Project

Mehdi

2023-04-04

```
library(tidycensus)
```

```
## Warning: package 'tidycensus' was built under R version 4.2.3
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.8     v purrr   1.0.1
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
```

```
#Get variable names (just for reference, next block of code does not require this)
variables <- load_variables(2005, "acs1", cache = TRUE)
```

#Get all MA, PA, and IL data

```
#B25064_001 = Median gross rent
#B25071_001 = Median gross rent as a percentage of household income
#C25032_002 = Owner occupied housing units
#C25032_007 = Renter occupied housing units
#B25001_001 = Total housing units
#B01003_001 = Total population
#B25014_009 = 0.51 to 1.00 occupants per room
#B25014_010 = 1.01 to 1.50 occupants per room
#B25014_011 = 1.51 to 2.00 occupants per room
#B25014_012 = 2.01 or more occupants per room
```

```
#years <- 2005:2019
#years <- append(years, 2021)
#state_names <- c("Massachusetts", "Pennsylvania", "Illinois")
#variable_labels <- c("B25064_001", "B25071_001", "C25032_002", "C25032_007", "B25001_001", "B01003_001
#combinedDF <- NULL

#for (state_name in state_names) {
# for (variable_label in variable_labels) {
#    for (year in years) {
#      df <- get_acs(geography = "place", state = state_name, survey = "acs1", variables = variable_lab
#      df$year <- year
#      if (is.null(combinedDF)) {
#        combinedDF <- df
#      } else {
#        combinedDF <- rbind(combinedDF, df)
#      }
#    }
#  }

#}
```

#Get Massachusetts City Data

```
#variable names:
#B25064_001 = Median gross rent
#B25071_001 = Median gross rent as a percentage of household income
#C25032_002 = Owner occupied housing units
#C25032_007 = Renter occupied housing units
#B25001_001 = Total housing units
#B01003_001 = Total population
#B25014_009 = 0.51 to 1.00 occupants per room
#B25014_010 = 1.01 to 1.50 occupants per room
#B25014_011 = 1.51 to 2.00 occupants per room
#B25014_012 = 2.01 or more occupants per room

#sequnce of years where data is available (data not available for 2020)
years <- 2005:2019
years <- append(years, 2021)

#get median rents
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B25064_001"
  df$year <- year
  if (exists("median_rent_ma_cities")) {
    median_rent_ma_cities <- rbind(median_rent_ma_cities, df)
  } else {
    median_rent_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Warning: * You have not set a Census API key. Users without a key are limited to 500
## queries per day and may experience performance limitations.
```

```
## i For best results, get a Census API key at
## http://api.census.gov/data/key_signup.html and then supply the key to the
## `census_api_key()` function to use it throughout your tidycensus session.
## This warning is displayed once per session.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
#get median rent as percent of household income
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B25071_001"
  df$year <- year
  if (exists("median_rent_percent_income_ma_cities")) {
```

```
    median_rent_percent_income_ma_cities <- rbind(median_rent_percent_income_ma_cities, df)
  } else {
    median_rent_percent_income_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```r
#get number of owner occupied housing units
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "C25032_002"
  df$year <- year
  if (exists("owner_occupied_units_ma_cities")) {
    owner_occupied_units_ma_cities <- rbind(owner_occupied_units_ma_cities, df)
  } else {
    owner_occupied_units_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
```

```
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2021 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```
#get number of renter occupied housing units
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "C25032_007"
  df$year <- year
  if (exists("renter_occupied_units_ma_cities")) {
    renter_occupied_units_ma_cities <- rbind(renter_occupied_units_ma_cities, df)
  } else {
    renter_occupied_units_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2006 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2007 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2008 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2009 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2010 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2011 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2012 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2013 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2014 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2015 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2016 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2017 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```r
#get total number of housing units
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B25001_001"
  df$year <- year
  if (exists("total_housing_units_ma_cities")) {
    total_housing_units_ma_cities <- rbind(total_housing_units_ma_cities, df)
  } else {
    total_housing_units_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
```

```
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2017 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2018 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2019 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2021 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```r
#get total population
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B01003_001"
  df$year <- year
  if (exists("total_population_ma_cities")) {
    total_population_ma_cities <- rbind(total_population_ma_cities, df)
  } else {
    total_population_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2006 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2007 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2008 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2009 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2010 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2011 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2012 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2013 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.

## Getting data from the 2014 1-year ACS

## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```r
#get occupants per room 0.51 to 1.00 (renters)
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B25014_009"
  df$year <- year
  if (exists("occupants_051_to_100_per_room_ma_cities")) {
    occupants_051_to_100_per_room_ma_cities <- rbind(occupants_051_to_100_per_room_ma_cities, df)
  } else {
    occupants_051_to_100_per_room_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
```

```
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```r
#get occupants per room 1.01 to 1.50 (renters)
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B25014_010"
  df$year <- year
  if (exists("occupants_101_to_150_per_room_ma_cities")) {
    occupants_101_to_150_per_room_ma_cities <- rbind(occupants_101_to_150_per_room_ma_cities, df)
  } else {
    occupants_101_to_150_per_room_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```r
#get occupants per room 1.51 to 1.00 (renters)
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B25014_011"
  df$year <- year
  if (exists("occupants_151_to_200_per_room_ma_cities")) {
    occupants_151_to_200_per_room_ma_cities <- rbind(occupants_151_to_200_per_room_ma_cities, df)
  } else {
    occupants_151_to_200_per_room_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
```

```
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

