R for Social Sciences, Public Policy and Humanities

Zahid Asghar

Agenda

Summarising Data Chapter 2

Discussion of various summary statistics

How to summarize numerical and categorical data

Introduction to R and Lab based demonstration plus HW

Why R?

Free

Flexible

future-proof (sort of)

atomcamp

What can R do for you?

R is a popular language & platform for data science & statistical computing. It is:

open source

expanding (increasing capabilities through add-ons)

able to open almost any data format

able to scrape data from the web

a decent tool for data wrangling

But R also:

- is a slightly awkward language for those with programming experience has a steep learning curve
- requires a willingness to write code and use scripts (cf. Tableau & co.) is less general than Python (but a bit easier to use for advanced statistical computing)

Why R in a Social Science, Humanities and Public Policy setting?

R is very versatile; it can be used in a variety of settings (cf. specialized tools for specific purposes)

R is open source and free

How might you use R?

Create dataviz for teaching

Introduce as a tool for students

Your own research

Case studies

Analyze economic & demographic data

Import data into R

If you're new to R and/or coding, this may look like overload- But! This is a good starting point for you.

Everything I'm doing you'll be able to reproduce on your own

Things I won't be able to show:

Intro to the R language itself We don't have the time, so learn by tweaking my code RMarkdown (using R to produce complete documents or slides) Text analysis in R R offers powerful packages! Links at the end of this workshop

R is a calculator

```
1 + 1
```

```
## [1] 2
```

R is an object-based language

```
students <- 16
papers <- 3
papers_to_grade <- students * papers
papers_to_grade
```

```
## [1] 48
```

Try for yourself!

How many papers would you have to grade if you were teaching two instead of one section?

```
students <- 16
papers <- 3
classes <- 2
papers_to_grade <- students * papers * classes
papers_to_grade
```

```
## [1] 96
```

R can be extended by using one of 12,621 packages

See (https://cran.r-project.org/web/packages/)

Install packages once, load them each time

For data input/output:

```
library("tidyverse")
#help(package = "tidyverse")
```

Example 1: Data from the CIA World Factbook (2014), prepared by OpenIntro Statistics

```r
cia <- read_csv("cia_factbook.csv")
#glimpse(cia)
#View(cia)
```

Life expectancy

```
ggplot(data = cia, aes(x = life_exp_at_birth)) + geom_histogram()
```

```
## Warning: Removed 35 rows containing non-finite values (stat_bin).
```

Try for yourself!

How is the net migration rate distributed?

```
ggplot(data = cia, aes(x = net_migration_rate)) + geom_histogram()
```

Life expectancy -> more emigration?

```
ggplot(data = cia, aes(x = life_exp_at_birth, y = net_migration_rate)) +
  geom_point() +
  geom_text(aes(label = country))
```

Let's un-clutter this:

```
filter(cia, net_migration_rate > 20 | net_migration_rate < -20)
```

Let's un-clutter this:

```
ggplot(data = cia, aes(x = life_exp_at_birth, y = net_migration_rate)) +
  geom_point() +
  geom_text(data = filter(cia, net_migration_rate > 20 | net_migration_rate < -20), aes(label = cou
```

Are the two variables related?

How does internet access vary around the world?

I could use internet_users, but the raw number is bad for comparison. So let's divide by population:

```
cia <- mutate(cia,
              internet_users_perc = internet_users / population * 100)
```

How does internet access vary around the world?

```
ggplot(data = cia, aes(x = internet_users_perc)) + geom_histogram()
```

```
## Warning: Removed 46 rows containing non-finite values (stat_bin).
```

Higher life expectancy -> more internet access?

```
ggplot(data = cia, aes(x = life_exp_at_birth, y = internet_users_perc)) +
  geom_point() +
  geom_smooth()
```

```
## Warning: Removed 51 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 51 rows containing missing values (geom_point).
```

