Small Multiples

Kieran Healy

February 28, 2024

Load our libraries

```
library(here)      # manage file paths
library(tidyverse) # your friend and mine
library(gapminder)  # gapminder data

## The astonishingly useful plot composer
library(patchwork)

## Data Visualization book: https://kjhealy.github.io/socviz
## install.packages("socviz")
library(socviz)    # data and some useful functions

## A COVID data package: https://kjhealy.github.io/covdata
## remotes::install_github("kjhealy/covdata")
library(covdata)
```

```
Attaching package: 'covdata'

The following object is masked from 'package:socviz':

    %nin%

The following object is masked from 'package:datasets':

    uspop
```

Grouped data and the `group` aesthetic

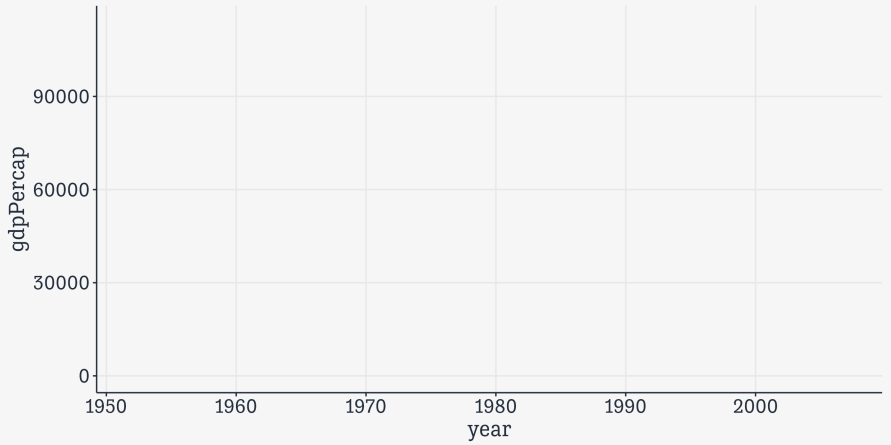
Try to make a lineplot

```
gapminder
```

```
# A tibble: 1,704 × 6
   country     continent  year lifeExp      pop
gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>
<dbl>
 1 Afghanistan Asia       1952    28.8  8425333
779.
 2 Afghanistan Asia       1957    30.3  9240934
821.
 3 Afghanistan Asia       1962    32.0 10267083
853.
 4 Afghanistan Asia       1967    34.0 11537966
836.
 5 Afghanistan Asia       1972    36.1 13079460
740.
 6 Afghanistan Asia       1977    38.4 14880372
786.
 7 Afghanistan Asia       1982    39.9 12881816
978.
```

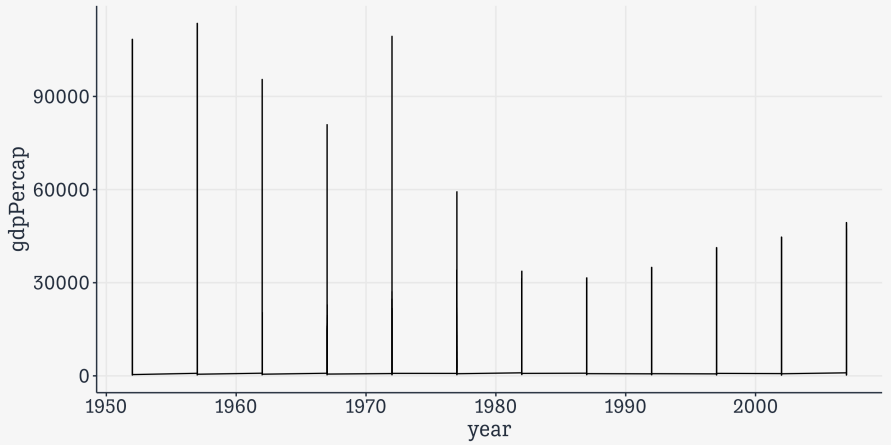
Try to make a lineplot

```
gapminder ▷
  ggplot(data = gapminder,
           mapping = aes(x = year,
                         y = gdpPercap))
```



Try to make a lineplot

```
gapminder ▷
  ggplot(data = gapminder,
           mapping = aes(x = year,
                         y = gdpPercap)) +
  geom_line()
```



