Spatial Data in R

Working with simple features

2021-08-15

Outline

1. Setup
2. Simple feature overview
3. Basic plotting
4. Reading and writing
5. Projecting
6. Attribute operations
7. Spatial operations
8. Geometry operations

Setup

- Download and open the worksheet *simple_features.Rmd*.
- Install packages

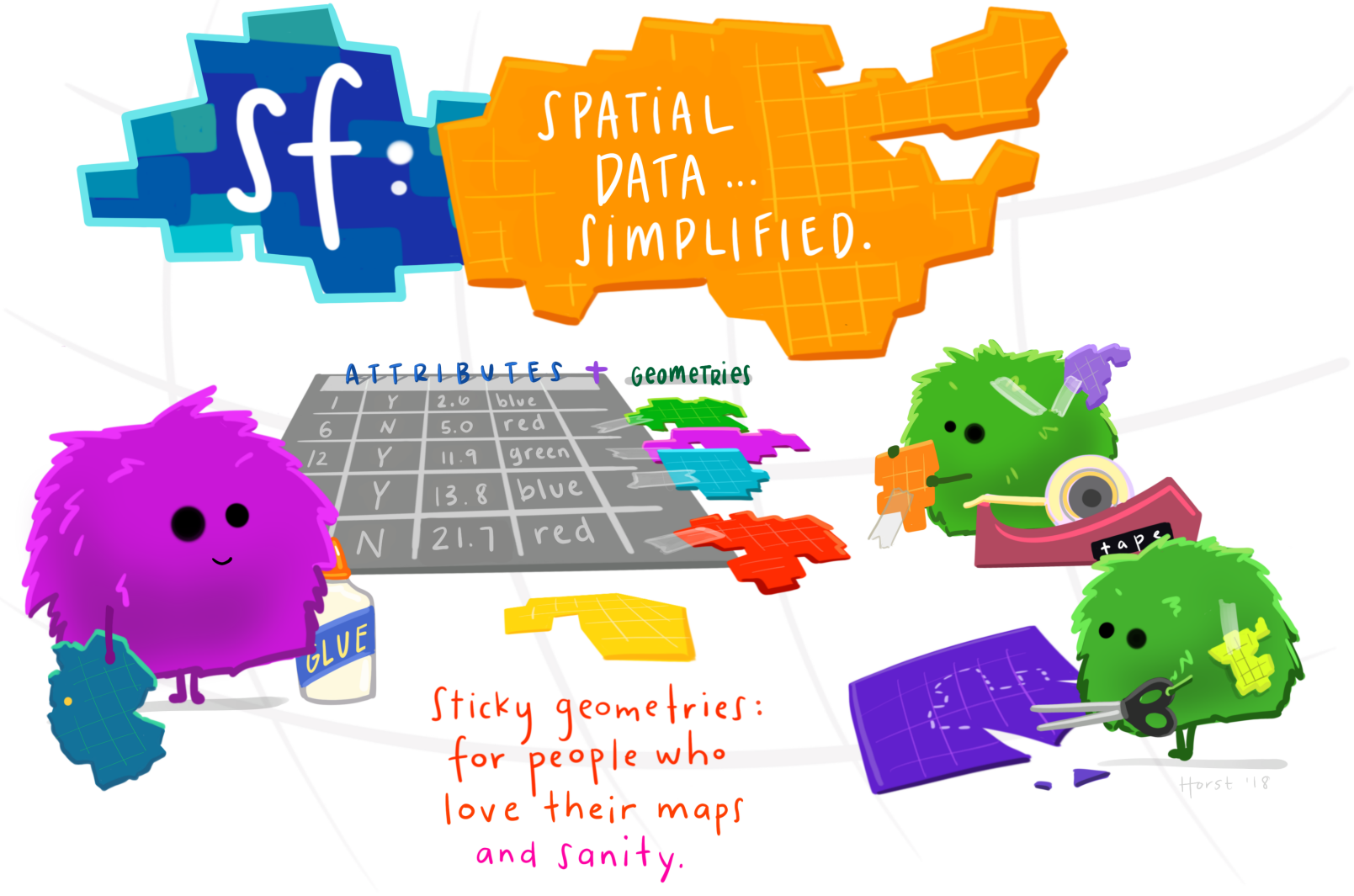
```
install.packages(c("dplyr", "here", "spData", "sf", "viridis"))

install.packages("spDataLarge",
                 repos = "https://nowosad.github.io/drat/",
                 type = "source")
```

- Attach packages

```
library(spData)
library(spDataLarge)
library(sf)
library(viridis)
library(dplyr)
```

Why `sf`?

1. It's **fast** processing,
2. It's **pretty** maps,
3. It's **rectangular** data,
4. It's **tidy** design, and
5. It's **consistent** syntax, just look for the `st` prefix



Artwork by Allison Horst.

