PROJECT TITLE: NYC_PROPERTY_SALES_DATA_VISUALISATION

Video Link : https://youtu.be/hXpbsfcXG5A

OBJECTIVE:

a) To communicate insights from "NYC_PROPERTY_SALES DATA" from 2014 to 2018 through visual representation.
b) To allow for easy understanding of relationships between variables in the data.
c) To wade through data and see distinct patterns and make analysis and observations accordingly.

QUESTIONS:

a) How does total sales changes over time?
b) What are the categories from each variable that have higher sales?
c) How does total sales differ as for weeks?

DATA SOURCE:

The datasets were taken from 3rd-party data products from AWS Data Exchange providers.

Link: https://aws.amazon.com/marketplace/pp/prodview-27ompcouk2o6i?qid=1596373247748&sr=0-31&ref_=srh_res_product_title#overview

The data provider provides 5 years of dataset which contains 22 variables which are

- Borough name
- Borough
- Neighborhood
- Building class category
- Tax class at present
- Block
- Lot
- Ease-ment
- Building class at present
- Address
- Apartment number
- Zip code
- Residential units
- Commercial units
- Total units
- Land square feet
- Gross square feet
- Year built
- Tax class at time of sale
- Building class at time of sale
- Sale price
- Sale date

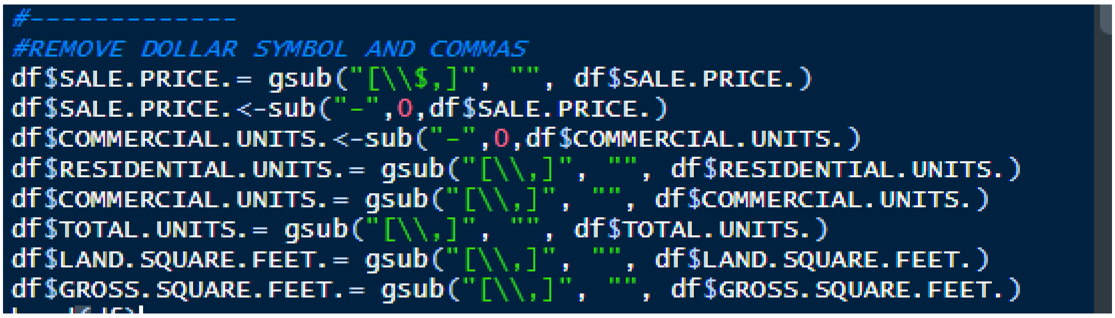
TOOL:

R programming language is used for the visualization and the flexdashboard library is used to create dashboard.

STEPS

i) DATA PREPARATION

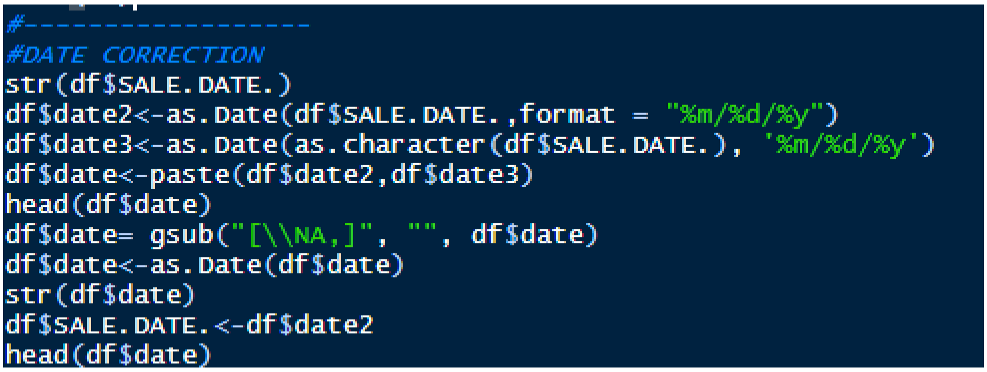
Some cleaning has to be done for the dataset where the unwanted commas, dollar signs and dash signs were removed. To improve to make sure the variables are saved in a proper datatype.



```
#-------------
#REMOVE DOLLAR SYMBOL AND COMMAS
df$SALE.PRICE.= gsub("[\\$,]", "", df$SALE.PRICE.)
df$SALE.PRICE.<-sub("-",0,df$SALE.PRICE.)
df$COMMERCIAL.UNITS.<-sub("-",0,df$COMMERCIAL.UNITS.)
df$RESIDENTIAL.UNITS.= gsub("[\\,]", "", df$RESIDENTIAL.UNITS.)
df$COMMERCIAL.UNITS.= gsub("[\\,]", "", df$COMMERCIAL.UNITS.)
df$TOTAL.UNITS.= gsub("[\\,]", "", df$TOTAL.UNITS.)
df$LAND.SQUARE.FEET.= gsub("[\\,]", "", df$LAND.SQUARE.FEET.)
df$GROSS.SQUARE.FEET.= gsub("[\\,]", "", df$GROSS.SQUARE.FEET.)
```

