#Import dataset

```r
BOROUGH <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/BOROUGH.xlsx")
NEIGHBORHOOD <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/NEIGHBORHOOD.xlsx")
BUILDING_CLASS <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/BUILDING_CLASS.xlsx
NYC_HISTORICAL <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/NYC_HISTORICAL.xlsx
```

#Install packages for analysis

```r
tinytex::install_tinytex()
```

```
## Warning: Detected an existing tlmgr at /usr/local/bin/tlmgr. It seems TeX
## Live has been installed (check tinytex::tinytex_root()). You are recommended
## to uninstall it, although TinyTeX should work well alongside another LaTeX
## distribution if a LaTeX document is compiled through tinytex::latexmk().

## TinyTeX installed to /Users/shimonyagrawal/Library/TinyTeX
```

```r
install.packages("readxl")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//Rtmp9bEnUj/downloaded_packages
```

```r
install.packages("DBI")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//Rtmp9bEnUj/downloaded_packages
```

```r
install.packages("odbc")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//Rtmp9bEnUj/downloaded_packages
```

```r
install.packages("tidyverse")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//Rtmp9bEnUj/downloaded_packages
```

```r
install.packages("lubridate")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//Rtmp9bEnUj/downloaded_packages
```

```r
install.packages("GGally")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//Rtmp9bEnUj/downloaded_packages
```

```r
install.packages("forecast")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//Rtmp9bEnUj/downloaded_packages
```

```r
library(readxl)
library(DBI)
library(odbc)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyvers

## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------------- tidyverse_con
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

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

```r
library (GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

##
## Attaching package: 'GGally'
```

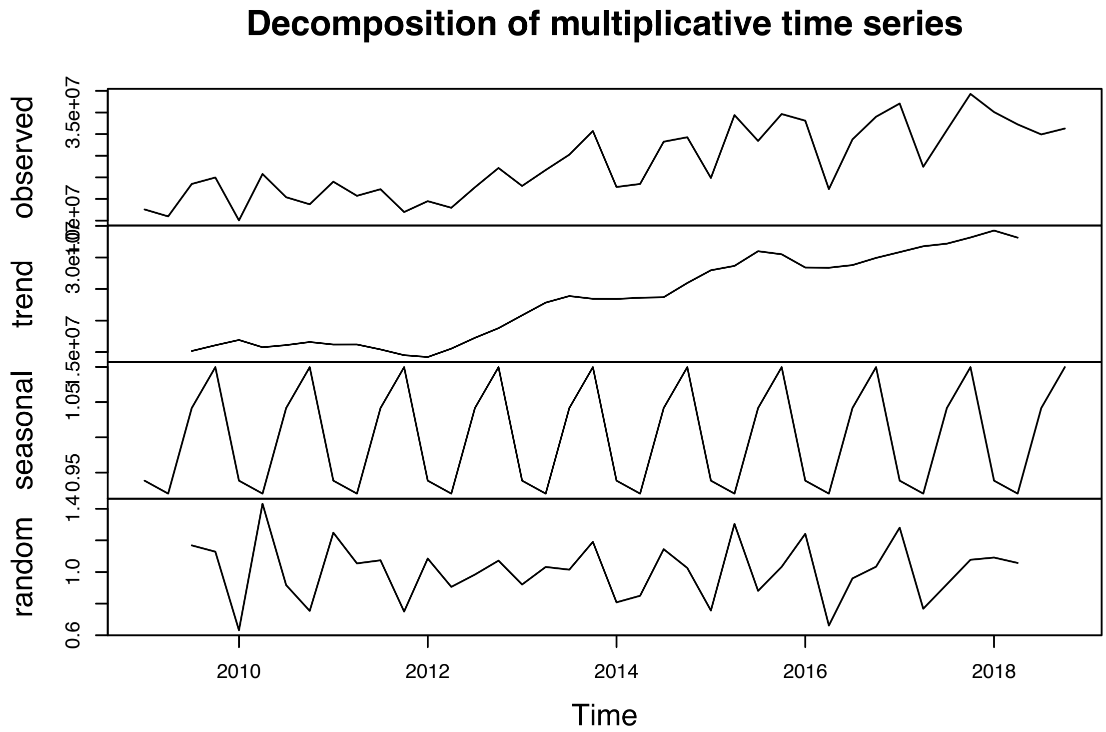
