ACS_extraction

ACS extraction code

Micah Hysong 09/15/25

Chosen based on metrics available in All of Us

Used https://github.com/geomarker-io/dep_index/tree/master/2018/R as guide

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```
library(tidycensus)
library(getopt)
library(purrr)
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```
census_api_key(Sys.getenv("CENSUS_API_KEY")) #, install = TRUE) #get your API key from https
```

```
To install your API key for use in future sessions, run this function with `install = TRUE`.
```

```
# Ran Sys.setenv(CENSUS_API_KEY="key") within R project
```

Percent households on public assisted income / SNAP

```
years_needed <- 2013:2023

acs_assisted_income_all <- map_df(years_needed, function(y) {
  get_acs(
    geography = "zcta",
    variables = "B19058_002",   # households w/ public assistance income
    summary_var = "B19058_001", # total households
    year = y,                   # 5-year ACS ending in this year
    survey = "acs5"           # 5-year estimates
  ) %>%
    mutate(
      year = y,
      fraction_assisted_income = estimate / summary_est,
       # extract the 5-digit ZCTA from NAME, then take first 3 digits
      zcta5 = stringr::str_extract(NAME, "\\d{5}"),
      zip3  = substr(zcta5, 1, 3)
    )
})
```

```
Getting data from the 2009-2013 5-year ACS

Getting data from the 2010-2014 5-year ACS

Getting data from the 2011-2015 5-year ACS

Getting data from the 2012-2016 5-year ACS

Getting data from the 2013-2017 5-year ACS

Getting data from the 2014-2018 5-year ACS
```

```
Getting data from the 2015-2019 5-year ACS

Getting data from the 2016-2020 5-year ACS

Getting data from the 2017-2021 5-year ACS

Getting data from the 2018-2022 5-year ACS

Getting data from the 2019-2023 5-year ACS
```

```r
# Population-weighted aggregation to 3-digit ZIP per year
acs_assisted_income_zip3 <- acs_assisted_income_all %>%
  group_by(year, zip3) %>%
  summarise(
    households_total = sum(summary_est, na.rm = TRUE),
    households_assisted = sum(estimate, na.rm = TRUE),
    fraction_assisted_income = households_assisted / households_total,
    .groups = "drop"
  )

#Remove zip codes for American Samoa', 'Guam', 'Northern Mariana Islands','Puerto Rico', #'U

territories <- c("967", "969", "006", "007", "009", "968", "008")

acs_assisted_income_zip3 <- acs_assisted_income_zip3 |> filter(!zip3 %in% territories)
```

EDUCATIONAL ATTAINMENT Population 25 years and over w/ High school graduate (includes equivalency)

```r
years_needed <- 2013:2023

acs_high_school_education <- map_df(years_needed, function(y) {
  get_acs(
    geography = "zcta",
    variables = paste0('B15003_0',17:25),   # high school and above
    summary_var = 'B15003_001', # total
    year = y,                   # 5-year ACS ending in this year
    survey = "acs5"             # 5-year estimates
  ) %>%
```

```
    summarise(
      high_school_edu = sum(estimate, na.rm = TRUE),
      total = unique(summary_est),
      .by = c(GEOID, NAME)
    ) %>%
    mutate(
      year = y,
      fraction_high_school_edu = high_school_edu / total,
       # extract the 5-digit ZCTA from NAME, then take first 3 digits
      zcta5 = stringr::str_extract(NAME, "\\d{5}"),
      zip3  = substr(zcta5, 1, 3)
    )
})
```

```
Getting data from the 2009-2013 5-year ACS

Getting data from the 2010-2014 5-year ACS

Getting data from the 2011-2015 5-year ACS

Getting data from the 2012-2016 5-year ACS

Getting data from the 2013-2017 5-year ACS

Getting data from the 2014-2018 5-year ACS

Getting data from the 2015-2019 5-year ACS

Getting data from the 2016-2020 5-year ACS

Getting data from the 2017-2021 5-year ACS

Getting data from the 2018-2022 5-year ACS

Getting data from the 2019-2023 5-year ACS
```

```r
# Population-weighted aggregation to 3-digit ZIP per year
acs_high_school_education_zip3 <- acs_high_school_education %>%
  group_by(year, zip3) %>%
  summarise(
    households_total = sum(total, na.rm = TRUE),
    households_high_school_edu = sum(high_school_edu, na.rm = TRUE),
    fraction_high_school_edu = households_high_school_edu / households_total,
    .groups = "drop"
  )

#Remove zip codes for American Samoa', 'Guam', 'Northern Mariana Islands','Puerto Rico', #'U
acs_high_school_education_zip3 <- acs_high_school_education_zip3 |> filter(!zip3 %in% territ
```

Median household income in the past 12 months in year inflation-adjusted dollars

```r
years_needed <- 2013:2023

acs_median_income <- map_df(years_needed, function(y) {
  get_acs(
    geography = "zcta",
    variables = "B19013_001",   # median household income
    summary_var = 'B19001_001', # total
    year = y,                   # 5-year ACS ending in this year
    survey = "acs5"             # 5-year estimates
  ) %>%
    mutate(
      year = y,
      median_income = estimate,
      total_households = summary_est,
       # extract the 5-digit ZCTA from NAME, then take first 3 digits
      zcta5 = stringr::str_extract(NAME, "\\d{5}"),
      zip3  = substr(zcta5, 1, 3)
    )
})
```

