FFx Houses

```
options(width = 200)
library(httr)
library(jsonlite)
library(dplyr)
library(lubridate)
library(ggplot2)
library(corrplot)
library(RColorBrewer)
```

Define a function to call the data.gov API via GET request, that converts the JSON response to dataframe.

```
get_data <- function(url) {
  request <- httr::GET(url = url)
  content <- content(request, as = 'text')
  flat <- fromJSON(content)
  df <- flat$features$properties
  print(nrow(df))
  return(df)
}
```

Use 'get_data' function to import all the housing data.

```
start = Sys.time()
dwelling = get_data('https://opendata.arcgis.com/datasets/53ee1065351c4273ab91ba2e6cfbbc6d_2.geo
json')
```

```
## [1] 333921
```

```
sales = get_data('https://opendata.arcgis.com/datasets/764b1798c0434003a862e2734ba2b705_1.geojso
n')
```

```
## [1] 1249678
```

```
values = get_data('https://opendata.arcgis.com/datasets/63b4c91c3a16425fb5ef9118dbce39ba_2.geojs
on')
```

```
## [1] 363384
```

```
legal = get_data('https://opendata.arcgis.com/datasets/0c3415baff124473832c0e821c0a4ddc_1.geojso
n')
```

```
## [1] 363722
```

```
land = get_data('https://opendata.arcgis.com/datasets/f1f0f31844cf49489134f9fa2b8f16f5_3.geojson')
```

```
## [1] 366281
```

```
parcels = get_data('https://opendata.arcgis.com/datasets/7607cf5046c5495183251d1c9dba0014_1.geojson')
```

```
## [1] 364580
```

```
owners = get_data('https://opendata.arcgis.com/datasets/753448c6e45c434d97b61e77ea752079_2.geojson')
```

```
## [1] 363722
```

```
print(Sys.time()-start)
```

```
## Time difference of 3.181716 mins
```

Find Assessed Value

```
values = values[,c('PARID', 'APRLAND', 'APRBLDG', 'APRTOT')]
```

Find most recent sale price for each parcel.

```
sales = sales[,c('PARID', 'SALEDT', 'PRICE')]
sales$SALEDT <- as.Date(sales$SALEDT, "%Y-%m-%d")
sales$PRICE <- as.numeric(sales$PRICE)
sales <- filter(sales, sales$PRICE != 0)

sales_latest <- sales %>%
  arrange(desc(SALEDT)) %>%
  group_by(PARID) %>%
  summarise('date'=max(SALEDT),'price'=first(PRICE))
```

Concatenate Address Columns for Legal and Owners to get Zipcode and city. Filter out properties that have owners who live elsewhere.

```
legal <- legal[,c('PARID', 'ADRNO', 'ADRADD', 'ADRDIR', 'ADRSTR', 'ADRSUF', 'TAXDIST_DESC')]
legal[is.na(legal)] <-''
legal$address <-  paste(legal$ADRNO, legal$ADRADD, legal$ADRDIR, legal$ADRSTR, legal$ADRSUF, sep
=' ')

owners <- owners[,c('PARID', 'ADRNO', 'ADRADD', 'ADRDIR', 'ADRSTR', 'ADRSUF','CITYNAME', 'STATEC
ODE', 'ZIP1')]
owners[is.na(owners)] <-''
owners$address <-  paste(owners$ADRNO, owners$ADRADD, owners$ADRDIR, owners$ADRSTR, owners$ADRSU
F, sep=' ')

address = dplyr::left_join(legal, owners[,c('PARID','address', 'ZIP1', 'CITYNAME')], by = 'PARI
D')
address <- filter(address, address$address.x == address$address.y)
address <- address[,c('PARID', 'CITYNAME', 'ZIP1', 'address.x','address.y')]
```

The data contains all types of properties (commercial, condo/apartment buildings, etc) so filter for only single family property types

```
single_fam = filter(parcels, parcels$LUC_DESC == 'Single-family, Detached ')
single_fam_dwelling = filter(dwelling, dwelling$PARID %in% single_fam$PARID)
```

Combine All Features

```
combined <- left_join(single_fam_dwelling, values, by = 'PARID') %>%
 left_join(., address, by = 'PARID') %>%
 left_join(., sales_latest, by = 'PARID')
```

Fill missing values for continuous dwelling variables

```
cols <- c('SFLA','RMBED','FIXBATH','FIXHALF','RECROMAREA','BSMTCAR')
for (c in cols){
  combined[c][is.na(combined[c])] <- 0

}
combined <- filter(combined, !is.na(combined$APRBLDG) )
```

