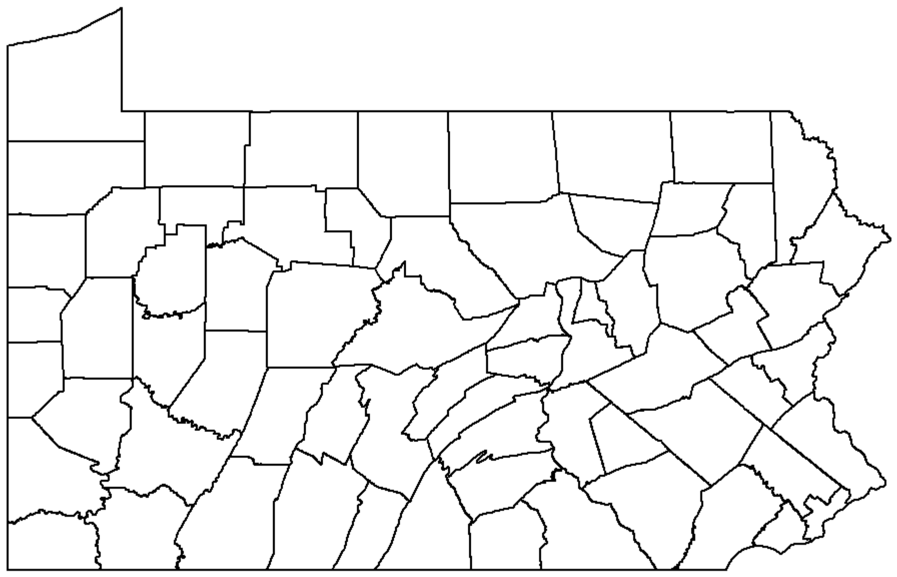
Chapter 5 Exercises

Meagan Fairfield-Peak

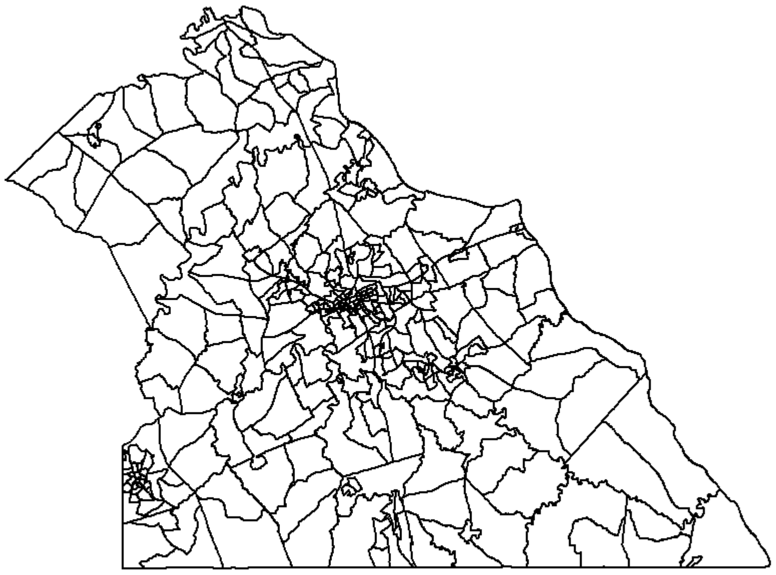
2/16/2022

- Give tigris a try for yourselves! Go through the available geographies in tigris and fetch data for a state and/or county of your choosing.
- Using your data, try some of the plotting options covered in this chapter. Plot the data with plot(), geom_sf(), and with mapview().
- Use suggest_crs() from the crsuggest package to identify an appropriate projected coordinate reference system for your data. Then, use st_transform() from the sf package to apply a CRS transformation to your data.

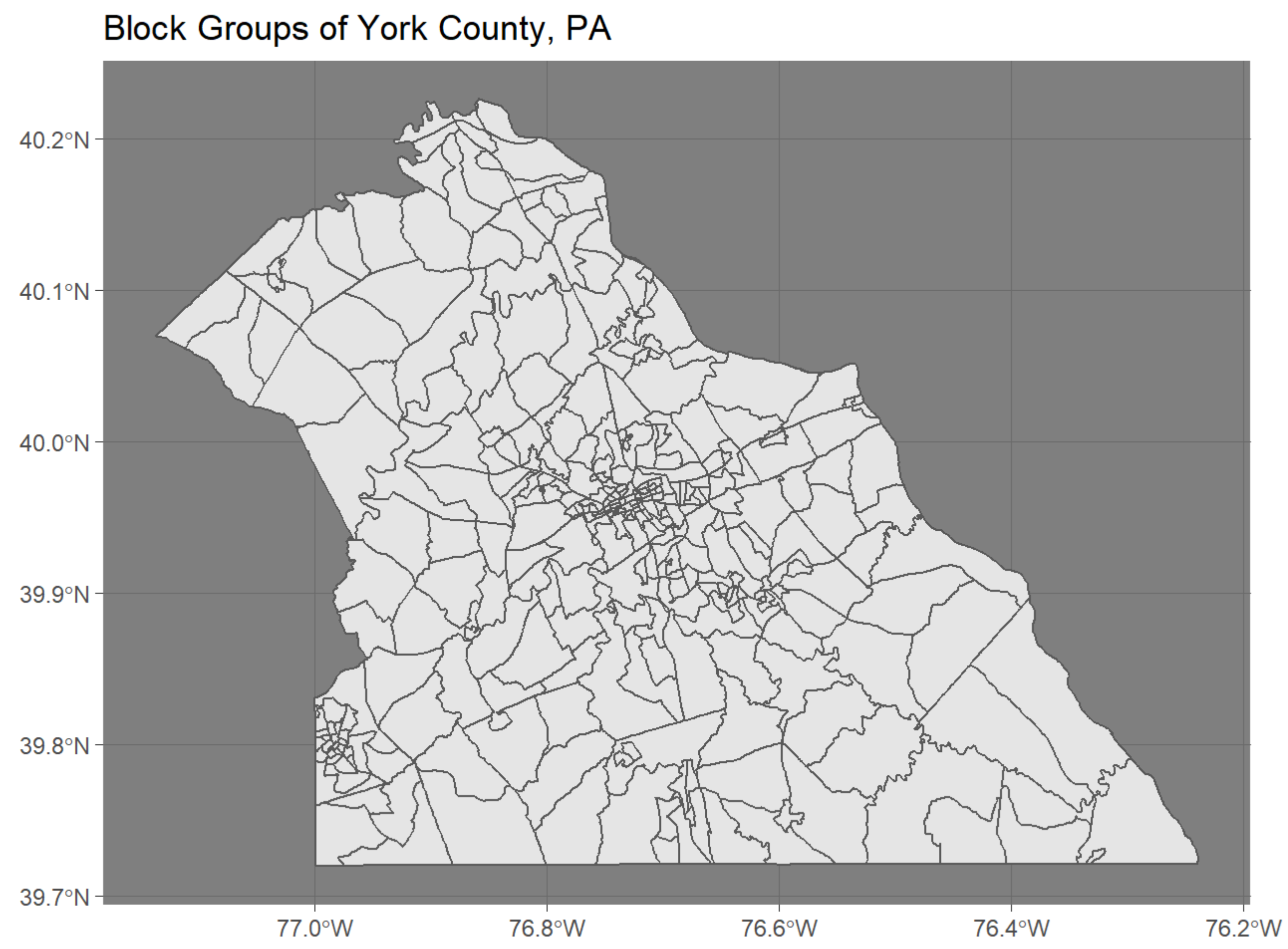
```
library(tigris)

pa_counties <- counties("PA")
pa_block <- block_groups("PA", "York")

plot(pa_counties$geometry)
```



```
plot(pa_block$geometry)
```

