Stat/CS 187: Homework 4

yourname

Spring 2022

Contents

```
## Set the default size of figures
knitr::opts_chunk$set(echo = TRUE)

knitr::opts_chunk$set(echo = TRUE,
                      fig.width=8,
                      fig.height=5)

# Load the packages we will be using
pacman::p_load(hexmode,ggthemes)
```

```
## Warning: package 'hexmode' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

## Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contri
##   cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.1/PACKAGES'

## Warning in p_install(package, character.only = TRUE, ...):

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'hexmode'

## Warning in pacman::p_load(hexmode, ggthemes): Failed to install/load:
## hexmode
```

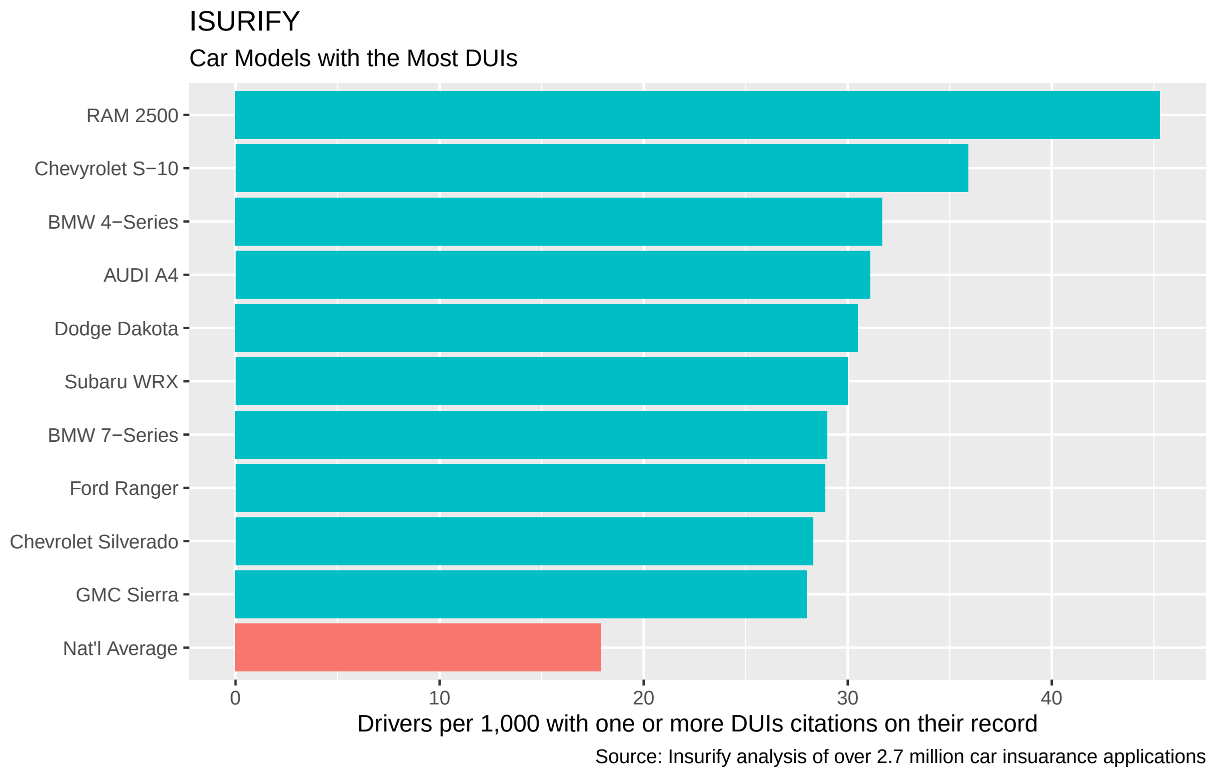
```
library(dplyr)
library(ggplot2)
library(ggthemes)
library(readr)
```

1. Insurify DUI Graph

```
dui <- tribble(
  ~ model,           ~ rate,    ~ specific,
  "RAM 2500",            45.3,       "Yes",
  "Chevyrolet S-10",     35.9,       "Yes",
  "BMW 4-Series",        31.7,       "Yes",
  "AUDI A4",             31.1,       "Yes",
  "Dodge Dakota",        30.5,       "Yes",
  "Subaru WRX",          30.0,       "Yes",
  "BMW 7-Series",        29.0,       "Yes",
  "Ford Ranger",         28.9,       "Yes",
  "Chevrolet Silverado", 28.3,       "Yes",
  "GMC Sierra",          28.0,       "Yes",
  "Nat'l Average",       17.9,        "No"
)
```