Find min and max to identify outliers

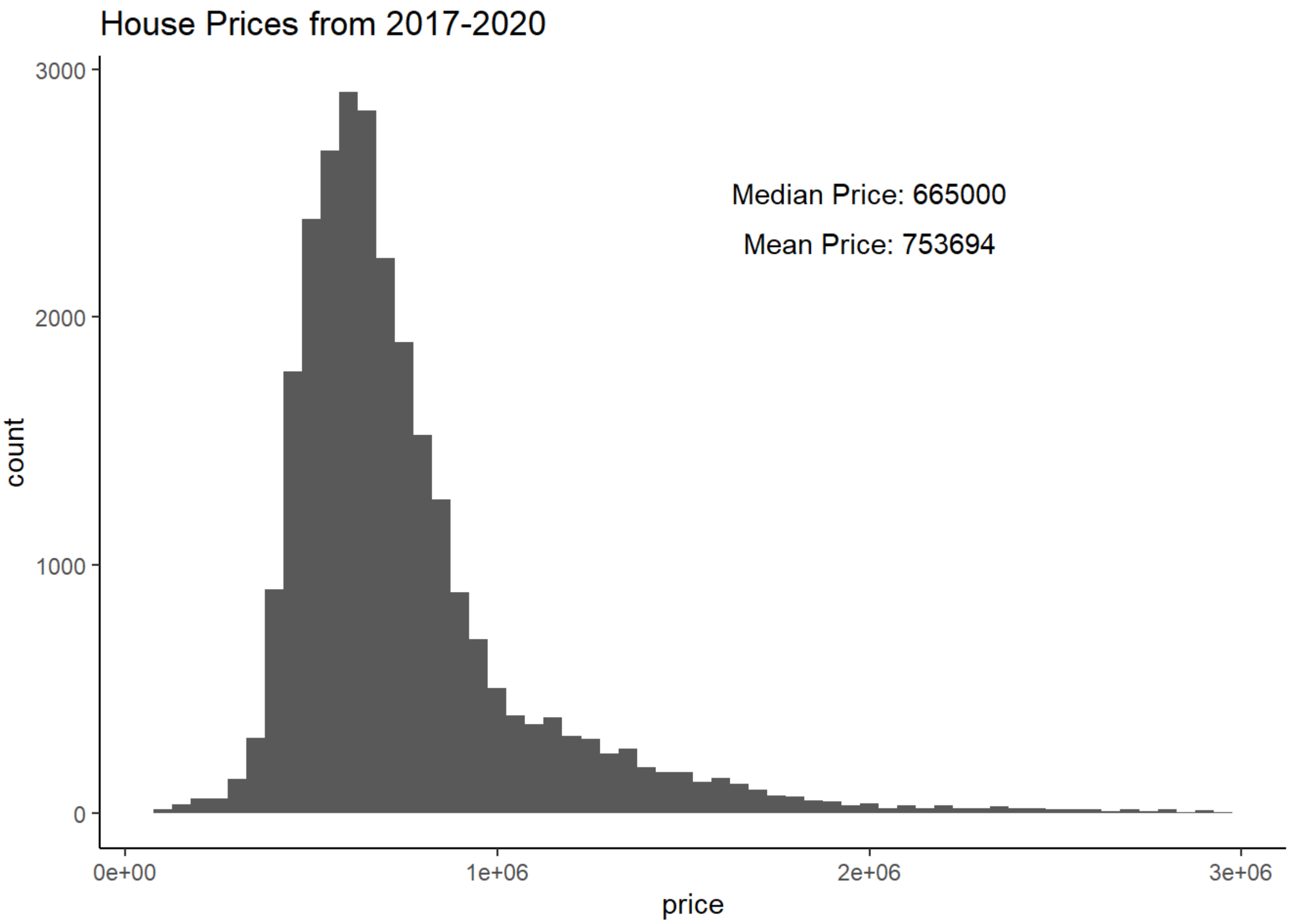
```
combined$YrSold <- year(combined$date)
recent <- filter(combined, combined$YrSold %in% c(2017,2018,2019,2020))
print(paste("Min Price:", min(recent$price), "Max Price:", max(recent$price)))
```

```
## [1] "Min Price: 3 Max Price: 4.3e+07"
```

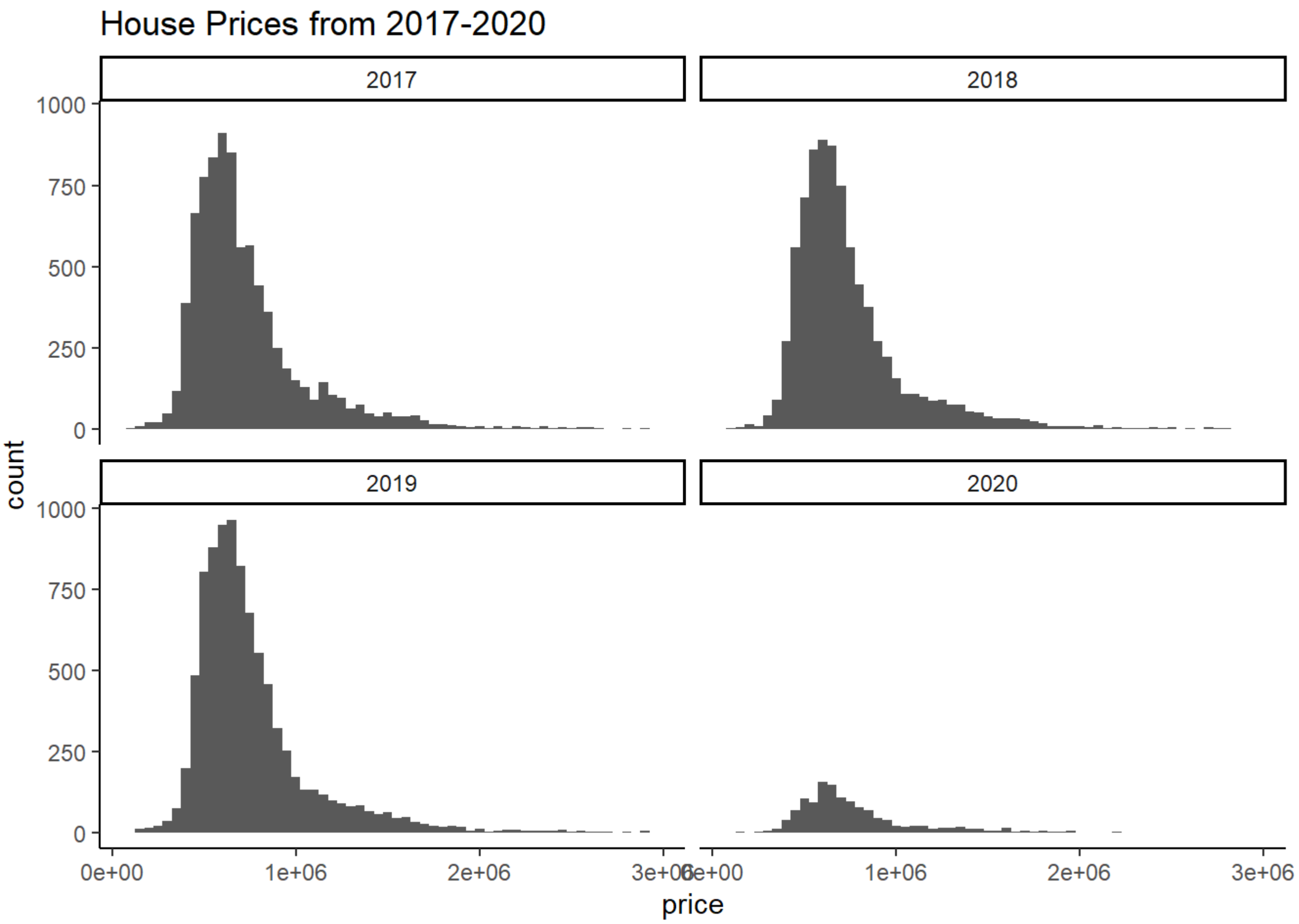
Filter out potential data quality issues (sale price of 3 dollars) and extremely expensive houses (set upper limit to $3M) Calculate ratio of Appraised Total (land + Building) to most recent sale price.

```
recent = recent[(recent$price > 100000 & recent$price < 3000000),]
recent$ratio = recent$APRTOT/recent$price
```

```r
h <- ggplot(data = recent, aes(x=price)) + geom_histogram(binwidth =5e4) + theme_classic() +ggtitle('House Prices from 2017-2020')
h +annotate(geom="text", x=2e6, y=2500, label=paste('Median Price:',median(recent$price)), color="black") +annotate(geom="text", x=2e6, y=2300, label=paste('Mean Price:',as.integer(mean(recent$price))), color="black")
```