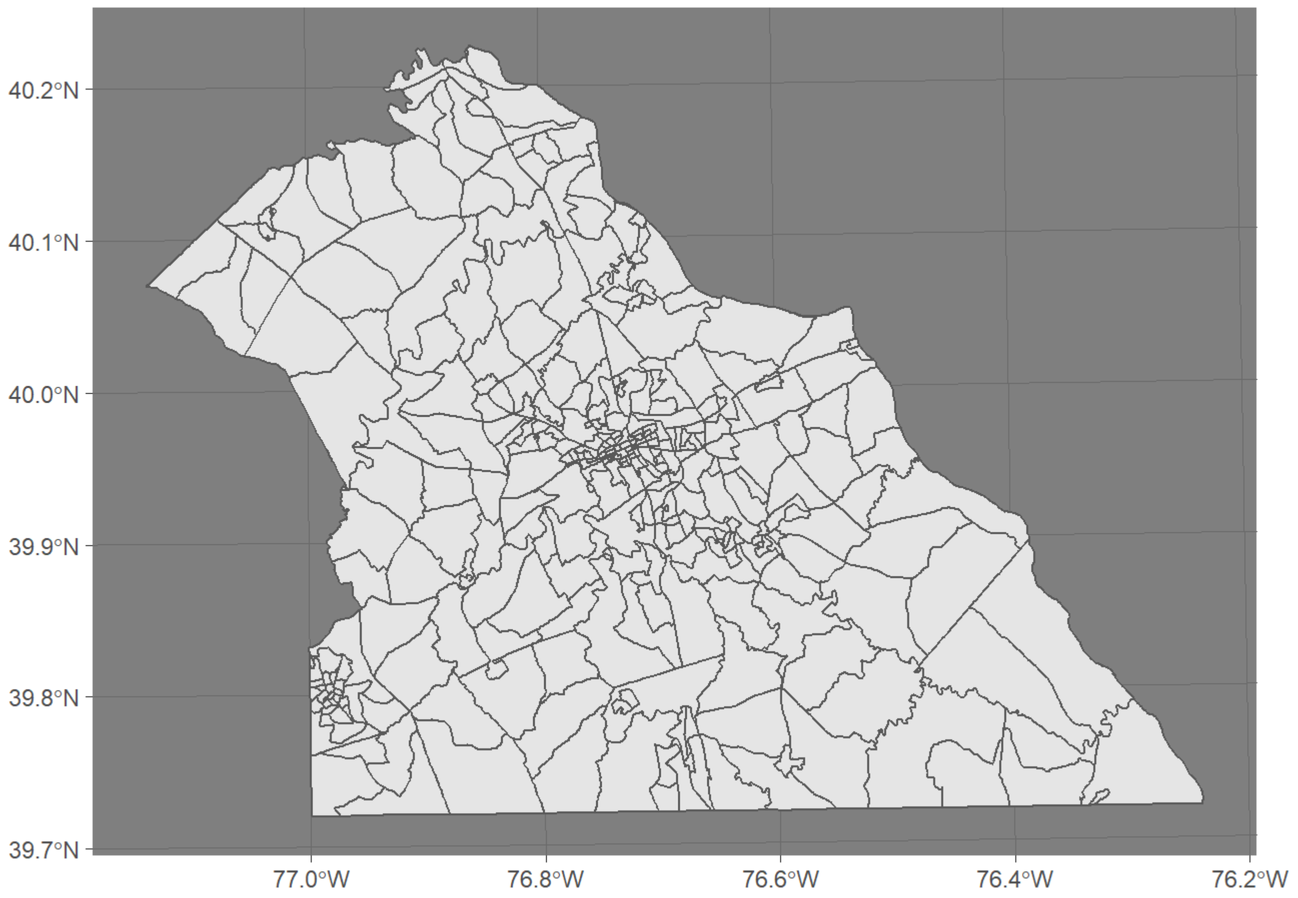


```
library(ggplot2)

ggplot(pa_block) +
  geom_sf() +
  theme_dark() +
  labs(title = "Block Groups of York County, PA")
```



```
library(mapview)

mapview(pa_block)


library(sf)
library(crsuggest)

block_crs <- suggest_crs(pa_block)

block_projected <- st_transform(pa_block, crs = 2272)

ggplot(block_projected) +
  geom_sf() +
  theme_dark() +
  labs(title = "Block Groups of York County, PA")
```

Block Groups of York County, PA



Chapter 6 Exercises

Meagan Fairfield-Peak

2/20/2022

Using one of the mapping frameworks introduced in this chapter (either ggplot2, tmap, or leaflet) complete the following tasks:

If you are just getting started with tidycensus/the tidyverse, make a race/ethnicity map by adapting the code provided in this section but for a different county.

