Making Maps with R - Part I
Displaying Maps

Xuemao Zhang
Department of Mathematics
East Stroudsburg University

June 26, 2019

Outline

- Overview
- Display maps using R packages
- Display maps using shapefiles
- Projections
- Interactive maps

Overview

- For analysis of geographical data, R can provide maps of data in the same computing environment that the data analysis is being performed in.
- A lot of R packages such as `maps` and `usmap` produce `ggplot` object which allows for adding ggplot layers.
- R has been acquiring much of the functionality of traditional GIS packages such as `sp`, `rgeos`, `spData`, `shapefiles`, `maptools` and `raster`.
- The main place to go to get an overview of the kinds and capabilities of the spatial packages in R is the Spatial Task Views on CRAN.

Overview

- Check whether all packages installed (install them if not), and then load all packages.

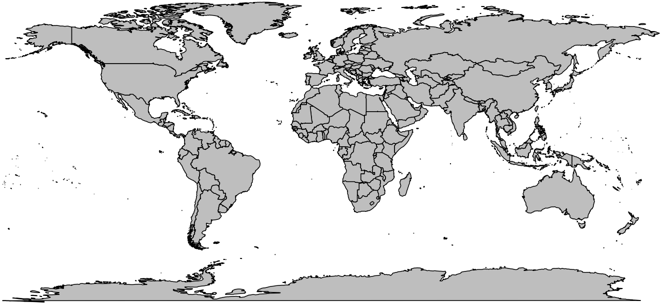
```
list.of.packages = c("ggplot2","maps","usmap","rgdal",
                      "leaflet","gridExtra");
new.packages = list.of.packages[!(list.of.packages %in%
installed.packages()[,"Package"])];
if(length(new.packages)) install.packages(new.packages);
```

Overview

- Geographical points are on a sphere while maps are ploted on a flat surface.
- There are two different types of Coordinate Reference Systems in GIS (Geographic Information System).
  - Geographic coordinate systems: coordinate systems that span the entire globe (e.g. latitude / longitude).
  - Projected coordinate systems: coordinate systems that are localized to minimize visual distortion in a particular region (e.g. Robinson, **UTM**, State Plane)
- The package `usmap` uses UTM Coordinates.
- For more information, see Penn State E-education UTM Coordinate System.
- The function `convUL()` in the package `PBSmapping` can be used to convert coordinates between UTM and Lon/Lat.

Display maps: World map

```
library(maps);
maps::map('world',fill=TRUE,border='black',col='gray', bg='white');
```



Display maps: World map

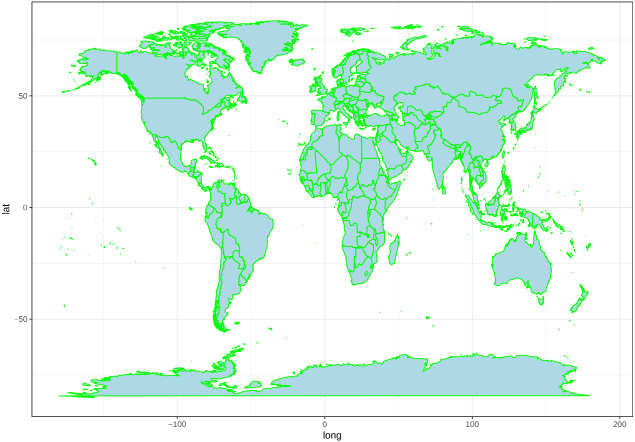
- To map individual countries, we use the regions parameter.

```
par(mfrow=c(1,3));
map("world", "italy");
map("world", "france");
map("world", "spain");
```

Display maps: World map

- map_data() in ggplot2 can easily turn data from the maps package in to a data frame suitable for plotting with ggplot2.

```
worldmap = ggplot2::map_data('world');
ggplot(worldmap, aes(long, lat, group=group)) +
  geom_polygon(fill='lightblue', colour="green") + theme_bw();
```



