Spatial Data in R

Working with simple features

2021-08-19

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

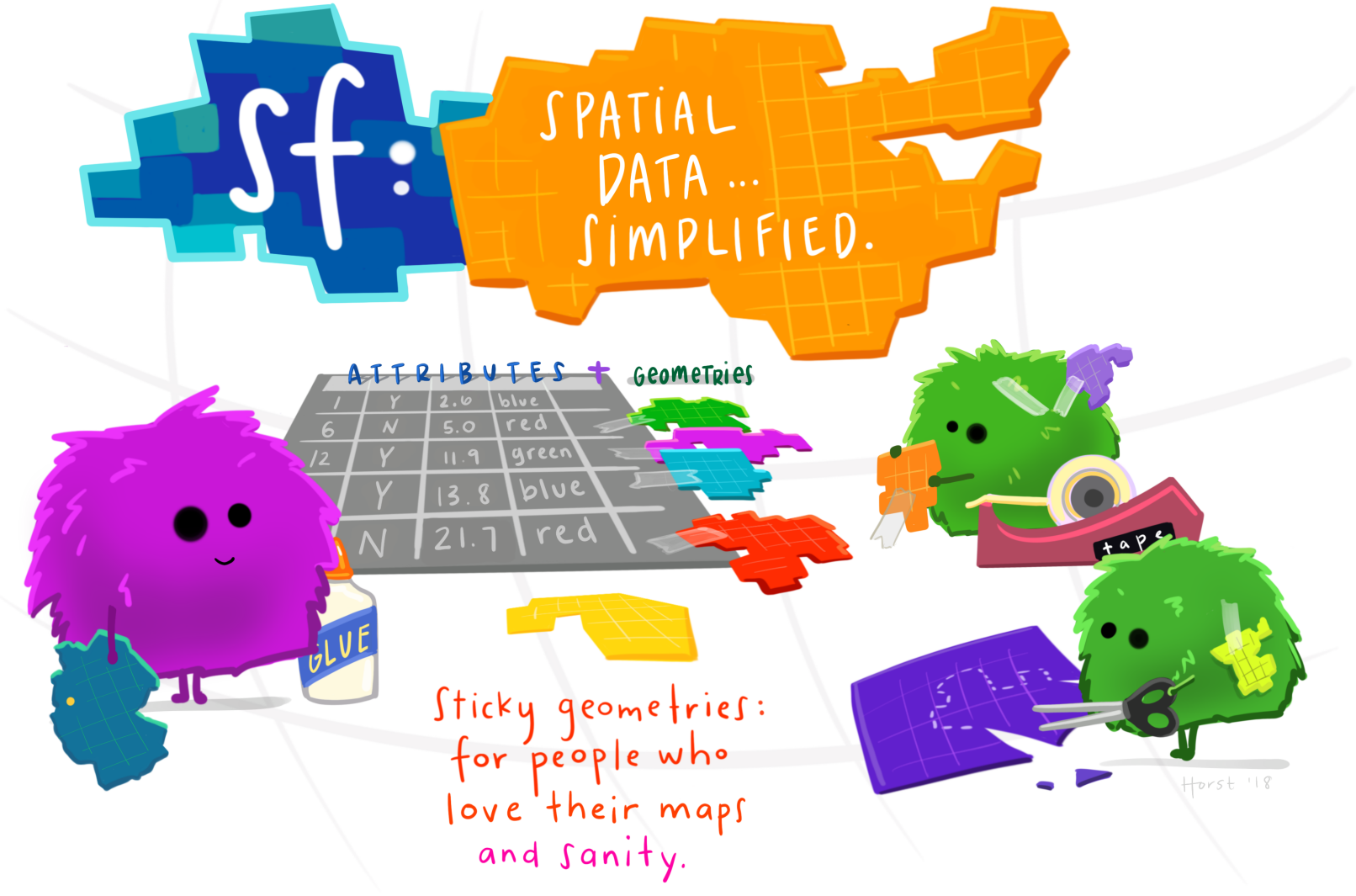
```
install.packages(c("dplyr", "here", "spData", "sf", "viridis"))

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

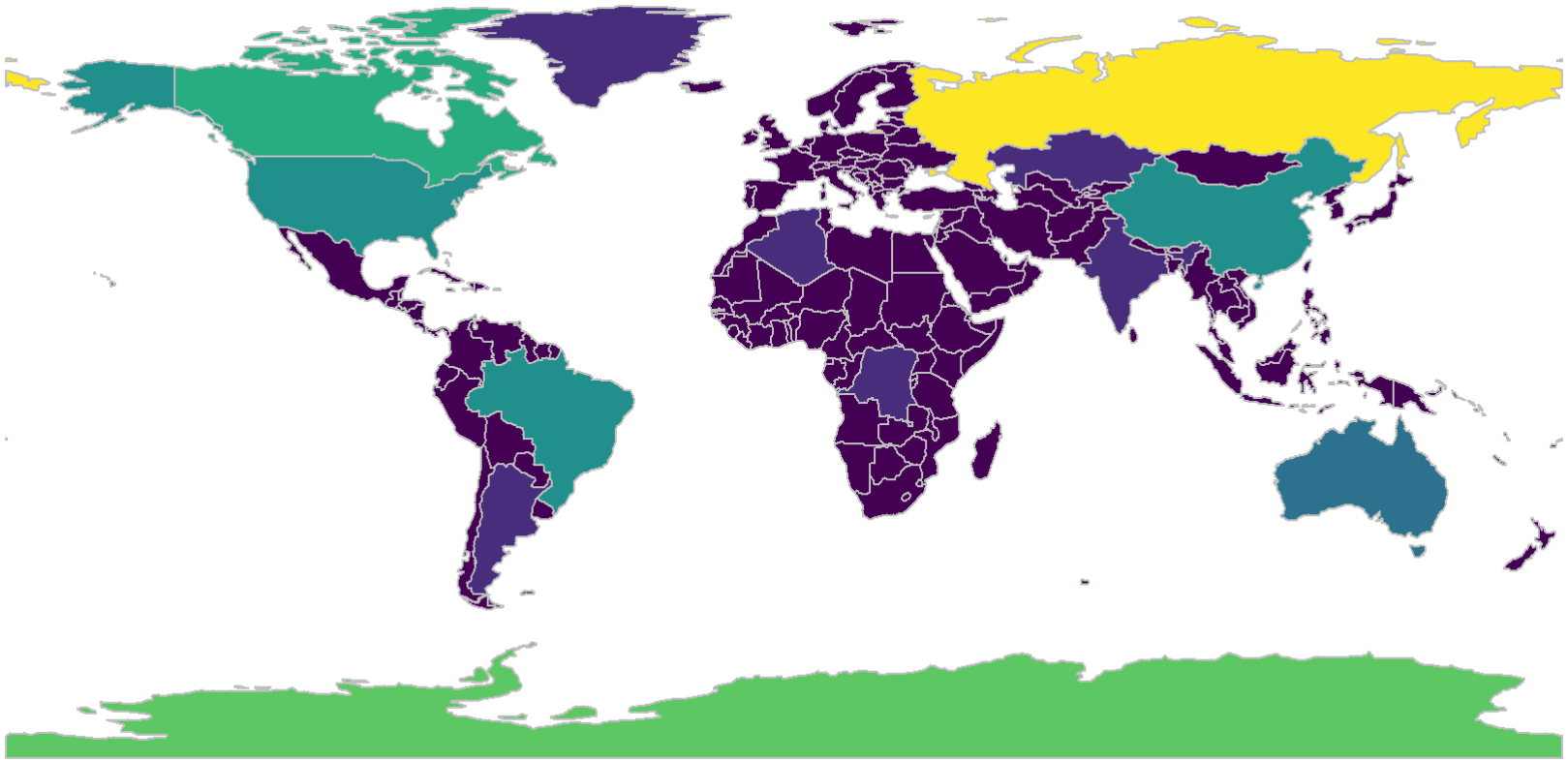
```
point1 <- st_point(c(432000, 4513100))           # create the feature geometry (sfg)
point2 <- st_point(c(436750, 4518500))

