

The Impact of Neurology Workforce Measures on State-level Parkinson's Disease Mortality Rates

QBS181 Project Aim 1

Savannah Gonzales, Amy Wang, Yifei Gao, Kath Paul

18 November 2024

DATA SOURCES

- National Plan and Provider Enumeration System (NPPES) Downloadable File
 - Input: `NPPES_FILE`
 - Source: National Bureau of Economic Research's Public Use Data Archive (<https://www.nber.org/research/data/national-plan-and-provider-enumeration-system-nppes>)
 - Download Instructions: Click on link above > scroll to File Downloads > CSV: Monthly > 2022 > `npi202212.csv`
 - NOTE: This file is 8.1GB. To avoid loading this on every knit, we load a file containing the subset of neurologists `NPPES_NEURO_FILE`
- Center for Medicare & Medicaid Services (CMS) Physician Compare National Downloadable File
 - Input: `CMS_FILE`
 - Source: CMS (<https://data.cms.gov/provider-data/archived-data/doctors-clinicians>)
 - Download Instructions: Scroll to 2022 archived data snapshots > Download the doctors_and_clinicians_12_2022.zip file > Extract/un-zip > `DAC_NatioinalDownloadableFile.csv`
- Parkinson's Disease Mortality Rates
 - Input: `PD_MORTALITY_FILE`
 - Source: CDC's National Center for Health Statistics (https://www.cdc.gov/nchs/pressroom/sosmap/parkinsons_disease_mortality/parkinsons_disease.htm)
 - Download Instructions: Select 2022 from drop-down box > Under the data table is a link to download data as a CSV
- US Adult Population Data
 - Input: `CENSUS_POP_FILE`
 - Source: US Census Bureau (<https://www.census.gov/data/tables/time-series/demo/popest/2020s-state-detail.html>)
- Medicare Monthly Enrollment File (for Medicare Beneficiary Population Data)
 - Input: `ENROLLMENT_FILE`
 - Source: CMS (<https://data.cms.gov/summary-statistics-on-beneficiary-enrollment/medicare-and-medicaid-reports/medicare-monthly-enrollment/data>)
 - Download Instructions: On the interactive data viewer, apply the following filters:
 - * YEAR Equals 2022
 - * MONTH Equals Year (specifies rows with annual counts, removes rows for monthly counts)
 - * BENE_GEO_LVL Equals State (specifies state-level rows, removes county-level rows)
- Social Deprivation Index (SDI)
 - Input: `SDI_COUNTY_FILE`
 - Source: The Graham Center (<https://www.graham-center.org/maps-data-tools/social-deprivation-index.html>)
 - Download Instructions: Click the link titled "The 2019 SDI at the County Level" (This was the most recent update at the time of this project)

- Rural-Urban Continuum Codes (RUCA) codes
 - Input: `RUCA_FILE`
 - Source: USDA Economic Research Service (<https://www.ers.usda.gov/data-products/rural-urban-commuting-area-codes/>)
 - Download Instructions: Scroll down to ‘Data Set’ > Select “2010 Rural-Urban Commuting Area Codes, ZIP code file” (for a CSV file, click the second icon)

Important acronyms frequently used in this document:

- NPES: National Plan and Provider Enumeration System
- NPI: National Provider Identifier
- CMS: Centers for Medicare & Medicaid Services
- M-A or MA: Medicare-accepting
- PD: Parkinson’s Disease

SETUP

LIBRARIES

All libraries used are loaded in the code chunk below

```
library(dplyr) # For data wrangling
library(data.table) # Provides `fread` function to read in data faster
library(usmapdata) # Provides geometry/shape files for mapping
library(magrittr) # Provides summarise() function for calculating average
library(ggplot2) # For plotting figures nicely
library(corrplot) # For correlation plots
library(psych) # Adds p-values to correlation plots
```

WORKING DIRECTORY

This will be different for each user. Add your own file path as your working directory.

```
#setwd("/Users/amyw/Documents/docmoenlab")
setwd("~/QBS/QBS181/group_project/")
```

FILE NAMES

Naming conventions often differ between scientists. Replace each file name below with the name (and path) as it is stored on your computer.

```
NPPES_FILE = 'npi202212.csv'
NPPES_NEURO_FILE = 'nppes_neuro.csv'
CMS_FILE = 'CMS_doctors_and_clinicians_12_2022/DAC_NationalDownloadableFile.csv'
PD_MORTALITY_FILE = 'statemortality.csv'
CENSUS_POP_FILE = '18+POP.csv'
ENROLLMENT_FILE = 'Medicare_Enrollment_2022/Medicare_Enrollment_2022.csv'
SDI_COUNTY_FILE = 'rgcsdi-2015-2019-county.csv'
RUCA_FILE = 'RUCA2010zipcode.csv'
```

IMPORTING NPPES DATASET AND FILTER

Selection criteria:

- Individual provider, not organization
- Neurology as primary, secondary, or tertiary specialty
- Must have medical degree (credentials column)

```
# Load data and select only columns of interest
nppes_raw = fread(NPPES_FILE,
                  select = c('npi',          # National provider identifier
                             'entity',       # 1=individual, 0=organizational
                             'ptaxcode1',    # Provider taxonomy codes (primary, secondary, tertiary)
                             'ptaxcode2',
                             'ptaxcode3',
                             'plocstatename',
                             'pcredential', 'pcredentialoth',
                             'ploczip'))

# Only include individual providers (entity=1), not organizational providers like hospital or clinic NPIs
nppes_indv = data.frame(nppes_raw[nppes_raw$entity == '1', ])

# Filter for neurologists using the taxonomy code
neuro_taxonomy_code = c('2084N0400X')

# Subset the data to only include NPIs who report neurology as their primary, secondary,
# or tertiary specialty, and store the results in a new dataframe
nppes_neurologists = subset(nppes_indv, ptaxcode1 %in% neuro_taxonomy_code |
                           ptaxcode2 %in% neuro_taxonomy_code |
                           ptaxcode3 %in% neuro_taxonomy_code)

# Filter data to only include clinicians with medical degrees
nppes_neuro_md = nppes_neurologists[with(nppes_neurologists,
                                         grepl("MD|M.D.", paste(pcredential, pcredentialoth))),]

# Create column to store 5-digit zip codes (from the ploczip column)
nppes_neuro_md$zipcode = substr(nppes_neuro_md$ploczip, 0, 5)

# Stored results into csv to avoid re-running this code chunk on every knit
#write.csv(nppes_neuro_md, 'neuro_npis.csv')
```

NOTE: The code chunk above is not run when knitting because the NPPES file is 8.1GB. Instead, we saved the results from this code chunk and load it as a CSV below.

```
# Load file of neurologist NPIs
neuro_npis = read.csv(NPPES_NEURO_FILE)
```

ADDING MEDICARE DATA

```
# Load CMS Physician Compare file (Medicare), only need NPI column
cms_npis = read.csv(CMS_FILE) %>% select('NPI')

# Check list of CMS NPIs for duplicates
dim(cms_npis) # Total of 2469709 npis
n_distinct(cms_npis$NPI) # 1247657 distinct npis, doesn't match above, bad

# Create list of unique CMS NPIs (don't include duplicates)
```

```

cms_npis_unique = unique(cms_npis$NPI)
length(cms_npis_unique) # 1247657, matches above, good

# Add column to neurologist NPI df indicating if each NPI is present in CMS NPI list
# (aka create binary variable indicating whether each NPI accepts Medicare)
neuro_npis$accepts_medicare = neuro_npis$npi %in% cms_npis_unique

sum(neuro_npis$accepts_medicare) # Total of 12904 neurologists accept Medicare
n_distinct(neuro_npis$npi) # Out of a total of 18,490 total neurologists
# This equals 69.78908% of all neurologists accepts Medicare

# Create new df for MA providers (subset to only include MA NPIs)
neuro_MA_npis = neuro_npis[neuro_npis$accepts_medicare,]

# Check that the new df only includes MA NPIs and matches number above
sum(neuro_MA_npis$accepts_medicare) #12904 good!
dim(neuro_MA_npis) #12904 good!

```

We now have two primary dataframes of NPIs to work with:

1. `neuro_npis` : a df of neurologists with columns for provider NPI, taxonomy codes (specialties), credentials, geographic identifiers (states, ZIP Codes, zctas), RUCA, and SDI.
2. `neuro_MA_npis` : a df of neurologists who accept Medicare insurance, with the same columns as `neuro_npis`

In the next step, we will calculate state-level measures and create columns for ‘all’ and ‘M-A’ neurologists in the same dataframe.

