

# Communicating Economics:

## Data Visualisation in R

---

Zaeen de Souza

Azim Premji University

6/05/21

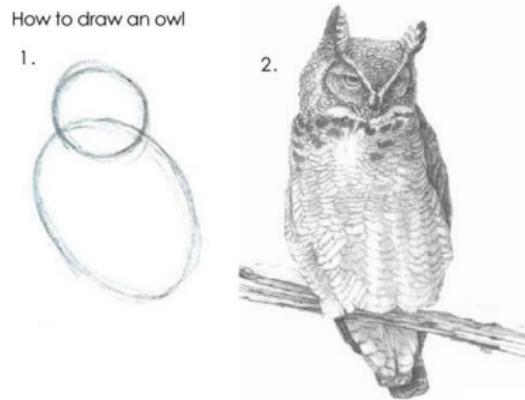
## Overview

---

You should be familiar with the following concepts:

1. **R syntax**
2. **ggplot/ggplot syntax**
3. **Aesthetics**
4. **Geometries**
5. **R help files**

However...learning ggplot sometimes feels like this



1. Draw some circles      2. Draw the rest of the fucking owl

Figure 1: A Step-by-Step Guide

## GIS data

How to read, use and graph geospatial data in R

## Colours/Scales

One variable vs Multiple variables

## Maps

---

## Bubble Maps

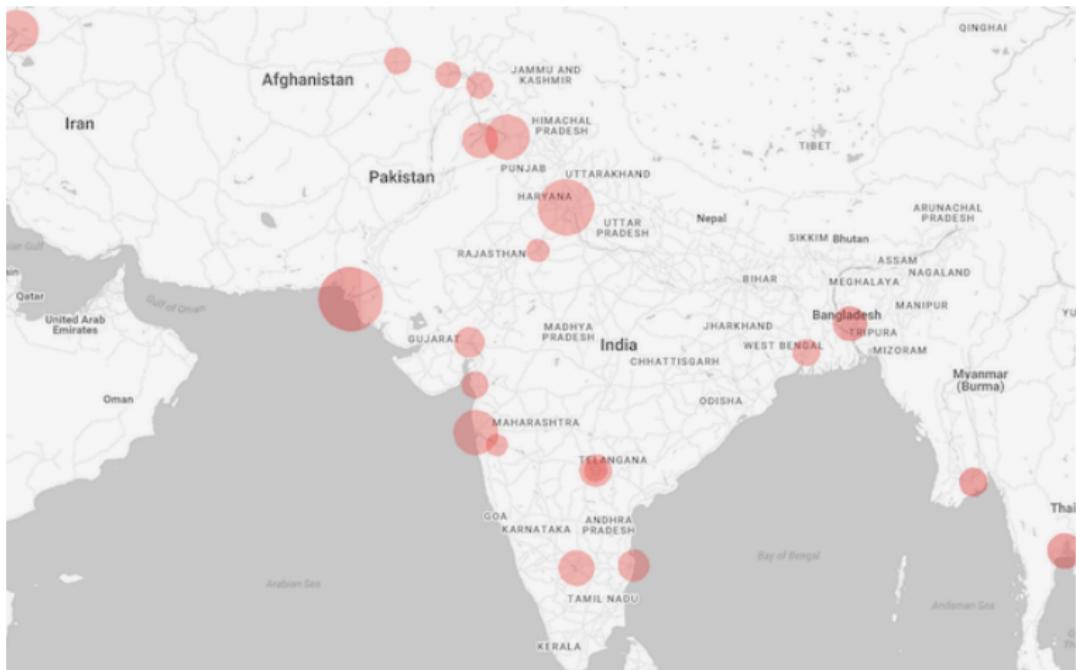
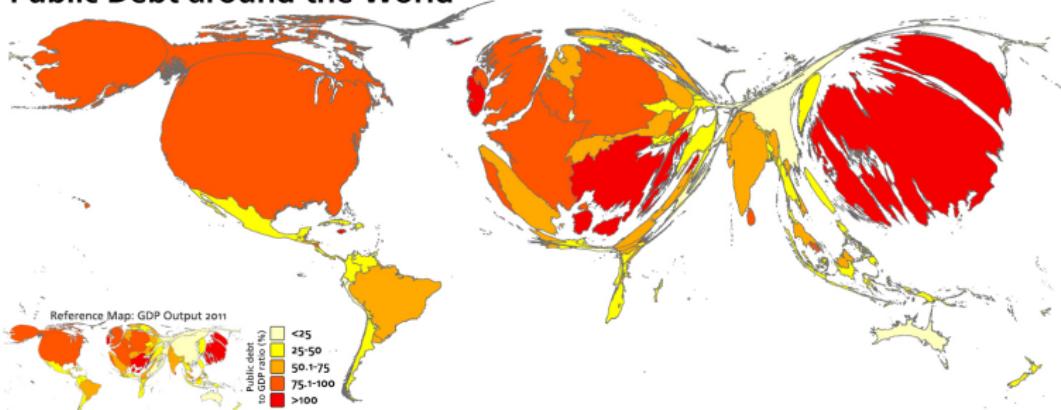


