Soil Data

John Buehler

4/20/2022

```
library(tidycensus)
library(dplyr)
library(tidyverse)
library(ggplot2)
library(tigris)
library(patchwork)
library(sf)
library(ggspatial)
library(units)
library(readr)
library(mapview)
library(rmapshaper)
library(corrr)
library(areal)
```

Montgomery County

Import Data

The primary goal is to see if the distribution of “Prime Farmland” soil correlates to median income. However, I also went ahead and gathered education, commute, and property size data from the American Community Survey.

```
#MoCo will be the geometry used to make area calculations
MoCo <- get_acs(
  geography = "tract",
  county = "Montgomery",
  state = "MD",
  variable = "B19013_001",
  geometry = TRUE,
  cb = TRUE
)
MoCo <- MoCo %>%
  st_transform(crs = 3857)
#renames "estimate" column
names(MoCo)[4] <- "median_inc"
```

```
#MoCo_com will be the receptacle for all variables
MoCo_com <- get_acs(geography = "tract",
                           variables = c("median_inc" = "B19013_001"),
                           year = 2020,
                           survey = "acs5",
                           state = c(24),
                           county = "Montgomery",
                           geometry = TRUE,
                           output = "wide")
MoCo_com <- MoCo_com %>%
  dplyr::select(GEOID, median_incE)%>%
  st_transform(crs = 3857)
```

```
#retrives Census data on educational attainment for each tract in Montgomery county.
#computes % of individuals who attained less than (NoDip) and equal to or more (higher_edu) than
a high school diploma
MoCo_edu <- get_acs(
  variables = c("total_edu" = "B15003_001",
                "Na_edu" = "B15003_002",
                "NS_edu" = "B15003_003",
                "kg_edu" = "B15003_004",
                "1st_edu" = "B15003_005",
                "2nd_edu" = "B15003_006",
                "3rd_ed" = "B15003_007",
                "4th_edu" = "B15003_008",
                "5th_edu" = "B15003_009",
                "6th_edu" = "B15003_010",
                "7th_edu" = "B15003_011",
                "8th_edu" = "B15003_012",
                "9th_edu" = "B15003_013",
                "10th_edu" = "B15003_014",
                "11th_edu" = "B15003_015",
                "12th_edu" = "B15003_016",
                "hsDip_edu" = "B15003_017",
                "GED_edu" = "B15003_018",
                "someC_edu" = "B15003_019",
                "someCNoD_edu"= "B15003_020",
                "AsD_edu" = "B15003_021",
                "BacD_edu" = "B15003_022",
                "MasD_edu" = "B15003_023",
                "ProfD_edu" = "B15003_024",
                "DocD_edu" = "B15003_025"
                ),
  year = 2020,
  geography = "tract",
  survey = "acs5",
  state = c(24),
  county = "Montgomery",
  geometry = TRUE,
  output = "wide")
MoCo_edu$NoDip <- (MoCo_edu$Na_eduE + MoCo_edu$NS_eduE + MoCo_edu$kg_eduE + MoCo_edu$`1st_eduE`+
MoCo_edu$`2nd_eduE` + MoCo_edu$`3rd_edE` + MoCo_edu$`4th_eduE`+ MoCo_edu$`5th_eduE`+ MoCo_edu$`6
th_eduE`+ MoCo_edu$`7th_eduE`+ MoCo_edu$`8th_eduE`+ MoCo_edu$`9th_eduE`+ MoCo_edu$`10th_eduE`+ M
oCo_edu$`11th_eduE`+ MoCo_edu$`12th_eduE`) / MoCo_edu$total_eduE

MoCo_edu$higher_edu <- (MoCo_edu$hsDip_eduE + MoCo_edu$GED_eduE + MoCo_edu$someC_eduE + MoCo_edu
$someCNoD_eduE + MoCo_edu$AsD_eduE + MoCo_edu$BacD_eduE + MoCo_edu$MasD_eduE + MoCo_edu$ProfD_ed
uE + MoCo_edu$DocD_eduE) / MoCo_edu$total_eduE

MoCo_edu <- MoCo_edu %>%
  dplyr::select(GEOID, NoDip, higher_edu,) %>%
  st_drop_geometry()
```

```r
#gathers data on the size of houses within each tract
MoCo_rooms <- get_acs(
  variable = c(
    "median_rms" = "B25018_001"),
  year = 2020,
  geography = "tract",
  survey = "acs5",
  state = "MD",
  county = "Montgomery",
  geometry = TRUE,
  output = "wide")

MoCo_rooms <- MoCo_rooms %>%
  dplyr::select(GEOID, median_rmsE) %>%
  st_drop_geometry()
```

```
#gathers commute duration data
MD_Commutes <- get_acs(
  variables = c("less_10" = "B08135_002",
                "10to14" = "B08135_003",
                "15to19" = "B08135_004",
                "20to24" = "B08135_005",
                "25to29" = "B08135_006",
                "30to34" = "B08135_007",
                "35to44" = "B08135_008",
                "45to59" = "B08135_009",
                "60or_more" = "B08135_010"),
  year = 2020,
  geography = "tract",
  survey = "acs5",
  county = "Montgomery",
  state = c(24),
  geometry = TRUE,
  output = "wide")

MD_Commutes$total <- MD_Commutes$less_10E + MD_Commutes$`10to14E` + MD_Commutes$`15to19E` + MD_Commutes$`20to24E` + MD_Commutes$`25to29E` + MD_Commutes$`30to34E` + MD_Commutes$`35to44E` + MD_Commutes$`45to59E`+ MD_Commutes$`60or_moreE`

MD_Commutes$less_30 <- (MD_Commutes$less_10E + MD_Commutes$`10to14E`+ MD_Commutes$`15to19E`+ MD_Commutes$`20to24E`+ MD_Commutes$`25to29E`) / MD_Commutes$total

MD_Commutes$`30or_more` <- (MD_Commutes$`30to34E` + MD_Commutes$`35to44E`+ MD_Commutes$`45to59E` + MD_Commutes$`60or_moreE`) / MD_Commutes$total

MD_Commutes$less_30[is.na(MD_Commutes$less_30)] <- 0
MD_Commutes$`30or_more`[is.na(MD_Commutes$`30or_more`)] <-0

MoCo_Commutes <- MD_Commutes %>%
  dplyr::select(GEOID, less_30, `30or_more`) %>%
  st_drop_geometry()
```

```
#joins the gathered data to the MoCo_com receptacle
MoCo_com <- left_join(x = MoCo_com, y = MoCo_edu, by = "GEOID")
MoCo_com <- left_join(x = MoCo_com, y = MoCo_rooms, by = "GEOID")
MoCo_com <- left_join(x = MoCo_com, y = MoCo_Commutes, by = "GEOID")
```

```
#import National Resource Conservation Service (NRCS) and re-project as Web-Mercator (units in meters)
soil <- st_read("F:/Soil/data/MD031/spatial/soilmu_a_md031.shp") %>%
  st_transform(crs = 3857)
```

```
## Reading layer `soilmu_a_md031' from data source
##   `F:\Soil\data\MD031\spatial\soilmu_a_md031.shp' using driver `ESRI Shapefile'
## Simple feature collection with 14415 features and 4 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -77.52727 ymin: 38.93447 xmax: -76.88865 ymax: 39.35414
## Geodetic CRS:  WGS 84
```

```
#import building footrint shapefile from 2020. projected in Web Mercator
MoCo_Buildings <- st_read("F:/Soil/data/MoCo_Buildings_Proj_Clip.shp")
```

```
## Reading layer `MoCo_Buildings_Proj_Clip' from data source
##   `F:\Soil\data\MoCo_Buildings_Proj_Clip.shp' using driver `ESRI Shapefile'
## Simple feature collection with 372702 features and 14 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -8629780 ymin: 4712773 xmax: -8560333 ymax: 4772297
## Projected CRS: WGS 84 / Pseudo-Mercator
```

```
#import shapefile with bodies of water in MoCo. projected in NAD83. reproject to Web Mercator
MoCo_water <- st_read("F:/Soil/data/MoCo_water.shp")
```

```
## Reading layer `MoCo_water' from data source `F:\Soil\data\MoCo_water.shp' using driver `ESRI
Shapefile'
## Simple feature collection with 3854 features and 7 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 1159993 ymin: 461432.6 xmax: 1345788 ymax: 617418.9
## Projected CRS: NAD83 / Maryland (ftUS)
```

```
MoCo_water <- MoCo_water %>%
  st_transform(crs=3857)
```