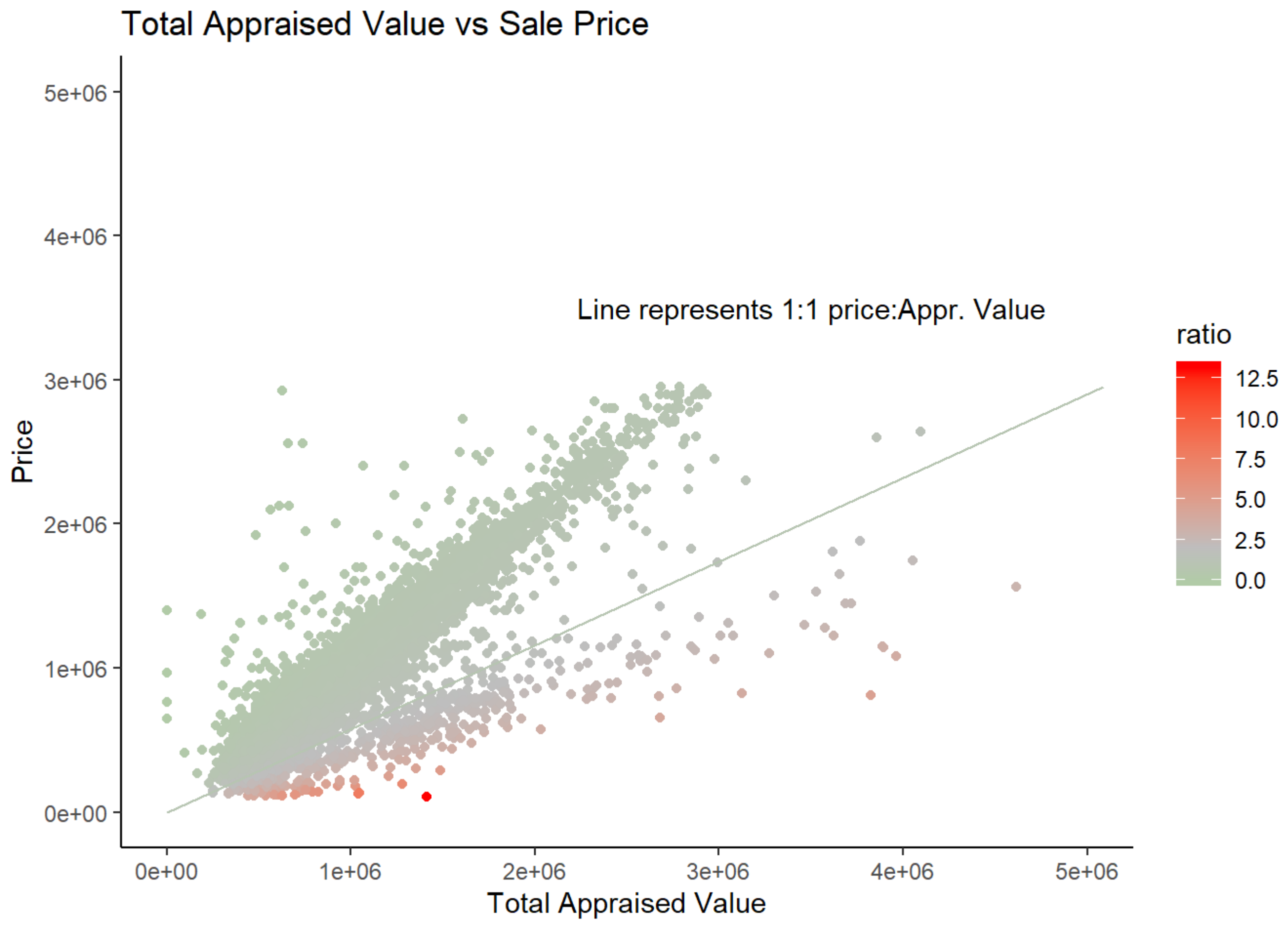


```r
h_facet = h + facet_wrap(~YrSold)
h_facet
```



