CLUS

In order to work with the data I will be using the tidyverse package. It is well designed to manipulate and work with large sets of data. The tidy census package is designed to download census data and automatically convert it into a tidy format for manipulation and model building.

```
pacman::p_load(tidyverse, tidycensus) #loading in and installing the packages
```

```
# The api key is necessary in order to get the tidy census package to work.
# This url can be used to obtain a census key in about 3 min.
# http://api.census.gov/data/key_signup.html
census_api_key("4164f675c6d35b601029ed68e31ec5150dac968f")

collect <- function(year, table, geometry = FALSE){
  temp <- get_acs(geography = 'county',
          state = "PA",
          year = year,
          table = table,
          cache_table = TRUE,
          geometry = geometry
          ) %>%
    rename(variable_c = variable) %>% #this is to prepare for adding in labels
    mutate(p_estimate = lead(estimate),
           p_moe = lead(moe), #this and the line above add 2 columns to replace the percentage rows
           year = year, #this adds in a year variable for easy filtering
           table = table) #this adds in a table variable for east filtering
  n <- seq(1, as.integer(count(temp)), 2)
  slice(temp, n) #this and the line above remove the now redundant percentage rows
}
```

First I create a function that collects the data from the American Community Survey and changes it slightly. The default setting makes it so every entry has 2 rows, one for the estimate of the value and its margin of error, and one for the percentage of the estimate of the value and its margin of error. The transformation I apply removes this second row and just adds another 2 columns to the data.

```
#An example of what the original function returns
get_acs(geography = 'county',
          state = "PA",
          year = 2014,
          table = "DP05",
          cache_table = TRUE,
)
```

```
## # A tibble: 10,854 x 5
##    GEOID NAME                       variable   estimate   moe
##    <chr> <chr>                      <chr>         <dbl> <dbl>
##  1 42001 Adams County, Pennsylvania DP05_0001  101566    NA
```

```
##  2 42001 Adams County, Pennsylvania DP05_0001P 101566    NA
##  3 42001 Adams County, Pennsylvania DP05_0002   50034   103
##  4 42001 Adams County, Pennsylvania DP05_0002P     49.3   0.1
##  5 42001 Adams County, Pennsylvania DP05_0003   51532   103
##  6 42001 Adams County, Pennsylvania DP05_0003P     50.7   0.1
##  7 42001 Adams County, Pennsylvania DP05_0004    5338     9
##  8 42001 Adams County, Pennsylvania DP05_0004P      5.3   0.1
##  9 42001 Adams County, Pennsylvania DP05_0005    5765   303
## 10 42001 Adams County, Pennsylvania DP05_0005P      5.7   0.3
## # ... with 10,844 more rows
```

```
#An example of what my modified version returns
collect(2014, "DP05")
```

```
## # A tibble: 5,427 x 9
##    GEOID NAME             variable_c estimate   moe p_estimate p_moe  year table
##    <chr> <chr>            <chr>         <dbl> <dbl>      <dbl> <dbl> <dbl> <chr>
##  1 42001 Adams County, P~ DP05_0001    101566    NA   101566    NA    2014 DP05
##  2 42001 Adams County, P~ DP05_0002     50034   103       49.3   0.1  2014 DP05
##  3 42001 Adams County, P~ DP05_0003     51532   103       50.7   0.1  2014 DP05
##  4 42001 Adams County, P~ DP05_0004      5338     9        5.3   0.1  2014 DP05
##  5 42001 Adams County, P~ DP05_0005      5765   303        5.7   0.3  2014 DP05
##  6 42001 Adams County, P~ DP05_0006      6514   294        6.4   0.3  2014 DP05
##  7 42001 Adams County, P~ DP05_0007      7275   157        7.2   0.2  2014 DP05
##  8 42001 Adams County, P~ DP05_0008      6753   113        6.6   0.1  2014 DP05
##  9 42001 Adams County, P~ DP05_0009     10358    96       10.2   0.1  2014 DP05
## 10 42001 Adams County, P~ DP05_0010     12353   107       12.2   0.1  2014 DP05
## # ... with 5,417 more rows
```

Here I am just running the function a bunch of times to collect all of the versions of the data that we want from all the different years and variables. If someone wants to optimize this code in the future it could probably be similar with a map function or a for loop.

```
#Housing Characteristics
HC_2014 <- collect(2014, "DP04")
HC_2015 <- collect(2015, "DP04")
HC_2016 <- collect(2016, "DP04")
HC_2017 <- collect(2017, "DP04")
HC_2018 <- collect(2018, "DP04")
HC_2019 <- collect(2019, "DP04")

#Demographic and Housing Characteristics

DHC_2014 <- collect(2014, "DP05")
DHC_2015 <- collect(2015, "DP05")
DHC_2016 <- collect(2016, "DP05")
DHC_2017 <- collect(2017, "DP05")
DHC_2018 <- collect(2018, "DP05")
DHC_2019 <- collect(2019, "DP05")

#Social Characteristics
SC_2014 <- collect(2014, "DP02")
SC_2015 <- collect(2015, "DP02")
```

```
SC_2016 <- collect(2016, "DP02")
SC_2017 <- collect(2017, "DP02")
SC_2018 <- collect(2018, "DP02")
SC_2019 <- collect(2019, "DP02")

#Economic characteristics
EC_2014 <- collect(2014, "DP03")
EC_2015 <- collect(2015, "DP03")
EC_2016 <- collect(2016, "DP03")
EC_2017 <- collect(2017, "DP03")
EC_2018 <- collect(2018, "DP03")
EC_2019 <- collect(2019, "DP03")
```

