House_report

2023-01-28

House report

Data

The data for median house price per square feet is extracted from website. The data for number of home sold and median days on the market is captured from the website

Included Area

The excel file created for data analytics contains towns that has the following property:

Target area This is the area we focus on. There are five areas, namely Scarsdale, Purchase, Rye, Bedford, Chappaqua. Scarsdale is the benchmark, which has a high house price.

(This is the testing dataset. I create a linear model based on other area to predict the potential house price per square feet. So that it can show if the price still has potential to increase)

High median price per square foot In accordance to move.org webpage, cities like San Francisco, Boston and New York City have the mosty expensive house price per unit. Along with these cities, I also choose Fremont, CA, Oakland, CA, San Jose, CA and Honolulu as focused cities. These cities can give a clear indication of the important factors that should be considered, if the house price is high, or is going to be high in the target city.

Other chosen city Seattle and Huston are also two major cities that have a high demand of investment. These cities can also be act as an benchmarks. River Edge is very similar to Chappaqua, so I include it in the data for comparison.

Variables

Table 1: HousePrice House price per square feet is included in the first table of the data. The table contains all the house price per square feet from 4th Feburary 2022 to 20th January 2023. The mean value for house price per square feet for each month is given in the aggragate columns in the last 12 rows, from BB to BM column in excel.

Table 2: #ofhousehold Table 2 consists the number of purchased house unit for every month in the last three years.

Table 3: Day on the market Table 3 consists the median time for the housing unit being purchased for every month in the last three years.

Table 4: all info Table 4 consists all the background information of the housing unit. In order to calculate the change in price over seasons so that the impact of the raise in interest rate can be reviewed, I choose to extract the data from Feb. 18th, May 13th, Aug 12th, Nov 18th and the latest available price, which is Jan 20, 2023. Columns C to K in excel are the calculated house values and the change in price is calculated in column H to K so that we can see the potential impact of rise in interest rate after August.

The rating of each city is also included in the table. This consist the overall rating of each city, as well as the rating for Housing, Education, Cost of living, Unemployment, Commute and Lifestyle. The rating is in a scale of 12. The scale is represented the following: A+: 12 A: 11 A-: 10 B+: 9 B: 8 B-: 7 C+: 6 C: 5 C-: 4 D+: 3 D: 2 D-: 1.

The description of the rating is evaluated from the following categories:

- Housing category: considers home inventory availability, median days on market, total new listing and estimated home sales.
- Cost of living: considers median home price and average cost per square foot.
- Education rating: evaluate the school grade and the percentage of Bachelor degree or greater people in the city.
- Unemployment: considers employment rate only.
- Life style: considers urban life style, including number of restaurant and bars, businesses and net migrations of the city.
- Commute rating: evaluate the median travel time to work, and percentage of people go to work by bike, public transit or on foot.

The table also contains the demographical details of each chosen cities, such as housing unit, median income and populations.

The last few columns contains the average school rating of top five schools in the city and the total number of schools in the city. The schools are classified into Elementary school, Middle school and High school.

Table 5: Population in thousand This is the population table for each target cities.

Table 6: Home for sale This is the table which consists the number of house unit available for every month in the last two years.

Table 7: Sale to list This is the table that store all of the sale price in relative to the list price in percentage. If the stored value is 100, it means that the sale price is exactly the same as the list value. If the value is higher than 100, than the sale price is higher than the list value.

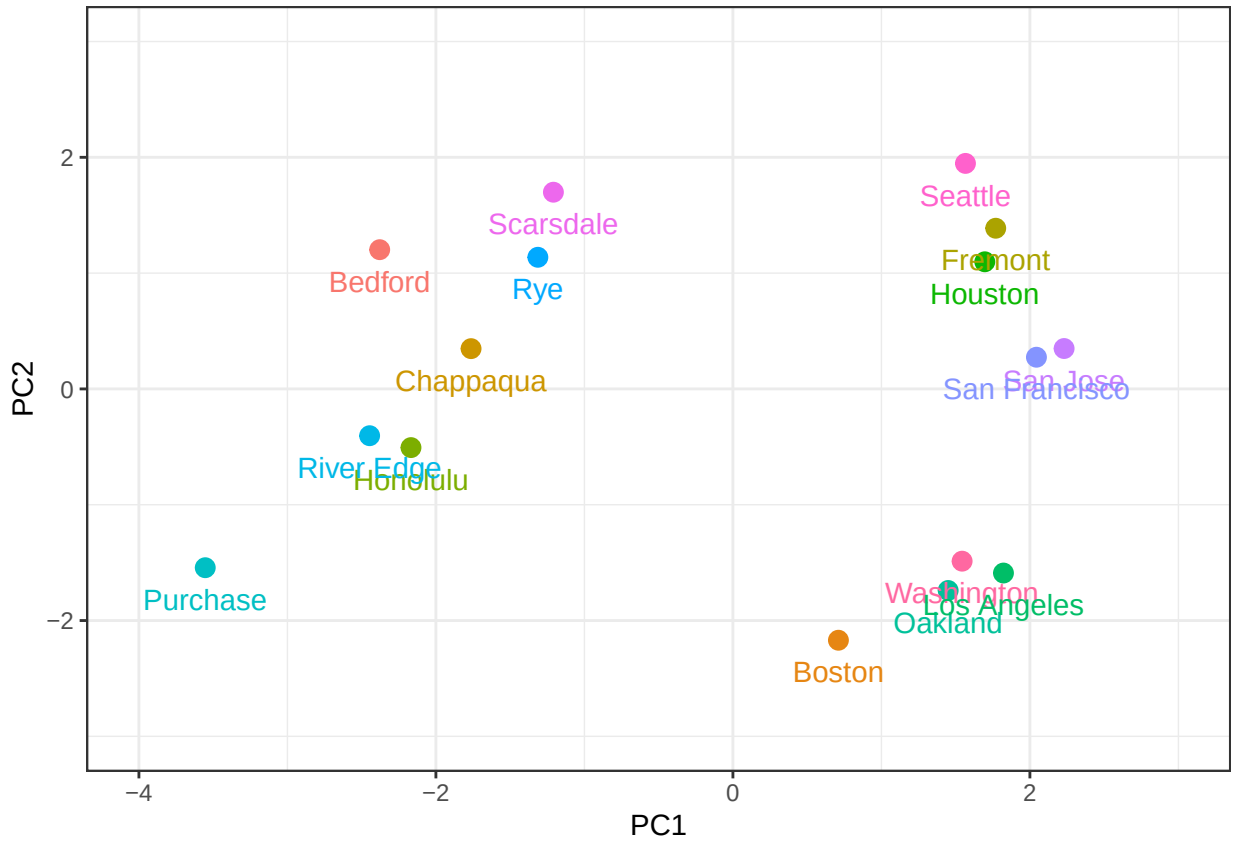
Table 4: House price in 3 yr This is another table for the house price in square feet, but with a scale value of 3 years. All values are the median value of each month.

Table 5: Other potential city It is an incomplete table which may be some potential cities that are worth to invest in.