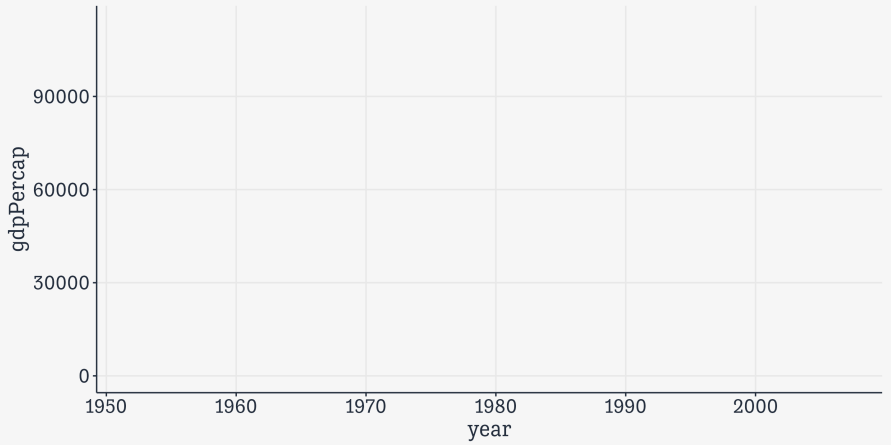
Try to make a lineplot

```
gapminder
```

```
# A tibble: 1,704 × 6
   country     continent  year lifeExp      pop
gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>
<dbl>
 1 Afghanistan Asia       1952    28.8  8425333
779.
 2 Afghanistan Asia       1957    30.3  9240934
821.
 3 Afghanistan Asia       1962    32.0 10267083
853.
 4 Afghanistan Asia       1967    34.0 11537966
836.
 5 Afghanistan Asia       1972    36.1 13079460
740.
 6 Afghanistan Asia       1977    38.4 14880372
786.
 7 Afghanistan Asia       1982    39.9 12881816
978.
```

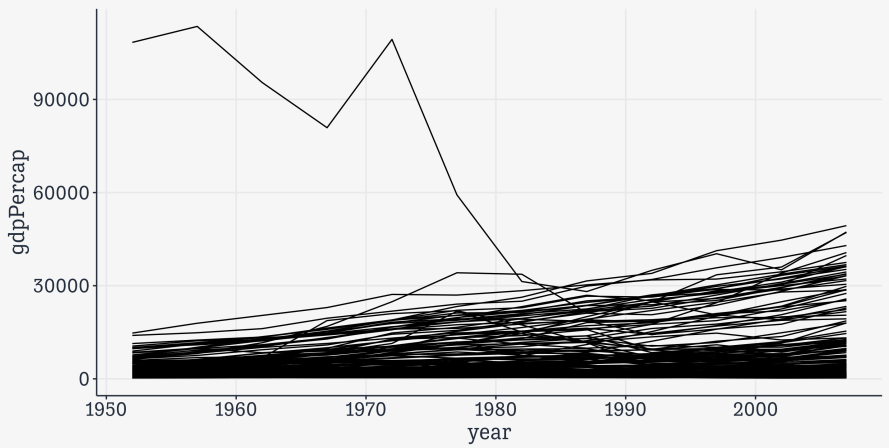
Try to make a lineplot

```
gapminder ▷
  ggplot(mapping = aes(x = year,
                       y = gdpPercap))
```



Try to make a lineplot

```
gapminder ▷
  ggplot(mapping = aes(x = year,
                       y = gdpPercap)) +
  geom_line(mapping = aes(group = country))
```



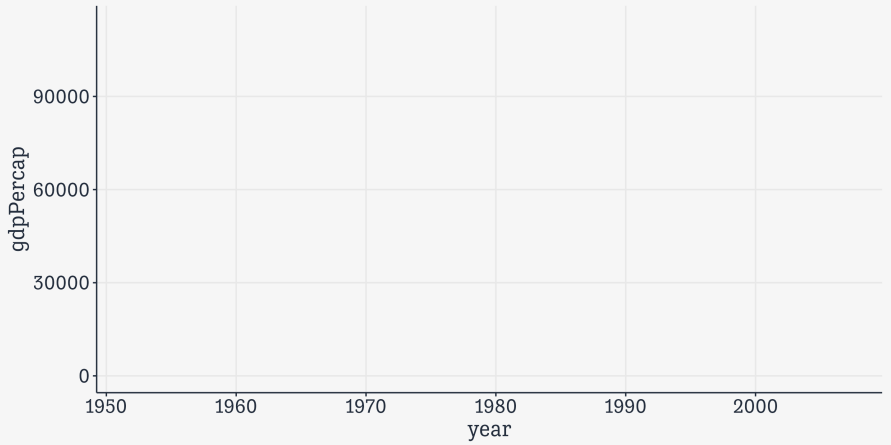
Facet the plot

```
gapminder
```

```
# A tibble: 1,704 × 6
   country     continent  year lifeExp      pop
gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>
<dbl>
 1 Afghanistan Asia       1952    28.8  8425333
779.
 2 Afghanistan Asia       1957    30.3  9240934
821.
 3 Afghanistan Asia       1962    32.0 10267083
853.
 4 Afghanistan Asia       1967    34.0 11537966
836.
 5 Afghanistan Asia       1972    36.1 13079460
740.
 6 Afghanistan Asia       1977    38.4 14880372
786.
 7 Afghanistan Asia       1982    39.9 12881816
978.
```

