Final Project

Nam Pham

2023-12-04

Name: Nam H. Pham

Section time: TR 12:00 - 1:15 PM

Loading in data

The data is downloaded with their names unchanged from: https://www.huduser.gov/portal/datasets/fmr.html#data_2024 (https://www.huduser.gov/portal/datasets/fmr.html#data_2024)

The website provide annual Fair Market Rent data. The results, which factor into various housing subsidy programs, represent the 40th percentile cost of monthly rent and (basic) utilities for “recent movers” in “standard quality” units, adjusted for the number of bedrooms.

There are two data sets that I want to investigate: (1) the Fair Market Rents (FMRs) data, which is the 40th percentile data for counties across the US, and (2) the Small Area Fair Market Rents (SAFMRs) data which looks at FMRs calculated for ZIP Codes within Metropolitan Areas.

Separating the two data sets

I want to see all the data I have in my data folder. This can be done with a for loop listing all the files:

```
data_files <- dir("../final_project/data/", pattern = "*")

files_list <- data.frame(name = data_files,
                         rows = NA,
                         cols = NA)

# “$” and “^” recognize line terminators within a string; otherwise, they match only at start and end of the input
for(i in 1:length(data_files)){
  if (grepl('.xls$', data_files[i], ignore.case = TRUE)){
    temp <- read_xls(paste0("../final_project/data/", data_files[i]))
    files_list[i, "rows"] <- nrow(temp)
    files_list[i, "cols"] <- ncol(temp)
  }else if (grepl('.xlsx$', data_files[i], ignore.case = TRUE)){
    temp <- read_xlsx(paste0("../final_project/data/", data_files[i]))
    files_list[i, "rows"] <- nrow(temp)
    files_list[i, "cols"] <- ncol(temp)
  }else if (grepl('.csv$', data_files[i], ignore.case = TRUE)){
    temp <- read_csv(paste0("../final_project/data/", data_files[i]))
    files_list[i, "rows"] <- nrow(temp)
    files_list[i, "cols"] <- ncol(temp)
  }
}

files_list
```

```
##                                      name  rows cols
## 1      FY2010_4050_Final_PostRDDs.xls  4763   20
## 2               FY2011_4050_Final.xls  4765   20
## 3               FY2012_4050_Final.xls  4765   17
## 4               FY2013_4050_Final.xls  4766   18
## 5            FY2014_4050_RevFinal.xls  4766   18
## 6            FY2015_4050_RevFinal.xls  4769   18
## 7         FY2016F-4050-RevFinal4.xlsx  4770   18
## 8  FY2017-4050-County-Level_Data.xlsx  4769   18
## 9           FY2018_4050_FMRs_rev.xlsx  4769   20
## 10         FY2019_4050_FMRs_rev2.xlsx  4767   20
## 11             fy2019_safmrs_rev.xlsx 26019   18
## 12          FY2020_4050_FMRs_rev.xlsx  4766   20
## 13             fy2020_safmrs_rev.xlsx 26090   18
## 14          FY2021_4050_FMRs_rev.xlsx  4766   16
## 15         fy2021_safmrs_revised.xlsx 27144   18
## 16           FY2022_FMRs_revised.xlsx  4765   14
## 17         fy2022_safmrs_revised.xlsx 27322   18
## 18           FY2023_FMRs_revised.xlsx  4764   14
## 19         fy2023_safmrs_revised.xlsx 27331   18
## 20                   FY2024_FMRs.xlsx  4764   14
## 21                 fy2024_safmrs.xlsx 27446   18
## 22              zip_code_database.csv 42735   15
```

Observation:

From the way that the number of rows and columns increasing and decreasing, some observations/feature might exist in one year and not exist in another. I would need some way to check the differences in values

Loading the data sets

Wow that actually worked! - Now I need to differentiate between FMRs and SAFMRs. - Extract the year value in `year` - Create a column for the names of the data imported in the workbook as `data_import`. This is done by using the part of the filename that is reusable (the value of year) and extracting it with stri_extract. Then using the paste function to create a column of variables for data_import using mutate - Using the column, load the data for small area. I’ll use a for loop

```
files_list <- files_list|>
  mutate(safmr_or_not = if_else(grepl('safmrs', name), TRUE, FALSE))|>
  mutate(year = as.double(stri_extract_first_regex(name, '\\p{N}+')))|>
  mutate(data_import = tolower(stri_extract(name, regex = '^.{1,6}')))|>
  mutate(data_import = if_else(
    safmr_or_not == FALSE,
    paste('fmr',data_import, sep = '_'),
    paste('safmr',data_import, sep = '_')
  ))
files_list
```

```
##                                   name  rows cols safmr_or_not ye
ar
## 1      FY2010_4050_Final_PostRDDs.xls  4763   20        FALSE 20
10
## 2               FY2011_4050_Final.xls  4765   20        FALSE 20
11
## 3               FY2012_4050_Final.xls  4765   17        FALSE 20
12
## 4               FY2013_4050_Final.xls  4766   18        FALSE 20
13
## 5            FY2014_4050_RevFinal.xls  4766   18        FALSE 20
14
## 6            FY2015_4050_RevFinal.xls  4769   18        FALSE 20
15
## 7         FY2016F-4050-RevFinal4.xlsx  4770   18        FALSE 20
16
## 8  FY2017-4050-County-Level_Data.xlsx  4769   18        FALSE 20
17
## 9           FY2018_4050_FMRs_rev.xlsx  4769   20        FALSE 20
18
## 10         FY2019_4050_FMRs_rev2.xlsx  4767   20        FALSE 20
19
## 11             fy2019_safmrs_rev.xlsx 26019   18         TRUE 20
19
## 12          FY2020_4050_FMRs_rev.xlsx  4766   20        FALSE 20
20
## 13             fy2020_safmrs_rev.xlsx 26090   18         TRUE 20
20
## 14          FY2021_4050_FMRs_rev.xlsx  4766   16        FALSE 20
21
## 15         fy2021_safmrs_revised.xlsx 27144   18         TRUE 20
21
## 16           FY2022_FMRs_revised.xlsx  4765   14        FALSE 20
22
## 17         fy2022_safmrs_revised.xlsx 27322   18         TRUE 20
22
```

```
## 18           FY2023_FMRs_revised.xlsx  4764   14        FALSE  20
23
## 19         fy2023_safmrs_revised.xlsx 27331   18         TRUE  20
23
## 20                   FY2024_FMRs.xlsx  4764   14        FALSE  20
24
## 21                 fy2024_safmrs.xlsx 27446   18         TRUE  20
24
## 22              zip_code_database.csv 42735   15        FALSE
NA
##     data_import
## 1    fmr_fy2010
## 2    fmr_fy2011
## 3    fmr_fy2012
## 4    fmr_fy2013
## 5    fmr_fy2014
## 6    fmr_fy2015
## 7    fmr_fy2016
## 8    fmr_fy2017
## 9    fmr_fy2018
## 10   fmr_fy2019
## 11 safmr_fy2019
## 12   fmr_fy2020
## 13 safmr_fy2020
## 14   fmr_fy2021
## 15 safmr_fy2021
## 16   fmr_fy2022
## 17 safmr_fy2022
## 18   fmr_fy2023
## 19 safmr_fy2023
## 20   fmr_fy2024
## 21 safmr_fy2024
## 22   fmr_zip_co
```

```
# seq_along would actually loop over all the rows in the data frame, where as
# length only give one value(final value)
# This is actually really useful so let's make it easier to see
# assign() function to assign variables
# paste() used to get a string value where it is used

for (i in seq_along(files_list[ , "name"])) {

  if (grepl('.xls$', files_list[i, "name"], ignore.case = TRUE)){

    assign(paste(files_list[i,"data_import"]),
         read_xls(paste0("../final_project/data/",files_list[i,"name"])))

  }else if (grepl('.xlsx$', files_list[i, "name"], ignore.case = TRUE)){

    assign(paste(files_list[i,"data_import"]),
         read_xlsx(paste0("../final_project/data/",files_list[i,"name"])))
  }else if (grepl('.csv$', data_files[i], ignore.case = TRUE)){

    assign(paste(files_list[i,"data_import"]),
         read_csv(paste0("../final_project/data/",files_list[i,"name"])))
  }
}
```

