Tidying Data

DATA 202 21FA (thanks: datasciencebox.org)

Q&A

> Can we use actual maps instead of scatterplots?

Yes, we need a *base layer*. We'll focus more on spatial data in a few weeks.

Review of Exercise 5

Why not just grab `states$population`?

Pivoting

Data: Sales

We have...

```
# A tibble: 2 × 4
  customer_id item_1 item_2
        <int> <chr>  <chr>
1           1 bread  milk
2           2 milk   toilet pap
```

Data: Sales

We have...

```
# A tibble: 2 × 4
  customer_id item_1 item_2
        <int> <chr>  <chr>
1           1 bread  milk
2           2 milk   toilet pap
```

We want...

```
# A tibble: 6 × 3
  customer_id item_num item
        <int> <chr>    <chr>
1           1 item_1   bread
2           1 item_2   milk
3           1 item_3   banana
4           2 item_1   milk
5           2 item_2   toilet p
6           2 item_3   <NA>
```

A grammar of data tidying

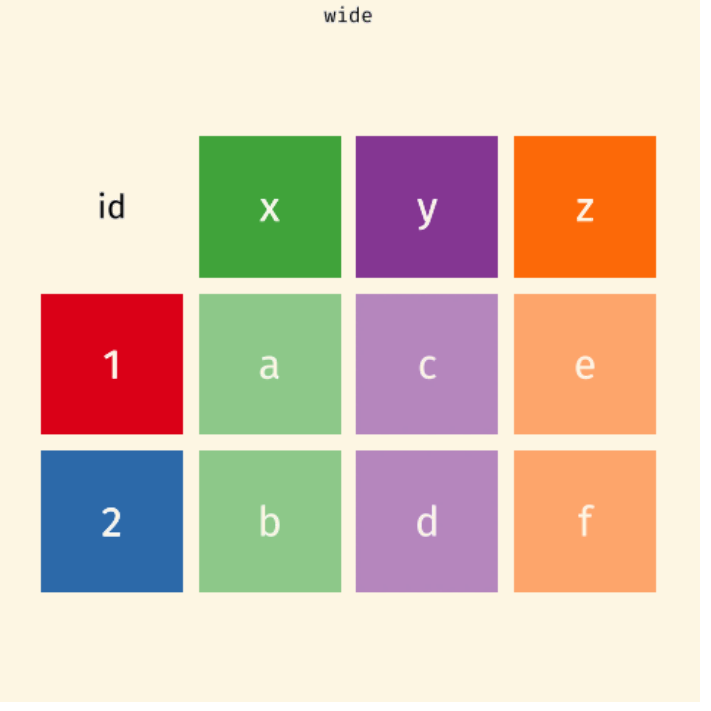
The goal of tidyr is to help you tidy your data via

- pivoting for going between wide and long data
- splitting and combining character columns
- nesting and unnesting columns
- clarifying how NAs should be treated

Pivoting data

Not this...

but this!



Wider vs. longer

wider

more columns

```
# A tibble: 2 × 4
  customer_id item_1 item_2
        <int> <chr>  <chr>
1           1 bread  milk
2           2 milk   toilet pap
```

Wider vs. longer

wider

more columns

```
# A tibble: 2 × 4
  customer_id item_1 item_2
        <int> <chr>  <chr>
1           1 bread  milk
2           2 milk   toilet pap
```

longer

more rows

```
# A tibble: 6 × 3
  customer_id item_num item
        <int> <chr>    <chr>
1           1 item_1   bread
2           1 item_2   milk
3           1 item_3   banana
4           2 item_1   milk
5           2 item_2   toilet p
6           2 item_3   <NA>
```

pivot_longer()

- `data` (as usual)

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  values_to = "value"
  )
```

pivot_longer()

- data (as usual)
- cols: columns to pivot into longer format

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  values_to = "value"
  )
```

`pivot_longer()`

- `data` (as usual)
- `cols`: columns to pivot into longer format
- `names_to`: name of the column where column names of pivoted variables go (character string)