```
Getting data from the 2009-2013 5-year ACS

Getting data from the 2010-2014 5-year ACS

Getting data from the 2011-2015 5-year ACS
```

```
Getting data from the 2012-2016 5-year ACS

Getting data from the 2013-2017 5-year ACS

Getting data from the 2014-2018 5-year ACS

Getting data from the 2015-2019 5-year ACS

Getting data from the 2016-2020 5-year ACS

Getting data from the 2017-2021 5-year ACS

Getting data from the 2018-2022 5-year ACS

Getting data from the 2019-2023 5-year ACS
```

```r
# Population-weighted aggregation to 3-digit ZIP per year
acs_median_income_zip3 <- acs_median_income %>%
  group_by(year, zip3) %>%
  summarise(
    median_income = sum(median_income * total_households, na.rm = TRUE) / sum(total_household
    total_households_zip3 = sum(total_households, na.rm = TRUE),
    .groups = "drop"
  )

#Remove zip codes for American Samoa', 'Guam', 'Northern Mariana Islands','Puerto Rico', #'U
acs_median_income_zip3 <- acs_median_income_zip3 |> filter(!zip3 %in% territories)
```

Fraction in poverty

```r
years_needed <- 2013:2023

acs_poverty <- map_df(years_needed, function(y) {
  get_acs(
    geography = "zcta",
    variables = 'B17001_002', #!Income in the past 12 months below poverty level:
    summary_var = 'B17001_001', #Total
    year = y,                   # 5-year ACS ending in this year
```

```
    survey = "acs5"             # 5-year estimates
  ) %>%
    mutate(
      year = y,
      poverty = estimate,
      total_households = summary_est,
       # extract the 5-digit ZCTA from NAME, then take first 3 digits
      zcta5 = stringr::str_extract(NAME, "\\d{5}"),
      zip3  = substr(zcta5, 1, 3)
    )
})
```

```
Getting data from the 2009-2013 5-year ACS

Getting data from the 2010-2014 5-year ACS

Getting data from the 2011-2015 5-year ACS

Getting data from the 2012-2016 5-year ACS

Getting data from the 2013-2017 5-year ACS

Getting data from the 2014-2018 5-year ACS

Getting data from the 2015-2019 5-year ACS

Getting data from the 2016-2020 5-year ACS

Getting data from the 2017-2021 5-year ACS

Getting data from the 2018-2022 5-year ACS

Getting data from the 2019-2023 5-year ACS
```

```r
# Population-weighted aggregation to 3-digit ZIP per year
acs_poverty_zip3 <- acs_poverty %>%
  group_by(year, zip3) %>%
  summarise(
    households_total = sum(total_households, na.rm = TRUE),
    households_poverty = sum(poverty, na.rm = TRUE),
    fraction_poverty = households_poverty / households_total,
    .groups = "drop"
  )

#Remove zip codes for American Samoa', 'Guam', 'Northern Mariana Islands','Puerto Rico', #'U
acs_poverty_zip3 <- acs_poverty_zip3 |> filter(!zip3 %in% territories)
```

No Health Insurance

```r
years_needed <- 2013:2023

acs_no_health_insurance <- map_df(years_needed, function(y) {
  get_acs(
    geography = "zcta",
    variables = paste0('B27010_0',c(17, 33, 50, 66)), #No health insurance for different age
    summary_var = 'B27010_001', #Total
    year = y,                   # 5-year ACS ending in this year
    survey = "acs5"             # 5-year estimates
  ) %>%
    mutate(
      year = y,
      no_health_insurance = estimate,
      total_households = summary_est,
       # extract the 5-digit ZCTA from NAME, then take first 3 digits
      zcta5 = stringr::str_extract(NAME, "\\d{5}"),
      zip3  = substr(zcta5, 1, 3)
    )
})
```

```
Getting data from the 2009-2013 5-year ACS

Getting data from the 2010-2014 5-year ACS

Getting data from the 2011-2015 5-year ACS
```

```
Getting data from the 2012-2016 5-year ACS

Getting data from the 2013-2017 5-year ACS

Getting data from the 2014-2018 5-year ACS

Getting data from the 2015-2019 5-year ACS

Getting data from the 2016-2020 5-year ACS

Getting data from the 2017-2021 5-year ACS

Getting data from the 2018-2022 5-year ACS

Getting data from the 2019-2023 5-year ACS
```

```r
# Population-weighted aggregation to 3-digit ZIP per year
acs_no_health_insurance_zip3 <- acs_no_health_insurance %>%
  group_by(year, zip3) %>%
  summarise(
    households_total = sum(total_households, na.rm = TRUE),
    households_no_health_insurance = sum(no_health_insurance, na.rm = TRUE),
    fraction_no_health_insurance = households_no_health_insurance / households_total,
    .groups = "drop"
  )