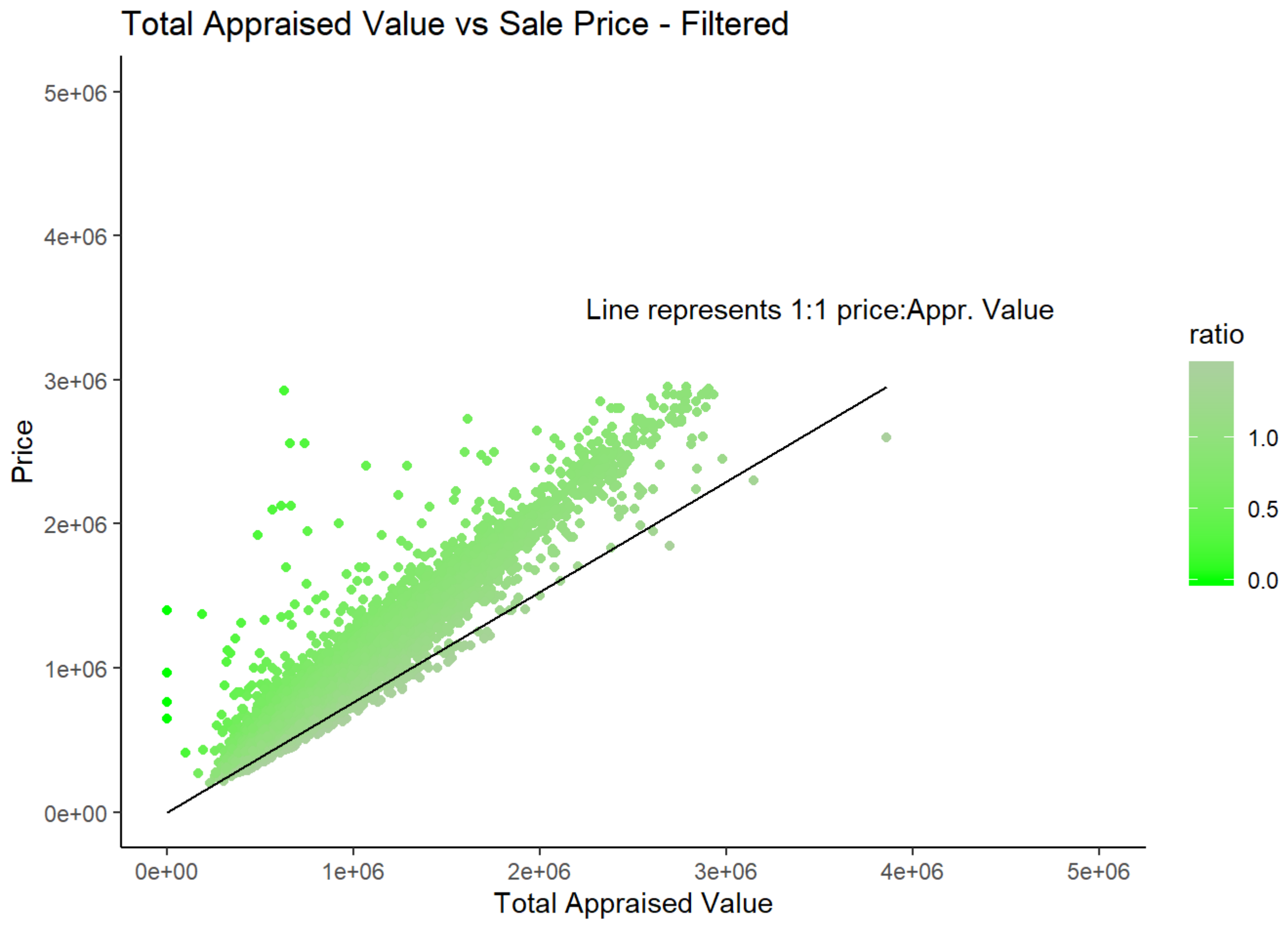
```
apr <- ggplot(data = recent, aes(APRTOT, price, color = ratio)) + geom_point() + scale_y_continu
ous('Price',limits = c(0,5000000)) + scale_x_continuous('Total Appraised Value', limits = c(0,50
00000)) + scale_color_gradient2(midpoint=2, low="green", mid = 'grey', high="red" ) + geom_segme
nt(x = 0, y=0, yend = max(recent$price), xend = max(recent$APRTOT))+theme_classic()
apr + ggtitle('Total Appraised Value vs Sale Price') + annotate(geom = 'text', y = 3.5e6, x =3.5
e6, label = "Line represents 1:1 price:Appr. Value" )
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

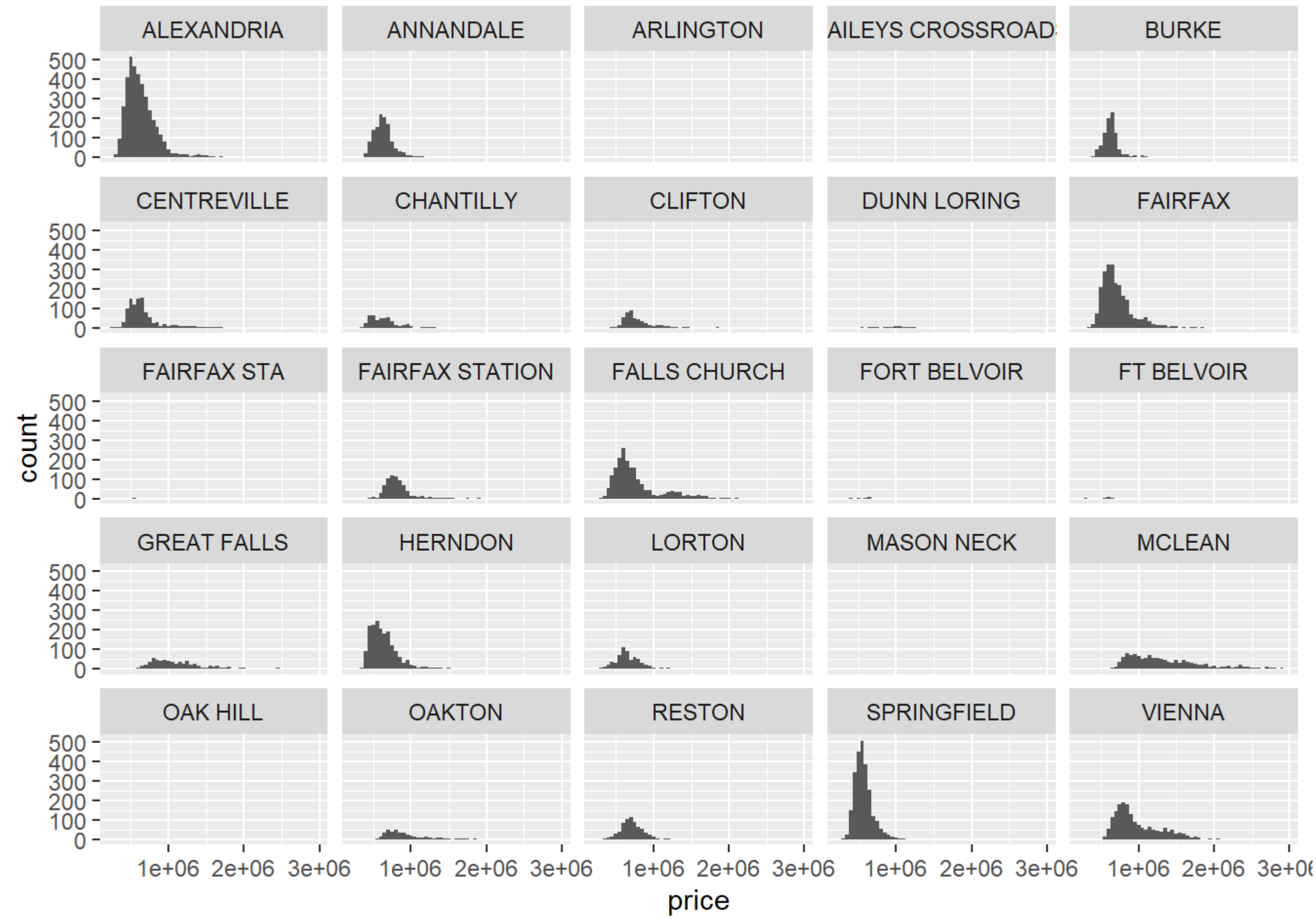


```
houses <- recent[recent$ratio < 1.5,]
```

```
apr <- ggplot(data = houses, aes(APRTOT, price, color = ratio)) + geom_point() + scale_y_continu
ous('Price',limits = c(0,5000000)) + scale_x_continuous('Total Appraised Value', limits = c(0,50
00000)) + scale_color_gradient2(midpoint=2, low="green", mid = 'grey', high="red" ) + geom_segme
nt(x = 0, y=0, yend = max(houses$price), xend = max(houses$APRTOT), color = "black")+theme_class
ic() + annotate(geom = 'text', y = 3.5e6, x =3.5e6, label = "Line represents 1:1 price:Appr. Val
ue" )
apr + ggtitle('Total Appraised Value vs Sale Price - Filtered')
```