Clean/Summarize the contents of the data

Looks like the SAFMRs data is quite well organized. Let’s make a combined data set called `safmrs19-24` that combines the information from all the spreadsheets - First, change the column names to lowercase and without spaces - Second, have uniform column values - Third, add a year column and separate the area value and the state value -

```
#A necessary package for deep cleaning

#install.packages("janitor")
#library(janitor)
```

```
# 1. Cleaning the column names
safmr_fy2019<-janitor::clean_names(safmr_fy2019)
safmr_fy2020<-janitor::clean_names(safmr_fy2020)
safmr_fy2021<-janitor::clean_names(safmr_fy2021)
safmr_fy2022<-janitor::clean_names(safmr_fy2022)
safmr_fy2023<-janitor::clean_names(safmr_fy2023)
safmr_fy2024<-janitor::clean_names(safmr_fy2024)
```

```
# 2. Changing column values for uniformity
safmr_new_names <- c("zip", "area_code", "area_name",
                     "safmr_0br", "safmr_0br_90", "safmr_0br_110",
                     "safmr_1br", "safmr_1br_90", "safmr_1br_110",
                     "safmr_2br", "safmr_2br_90", "safmr_2br_110",
                     "safmr_3br", "safmr_3br_90", "safmr_3br_110",
                     "safmr_4br", "safmr_4br_90", "safmr_4br_110")

# The column data is quite uniform already so this works
colnames(safmr_fy2019) <- safmr_new_names
colnames(safmr_fy2020) <- safmr_new_names
colnames(safmr_fy2021) <- safmr_new_names
colnames(safmr_fy2022) <- safmr_new_names
colnames(safmr_fy2023) <- safmr_new_names
colnames(safmr_fy2024) <- safmr_new_names
```

```
# 3. Add year column
safmr_fy2019 <- safmr_fy2019|>
  mutate(year = 2019)|>
  mutate(state = gsub("[, ]", "",stri_extract_first_regex(area_name,", \\p{L}+ ")))|>
  mutate(area = gsub("[,]", "", stri_extract_all_regex(area_name,".*[^,],")))

safmr_fy2020 <- safmr_fy2020|>
  mutate(year = 2020)|>
  mutate(state = gsub("[, ]", "",stri_extract_first_regex(area_name,", \\p{L}+ ")))|>
  mutate(area = gsub("[,]", "", stri_extract_all_regex(area_name,".*[^,],")))

safmr_fy2021 <- safmr_fy2021|>
  mutate(year = 2021)|>
  mutate(state = gsub("[, ]", "",stri_extract_first_regex(area_name,", \\p{L}+ ")))|>
  mutate(area = gsub("[,]", "", stri_extract_all_regex(area_name,".*[^,],")))

safmr_fy2022 <- safmr_fy2022|>
  mutate(year = 2022)|>
  mutate(state = gsub("[, ]", "",stri_extract_first_regex(area_name,", \\p{L}+ ")))|>
  mutate(area = gsub("[,]", "", stri_extract_all_regex(area_name,".*[^,],")))

safmr_fy2023 <- safmr_fy2023|>
  mutate(year = 2023)|>
  mutate(state = gsub("[, ]", "",stri_extract_first_regex(area_name,", \\p{L}+ ")))|>
  mutate(area = gsub("[,]", "", stri_extract_all_regex(area_name,".*[^,],")))

safmr_fy2024 <- safmr_fy2024|>
  mutate(year = 2024)|>
  mutate(state = gsub("[, ]", "",stri_extract_first_regex(area_name,", \\p{L}+ ")))|>
  mutate(area = gsub("[,]", "", stri_extract_all_regex(area_name,".*[^,],")))

#Print one table
safmr_fy2019
```

```
## # A tibble: 26,019 x 21
##    zip   area_code  area_name safmr_0br safmr_0br_90 safmr_0br_1
10 safmr_1br
##    <chr> <chr>      <chr>         <dbl>        <dbl>         <db
l>     <dbl>
##  1 76437 METRO1018~ Abilene,~       510          459            5
61       550
##  2 76443 METRO1018~ Abilene,~       510          459            5
61       550
##  3 76464 METRO1018~ Abilene,~       510          459            5
61       550
##  4 76469 METRO1018~ Abilene,~       520          468            5
72       550
##  5 79501 METRO1018~ Abilene,~       600          540            6
60       630
##  6 79503 METRO1018~ Abilene,~       550          495            6
05       570
##  7 79504 METRO1018~ Abilene,~       540          486            5
94       560
##  8 79508 METRO1018~ Abilene,~       650          585            7
15       670
##  9 79510 METRO1018~ Abilene,~       610          549            6
71       640
## 10 79519 METRO1018~ Abilene,~       600          540            6
60       630
## # i 26,009 more rows
## # i 14 more variables: safmr_1br_90 <dbl>, safmr_1br_110 <dbl>,
## #   safmr_2br <dbl>, safmr_2br_90 <dbl>, safmr_2br_110 <dbl>,
## #   safmr_3br <dbl>, safmr_3br_90 <dbl>, safmr_3br_110 <dbl>,
## #   safmr_4br <dbl>, safmr_4br_90 <dbl>, safmr_4br_110 <dbl>, ye
ar <dbl>,
## #   state <chr>, area <chr>
```

The variables are:

- **zip_code**: The area zip code
- **area_code**: area code for the metro the small area is associated with

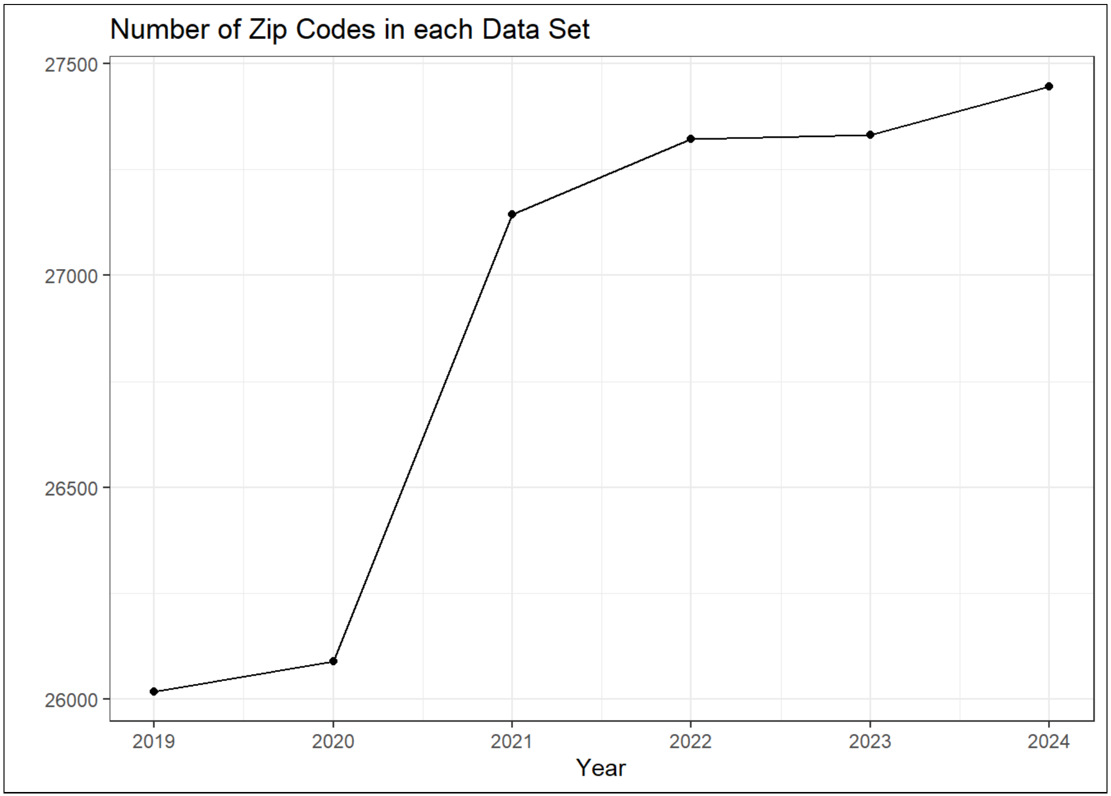
- **area_name**: Full name of the area with state and metro
- **safmr_0br** to **safmr_4br**: the small area fair market rents for rooms 0 bedroom to 4 bedrooms
- **safmr_0br_90**: 0.9 times of the safmr value
- **safmr_0br_110**: 1.1 times of the safmr value
- **year**: the year that the spreadsheet collected data for
- **state**: places in the US
- **area**: name of the area

