# Exploratory Data Analysis

*Lea Collin*

*4/9/2019*

```
all.data <- fread("Data/NYCRealEstateFullData.csv")
```

```
avg.sale.price.by <- function(data, by.column.names){
  mean.price <- data[, .(`Avg. Price` = mean(get(sale.price.name), na.rm=TRUE)), by = by.column.names]
  return (mean.price)
}
```

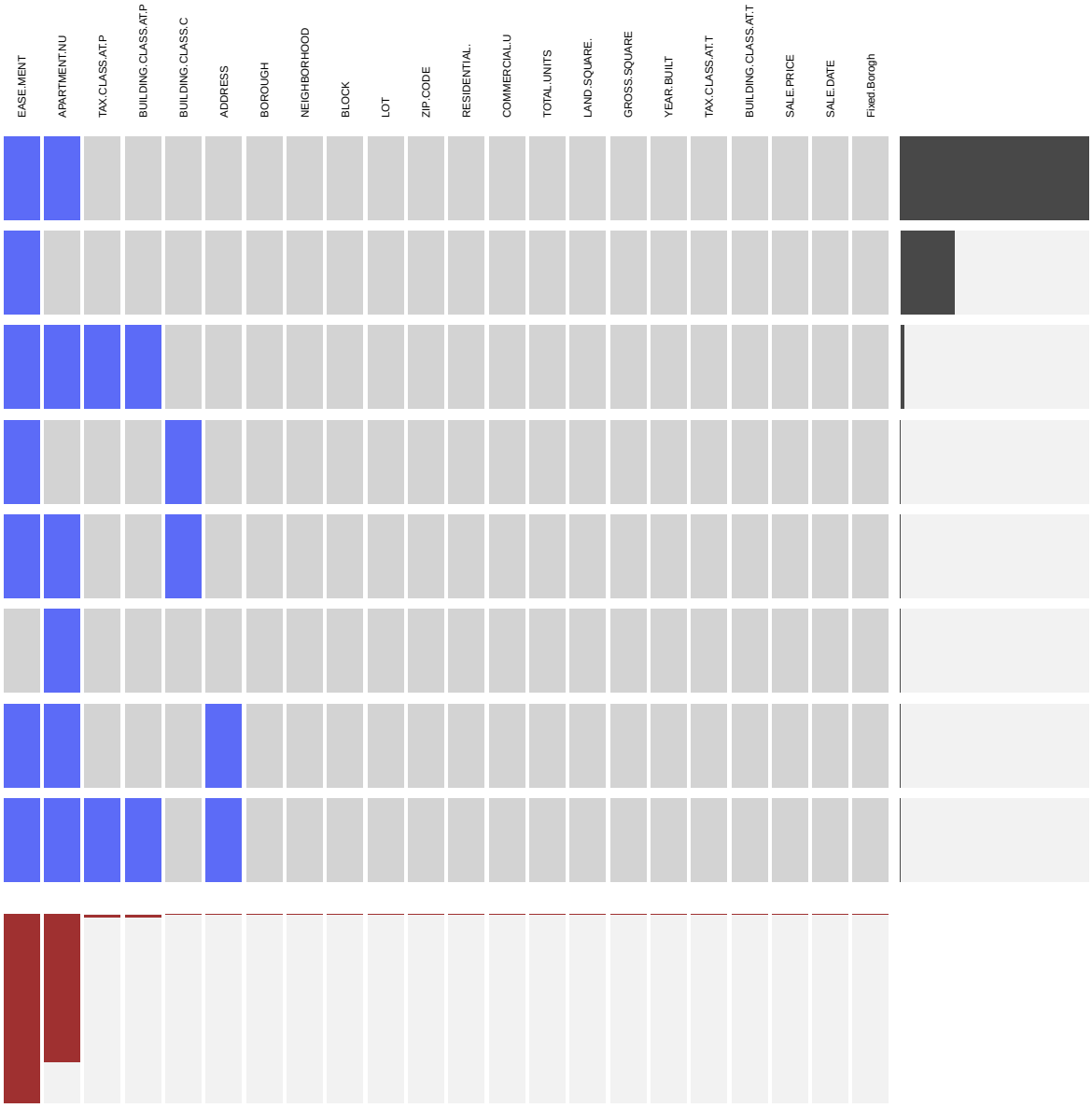
```
old.borough.name <- "BOROUGH"
borough.name <- "Fixed Borough"
neighborhood.name <- "NEIGHBORHOOD"
building.class.name <- "BUILDING CLASS CATEGORY"
tax.class.name <- "TAX CLASS AT PRESENT"
block.name <- "BLOCK"
lot.name <- "LOT"
easement.name <- "EASE-MENT"
building.class.present.name <- "BUILDING CLASS AT PRESENT"
address.name <- "ADDRESS"
apartment.number.name <- "APARTMENT NUMBER"
zip.name <- "ZIP CODE"
residential.name <- "RESIDENTIAL UNITS"
commercial.name <- "COMMERCIAL UNITS"
total.units.name <- "TOTAL UNITS"
land.square.feet.name <- "LAND SQUARE FEET"
gross.square.feet.name <- "GROSS SQUARE FEET"
year.built.name <- "YEAR BUILT"
tax.class.sale.name <- "TAX CLASS AT TIME OF SALE"
building.class.sale.name <- "BUILDING CLASS AT TIME OF SALE"
sale.price.name <- "SALE PRICE"
sale.date.name <- "SALE DATE"
sale.year.name <- "Sale Year"
log.price.name <- "Log Price"
```

```
colSums(is.na(all.data))
```

```
##                         BOROUGH                    NEIGHBORHOOD
##                               0                               0
##         BUILDING CLASS CATEGORY            TAX CLASS AT PRESENT
##                            1481                           20640
##                           BLOCK                             LOT
##                               0                               0
##                       EASE-MENT       BUILDING CLASS AT PRESENT
##                         1432447                           20640
##                         ADDRESS                APARTMENT NUMBER
##                               7                         1119671
##                        ZIP CODE              RESIDENTIAL UNITS
##                               0                               0
##                 COMMERCIAL UNITS                    TOTAL UNITS
##                               0                               0
```

```
##                LAND SQUARE FEET                GROSS SQUARE FEET
##                               0                                0
##                      YEAR BUILT      TAX CLASS AT TIME OF SALE
##                               0                                0
## BUILDING CLASS AT TIME OF SALE                      SALE PRICE
##                               0                                0
##                       SALE DATE                  Fixed Borough
##                               0                                0
```

```
visna(all.data, sort = "b")
```