```
houses <- filter(houses, (!is.na(houses$CITYNAME) & houses$CITYNAME != ''))

ggplot(houses, aes(x=price)) + geom_histogram(binwidth = 5e4) +facet_wrap(~CITYNAME)
```

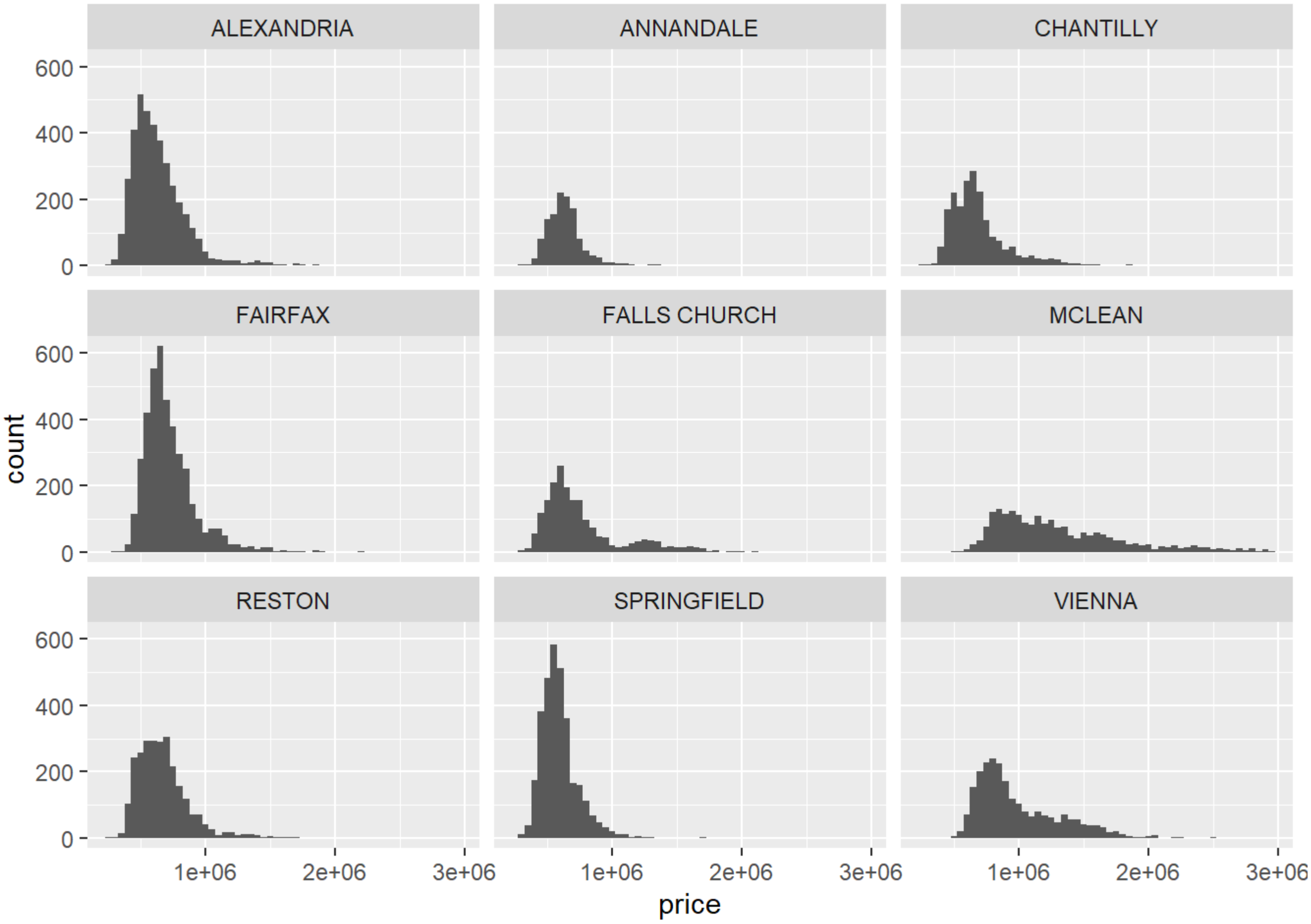


There were some inconsistent city names and cities that had very few observations. I grouped based geography and corrected where the same city had different names in the data.