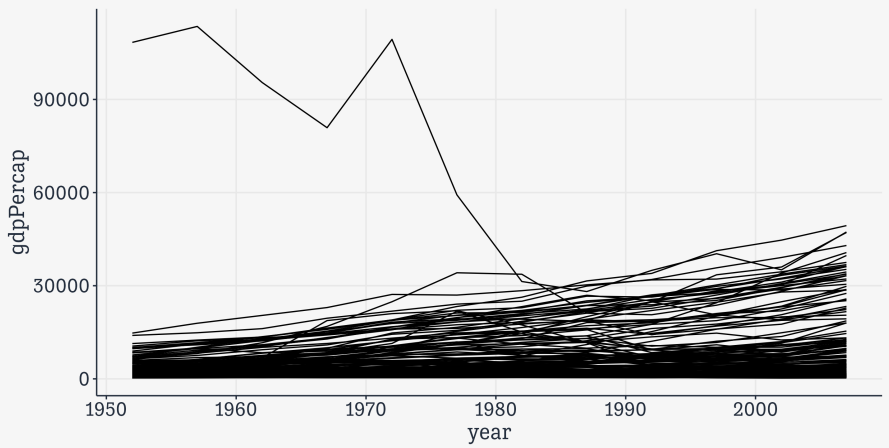
Facet the plot

```
gapminder ▷
  ggplot(mapping =
           aes(x = year,
           y = gdpPercap))
```



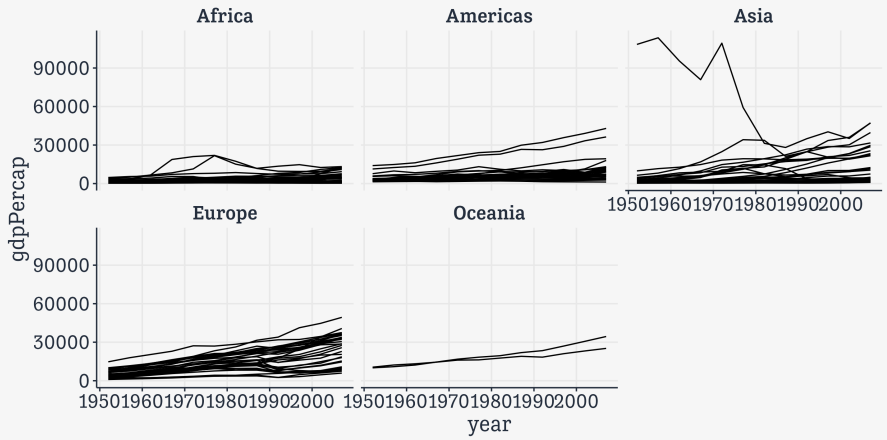
Facet the plot

```
gapminder ▷
  ggplot(mapping =
           aes(x = year,
           y = gdpPercap)) +
  geom_line(mapping = aes(group = country))
```



Facet the plot

```
gapminder ▷
  ggplot(mapping =
           aes(x = year,
           y = gdpPercap)) +
  geom_line(mapping = aes(group = country)) +
  facet_wrap(~ continent)
```



Faceting is very powerful

Faceting

A facet is not a geom; it's a way of arranging repeated geoms by some additional variable

Facets use R's "formula" syntax: `facet_wrap(~ continent)`

Read the **~** as "on" or "by"