CALCULATE NEUROLOGIST SUPPLY BY STATE

Supply for all neurologists (`neuro_npis`)

```

# Convert to table then back to df to count neurologist supply
state_neuro_supply = table(neuro_npis$plocstatename) %>% as.data.frame %>% arrange(desc(Freq))

# Rename columns
names(state_neuro_supply)[1:2] = c('State', 'Neurologist_Supply')

# Manually filter out non-US states
state_neuro_supply = state_neuro_supply %>%
  filter(!(State %in% c('AA', 'AE', 'BC', 'CEARA', 'DC', 'GU', 'ISRAEL', 'KERALA', 'NEWFOUNDLAND',
    'NOVA SCOTIA', 'ON', 'ONTARIO', 'PR', 'SONORA', 'VI'))))

```

Repeat for M-A neurologists (`neuro_MA_npis`)

```

# Count MA neurologists per state, store in new df
state_MA_neuro_supply = table(neuro_MA_npis$plocstatename) %>% as.data.frame %>% arrange(desc(Freq))

# Rename columns
names(state_MA_neuro_supply)[1:2] = c('State', 'MA_Neurologist_Supply')

# Manually filter out non-US states
state_MA_neuro_supply = state_MA_neuro_supply %>%
  filter(!(State %in% c('AA', 'AE', 'BC', 'CEARA', 'DC', 'GU', 'ISRAEL', 'KERALA', 'NEWFOUNDLAND',
    'NOVA SCOTIA', 'ON', 'ONTARIO', 'PR', 'SONORA', 'VI'))))

```

Combine these two dataframes to create one dataframe containing all state-level measures

```
# Join by state
state_neuro = full_join(state_neuro_supply, state_MA_neuro_supply, by='State')
```

Calculate the percentage of neurologists that accept Medicare in each state, and store as new column for later visualizations.

```
# Round to 2 digits and multiply by 100 to get percentage
state_neuro$Percentage_MA = round(state_neuro$MA_Neurologist_Supply/state_neuro$Neurologist_Supply, 2)*100
head(state_neuro)
```

| | State | Neurologist_Supply | MA_Neurologist_Supply | Percentage_MA |
|---|-------|--------------------|-----------------------|---------------|
| 1 | CA | 2044 | 1346 | 66 |
| 2 | NY | 1601 | 1122 | 70 |
| 3 | FL | 1214 | 909 | 75 |
| 4 | TX | 1188 | 844 | 71 |
| 5 | MA | 879 | 574 | 65 |
| 6 | PA | 827 | 571 | 69 |

ADD PARKINSON'S DISEASE MORTALITY DATA

```
# Read in PD mortality data
mortality = read.csv(PD_MORTALITY_FILE, colClasses=c(YEAR = 'character'))

# Filter to only 2022 data
mortality2022 = mortality %>% filter(YEAR %in% c('2022')) %>%
  # Remove DC (don't have sufficient agricultural data, decided to exclude from all analyses)
  filter(!(STATE %in% c('District of Columbia'))) %>%
  # Remove year and url columns since we won't use them
  select(-YEAR, -URL)

# Add mortality data to df of state-level measures (state_neuro)
state_neuro = full_join(state_neuro, mortality2022, by=c('State'='STATE'))

# Rename rate and death columns to specify PD
colnames(state_neuro)[5:6] = c('PD_Mortality_Rate', 'PD_Deaths')

# Change mortality values to numeric and remove commas
state_neuro$PD_Deaths = as.numeric(gsub(',', '', state_neuro$PD_Deaths))

head(state_neuro)
```

| | State | Neurologist_Supply | MA_Neurologist_Supply | Percentage_MA |
|---|-------|--------------------|-----------------------|---------------|
| 1 | CA | 2044 | 1346 | 66 |
| 2 | NY | 1601 | 1122 | 70 |
| 3 | FL | 1214 | 909 | 75 |
| 4 | TX | 1188 | 844 | 71 |
| 5 | MA | 879 | 574 | 65 |
| 6 | PA | 827 | 571 | 69 |

| | PD_Mortality_Rate | PD_Deaths |
|---|-------------------|-----------|
| 1 | 9.4 | 4289 |
| 2 | 6.5 | 1751 |
| 3 | 8.5 | 3169 |
| 4 | 11.1 | 3091 |
| 5 | 8.0 | 747 |
| 6 | 9.2 | 1725 |

CALCULATE NEUROLOGIST DENSITY

READ IN CENSUS POPULATION DATA

```
# Read in US Census population data
popdata = read.csv(CENSUS_POP_FILE, header=T)

# Filter data for only states we're interested in (50 US states, exclude DC row 17)
pop50states = popdata[c(9:16,18:59), c(1,3)]

# Rename columns
colnames(pop50states) = c('State','PopulationOver18')

# Add state abbreviations
pop50states$StateAbbrev = datasets::state.abb

# Select only columns for abbreviated state and population
pop50states = select(pop50states, StateAbbrev, PopulationOver18)

# Join with state mortality and neurologist df
state_neuro = left_join(state_neuro, pop50states, join_by('State'=='StateAbbrev'))

# Change population values to numeric and remove commas
state_neuro$PopulationOver18 = as.numeric(gsub(',', '',state_neuro$PopulationOver18))
```

READ IN CMS MEDICARE ENROLLMENT DATA

```
# Read in CMS Medicare Enrollment Data
medicare_enrollment = read.csv(ENROLLMENT_FILE) %>%
  # Select only columns for abbreviated state and total number of beneficiaries
  select('BENE_STATE_ABRVTN', 'TOT_BENES')

# Rename columns
colnames(medicare_enrollment) = c('State','Total_Medicare_Beneficiaries')

# Join with state-level measures df
state_neuro = left_join(state_neuro, medicare_enrollment, by='State')
# Note: left_join automatically removes territories not in our state_neuro df, including DC
# Double check that we have 50 rows
dim(state_neuro) # 50, 8 -- good!

# Change values to numeric and remove commas
state_neuro$Total_Medicare_Beneficiaries = as.numeric(state_neuro$Total_Medicare_Beneficiaries)
```

CALCULATE NEUROLOGIST DENSITY PER 100K PER STATE

```
# Calculate the number of neurologists per 100,000 people over 18 years old
state_neuro$NeurologistDensityPer100k =
  round((state_neuro$Neurologist_Supply/state_neuro$PopulationOver18)*100000, digits=2)

# Repeat for MA Neurologist density, use number of beneficiaries enrolled instead of overall population
state_neuro$MA_NeurologistDensityPer100k =
  round((state_neuro$MA_Neurologist_Supply/state_neuro$Total_Medicare_Beneficiaries)*100000, digits=2)

head(state_neuro, 3)
```

| | State | Neurologist_Supply | MA_Neurologist_Supply | Percentage_MA |
|---|-------|--------------------|-----------------------|---------------|
| 1 | CA | 2044 | 1346 | 66 |
| 2 | NY | 1601 | 1122 | 70 |
| 3 | FL | 1214 | 909 | 75 |

| | PD_Mortality_Rate | PD_Deaths | PopulationOver18 | Total_Medicare_Beneficiaries |
|---|-------------------|-----------|------------------|------------------------------|
| 1 | 9.4 | 4289 | 30519524 | 6625047 |
| 2 | 6.5 | 1751 | 15611308 | 3761501 |
| 3 | 8.5 | 3169 | 18229883 | 4921208 |

| | NeurologistDensityPer100k | MA_NeurologistDensityPer100k |
|---|---------------------------|------------------------------|
| 1 | 6.70 | 20.32 |
| 2 | 10.26 | 29.83 |
| 3 | 6.66 | 18.47 |

ADDING SDI DATA

```
# Read in SDI data, convert county FIPS codes to character class
sdiByCounty = read.csv(SDI_COUNTY_FILE, colClasses=c(COUNTY_FIPS = 'character')) %>%
  # Select only the necessary columns
  select(COUNTY_FIPS, SDI_score)

# Load state FIPS data to use as crosswalk (comes from usmapdata library)
state_FIPS = fips_data('states')

# Extract the first two digits of the county FIPS codes, since these determine the state
# i.e. Texas state FIPS code is 48, so all counties in Texas have FIPS codes beginning with 48
sdiByCounty$two_digit_fips = substr(sdiByCounty$COUNTY_FIPS, 0, 2)

#joining to correlate SDI scores per county with the state
# Use crosswalk to add state abbreviations
sdiByCounty = left_join(sdiByCounty, state_FIPS, by=join_by('two_digit_fips'=='fips'))

# Calculate average SDI scores per state
sdiByState = sdiByCounty %>% select(abbr, SDI_score) %>%
  # Group by state
  group_by(abbr) %>%
  # Store average in new column named 'Average_SDI'
  summarise(Average_SDI = mean(SDI_score))

# Join state-level SDI scores to working df
state_neuro = left_join(state_neuro, sdiByState, by=c('State'='abbr'))

head(state_neuro, 3)
```

| | State | Neurologist_Supply | MA_Neurologist_Supply | Percentage_MA |
|---|-------|--------------------|-----------------------|---------------|
| 1 | CA | 2044 | 1346 | 66 |
| 2 | NY | 1601 | 1122 | 70 |
| 3 | FL | 1214 | 909 | 75 |

| | PD_Mortality_Rate | PD_Deaths | PopulationOver18 | Total_Medicare_Beneficiaries |
|---|-------------------|-----------|------------------|------------------------------|
| 1 | 9.4 | 4289 | 30519524 | 6625047 |
| 2 | 6.5 | 1751 | 15611308 | 3761501 |
| 3 | 8.5 | 3169 | 18229883 | 4921208 |

| | NeurologistDensityPer100k | MA_NeurologistDensityPer100k | Average_SDI |
|---|---------------------------|------------------------------|-------------|
| 1 | 6.70 | 20.32 | 60.87931 |
| 2 | 10.26 | 29.83 | 44.72581 |
| 3 | 6.66 | 18.47 | 54.00000 |