Let's improve this plot!

```
## Warning: Removed 51 rows containing non-finite values (stat_smooth).

## Warning: Removed 51 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_text).
```

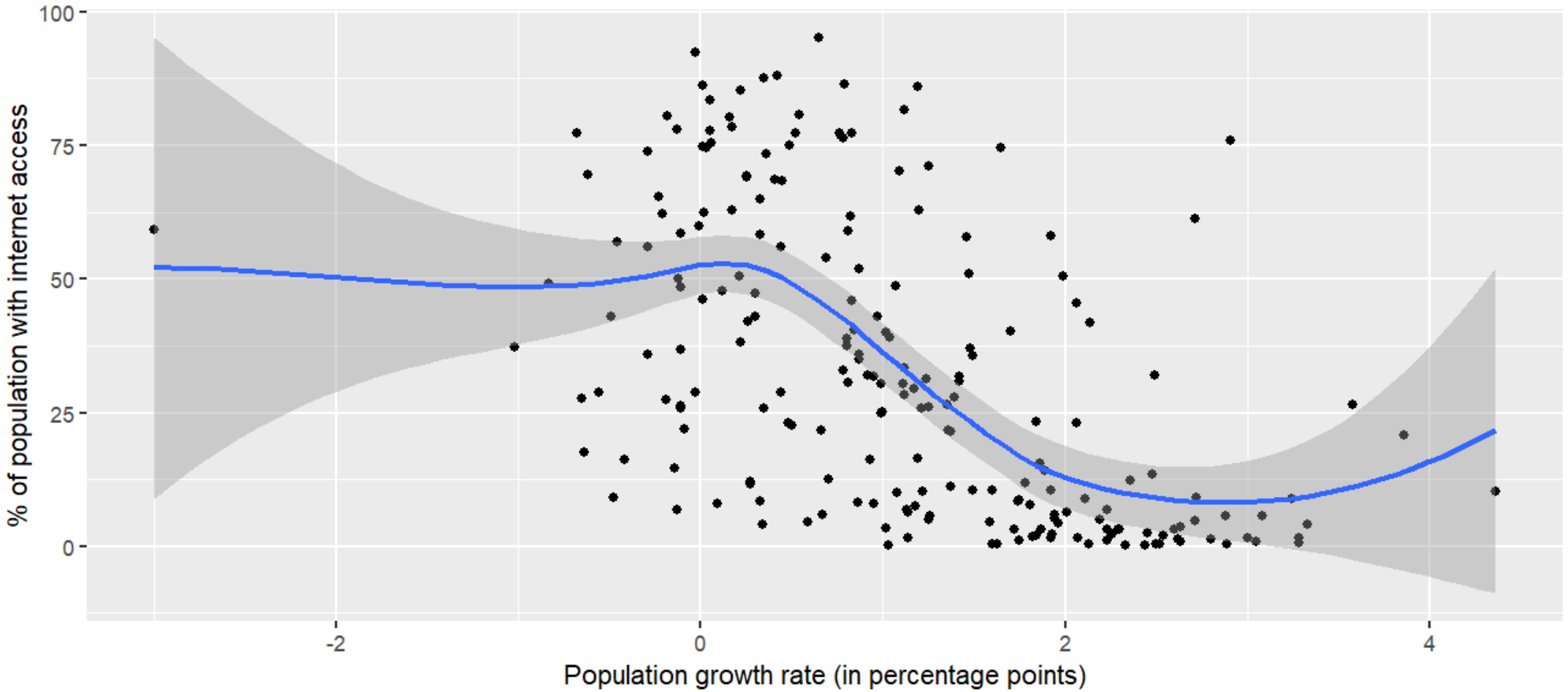
Try for yourself!

How would you plot internet access against population growth (population_growth_rate)?

