text(angle = 90))+ ggtitle("Traffic Accident Counts by Type per
QUADRANT")
```

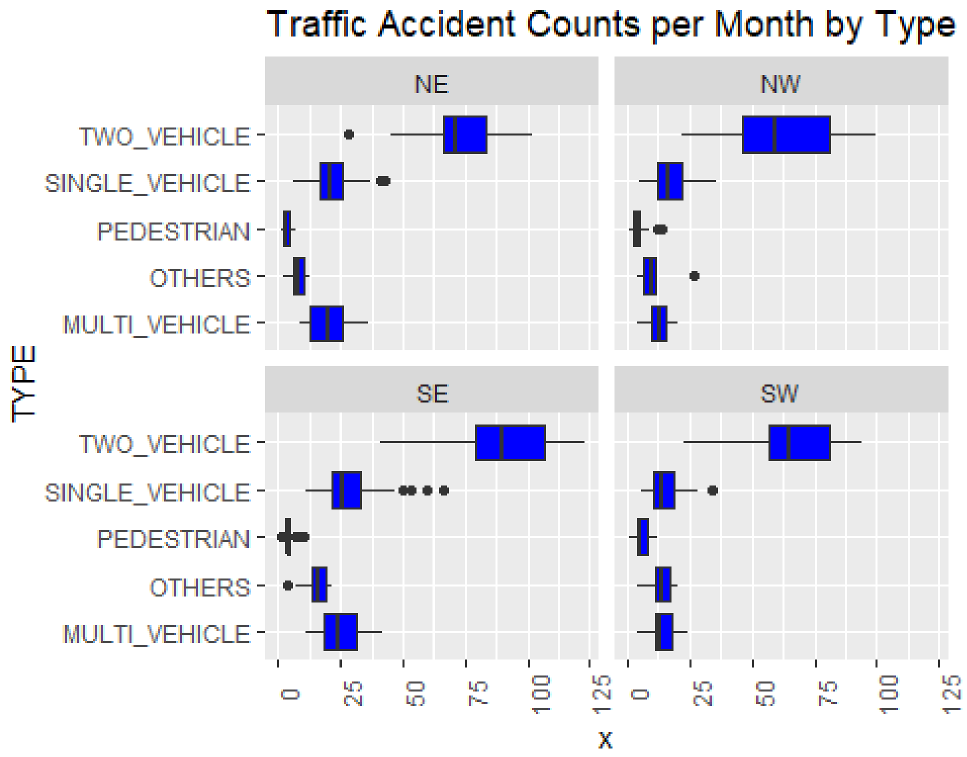


```
# SW has hieghest number of pedestrain accident and SE has highest number
Two_Vehicle accidents
```

We calculated the number of accidents per month per quadrant, and type. This helped us investigate monthly variations in accident types by quadrant. The boxplot of monthly accidents by type per quadrant shows that NE has the least amount of variation by month for two-vehicle accidents and NW has the most.

Also, east quadrants (SE and NE) have more variations in monthly multi-vehicle accidents compared to the west quadrants.

```
#Agregate level: year,month, quadrant , and Type
trafficAccident_agg = aggregate(trafficAccident_wantedColumn_DF$Count, by=
list(YEAR=trafficAccident_wantedColumn_DF$YEAR,
MONTH=trafficAccident_wantedColumn_DF$MONTH,
QUADRANT=trafficAccident_wantedColumn_DF$QUADRANT ,
TYPE=trafficAccident_wantedColumn_DF$TYPE), FUN=sum, na.rm=T)
head(trafficAccident_agg,4)

##   YEAR MONTH QUADRANT          TYPE  x
## 1 2018     1       NE MULTI_VEHICLE 13
## 2 2017     2       NE MULTI_VEHICLE  9
## 3 2018     2       NE MULTI_VEHICLE 26
## 4 2017     3       NE MULTI_VEHICLE 12

ggplot(data= trafficAccident_agg, aes( x=TYPE, y=x))+
geom_boxplot(fill='blue')+facet_wrap(~QUADRANT) +theme(axis.text.x =
element_text(angle = 90))+coord_flip() + ggtitle("Traffic Accident Counts per Month by Type per QUADRANT ")
```



```
#One of the things this shows is that NE has least amount of variation by month for two vehicle accidents and NW has the most
# East quadrants have more variation in monthly multi vehicle accident compared to the west
#explain how you have aggregated the data and what are people looking at
```

Based on above visualization we see that type of accident is dependent to the quadrant. For example two vehicle accidents are more common in SE while pedestrain accident are more common in SW.

To statistically validate this, we defined our statistical hypotheses as following.

Ho: Types of accidents and the quadrants of the city are Independent.

HA: Types of accidents and the quadrants of the city are dependent.

The test statistics in the categorical bivarite data is χ2obs.

First step to apply test of independence is creating contingency table.

```
contTableTypeQuadrant = tally(~ QUADRANT+TYPE    , margins=TRUE, data =
trafficAccident_wantedColumn_DF   )
# Out put of tally had a extra row of zero that I removed it
contTableTypeQuadrant = contTableTypeQuadrant[2:6,]
contTableTypeQuadrant
```

```
##          TYPE
## QUADRANT MULTI_VEHICLE OTHERS PEDESTRIAN SINGLE_VEHICLE TWO_VEHICLE Total
##    NE              467    190         74            514        1669  2914
##    NW              277    210        101            409        1449  2446
##    SE              571    367        102            701        2089  3830
##    SW              332    309        136            343        1530  2650
##    Total          1647   1076        413           1967        6737 11840
```

Chi-squared test has a condition that the Eij≥5. As we can see in the contingency table, this condition is met.

```
xchisq.test(contTableTypeQuadrant, correct=FALSE)

## 
##  Pearson's Chi-squared test
## 
## data:  x
## X-squared = 144.48, df = 20, p-value < 0.00000000000000022
## 
##    467.00     190.00      74.00     514.00    1669.00    2914.00
## (  405.35) (  264.82) (  101.65) (  484.11) ( 1658.08) ( 2914.00)
## [ 9.376] [21.139] [ 7.519] [ 1.846] [ 0.072] [ 0.000]
## < 3.06>  <-4.60>  <-2.74>  < 1.36>  < 0.27>  < 0.00>
## 
##    277.00     210.00     101.00     409.00    1449.00    2446.00
## (  340.25) (  222.29) (   85.32) (  406.36) ( 1391.78) ( 2446.00)
## [11.758] [ 0.679] [ 2.881] [ 0.017] [ 2.352] [ 0.000]
## <-3.43>  <-0.82>  < 1.70>  < 0.13>  < 1.53>  < 0.00>
## 
##    571.00     367.00     102.00     701.00    2089.00    3830.00
## (  532.77) (  348.06) (  133.60) (  636.28) ( 2179.28) ( 3830.00)
## [ 2.743] [ 1.030] [ 7.473] [ 6.582] [ 3.740] [ 0.000]
## < 1.66>  < 1.01>  <-2.73>  < 2.57>  <-1.93>  < 0.00>
## 
##    332.00     309.00     136.00     343.00    1530.00    2650.00
## (  368.63) (  240.83) (   92.44) (  440.25) ( 1507.86) ( 2650.00)
## [ 3.639] [19.298] [20.530] [21.482] [ 0.325] [ 0.000]
## <-1.91>  < 4.39>  < 4.53>  <-4.63>  < 0.57>  < 0.00>
## 
##   1647.00    1076.00     413.00    1967.00    6737.00   11840.00
## ( 1647.00) ( 1076.00) (  413.00) ( 1967.00) ( 6737.00) (11840.00)
## [ 0.000] [ 0.000] [ 0.000] [ 0.000] [ 0.000] [ 0.000]
## < 0.00>  < 0.00>  < 0.00>  < 0.00>  < 0.00>  < 0.00>
## 
## key:
##  observed
##  (expected)
##  [contribution to X-squared]
##  <Pearson residual>
```

From this table we can see the Expected Count per quadrant by type (second row of each quadrant in the table). For example, the observed count for multi-vehicle accidents in NE is 467 while the expected count is 405. This expected value was calculated by taking the row total for NE (2914) times the column total for multi-vehicle accidents and then dividing the result by the sample size (11840). The same procedure was conducted for each cell. The general concept is that if the expected and observed counts are not too different, then the two variables are not related (i.e. are independent). In contrast, if the observed counts were much different than expected, we would conclude that there is an association (i.e. dependence) between the two variables.

The third row of each quadrant in the table shows the Chi-square contribution of each accident type to the test statistics for that quadrant. For example, the chi-square contribution of 9.38 for multi-vehicle accidents in NE was calculated by taking the squared difference between the Observed Count (467) and the Expected Count (405) then dividing by the Expected Count ((467- 405) ^2 /405). The chi-square of all accident type/quadrant cells (sum of the all cells in the third rows of all quadrants) adds up to the chi-square test statistic of 144.48.

The p-value for the Pearson's Chi-Square test summarizes these calculations in an interpretable fashion. This p-value of accident types being independent from the quadrants is 2E-16 (practically zero) which is below 0.05, so we declare that the result is statistically significant. From this result, we infer that the "types of accidents" and "quadrants" are dependent.

Now that we found out these categories are related, we go one step further to compare the risks of each type of accidents in each quadrant. We pursue this by "Test of Equal" or "Given Proportions".

H0: The null hypothesis is that the four populations (NW, NE, SW, SE) in which the multi-vehicle accidents have happened, have the same true proportion of total accidents.

HA: The alternative is that this proportion is different in at least one of the populations.

