#Read raw data  
df<-read.csv("toronto.csv", na.strings=c("","No Information","Pending","UNKNOWN","NOT LISTED, PLEASE SPECIFY","OTHER"))

#See data structure  
head(df)

## X\_id Assigned\_ID Outbreak.Associated Age.Group Neighbourhood.Name FSA  
## 1 1 1 Sporadic 50 to 59 Years Willowdale East M2N  
## 2 2 2 Sporadic 50 to 59 Years Willowdale East M2N  
## 3 3 3 Sporadic 20 to 29 Years Parkwoods-Donalda M3A  
## 4 4 4 Sporadic 60 to 69 Years Church-Yonge Corridor M4W  
## 5 5 5 Sporadic 60 to 69 Years Church-Yonge Corridor M4W  
## 6 6 6 Sporadic 50 to 59 Years Newtonbrook West M2R  
## Source.of.Infection Classification Episode.Date Reported.Date Client.Gender  
## 1 Travel CONFIRMED 1/22/2020 1/23/2020 FEMALE  
## 2 Travel CONFIRMED 1/21/2020 1/23/2020 MALE  
## 3 Travel CONFIRMED 2/5/2020 2/21/2020 FEMALE  
## 4 Travel CONFIRMED 2/16/2020 2/25/2020 FEMALE  
## 5 Travel CONFIRMED 2/20/2020 2/26/2020 MALE  
## 6 Travel CONFIRMED 2/24/2020 2/27/2020 MALE  
## Outcome Currently.Hospitalized Currently.in.ICU Currently.Intubated  
## 1 RESOLVED No No No  
## 2 RESOLVED No No No  
## 3 RESOLVED No No No  
## 4 RESOLVED No No No  
## 5 RESOLVED No No No  
## 6 RESOLVED No No No  
## Ever.Hospitalized Ever.in.ICU Ever.Intubated  
## 1 No No No  
## 2 Yes No No  
## 3 No No No  
## 4 No No No  
## 5 No No No  
## 6 No No No

summary(df)

## X\_id Assigned\_ID Outbreak.Associated Age.Group   
## Min. : 1 Min. : 1 Length:370892 Length:370892   
## 1st Qu.: 92724 1st Qu.: 96457 Class :character Class :character   
## Median :185447 Median :192070 Mode :character Mode :character   
## Mean :185447 Mean :191326   
## 3rd Qu.:278169 3rd Qu.:286249   
## Max. :370892 Max. :381171   
## Neighbourhood.Name FSA Source.of.Infection Classification   
## Length:370892 Length:370892 Length:370892 Length:370892   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## Episode.Date Reported.Date Client.Gender Outcome   
## Length:370892 Length:370892 Length:370892 Length:370892   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## Currently.Hospitalized Currently.in.ICU Currently.Intubated  
## Length:370892 Length:370892 Length:370892   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## Ever.Hospitalized Ever.in.ICU Ever.Intubated   
## Length:370892 Length:370892 Length:370892   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##

#See missing values  
sum(is.na(df))

## [1] 204227

#Format date column  
df$Episode.Date <- as.Date(df$Episode.Date, format = "%m/%d/%Y")  
df$Reported.Date <- as.Date(df$Reported.Date, format = "%m/%d/%Y")  
#Ensure format column correct  
class(df$Episode.Date)

## [1] "Date"

#Factorize column  
col <- c("Outbreak.Associated","Age.Group","Neighbourhood.Name","FSA","Source.of.Infection","Classification","Client.Gender", "Outcome","Currently.Hospitalized",  
 "Currently.in.ICU","Currently.Intubated","Ever.Hospitalized","Ever.in.ICU","Ever.Intubated")  
  
df[col] <-lapply(df[col],factor)

#Include only complete cases and remove unwanted columns  
df1<-df[complete.cases(df), ]  
  
df1 <- df1[ -c(1:2,13:15) ]

#Create month data  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

df1$Month <- format(df1$Episode.Date, "%m")  
df1$Month<-as.numeric(gsub("0", "", df1$Month))  
df1$Month <- as.factor(df1$Month)  
df1<-df1 %>% dplyr::select(-c(Episode.Date,Reported.Date,Classification))

#Take out levels that has undetermined results  
df2<-df1  
df2<-df2[!grepl('ACTIVE', df2$Outcome),]  
df2$Outcome<-droplevels(df2$Outcome)

#Join data from 201 Census data, and import census data  
census<-read.csv("join.csv")  
head(census)

