# 27. 2010-2020 Race Proportion Differences by City

Daniel Zoleikhaeian

2023-04-19

## knit set-up

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(tidy = TRUE)
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

## Dependencies

```
library(plyr)
library(dplyr)
```

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

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## 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(readxl)
```

## Datasets

```r
dat_2010 <- read_xlsx("C:\\Users\\danie\\Documents\\Joshi Lab Materials\\3 Studies Dataset\\Dataset Merg
dat_2020 <- read_xlsx("C:\\Users\\danie\\Documents\\Joshi Lab Materials\\3 Studies Dataset\\Dataset Merg
View(dat_2010)
View(dat_2020)

dat_2010_races <- dat_2010[, 2:8]
dat_2020_races <- dat_2020[, 2:8]

rownames(dat_2010_races) <- dat_2010$City
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```r
rownames(dat_2020_races) <- dat_2020$City
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```r
View(dat_2010_races)
```

# Making proportions

```r
# Making proportions
prop_2010 <- as.data.frame(t(apply(dat_2010_races, MARGIN = 1,
    function(x) {
        x/sum(x)
    }, simplify = T)))

# Making a city column
prop_2010$City <- rownames(prop_2010)
prop_2010 <- prop_2010[, c(ncol(prop_2010), 1:(ncol(prop_2010) -
    1))]
rownames(prop_2010) <- NULL

# reformatting the City vector
prop_2010$City <- lapply(prop_2010$City, function(x) {
    substr(x, 1, unlist(gregexpr(",", x))[1] - 1)
})

# Adding a Year column
prop_2010$Year <- rep(2010, nrow(prop_2010))

prop_2020 <- as.data.frame(t(apply(dat_2020_races, MARGIN = 1,
    function(x) {
        x/sum(x)
    }, simplify = T)))

prop_2020$City <- rownames(prop_2020)
prop_2020 <- prop_2020[, c(ncol(prop_2020), 1:(ncol(prop_2020) -
    1))]
rownames(prop_2020) <- NULL
```

```r
prop_2020$City <- lapply(prop_2020$City, function(x) {
    substr(x, 1, unlist(gregexpr(",", x))[1] - 1)
})

prop_2020$Year <- rep(2020, nrow(prop_2020))

# stacking the dataframes on top of each other
prop_2010_2020 <- rbind(prop_2010, prop_2020)

# checking if the rows are equal
nrow(prop_2010_2020) == (nrow(prop_2010) + nrow(prop_2020))
```

```
## [1] TRUE
```

```r
# standardizing acronyms
colnames(prop_2010_2020)[colnames(prop_2010_2020) == "NA"] <- "AE"
colnames(prop_2010_2020)[colnames(prop_2010_2020) == "PI"] <- "NH"
```

## Visualization 1: Plots of the proportions in 2010 and 2020

```r
# Plots for each race Lines for each city

prop_2010_2020$City <- factor(as.character(prop_2010_2020$City))

race_acrs <- c("AA", "AS", "CA", "AE", "NH", "MR", "OT")

plot_race_cities <- function(race, data_in) {
    ggplot(data_in, aes_string(x = "Year", y = race, col = "City")) +
        geom_point() + ylab("Proportion") + geom_smooth(method = lm,
        se = FALSE) + ggtitle(paste("Race Proportions for", race))
}

plot_collection <- lapply(race_acrs, plot_race_cities, data_in = prop_2010_2020)
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation ideoms with `aes()`
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
# plot_collection
```

Conclusions: * Race proportions not stable over time * Shifts generally less than 0.05 for most cities

# Hypothesis Testing: Proportion Difference between 2010 and 2020

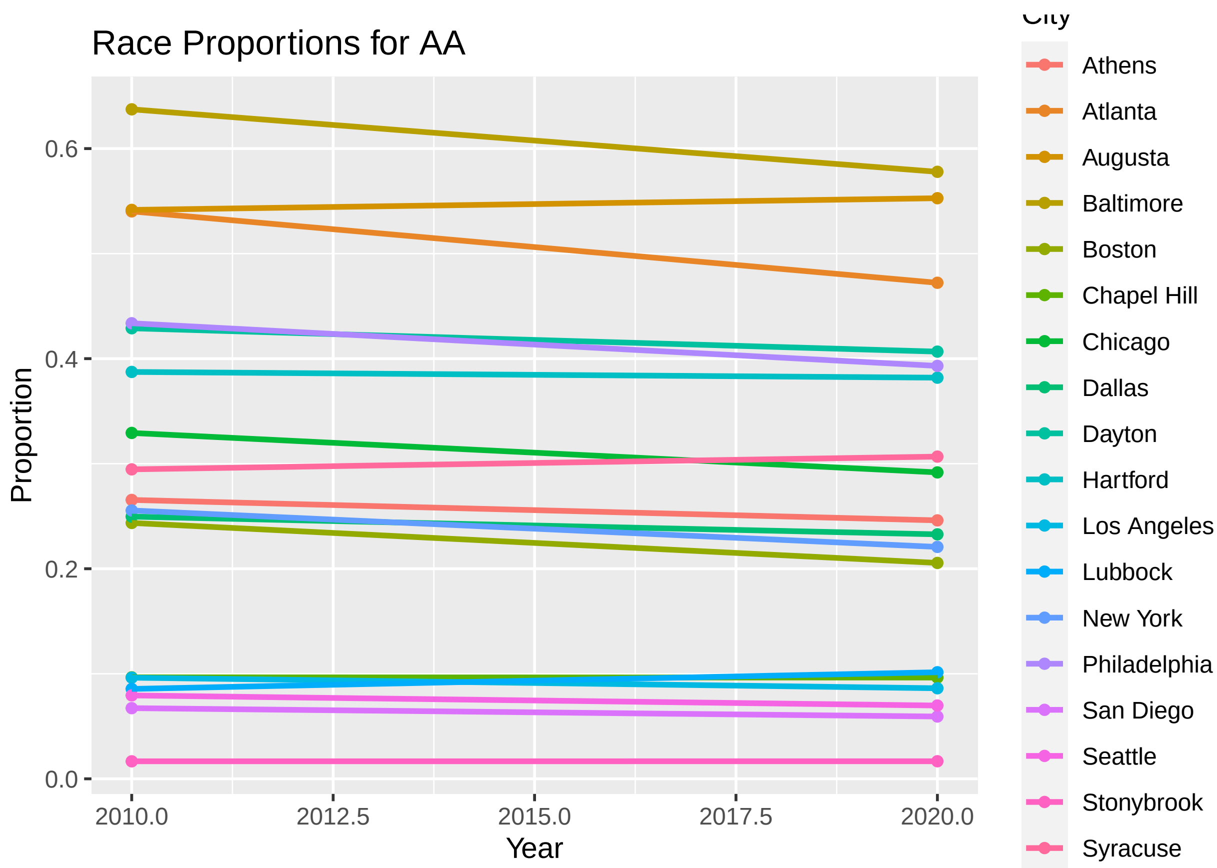
## Calculating proportion differences between 2010 and 2020

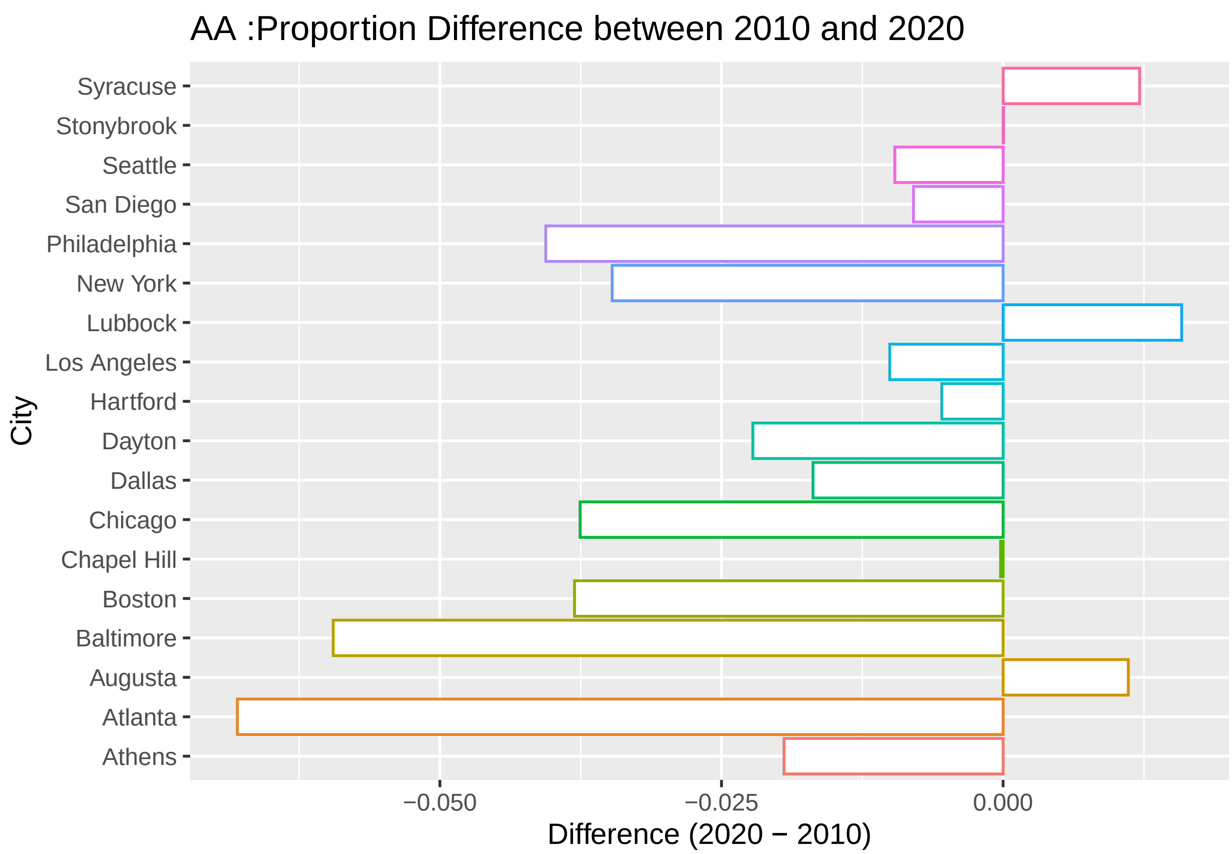
```
# Calculated as proportion in 2020 - proportion in 2010

