

Fundamentals of Spatial Analysis in R

Marc Weber

2020-02-25

Contents

1	Introduction	5
1.1	Workshop agenda	7
1.2	Spatial Data in R	7
1.3	Code along	7
1.4	Challenge: Does this work?	9
2	Vector data with sf	11
3	Raster data	13
4	Applications	15
4.1	Example one	15
4.2	Example two	15
5	Final Words	17
	References	19

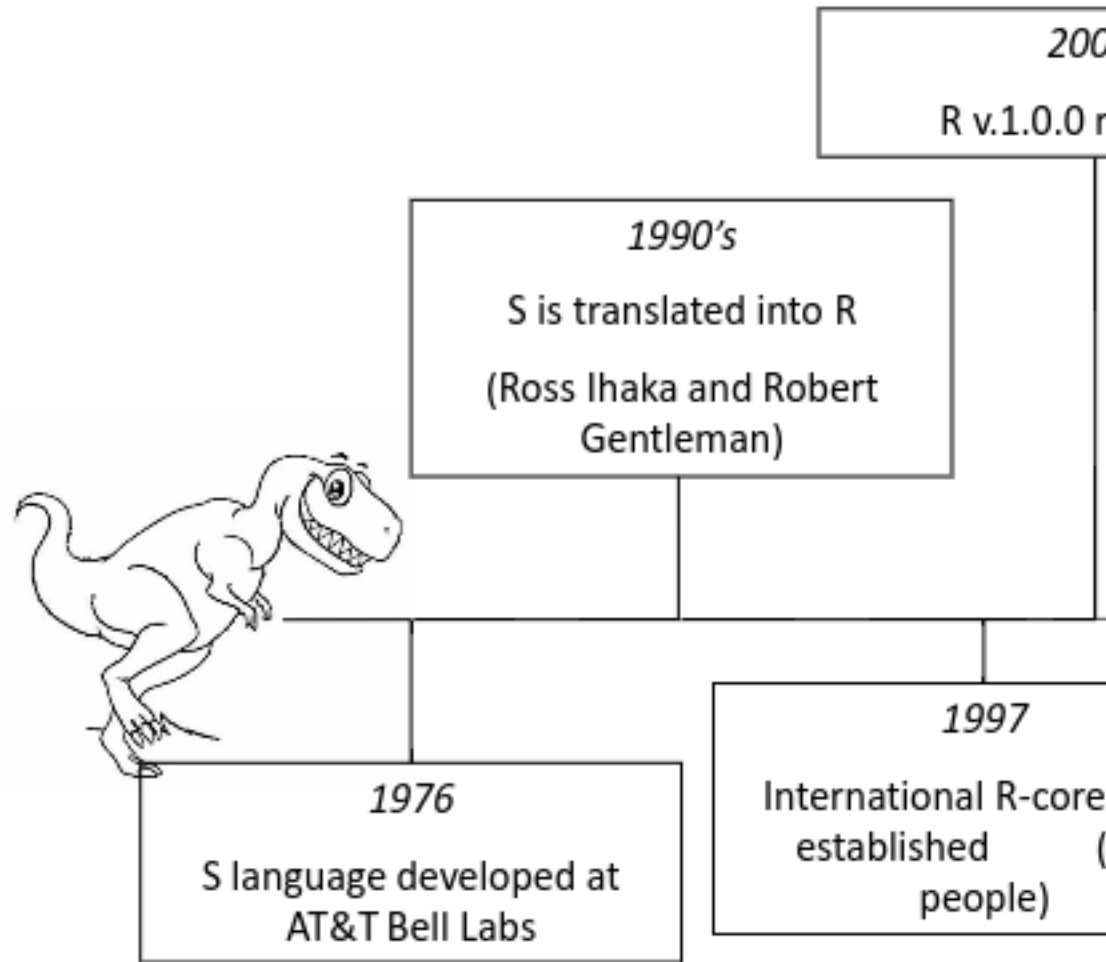
Chapter 1

Introduction

- A bit about me
- Intros
- What is everyone's experience level?
- What are you expected to know?
 - Basic R objects and methods
 - **tidyverse** packages and syntax
 - * i.e. **ggplot2**, **dplyr**, **readr**, **tidyr**, the pipe operator `%>%`
 - If these are new, don't sweat it - google them, we'll talk through them as we go if needed
- This portion of workshop expectation no expectation of experience with spatial in R - we'll cover all the basics

What is R and why should we use R for spatial analysis? Let's break that into two questions - first, what is R and why should we use it?

