

R Notebook

This is an R Markdown (<http://rmarkdown.rstudio.com>) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Part 1. installing libraries and packages

```
#Downloading Packages
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(ggplot2)  
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.0.5
```

```
## Linking to GEOS 3.9.0, GDAL 3.2.1, PROJ 7.2.1
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

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

```
## v tibble  3.1.1      v dplyr   1.0.5  
## v tidyr   1.1.3      v stringr 1.4.0  
## v readr   1.4.0      v forcats 0.5.1  
## v purrr   0.3.4
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

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

```
library(tidycensus)
```

```
## Warning: package 'tidycensus' was built under R version 4.0.4
```

```
library(mapview)
```

```
## Warning: package 'mapview' was built under R version 4.0.5
```

```
## GDAL version >= 3.1.0 | setting mapviewOptions(fgb = TRUE)
```

```
library(stringr)  
library(scales)
```

```
##  
## Attaching package: 'scales'
```

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

```
## The following object is masked from 'package:readr':  
##  
## col_factor
```

```
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.0.5
```

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

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

```
library(readr)
```

Part 2. Installing Census API Key

```
library(tidycensus)
options(tigris_class = "sf")
options(tigris_use_cache = TRUE)
census_api_key("a956681143e4acebe4095b02c107771b66635f8b", overwrite = TRUE, install = TRUE)
```

```
## Your original .Renvirom will be backed up and stored in your R HOME directory if needed.
```

```
## Your API key has been stored in your .Renvirom and can be accessed by Sys.getenv("CENSUS_API_KEY").
## To use now, restart R or run `readRenvirom("~/Renvirom")`
```

```
## [1] "a956681143e4acebe4095b02c107771b66635f8b"
```

```
readRenvirom("~/Renvirom")
```

Part 3. Bringing Full Census Datalist into R

```
acs_variable_list = load_variables(2019,"acs5", cache= TRUE)

write.csv(acs_variable_list,'acs_variable_list_2018.csv', row.names = FALSE)

v16 = load_variables(year = 2019, dataset = "acs5", cache = TRUE)

v16
```

name <chr>	label <chr>	
B01001_001	Estimate!!Total:	
B01001_002	Estimate!!Total:!!Male:	
B01001_003	Estimate!!Total:!!Male:!!Under 5 years	
B01001_004	Estimate!!Total:!!Male:!!5 to 9 years	
B01001_005	Estimate!!Total:!!Male:!!10 to 14 years	
B01001_006	Estimate!!Total:!!Male:!!15 to 17 years	
B01001_007	Estimate!!Total:!!Male:!!18 and 19 years	
B01001_008	Estimate!!Total:!!Male:!!20 years	
B01001_009	Estimate!!Total:!!Male:!!21 years	
B01001_010	Estimate!!Total:!!Male:!!22 to 24 years	
1-10 of 10,000 rows 1-2 of 3 columns		Previous 1 2 3 4 5 6 ... 1000 Next

```
drop(BaltCitytractdata2019)
```

Part 4. Getting Census Data for County

```
BaltCitytractdata2019 = get_acs(geography = "tract", year=2019, state = "MD", survey="acs5", county = "Baltimore City",
                                variables = c("Total Pop" = "B01003_001",
"Black Pop" = "B01001B_001",
"Black Inc" = "B19013B_001",
"White Pop" = "B01001A_001",
"White Inc" = "B19013A_001",
"Incunder10k" = "B19001_002",
"10kto15k" = "B19001_003",
"15kto20k" = "B19001_004",
"20kto25k" = "B19001_005",
"25kto30k" = "B19001_006",
"30kto35k" = "B19001_007",
"35kto40k" = "B19001_008",
"40kto45k" = "B19001_009",
"45kto50k" = "B19001_010",
"50kto60k" = "B19001_011",
"60kto75k" = "B19001_012",
"75kto100k" = "B19001_013",
"100kto125k" = "B19001_014",
"125kto150k" = "B19001_015",
"150kto200k" = "B19001_016",
"incover200k" = "B19001_017",
"monthlycost" = "B25105_001",
"highschool" = "B15003_017",
"Associates" = "B15003_021",
"Bachelors" = "B15003_022",
"Masters" = "B15003_023",
"professional" = "B15003_024",
"Doctorate" = "B15003_025",
"employed" = "B23025_004",
"unemployed" = "B23025_005",
"laborforce" = "B23025_003",
"Gini" = "B19083_001",
"povertypop" = "B17001_002",
"medHouseprice" = "B25077_001",
"noncitizen" = "B05001_006",
"med_age" = "B01002_001"),
                                geometry = TRUE,
                                output = "wide") %>% clean_names()
```