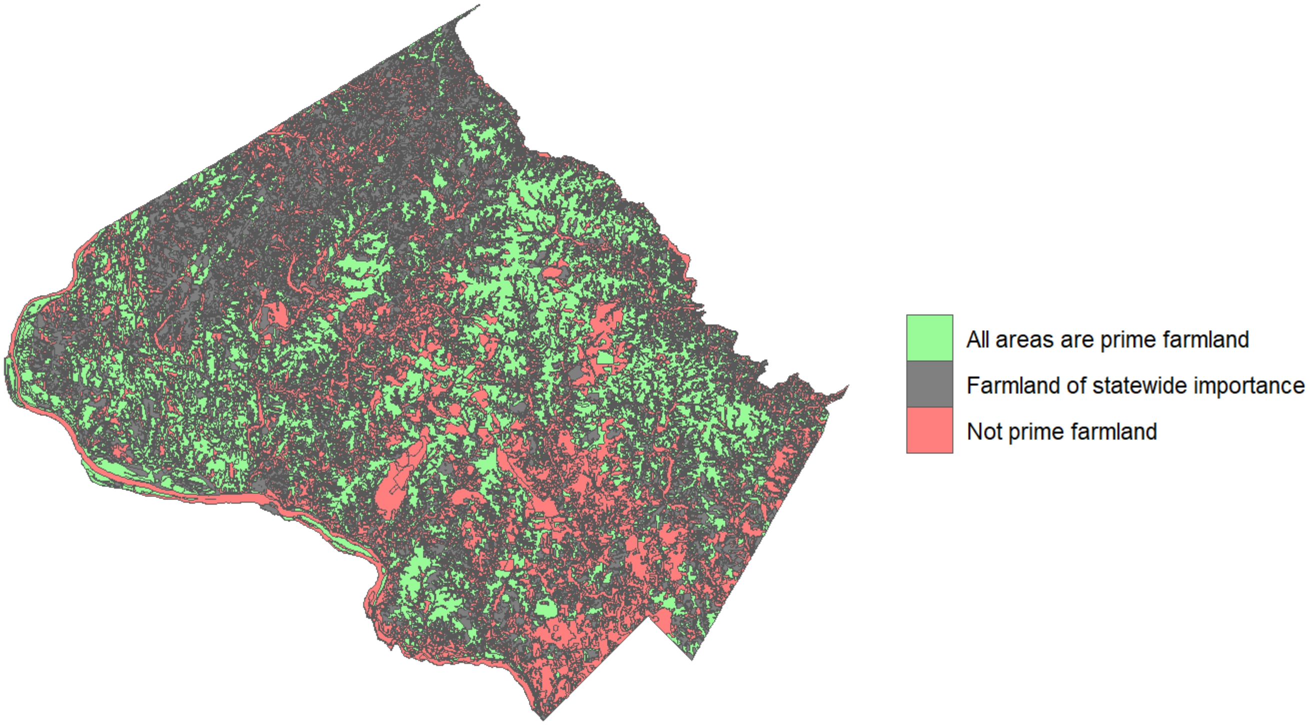
```
#import mapunit.txt ("|" delimited), included in the "tabular" folder of data downloaded from NR
CS. Using dplyr::select to remove unnecessary columns.
map_key <- read_delim("F:/Soil/data/MD031/tabular/mapunit.txt",
    delim = "|", escape_double = FALSE, col_names = FALSE,
    trim_ws = TRUE)
map_key <- map_key %>%
  dplyr::select(X1, X2, X12)
```

```
#renaming column names with base r
names(map_key)[1] <- "MUSYM"
names(map_key)[2] <- "Type"
names(map_key)[3] <- "Farmland"
```

```
#joins soil and map_key tables to make a complete soil dataset that specifies which areas are prime farmland or "Not prime."
soil_com <- left_join(x = soil, y = map_key, by = "MUSYM")

gg4 <- ggplot(soil_com)+
  geom_sf(aes(fill = Farmland), lwd = 0)+
  ggtitle(label = "Soil Classifications", subtitle = "'Prime' meets NRCS physical standards, other classes do not")+
  scale_fill_manual(values = c("#98fb98", "#808080", "#FF7F7F"))+
  theme_void()+
  labs(fill = "")
gg4
```