```
contTableTypeQuadrant
```

```
##         TYPE
## QUADRANT MULTI_VEHICLE OTHERS PEDESTRIAN SINGLE_VEHICLE TWO_VEHICLE Total
##    NE              467    190         74            514        1669  2914
##    NW              277    210        101            409        1449  2446
##    SE              571    367        102            701        2089  3830
##    SW              332    309        136            343        1530  2650
##    Total          1647   1076        413           1967        6737 11840
```

```
#prop test between MULTI_VEHICLE in each Quadrant
prop.test(contTableTypeQuadrant[1:4,"MULTI_VEHICLE"],contTableTypeQuadrant[1:
4,"Total"], alternative = "two.sided")
```

```
## 
##  4-sample test for equality of proportions without continuity
##  correction
```

```
##
## data:  contTableTypeQuadrant[1:4, "MULTI_VEHICLE"] out of
contTableTypeQuadrant[1:4, "Total"]
## X-squared = 31.962, df = 3, p-value = 0.000000533
## alternative hypothesis: two.sided
## sample estimates:
##    prop 1    prop 2    prop 3    prop 4
## 0.1602608 0.1132461 0.1490862 0.1252830
```

Based on the p-value= 0.000000533, we can infer that proportion of multi-vehicle accidents is different in at least one of the four quadrants. It also provides us with the proportion of success (multi-vehicle accident happening) in each quadrant. The following table summarizes the proportion (%) of accidents happening in each quadrant by type. We can create proportion of success for all scenarios as below:

```
propotionTableTypeQuadrant = contTableTypeQuadrant
percentTableTypeQuadrant = contTableTypeQuadrant
for (i in 1:5){
propotionTableTypeQuadrant[i,] =
contTableTypeQuadrant[i,]/contTableTypeQuadrant[i,6]
percentTableTypeQuadrant[i,] =
round((contTableTypeQuadrant[i,]/contTableTypeQuadrant[i,6]*100),1)
}
propotionTableTypeQuadrant

##          TYPE
## QUADRANT MULTI_VEHICLE     OTHERS PEDESTRIAN SINGLE_VEHICLE  TWO_VEHICLE
##    NE       0.16026081 0.06520247 0.02539465     0.17638984  0.57275223
##    NW       0.11324612 0.08585446 0.04129191     0.16721177  0.59239575
##    SE       0.14908616 0.09582245 0.02663185     0.18302872  0.54543081
##    SW       0.12528302 0.11660377 0.05132075     0.12943396  0.57735849
##    Total    0.13910473 0.09087838 0.03488176     0.16613176  0.56900338
##          TYPE
## QUADRANT      Total
##    NE    1.00000000
##    NW    1.00000000
##    SE    1.00000000
##    SW    1.00000000
##    Total 1.00000000

percentTableTypeQuadrant

##          TYPE
## QUADRANT MULTI_VEHICLE OTHERS PEDESTRIAN SINGLE_VEHICLE TWO_VEHICLE Total
##    NE             16.0    6.5        2.5           17.6        57.3 100.0
##    NW             11.3    8.6        4.1           16.7        59.2 100.0
##    SE             14.9    9.6        2.7           18.3        54.5 100.0
##    SW             12.5   11.7        5.1           12.9        57.7 100.0
##    Total          13.9    9.1        3.5           16.6        56.9 100.0
```

The proportion/percentage table shows that more than 50% of accidents in all quadrants are two-vehicle accidents. The second most common accidents type in all areas are single-vehicle accidents followed by multi-vehicle accidents.

Additionally, we can compare the risks by using proportion/percentage table. We define risk as a bad outcome (accident happening) and it can be expressed either as the proportion or percentage of a group that experiences the outcome. For example, risk of a pedestrian accident in SW is 5.1% (Row 4, column 4). This means that 5 percent of all accidents in SW involves a pedestrian. As another example we can say that 3.5 percent of all accidents in all areas (Row 5, column 4) involves a pedestrian.

Relative Risk and Percent increased risk are another two measure that can be used to compare the risk of a particular outcome in two different groups. They are calculated as following.

Relative risks = Risk in group1/Risk in Group2

Percent increased risk = (Risk in group1 - Risk in Group2)/Risk in Group2

We compared two quadrants with the highest and lowest risks for each type of accident and summarized as bellow: for MULTI_VEHICLE type NE has highest risk 16.0 and NW has lowest 11.3%. Reletive risk MULTI_VEHICLE between NE and NW is 1.4% which is 41.5 percent increased risk. for PEDESTRIAN SW has highest risk 5.1 and NE has lowest 2.5%. Reletive risk PEDESTRIAN between SW and NE is 2.04 % which is 41.8 percent increased risk. For SINGLE_VEHICLE SE has highest risk 18.3 and SW has lowest 12.9%. Reletive risk SINGLE_VEHICLE between SE and SW is 1.4 % which is 104 percent increased risk. For TWO_VEHICLE NW has highest risk 59.2 and SE has lowest 54.5%. Reletive risk TWO_VEHICLE between NW and SE is 1.08% which is 8.6 percent increased risk.

```
#MULTI_VEHICLE , NE and NW
16/11.3

## [1] 1.415929

(16-11.3)/11.3 *100

## [1] 41.59292

(16/(100-16))/(11.3 / (100-11.3))

## [1] 1.495154

# PEDESTRIAN SW and NE
5.1 /2.5

## [1] 2.04

(5.1-2.5) /2.5 *100

## [1] 104

(5.1/(100-5.1))/(2.5/ (100-2.5))
```

```
## [1] 2.09589

# SINGLE_VEHICLE SE and SW
18.3/12.9

## [1] 1.418605

(18.3-12.9)/12.9 *100

## [1] 41.86047

(18.3/(100-18.3))/(12.9/ (100-12.9))

## [1] 1.512368

# TWO_VEHICLE NW and SE
59.2/54.5

## [1] 1.086239

(59.2-54.5)/54.5 *100

## [1] 8.623853

(59.2/(100-59.2))/(54.5/ (100-54.5))