prop_difs <- prop_2020[, 2:8] - prop_2010[, 2:8]
prop_difs$City <- prop_2020$City

# standardizing acronyms
colnames(prop_difs)[colnames(prop_difs) == "NA"] <- "AE"
colnames(prop_difs)[colnames(prop_difs) == "PI"] <- "NH"

# re-ordering columns
prop_difs <- prop_difs[, c("City", race_acrs)]

# making sub-dataframes for AA, AS, and CA
AA_df <- prop_difs[, c("City", "AA")]
AA_df$p_val <- rep(-1, nrow(AA_df))

AS_df <- prop_difs[, c("City", "AS")]
AS_df$p_val <- rep(-1, nrow(AS_df))

CA_df <- prop_difs[, c("City", "CA")]
CA_df$p_val <- rep(-1, nrow(CA_df))
```

## Visualization 2: Proportion Differences of Races across Cities

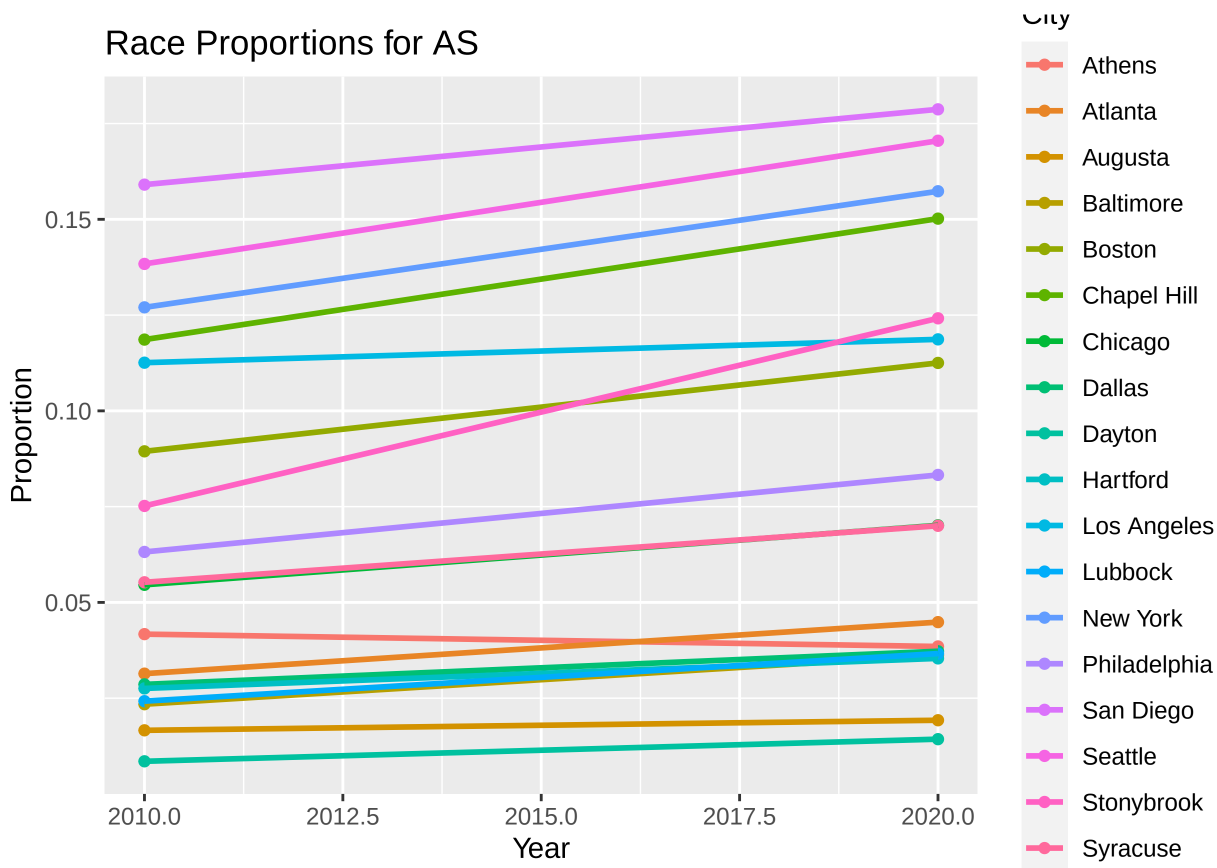
```
bar_plot_race_cities <- function(race_acr, data_in) {
    ggplot(data = data_in, aes_string(x = "City", y = race_acr,
        color = "City")) + geom_bar(stat = "identity", fill = "white") +
        ylab("Difference (2020 - 2010)") + ggtitle(paste(race_acr,
        ":Proportion Difference between 2010 and 2020")) + xlab("City") +
        coord_flip() + theme(legend.position = "none")
}

prop_difs$City <- as.factor(unlist(prop_difs$City))

plot_collection2 <- lapply(race_acrs, bar_plot_race_cities, data_in = prop_difs)