Soil Classifications

'Prime' meets NRCS physical standards, other classes do not


🟩 All areas are prime farmland
⬛ Farmland of statewide importance
🟥 Not prime farmland

```
#remove geometry that contains "state wide importance" and "Not prime farmland"
soil_prime2 <- soil_com %>%
  filter(Farmland == c("All areas are prime farmland"))
```

Erase Unusable Areas From MoCo Tracts and Prime Soil

```
MoCo_erase_b <- rmapshaper::ms_erase(MoCo_com, MoCo_Buildings)
```

```
MoCo_erase_bw <- rmapshaper::ms_erase(MoCo_erase_b, MoCo_water)
```

Alternative to Rmapshaper

st_erase = function(x, y) st_difference(x, st_union(st_combine(y))) | tracts_prime <- st_erase(MoCo_com, MoCo_Buildings) **takes forever tho**

Calculate Usable Tract Area

```
MoCo_erase_bw$Usable_Area <- st_area(MoCo_erase_bw)
```

```
MoCo_erase_bw2 <- MoCo_erase_bw %>%
  st_drop_geometry() %>%
  dplyr::select(GEOID, Usable_Area) #bw2 will be used later
```

```
MoCo_erase_bw2 <- sf::st_buffer(MoCo_erase_bw, dist = 0) # added this line because of issues rel
ated to st_intersect. idk what happened.
```

