RStudio for Practical Geospatial Analysis

Roger Andre, Solutions Engineer, Posit PBC

Practical Geospatial Analysis…

‘Practical’ as in, *“We need to know…”*

- *How many people live in a service area?*
- *How far apart are 2 businesses from one another?*
- *What are the geographic coordinates for an address?*
- *How many businesses are within N miles of an address?*

“…by tomorrow!”

R widely used in Data Science but…

…using R/RStudio for spatial analysis still isn't very common.

- Spatial Data Analysis (GIS) in R is still considered "niche" by some people
- Older spatial packages (like `sp`) implemented very R-centric workflows
- There are PLENTY of other, good GIS tools available

My goal here is to convince you to try it!

What is R?

> *"R is a free software environment for statistical computing and graphics."*

- An interpreted language that is similar in some ways to Python
- Multipurpose, but highly optimized for working with structured data
- Pandas library in Python is based on core functionality in R

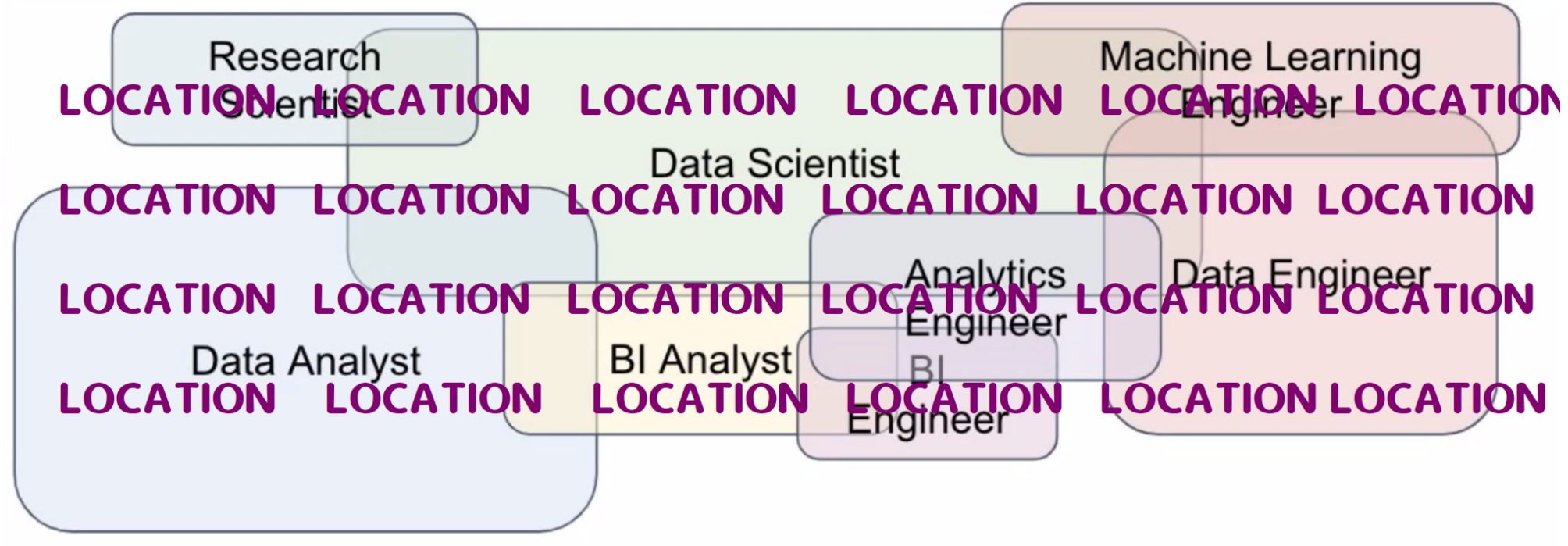
What is RStudio?

- Open Source IDE with tools for working in R (and Python)
- Available in free (as in speech) and commercial editions
- Desktop app. on many OS and server application on Linux

`RStudio` was also formerly the name of the company that funds work on the IDE (and other OS projects).

Why should you use R/RStudio for Geospatial Data Analysis?

Because you're probably already using RStudio for ~~Geospatial~~ Data Analysis!