Results

Clustering results

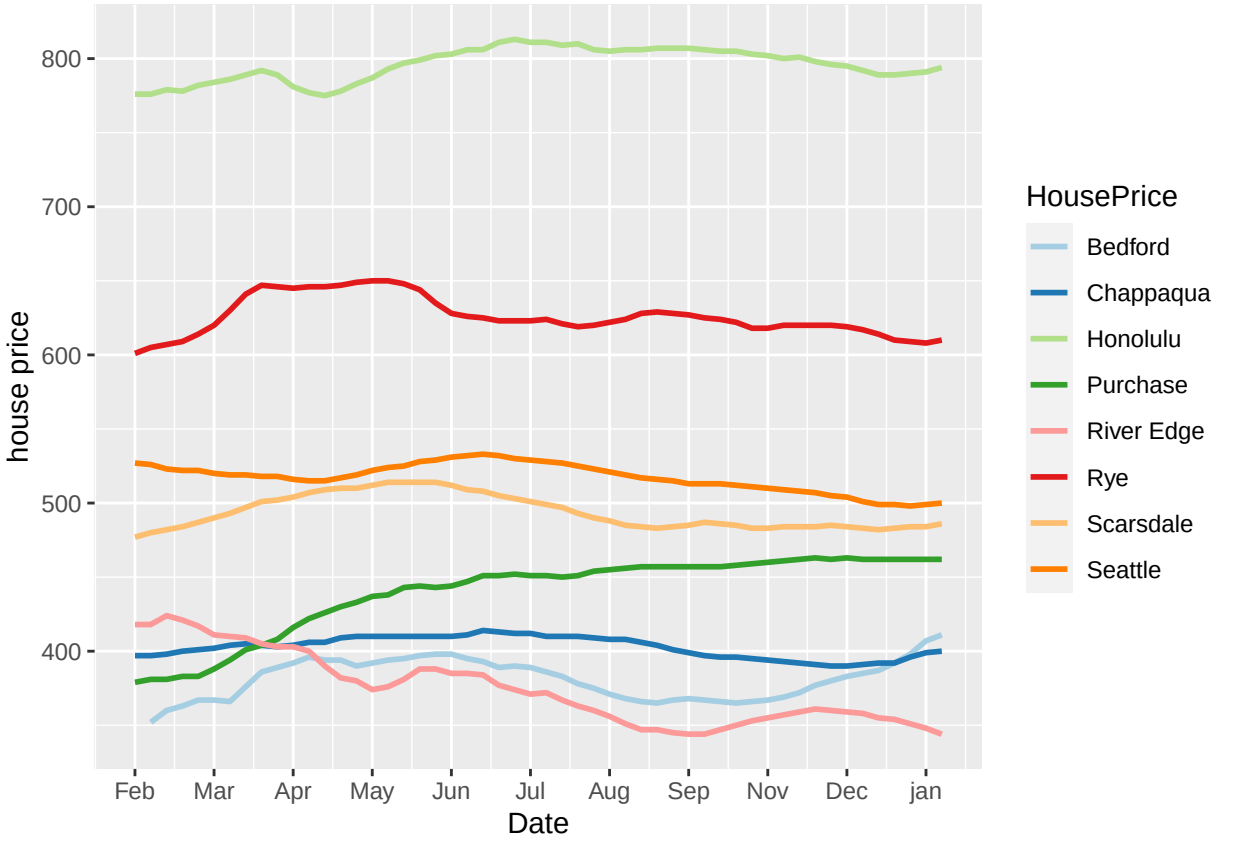
```
selected_info =
  all_info %>%
  janitor::clean_names() %>%
  select(overall_rating, housing, education, cost_of_living, unemployment, commute, lifestyle)
rownames(selected_info) = c("Chappaqua", "Bedford","Rye","Purchase","Scarsdale","Fremont", "Oakland", "S

pca_res <- prcomp(selected_info, scale. = TRUE)




df <- as.data.frame(pca_res$x)
df$city <- rownames(df)

ggplot(df, aes(PC1, PC2, color = city)) +
  geom_point(size = 3) +
  geom_text(aes(label = city) , vjust = 2) +
  lims(x = c(-4, 3), y = c(-3, 3)) +
  theme_bw() +
  theme(legend.position = "none")
```



Honolulu has a rating that is close to the four target areas, so I include the city in the price time series plot. Seattle is also included for a reference as it has a high house price per square feet and it is a well developed city.

House price time series

```
time = colnames(house_price)[3:54]
df1 = t(house_price[,3:54])
data_df1 = melt(df1)
data_df1_named =
  data_df1 %>%
  mutate(Var2 = ifelse(Var2==1,"Chappaqua",ifelse(Var2==2, "Bedford", ifelse(Var2==3, "Rye", ifelse(Var
         ind = rep(1:length(time),8),
         HousePrice = Var2)
ggplot(data_df1_named, aes(x=ind, y=value, color = HousePrice))+
  geom_line(size=1)+
  labs(
    y="house price",
    x="Date"
  )+
  scale_x_continuous(
    breaks = seq(1,length(time),5),
    labels = c("Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Nov","Dec","jan"),
    limits = c(1,length(time)))+
  scale_color_brewer(palette="Paired")
```



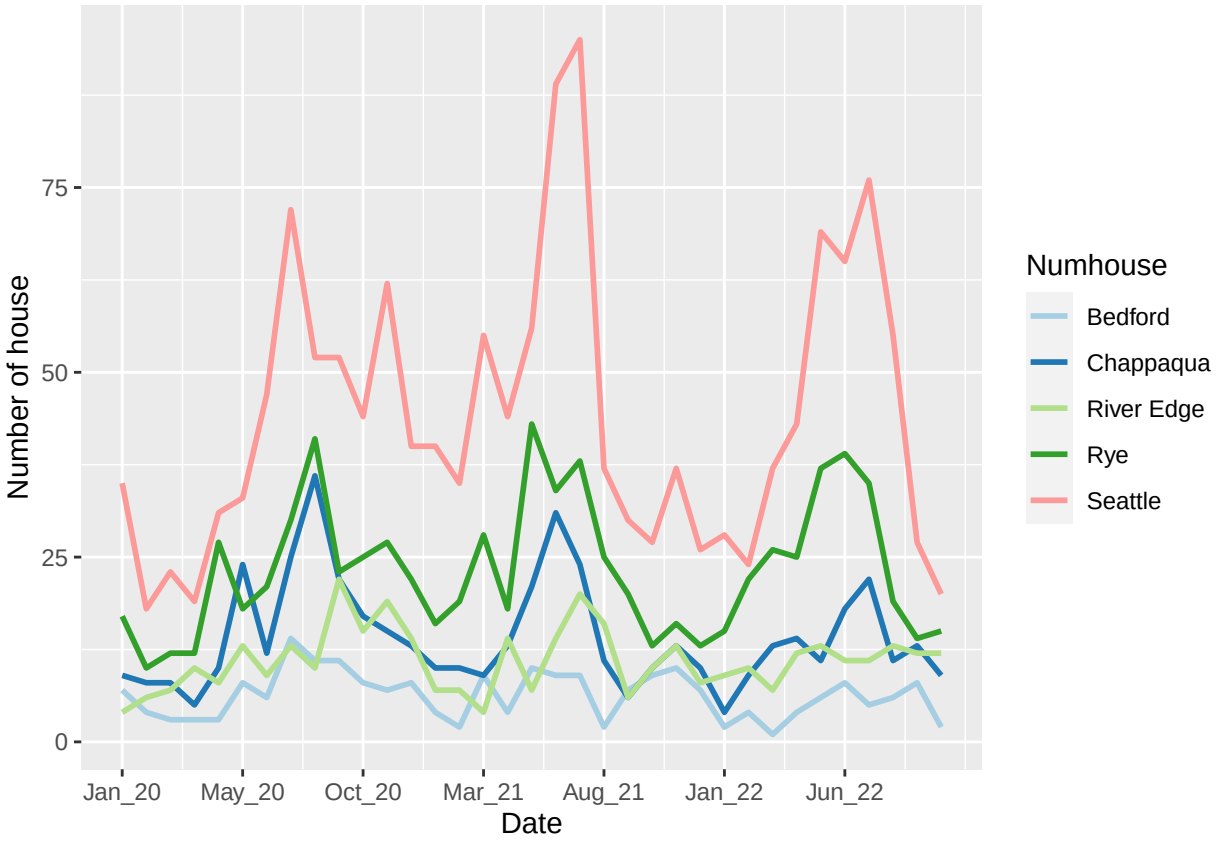

```
data_df1_named %>% group_by(Var2) %>% summarize(mean_price = mean(value))
```

```
## # A tibble: 8 x 2
##   Var2       mean_price
##   <chr>           <dbl>
## 1 Bedford           NA
## 2 Chappaqua        403.
## 3 Honolulu         796.
## 4 Purchase         439.
## 5 River Edge       375.
## 6 Rye              626.
## 7 Scarsdale        494.
## 8 Seattle          517.
```

According to the time series plot for house price per square foot, Bedford has the most fluctuated price over the time. It is one of the most competitive homes sell in 18 days. There is a net population gain in Bedford, with a net search flow of approximately 1100. The house price for Bedford decreases after the increment of interest rate. Rye has the highest house price per square foot among target cities (Rye, Purchase, Chappaqua, Bedford) , and it is relatively stable. The house price per square feet for Purchase does not decrease after the rise in interest rate and it has a persistent increasing trend.