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)
```



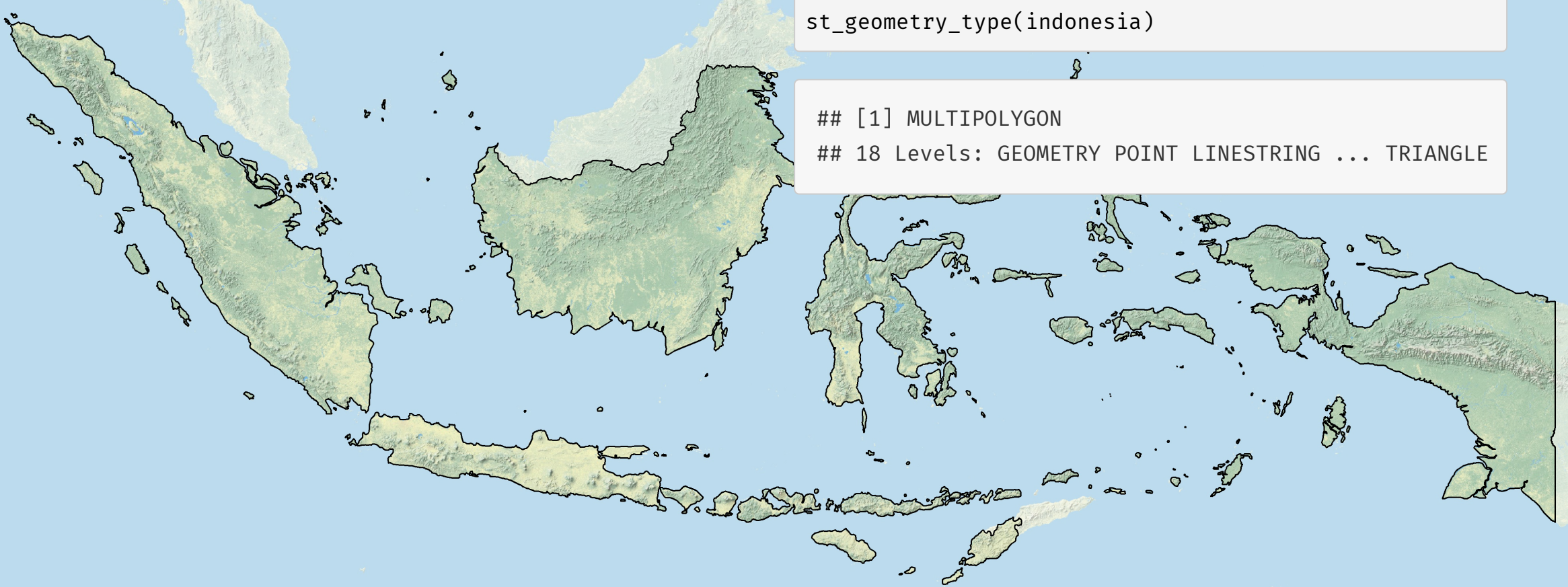
Plot Geometry