Anatomy of a Simple Feature

```
world
```

```
## Simple feature collection with 177 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -90 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 177 x 6
##   name_long  continent  area_km2     pop gdpPercap                       geom
##   <chr>      <chr>         <dbl>   <dbl>     <dbl>         <MULTIPOLYGON [°]>
## 1 Fiji       Oceania      1.93e4  8.86e5     8222. (((180 -16.06713, 180 -16.55~
## 2 Tanzania   Africa       9.33e5  5.22e7     2402. (((33.90371 -0.95, 34.07262 ~
## 3 Western S~ Africa       9.63e4 NA            NA  (((-8.66559 27.65643, -8.665~
## 4 Canada     North Ame~   1.00e7  3.55e7    43079. (((-122.84 49, -122.9742 49,~
## 5 United St~ North Ame~   9.51e6  3.19e8    51922. (((-122.84 49, -120 49, -117~
## # ... with 172 more rows
```

Feature Geometry (sfg)

From the ground up

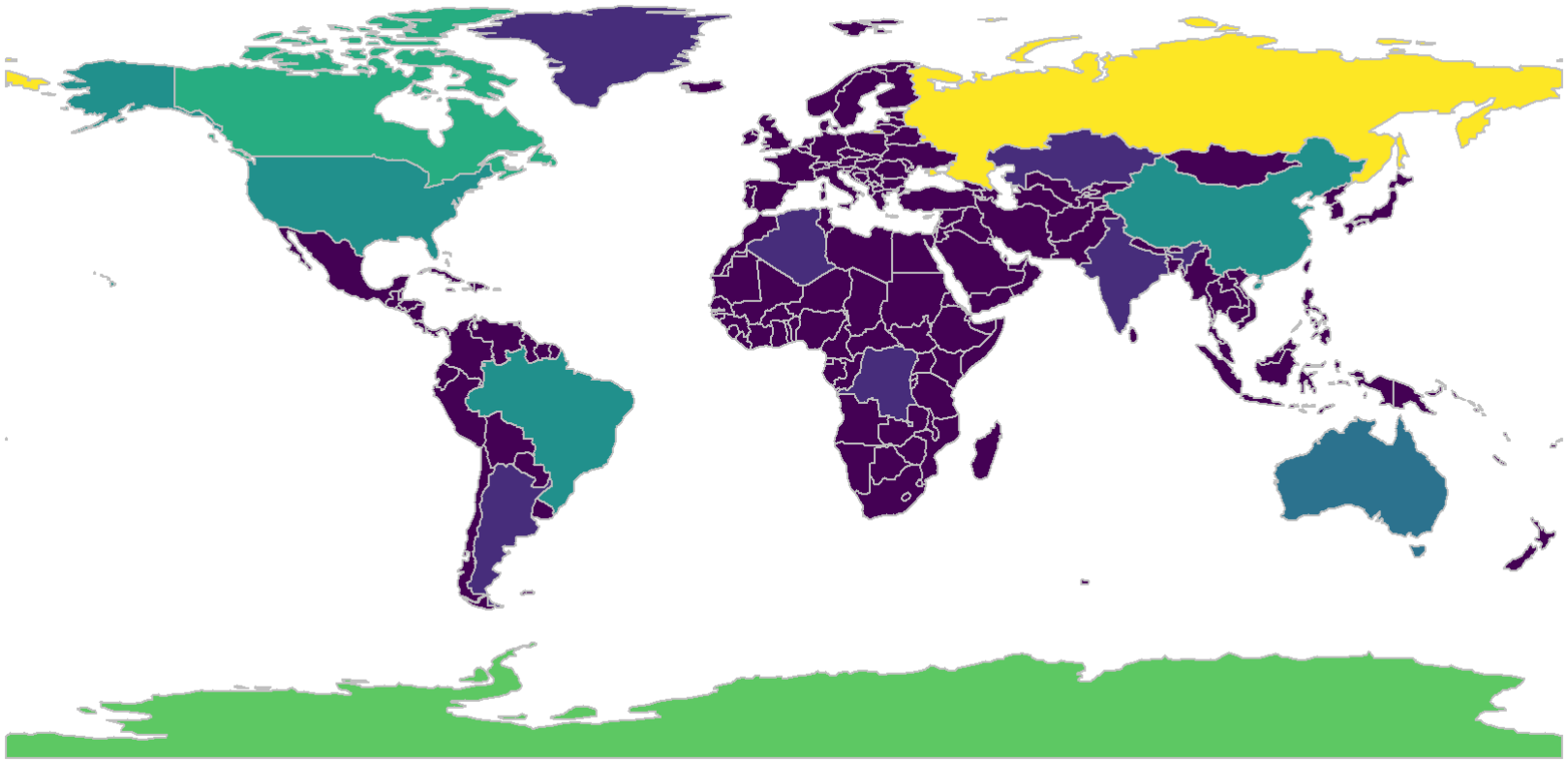
```
point1 <- st_point(c(432000, 4513100))           # create the feature geometry (sfg)
point2 <- st_point(c(436750, 4518500))

sf_column <- st_sfc(point1, point2, crs = 26912) # create the geometry column (sfc)

st_sf(id = 1:2,                                  # create the simple feature table
      location = "red butte canyon",
      geometry = sf_column)
```

```
## Simple feature collection with 2 features and 2 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 432000 ymin: 4513100 xmax: 436750 ymax: 4518500
## Projected CRS: NAD83 / UTM zone 12N
##   id         location               geometry
## 1  1 red butte canyon POINT (432000 4513100)
## 2  2 red butte canyon POINT (436750 4518500)
```

Similar to `data.frame(col1 = ..., col2 = ...)`, but with a geometry column.