ADDING RUCA DATA

```
# Read in RUCA file with the zip code column as character to keep leading zeros
ruca = read.csv(RUCA_FILE)

# Rename first column because it loads as "ZIP_CODE" or X..ZIP_CODE.. because of the quotes
colnames(ruca)[1] = 'ZIP_CODE'

# Calculate average per state
# We use RUCA1 because it reflects the primary commuting pattern in the geographic area
state_ruca = ruca %>% select(STATE, RUCA1) %>%
  group_by(STATE) %>%
  # Store average in new column named 'Average_RUCA' rounded to 2 decimal places
  summarise(Average_RUCA = round(mean(RUCA1), 2))

# Join state-level RUCA to working df
state_neuro = left_join(state_neuro, state_ruca, by=c('State'='STATE'))

head(state_neuro, 3)
```

| | State | Neurologist_Supply | MA_Neurologist_Supply | Percentage_MA |
|---|-------|--------------------|-----------------------|---------------|
| 1 | CA | 2044 | 1346 | 66 |
| 2 | NY | 1601 | 1122 | 70 |
| 3 | FL | 1214 | 909 | 75 |

| | PD_Mortality_Rate | PD_Deaths | PopulationOver18 | Total_Medicare_Beneficiaries |
|---|-------------------|-----------|------------------|------------------------------|
| 1 | 9.4 | 4289 | 30519524 | 6625047 |
| 2 | 6.5 | 1751 | 15611308 | 3761501 |
| 3 | 8.5 | 3169 | 18229883 | 4921208 |

| | NeurologistDensityPer100k | MA_NeurologistDensityPer100k | Average_SDI |
|---|---------------------------|------------------------------|-------------|
| 1 | 6.70 | 20.32 | 60.87931 |
| 2 | 10.26 | 29.83 | 44.72581 |
| 3 | 6.66 | 18.47 | 54.00000 |

| | Average_RUCA |
|---|--------------|
| 1 | 2.56 |
| 2 | 3.48 |
| 3 | 1.78 |

STATISTICAL ANALYSIS

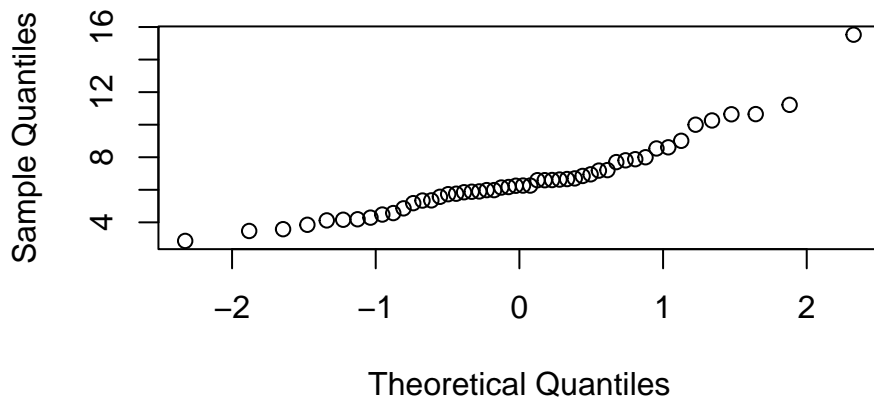
We want to assess the association between the neurologist workforce measures (supply, density, MA supply, MA density) and the state-level PD mortality rates.

Our initial plan is to use Pearson's Correlation Coefficient, which relies on the assumption that our data is normally distributed.

CHECK ASSUMPTIONS

```
# Visually check if our data is normally distributed
qqnorm(state_neuro$NeurologistDensityPer100k)
```


Normal Q-Q Plot



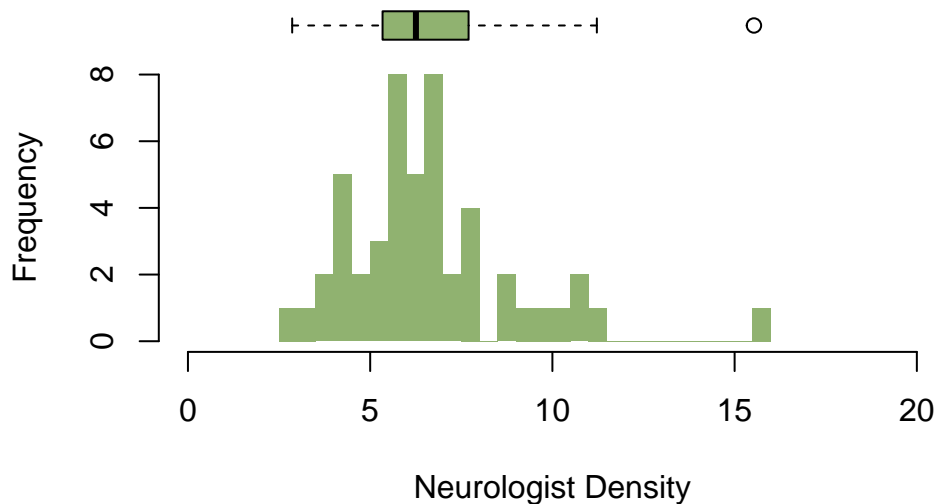
```
# Use the Shapiro-Wilk test to objectively determine if data is normally distributed
shapiro.test(state_neuro$NeurologistDensityPer100k) # not normally distributed
```

Shapiro-Wilk normality test

```
data: state_neuro$NeurologistDensityPer100k
W = 0.90731, p-value = 0.0008451
```

```
# Make combined boxplot and histogram for presentation slides
layout(mat=matrix(c(1,2),2,1, byrow=TRUE), height=c(1,4))
par(mar=c(0, 4.5, 1, 0))
boxplot(state_neuro$NeurologistDensityPer100k, horizontal=T, ylim=c(0,20), axes=F, col='#90b270',
        main='Distribution of Neurologist Density')
par(mar=c(4.5, 4.5, 0, 0))
hist(state_neuro$NeurologistDensityPer100k, breaks=25, col='#90b270', border=F,
     xlab="Neurologist Density", main="", xlim=c(0,20))
```

Distribution of Neurologist Density



```
# Need both variables to be normally distributed, but stopped after the first failed
```

Since our Neurologist density data isn't normally distributed, it fails the required assumptions for Pearson's Correlation Coefficient. Instead, we will calculate Spearman's Correlation Coefficient.

SPEARMAN'S CORRELATION

```
# Calculate spearman pair-wise correlations and store results
corr_results = corr.test(state_neuro[,1:12] %>% select(-State), method='spearman')

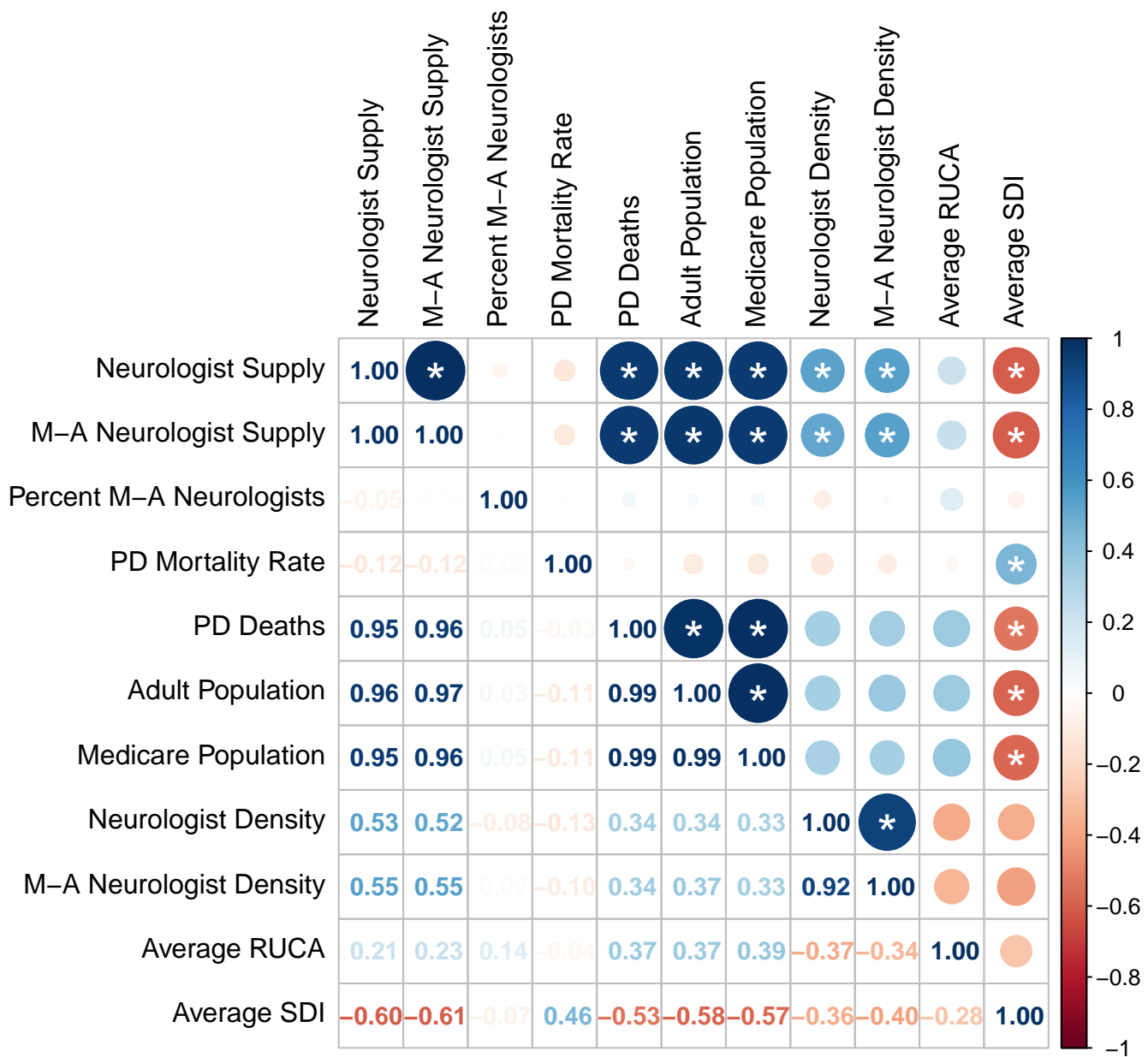
# `r` indicates the correlation matrix, store in a more descriptive variable
corr_matrix = corr_results$r

# `p` indicates p-values
corr_pval = corr_results$p

# Change labels of each column and row to make plot look nice
corr_names = c("Neurologist Supply", "M-A Neurologist Supply", "Percent M-A Neurologists",
               "PD Mortality Rate", "PD Deaths", "Adult Population", "Medicare Population",
               "Neurologist Density", "M-A Neurologist Density", "Average RUCA", "Average SDI")
colnames(corr_matrix) = corr_names
rownames(corr_matrix) = corr_names
colnames(corr_pval) = corr_names
rownames(corr_pval) = corr_names

# Specifications for correlation plot (both are plotted on the same figure)
corrplot(corr_matrix, type = "upper", # Upper triangle of plot
         method = "circle", # Size of circles shows absolute values of correlation coefficient
         tl.pos = "lt", tl.col = "black", # Text color and position
         tl.cex=1, # Text size scaling
         # Adds white stars for significant correlations, according to p-values stored above
         p.mat = corr_pval, insig = "label_sig", pch.col = "white", pch.cex = 2, diag = T,)

corrplot(corr_matrix, type="lower", # Lower triangle of plot
         method = "number", # Adds the correlation coefficient to each cell as a number
         add = T, # Adds plot to the one above
         diag = T, # Does not add diagonal values
         cl.pos = "n", tl.pos = "n", # Hides legend and text, since already on plot above
         number.cex = 0.9) # Scales size of numeric text
```