And since you're already using RStudio…

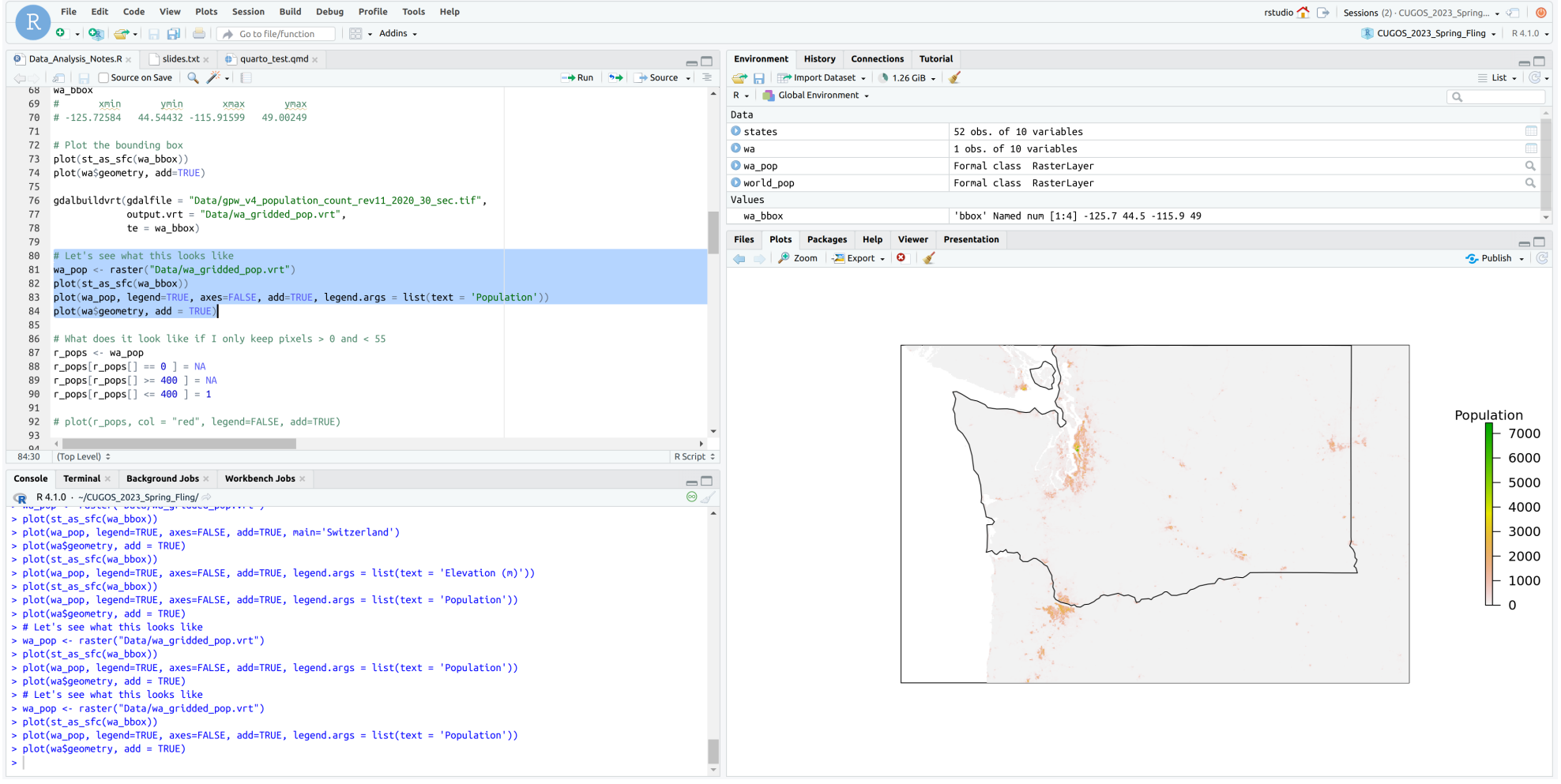
RStudio has built-in features that are well suited to GIS!