Number of house

```
time2 = colnames(num_sold)[3:37]
num_sold_num = num_sold %>% filter(City_name!="Purchase") %>% filter(City_name!="Seattle") %>% filter(C
df2 = t(num_sold_num[,3:37])
data_df2 = melt(df2)
data_df2_named =
  data_df2 %>%
  mutate(Var2 = ifelse(Var2==1,"Chappaqua",ifelse(Var2==2, "Bedford", ifelse(Var2==3, "Rye", ifelse(Var
         ind = rep(1:length(time2),5),
         Numhouse = Var2)
ggplot(data_df2_named, aes(x=ind, y=value, color = Numhouse))+
  geom_line(size=1)+
  labs(
    y="Number of house",
    x="Date"
  )+
  scale_x_continuous(
    breaks = seq(1,length(time2),5),
    labels = c("Jan_20","May_20","Oct_20","Mar_21","Aug_21","Jan_22","Jun_22"),
    limits = c(1,length(time2)))+
    scale_color_brewer(palette="Paired")
```




```
data_df2_named %>% group_by(Var2) %>% summarize(mean_number = mean(value))
```

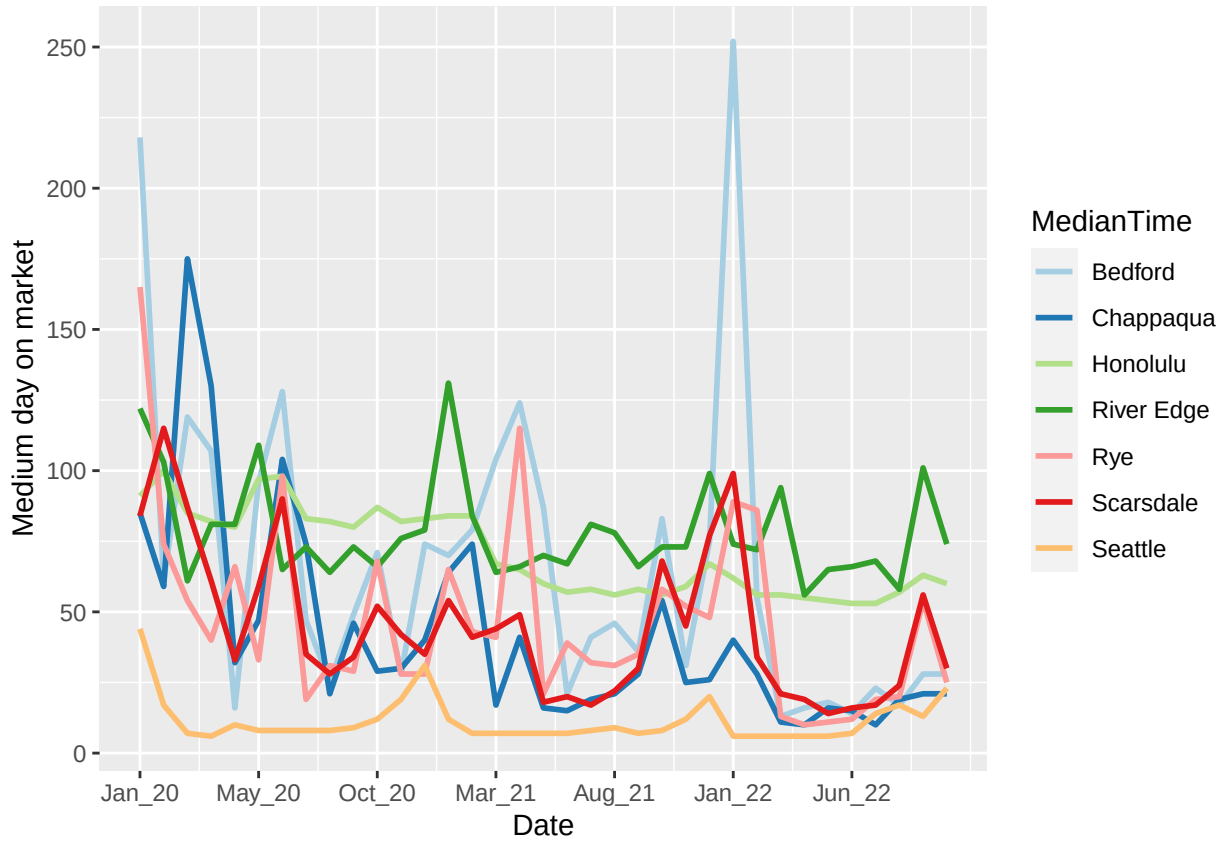
```
## # A tibble: 5 x 2
```

```
##   Var2       mean_number
##   <chr>            <dbl>
## 1 Bedford           6.31
## 2 Chappaqua        14.2
## 3 River Edge       11.0
## 4 Rye              23.3
## 5 Seattle          44.1
```

Purchase data is not available online. Rye is the city that has the highest number of house available among three target cities.

Medium day on the market

```
time3 = colnames(medium_time)[3:37]
medium_time_num = medium_time %>% filter(City_name!="Purchase")
df3 = t(medium_time_num[,3:37])
data_df3 = melt(df3)
data_df3_named =
  data_df3 %>%
  mutate(Var2 = ifelse(Var2==1,"Chappaqua",ifelse(Var2==2, "Bedford", ifelse(Var2==3, "Rye",ifelse(Var2=
         ind = rep(1:length(time3),7),
         MedianTime = Var2)
ggplot(data_df3_named, aes(x=ind, y=value, color = MedianTime))+
  geom_line(size=1)+
  labs(
    y="Medium day on market",
    x="Date"
  )+
  scale_x_continuous(
    breaks = seq(1,length(time2),5),
    labels = c("Jan_20","May_20","Oct_20","Mar_21","Aug_21","Jan_22","Jun_22"),
    limits = c(1,length(time2)))+
    scale_color_brewer(palette="Paired")
```




```
data_df3_named %>% group_by(Var2) %>% summarize(mean_day = mean(value))
```

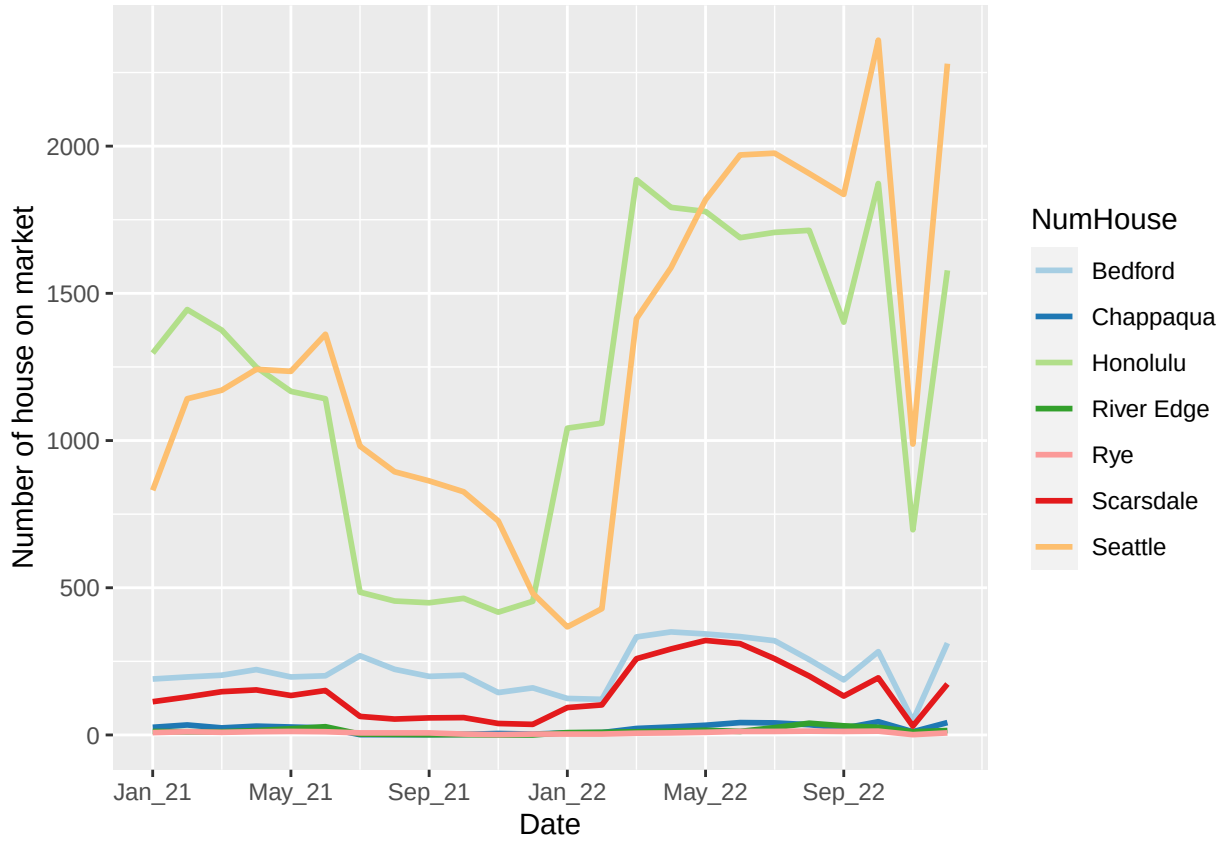
```
## # A tibble: 7 x 2
##   Var2       mean_day
##   <chr>         <dbl>
## 1 Bedford        65.7
## 2 Chappaqua      41.8
## 3 Honolulu       70.6
## 4 River Edge     78.1
## 5 Rye            47.2
## 6 Scarsdale      44.9
## 7 Seattle        11.5
```

Bedford has a relatively high number of days on the market before sold.