# plot_collection2

## Running plot collections side by side
for (i in 1:7) {
    print(plot_collection[[i]])
    print(plot_collection2[[i]])
}
```

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

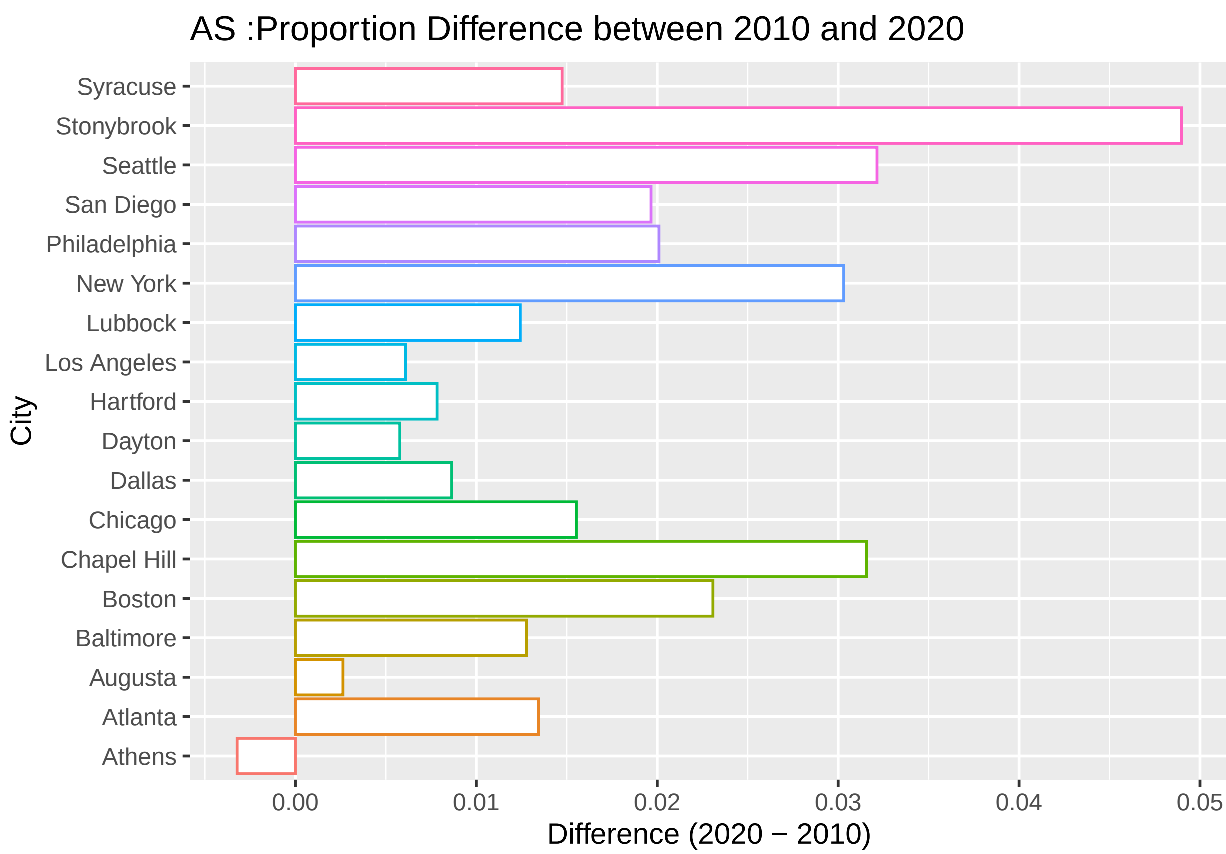

Race Proportions for AA
Proportion
0.0
0.2
0.4
0.6
2010.0
2012.5
2015.0
2017.5
2020.0
Year
City
Athens
Atlanta
Augusta
Baltimore
Boston
Chapel Hill
Chicago
Dallas
Dayton
Hartford
Los Angeles
Lubbock
New York
Philadelphia
San Diego
Seattle
Stonybrook
Syracuse



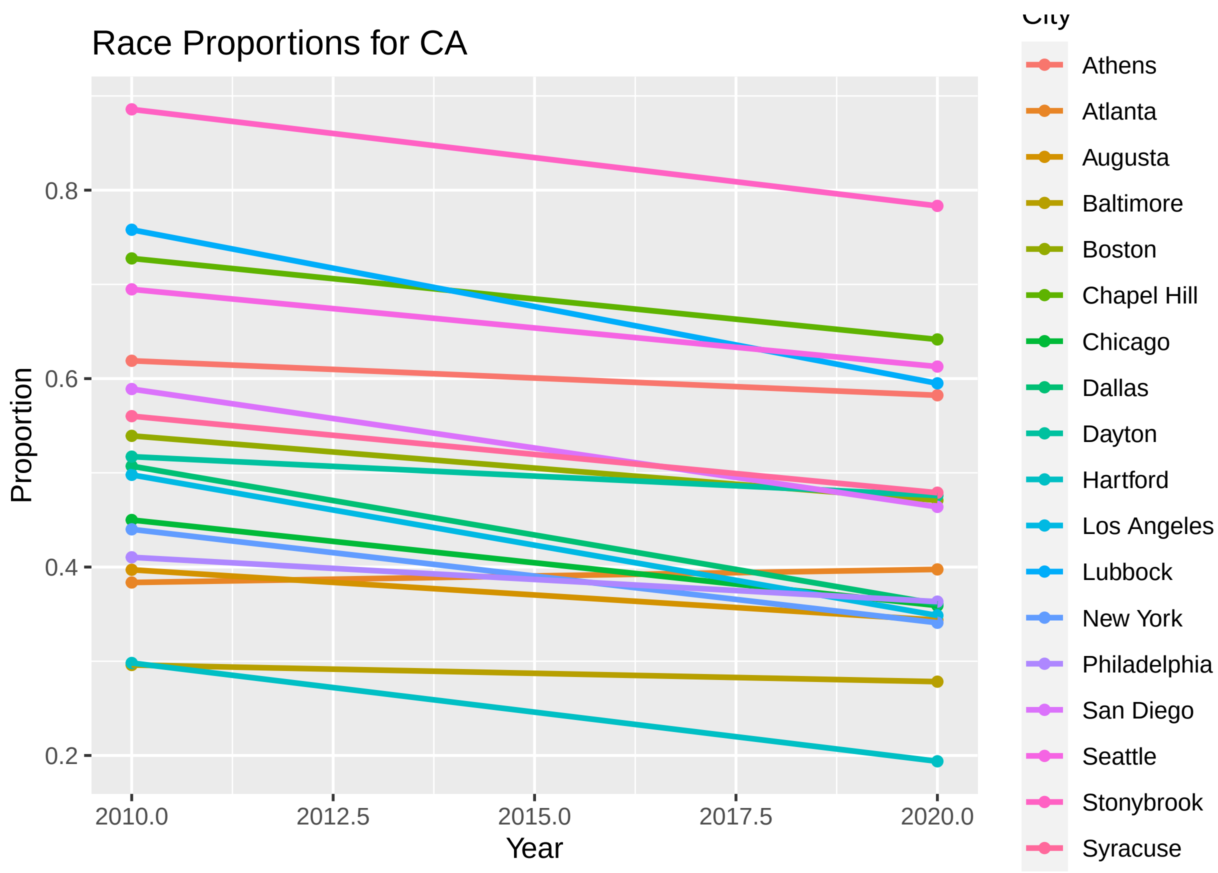

