SF Summary

2022-03-14

Get the input date

```
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.1.3
```

```
## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

Creat simple sf object

```
p <- rbind(c(10, 40), c(40, 30), c(20, 20), c(30, 10))
(mp <- st_multipoint(p))
```

```
## MULTIPOINT ((10 40), (40 30), (20 20), (30 10))
```

```
?st_multipoint()
```

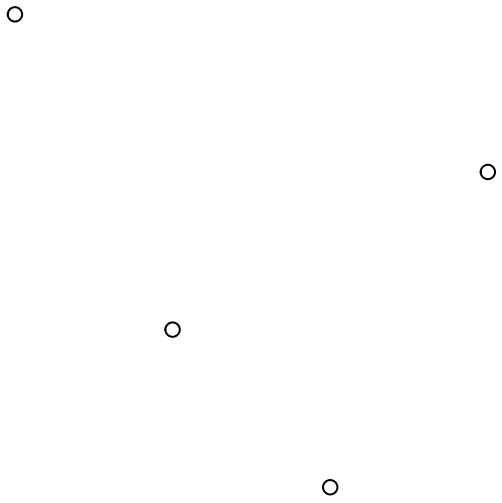
```
## starting httpd help server ...
```

```
##  done
```

```
class(mp)
```

```
## [1] "XY"         "MULTIPOINT" "sfg"
```

```
plot(mp)
```



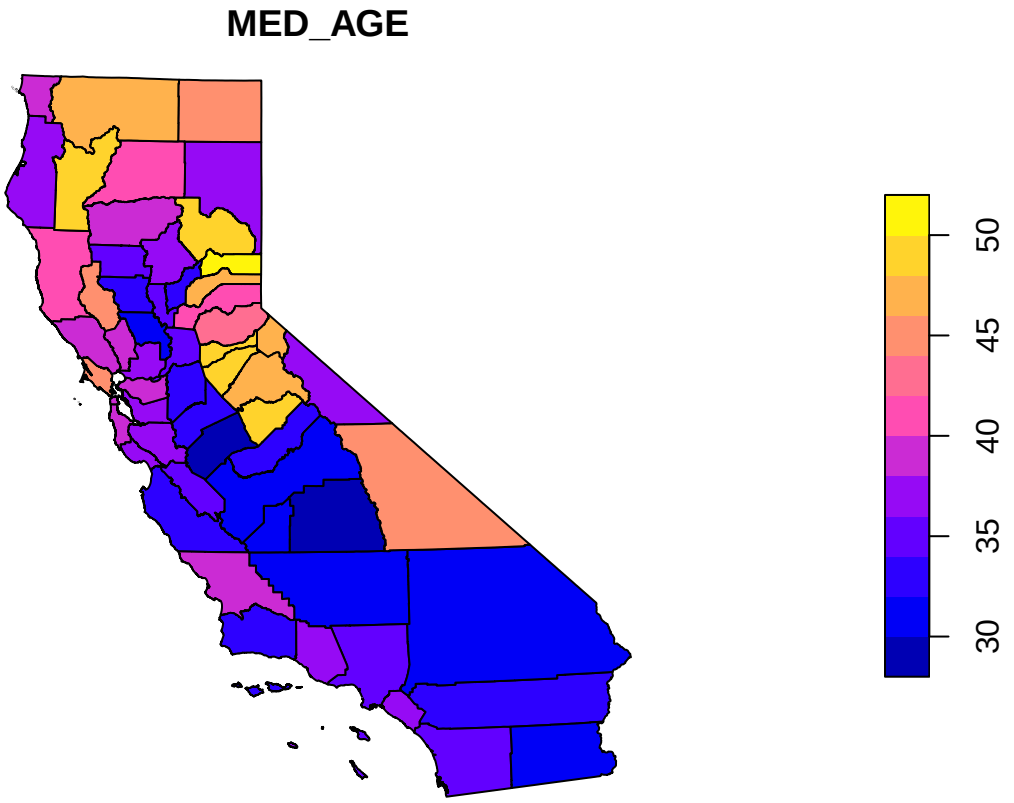
County data

- An `sf` object is a standard R `data.frame` extended with spatial data (geometry) and spatial operations.

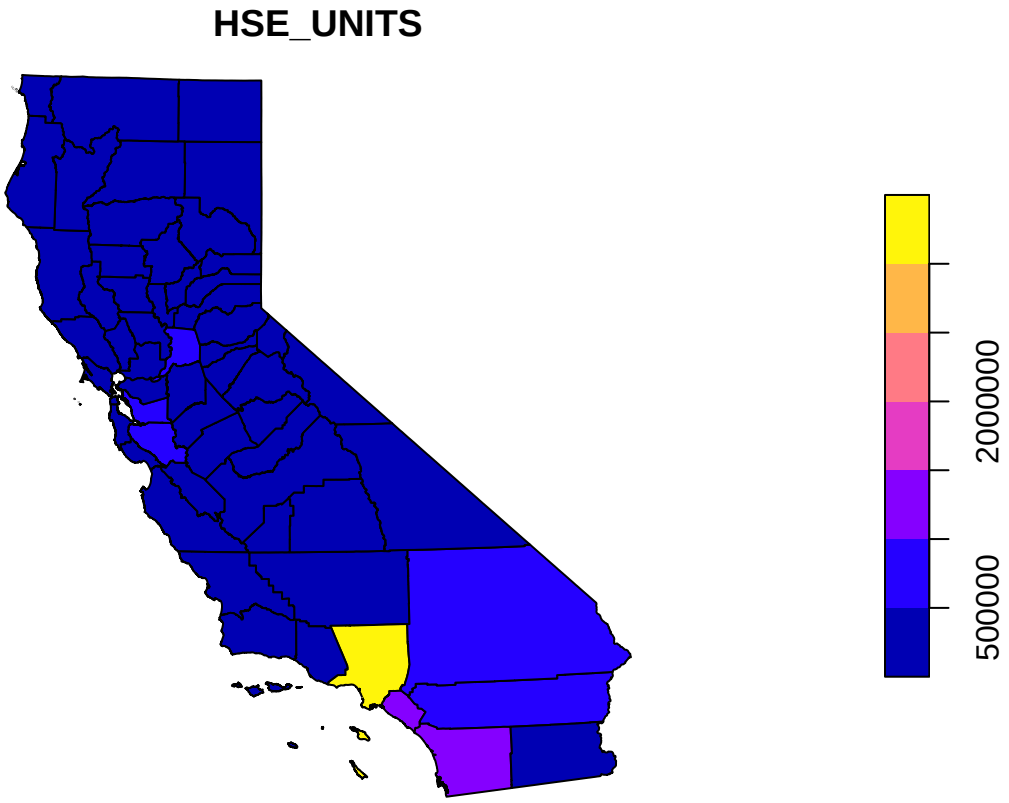
```
counties <- st_read("~/GitHub/R-Geospatial-Fundamentals/notebook_data/california_counties")
```