Home for sale

```
time4 = colnames(num_house)[3:26]
num_house_num = num_house %>% filter(City_name!="Chappaqua")
df4 = t(num_house_num[,3:26])
data_df4 = melt(df4)
data_df4_named =
  data_df4 %>%
```

```
  mutate(Var2 = ifelse(Var2==1,"Chappaqua",ifelse(Var2==2, "Bedford", ifelse(Var2==3, "Rye",ifelse(Var2=
         ind = rep(1:length(time4),7),
         NumHouse = Var2)
ggplot(data_df4_named, aes(x=ind, y=value, color = NumHouse))+
  geom_line(size=1)+
  labs(
    y="Number of house on market",
    x="Date"
  )+
  scale_x_continuous(
    breaks = seq(1,length(time4),4),
    labels = c("Jan_21","May_21","Sep_21","Jan_22","May_22","Sep_22"),
    limits = c(1,length(time4)))+
    scale_color_brewer(palette="Paired")
```




```
data_df4_named %>% group_by(Var2) %>% summarize(mean_number = mean(value))
```

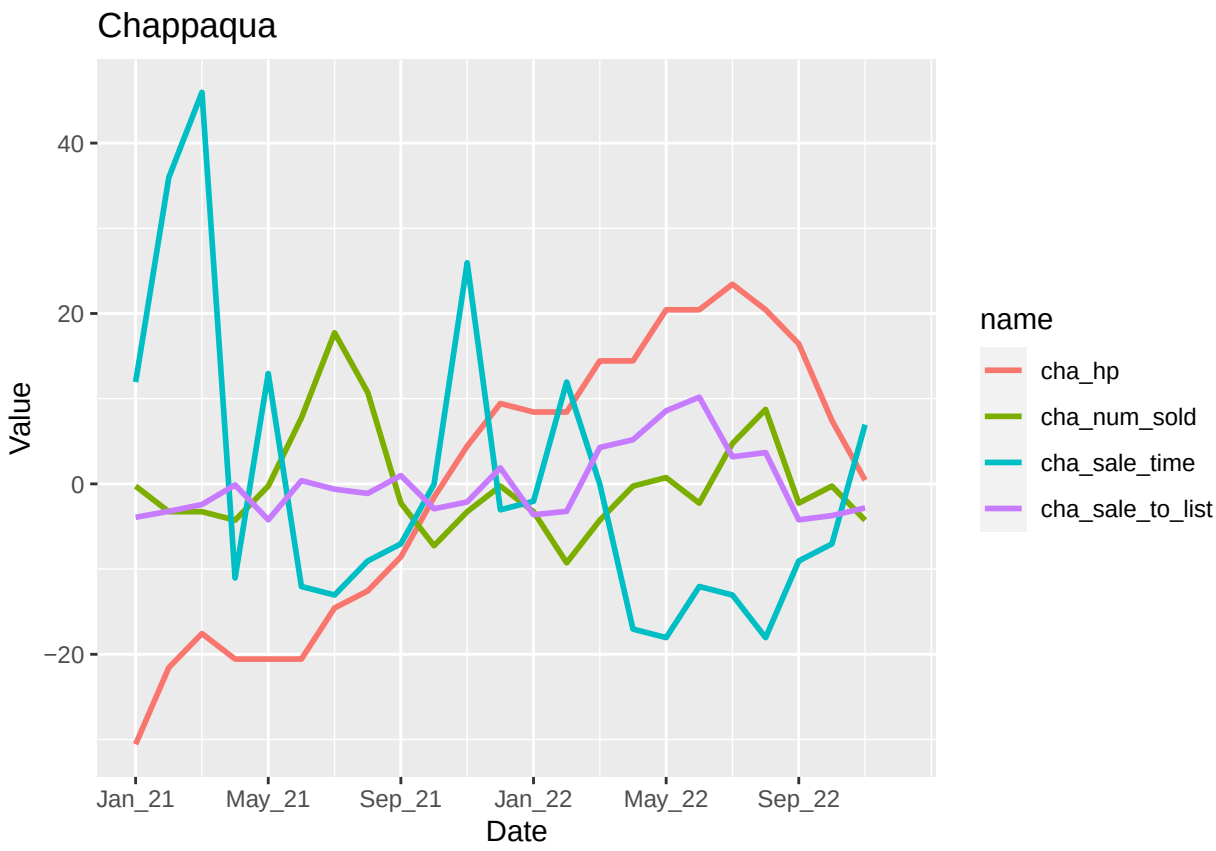
```
## # A tibble: 7 x 2
##   Var2       mean_number
##   <chr>            <dbl>
## 1 Bedford          226.
## 2 Chappaqua         21.4
## 3 Honolulu        1192.
## 4 River Edge        13.4
## 5 Rye                7.83
```

```
## 6 Scarsdale       146.
## 7 Seattle        1279.
```

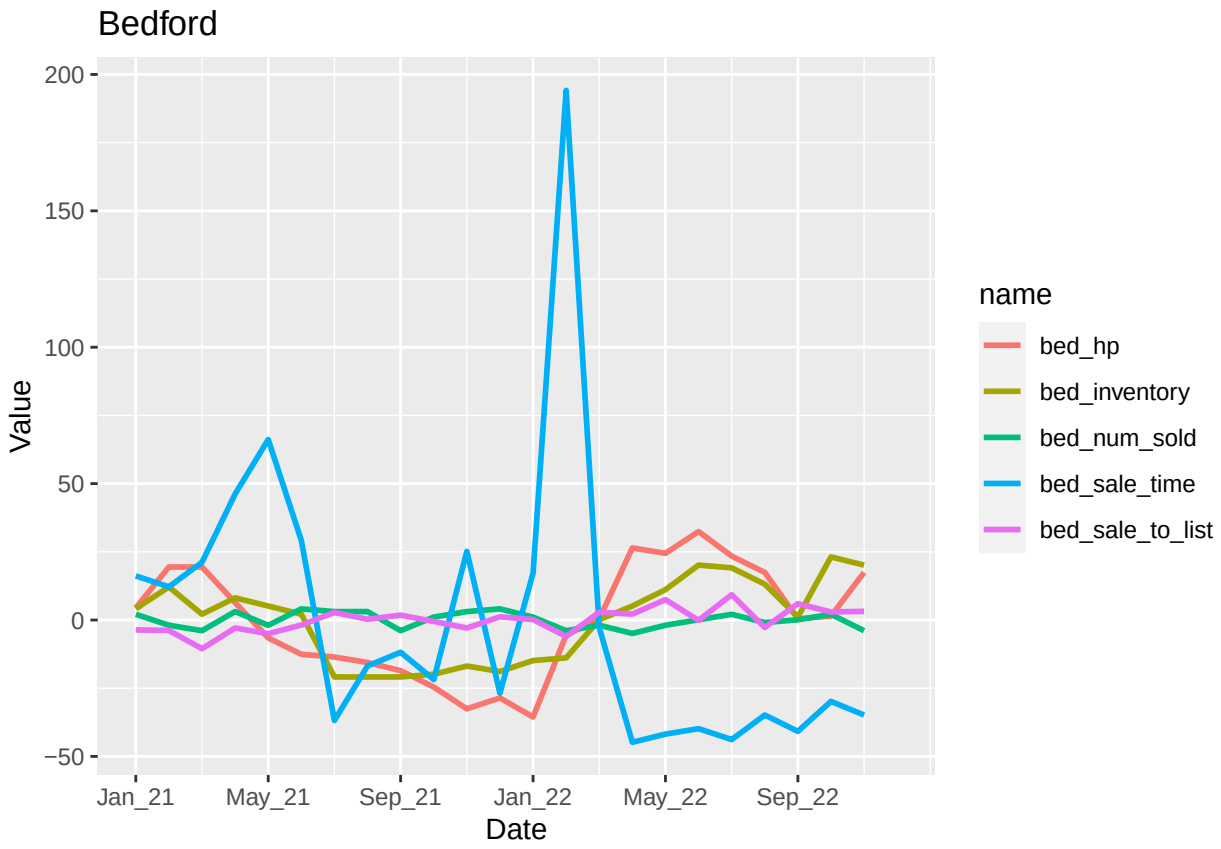
Time series for multiple indicator

This is the plotting that stores all the house price, number of available house, sold number, number of day and the sale to list price is stored in one plotting for each city. Since they are all in different scale, I transfer them into the same scale using standardization so that all trends are captured in each of the plotting.

