Tidy

K Arnold

Homework 3

- GitHub repos created for you!
- Comment out 2012 data if it's giving you trouble

Facial Recognition (Discussion 2)

Sit near your team (see attandance sheet)

Why cohorts?

- Relationships in a socially distant time
- Teamwork as a "soft skill"
- Learn collaborative workflow using GitHub
- Help each other within a cohort
- Course staff can meet with cohorts together
- Possible final project teams

For each assignment, each *cohort* randomly split into 2 *teams*.

Slow it down...

- HW 3 due date extension
- HW 4 small

Friday

- Reminder: Quiz (including feedback)
- Discussion: reply instructions posted
- HW4 and Prep 4 posted this afternoon.
- Which two cohorts want to meet with me next week?

Facial Recognition Surveillance

- What good points did others raise?

Ethical Frameworks

- Utilitarian ("do the benefits outweigh the harms? to whom?")
- Virtue ("does this align with my core values?")
- Analogical ("is there a simpler situation I can compare this to?")
- Deontological ("is this decision lawful?")

Which did you use? Which did your peers use?

Tidying and Joining Data

Outline:

- the dataset
- tidying
- joining
- plotting

Data wrangling often takes a *lot* of time and effort, so buckle in.

JHU COVID-19 data

As you might imagine, keeping a comprehensive list of all COVID-19 cases worldwide involves pulling data from numerous sources (and frequent updating). Fortunately, some folks at Johns Hopkins have been doing that work and putting the resulting data into a github repository that anyone can access.

Navigating the GitHub Repo

You can visit their github project at https://github.com/CSSEGISandData/COVID-19. There you will find

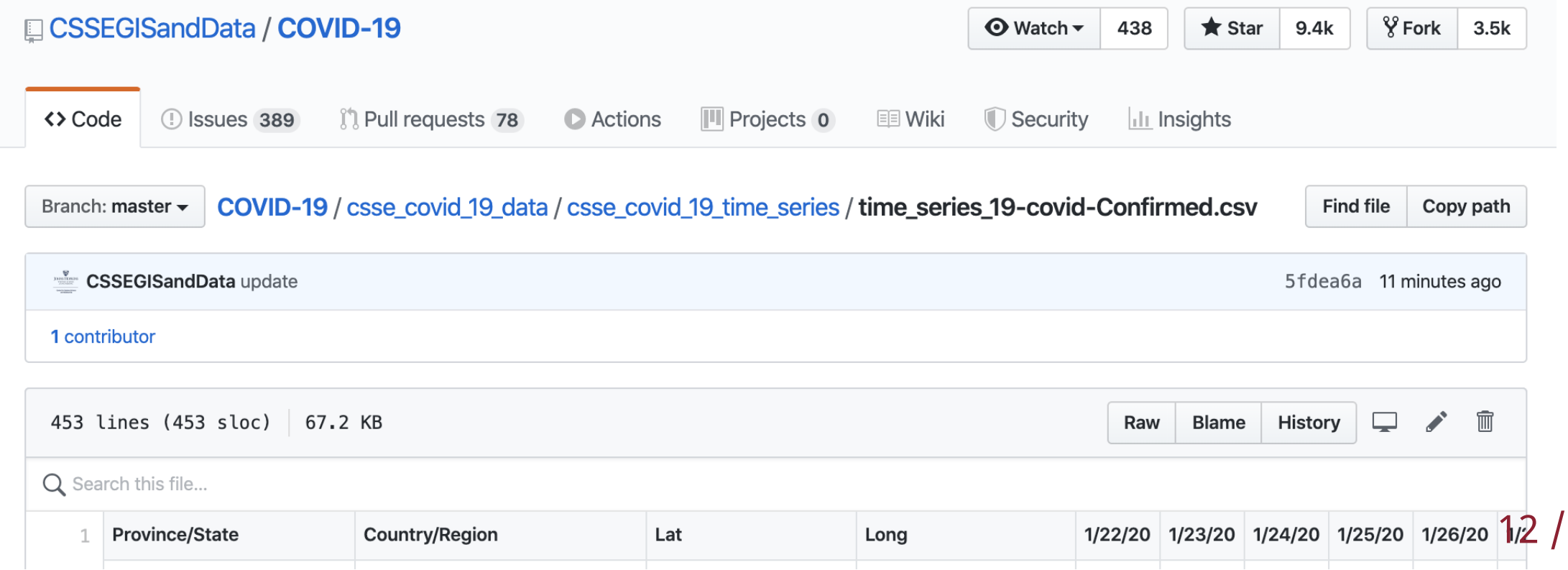
- details about what sources were used for the data,
- what sorts of data are available, and
- some places the data have been used

You will also see this note:

> The Website relies upon publicly available data from multiple sources, that do not always agree.