- A language and environment for statistical computing and graphics
- R is lightweight, free, open-source and cross-platform
- Works with contributed packages - currently 12,938 -extensibility
- Automation and recording of workflow (reproducibility)
- Optimized work flow - data manipulation, analysis and visualization all in one place
- R does not alter underlying data - manipulation and visualization in memory
- R is great for repetitive graphics



of R.bb

Figure 1.1: History of R

1.1 Workshop agenda

1.2 Spatial Data in R

1.3 Code along

Just a sampling of what we'll cover. Run code, examine output, ask any questions - we'll explore it all in more detail through the morning.

1.3.1 geocoding example with tmaptools using open street map

```
# uses OSM
library(tmap)
library(tmaptools)
library(dplyr)
tex_cap <- tmaptools::geocode_OSM("Texas Capital",
  as.sf = TRUE) %>%
  glimpse()

## Observations: 1
## Variables: 8
## $ query      <chr> "Texas Capital"
## $ lat        <dbl> -31.46748
## $ lon        <dbl> -64.22844
## $ lat_min    <dbl> -31.46748
## $ lat_max    <dbl> -31.46748
## $ lon_min    <dbl> -64.22995
## $ lon_max    <dbl> -64.22723
## $ geometry   <POINT [°]> POINT (-64.22844 -31.46748)
```

1.3.2 interactive mapping

```
library(mapview)
mapview(tex_cap)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed,
```

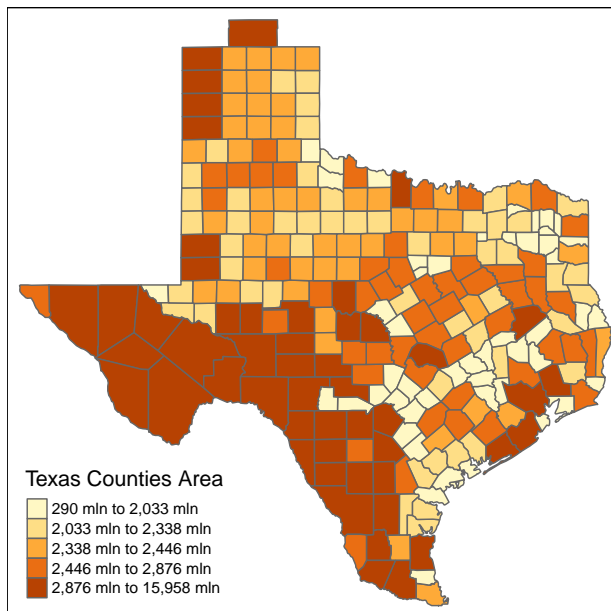
1.3.3 Choropleth map

The package maps (automatically installed and loaded with ggplot2) provides maps of the USA, with state and county borders, that can be retrieved and converted as sf objects:

```
library(sf)
library(maps)
counties <- st_as_sf(map("county", plot = FALSE, fill = TRUE))
counties <- subset(counties, grepl("texas", counties$ID) & !grepl('missouri,texas', counties$ID))
counties$area <- as.numeric(st_area(counties))
head(counties)
```

```
## Simple feature collection with 6 features and 2 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -103.0751 ymin: 28.14942 xmax: -94.13123 ymax: 37.00161
## epsg (SRID):    4326
## proj4string:     +proj=longlat +datum=WGS84 +no_defs
##               geometry              ID          area
## 2168 MULTIPOLYGON (((-101.6255 3... oklahoma,texas 5434507068
## 2491 MULTIPOLYGON (((-95.75271 3... texas,anderson 2817584981
## 2492 MULTIPOLYGON (((-102.2042 3... texas,andrews 3962852909
## 2493 MULTIPOLYGON (((-94.13123 3... texas,angelina 2200352194
## 2494 MULTIPOLYGON (((-96.80122 2... texas,aransas 290370313
## 2495 MULTIPOLYGON (((-98.42269 3... texas,archer 2422607253

tm_shape(counties) +
  tm_polygons("area",
    style="quantile",
    title="Texas Counties Area")
```



example-1.bb

1.4 Challenge: Does this work?

Did my .css styling adjustment work?