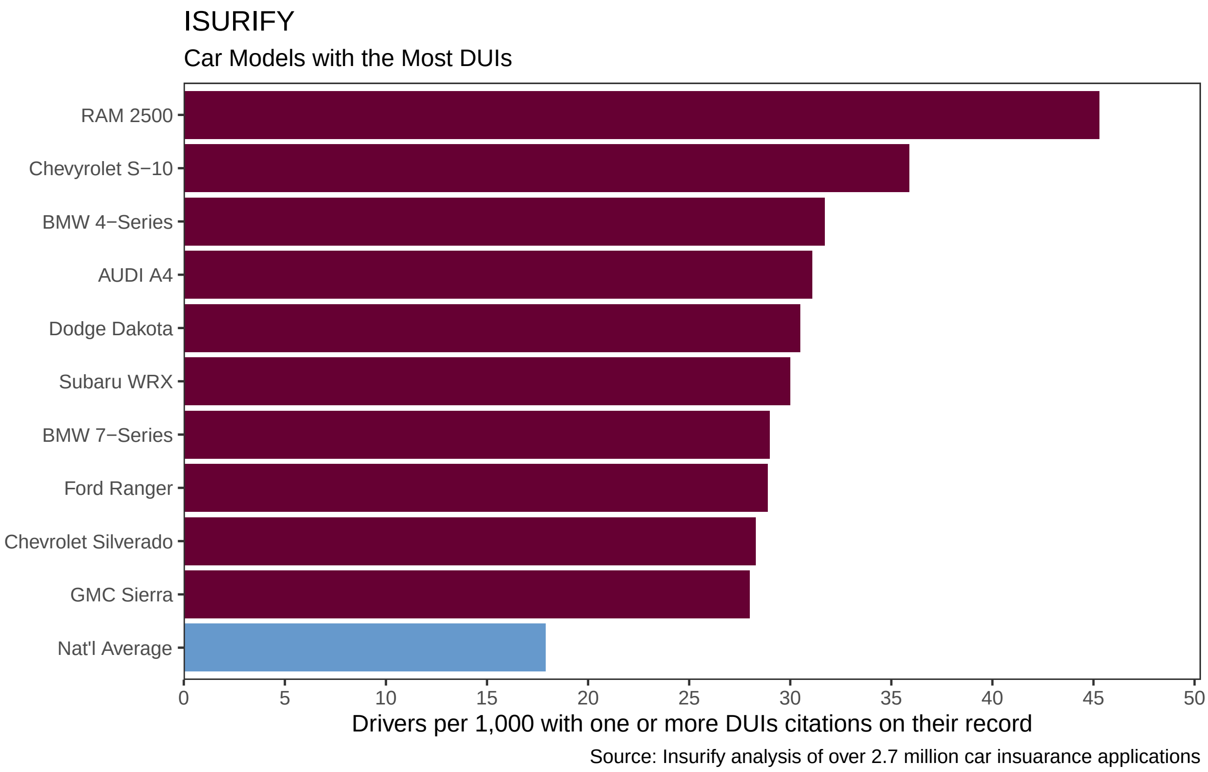
Part 1a)

```
# plot gg_dui as basic plot
gg_dui<-
  ggplot(data=dui,
         mapping=aes(x=rate,
                     y=reorder(model,rate),  # reorder
                     fill=specific))+   # different color for groups
  geom_col(show.legend = FALSE)+     # hide legend for specific
# add title, subtitle,xlab and caption, also remove ylab
  labs(title= "ISURIFY",
       subtitle = "Car Models with the Most DUIs",
       x= "Drivers per 1,000 with one or more DUIs citations on their record",
       y="",
       caption = "Source: Insurify analysis of over 2.7 million car insuarance applications")




# Making sure your knitted document shows the resulting plot
gg_dui
```