#Remove zip codes for American Samoa', 'Guam', 'Northern Mariana Islands','Puerto Rico', #'U
acs_no_health_insurance_zip3 <- acs_no_health_insurance_zip3 |> filter(!zip3 %in% territorie
```

Fraction Vacant Housing

```r
years_needed <- 2013:2023

acs_vacant_housing <- map_df(years_needed, function(y) {
  get_acs(
    geography = "zcta",
    variables = 'B25002_003', #Vacant Housing
    summary_var = 'B25002_001', #Total
```

```
    year = y,                    # 5-year ACS ending in this year
    survey = "acs5"              # 5-year estimates
  ) %>%
    mutate(
      year = y,
      vacant_housing = estimate,
      total_households = summary_est,
       # extract the 5-digit ZCTA from NAME, then take first 3 digits
      zcta5 = stringr::str_extract(NAME, "\\d{5}"),
      zip3  = substr(zcta5, 1, 3)
    )
})
```

```
Getting data from the 2009-2013 5-year ACS

Getting data from the 2010-2014 5-year ACS

Getting data from the 2011-2015 5-year ACS

Getting data from the 2012-2016 5-year ACS

Getting data from the 2013-2017 5-year ACS

Getting data from the 2014-2018 5-year ACS

Getting data from the 2015-2019 5-year ACS

Getting data from the 2016-2020 5-year ACS

Getting data from the 2017-2021 5-year ACS

Getting data from the 2018-2022 5-year ACS

Getting data from the 2019-2023 5-year ACS
```

```r
# Population-weighted aggregation to 3-digit ZIP per year
acs_vacant_housing_zip3 <- acs_vacant_housing %>%
  group_by(year, zip3) %>%
  summarise(
    households_total = sum(total_households, na.rm = TRUE),
    households_vacant_housing = sum(vacant_housing, na.rm = TRUE),
    fraction_vacant_housing = households_vacant_housing / households_total,
    .groups = "drop"
  )

#Remove zip codes for American Samoa', 'Guam', 'Northern Mariana Islands','Puerto Rico', #'U
acs_vacant_housing_zip3 <- acs_vacant_housing_zip3 |> filter(!zip3 %in% territories)
```

Merge all Data

```r
# Make sure all your data frames exist
dfs <- list(
  acs_assisted_income_zip3,
  acs_high_school_education_zip3,
  acs_median_income_zip3,
  acs_no_health_insurance_zip3,
  acs_poverty_zip3,
  acs_vacant_housing_zip3
)

# Reduce with only fraction/median columns
dfs_clean <- lapply(dfs, function(df) {
  df %>% select(zip3, year, starts_with("fraction"), starts_with("median"))
})

# Combine them by left join on zip3 and year
combined_data <- purrr::reduce(dfs_clean, function(x, y) left_join(x, y, by = c("zip3", "yea

#Flip as necessary
combined_data <- combined_data %>%
    mutate(
      fraction_high_school_edu = 1 - fraction_high_school_edu,
      median_income = 1 - (median_income - min(median_income, na.rm = TRUE)) /
                          (max(median_income, na.rm = TRUE) - min(median_income, na.rm =
    )
```

PCA for Deprivation Index

```
all_PC1_proportion <- list()
all_dep_weights <- list()
all_d <- list()

for (year_var in unique(combined_data$year)) {

  year_data <- combined_data %>% filter(year == year_var)

  # Run PCA
  pca <- year_data %>%
    na.omit() %>%
    select(-c(zip3, year)) %>%
    prcomp(center = TRUE, scale. = TRUE)

  # Get PC1 variance explained
  PC1_proportion <- summary(pca)$importance %>%
    as_tibble() %>%
    mutate(measure = row.names(summary(pca)$importance),
           year = year_var) %>%
    pivot_longer(cols = -c(measure, year), names_to = "component", values_to = "value") %>%
    filter(component == "PC1", measure == "Proportion of Variance")

  # Loadings
  dep_weights <- pca$rotation %>%
    as_tibble() %>%
    mutate(measure = row.names(pca$rotation),
           year = year_var) %>%
    pivot_longer(cols = starts_with("PC"), names_to = "component", values_to = "weight")

  # PCA scores scaled 0-1
  dep_index <- as_tibble(pca$x) %>%
  select(PC1) %>%
  mutate(deprivation_index = 1 - (PC1 - min(PC1)) / diff(range(PC1)),
         zip3 = year_data %>% na.omit() %>% pull(zip3),
         year = year_var) %>%
    select(zip3, year, deprivation_index)

  # Combine with original data
  d <- left_join(year_data, dep_index, by = c("zip3", "year"))
```

```
  all_PC1_proportion[[as.character(year_var)]] <- PC1_proportion
  all_dep_weights[[as.character(year_var)]] <- dep_weights
  all_d[[as.character(year_var)]] <- d
}

# Combine all years
PC1_proportion_all <- bind_rows(all_PC1_proportion)
dep_weights_all <- bind_rows(all_dep_weights)
d_all <- bind_rows(all_d)

write.table(d_all,file="ACS_data_2013_2023.csv", sep=",",row.names=FALSE, quote = F)
```