Figure 2: These are weighted centroids/individual points

\$54 Trillion

## Public Debt around the World



Data Sources: Compiled from IMF, World Economic Outlook Database 04/2011, with additional data from IMF & EUROSTAT  
Map created by Benjamin D. Hennig, Sasi Research Group, University of Sheffield

[www.viewsoftheworld.net](http://www.viewsoftheworld.net)

Figure 3: What's going on here?



**Figure 4:** These could be used to show networks/flows, trade, flights etc

## The data

---

This is a geojson file. You can read more about spatial data [here](#)

```
india <- geojson_read("my_map.geojson",
                      what = "sp")
```

## The data

geojson files have a few different features.

```
summary(india)
```

	Length	Class	Mode
##	36	SpatialPolygonsDataFrame	S4

1. Coordinates
2. Attributes

You can also use shape files, raster files etc.

```
library(broom)
names(map_data)[names(map_data) == 'ST_NM'] <- 'id'
```

We need to first, create a linking `id` across both datasets—the gis data and the survey data we want to merge to it.

```
# tidying the map data
india_df <- tidy(india, region = "ST_NM")
# merging by id
india_merge <- left_join(india_df,
                           map_data,
                           by = "id")
```

Your merged data should look like this

```
## Rows: 26,544
## Columns: 13
## $ long      <dbl> ~
## $ lat       <dbl> ~
## $ order     <int> ~
## $ hole      <lgl> ~
## $ piece     <fct> ~
## $ group     <fct> ~
## $ id        <chr> ~
## $ recovery_male <dbl> ~
## $ noeffect_male <dbl> ~
## $ recovery_female <dbl> ~
## $ noeffect_female <dbl> ~
## $ recovery_total <dbl> ~
## $ noeffect_total <dbl> ~
```

For spatial objects, there are some new **geoms**:

1. **geom\_polygon**
2. **geom\_sf**

What do you get if you plot the following data?