Finding some data

You could clone the repository, but you also just pull the data directly from their repository. They've split the data into "daily reports" (one CSV per day, all measures) and "time series" (one CSV per measure, all days). Here is an example page showing one of the data sets available to you.



We want raw data

GitHub renders CSVs in a fancy way, but you can get the plain old CSV if you click the Raw button. We're mostly interested in the URL for this file, since that will let us pull the data into R.

```
confirmed_global_url <- paste0(
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/",
  "csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_",
  "confirmed", # also: "deaths", "recovered"
  "_global.csv"
)
```

Read it in.

```
confirmed_global <- confirmed_global_url %>%
  pins::pin() %>%
  read_csv(col_types = cols(
    .default = col_double(),
    `Province/State` = col_character(),
    `Country/Region` = col_character()
  )) %>%
  rename(
    country_or_region = `Country/Region`,
    province_or_state = `Province/State`
  )
```

Search

province_or_state	country_or_region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/2
	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	
	Albania	41.1533	20.1683	0	0	0	0	0	0	
	Algeria	28.0339	1.6596	0	0	0	0	0	0	
	Andorra	42.5063	1.5218	0	0	0	0	0	0	
	Angola	-11.2027	17.8739	0	0	0	0	0	0	

1–5 of 266 rows

Previous **1** 2 3 4 5 … 54 Next

Notice that each day's count of confirmed cases is in a separate column. Suppose we want to plot the number of cases over time. What about the structure of this table might give us trouble?

Search

province_or_state	country_or_region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/2
	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	
	Albania	41.1533	20.1683	0	0	0	0	0	0	
	Algeria	28.0339	1.6596	0	0	0	0	0	0	
	Andorra	42.5063	1.5218	0	0	0	0	0	0	
	Angola	-11.2027	17.8739	0	0	0	0	0	0	

1–5 of 266 rows

Previous 1 2 3 4 5 … 54 Next

Notice that each day's count of confirmed cases is in a separate column.
Suppose we want to plot the number of cases over time. What about the structure of this table might give us trouble?
How many observations does each row represent?

Think and discuss

Suppose we want to plot the number of cases over time.
What *should* the table look like?

Tidying Step 1: `pivot_longer`

- We have: Lots of observations per row
- We want: one observation per row.
- So: we're gonna need a *longer* (and narrower) dataset.

enter `pivot_longer`!

```
confirmed_global %>%
  pivot_longer(
    -(1:4) # the first 4 columns are not part of the pivot
  )
```

```
## # A tibble: 68,628 x 6
##   province_or_state country_or_region   Lat  Long name    value
##   <chr>             <chr>             <dbl> <dbl> <chr>   <dbl>
## 1 <NA>              Afghanistan        33.9  67.7 1/22/20     0
## 2 <NA>              Afghanistan        33.9  67.7 1/23/20     0
## 3 <NA>              Afghanistan        33.9  67.7 1/24/20     0
## 4 <NA>              Afghanistan        33.9  67.7 1/25/20     0
## 5 <NA>              Afghanistan        33.9  67.7 1/26/20     0
## 6 <NA>              Afghanistan        33.9  67.7 1/27/20     0
## # … with 68,622 more rows
```