```
## Reading layer 'CaliforniaCounties' from data source
##   'C:\Users\bill.chung\OneDrive - Danaher\Documents\GitHub\R-Geospatial-Fundamentals\notebook_data\ca
##   using driver 'ESRI Shapefile'
## Simple feature collection with 58 features and 24 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -374445.4 ymin: -604500.7 xmax: 540038.5 ymax: 450022
## Projected CRS: NAD83 / California Albers
```

```
#This returns a **choropleth** map of median male age, by county!
plot(counties['MED_AGE'])
```




```
plot(counties['HSE_UNITS'])
```



- Based on the output of the head command below, can you guess why they call the geometry column `sticky`?

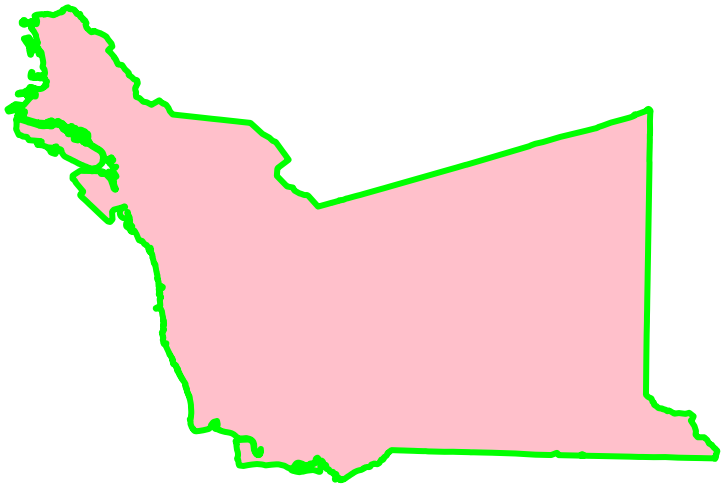
```
counties2 = counties[c('NAME','POP2012','MED_AGE')]
head(counties2)
```