```
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
```



Read and write to file

Read sf

```
nc <- st_read(dsn = "data/north_carolina.shp",          # Data Source Name
              quiet = TRUE,                             # suppress Read messages
              stringsAsFactors = FALSE,                 # do not covert strings to factors
              as_tibble = TRUE)                         # load as tibble, not data.frame
```

Equivalent:

```
nc <- read_sf(dsn = "data/north_carolina.shp")
```

Write sf

```
st_write(nc,
         dsn = "data/north_carolina.shp",          # Data Source Name
         quiet = TRUE,                             # suppress write messages
         append = FALSE,                           # do not add to existing layer
         delete_layer = TRUE)                      # overwrite existing layer
```

Equivalent:

```
write_sf(nc, dsn = "data/north_carolina.shp")
```

Read and write to database

** A product of the OGC.*

```
st_layers("data/utah.gpkg")
```

```
## Driver: GPKG
## Available layers:
##       layer_name     geometry_type features fields
## 1           utah           Polygon        1      1
## 2 municipalities             Point      461      5
## 3          roads Multi Line String     3442      2
## 4  blm_monuments           Polygon       10      2
## 5      nps_parks     Multi Polygon       13      2
```

OGC GeoPackage

```
ut_parks <- read_sf(dsn = "data/utah.gpkg", layer = "nps_parks")
```

```
## Simple feature collection with 13 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -12604520 ymin: 4439191 xmax: -12138860 ymax: 5112397
## Projected CRS: WGS 84 / Pseudo-Mercator
## # A tibble: 13 x 3
##   name                              type                                  geom
##   <chr>                             <chr>                       <MULTIPOLYGON [m]>
## 1 Timpanogos Cave National Monument National Monument (((-12434775 4929486, -12~
## 2 Hovenweep National Monument       National Monument (((-12138862 4495150, -12~
## 3 Bryce Canyon National Park        National Park     (((-12482135 4520168, -12~
## 4 Canyonlands National Park         National Park     (((-12266249 4646106, -12~
## 5 Rainbow Bridge National Monument  National Monument (((-12352122 4449738, -12~
## # ... with 8 more rows
```

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
```