```
## 'geom_smooth()' using formula = 'y ~ x'
```

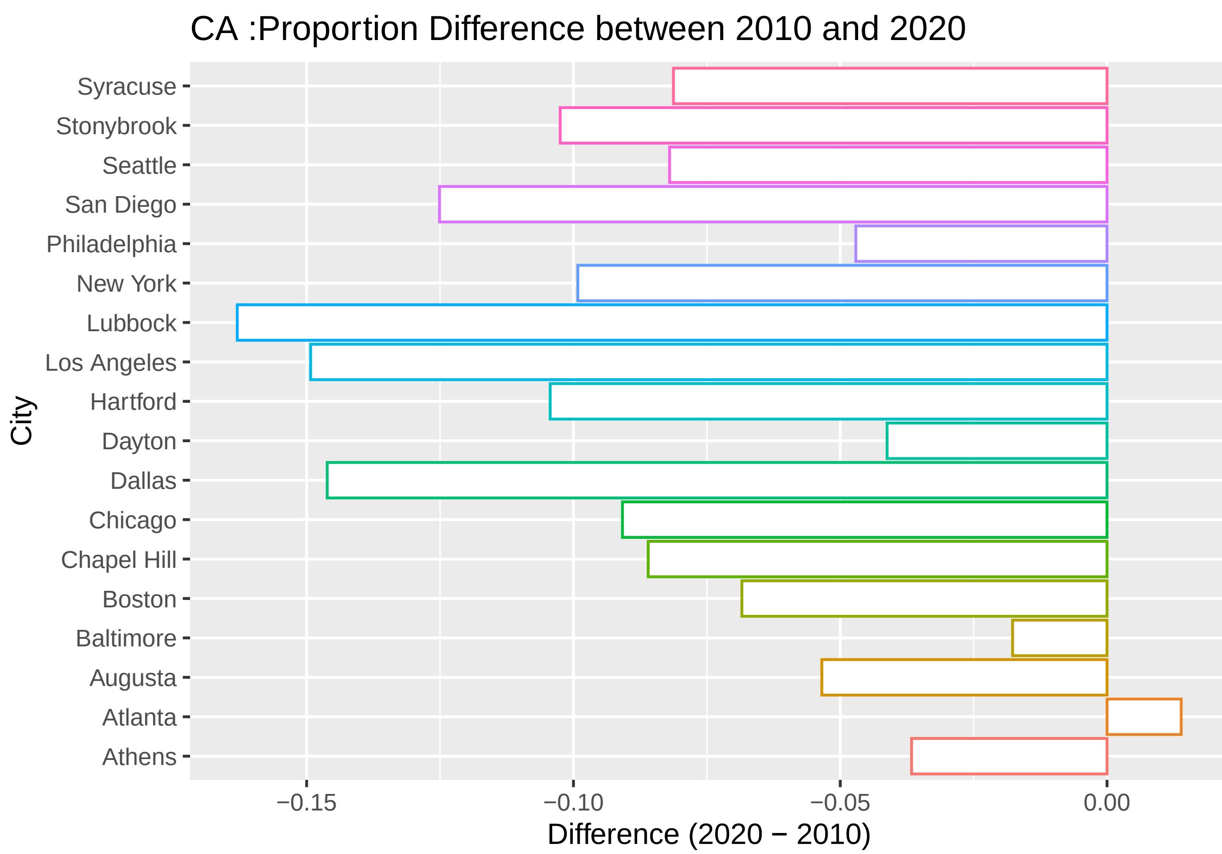


Race Proportions for AS
City
Proportion
0.15
0.10
0.05
2010.0
2012.5
2015.0
2017.5
2020.0
Year
Athens
Atlanta
Augusta
Baltimore
Boston
Chapel Hill
Chicago
Dallas
Dayton
Hartford
Los Angeles
Lubbock
New York
Philadelphia
San Diego
Seattle
Stonybrook
Syracuse

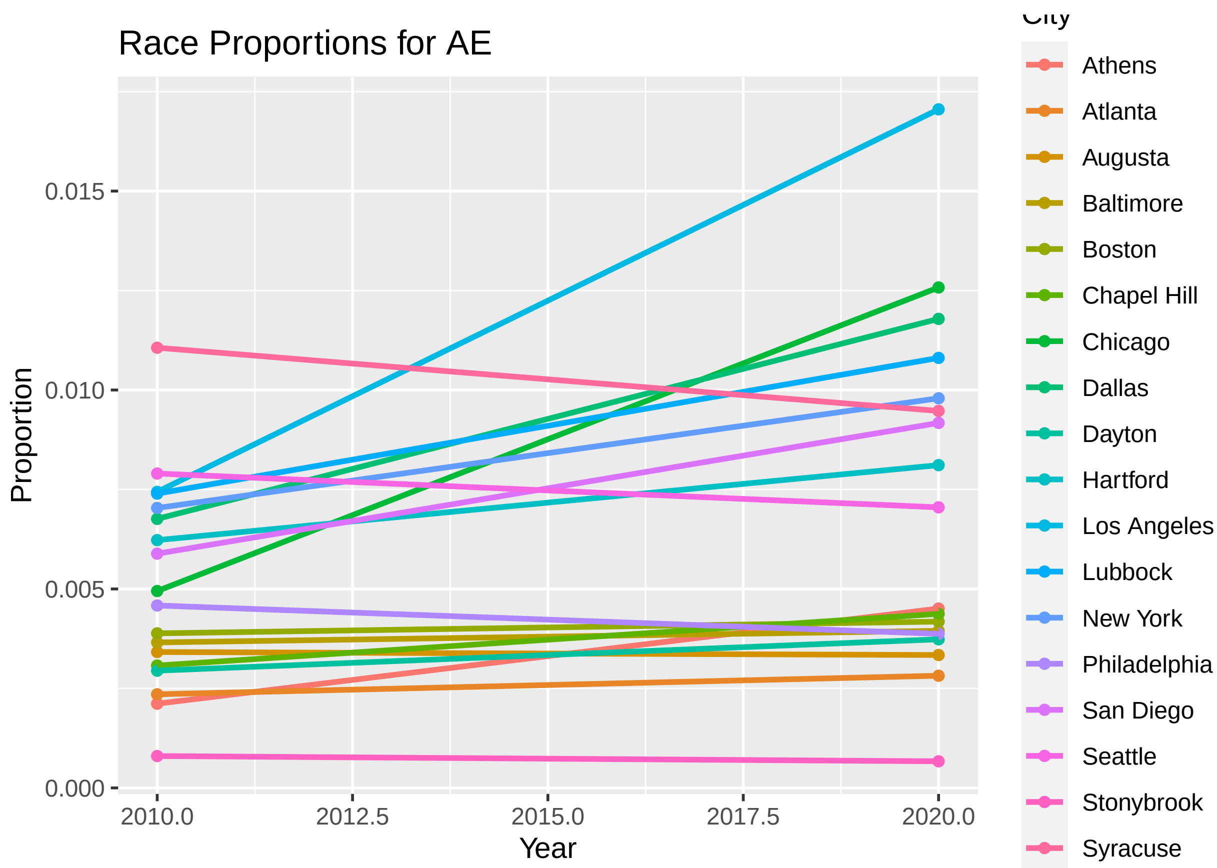


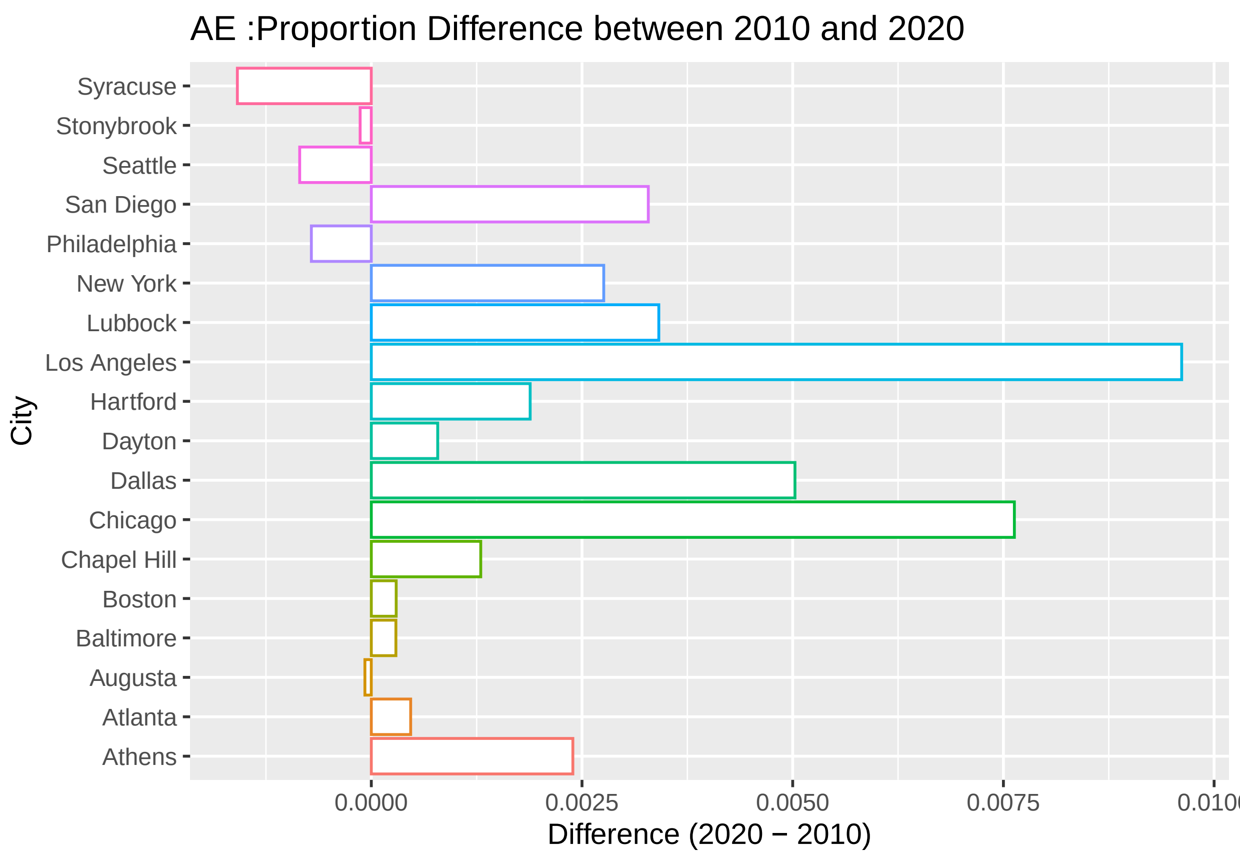