## Neighbourhood.Name Income Density  
## 1 City of Toronto 81,495 4,334  
## 2 Agincourt North 427,037 3,929  
## 3 Agincourt South-Malvern West 278,390 3,034  
## 4 Alderwood 168,602 2,435  
## 5 Annex 792,507 10,863  
## 6 Banbury-Don Mills 493,486 2,775

census$Neighbourhood.Name<-as.factor(census$Neighbourhood.Name)  
  
#merge by neighborhood name to get income & density by neighborhood  
df3<- df2 %>% left\_join(census, by= c("Neighbourhood.Name"="Neighbourhood.Name"))  
  
  
#remove "," within the incomes  
df3$Income<-as.numeric(gsub(",", "", df3$Income))  
df3$Density<-as.numeric(gsub(",", "", df3$Density))  
df3<-df3 %>% dplyr::select(-c(FSA,Neighbourhood.Name))  
sum(is.na(df3$Income))

## [1] 0

#check the statistical description of fatal cases  
nrow(df3)

## [1] 183045

fatal<-df3[grep("FATAL", df3$Outcome), ]  
nrow(fatal)

## [1] 3485

#check the statistical description of resolved cases  
resolved<-df3[grep("RESOLVED", df3$Outcome), ]  
nrow(resolved)

## [1] 179560

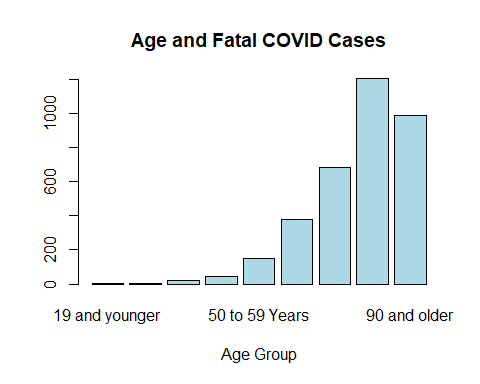
#relocate outcome column as last column  
df3 <- df3 %>% relocate(Outcome, .after = Density)  
#check structure of final data set  
str(df3)

## 'data.frame': 183045 obs. of 11 variables:  
## $ Outbreak.Associated: Factor w/ 2 levels "Outbreak Associated",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Age.Group : Factor w/ 9 levels "19 and younger",..: 5 5 2 6 6 5 8 6 5 6 ...  
## $ Source.of.Infection: Factor w/ 7 levels "Close Contact",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ Client.Gender : Factor w/ 6 levels "FEMALE","MALE",..: 1 2 1 1 2 2 2 2 2 2 ...  
## $ Ever.Hospitalized : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 2 1 1 ...  
## $ Ever.in.ICU : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ever.Intubated : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Month : Factor w/ 11 levels "1","2","3","4",..: 1 1 2 2 2 2 2 2 2 2 ...  
## $ Income : num 572155 572155 623453 443734 443734 ...  
## $ Density : num 10087 10087 4691 23044 23044 ...  
## $ Outcome : Factor w/ 2 levels "FATAL","RESOLVED": 2 2 2 2 2 2 2 2 2 2 ...

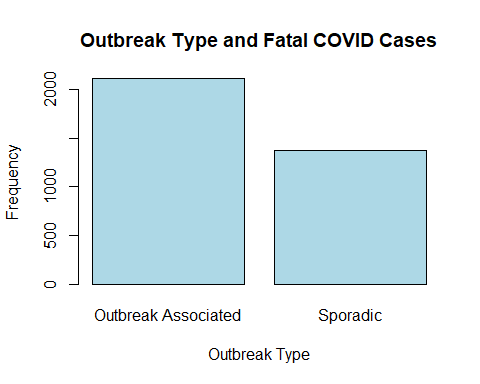
#save temp data  
write.csv(df3,"C:/KZ/S/CIND820\\df3.csv", row.names = FALSE)

VISUALIZE DATA

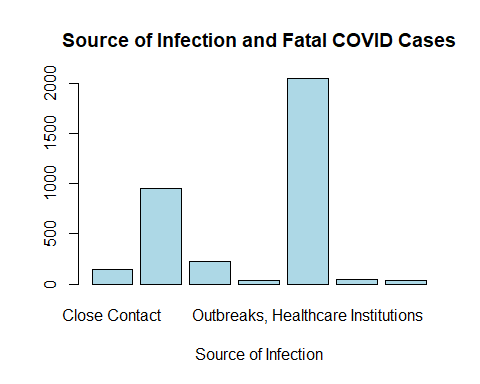
#bar plot for age vs outcome  
fatal$Outcome<-droplevels(fatal$Outcome)  
age <- table(fatal$Outcome,fatal$Age.Group)  
  
barplot(age, main="Age and Fatal COVID Cases",  
 xlab="Age Group", col=c("lightblue"))