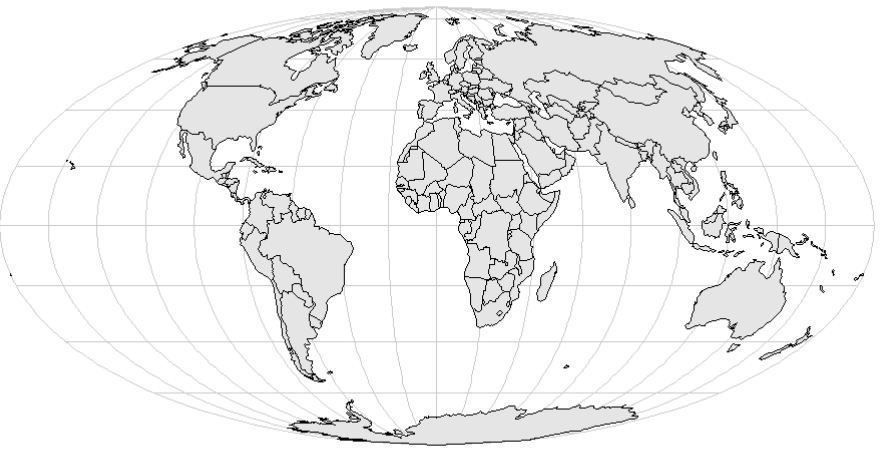
Re-projecting Simple Features

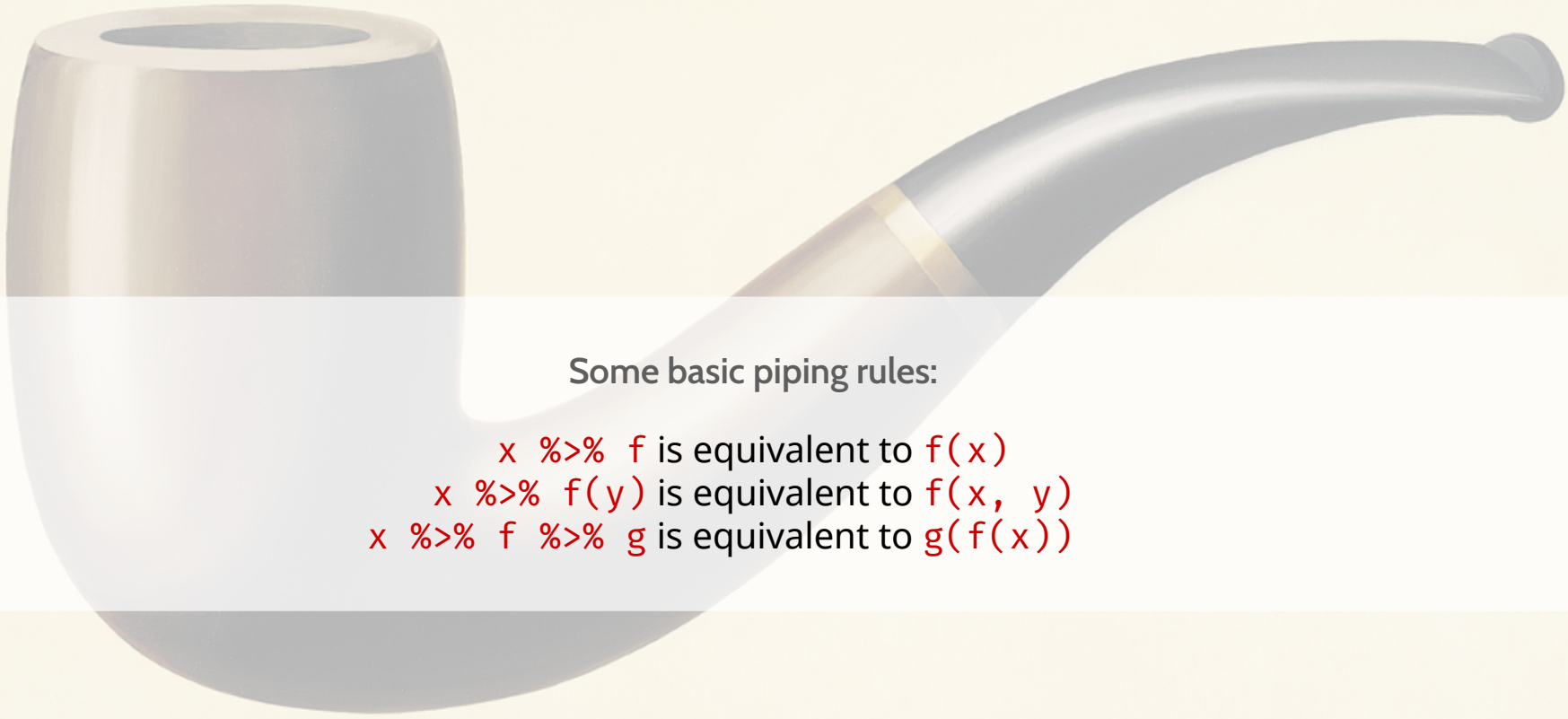
Google Pseudo-Mercator

```
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

We select just one column, but...

```
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
```

...the geometry comes along anyway. It's sticky!

`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 Rico     North America    9225. 3534874    35066. (((-66.28243 18.51476, -~
## 2 Palestine       Asia             5037. 4294682     4320. (((35.39756 31.48909, 35~
## 3 Vanuatu         Oceania          7490.  258850     2892. (((166.7932 -15.66881, 1~
## 4 Luxembourg      Europe           2417.  556319    93655. (((6.043073 50.12805, 5.~
## 5 Northern Cyprus Asia             3786.      NA       NA  (((32.73178 35.14003, 32~
## # ... with 2 more rows
```