```r
#get occupants per room 2.01 or more (renters)
for (year in years) {
  df <- get_acs(geography = "place", state = "Massachusetts", survey = "acs1", variables = "B25014_012"
  df$year <- year
  if (exists("occupants_201_per_room_ma_cities")) {
    occupants_201_per_room_ma_cities <- rbind(occupants_201_per_room_ma_cities, df)
  } else {
    occupants_201_per_room_ma_cities <- df
  }
}
```

```
## Getting data from the 2005 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2006 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2007 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2008 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

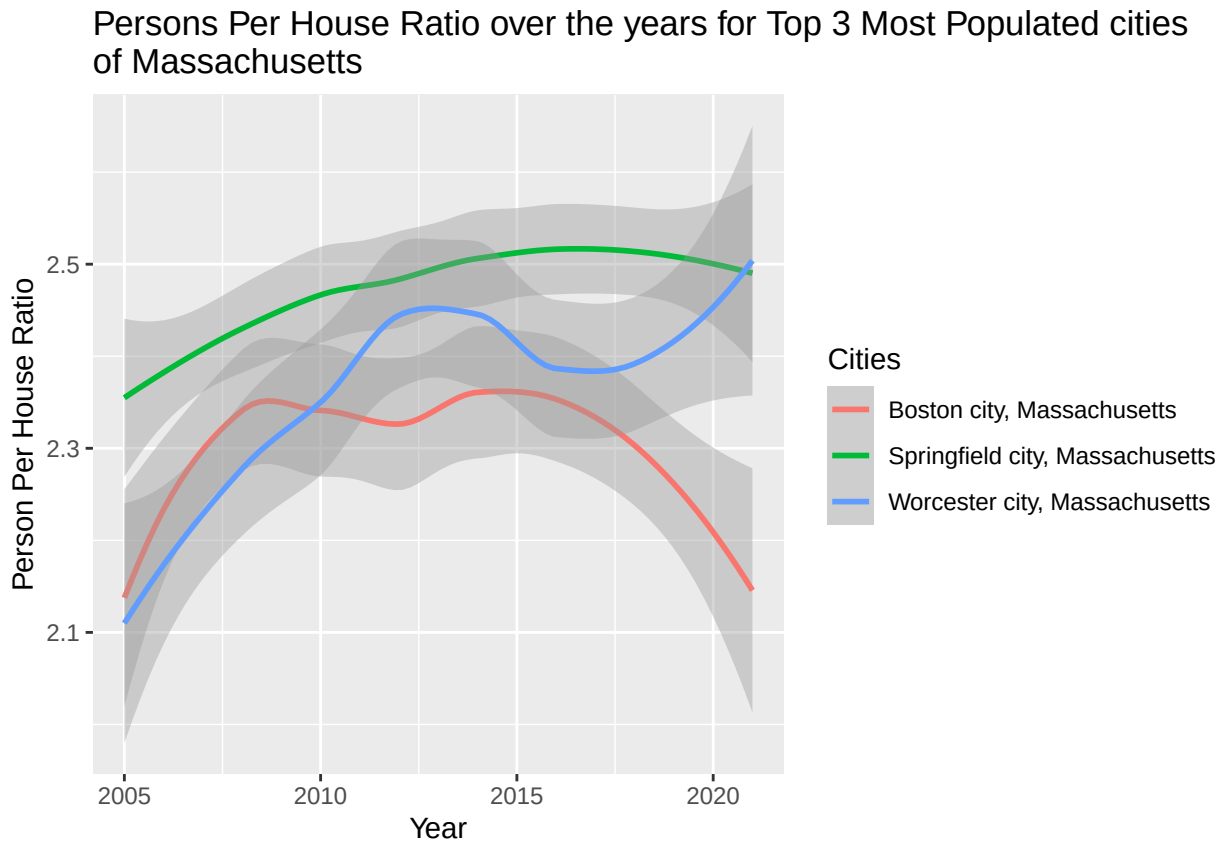
```
## Getting data from the 2009 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2010 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2011 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2012 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2013 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2014 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2015 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2016 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2017 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2018 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2019 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
## Getting data from the 2021 1-year ACS
## The 1-year ACS provides data for geographies with populations of 65,000 and greater.
```