```
## Simple feature collection with 6 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -267387.9 ymin: -578158.6 xmax: 216677.6 ymax: 352693.6
## Projected CRS: NAD83 / California Albers
##          NAME POP2012 MED_AGE                       geometry
## 1        Kern  851089    30.7 MULTIPOLYGON (((213672.6 -2...
## 2       Kings  155039    31.1 MULTIPOLYGON (((12524.03 -1...
## 3        Lake   65253    45.0 MULTIPOLYGON (((-235734.3 1...
## 4      Lassen   35039    37.0 MULTIPOLYGON (((12.28914 35...
## 5 Los Angeles 9904341    34.8 MULTIPOLYGON (((173874.5 -4...
## 6      Madera  153025    33.1 MULTIPOLYGON (((16681.16 -1...
```

How to subset county

```
alameda_county = counties[counties$NAME == 'Alameda',]
# Plot our newly subsetted sf object
plot(alameda_county$geometry, col='pink', border='green', lwd=3, main='Alameda County, Why not?')
```

Alameda County, Why not?



How to save a file

```
st_write(alameda_county,
         paste0(getwd(),"/alameda_county/","alameda_county.shp"),
         delete_dsn = T)
```

```
## Deleting source 'C:/Users/bill.chung/OneDrive - Danaher/Documents/GitHub/R-Geospatial-Fundamentals/al
## Writing layer 'alameda_county' to data source
##   'C:/Users/bill.chung/OneDrive - Danaher/Documents/GitHub/R-Geospatial-Fundamentals/alameda_county/a
## Writing 1 features with 24 fields and geometry type Multi Polygon.
```

```
st_write(alameda_county,
         paste0(getwd(),"/alameda_county/","alameda_county.json"),
         driver = "GeoJSON",
         delete_dsn = T)
```

```
## Deleting source 'C:/Users/bill.chung/OneDrive - Danaher/Documents/GitHub/R-Geospatial-Fundamentals/al
## Writing layer 'alameda_county' to data source
##   'C:/Users/bill.chung/OneDrive - Danaher/Documents/GitHub/R-Geospatial-Fundamentals/alameda_county/a
## Writing 1 features with 24 fields and geometry type Multi Polygon.
```

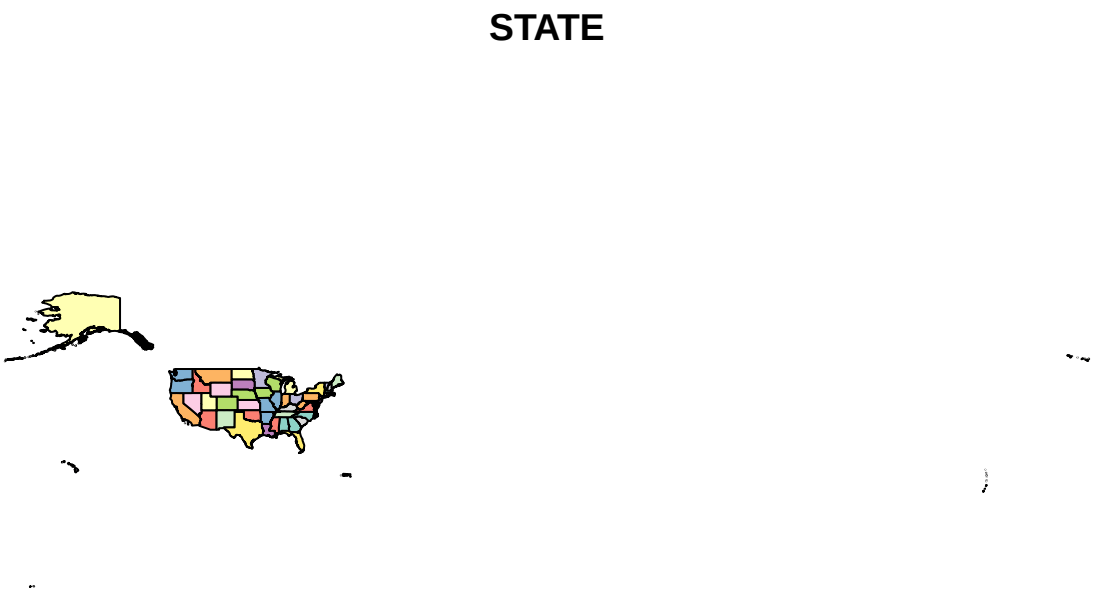
Coordiate translation

US State shape file