Questions

Question 1: Visualize the small area data, how many observations are in each? To do this all 6 of the data sets must be merged using `bind_rows`. Now ‘zip’ and ‘year’ is the primary key - create a new variable called `safmr_merged` by using function `bind_rows` on all the safmr data - Make a line-plot showing the change in observations

```
theme_set(theme_bw())
```

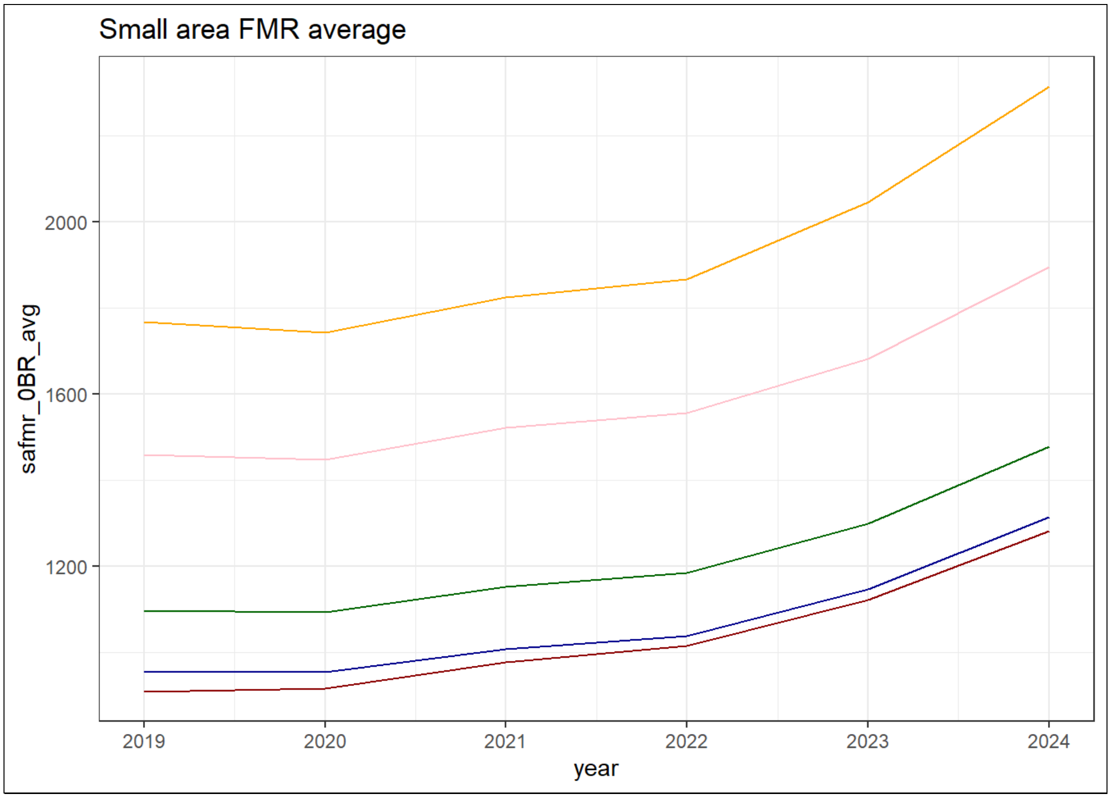
```
safmr_merged <- bind_rows(safmr_fy2019, safmr_fy2020, safmr_fy2021,
                          safmr_fy2022, safmr_fy2023, safmr_fy2024)

g1 <- safmr_merged|>
  group_by(year)|>
  summarize(counts = n())|>
  ggplot(aes(x = year, y = counts))



g1 +
  geom_line()+
  geom_point()+
  labs(
    title = "Number of Zip Codes in each Data Set",
    x = "Year",
    y = ""
  )
```



Question 2: So there is a big difference in the number of observations between years. 2024 increased by almost 1500 observations. That means a lot of new areas are considered to be included in the small area fair market rent with new zip codes.

But let's take a look close to home. Filter out from the `area` column only the city of `Richmond`, and from the `state` column the value `VA` for Virginia. Then make some summary data: - How many observations each year? - Average 0-4 bedroom rents?

```
safmr_merged|>
  filter(area == "Richmond", state == "VA")|>
  group_by(year)|>
  summarize(counts = n(),
            safmr_0BR_avg = sum(safmr_0br)/n(),
            safmr_1BR_avg = sum(safmr_1br)/n(),
            safmr_2BR_avg = sum(safmr_2br)/n(),
            safmr_3BR_avg = sum(safmr_3br)/n(),
            safmr_4BR_avg = sum(safmr_4br)/n()
            ) |>
  ggplot(aes(x = year))+
    geom_line(aes(y = safmr_0BR_avg), color = "darkred")+
    geom_line(aes(y = safmr_1BR_avg), color = "darkblue")+
    geom_line(aes(y = safmr_2BR_avg), color = "darkgreen")+
    geom_line(aes(y = safmr_3BR_avg), color = "pink")+
    geom_line(aes(y = safmr_4BR_avg), color = "orange")+
    labs(
      title = "Small area FMR average ",

    )
```



Question 3: Perform the same plot but inflation adjusted by multiplying each year's average by a scaling constant compared to 2018:

Why 2018?: Because it's 2024 is not over yet. I'll assume the 2019 values were calculated using 2018 money. I'll multiply by the cumulative inflation https://smartasset.com/investing/inflation-calculator (https://smartasset.com/investing/inflation-calculator)

2018 - 2019: 1.81% , 2018 - 2020: 3.07%, 2018 - 2021: 7.91%, 2018 - 2022: 16.86%, 2018 - 2023: 21.02%

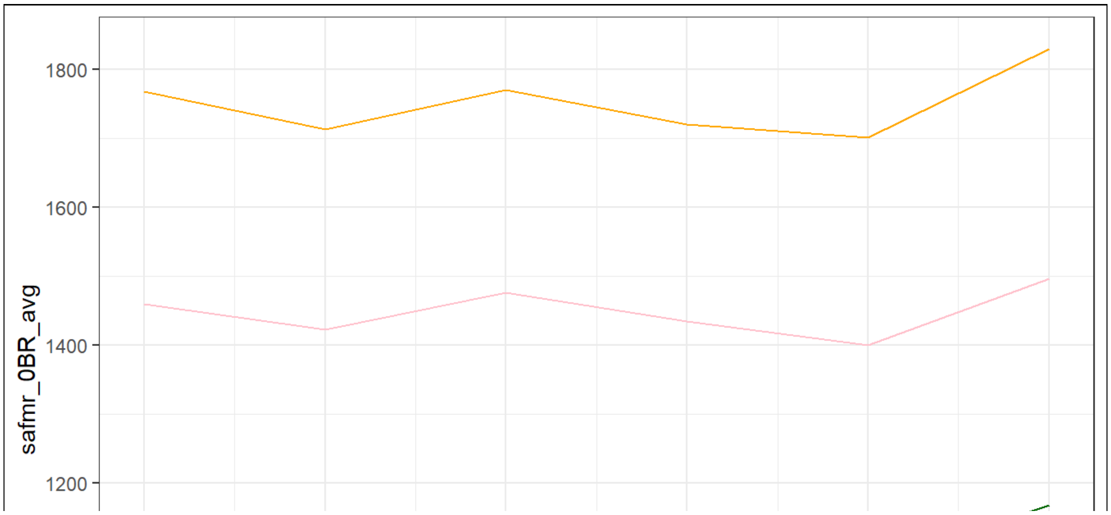
ChatGPT: When comparing Fair Market Rents (FMRs) or Small Area FMRs (SAFMRs) across different years, it's crucial to consider not only the inherent inflation adjustments within each year but also the relative inflation or changes in the value of money over the years.