Find Where Usable Tract Area Intersects Prime Soil

```
#create a shapefile that shows where MoCo tracts intersects with soil_prime polygons
tracts_prime_imp <- st_intersection(x = MoCo_erase_bw, y = soil_prime2)
```

```
#replicate "dissolve" tool in Arc using dplyr::group_by
tracts_prime_imp_dis <- tracts_prime_imp %>% group_by(GEOID) %>% summarize()
```

```
#calculate prime farmland area for each tract (GEOID)
tracts_prime_imp_dis$prime_imp_area <- st_area(tracts_prime_imp_dis)
```

```
#remove geometry from prime polygon so the data can be joined to complete tract polygons
prime_imp_area <- tracts_prime_imp_dis %>%
  st_drop_geometry()
```

Calculate Ratio of Prime Area to Usable Tract Area

```
#joins tables so that the tract table has prime area for each tract
MoCo_prime_join <- left_join(x = MoCo_com, y = prime_imp_area, by = "GEOID")
```

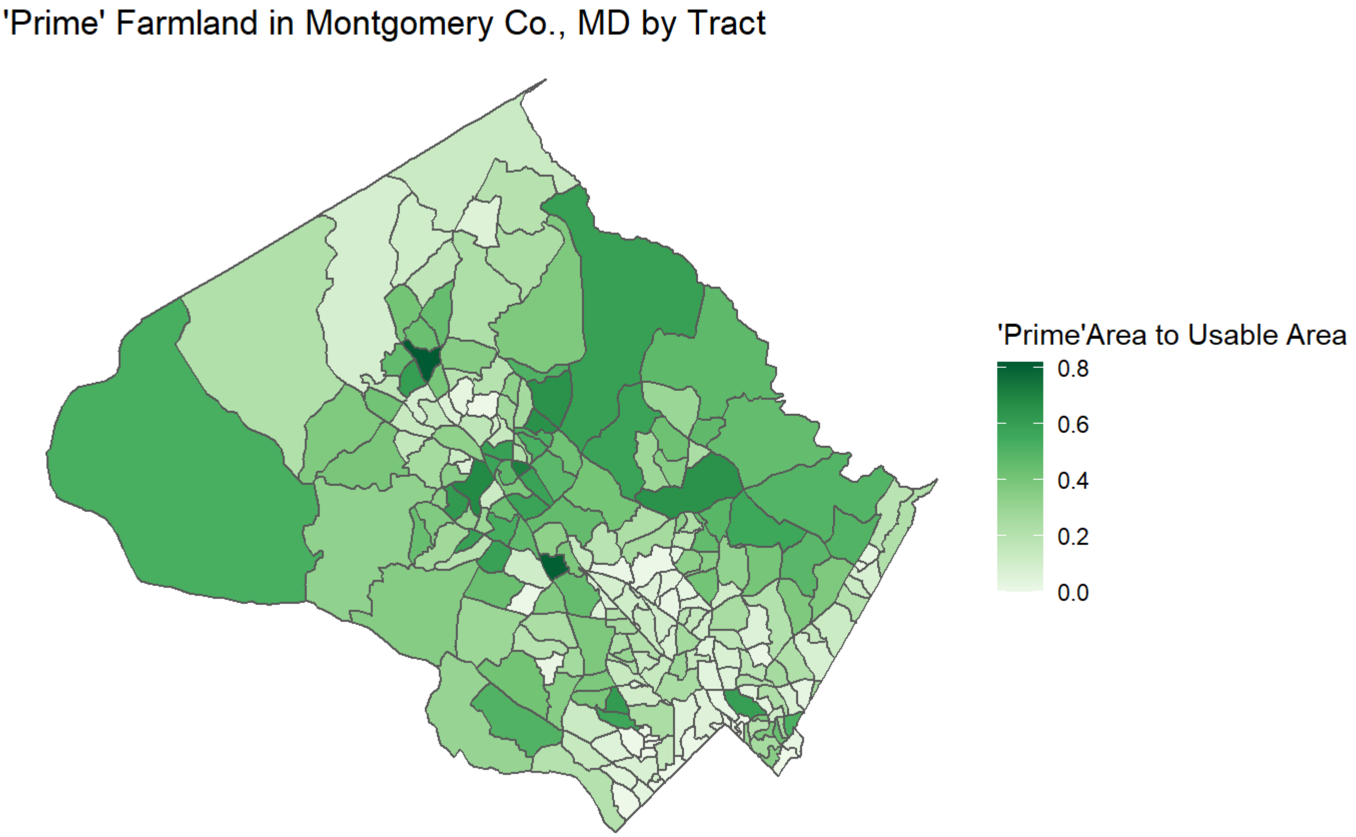
```
MoCo_erase_bw2 = st_drop_geometry(MoCo_erase_bw)
MoCo_prime_usable_join <- left_join(x = MoCo_prime_join, y = MoCo_erase_bw2, by = "GEOID")
```

```
#calculate the ratio of prime area to usable tract area
MoCo_prime_usable_join$Prime_to_Area <- MoCo_prime_usable_join$prime_imp_area/ MoCo_prime_usable_join$Usable_Area
#makes a copy of the Prime_to_Area field without units
MoCo_prime_usable_join$Prime_to_Area1 <- as.numeric(MoCo_prime_usable_join$Prime_to_Area)
#replaces NA with 0
MoCo_prime_usable_join$Prime_to_Area1[is.na(MoCo_prime_usable_join$Prime_to_Area1)] <- 0
MoCo_final <- MoCo_prime_usable_join #fixing terrible object name
```