VISUALIZATION 1:

```
popHousing<-inner_join(total_housing_units_ma_cities, total_population_ma_cities
                       , by=c("NAME","year"))
popHousing<-popHousing%>%select(NAME,estimate.x,estimate.y,year)
colnames(popHousing)[2] ="Housing_Units"
colnames(popHousing)[3] ="Population"


top3CitiesHPop<-popHousing%>%filter(NAME=="Boston city, Massachusetts" |
                                    NAME=="Worcester city, Massachusetts" |
                                    NAME=="Springfield city, Massachusetts")

ggplot(data=top3CitiesHPop, aes(x=year, y=Population/Housing_Units, group=NAME,color=NAME)) +
  labs(title="Persons Per House Ratio over the years for Top 3 Most Populated cities
of Massachusetts",
```

```
       x="Year", y="Person Per House Ratio",color="Cities") +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



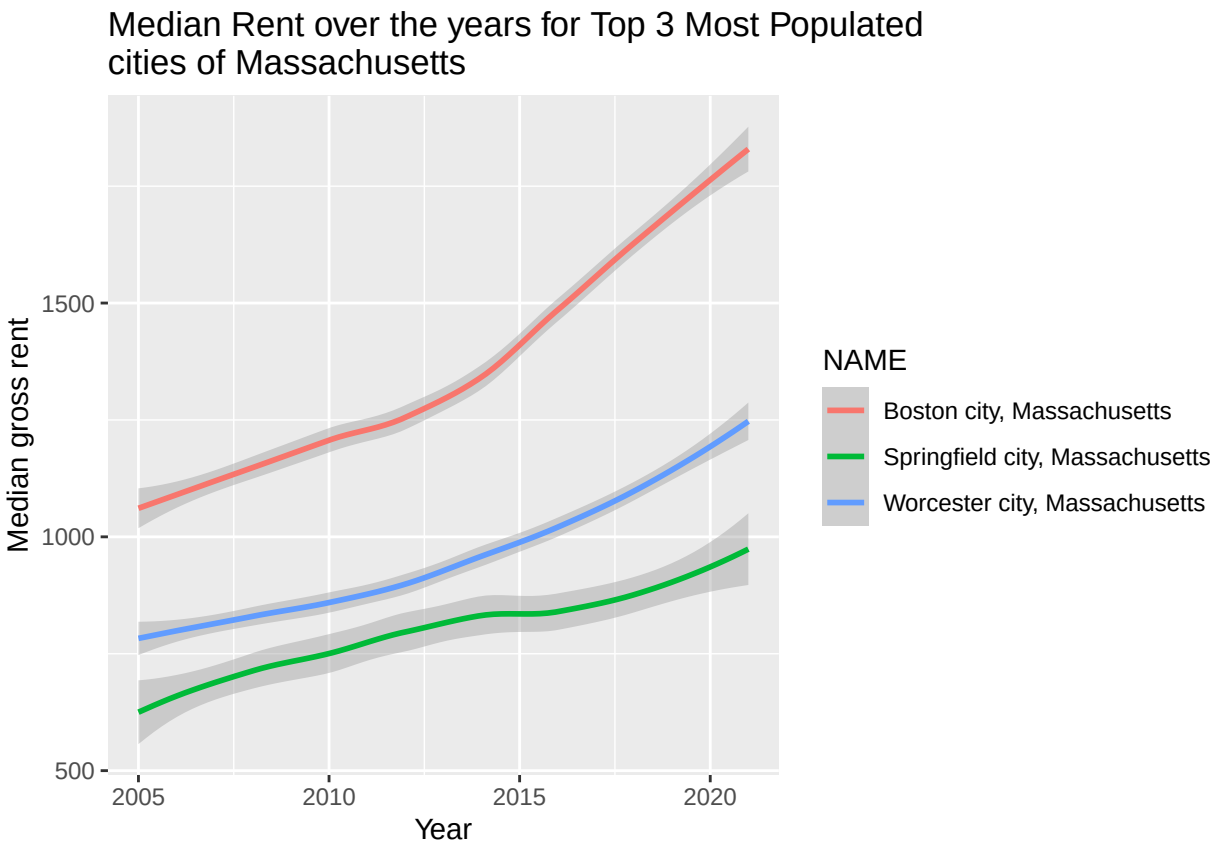
OBSERVATIONS:

According to the website https://www.massachusetts-demographics.com/cities_by_population the top 3 populated cities of Massachusetts are Boston, Worcester and Springfield. Above visualization shows the Person Per House ratio over the years in these cities. As we can see Boston's PersonPerHouse ratio is decreasing while that of Springfield is almost stable and Worcester's PersonPerHouse ratio is increasing very fast.

VISUALIZATION 2:

```
rent<-median_rent_ma_cities%>%select(NAME,estimate,year)

top3CitiesRent<-rent%>%filter(NAME=="Boston city, Massachusetts" |
                                NAME=="Worcester city, Massachusetts" |
                                NAME=="Springfield city, Massachusetts")

ggplot(data=top3CitiesRent, aes(x=year, y=estimate, group=NAME,color=NAME)) +
  labs(title="Median Rent over the years for Top 3 Most Populated
cities of Massachusetts",
       x="Year", y="Median gross rent") +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



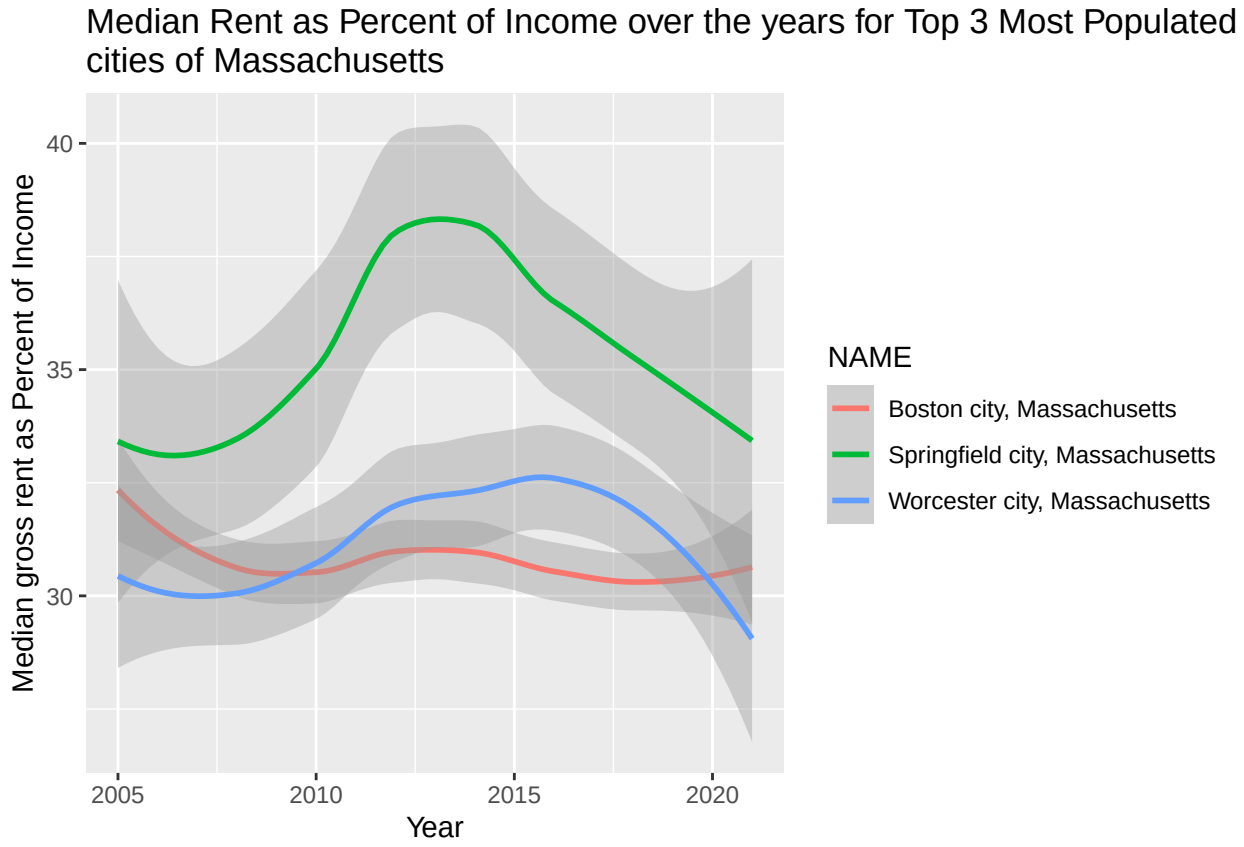
OBSERVATIONS:

For the same top 3 populated cities, we have visualized the median rent over the years. The median rent in Boston seems to have increased rapidly, while it is not that rapid increase in Worcester and Springfield.