We see that the most common columns that have missing values are "EASE.MENT" and "APARTMENT NUMBER". Both of these have missing values in more than half of all rows. It is probably safe to get rid of these columns and ignore them both in our exploratory analysis and our machine learning model as they likely do not hold any valuable information with so many missing values. It may have been interesting to look

at apartment number, to see how the floor of the apartment (ie floor in the building) affects the apartments price. Otherwise the only other missing valued columns are "TAX CLASS AT PRESENT", "BUILDING CLASS AT PRESENT", "BUILDING CLASS CATEGORY", and "ADDRESS". Address has only 7 missing values in over 1.4 million rows and there are many other columns that indicate the location of these buildings. What is interesting is that whenever Tax Class at Present is missing, so is Building Class at Present. These have the next highest number of missing values, but still only have about 20,000 NA which is not much when compared to the number of rows.

```
all.data$`EASE-MENT` <- NULL
all.data$`APARTMENT NUMBER` <- NULL
```

```
all.data[get(sale.price.name) == 0, .N]
```

```
## [1] 431820
```

```
all.data[get(sale.price.name) < 50000, .N]
```

```
## [1] 492640
```

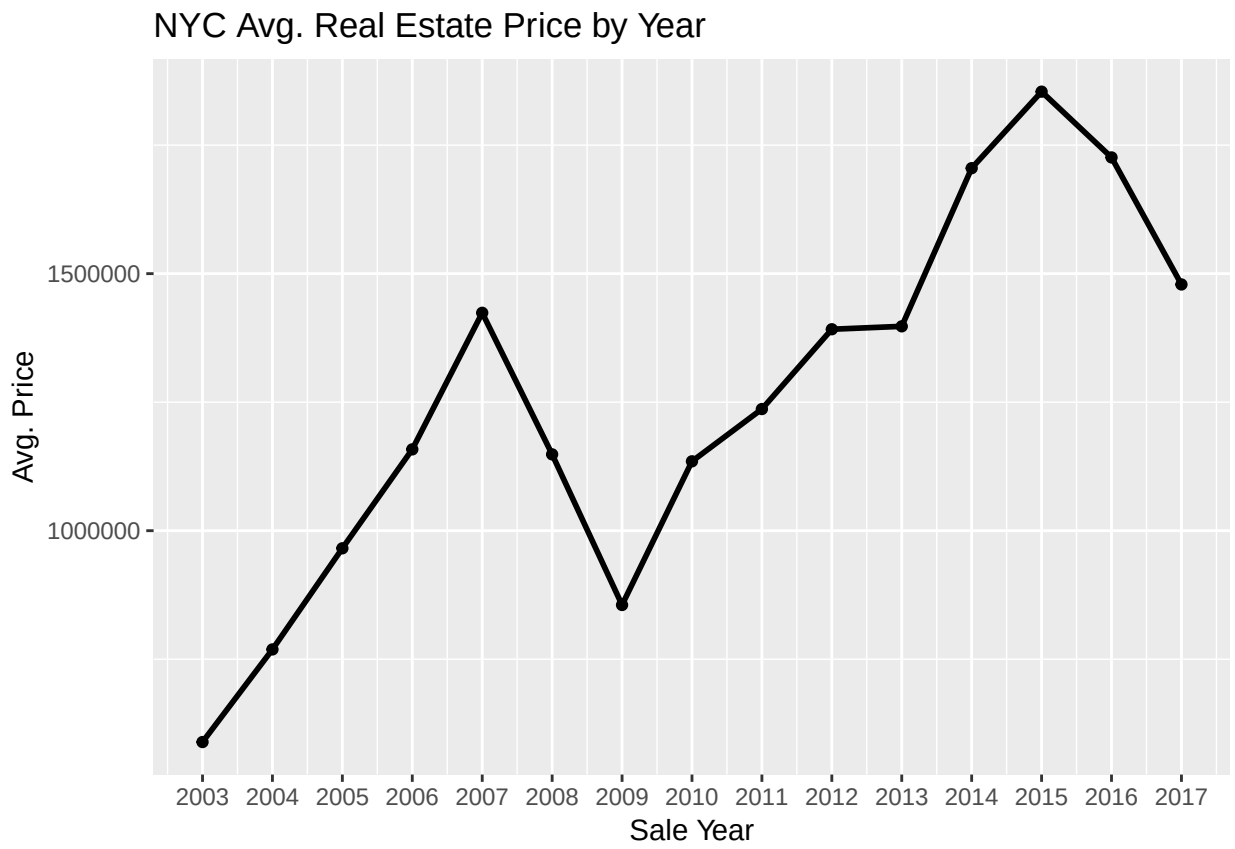
While looking at the data when we were choosing a dataset, we noticed that some sales prices are listed as 0. The documentation for the data claims:

A $0 sale indicates that there was a transfer of ownership without a cash consideration. There can be a number of reasons for a $0 sale including transfers of ownership from parents to children.

It may be interesting to look at what kinds of properties are most common in transfers of ownership with no cash consideration, especially since they make up almost a third of our dataset. Furthermore, there are other "weird" values. Such as values of 1 or 10. We'll take a closer look at the types of buildings that list these as their sale price. For now, we will only look at data that has a price of more than 50,000 and continue with our analysis.

```
# also taking the log for future graphing
all.data <- all.data[, `Log Price` := log(get(sale.price.name))]
dat <- all.data[get(sale.price.name) > 50000]
```