Display maps: World map

```
str(worldmap);
```

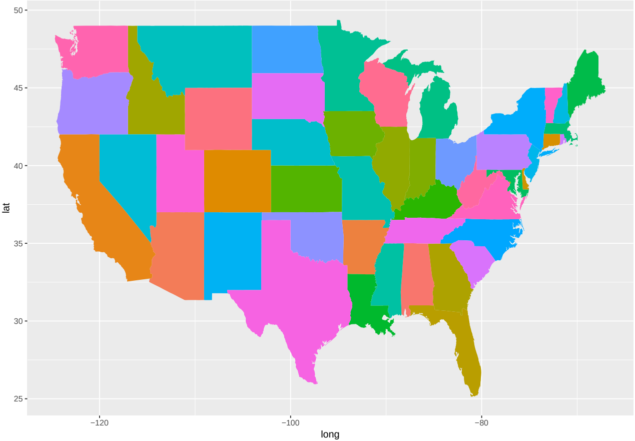
```
## 'data.frame':    99338 obs. of  6 variables:
##  $ long     : num  -69.9 -69.9 -69.9 -70 -70.1 ...
##  $ lat      : num  12.5 12.4 12.4 12.5 12.5 ...
##  $ group    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ order    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ region   : chr  "Aruba" "Aruba" "Aruba" "Aruba" ...
##  $ subregion: chr  NA NA NA NA ...
```

```
head(worldmap);
```

```
##        long      lat group order region subregion
## 1 -69.89912 12.45200     1     1  Aruba      <NA>
## 2 -69.89571 12.42300     1     2  Aruba      <NA>
## 3 -69.94219 12.43853     1     3  Aruba      <NA>
## 4 -70.00415 12.50049     1     4  Aruba      <NA>
## 5 -70.06612 12.54697     1     5  Aruba      <NA>
## 6 -70.05088 12.59707     1     6  Aruba      <NA>
```

Display maps: US map

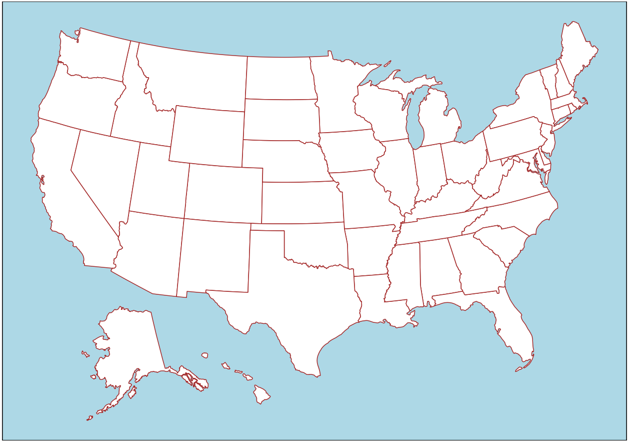
```
usmap = map_data("state");
ggplot(usmap, aes(long, lat, group=group, fill=region)) +
  geom_polygon(show.legend = F);
```



Display maps: US map including Alaska and Hawaii

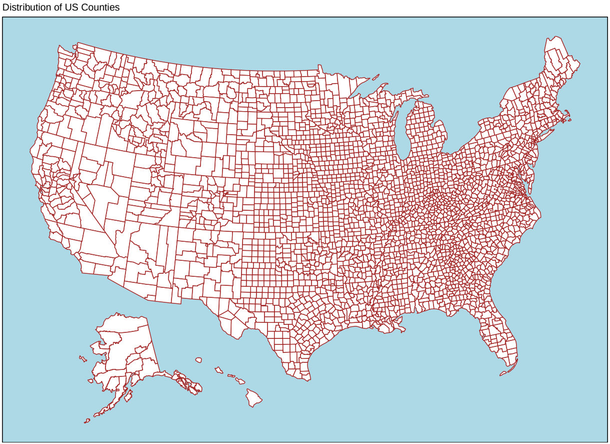
- theme() in ggplot2 is used to modify a plot's theme.

```
library(usmap);
usmap::plot_usmap(regions = "states", lines = "brown")+
  theme(panel.background = element_rect(colour = "black",
                          fill = "lightblue"));
```



