GSA 2022 Abstract Code

Peter Sun

July 30, 2022

Contents

1 Load Packages and Data **2**

2 Re-Read Project-Level Environmental Variables for HRS Dataset File Paths **3**

3 Import Work, Age, Wave 14 (2018) Flag, and Nursing Home Status **4**

4 Import Geography Data **5**

5 Import Volunteering Data **6**

6 Import Caregiving Data **7**

7 Merge Datasets and Create Multiple Productive Activities **10**
7.1 Study Sample Size 10

8 Helper Functions **12**

9 Results **13**
9.1 Census Region Statistics 13
9.2 Census Division Statistics 14
9.3 Rural-Urban Comparisons (Two-Proportion Z Tests) 15

10 Plots **18**

1 Load Packages and Data

```
library(tidyverse)
library(haven)
library(sjlabelled)
library(ggpubr)
library(kableExtra)
library(scales)
library(choroplethr)
library(choroplethrMaps)
library(glue)
library(tigris)
data(state.regions)
library(janitor)
library(fips)
library(cowplot)
library(biscale)

# Avoid select clashes
select <- dplyr::select
recode <- dplyr::recode
summarize <- dplyr::summarize
```

2 Re-Read Project-Level Environmental Variables for HRS Dataset File Paths

```
readRenviron(".Renviron")
```

3 Import Work, Age, Wave 14 (2018) Flag, and Nursing Home Status

```r
# Import "randhrs1992_2018v1.dta"
rand.long <- read_dta(Sys.getenv("HRS_LONG"),
  col_select = c(hhid, pn, r14agey_e, inw14,
                 r14work, # Currently Working for Pay
                 s14hhidpn)) %>% # Spouse Identifier for Caregiving
  rename(worker = r14work) %>%
  haven::zap_formats() %>%
  sjlabelled::remove_all_labels() %>%
  as_tibble()

# Inspect worker
table(rand.long$worker, useNA = "always") # 6711 workers
```

```
##
##     0     1  <NA>
## 10363  6711 25159
```

```r
# Import nursing home status from "trk2018tr_r.dta"
tracker <- read_dta(Sys.getenv("HRS_TRACKER_2018_20"),
  col_select = c(hhid, pn, qnurshm))
```

4 Import Geography Data

```
# Import "HRSXREGION18.dta"
geo <- read_dta(Sys.getenv("HRS_REGION_2018_82"),
    col_select = c(hhid, pn, beale2013_18, region18)) %>%
  rename(rural = beale2013_18) %>%
  mutate(rural = recode(rural,
    `1` = "Urban",
    `2` = "Urban",
    `3` = "Rural",
    .default = NA_character_)) %>%
  mutate(region = recode(region18,
    `1` = "northeast",
    `2` = "northeast",
    `3` = "midwest",
    `4` = "midwest",
    `5` = "south",
    `6` = "south",
    `8` = "west",
    `9` = "west",
    .default = NA_character_)) %>%
  mutate(division = recode(region18,
    `1` = "New England",
    `2` = "Middle Atlantic",
    `3` = "East North Central",
    `4` = "West North Central",
    `5` = "South Atlantic",
    `6` = "East South Central",
    `7` = "West South Central",
    `8` = "Mountain",
    `9` = "Pacific", .default = NA_character_)) %>%
  haven::zap_formats() %>%
  sjlabelled::remove_all_labels() %>%
  as_tibble()
```

5 Import Volunteering Data

```
vol.18 <- read_dta(Sys.getenv("HRS_2018_FAT"),
    col_select = c(hhid, pn, qg086)) %>%
  rename(volunteer = qg086) %>%
  mutate(volunteer = recode(volunteer,
    `1` = 1,
    `5` = 0,
    .default = NA_real_))
table(vol.18$volunteer, useNA = "always") # 5772 volunteers
```

```
##
##     0     1  <NA>
## 11328  5772    46
```