```
houses <- mutate( houses, CityNew =
                    case_when (CITYNAME %in% c('BURKE','FAIRFAX STA','FAIRFAX STATION') ~ 'FAIRF
AX',
                               CITYNAME %in% c('FORT BELVOIR','FT BELVOIR','LORTON', 'MASON NEC
K') ~ 'SPRINGFIELD',
                               CITYNAME %in% c('OAKTON','DUNN LORING') ~ 'VIENNA',
                               CITYNAME %in% c('OAK HILL', 'CENTREVILLE', 'CLIFTON') ~ 'CHANTILL
Y',
                               CITYNAME %in% c('HERNDON') ~ 'RESTON',
                               CITYNAME %in% c('GREAT FALLS') ~ 'MCLEAN',
                               CITYNAME %in% c('BAILEYS CROSSROADS', 'ARLINGTON') ~ 'ALEXANDRIA'
,
                               TRUE ~ CITYNAME))
```

Now down to 9 cities, each with at least a few hundred rows.

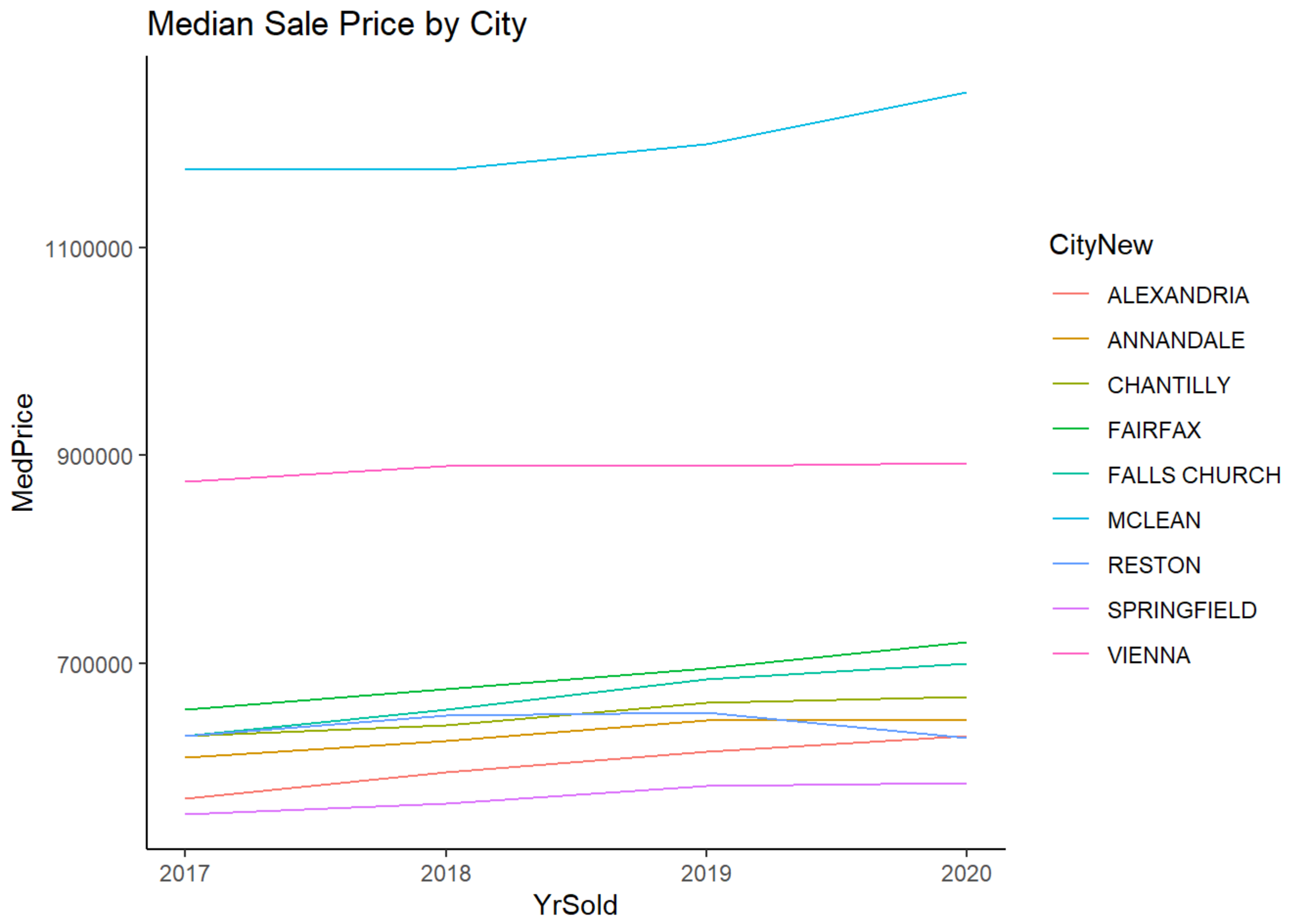
```
ggplot(houses, aes(x=price)) + geom_histogram(binwidth = 5e4) +facet_wrap(~CityNew)
```



Find median home price sale price per year for each city

```
medians <- houses %>%
  group_by(CityNew, YrSold) %>%
  summarise('MedPrice' = median(price))

ggplot(medians, aes(x=YrSold, y = MedPrice, color = CityNew)) + geom_line(aes(linestyle = CityNe
w)) + theme_classic() + ggtitle('Median Sale Price by City')
```

```
## Warning: Ignoring unknown aesthetics: linestyle
```



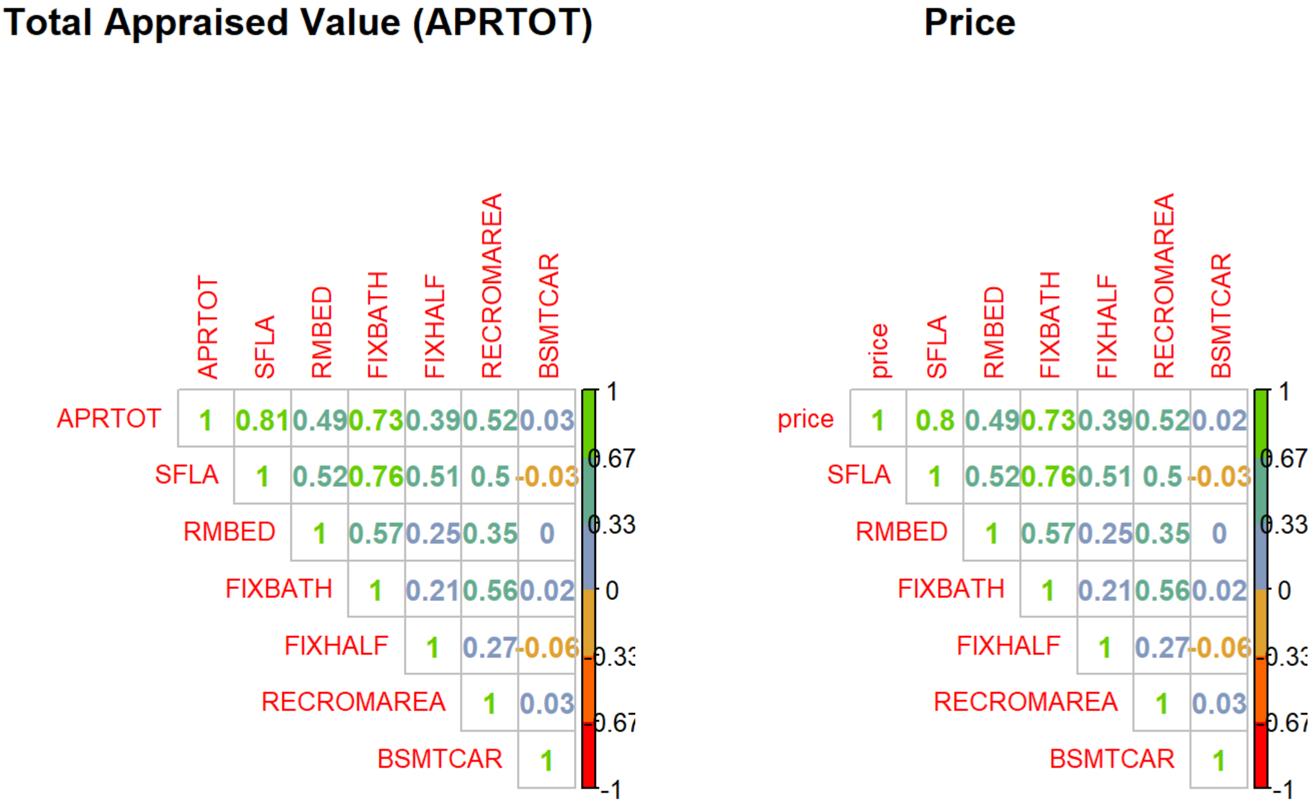
Check One way anova for dwelling variables and City