Plot Attribute

```
plot(world["area_km2"], pal = viridis)
```



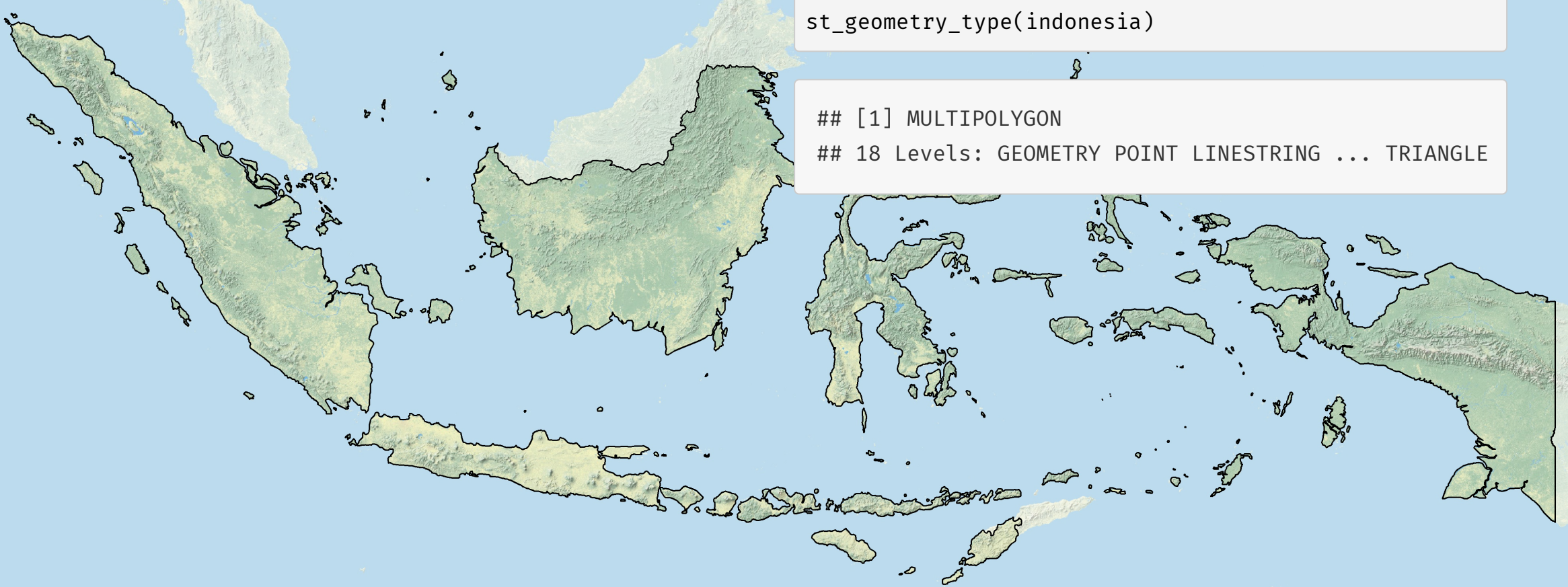
Plot Geometry

```
plot(st_geometry(world), col = "gray90")
```

Geometry type?

```
indonesia <- subset(world, name_long == "Indonesia")

st_geometry_type(indonesia)
```

```
## [1] MULTIPOLYGON
## 18 Levels: GEOMETRY POINT LINESTRING ... TRIANGLE
```



Coordinate Reference System

```
st_crs(world)
```

```
## Coordinate Reference System:
##   User input: EPSG:4326
##   wkt:
## GEOGCRS["WGS 84",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["Horizontal component of 3D system."],
##         AREA["World."],
##         BBOX[-90,-180,90,180]],
##     ID["EPSG",4326]]
```

```
st_crs(world)$epsg
```

```
## [1] 4326
```

