615FinalProject

Kaiyu Yan

December 11, 2018

Introduction

For this project, I'm investigarte dataset and trying to determine if it was fabricated or altered in some way.

Description of data

The data is NYC Manhattan property sale in year 2017 that obtained from NYC Department of Finance https://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page. The data include the following useful variables for our future analysis:

1)NEIGHBORHOOD.

2)BUILDING CLASS CATEGORY.

3)BUILDING CLASS AT TIME OF SALE.

4)ZIPCODE.

5)LAND SQUARE FEET.

6)GROSS SQUARE FEET.

7)YEAR BUILT.

8)SALE PRICE.

9)SALE DATE.

The interesting variable I am using in the dataset are Sale price,neighborhood, land and gross square feet, and sale date.

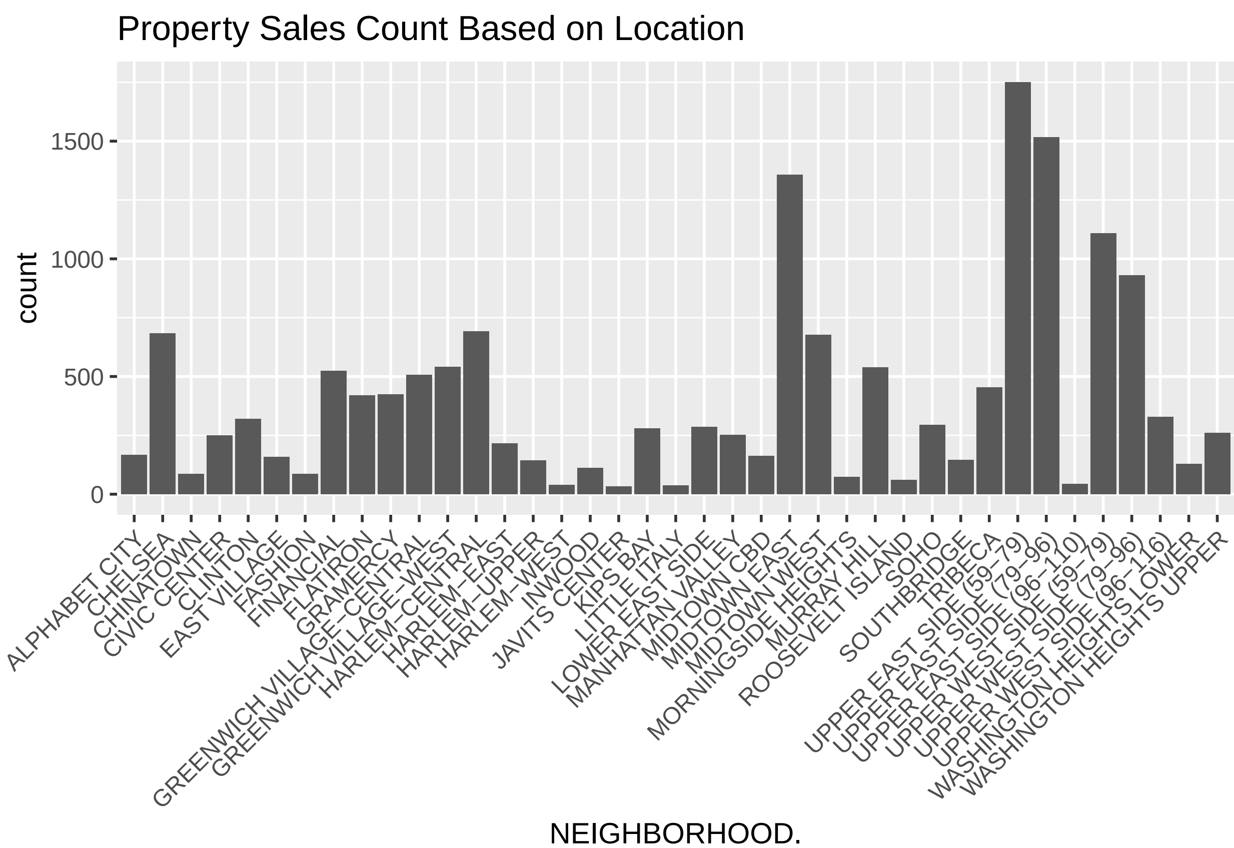
For Benford analysis, I will analyze the sale price and see if the price was faked or not.

EDA

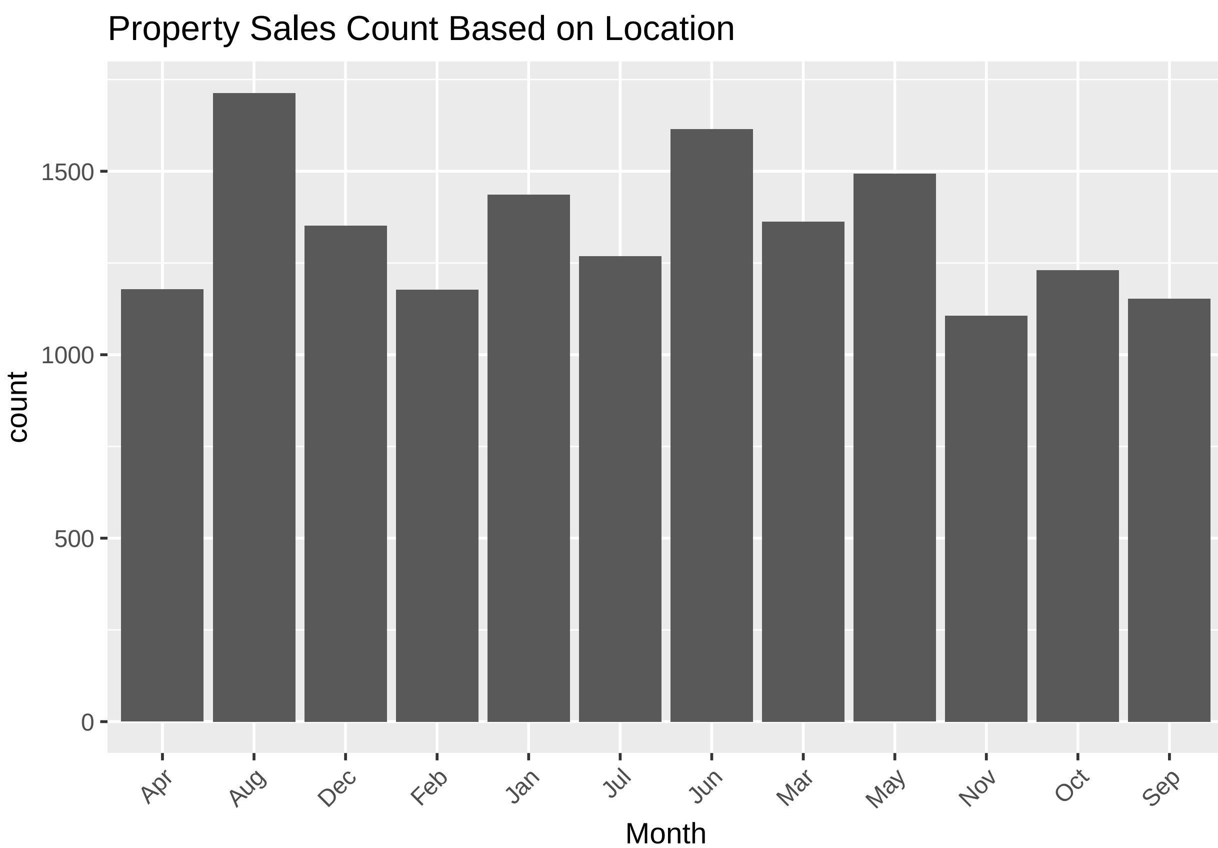
Data Import and Cleaning

```
Manhattan <- read.csv("2017_manhattan.csv") %>%
  select(2,9,11,12,13,15,16,17,19,20,21)
```