Native graphic support for vector “maps”

Charts, plots, images…

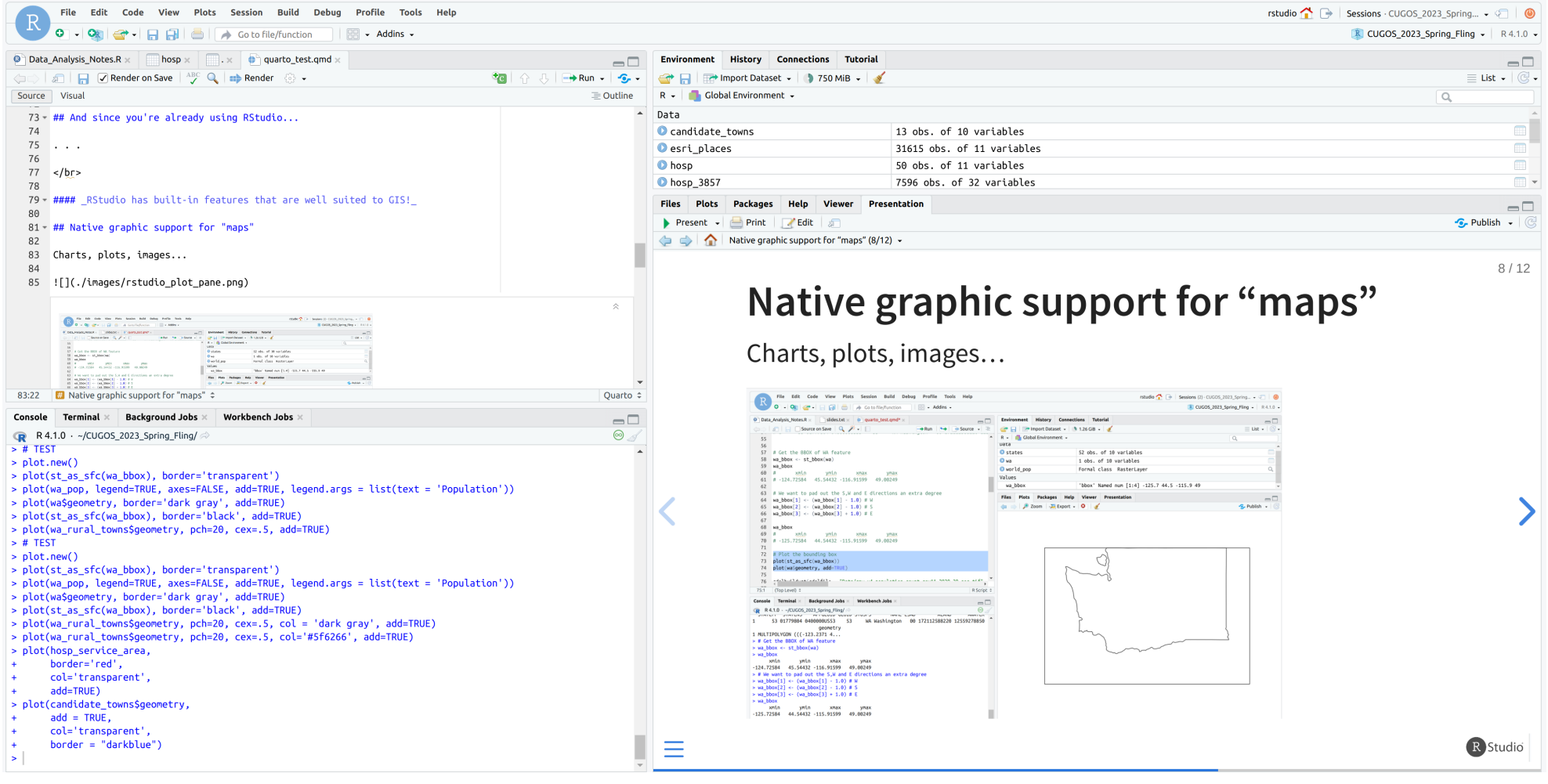
...and raster plots

Gridded Population



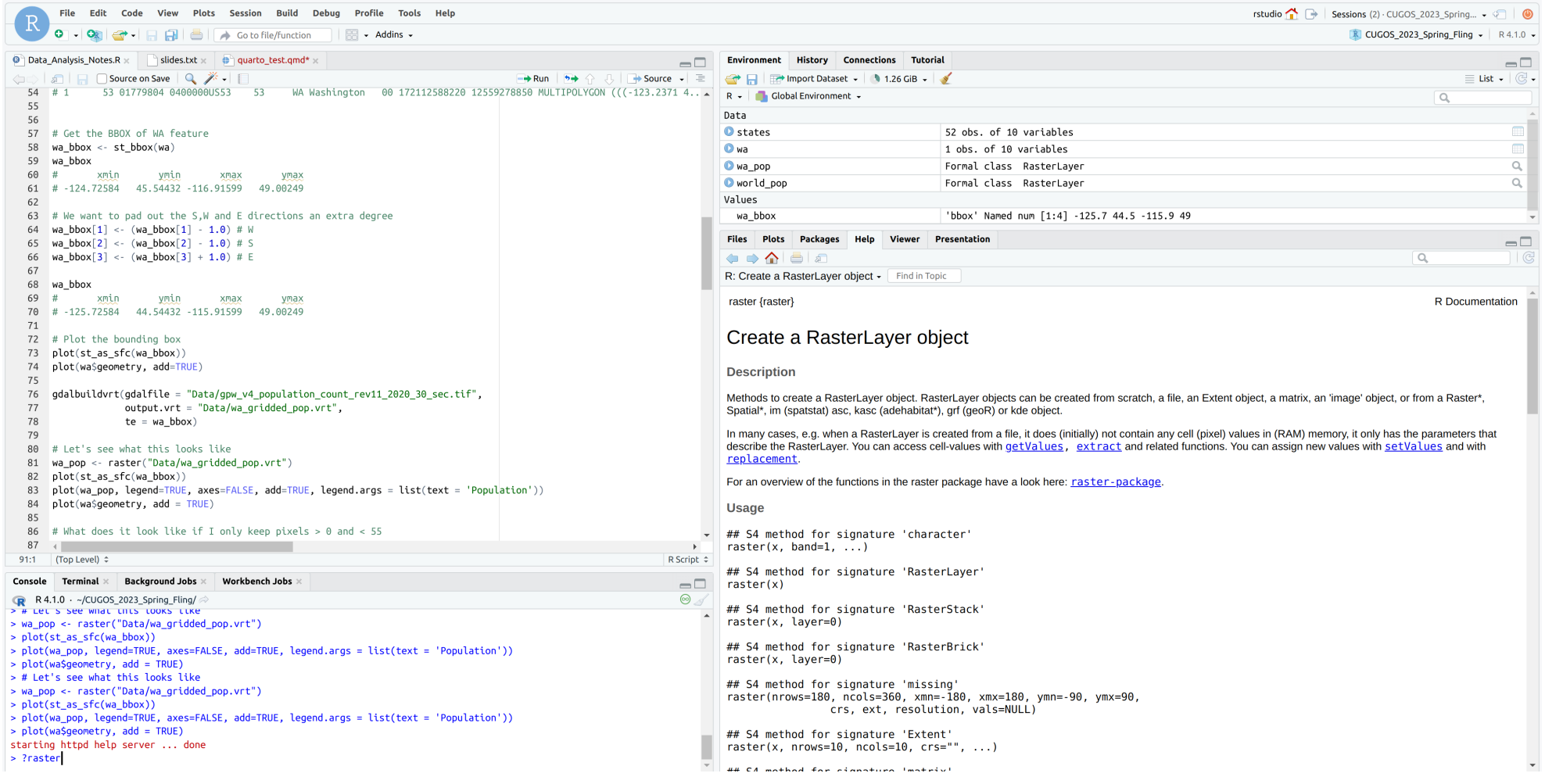
…and other things

(my slides under development)



Built-in Help doc viewer

?raster



R itself has good geospatial support

- `raster` (which is now superseded by `terra`)

> The raster package provides classes and functions to manipulate geographic (spatial) data in 'raster' format.

- `sf`

> Support for simple features, a standardized way to encode spatial vector data. Binds to 'GDAL' for reading and writing data, to 'GEOS' for geometrical operations, and to 'PROJ' for projection conversions and datum transformations.