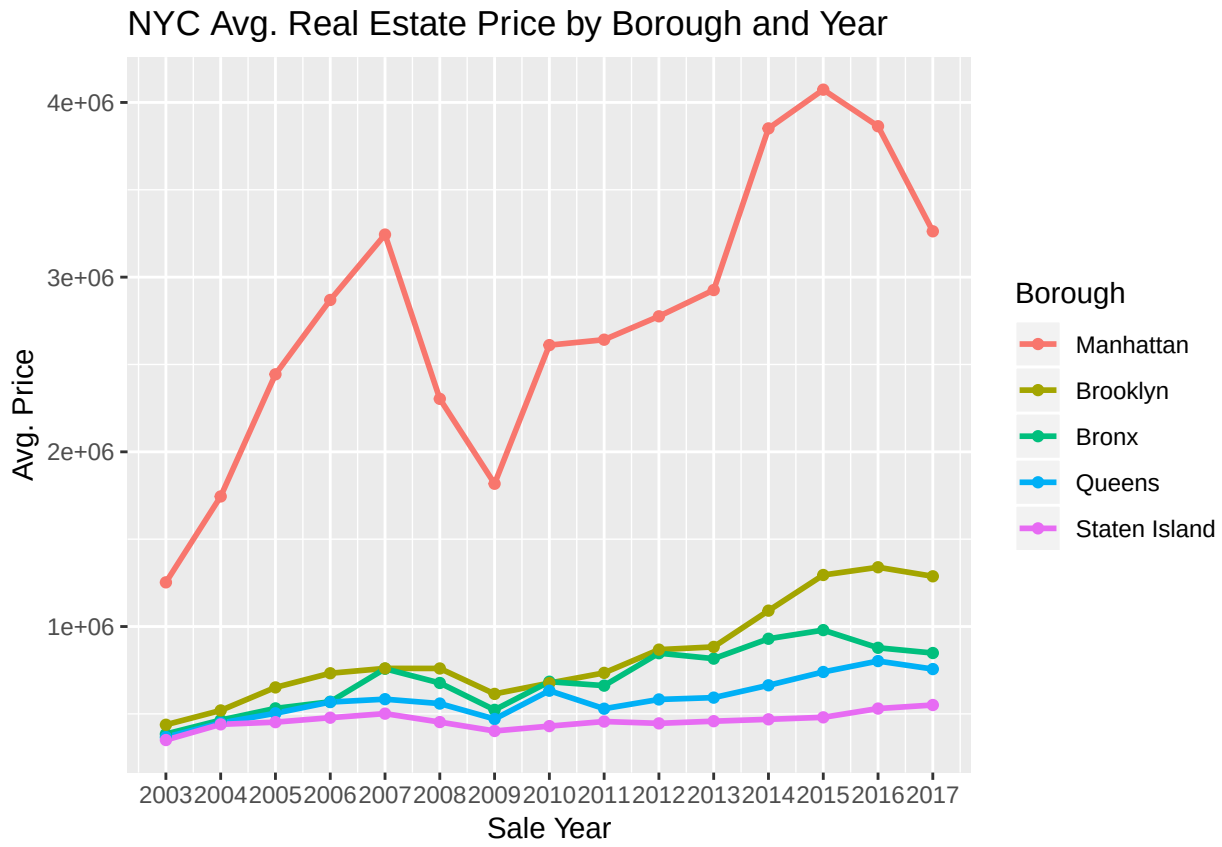
```
dat <- dat[, `Sale Year` := year(get(sale.date.name))]
mean.year.price <- avg.sale.price.by(dat, c(sale.year.name))
setorderv(x = mean.year.price, cols = "Sale Year", order = 1)

year.price.plot <- ggplot(mean.year.price, aes(`Sale Year`, as.integer(`Avg. Price`))) +
                   geom_line(size = 1) + geom_point(aes(`Sale Year`,as.integer(`Avg. Price`))) +
                   xlab("Sale Year") + ylab("Avg. Price") +
                   scale_x_continuous(breaks = scales::pretty_breaks(length(mean.year.price$`Sale Year`
                   ggtitle("NYC Avg. Real Estate Price by Year")

year.price.plot
```




Though this plot isn't terribly exciting, we do see a huge dip in price from 2007 to 2009, most likely due to the 2008 financial/housing crisis. We could also control these prices for inflation since they are taking into account data over a 14 year range. (will try and do this later)

Let's look at the trends over the years, but now also by borough.