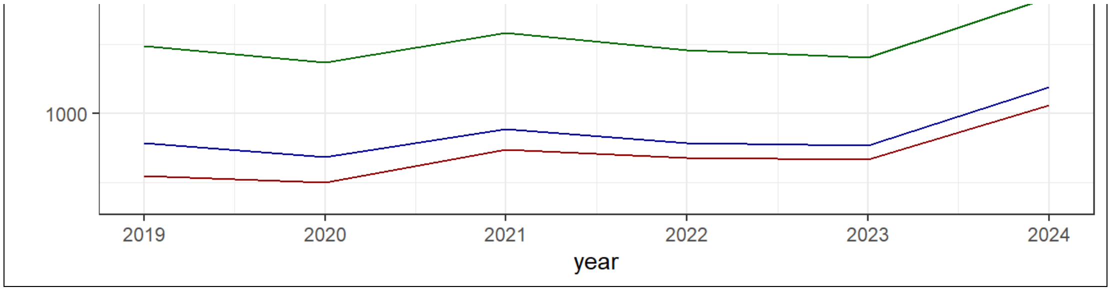
If you want to compare the FMRs from 2020 to 2024 with the FMR from 2019 in terms of their real value, you would indeed need to apply an additional inflation factor for each year. This ensures that you are adjusting for the changes in purchasing power over time.

The process would involve scaling the FMR from each subsequent year by the cumulative inflation factor from 2019 to that specific year. For example, if you have inflation rates for each individual year (e.g., 2019 to 2020, 2020 to 2021, and so on), you would multiply the FMR from each year by the product of the corresponding inflation factors.

This way, you're normalizing the FMR values to a consistent reference point, allowing for a more meaningful comparison of their real economic impact over the specified time frame. It's a thoughtful approach to ensure that your comparisons accurately reflect changes in the actual purchasing power of the currency across the years.

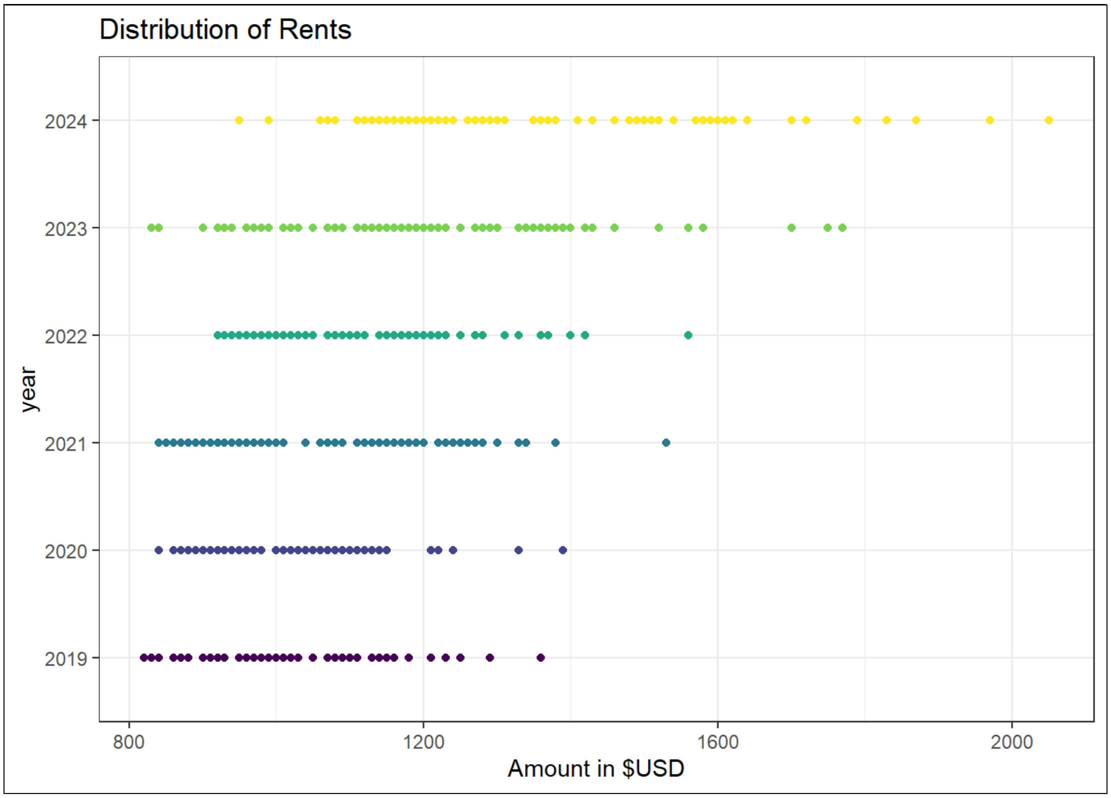
```
safmr_merged|>
  filter(area == "Richmond", state == "VA")|>
  group_by(year)|>
  summarize(counts = n(),
            safmr_0BR_avg = sum(safmr_0br)/n(),
            safmr_1BR_avg = sum(safmr_1br)/n(),
            safmr_2BR_avg = sum(safmr_2br)/n(),
            safmr_3BR_avg = sum(safmr_3br)/n(),
            safmr_4BR_avg = sum(safmr_4br)/n()
            ) |>
  mutate(safmr_0BR_avg = (1-c(0,0.0181,0.0307,0.0791,0.1686,0.210
         2))*safmr_0BR_avg,
         safmr_1BR_avg = (1-c(0,0.0181,0.0307,0.0791,0.1686,0.210
         2))*safmr_1BR_avg,
         safmr_2BR_avg = (1-c(0,0.0181,0.0307,0.0791,0.1686,0.210
         2))*safmr_2BR_avg,
         safmr_3BR_avg = (1-c(0,0.0181,0.0307,0.0791,0.1686,0.210
         2))*safmr_3BR_avg,
         safmr_4BR_avg = (1-c(0,0.0181,0.0307,0.0791,0.1686,0.210
         2))*safmr_4BR_avg,
         )|>
  ggplot(aes(x = year))+
    geom_line(aes(y = safmr_0BR_avg), color = "darkred")+
    geom_line(aes(y = safmr_1BR_avg), color = "darkblue")+
    geom_line(aes(y = safmr_2BR_avg), color = "darkgreen")+
    geom_line(aes(y = safmr_3BR_avg), color = "pink")+
    geom_line(aes(y = safmr_4BR_avg), color = "orange")
```





Question 4: There should be a better way to visualize this. Instead of using lines and averages, why not use points to indicate each observation as a point.

```
safmr_merged|>
  filter(area == "Richmond", state == "VA")|>
  group_by(year)|>
  mutate(year = as.character(year))|>
  ggplot(aes(x = safmr_1br, y = year))+
    geom_point(aes(color = year))+
    scale_color_viridis_d()+
    theme(legend.position = "none")+
    labs(
      title = "Distribution of Rents",
      x = "Amount in $USD",
    )
```