Basic shapefile operations

- Open and read a shapefile

```
library(sf)
states <- st_read("Data/cb_2018_us_state_20m.shp")

Reading layer `cb_2018_us_state_20m' from data source
  `/home/rstudio/CUGOS_2023_Spring_Fling/Data/cb_2018_us_state_20m.shp' usi
Simple feature collection with 52 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -179.1743 ymin: 17.91377 xmax: 179.7739 ymax: 71.35256
Geodetic CRS:  4269
```

Basic shapefile operations (cont.)

- Reproject to new CRS

```
states <- st_transform(states, crs = 4326)
states

Simple feature collection with 52 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -179.1743 ymin: 17.91377 xmax: 179.7739 ymax: 71.35256
Geodetic CRS:  EPSG:4326
```

Basic shapefile operations (cont.)

- Buffer around a set of points

```
wa_town_albers

Simple feature collection with 124 features and 9 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -2134539 ymin: 2773350 xmax: -1566342 ymax: 3155027
Projected CRS: EPSG:5070

# 30 miles approx 48,300 meters
wa_town_albers_buffer <- st_buffer(wa_town_albers, 48300)

wa_town_albers_buffer
Simple feature collection with 124 features and 9 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: -2182839 ymin: 2725050 xmax: -1518042 ymax: 3203327
Projected CRS: EPSG:5070
```

Basic raster operations

- Open a geotiff and display its extents

```
library(raster)
world_pop <- raster("Data/gpw_v4_population_count_rev11_2020_30_sec.tif")
world_pop

class      : RasterLayer
dimensions : 21600, 43200, 933120000  (nrow, ncol, ncell)
resolution : 0.008333333, 0.008333333  (x, y)
extent     : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
crs        : +proj=longlat +datum=WGS84 +no_defs
source     : gpw_v4_population_count_rev11_2020_30_sec.tif
names      : gpw_v4_population_count_rev11_2020_30_sec
```

Basic raster operations (cont.)

- Create a VRT from larger raster using bbox as extents

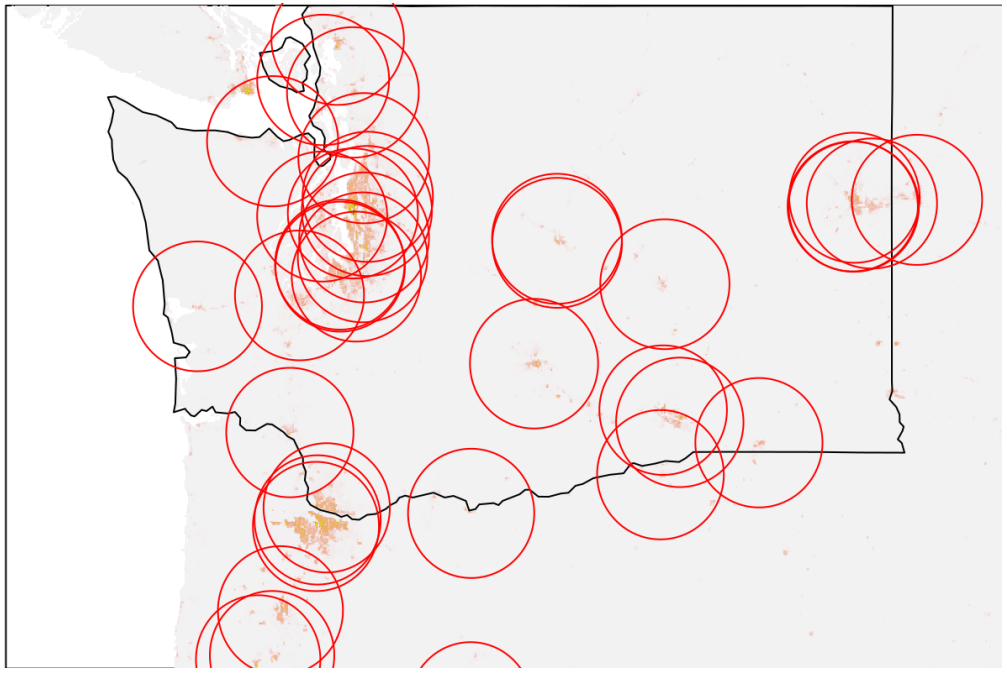
```
library(gdalUtilities)
gdalbuildvrt(gdalfile = "Data/gpw_v4_population_count_rev11_2020_30_sec.tif
             output.vrt = "Data/wa_gridded_pop.vrt",
             te = wa_bbox)
```

