Script Combination with Outputs

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(fredr)
library(haven)
library(ipumsr)
library(tidycensus)
library(readxl)
```

00_get_fred_data

```
#Access FRED API: https://fred.stlouisfed.org/docs/api/api_key.html
#Sys.setenv("FRED_API_KEY" = "")
api_key <- Sys.getenv("FRED_API_KEY")


#Function to get FRED Housing data for full usa, 20 city composite,
# and 20 cities individually:
get_fred_housing <- function(series_id_code){
  #Function to get FRED housing data in a standard way
  #Input: "series_id_code" (string) - the FRED code
  #Output: FRED data in a dataframe

  #print(series_id_code)

  fred_data <- fredr(
    series_id = series_id_code,
    observation_start = as.Date("1987-01-01"),
    observation_end = as.Date("2022-02-01"),
    frequency = "a"
```

```
  )


  return(fred_data)
}

#Full usa:
housing_usa_cs <- get_fred_housing("CSUSHPINSA")

#20 city composite:
housing_20_city_cs <- get_fred_housing("SPCS20RSA")


#20 cities individually
fred_codes_housing <- c("SFXRSA", #San Francisco
                        "LXXRSA", #Los Angeles    - potentially (CMSA through Jan 2018)
                        "CHXRSA", #Chicago
                        "NYXRSA", #New York
                        "SDXRSA", #San Diego  - potentially (only by year through 2021)
                        "SEXRSA", #Seattle
                        "PHXRSA", #Phoenix    - potentially (only 2002 - 2021 and by year)
                        "MIXRSA", #Miami
                        "DNXRSA", #Denver     - potentially (only by year through 2021)
                        "BOXRSA", #Boston
                        "DAXRSA", #Dallas
                        "ATXRSA", #Atlanta
                        "POXRSA", #Portland   - potentially (only by year, discontinued in 2017)
                        "WDXRSA", #Washington DC - potentially (discontinued 2017)
                        "TPXRSA", #Tampa      - potentially (only year)
                        #"LVXRSA", #Las Vegas   - NOPE
                        "MNXRSA", #Minneapolis - by year
                        #"CRXRSA", #Charlotte   - NOPE
                        "DEXRSA", #Detroit
                        "CEXRSA"  #Cleveland   -yes discontinued in 2017
                        )
#Cities here are: San Francisco, Los Angeles, Chicago, New York,
#                 San Diego, Seattle, Phoenix, Miami,
#                 Denver, Boston, Dallas, Atlanta,
#                 Portland, Washington DC, Tampa, Las Vegas
#                 Minnesota, Charlotte, Detroit, Cleveland

housing_city_cs <- lapply(fred_codes_housing, get_fred_housing)
housing_city_cs <- bind_rows(housing_city_cs)


#Get Rental Data: Rent of Primary residence:

#CBSA RENT CODES:
#cbsa_rent_codes <- lapply(cbsa_home_codes, convert_home_to_rent_code)
cbsa_rent_codes <- c("CUURA101SEHA",  #New York
                     "CUURA319SEHA", #Atlanta
                     "CUURA422SEHA", #San Francisco
                     "CUURA103SEHA", #Boston
```

```
                      "CUURA208SEHA", #Detroit
                      "CUURA423SEHA", #Seattle
                      "CUURA320SEHA", #Miami
                      "CUURA207SEHA", #Chicago
                      "CUURA316SEHA",  #Dallas-Ft Worth

                      "CUURA421SEHA", #Los Angeles

                      "CUUSA424SEHA", #San Diego -a
                      "CUUSA429SEHA", #Phoenix -a
                      "CUUSA433SEHA", #Denver - a
                      "CUUSA425SEHA", #Portland -a
                      "CUURA311SEHA", #Washington DC
                      "CUUSA321SEHA", #Tampa -a
                      "CUUSA211SEHC01", #Minneapolis - a
                      "CUUSA210SEHA" #Cleveland- # - a
)
rent_city_bls <- lapply(cbsa_rent_codes, get_fred_housing)

rent_city_bls <- bind_rows(rent_city_bls)

rent_all_urban_bls <- get_fred_housing("CUSR0000SEHA")