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  values_to = "value"
  )
```

pivot_longer()

- data (as usual)
- cols: columns to pivot into longer format
- names_to: name of the column where column names of pivoted variables go (character string)
- values_to: name of the column where data in pivoted variables go (character string)

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  values_to = "value"
  )
```

Customers → purchases

```
purchases <- customers %>%
  pivot_longer(
    cols = item_1:item_3,  # variables item_1 to item_3
    names_to = "item_num", # column names -> new column called i
    values_to = "item"     # values in columns -> new column cal
    )

purchases
```

```
# A tibble: 6 × 3
  customer_id item_num item
        <int> <chr>    <chr>
1           1 item_1   bread
2           1 item_2   milk
3           1 item_3   banana
4           2 item_1   milk
5           2 item_2   toilet paper
6           2 item_3   <NA>
```

Why pivot?

Most likely, because the next step of your analysis needs it

Why pivot?

Most likely, because the next step of your analysis needs it

```
prices
```

```
# A tibble: 5 × 2
  item          price
  <chr>         <dbl>
1 avocado         0.5
2 banana          0.15
3 bread           1
4 milk            0.8
5 toilet paper    3
```

```
purchases %>%
  left_join(prices)
```

```
# A tibble: 6 × 4
  customer_id item_num item
        <int> <chr>    <chr>
1           1 item_1   bread
2           1 item_2   milk
3           1 item_3   banana
4           2 item_1   milk
5           2 item_2   toilet p
6           2 item_3   <NA>
```

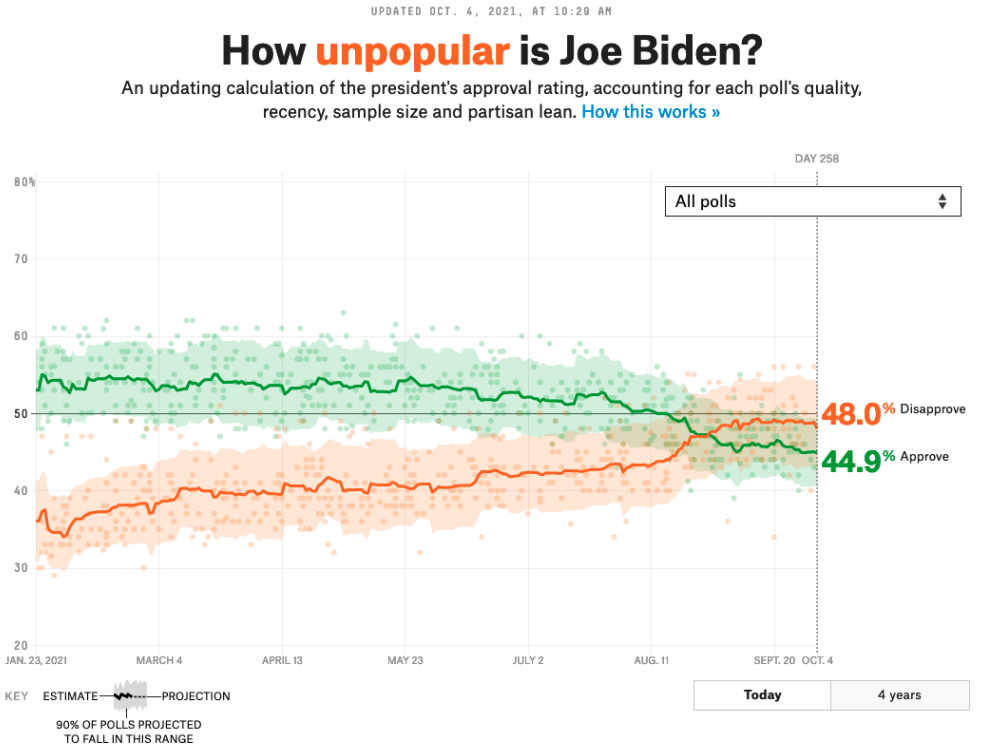
Purchases → customers

- `data` (long)
- `names_from`: tells us what column to put each value in
- `values_from`: tells us what to put in that column

```
purchases %>%
  pivot_wider(
    names_from = item_num,
    values_from = item
  )
```

```
# A tibble: 2 × 4
  customer_id item_1 item_2       item_3
        <int> <chr>  <chr>        <chr>
1           1 bread  milk         banana
2           2 milk   toilet paper <NA>
```