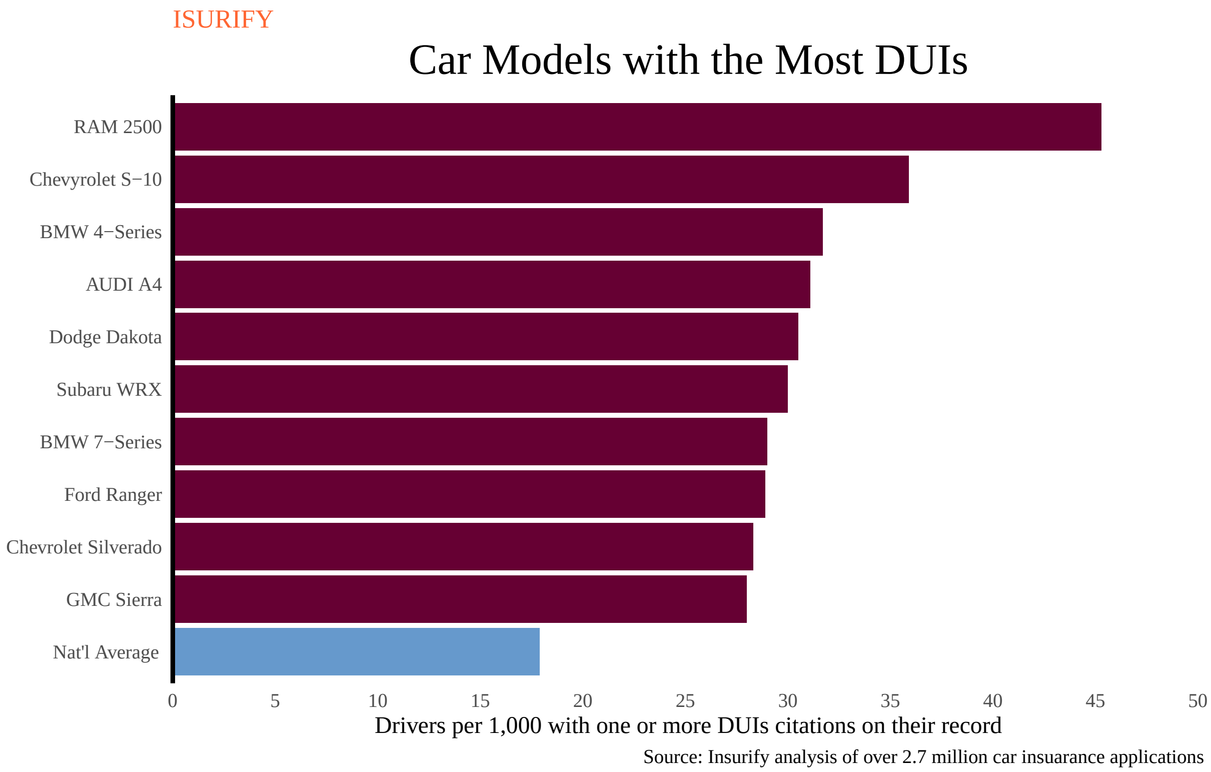
Part 1b)

```
# choose color
color_manual<-c("#6699CC","#660033")

# plot gg_dui2 based on gg_dui
gg_dui2 <-
  gg_dui + theme_test()+     # change theme
# set scales and change breaks for x-axis
  scale_x_continuous(breaks=c(0,5,10,15,20,25,30,35,40,45,50),
                     expand = expansion(mult = 0,
                                        add = c(0, 5)))+
# apply color designed into groups for dui$specific
  scale_fill_manual(values=color_manual)



# show gg_dui2
gg_dui2
```



Part 1c)

```
# change theme for non-data ink options
gg_dui2 + theme_tufte(ticks=FALSE)+
# add vertical line at x=0
  theme(axis.line.y = element_line(size=1,colour="black",linetype=1),
# change color & size of subtitle and size & position of subtitle
        plot.title= element_text(size=12,colour="#FF6633"),
        plot.subtitle = element_text(size=20,hjust=0.5))
```



Car Models with the Most DUIs

Source: Insurify analysis of over 2.7 million car insuarance applications

2. Mapping income and unemployment

```
# Read in data files:
US_2019_C <- read_csv("USDA_2019_County.csv")
```

```
## Rows: 3144 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): stabb, state
## dbl (7): FIPS, Civilian_labor_force_2019, Employed_2019, Unemployed_2019, Un...
##
## 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.
```

```
US_2019_S <- read_csv("USDA_2019_State.csv")
```