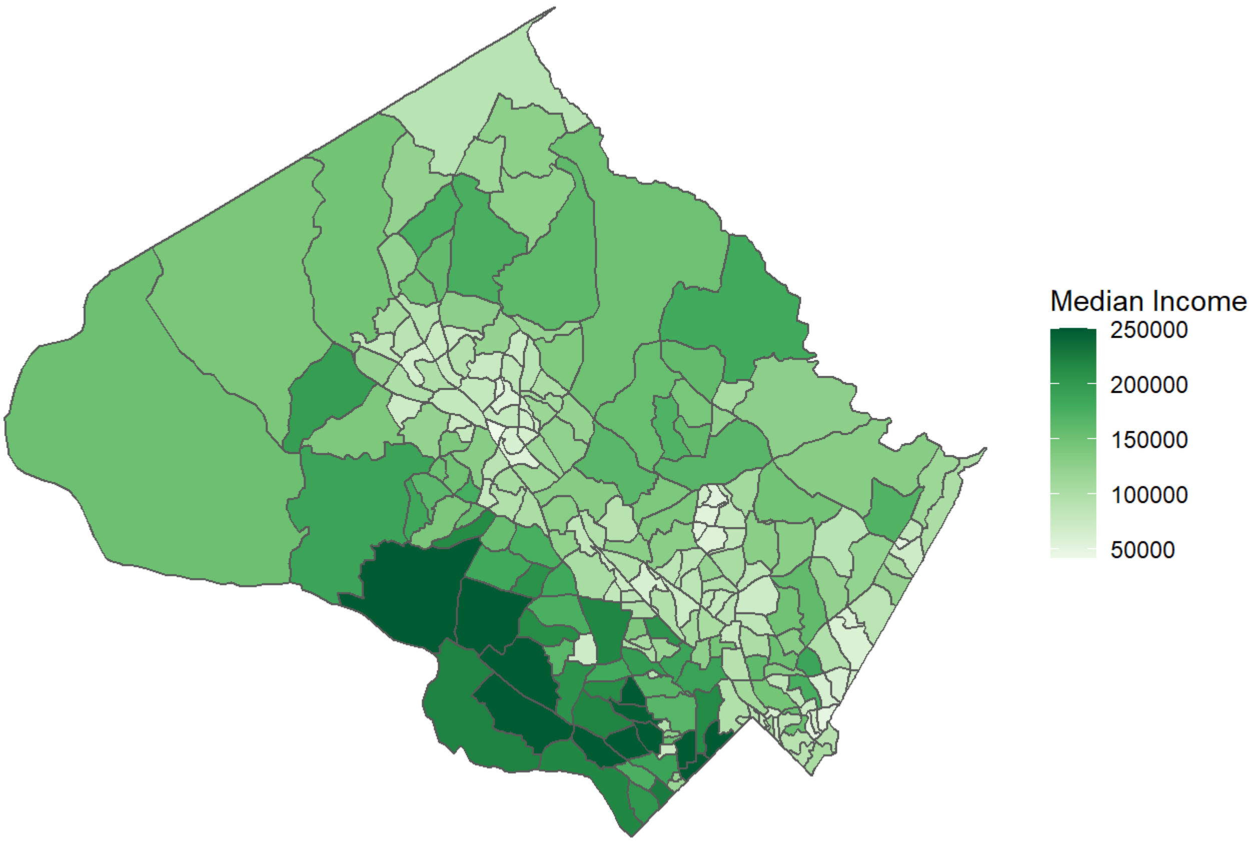
Last Second Column Editing

```
names(MoCo_final)[2] <- "med_inc"
names(MoCo_final)[3] <- "No_Dip"
names(MoCo_final)[4] <- "Hi_Edu"
names(MoCo_final)[5] <- "med_rms"
names(MoCo_final)[6] <- "und_30"
names(MoCo_final)[7] <- "mr_30"

MoCo_final <- MoCo_final %>%
  dplyr::select(GEOID, med_inc, No_Dip, Hi_Edu, med_rms, und_30, mr_30, Prime_to_Area1)
```

```
gg1 <- ggplot(MoCo_final)+
  geom_sf(aes(fill = Prime_to_Area1))+
  scale_fill_distiller(palette = "Greens", direction = 1)+
  ggtitle(label = "'Prime' Farmland in Montgomery Co., MD by Tract" )+
  theme_void()+
  labs(fill = "'Prime'Area to Usable Area")
gg2 <- ggplot(MoCo_final)+
  geom_sf(aes(fill = med_inc))+
  scale_fill_distiller(palette = "Greens", direction = 1)+
  ggtitle(label = "Median Household Income in Montgomery Co., MD, by Tract")+
  theme_void()+
  labs(fill = "Median Income")

gg1
```


