R Notebook

Resources

Here are some links I found helpful:

- https://www.computerworld.com/article/3175623/mapping-in-r-just-got-a-whole-lot-easier.html
- https://cran.r-project.org/web/packages/tmap/vignettes/tmap-getstarted.html
- http://zevross.com/blog/2018/10/02/creating-beautiful-demographic-maps-in-r-with-the-tidycensus-and-tmap-packages/
- https://geocompr.robinlovelace.net/adv-map.html#introduction-5

Imports

These are the libraries I'm using (I silence masking warnings):

Show the current software versions I'm working with.

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] reshape2_1.4.3     RColorBrewer_1.1-2 spDataLarge_0.3.1  spData_0.3.2
##  [5] tmaptools_2.0-2    tmap_2.3-1         raster_3.0-7       sp_1.3-2
##  [9] sf_0.8-0           ggplot2_3.2.1      knitr_1.27         dplyr_0.8.3
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.5   xfun_0.11          purrr_0.3.3        lattice_0.20-38
##  [5] colorspace_1.4-1   viridisLite_0.3.0  htmltools_0.4.0    yaml_2.2.0
##  [9] XML_3.98-1.20      rlang_0.4.2        e1071_1.7-3        pillar_1.4.2
## [13] later_1.0.0        glue_1.3.1         withr_2.1.2        DBI_1.1.0
## [17] plyr_1.8.5         lifecycle_0.1.0    stringr_1.4.0      rgeos_0.5-2
## [21] munsell_0.5.0      gtable_0.3.0       htmlwidgets_1.5.1  codetools_0.2-16
## [25] leafsync_0.1.0     evaluate_0.14      fastmap_1.0.1      httpuv_1.5.2
## [29] crosstalk_1.0.0    class_7.3-15       Rcpp_1.0.3         KernSmooth_2.23-16
```

```
## [33] xtable_1.8-4       scales_1.1.0       promises_1.1.0     classInt_0.4-2
## [37] lwgeom_0.1-7       leaflet_2.0.3      mime_0.8           digest_0.6.23
## [41] stringi_1.4.3      shiny_1.4.0        grid_3.6.2         rgdal_1.4-8
## [45] tools_3.6.2        magrittr_1.5       lazyeval_0.2.2     tibble_2.1.3
## [49] dichromat_2.0-0    crayon_1.3.4       pkgconfig_2.0.3    assertthat_0.2.1
## [53] rmarkdown_2.0      R6_2.4.1           units_0.6-5        compiler_3.6.2
```

Load Raw Data

This data was available from: https://www.kaggle.com/mikejohnsonjr/united-states-crime-rates-by-county/ data with the following license: CC0 1.0 Universal (CC0 1.0)

```
df <- read.csv("crime_data.csv")
head(df)
```

```
##             county_name crime_rate_per_100000 index EDITION PART IDNO CPOPARST
## 1    St. Louis city, MO              1791.995     1       1    4 1612   318667
## 2 Crittenden County, AR              1754.915     2       1    4  130    50717
## 3  Alexander County, IL              1664.700     3       1    4  604     8040
## 4     Kenedy County, TX              1456.311     4       1    4 2681      444
## 5    De Soto Parish, LA              1447.402     5       1    4 1137    26971
## 6    Baltimore city, MD              1419.538     6       1    4 1227   625474
##   CPOPCRIM AG_ARRST AG_OFF COVIND INDEX MODINDX MURDER RAPE ROBBERY AGASSLT
## 1   318667       15     15    100  5706   22329    119  200    1778    3609
## 2    50717        4      4    100   873    3424      8   38     165     662
## 3     8040        2      2    100   127     278      1    2       5     119
## 4      444        1      1    100     6      13      0    3       1       2
## 5    26971        3      3    100   392     703      3    4      17     368
## 6   625474        9      9    100  8831   29868    216  317    3638    4660
##   BURGLRY LARCENY MVTHEFT ARSON population FIPS_ST FIPS_CTY
## 1    4995   13791    3543   464     318416      29      510
## 2    1482    1753     189    28      49746       5       35
## 3      82     184      12     2       7629      17        3
## 4       5       4       4     0        412      48      261
## 5     149     494      60     0      27083      22       31
## 6    7804   18055    4009   251     622104      24      510
```

Data Preparation

Notice how the state name is part of the "county name" variable? I'll create a new column for the state name.