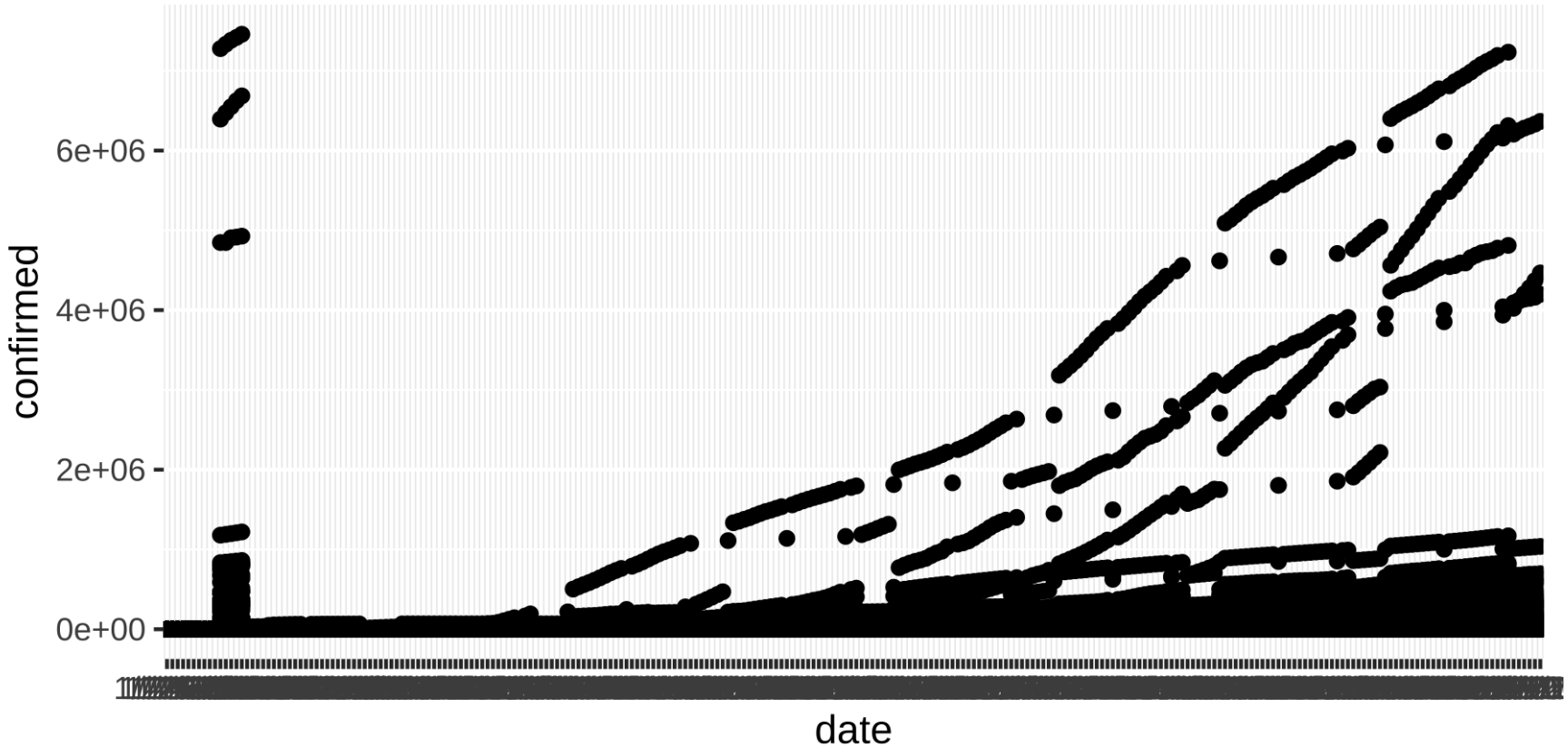
```
confirmed_global %>%
  pivot_longer(
    -(1:4),
    names_to = "date"
  )
```

```
## # A tibble: 68,628 x 6
##   province_or_state country_or_region   Lat  Long date    value
##   <chr>             <chr>             <dbl> <dbl> <chr>   <dbl>
## 1 <NA>              Afghanistan        33.9  67.7 1/22/20     0
## 2 <NA>              Afghanistan        33.9  67.7 1/23/20     0
## 3 <NA>              Afghanistan        33.9  67.7 1/24/20     0
## 4 <NA>              Afghanistan        33.9  67.7 1/25/20     0
## 5 <NA>              Afghanistan        33.9  67.7 1/26/20     0
## 6 <NA>              Afghanistan        33.9  67.7 1/27/20     0
## # … with 68,622 more rows
```

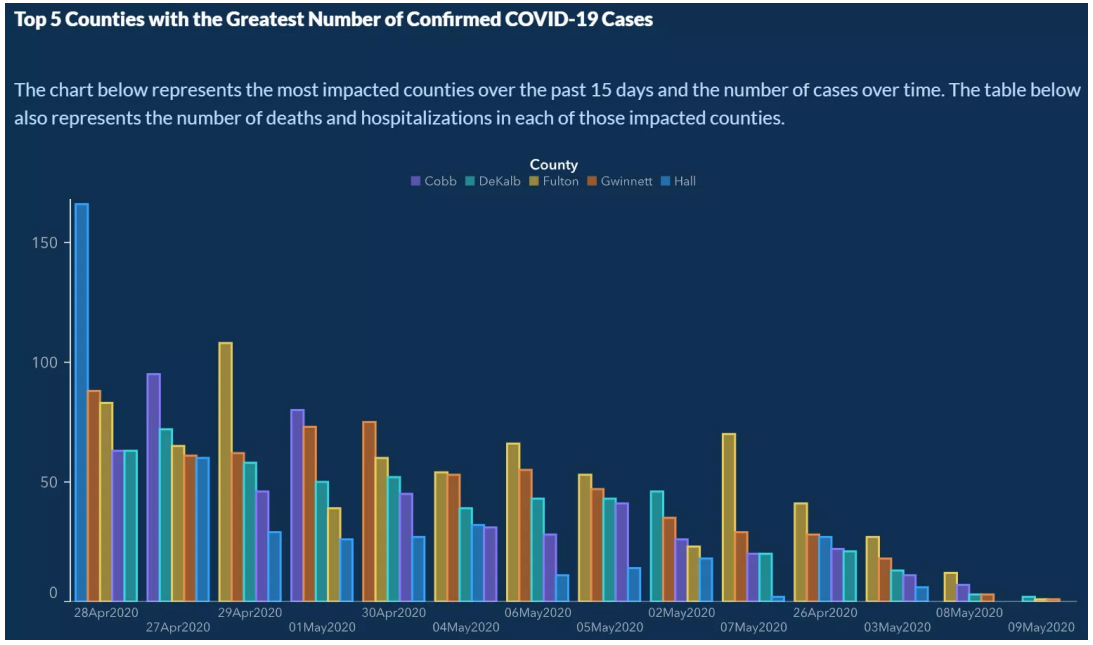
```r
confirmed_global_long <-
  confirmed_global %>%
  pivot_longer(
    -(1:4),
    names_to = "date",
    values_to = "confirmed"
  )
confirmed_global_long
```

```
## # A tibble: 68,628 x 6
##   province_or_state country_or_region   Lat  Long date    confirmed
##   <chr>             <chr>              <dbl> <dbl> <chr>       <dbl>
## 1 <NA>              Afghanistan         33.9  67.7 1/22/20         0
## 2 <NA>              Afghanistan         33.9  67.7 1/23/20         0
## 3 <NA>              Afghanistan         33.9  67.7 1/24/20         0
## 4 <NA>              Afghanistan         33.9  67.7 1/25/20         0
## 5 <NA>              Afghanistan         33.9  67.7 1/26/20         0
## 6 <NA>              Afghanistan         33.9  67.7 1/27/20         0
## # … with 68,622 more rows
```

```
ggplot(confirmed_global_long, aes(x = date, y = confirmed)) +
  geom_point()
```