Figure 1: Example of code to remove them

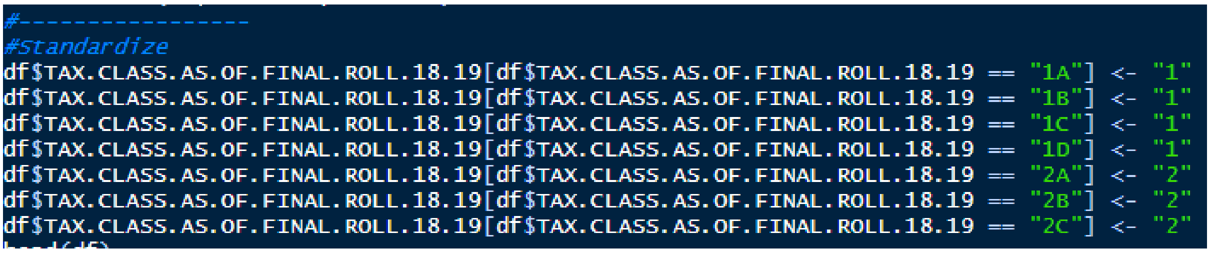
The sales date was one of the toughest parts to handle since it was so distorted.



```
#------------------
#DATE CORRECTION
str(df$SALE.DATE.)
df$date2<-as.Date(df$SALE.DATE.,format = "%m/%d/%y")
df$date3<-as.Date(as.character(df$SALE.DATE.), '%m/%d/%y')
df$date<-paste(df$date2,df$date3)
head(df$date)
df$date= gsub("[\\NA,]", "", df$date)
df$date<-as.Date(df$date)
str(df$date)
df$SALE.DATE.<-df$date2
head(df$date)
```

Figure 2: Example of code to correct the date

Some of the variables also needed standardization as it was all messed up in data collection



```
#----------------
#Standardize
df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19[df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19 == "1A"] <- "1"
df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19[df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19 == "1B"] <- "1"
df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19[df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19 == "1C"] <- "1"
df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19[df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19 == "1D"] <- "1"
df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19[df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19 == "2A"] <- "2"
df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19[df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19 == "2B"] <- "2"
df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19[df$TAX.CLASS.AS.OF.FINAL.ROLL.18.19 == "2C"] <- "2"
```

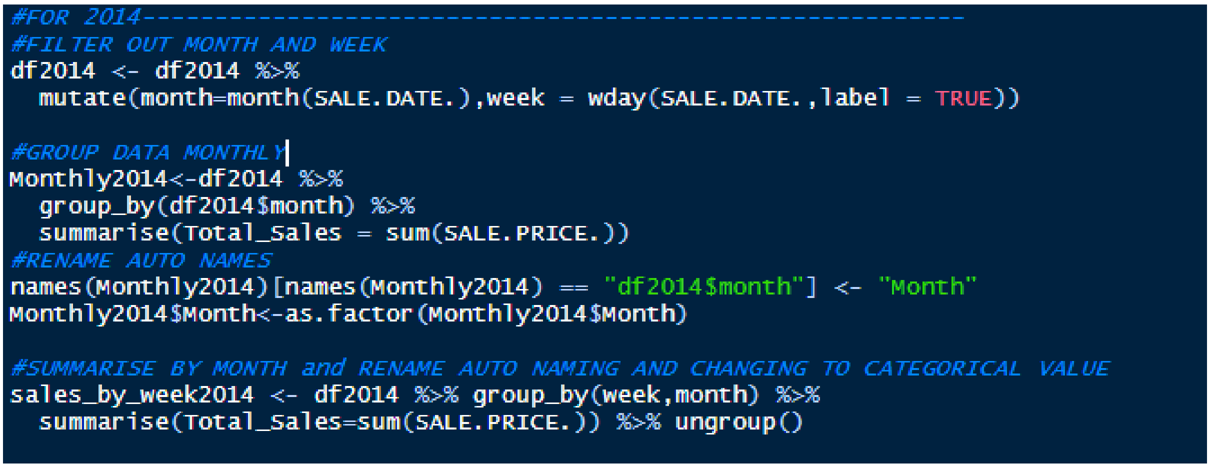
Figure 3: Example of code used for standardization

Some unwanted columns were also removed like address and lot number which are not much usable for the visualization purpose. All the dataset was cleaned, standardized and prepared separately and were saved as a modified file.