```
# note that we trim whitespace and explicitly convert
# to a factor
df$state_abbr <- as.factor(trimws(sub('.*,\\s*', '', df$county_name)))
head(df)
```

```
##             county_name crime_rate_per_100000 index EDITION PART IDNO CPOPARST
## 1    St. Louis city, MO              1791.995     1       1    4 1612   318667
## 2 Crittenden County, AR              1754.915     2       1    4  130    50717
## 3  Alexander County, IL              1664.700     3       1    4  604     8040
## 4     Kenedy County, TX              1456.311     4       1    4 2681      444
## 5    De Soto Parish, LA              1447.402     5       1    4 1137    26971
## 6    Baltimore city, MD              1419.538     6       1    4 1227   625474
```

```
##   CPOPCRIM AG_ARRST AG_OFF COVIND INDEX MODINDX MURDER RAPE ROBBERY AGASSLT
## 1   318667       15     15    100  5706   22329    119  200    1778    3609
## 2    50717        4      4    100   873    3424      8   38     165     662
## 3     8040        2      2    100   127     278      1    2       5     119
## 4      444        1      1    100     6      13      0    3       1       2
## 5    26971        3      3    100   392     703      3    4      17     368
## 6   625474        9      9    100  8831   29868    216  317    3638    4660
##   BURGLRY LARCENY MVTHEFT ARSON population FIPS_ST FIPS_CTY state_abbr
## 1    4995   13791    3543   464     318416      29      510         MO
## 2    1482    1753     189    28      49746       5       35         AR
## 3      82     184      12     2       7629      17        3         IL
## 4       5       4       4     0        412      48      261         TX
## 5     149     494      60     0      27083      22       31         LA
## 6    7804   18055    4009   251     622104      24      510         MD
```

But this gives us abbreviations, we really want the names. There's a file in this directory with the mapping.

```
states <- read.csv("state.csv")
# note that we trim whitespace and explicitly convert
# to a factor
states$state_abbr <- as.factor(trimws(states$state_abbr))
head(states)
```

```
##             state_name state_abbr
## 1 District of Columbia         DC
## 2              Alabama         AL
## 3               Alaska         AK
## 4              Arizona         AZ
## 5             Arkansas         AR
## 6           California         CA
```

We'll add the state names using a right join:

```
df <- merge(x = df, y = states, by = "state_abbr", all.x = TRUE)
head(df)
```

```
##   state_abbr                 county_name crime_rate_per_100000 index EDITION
## 1         AK   Anchorage Municipality, AK              824.7217    50       1
## 2         AK  Juneau City and Borough, AK              352.1127   667       1
## 3         AK    Kodiak Island Borough, AK              856.0311    44       1
## 4         AK   Sitka City and Borough, AK              133.0377  2018       1
## 5         AK Northwest Arctic Borough, AK             1014.9642    21       1
## 6         AK         Nome Census Area, AK              232.5111  1267       1
##   PART IDNO CPOPARST CPOPCRIM AG_ARRST AG_OFF COVIND INDEX MODINDX MURDER RAPE
## 1    4   71   299143   299143        3      3    100  2482   10728     15  303
## 2    4   77    32553    32553        1      1    100   115    1099      0    9
## 3    4   80     6332     6332        1      1    100   121     254      0    8
## 4    4   87     9060     9060        1      1    100    12     143      0    1
## 5    4   85     3334     3334        1      1    100    78     210      0   23
## 6    4   83     3776     3776        1      1    100    23      44      0    3
##   ROBBERY AGASSLT BURGLRY LARCENY MVTHEFT ARSON population FIPS_ST FIPS_CTY
## 1     488    1676    1159    8724     845    98     300950       2       20
## 2      16      90      94     975      30     9      32660       2      110
## 3      10     103      23     200      31     3      14135       2      150
## 4       1      10      17     120       6     0       9020       2      220
## 5      12      43      46     126      38     2       7685       2      188
## 6       2      18      12      23       9     2       9892       2      180
```