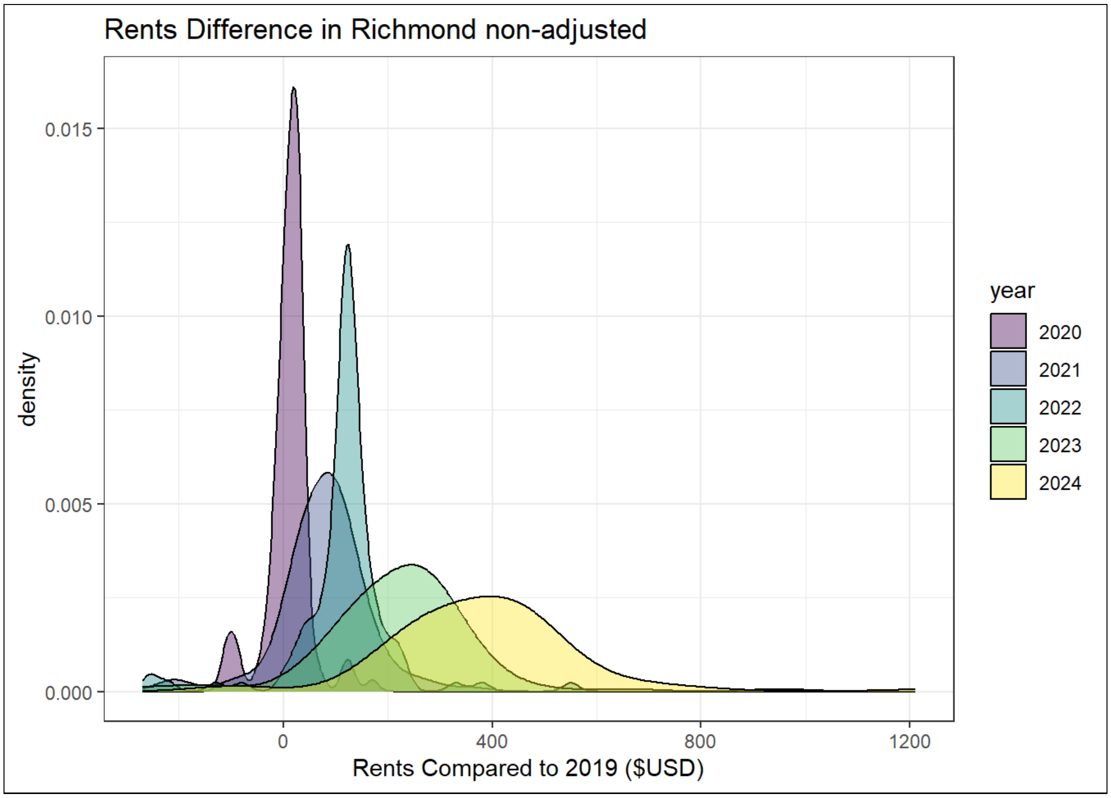


Question 5: We can use the rents 2019 as a baseline to calculate changes across year, that basically means calculating the difference between years. This would effectively lose 1 “degree of freedom” but give us a value that represents change

To do this, we would nee to pivot the data wider, including columns that represent years such as y_19_0br, y_20_0br.

a. Looking at the points, I’m reminded of density plots so why not try doing that this time around.
b. Inflation adjusted so we get an unbiased view

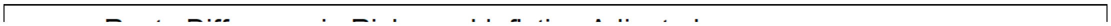
```
safmr_merged|>
  # pivot wider so that we can isolate the similar zip codes
  select(
    zip, area_code, area, state, year,
    safmr_0br, safmr_1br, safmr_2br, safmr_3br, safmr_4br
  )|>
  filter(area == "Richmond", state == "VA")|>
  pivot_wider(
    names_from = year,
    names_prefix = "y",
    values_from = c(safmr_0br, safmr_1br,safmr_2br, safmr_3br, safm
         r_4br)
    )|>
  na.omit()|>
  # A lot of NAs in the wide table 131 to 109, lost 22

  # calculate the difference
  group_by(zip)|>
  summarize(
    safmr_0br_19vs20 = safmr_0br_y2020 - safmr_0br_y2019,
    safmr_0br_19vs21 = safmr_0br_y2021 - safmr_0br_y2019,
    safmr_0br_19vs22 = safmr_0br_y2022 - safmr_0br_y2019,
    safmr_0br_19vs23 = safmr_0br_y2023 - safmr_0br_y2019,
    safmr_0br_19vs24 = safmr_0br_y2024 - safmr_0br_y2019,
  )|>
  pivot_longer(
    -c(zip),
    names_to = "type",
    values_to = "difference"
    )|>

  ggplot(aes(x = difference, group = type, fill = type))+
    geom_density(adjust=1.5, alpha=.4)+
    labs(
      title = "Rents Difference in Richmond non-adjusted",
      x = "Rents Compared to 2019 ($USD)",
      fill = "year"
    )+
    scale_fill_viridis_d( labels = c('2020','2021','2022','2023','2024'))
```

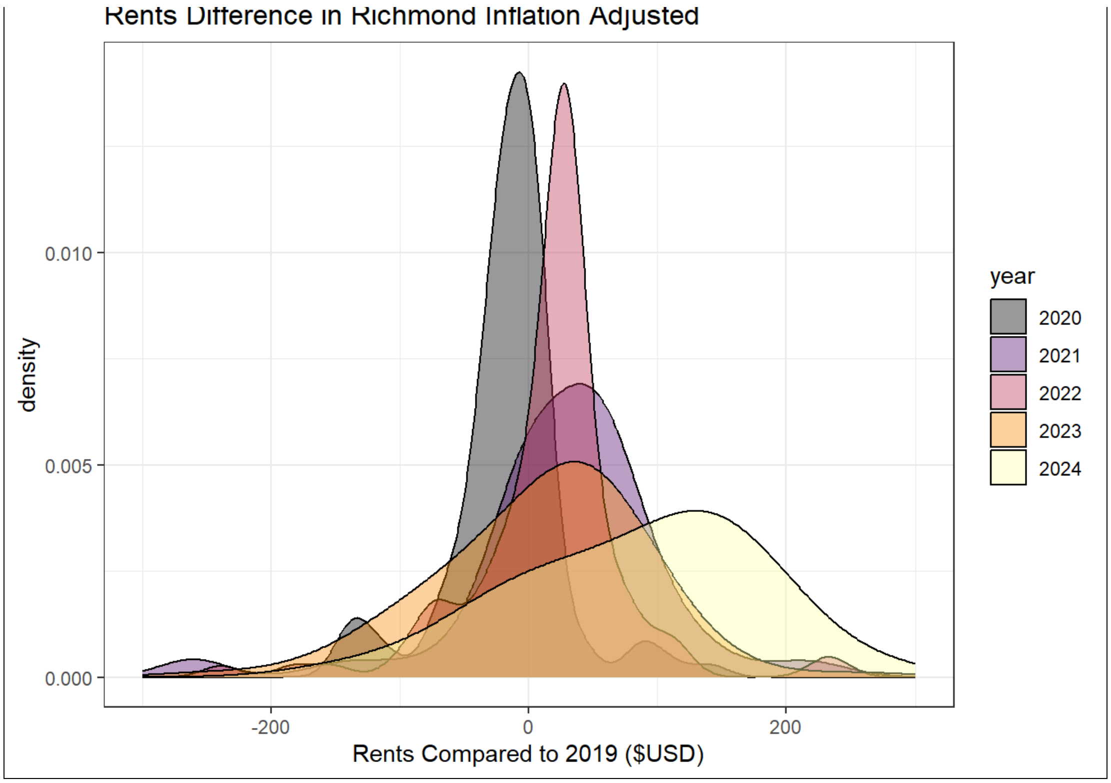


```
# Easily change the number of bedrooms
num_br <- "safmr_1br"

safmr_merged|>
  # pivot wider so that we can isolate the similar zip codes
  select(
    zip, area_code, area, state, year,
    all_of(num_br)
  )|>
  filter(area == "Richmond", state == "VA")|>
  pivot_wider(
    names_from = year,
    names_prefix = "y",
    values_from = c(num_br)
    )|>
  na.omit()|>
  # A lot of NAs in the wide table 131 to 109, lost 22

  # scale for inflation by multiplying the appropriate year rent value with
  # a inflation adjusted constant
  mutate(
    y2020 = y2020 * (1-0.0181),
    y2021 = y2021 * (1-0.0307),
    y2022 = y2022 * (1-0.0791),
    y2023 = y2023 * (1-0.1686),
    y2024 = y2024 * (1-0.2102)
    )|>

  # calculate the difference
  group_by(zip)|>
  summarize(
    diff19vs20 = y2020 - y2019,
    diff19vs21 = y2021 - y2019,
    diff19vs22 = y2022 - y2019,
    diff19vs23 = y2023 - y2019,
    diff19vs24 = y2024 - y2019,
  )|>
  pivot_longer(
    -c(zip),
    names_to = "type",
    values_to = "difference"
    )|>

  ggplot(aes(x = difference, group = type, fill = type))+
    geom_density(adjust=1.5, alpha=.4)+
    labs(
      title = "Rents Difference in Richmond Inflation Adjusted",
      x = "Rents Compared to 2019 ($USD)",
      fill = "year"
    )+
    scale_x_continuous(limits = c(-300, 300))+
    scale_fill_viridis_d(option = "inferno", labels = c('2020','2021','2022','2023','2024'))
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(num_br)
##
##   # Now:
##   data %>% select(all_of(num_br))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 18 rows containing non-finite values (`stat_density()`).
```





That… worked too well. Now it is easier to see that from the distribution that even when inflation adjusted, the rents are still significantly higher than 2019 rents

Question 6: Building a Heatmap with geometry values.