##Clean the acquired data:
rent <- bind_rows(rent_city_bls, rent_all_urban_bls) %>%
  rename(real_rent= value) %>%
  mutate(
    data = "real_rent",
    year = str_sub(date, 1, 4),
    geography  = case_when(
                str_detect(series_id, "CUURA101") ~ "New York",
                str_detect(series_id, "CUURA319") ~ "Atlanta",
                str_detect(series_id, "CUURA422") ~ "San Francisco",
                str_detect(series_id, "CUURA103") ~ "Boston",
                str_detect(series_id, "CUURA208") ~ "Detroit",
                str_detect(series_id, "CUURA423") ~ "Seattle",
                str_detect(series_id, "CUURA320") ~ "Miami",
                str_detect(series_id, "CUURA207") ~ "Chicago",
                str_detect(series_id, "CUURA316") ~ "Dallas",
                str_detect(series_id, "CUURA421") ~ "Los Angeles",
                str_detect(series_id, "CUUSA424") ~ "San Diego",
                str_detect(series_id, "CUUSA429") ~ "Phoenix",
                str_detect(series_id, "CUUSA433") ~ "Denver",
                str_detect(series_id, "CUUSA425") ~ "Portland",
                str_detect(series_id, "CUURA311") ~ "Washington DC",
                str_detect(series_id, "CUUSA211") ~ "Minneapolis",
                str_detect(series_id, "CUUSA210") ~ "Cleveland",
                str_detect(series_id, "CUUSA321") ~ "Tampa",
                str_detect(series_id, "CUSR0000") ~ "USA")) %>%
  select(year, geography, real_rent) %>%
  mutate(geog_join = geography)
```

```r
home <- bind_rows(housing_20_city_cs, housing_city_cs, housing_usa_cs) %>%
  select(-c(realtime_start, realtime_end)) %>%
  rename(cs_home = value) %>%
  mutate(
    year = str_sub(date, 1, 4),
    geography_to_keep = case_when(
      str_detect(series_id, "NYXRSA") ~ "New York",
      str_detect(series_id, "ATXRSA") ~ "Atlanta",
      str_detect(series_id, "SFXRSA") ~ "San Francisco",
      str_detect(series_id, "BOXRSA") ~ "Boston",
      str_detect(series_id, "DEXRSA") ~ "Detroit",
      str_detect(series_id, "SEXRSA") ~ "Seattle",
      str_detect(series_id, "MIXRSA") ~ "Miami",
      str_detect(series_id, "CHXRSA") ~ "Chicago",
      str_detect(series_id, "DAXRSA") ~ "Dallas",
      str_detect(series_id, "LXXRSA") ~ "Los Angeles",
      str_detect(series_id, "SDXRSA") ~ "San Diego",
      str_detect(series_id, "PHXRSA") ~ "Phoenix",
      str_detect(series_id, "DNXRSA") ~ "Denver",
      str_detect(series_id, "POXRSA") ~ "Portland",
      str_detect(series_id, "WDXRSA") ~ "Washington DC",
      str_detect(series_id, "MNXRSA") ~ "Minneapolis",
      str_detect(series_id, "CUUSA210") ~ "Cleveland",
      str_detect(series_id, "TPXRSA") ~ "Tampa",
      str_detect(series_id, "CEXRSA") ~ "Cleveland",
      str_detect(series_id, "SPCS20RSA") ~ "20 City",
      str_detect(series_id, "CSUSHPINSA") ~ "USA")) %>%
  select(year, geography_to_keep, cs_home) %>%
  mutate(geog_join = ifelse(geography_to_keep == "20 City",
                            "USA",
                            geography_to_keep))



full <- left_join(home, rent, by = c("year", "geog_join")) %>%
  select(year, geography = geography_to_keep, cs_home, real_rent)