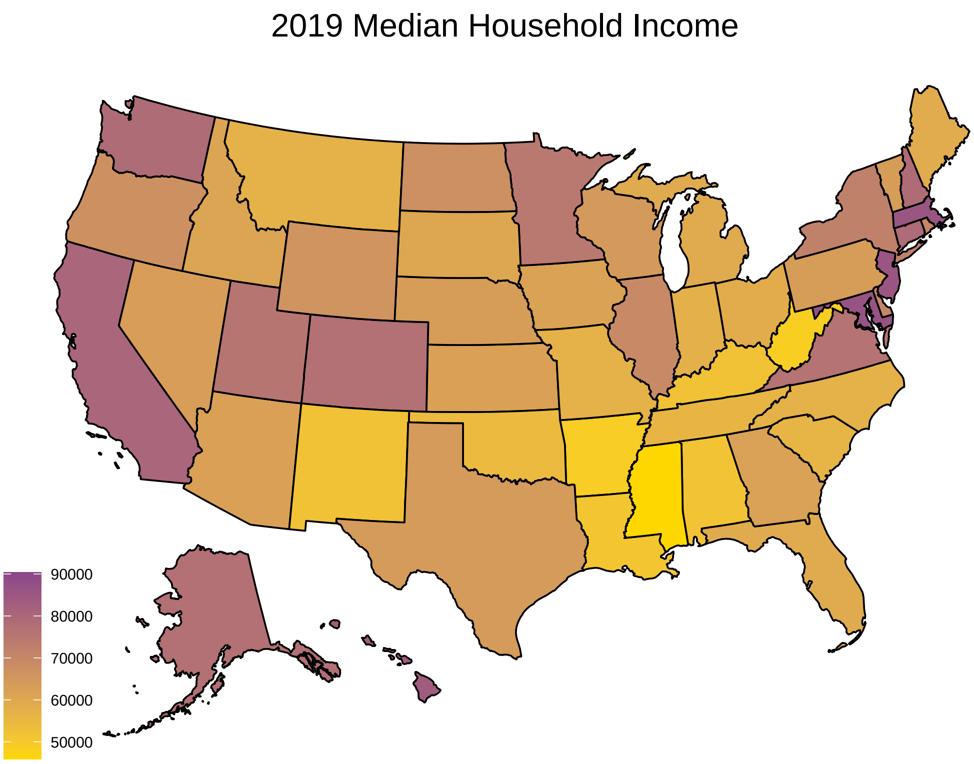
```
## Rows: 51 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): FIPS, StAbb, state
## dbl (5): Civilian_labor_force_2019, Employed_2019, Unemployed_2019, Unemploy...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Set the default size of figures
knitr::opts_chunk$set(fig.width=8, fig.height=5)

# Changing the default theme
theme_set(theme_bw())
```

```r
# Load the libraries we will be using
pacman::p_load(gapminder,socviz, tidyverse, grid,usmap, maps, statebins, viridis, leaflet)


# A map theme provided by our book:
theme_map <- function(base_size=9, base_family="") {
    require(grid)
    theme_bw(base_size=base_size, base_family=base_family) %+replace%
        theme(axis.line=element_blank(),
              axis.text=element_blank(),
              axis.ticks=element_blank(),
              axis.title=element_blank(),
              panel.background=element_blank(),
              panel.border=element_blank(),
              panel.grid=element_blank(),
              panel.spacing=unit(0, "lines"),
              plot.background=element_blank(),
              legend.justification = c(0,0),
              legend.position = c(0,0)
              )
}
```

2a) Simple US Map

```r
#  Use package usmap, and employment data
plot_usmap(regions = c('state'),
           data = data.frame(US_2019_S),
           values= "Median_Household_Income_2019") +
# choose color reasonably close
  scale_fill_continuous(low = 'gold1', high = 'orchid4')+
# add title at center and remove legend title
  labs(title="2019 Median Household Income") +
  theme(plot.title = element_text(size=15,hjust=0.5),
        legend.title = element_blank())
```



2b) For more serious maps: Merge state lines info with USDA data.