Libraries:

```
library(tidycensus)
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   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(crsuggest)
```

```
## Using the EPSG Dataset v10.019, a product of the International Association of Oil & Gas Produ
cers.
## Please view the terms of use at https://epsg.org/terms-of-use.html.
```

```
library(sf)
```

```
## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
library(tmap)
```

Data:

```
york_race <- get_acs(
  geography = "tract",
  state = "PA",
  county = "York",
  variables = c(White = "B03002_003",
                Black = "B03002_004",
                Native = "B03002_005",
                Asian = "B03002_006",
                Hispanic = "B03002_012"),
  summary_var = "B03002_001",
  geometry = TRUE
) %>%
  mutate(percent = 100 * (estimate / summary_est))
```

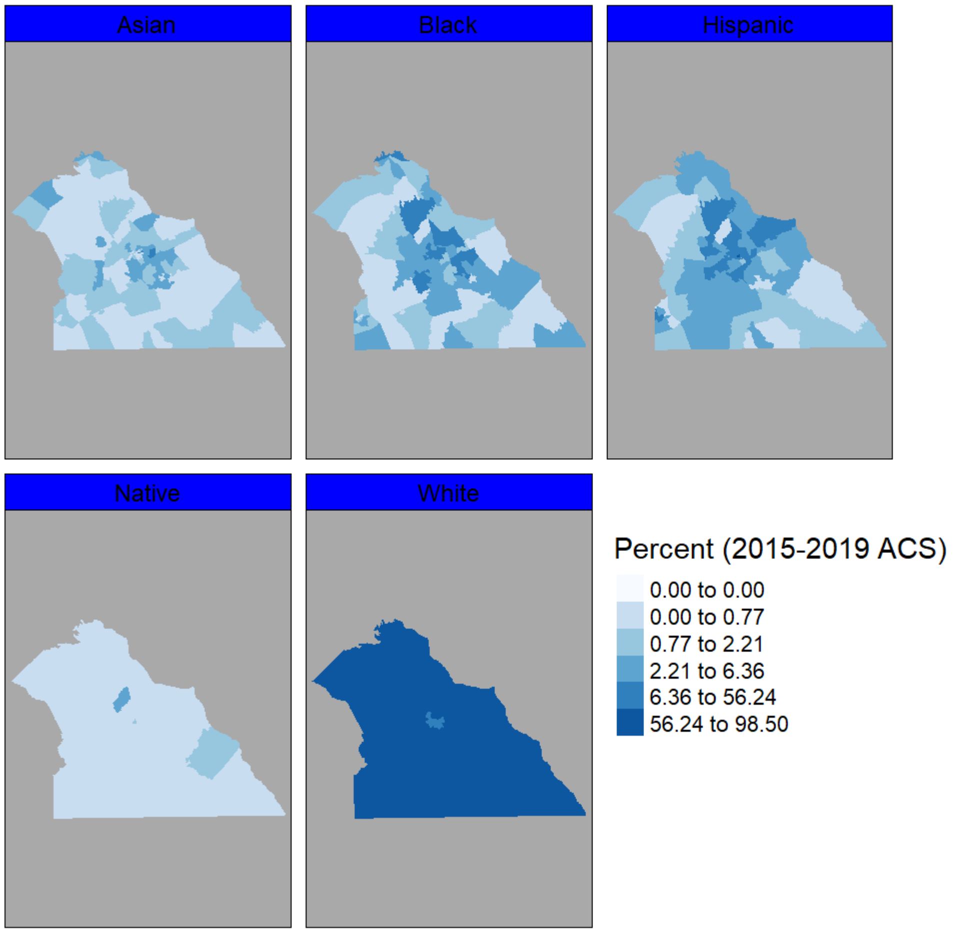
```
## Getting data from the 2015-2019 5-year ACS
```

```
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future
sessions, set `options(tigris_use_cache = TRUE)`.
```

```
york_crs<-suggest_crs(york_race)
```

Visualizing:

```
tm_shape(york_race,
         projection = sf::st_crs(6565)) +
  tm_facets(by = "variable", scale.factor = 4) +
  tm_fill(col = "percent",
          style = "quantile",
          n = 6,
          palette = "Blues",
          title = "Percent (2015-2019 ACS)",) +
  tm_layout(bg.color = "darkgrey",
            legend.position = c(-0.7, 0.2),
            panel.label.bg.color = "blue")
```



Next, find a different variable to map with tidycensus::load_variables(). Review the discussion of cartographic choices in this chapter and visualize your data appropriately.