```
## `geom_smooth()` using formula = 'y ~ x'
```


Race Proportions for CA
Proportion
0.2
0.4
0.6
0.8
Year
2010.0
2012.5
2015.0
2017.5
2020.0
City
Athens
Atlanta
Augusta
Baltimore
Boston
Chapel Hill
Chicago
Dallas
Dayton
Hartford
Los Angeles
Lubbock
New York
Philadelphia
San Diego
Seattle
Stonybrook
Syracuse

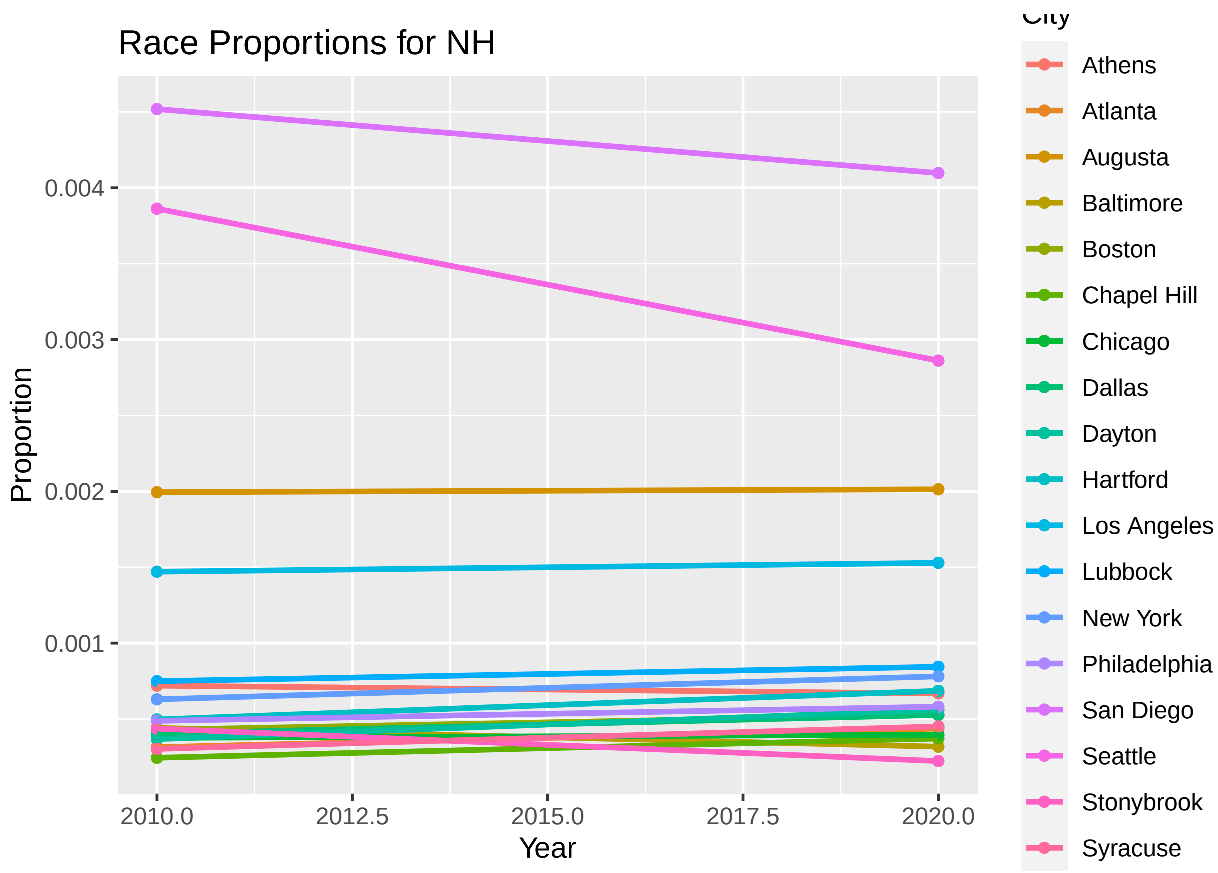


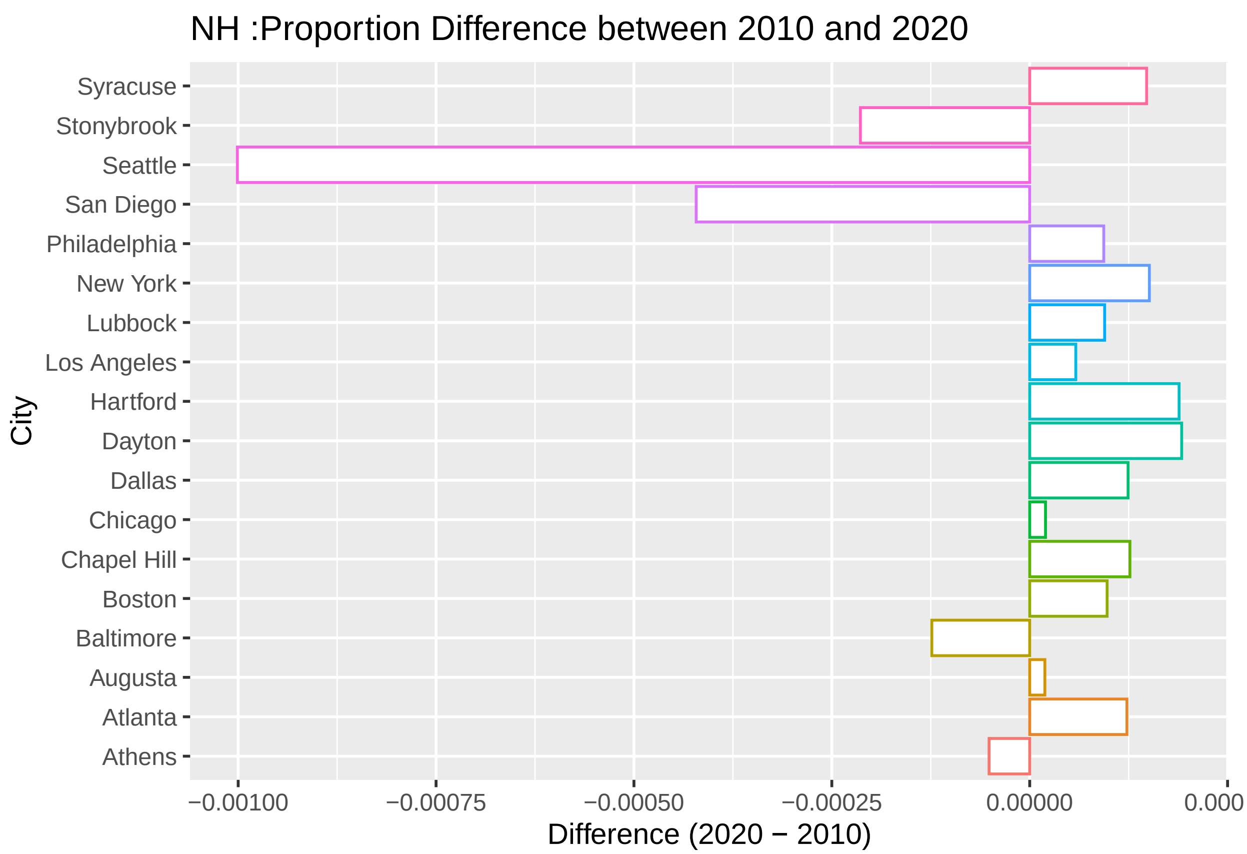