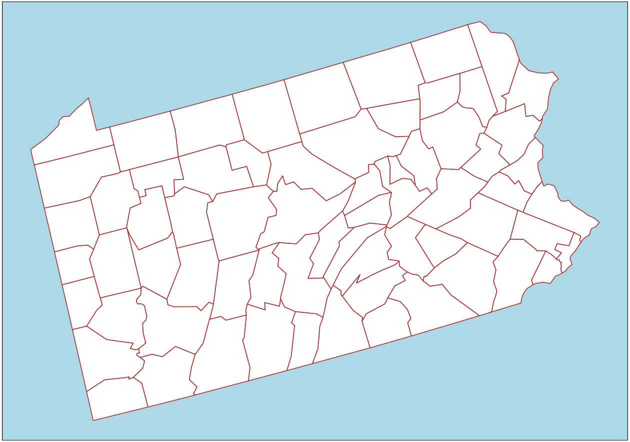
Display maps: US map including Alaska and Hawaii

```r
usmap::plot_usmap(regions = "counties",lines = "brown") +
  labs(title = "Distribution of US Counties") +
  theme(panel.background = element_rect(colour = "black",
                          fill = "lightblue"));
```



Display maps: US map with specified states

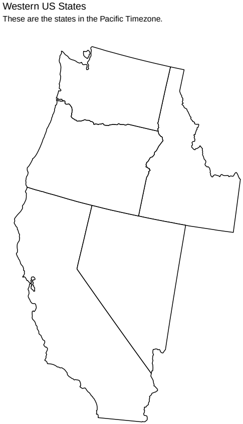
```
plot_usmap(regions = "counties", include = c("PA"),
           lines = "brown") +
theme(panel.background = element_rect(colour = "black",
            fill = "lightblue"));
```



Display maps

- You can also specify multiple regions to map them all at once.

```
plot_usmap(include = c("CA", "ID", "NV", "OR", "WA")) +
labs(title = "Western US States",
subtitle = "These are the states in the Pacific Timezone.");
```



Display maps

- You have to know the names of the regions in the database though. To find that out, enter the following in the console.
- “fips” is for each state’s FIPS code. FIPS stands for “Federal Information Processing Standard”, and pretty much any geography-based data from the government uses it.

```
state.fips;  # See state region names
```

```
##     fips ssa region division abb                      polyname
## 1      1   1      3        6  AL                       alabama
## 2      4   3      4        8  AZ                       arizona
## 3      5   4      3        7  AR                      arkansas
## 4      6   5      4        9  CA                    california
## 5      8   6      4        8  CO                      colorado
## 6      9   7      1        1  CT                   connecticut
## 7     10   8      3        5  DE                      delaware
## 8     11   9      3        5  DC          district of columbia
## 9     12  10      3        5  FL                       florida
## 10    13  11      3        5  GA                       georgia
## 11    16  13      4        8  ID                         idaho
```

Display maps using shapefiles

- The disadvantage of using R packages is that you're limited to the regions that the package offers.
- Using shapefiles provides more flexibility.
- Shapefiles are specifcally for geographic data. The file format encodes points, lines, and polygons in geographic space and is a common way to distribute spatial data. The file extension is .shp.
- The nice thing about R, using the `rgdal` package, is that you can load this data in a straightforward way.
- The rgdal package lets you load the file with the `readOGR()` function.

Display maps using shapefiles: US

- First, go to Census Bureau site https://www.census.gov/geo/maps-data/data/cbf/cbf_state.html download the files. The files should be already downloaded. Use `setwd()` to set working directory.

```
library(rgdal);
usa = rgdal::readOGR("F:/DataCamp/data/cb_2017_us_state_20m.shp"); #
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "F:\DataCamp\data\cb_2017_us_state_20m.shp", layer: "cb_2
## with 52 features
## It has 9 fields
## Integer64 fields read as strings:  ALAND AWATER
```

Display maps using shapefiles: US

- ggplot2::fortify or broom::tidy converts the shape file to data.frame #https://cran.r-project.org/web/packages/broom/vignettes/broom.html

```
library(ggplot2);
usadf = ggplot2::fortify(usa);
```

```
## Regions defined for each Polygons
```

```
str(usadf); head(usadf);
```

```
## 'data.frame':    13832 obs. of  7 variables:
##  $ long : num  -168 -168 -168 -167 -167 ...
##  $ lat  : num  65.7 65.7 65.8 65.9 66.1 ...
##  $ order: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ hole : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ piece: Factor w/ 47 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1
##  $ id   : chr  "0" "0" "0" "0" ...
##  $ group: Factor w/ 132 levels "0.1","0.2","0.3",..: 1 1 1 1 1 1
```

```
##        long      lat order  hole piece id group
## 1 -168.1289 65.65574     1 FALSE     1  0   0.1
## 2 -167.9799 65.72797     2 FALSE     1  0   0.1
```