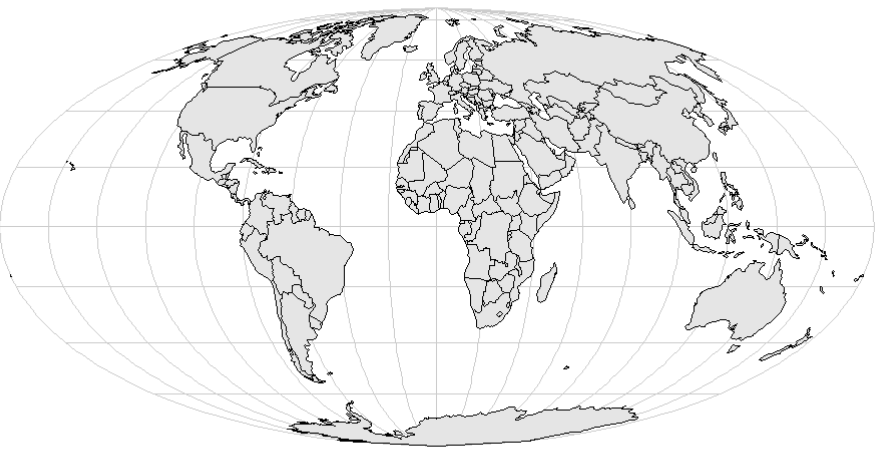
Re-projecting Simple Features

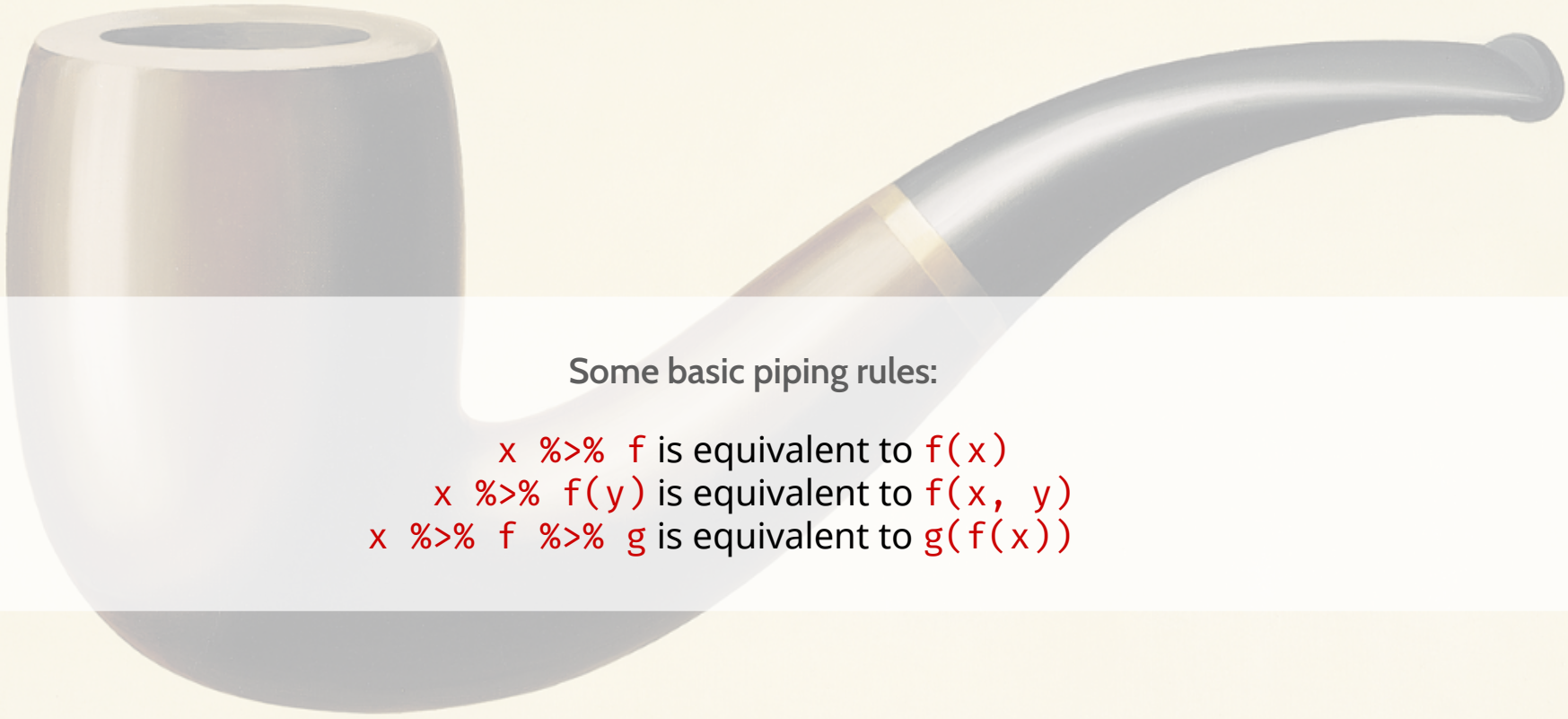
Google Pseudo-Mercator

```
st_transform(world, crs = 3857)
```

World Mollweide

```
st_transform(world, crs = "+proj=moll")
```

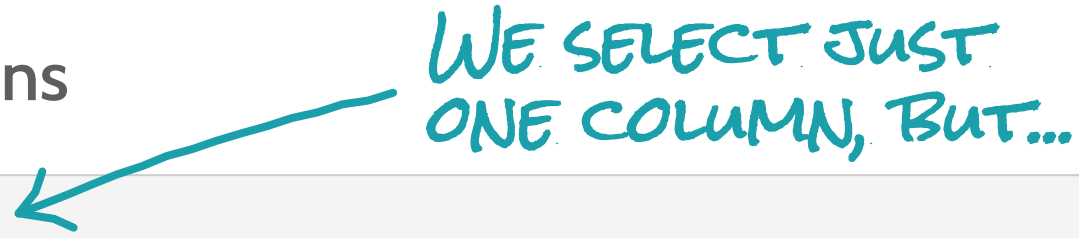




Some basic piping rules:

`x %>% f` is equivalent to `f(x)`
`x %>% f(y)` is equivalent to `f(x, y)`
`x %>% f %>% g` is equivalent to `g(f(x))`

`select()` columns




```
world %>% select(name_long)
```

```
## Simple feature collection with 177 features and 1 field
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 177 x 2
##   name_long                                                       geom
##   <chr>                                                <MULTIPOLYGON [°]>
## 1 Fiji          (((-180 -16.55522, -179.9174 -16.50178, -179.7933 -16.02088, -1~
## 2 Tanzania      (((33.90371 -0.95, 31.86617 -1.02736, 30.76986 -1.01455, 30.419~
## 3 Western Saha~ (((-8.66559 27.65643, -8.817828 27.65643, -8.794884 27.1207, -9~
## 4 Canada        (((-132.71 54.04001, -133.18 54.16998, -133.2397 53.85108, -133~
## 5 United States (((-171.7317 63.78252, -171.7911 63.40585, -171.5531 63.31779, ~
## # ... with 172 more rows
```