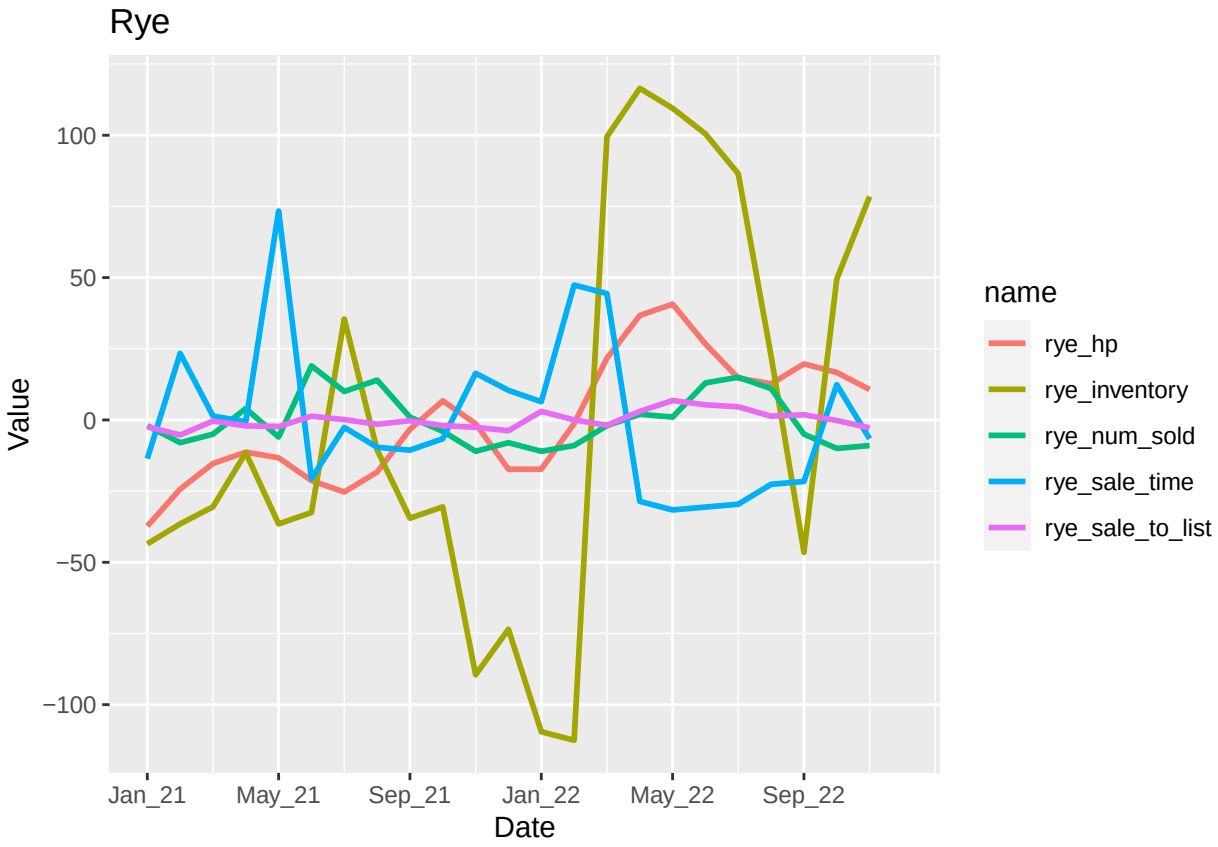
```
std <- function(x){
  return (x-mean(x))/sd(x)
}


#take value from year 2021
cha_num_sold = as.numeric(num_sold[1,][15:37]) %>% std()
#get rid of November as not all dataset contains Nov, this is the 25th row
cha_sale_time = as.numeric(medium_time[1,][15:38][-23]) %>% std()
#get rid of November as not all dataset contains Nov, this is the 25th row
cha_inventory =  as.numeric(num_house[1,][3:26][-23]) %>% std()
cha_sale_to_list =  as.numeric(sale_to_list[1,][15:37]) %>% std()
cha_hp =  as.numeric(hp_three_yr[1,][15:37]) %>% std()

date = seq(1,23,1)
#time_val = format(date, "%B %Y")
#time_val[23]="December 2022"


chapa_df = as.data.frame(cbind(date,cha_num_sold,cha_sale_time,cha_inventory,cha_sale_to_list,cha_hp))
chapa_df_longer <- pivot_longer(chapa_df, c(cha_num_sold,cha_sale_time,cha_inventory,cha_sale_to_list,cl

ggplot(chapa_df_longer, aes(x=date, y=value, color = name))+
  geom_line(size=1)+
  labs(
    y="Value",
    x="Date"
  )+
  scale_x_continuous(
    breaks = seq(1,length(time4),4),
    labels = c("Jan_21","May_21","Sep_21","Jan_22","May_22","Sep_22"),
    limits = c(1,length(time4)))+
  ggtitle("Chappaqua")
```


Chappaqua
40
20
0
−20
Value
Jan_21
May_21
Sep_21
Jan_22
May_22
Sep_22
Date
name
cha_hp
cha_num_sold
cha_sale_time
cha_sale_to_list

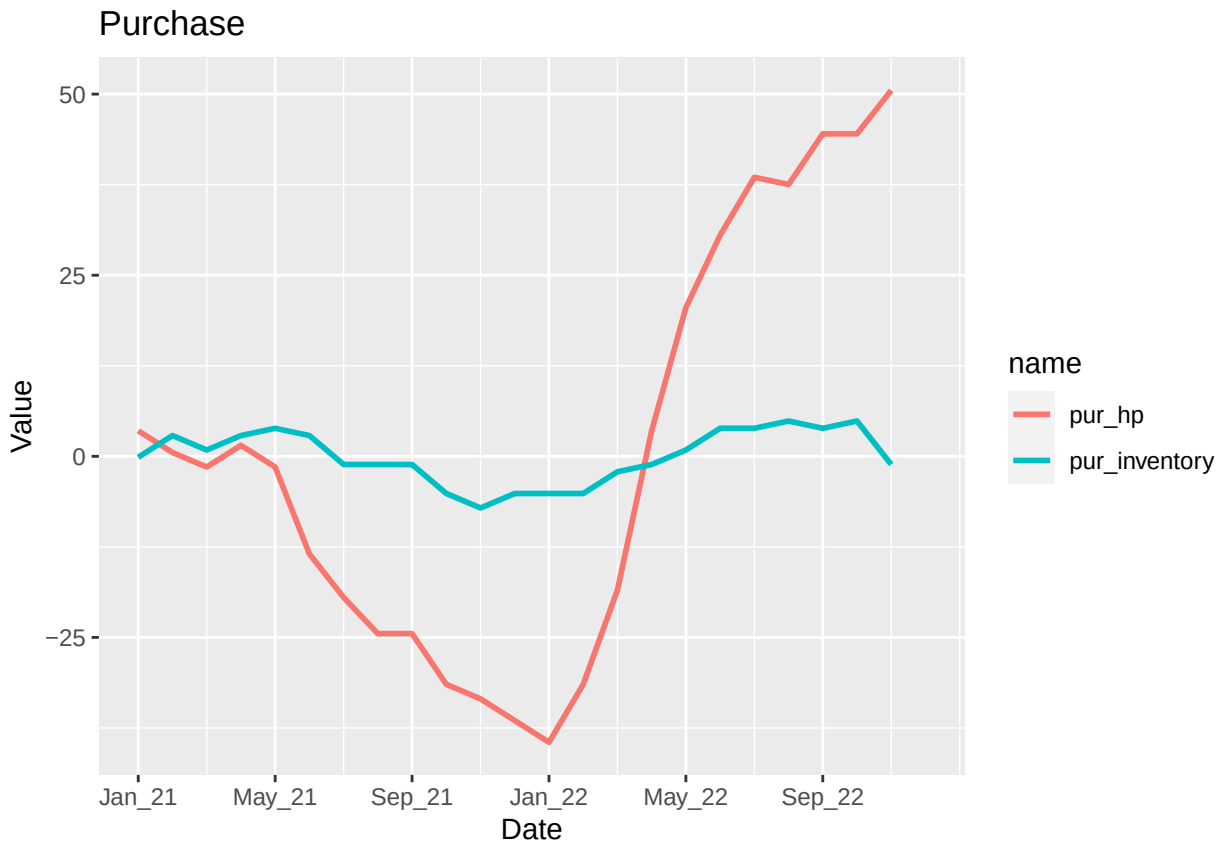
Chappaqua


Bedford
200
150
100
50
0
−50
Value
Jan_21
May_21
Sep_21
Jan_22
May_22
Sep_22
Date
name
bed_hp
bed_inventory
bed_num_sold
bed_sale_time
bed_sale_to_list