I have included an extra data set which includes geometric information of each zip code.

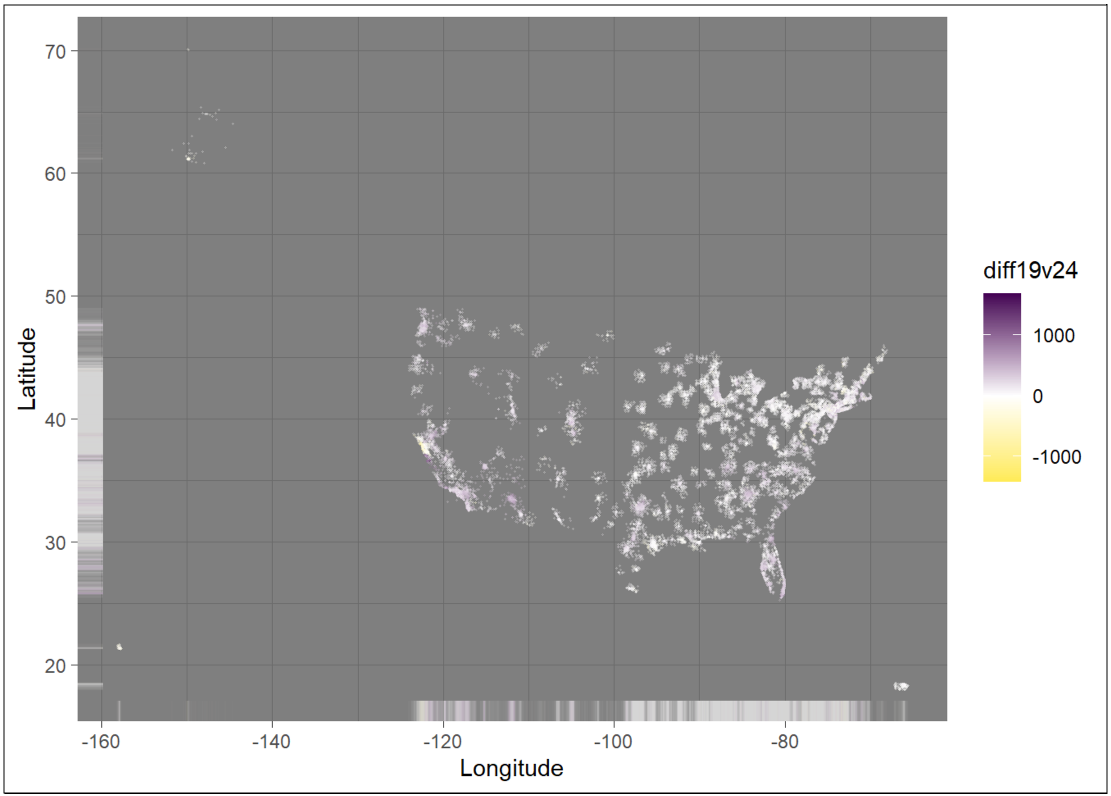
https://www.unitedstateszipcodes.org/zip-code-database/ (https://www.unitedstateszipcodes.org/zip-code-database/)

a. look at the `fmr_zip_co` file.
b. use `left_join` to join ‘fmr_zip_co’ to `safmr_merged`.

```
fmr_zip_co
```

```
## # A tibble: 42,735 x 15
##    zip   type     decommissioned primary_city acceptable_cities
##    <chr> <chr>             <dbl> <chr>        <chr>
##  1 00501 UNIQUE                0 Holtsville   <NA>
##  2 00544 UNIQUE                0 Holtsville   <NA>
##  3 00601 STANDARD              0 Adjuntas     <NA>
##  4 00602 STANDARD              0 Aguada       <NA>
##  5 00603 STANDARD              0 Aguadilla    Ramey
##  6 00604 PO BOX                0 Aguadilla    Ramey
##  7 00605 PO BOX                0 Aguadilla    <NA>
##  8 00606 STANDARD              0 Maricao      <NA>
##  9 00610 STANDARD              0 Anasco       <NA>
## 10 00611 PO BOX                0 Angeles      <NA>
## # i 42,725 more rows
## # i 10 more variables: unacceptable_cities <chr>, state <chr>,
## #   county <chr>, timezone <chr>, area_codes <chr>, world_region
<chr>,
## #   country <chr>, latitude <dbl>, longitude <dbl>,
## #   irs_estimated_population <dbl>
```

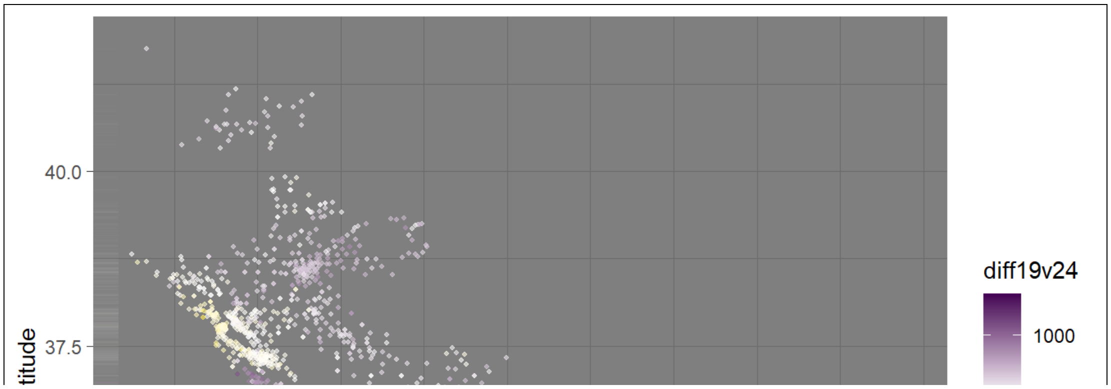
```
safmr_merged|>
  left_join(fmr_zip_co, by = c("zip", "state"))|>
  select(
    zip, area_code, state, area, county, latitude, longitude,
    safmr_0br, year
  )|>

  pivot_wider(
    names_from = year,
    names_prefix = "y",
    values_from = safmr_0br
  )|>
  na.omit()|>
  # A lot of NAs in the wide table 131 to 109, lost 22

  # scale for inflation by multiplying the appropriate year rent value with
  # a inflation adjusted constant
  mutate(
    y2020 = y2020 * (1-0.0181),
    y2021 = y2021 * (1-0.0307),
    y2022 = y2022 * (1-0.0791),
    y2023 = y2023 * (1-0.1686),
    y2024 = y2024 * (1-0.2102)
    )|>

  # calculate the difference
  mutate(
    diff19v20 = y2020 - y2019,
    diff19v21 = y2021 - y2019,
    diff19v22 = y2022 - y2019,
    diff19v23 = y2023 - y2019,
    diff19v24 = y2024 - y2019
    )|>
  select(
    latitude, longitude, diff19v20, diff19v21, diff19v22, diff19v23, diff19v24
  )|>
  ggplot(aes(x = longitude, y = latitude, color = diff19v24))+
    geom_point(alpha = 0.2, size = 0.2)+
    geom_rug(alpha = 0.01)+
    theme_dark()+
    scale_color_gradient2(
      low = "#FDE725FF",
      mid = "white",
      high = "#440154FF"
    )+
    labs(
      Title = "Rents Change across the US",
      x = "Longitude",
      y = "Latitude"
    )
```