```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
library(forecast)
```

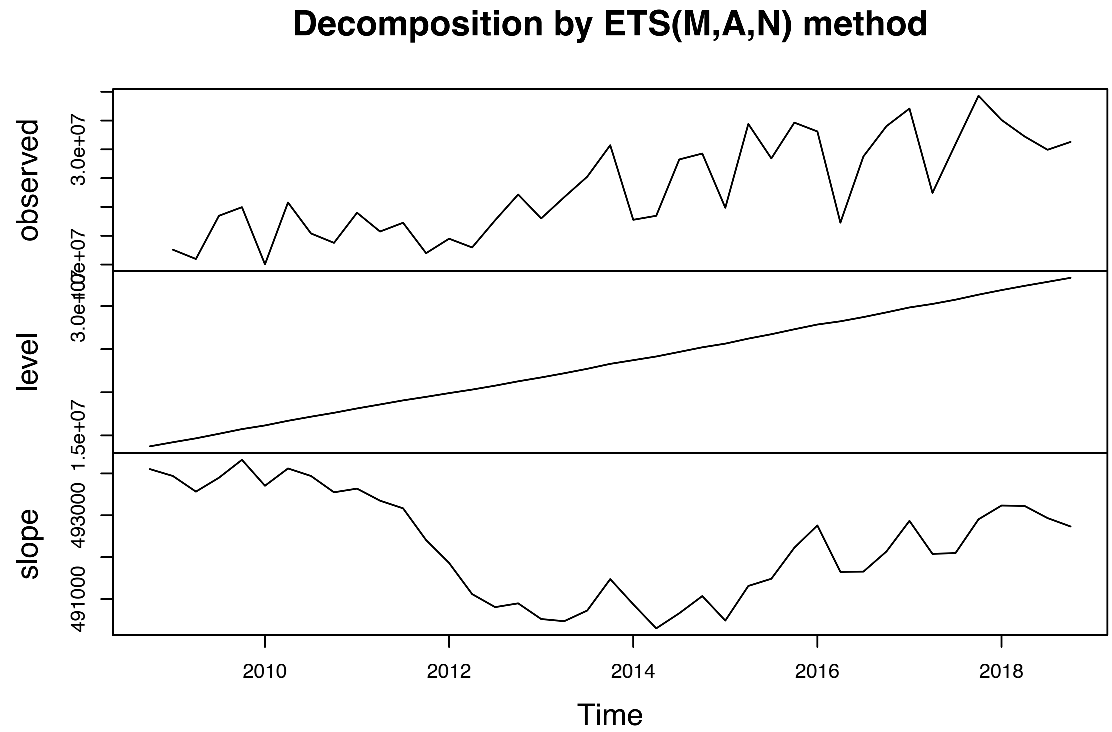
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#Predictive Statistics for Neighborhood Madison (149) to forecast sales for next 8 quarters
#create dataframe with required data and filter N/A or missing values

NYCdf <- NYC_HISTORICAL %>%
  left_join(NEIGHBORHOOD, by= "NEIGHBORHOOD_ID") %>%
  left_join(BUILDING_CLASS, by= c("BUILDING_CLASS_FINAL_ROLL"="BUILDING_CODE_ID")) %>%
  select (NEIGHBORHOOD_ID, NEIGHBORHOOD_NAME, SALE_DATE, SALE_PRICE, GROSS_SQUARE_FEET, RESIDENTIAL_UNI
  filter(SALE_PRICE >0, TYPE == "RESIDENTIAL", GROSS_SQUARE_FEET > 0 ) %>%
  mutate(Year = year(SALE_DATE), Quarter = quarter(SALE_DATE)) %>%
  select(TYPE, SALE_PRICE, Quarter, Year, NEIGHBORHOOD_ID)

view(NYCdf)
```

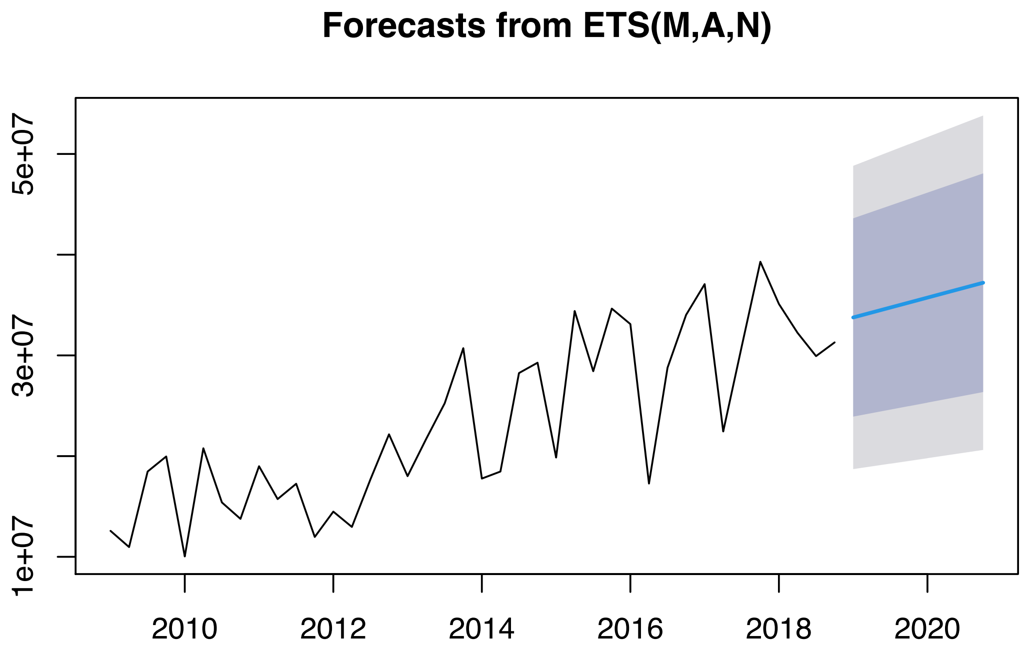
```
#Time series analysis on the total dollar amount of residential real estate sales in Madison using sale

forecast <- NYCdf %>%
  filter(NEIGHBORHOOD_ID == 149, SALE_PRICE > 0, Year > 2008) %>%
  mutate(t = as.numeric(Year)*4 + Quarter - 2009*4) %>%
  group_by(t) %>%
  summarise (TotalSales = sum(SALE_PRICE))


Timeseries_Madison <- ts(forecast$TotalSales, start = c(2009,1), frequency = 4)
ets_madison <- ets(Timeseries_Madison, model = "MAN")
Forecast_Madison <- forecast (ets_madison, 8)
decomposets <- decompose(Timeseries_Madison, type  = "multiplicative")

plot(decomposets)
```




```
plot(ets_madison)
```




```r
plot(Forecast_Madison)
```

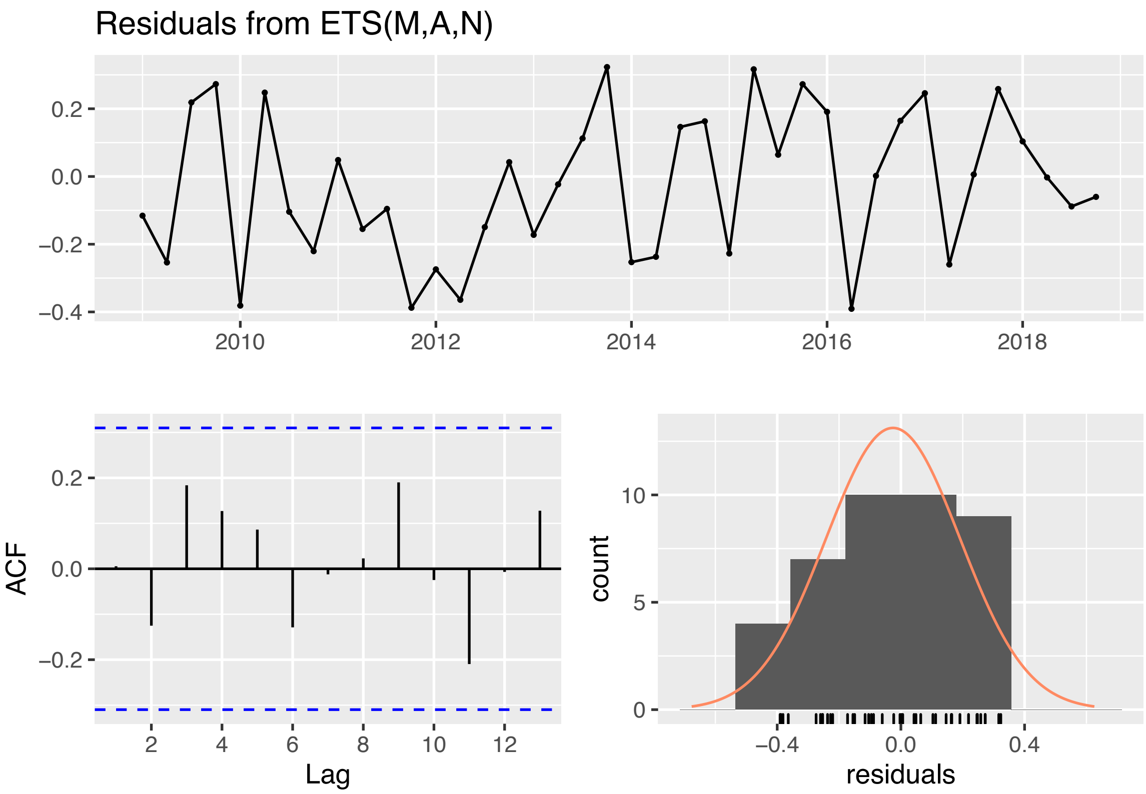



```
summary(Forecast_Madison)
```

```
##
## Forecast method: ETS(M,A,N)
##
## Model Information:
## ETS(M,A,N)
##
## Call:
##  ets(y = Timeseries_Madison, model = "MAN")
##
##   Smoothing parameters:
##     alpha = 0.011
##     beta  = 1e-04
##
##   Initial states:
##     l = 13732187.4752
##     b = 494103.341
##
##   sigma:  0.2275
##
##      AIC     AICc      BIC
## 1390.764 1392.529 1399.208
##
## Error measures:
```

```
##                     ME    RMSE     MAE       MPE     MAPE      MASE       ACF1
## Training set -342565.5 5000256 4227643 -8.138045 21.20466 0.8195974 0.01219138
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019 Q1       33765836 23921020 43610652 18709489 48822183