## [1] 1.211369
```

#H0:B=0(month CAN NOT be expressed as a positive linear function of the number of traffic Incident) #HA:B≠0(month CAN be expressed as a positive linear function of the number of traffic Incident)

head(trafficAccident_wantedColumn_DF)

Yearly_Monthly_grouped = aggregate(trafficAccident_wantedColumn_DF$Count, by = list(YEAR = trafficAccident_{w}antedColumn_{D}F$YEAR,MONTH=trafficAccident_wantedColumn_DF$MONTH), FUN=sum, na.rm=T)

two_vehicle_Incidend_DF =filter(trafficAccident_wantedColumn_DF, TYPE == 'TWO_VEHICLE')

Yearly_Monthly_TwoVehicle = aggregate(two_vehicle_Incidend_DF$Count, by = list(YEAR = two_{v}ehicle_{I}ncidend_{D}F$YEAR, MONTH=two_vehicle_Incidend_DF$MONTH), FUN=sum, na.rm=T)

Total_Two_Inc = data.frame(Total = Yearly_Monthly_grouped$x, TWoVehicle = Yearly_{M}onthly_{T}woVehicle$x)

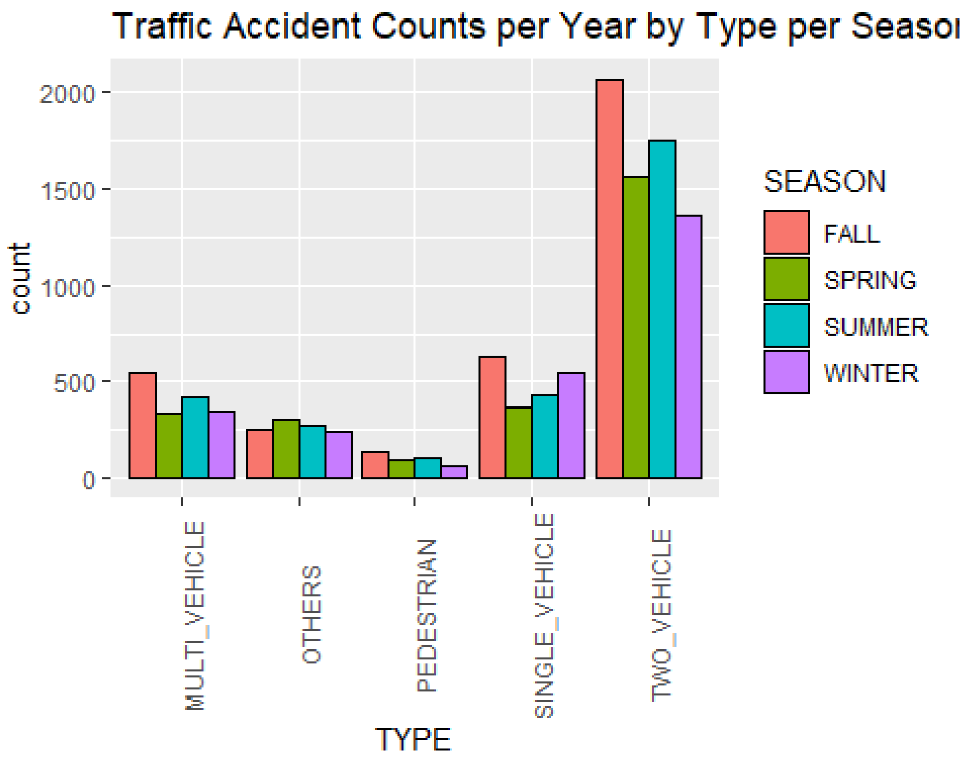
ggplot(data=Total_Two_Inc, aes(x = Total, y = TWoVehicle)) + geom_point(col="blue", size=2, position="jitter") + xlab("Total Accident") + ylab("Two Vehicle Accident") + ggtitle("Scatterplot of Monthly Total Accident toTwo Vehicle Accident") +stat_smooth(method="lm", col='red')

```
#Check the strength of the relation cor(~Total, ~TWoVehicle, data=Total_Two_Inc)

predictTotalAcc = lm( Total~TWoVehicle, data=Total_Two_Inc)

predictHrat = predictTotalAcc$fitted.values #place the predicted values of y for each
observed x into a vector eisHrat = predictTotalAcc$residuals #pull out the residuals
predictionHrat6G = data.frame(predictHrat, eisHrat)

ggplot(predictionHrat6G, aes(sample=eisHrat)) + stat_qq(col=‘blue’, size=2) +
stat_qqline(col=‘red’) + ggtitle(“Normal Probability Plot of the Residuals”)

ggplot(predictionHrat6G, aes(x = predictHrat, y = eisHrat)) + geom_point(size=2, col=‘blue’,
position=“jitter”) + xlab(“Predicted Value”) + ylab(“Residuals”) + ggtitle(“Plot of Fits to
Residuals”) + geom_hline(yintercept=0, color=“red”, linetype=“dashed”)

aov(predictTotalAcc)

options(scipen=999) summary(aov(predictTotalAcc))
```

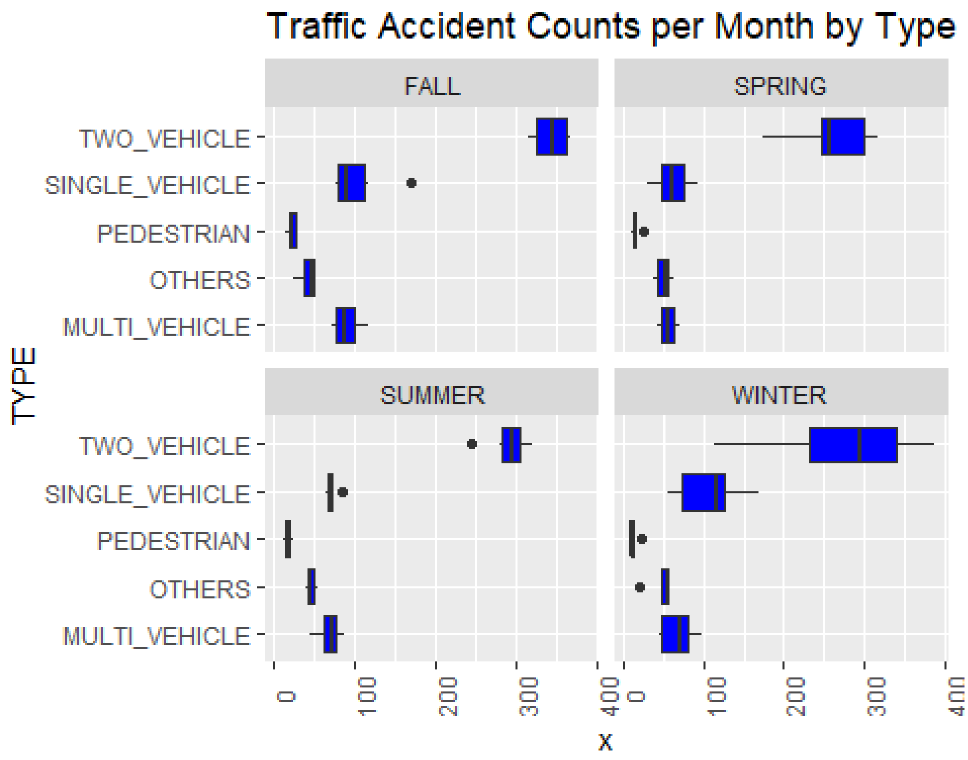
3) Does the type of accident depend on the season?

We are examining whether the type of accident is independent from the season when it occurred. To do this we summarise the count of accidents by season for each type. This can be done using a bar chart.

```
ggplot ( data= trafficAccident_wantedColumn_DF , aes ( x = TYPE, fill =
SEASON)) + geom_bar ( col = 'black', position = "dodge") + theme (
axis.text.x = element_text ( angle = 90)) + ggtitle ("Traffic Accident Counts
per Year by Type per Season ")
```



```
#   Group data by year, month, season, and type
#   There is a count column in dataframe. Every entry appears to be 1. So,
taking the sum of the Count column will give the
#   number of each accident for each season.
seasontraffic_agg = aggregate ( trafficAccident_wantedColumn_DF$Count, by =
list ( YEAR = trafficAccident_wantedColumn_DF$YEAR, MONTH =
trafficAccident_wantedColumn_DF$MONTH, SEASON =
trafficAccident_wantedColumn_DF$SEASON ,  TYPE =
trafficAccident_wantedColumn_DF$TYPE), FUN = sum, na.rm = T)

head (trafficAccident_agg, 4)

##   YEAR MONTH QUADRANT          TYPE  x
## 1 2018     1       NE MULTI_VEHICLE 13
## 2 2017     2       NE MULTI_VEHICLE  9
## 3 2018     2       NE MULTI_VEHICLE 26
## 4 2017     3       NE MULTI_VEHICLE 12

#   Series of boxplots, divided by season using the monthly counts
ggplot ( data = seasontraffic_agg, aes ( x = TYPE, y = x))+ geom_boxplot (
fill = 'blue') + facet_wrap ( ~ SEASON) + theme ( axis.text.x = element_text
( angle = 90)) + coord_flip () + ggtitle ("Traffic Accident Counts per Month
by Type per Season ")
```



From the above plots, Winter shows the greatamount of distribution. We also see variation in single vehicle accidents which suggests there might be a dependency based on season.

A tally table summarises the count of types of traffic accidents by season. Essentially, a tabular form of the bar chart rendered above.