Here I am creating another function to load in some meta data about the American Community survey that I downloaded from the census. This enables us to get some labels about what the variables are in the tables that we created earlier. I use some regexes to make them more readable, however this makes some of the labels harder to understand as it removes some clarification. If you have the census table open in a browser it makes it much easier to understand with the combination, however more work could be done here to make it easier to use.

```
get_labels <- function(x) {
  df <-  read_csv(x, col_names = FALSE, skip = 2)

  df <- df %>%
    slice(seq(1, as.integer(count(df)), 4))
  df %>%
    mutate(variable_c = str_remove(df$X1,"E"),
           variable_l = str_remove(df$X2, ".*!!")) %>%
    select(variable_c, variable_l)
}

DP02_names <- get_labels("ACSDP5Y2019.DP02_metadata_2021-03-09T174642.csv")
DP03_names <- get_labels("ACSDP5Y2019.DP03_metadata_2021-03-09T174642.csv")
DP04_names <- get_labels("ACSDP5Y2019.DP04_metadata_2021-03-09T174642.csv")
DP05_names <- get_labels("ACSDP5Y2019.DP05_metadata_2021-03-09T174642.csv")

#This line combines the labels that we created from the earlier lines of code
#into one large data set that contains all of the labels.

census_labels <- bind_rows(DP02_names, DP03_names, DP04_names, DP05_names)
```

Here we create the ACS dataset which combines all of the datasets created earlier into one large data set, and also creates a column that uses the labels created earlier to aid in readability of the dataset.

```
ACS <- left_join(bind_rows(SC_2014, SC_2015, SC_2016, SC_2017, SC_2018, SC_2019,
          EC_2014, EC_2015, EC_2016, EC_2017, EC_2018, EC_2019,
          HC_2014, HC_2015, HC_2016, HC_2017, HC_2018, HC_2019,
          DHC_2014, DHC_2015, DHC_2016, DHC_2017, DHC_2018, DHC_2019),
          census_labels, by = "variable_c")
ACS <- ACS %>%
  mutate(state = "PA",
         county = str_remove(ACS$NAME, "[[:blank:]]County.*")) %>%
  rename(geoid = GEOID) %>%
```

```
  select(geoid, state, county, year, table, variable_c, variable_l, estimate, moe, p_estimate, p_moe)

#Here is what the current final form of the data set looks like
ACS
```

```
## # A tibble: 208,169 x 11
##    geoid state county  year table variable_c variable_l estimate   moe
##    <chr> <chr> <chr>  <dbl> <chr> <chr>      <chr>         <dbl> <dbl>
##  1 42001 PA    Adams   2014 DP02  DP02_0001  Total hou~    37956   426
##  2 42001 PA    Adams   2014 DP02  DP02_0002  Married-c~    27152   430
##  3 42001 PA    Adams   2014 DP02  DP02_0003  With own ~    10395   386
##  4 42001 PA    Adams   2014 DP02  DP02_0004  Cohabitin~    21925   453
##  5 42001 PA    Adams   2014 DP02  DP02_0005  With own ~     7577   353
##  6 42001 PA    Adams   2014 DP02  DP02_0006  Male hous~     1607   225
##  7 42001 PA    Adams   2014 DP02  DP02_0007  With own ~      761   161
##  8 42001 PA    Adams   2014 DP02  DP02_0008  Household~     3620   367
##  9 42001 PA    Adams   2014 DP02  DP02_0009  65 years ~     2057   265
## 10 42001 PA    Adams   2014 DP02  DP02_0010  Female ho~    10804   482
## # ... with 208,159 more rows, and 2 more variables: p_estimate <dbl>,
## #   p_moe <dbl>
```

```
#Here is a summery of the current final form of the data set
summary(ACS)
```

```
##     geoid              state              county              year
##  Length:208169      Length:208169      Length:208169      Min.   :2014
##  Class :character   Class :character   Class :character   1st Qu.:2015
##  Mode  :character   Mode  :character   Mode  :character   Median :2017
##                                                           Mean   :2017
##                                                           3rd Qu.:2018
##                                                           Max.   :2019
##
##     table            variable_c         variable_l          estimate
##  Length:208169      Length:208169      Length:208169      Min.   :      0
##  Class :character   Class :character   Class :character   1st Qu.:    519
##  Mode  :character   Mode  :character   Mode  :character   Median :   3451
##                                                           Mean   :  27513
##                                                           3rd Qu.:  18803
##                                                           Max.   :1579075
##                                                           NA's   :9179
##       moe             p_estimate           p_moe
##  Min.   :    0.01   Min.   :      0.0   Min.   :  0.10
##  1st Qu.:   85.00   1st Qu.:      3.0   1st Qu.:  0.20
##  Median :  230.00   Median :     12.3   Median :  0.60
##  Mean   :  400.60   Mean   :  11728.0   Mean   :  1.27
##  3rd Qu.:  508.00   3rd Qu.:     50.6   3rd Qu.:  1.30
##  Max.   :11702.00   Max.   :1579075.0   Max.   :100.00
##  NA's   :12906      NA's   :16482       NA's   :45072
```

Below here is just of copy past of the variable that were asked for in the document. This is just to aid my work flow

Selected housing characteristics: housing occupancy (occupied housing units, vacant housing units, home-owner vacancy rate, rental vacancy rate); Units in structure; year structure built; # of rooms; #of bedrooms;

housing tenure (owner vs. renter); year householder moved in; house heating fuel; plumbing, kitchen, and telephone service; occupants per room; mortgage status;

Race and ethnicity (when downloading the table, keep estimate and percent) Population and people (age and sex) Selected social characteristics (household by type, marital status, educational attainment, disability status, U.S. citizenship status and year of entry, computers and internet use) Financial characteristics (household income, monthly housing costs, monthly housing costs as a percentage of household income, mortgage status, ratio of value to household income, real estate taxes)