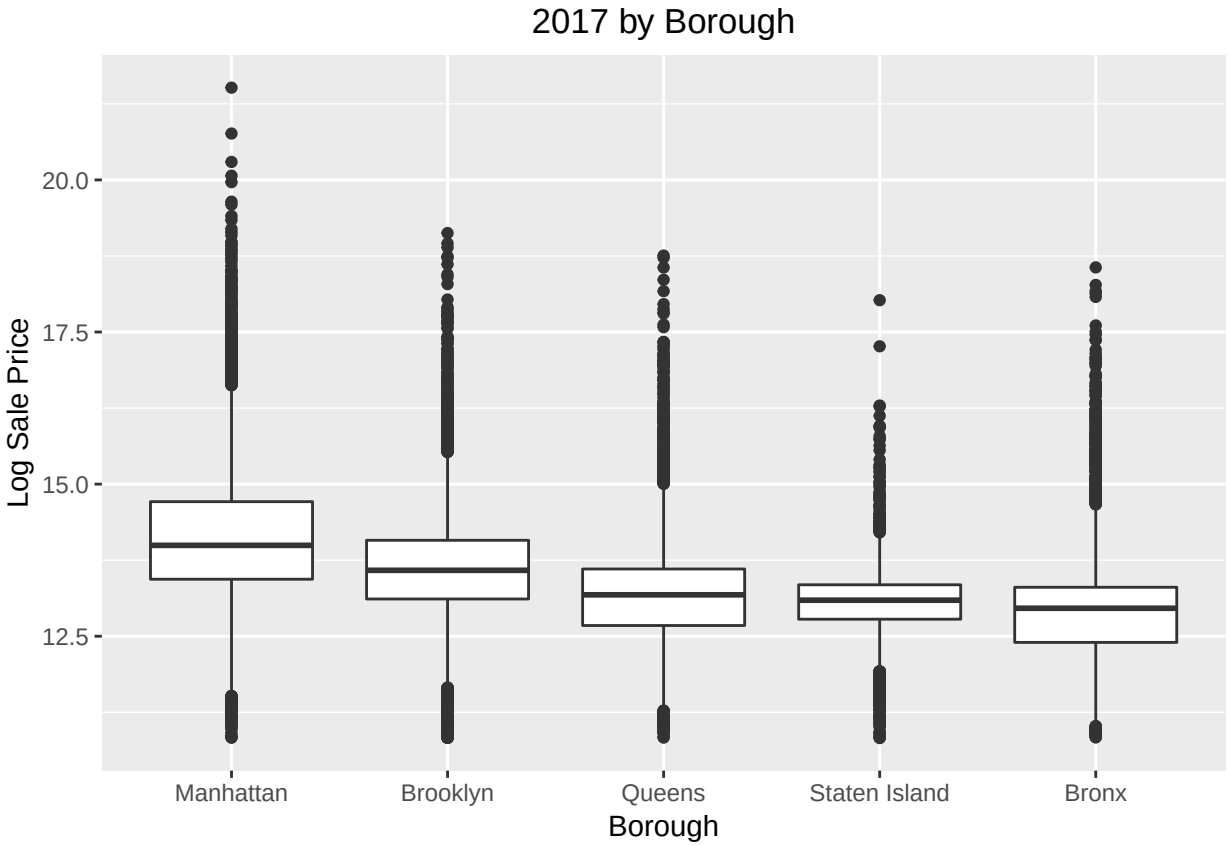
```
mean.year.borough.price <- avg.sale.price.by(dat, c(sale.year.name, borough.name))
setorderv(x = mean.year.borough.price, cols = "Sale Year", order = 1)
mean.year.borough.price <- mean.year.borough.price %>% mutate(`Fixed Borough` = forcats::fct_reorder2(`

year.borough.price.plot <- ggplot(mean.year.borough.price, aes(`Sale Year`, as.integer(`Avg. Price`), co
                  geom_line(size = 1) + geom_point(aes(`Sale Year`,as.integer(`Avg. Price`))) +
                  xlab("Sale Year") + ylab("Avg. Price") + labs(color = "Borough") +
                  scale_x_continuous(breaks = scales::pretty_breaks(length(mean.year.price$`Sale Year`
                  ggtitle("NYC Avg. Real Estate Price by Borough and Year")

year.borough.price.plot
```