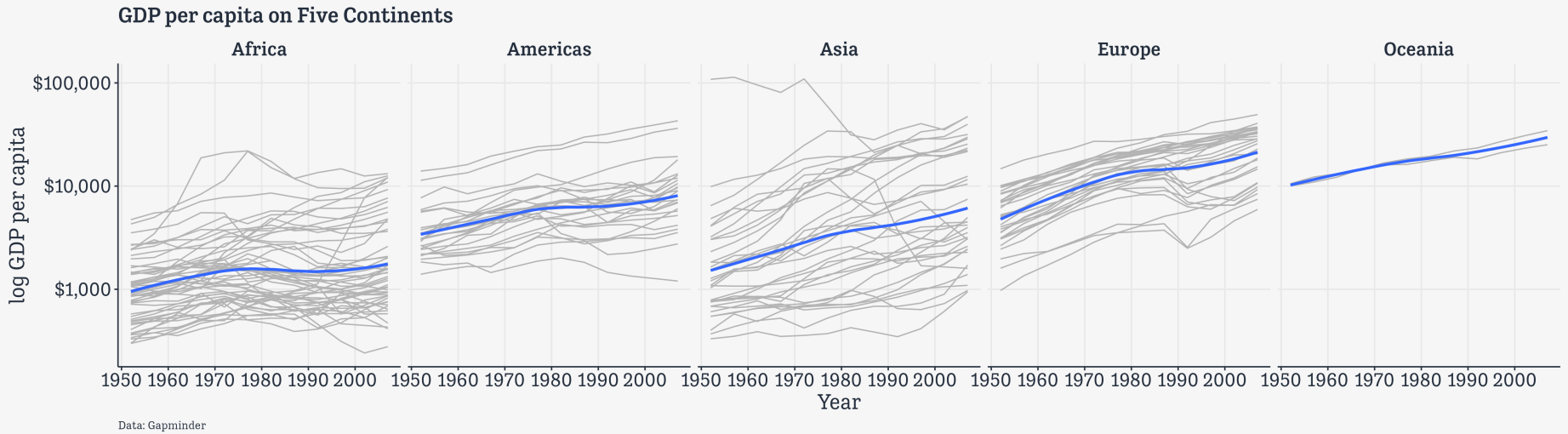
Faceting

You can also use this syntax: `facet_wrap(vars(continent))`

This is newer, and consistent with other ways of referring to variables within tidyverse functions.

Facets in action

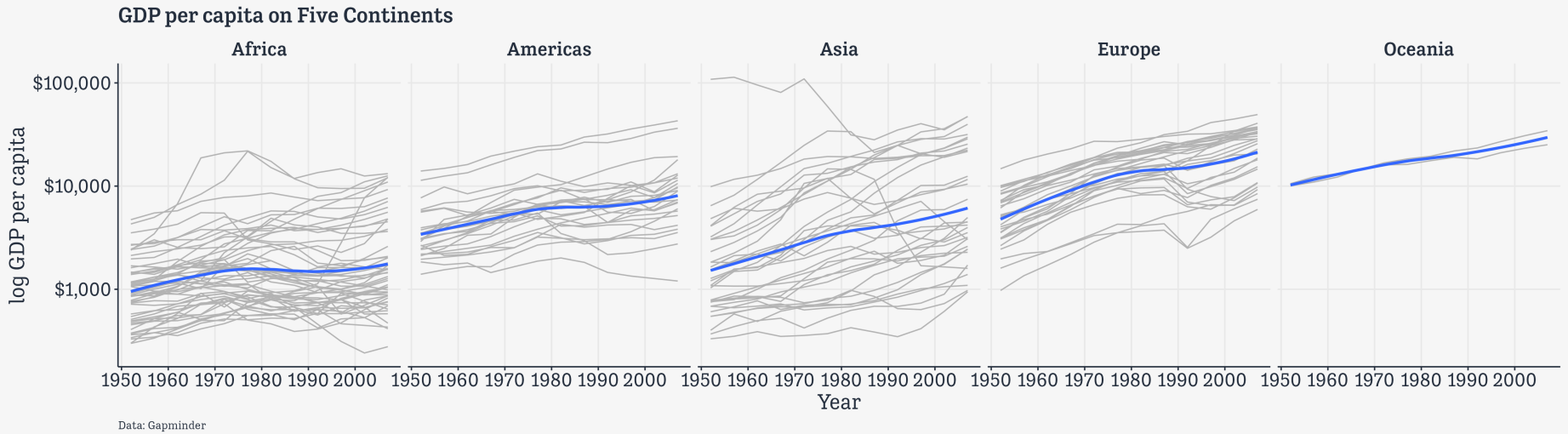
```
p ← ggplot(data = gapminder,
           mapping = aes(x = year,
                         y = gdpPercap))

p_out ← p + geom_line(color="gray70",
              mapping=aes(group = country)) +
    geom_smooth(size = 1.1,
                method = "loess",
                se = FALSE) +
    scale_y_log10(labels=scales::label_dollar()) +
    facet_wrap(~ continent, ncol = 5) +
    labs(x = "Year",
         y = "log GDP per capita",
         title = "GDP per capita on Five Continents",
         caption = "Data: Gapminder")
```



A more polished faceted plot.