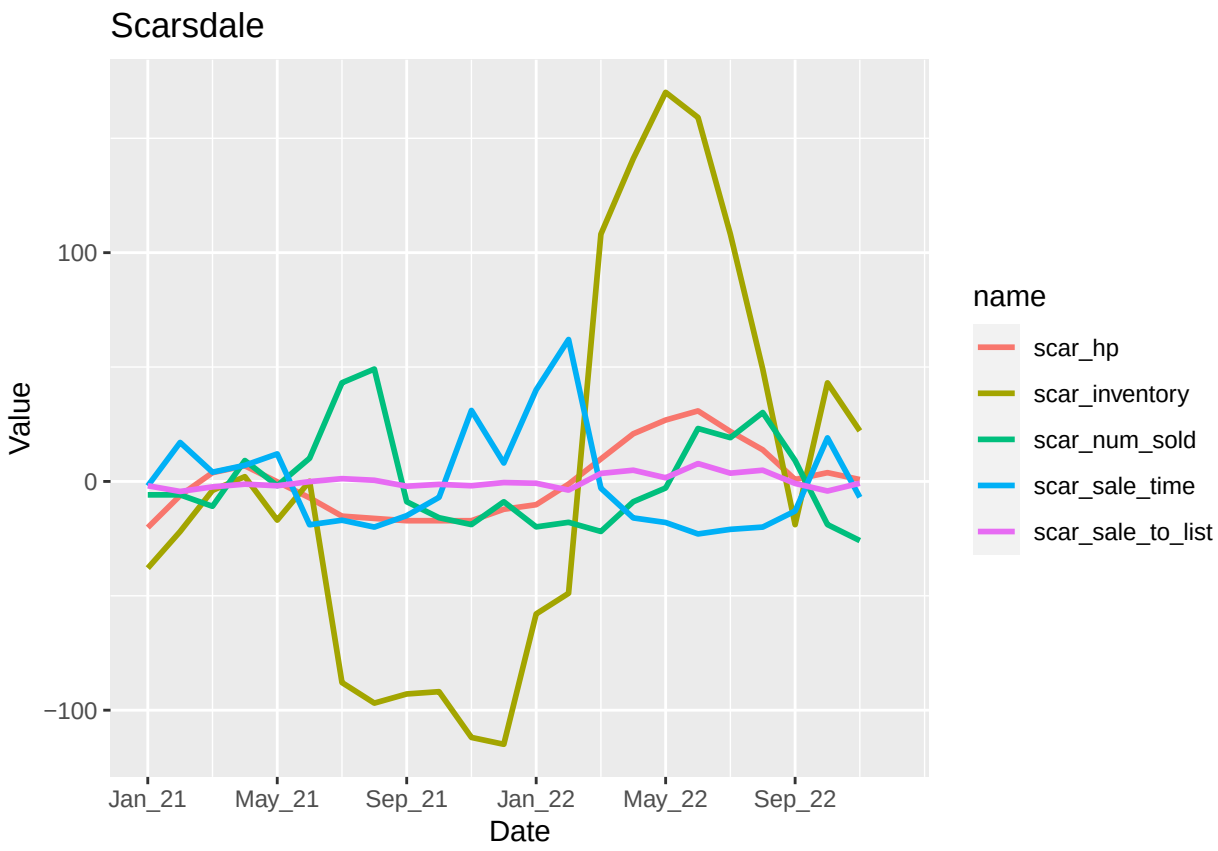
Bedford


Rye
100
50
0
−50
−100
Value
Jan_21
May_21
Sep_21
Jan_22
May_22
Sep_22
Date
name
rye_hp
rye_inventory
rye_num_sold
rye_sale_time
rye_sale_to_list

Rye


Purchase
50
25
0
−25
Value
Jan_21
May_21
Sep_21
Jan_22
May_22
Sep_22
Date
name
pur_hp
pur_inventory

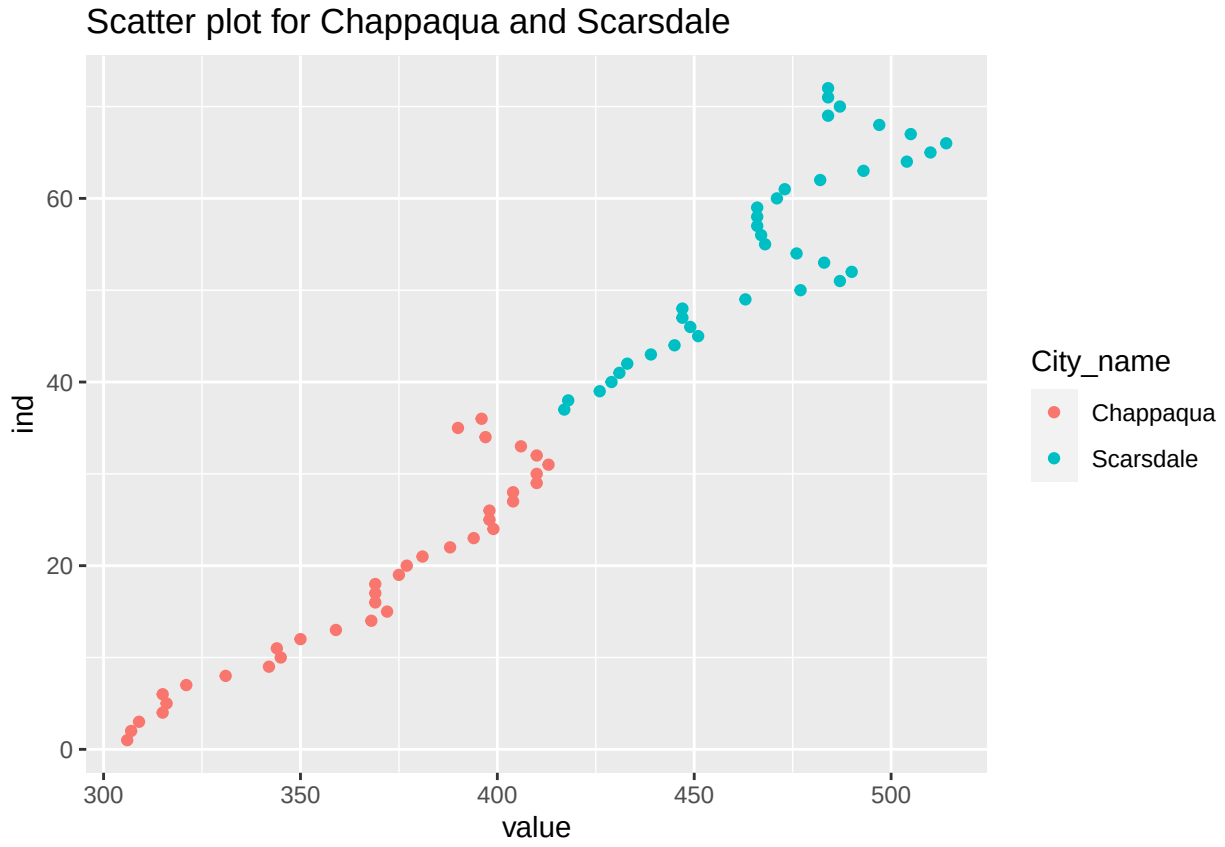
Purchase


Scarsdale
100
0
−100
Value
Jan_21
May_21
Sep_21
Jan_22
May_22
Sep_22
Date
name
scar_hp
scar_inventory
scar_num_sold
scar_sale_time
scar_sale_to_list

Scarsdale

Scatter plot for chappaqua and scarsdale

```
all_val =
  hp_three_yr %>%
  filter(City_name=="Chappaqua" | City_name=="Scarsdale") %>%
  pivot_longer(
    `43831`:`44927`,
    names_to = "date"
  ) %>%
  mutate(date_val = as.Date(as.numeric(date),origin="1899-12-30"),
         ind = seq(1,72,1))

ggplot(all_val,aes(x=value, y=ind, color = City_name))+
  geom_point()+
  labs(title="Scatter plot for Chappaqua and Scarsdale")
```