#Goal now is to reindex both cs_home and real_rent to the first year in
# which they both exist in the dataset:
year_first_both <- full %>%
  mutate(both_have_data = ifelse(!is.na(cs_home) & !is.na(real_rent), TRUE, FALSE)) %>%
  filter(both_have_data ==TRUE) %>%
  group_by(geography) %>%
  slice_min(order_by = year) %>%
  select(geography, year_first_both = year)

full <- full %>%
  left_join(year_first_both)
```

```
## Joining, by = "geography"
```

```r
reindex_value <- full %>%
  filter(year == year_first_both) %>%
  select(geography, cs_home_reindex_val = cs_home, real_rent_reindex_val = real_rent)
```

```
full <- full %>%
  left_join(reindex_value) %>%
  mutate(cs_home_reindexed = cs_home/cs_home_reindex_val,
         real_rent_reindexed = real_rent/real_rent_reindex_val) %>%
  select(year, geography, real_rent, cs_home, real_rent_reindexed, cs_home_reindexed)
```

```
## Joining, by = "geography"
```

```
#Description:
# The table above join two sets of disparate datasets.
#cs_home is the Case Shiller Home Price Index for the MSA or CBSA or the
#city listed listed. For every
# one of the 18 cities, the Case Shiller Index refers specifically to that city.
# The "20 City" refers to the 20 City Case Shiller Index. The real-rent data
# comes from the BLS real rent index. The all USA Case Shiller AND the 20 city
# case Shiller are paired with the BLS data for all Urban consumers (neither
# match is perfect, but both offer reasonable comparisons. )

#The reindexed values are the initial values acquired divided by the first year
# in which the data has values both for the real rent data from BLS and for the
#Case Shiller index.

head(full)
```

```
## # A tibble: 6 x 6
##   year  geography real_rent cs_home real_rent_reindexed cs_home_reindexed
##   <chr> <chr>         <dbl>   <dbl>               <dbl>             <dbl>
## 1 2000  20 City        184.    106.                1                 1
## 2 2001  20 City        192.    117.                1.04              1.10
## 3 2002  20 City        200.    128.                1.09              1.20
## 4 2003  20 City        206.    143.                1.12              1.34
## 5 2004  20 City        211.    164.                1.15              1.55
## 6 2005  20 City        217.    190.                1.18              1.79
```

```
write_csv(full, "data/processed/rent_home_prices.csv")
```

01_rent_home_plots