```
conttable_typeseason = tally ( ~ SEASON + TYPE, margins = FALSE, data =
trafficAccident_wantedColumn_DF)
conttable_typeseason

##          TYPE
## SEASON   MULTI_VEHICLE OTHERS PEDESTRIAN SINGLE_VEHICLE TWO_VEHICLE
##   FALL             544    256        136            625        2065
##   SPRING           340    304         99            370        1561
##   SUMMER           417    277        109            431        1746
##   WINTER           346    239         69            541        1365
```

To test the independence of the two categorical variables we will use the xchi_sq function to determine the test statistic and the p-value.

The hypotheses being tested are:

$$ H_0 : The \ type \ of \ incident \ is \ independent \ of \ the \ season \ when \ it \ occurred   \\ H_A : The \ type \ of \ incident \ is \ not \ independent \ of \ the \ season \ when \ it \ occurred $$

```
xchisq.test ( conttable_typeseason, correct = FALSE, simulate.p.value =
FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  x
## X-squared = 105.43, df = 12, p-value < 0.00000000000000022
##
##    544      256      136      625     2065
## ( 504.39) ( 329.52) ( 126.48) ( 602.39) (2063.21)
## [ 3.1100] [16.4052] [ 0.7164] [ 0.8484] [ 0.0016]
## < 1.764> <-4.050> < 0.846> < 0.921> < 0.039>
##
##    340      304       99      370     1561
## ( 371.97) ( 243.01) (  93.27) ( 444.24) (1521.52)
## [ 2.7471] [15.3078] [ 0.3515] [12.4056] [ 1.0247]
## <-1.657> < 3.913> < 0.593> <-3.522> < 1.012>
##
##    417      277      109      431     1746
## ( 414.53) ( 270.82) ( 103.95) ( 495.07) (1695.63)
## [ 0.0147] [ 0.1411] [ 0.2456] [ 8.2923] [ 1.4963]
## < 0.121> < 0.376> < 0.496> <-2.880> < 1.223>
##
##    346      239       69      541     1365
## ( 356.11) ( 232.65) (  89.30) ( 425.30) (1456.65)
## [ 0.2869] [ 0.1734] [ 4.6136] [31.4771] [ 5.7663]
## <-0.536> < 0.416> <-2.148> < 5.610> <-2.401>
##
## key:
##  observed
##  (expected)
##  [contribution to X-squared]
##  <Pearson residual>
```

The test returns a chi_squared test statistic of 105.43 and a p-value of 0.00000000000000022. From the p-value, the null hypothesis is rejected and we can infer that the type of traffic accident is dependent on the season when it occurred in some way.

Since the null hypothesis has been rejected and we have determined that type of accident is dependent on the season, we can now where the dependencies likely lie. Using the porp.test, we can examine each type of accident individually and perform a difference of proportions test comparing each season against the others, six comparisons, to find where these differences might be.

The general hypotheses for the difference of proportions are

$$ H_0 : p_{season 1} - p_{season 2} = 0 \\ H_A : p_{season 1} - p_{season 2} \neq 0 $$

We do not start with any assumptions about the proportion for one season being high or lower than that for another season, so, we perform a series of two-sided prop tests.

Computation of seasonal difference of proportions for pedestrian related accidents.

```r
#   p_spring - p_summer
prop.test (c(99, 109), c(2674, 2980), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(99, 109) out of c(2674, 2980)
## X-squared = 0.0079117, df = 1, p-value = 0.9291
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.009384787  0.010276798
## sample estimates:
##     prop 1     prop 2
## 0.03702319 0.03657718
```

```r
#   p_summer - p_fall
prop.test (c(109, 136), c(2980, 3626), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(109, 136) out of c(2980, 3626)
## X-squared = 0.03959, df = 1, p-value = 0.8423
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.010076925  0.008217498
## sample estimates:
##     prop 1     prop 2
## 0.03657718 0.03750689
```

```r
#   p_fall - p_winter
prop.test (c(136, 69), c(3626, 2560), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(136, 69) out of c(3626, 2560)
## X-squared = 5.2163, df = 1, p-value = 0.02238
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.001744675 0.019362864
## sample estimates:
##     prop 1     prop 2
## 0.03750689 0.02695313
```

```
#   p_winter - p_spring
prop.test (c(69, 99), c(2560, 2674), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(69, 99) out of c(2560, 2674)
## X-squared = 4.269, df = 1, p-value = 0.03881
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0195870532 -0.0005530692
## sample estimates:
##     prop 1     prop 2
## 0.02695313 0.03702319

#   p_winter - p_summer
prop.test (c(69, 109), c(2560, 2980), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(69, 109) out of c(2560, 2980)
## X-squared = 4.1014, df = 1, p-value = 0.04285
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0188317267 -0.0004163857
## sample estimates:
##     prop 1     prop 2
## 0.02695313 0.03657718

#   p_spring - p_fall
prop.test (c(99, 136), c(2674, 3626), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(99, 136) out of c(2674, 3626)
## X-squared = 0.010028, df = 1, p-value = 0.9202
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.009942219  0.008974802
## sample estimates:
##     prop 1     prop 2
## 0.03702319 0.03750689
```

From the results of the six proportionality tests, the p-values of three differences of proportions suggested the rejection of the null hypothesis. These three differences are: p_fall - p_winter (p-value = 0.022), p_winter - p_spring (p-value = 0.038), and p_winter - p_summer (p-value = 0.043). The 95% confidence intervals for each suggest, fall has 0.174% to 1.94% more pedestrian related accident compared to winter, spring has 0.055% to 1.96% more accidents than winter, and summer has 0.042% to 1.88% more accidents than winter.

Computation of seasonal difference of proportions for multi-vehicle accidents.

```
#   p_spring - p_summer
prop.test (c(340, 417), c(2674, 2980), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(340, 417) out of c(2674, 2980)
## X-squared = 1.9858, df = 1, p-value = 0.1588
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.03051898  0.00495388
## sample estimates:
##    prop 1    prop 2
## 0.1271503 0.1399329
```

```
#   p_summer - p_fall
prop.test (c(417, 544), c(2980, 3626), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(417, 544) out of c(2980, 3626)
## X-squared = 1.3409, df = 1, p-value = 0.2469
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.027131110  0.006941725
## sample estimates:
##    prop 1    prop 2
## 0.1399329 0.1500276
```

```
#   p_fall - p_winter
prop.test (c(544, 346), c(3626, 2560), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
```

```
##
## data:  c(544, 346) out of c(3626, 2560)
## X-squared = 2.6943, df = 1, p-value = 0.1007
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.002749589  0.032492246
## sample estimates:
##    prop 1    prop 2
## 0.1500276 0.1351563
```

```
#   p_winter - p_spring
prop.test (c(346, 340), c(2560, 2674), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(346, 340) out of c(2560, 2674)
## X-squared = 0.73606, df = 1, p-value = 0.3909
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.01029267  0.02630450
## sample estimates:
##    prop 1    prop 2
## 0.1351563 0.1271503
```

```
#   p_winter - p_summer
prop.test (c(346, 417), c(2560, 2980), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(346, 417) out of c(2560, 2980)
## X-squared = 0.26456, df = 1, p-value = 0.607
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.02295748  0.01340421
## sample estimates:
##    prop 1    prop 2
## 0.1351563 0.1399329
```

```
#   p_spring - p_fall
prop.test (c(340, 544), c(2674, 3626), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
```

```
##
## data:  c(340, 544) out of c(2674, 3626)
## X-squared = 6.6774, df = 1, p-value = 0.009764
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.040039250 -0.005715234
## sample estimates:
##    prop 1    prop 2
## 0.1271503 0.1500276
```

From the results of the proportionality tests, only the p-value for the comparison between spring and fall (p-value = 0.0098) showed a significant difference between the proportions of multi-vehicle accidents. From the 95% confident interval, you would expect there to be 0.057% to 4.00% more multi-vehicle in fall compared to spring.

Computation of seasonal difference of proportions for two-vehicle accidents.