'Prime' Farmland in Montgomery Co., MD by Tract
'Prime'Area to Usable Area
0.8
0.6
0.4
0.2
0.0

```
gg2
```



Median Household Income in Montgomery Co., MD, by Tract

Areal Interpolation

```
MoCo_grid <- st_make_grid(MoCo,
                          2 * 1000,
                          crs = 3857,
                          what = "polygon",
                          square = FALSE)
MoCo_grid <- st_sf(index = 1:length(lengths(MoCo_grid)), MoCo_grid)
MoCo_shape <- st_union(MoCo)
MoCo_grid.intersects <- st_intersects(MoCo_shape, MoCo_grid)
MoCo_grid.subset <- MoCo_grid[MoCo_grid.intersects[[1]],]
```

```
ar_validate(source = MoCo_final, target = MoCo_grid.subset, varList = "Prime_to_Area1", method =
"aw")
```

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

```
ar_validate(source = MoCo_final, target = MoCo_grid.subset, varList = "med_inc", method = "aw")
```

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

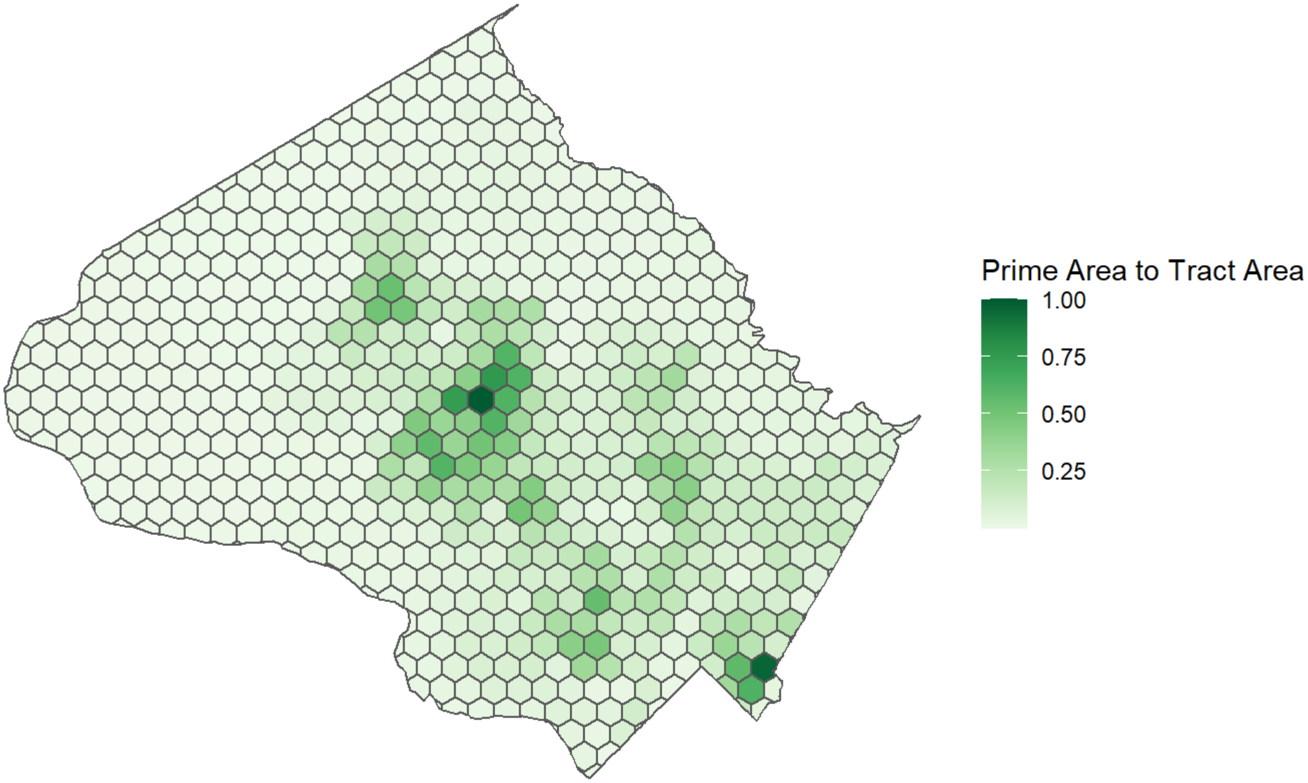
```
MoCo_grid_prime <- aw_interpolate(MoCo_grid.subset, tid = index, source = MoCo_final, sid = "GEOID", output = "sf", weight = "sum", extensive = "Prime_to_Area1")

MoCo_grid_inc <- aw_interpolate(MoCo_grid.subset, tid = index, source = MoCo_final, sid = "GEOID", output = "sf", weight = "sum", extensive = "med_inc")

MoCo_grid_prime.crop <- st_intersection(MoCo_grid_prime, MoCo_shape)
```

```
## Warning: attribute variables are assumed to be spatially constant throughout all
## geometries
```

```
MoCo_grid_inc.crop <- st_intersection(MoCo_grid_inc, MoCo_shape)
```

