Exercise 03 Ray Curtis EC 137

Curtis

4/2/2021

What is BLS Scraper?

```
# not for the class, but read int the api key
set_bls_key("f8cf85a384824d14af5a42667fad7f77", overwrite = TRUE)
```

```
## [1] "f8cf85a384824d14af5a42667fad7f77"
```

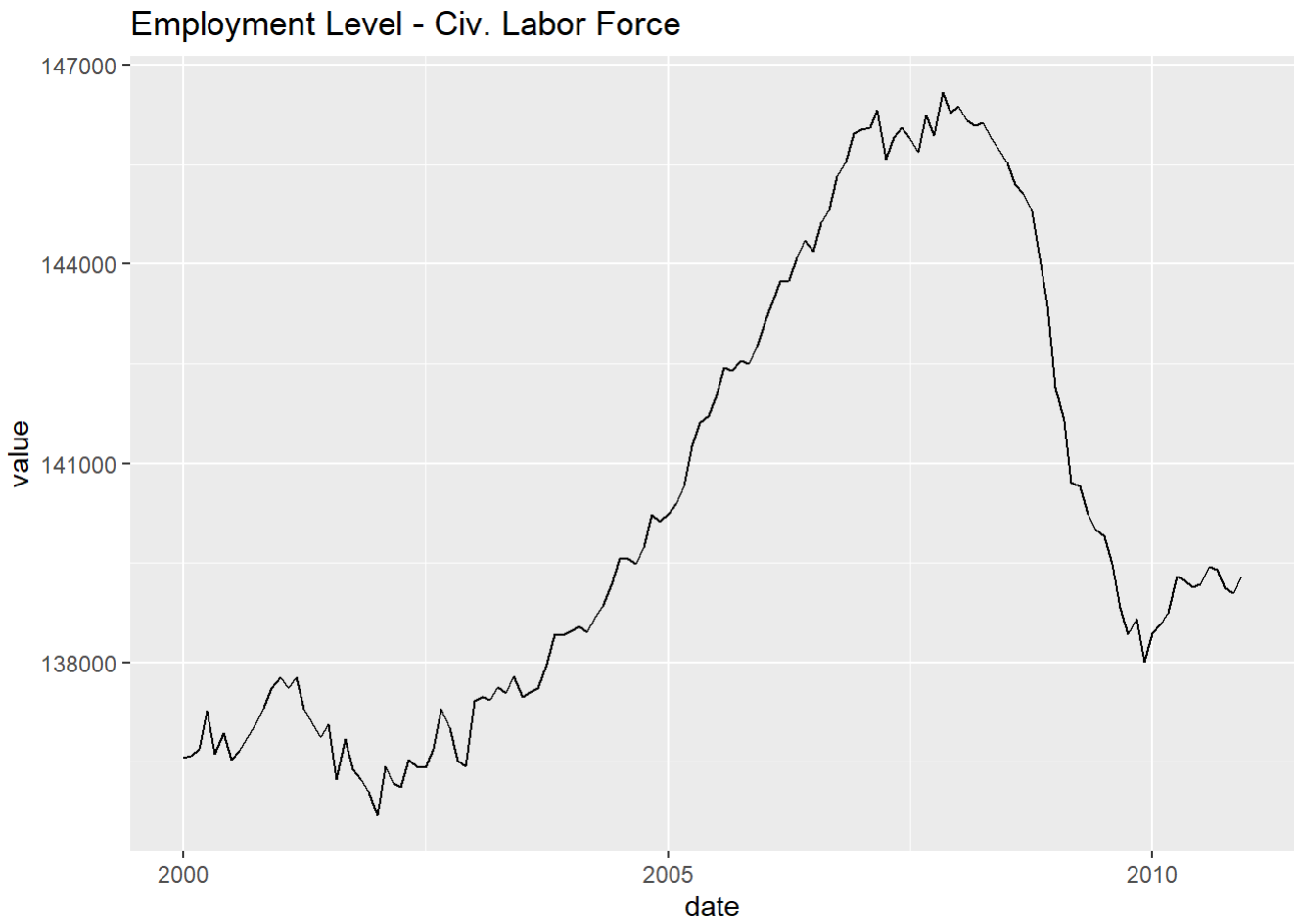
```
readRenviron("~/.Renviron")

Sys.getenv("BLS_KEY")
```

```
## [1] "f8cf85a384824d14af5a42667fad7f77"
```

```
library(blscrapeR)