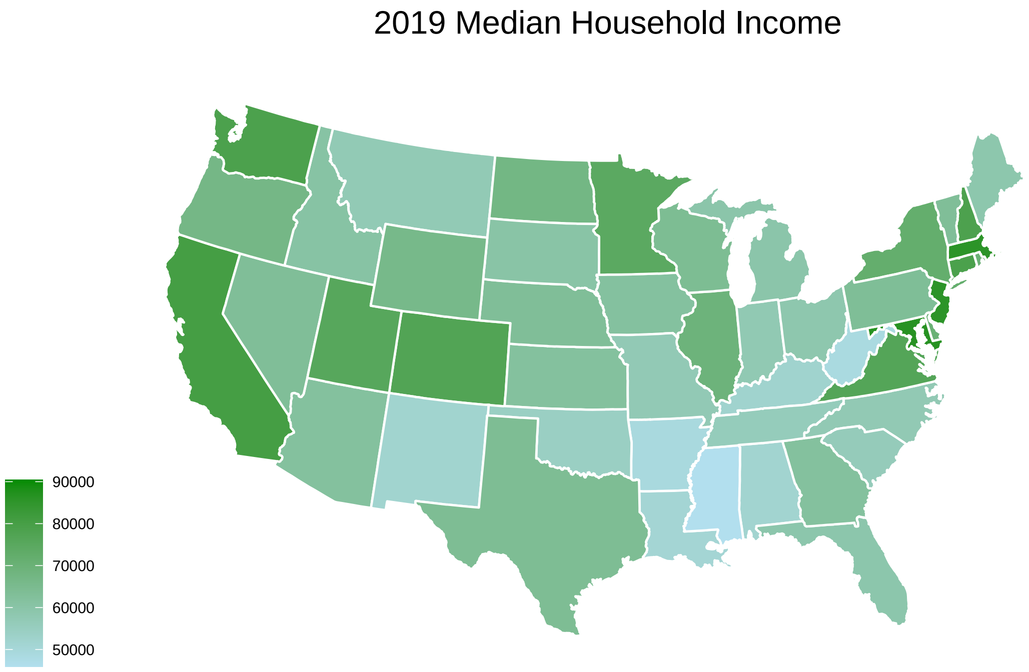
```
# create state_lines using the map_data function like we did in class
state_lines <- map_data("state")
head(state_lines)
```

```
##        long      lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama      <NA>
## 2 -87.48493 30.37249     1     2 alabama      <NA>
## 3 -87.52503 30.37249     1     3 alabama      <NA>
## 4 -87.53076 30.33239     1     4 alabama      <NA>
## 5 -87.57087 30.32665     1     5 alabama      <NA>
## 6 -87.58806 30.32665     1     6 alabama      <NA>
```

```
# clean data and change region to state for merge
state_lines<- state_lines%>%
  mutate(state=region)%>%
  select(-region,-subregion)

# change state to all lowercase
US_2019_S$state<-tolower(US_2019_S$state)

# merge data
US <- US_2019_S %>%
  left_join(y = state_lines,
            by = 'state')
```

2c) Map a Numeric variable by State: Median Income in 2019

```
ggplot(data = US,
         mapping = aes(x = long,
                       y = lat,
                       group = group))  +

  geom_polygon(aes(fill = US$Median_Household_Income_2019),color="white")  +

  theme_map() +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45)+

# choose color reasonably close
  scale_fill_gradient(low = 'lightblue2', high = 'green4' )+

# add title at center and remove legend title
  labs(title="2019 Median Household Income") +

  theme(plot.title = element_text(size=15,hjust=0.5),
        legend.title = element_blank())
```



2d) Now map data at a county level. First, the data, merged with county map info

```
# county_map is a data frame in socviz -- do help(county_map) to see more
?county_map
```

```
## starting httpd help server ... done
```