```
##   state_name
## 1     Alaska
## 2     Alaska
## 3     Alaska
## 4     Alaska
## 5     Alaska
## 6     Alaska
```

Next we need some aggregates by state. I'll just take the sum here, but we could choose different functions if we wanted to.

```
data <- df %>%
          dplyr::select(state_name, population, ARSON, ROBBERY, MURDER) %>%
          group_by(state_name) %>%
          summarise_all(sum)

head(data)
```

```
## # A tibble: 6 x 5
##   state_name population ARSON ROBBERY MURDER
##   <fct>           <int> <int>   <int>  <int>
## 1 Alabama       4833722  1025    4988    341
## 2 Alaska         718165   136     591     21
## 3 Arizona       6626624  1405    7263    325
## 4 Arkansas      2959373   564    2302    173
## 5 California   38332521  7395   56484   1879
## 6 Colorado      5268367   909    3372    152
```

Maps!

First we append our data into the map object. Note the warning that we only have data for the lower 48.

```
map_obj <- append_data(us_states,
                       data,
                       key.shp = "NAME",
                       key.data = "state_name")
```

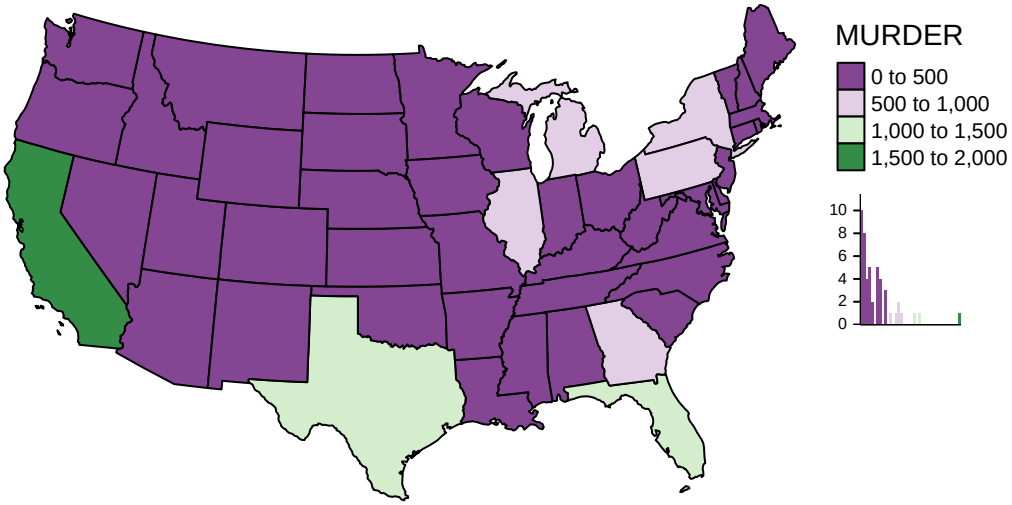
```
## Warning: This function is deprecated and has been migrated to github.com/
## mtennekes/oldtmaptools
```

```
## Over coverage: 2 out of 51 data records were not appended. Run over_coverage() to get the correspondi
```

Now we can plot this bad boi!

```
MURDER <- tm_shape(map_obj, projection = 2163) +
  tm_polygons("MURDER",
              title = "MURDER",
                id = "NAME",
                legend.hist = TRUE,
                palette = "PRGn") +
    tm_style("col_blind", legend.outside=TRUE) +
    tm_layout(frame = FALSE)

MURDER
```


MURDER
0 to 500
500 to 1,000
1,000 to 1,500
1,500 to 2,000
10
8
6
4
2
0