6 Import Caregiving Data

```r
# Import 2018 RAND Fat File (File name: h04f1c.dta)
care18 <- read_dta(Sys.getenv("HRS_2018_FAT"),
  col_select = c(
    "hhidpn", "hhid", "pn",

    # adl helpers
    starts_with("qg033_"),

    # iadl helpers
    starts_with("qg055_"),

    # caregiving grandchildren
    "qe060",

    # caregiving parental personal
    "qf119",

    # caregiving parental errands
    "qf139"
  )
) %>%
  haven::zap_formats() %>%
  sjlabelled::remove_all_labels() %>%
  as_tibble()

# Identify participants who had a spouse/partner ADL or IADL helper
# QG033_x = ADL helpers
# QG055_x = IADL helpers
# TODO: those who did not report having ADL or IADL needs should be marked as
# not having an ADL/IADL caregiver
care18b <- care18 %>%
  # Temporarily set 2s to 1, non-2 numbers to 0, and keep NAs as NAs
  # Then count the number of 1s (which is equivalent to the number of 2s)
  mutate(spouse_helper_sum = rowSums(
    ifelse(select(., starts_with("qg033"), starts_with("qg055")) == 2, 1, 0),
    na.rm = T
  )) %>%

  # Because rowSums in the above syntax will count all NAs as 0,
  # use !is.na(...) == 0 to set all
  # non-missing values to TRUE and missing values to FALSE
  # Then use rowSums to count the number of non-missing values
  # Because FALSE = missing, if all of the columns are FALSE/NA/Missing,
  # then the sum will be 0
  # Finally, set spouse_helper_sum to NA_real_ if all of the columns are False/NA/Missing
  mutate(spouse_helper_sum = ifelse(
    rowSums(!is.na(select(., starts_with("qg033"), starts_with("qg055")))) == 0,
    NA_real_, spouse_helper_sum
  )) %>%

  # spouse_helper_sum counts the number of 2s (spouse/partner) in QG033_x and QG055_x
```

```
  # has_spouse_helper coding:
  # 1 = if there is at least one 2s across all QG033_x and QG055_x variables
  # 0 = if there is at least one non-missing variable and no 2s
  # NA = all QG033_x and QG055_x variables are NAs
  mutate(has_spouse_helper = ifelse(spouse_helper_sum >= 1, 1,
                             ifelse(is.na(spouse_helper_sum), NA_real_, 0)))
table(care18b$has_spouse_helper, useNA = "always")
```

```
##
##     0     1  <NA>
##  1415  1022 14709
```

```
# Extract participants who have a spousal ADL/IADL caregiver
# Merge their spouse PN
# Then create a dataset with hhid and pn of spouse and an indication of
# whether or not they are a spousal caregiver
spousal_caregivers.18 <- care18b %>%
  filter(has_spouse_helper == 1) %>%
  select(hhid, pn, has_spouse_helper) %>%

  # Left join with respondents' spouse/partner's hhidpn in wave 14 (2018)
  # "If there is no spouse in a given wave, SwHHIDPN is set to zero. If SwHHIDPN is
  # unknown, and the marital status in a particular wave
  # is either missing (.M) or married, SwHHIDPN is set to a special missing code of .M."
  left_join(rand.long %>% select(hhid, pn, s14hhidpn) %>%
              filter(!is.na(s14hhidpn), s14hhidpn != 0, !is.na(hhid), !is.na(pn)),
    by = c("hhid", "pn")
  ) %>%
  select(hhidpn = s14hhidpn, caregiver_spousal = has_spouse_helper)
spousal_caregivers.18
```

```
## # A tibble: 1,022 x 2
##      hhidpn caregiver_spousal
##       <dbl>             <dbl>
##  1 10059020                 1
##  2 10106010                 1
##  3 10453020                 1
##  4 10533811                 1
##  5 10648020                 1
##  6       NA                 1
##  7 11071020                 1
##  8 11332010                 1
##  9 12135020                 1
## 10 12232040                 1
## # ... with 1,012 more rows
## # i Use `print(n = ...)` to see more rows
```

```
table(is.na(spousal_caregivers.18$hhidpn), useNA = "always") # 27 spouses not in dataset
```

```
##
## FALSE  TRUE  <NA>
##   995    27     0
```

```
# Merge the spousal_caregivers data back to the dataset
care18c <- care18b %>%
  left_join(spousal_caregivers.18, by = c("hhidpn"))
```

```r
# Format parental/grandchildren caregivers
# qe060 = grandchildren caregiver
# qf119 = parental caregiver (personal needs) (1 = yes, 5 = no, 8 = DK, 9 = RF)
# qf139 = parental caregiver (errands)
care18d <- care18c %>%
  mutate(across(
    .cols = c(qe060, qf119, qf139),
    ~ recode(.,
      `1` = 1, `5` = 0, `8` = NA_real_, `9` = NA_real_,
      .default = NA_real_
    )
  )) %>%
  rename(
    caregiver_grandchildren = qe060,
    caregiver_parental_personal = qf119,
    caregiver_parental_errands = qf139
  ) %>%
  mutate(
    caregiver_parental =
      ifelse(caregiver_parental_personal == 1 | caregiver_parental_errands == 1, 1,
      ifelse(is.na(caregiver_parental_personal) & is.na(caregiver_parental_errands),
             NA_real_, 0)))
table(care18d$caregiver_parental, useNA = "always")
```

```
##
##     0     1  <NA>
##  3545  1938 11663
```

```r
# Caregiver Coding
# 1 = If at least one type of caregiver
# NA = Two or more NAs
# 0 = All other cases (at least two zeroes)
care18e <- care18d %>%
  select(hhid, pn,
    cs = caregiver_spousal,
    cp = caregiver_parental,
    cg = caregiver_grandchildren
  ) %>%
  mutate(
    Caregiver_Sum = rowSums(select(., cs:cg), na.rm = T),
    # Set caregiver to NA if two or more NAs
    Caregiver_NACount = rowSums(is.na(select(., cs:cg))),
    Caregiver_Sum = ifelse(Caregiver_NACount >= 2, NA_real_, Caregiver_Sum),
    caregiver = ifelse(Caregiver_Sum >= 1, 1,
                ifelse(is.na(Caregiver_Sum), NA_real_, 0))
  ) %>%
  select(hhid, pn, caregiver)
table(care18e$caregiver, useNA = "always") # 2640 caregivers
```

```
##
##     0     1  <NA>
##  1223  2640 13283
```