# Grab several data sets from the BLS at onece.
# NOTE on series IDs:
# EMPLOYMENT LEVEL - Civilian labor force - LNS12000000
# UNEMPLOYMENT LEVEL - Civilian labor force - LNS13000000
# UNEMPLOYMENT RATE - Civilian labor force - LNS14000000
df <- bls_api(c("LNS12000000", "LNS13000000", "LNS14000000"),
              startyear = 2000, endyear = 2010, Sys.getenv("BLS_KEY")) %>%
    # Add time-series dates
    dateCast()
```

```
## REQUEST_SUCCEEDED
```

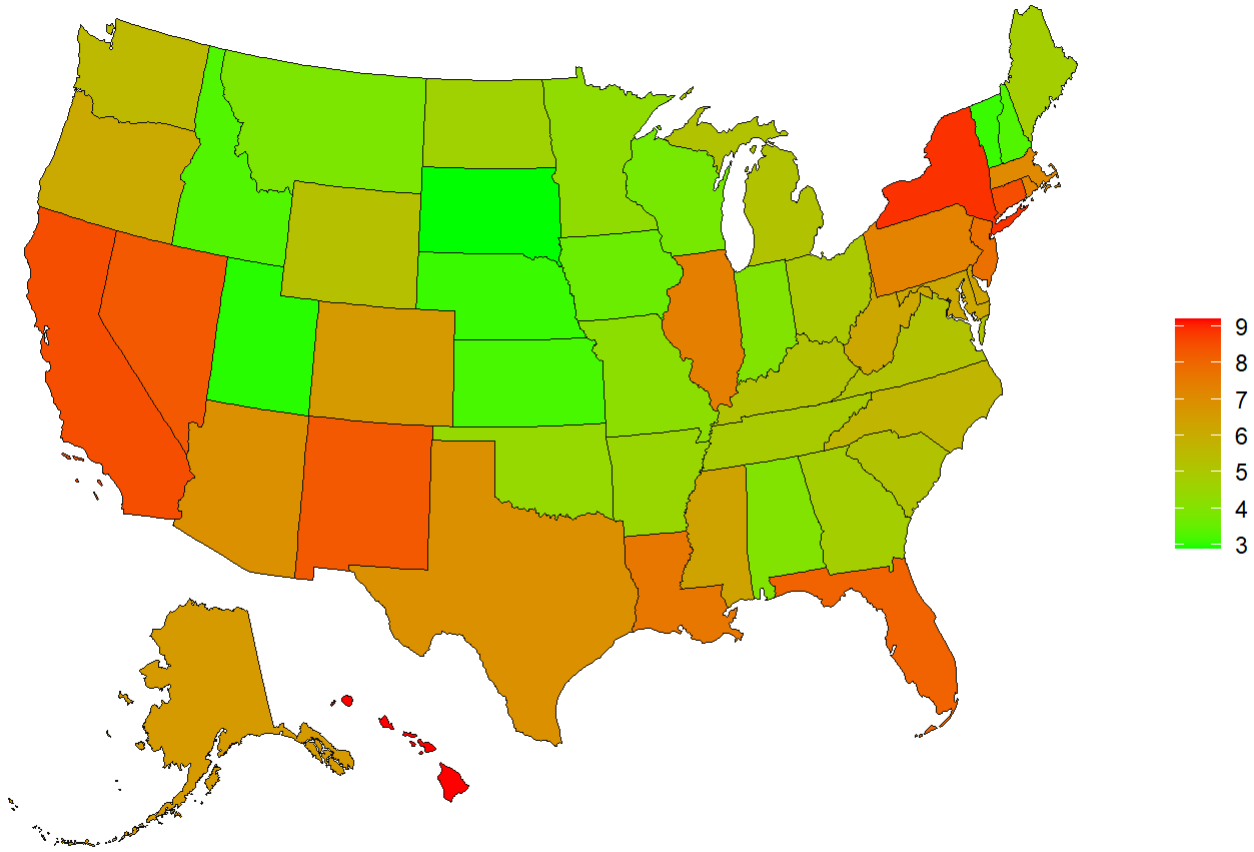
```
# Plot employment level
library(ggplot2)
gg1200 <- subset(df, seriesID=="LNS12000000")
library(ggplot2)
ggplot(gg1200, aes(x=date, y=value)) +
    geom_line() +
    labs(title = "Employment Level - Civ. Labor Force")
```




```
df2 <- get_bls_state()

tibble(df2)
```

```
## # A tibble: 50 x 12
##    state civ_pop labor_force labor_force_rate employed employed_rate unemployed
##    <chr>   <dbl>       <dbl>            <dbl>    <dbl>         <dbl>      <dbl>
##  1 Alab~  3.89e6     2249475             57.8  2158410          55.5      91065
##  2 Alas~  5.45e5      351042             64.4   327734          60.2      23308
##  3 Ariz~  5.94e6     3580159             60.2  3333218          56.1     246941
##  4 Arka~  2.37e6     1363154             57.6  1301585          55        61569
##  5 Cali~  3.11e7    18944536             