```
anv <- aov(price ~ SFLA+FIXBATH+RMBED+RECROMAREA+CityNew, data = houses)
summary(anv)
```

```
##                 Df    Sum Sq   Mean Sq  F value Pr(>F)    
## SFLA             1 1.644e+15 1.644e+15 74874.23 <2e-16 ***
## FIXBATH          1 8.473e+13 8.473e+13  3858.19 <2e-16 ***
## RMBED            1 1.880e+12 1.880e+12    85.59 <2e-16 ***
## RECROMAREA       1 1.604e+13 1.604e+13   730.40 <2e-16 ***
## CityNew          8 3.126e+14 3.907e+13  1779.06 <2e-16 ***
## Residuals    23289 5.115e+14 2.196e+10                    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

View Correlations for both Price and Appraised Value. Not Surprising, the correlations are about identical.

```
col1 <- colorRampPalette(c('red', 'orange', 'cornflowerblue', 'chartreuse3'))

corrs <- cor(houses[,c('APRTOT', 'SFLA','RMBED','FIXBATH','FIXHALF','RECROMAREA','BSMTCAR')])
corrs1 <- cor(houses[,c('price', 'SFLA','RMBED','FIXBATH','FIXHALF','RECROMAREA','BSMTCAR')])

par(mfrow =c(1,2))
corrplot(corrs, type = 'upper', method = 'number', tl.cex = .8, number.cex = .9, col = col1(6),
 title = 'Total Appraised Value (APRTOT)',mar=c(0,0,1,0))
corrplot(corrs1, type = 'upper', method = 'number',tl.cex = .8, number.cex = .9,col = col1(6),ti
tle = 'Price',mar=c(0,0,1,0))
```

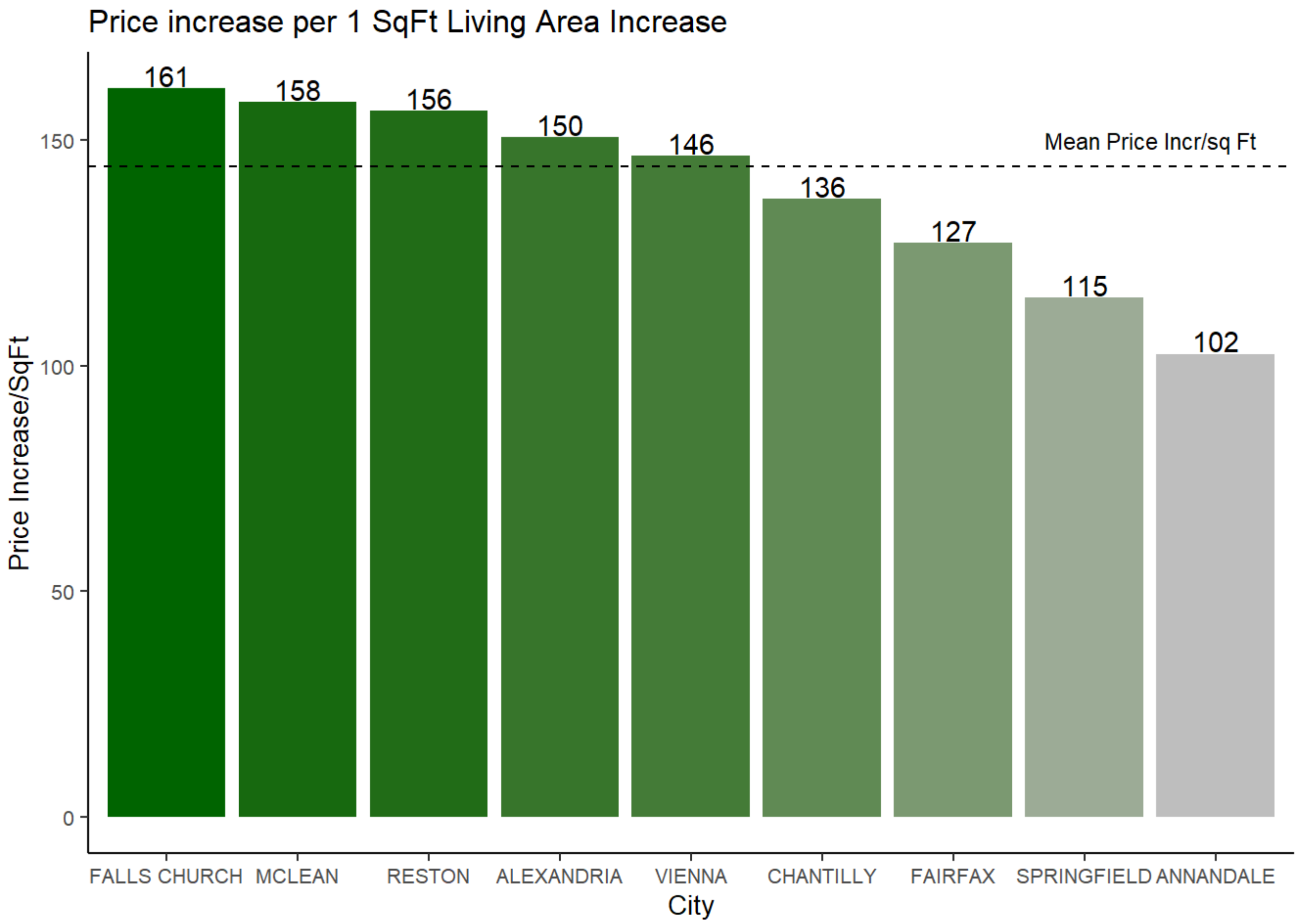


Find how much SFLA influences price for all houses

```
mod_all <- lm(APRTOT ~ SFLA+FIXBATH+RMBED+RECROMAREA+CityNew, data = houses)
coef_all_apr <- summary(mod_all)$coef[2]

mod_all1 <- lm(price ~ SFLA+FIXBATH+RMBED+RECROMAREA+CityNew, data = houses)
coef_all_price <- summary(mod_all1)$coef[2]
```