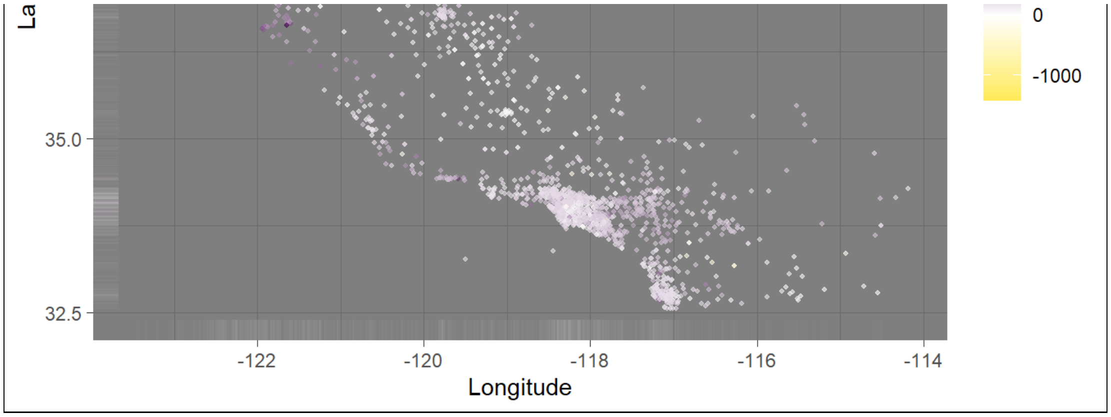


Observation 6b: Plotting the difference of each year at a time like this is not very effective. Maybe another statistic might be more appropriate for this plot. Something like rate of change would be easier as we can see a comparison of rate between each zip code, with a map!

```
# Let's see how California is fairing
var_state = "CA"

safmr_merged|>
  left_join(fmr_zip_co, by = c("zip", "state"))|>
  select(
    zip, area_code, state, area, county, latitude, longitude,
    safmr_0br, year
  )|>
  filter(state == all_of(var_state))|>
  pivot_wider(
    names_from = year,
    names_prefix = "y",
    values_from = safmr_0br
  )|>
  na.omit()|>
  # A lot of NAs in the wide table 131 to 109, lost 22

  # scale for inflation by multiplying the appropriate year rent va
          lue with
  # a inflation adjusted constant
  mutate(
    y2020 = y2020 * (1-0.0181),
    y2021 = y2021 * (1-0.0307),
    y2022 = y2022 * (1-0.0791),
    y2023 = y2023 * (1-0.1686),
    y2024 = y2024 * (1-0.2102)
    )|>

  # calculate the difference
  mutate(
    diff19v20 = y2020 - y2019,
    diff19v21 = y2021 - y2019,
    diff19v22 = y2022 - y2019,
    diff19v23 = y2023 - y2019,
    diff19v24 = y2024 - y2019
    )|>
  select(
    latitude, longitude, diff19v20, diff19v21, diff19v22, diff19v23, diff19v24
  )|>
  ggplot(aes(x = longitude, y = latitude, color = diff19v24))+
    geom_rug(alpha = 0.01)+
    geom_point(alpha = 0.6, size = 0.8)+
    theme_dark()+
    scale_color_gradient2(
      low = "#FDE725FF",
      mid = "white",
      high = "#440154FF"
    )+
    labs(
      Title = "Rents Change across the US",
      x = "Longitude",
      y = "Latitude",
      fill = "Amount $USD"
    )
```

```
## Warning: There was 1 warning in `filter()`.
## i In argument: `state == all_of(var_state)`.
## Caused by warning:
## ! Using `all_of()` outside of a selecting function was deprecated in
##   tidyselect 1.2.0.
## i See details at
##   <https://tidyselect.r-lib.org/reference/faq-selection-context.html>
```