VISUALIZATION 3:

```
rentIncome<-median_rent_percent_income_ma_cities%>%select(NAME,estimate,year)

top3CitiesRentIncome<-rentIncome%>%filter(NAME=="Boston city, Massachusetts" |
                                  NAME=="Worcester city, Massachusetts" |
                                  NAME=="Springfield city, Massachusetts")

ggplot(data=top3CitiesRentIncome, aes(x=year, y=estimate, group=NAME,color=NAME)) +
  labs(title="Median Rent as Percent of Income over the years for Top 3 Most Populated
cities of Massachusetts",
       x="Year", y="Median gross rent as Percent of Income") +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



OBSERVATIONS:

The rent as percent of income is higher i.e. affordability is lower for Springfield as compared to Boston and Worcester. Rent as percent of income initially increased for Springfield and Worcester and now decreasing meaning that affordability is now increasing for these cities. For Boston it is not changing that much.

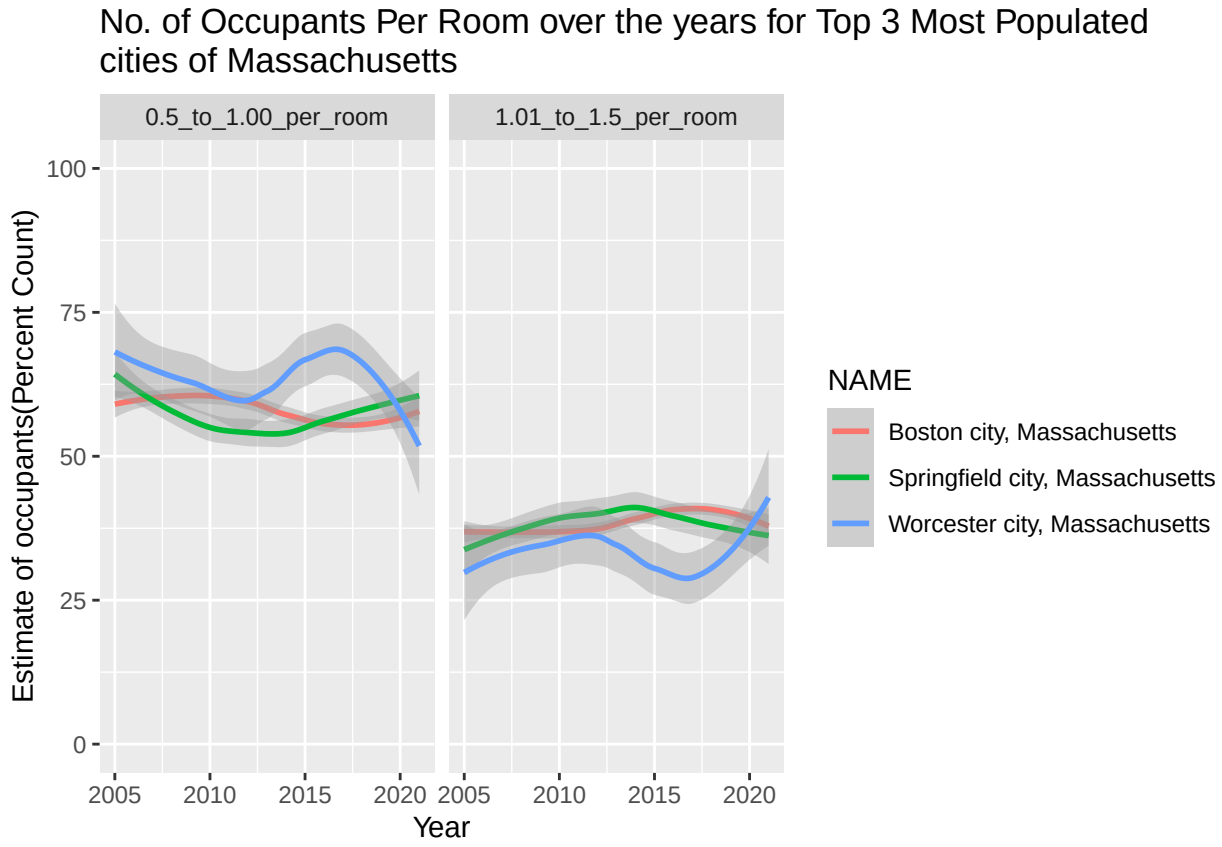
VISUALIZATION 4:

```
range1<-occupants_051_to_100_per_room_ma_cities%>%select(NAME,estimate,year)
range2<-occupants_101_to_150_per_room_ma_cities%>%select(NAME,estimate,year)
range3<-occupants_151_to_200_per_room_ma_cities%>%select(NAME,estimate,year)
range4<-occupants_201_per_room_ma_cities%>%select(NAME,estimate,year)


colnames(range1)[2] ="0.5_to_1.00_per_room"
colnames(range2)[2] ="1.01_to_1.5_per_room"
colnames(range3)[2] ="1.51_to_2.0_per_room"
colnames(range4)[2] ="2.01_or_more_per_room"


rangeMerge<-inner_join(range1, range2
                       , by=c("NAME","year"))
rangeMerge<-inner_join(rangeMerge,range3
                       , by=c("NAME","year"))
rangeMerge<-inner_join(rangeMerge,range4
```

```
                        , by=c("NAME","year"))


top3CitiesRangeMerge<-rangeMerge%>%filter(NAME=="Boston city, Massachusetts" |
                                    NAME=="Worcester city, Massachusetts" |
                                    NAME=="Springfield city, Massachusetts")

top3CitiesRangeMerge<-top3CitiesRangeMerge%>%mutate(totalPopulation = `0.5_to_1.00_per_room`+
                                  `1.01_to_1.5_per_room`+
                                  `1.51_to_2.0_per_room`+
                                  `2.01_or_more_per_room`)

top3CitiesRangeMerge$`0.5_to_1.00_per_room`<-
    top3CitiesRangeMerge$`0.5_to_1.00_per_room`/
top3CitiesRangeMerge$totalPopulation*100

top3CitiesRangeMerge$`1.01_to_1.5_per_room`<-
    top3CitiesRangeMerge$`1.01_to_1.5_per_room`/
  top3CitiesRangeMerge$totalPopulation*100

top3CitiesRangeMerge$`1.51_to_2.0_per_room`<-
    top3CitiesRangeMerge$`1.51_to_2.0_per_room`/
  top3CitiesRangeMerge$totalPopulation*100

top3CitiesRangeMerge$`2.01_or_more_per_room`<-
    top3CitiesRangeMerge$`2.01_or_more_per_room`/
  top3CitiesRangeMerge$totalPopulation*100



top3CitiesRangeMerge<-pivot_longer(top3CitiesRangeMerge,
                                   cols=c(`0.5_to_1.00_per_room`,
                                          `1.01_to_1.5_per_room`,
                                          `1.51_to_2.0_per_room`,
                                          `2.01_or_more_per_room`),
             names_to = "range", values_to = "estimate")

ggplot(data=top3CitiesRangeMerge%>%filter(range=="0.5_to_1.00_per_room" |
                                            range=="1.01_to_1.5_per_room" ),
                                            aes(x=year, y=estimate,
                                            group=NAME,color=NAME)) +
  coord_cartesian(ylim=c(0,100))+
  facet_wrap(~range)+
  labs(title="No. of Occupants Per Room over the years for Top 3 Most Populated
cities of Massachusetts",
       x="Year", y="Estimate of occupants(Percent Count)") +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