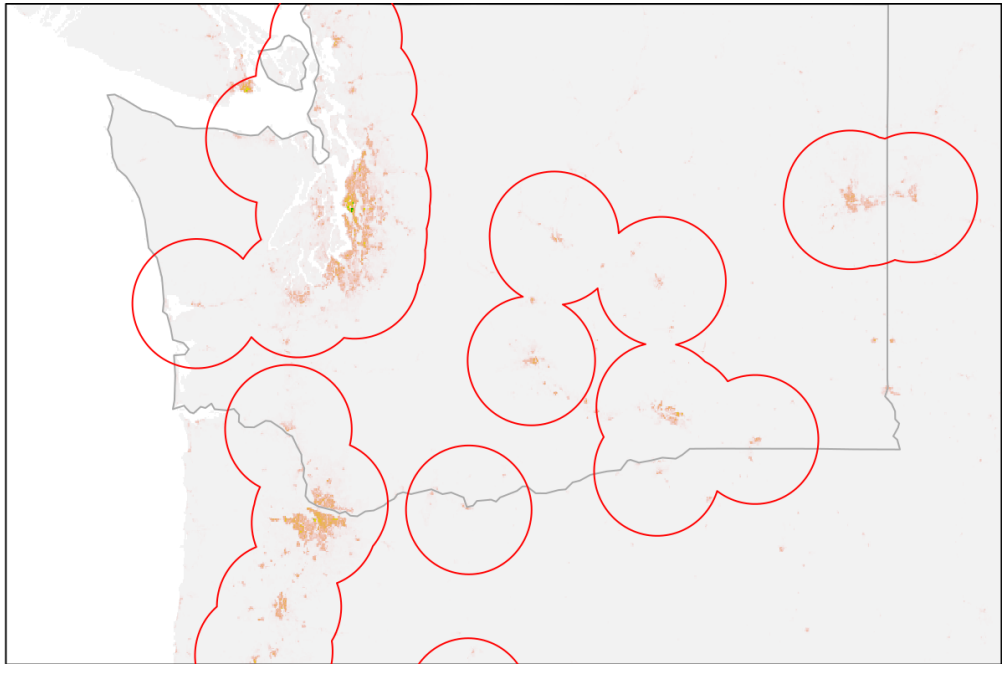
- Load the VRT as a raster

```
wa_pop <- raster("Data/wa_gridded_pop.vrt")

setMinMax(wa_pop)
class       : RasterLayer
dimensions  : 535, 1177, 629695  (nrow, ncol, ncell)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -125.7258, -115.9175, 44.54416, 49.00249  (xmin, xmax, ymin, y
crs         : +proj=longlat +datum=WGS84 +no_defs
source      : wa_gridded_pop.vrt
names       : wa_gridded_pop
values      : 0, 7647.292  (min, max)
```

Dissolve operation

```
# Add 'type' field to dissolve on
hosp_buffer$type <- "combined_service_areas"

hosp_service_area <- hosp_buffer %>%
  group_by(type) %>%
  summarise()
```





A practical example

> Assume that you work for an agency that is reponsible for helping to expand access to healthcare in rural America. You have been tasked to come up with a list of 3 rural cities, or towns, in the State of Washington that would most benefit from the construction of a new hospital.