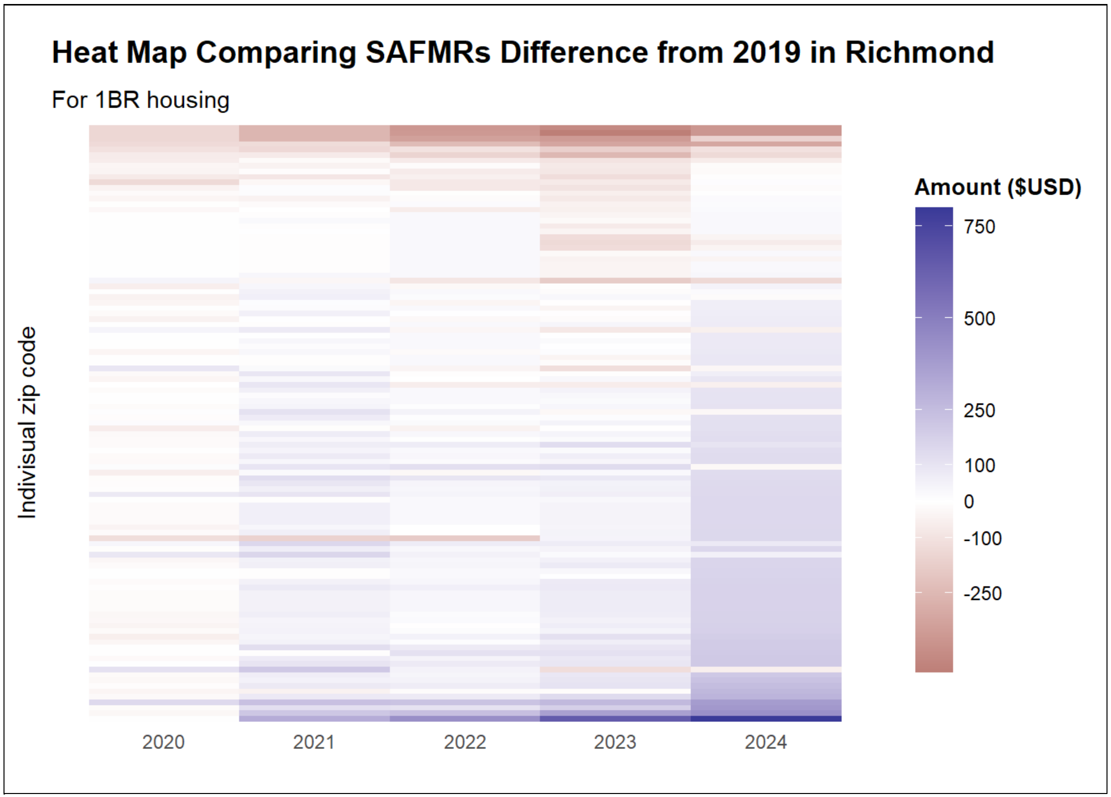




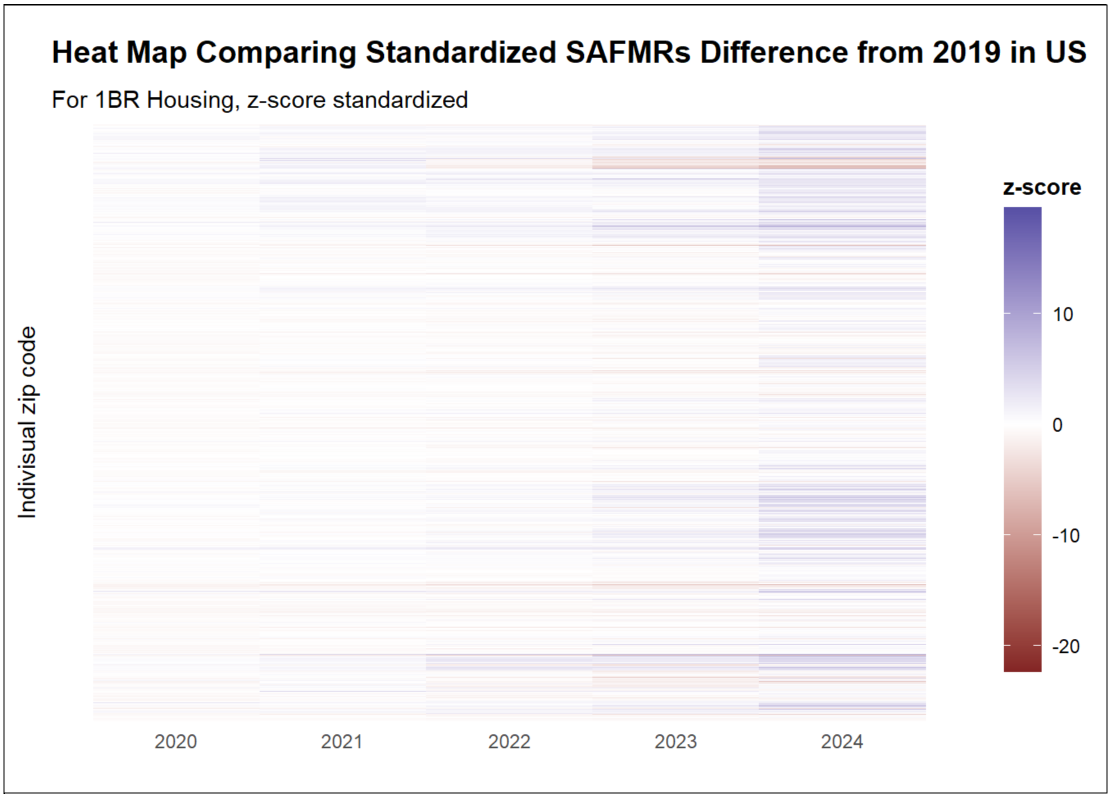
Question 7: Ok so that didn’t look as good as I thought it would, and it is also limited to showing only 1 year at a time.

But this a heatmap comparing the distribution of difference would be way better! a. Heat map with raw data b. Heat map with normalized data -

```
# Easily change the number of bedrooms
num_br <- "safmr_1br"

safmr_merged|>
  select(
    zip, area_code, state, area,
    num_br, year
  )|>
  filter(area == "Richmond", state == "VA")|>
  pivot_wider(
    names_from = year,
    names_prefix = "y",
    values_from = num_br
  )|>
  na.omit()|>
  # A lot of NAs in the wide table 131 to 109, lost 22

  # scale for inflation by multiplying the appropriate year rent value with
  # a inflation adjusted constant
  mutate(
    y2020 = y2020 * (1-0.0181),
    y2021 = y2021 * (1-0.0307),
    y2022 = y2022 * (1-0.0791),
    y2023 = y2023 * (1-0.1686),
    y2024 = y2024 * (1-0.2102)
    )|>

  # calculate the difference
  mutate(
    diff19v20 = y2020 - y2019,
    diff19v21 = y2021 - y2019,
    diff19v22 = y2022 - y2019,
    diff19v23 = y2023 - y2019,
    diff19v24 = y2024 - y2019
    )|>
  select(
    zip, diff19v20, diff19v21, diff19v22, diff19v23, diff19v24
  )|>

  # Prep data and heat map
  pivot_longer(
    -c(zip),
    names_to = 'year',
    values_to = 'value'
  )|>
  mutate(year = case_match(
    year,
    "diff19v20" ~ 2020,
    "diff19v21" ~ 2021,
    "diff19v22" ~ 2022,
    "diff19v23" ~ 2023,
    "diff19v24" ~ 2024
  ))|>
  arrange(desc(value))|>
  ggplot(aes(y = fct_inorder(zip), x = year, fill = value))+
    geom_tile()+
    scale_fill_gradient2(
      breaks =  c(-250, -100, 0, 100, 250, 500, 750),
      labels =  c(-250, -100, 0, 100, 250, 500, 750)
    )+
    theme_tufte()+
    theme(
      axis.text.y = element_blank(),
      axis.ticks.y = element_blank(),
      plot.title = element_text(face = "bold",
                                margin = margin(10, 0, 10, 0),
                                size = 14),
      legend.position = "right",
      legend.title = element_text(size = 10, face = "bold"),
      legend.key.height = unit(1.5,"cm")
      )+
    labs(
      title = "Heat Map Comparing SAFMRs Difference from 2019 in Richmond",
      subtitle = "For 1BR housing",
      x = "",
      y = "Indivisual zip code",
      fill = "Amount ($USD)"
    )
```



```
# Easily change the number of bedrooms
num_br <- "safmr_1br"

safmr_merged|>
  select(
    zip, area_code, state, area,
    num_br, year
  )|>

  pivot_wider(
    names_from = year,
    names_prefix = "y",
    values_from = num_br
  )|>
  na.omit()|>
  # A lot of NAs in the wide table 131 to 109, lost 22

  # scale for inflation by multiplying the appropriate year rent value with
  # a inflation adjusted constant
  mutate(
    y2020 = y2020 * (1-0.0181),
    y2021 = y2021 * (1-0.0307),
    y2022 = y2022 * (1-0.0791),
    y2023 = y2023 * (1-0.1686),
    y2024 = y2024 * (1-0.2102)
    )|>

  # calculate the difference
  mutate(
    diff19v20 = y2020 - y2019,
    diff19v21 = y2021 - y2019,
    diff19v22 = y2022 - y2019,
    diff19v23 = y2023 - y2019,
    diff19v24 = y2024 - y2019
    )|>
  select(
    zip, diff19v20, diff19v21, diff19v22, diff19v23, diff19v24
  )|>

  # z-score standardized data
  mutate(
    diff19v20 = (diff19v20 - mean(c(diff19v20, diff19v21, diff19v2
         2, diff19v23, diff19v24))) / sd(c(diff19v20, diff19v21, di
         ff19v22, diff19v23, diff19v24)),
    diff19v21 = (diff19v21 - mean(c(diff19v20, diff19v21, diff19v2
         2, diff19v23, diff19v24))) / sd(c(diff19v20, diff19v21, di
         ff19v22, diff19v23, diff19v24)),
    diff19v22 = (diff19v22 - mean(c(diff19v20, diff19v21, diff19v2
         2, diff19v23, diff19v24))) / sd(c(diff19v20, diff19v21, di
         ff19v22, diff19v23, diff19v24)),
    diff19v23 = (diff19v23 - mean(c(diff19v20, diff19v21, diff19v2
         2, diff19v23, diff19v24))) / sd(c(diff19v20, diff19v21, di
         ff19v22, diff19v23, diff19v24)),
    diff19v24 = (diff19v24 - mean(c(diff19v20, diff19v21, diff19v2
         2, diff19v23, diff19v24))) / sd(c(diff19v20, diff19v21, di
         ff19v22, diff19v23, diff19v24))
  )|>

  # Prep data and heat map
  pivot_longer(
    -c(zip),
    names_to = 'year',
    values_to = 'value'
  )|>
  mutate(year = case_match(
    year,
    "diff19v20" ~ 2020,
    "diff19v21" ~ 2021,
    "diff19v22" ~ 2022,
    "diff19v23" ~ 2023,
    "diff19v24" ~ 2024
  ))|>
  arrange(desc(value))|>
  ggplot(aes(y = zip, x = year, fill = value))+
    geom_tile()+
    scale_fill_gradient2()+
    theme_tufte()+
    theme(
      axis.text.y = element_blank(),
      axis.ticks.y = element_blank(),
      plot.title = element_text(face = "bold",
                                margin = margin(10, 0, 10, 0),
                                size = 14),
      legend.position = "right",
      legend.title = element_text(size = 10, face = "bold"),
      legend.key.height = unit(1.5,"cm")
      )+
    labs(
      title = "Heat Map Comparing Standardized SAFMRs Difference from 2019 in US",
      subtitle = "For 1BR Housing, z-score standardized",
      x = "",
      y = "Indivisual zip code",
      fill = "z-score"
    )
```



Observation 7b: You lose a lot of information by standardizing each sample separately with (x1 - x1hat) / sd1 This made the heat map plot less appropriate because now each sample is not on the same scale as each other.

FIXED: Calculate an estimate of the mean of difference of d2020 - d2024 *and not* the individual mean of d2020 only. The formula becomes:

x - xhat / sd where x is any observation of difference of any year xhat is the mean of all observations d2020 - d2024 sd is the sd of all observations d2020 - d2024