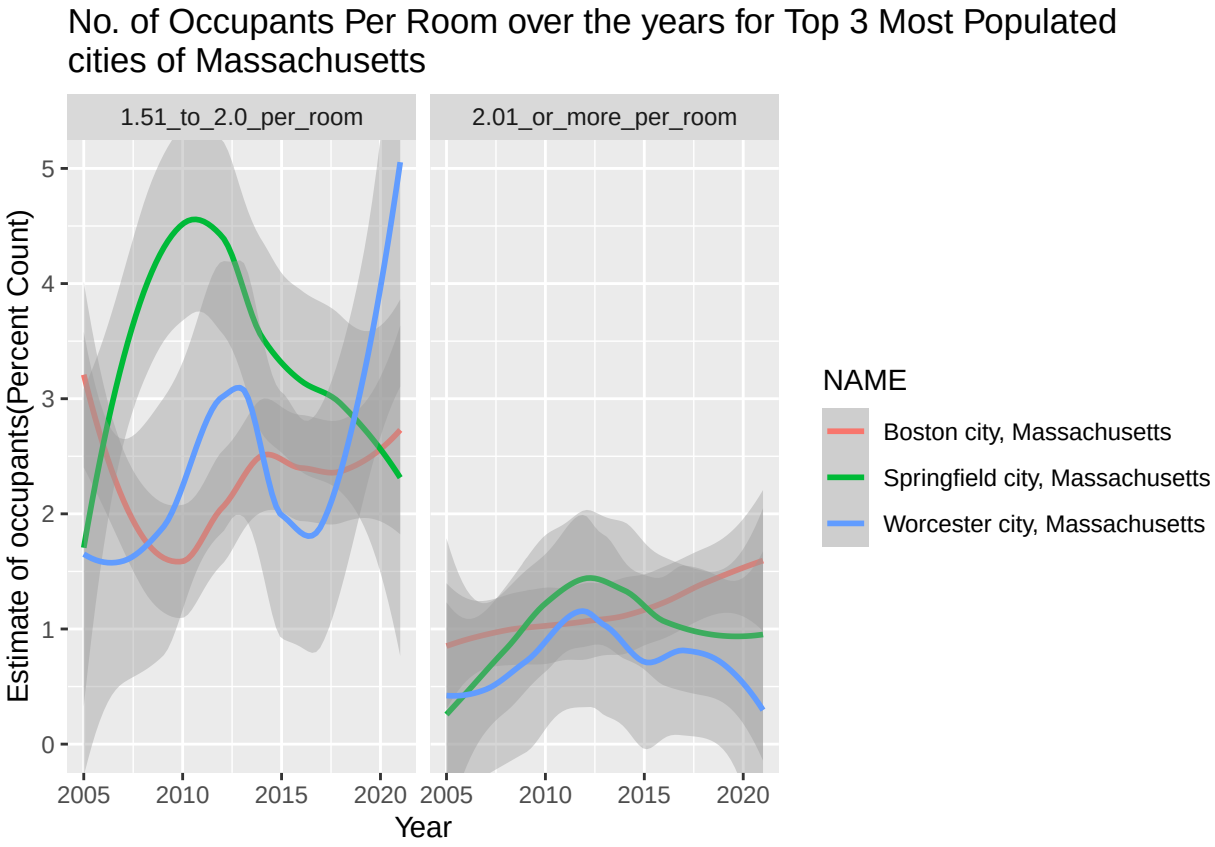
```
## Warning: Removed 4 rows containing non-finite values (`stat_smooth()`).
```




```
ggplot(data=top3CitiesRangeMerge%>%filter(range=="1.51_to_2.0_per_room" |
                                            range=="2.01_or_more_per_room" ),
                                             aes(x=year, y=estimate,
                                             group=NAME,color=NAME)) +
  coord_cartesian(ylim=c(0,5))+
  facet_wrap(~range)+
  labs(title="No. of Occupants Per Room over the years for Top 3 Most Populated
cities of Massachusetts",
       x="Year", y="Estimate of occupants(Percent Count)") +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_smooth()`).
```



OBSERVATIONS:

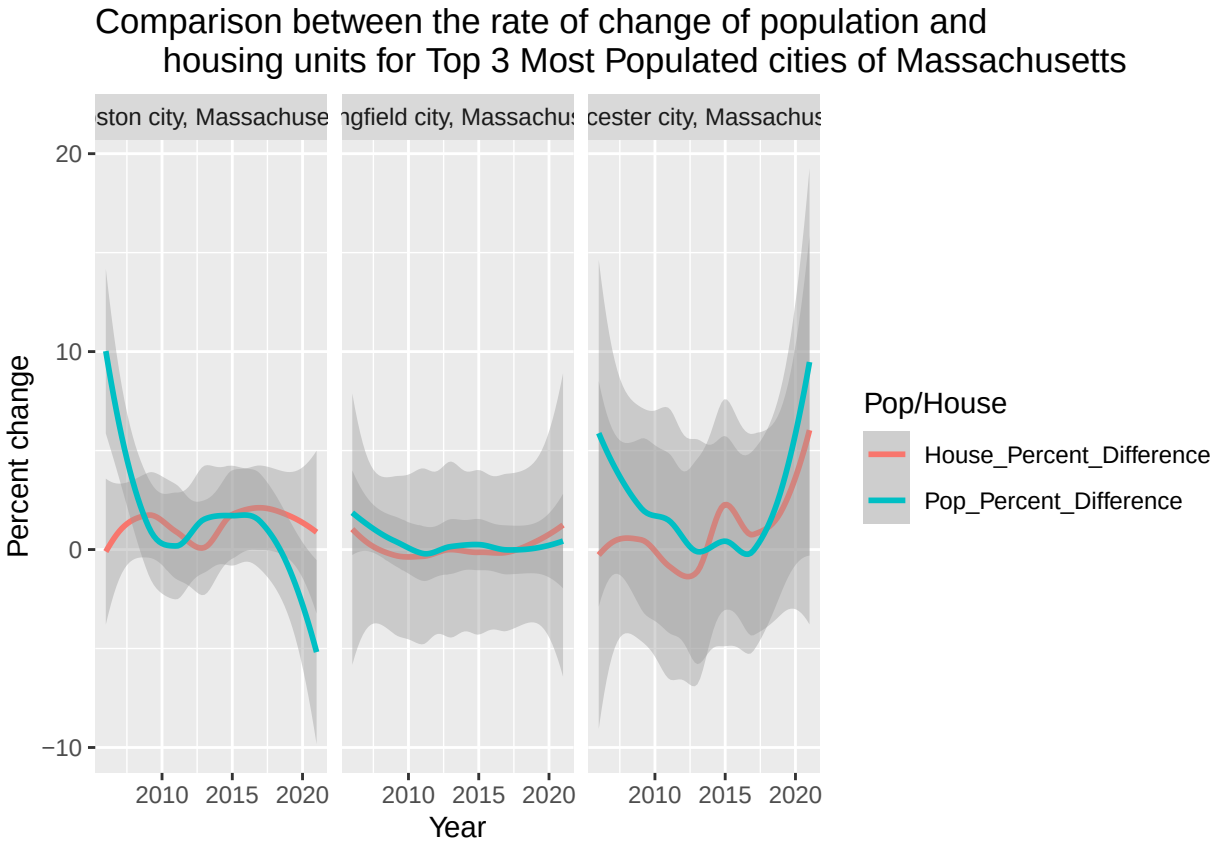
The percent count for estimate of 1.51 or more occupants per room is less than 5% for all three cities which is negligible compared to the percent count for estimates less than 1.51 occupants per room. The percent count for estimate of 0.5 to 1.0 occupants per room is more than 50% for all three cities and that of 1.01 to 1.5 occupants per room is between 25% to 40%. For Worcester city, the percent count for estimate of 0.5 to 1.0 occupant per room is recently showing a decreasing trend, whereas it is increasing for Boston and Springfield. An opposite trend is observed for estimate of 1.01 to 1.5 occupants per room.

VISUALIZATION 5:

```
pop<-total_population_ma_cities%>%select(NAME,estimate,year)

house<-total_housing_units_ma_cities%>%select(NAME,estimate,year)


pop_diff <-
  pop%>%group_by(NAME)%>%
  mutate(Pop_Percent_Difference = ((estimate - lag(estimate))/lag(estimate))*100)%>%
  na.omit()

pop_diff<-pop_diff%>%filter(NAME=="Boston city, Massachusetts" |
                                  NAME=="Worcester city, Massachusetts" |
                                  NAME=="Springfield city, Massachusetts")
```

```r
house_diff <-
  house%>%group_by(NAME)%>%
  mutate(House_Percent_Difference = ((estimate - lag(estimate))/lag(estimate))*100)%>%
  na.omit()

house_diff<-house_diff%>%filter(NAME=="Boston city, Massachusetts" |
                                  NAME=="Worcester city, Massachusetts" |
                                  NAME=="Springfield city, Massachusetts")



pop_house_diff<-inner_join(pop_diff%>%select(NAME,year,Pop_Percent_Difference),
                           house_diff%>%select(NAME,year,House_Percent_Difference)
                       , by=c("NAME","year"))


pop_house_diff<-pivot_longer(pop_house_diff,
                                   cols=c(`Pop_Percent_Difference`,
                                          `House_Percent_Difference`),
             names_to = "Pop/House", values_to = "estimate")

ggplot(data=pop_house_diff, aes(x=year, y=estimate,color=`Pop/House`))+
  facet_wrap(~NAME)+
  labs(title="Comparison between the rate of change of population and
       housing units for Top 3 Most Populated cities of Massachusetts",
       x="Year", y="Percent change")+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



OBSERVATIONS:

It seems that rate of change of population is very similar to that of number of houses. For Boston it is decreasing, For Springfield it is almost constant and for Worcester it is increasing.