Display maps using shapefiles: US

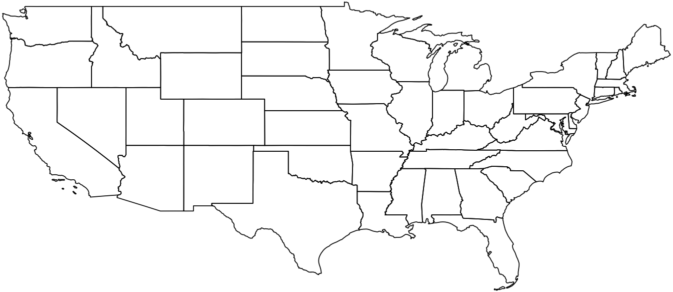
- Then use plot() to draw the map.
- Because it's from the Census Bureau, you get boundaries not only for the conterminous United States, but also Alaska, Hawaii, and Puerto Rico.

```
plot(usa);
```

Display maps using shapefiles: US

- For only the conterminous United States, you can subset out using FIPS codes. The codes for Alaska, Hawaii, and Puerto Rico are 02, 15, and 72, respectively.

```
conterm = usa[usa$STATEFP != "02" & usa$STATEFP != "15" &
                usa$STATEFP != "72",];
plot(conterm);
```



Display maps using shapefiles: World

- The process is the same for other shapefiles. For example, using shapefiles from Natural Earth, here's how to load country boundaries for the world.

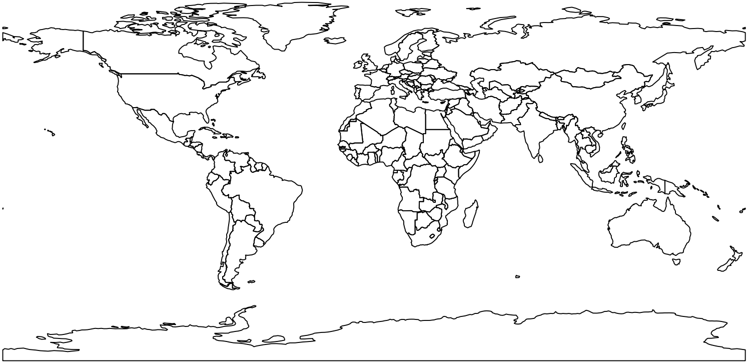
```
countries=readOGR("F:/DataCamp/data/ne_110m_admin_0_countries.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "F:\DataCamp\data\ne_110m_admin_0_countries.shp", layer:
## with 177 features
## It has 94 fields
## Integer64 fields read as strings:  POP_EST NE_ID
```

Display maps using shapefiles: World

- Here's the result.

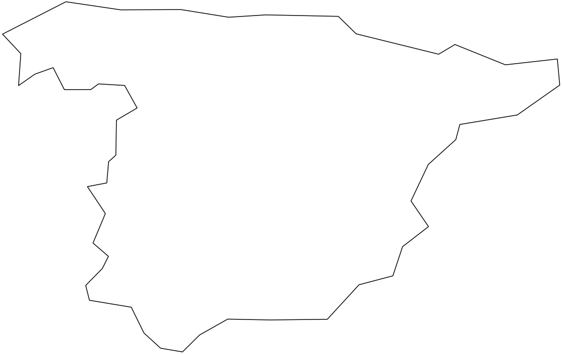
```r
plot(countries);
```



Display maps using shapefiles: World

- You can also subset for specific countries in the world file.

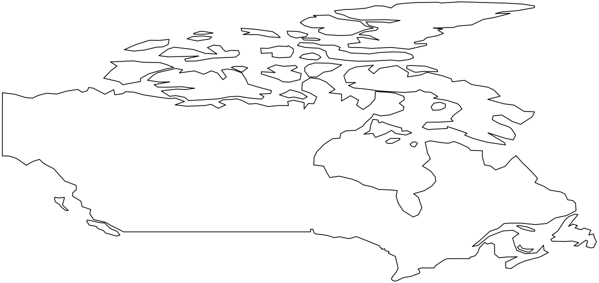
```
spain=countries[countries$NAME == 'Spain',];
plot(spain);
```



Display maps using shapefiles: World

- You can also subset for specific countries in the world file.

```
canada=countries[countries$NAME == 'Canada',];
plot(canada);
```



Display maps using shapefiles: World

- We can see all other country names.