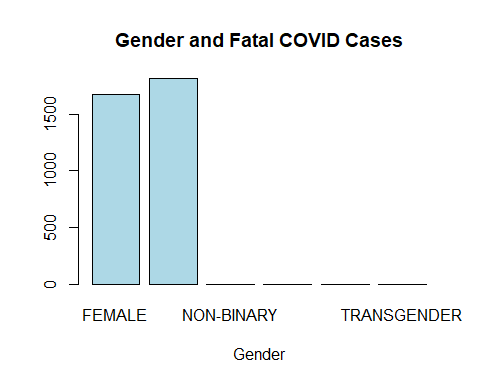
#bar plot for outbreak vs outcome  
out <- table(fatal$Outcome,fatal$Outbreak.Associated)  
barplot(out, main="Outbreak Type and Fatal COVID Cases",  
 xlab="Outbreak Type", ylab="Frequency", col=c("lightblue"))



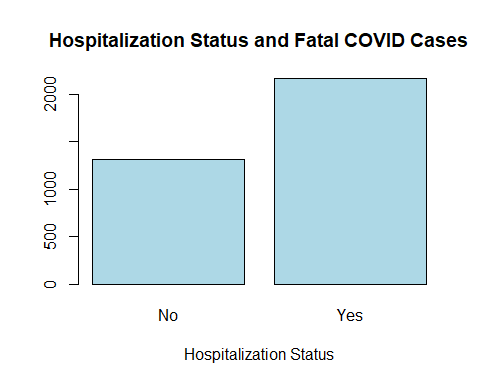
#bar plot for source vs outcome  
sou <- table(fatal$Outcome,fatal$Source.of.Infection)  
barplot(sou, main="Source of Infection and Fatal COVID Cases",  
 xlab="Source of Infection", col=c("lightblue"))



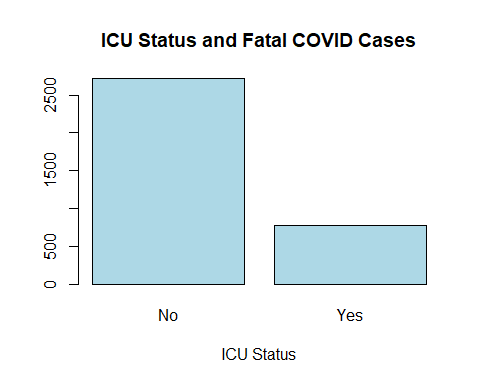
#bar plot for gender vs outcome  
gen <- table(fatal$Outcome,fatal$Client.Gender)  
barplot(gen, main="Gender and Fatal COVID Cases",  
 xlab="Gender", col=c("lightblue"))



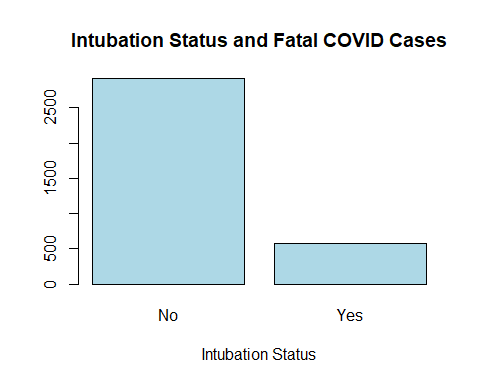
#bar plot for hospitalization vs outcome  
hou <- table(fatal$Outcome,fatal$Ever.Hospitalized)  
barplot(hou, main="Hospitalization Status and Fatal COVID Cases",  
 xlab="Hospitalization Status", col=c("lightblue"))



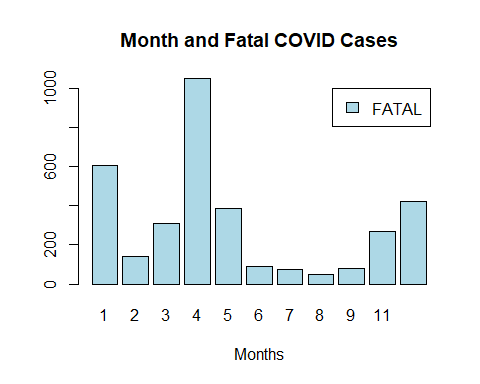
#bar plot for ICU vs outcome  
icu <- table(fatal$Outcome,fatal$Ever.in.ICU)  
barplot(icu, main="ICU Status and Fatal COVID Cases",  
 xlab="ICU Status", col=c("lightblue"))