Reorder your facets

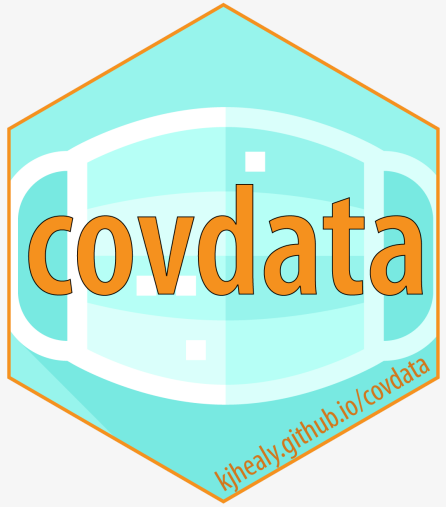
```
p ← ggplot(data = gapminder,
           mapping = aes(x = year,
                         y = gdpPercap))

p_out ← p + geom_line(color="gray70",
              mapping=aes(group = country)) +
    geom_smooth(size = 1.1,
                method = "loess",
                se = FALSE) +
    scale_y_log10(labels=scales::label_dollar()) +
    facet_wrap(~ reorder(continent, gdpPercap), ncol = 5) +
    labs(x = "Year",
         y = "log GDP per capita",
         title = "GDP per capita on Five Continents",
         caption = "Data: Gapminder")
```



Reorder by the mean of GDP per capita.

The covdata package

```
library(covdata)
```



A data package that collects and bundles datasets related to the COVID-19 pandemic from a variety of sources.

Cross-national case and mortality data; data for states and counties in the U.S.; mobility data from Apple; other material.

Let’s take a look

```
stmf
```

```
# A tibble: 580,395 × 17
   country_code cname    iso2  continent iso3   year  week sex   split split_sex
   <chr>        <chr>    <chr> <chr>     <chr> <dbl> <dbl> <chr> <dbl>     <dbl>
 1 AUS          Austral… AU    Oceania   AUS    2015     1 m         1         0
 2 AUS          Austral… AU    Oceania   AUS    2015     1 m         1         0
 3 AUS          Austral… AU    Oceania   AUS    2015     1 m         1         0
 4 AUS          Austral… AU    Oceania   AUS    2015     1 m         1         0
 5 AUS          Austral… AU    Oceania   AUS    2015     1 m         1         0
 6 AUS          Austral… AU    Oceania   AUS    2015     1 f         1         0
 7 AUS          Austral… AU    Oceania   AUS    2015     1 f         1         0
 8 AUS          Austral… AU    Oceania   AUS    2015     1 f         1         0
 9 AUS          Austral… AU    Oceania   AUS    2015     1 f         1         0
10 AUS          Austral… AU    Oceania   AUS    2015     1 f         1         0
# ℹ 580,385 more rows
# ℹ 7 more variables: forecast <dbl>, approx_date <date>, age_group <chr>,
#   death_count <dbl>, death_rate <dbl>, deaths_total <dbl>, rate_total <dbl>
```

Let’s take a look

Short Term Mortality Fluctuations (STMF) data series

Source: `R/data.R`

Human Mortality Database (HMD) series of weekly death counts across countries.

```
stmf
```

Format

A tibble with 537,240 rows and 17 variables:

`country_code`
: Mortality database country code

`cname`
: character Country name

`iso2`
: character ISO2 country code

`iso3`

What we're interested in

```
stmf ▷
  filter(sex == "b", year > 2014 & year < 2020) ▷
  select(cname, iso3, year:sex, age_group, death_rate, rate_total)
```

```
# A tibble: 49,310 × 8
   cname     iso3   year  week sex   age_group death_rate rate_total
   <chr>     <chr> <dbl> <dbl> <chr> <chr>          <dbl>      <dbl>
 1 Australia AUS    2015     1 b     0-14        0.000360    0.00639
 2 Australia AUS    2015     1 b     15-64       0.00184     0.00639
 3 Australia AUS    2015     1 b     65-74       0.0112      0.00639
 4 Australia AUS    2015     1 b     75-84       0.0378      0.00639
 5 Australia AUS    2015     1 b     85+         0.127       0.00639
 6 Australia AUS    2015     2 b     0-14        0.000334    0.00605
 7 Australia AUS    2015     2 b     15-64       0.00172     0.00605
 8 Australia AUS    2015     2 b     65-74       0.0108      0.00605
 9 Australia AUS    2015     2 b     75-84       0.0341      0.00605
10 Australia AUS    2015     2 b     85+         0.124       0.00605
# ℹ 49,300 more rows
```

Mortality rankings: mean and max

```
rate_rank ← stmf ▷
  filter(sex == "b", year > 2014 & year < 2020) ▷
  group_by(country_code) ▷
  summarize(mean_rate = mean(rate_total, na.rm = TRUE)) ▷
  mutate(rate_rank = rank(mean_rate))


rate_max_rank ← stmf ▷
  filter(sex == "b", year == 2020) ▷
  group_by(country_code) ▷
  summarize(covid_max = max(rate_total, na.rm = TRUE)) ▷
  mutate(covid_max_rank = rank(covid_max))
```

Mortality rankings: mean and max

```
rate_rank
```

```
# A tibble: 38 × 3
   country_code mean_rate rate_rank
   <chr>            <dbl>     <dbl>
 1 AUS            0.00653         5
 2 AUT            0.00923        19
 3 BEL            0.00960        21
 4 BGR            0.0153         38
 5 CAN            0.00750         9
 6 CHE            0.00790        11
 7 CHL            0.00604         3
 8 CZE            0.0104         26
 9 DEUTNP         0.0113         30
10 DNK            0.00928        20
# ℹ 28 more rows
```

```
rate_max_rank
```

```
# A tibble: 38 × 3
   country_code covid_max covid_max_rank
   <chr>            <dbl>          <dbl>
 1 AUS            0.00707              3
 2 AUT            0.0149              18
 3 BEL            0.0193              26
 4 BGR            0.0356              38
 5 CAN            0.00950              8
 6 CHE            0.0133              15
 7 CHL            0.0116              12
 8 CZE            0.0206              31
 9 DEUTNP         0.0160              22
10 DNK            0.0114              11
# ℹ 28 more rows
```