```
#   p_spring - p_summer
prop.test (c(1561, 1746), c(2674, 2980), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(1561, 1746) out of c(2674, 2980)
## X-squared = 0.026494, df = 1, p-value = 0.8707
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.02786236  0.02358955
## sample estimates:
##    prop 1    prop 2
## 0.5837696 0.5859060

#   p_summer - p_fall
prop.test (c(1746, 2065), c(2980, 3626), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(1746, 2065) out of c(2980, 3626)
## X-squared = 1.8041, df = 1, p-value = 0.1792
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.007518896  0.040334838
## sample estimates:
##    prop 1    prop 2
## 0.5859060 0.5694981
```

```r
#   p_fall - p_winter
prop.test (c(2065, 1365), c(3626, 2560), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(2065, 1365) out of c(3626, 2560)
## X-squared = 8.002, df = 1, p-value = 0.004673
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.01113096 0.06145893
## sample estimates:
##    prop 1    prop 2
## 0.5694981 0.5332031

#   p_winter - p_spring
prop.test (c(1365, 1561), c(2560, 2674), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(1365, 1561) out of c(2560, 2674)
## X-squared = 13.566, df = 1, p-value = 0.0002303
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.07744691 -0.02368610
## sample estimates:
##    prop 1    prop 2
## 0.5332031 0.5837696

#   p_winter - p_summer
prop.test (c(1365, 1746), c(2560, 2980), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(1365, 1746) out of c(2560, 2980)
## X-squared = 15.535, df = 1, p-value = 0.000081
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.07889919 -0.02650664
## sample estimates:
##    prop 1    prop 2
## 0.5332031 0.5859060
```

```
#   p_spring - p_fall
prop.test (c(1561, 2065), c(2674, 3626), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(1561, 2065) out of c(2674, 3626)
## X-squared = 1.2832, df = 1, p-value = 0.2573
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.01040243  0.03894556
## sample estimates:
##    prop 1    prop 2
## 0.5837696 0.5694981
```

From the results of the proportionality tests, the p-values for three tests showed a statistically significant difference between proportions. These three differences are: p_fall - p_winter (p-value = 0.0047), p_winter - p_spring (p-value = 0.00023), and p_winter - p_summer (p-value = 0.000081). The 95% confidence intervals for each suggest we would expect fall to have 1.11% to 6.15% more multi-vehicle accidents compared to winter, spring has 2.37% to 7.74% more accidents compared to winter, and summer has 2.65% to 7.89% more accidents compared to winter.

Computation of seasonal difference of proportions for single-vehicle accidents.

```
#   p_spring - p_summer
prop.test (c(370, 431), c(2674, 2980), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(370, 431) out of c(2674, 2980)
## X-squared = 0.45439, df = 1, p-value = 0.5003
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.02444796  0.01192519
## sample estimates:
##    prop 1    prop 2
## 0.1383695 0.1446309

#   p_summer - p_fall
prop.test (c(431, 625), c(2980, 3626), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
```

```
##
## data:  c(431, 625) out of c(2980, 3626)
## X-squared = 9.369, df = 1, p-value = 0.002207
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.04535946 -0.01011128
## sample estimates:
##    prop 1    prop 2
## 0.1446309 0.1723662

#   p_fall - p_winter
prop.test (c(625, 541), c(3626, 2560), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(625, 541) out of c(3626, 2560)
## X-squared = 14.892, df = 1, p-value = 0.0001138
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.05899262 -0.01893114
## sample estimates:
##    prop 1    prop 2
## 0.1723662 0.2113281

#   p_winter - p_spring
prop.test (c(541, 370), c(2560, 2674), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(541, 370) out of c(2560, 2674)
## X-squared = 48.427, df = 1, p-value = 0.000000000003429
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.05243125 0.09348603
## sample estimates:
##    prop 1    prop 2
## 0.2113281 0.1383695

#   p_winter - p_summer
prop.test (c(541, 431), c(2560, 2980), alternative = "two.sided", correct =
FALSE)

##
##  2-sample test for equality of proportions without continuity
##  correction
```

```
##
## data:  c(541, 431) out of c(2560, 2980)
## X-squared = 42.344, df = 1, p-value = 0.00000000007656
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.04645933 0.08693517
## sample estimates:
##    prop 1    prop 2
## 0.2113281 0.1446309
```

```
#   p_spring - p_fall
prop.test (c(370, 625), c(2674, 3626), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(370, 625) out of c(2674, 3626)
## X-squared = 13.375, df = 1, p-value = 0.000255
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.05195250 -0.01604102
## sample estimates:
##    prop 1    prop 2
## 0.1383695 0.1723662
```

From the results of the proportionality tests, only the p-value for the comparison between spring and summer (p-value = 0.500) did not show a statistically significant difference. The others that did show a significant difference are: p_summer - p_fall (p-value = 0.0022), p_fall - p_winter (p-value = 0.00011), p_winter - p_spring (p-value = 0.000000000034), p_winter - p_summer (p-value = 0.000000000077), and p_spring - p_fall (p-value = 0.00026). The 95% confidence interval for each suggest, fall has 1.01% to 4.54% more single vehicle accidents compared to fall, winter has 1.89% to 5.90% more accidents than fall, winter has 5.24 %to 9.35% more accidents compared to spring, winter has 4.65% to 8.69% more accidents compared to summer, and fall has 1.60% to 5.20% more accidents compared to spring.

Computation of seasonal difference of proportions for incidents categorised as Other.

```
#   p_spring - p_summer
prop.test (c(304, 277), c(2674, 2980), alternative = "two.sided", correct =
FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(304, 277) out of c(2674, 2980)
## X-squared = 6.5716, df = 1, p-value = 0.01036
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.004814509 0.036654170
## sample estimates:
##     prop 1     prop 2 
## 0.11368736 0.09295302
```

```r
#   p_summer - p_fall
prop.test (c(277, 256), c(2980, 3626), alternative = "two.sided", correct =
FALSE)
```

```
## 
##  2-sample test for equality of proportions without continuity
##  correction
## 
## data:  c(277, 256) out of c(2980, 3626)
## X-squared = 11.017, df = 1, p-value = 0.0009026
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.009002591 0.035701022
## sample estimates:
##     prop 1     prop 2 
## 0.09295302 0.07060121
```

```r
#   p_fall - p_winter
prop.test (c(256, 239), c(3626, 2560), alternative = "two.sided", correct =
FALSE)
```

```
## 
##  2-sample test for equality of proportions without continuity
##  correction
## 
## data:  c(256, 239) out of c(3626, 2560)
## X-squared = 10.557, df = 1, p-value = 0.001157
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.036777042 -0.008739282
## sample estimates:
##     prop 1     prop 2 
## 0.07060121 0.09335937
```