`filter()` rows

```
world %>% filter(area_km2 < 10000)
```

```
## Simple feature collection with 7 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -67.24243 ymin: -16.59785 xmax: 167.8449 ymax: 50.12805
## Geodetic CRS:  WGS 84
## # A tibble: 7 x 6
##   name_long  continent area_km2     pop gdpPercap                       geom
## * <chr>      <chr>        <dbl>   <dbl>     <dbl>             <MULTIPOLYGON [°]>
## 1 Puerto Ri~ North Am~    9225. 3534874    35066. (((-66.28243 18.51476, -67.10~
## 2 Palestine  Asia         5037. 4294682     4320. (((35.39756 31.48909, 35.5452~
## 3 Vanuatu    Oceania      7490.  258850     2892. (((166.7932 -15.66881, 167.00~
## 4 Luxembourg Europe       2417.  556319    93655. (((6.043073 50.12805, 5.78241~
## 5 Northern ~ Asia         3786.      NA       NA  (((32.73178 35.14003, 32.9195~
## # ... with 2 more rows
```

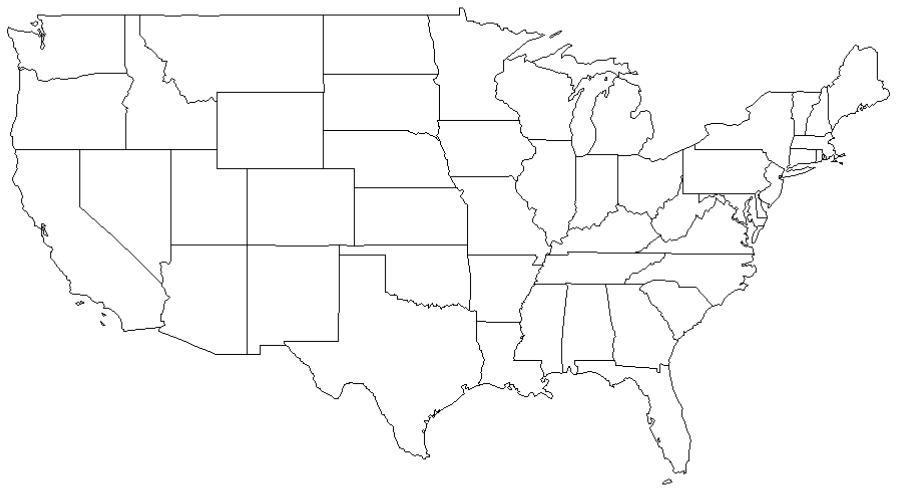
`mutate()` variables

```
world %>% mutate(pop_dens = (pop / area_km2))
```

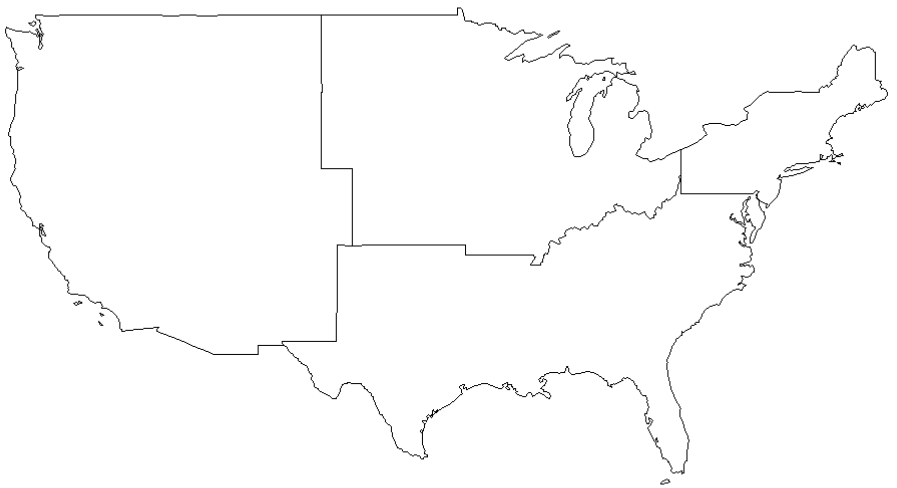
```
## Simple feature collection with 177 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 177 x 5
##   name_long   pop_dens      pop area_km2                                geom
##   <chr>          <dbl>    <dbl>    <dbl>                     <MULTIPOLYGON [°]>
## 1 Fiji            45.9    8.86e5   1.93e4 (((-180 -16.55522, -179.9174 -16.50178~
## 2 Tanzania        56.0    5.22e7   9.33e5 (((33.90371 -0.95, 31.86617 -1.02736, ~
## 3 Western Sa~     NA     NA        9.63e4 (((-8.66559 27.65643, -8.817828 27.656~
## 4 Canada           3.54   3.55e7   1.00e7 (((-132.71 54.04001, -133.18 54.16998,~
## 5 United Sta~     33.5    3.19e8   9.51e6 (((-171.7317 63.78252, -171.7911 63.40~
## # ... with 172 more rows
```

`summarize()` groups

```
states %>%
  # do something here
  # do another thing
  st_geometry() %>%
  plot()
```



```
states %>%
  group_by(REGION) %>%
  summarize(mean_area = mean(AREA)) %>%
  st_geometry() %>%
  plot()
```



`summarize()` groups

```
states %>%
  select(NAME, REGION, AREA, total_pop_15) %>%
  group_by(REGION) %>%
  summarize(pop = sum(total_pop_15),
            n_states = n()) %>%
  st_drop_geometry()
```

```
## # A tibble: 4 x 3
##   REGION         pop n_states
## * <fct>        <dbl>    <int>
## 1 Norteast  55989520        9
## 2 Midwest   67546398       12
## 3 South    118575377       17
## 4 West      72264052       11
```