60.9 17334333          55.8    1610203
##  6 Colo~  4.65e6     3187192             68.6  2977934          64.1     209258
##  7 Conn~  2.88e6     1712892             59.5  1566755          54.4     146137
##  8 Dela~  7.97e5      488849             61.3   457878          57.5      30971
##  9 Flor~  5.84e5      408453             69.9   375176          64.2      33277
## 10 Geor~  8.35e6     5145311             61.6  4899537          58.7     245774
## # ... with 40 more rows, and 5 more variables: unemployed_rate <dbl>,
## #   month <date>, fips_state <chr>, state_abb <chr>, gnisid <chr>
```

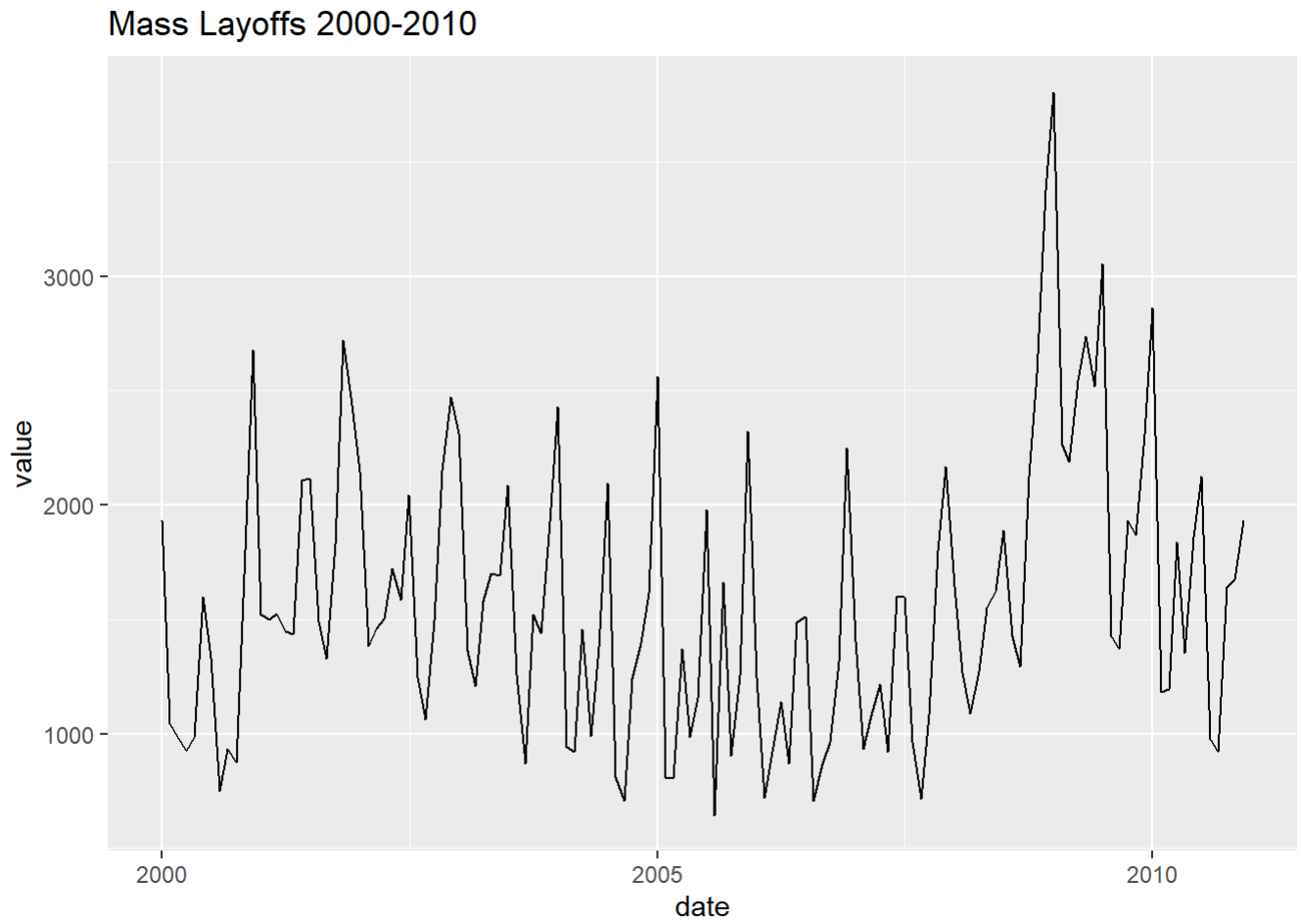
```
# use the map_bls function
map_bls(map_data = df2, fill_rate = "unemployed_rate")
```




```
# to pull a new variable from bls, all we have to do is go to the link below, and find the Serie
s ID code to pull it into R

# Link:https://www.bls.gov/help/hlpforma.htm#ML


df3 <- bls_api(c("MLUMS00NN0001003"),
               startyear = 2000, endyear = 2010, Sys.getenv("BLS_KEY")) %>%
    # Add time-series dates
    dateCast()
```

```
## REQUEST_SUCCEEDED
```

```
# lets plot the mass layoff statistics from 2000-2010


ggplot(df3, aes(x=date, y=value)) +
    geom_line() +
    labs(title = "Mass Layoffs 2000-2010")
```



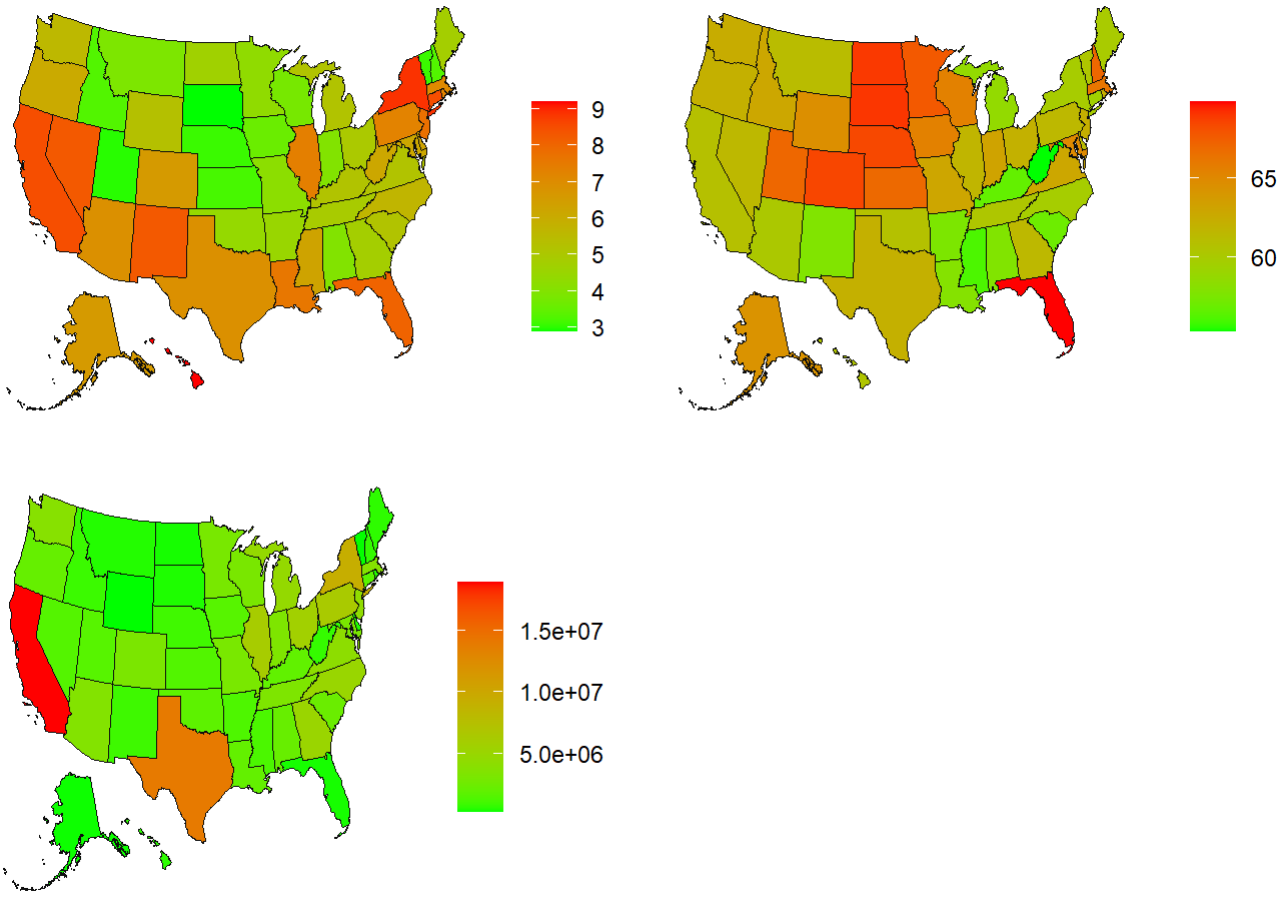

```
df2 <- get_bls_state()