```r
#   p_winter - p_spring
prop.test (c(239, 304), c(2560, 2674), alternative = "two.sided", correct =
FALSE)
```

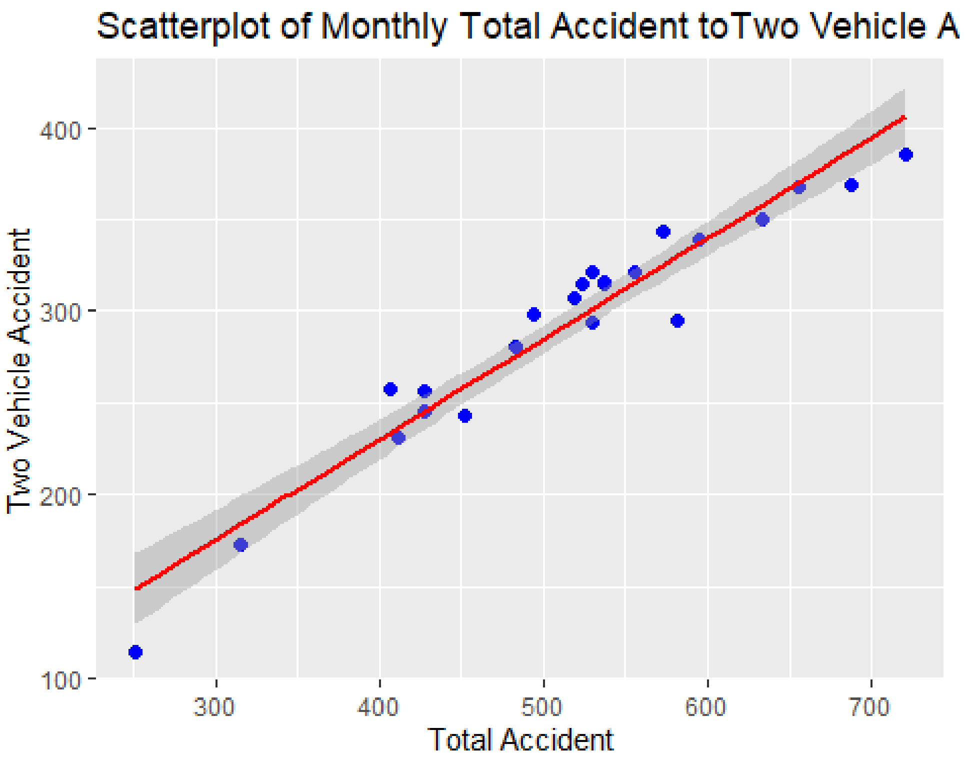
```
## 
##  2-sample test for equality of proportions without continuity
##  correction
## 
## data:  c(239, 304) out of c(2560, 2674)
## X-squared = 5.8124, df = 1, p-value = 0.01591
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.036813386 -0.003842584
## sample estimates:
##     prop 1     prop 2 
## 0.09335937 0.11368736 

#   p_winter - p_summer
prop.test (c(239, 277), c(2560, 2980), alternative = "two.sided", correct =
FALSE)

## 
##  2-sample test for equality of proportions without continuity
##  correction
## 
## data:  c(239, 277) out of c(2560, 2980)
## X-squared = 0.002692, df = 1, p-value = 0.9586
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.01494614  0.01575885
## sample estimates:
##     prop 1     prop 2 
## 0.09335937 0.09295302 

#   p_spring - p_fall
prop.test (c(304, 256), c(2674, 3626), alternative = "two.sided", correct =
FALSE)

## 
##  2-sample test for equality of proportions without continuity
##  correction
## 
## data:  c(304, 256) out of c(2674, 3626)
## X-squared = 35.278, df = 1, p-value = 0.000000002858
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.02844816 0.05772414
## sample estimates:
##     prop 1     prop 2 
## 0.11368736 0.07060121 
```

From the results of the proportionality tests, only the p-value for the comparison between winter and summer (p-value = 0.959) did not show a statistically significant difference. The others that did show a significant difference are: p_spring - p_summer (p-value = 0.011), p_summer - p_fall (p-value = 0.00090), p_fall - p_winter (p-value = 0.0.0012), p_winter - p_spring (p-value = 0.016), and p_spring - p_fall (p-value = 0.0000000029). The 95% confidence interval for each suggest, spring has 0.48% to 3.67% more Other categorized incidents compared to summer, summer has 0.90% to 3.57% more indicents than fall, winter has 0.87% to 3.67% more incidents compared to fall, 0.38% to 3.68% more incidents compared to spring, and spring has 2.84% to 5.77% more incidents compared to fall.

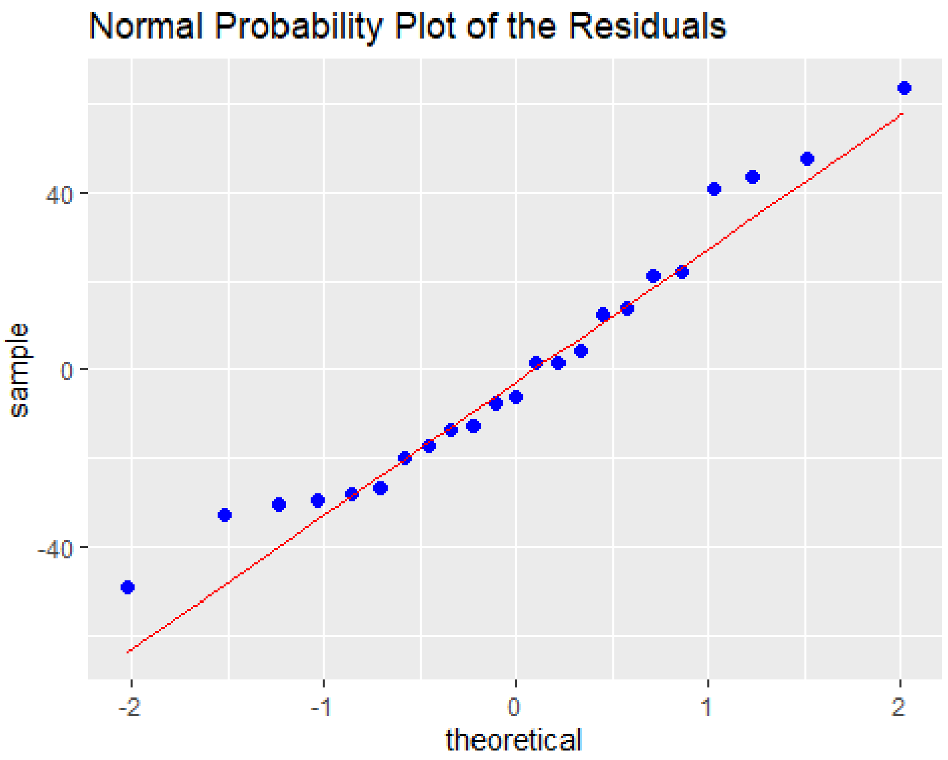
4) Can we create a linear regression model for the number of traffic incidences vs time?

#H0:B=0(month CAN NOT be expressed as a positive linear function of the number of traffic Incident) #HA:B≠0(month CAN be expressed as a positive linear function of the number of traffic Incident)

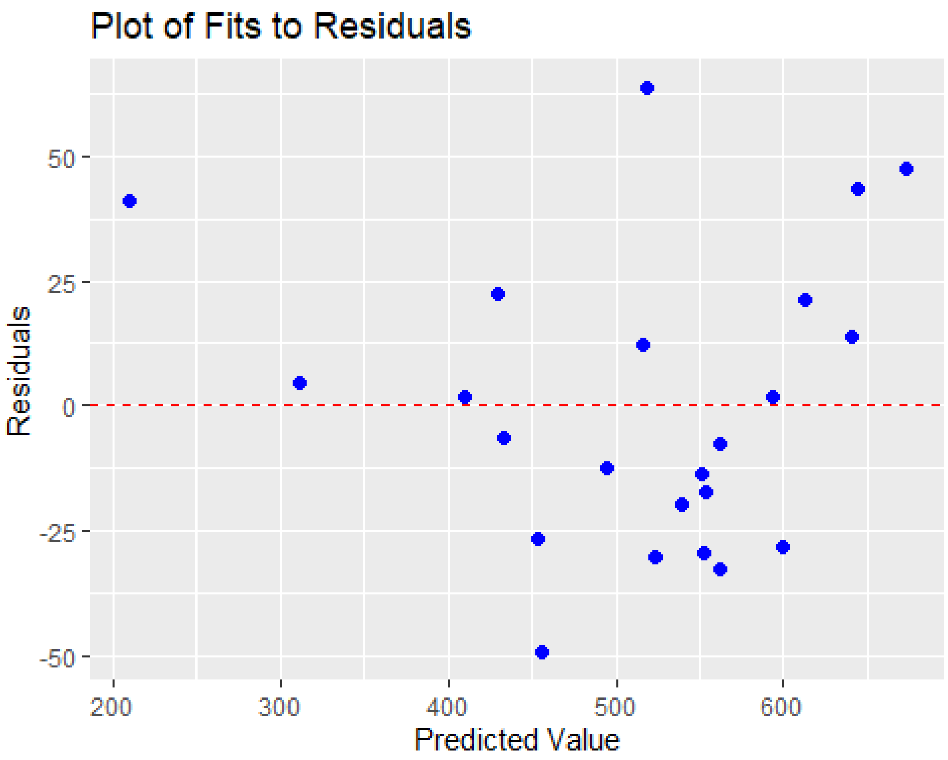
```
head(trafficAccident_wantedColumn_DF)