CALCULATE RATIOS

```
state_neuro$mortality_to_density_ratio =
  state_neuro$PD_Mortality_Rate/state_neuro$NeurologistDensityPer100k
state_neuro$deaths_to_supply_ratio =
  state_neuro$PD_Deaths/state_neuro$Neurologist_Supply

state_neuro$mortality_to_MA_density_ratio =
  state_neuro$PD_Mortality_Rate/state_neuro$MA_NeurologistDensityPer100k
state_neuro$deaths_to_MA_supply_ratio =
  state_neuro$PD_Deaths/state_neuro$MA_Neurologist_Supply
```

MAPPING

GEOMETRY SETUP

```
# Load state map geometry data
state_geom = us_map(regions='state')

# Add state geometry column to df of state measures
state_neuro = left_join(state_neuro, state_geom %>% select(abbr, geom), by=join_by(State==abbr))
# head(state_neuro)
```

COLORS SETUP

```
# Create list of colors for mapping
map_colors = c('#FAF3DD', '#C8D5B9', '#8FC0A9', '#68B0AB', '#4A7C59')
population_colors = c('#F1E7A5', '#EFD56C', '#F0B71A', '#E5830C', '#C2510A')
mortality_colors = c('#feebe2', '#fbb4b9', '#f768a1', '#c51b8a', '#7a0177')
ratio_colors = c('#809964', '#cddbb7', '#e4dfca', '#d9b0b1', '#E791AA')
RUCA_colors = c('#ffffd4', '#fed98e', '#fe9929', '#d95f0e', '#993404')
SDI_colors = c('#EFF3FF', '#C6DBEF', '#9ECAE1', '#3182bd', '#08519C')
```

LEGEND SETUP

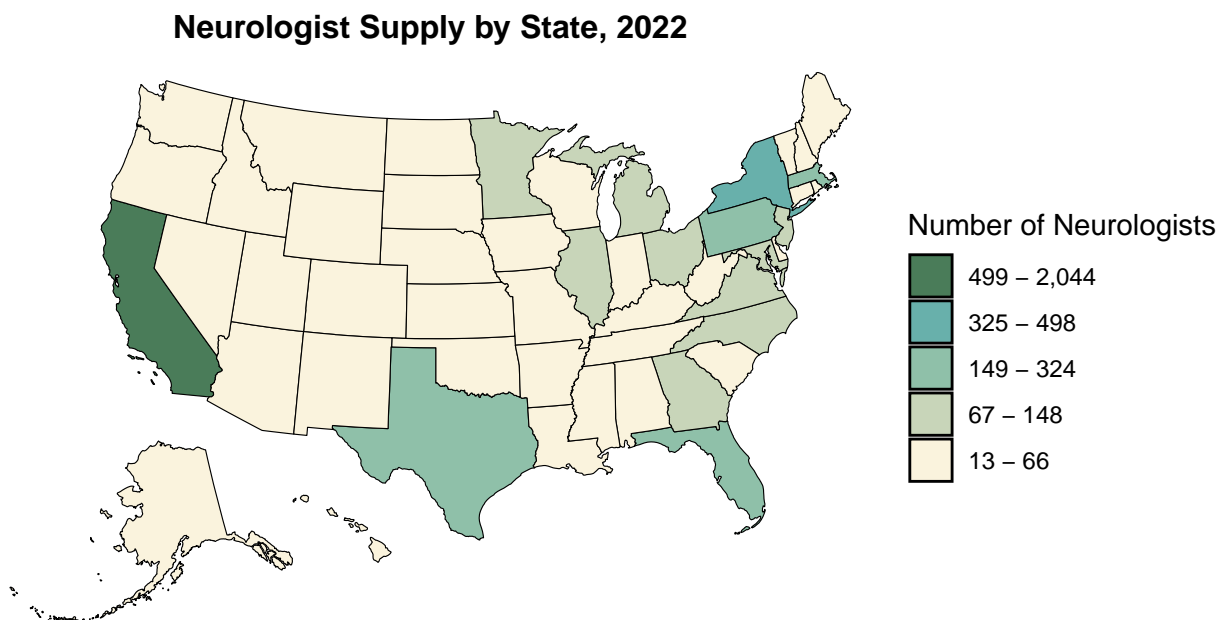
```
# Function to make ranges for pretty legends
ranges <- function(d, count_var, as_integer=TRUE, num_cats=5) {
  d$counts = count_var # Add count_variable to temp df
  d$cat = ntile(count_var, num_cats) # Assign number of categories to count_variable
  lower = c()
  upper = c()
  for (i in 1:num_cats) {
    # Calculate lower/upper bounds for each category
    lower = c(lower, min(d[d$cat==i,]$counts))
    upper = c(upper, max(d[d$cat==i,]$counts))
  }
  # Depending on input, round to integer/decimal and add 1/0.1 to create lower bounds
  if (as_integer) {
    # Ensure no decimals
    lower = as.integer(lower)
    upper = as.integer(upper)
    # Make lower bounds match previous upper bounds (+1), add commas to large numbers
    lower = prettyNum(c(lower[1], upper+1), big.mark=',')
    upper = prettyNum(upper, big.mark=',')
  } else {
    # Round decimals to one digit
    lower = round(lower, 1)
    upper = round(upper, 1)
    # Make lower bounds match previous upper bounds (+0.1), add commas to large numbers
    lower = prettyNum(c(lower[1], upper+0.1), big.mark=',')
    upper = prettyNum(upper, big.mark=',')
  }
  # Create list of ranges as nicely format strings
  ls = paste0(lower, ' - ', upper)
  # If smallest range is 0-0, replace with just 0 to look nicer
  if (ls[1]=='0 - 0') {ls[1]<-'0'}
  return(ls[1:5])
}
```

MAP WORKFORCE MEASURES BY STATE

```
# Overall Neurologist Supply

# Use ranges functions to create categorical values for the variable you want to map
state_neuro$Neurologist_Supply_cat = cut(state_neuro$Neurologist_Supply, breaks=5,
                                          labels=ranges(state_neuro,
                                                         state_neuro$Neurologist_Supply,
                                                         as_integer=T))

ggplot(data=state_neuro, aes(fill=Neurologist_Supply_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=map_colors, name='Number of Neurologists',
                   labels=ranges(state_neuro, state_neuro$Neurologist_Supply, as_integer=T),
                   guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Neurologist Supply by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

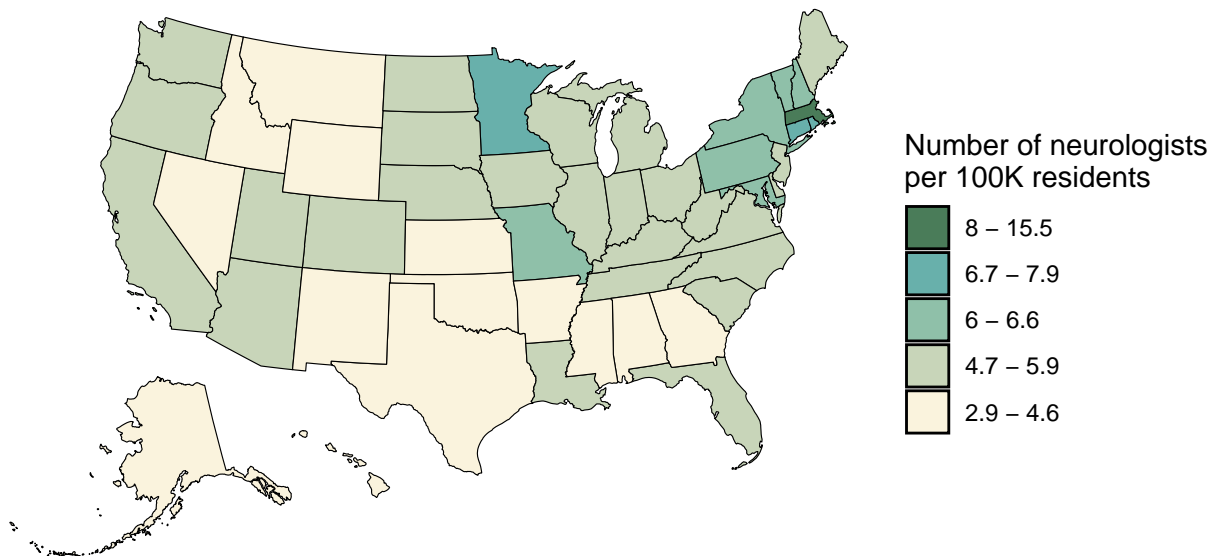


Overall Neurologist Density

```
state_neuro$NeurologistDensityPer100k_cat = cut(state_neuro$NeurologistDensityPer100k, breaks=5,
                                                labels=ranges(state_neuro,
                                                                state_neuro$NeurologistDensityPer100k,
                                                                as_integer=F))

ggplot(data=state_neuro, aes(fill=NeurologistDensityPer100k_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=map_colors, name='Number of neurologists \nper 100K residents',
                    labels=ranges(state_neuro, state_neuro$NeurologistDensityPer100k, as_integer=F),
                    guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Neurologist Density by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