tibble(df2)
```

```
## # A tibble: 50 x 12
##    state civ_pop labor_force labor_force_rate employed employed_rate unemployed
##    <chr>   <dbl>       <dbl>            <dbl>    <dbl>         <dbl>      <dbl>
##  1 Alab~  3.89e6     2249475             57.8  2158410          55.5      91065
##  2 Alas~  5.45e5      351042             64.4   327734          60.2      23308
##  3 Ariz~  5.94e6     3580159             60.2  3333218          56.1     246941
##  4 Arka~  2.37e6     1363154             57.6  1301585          55        61569
##  5 Cali~  3.11e7    18944536             60.9 17334333          55.8    1610203
##  6 Colo~  4.65e6     3187192             68.6  2977934          64.1     209258
##  7 Conn~  2.88e6     1712892             59.5  1566755          54.4     146137
##  8 Dela~  7.97e5      488849             61.3   457878          57.5      30971
##  9 Flor~  5.84e5      408453             69.9   375176          64.2      33277
## 10 Geor~  8.35e6     5145311             61.6  4899537          58.7     245774
## # ... with 40 more rows, and 5 more variables: unemployed_rate <dbl>,
## #   month <date>, fips_state <chr>, state_abb <chr>, gnisid <chr>
```

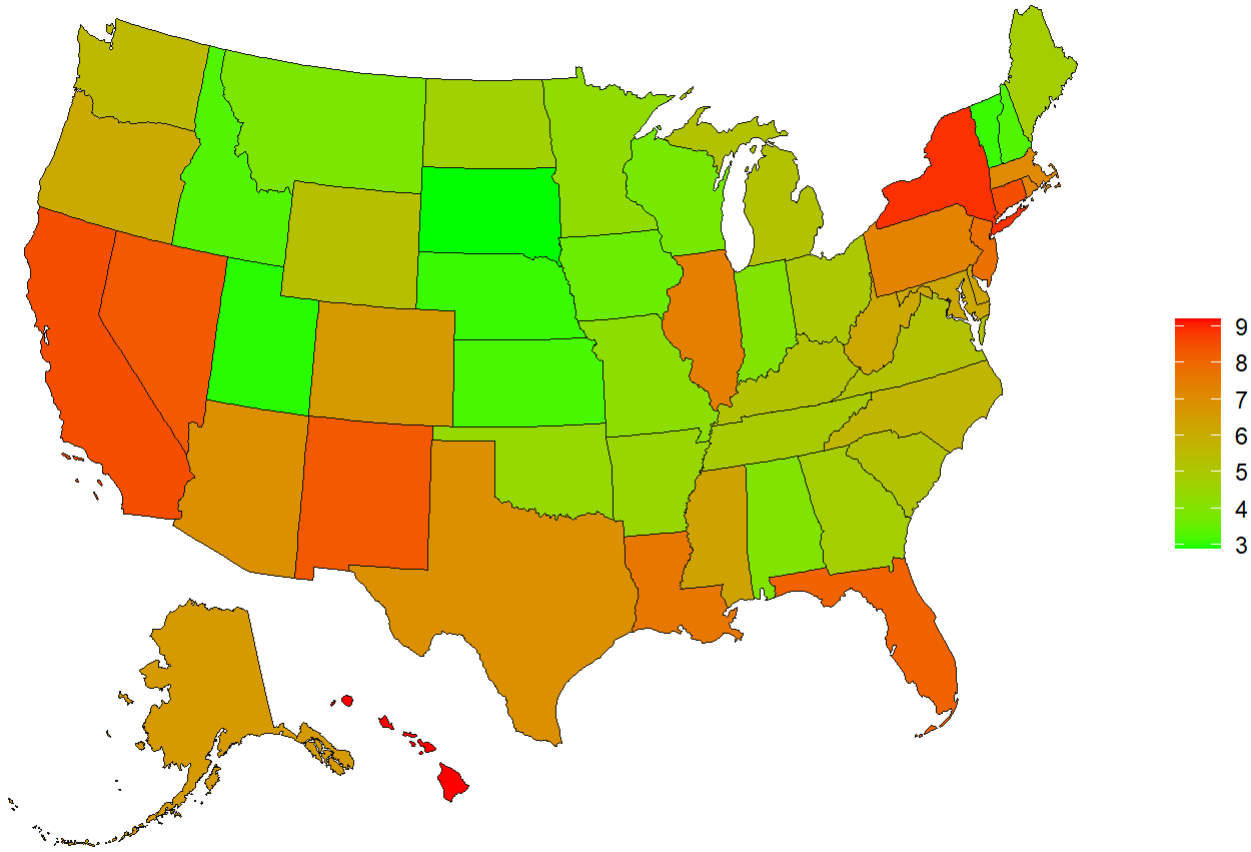
```
# use the map_bls function
graph1 <- map_bls(map_data = df2, fill_rate = "unemployed_rate")