7 Merge Datasets and Create Multiple Productive Activities

```
df <- rand.long %>%
  left_join(tracker, by = c("hhid", "pn")) %>%
  filter(inw14 == 1) %>% # in wave 14
  filter(qnurshm %in% c(5, 6, 7)) %>% # community-dwelling
  filter(r14agey_e >= 65) %>% # age 65+
  left_join(geo, by = c("hhid", "pn")) %>%
  left_join(vol.18, by = c("hhid", "pn")) %>%
  left_join(care18e, by = c("hhid", "pn")) %>%

  # Count 0s and 1s
  mutate(multi_zeroes = rowSums(select(., volunteer, caregiver, worker) == 0,
                               na.rm = T)) %>%
  mutate(multi_ones = rowSums(select(., volunteer, caregiver, worker) == 1,
                             na.rm = T)) %>%

  # Set multi to 0 if at least one 0, otherwise NA
  mutate(multi = ifelse(multi_zeroes >= 1, 0, NA_real_)) %>%

  # Set multi to 1 if at least one productive activity
  mutate(multi = ifelse(multi_ones >= 1, 1, multi))
  # rowwise() %>%
  # mutate(multi_sum = sum(c_across(c(volunteer, caregiver, worker)), na.rm = T)) %>%
  # ungroup()

table(df$volunteer, useNA = "always")
```

```
##
##    0    1 <NA>
## 5851 2853   24
```

```
table(df$caregiver, useNA = "always") # likely an underestimation of non-caregivers
```

```
##
##    0    1 <NA>
##  254  877 7597
```

```
table(df$worker, useNA = "always")
```

```
##
##    0    1 <NA>
## 7088 1607   33
```

```
table(df$multi, useNA = "always")
```

```
##
##    0    1 <NA>
## 4490 4219   19
```

```
#df %>% select(volunteer, caregiver, worker, multi_zeroes, multi_ones, multi) %>% view()
```