```
library(tidyverse)
#Plot Change Over Time of Rent and Home Indices

library(tidyverse)
library(gtable)
library(lemon)
```

```
##
## Attaching package: 'lemon'
```

```
## The following object is masked from 'package:purrr':
##
##     %||%
```

```
## The following objects are masked from 'package:ggplot2':
##
##     CoordCartesian, element_render
```

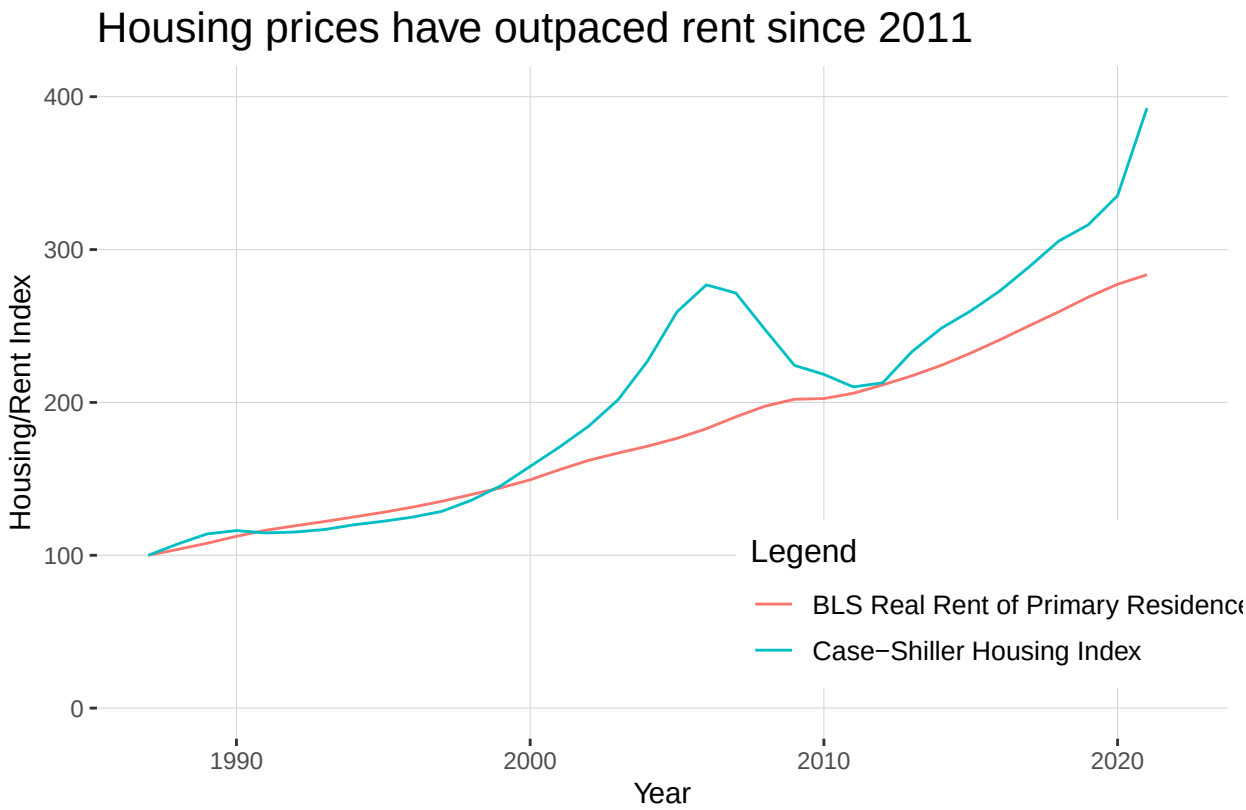
```
library(ggthemes)

rh <- read_csv("data/processed/rent_home_prices.csv") #rh for rent and house
```

```
## Rows: 679 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geography
## dbl (5): year, real_rent, cs_home, real_rent_reindexed, cs_home_reindexed
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Filter to appropriate Geography:
us_rh <- rh %>%
  filter(geography == "USA") %>%
  pivot_longer(!c("year", "geography"), names_to = "index", values_to = "value") %>%
  mutate(value = value * 100) %>%
  mutate(Index = case_when(
    index == "cs_home" ~ "Case-Shiller Housing Index",
    index == "cs_home_reindexed" ~ "Case-Shiller Housing Index ",
    index == "real_rent" ~ "BLS Real Rent of Primary Residence",
    index == "real_rent_reindexed" ~ "BLS Real Rent of Primary Residence")) %>%
  filter(index == "cs_home_reindexed" | index == "real_rent_reindexed")

top_20 <- rh %>%
  filter(geography == "20 City")

cbsa_rh <- rh %>%
  filter(!(geography %in% c("20 City", "USA"))) %>%
  pivot_longer(!c("year", "geography"), names_to = "index", values_to = "value") %>%
  mutate(value = value * 100) %>%
  mutate(Index = case_when(
    index == "cs_home" ~ "Case-Shiller Housing Index",
    index == "cs_home_reindexed" ~ "Case-Shiller Housing Index ",
    index == "real_rent" ~ "BLS Real Rent of Primary Residence",
    index == "real_rent_reindexed" ~ "BLS Real Rent (Primary Residence)"))

cbsa_rh_index <- cbsa_rh %>%
  filter(index %in% c("cs_home_reindexed", "real_rent_reindexed"))