## 2019 Q2       34258569 24269463 44247675 18981550 49535588
## 2019 Q3       34751302 24617895 44884709 19253593 50249011
## 2019 Q4       35244035 24966315 45521755 19525619 50962451
## 2020 Q1       35736768 25314723 46158813 19797626 51675910
## 2020 Q2       36229501 25663119 46795883 20069614 52389388
## 2020 Q3       36722234 26011502 47432965 20341584 53102884
## 2020 Q4       37214967 26359874 48070060 20613534 53816400
```

```r
checkresiduals(Forecast_Madison)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,N)
## Q* = 4.1936, df = 4, p-value = 0.3804
##
## Model df: 4.   Total lags used: 8
```

```r
#Use a multiple regression model to come up with another forecast for the next 8 quarters of sales. Inc

forecast <- cbind(forecast, c("Q1", "Q2","Q3","Q4"))
names(forecast)[3] <- "Quarter"

#regression including time
regression_time <- lm ( data=forecast, formula = TotalSales~t)
summary(regression_time)
```

```
##
## Call:
## lm(formula = TotalSales ~ t, data = forecast)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -11459830  -3445107   -345006   4020272   7964821
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11535400    1607625   7.175 1.43e-08 ***
## t             573281      68332   8.390 3.54e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4989000 on 38 degrees of freedom
## Multiple R-squared:  0.6494, Adjusted R-squared:  0.6402
## F-statistic: 70.39 on 1 and 38 DF,  p-value: 3.537e-10
```

```r
x <- data.frame(t=c(41,42,43,44,45,46,47,48), TotalSales = c(0,0,0,0,0,0,0,0), Quarter=c("Q1", "Q2", "Q
predict.lm(regression_time, x, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 35039921 31785454 38294389
## 2 35613202 32237517 38988888
## 3 36186483 32688309 39684657
## 4 36759764 33137961 40381567
## 5 37333045 33586585 41079506
## 6 37906326 34034279 41778373
## 7 38479607 34481133 42478082
## 8 39052888 34927222 43178554
```

```r
#regression using time and seasonality
regression_timeseason <- lm(data = forecast, TotalSales~t+Quarter)