Data:

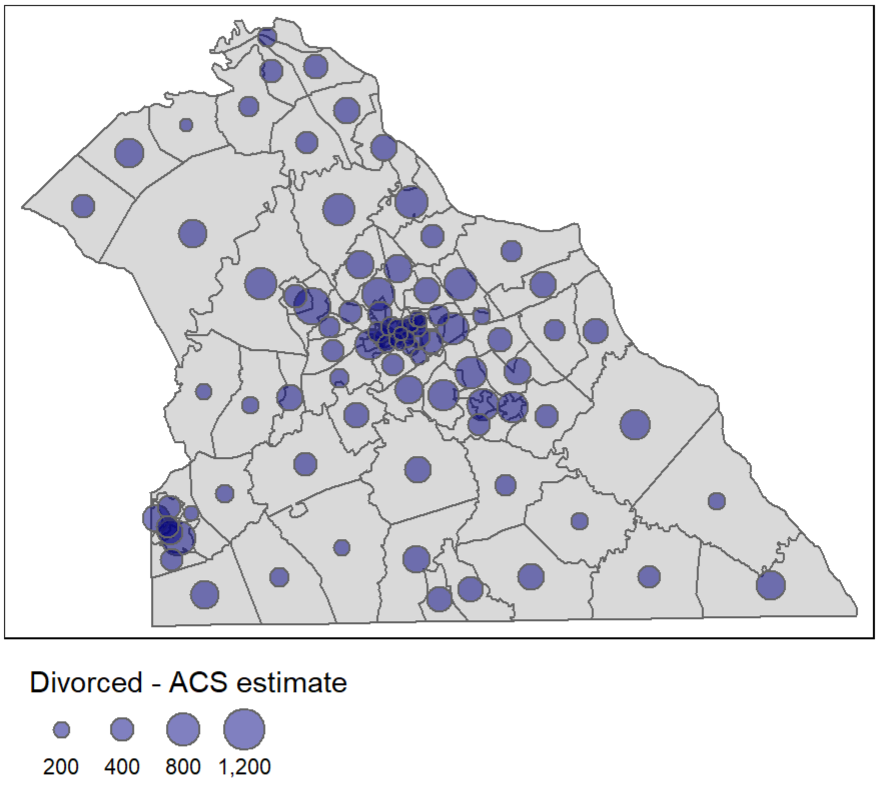
```
v19 <- load_variables(2019, "acs5", cache = TRUE)

york_divorced <- get_acs(
  geography = "tract",
  state = "PA",
  county = "York",
  variables = "B06008_004",
  geometry = TRUE
  )
```

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

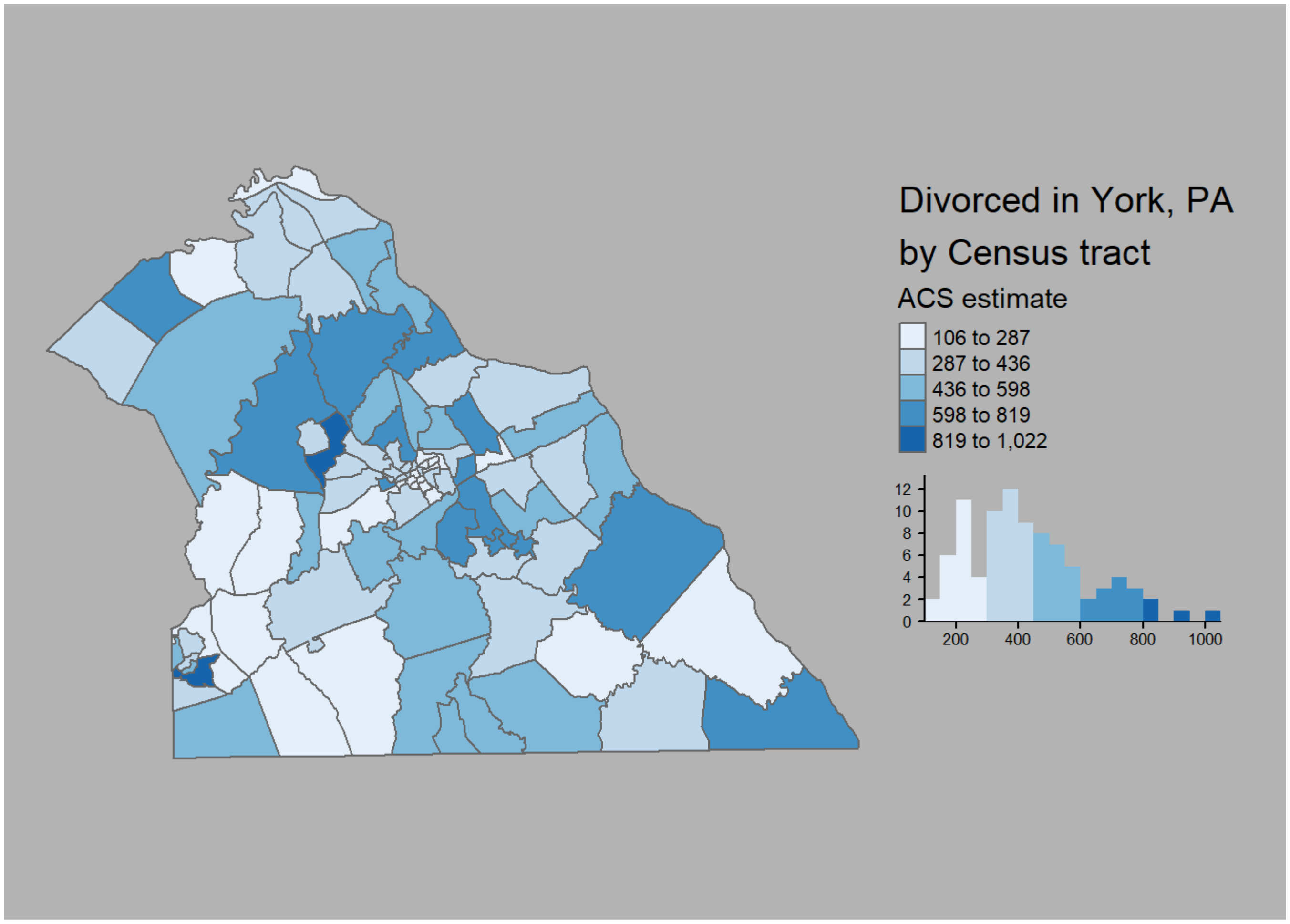
```
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future
sessions, set `options(tigris_use_cache = TRUE)`.
```