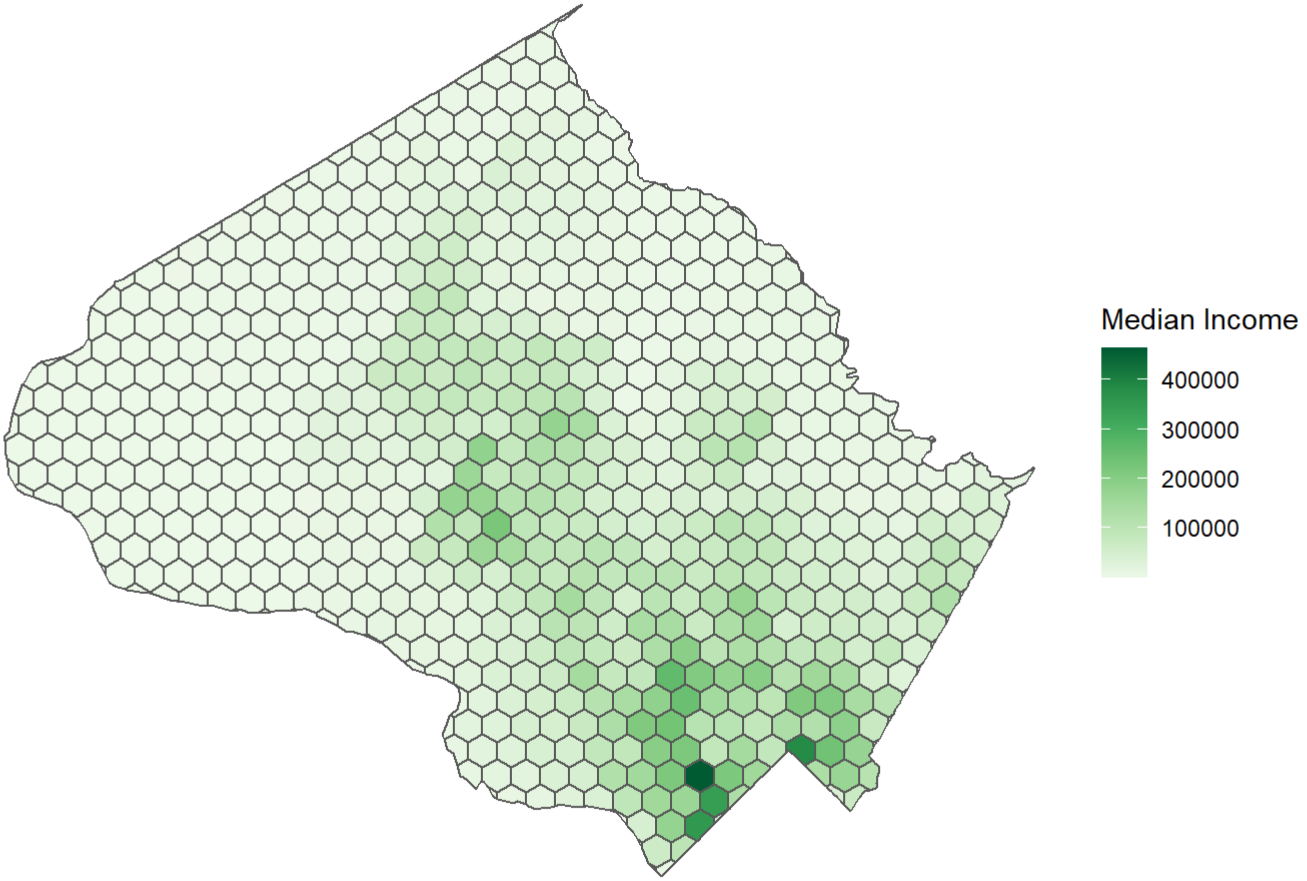
```
## Warning: attribute variables are assumed to be spatially constant throughout all
## geometries
```

```
ai1 <- ggplot( MoCo_grid_prime.crop)+
  geom_sf(aes(fill = Prime_to_Area1))+
  scale_fill_distiller(palette = "Greens", direction = 1)+
  labs(fill = "Prime Area to Tract Area")+
  theme_void()
ai1
```



```
ggsave("Prime_Grid.pdf")
```

```
options(scipen = 999)
ai2 <- ggplot( MoCo_grid_inc.crop)+
  geom_sf(aes(fill = med_inc))+
  scale_fill_distiller(palette = "Greens", direction = 1)+
  labs(fill = "Median Income")+
  theme_void()
ai2
```