This happens!



Source: https://www.vox.com/covid-19-coronavirus-us-response-trump/2020/5/18/21262265/georgia-covid-19-cases-declining-reopening

Aside: lots of Covid visualizations are problematic

- https://sirota.substack.com/p/georgias-misleading-covid-map
- https://medium.com/nightingale/ten-considerations-before-you-create-another-chart-about-covid-19-27d3bd691be8

```
"2020-02-01" %>%
  parse_date() %>%
  lubridate::month()
```

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

```
"2/1/20" %>% parse_date() #<< Fail: date parser needs help!
```

```
"2/1/20" %>%
  parse_date_time("%m/%d/%y!*") %>%
  lubridate::month()
```

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

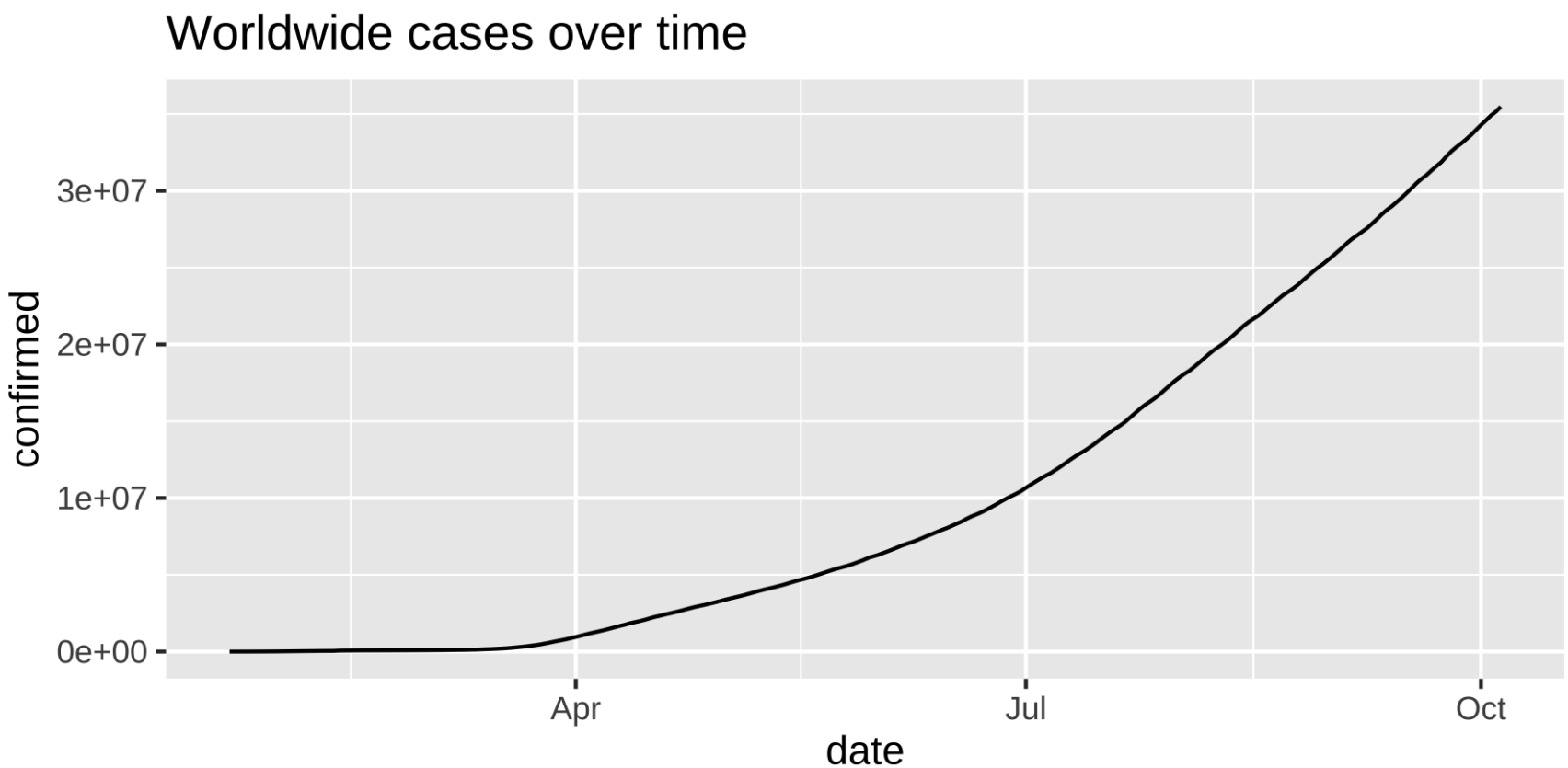
```
confirmed_global_long <-
  confirmed_global %>%
  pivot_longer(
    -(1:4), # the first 4 columns are not part of the pivot
    names_to = "date", # names of the remaining columns will be put into a date column
    values_to = "confirmed"
  ) %>% # values will be put into a column called confirmed
  mutate(date = lubridate::parse_date_time(date, "%m/%d/%y!*")) # convert to date objects

confirmed_global_long
```