Neurologist Density by State, 2022



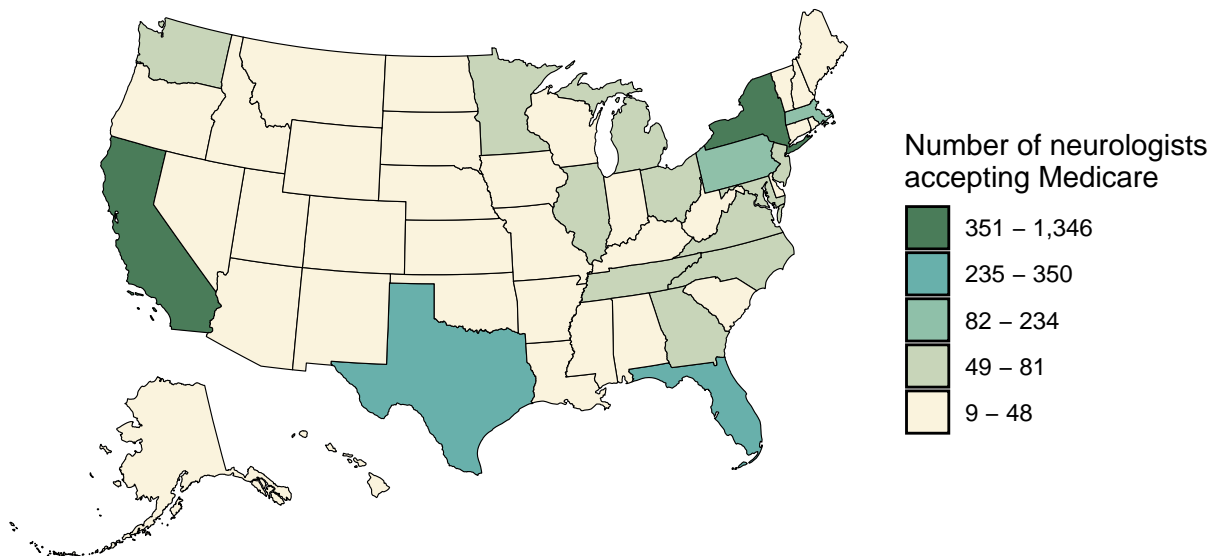
```

# M-A Neurologist Supply
state_neuro$MA_Neurologist_Supply_cat = cut(state_neuro$MA_Neurologist_Supply, breaks=5,
                                             labels=ranges(state_neuro,
                                                             state_neuro$MA_Neurologist_Supply,
                                                             as_integer=T))

ggplot(data=state_neuro, aes(fill=MA_Neurologist_Supply_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=map_colors, name='Number of neurologists \naccepting Medicare',
                    labels=ranges(state_neuro, state_neuro$MA_Neurologist_Supply, as_integer=T),
                    guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Medicare-Accepting Neurologist Supply by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))

```

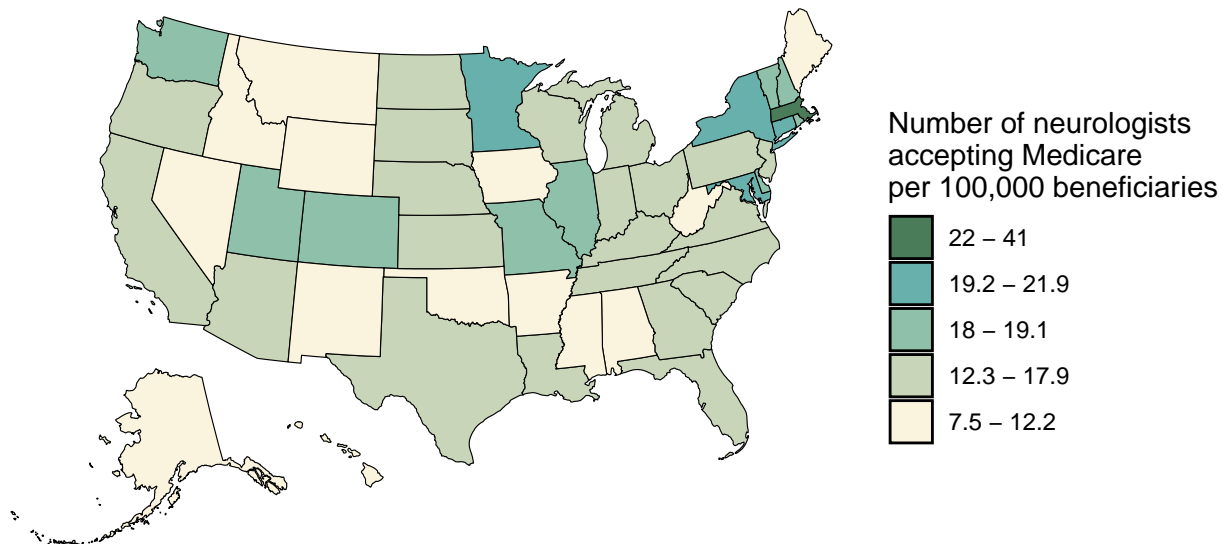
Medicare–Accepting Neurologist Supply by State, 2022



```
# M-A Neurologist Density
state_neuro$MA_NeurologistDensityPer100k_cat = cut(state_neuro$MA_NeurologistDensityPer100k, breaks=5,
                                                    labels=ranges(state_neuro,
                                                                    state_neuro$MA_NeurologistDensityPer100k,
                                                                    as_integer=T))

ggplot(data=state_neuro, aes(fill=MA_NeurologistDensityPer100k_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=map_colors,
                    name='Number of neurologists \naccepting Medicare \nper 100,000 beneficiaries',
                    labels=ranges(state_neuro, state_neuro$MA_NeurologistDensityPer100k, as_integer=F),
                    guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Medicare-Accepting Neurologist Density by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

Medicare–Accepting Neurologist Density by State, 2022

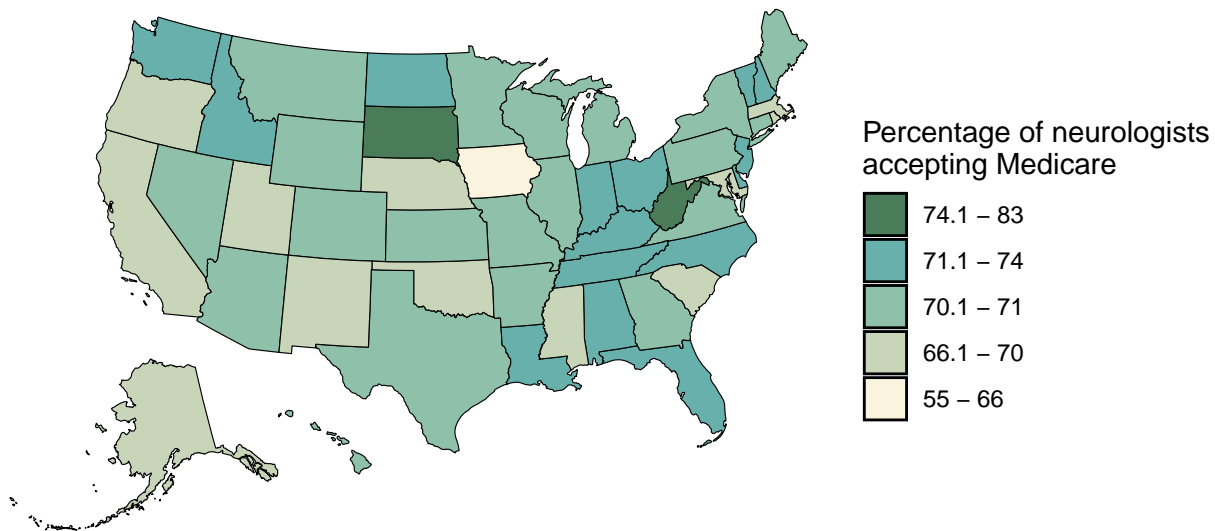


MAP PERCENTAGE OF NEUROLOGISTS ACCEPTING MEDICARE BY STATE

```
# Percent M-A Neurologists
state_neuro$Percentage_MA_cat = cut(state_neuro$Percentage_MA, breaks=5,
                                   labels=ranges(state_neuro,
                                                  state_neuro$Percentage_MA,
                                                  as_integer=T))

ggplot(data=state_neuro, aes(fill=Percentage_MA_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=map_colors, name='Percentage of neurologists \naccepting Medicare',
                   labels=ranges(state_neuro, state_neuro$Percentage_MA, as_integer=F),
                   guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Percentage of Medicare-Accepting Neurologists by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 10))
```

Percentage of Medicare–Accepting Neurologists by State, 2022

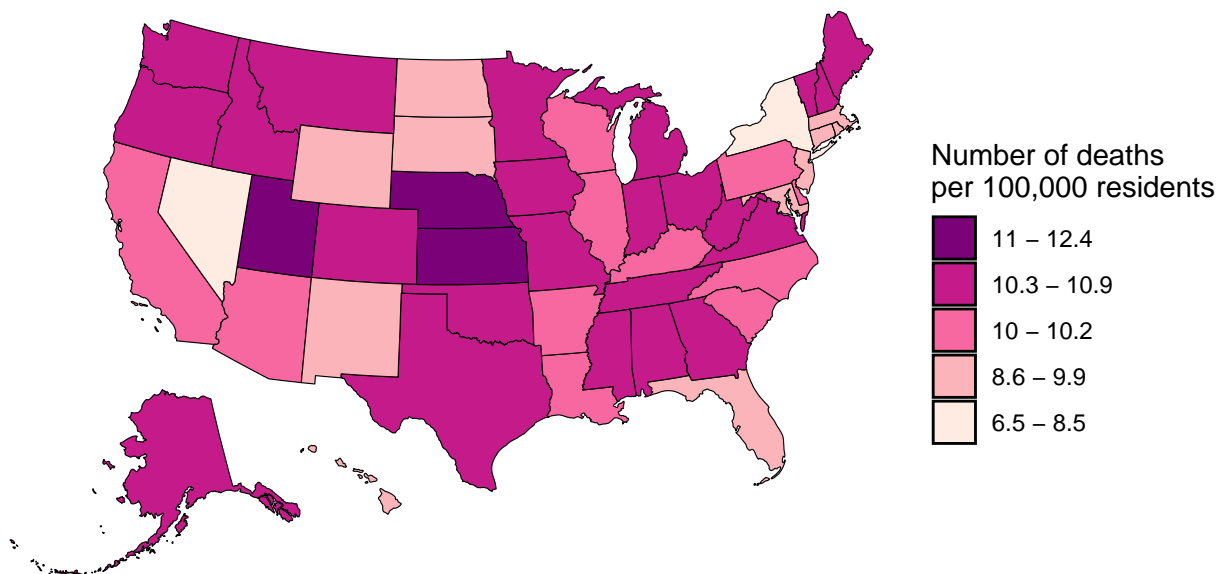


MAP PD MORTALITY BY STATE