```
states <- st_read("~/GitHub/R-Geospatial-Fundamentals/notebook_data/us_states")
```

```
## Reading layer ‘us_states’ from data source
##   ‘C:\Users\bill.chung\OneDrive - Danaher\Documents\GitHub\R-Geospatial-Fundamentals\notebook_data\us
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 56 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -179.1482 ymin: -14.37374 xmax: 179.7739 ymax: 71.35256
## Geodetic CRS:  WGS 84
```

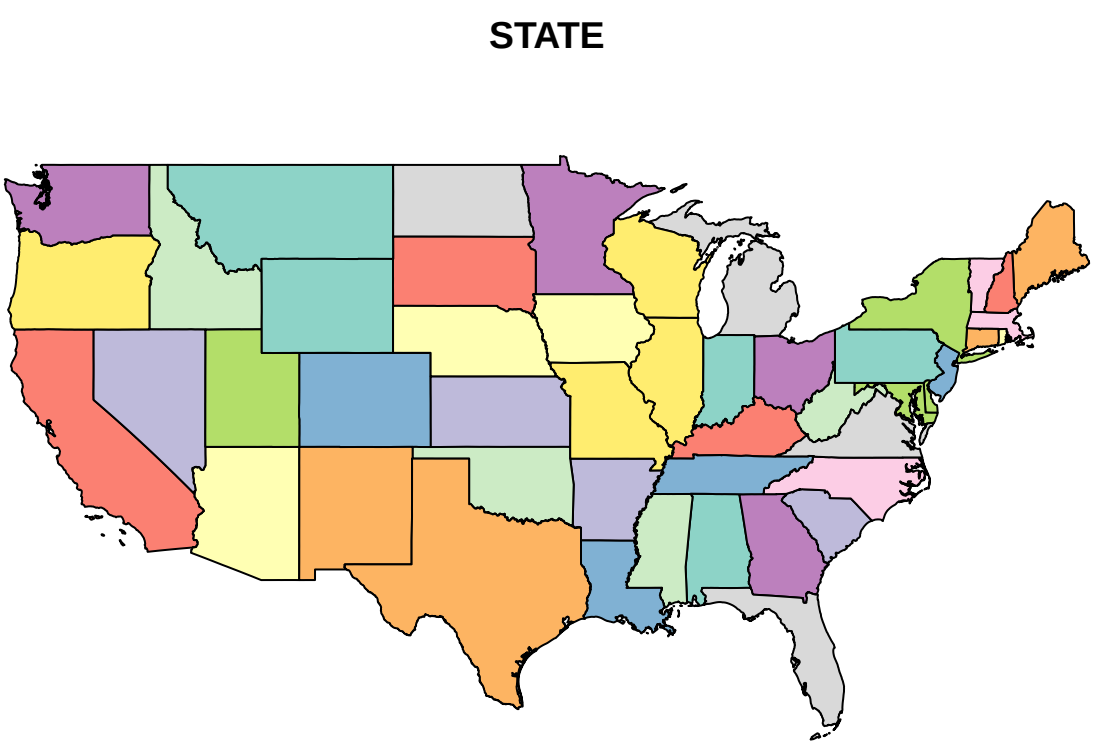
```
# Plot our states data
plot(states['STATE'])
```



Beyond the 50 states we seem to have American Samoa, Puerto Rico, Guam, Commonwealth of the Northern Mariana Islands, and United States Virgin Islands included in this spatial dataframe. To make our map cleaner, let's limit the states to the contiguous states (so we'll also exclude Alaska and Hawaii).

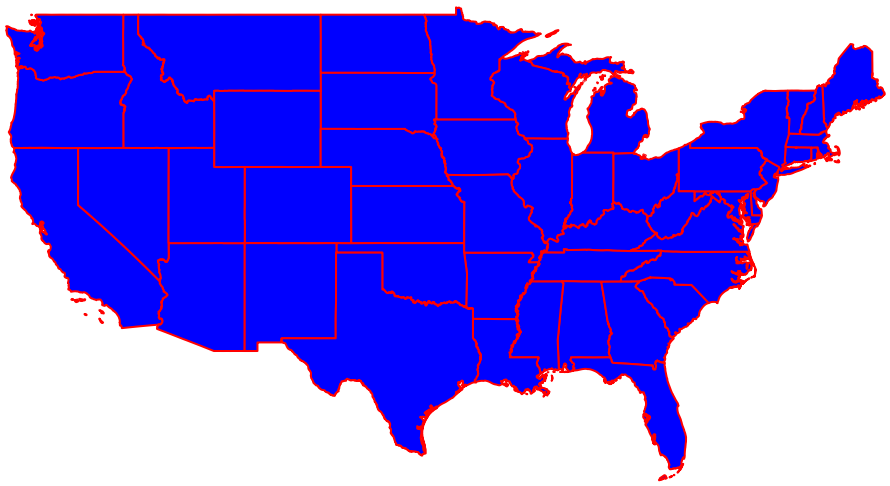
```
# Define list of non-contiguous states
non_contiguous_us = c('American Samoa',
                      'Puerto Rico',
                      'Guam',
                      'Commonwealth of the Northern Mariana Islands',
                      'United States Virgin Islands',
                      'Alaska',
                      'Hawaii')