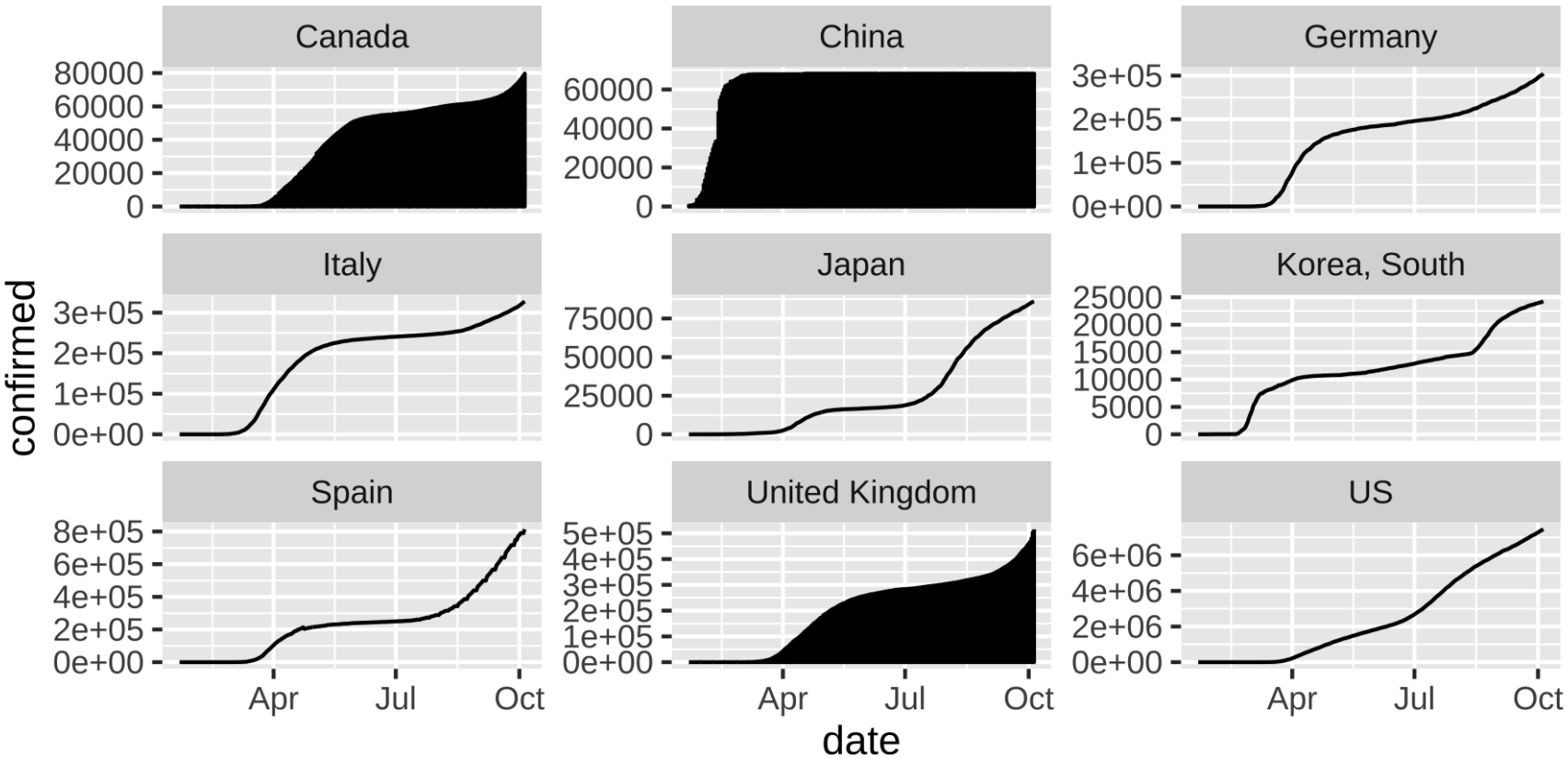
```
## # A tibble: 68,628 x 6
##   province_or_state country_or_region   Lat  Long date                confirmed
##   <chr>             <chr>             <dbl> <dbl> <dttm>                  <dbl>
## 1 <NA>              Afghanistan        33.9  67.7 2020-01-22 00:00:00         0
## 2 <NA>              Afghanistan        33.9  67.7 2020-01-23 00:00:00         0
## 3 <NA>              Afghanistan        33.9  67.7 2020-01-24 00:00:00         0
## 4 <NA>              Afghanistan        33.9  67.7 2020-01-25 00:00:00         0
## 5 <NA>              Afghanistan        33.9  67.7 2020-01-26 00:00:00         0
## 6 <NA>              Afghanistan        33.9  67.7 2020-01-27 00:00:00         0
## # … with 68,622 more rows
```