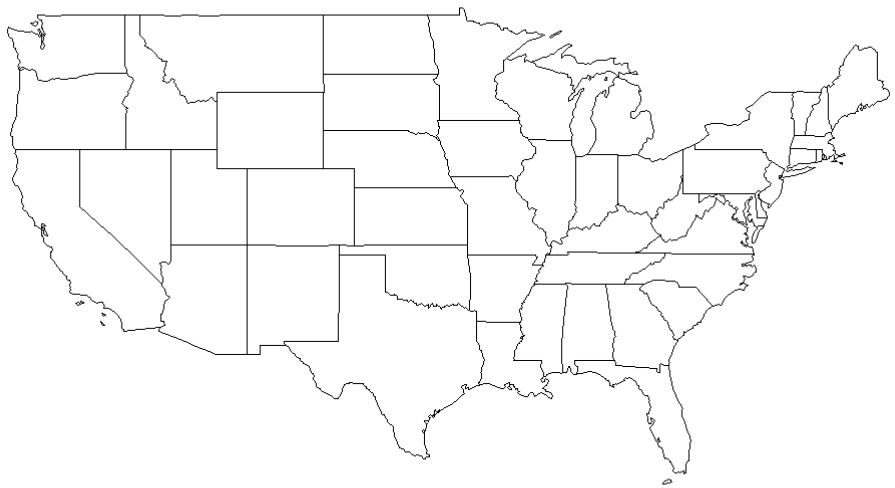
`mutate()` variables

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

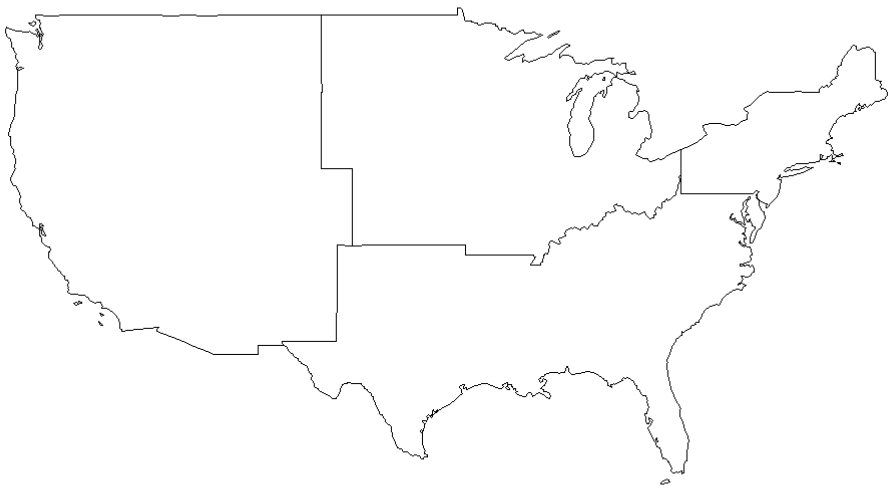
```
## Simple feature collection with 177 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 177 x 5
##   name_long      pop_dens       pop  area_km2                              geom
##   <chr>             <dbl>     <dbl>     <dbl>                <MULTIPOLYGON [°]>
## 1 Fiji               45.9     885806    19290. (((-180 -16.55522, -179.9174 -16.~
## 2 Tanzania           56.0   52234869   932746. (((33.90371 -0.95, 31.86617 -1.02~
## 3 Western Sahara     NA           NA    96271. (((-8.66559 27.65643, -8.817828 2~
## 4 Canada              3.54  35535348 10036043. (((-132.71 54.04001, -133.18 54.1~
## 5 United States      33.5  318622525  9510744. (((-171.7317 63.78252, -171.7911 ~
## # ... 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
```