Identify the outliers

```
filter(cia, population_growth_rate < -5 | population_growth_rate > 5)
```

Let's try again, w/o outliers



Visualize data on a map

First, use the built-in map tools in ggplot2:

```
library(ggplot2)
library(tidyverse)
worldmap <- map_data("world")
glimpse(worldmap)
```

```
## Rows: 99,338
## Columns: 6
## $ long      <dbl> -69.89912, -69.89571, -69.94219, -70.00415, -70.06612, -70.05…
## $ lat       <dbl> 12.45200, 12.42300, 12.43853, 12.50049, 12.54697, 12.59707, 1…
## $ group     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2…
## $ order     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19…
## $ region    <chr> "Aruba", "Aruba", "Aruba", "Aruba", "Aruba", "Aruba", "Aruba"…
## $ subregion <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
```

Clean some country names

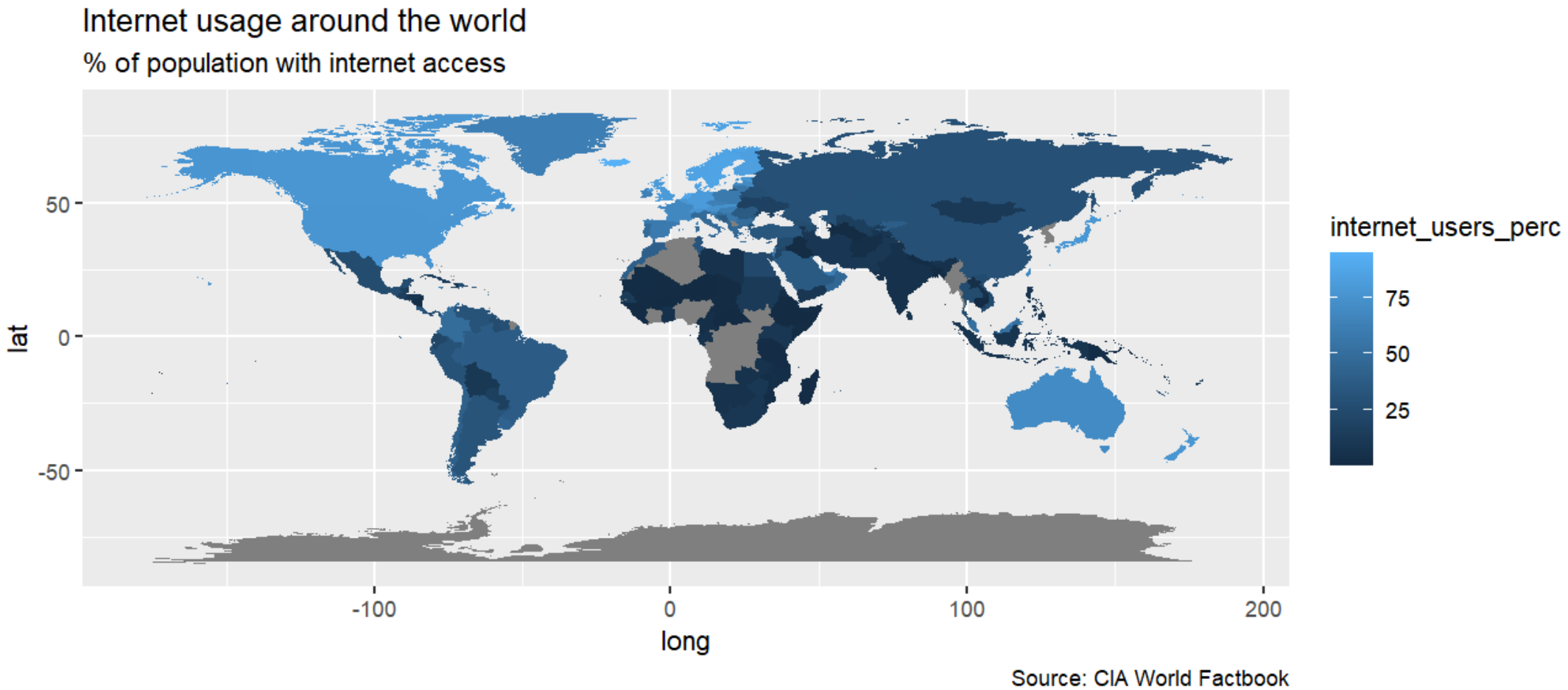
```
cia <- mutate(cia,
              country = ifelse(country == "United States", "USA", country))
cia <- mutate(cia,
              country = ifelse(country == "United Kingdom", "UK", country))
```