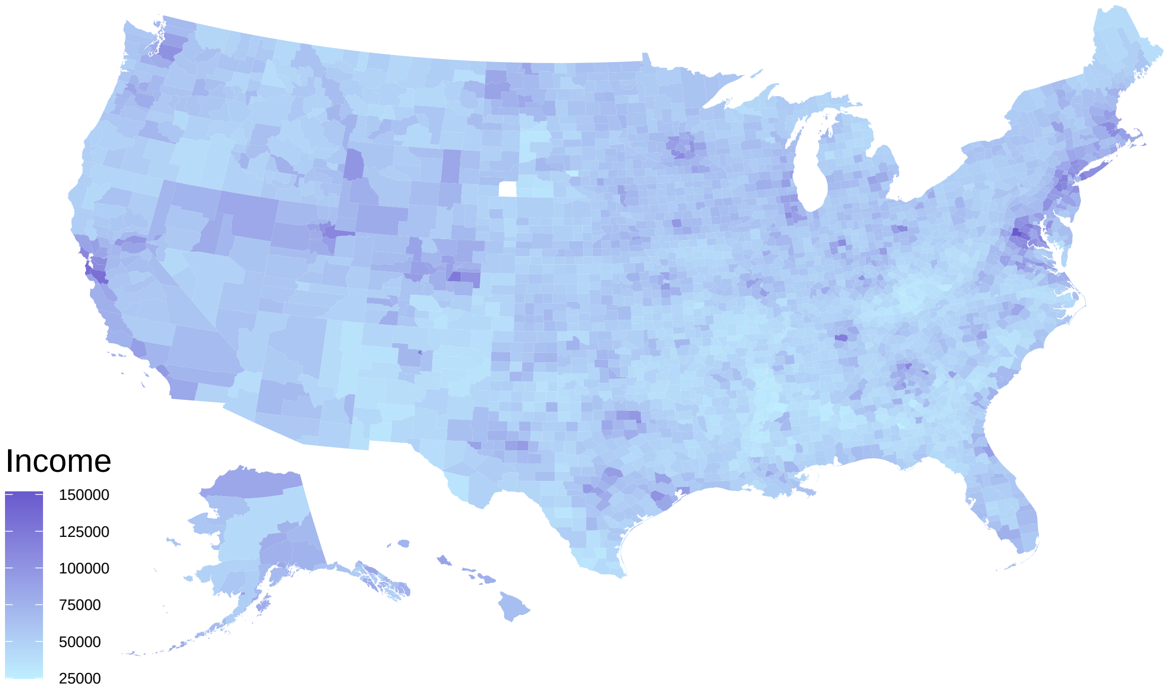
```
#create a new data frame using county_map data
new_cmap<-county_map%>%
  select_all()%>%
# create a variable called FIPS
  mutate(FIPS = id)%>%
# drop id
  select(-id)
# change FIPS to numeric values
new_cmap$FIPS <- as.numeric(new_cmap$FIPS)


# merge the two together
US2 <- US_2019_C %>%
  left_join(y = new_cmap,
            by = 'FIPS')

# Print the first 6 rows of the resulting data set
head(US2)
```

```
## # A tibble: 6 x 15
##    FIPS stabb state              Civilian_labor_f~ Employed_2019 Unemployed_2019
##   <dbl> <chr> <chr>                          <dbl>         <dbl>           <dbl>
## 1  1001 AL    Autauga County, AL             26172         25458             714
## 2  1001 AL    Autauga County, AL             26172         25458             714
## 3  1001 AL    Autauga County, AL             26172         25458             714
## 4  1001 AL    Autauga County, AL             26172         25458             714
## 5  1001 AL    Autauga County, AL             26172         25458             714
## 6  1001 AL    Autauga County, AL             26172         25458             714
## # ... with 9 more variables: Unemployment_rate_2019 <dbl>,
## #   Median_Household_Income_2019 <dbl>,
## #   Med_HH_Income_Percent_of_State_Total_2019 <dbl>, long <dbl>, lat <dbl>,
## #   order <int>, hole <lgl>, piece <fct>, group <fct>
```

2e) Map Median Household Income by county:

Map income by county, matching the map below, in terms of colors (pretty close is fine), title, legend label, font size, theme, etc.

```
US2$Income<-US2$Median_Household_Income_2019
ggplot(data = US2,
         mapping = aes(x = long, y = lat,
                       fill=Income,
                       group = group
                       ))+

  geom_polygon(size=0.05)+
  theme_map()+
# choose color reasonably close
#  scale_fill_continuous(low="white",high="blue")+
  scale_fill_gradient(low="lightblue1",high="slateblue3")+

# add title at center and remove legend title
  labs(title="2019 Median Income") +
  theme(plot.title = element_text(size=15,hjust=0.5),
        legend.title=element_text(size=15))
```



2019 Median Income