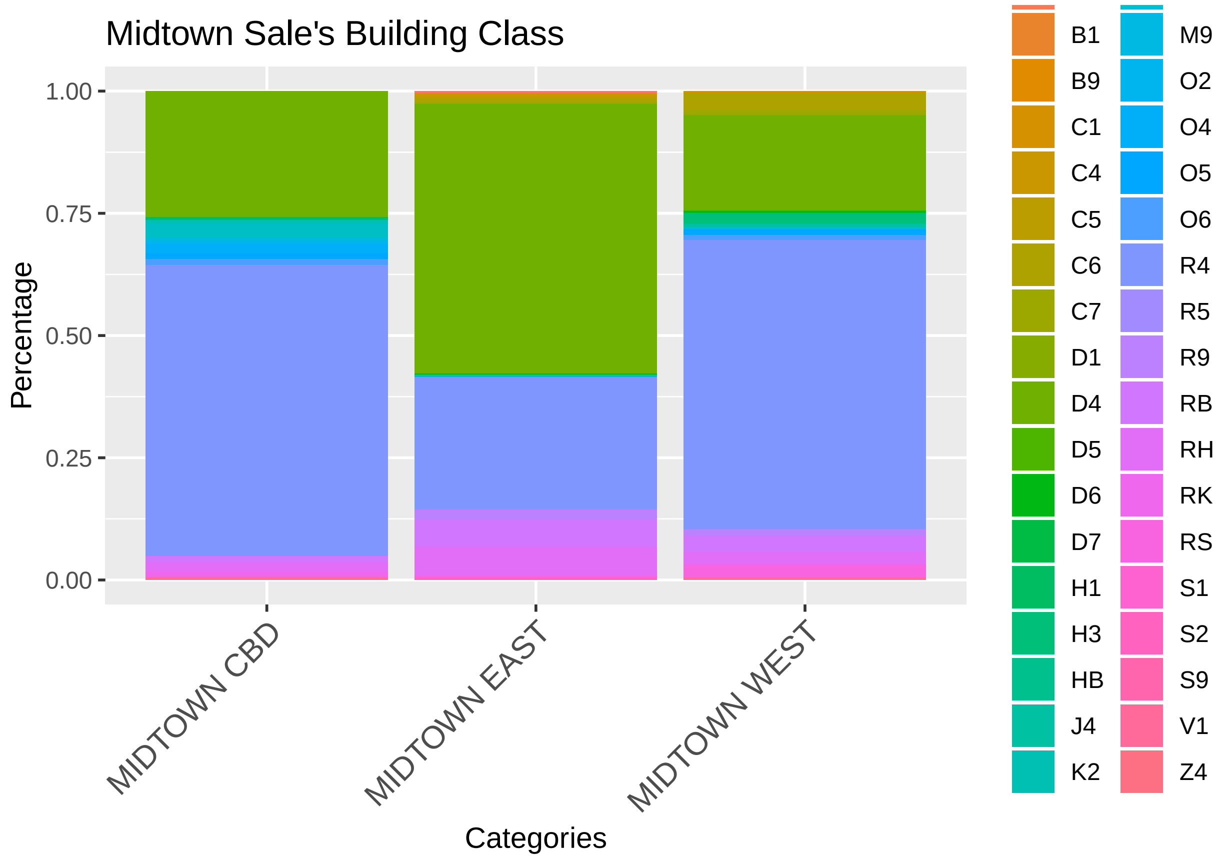
```
ggplot(Manhattan)+
   geom_bar(mapping = aes(x = NEIGHBORHOOD.))+
   theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ggtitle("Property Sales Count Based on Location ")
```



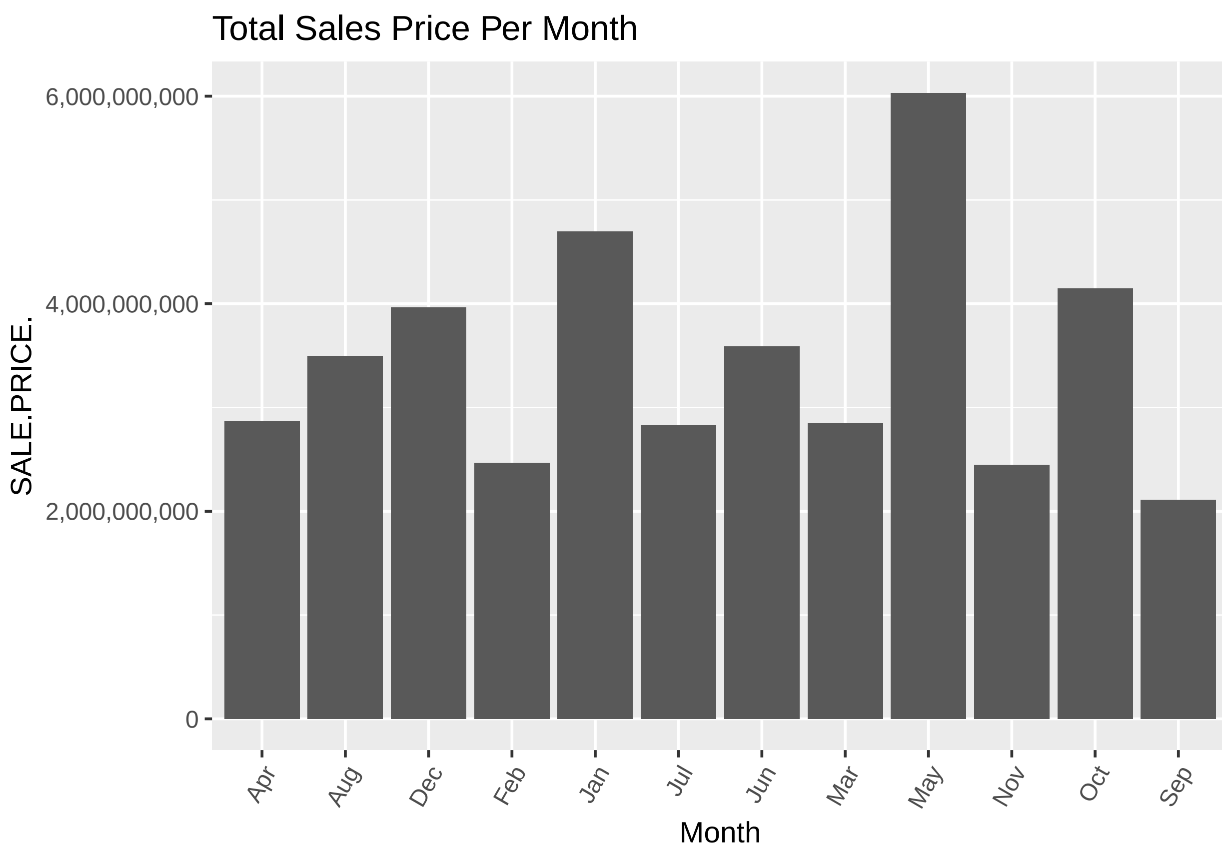

```r
ggplot(Manhattan)+
   geom_bar(mapping = aes(x = Month))+
   theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ggtitle("Property Sales Count Based on Location ")
```