# we can plot other things as well on this
# We should tibble this get_bls_state to see what other variables we can use to plot
tibble(df2)
```

```
## # A tibble: 50 x 12
##    state civ_pop labor_force labor_force_rate employed employed_rate unemployed
##    <chr>   <dbl>       <dbl>            <dbl>    <dbl>         <dbl>      <dbl>
##  1 Alab~  3.89e6     2249475             57.8  2158410          55.5      91065
##  2 Alas~  5.45e5      351042             64.4   327734          60.2      23308
##  3 Ariz~  5.94e6     3580159             60.2  3333218          56.1     246941
##  4 Arka~  2.37e6     1363154             57.6  1301585          55        61569
##  5 Cali~  3.11e7    18944536             60.9 17334333          55.8    1610203
##  6 Colo~  4.65e6     3187192             68.6  2977934          64.1     209258
##  7 Conn~  2.88e6     1712892             59.5  1566755          54.4     146137
##  8 Dela~  7.97e5      488849             61.3   457878          57.5      30971
##  9 Flor~  5.84e5      408453             69.9   375176          64.2      33277
## 10 Geor~  8.35e6     5145311             61.6  4899537          58.7     245774
## # ... with 40 more rows, and 5 more variables: unemployed_rate <dbl>,
## #   month <date>, fips_state <chr>, state_abb <chr>, gnisid <chr>
```

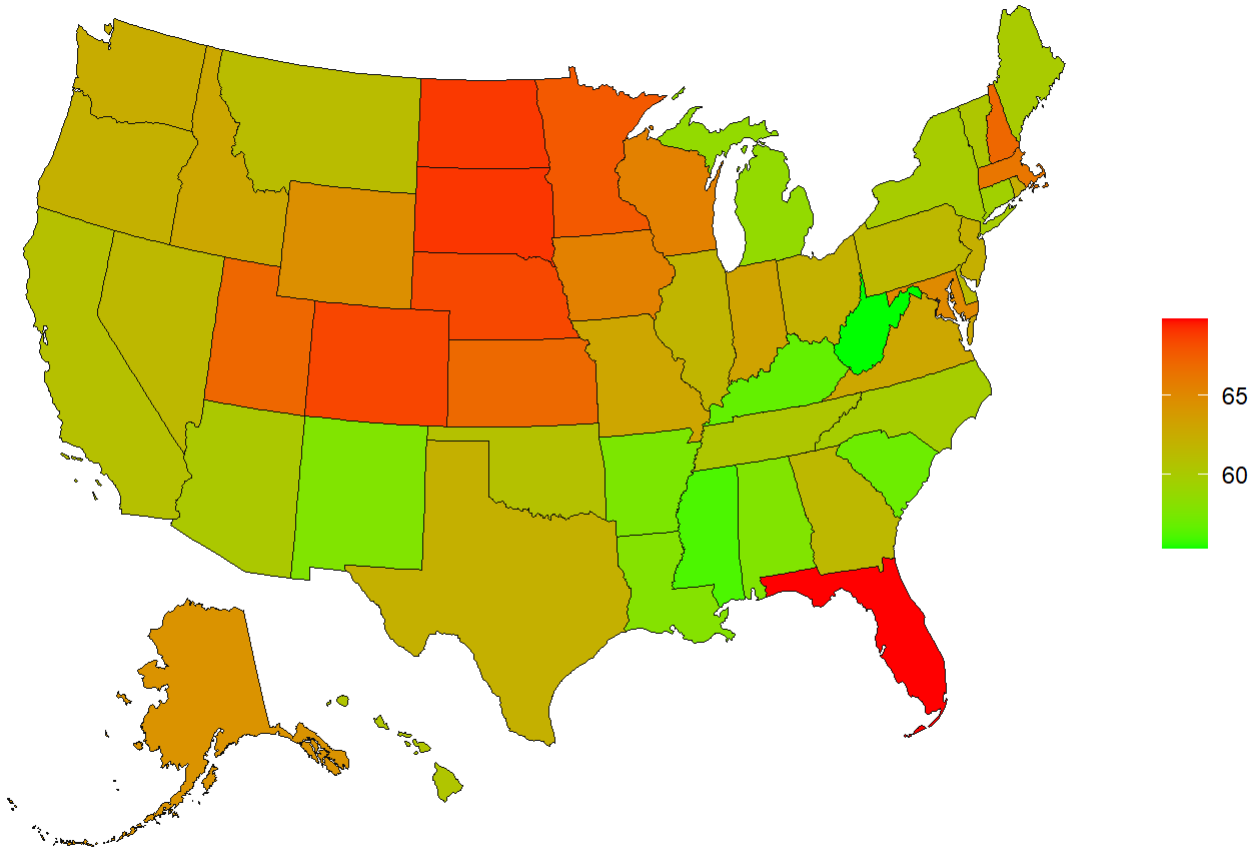
