Mapping State Cities

```
# load packages for data wrangling
library(tidyverse)
library(ggthemes)
library(usdata)
library(scales)
```

In this mini-assignment, you will be shown an example code that creates a specific map. After you examine the example code, please do the exercises in each part.

Mapping State Counties and Cities. Here, we are using the shapefiles provided by the `urbnmapr` package, which is from the Urban Institute. For the R code block below, examine the R code and solve the following:

a. Choose a US state to map other than the example.
b. Use the `county` data set from the `usdata` package where you use the `pop_change` variable and convert it into an ordinal categorical variable with 6 levels.
c. Combine the shapefile dataframe and part (b)s dataframe where you match the county and the state names.
d. Plot the county-level state map where you color each county proportional to the ordinal categorical variable done in part (b). Choose a color diverging scheme from ColorBrewer 2.0.
e. Use the `us.cites` data set from the `maps` package to put point onto your map.

Make sure to adjust your figure widths and heights to avoid distortions or overlapping labels. No need to describe your observations.

```
library(sf) # classes and functions for vector data (or shapefiles)
library(maps) # other maps with cities data

# Urban Institute shape files: https://urbaninstitute.github.io/urbnmapr/
# install.packages("devtools")
# devtools::install_github("UrbanInstitute/urbnmapr")
library(urbnmapr) # provides state and county shapefiles
```

```
#### Start: (1) Choose a US State here
state_full <- "California"
state_abv <- "CA"
#### End: (1) Choose a US State here

# Load the shapefile object of US counties and filter only the state of choice
#   using the urbnmapr package
counties_sf <- get_urbn_map("counties", sf = TRUE)
counties_sf_ca <- counties_sf %>% filter(state_name == state_full)
```

```
#### Start: (2) Wrangle the county data set here
# subset of the county data set
county_sub <- county %>%
  select(name,state,pop_change)
# cities data set
crs_use <- "+proj=laea +lat_0=30 +lon_0=-95" # Albers projection with curvature
us_cities <- us.cities %>%
  mutate(name_n = str_remove_all(name, state_abv)) %>%
  mutate(name_n = str_remove_all(name_n, " ")) %>%
  filter(country.etc == state_abv)
d_points <-  us_cities %>%
  st_as_sf(coords = c("long", "lat"), crs = 4326) %>%
  st_transform(crs = crs_use)
#### End: (2) Wrangle the county data set here

#### Start: (3) Update the ggplot pipeline here
ggplot(data = counties_sf_ca, aes()) +
  geom_sf() +
  labs(title = paste("Population Change County-Level Map of",state_full)) +
#### End: (3) Update the ggplot pipeline here
theme(legend.position="bottom",
        panel.background=element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank())
```

Population Change County−Level Map of California