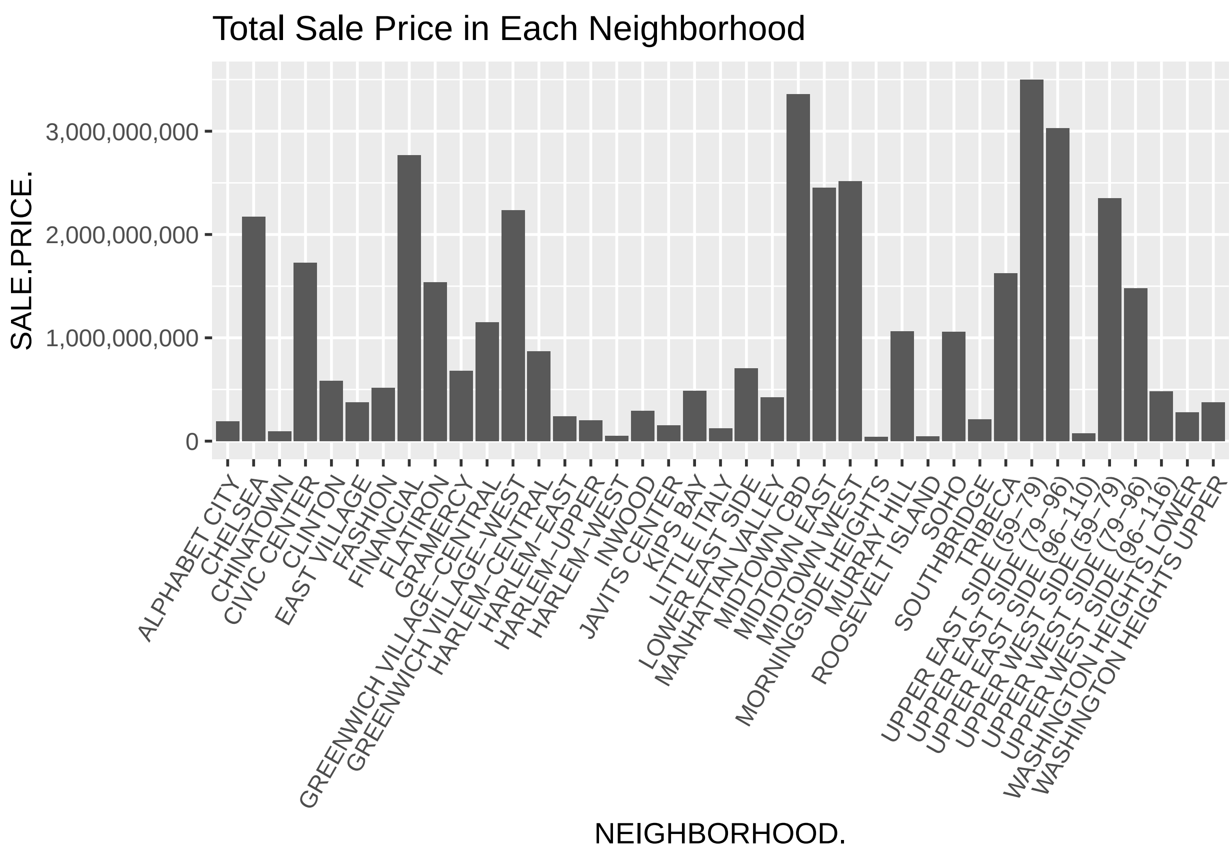

```r
A <- ggplot(Midtown,aes(NEIGHBORHOOD., fill =BUILDING.CLASS.AT.TIME.OF.SALE.)) +
        geom_bar(position = "fill") +

        xlab ("Categories") +
        ylab("Percentage") +
        theme(axis.text.x = element_text(size = 12, angle =45, hjust = 1)) +
        guides(fill=guide_legend(title="Answers"))+
  ggtitle("Midtown Sale's Building Class")
A
```



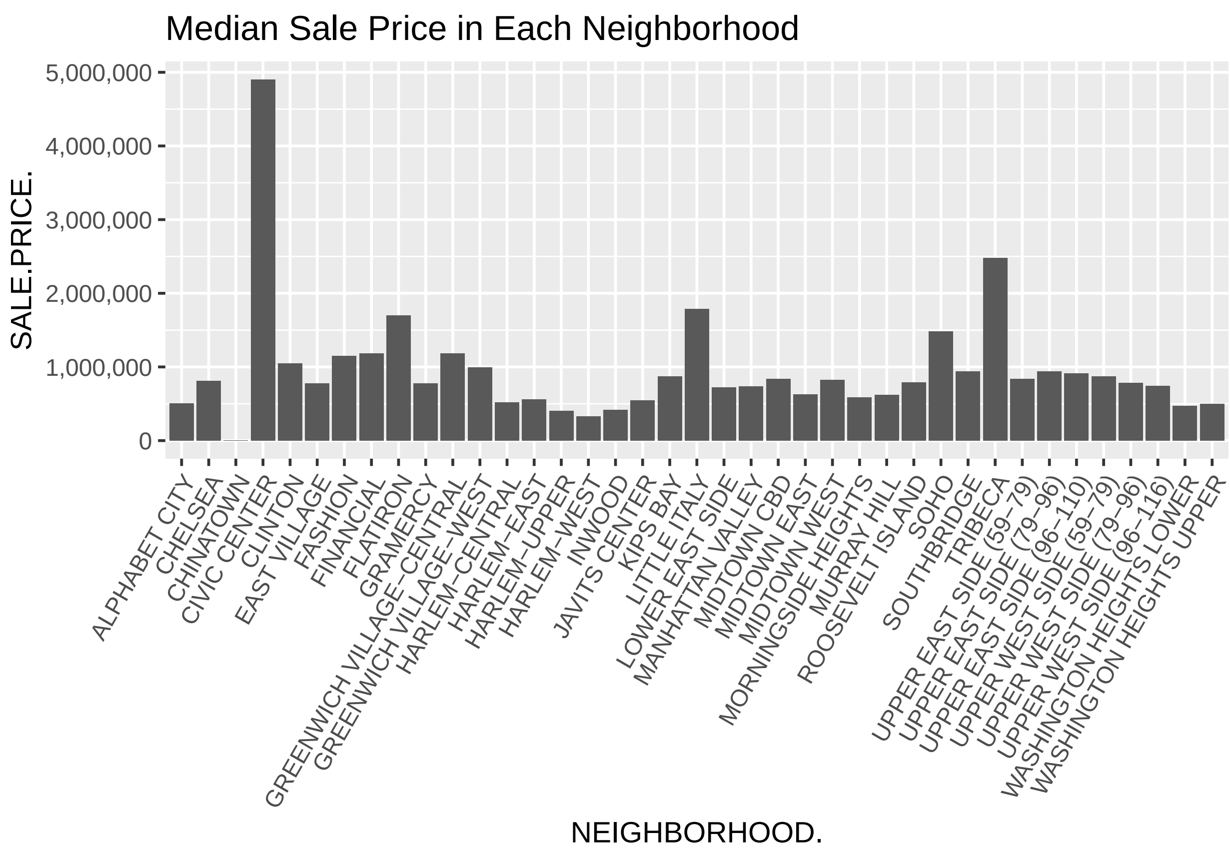

```
ggplot(data = Manhattan,
  aes(Month, SALE.PRICE.)) +
  stat_summary(fun.y = sum, # adds up all observations for the month
    geom = "bar") + # or "line"
  # custom x-axis labels
   scale_y_continuous(labels = comma)+
 theme(axis.text.x = element_text(angle =60, hjust = 1))+
  ggtitle("Total Sales Price Per Month")
```



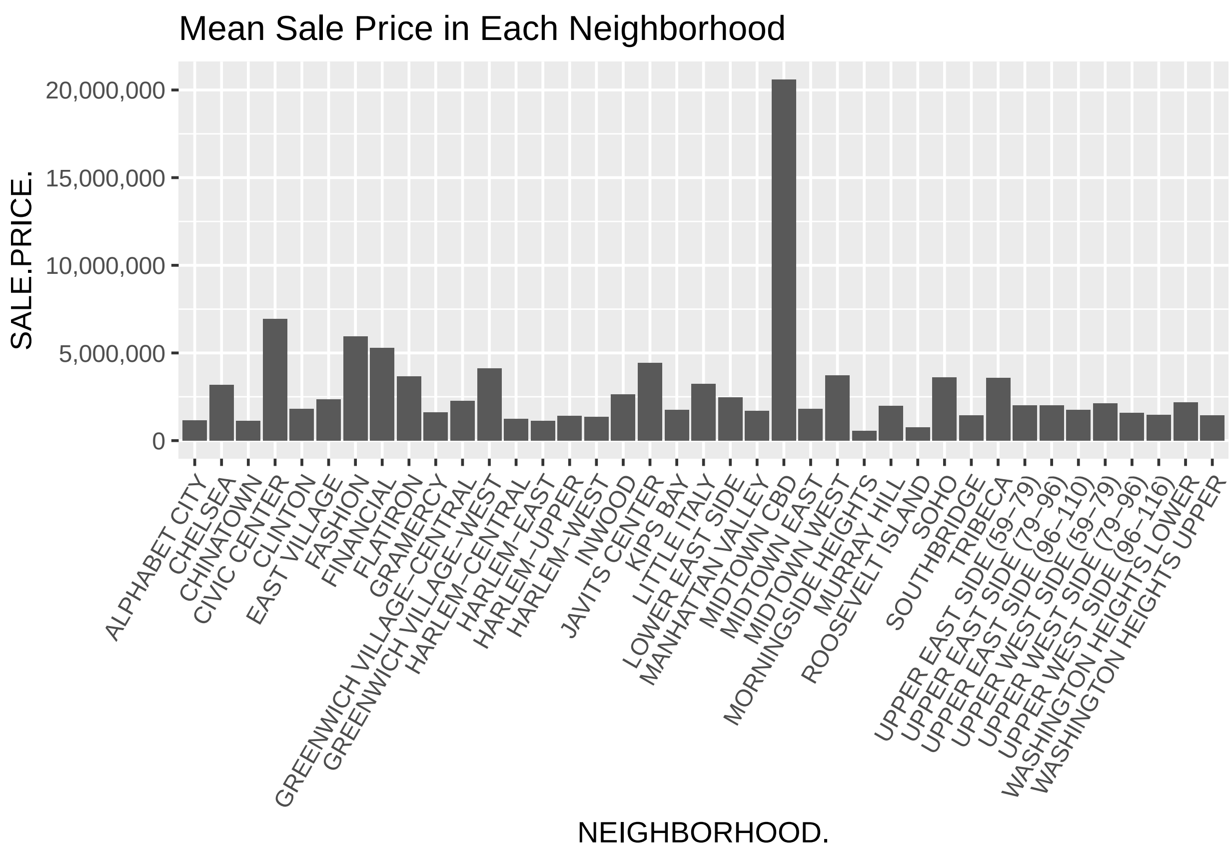

```r
ggplot(data = Manhattan,
  aes(NEIGHBORHOOD., SALE.PRICE.)) +
  scale_y_continuous(labels = comma)+
  stat_summary(fun.y = sum, # adds up all observations for the month
    geom = "bar") + # or "line"
   theme(axis.text.x = element_text(angle =60, hjust = 1))+
  ggtitle("Total Sale Price in Each Neighborhood")
```