##   Count MONTH YEAR QUADRANT           TYPE DAY SEASON    HOURTYPE
## 1     1     2 2017       SE SINGLE_VEHICLE   8 WINTER    RUSHHOUR
## 2     1     2 2017       SE  MULTI_VEHICLE   8 WINTER NOTRUSHHOUR
## 3     1     2 2017       NE    TWO_VEHICLE   8 WINTER NOTRUSHHOUR
## 4     1     2 2017       SE    TWO_VEHICLE   8 WINTER NOTRUSHHOUR
## 5     1     2 2017       NW  MULTI_VEHICLE   8 WINTER NOTRUSHHOUR
## 6     1     2 2017       NE    TWO_VEHICLE   8 WINTER NOTRUSHHOUR
##    PROPORTION
## 1 0.000251067
## 2 0.000127275
## 3 0.000127275
## 4 0.000127275
## 5 0.000127275
## 6 0.000127275

Yearly_Monthly_grouped = aggregate(trafficAccident_wantedColumn_DF$Count, by=
list(
YEAR=trafficAccident_wantedColumn_DF$YEAR,MONTH=trafficAccident_wantedColumn_
DF$MONTH), FUN=sum, na.rm=T)

two_vehicle_Incidend_DF  =filter(trafficAccident_wantedColumn_DF, TYPE ==
'TWO_VEHICLE')

Yearly_Monthly_TwoVehicle = aggregate(two_vehicle_Incidend_DF$Count, by=
list( YEAR=two_vehicle_Incidend_DF$YEAR,
MONTH=two_vehicle_Incidend_DF$MONTH), FUN=sum, na.rm=T)

Total_Two_Inc = data.frame( Total = Yearly_Monthly_grouped$x , TWoVehicle
=Yearly_Monthly_TwoVehicle$x)

ggplot(data=Total_Two_Inc, aes(x = Total, y = TWoVehicle)) +
geom_point(col="blue", size=2, position="jitter") + xlab("Total Accident") +
ylab("Two Vehicle Accident") + ggtitle("Scatterplot of Monthly Total Accident
toTwo Vehicle Accident") +stat_smooth(method="lm", col='red')
```



```
#Check the strength of the relation
cor(~Total, ~TWoVehicle,  data=Total_Two_Inc)

## [1] 0.9648007

predictTotalAcc = lm( Total~TWoVehicle, data=Total_Two_Inc)


predictHrat = predictTotalAcc$fitted.values #place the predicted values of y
for each observed x into a vector
eisHrat = predictTotalAcc$residuals      #pull out the residuals
predictionHrat6G = data.frame(predictHrat, eisHrat)


ggplot(predictionHrat6G, aes(sample=eisHrat)) + stat_qq(col='blue', size=2) +
stat_qqline(col='red') + ggtitle("Normal Probability Plot of the Residuals")
```

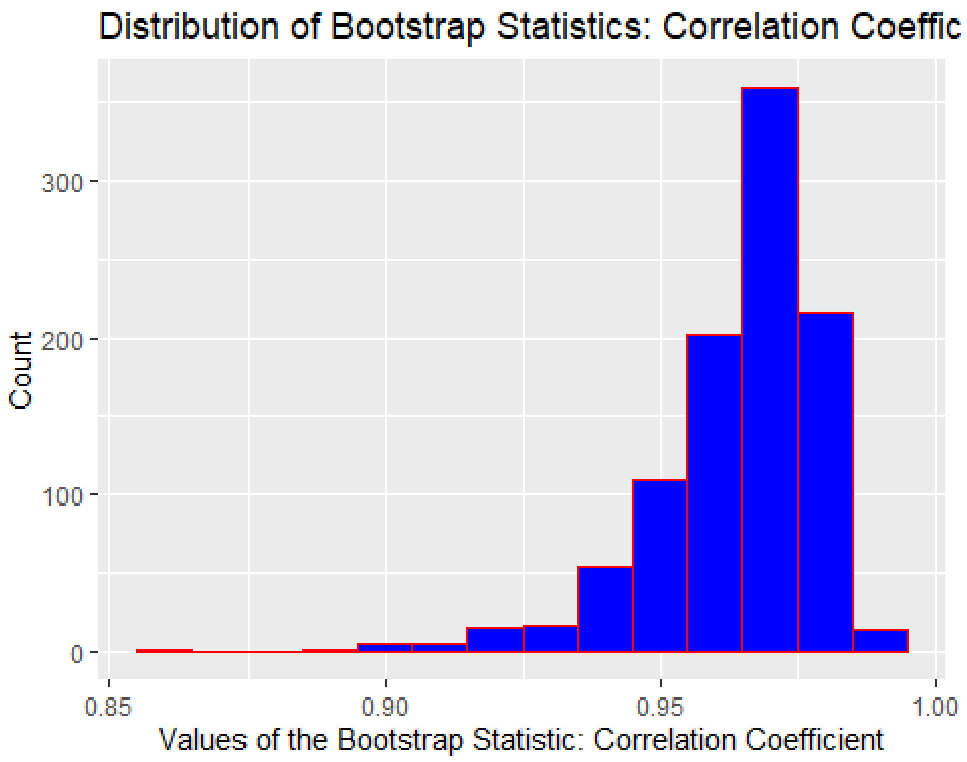


```
ggplot(predictionHrat6G, aes(x = predictHrat, y = eisHrat)) +
geom_point(size=2, col='blue', position="jitter") + xlab("Predicted Value") +
ylab("Residuals") + ggtitle("Plot of Fits to Residuals") +
geom_hline(yintercept=0, color="red", linetype="dashed")
```

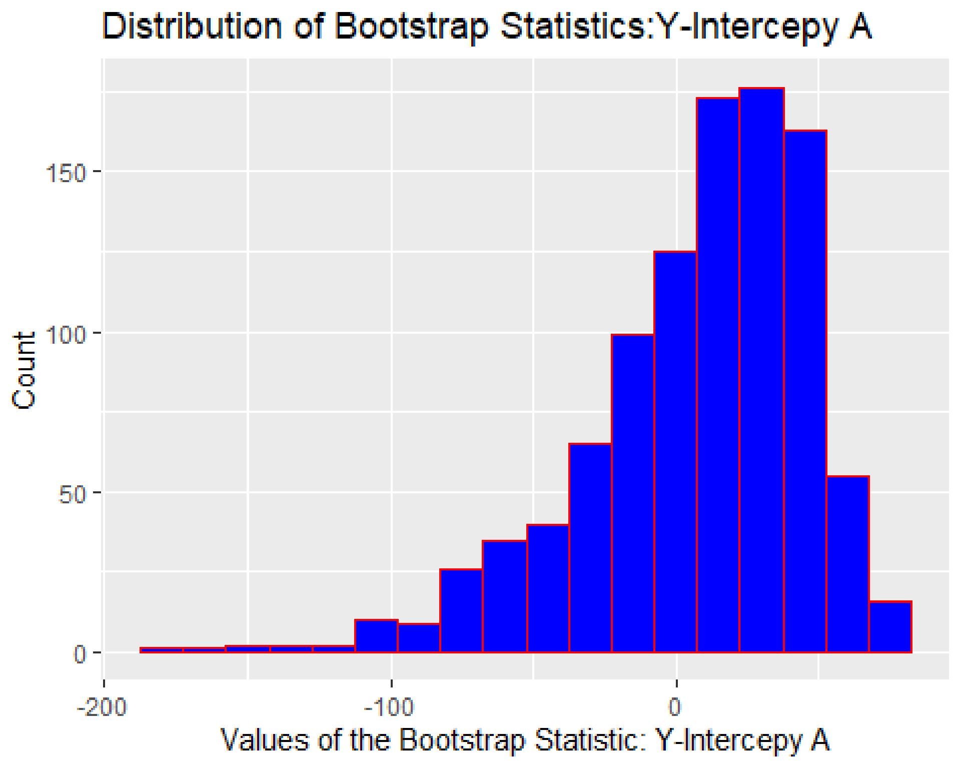