7.1 Study Sample Size

```
study_n <- nrow(df)
study_n # 2018 HRS sample of age 65+ community-dwelling individuals
```

```
## [1] 8728
```

8 Helper Functions

```
# Function for contingency table
# The denominator is currently assumed to be the sum of 0s and NAs, that is,
# the total population in each region/division. This is a
# limitation in this current study, because of the possibility of non-response bias.
get_kab <- function(data, geo, iv) {
  data %>%
    count({{ geo }}, rural, {{ iv }}) %>%
    group_by({{ geo }}, rural) %>%
    mutate(denom = sum(n),
           pct = n / denom) %>%
    ungroup() %>%
    filter({{ iv }} == 1) %>%
    filter(!is.na({{ geo }}), !is.na(rural), !is.na({{ iv }})) %>%
    select(-{{ iv }})
}

# Function for two proportion z-test
get_prop <- function(data, geo, iv) {
  data %>%
    count({{ geo }}, rural, {{ iv }}) %>%
    group_by({{ geo }}, rural) %>%
    mutate(sum = sum(n)) %>%
    ungroup() %>%
    filter({{ iv }} == 1) %>%
    filter(!is.na({{ geo }}), !is.na(rural), !is.na({{ iv }})) %>%
    pivot_wider(names_from = "rural", values_from = n:sum) %>%
    rowwise() %>%
    mutate(p = prop.test(x = c(n_Rural, n_Urban),
                         n = c(sum_Rural, sum_Urban))$p.value,
           rural_prop = prop.test(x = c(n_Rural, n_Urban),
                                  n = c(sum_Rural, sum_Urban))$estimate[1],
           urban_prop = prop.test(x = c(n_Rural, n_Urban),
                                  n = c(sum_Rural, sum_Urban))$estimate[2]) %>%
    mutate(rural_prop = percent(rural_prop, accuracy = .1),
           urban_prop = percent(urban_prop, accuracy = .1)) %>%
    kbl(booktabs = T, linesep = "", digits = 3) %>%
    kable_styling(position = "center") %>%
    kable_styling(latex_options = c("striped", "hold_position"))
}
```

9 Results

9.1 Census Region Statistics

```
r1 <- get_kab(df, region, worker) %>% rename(worker = pct)
r2 <- get_kab(df, region, volunteer) %>% rename(volunteer = pct)
r3 <- get_kab(df, region, caregiver) %>% rename(caregiver = pct)
r4 <- get_kab(df, region, multi) %>% rename(multiple = pct)
r1 %>%
  left_join(r2, by = c("region", "rural")) %>%
  left_join(r3, by = c("region", "rural")) %>%
  left_join(r4, by = c("region", "rural")) %>%
  select(-starts_with("n"), -starts_with("denom")) %>%
  mutate(across(where(is.numeric), scales::percent, 0.1)) %>%
  kbl(booktabs = T, linesep = "") %>%
  kable_styling(position = "center") %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

region	rural	worker	volunteer	caregiver	multiple
midwest	Rural	20.9%	42.0%	11.0%	56.8%
midwest	Urban	14.7%	37.5%	9.4%	49.2%
northeast	Rural	16.2%	29.3%	10.8%	43.1%
northeast	Urban	19.2%	31.4%	8.6%	48.0%
south	Rural	18.3%	31.0%	10.5%	46.5%
south	Urban	19.2%	33.0%	9.7%	48.8%
west	Rural	19.2%	36.8%	13.8%	53.1%
west	Urban	20.5%	30.1%	10.4%	48.3%

9.2 Census Division Statistics