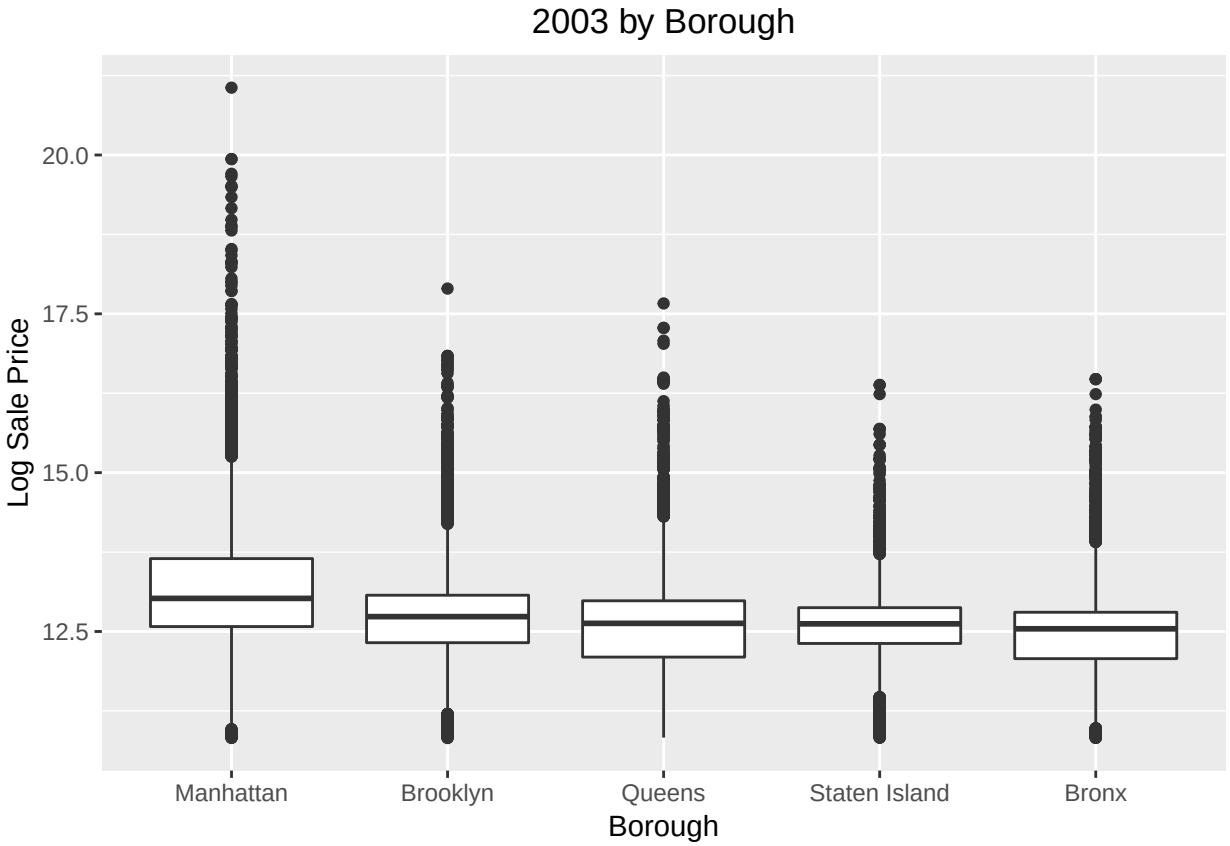

Not too surprisingly, Manhattan sales prices are the highest among the five boroughs. In recent years, we have seen Brooklyn becoming a more sought-after neighborhood which has led to increases in higher real estate prices, as corroborated by this graph. We should note that since this dataset does not only include apartment buildings, that that is probably also the reason that we see such high real estate prices in Manhattan. Manhattan has many more large buildings than, say, Staten Island which is much more residential. What is also interesting to note is that the Manhattan line follows basically the same trend as the overall city trend, likely indicating that most of the data we have is from Manhattan. Finally, for all five boroughs, we see the same decline ine prices from 2007 to 2009 as we did in the overall city trend, though no decline is as noticeable as the one seen in Manhattan.

Now we will look at the distribution of each borough's sale prices over the years. For this static report, we will focus only on 2003 and 2017 and look at the distribution for each borough. In our Shiny application, we will give the user a chance to look at each year, however, this is just too much variable information to show in a static report.

```
box.borough.2017 <- dat[get(sale.year.name) == 2017]
box.borough.2017$`Sale Year` <- as.factor(box.borough.2017$`Sale Year`)

box.borough.2017.plot <- ggplot(box.borough.2017, aes(x = reorder(`Fixed Borough`, -1*`Log Price`, FUN=m
                    geom_boxplot() +
                    xlab("Borough") + ylab("Log Sale Price") + ggtitle("2017 by Borough") +
                    theme(plot.title = element_text(hjust = 0.5))

box.borough.2017.plot
```




```
box.borough.2003 <- dat[get(sale.year.name) == 2003]
box.borough.2003$`Sale Year` <- as.factor(box.borough.2003$`Sale Year`)

box.borough.2003.plot <- ggplot(box.borough.2003, aes(x = reorder(`Fixed Borough`, -1*`Log Price`, FUN=m
                    geom_boxplot() +
                    xlab("Borough") + ylab("Log Sale Price") + ggtitle("2003 by Borough") +
                    theme(plot.title = element_text(hjust = 0.5))

box.borough.2003.plot
```




```
keep_cols = c(zip.name, sale.price.name)
data(zip.regions)
zip.prices <- dat[get(sale.year.name) == 2017, ..keep_cols]
zip.prices <- zip.prices[, mean(get(sale.price.name), na.rm = TRUE), by = zip.name]
colnames(zip.prices) <- c("region", "value")
zip.prices$value <- as.numeric(zip.prices$value)
zip.prices$region <- as.character(zip.prices$region)
zip.prices <- zip.prices[region %in% zip.regions$region,]
zip.prices <- zip.prices[value > 0,]

zip_choropleth(zip.prices,
               zip_zoom = zip.prices$region,
               title       = "2017 Average Sale Price",
               legend      = "Average Sale Price")
```

2017 Average Sale Price