```
# PD Mortality Rates
state_neuro$PD_Mortality_Rate_cat = cut(state_neuro$PD_Mortality_Rate, breaks=5,
                                         labels=ranges(state_neuro,
                                                         state_neuro$PD_Mortality_Rate,
                                                         as_integer=F))

ggplot(data=state_neuro, aes(fill=PD_Mortality_Rate_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=mortality_colors, name='Number of deaths \nper 100,000 residents',
                    labels=ranges(state_neuro, state_neuro$PD_Mortality_Rate, as_integer=F),
                    guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Parkinson\'s Disease Mortality Rates by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

Parkinson's Disease Mortality Rates by State, 2022



```

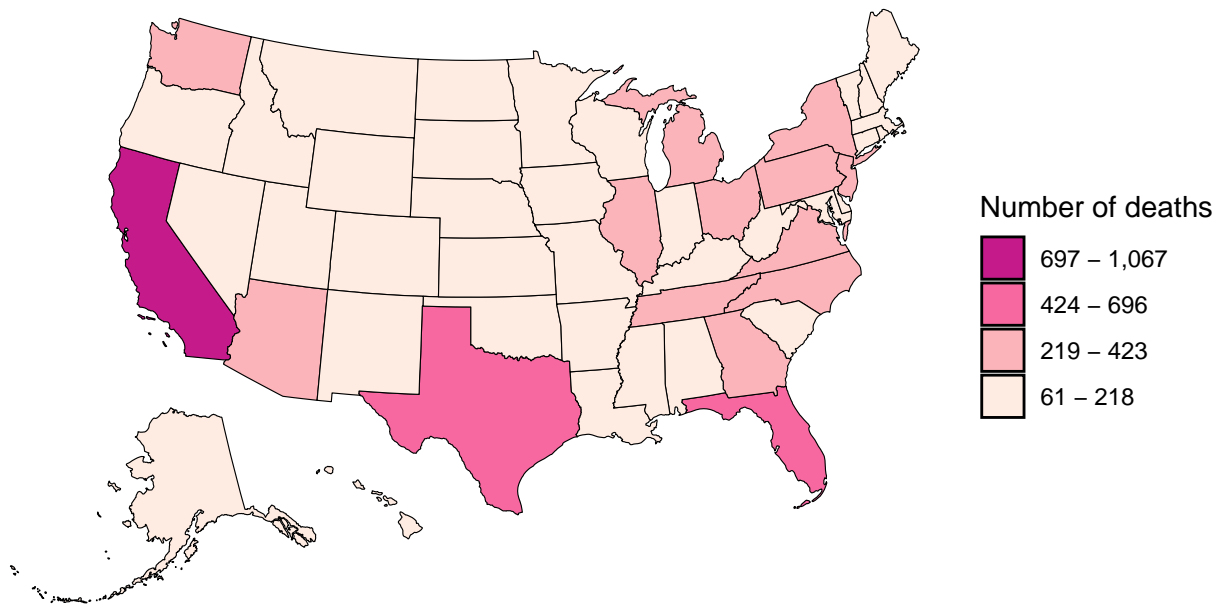
# Number of PD Deaths
state_neuro$PD_Deaths_cat = cut(state_neuro$PD_Deaths, breaks=5,
                                labels=ranges(state_neuro,
                                                state_neuro$PD_Deaths,
                                                as_integer=F))

ggplot(data=state_neuro, aes(fill=PD_Deaths_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=mortality_colors, name='Number of deaths',
                    labels=ranges(state_neuro, state_neuro$PD_Deaths, as_integer=T),
                    guide=guide_legend(reverse=T)) +

  theme_void() +
  ggtitle('Parkinson\'s Disease Deaths by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))

```

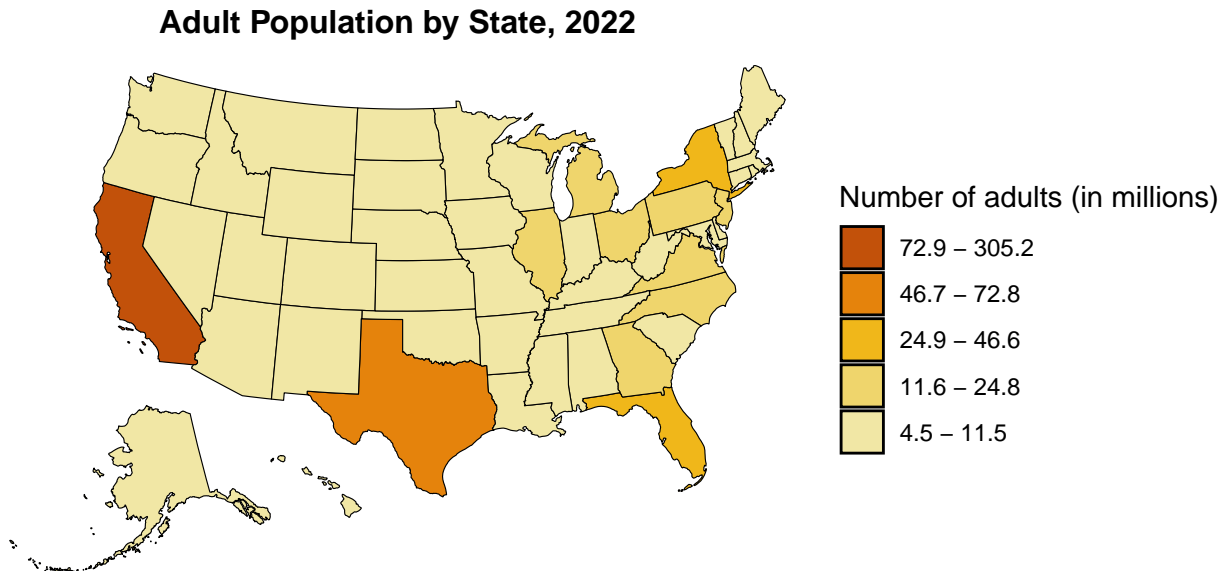
Parkinson's Disease Deaths by State, 2022



MAP STATE POPULATIONS

```
# Census population data (adults over 18 years old)
state_neuro$PopulationOver18_cat = cut(state_neuro$PopulationOver18/1000000, breaks=5,
                                       labels=ranges(state_neuro,
                                                      state_neuro$PopulationOver18/1000000,
                                                      as_integer=F))

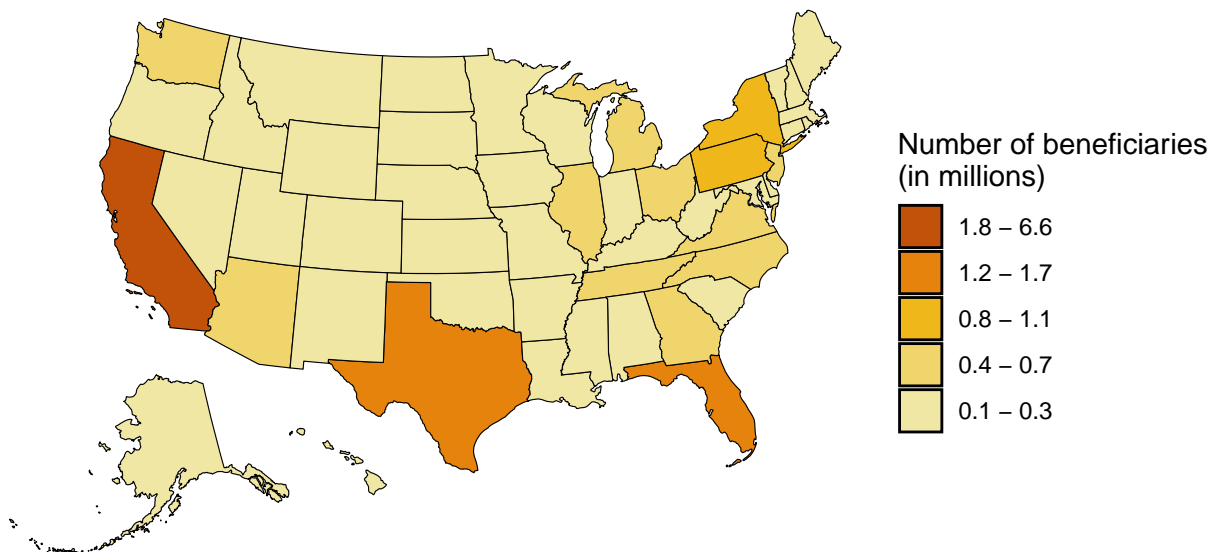
ggplot(data=state_neuro, aes(fill=PopulationOver18_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=population_colors, name='Number of adults (in millions)',
                   labels=ranges(state_neuro, state_neuro$PopulationOver18/100000, as_integer=F),
                   guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Adult Population by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```