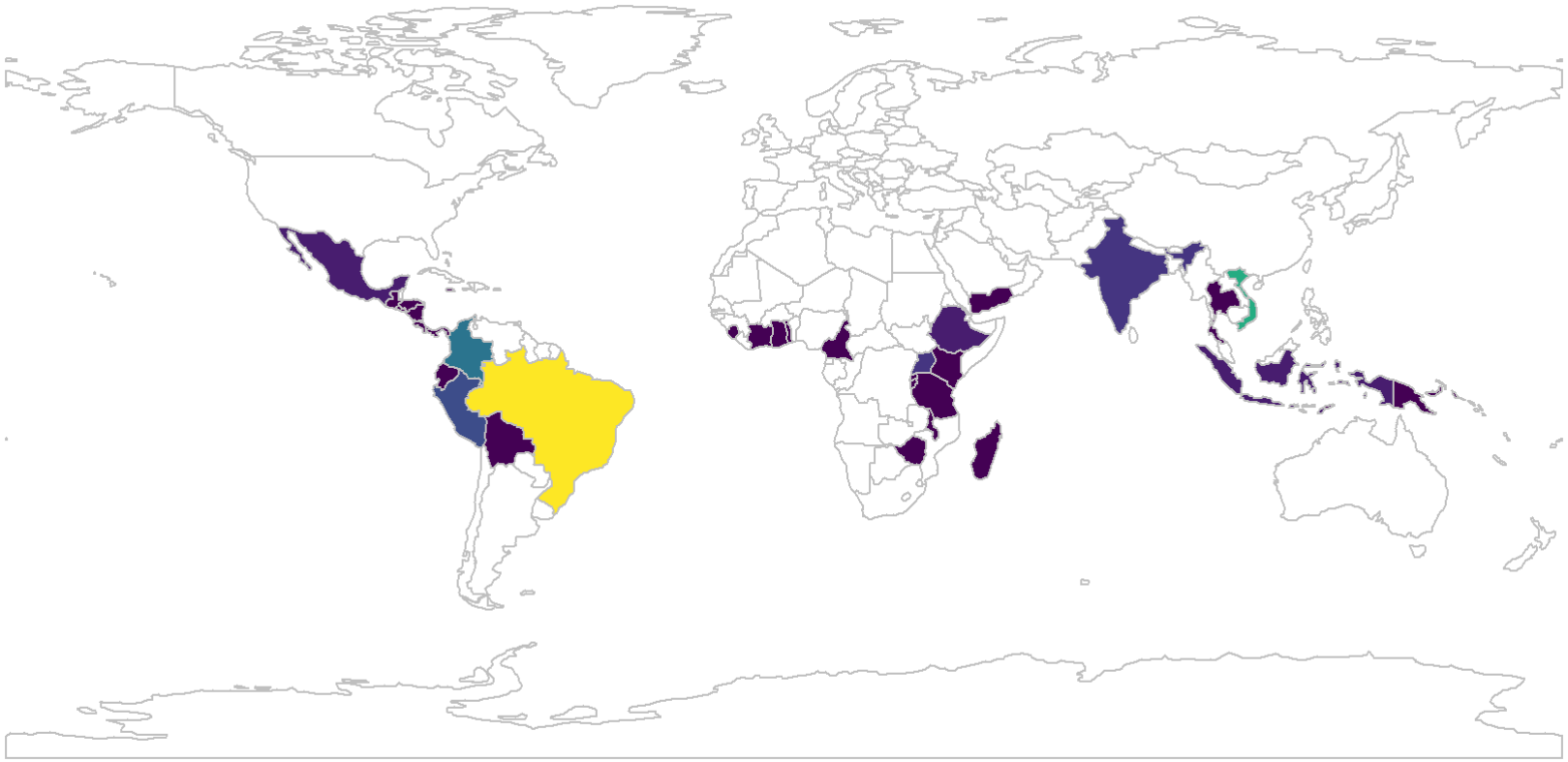
`left_join()` tables

```
world_coffee <- world %>% left_join(coffee_data)

world_coffee %>% select(name_long, coffee_production_2017)
```

```
## Simple feature collection with 177 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 177 x 3
##   name_long    coffee_production~                                    geom
##   <chr>                     <int>                            <MULTIPOLYGON [°]>
## 1 Fiji                         NA (((-180 -16.55522, -179.9174 -16.50178, -179.~
## 2 Tanzania                     66 (((33.90371 -0.95, 31.86617 -1.02736, 30.7698~
## 3 Western Sah~                 NA (((-8.66559 27.65643, -8.817828 27.65643, -8.~
## 4 Canada                       NA (((-132.71 54.04001, -133.18 54.16998, -133.2~
## 5 United Stat~                 NA (((-171.7317 63.78252, -171.7911 63.40585, -1~
## # ... with 172 more rows
```

`left_join()` tables

```
plot(world_coffee["coffee_production_2017"], pal = viridis)
```