```
# lets choose to do labor force and labor force rate instead

# all we're changing for ggplot is the variable of the bls_state data frame we want to plot
# i.e. labor_force_rate or labor_force

graph2 <- map_bls(map_data = df2, fill_rate = "labor_force_rate")

graph3 <- map_bls(map_data = df2, fill_rate = "labor_force")


ggarrange(graph1, graph2, graph3)
```



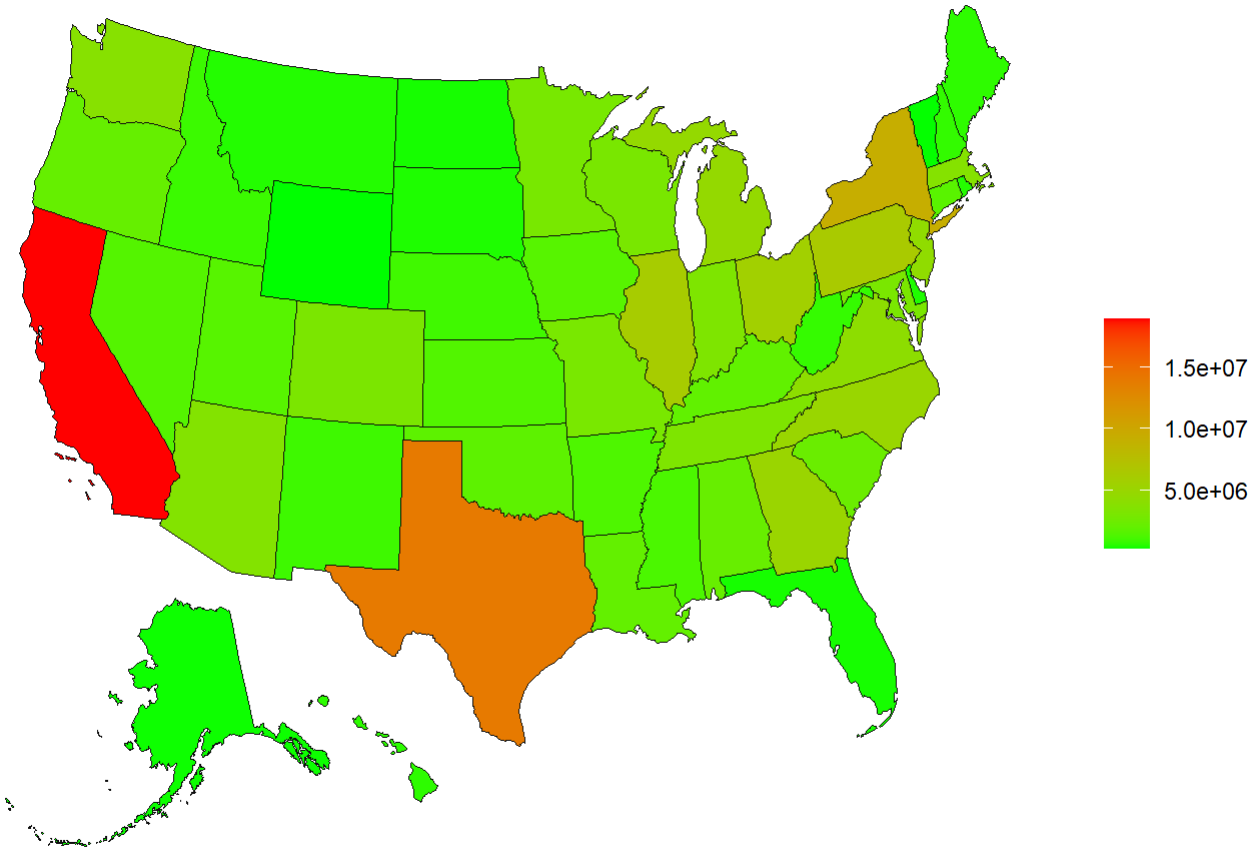

```
# or
graph1
```




```
graph2
```



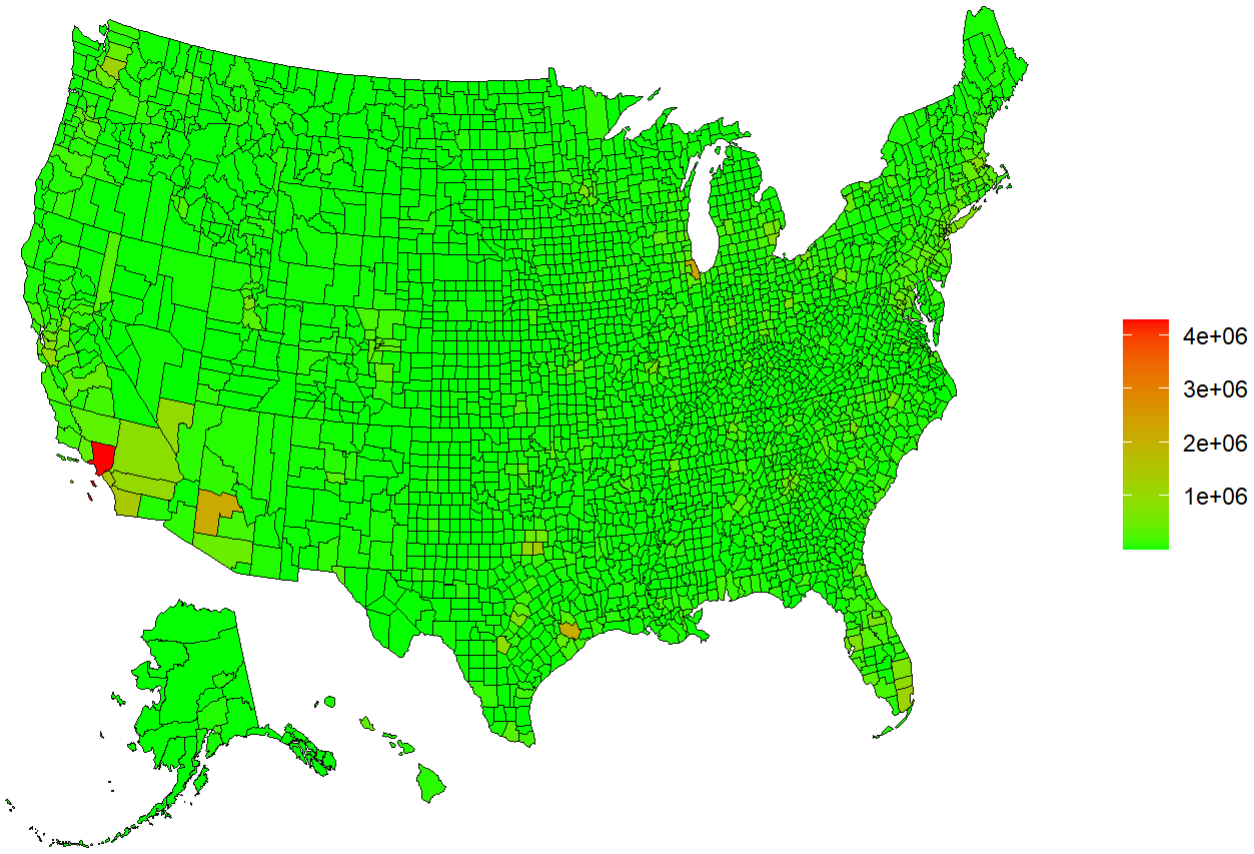

```
graph3
```




```
# In the bls scraper, it also contains county level data, so you can see the distribution of the
se varibles at a state level

# name a new data frame
df4 <- get_bls_county() # same as before, just count

# tibble to get a quick view
tibble(df4)
```

```
## # A tibble: 3,219 x 10
##    area_code fips_state fips_county area_title period     labor_force employed
##    <chr>     <chr>      <chr>       <chr>      <date>           <dbl>    <dbl>
##  1 CN010010~ 01         001         Autauga C~ 2021-01-01       25757    24904
##  2 CN010030~ 01         003         Baldwin C~ 2021-01-01       95885    92512
##  3 CN010050~ 01         005         Barbour C~ 2021-01-01        8391     7826
##  4 CN010070~ 01         007         Bibb Coun~ 2021-01-01        8622     8270
##  5 CN010090~ 01         009         