cbsa_rh_no_index <- cbsa_rh %>%
  filter(!(index %in% c("cs_home_reindexed", "real_rent_reindexed")))

ggplot(data = us_rh, aes(x = year, y = value, color = Index)) +
  geom_line() +
  labs(x = "Year",
       y = "Housing/Rent Index",
       color = "Legend",
       title = "Housing prices have outpaced rent since 2011",
       caption = "Index = 100 in 1987. Data From FRED. Rent for U.S. City Average. Housing Index for al
  scale_y_continuous(limits = c(0, 400), breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.text.y = element_text(angle = 0),
```

```
        panel.grid.major = element_line(colour = "lightgrey", size = .15),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        strip.text.x = element_text(size = 11),
        strip.background =element_rect(fill="#EAEAEA"),
        legend.key = element_rect(colour = NA, fill = NA),
        legend.title = element_text(size=12), #change legend title font size
        legend.text = element_text(size=10),
        plot.title = element_text(size=16),
        plot.caption =  element_text(size = 9, hjust = 0),
        legend.position = c(0.8, 0.2)
  )
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



Index = 100 in 1987. Data From FRED. Rent for U.S. City Average. Housing Index for all of the US.

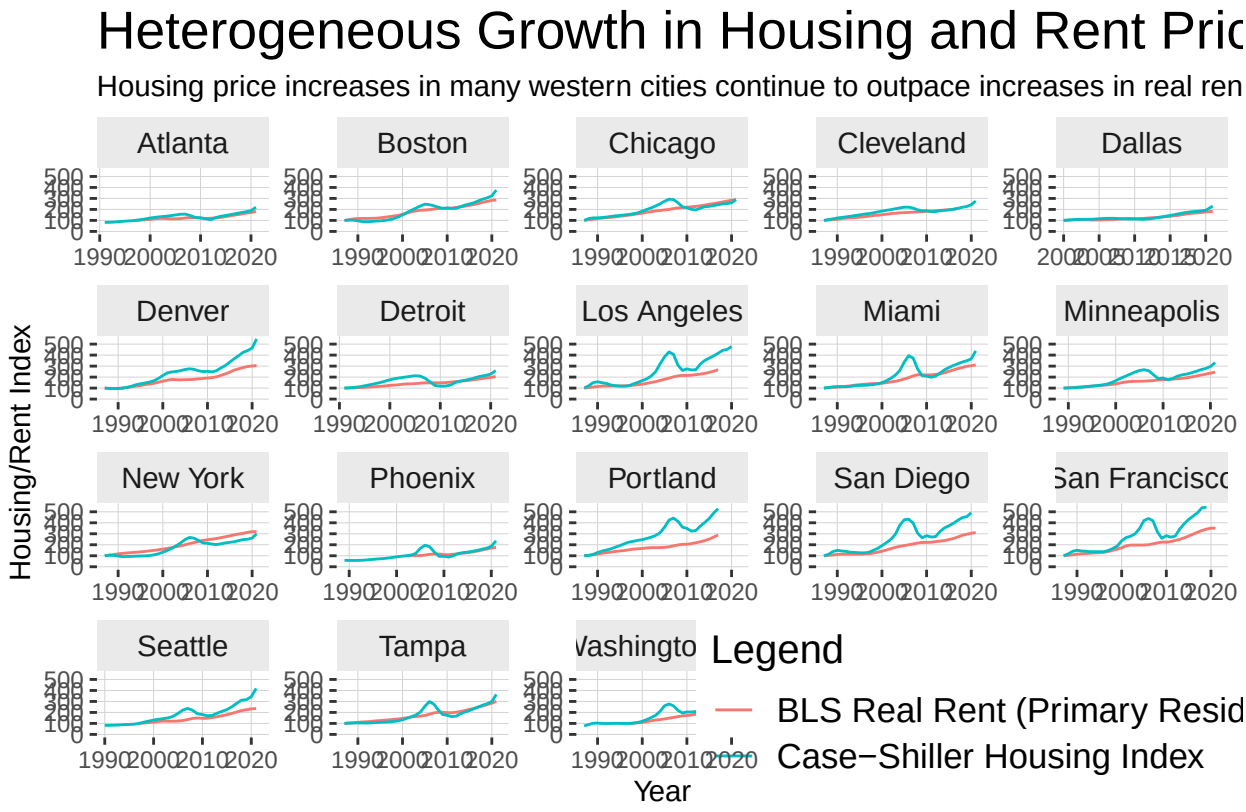
```
shift_legend2 <- function(p) {
  #CODE COPIED FROM: https://stackoverflow.com/questions/54438495/shift-legend-into-empty-facets-of-a-f
  # ...
  # to grob
  gp <- ggplotGrob(p)
  facet.panels <- grep("^panel", gp[["layout"]][["name"]])
  empty.facet.panels <- sapply(facet.panels, function(i) "zeroGrob" %in% class(gp[["grobs"]][[i]]))
  empty.facet.panels <- facet.panels[empty.facet.panels]

  # establish name of empty panels
  empty.facet.panels <- gp[["layout"]][empty.facet.panels, ]
  names <- empty.facet.panels$name
```

```
  # example of names:
  #[1] "panel-3-2" "panel-3-3"

  # now we just need a simple call to reposition the legend
  reposition_legend(p, 'center', panel=names)
}

p <-ggplot(data = cbsa_rh_index) +
  geom_line(aes(x = year, y = value, color = Index)) +
  facet_wrap(geography ~ ., scales='free') +
  labs(x = "Year",