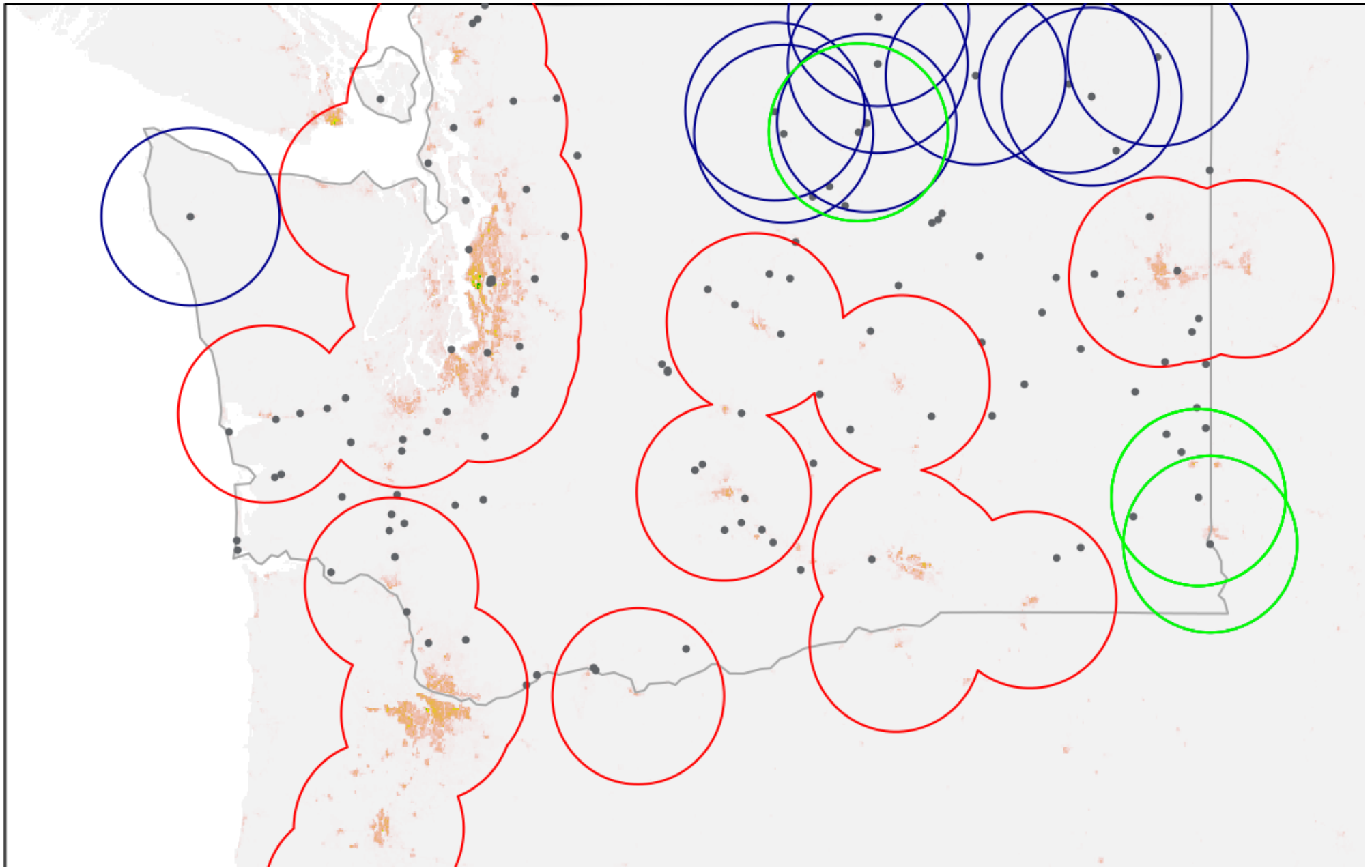
Provide the names for the top 3 cities or towns and the estimated population that would be covered by a 30 mile service area around each of them.

Combining rasters and vectors

Where this gets fun!

```
st_as_sf(raster::extract(wa_pop,
  candidate_towns,
  fun=sum,
  na.rm=TRUE,
  sp=TRUE)) -> candidate_town_pops

st_drop_geometry(candidate_town_pops) %>%
  select(NAME, wa_gridded_pop) %>%
  arrange(desc(wa_gridded_pop)) %>%
  slice(1:3)

       NAME wa_gridded_pop
1    Colton      144817.03
2    Asotin      133227.72
3 Okanogan       36177.21
```

Final results - Top 3 candidates



Additional inspiration

- Shiny apps that showcase interactivity with maps

https://fitzlab.shinyapps.io/cityapp/

https://alexh5.shinyapps.io/INFO201FinalProject/

https://vac-lshtm.shinyapps.io/ncov_tracker/