```
# Medicare Beneficiaries population data
state_neuro$Total_Medicare_Beneficiaries_cat = cut(state_neuro$Total_Medicare_Beneficiaries/1000000,
                                                    breaks=5,
                                                    labels=ranges(state_neuro,
                                                                    state_neuro$Total_Medicare_Beneficiaries/1000000,
                                                                    as_integer=F))

ggplot(data=state_neuro, aes(fill=Total_Medicare_Beneficiaries_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=population_colors, name='Number of beneficiaries \n(in millions)',
                    labels=ranges(state_neuro,
                                state_neuro$Total_Medicare_Beneficiaries/1000000, as_integer=F),
                    guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Medicare Beneficiary Population by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

Medicare Beneficiary Population by State, 2022



```

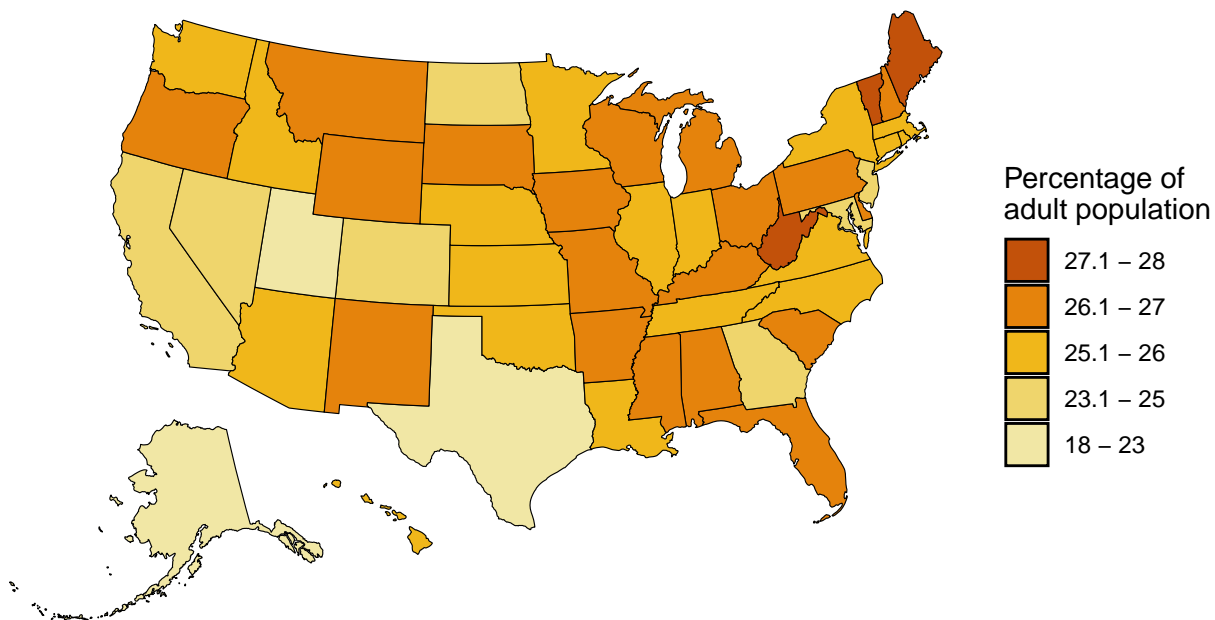
# Percentage of adult population on Medicare
state_neuro$Percent_Beneficiaries =
  round(state_neuro$Total_Medicare_Beneficiaries/state_neuro$PopulationOver18, 2)*100

state_neuro$Percent_Beneficiaries_cat = cut(state_neuro$Percent_Beneficiaries, breaks=5,
                                           labels=ranges(state_neuro,
                                                           state_neuro$Percent_Beneficiaries,
                                                           as_integer=F,
                                                           num_cats=6))

ggplot(data=state_neuro, aes(fill=Percent_Beneficiaries_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=population_colors, name='Percentage of \nadult population',
                   labels=ranges(state_neuro, state_neuro$Percent_Beneficiaries, as_integer=F, num_cats=6),
                   guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Percentage of Adults with Medicare by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))

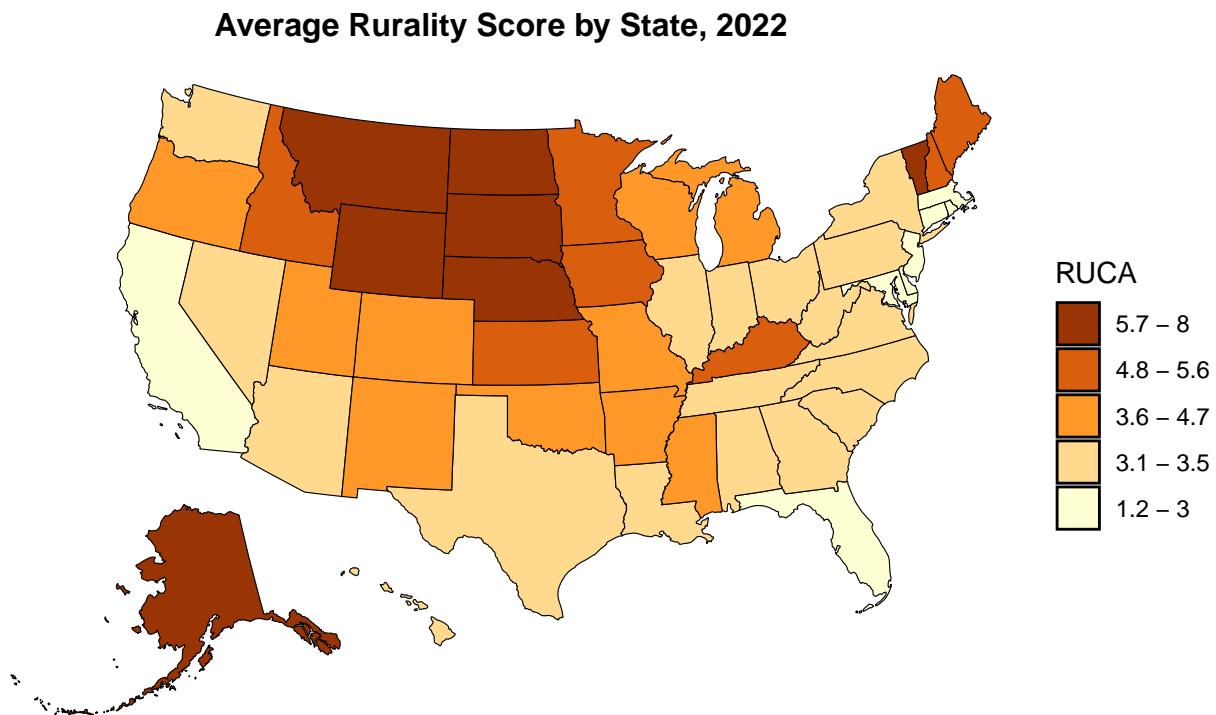
```

Percentage of Adults with Medicare by State, 2022



MAP STATE-LEVEL RUCA

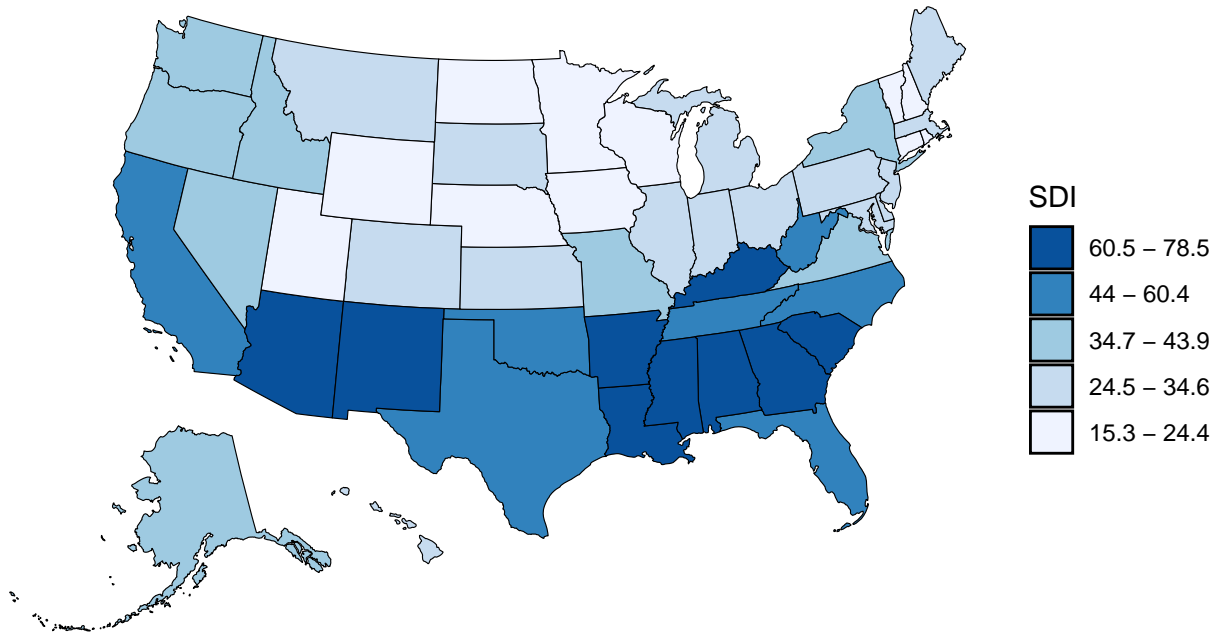
```
state_neuro$Average_RUCA_cat = cut(state_neuro$Average_RUCA, breaks=5,  
                                   labels=ranges(state_neuro,  
                                                 state_neuro$Average_RUCA,  
                                                 as_integer=F))  
  
ggplot(data=state_neuro, aes(fill=Average_RUCA_cat, group=State)) +  
  geom_sf(aes(geometry=geom), color='black') +  
  scale_fill_manual(values=RUCA_colors, name='RUCA',  
                   labels=ranges(state_neuro, state_neuro$Average_RUCA, as_integer=F),  
                   guide=guide_legend(reverse=T)) +  
  theme_void() +  
  ggtitle('Average Rurality Score by State, 2022') +  
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```



MAP STATE-LEVEL SDI