Blount Co~ 2021-01-01       24792    24190
##  6 CN010110~ 01         011         Bullock C~ 2021-01-01        4781     4554
##  7 CN010130~ 01         013         Butler Co~ 2021-01-01        8821     8232
##  8 CN010150~ 01         015         Calhoun C~ 2021-01-01       46374    44284
##  9 CN010170~ 01         017         Chambers ~ 2021-01-01       15904    15163
## 10 CN010190~ 01         019         Cherokee ~ 2021-01-01       11453    11136
## # ... with 3,209 more rows, and 3 more variables: unemployed <dbl>,
## #   unemployed_rate <chr>, fips <chr>
```

```
# Now let's make a map in the same fashion for those employed in each county of each state
graph4 <- map_bls(map_data = df4, fill_rate = "employed")

graph4
```

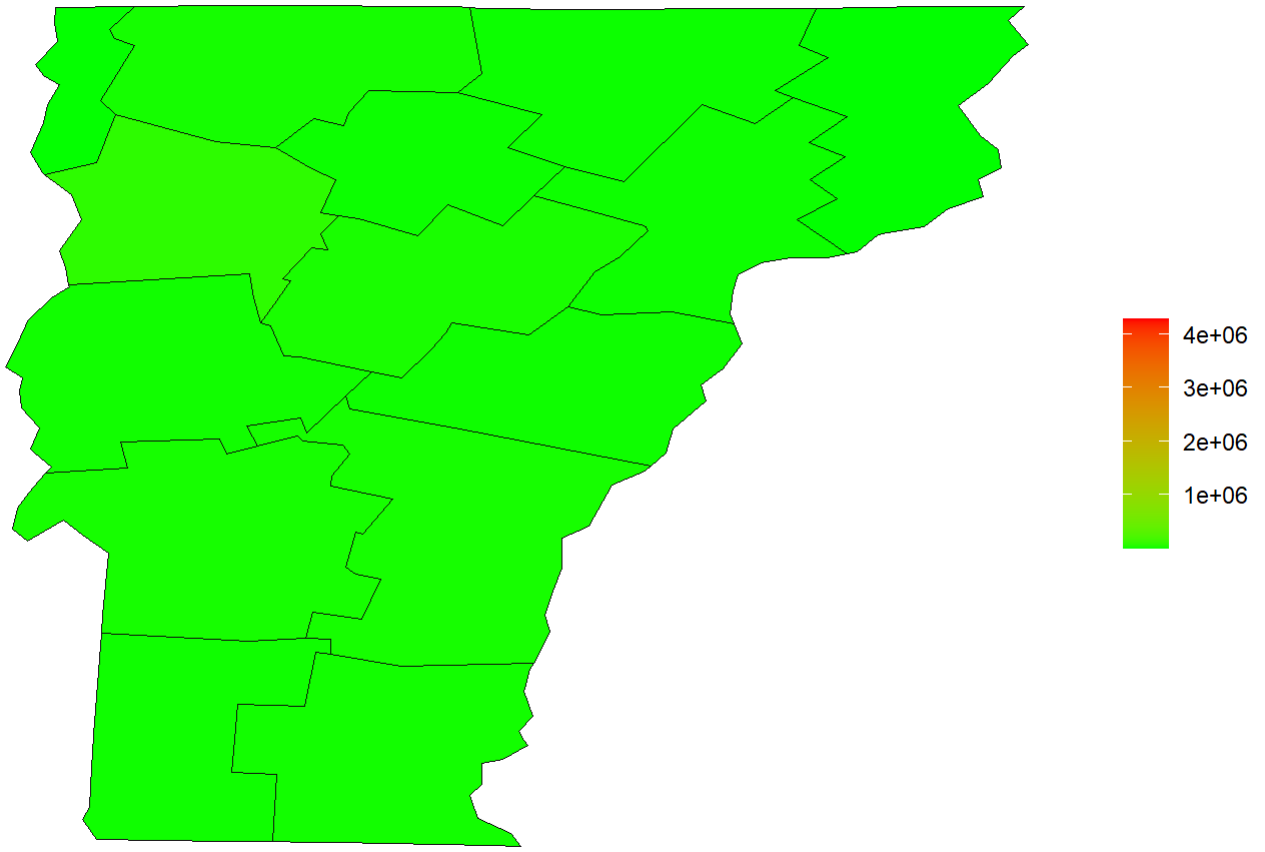



```
# although this is hard to read, it is interesting to see where certain hot spots are

# we can also analyze certain states if we wanted to
graph5 <- bls_map_county(map_data = df4, fill_rate = "employed", stateName = "Vermont")
```

```
## Warning in bls_map_county(map_data = df4, fill_rate = "employed", stateName =
## "Vermont"): This function has been deprecated, consider using map_bls() instead.
```

```
## Warning: Ignoring unknown aesthetics: x, y
```

```
# this graph will show only Vermont's employment levels
graph5
```




```