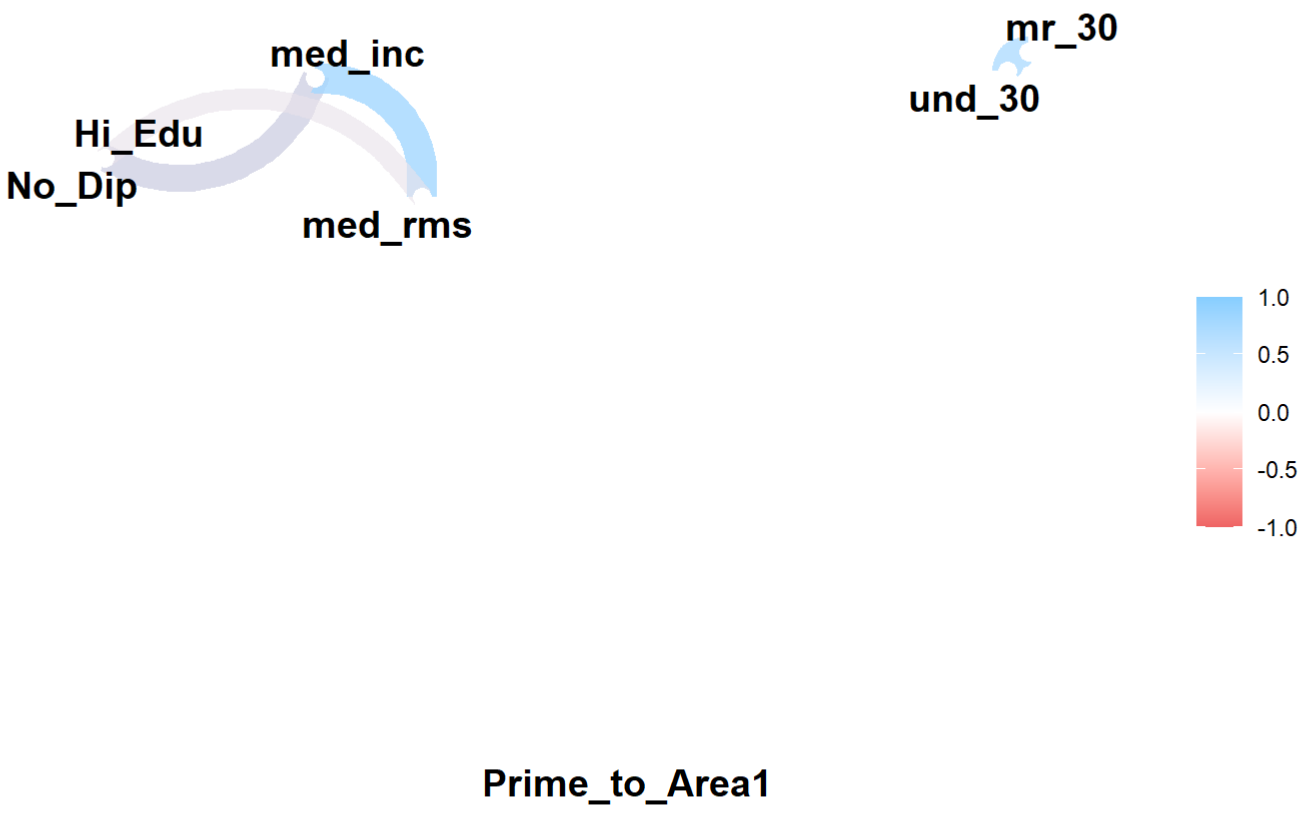
```
ggsave("Inc_Grid.pdf")
```

Does it Correlate???

```
#Using corrr to see if prime area to tract area correlates to median income and other variables
MoCo_corr <- MoCo_final %>%
  dplyr::select(-GEOID) %>%
  st_drop_geometry()
correlations <- correlate(MoCo_corr, method = "pearson")
```

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
network_plot(correlations) #sadge... prime soil area does not correlate
```



nope.