Plotting the data

```
confirmed_global_long %>%
  group_by(date) %>%
  summarize(confirmed = sum(confirmed)) %>%
  ggplot(aes(x = date, y = confirmed)) +
    geom_line() +
    labs(title="Worldwide cases over time")
```



```
confirmed_global_long %>%
  filter(country_or_region %in%
    c("US", "Canada", "China", "Japan", "Korea, South", "Italy", "Germany", "Spain", "United Kingdor
  ggplot(aes(x = date, y = confirmed)) +
    geom_line() +
    facet_wrap(~country_or_region, scales = "free_y")
```

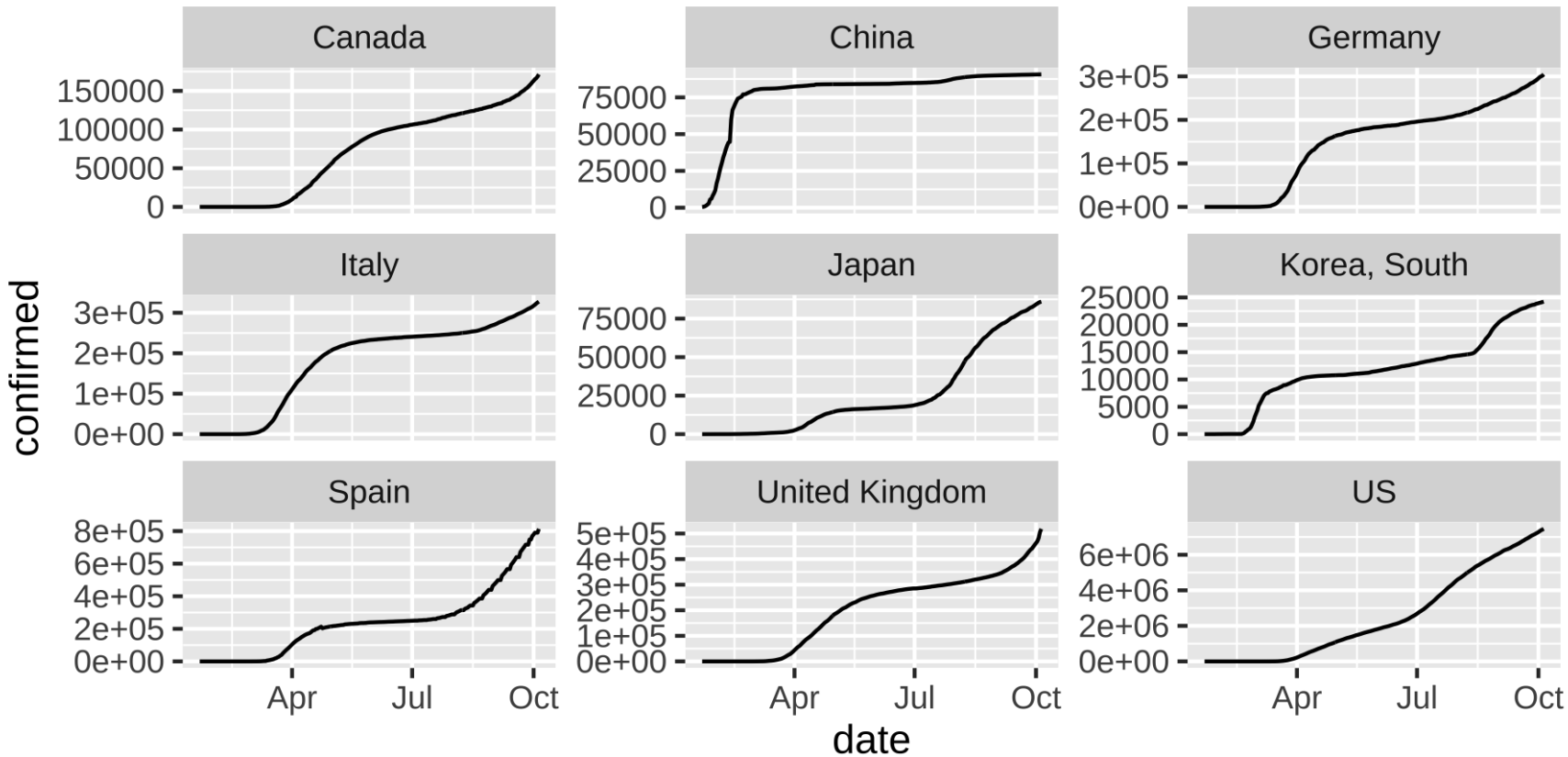


Why are the plots for Canada, China, and the UK so weird? Let's look at the data...

```
confirmed_global %>%
  count(country_or_region) %>%
  filter(n > 1)
```

```
## # A tibble: 7 x 2
##   country_or_region     n
##   <chr>             <int>
## 1 Australia             8
## 2 Canada               14
## 3 China                33
## 4 Denmark               3
## 5 France               11
## 6 Netherlands           5
## # … with 1 more row
```

```
confirmed_global_long %>%
  filter(country_or_region %in%
    c("US", "Canada", "China", "Japan", "Korea, South", "Italy", "Germany", "Spain", "United Kingdor
  group_by(country_or_region, date) %>% # <<
  summarize(confirmed = sum(confirmed)) %>% # <<
  ggplot(aes(x = date, y = confirmed)) +
  geom_line() +
  facet_wrap(~country_or_region, scales = "free_y")
```



Per Capita?

A data source: the World Bank.

```
reload_data <- FALSE
if (reload_data) {
  wbstats::wb_data("SP.POP.TOTL",
    mrnev = 1 # most recent non-empty value
  ) %>% write_csv("data/worldbank_sp_pop_totl.csv")
}

population <-
  read_csv(
    "data/worldbank_sp_pop_totl.csv",
    col_types = cols_only(
      iso2c = col_character(),
      iso3c = col_character(),
      country = col_character(),
      date = col_double(),
      SP.POP.TOTL = col_double(),
      footnote = col_character()
    )
  ) %>%
  select(iso2c, iso3c, country, population = SP.POP.TOTL)
```

```
population %>% reactable::reactable()
```

iso2c	iso3c	country	population
AW	ABW	Aruba	106314
AF	AFG	Afghanistan	38041754
AO	AGO	Angola	31825295
AL	ALB	Albania	2854191
AD	AND	Andorra	77142
AE	ARE	United Arab Emirates	9770529
AR	ARG	Argentina	44938712
AM	ARM	Armenia	2957731
AS	ASM	American Samoa	55312
AG	ATG	Antigua and Barbuda	97118