1.4.1 Answer

1. Yes
2. No

Chapter 2

Vector data with sf

Load `tidycensus` - you'll need to set your Census API key. A key can be obtained from [here](#).

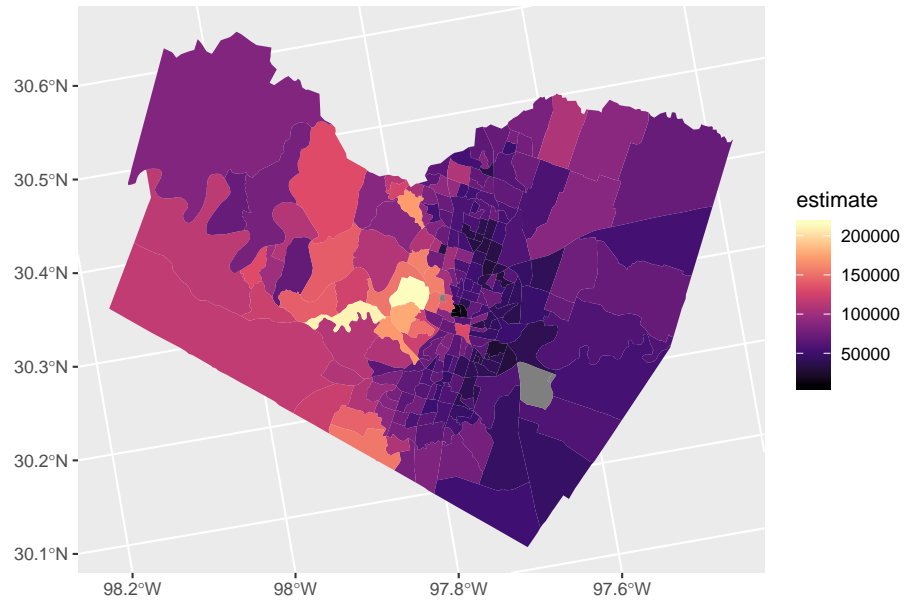
```
library(tidycensus)
library(tidyverse)

census_api_key("YOUR API KEY GOES HERE")

library(tidycensus)
library(ggplot2)
options(tigris_use_cache = TRUE)
austin_tracts <- get_acs(state = 'TX', county = 'Travis', geography = "tract",
                        variables = "B19013_001", geometry = TRUE)
```

Getting data from the 2013-2017 5-year ACS

```
austin_tracts %>%
  ggplot(aes(fill = estimate)) +
  geom_sf(color = NA) +
  coord_sf(crs = 26911) +
  scale_fill_viridis_c(option = "magma")
```



Chapter 3

Raster data

Chapter 4

Applications

Some *significant* applications are demonstrated in this chapter.

4.1 Example one

4.2 Example two

Chapter 5

Final Words

We have finished a nice book.

References

5.0.1 R Spatial Resources

- R Spatial - **Spatial Data Science with R**
- **Geocomputation with R**
- **R Spatial Task View**
- **Modern Geospatial Data Analysis with R** by Zev Ross
- **Spatial Data Science - Pebesma and Bivand**
- **Spatial Data Science Course- Prof. Adam Wilson**
- **Introduction to Mapping and Spatial Analysis with R**
- **Google R Style Guide**
- **Advanced R** by Hadley Wickham
- **Intro to GIS and Spatial Analysis** by Manuel Gimond
- **FOSS4G2019 R for Geospatial Processing**
- **An Introduction to Spatial Analysis and Mapping in R**

5.0.2 R Vector Processing / Simple Features Resources

- **Simple Features for R**
- **Spatial Data in R: New Directions**
- **sp-sf Migration**
- **An Exploration of Simple Features for R**
- **Simple Features: Building Spatial Data Pipelines in R**
- **Tidy spatial data in R: using dplyr, tidyr, and ggplot2 with sf**

5.0.3 R Raster Resources

- Wageningen University **Intro to Raster**
- Wageningen University **Advanced Raster Analysis**
- **The Visual Raster Cheat Sheet GitHub Repo**
- **Rastervis**
- **stars - spatiotemporal arrays**

5.0.4 R Mapping Resources

- **mapview**
- **Leaflet for R**
- **tmap**
- Zev Ross **Creating beautiful demographic maps in R with the `tidycensus` and `tmap` packages**
- Geocomputation with R: **Making maps with R**
- Ryan Peek: **Mapping in R**