# bls scraper also offers some quick ways to pull employment statistics

df5 <-quick_employed_level()
```

```
## REQUEST_SUCCEEDED
```

```
tibble(df5)
```

```
## # A tibble: 27 x 7
##     year period periodName latest  value footnotes                    seriesID
##    <dbl> <chr>  <chr>      <chr>   <dbl> <chr>                        <chr>
##  1  2021 M03    March      true   150848 ""                           LNS12000~
##  2  2021 M02    February   <NA>   150239 ""                           LNS12000~
##  3  2021 M01    January    <NA>   150031 "1 Data affected by changes ~ LNS12000~
##  4  2020 M12    December   <NA>   149830 ""                           LNS12000~
##  5  2020 M11    November   <NA>   149809 ""                           LNS12000~
##  6  2020 M10    October    <NA>   149669 ""                           LNS12000~
##  7  2020 M09    September  <NA>   147543 ""                           LNS12000~
##  8  2020 M08    August     <NA>   147276 ""                           LNS12000~
##  9  2020 M07    July       <NA>   143777 ""                           LNS12000~
## 10  2020 M06    June       <NA>   142100 ""                           LNS12000~
## # ... with 17 more rows
```

```
# this works with several other economic variables that can be  found here:
# https://cran.r-project.org/web/packages/blscrapeR/blscrapeR.pdf
```