Select the columns; merge the ranks

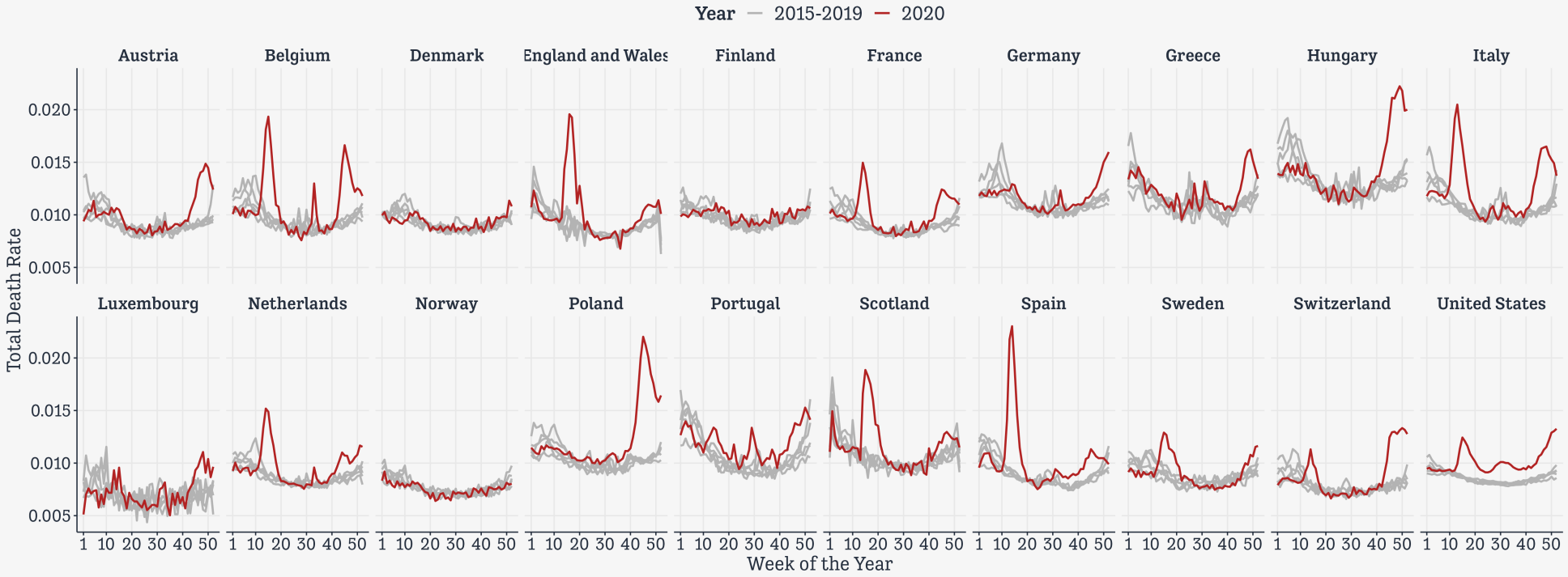
```
df ← stmf ▷
  filter(sex == "b", year > 2014, year < 2021,
         country_code %in% c("AUT", "BEL", "CHE", "DEUTNP", "DNK", "ESP", "FIN", "FRATNP", "GBR_SCO", "GBRTENW",
                             "GRC", "HUN", "ITA", "LUX", "POL", "NLD", "NOR", "PRT", "SWE", "USA")) ▷
  filter(!(year == 2020 & week > 53)) ▷
  group_by(cname, year, week) ▷
  mutate(yr_ind = year %in% 2020) ▷
  slice(1) ▷
  left_join(rate_rank, by = "country_code") ▷
  left_join(rate_max_rank, by = "country_code")

df
```

```
# A tibble: 6,279 × 22
# Groups:   cname, year, week [6,279]
   country_code cname   iso2  continent iso3   year  week sex   split split_sex
   <chr>        <chr>   <chr> <chr>     <chr> <dbl> <dbl> <chr> <dbl>     <dbl>
 1 AUT          Austria AT    Europe    AUT    2015     1 b         0         0
 2 AUT          Austria AT    Europe    AUT    2015     2 b         0         0
 3 AUT          Austria AT    Europe    AUT    2015     3 b         0         0
 4 AUT          Austria AT    Europe    AUT    2015     4 b         0         0
 5 AUT          Austria AT    Europe    AUT    2015     5 b         0         0
 6 AUT          Austria AT    Europe    AUT    2015     6 b         0         0
 7 AUT          Austria AT    Europe    AUT    2015     7 b         0         0
 8 AUT          Austria AT    Europe    AUT    2015     8 b         0         0
 9 AUT          Austria AT    Europe    AUT    2015     9 b         0         0
10 AUT          Austria AT    Europe    AUT    2015    10 b         0         0
# ℹ 6,269 more rows
# ℹ 12 more variables: forecast <dbl>, approx_date <date>, age_group <chr>,
#   death_count <dbl>, death_rate <dbl>, deaths_total <dbl>, rate_total <dbl>,
#   yr_ind <lgl>, mean_rate <dbl>, rate_rank <dbl>, covid_max <dbl>,
#   covid_max_rank <dbl>
```

And make a plot

```
p_out ← df ▷
  ggplot(aes(x = week, y = rate_total, color = yr_ind, group = year)) +
  scale_color_manual(values = c("gray70", "firebrick"), labels = c("2015-2019", "2020")) +
  scale_x_continuous(limits = c(1, 52),
                     breaks = c(1, seq(10, 50, 10)),
                     labels = as.character(c(1, seq(10, 50, 10)))) +
  facet_wrap(~ cname, ncol = 10) +
  geom_line(linewidth = 0.9) +
  labs(x = "Week of the Year", y = "Total Death Rate",
       color = "Year", title = "Overall Weekly Death Rates",
       caption = "Graph: @kjhealy. Data: Human Mortality Database, mortality.org")
```