# Limit data according to above list
states_limited = states[!states$STATE %in% non_contiguous_us, ]
```

```
# Plot it
plot(states_limited['STATE'])
```



To prepare for our mapping overlay, let's make our states a nice, bold color.

```
plot(states_limited$geometry,
     col = 'blue',
     border = 'red',
     lwd = 1)
```



Plotting maps with different coordiate system

```
plot(counties$geometry, col = 'lightgrey', border = 'white')
plot(states_limited$geometry, col = 'blue', border = 'red', lwd = 5, add = T)
```

```
print(st_bbox(counties))
```

```
##      xmin      ymin      xmax      ymax
## -374445.4 -604500.7  540038.5  450022.0
```

```
print(st_bbox(states_limited))
```

```
##       xmin       ymin       xmax       ymax
## -124.73183   24.54547  -66.97626   49.38436
```

```
st_crs(counties)[1]
```

```
## $input
## [1] "NAD83 / California Albers"
```

```
st_crs(states_limited)[1]
```

```
## $input
## [1] "WGS 84"
```

Converting

```
# Convert the states data to Web Mercator
states_limited_3857 = st_transform(states_limited, crs = 3857)
counties_3857 = st_transform(counties, crs = 3857)

st_crs(states_limited_3857)[1]
```

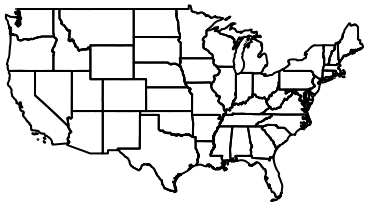
```
## $input
## [1] "EPSG:3857"
```

- Another way of converting

We can also do the transformation the following way to make sure the CRS values match:

```
# Make sure the CRSs match!
counties_3857 = st_transform(counties, st_crs(states_limited_3857))
```

```
par(mfrow=c(1,2))

# plot geographic sf data.frame
plot(states_limited$geometry, asp = 1)

# plot utm sf data.frame
plot(states_limited_3857$geometry, asp = 1)
```



```
# reset plot rows/cols
par(mfrow=c(1,1))
```

```
plot(states_limited_3857$geometry, col = 'lightgrey')
plot(counties_3857$geometry, col='darkgreen', add=T)
```