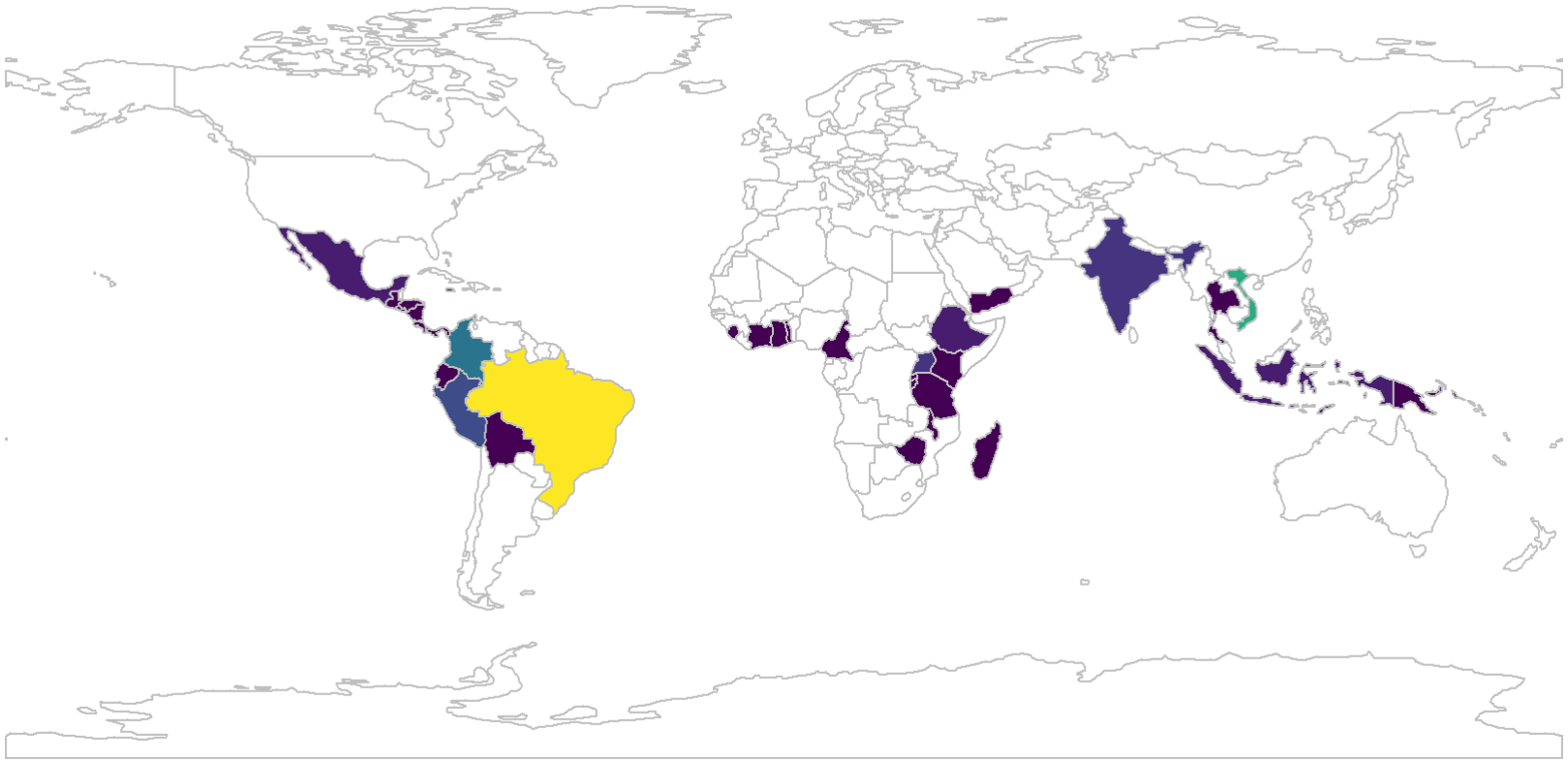
`left_join()` tables

```
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_2017                                geom
##   <chr>                           <int>                      <MULTIPOLYGON [°]>
## 1 Fiji                               NA (((-180 -16.55522, -179.9174 -16.50178,~
## 2 Tanzania                           66 (((33.90371 -0.95, 31.86617 -1.02736, 3~
## 3 Western Sahara                     NA (((-8.66559 27.65643, -8.817828 27.6564~
## 4 Canada                             NA (((-132.71 54.04001, -133.18 54.16998, ~
## 5 United States                      NA (((-171.7317 63.78252, -171.7911 63.405~
## # ... with 172 more rows
```