Join CIA and map data

```
iumap <- left_join(x = worldmap,
                   y = cia,
                   by = c("region" = "country"))
glimpse(iumap)
```

```
## Rows: 99,338
## Columns: 17
## $ long                  <dbl> -69.89912, -69.89571, -69.94219, -70.00415, -70…
## $ lat                   <dbl> 12.45200, 12.42300, 12.43853, 12.50049, 12.5469…
## $ group                 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,…
## $ order                 <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, …
## $ region                <chr> "Aruba", "Aruba", "Aruba", "Aruba", "Aruba", "A…
## $ subregion             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ area                  <dbl> 180, 180, 180, 180, 180, 180, 180, 180, 180, 18…
## $ birth_rate            <dbl> 12.65, 12.65, 12.65, 12.65, 12.65, 12.65, 12.65…
## $ death_rate            <dbl> 8.09, 8.09, 8.09, 8.09, 8.09, 8.09, 8.09, 8.09,…
## $ infant_mortality_rate <dbl> 11.74, 11.74, 11.74, 11.74, 11.74, 11.74, 11.74…
## $ internet_users        <dbl> 24000, 24000, 24000, 24000, 24000, 24000, 24000…
## $ life_exp_at_birth     <dbl> 76.35, 76.35, 76.35, 76.35, 76.35, 76.35, 76.35…
## $ maternal_mortality_rate <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 460, 46…
## $ net_migration_rate    <dbl> 9.04, 9.04, 9.04, 9.04, 9.04, 9.04, 9.04, 9.04,…
## $ population            <dbl> 110663, 110663, 110663, 110663, 110663, 110663,…
```

First take: a chloropleth map



Some improvements

Map projection Labels Remove Antarctica Legend placement

```
worldmap_noant <- filter(worldmap,
                         region != "Antarctica")
iumap <- left_join(x = worldmap_noant,
                   y = cia,
                   by = c("region" = "country"))
```