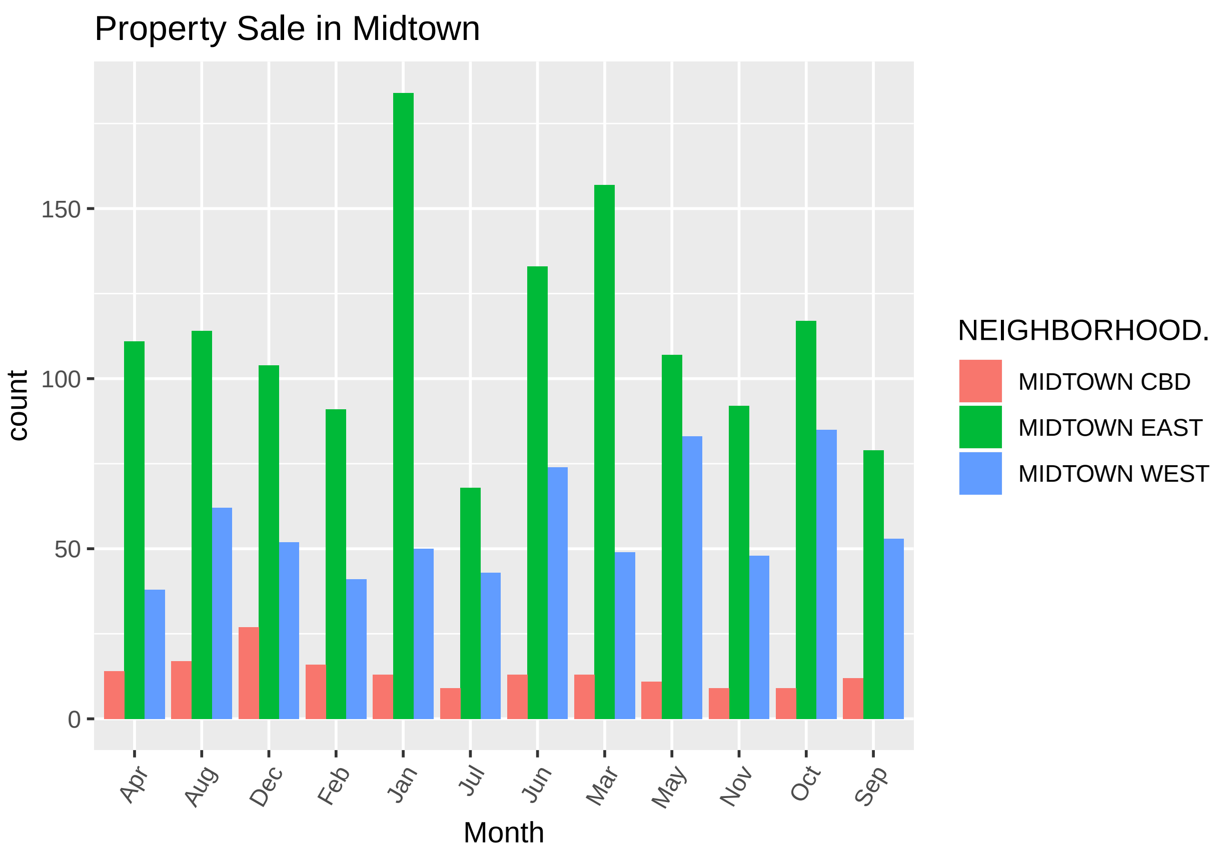

```r
ggplot(data = Manhattan,
  aes(NEIGHBORHOOD., SALE.PRICE.)) +
  scale_y_continuous(labels = comma)+
  stat_summary(fun.y = median, # adds up all observations for the month
    geom = "bar") +
   theme(axis.text.x = element_text(angle =60, hjust = 1))+
  ggtitle("Median Sale Price in Each Neighborhood")
```




```r
ggplot(data = Manhattan,
  aes(NEIGHBORHOOD., SALE.PRICE.)) +
  scale_y_continuous(labels = comma)+
  stat_summary(fun.y = mean, # adds up all observations for the month
    geom = "bar") +
   theme(axis.text.x = element_text(angle =60, hjust = 1))+
  ggtitle("Mean Sale Price in Each Neighborhood")
```

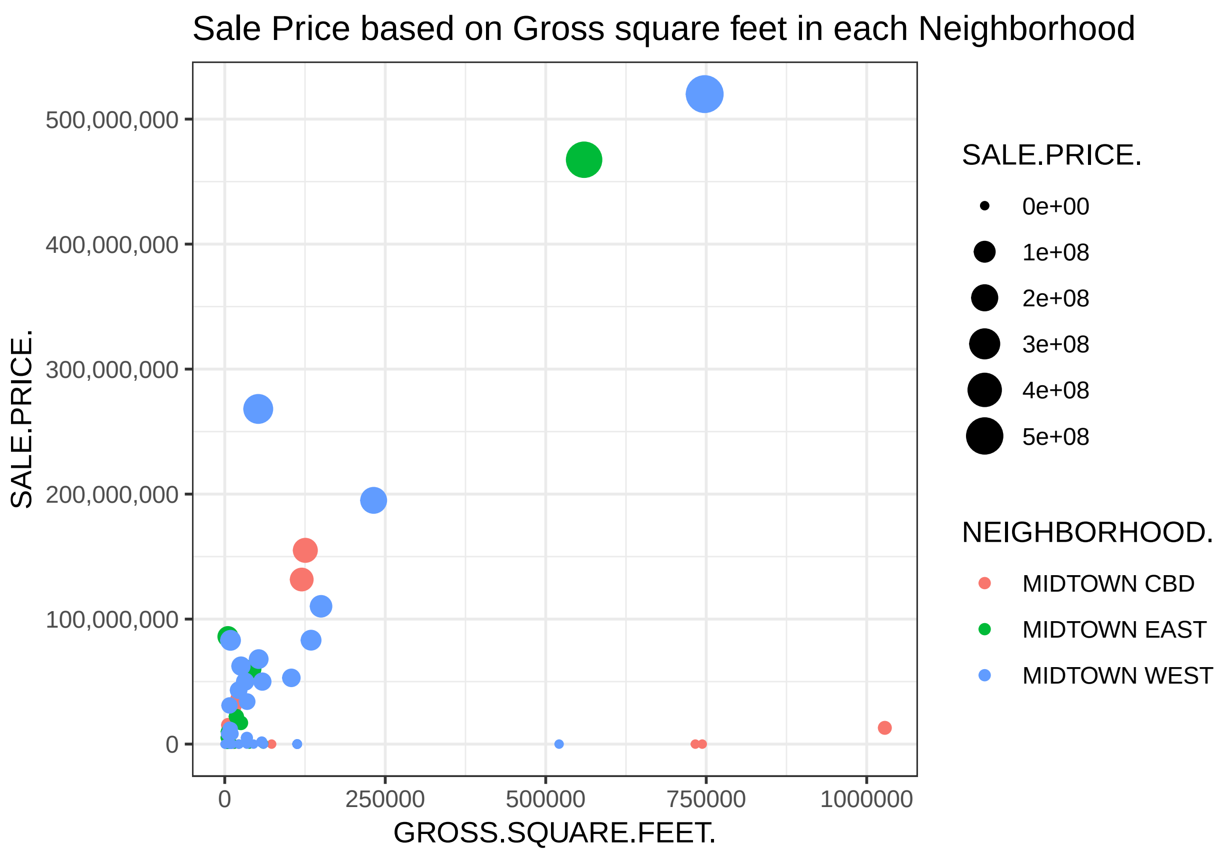



```
ggplot(data = Midtown,
  aes(Month, fill = NEIGHBORHOOD.)) +
   theme(axis.text.x = element_text(angle =60, hjust = 1))+
   geom_bar(position = "dodge")+
  ggtitle("Property Sale in Midtown")
```