Case study: Biden Approval Rating



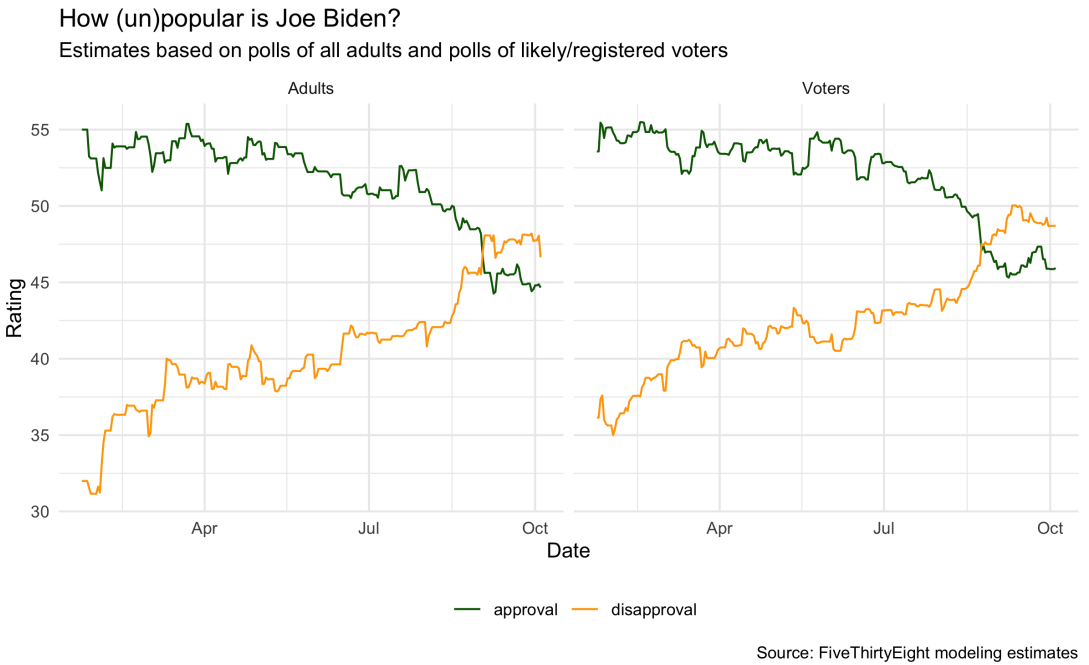
Source: FiveThirtyEight

Data

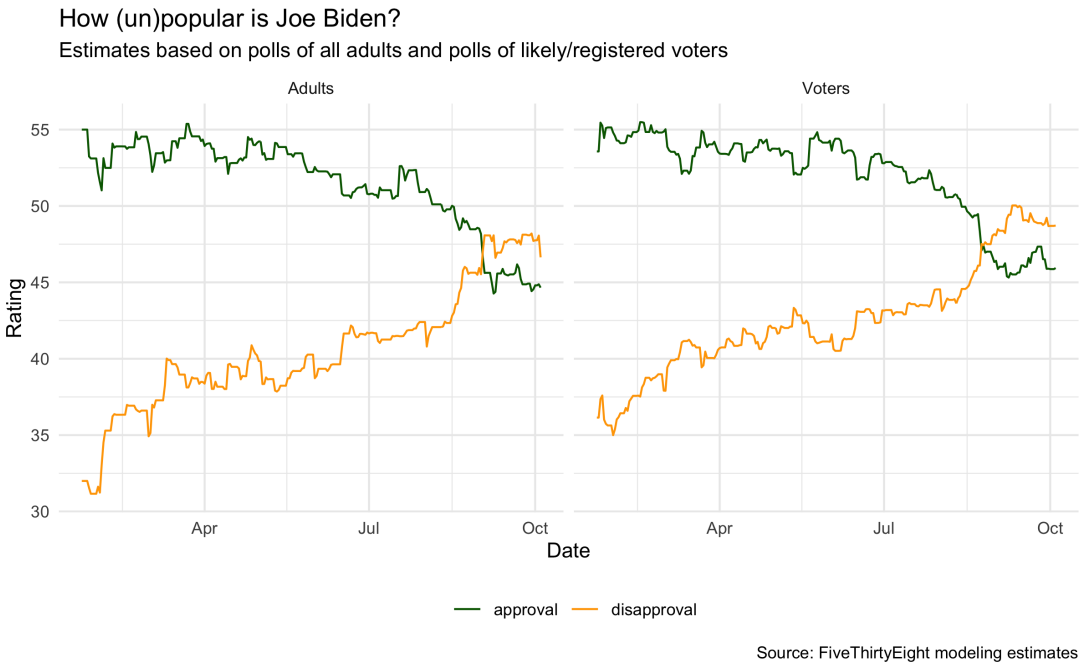
```
approval
```

```
# A tibble: 510 × 4
  subgroup date       approval disapproval
  <chr>    <date>        <dbl>       <dbl>
1 Voters   2021-10-04     45.9        48.7
2 Adults   2021-10-04     44.7        46.6
3 Voters   2021-10-03     45.9        48.7
4 Adults   2021-10-03     44.9        48.1
5 Adults   2021-10-02     44.8        47.8
6 Voters   2021-10-02     45.9        48.7
# … with 504 more rows
```

Goal


How (un)popular is Joe Biden?
Estimates based on polls of all adults and polls of likely/registered voters
Adults
Voters
Rating
55
50
45
40
35
30
Apr
Jul
Oct
Date
approval
disapproval
Source: FiveThirtyEight modeling estimates