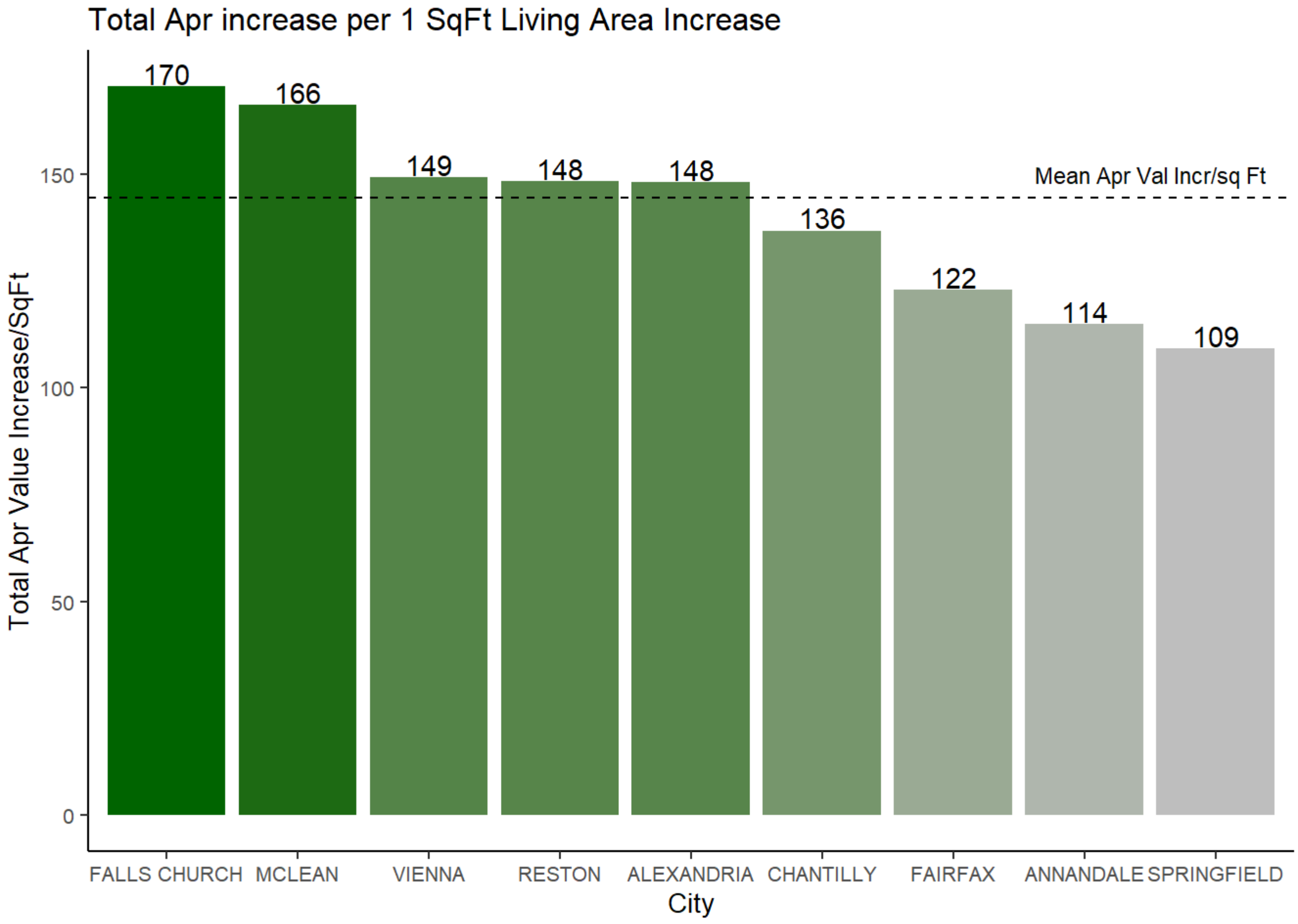
Multiple regressions for each city to see how sale price changes based on house size (measured by Sq Ft Living Area) across each city while holding other significant variables constant

```r
cities <- unique(houses$CityNew)
sqftdf <- data.frame()
i = 1
for (city in cities){
  temp <- houses[houses$CityNew == city,]
  mod <- lm(price ~ SFLA+FIXBATH+RMBED+RECROMAREA, data = temp)
  sqftdf[i,1] <- as.character(city)
  sqftdf[i,2] <- summary(mod)$coef[2]
  i = i+1

}

ggplot(sqftdf, aes(x = reorder(V1,-V2), y = V2,fill = V2) ) + geom_bar(stat = "identity") + xlab
('City') + ylab('Price Increase/SqFt') + ggtitle('Price increase per 1 SqFt Living Area Increas
e') + theme_classic() + theme(legend.position = "none") + geom_text(aes(label = as.integer(V2),
 vjust = -.1)) +  scale_fill_continuous(high = 'darkgreen', low = 'grey')+ geom_abline(intercept
= coef_all_price, slope = 0, linetype = 'dashed') + annotate(geom = 'text', x = 8.5, y = 150, la
bel = "Mean Price Incr/sq Ft", size = 3)+theme(text = element_text(size=10))
```



Multiple regressions for each city to see how Total Appraisal Value changes based on house size (measured by Sq Ft Living Area) across each city while holding other significant variables constant

```
sqftdf1 <- data.frame()
i = 1
for (city in cities){
  temp <- houses[houses$CityNew == city,]
  mod <- lm(APRTOT ~ SFLA+FIXBATH+RMBED+RECROMAREA, data = temp)
  sqftdf1[i,1] <- as.character(city)
  sqftdf1[i,2] <- summary(mod)$coef[2]
  i = i+1

}

ggplot(sqftdf1, aes(x = reorder(V1,-V2), y = V2,fill = V2) ) + geom_bar(stat = "identity") + xla
b('City') + ylab('Total Apr Value Increase/SqFt') + ggtitle('Total Apr increase per 1 SqFt Livin
g Area Increase') + theme_classic() + theme(legend.position = "none") + geom_text(aes(label = a
s.integer(V2), vjust = -.1)) +  scale_fill_continuous(high = 'darkgreen', low = 'grey')+ geom_ab
line(intercept = coef_all_apr, slope = 0, linetype = 'dashed') + annotate(geom = 'text', x = 8.5
, y = 150, label = "Mean Apr Val Incr/sq Ft", size = 3) +theme(text = element_text(size=10))
```


Total Apr increase per 1 SqFt Living Area Increase
Total Apr Value Increase/SqFt
Mean Apr Val Incr/sq Ft
FALLS CHURCH
MCLEAN
VIENNA
RESTON
ALEXANDRIA
CHANTILLY
FAIRFAX
ANNANDALE
SPRINGFIELD
City