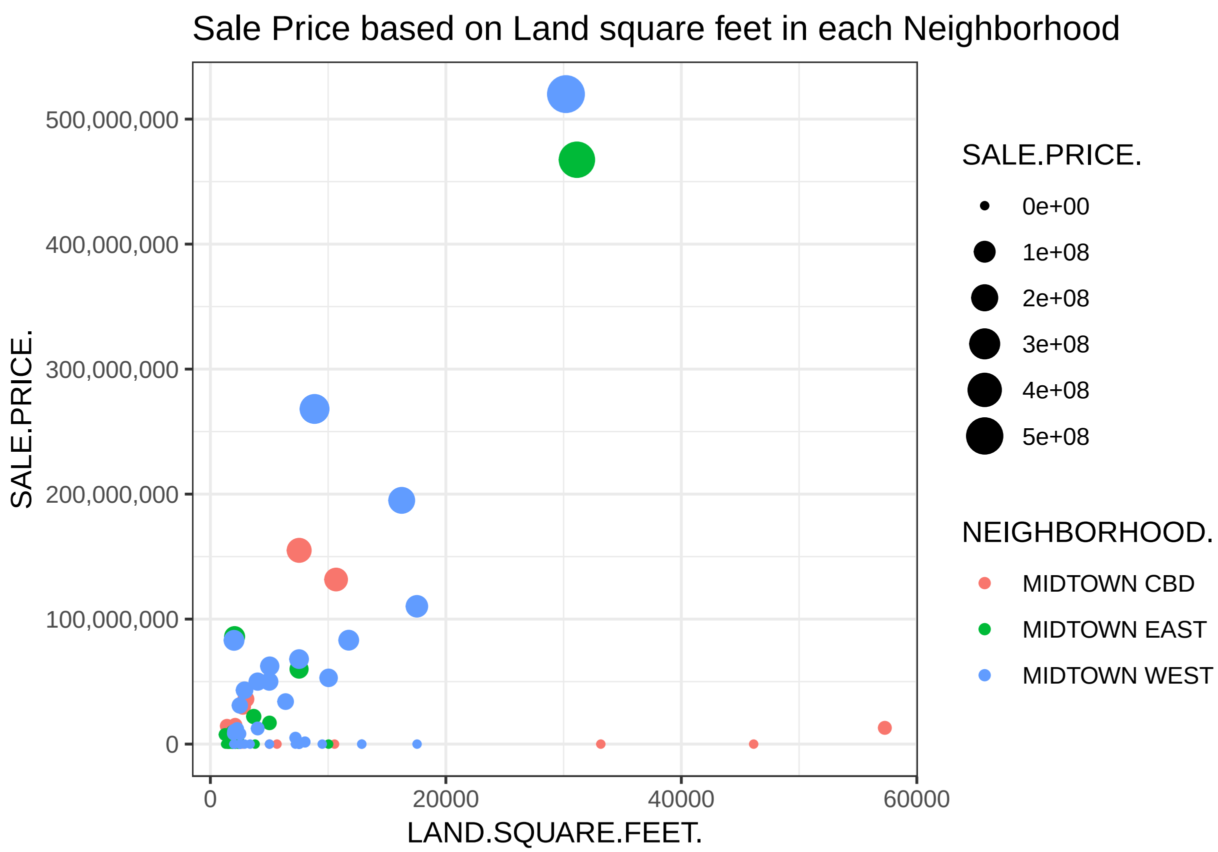

```
p <-  Midtown%>%
  filter(SALE.PRICE.<2210000000) %>% ggplot( aes(GROSS.SQUARE.FEET., SALE.PRICE., size = SALE.PRICE., co
  geom_point() +
  scale_y_continuous(labels = comma)+
    theme_bw()+
  ggtitle("Sale Price based on Gross square feet in each Neighborhood")
p
```

```
## Warning: Removed 2094 rows containing missing values (geom_point).
```

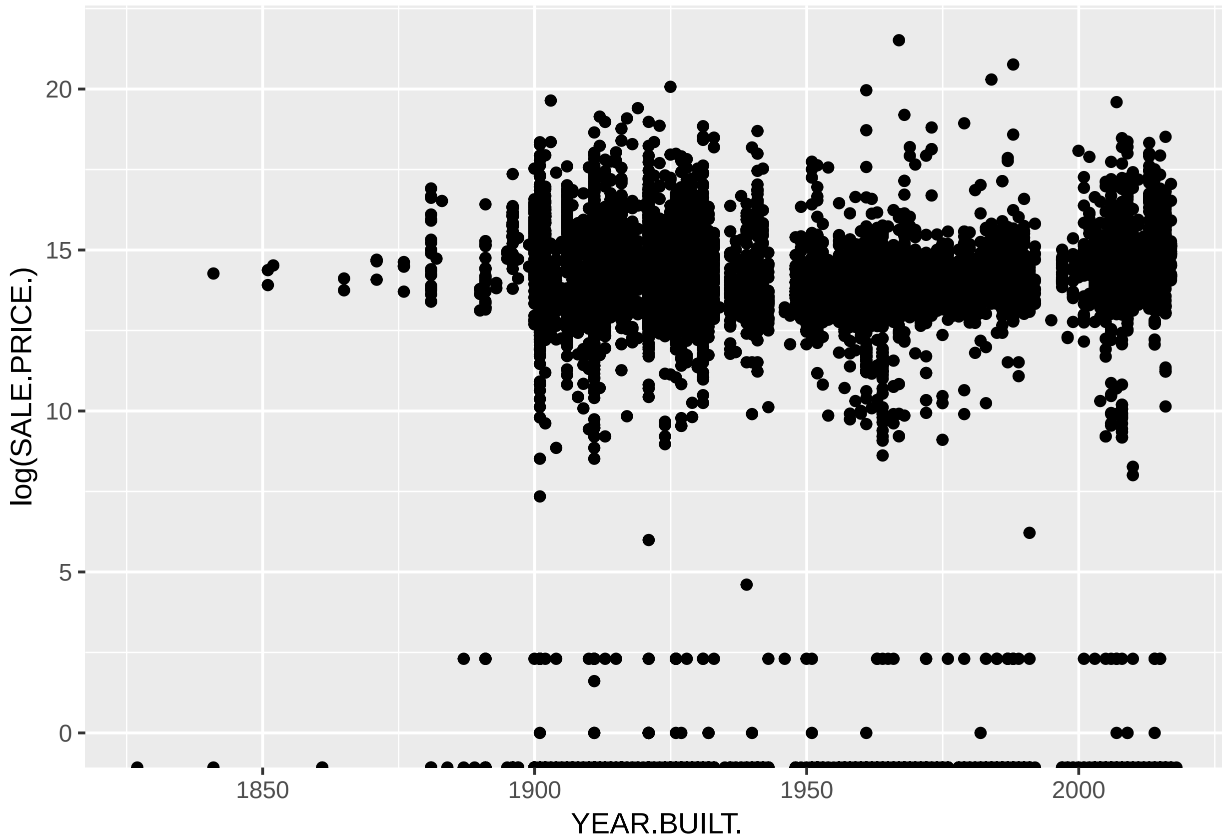



```
gf <-  Midtown%>%
  filter(SALE.PRICE.<2210000000) %>% ggplot( aes(LAND.SQUARE.FEET., SALE.PRICE., size = SALE.PRICE., co
  geom_point() +
  scale_y_continuous(labels = comma)+
    theme_bw()+
  ggtitle("Sale Price based on Land square feet in each Neighborhood")
gf
```

```
## Warning: Removed 2088 rows containing missing values (geom_point).
```




```r
ggplot(Manhattan)+aes(x=YEAR.BUILT.,y=log(SALE.PRICE.))+geom_point()
```