```
d1 <- get_kab(df, division, worker) %>% rename(worker = pct)
d2 <- get_kab(df, division, volunteer) %>% rename(volunteer = pct)
d3 <- get_kab(df, division, caregiver) %>% rename(caregiver = pct)
d4 <- get_kab(df, division, multi) %>% rename(multiple = pct)
d1 %>%
  left_join(d2, by = c("division", "rural")) %>%
  left_join(d3, by = c("division", "rural")) %>%
  left_join(d4, by = c("division", "rural")) %>%
  select(-starts_with("n"), -starts_with("denom")) %>%
  mutate(across(where(is.numeric), scales::percent, 0.1)) %>%
  kbl(booktabs = T, linesep = "") %>%
  kable_styling(position = "center") %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

division	rural	worker	volunteer	caregiver	multiple
East North Central	Rural	21.3%	39.0%	12.3%	55.2%
East North Central	Urban	14.9%	36.8%	9.1%	48.9%
East South Central	Rural	18.9%	28.8%	15.3%	48.2%
East South Central	Urban	20.7%	32.0%	10.7%	48.5%
Middle Atlantic	Rural	14.8%	26.6%	12.5%	39.8%
Middle Atlantic	Urban	19.5%	32.4%	9.2%	49.4%
Mountain	Rural	23.4%	35.4%	11.4%	53.1%
Mountain	Urban	16.9%	31.3%	10.6%	46.5%
New England	Rural	20.5%	38.5%	5.1%	53.8%
New England	Urban	18.2%	28.5%	6.9%	43.8%
Pacific	Rural	7.8%	40.6%	20.3%	53.1%
Pacific	Urban	21.8%	29.7%	10.3%	49.1%
South Atlantic	Rural	18.0%	31.9%	8.5%	45.9%
South Atlantic	Urban	18.9%	33.2%	9.5%	48.9%
West North Central	Rural	20.2%	46.8%	9.1%	59.3%
West North Central	Urban	14.3%	39.2%	9.9%	50.0%
West South Central	Rural	14.5%	28.6%	10.9%	41.8%
West South Central	Urban	17.7%	24.5%	10.2%	42.4%

9.3 Rural-Urban Comparisons (Two-Proportion Z Tests)

9.3.1 Region: Worker, Volunteer, Caregiver, and Multiple

```
get_prop(df, region, worker)
```

region	worker	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
midwest	1	159	171	761	1162	0.001	20.9%	14.7%
northeast	1	27	203	167	1057	0.408	16.2%	19.2%
south	1	140	374	765	1947	0.625	18.3%	19.2%
west	1	46	298	239	1454	0.721	19.2%	20.5%

```
get_prop(df, region, volunteer)
```

region	volunteer	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
midwest	1	320	436	761	1162	0.052	42.0%	37.5%
northeast	1	49	332	167	1057	0.655	29.3%	31.4%
south	1	237	643	765	1947	0.328	31.0%	33.0%
west	1	88	438	239	1454	0.046	36.8%	30.1%

```
get_prop(df, region, caregiver)
```

region	caregiver	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
midwest	1	84	109	761	1162	0.269	11.0%	9.4%
northeast	1	18	91	167	1057	0.442	10.8%	8.6%
south	1	80	189	765	1947	0.605	10.5%	9.7%
west	1	33	151	239	1454	0.143	13.8%	10.4%

```
get_prop(df, region, multi)
```

region	multi	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
midwest	1	432	572	761	1162	0.001	56.8%	49.2%
northeast	1	72	507	167	1057	0.279	43.1%	48.0%
south	1	356	950	765	1947	0.310	46.5%	48.8%
west	1	127	703	239	1454	0.193	53.1%	48.3%

9.3.2 Division: Worker and Volunteer

```
get_prop(df, division, worker)
```

division	worker	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
East North Central	1	99	122	464	820	0.004	21.3%	14.9%
East South Central	1	42	68	222	328	0.680	18.9%	20.7%
Middle Atlantic	1	19	153	128	783	0.256	14.8%	19.5%
Mountain	1	41	67	175	396	0.086	23.4%	16.9%
New England	1	8	50	39	274	0.904	20.5%	18.2%
Pacific	1	5	231	64	1058	0.012	7.8%	21.8%
South Atlantic	1	98	306	543	1619	0.706	18.0%	18.9%
West North Central	1	60	49	297	342	0.062	20.2%	14.3%
West South Central	1	45	118	311	665	0.236	14.5%	17.7%

```
get_prop(df, division, volunteer)
```