summary(regression_timeseason)
```

```
##
## Call:
## lm(formula = TotalSales ~ t + Quarter, data = forecast)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9124795 -3815801   510591  3139841 10338911
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11038305    1984713   5.562 2.92e-06 ***
## t             561321      66565   8.433 6.01e-10 ***
## QuarterQ2   -1566033    2164134  -0.724    0.474
## QuarterQ3    1211001    2167203   0.559    0.580
## QuarterQ4    3324139    2172309   1.530    0.135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4837000 on 35 degrees of freedom
## Multiple R-squared:  0.6964, Adjusted R-squared:  0.6617
## F-statistic: 20.07 on 4 and 35 DF,  p-value: 1.147e-08
```

```
y <- data.frame(t=c(41,42,43,44,45,46,47,48), TotalSales = c(0,0,0,0,0,0,0,0), Quarter = c("Q1", "Q2", 
predict.lm(regression_timeseason, y, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 34052463 29753575 38351351
## 2 33047750 28748863 37346638
## 3 36386105 32087218 40684993
## 4 39060565 34761677 43359453
## 5 36297746 31608758 40986734
## 6 35293034 30604046 39982022
## 7 38631389 33942401 43320377
## 8 41305848 36616860 45994836
```

```
#   Use a multiple regression model to determine the sale of a given residential property in your neigh
#a.Sale Date
#b.Year built
#c.Building type (categorical)
#d.Gross Square Feet
#e.Number of Units

NYC.df1 <- NYC_HISTORICAL %>%
  left_join(NEIGHBORHOOD, by= "NEIGHBORHOOD_ID") %>%
  left_join(BUILDING_CLASS, by= c("BUILDING_CLASS_FINAL_ROLL"="BUILDING_CODE_ID")) %>%
  select (NEIGHBORHOOD_ID, NEIGHBORHOOD_NAME, SALE_DATE, DESCRIPTION, YEAR_BUILT, ADDRESS, SALE_PRICE,GI
  filter(SALE_PRICE >0, TYPE == "RESIDENTIAL", GROSS_SQUARE_FEET > 0 ) %>%
  mutate (Year = year(SALE_DATE),Quarter = quarter(SALE_DATE))%>%
  select (BUILDING_CLASS_FINAL_ROLL, NEIGHBORHOOD_ID, TYPE,SALE_DATE, DESCRIPTION, ADDRESS, Year, YEAR_I

view(NYC.df1)

Madison_Regression <- NYC.df1 %>%
  filter (Year>2008, NEIGHBORHOOD_ID == 149) %>%
  select (SALE_DATE,YEAR_BUILT, SALE_PRICE, GROSS_SQUARE_FEET,BUILDING_CLASS_FINAL_ROLL,RESIDENTIAL_UNI'

Madison_Model <- lm(SALE_PRICE~. , data = Madison_Regression)