```
state_neuro$Average_SDI_cat = cut(state_neuro$Average_SDI, breaks=5,  
                                  labels=ranges(state_neuro,  
                                                state_neuro$Average_SDI,  
                                                as_integer=F))  
  
ggplot(data=state_neuro, aes(fill=Average_SDI_cat, group=State)) +  
  geom_sf(aes(geometry=geom), color='black') +  
  scale_fill_manual(values=SDI_colors, name='SDI',  
                   labels=ranges(state_neuro, state_neuro$Average_SDI, as_integer=F),  
                   guide=guide_legend(reverse=T)) +  
  theme_void() +  
  ggtitle('Average Social Deprivation Index (SDI) by State, 2022') +  
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

Average Social Deprivation Index (SDI) by State, 2022




```
# PD Mortality Rate to Neurologist Density
state_neuro$mortality_to_density_ratio_cat = cut(state_neuro$mortality_to_density_ratio, breaks=5,
                                                  labels=ranges(state_neuro,
                                                                state_neuro$mortality_to_density_ratio,
                                                                as_integer=F))

ggplot(data=state_neuro, aes(fill=mortality_to_density_ratio_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=ratio_colors, name='Ratio',
                    labels=ranges(state_neuro, state_neuro$mortality_to_density_ratio, as_integer=F),
                    guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Ratio of PD Mortality Rate to Neurologist Density by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

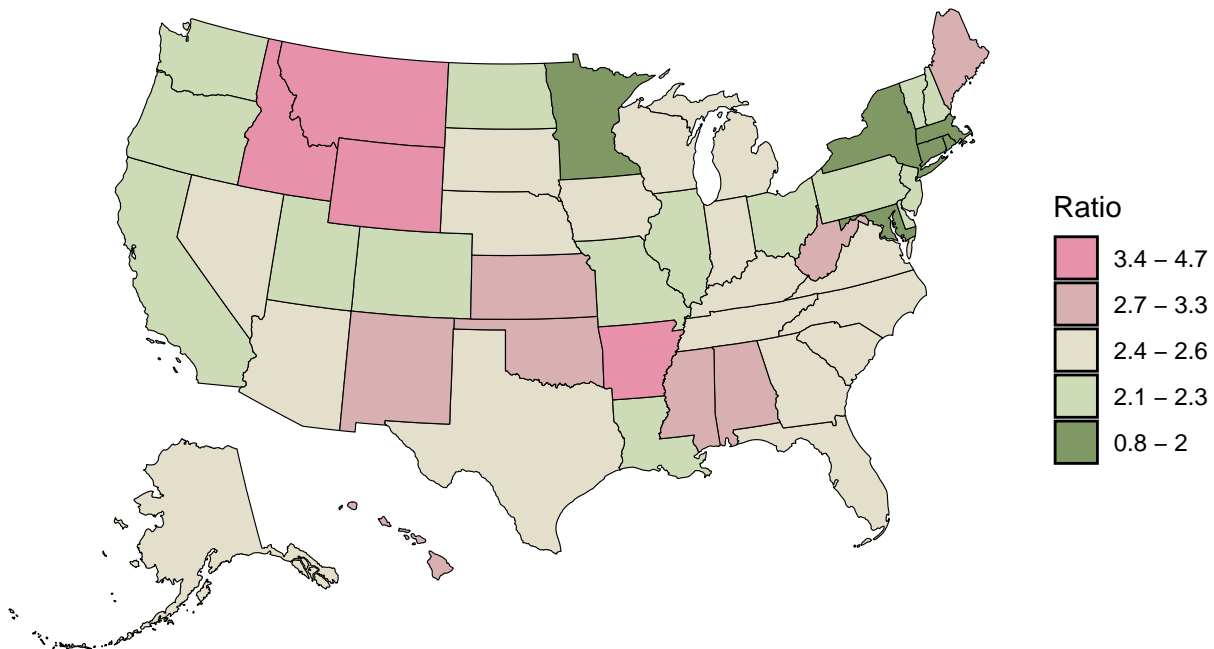
Ratio

- 2.0 – 3.1
- 1.8 – 1.9
- 1.6 – 1.7
- 1.3 – 1.5
- 0.5 – 1.2

```
# PD Deaths to Neurologist Supply
state_neuro$deaths_to_supply_ratio_cat = cut(state_neuro$deaths_to_supply_ratio, breaks=5,
                                             labels=ranges(state_neuro,
                                                             state_neuro$deaths_to_supply_ratio,
                                                             as_integer=F))

ggplot(data=state_neuro, aes(fill=deaths_to_supply_ratio_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=ratio_colors, name='Ratio',
                   labels=ranges(state_neuro, state_neuro$deaths_to_supply_ratio, as_integer=F),
                   guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Ratio of PD Deaths to Neurologist Supply by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

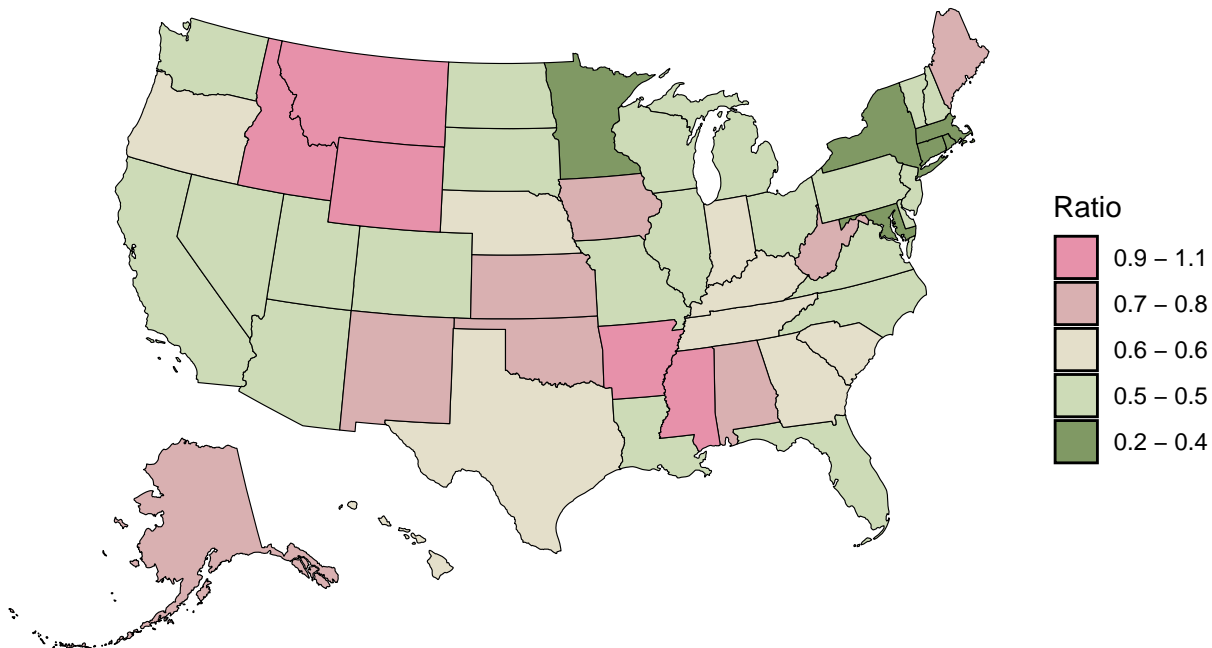
Ratio of PD Deaths to Neurologist Supply by State, 2022



```
# PD Mortality Rate to M-A Neurologist Density
state_neuro$mortality_to_MA_density_ratio_cat = cut(state_neuro$mortality_to_MA_density_ratio, breaks=5,
                                                    labels=ranges(state_neuro,
                                                                    state_neuro$mortality_to_MA_density_ratio,
                                                                    as_integer=F))

ggplot(data=state_neuro, aes(fill=mortality_to_MA_density_ratio_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=ratio_colors, name='Ratio',
                    labels=ranges(state_neuro, state_neuro$mortality_to_MA_density_ratio, as_integer=F),
                    guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Ratio of PD Mortality Rate to Medicare-Accepting Neurologist Density by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 10))
```

Ratio of PD Mortality Rate to Medicare–Accepting Neurologist Density by State, 2022



```
# PD Deaths to M-A Neurologist Supply
state_neuro$deaths_to_MA_supply_ratio_cat = cut(state_neuro$deaths_to_MA_supply_ratio, breaks=5,
                                                labels=ranges(state_neuro,
                                                                state_neuro$deaths_to_MA_supply_ratio,
                                                                as_integer=F))

ggplot(data=state_neuro, aes(fill=deaths_to_MA_supply_ratio_cat, group=State)) +
  geom_sf(aes(geometry=geom), color='black') +
  scale_fill_manual(values=ratio_colors, name='Ratio',
                   labels=ranges(state_neuro, state_neuro$deaths_to_MA_supply_ratio, as_integer=F),
                   guide=guide_legend(reverse=T)) +
  theme_void() +
  ggtitle('Medicare-Accepting Neurologist Supply by State, 2022') +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 12))
```

Medicare-Accepting Neurologist Supply by State, 2022