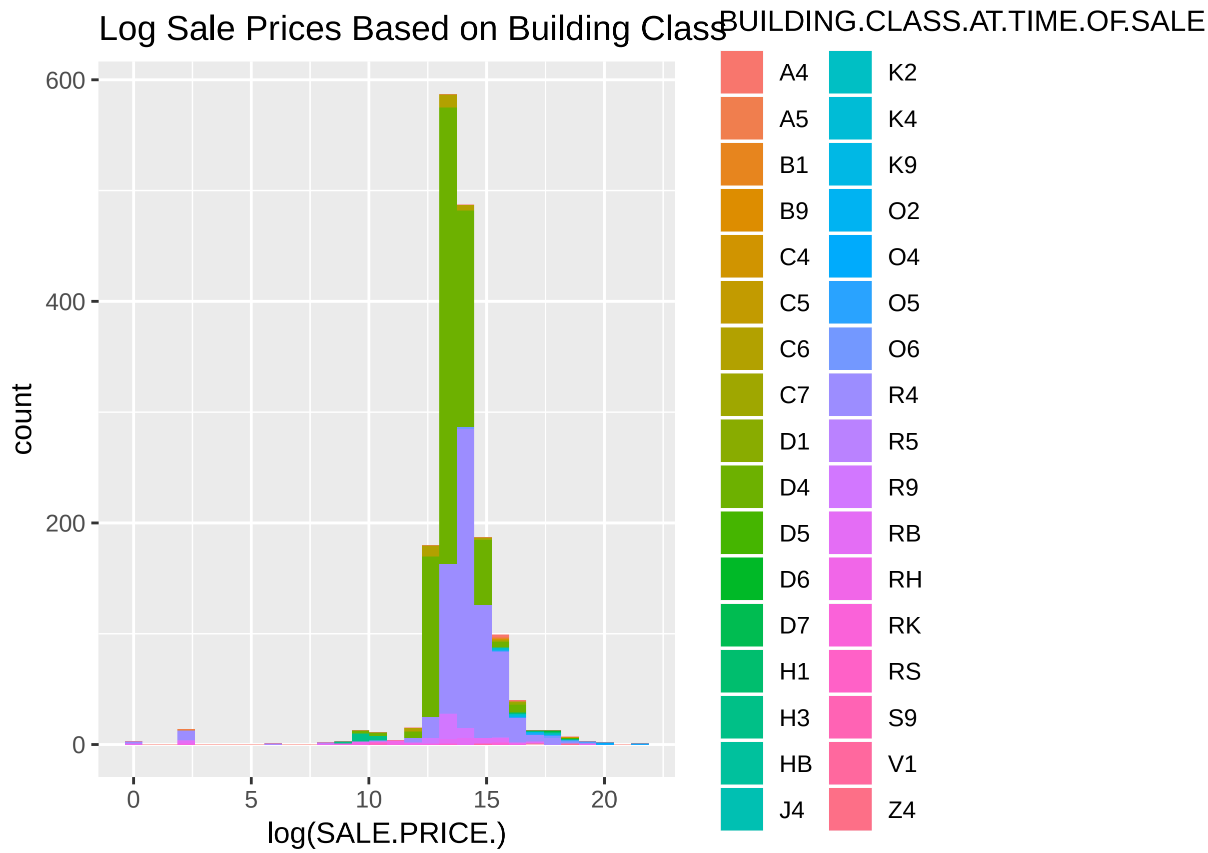

```r
B <- ggplot(data = Midtown,aes(x=log(SALE.PRICE.),fill = NEIGHBORHOOD.))+
  geom_histogram()+
  ggtitle("Log Sale Prices Based on Location")
B
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 513 rows containing non-finite values (stat_bin).
```

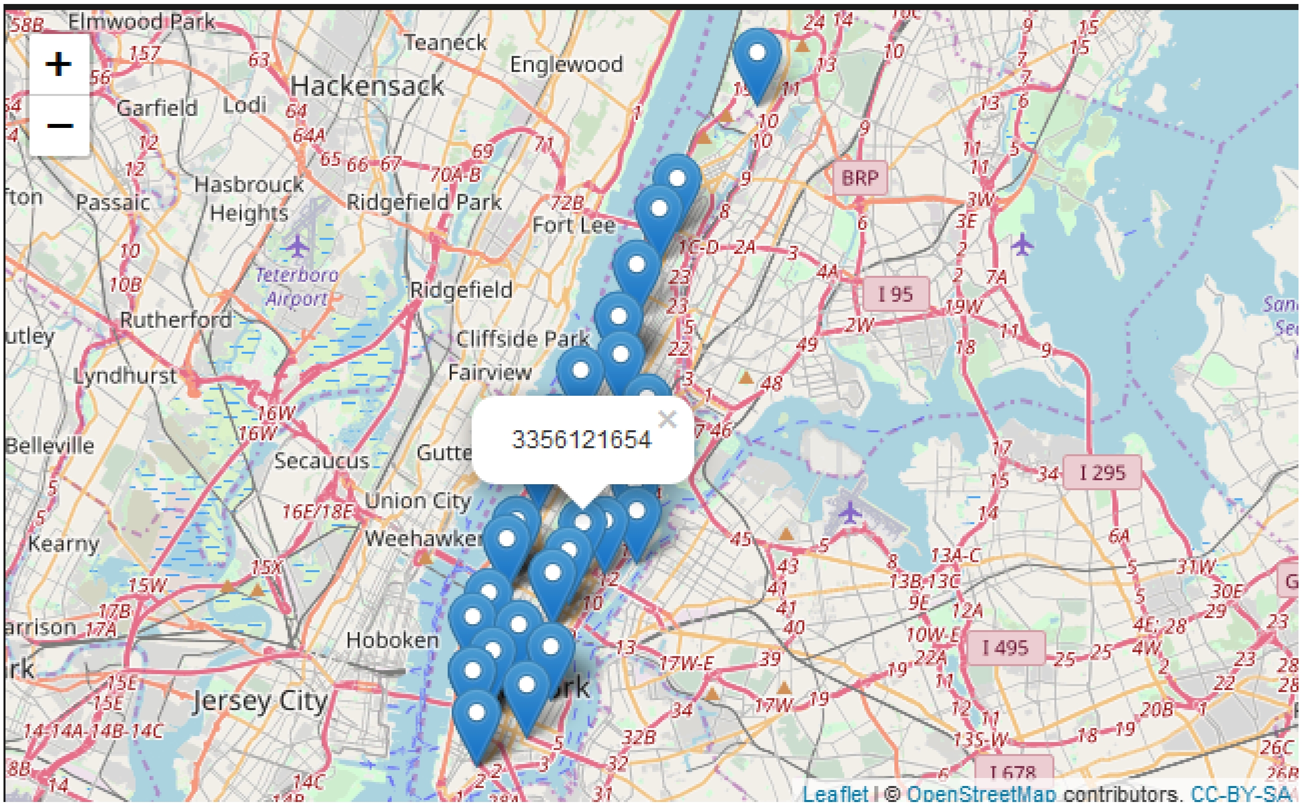

```
C <- ggplot(data = Midtown,aes(x=log(SALE.PRICE.),fill = BUILDING.CLASS.AT.TIME.OF.SALE.))+
  geom_histogram()+
  ggtitle("Log Sale Prices Based on Building Class")
C
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 513 rows containing non-finite values (stat_bin).
```