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

Part 5. Creating Variables for Analysis

#Calculating number of college educated people in census tract

BaltCitytractdata2019\$educated =

```
((BaltCitytractdata2019$bachelors_e +
  BaltCitytractdata2019$associates_e +
  BaltCitytractdata2019$masters_e +
  BaltCitytractdata2019$doctorate_e))
```

BaltCitytractdata2019\$educatedprop =

```
BaltCitytractdata2019$educated / BaltCitytractdata2019$total_pop_e
```

BaltCitytractdata2019\$higheducated =

```
(BaltCitytractdata2019$masters_e + BaltCitytractdata2019$doctorate_e +
  BaltCitytractdata2019$professional_e)
```

BaltCitytractdata2019\$higheducatedprop =

```
(BaltCitytractdata2019$higheducated / BaltCitytractdata2019$total_pop_e)
```

#Calculating employment and unemployment rate

BaltCitytractdata2019\$unemployedrate = BaltCitytractdata2019\$unemployed_e / BaltCitytractdata2019\$laborforce_e

BaltCitytractdata2019\$employedrate =

```
BaltCitytractdata2019$employed_e / BaltCitytractdata2019$laborforce_e
```

#Calculating number of people Making over and under 30k (poor population)

BaltCitytractdata2019\$popunder30k= ((BaltCitytractdata2019\$incunder10k_e + BaltCitytractdata2019\$x10kto15k_e + BaltCitytractdata2019\$x15kto20k_e + BaltCitytractdata2019\$x25kto30k_e))

BaltCitytractdata2019\$popover30k=

```
((BaltCitytractdata2019$x35kto40k_e + BaltCitytractdata2019$x45kto50k_e + BaltCitytractdata2019$x50kto60k_e + BaltCitytractdata2019$x60kto75k_e +
  BaltCitytractdata2019$x75kto100k_e +
  BaltCitytractdata2019$x100kto125k_e +
  BaltCitytractdata2019$x125kto150k_e +
  BaltCitytractdata2019$x150kto200k_e +
  BaltCitytractdata2019$incover200k_e))
```

#Calculating population making over and under 150k (rich population)

BaltCitytractdata2019\$popover150k =

```
(BaltCitytractdata2019$incover200k_e + BaltCitytractdata2019$x150kto200k_e)
```

```

BaltCitytractdata2019$popunder150k =
  (BaltCitytractdata2019$x35kto40k_e +
   BaltCitytractdata2019$x45kto50k_e +
   BaltCitytractdata2019$x50kto60k_e +
   BaltCitytractdata2019$x60kto75k_e +
   BaltCitytractdata2019$x75kto100k_e +
   BaltCitytractdata2019$x100kto125k_e +
   BaltCitytractdata2019$x125kto150k_e +
   BaltCitytractdata2019$incunder10k_e +
   BaltCitytractdata2019$x10kto15k_e +
   BaltCitytractdata2019$x15kto20k_e +
   BaltCitytractdata2019$x25kto30k_e)

#Calculating total rich and total poor populations for county
BaltCitytractdata2019$poorcountypop = sum(BaltCitytractdata2019$popunder150k)

BaltCitytractdata2019$richcountypop =
  sum(BaltCitytractdata2019$popover150k)

#Calculating rich and poor population ratios for census tracts

BaltCitytractdata2019$nonpoorprop =      (BaltCitytractdata2019$popover30k/
  (BaltCitytractdata2019$popover30k + BaltCitytractdata2019$popunder30k))

BaltCitytractdata2019$poorprop =      (BaltCitytractdata2019$popunder30k/
  (BaltCitytractdata2019$popunder30k + BaltCitytractdata2019$popover30k))

BaltCitytractdata2019$richprop =
  (BaltCitytractdata2019$popover150k/
   (BaltCitytractdata2019$popover150k + BaltCitytractdata2019$popunder150k))

BaltCitytractdata2019$nonrichprop =
  (BaltCitytractdata2019$popunder150k/
   (BaltCitytractdata2019$popunder150k + BaltCitytractdata2019$popover150k))

```

Part 6. Turning County Census data into QGIS Map

```
st_write(BaltCitytractdata2019, "BaltCityACS2019.geojson")
```

After this was done, the process performed in Part 4 and Part 5 were repeated for each of the 7 counties in the Baltimore Metropolitan Statistical Area*

...

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.