Visualizing:

```
tm_shape(york_divorced,
         projection = sf::st_crs(6565)) +
  tm_polygons() +
  tm_bubbles(size = "estimate", alpha = 0.5,
             col = "navy",
             title.size = "Divorced - ACS estimate") +
  tm_layout(legend.outside = TRUE,
            legend.outside.position = "bottom")
```



```
tm_shape(york_divorced,
         projection = sf::st_crs(6565)) +
  tm_polygons(col = "estimate",
          style = "jenks",
          n = 5,
          palette = "Blues",
          title = "ACS estimate",
          legend.hist = TRUE) +
  tm_layout(title = "Divorced in York, PA \nby Census tract",
            frame = FALSE,
            legend.outside = TRUE,
            bg.color = "grey70",
            legend.hist.width = 5,
            fontfamily = "Verdana")
```


Divorced in York, PA
by Census tract
ACS estimate
106 to 287
287 to 436
436 to 598
598 to 819
819 to 1,022
12
10
8
6
4
2
0
200
400
600
800
1000

Chapter 7 Exercises

Meagan Fairfield-Peak

2/20/2022

Libraries

```
library(tidycensus)
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   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(tigris)
```

```
## To enable
## caching of data, set `options(tigris_use_cache = TRUE)` in your R script or .Rprofile.
```

```
##
## Attaching package: 'tigris'
```

```
## The following object is masked from 'package:tidycensus':
##
##     fips_codes
```

```
library(sf)
```

```
## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
library(spdep)
```

```
## Loading required package: sp
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

1. Identify a different core-based statistical area of interest and use the methods introduced in this chapter to extract Census tracts or block groups for that CBSA.

```
md_tracts <- map_dfr(c("MD"), ~{
  tracts(.x, cb = TRUE, year = 2020)
}) %>%
  st_transform(26985)

md_metro <- core_based_statistical_areas(cb = TRUE, year = 2020) %>%
  filter(str_detect(NAME, "Baltimore-Columbia-Towson")) %>%
  st_transform(26985)


md_tracts_within <- md_tracts %>%
  st_filter(md_metro, .predicate = st_within)


ggplot() +
  geom_sf(data = md_tracts_within, fill = "white", color = "grey") +
  geom_sf(data = md_metro, fill = NA, color = "purple") +
  theme_void()
```

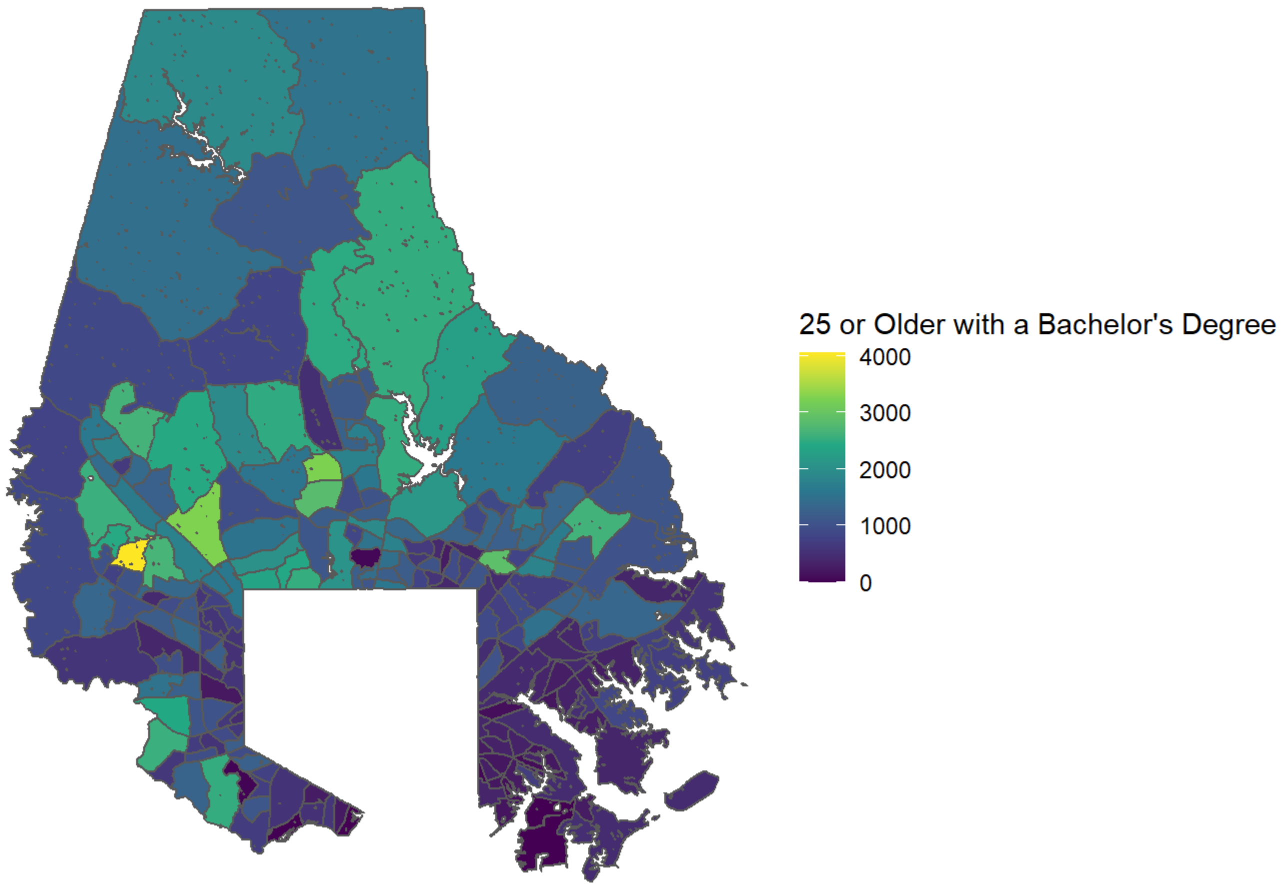
2. Replicate the st_erase() cartographic workflow for a different county with significant water area; a good choice is King County, Washington. Be sure to transform your data to an appropriate projected coordinate system (selected with suggest_crs()) first. If the operation creates too many “holes” for water areas, try filtering down the water dataset by its AWATER column and retaining only the largest water areas, then re-run and see what you get.