`left_join()` tables

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



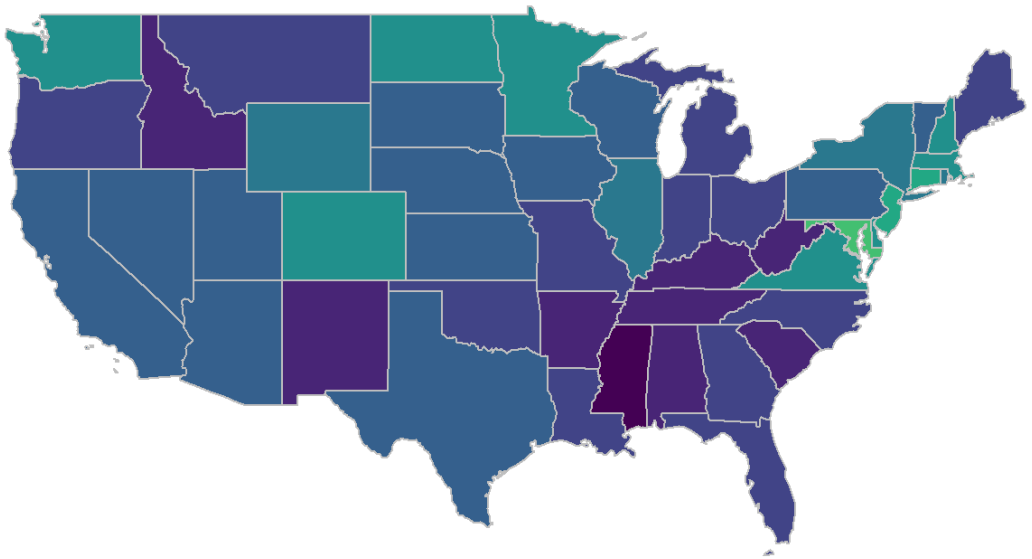
`left_join()` tables

```
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

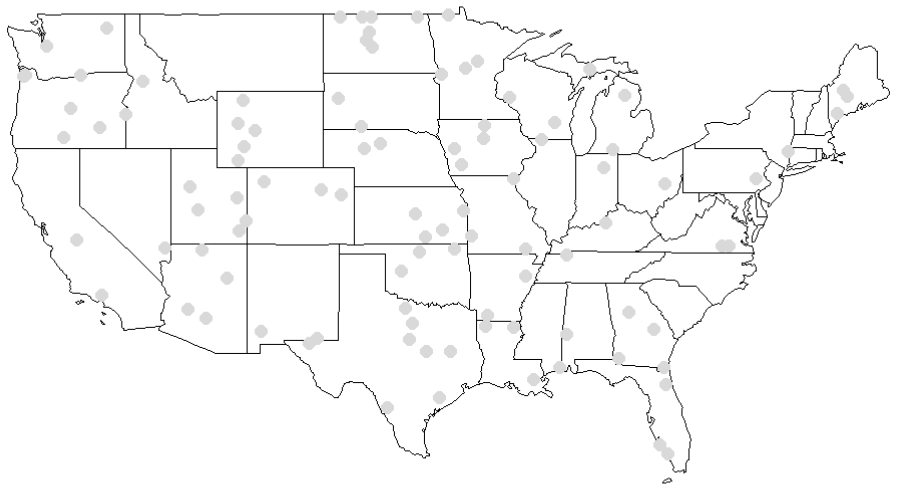
```
plot(state_data["median_income_15"], pal = viridis)
```

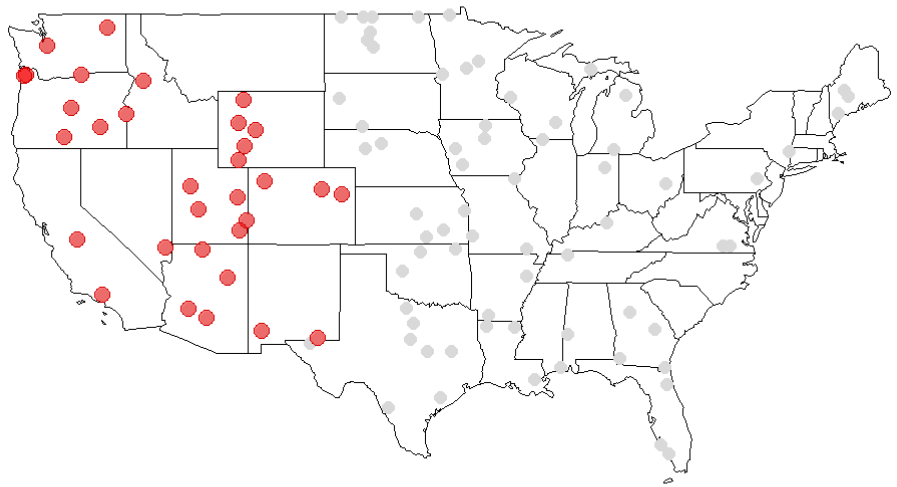


st_filter() geometries

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

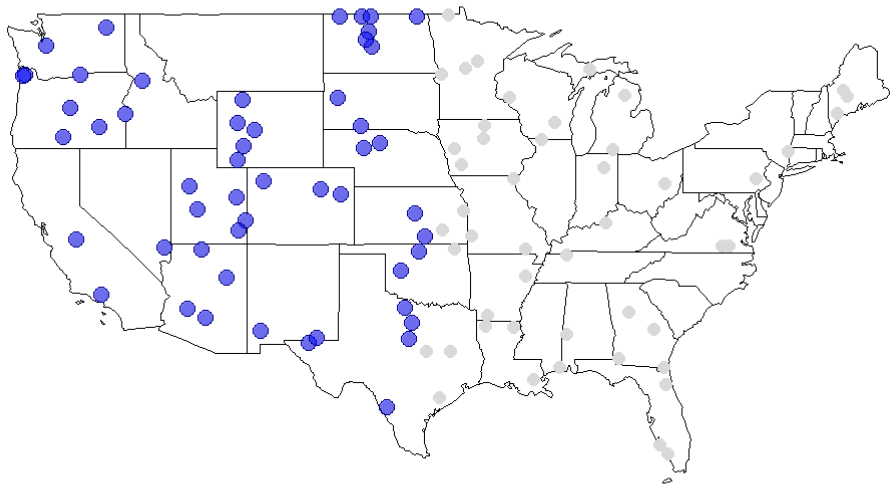
```
the_west <- states %>% filter(REGION == "West")
# spatial filter
westers <- rand_ohs %>%
  st_filter(the_west, .predicate = st_intersects)
```



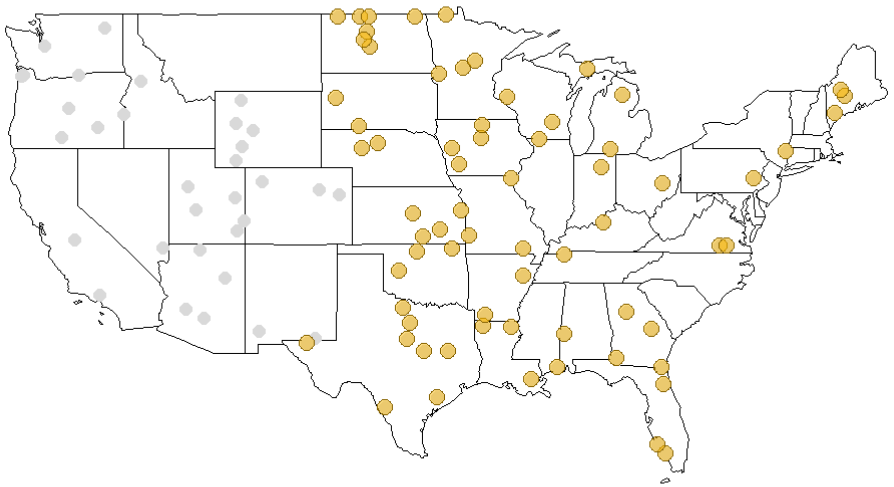


`st_filter()` geometries