Finally, the data were saved according to month and year and saved too for further use. This is because some manual cleaning needed to be done for the dataset because the date for one of the datasets were severely disported.

ii) DATA VISUALISATION

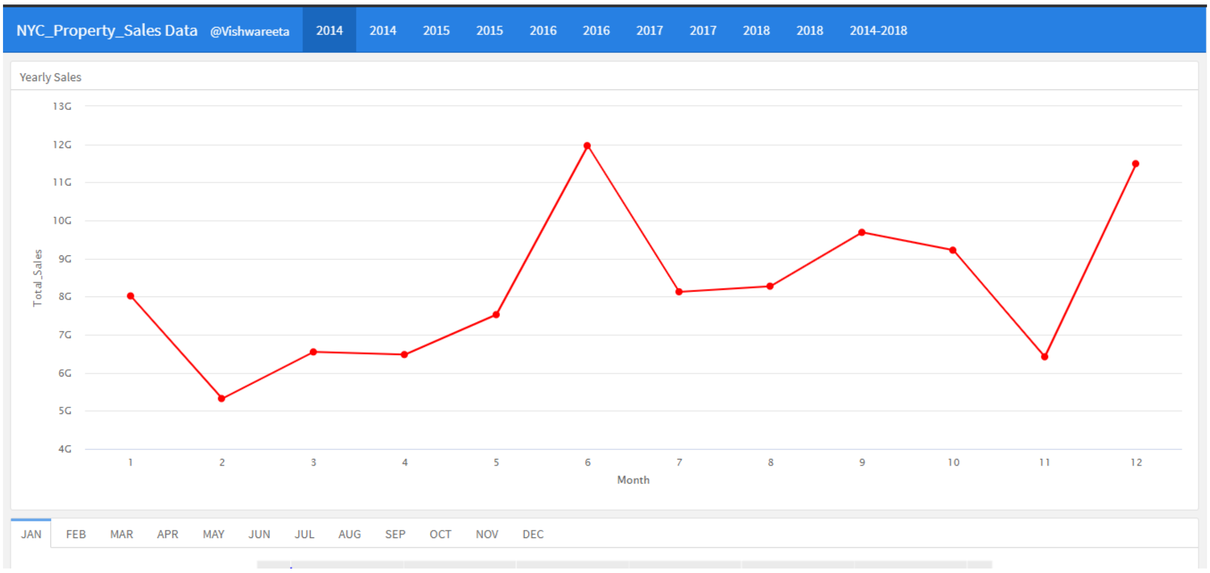
The visualization was done in dashboard which were saved in .Rmd file. The other important coding that needed to produce them were saved in separate .R files according to the year and called in the .Rmd file. This to make sure the coding is systematic and it will be easier to detect error or easier to make modifications.