```
md <- get_acs(
  geography = "tract",
  variables = "C15010_001",
  state = "MD",
  county = "Baltimore County",
  year = 2019,
  geometry = TRUE,
  cb= FALSE
) %>%
  st_transform(26985)
```

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

```
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future
sessions, set `options(tigris_use_cache = TRUE)`.
```

```r
st_erase <- function(x, y) {
  st_difference(x, st_union(y))
}

md_water <- area_water("MD", "Baltimore County", year = 2019) %>%
  st_transform(26985)

md_erase <- st_erase(md, md_water)

ggplot(md_erase) +
  geom_sf(aes(fill = estimate)) +
  scale_fill_viridis_c(labels =) +
  theme_void() +
  labs(fill = "25 or Older with a Bachelor's Degree")
```



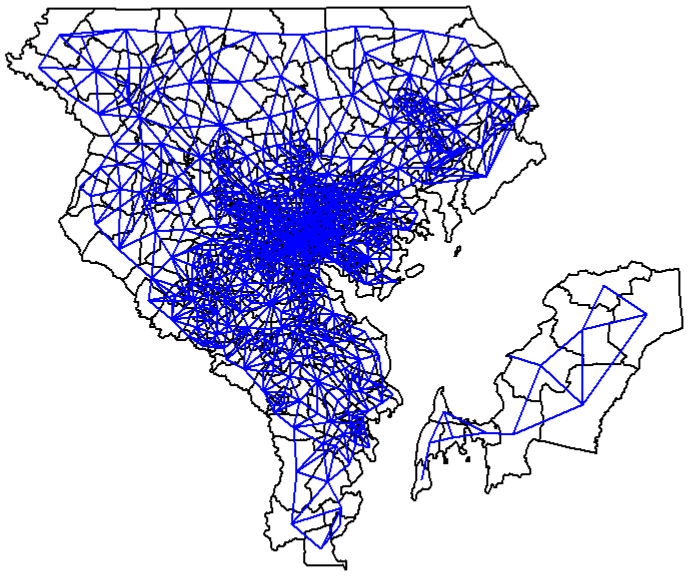
3. Acquire a spatial dataset with tidycensus for a region of interest and a variable of interest to you. Follow the instructions in this chapter to generate a spatial weights matrix, then compute a hot-spot analysis with localG()

```
balt_area <- core_based_statistical_areas(cb = TRUE, year = 2019) %>%
  filter(str_detect(NAME, "Baltimore-Columbia-Towson")) %>%
  st_transform(26985)

b_tracts <- get_acs(
  geography = "tract",
  variables = "B01001_035",
  state = "MD",
  year = 2019,
  geometry = TRUE
) %>%
  st_transform(26985) %>%
  st_filter(balt_area, .predicate = st_within) %>%
  na.omit()
```

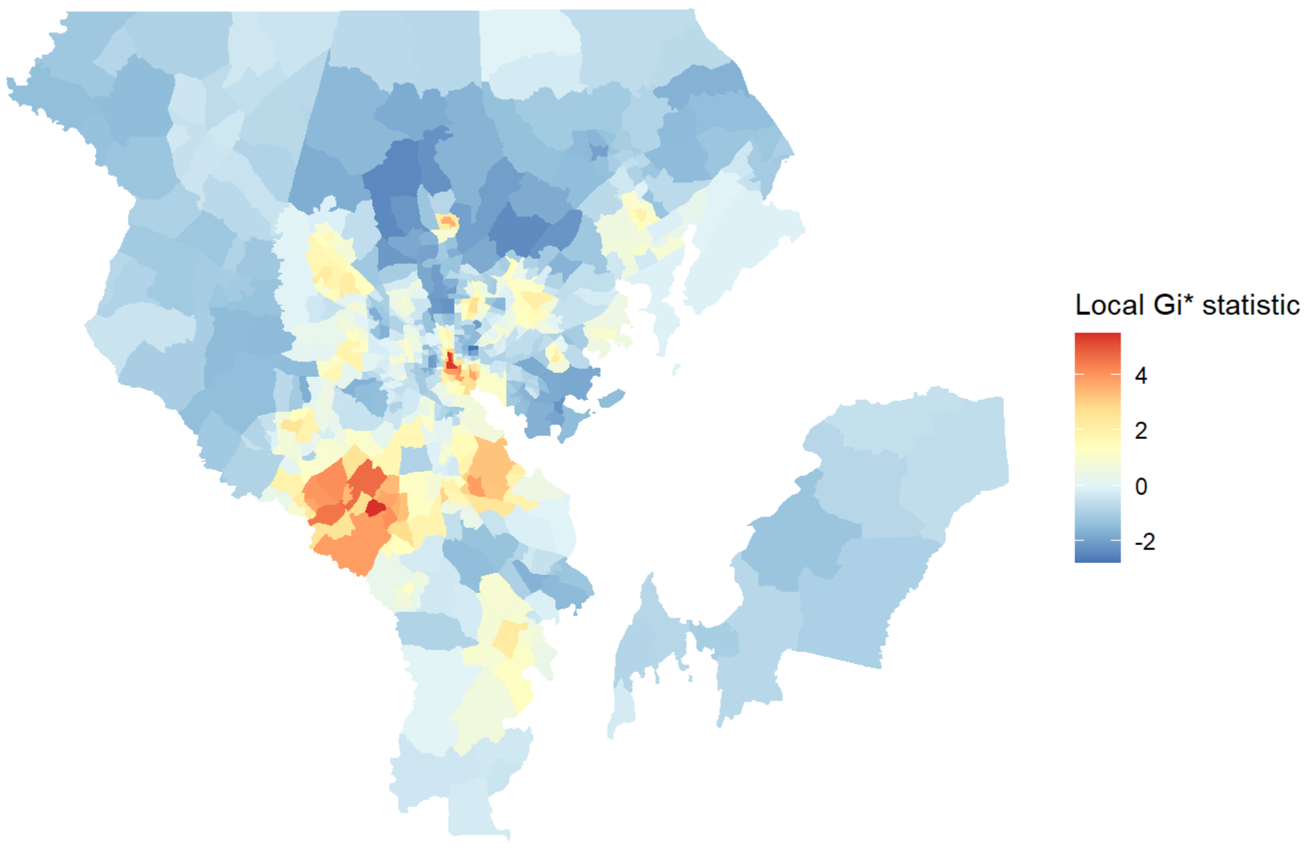
```
## Getting data from the 2015-2019 5-year ACS
```