       y = "Housing/Rent Index",
       color = "Legend",
       title = "Heterogeneous Growth in Housing and Rent Prices by City",
       subtitle = "Housing price increases in many western cities continue to outpace increases in real
       caption = "Index = 100 in the first year of data for both cities. Data From FRED. Rent for U.S. C
  scale_y_continuous(limits = c(0, 550), breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.text.y = element_text(angle = 0),
        panel.grid.major = element_line(colour = "lightgrey", size = .15),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        strip.text.x = element_text(size = 11),
        strip.background =element_rect(fill="#EAEAEA"),
        legend.key = element_rect(colour = NA, fill = NA),
        legend.title = element_text(size=15), #change legend title font size
        legend.text = element_text(size=12.5),
        plot.title = element_text(size=20),
        plot.caption =  element_text(size = 9, hjust = 0)
  )
p <- shift_legend2(p)
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



Index = 100 in the first year of data for both cities. Data From FRED. Rent for U.S. City Average. Housing Inc

02_clean_analyze_mian_sufi

```
#Read data:
m <- read_dta("data/raw/miansufieconometrica_countylevel.dta")
x <- read_excel("data/raw/list1.xls", skip=2)

#Process xwalk:
xwalk <- x %>%
  janitor::clean_names() %>%
  mutate(fips = str_c(fips_state_code, fips_county_code)) %>%
  select(fips, cbsa_code, cbsa_title, metropolitan_micropolitan_statistical_area)

#Process mian sufi:
ms <- m %>%
  mutate(fips = str_pad(fips, width = 5, side = "left", pad = "0")) %>%
  select(fips, netwp_h)

write.csv(ms, "data/processed/ms.csv")


#Get 2007-ish housing count by county via tidycensus:

#SET CENSUS API KEY: https://walker-data.com/tidycensus/reference/census_api_key.html
#census_api_key()
housing_county_07 <- get_acs(geography = "county",
                             variables = "B25008_002",
```

```
                          year = 2009,
                          survey = "acs5")

## Getting data from the 2005-2009 5-year ACS
housing_county_07 <- housing_county_07 %>%
  select(fips = GEOID, count_housing = estimate)