```
## `geom_smooth()` using formula = 'y ~ x'
```


Race Proportions for AE
City
Athens
Atlanta
Augusta
Baltimore
Boston
Chapel Hill
Chicago
Dallas
Dayton
Hartford
Los Angeles
Lubbock
New York
Philadelphia
San Diego
Seattle
Stonybrook
Syracuse
Proportion
0.015
0.010
0.005
0.000
2010.0
2012.5
2015.0
2017.5
2020.0
Year

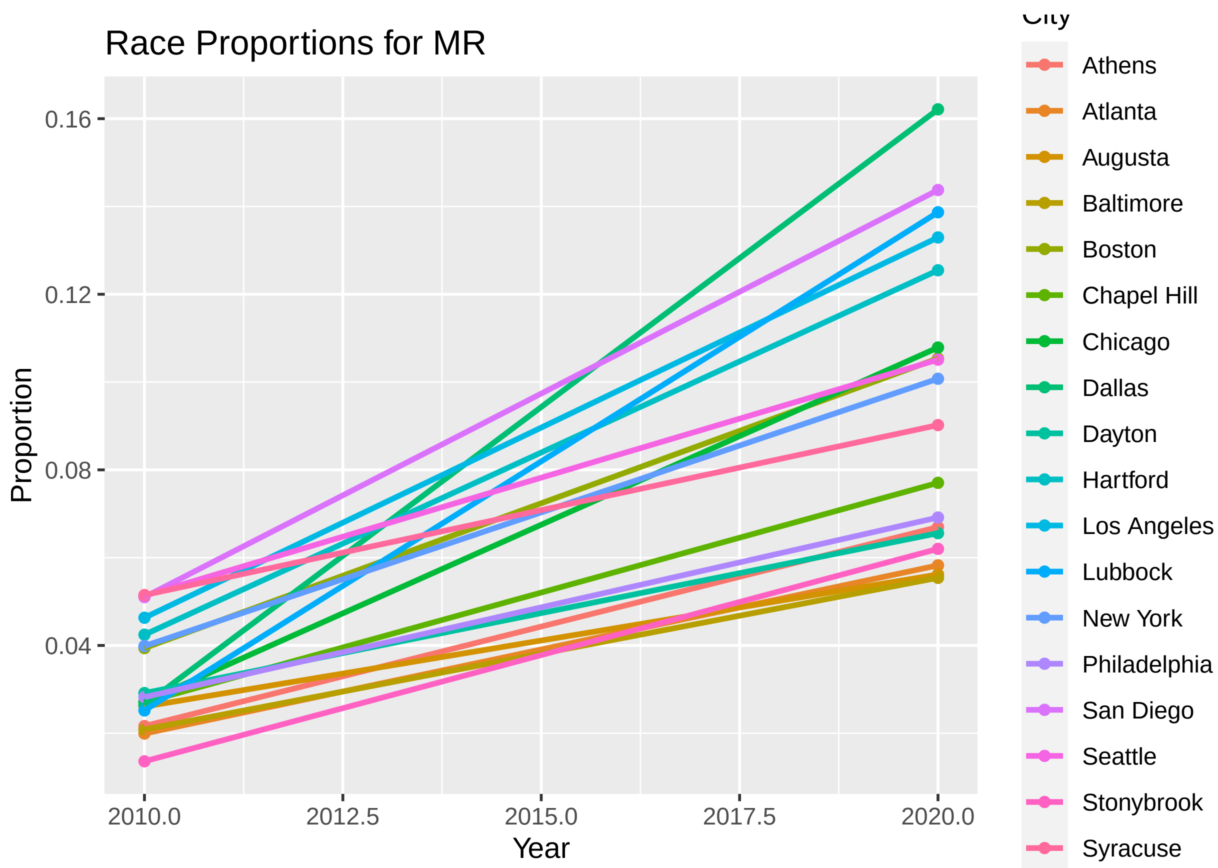


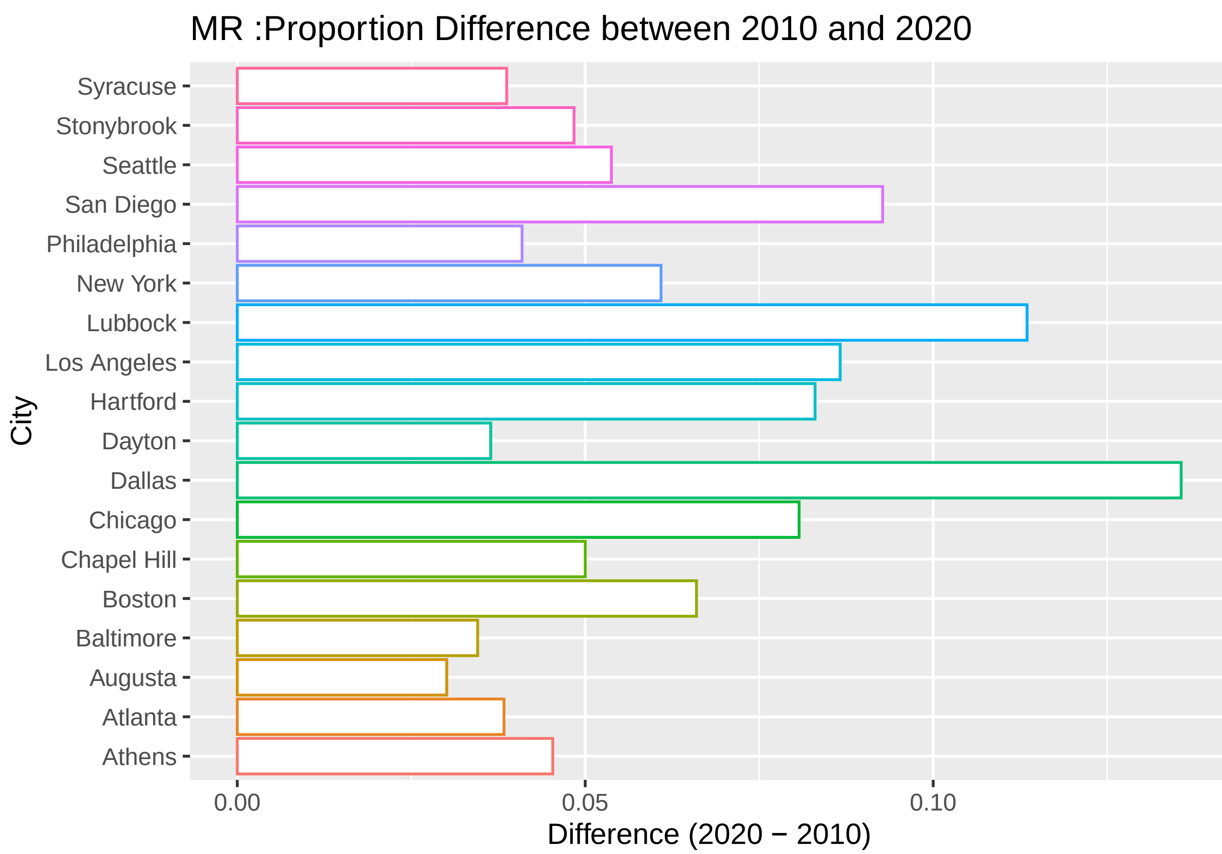