Goal



Aesthetic mappings:

✅ x = `date`

❌ y = `rating_value`

❌ color = `rating_type`

Facet:

✅ `subgroup` (Adults and Voters)

Pivot

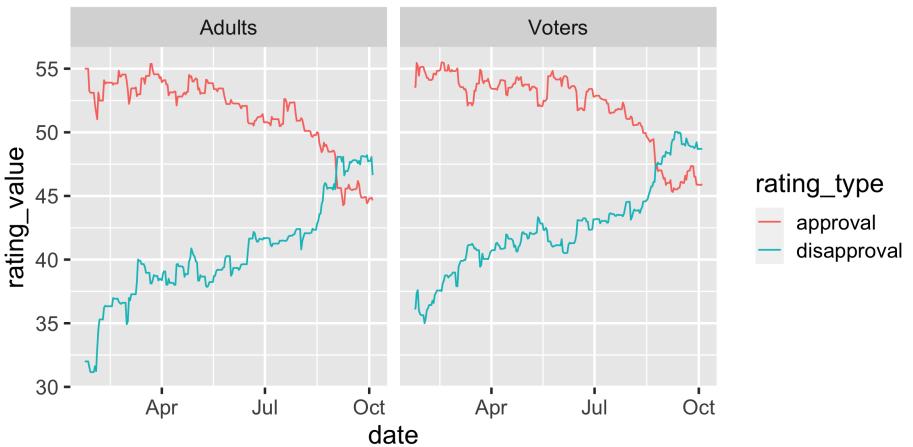
```
approval_longer <- approval %>%
  pivot_longer(
    cols = c(approval, disapproval),
    names_to = "rating_type",
    values_to = "rating_value"
  )

approval_longer
```

```
# A tibble: 1,020 × 4
  subgroup date       rating_type rating_value
  <chr>    <date>     <chr>              <dbl>
1 Voters   2021-10-04 approval            45.9
2 Voters   2021-10-04 disapproval         48.7
3 Adults   2021-10-04 approval            44.7
4 Adults   2021-10-04 disapproval         46.6
5 Voters   2021-10-03 approval            45.9
6 Voters   2021-10-03 disapproval         48.7
# … with 1,014 more rows
```