1. **latitude**
2. **longitude**

A map is basically a scatterplot

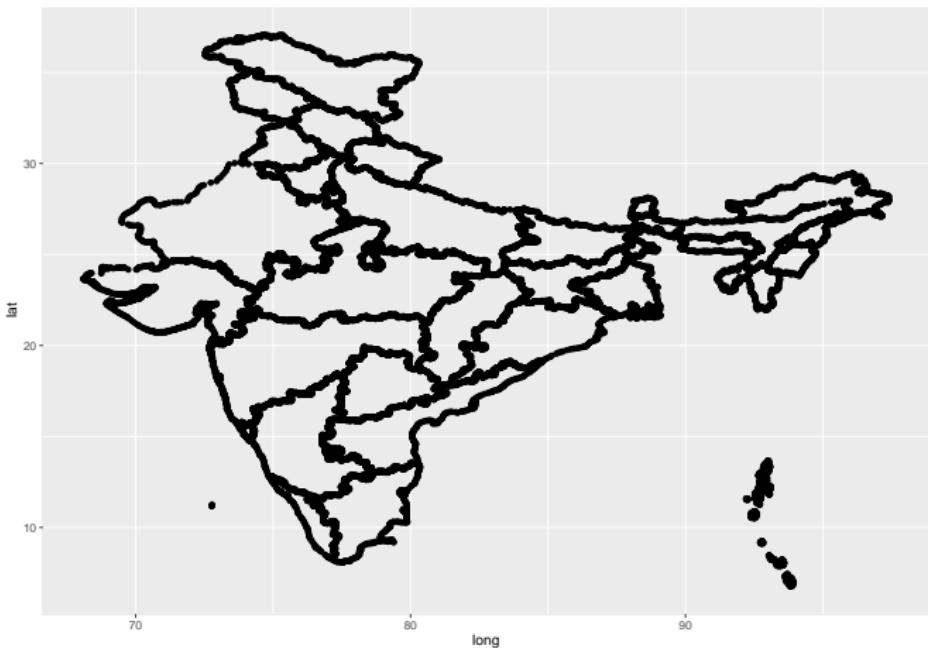


Figure 5: A Scatter plot!

If you colour areas by state, you get this

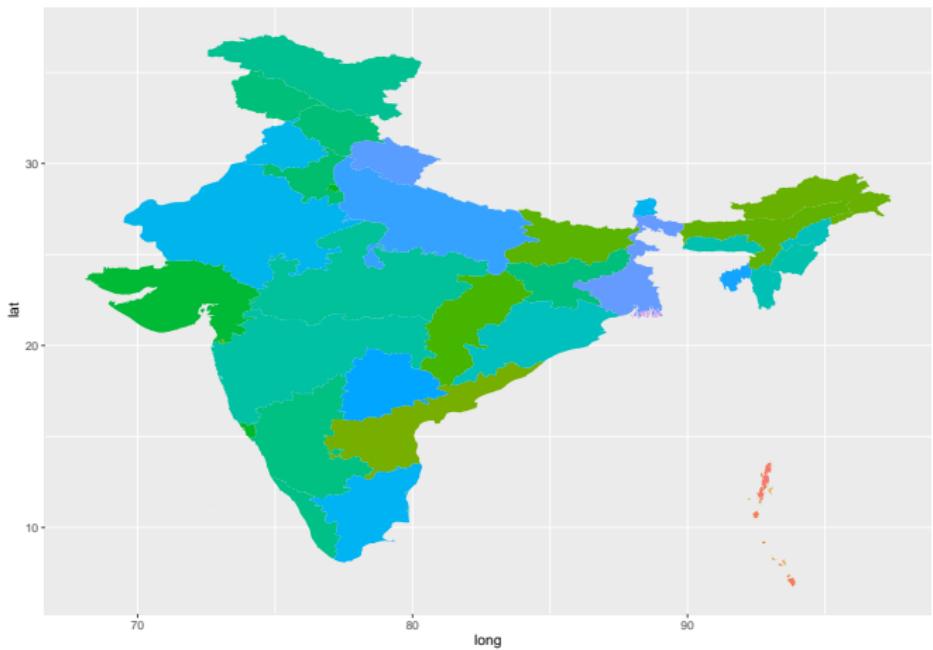


Figure 6: A basic map

But..., why is the map squashed?

```
ggplot() +  
  geom_polygon(data = india_merge,  
               aes(x = long,  
                   y = lat,  
                   group = group,  
                   fill = id),  
               show.legend = F) +  
  coord_map() # this corrects the dimensions!
```