```
## `geom_smooth()` using formula = 'y ~ x'
```


Race Proportions for NH
Proportion
0.004
0.003
0.002
0.001
2010.0
2012.5
2015.0
2017.5
2020.0
Year
City
Athens
Atlanta
Augusta
Baltimore
Boston
Chapel Hill
Chicago
Dallas
Dayton
Hartford
Los Angeles
Lubbock
New York
Philadelphia
San Diego
Seattle
Stonybrook
Syracuse

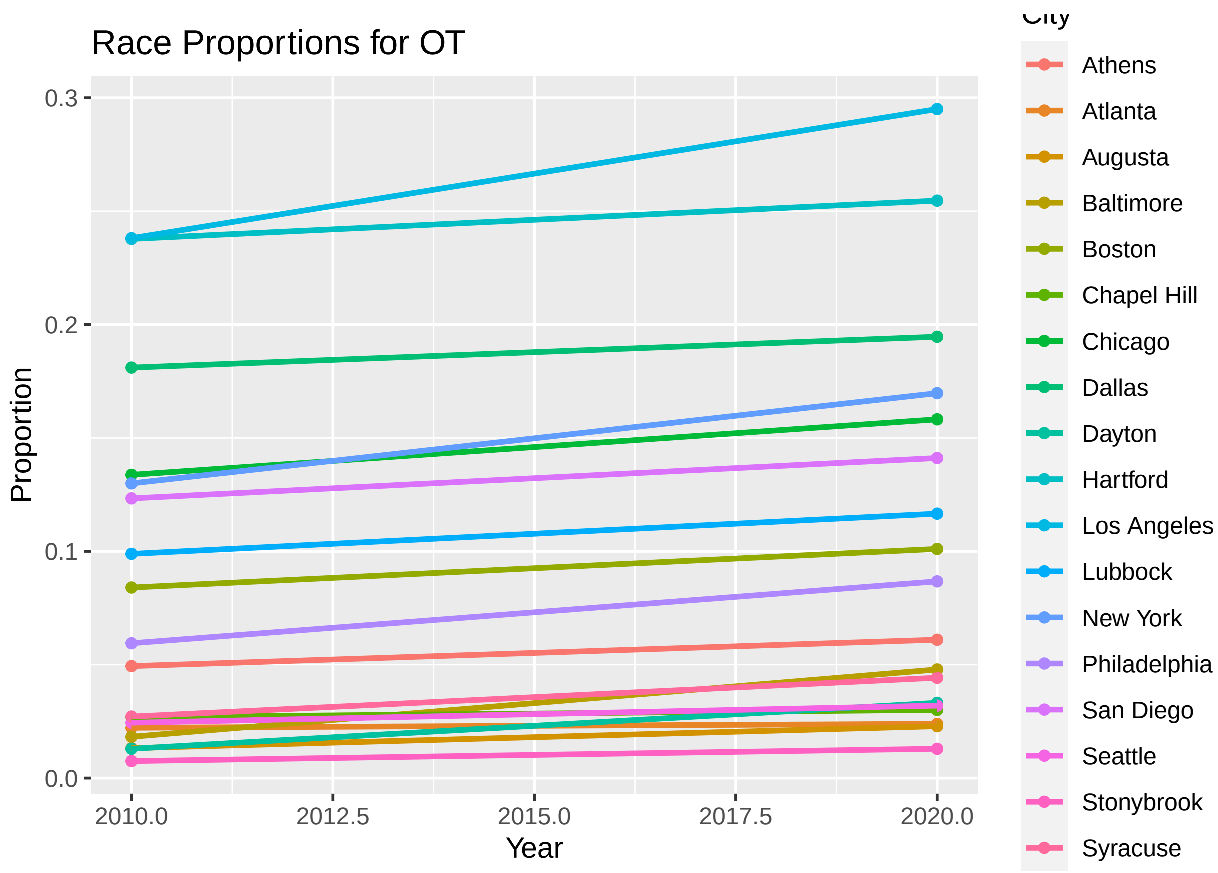



```
## `geom_smooth()` using formula = 'y ~ x'
```

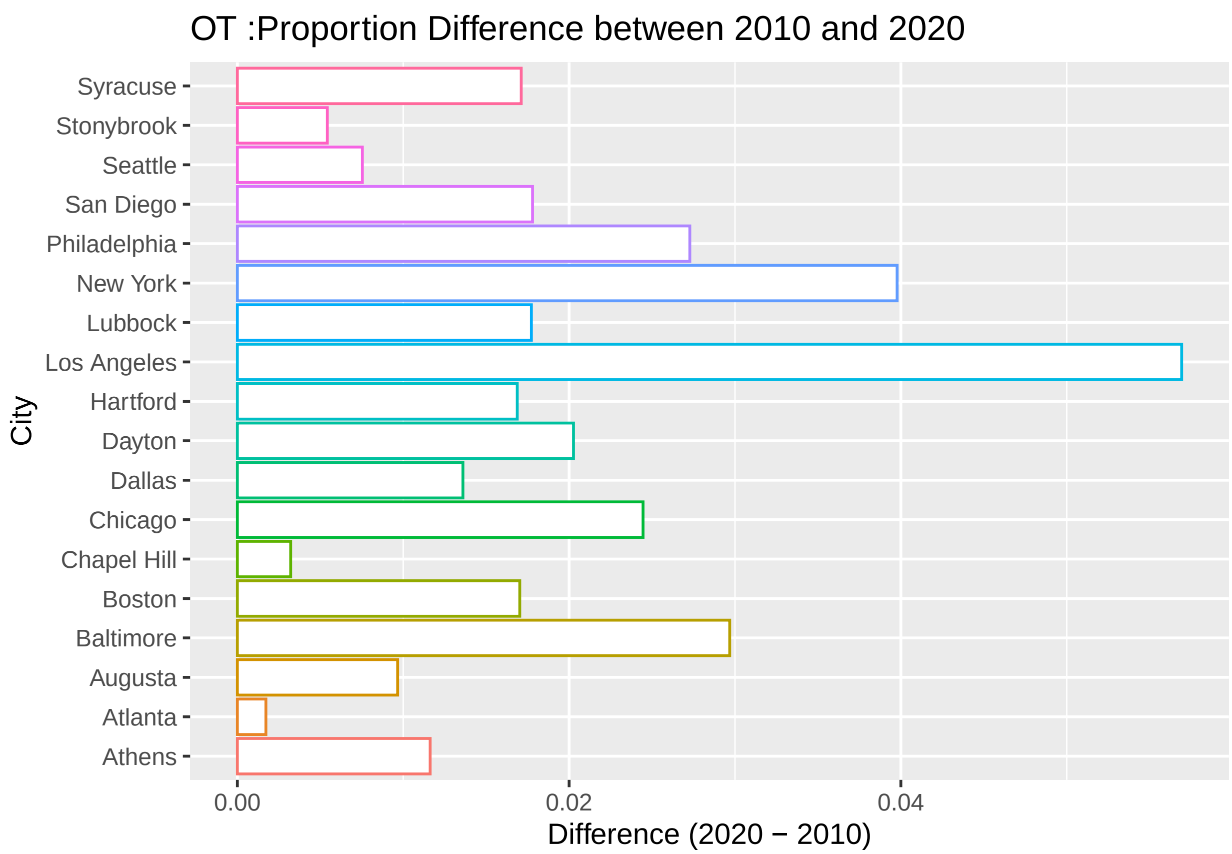

Race Proportions for MR
Proportion
0.16
0.12
0.08
0.04
2010.0
2012.5
2015.0
2017.5
2020.0
Year
City
Athens
Atlanta
Augusta
Baltimore
Boston
Chapel Hill
Chicago
Dallas
Dayton
Hartford
Los Angeles
Lubbock
New York
Philadelphia
San Diego
Seattle
Stonybrook
Syracuse




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



Race Proportions for OT
City
Proportion
Year
0.0
0.1
0.2
0.3
2010.0
2012.5
2015.0
2017.5
2020.0
Athens
Atlanta
Augusta
Baltimore
Boston
Chapel Hill
Chicago
Dallas
Dayton
Hartford
Los Angeles
Lubbock
New York
Philadelphia
San Diego
Seattle
Stonybrook
Syracuse




### Testing if proportion differences are significant for each city