Plot

```
ggplot(approval_longer,
       aes(x = date, y = rating_value, color = rating_type)) +
  geom_line() +
  facet_wrap(vars(subgroup))
```



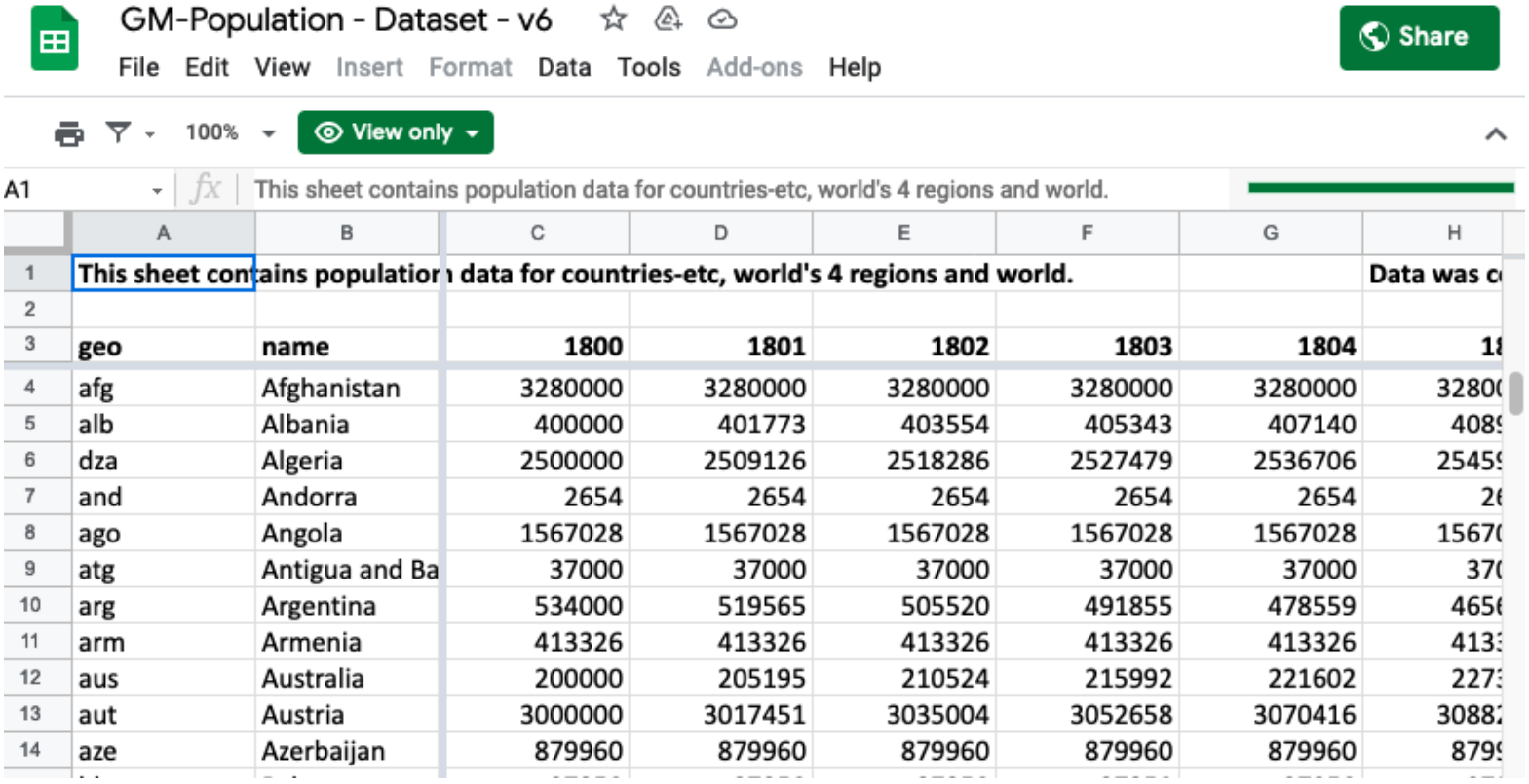
Code Plot

```
ggplot(approval_longer,
       aes(x = date, y = rating_value,
           color = rating_type)) +
  geom_line() +
  facet_wrap(vars(subgroup)) +
  scale_color_manual(values = c("darkgreen", "orange")) +
  labs(
    x = "Date", y = "Rating",
    color = NULL,
    title = "How (un)popular is Joe Biden?",
    subtitle = "Estimates based on polls of all adults and polls
    caption = "Source: FiveThirtyEight modeling estimates"
  )
```