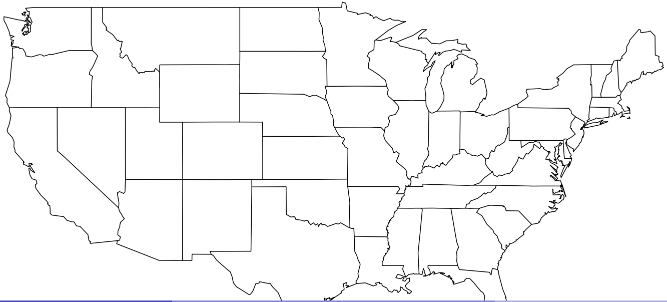
```
countries$NAME;
```

```
##   [1] Fiji                     Tanzania
##   [3] W. Sahara                Canada
##   [5] United States of America Kazakhstan
##   [7] Uzbekistan               Papua New Guinea
##   [9] Indonesia                Argentina
##  [11] Chile                    Dem. Rep. Congo
##  [13] Somalia                  Kenya
##  [15] Sudan                    Chad
##  [17] Haiti                    Dominican Rep.
##  [19] Russia                   Bahamas
##  [21] Falkland Is.             Norway
##  [23] Greenland                Fr. S. Antarctic Lands
##  [25] Timor-Leste              South Africa
##  [27] Lesotho                  Mexico
##  [29] Uruguay                  Brazil
##  [31] Bolivia                  Peru
##  [33] Colombia                 Panama
```

Projections

- Longitude/latitude position points on a sphere; maps are drawn on a flat surface.
- Consider the `maps` package.
- The default uses a **rectangular projection** with the aspect ratio chosen so that longitude and latitude scales are equivalent at the center of the map.

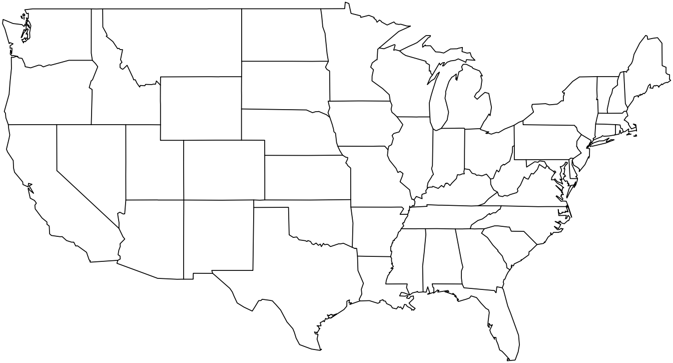
```r
maps::map("state");
```



Projections

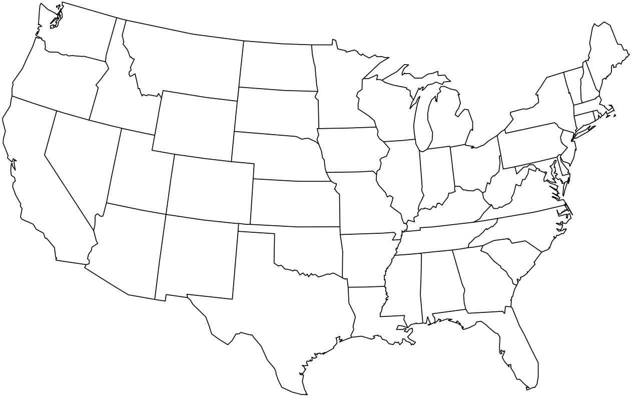
- Other projections try to preserve angles or areas.

```r
maps::map("state", project = "mercator");
```



Projections

```
maps::map("state", project = "bonne", param = 45);
```



Interactive Maps by R:leaflet

- Leaflet is one of the most popular open-source JavaScript libraries for interactive maps. It's used by websites ranging from The New York Times and The Washington Post to GitHub and Flickr, as well as GIS specialists like OpenStreetMap, Mapbox, and CartoDB.
- Run the following code in RStudio.

```
library(leaflet);
m=leaflet() %>% addTiles() %>%
# Add map tiles (Default: OpenStreetMap)
 addMarkers(lng=-75.173, lat=40.997,
  popup="East Stroudsburg University");
options(viewer = NULL); #view in a browser
m;  #print the map
```

Questions?

?