division	volunteer	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
East North Central	1	181	302	464	820	0.475	39.0%	36.8%
East South Central	1	64	105	222	328	0.484	28.8%	32.0%
Middle Atlantic	1	34	254	128	783	0.221	26.6%	32.4%
Mountain	1	62	124	175	396	0.384	35.4%	31.3%
New England	1	15	78	39	274	0.275	38.5%	28.5%
Pacific	1	26	314	64	1058	0.087	40.6%	29.7%
South Atlantic	1	173	538	543	1619	0.592	31.9%	33.2%
West North Central	1	139	134	297	342	0.063	46.8%	39.2%
West South Central	1	89	163	311	665	0.198	28.6%	24.5%

9.3.3 Division: Caregiver and Multiple

```
get_prop(df, division, caregiver)
```

division	caregiver	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
East North Central	1	57	75	464	820	0.092	12.3%	9.1%
East South Central	1	34	35	222	328	0.138	15.3%	10.7%
Middle Atlantic	1	16	72	128	783	0.312	12.5%	9.2%
Mountain	1	20	42	175	396	0.884	11.4%	10.6%
New England	1	2	19	39	274	0.936	5.1%	6.9%
Pacific	1	13	109	64	1058	0.022	20.3%	10.3%
South Atlantic	1	46	154	543	1619	0.523	8.5%	9.5%
West North Central	1	27	34	297	342	0.818	9.1%	9.9%
West South Central	1	34	68	311	665	0.823	10.9%	10.2%

```
get_prop(df, division, multi)
```

division	multi	n_Rural	n_Urban	sum_Rural	sum_Urban	p	rural_prop	urban_prop
East North Central	1	256	401	464	820	0.036	55.2%	48.9%
East South Central	1	107	159	222	328	1.000	48.2%	48.5%
Middle Atlantic	1	51	387	128	783	0.055	39.8%	49.4%
Mountain	1	93	184	175	396	0.167	53.1%	46.5%
New England	1	21	120	39	274	0.313	53.8%	43.8%
Pacific	1	34	519	64	1058	0.614	53.1%	49.1%
South Atlantic	1	249	791	543	1619	0.245	45.9%	48.9%
West North Central	1	176	171	297	342	0.024	59.3%	50.0%
West South Central	1	130	282	311	665	0.913	41.8%	42.4%