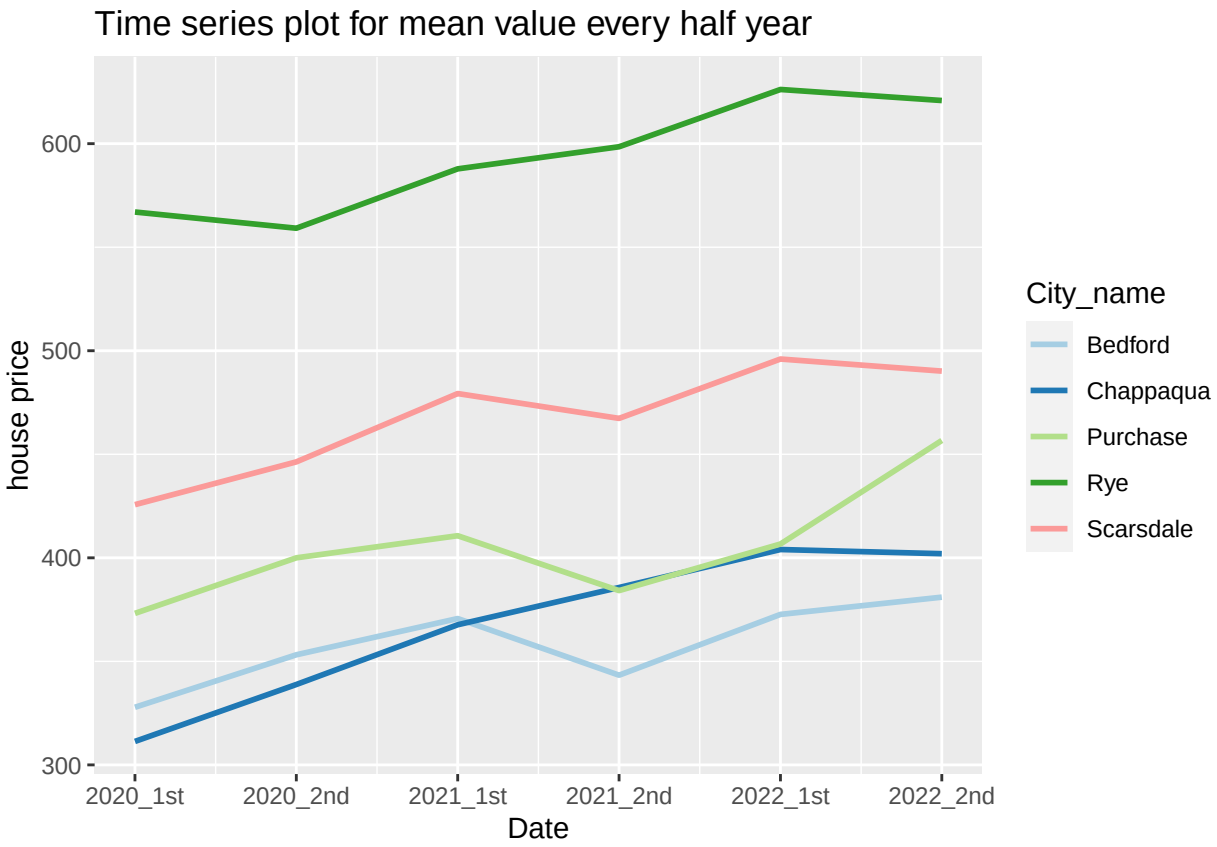
The standard deviation of Chappaqua is `34.948` with a range of `107` dollar per square feet. Scarsdale, on the other hand, has a standard deviation of `26.455` with a range of `97` dollar per square feet. The table shows all the information:

```
## # A tibble: 2 x 4
##   City_name mean_val sd_val range_dollar
##   <chr>        <dbl>  <dbl>        <dbl>
## 1 Chappaqua     368.   34.9          107
## 2 Scarsdale     467.   26.5           97
```

Mean value for each half year

```
## `summarise()` has grouped output by 'City_name'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 5 x 7
##   City_name first_2020 first_2021 first_2022 second_2020 second_2021 second_2022
##   <chr>          <dbl>      <dbl>      <dbl>       <dbl>       <dbl>       <dbl>
## 1 Bedford         328.       371.       373.        353.        343.        381
## 2 Chappaqua       311.       368.       404         339.        386.        402
## 3 Purchase        373.       411.       407.        400         384.        457.
## 4 Rye             567        588.       626.        559.        598.        621.
## 5 Scarsdale       426.       479.       496         446.        467.        490.
```

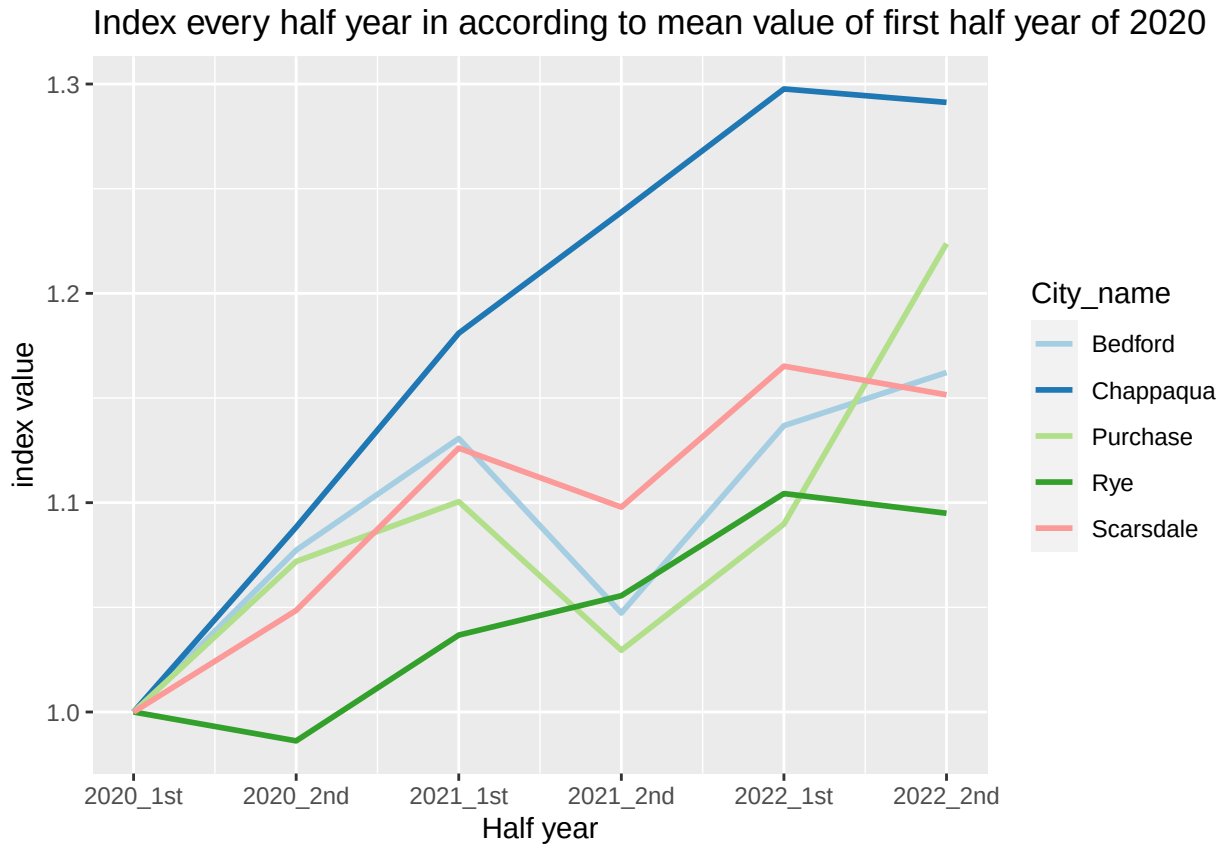


Index

Yearly index

```
## # A tibble: 5 x 7
##   City_name first_2020_index second_2020_index first_2~1 secon~2 first~3 secon~4
##   <chr>                <dbl>             <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 Bedford                  1             1.08       1.13    1.05    1.14    1.16
## 2 Chappaqua                1             1.09       1.18    1.24    1.30    1.29
## 3 Purchase                 1             1.07       1.10    1.03    1.09    1.22
## 4 Rye                      1             0.986      1.04    1.06    1.10    1.09
## 5 Scarsdale                1             1.05       1.13    1.10    1.17    1.15
## # ... with abbreviated variable names 1: first_2021_index,
## #   2: second_2021_index, 3: first_2022_index, 4: second_2022_index
```

Yearly index plot:



Quarterly index

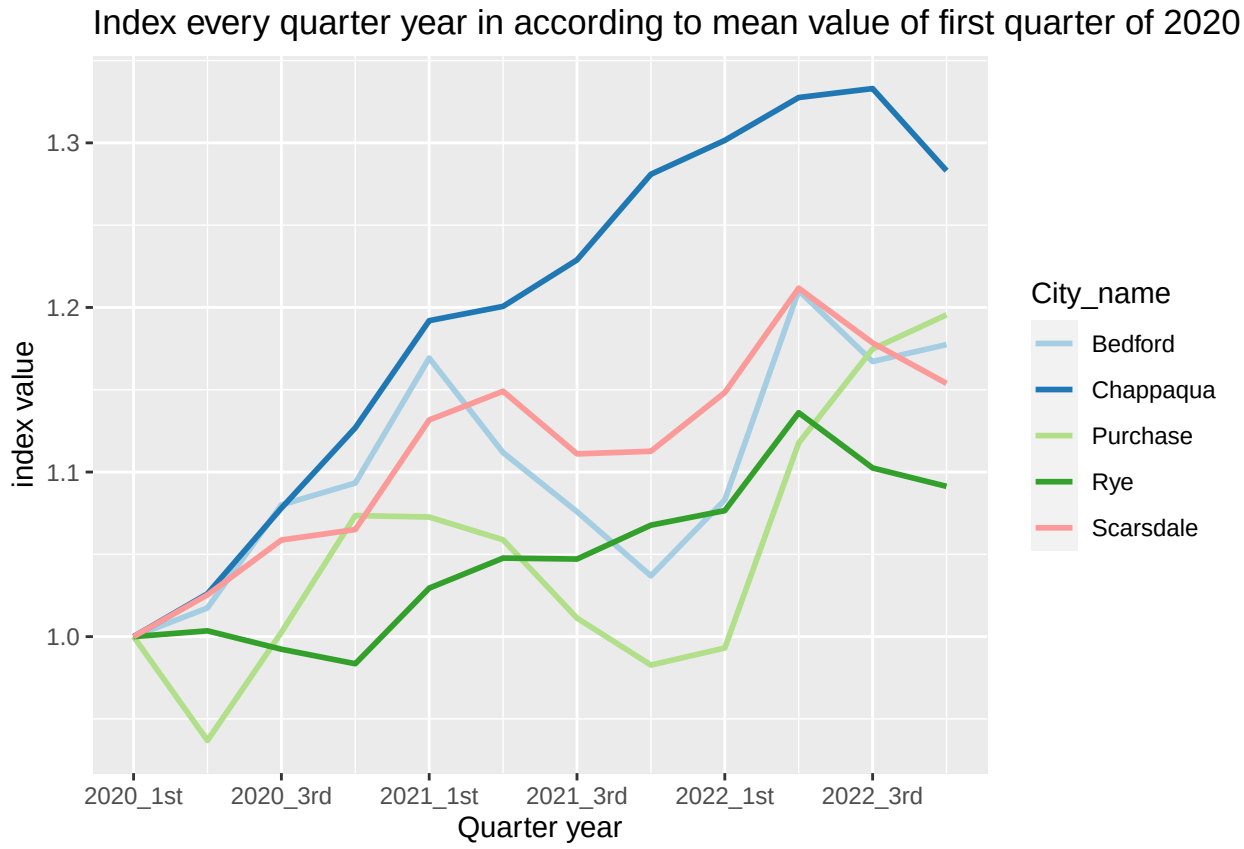
```
## `summarise()` has grouped output by 'City_name'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 5 x 5
##   City_name first_2020_index second_2020_index third_2020_index fourth_2020_in~1
##   <chr>                <dbl>             <dbl>            <dbl>            <dbl>
```

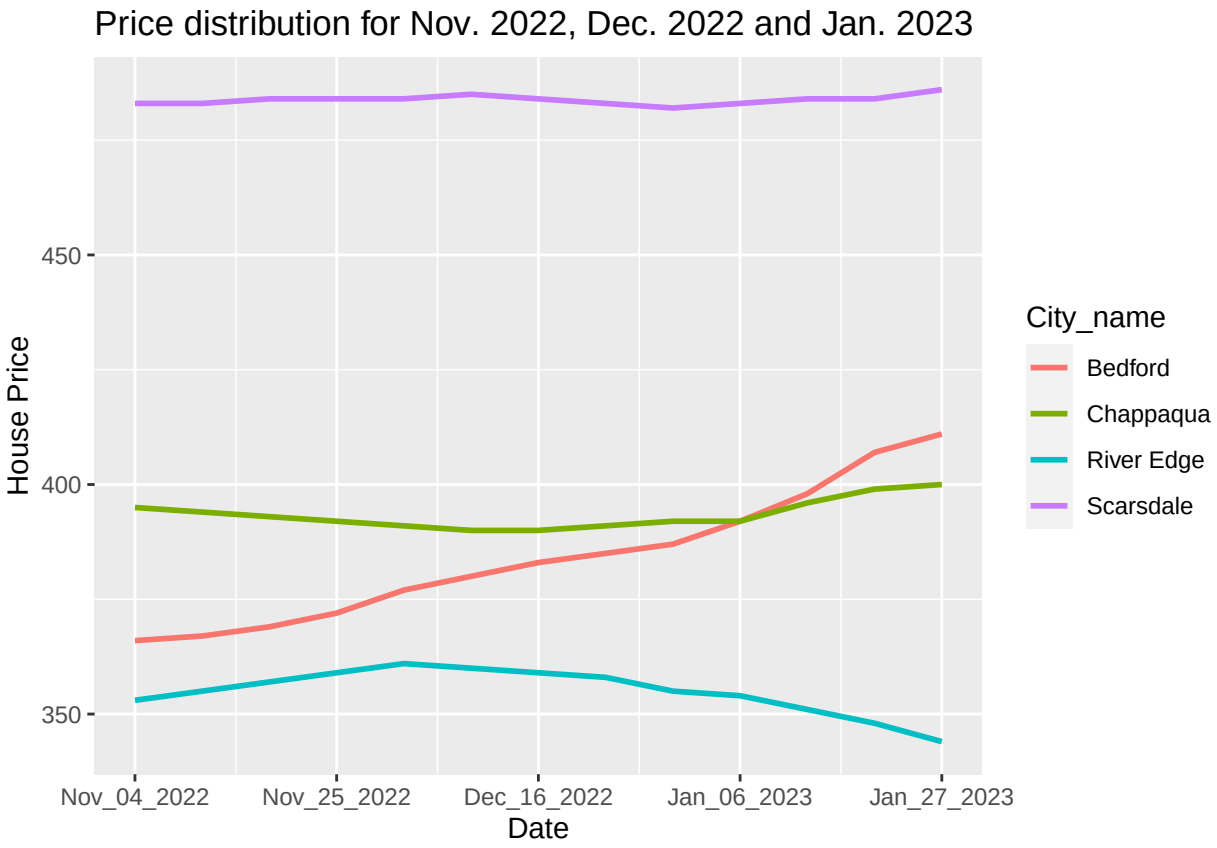
```
## 1 Bedford                   1             1.02             1.08             1.09
## 2 Chappaqua                 1             1.03             1.08             1.13
## 3 Purchase                  1             0.937            1.00             1.07
## 4 Rye                       1             1.00             0.992            0.984
## 5 Scarsdale                 1             1.03             1.06             1.07
## # ... with abbreviated variable name 1: fourth_2020_index
```

```
## # A tibble: 5 x 4
##   first_2021_index second_2021_index third_2021_index fourth_2021_index
##              <dbl>             <dbl>            <dbl>             <dbl>
## 1             1.17              1.11             1.08              1.04
## 2             1.19              1.20             1.23              1.28
## 3             1.07              1.06             1.01              0.983
## 4             1.03              1.05             1.05              1.07
## 5             1.13              1.15             1.11              1.11
```

```
## # A tibble: 5 x 4
##   first_2022_index second_2022_index third_2022_index fourth_2022_index
##              <dbl>             <dbl>            <dbl>             <dbl>
## 1             1.08              1.21             1.17              1.18
## 2             1.30              1.33             1.33              1.28
## 3            0.993              1.12             1.17              1.20
## 4             1.08              1.14             1.10              1.09
## 5             1.15              1.21             1.18              1.15
```



Price distribution for last three month



Summary of the four cities in the range from November 2022 to January 2023

```
## # A tibble: 4 x 4
##   City_name  mean_val sd_val range_dollar
##   <chr>         <dbl>  <dbl>        <dbl>
## 1 Bedford        384.  14.7            45
## 2 Chappaqua      393.   3.23           10
## 3 River Edge     355.   4.97           17
## 4 Scarsdale      484.   1.01            4
```