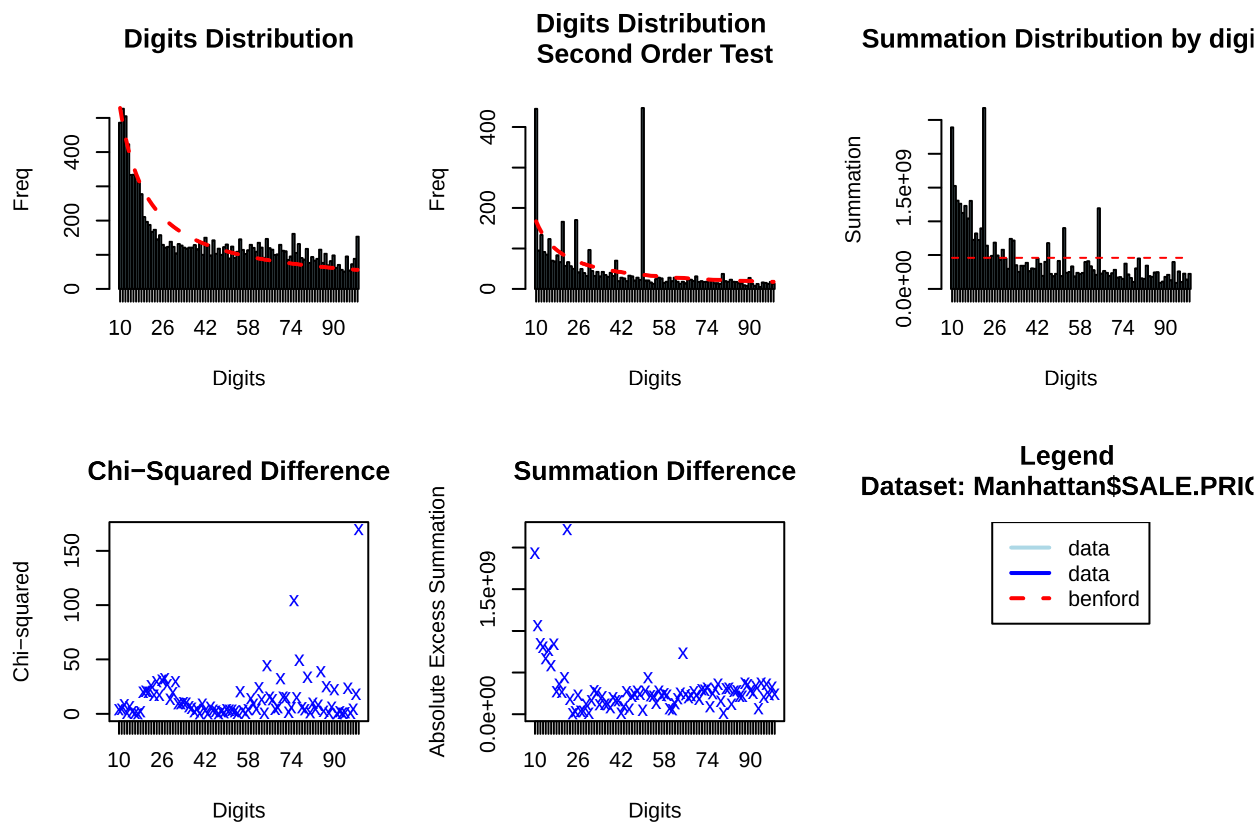

```
data("zipcode")
colnames(Manhattan)[3] <- "zip"
Manhattan$zip <- as.character(Manhattan$zip)
Manhattan_zip <- inner_join(Manhattan,zipcode,by="zip")
Manhattan_zip$latitude <- as.numeric(Manhattan_zip$latitude)
Manhattan_zip$longitude <- as.numeric(Manhattan_zip$longitude)
Ne_sum <- aggregate(SALE.PRICE.~NEIGHBORHOOD.,data = Manhattan_zip,sum)

ASDF <- inner_join(Ne_sum, Manhattan_zip,by = "NEIGHBORHOOD.")
ASDF <- distinct(ASDF,NEIGHBORHOOD.,.keep_all = TRUE)

# m <- leaflet(data =ASDF ) %>%
#   addTiles() %>%  # Add default OpenStreetMap map tiles
#   addMarkers( ~longitude,~latitude
# , popup = ~as.character(SALE.PRICE..x), label = ~as.character(NEIGHBORHOOD.))
# m  # Print the map
```

```
img_path1 <- "Capture.PNG"
img1 <- readPNG(img_path1, native = TRUE, info = TRUE)
# Small fig.width
include_graphics(img_path1)
```



Benford Law Analysis

```
####### Benford analysis
bfd <- benford(Manhattan$SALE.PRICE.)
plot(bfd)
```



The original data is in dark color and the expected frequency are the red dash line. From the plot, we found that the first digits indicate the most data have a tendency to follows Benford's distribution. The digits distribution of second order test shows that there is a clear discrepancy around 50.

Then, I print the main results of the analysis:

```
bfd
```

```
##
## Benford object:
##
## Data: Manhattan$SALE.PRICE.
## Number of observations used = 12779
## Number of obs. for second order = 4038
## First digits analysed = 2
##
## Mantissa:
##
##     Statistic  Value
##          Mean  0.542
##           Var  0.095
##   Ex.Kurtosis -1.331
##      Skewness -0.222
##
##
## The 5 largest deviations:
##
##    digits absolute.diff
```

```
## 1     99         97.22
## 2     75         87.49
## 3     24         81.56
## 4     27         80.83
## 5     26         80.45
##
## Stats:
##
##  Pearson's Chi-squared test
##
## data:  Manhattan$SALE.PRICE.
## X-squared = 1205.6, df = 89, p-value < 2.2e-16
##
##
##  Mantissa Arc Test
##
## data:  Manhattan$SALE.PRICE.
## L2 = 0.029055, df = 2, p-value < 2.2e-16
##
## Mean Absolute Deviation: 0.00256272
## Distortion Factor: 12.00989
##
## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!
```

The numbers of Mantissa from summary are Mean as 0.542, Var as 0.095, Ex.Kurtosis are -1.331, and Skewness are -0.222, which are close to expect value. Degree of freedom equals 89 and p-value is small enough that we are supposed to reject the benford's law. However, thinking of reality, price start with two digits of 99 or 89 are acceptable.

```r
suspects <- getSuspects(bfd, Manhattan)
suspects
```

```
##               NEIGHBORHOOD.                   ADDRESS.   zip
##   1:          ALPHABET CITY  544 EAST 11TH STREET, 3A 10009
##   2:          ALPHABET CITY     399 EAST 8TH   STREET 10009
##   3:          ALPHABET CITY              143 AVENUE B 10009
##   4:          ALPHABET CITY      525 EAST 11TH STREET 10009
##   5:          ALPHABET CITY      643 EAST 11TH STREET 10009
##  ---
## 310: WASHINGTON HEIGHTS UPPER       69 PINEHURST AVENUE 10033
## 311: WASHINGTON HEIGHTS UPPER     804 WEST 180TH ST, 41 10033
## 312: WASHINGTON HEIGHTS UPPER 860 WEST 181ST STREET, 66 10033
## 313: WASHINGTON HEIGHTS UPPER 860 WEST 181ST STREET, 56 10033
## 314: WASHINGTON HEIGHTS UPPER 360 CABRINI BOULEVARD, 8D 10040
##      RESIDENTIAL.UNITS. COMMERCIAL.UNITS. LAND.SQUARE.FEET.
##   1:                  0                 0                NA
##   2:                  1                 0                NA
##   3:                  1                 0                NA
##   4:                  1                 0                NA
##   5:                  1                 0                NA
##  ---
## 310:                 30                 0              7625
## 311:                  0                 0                NA
## 312:                  0                 0                NA
## 313:                  0                 0                NA
```

```
## 314:                  0                 0                NA
##      GROSS.SQUARE.FEET. YEAR.BUILT. BUILDING.CLASS.AT.TIME.OF.SALE.
##   1:                 NA  1928-12-15                              C6
##   2:                 NA  2014-12-15                              R4
##   3:                 NA  1928-12-15                              R4
##   4:                 NA  1965-12-15                              R4
##   5:                 NA  2006-12-15                              R1
##  ---
## 310:              26730  1924-12-15                              C1
## 311:                 NA  1910-12-15                              D4
## 312:                 NA  1923-12-15                              D4
## 313:                 NA  1923-12-15                              D4
## 314:                 NA  1942-12-15                              D4
##      SALE.PRICE. SALE.DATE. Month
##   1:      750000 2017-01-20   Jan
##   2:      753421 2017-02-01   Feb
##   3:      999999 2017-09-14   Sep
##   4:      750000 2017-03-02   Mar
##   5:      995000 2017-07-18   Jul
##  ---
## 310:     7500000 2017-03-30   Mar
## 311:      754000 2017-12-20   Dec
## 312:      755000 2017-06-07   Jun
## 313:      758000 2017-07-13   Jul
## 314:      755000 2017-10-31   Oct
```

```r
manhattan_price <- getBfd(benford(Manhattan$SALE.PRICE.))
kable(manhattan_price)
```