Some improvements

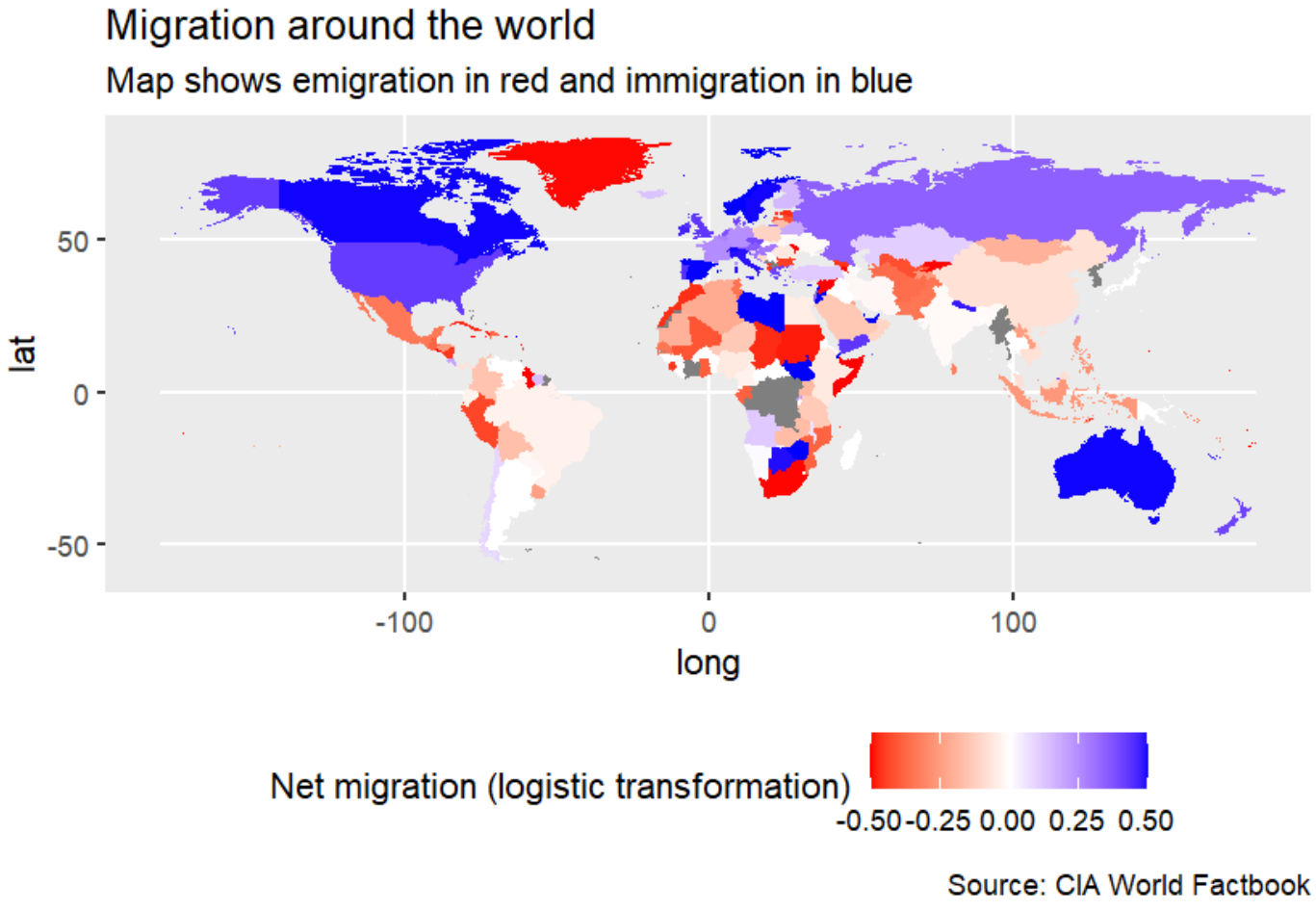
Map projection Labels Remove Antarctica Legend placement

```
ggplot(data = iumap, aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = life_exp_at_birth)) +
  labs(title = "Internet usage around the world",
       subtitle = "% of population with internet access",
       caption = "Source: CIA World Factbook",
       fill = "% of population with internet access") +
  coord_map(projection = "rectangular", lat0 = 0, xlim = c(-180, 180)) +
  theme(legend.position = "bottom")
```

atomcamp

Try for yourself:

Map migration rates around the world!



Adding locations is also easy. Let's pick capitals...

First, I scrape location data from the web (using the "rvest" package): Google points me to http://techslides.com/list-of-countries-and-capitals...

```
library("rvest")
cap_url <- read_html("http://techslides.com/list-of-countries-and-capitals")
cap_nodes <- html_nodes(cap_url, "table")
cap_table <- html_table(cap_nodes[1], fill = TRUE, header = TRUE)[[1]]
glimpse(cap_table)
```

```
## Rows: 245
## Columns: 6
## $ `Country Name`      <chr> "Afghanistan", "Aland Islands", "Albania", "Algeria…
## $ `Capital Name`      <chr> "Kabul", "Mariehamn", "Tirana", "Algiers", "Pago Pa…
## $ `Capital Latitude`  <dbl> 34.516667, 60.116667, 41.316667, 36.750000, -14.266…
## $ `Capital Longitude` <dbl> 69.183333, 19.900000, 19.816667, 3.050000, -170.700…
## $ `Country Code`      <chr> "AF", "AX", "AL", "DZ", "AS", "AD", "AO", "AI", "AQ…
## $ `Continent Name`    <chr> "Asia", "Europe", "Europe", "Africa", "Australia", …
```