```
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: -123.662 ymin: 26.07254 xmax: -69.76945 ymax: 48.97653
## Geodetic CRS:  NAD83
## First 5 features:
##         NAME                    geometry
## 1       Iowa  POINT (-93.55588 42.5097)
## 2    Arizona POINT (-112.9006 33.63792)
## 3   Arkansas POINT (-90.84833 35.36545)
## 4 Washington   POINT (-122.16 47.36634)
## 5  Wisconsin POINT (-88.93779 43.39093)
```

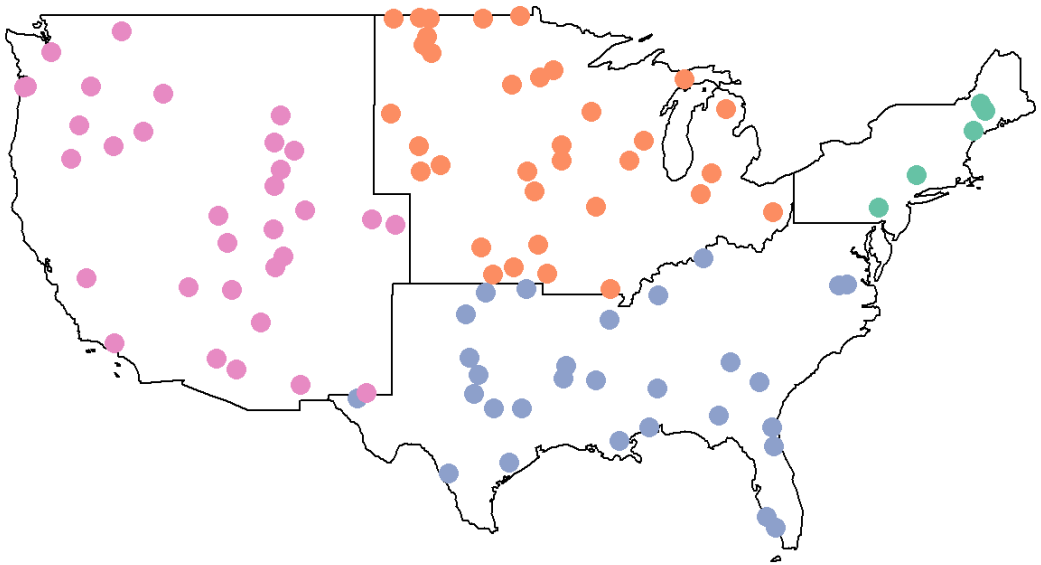
st_join() geometries

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

```
## Simple feature collection with 1731 features and 1 field
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -123.662 ymin: 26.07254 xmax: -69.76945 ymax: 48.97653
## Geodetic CRS:  NAD83
## First 5 features:
##        NAME                   geometry
## 1   Alabama POINT (-93.55588 42.5097)
## 2  Colorado POINT (-93.55588 42.5097)
## 3   Indiana POINT (-93.55588 42.5097)
## 4    Kansas POINT (-93.55588 42.5097)
## 5 Minnesota POINT (-93.55588 42.5097)
```

st_join() geometries

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



`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: -123.662 ymin: 26.07254 xmax: -69.76945 ymax: 48.97653
## Geodetic CRS:  NAD83
## # A tibble: 4 x 3
##   REGION         pop                                                geometry
##   <fct>        <dbl>                                          <MULTIPOINT [°]>
## 1 Norteast  36440033 ((-70.02224 45.0453), (-69.76945 44.74112), (-70.43562 43.~
## 2 Midwest  146440000 ((-86.62622 46.13904), (-94.7542 46.20103), (-93.98669 46.~
## 3 South    387482382 ((-90.28253 29.93282), (-91.59847 32.65944), (-93.4401 32.~
## 4 West     184468938 ((-109.2935 44.50244), (-109.674 43.32569), (-122.16 47.36~
```

`st_distance()` between sf

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

```
## Units: [m]
##         [,1]    [,2]      [,3]      [,4]    [,5]
## [1,]       0 1396320 1257282.4 1470660.7 3326083
## [2,] 1396320       0 2507890.7 2540962.1 3972001
## [3,] 1257282 2507891       0.0  449435.8 2374672
## [4,] 1470661 2540962  449435.8       0.0 1960816
## [5,] 3326083 3972001 2374672.2 1960816.5       0
```