summary (Madison_Model)
```

```
##
```

```
## Call:
## lm(formula = SALE_PRICE ~ ., data = Madison_Regression)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1308950  -106884    10921   117966  1929248
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -5.624e+05  2.393e+05  -2.350 0.018938 *
## SALE_DATE                    1.458e-03  8.808e-05  16.552  < 2e-16 ***
## YEAR_BUILT                  -4.536e+02  1.024e+02  -4.429 1.03e-05 ***
## GROSS_SQUARE_FEET            2.284e+02  1.486e+01  15.374  < 2e-16 ***
## BUILDING_CLASS_FINAL_ROLLA2 -2.438e+04  8.905e+04  -0.274 0.784330
## BUILDING_CLASS_FINAL_ROLLA3  1.032e+06  1.671e+05   6.173 9.03e-10 ***
## BUILDING_CLASS_FINAL_ROLLA5 -2.133e+05  2.447e+04  -8.717  < 2e-16 ***
## BUILDING_CLASS_FINAL_ROLLA9 -1.258e+05  2.962e+04  -4.247 2.33e-05 ***
## BUILDING_CLASS_FINAL_ROLLB1 -1.377e+04  3.131e+04  -0.440 0.660029
## BUILDING_CLASS_FINAL_ROLLB2  1.298e+05  3.452e+04   3.761 0.000177 ***
## BUILDING_CLASS_FINAL_ROLLB3  1.191e+05  3.403e+04   3.499 0.000483 ***
## BUILDING_CLASS_FINAL_ROLLB9 -1.916e+04  7.760e+04  -0.247 0.804998
## BUILDING_CLASS_FINAL_ROLLC0  1.967e+05  5.325e+04   3.694 0.000230 ***
## BUILDING_CLASS_FINAL_ROLLC2  4.052e+05  2.939e+05   1.379 0.168244
## BUILDING_CLASS_FINAL_ROLLC3  4.155e+05  9.851e+04   4.218 2.64e-05 ***
## BUILDING_CLASS_FINAL_ROLLD1  1.815e+06  3.529e+05   5.143 3.14e-07 ***
## BUILDING_CLASS_FINAL_ROLLD3 -1.692e+06  2.793e+05  -6.058 1.82e-09 ***
## RESIDENTIAL_UNITS           -2.148e+05  2.311e+04  -9.293  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 278200 on 1255 degrees of freedom
## Multiple R-squared:  0.4827, Adjusted R-squared:  0.4757
## F-statistic: 68.87 on 17 and 1255 DF,  p-value: < 2.2e-16
```

```r
#Properties that are the biggest bargains and most expensive

Madison_Address <- NYC.df1 %>%
  filter (Year>2008, NEIGHBORHOOD_ID == 149) %>%
  select (ADDRESS,DESCRIPTION)

Madison_Regression["Residuals"] <- Madison_Model$residuals
Madison_Regression["Address"] <- Madison_Address$ADDRESS
Madison_Regression["Description"] <- Madison_Address$DESCRIPTION
```