```
dat_2010_races$City_n <- rowSums(dat_2010_races)
dat_2020_races$City_n <- rowSums(dat_2020_races)

dat_2010_races <- as.data.frame(dat_2010_races)
dat_2020_races <- as.data.frame(dat_2020_races)

# Probabilities deviate far from 0.5, so do not use Z-test
# Use Fisher exact test instead Benefits: Works with small
# counts Drawbacks: Conservative; hard to reject null
# hypothesis

## EXAMPLE: AA in Athens between 2010 and 2020
df <- data.frame(AA_yes = c(dat_2010_races[1, "AA"], dat_2020_races[1,
    "AA"]), AA_no = c(dat_2010_races$City_n[1] - dat_2010_races[1,
    "AA"], dat_2020_races$City_n[1] - dat_2020_races[1, "AA"]),
    row.names = c("2010", "2020"))
View(df)
res <- fisher.test(df)
res$p.value
```

```
## [1] 2.791439e-28
```

```r
# Looping through all the columns of AA, AS, and CA

add_fish_p_val <- function(race_acr) {

    p_vals <- rep(0, nrow(dat_2010_races))

    for (i in 1:nrow(dat_2010_races)) {
        df <- data.frame(Race_yes = c(dat_2010_races[i, race_acr],
            dat_2020_races[i, race_acr]), Race_no = c(dat_2010_races$City_n[i] -
            dat_2010_races[i, race_acr], dat_2020_races$City_n[i] -
            dat_2020_races[i, race_acr]), row.names = c("2010",
            "2020"))
        res <- fisher.test(df)
        p_vals[i] <- res$p.value
    }

    return(p_vals)

}

AA_df$p_val <- add_fish_p_val("AA")
AS_df$p_val <- add_fish_p_val("AS")
CA_df$p_val <- add_fish_p_val("CA")
```

### Dataframe of proportion differences and p-values

```r
## African-Americans
AA_df
```

```
##            City            AA         p_val
## 1        Athens -1.944982e-02  2.791439e-28
## 2       Atlanta -6.798029e-02  0.000000e+00
## 3       Augusta  1.111559e-02  1.059690e-12
## 4     Baltimore -5.947306e-02  0.000000e+00
## 5        Boston -3.804927e-02  0.000000e+00
## 6   Chapel Hill -2.216741e-04  8.984536e-01
## 7       Chicago -3.755375e-02  0.000000e+00
## 8        Dallas -1.687833e-02 1.817408e-213
## 9        Dayton -2.222015e-02  1.170440e-32
## 10     Hartford -5.441278e-03  5.612947e-03
## 11  Los Angeles -1.006447e-02  0.000000e+00
## 12      Lubbock  1.585400e-02  3.683919e-80
## 13     New York -3.470744e-02  0.000000e+00
## 14 Philadelphia -4.060082e-02  0.000000e+00
## 15    San Diego -7.955680e-03 1.883197e-158
## 16      Seattle -9.614377e-03  2.353079e-99
## 17   Stonybrook  4.084057e-05  1.000000e+00
## 18     Syracuse  1.212193e-02  7.963213e-13
```

```r
## Asian-Americans
AS_df
```

```
##            City           AS         p_val
## 1        Athens -0.003216066  5.071252e-05
## 2       Atlanta  0.013449487 1.450537e-246
## 3       Augusta  0.002631509  2.625173e-10
## 4     Baltimore  0.012784381  0.000000e+00
## 5        Boston  0.023080077  0.000000e+00
## 6   Chapel Hill  0.031574630  2.038002e-57
## 7       Chicago  0.015529183  0.000000e+00
## 8        Dallas  0.008645349 6.408031e-321
## 9        Dayton  0.005776411  2.574704e-47
## 10     Hartford  0.007835322  8.041077e-29
## 11  Los Angeles  0.006088298 1.431786e-153
## 12      Lubbock  0.012430745 1.601121e-140
## 13     New York  0.030309945  0.000000e+00
## 14 Philadelphia  0.020098289  0.000000e+00
## 15    San Diego  0.019659687  0.000000e+00
## 16      Seattle  0.032148101  0.000000e+00
## 17   Stonybrook  0.048973392  9.586896e-42
## 18     Syracuse  0.014744846  3.571064e-61
```

```r
## Caucasian-Americans
CA_df
```

```
##            City          CA         p_val
## 1        Athens -0.03663271  2.081849e-76
## 2       Atlanta  0.01389718  3.998200e-42
## 3       Augusta -0.05345118 2.677027e-273
## 4     Baltimore -0.01770108 2.552472e-102
## 5        Boston -0.06843315  0.000000e+00
## 6   Chapel Hill -0.08599455 8.330654e-224
## 7       Chicago -0.09081274  0.000000e+00
## 8        Dallas -0.14614949  0.000000e+00
## 9        Dayton -0.04122114 3.719557e-105
## 10     Hartford -0.10436239  0.000000e+00
## 11  Los Angeles -0.14925946  0.000000e+00
## 12      Lubbock -0.16300498  0.000000e+00
## 13     New York -0.09918037  0.000000e+00
## 14 Philadelphia -0.04708064  0.000000e+00
## 15    San Diego -0.12509969  0.000000e+00
## 16      Seattle -0.08198255  0.000000e+00
## 17   Stonybrook -0.10248567 4.617870e-116
## 18     Syracuse -0.08125801  0.000000e+00
```

Conclusions:

- Magnitude of differences is small for AS and AA
  - All differences are significant at the 0.05 level for AS
  - Some are not significant for AA
  - High power to detect small difference due to large sample size

- Magnitude of difference for CA much higher than for AS and AA
  - General decrease in proportion of CA between 2010 and 2020
  - All differences are significant
- Suggestion:
  - Designate a magnitude and p-value cutoff to determine which differences are consequential for the census "average"

## Final Thoughts:

- Cannot use "average" 2010 and 2020 for the null hypothesis
  - 2010 and 2020 are different, but magnitude may not be consequential