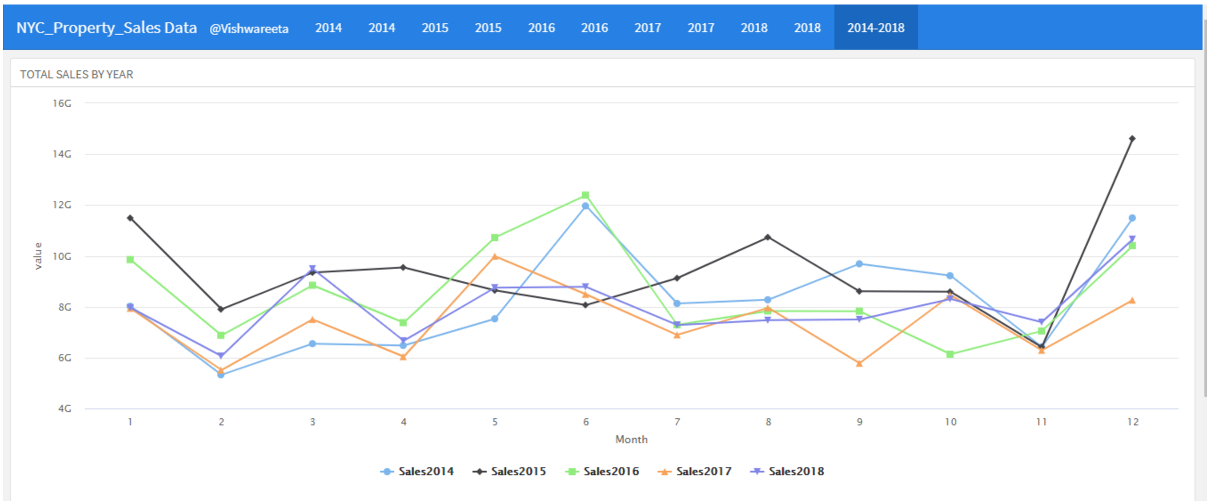


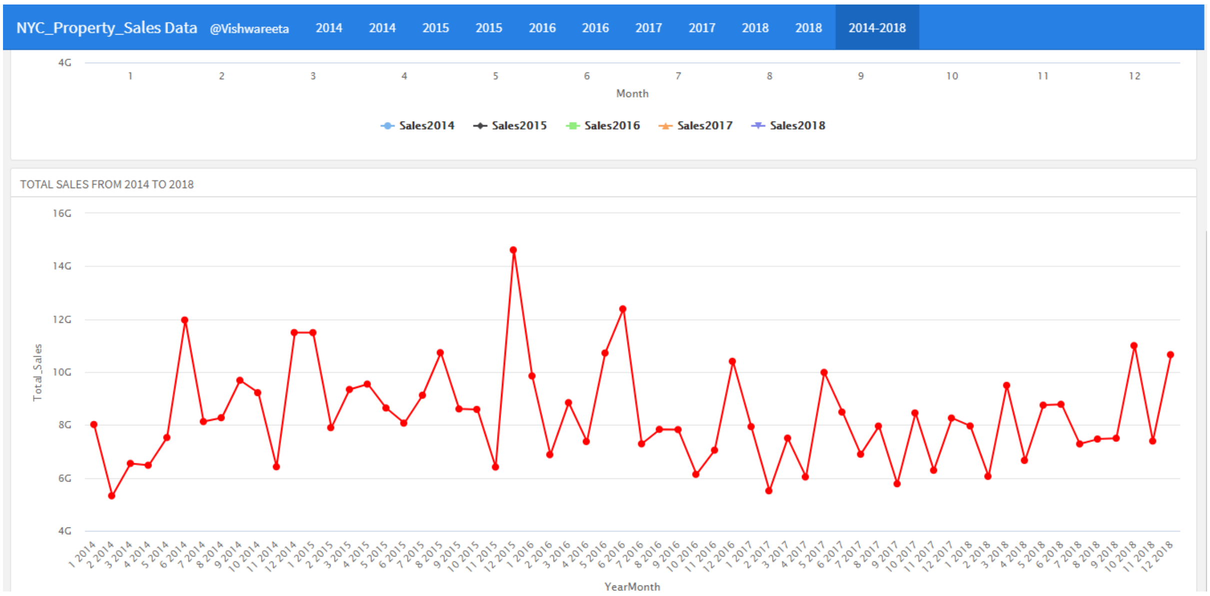
```
#FOR 2014-------------------------------------------------------
#FILTER OUT MONTH AND WEEK
df2014 <- df2014 %>%
  mutate(month=month(SALE.DATE.),week = wday(SALE.DATE.,label = TRUE))

#GROUP DATA MONTHLY
Monthly2014<-df2014 %>%
  group_by(df2014$month) %>%
  summarise(Total_Sales = sum(SALE.PRICE.))
#RENAME AUTO NAMES
names(Monthly2014)[names(Monthly2014) == "df2014$month"] <- "Month"
Monthly2014$Month<-as.factor(Monthly2014$Month)

#SUMMARISE BY MONTH and RENAME AUTO NAMING AND CHANGING TO CATEGORICAL VALUE
sales_by_week2014 <- df2014 %>% group_by(week,month) %>%
  summarise(Total_Sales=sum(SALE.PRICE.)) %>% ungroup()
```

Figure 4: The part of code to separate data according to week and month for time series plot

OUTPUT:







Every dataset has 2 pages of visualization. One to show time series plot and another to show visualization of other variables in the dataset. At last there is one last page to show time series visualization from 2014 to 2018.