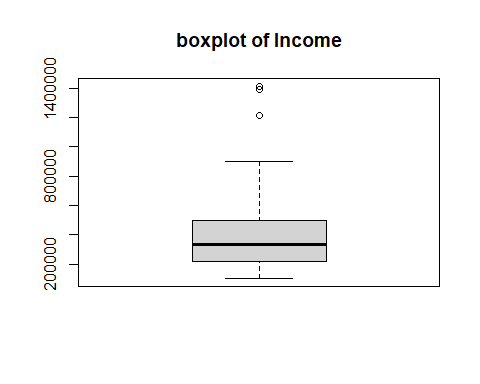
#bar plot for Intubation vs outcome  
int <- table(fatal$Outcome,fatal$Ever.Intubated)  
barplot(int, main="Intubation Status and Fatal COVID Cases",  
 xlab="Intubation Status", col=c("lightblue"))



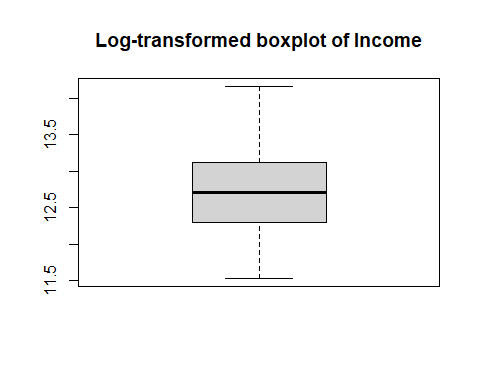
#Bar plot for Month  
mon <- table(fatal$Outcome,fatal$Month)  
barplot(mon, main="Month and Fatal COVID Cases",  
 xlab="Months", col=c("lightblue","pink"),  
 legend = rownames(mon), beside=FALSE)



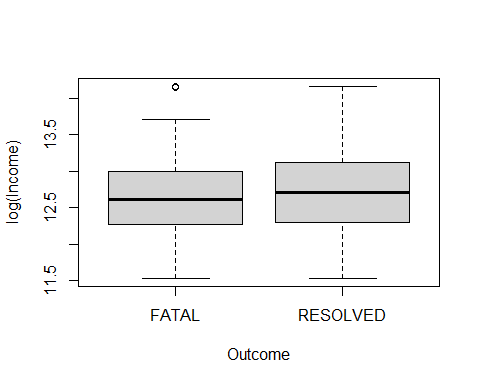
#boxplot for income  
boxplot(df3$Income,main="boxplot of Income")



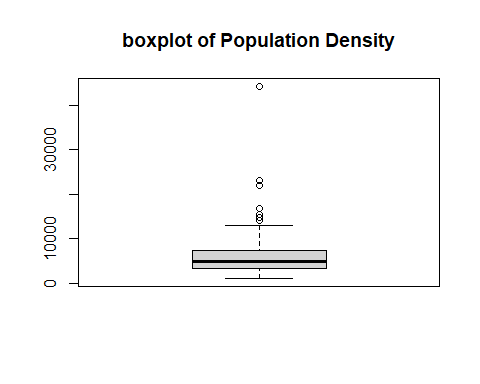
#log transform   
loginc <- log(as.data.frame(df3$Income))  
boxplot(loginc, main="Log-transformed boxplot of Income")



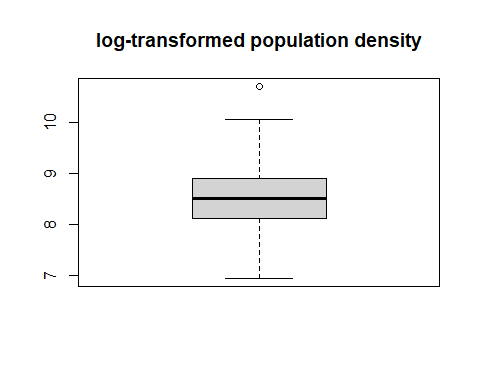
boxplot(log(Income)~Outcome,df3)



#boxplot for Density  
boxplot(df3$Density,main="boxplot of Population Density")



#log transform  
boxplot(log(df3$Density), main="log-transformed population density")



boxplot(log(Density)~Outcome,df3)