## Map Projections: this is much better!

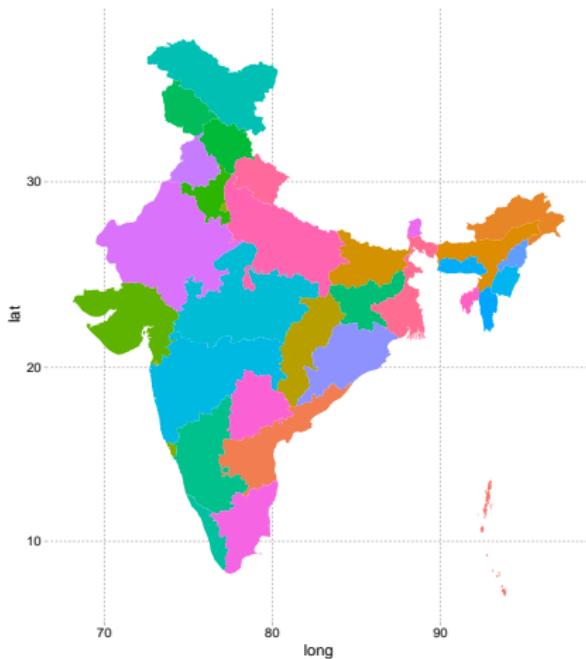


Figure 7: A basic map

We need to project a sphere onto a 2d plane—this distorts the scale, and as such, squashes the map. Usually, this is corrected with the function `coord_map()`. I think the distortion is less severe for countries around the equator

Suppose you want to shade different regions by the variable `recovery_total`.

```
ggplot() +  
  geom_polygon(data = india_merge,  
               aes(x = long,  
                   y = lat,  
                   group = group,  
                   fill = recovery_total*100),  
               color="white",  
               show.legend = T) +  
  # Choose a max/min colour code, and add here  
  scale_fill_gradient(low = "#F9F871",  
                      high = "#845EC2",  
  # Adding this to show how to deal with NA's/that there ARE missing values  
                      na.value = "black") +  
  coord_map() +  
  theme_void()
```

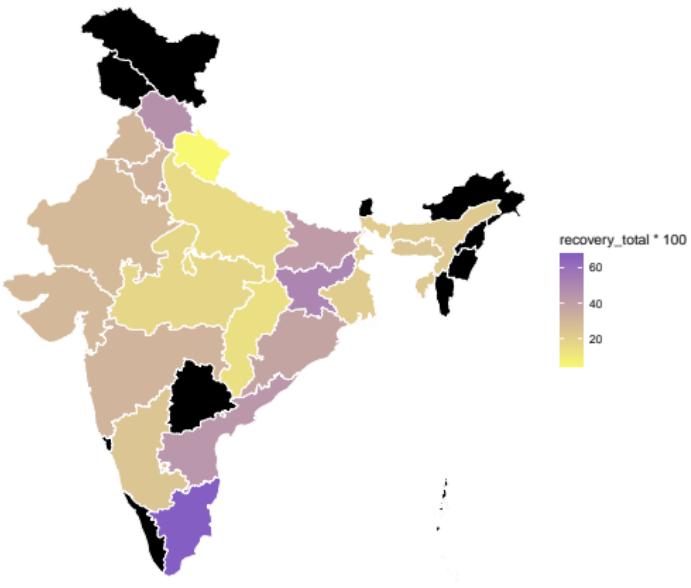
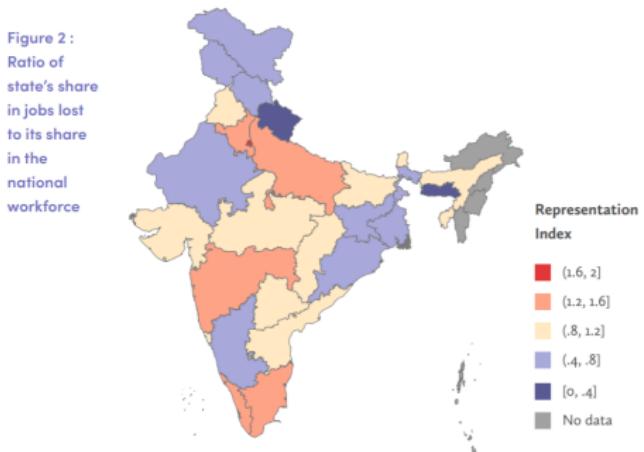


Figure 8: A basic map

## What if data is multivariate?

One solution is use to some kind of dimension reduction—an index, or create a new variable, out of 2 or more, as was done here:



Sources and notes: Authors' calculations based on CMIE-CPHS. Data is for the months of September, October, November, December of 2019 and 2020. See [Appendix section 2](#) for details. The representation index is a ratio of the state's share in employment loss to its share in the pre-Covid total workforce.