10 Plots

```r
# Obtain a list of states and FIPS codes
fp <- tidycensus::fips_codes %>%
  as_tibble() %>%
  group_by(state) %>%
  slice(1) %>%
  ungroup() %>%
  select(state_code = state, GEOID = state_code) %>%
  arrange(GEOID)

# Join FIPS codes with Census divisions
rd <- read_csv("private/poster/us_census_regions_and_divisions.csv") %>%
  janitor::clean_names() %>%
  left_join(fp, by = "state_code") %>%
  arrange(GEOID) %>%
  select(GEOID, division)
```

```
## Rows: 51 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): State, State Code, Region, Division
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Get list of states per division
read_csv("private/poster/us_census_regions_and_divisions.csv") %>%
  janitor::clean_names() %>%
  left_join(fp, by = "state_code") %>%
  group_by(division) %>%
  summarize(states = paste0(state_code, collapse = ", "))
```

```
## Rows: 51 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): State, State Code, Region, Division
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## # A tibble: 9 x 2
##   division           states
##   <chr>              <chr>
## 1 East North Central IL, IN, MI, OH, WI
## 2 East South Central AL, KY, MS, TN
## 3 Middle Atlantic    NJ, NY, PA
## 4 Mountain           AZ, CO, ID, MT, NM, NV, UT, WY
## 5 New England        CT, MA, ME, NH, RI, VT
## 6 Pacific            AK, CA, HI, OR, WA
## 7 South Atlantic     DC, DE, FL, GA, MD, NC, SC, VA, WV
## 8 West North Central IA, KS, MN, MO, ND, NE, SD
## 9 West South Central AR, LA, OK, TX
```

```r
# Get proportions of older adults with at least one productive activity by division
multi <- get_kab(df, division, multi) %>%
```

```r
  select(-n, -denom) %>%
  pivot_wider(names_from = "rural", values_from = "pct")
multi
```

```
## # A tibble: 9 x 3
##   division           Rural Urban
##   <chr>              <dbl> <dbl>
## 1 East North Central 0.552 0.489
## 2 East South Central 0.482 0.485
## 3 Middle Atlantic    0.398 0.494
## 4 Mountain           0.531 0.465
## 5 New England        0.538 0.438
## 6 Pacific            0.531 0.491
## 7 South Atlantic     0.459 0.489
## 8 West North Central 0.593 0.5
## 9 West South Central 0.418 0.424
```

```r
# Join data with FIPS/divisions
rd_multi <- rd %>%
  left_join(multi, by = "division")

# Download a states map (filter out Puerto Rico)
# Transform to USA Contiguous Albers Equal Area Conic ('ESRI:102003')
states_tmp <- tigris::states(class = "sf", resolution = "20m", cb = T) %>%
  filter(GEOID != "72") %>% # filter out Puerto Rico (PR)
  shift_geometry()
```

```
## Retrieving data for the year 2020

##   |                                                                              |
```

```r
# Merge sf map with data
states <- states_tmp %>%
  left_join(rd_multi, by = "GEOID")

# Function to produce a bivariate choropleth map
gen_map <- function(mypal) {
  mydim <- 3
  mystyle <- "quantile"
  data <- bi_class(states, x = Rural, y = Urban, style = mystyle, dim = mydim)
  breaks <- bi_class_breaks(states, x = Rural, y = Urban, style = mystyle, dim = mydim)
  map <- ggplot() +
    geom_sf(
      data = data, mapping = aes(fill = bi_class), show.legend = FALSE,
      color = "gray65", size = 0.5
    ) +
    bi_scale_fill(pal = mypal, dim = mydim) +
    # geom_sf_label(data = data, aes(label = NAME)) +
    bi_theme(bg_color = "transparent") +
    theme(
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.background = element_rect(fill = "transparent", colour = NA),
      plot.background = element_rect(fill = "transparent", colour = NA),
      plot.title = element_text(color = "gray80", size = 30, face = "bold"),
      legend.background = element_rect(fill = "transparent", color = NA),
```

```
      legend.position = "bottom",
      plot.margin = margin(0, 200, 0, 0, "pt") # top, right, bottom, left
    )
  default_background_color <- "transparent"
  legend <- bi_legend(
    pal = mypal,
    dim = mydim,
    xlab = "Higher % PE in Rural",
    ylab = "Higher % PE in Urban",
    size = 12
  ) +
    theme(
      plot.background = element_rect(
        fill = default_background_color,
        color = NA
      ),
      panel.background = element_rect(
        fill = default_background_color,
        color = NA
      ),
      legend.background = element_rect(
        fill = default_background_color,
        color = NA
      ),
      text = element_text(color = "white")
    )
  finalPlot <- ggdraw() +
    draw_plot(map, 0, 0, 1, 1) +
    draw_plot(legend, 0.6, .1, 0.35, 0.35)
  finalPlot
}

# Produce map based on given bivariate color scheme
# pals <- c("Bluegill", "BlueGold", "BlueOr", "BlueYl", "Brown", "Brown2",
# "DkBlue", "DkBlue2", "DkCyan", "DkCyan2", "DkViolet", "DkViolet2", "GrPink",
# "GrPink2", "PinkGrn", "PurpleGrn", "PurpleOr")
pals <- "DkBlue"
for (p in pals) {
  # mypal <- "DkBlue2"
  mymap <- gen_map(p)
  print(mymap)
  map_filename <- glue("map_{p}.png")
  ggsave(map_filename,
    plot = mymap, width = 12, height = 7, dpi = 600,
    bg = "transparent"
  )
}
```