digits	data.dist	data.second.order.dist	benford.dist	data.second.order.dist.freq	data.dist.freq	benford.dist.freq
10	0.0380311	0.1102031	0.0413927	445	486	528.95712
11	0.0412395	0.0235265	0.0377886	95	527	482.90002
12	0.0395180	0.0329371	0.0347621	133	505	444.22496
13	0.0331012	0.0225359	0.0321847	91	423	411.28807
14	0.0260584	0.0210500	0.0299632	85	333	382.90003
15	0.0261366	0.0304606	0.0280287	123	334	358.17906
16	0.0252758	0.0173353	0.0263289	70	323	336.45751
17	0.0243368	0.0168400	0.0248236	68	311	317.22058
18	0.0216762	0.0205547	0.0234811	83	277	300.06492
19	0.0164332	0.0163447	0.0222764	66	210	284.67005
20	0.0153377	0.0411095	0.0211893	166	196	270.77805
21	0.0146334	0.0141159	0.0202034	57	187	258.17907
22	0.0130683	0.0163447	0.0193052	66	167	246.70058
23	0.0135378	0.0138683	0.0184834	56	173	236.19944
24	0.0113467	0.0123824	0.0177288	50	145	226.55591
25	0.0122858	0.0421000	0.0170333	170	157	217.66904
26	0.0100947	0.0101535	0.0163904	41	129	209.45313
27	0.0094687	0.0121347	0.0157943	49	121	201.83494
28	0.0096252	0.0096582	0.0152400	39	123	194.75153
29	0.0107990	0.0079247	0.0147233	32	138	188.14850
30	0.0096252	0.0237741	0.0142404	96	123	181.97857
31	0.0081384	0.0108965	0.0137883	44	104	176.20049
32	0.0102512	0.0079247	0.0133640	32	131	170.77806
33	0.0099382	0.0104012	0.0129650	42	127	165.67944

digits	data.dist	data.second.order.dist	benford.dist	data.second.order.dist.freq	data.dist.freq	benford.dist.freq
34	0.0094687	0.0076771	0.0125891	31	121	160.87646
35	0.0092339	0.0104012	0.0122345	42	118	156.34412
36	0.0094687	0.0084200	0.0118992	34	121	152.06017
37	0.0094687	0.0074294	0.0115819	30	121	148.00475
38	0.0100164	0.0099059	0.0112810	40	128	144.16003
39	0.0090774	0.0079247	0.0109954	32	116	140.51002
40	0.0107207	0.0173353	0.0107239	70	137	137.04028
41	0.0078253	0.0047053	0.0104654	19	100	133.73778
42	0.0117380	0.0069341	0.0102192	28	150	130.59071
43	0.0100947	0.0064388	0.0099842	26	129	127.58836
44	0.0076688	0.0047053	0.0097598	19	98	124.72096
45	0.0111120	0.0081724	0.0095453	33	142	121.97962
46	0.0080601	0.0076771	0.0093400	31	103	119.35620
47	0.0092339	0.0052006	0.0091434	21	118	116.84325
48	0.0078253	0.0069341	0.0089548	28	100	114.43393
49	0.0095469	0.0054482	0.0087739	22	122	112.12198
50	0.0101729	0.1106984	0.0086002	447	130	109.90159
51	0.0069646	0.0052006	0.0084332	21	89	107.76745
52	0.0097034	0.0052006	0.0082725	21	124	105.71461
53	0.0070428	0.0039624	0.0081179	16	90	103.73852
54	0.0085296	0.0032194	0.0079689	13	109	101.83495
55	0.0113467	0.0059435	0.0078253	24	145	99.99999
56	0.0089209	0.0069341	0.0076868	28	114	98.22998
57	0.0079818	0.0064388	0.0075531	26	102	96.52155
58	0.0088426	0.0037147	0.0074240	15	113	94.87153
59	0.0100947	0.0044577	0.0072992	18	129	93.27697
60	0.0093904	0.0069341	0.0071786	28	120	91.73513
61	0.0085296	0.0047053	0.0070619	19	109	90.24344
62	0.0105642	0.0069341	0.0069489	28	135	88.79948
63	0.0094687	0.0044577	0.0068394	18	121	87.40101
64	0.0071211	0.0032194	0.0067334	13	91	86.04590
65	0.0114250	0.0044577	0.0066306	18	146	84.73217
66	0.0093122	0.0034671	0.0065309	14	119	83.45795
67	0.0089991	0.0056959	0.0064341	23	115	82.22149
68	0.0077471	0.0056959	0.0063402	23	99	81.02114
69	0.0079036	0.0049529	0.0062489	20	101	79.85532
70	0.0100947	0.0076771	0.0061603	31	129	78.72258
71	0.0087644	0.0042100	0.0060741	17	112	77.62153
72	0.0086079	0.0047053	0.0059904	19	110	76.55086
73	0.0066515	0.0042100	0.0059089	17	85	75.50932
74	0.0074341	0.0044577	0.0058295	18	95	74.49574
75	0.0125988	0.0059435	0.0057523	24	161	73.50901
76	0.0082166	0.0052006	0.0056771	21	105	72.54808
77	0.0102512	0.0034671	0.0056039	14	131	71.61195
78	0.0070428	0.0034671	0.0055325	14	90	70.69967
79	0.0067298	0.0029718	0.0054629	12	86	69.81034
80	0.0091556	0.0091630	0.0053950	37	117	68.94311
81	0.0059473	0.0047053	0.0053288	19	76	68.09716
82	0.0072776	0.0044577	0.0052642	18	93	67.27172
83	0.0064950	0.0056959	0.0052012	23	83	66.46605
84	0.0068863	0.0044577	0.0051396	18	88	65.67946
85	0.0089991	0.0042100	0.0050795	17	115	64.91126

digits	data.dist	data.second.order.dist	benford.dist	data.second.order.dist.freq	data.dist.freq	benford.dist.freq
86	0.0059473	0.0034671	0.0050208	14	76	64.16082
87	0.0080601	0.0034671	0.0049634	14	103	63.42754
88	0.0053995	0.0022288	0.0049073	9	69	62.71083
89	0.0063385	0.0019812	0.0048525	8	81	62.01013
90	0.0076688	0.0066865	0.0047989	27	98	61.32492
91	0.0048517	0.0029718	0.0047464	12	62	60.65469
92	0.0054777	0.0017335	0.0046951	7	70	59.99895
93	0.0043822	0.0029718	0.0046449	12	56	59.35724
94	0.0039909	0.0012382	0.0045958	5	51	58.72911
95	0.0074341	0.0039624	0.0045476	16	95	58.11413
96	0.0042257	0.0037147	0.0045005	15	54	57.51191
97	0.0056342	0.0029718	0.0044543	12	72	56.92203
98	0.0068863	0.0042100	0.0044091	17	88	56.34413
99	0.0119728	0.0032194	0.0043648	13	153	55.77785

```
kable(head(suspectsTable(benford(Manhattan$SALE.PRICE.)),10))
```

digits	absolute.diff
99	97.22215
75	87.49099
24	81.55591
27	80.83494
26	80.45313
22	79.70058
20	74.77805
19	74.67005
31	72.20049
28	71.75153
# Conclus	ion

By looking at the five output result for Benford's law, it looks like that they are following Benford distribution. Although there are some deviation for some digits, but it may be due to some marketing strategy. Therefore, I think this dataset for sale price are good to trust.