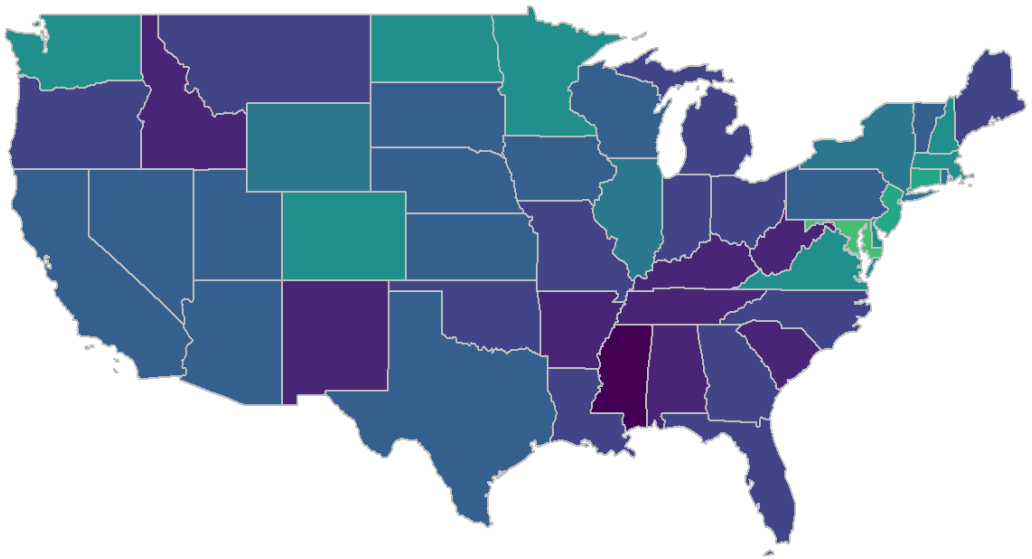
`left_join()` tables

```
state_data <- states %>% left_join(us_states_df, by = c("NAME" = "state"))

state_data %>% select(NAME, median_income_15)
```

```
## Simple feature collection with 49 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -124.7042 ymin: 24.55868 xmax: -66.9824 ymax: 49.38436
## Geodetic CRS:  NAD83
## First 5 features:
##          NAME median_income_15                       geometry
## 1     Alabama            22890 MULTIPOLYGON (((-88.20006 3...
## 2     Arizona            26156 MULTIPOLYGON (((-114.7196 3...
## 3    Colorado            30752 MULTIPOLYGON (((-109.0501 4...
## 4 Connecticut            33226 MULTIPOLYGON (((-73.48731 4...
## 5     Florida            24654 MULTIPOLYGON (((-81.81169 2...
```

`left_join()` tables

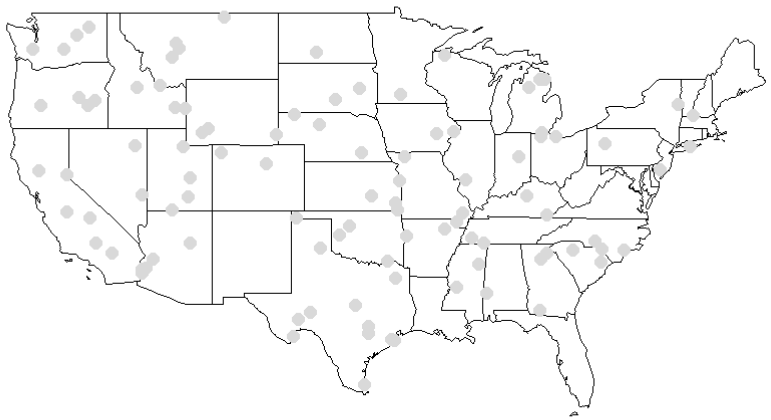
```
plot(state_data["median_income_15"], pal = viridis)
```

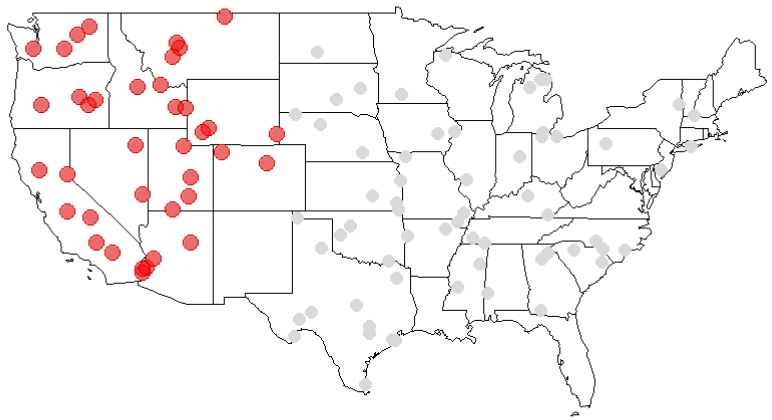


st_filter() geometries

```
# sample
rand_ohs <- states %>%
  st_sample(size = 100) %>%
  st_sf()
```

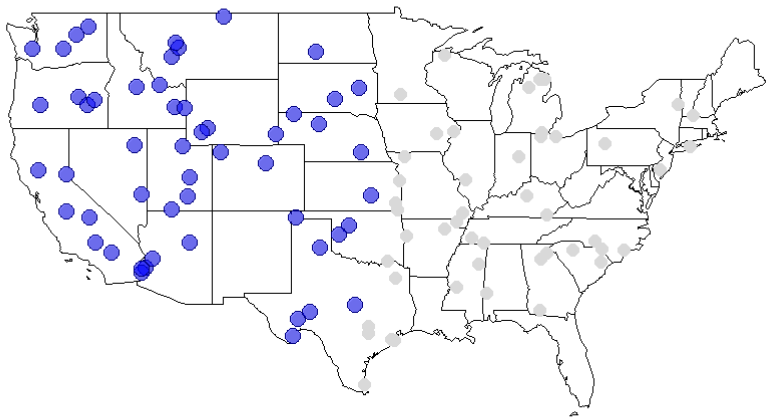
```
the_west <- states %>% filter(REGION == "West")
# spatial filter
westers <- rand_ohs %>%
  st_filter(the_west, .predicate = st_intersects)
```