Code Plot

```
ggplot(approval_longer,
       aes(x = date, y = rating_value,
           color = rating_type)) +
  geom_line() +
  facet_wrap(vars(subgroup)) +
  scale_color_manual(values = c("darkgreen", "orange")) +
  labs(
    x = "Date", y = "Rating",
    color = NULL,
    title = "How (un)popular is Joe Biden??",
    subtitle = "Estimates based on polls of all adults and polls
    caption = "Source: FiveThirtyEight modeling estimates"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

Case study: Gapminder



GM-Population - Dataset - v6

File Edit View Insert Format Data Tools Add-ons Help

Share

100% View only

A1 | This sheet contains population data for countries-etc, world's 4 regions and world.

	A	B	C	D	E	F	G	H
1	**This sheet con**	**ains populatio**	**data for countries-etc, world's 4 regions and world.**					**Data was c**
2								
3	**geo**	**name**	**1800**	**1801**	**1802**	**1803**	**1804**	**18**
4	afg	Afghanistan	3280000	3280000	3280000	3280000	3280000	3280(
5	alb	Albania	400000	401773	403554	405343	407140	408
6	dza	Algeria	2500000	2509126	2518286	2527479	2536706	2545
7	and	Andorra	2654	2654	2654	2654	2654	2(
8	ago	Angola	1567028	1567028	1567028	1567028	1567028	1567(
9	atg	Antigua and Ba	37000	37000	37000	37000	37000	37(
10	arg	Argentina	534000	519565	505520	491855	478559	465(
11	arm	Armenia	413326	413326	413326	413326	413326	413
12	aus	Australia	200000	205195	210524	215992	221602	227
13	aut	Austria	3000000	3017451	3035004	3052658	3070416	3088
14	aze	Azerbaijan	879960	879960	879960	879960	879960	879

Source: https://www.gapminder.org/data/documentation/gd003/

We want...

```
gm_pop_wide <- read_csv("data/GM-Population - Dataset - v6 - dat
gm_pop_wide %>% distinct(geo, name)
```

```
# A tibble: 204 × 2
  geo   name
  <chr> <chr>
1 afg   Afghanistan
2 alb   Albania
3 dza   Algeria
4 and   Andorra
5 ago   Angola
6 atg   Antigua and Barbuda
# … with 198 more rows
```

```
gm_pop_long <- gm_pop_wide %>%
  pivot_longer(
    cols = -c(geo, name),
    names_to = "year",
    values_to = "population"
  ) %>%
  mutate(
```

Case Study: is college worth it?

```
# https://www.insidehighered.com/news/2019/06/10/new-data-show-e
# https://www.newyorkfed.org/medialibrary/media/research/college
labor_fed <- read_csv("data/college-labor-data.csv", show_col_ty
labor_fed
```

```
# A tibble: 31 × 5
  Date   `Bachelor's degree… `Bachelor's degr… `Bachelor's degre…
  <chr>                <dbl>             <dbl>              <dbl>
1 1/1/90               35369             46311              59434
2 1/1/91               33077             43723              57030
3 1/1/92               33208             44277              55347
4 1/1/93               32250             43000              55542
5 1/1/94               31303             41912              54137
6 1/1/95               29727             40769              52660
# … with 25 more rows, and 1 more variable:
#   High school diploma: median <dbl>
```

```
labor_fed %>%
  pivot_longer(
    cols = -Date,
    names_to = "education",
    values_to = "wage"
  ) %>%
  separate(education, into = c("degree", "measure"), sep = ": ")
```