1–10 of 217 rows

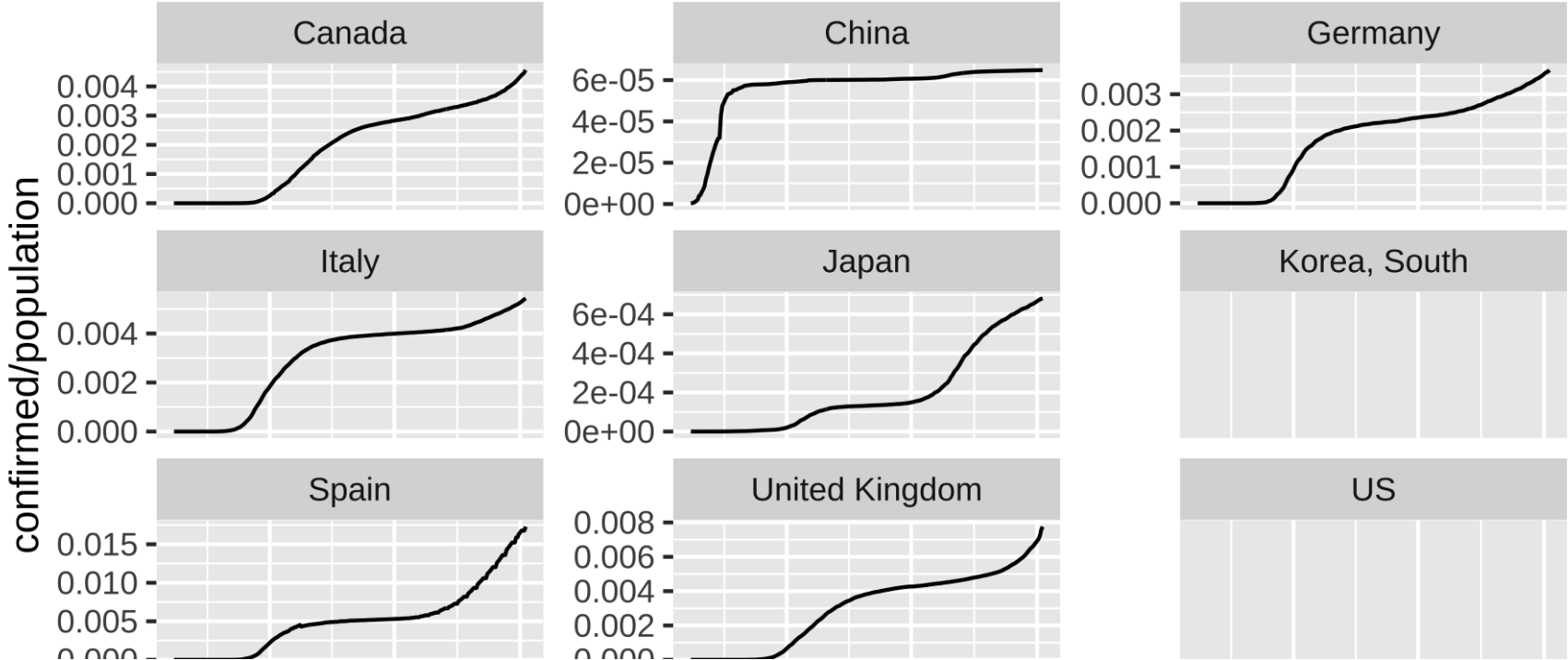
Previous **1** 2 3 4 5 ... 22 Next

Join cases table with population table

```
cases_with_population <- confirmed_global_long %>%
  rename(country = country_or_region) %>%
  left_join(
    population,
    by = "country"
  )
```

```
cases_with_population %>%
  filter(country %in%
    c("US", "Canada", "China", "Japan", "Korea, South", "Italy", "Germany", "Spain", "United Kingdor
  group_by(country, date, population) %>% # <<
  summarize(confirmed = sum(confirmed)) %>% # <<
  ggplot(aes(x = date, y = confirmed / population)) +
  geom_line() +
  facet_wrap(~country, scales = "free_y")
```

```
## Warning: Removed 258 row(s) containing missing values (geom_path).
```



Debugging a join

```
full_join_results <- confirmed_global_long %>%
  rename(country = country_or_region) %>%
  full_join( # <<
    population,
    by = "country"
  )
```

Countries with population but no case count:

```
full_join_results %>%
  filter(is.na(confirmed)) %>%
  distinct(country)
```

```
## # A tibble: 55 x 1
##   country
##   <chr>
## 1 Aruba
## 2 American Samoa
## 3 Bahamas, The
## 4 Bermuda
## 5 Brunei Darussalam
## 6 Channel Islands
## # … with 49 more rows
```

Countries with case count but no population:

```
full_join_results %>%
  filter(is.na(population)) %>%
  distinct(country)
```

```
## # A tibble: 26 x 1
##   country
##   <chr>
## 1 Bahamas
## 2 Brunei
## 3 Burma
## 4 Congo (Brazzaville)
## 5 Congo (Kinshasa)
## 6 Czechia
## # … with 20 more rows
```

Recoding

```
recoded_cases <-
  confirmed_global_long %>%
  mutate(country = case_when(
    country_or_region == "US" ~ "United States",
    country_or_region == "Russia" ~ "Russian Federation",
    country_or_region == "Korea, South" ~ "Korea, Rep.",
    TRUE ~ country_or_region
  ))
cases_with_population <- inner_join(
  recoded_cases %>% select(country, date, confirmed),
  population %>% select(country, population),
  by = "country"
)
```

```
cases_with_population %>%
  filter(country %in%
    c("United States", "Canada", "China", "Japan", "Korea, Rep.", "Italy", "Germany", "Spain", "Unit
  group_by(country, population, date) %>% # <<
  summarize(confirmed = sum(confirmed)) %>% # <<
  ggplot(aes(x = date, y = confirmed / population)) +
  geom_line() +
  facet_wrap(~country, scales = "fixed")
```