```
aov(predictTotalAcc)

## Call:
##    aov(formula = predictTotalAcc)
##
## Terms:
##                 TWoVehicle Residuals
## Sum of Squares    257394.0   19123.9
## Deg. of Freedom          1        21
##
## Residual standard error: 30.17717
## Estimated effects may be unbalanced

options(scipen=999)
summary(aov(predictTotalAcc))

##             Df Sum Sq Mean Sq F value            Pr(>F)
## TWoVehicle   1 257394  257394   282.6 0.000000000000117 ***
## Residuals   21  19124     911
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Nbootstraps = 1000 #resample n =  14, 3000 times
cor.boot = numeric(Nbootstraps) #define a vector to be filled by the cor boot
stat
a.boot = numeric(Nbootstraps) #define a vector to be filled by the a boot
stat
```

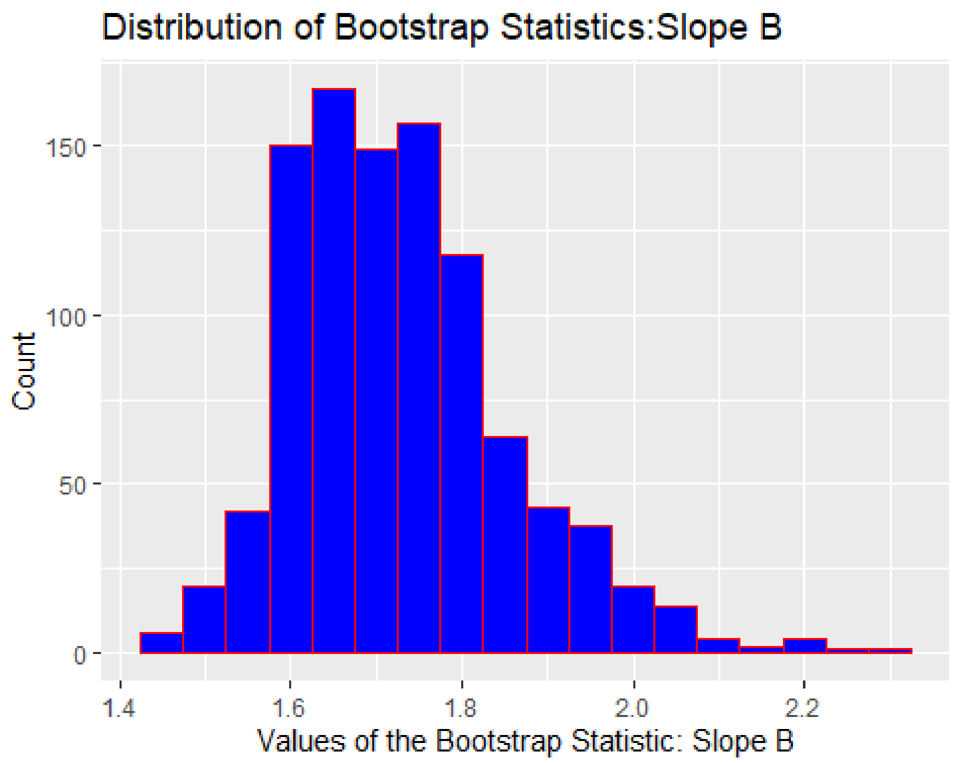
```r
b.boot = numeric(Nbootstraps) #define a vector to be filled by the b boot
stat
#ymean.boot = numeric(Nbootstraps) #define a vector to be filled by the
predicted y boot stat

nsize = dim(Total_Two_Inc)[1]  #set the n to be equal to the number of
bivariate cases, number of rows
#start of the for loop
for(i in 1:Nbootstraps)
{   #start of the loop
    index = sample(nsize, replace=TRUE)  #randomly picks n- number between 1
and n, assigns as index
    TWOV.boot = Total_Two_Inc[index, ] #accesses the i-th row of the
SAT_2010High data frame
    #
    cor.boot[i] = cor( ~Total,~TWoVehicle, data=TWOV.boot) #computes
correlation for each bootstrap sample
    SAT.lm = lm( Total~TWoVehicle, data=TWOV.boot)  #set up the linear model
    a.boot[i] = coef(SAT.lm)[1] #access the computed value of a, in position
1
    b.boot[i] = coef(SAT.lm)[2] #access the computed valeu of b, in position
2
   # ymean.boot[i] = a.boot[i] + (b.boot[i]*xvalue)
}
#end the loop
#create a data frame that holds the results of teach of he Nbootstraps
bootstrapresultsdf = data.frame(cor.boot, a.boot, b.boot)

ggplot(bootstrapresultsdf, aes(x = cor.boot)) + geom_histogram(col="red",
fill="blue", binwidth=0.01) + xlab("Values of the Bootstrap Statistic:
Correlation Coefficient") + ylab("Count") + ggtitle("Distribution of
Bootstrap Statistics: Correlation Coefficient")
```



```
ggplot(bootstrapresultsdf, aes(x = a.boot)) + geom_histogram(col="red",
fill="blue", binwidth=15) + xlab("Values of the Bootstrap Statistic: Y-
Intercepy A") + ylab("Count") + ggtitle("Distribution of Bootstrap
Statistics:Y-Intercepy A")
```



```
ggplot(bootstrapresultsdf, aes(x = b.boot)) + geom_histogram(col="red",
fill="blue", binwidth=0.05) + xlab("Values of the Bootstrap Statistic: Slope
B") + ylab("Count") + ggtitle("Distribution of Bootstrap Statistics:Slope B")
```



```
boot.amean = favstats(~a.boot, data=bootstrapresultsdf)$mean
boot.bmean = favstats(~b.boot, data=bootstrapresultsdf)$mean
   boot.amean

## [1] 7.363953

   boot.bmean

## [1] 1.728649

cat("The model to predict Total Accidend is Total=" ,boot.amean ,"+
(",boot.bmean ,"TwoVehiclei)+ei\n")

## The model to predict Total Accidend is Total= 7.363953 + ( 1.728649
*TwoVehiclei)+ei

cat ("The Total Accidend is " ,boot.amean +(boot.bmean *367))

## The Total Accidend is  641.778

df2019 = filter(trafficAccidentDF, YEAR==2019, MONTH<6)
actual_Data2019 = aggregate(df2019$Count, by= list( YEAR=df2019$YEAR,
MONTH=df2019$MONTH), FUN=sum, na.rm=T)
df2019TwoVehicle = filter(df2019, TYPE=='TWO_VEHICLE' )
df2019_Two_Vehicledf = aggregate(df2019TwoVehicle$Count, by= list(
YEAR=df2019TwoVehicle$YEAR, MONTH=df2019TwoVehicle$MONTH), FUN=sum, na.rm=T)

predictFit =numeric( dim(actual_Data2019)[1])
predictLwr =numeric( dim(actual_Data2019)[1])
predictUpr =numeric( dim(actual_Data2019)[1])
 for(i in 1:5){
   prediction = predict(predictTotalAcc, newdata=data.frame(TWoVehicle =
df2019_Two_Vehicledf$x[i]), interval="predict", conf.level=0.95)
   predictFit[i]= prediction[1]
   predictLwr[i]= prediction[2]
   predictUpr[i]= prediction[3]
 }
prediction_2019_df = data.frame(month =actual_Data2019$MONTH, TwoVehicle =
df2019_Two_Vehicledf$x, Lower = predictLwr, Total = actual_Data2019$x, Upper
= predictUpr )
 print(prediction_2019_df)

##   month TwoVehicle    Lower Total    Upper
## 1     1        313 484.7412   545 613.2334
## 2     2        367 574.9614   692 706.9188
## 3     3        201 291.3040   377 425.2365
## 4     4        154 207.7687   255 348.7058
## 5     5        222 328.2059   366 459.8535
```