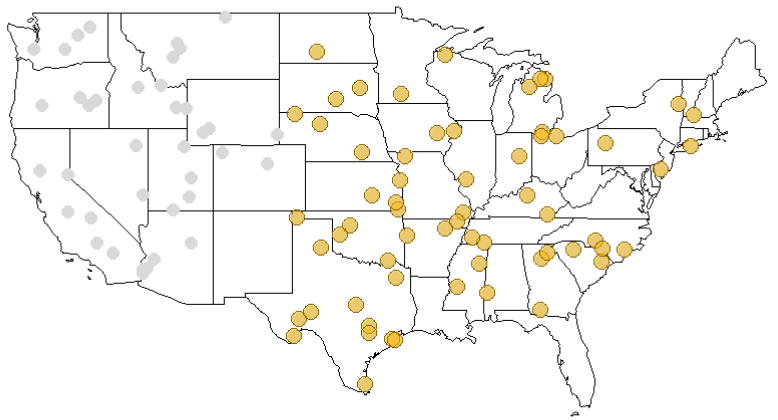


`st_filter()` geometries

```
goats <- rand_ohs %>%
  st_filter(the_west,
            .predicate = st_is_within_distance,
            dist = 500000)
```



```
easters <- rand_ohs %>%
  # made-up function
  st_filter(the_west,
            .predicate = st_not_intersects)
```



`st_join()` geometries

```
rand_ohs %>% st_join(states["NAME"])
```

```
## Simple feature collection with 100 features and 1 field
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -122.7003 ymin: 26.44266 xmax: -72.42611 ymax: 48.79554
## Geodetic CRS:  NAD83
## First 5 features:
##       NAME                   geometry
## 1 Michigan POINT (-83.97332 41.78423)
## 2 Colorado  POINT (-108.3684 40.5151)
## 3  Wyoming POINT (-109.3914 42.02838)
## 4 Michigan POINT (-83.74951 44.99639)
## 5     Iowa POINT (-90.70396 41.84292)
```

st_join() geometries

```
rand_ohs %>% st_join(states["NAME"],
                     join = st_is_within_distance,
                     dist = 1000000)
```

```
## Simple feature collection with 1810 features and 1 field
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -122.7003 ymin: 26.44266 xmax: -72.42611 ymax: 48.79554
## Geodetic CRS:  NAD83
## First 5 features:
##          NAME                    geometry
## 1     Alabama POINT (-83.97332 41.78423)
## 2 Connecticut POINT (-83.97332 41.78423)
## 3     Georgia POINT (-83.97332 41.78423)
## 4     Indiana POINT (-83.97332 41.78423)
## 5      Kansas POINT (-83.97332 41.78423)
```

`st_join()` geometries

```
rand_ohs %>% st_join(states[, "REGION"])
```

```
## Simple feature collection with 100 features and 1 field
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -122.7003 ymin: 26.44266 xmax: -72.42611 ymax: 48.79554
## Geodetic CRS:  NAD83
## First 5 features:
##    REGION                   geometry
## 1 Midwest POINT (-83.97332 41.78423)
## 2    West  POINT (-108.3684 40.5151)
## 3    West POINT (-109.3914 42.02838)
## 4 Midwest POINT (-83.74951 44.99639)
## 5 Midwest POINT (-90.70396 41.84292)
```

st_join() geometries

```
rand_ohs %>%
  st_join(states[, c("REGION", "total_pop_15")]) %>%
  group_by(REGION) %>%
  summarize(pop = sum(total_pop_15))
```

```
## Simple feature collection with 4 features and 2 fields
## Geometry type: MULTIPOINT
## Dimension:     XY
## Bounding box:  xmin: -122.7003 ymin: 26.44266 xmax: -72.42611 ymax: 48.79554
## Geodetic CRS:  NAD83
## # A tibble: 4 x 3
##   REGION        pop                                                geometry
##   <fct>       <dbl>                                          <MULTIPOINT [°]>
## 1 Norteast   6.24e7 ((-74.95319 39.49451), (-79.18072 41.06741), (-73.575 43.46~
## 2 Midwest    1.31e8 ((-83.74951 44.99639), (-84.0922 45.0179), (-91.38018 46.43~
## 3 South      4.18e8 ((-90.444 32.35973), (-95.41068 29.18533), (-95.19731 29.15~
## 4 West       3.47e8 ((-108.1109 48.79554), (-111.8395 47.19768), (-111.6086 46.~
```

`st_distance()` between sf

```
rand_ohs %>%
  sample_n(5) %>%
  st_distance()
```

```
## Units: [m]
##          [,1]      [,2]      [,3]      [,4]     [,5]
## [1,]      0.0  895327.5  614695.8  504480.2 774539.2
## [2,] 895327.5       0.0 1125035.8 1354605.0 944144.4
## [3,] 614695.8 1125035.8       0.0  525373.1 349190.3
## [4,] 504480.2 1354605.0  525373.1       0.0 855288.6
## [5,] 774539.2  944144.4  349190.3  855288.6      0.0
```