#Get names of cities:
rent_home_prices <- read.csv("data/processed/rent_home_prices.csv")
cities <- unique(rent_home_prices$geography)
cbsa_codes_18_cities <- c("31080",  #Los Angeles
                          "16980", #Chicago
                          "38060", #Phoenix
                          "41740", #San Diego
                          "35620",  #New York
                          "19100", #Dallas
                          "19820", #Detroit
                          "42660", #Seattle
                          "33100", #Miami
                          "17460", #Cleveland
                          "33460", #Minneapolis
                          "41180", #St. Louis
                          "45300", #Tampa
                          "47900", #Washington DC.
                          "17460", #Cleveland
                          "12060", #Atlanta
                          "14460", #Boston
                          "38900", #Portland
                          "19740",  #Denver
                          "41860" #San Francisco
                          )

#Get the weighted sum of mian and sufi counties where weight is number of
#owner-occupied housing units from acs 2005- 2009
ms_weighted <- ms %>%
  left_join(housing_county_07) %>% #Join housing data
  left_join(xwalk) %>% #Join crosswalk
  #Use only mian and sufi data for the 18 cities
  filter(cbsa_code %in% cbsa_codes_18_cities,
         !is.na(netwp_h)) %>% #And for where it exists

  #get weighted sum of netwp_h where weight is number of housing units
  group_by(cbsa_code, cbsa_title) %>%
  summarize(netwp_h_weighted_sum = sum(netwp_h * count_housing) / sum(count_housing)) %>%

  #Get geography column for future join
  mutate(geography = sub("-.*", "", cbsa_title),
         geography = ifelse(geography == "St. Louis, MO",
                            "St. Louis",
                            geography),
         geography = ifelse(geography == "Washington",
                            "Washington DC",
                            geography)) %>%
```

```
  select(-cbsa_title)
```

```
## Joining, by = "fips"
```

```
## Joining, by = "fips"
```

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

```
write.csv(ms_weighted, "data/processed/ms_weighted.csv")

rent_home_ms <- left_join(rent_home_prices, ms_weighted)
```

```
## Joining, by = "geography"
```

```
#######

#Base Regression Set-Up:
base_reg <- rent_home_ms %>%
  filter(geography != "20 City" & geography != "USA") %>%
  filter(year == 2006) %>%
  mutate(home_rent_div = cs_home/real_rent,
         home_rent_reindex_div = cs_home_reindexed / real_rent_reindexed)

base_model <- lm(netwp_h_weighted_sum ~ home_rent_div, data = base_reg)
reindex_model <- lm(netwp_h_weighted_sum ~ home_rent_reindex_div, data = base_reg)


#Results using 2006 to predict Mian and Sufi 2007 - 2009 drop
summary(base_model)
```

```
##
## Call:
## lm(formula = netwp_h_weighted_sum ~ home_rent_div, data = base_reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10227 -0.02515  0.01059  0.03139  0.07872
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.01850    0.03149  -0.587  0.56511
## home_rent_div -0.11573    0.03067  -3.774  0.00166 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05103 on 16 degrees of freedom
## Multiple R-squared:  0.4709, Adjusted R-squared:  0.4379
## F-statistic: 14.24 on 1 and 16 DF,  p-value: 0.001662
```

```
summary(reindex_model)
```

```
##
## Call:
## lm(formula = netwp_h_weighted_sum ~ home_rent_reindex_div, data = base_reg)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
```

```
## -0.099307 -0.027304 -0.001456  0.025676  0.116536
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.05809    0.05958   0.975  0.34404
## home_rent_reindex_div -0.11032    0.03442  -3.205  0.00552 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05475 on 16 degrees of freedom
## Multiple R-squared:  0.391,  Adjusted R-squared:  0.353
## F-statistic: 10.27 on 1 and 16 DF,  p-value: 0.005516
```

```
#stargazer::stargazer(reindex_model, type="html")
#Note ^This is commented out, but I used it to create the regression output in the paper
```

03_clean_analyze_census_data