Figure 9: Mapping >1 variable

## What if data is multivariate?

---

What else could work? What are potential ways to represent multivariate data onto a choropleth map?

## Multivariate Scales: Take Home Exercise

You need two variables and two scales, going low to high, from left to right—note, the low colour is common across scales.

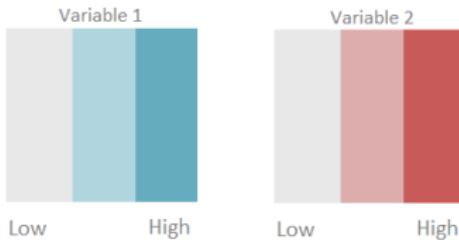


Figure 10: Mapping >1 variable

## Multivariate Scales: Take Home Exercise

Step 2: Rotate the first 90 degrees, so it goes low to high, from bottom to top.

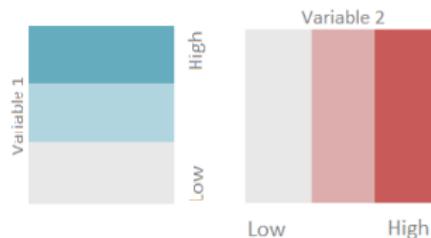


Figure 11: Mapping >1 variable

## Multivariate Scales: Take Home Exercise

Overlay the two scales as such:

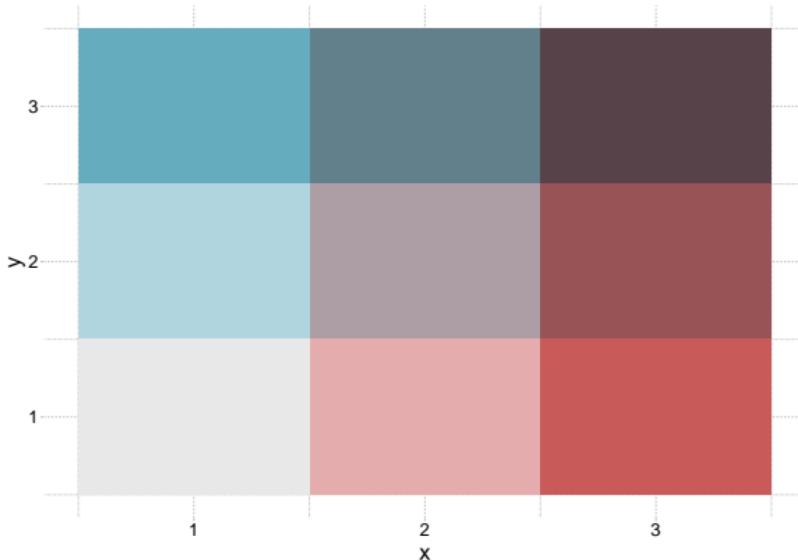


Figure 12: Overlaying one scale on another

# A Small Example (With Incomplete Data!)

## Male and Female Job Recovery Doesn't Go Hand-in-Hand

Something something something something something

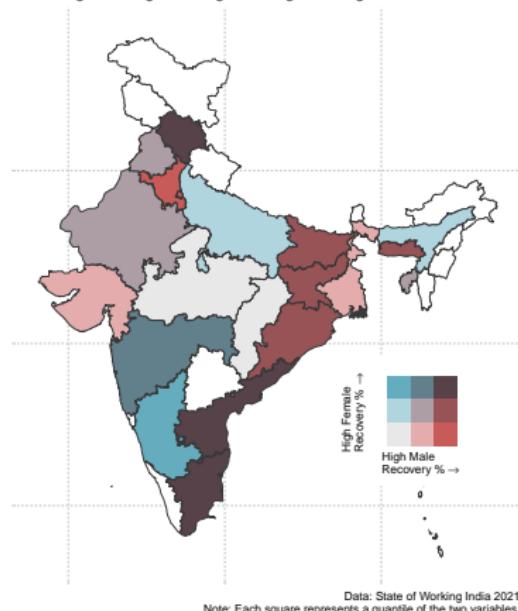


Figure 13: Try to replicate/build scales like this

However, these tend to look better with more granular data

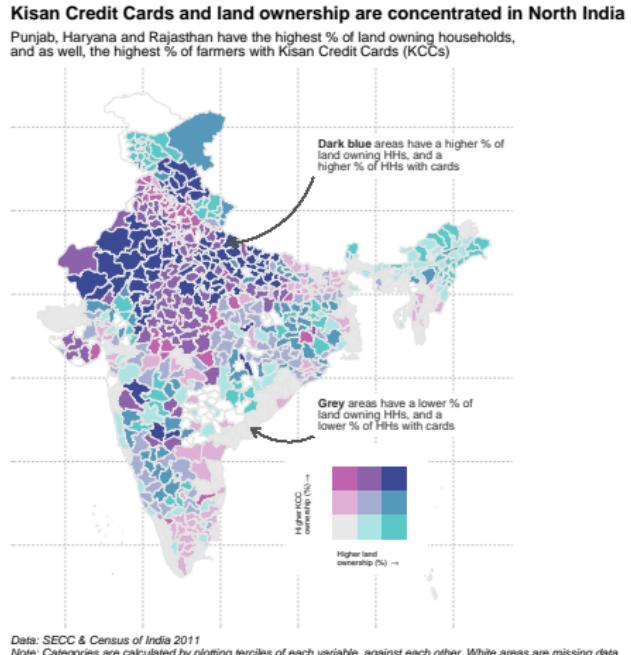


Figure 14: Try to make something similar!

Use what we saw in the last class to make an annotated map like that—use the following:

1. `geom_polygon` OR `geom_sf` : This is the spatial geom
2. `geom_curve` : This is an arrow geom
3. `annotate` : this is for annotation
4. `geom_richtext` : this is for annotation

As well as any other ggplot compatible function from the list shared earlier.

Thank You!

## Extra Slides

---

R resources:

1. [Colour scales in ggplot](#)
2. [Map colour picker](#)
3. [Colour in map design](#)
4. [Geocomputation with R](#)

Interesting articles:

1. [Gerardus Mercator revolutionized mapmaking. He was almost executed for it.](#)
2. [John Snow's data journalism: the cholera map that changed the world](#)
3. [Storytelling by Martin Krzywinski & Alberto Cairo](#)

## Adding annotation layers to ggplot maps

While the code itself is identical to the earlier example of annotation we saw in the dumbbell example, referencing coordinates is slightly different. An extract from the annotated district map is here:

```
# this adds the curved arrow that points to some part of the map
geom_curve(aes(
# these coordinates will be lat/long points
# that you can manually find from looking at the map axes
x = 86,
y = 32,
xend = 80.5,
yend = 28),
colour = "#555555",
curvature = -0.3,
size = 0.1,
arrow = arrow(length = unit(0.02, "npc")), data = bi)
```

## Adding annotation layers to ggplot maps

This adds the text label at the start of the arrow.

```
geom_richtext(aes(  
  x = 86,  
  y = 32,  
  # note the markdown syntax to add bold text and line breaks  
  label = "★★Dark blue★★ areas have a higher % of<br>land  
  owning HHs, and a<br>higher % of HHs with cards"),  
  lineheight = 0.8,  
  colour = "#2b2b2b",  
  fill = "white",  
  vjust = 0,  
  hjust = 0,  
  label.size = NA,  
  size = 3)
```

Extra, Extra Slides

---

1. Not So Standard Deviations by Hilary Parker, Roger Peng
2. Tidy Tuesday by Jon Harmon
3. Florence Nightingale Data Viz Pioneer by Tim Harford
4. Weapons of Math Destruction by Cathy O'Neil
5. Coded Bias (Netflix)
6. R4DS Office Hours