Fixing a few country names and removing mini-states

```
cap_table <- mutate(cap_table,
                    `Country Name` = ifelse(`Country Name` == "United States", "USA", `Country Name`
cap_table <- mutate(cap_table,
                    `Country Name` = ifelse(`Country Name` == "United Kingdom", "UK", `Country Name`
cia_with_caps <- left_join(x = cia,
                   y = cap_table,
                   by = c("country" = "Country Name"))
cia_with_caps <- mutate(cia_with_caps,
                        no_ministates = ifelse(population >= 1000000,
                                               1,
                                               0))
```

Internet access, with capitals

```
ggplot(data = iumap, aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = internet_users_perc)) +
  geom_point(data = filter(cia_with_caps, no_ministates == 1),
             aes(x = `Capital Longitude`, y = `Capital Latitude`, group = NULL),
             color = "orange", size = 1) +
  labs(title = "Internet usage around the world",
       subtitle = "% of population with internet access",
       caption = "Source: CIA World Factbook",
       fill = "% of population with internet access") +
  coord_map(projection = "rectangular", lat0 = 0, xlim = c(-180, 180)) +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```

Instead of building your own...

you can use some built-in mapping tools, too!

Let's look at some economic data for the tri-state area, using the "blscrapeR" package to pull data from the API of the U.S. Bureau of Labor Statistics.

atomcamp

Example 2 : #oscarssowhite

What do we know about diversity among Academy Award winners over time?

I use data provided by Crowdflower/FigureEight: https://data.world/crowdflower/academy-awards-demographics

```
##aa <- import("Data/crowdflower-academy-awards-demographics/data/oscars_demographics_dfe.csv")
aa<-read_csv("Oscars-demographics-DFE.csv")
glimpse(aa)
```

```
## Rows: 441
## Columns: 27
## $ `_unit_id`                    <dbl> 670454353, 670454354, 670454355, 670454…
## $ `_golden`                     <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALS…
## $ `_unit_state`                 <chr> "finalized", "finalized", "finalized", …
## $ `_trusted_judgments`          <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, …
## $ `_last_judgment_at`           <chr> "2/10/15 3:45", "2/10/15 2:03", "2/10/1…
## $ birthplace                    <chr> "Chisinau, Moldova", "Glasgow, Scotland…
## $ `birthplace:confidence`       <dbl> 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,…
## $ date_of_birth                 <chr> "30-Sep-1895", "2-Feb-1886", "30-Sep-18…
## $ `date_of_birth:confidence`    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ race_ethnicity                <chr> "White", "White", "White", "White", "Wh…
## $ `race_ethnicity:confidence`   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ religion                      <chr> "Na", "Na", "Na", "Na", "Roman Catholic…
```

Which awards are in the dataset?

```
table(aa$award)
```

```
##
##               Best Actor              Best Actress           Best Director
##                       88                        95                      91
##   Best Supporting Actor Best Supporting Actress
##                       82                        85
```

AA winners overall

```
ggplot(data = aa, aes(x = race_ethnicity)) + geom_bar()
```

AA winners over time

First, collapse the data:

```
aa_year <- summarize(group_by(aa, year_of_award, race_ethnicity),
                     awards = n())
```

AA winners over time

Then, create the plot:

```
ggplot(data = aa_year,
       aes(x = year_of_award, y = awards, color = race_ethnicity)) +
  geom_point() +
  ylim(0, NA)
```

More recent trends since 1960

```
ggplot(data = filter(aa_year, year_of_award >= 1960), aes(x = year_of_award, y = awards, fill = race
  geom_col() +
  ylim(0, NA)
```