```
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future
sessions, set `options(tigris_use_cache = TRUE)`.
```

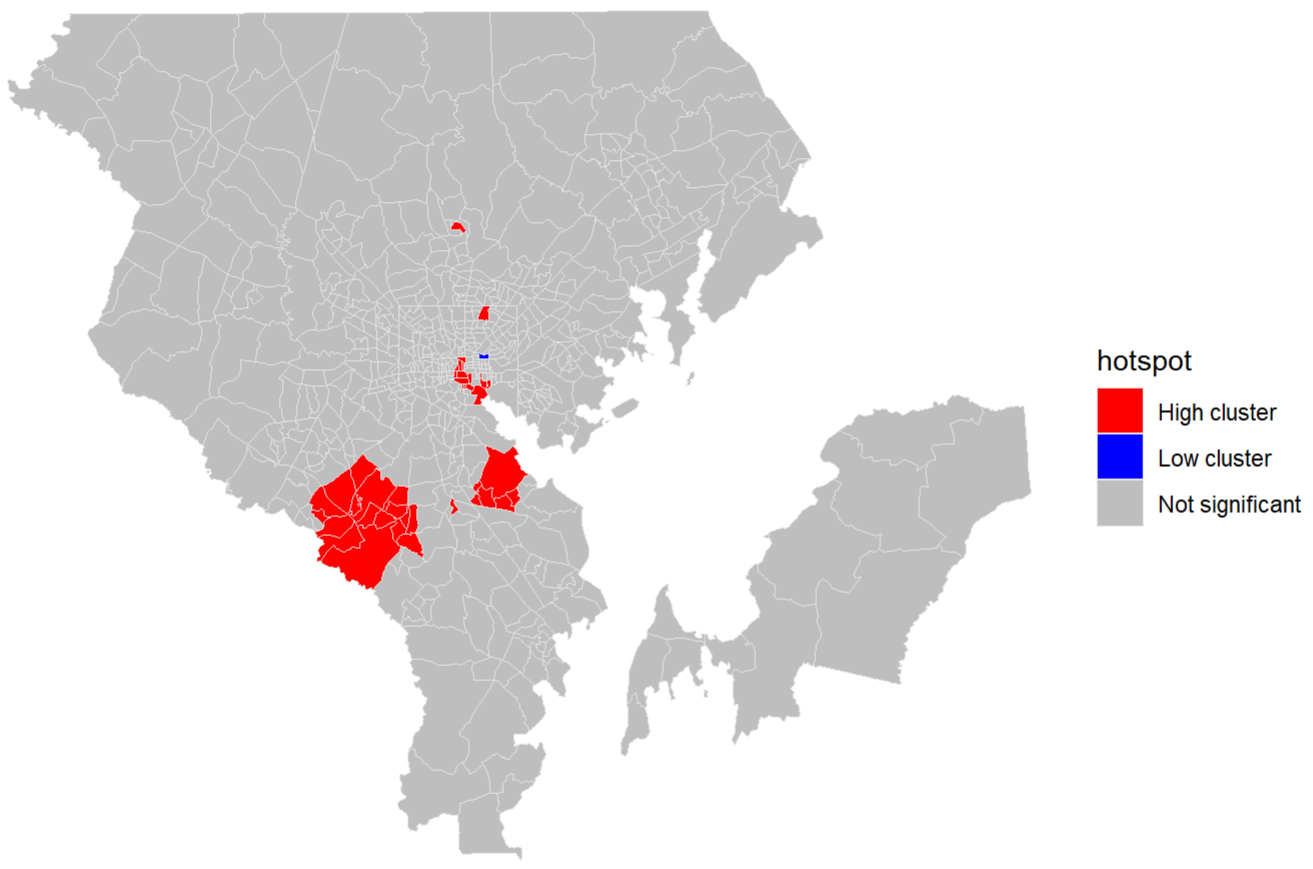
```
neighbors <- poly2nb(b_tracts, queen = TRUE)

summary(neighbors)

b_coords <- b_tracts %>%
  st_centroid() %>%
  st_coordinates()

plot(b_tracts$geometry)
plot(neighbors,
     coords = b_coords,
     add = TRUE,
     col = "blue",
     points = FALSE)
```



```
neighbors[[1]]

weights <- nb2listw(neighbors, style = "W")

weights$weights[[1]]

b_tracts$lag_estimate <- lag.listw(weights, b_tracts$estimate)


# For Gi*, re-compute the weights with `include.self()`
localg_weights <- nb2listw(include.self(neighbors))

b_tracts$localG <- localG(b_tracts$estimate, localg_weights)

ggplot(b_tracts) +
  geom_sf(aes(fill = localG), color = NA) +
  scale_fill_distiller(palette = "RdYlBu") +
  theme_void() +
  labs(fill = "Local Gi* statistic")
```



```
b_tracts <- b_tracts %>%
  mutate(hotspot = case_when(
    localG >= 2.56 ~ "High cluster",
    localG <= -2.56 ~ "Low cluster",
    TRUE ~ "Not significant"
  ))

ggplot(b_tracts) +
  geom_sf(aes(fill = hotspot), color = "grey90", size = 0.1) +
  scale_fill_manual(values = c("red", "blue", "grey")) +
  theme_void()
```


hotspot
🟥 High cluster
🟦 Low cluster
⬜ Not significant

Language Spoken Patchwork

Meagan Fairfield-Peak

2/20/2022

Libraries

```
library(tidycensus)
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   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(ggplot2)
library(geofacet)
```

Obtaining Data

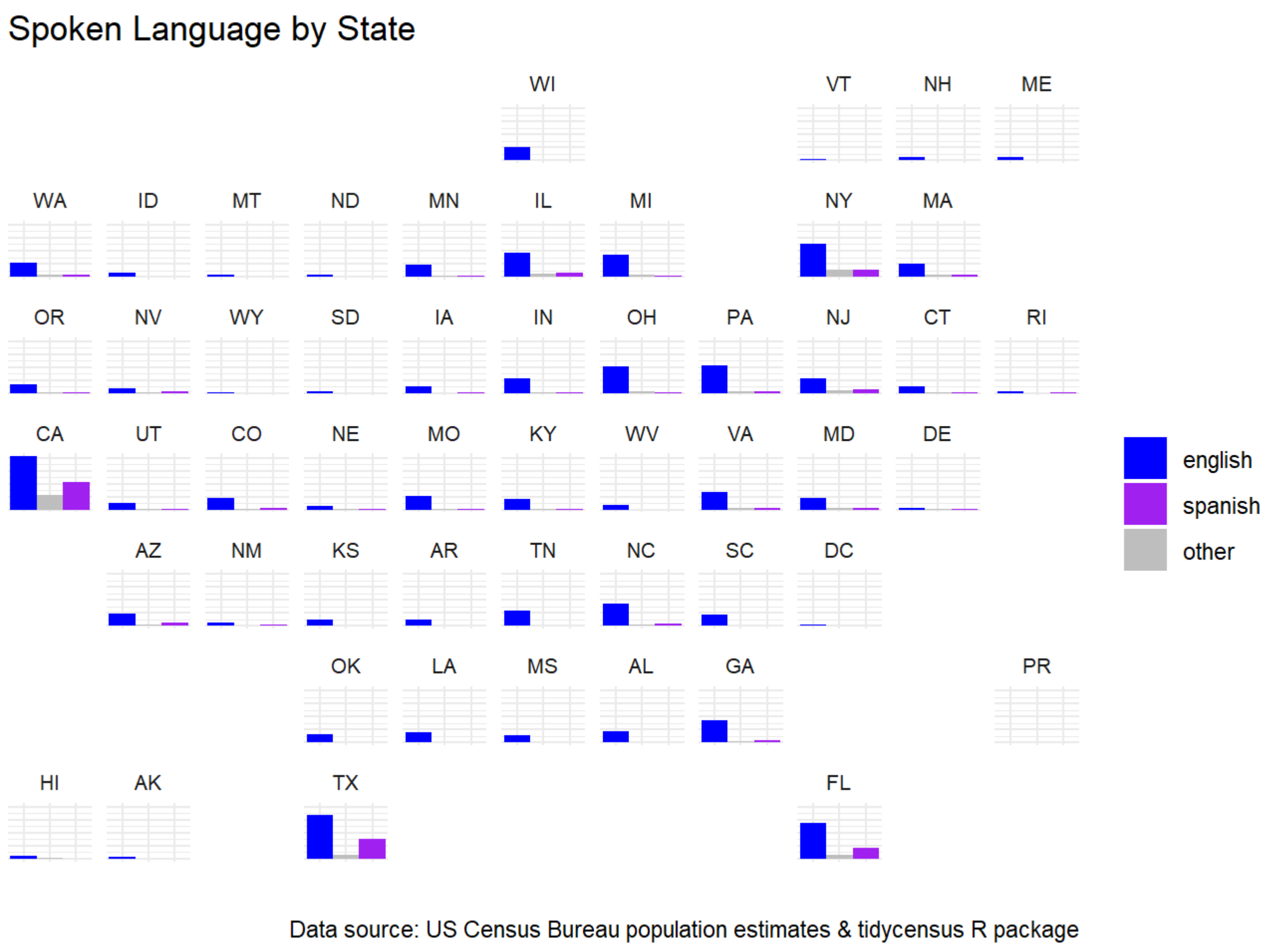
```
v19 <- load_variables(2019, "acs5", cache = TRUE)

language_pyramid <- get_acs(
  geography = "state",
  variables = c(
    english = "B06007_002",
    spanish = "B06007_003",
    other = "B06007_006"
  ))
```

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

Visualizing Data

```
ggplot(language_pyramid, aes(x = variable, y = estimate, fill = variable)) +
  geom_col(width = 1) +
  theme_minimal() +
  scale_fill_manual(values = c("english"="blue", "spanish"="purple", "other"="grey")) +
  facet_geo(~NAME, grid = "us_state_with_DC_PR_grid2",
            label = "code") +
  theme(axis.text = element_blank(),
        strip.text.x = element_text(size = 8)) +
  labs(x = "",
       y = "",
       title = "Spoken Language by State",
       fill = "",
       caption = "Data source: US Census Bureau population estimates & tidycensus R package")
```

```
## Warning: Removed 3 rows containing missing values (position_stack).
```



I would have preferred to have the bars stacked and fulling the entire space for each space but I am still working on editing the code to visualize in that way.

Meagan Fairfield-Peak
GES 687

Lab Write-Up

Lab 2
I used Rstudio in 688 but did very little development of my own code so these exercises were great to better understand how to edit and change code for my own purpose. Just as in Lab 1, any error I received I was able to find a solution online, or was a connection error and just needed rerunning. I am currently finding all of the code to consider and keep in mind a little overwhelming but I know in the beginning it is a lot of copying and pasting. It will take plenty of time to become familiar with it all.