Overall Weekly Death Rates



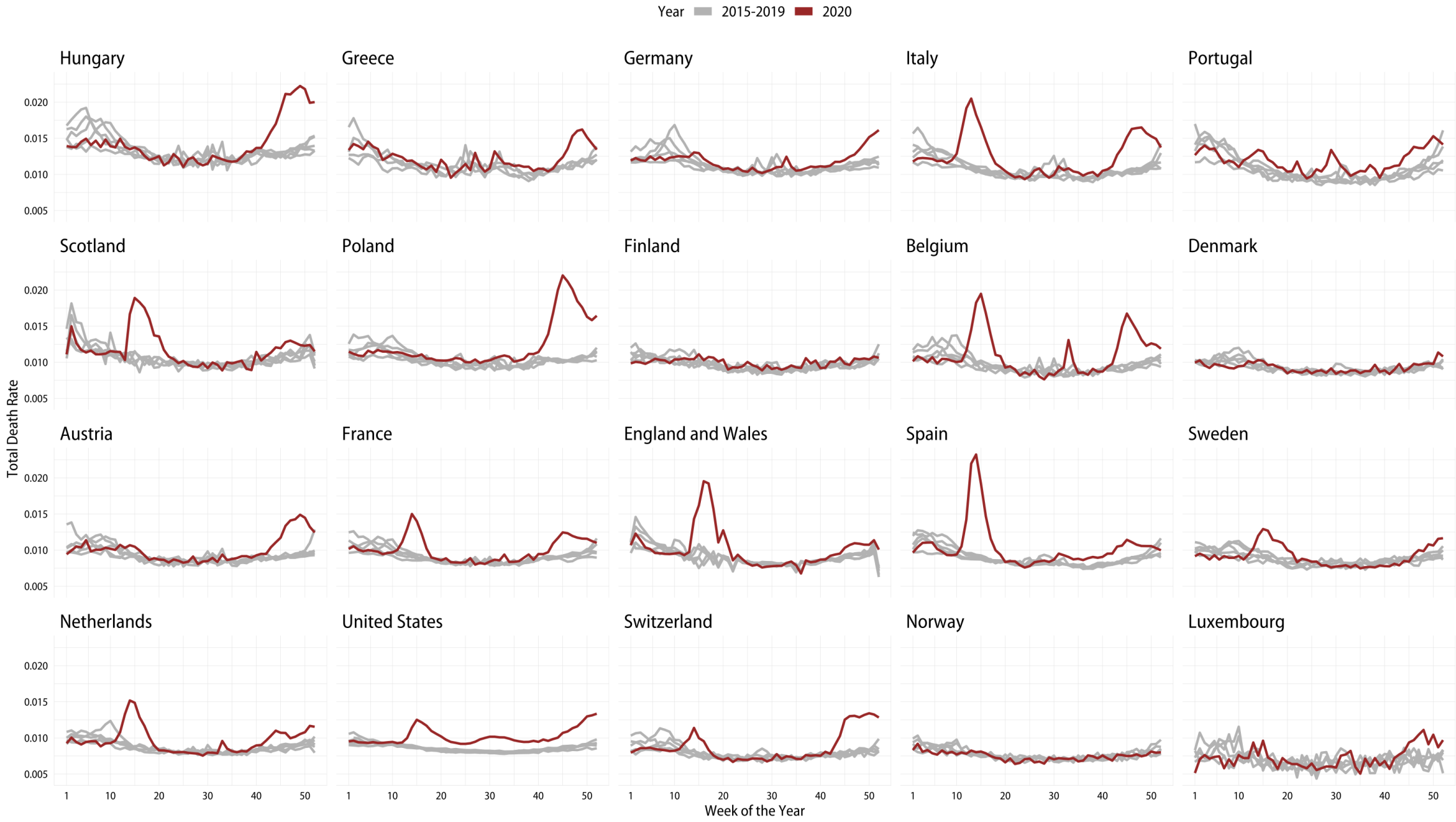
Graph: @kjhealy. Data: Human Mortality Database, mortality.org

Reorder the facets

```
p_out ← df ▷
  ggplot(aes(x = week, y = rate_total, color = yr_ind, group = year)) +
  scale_color_manual(values = c("gray70", "firebrick"), labels = c("2015-2019", "2020")) +
  scale_x_continuous(limits = c(1, 52),
                     breaks = c(1, seq(10, 50, 10)),
                     labels = as.character(c(1, seq(10, 50, 10)))) +
  geom_line(linewidth = 0.9) +
  facet_wrap(~ reorder(cname, rate_rank, na.rm = TRUE), ncol = 10) +
  labs(x = "Week of the Year", y = "Total Death Rate",
       color = "Year", title = "Overall Weekly Death Rates",
       caption = "Graph: @kjhealy. Data: Human Mortality Database, mortality.org")
```

Overall Weekly Death Rates

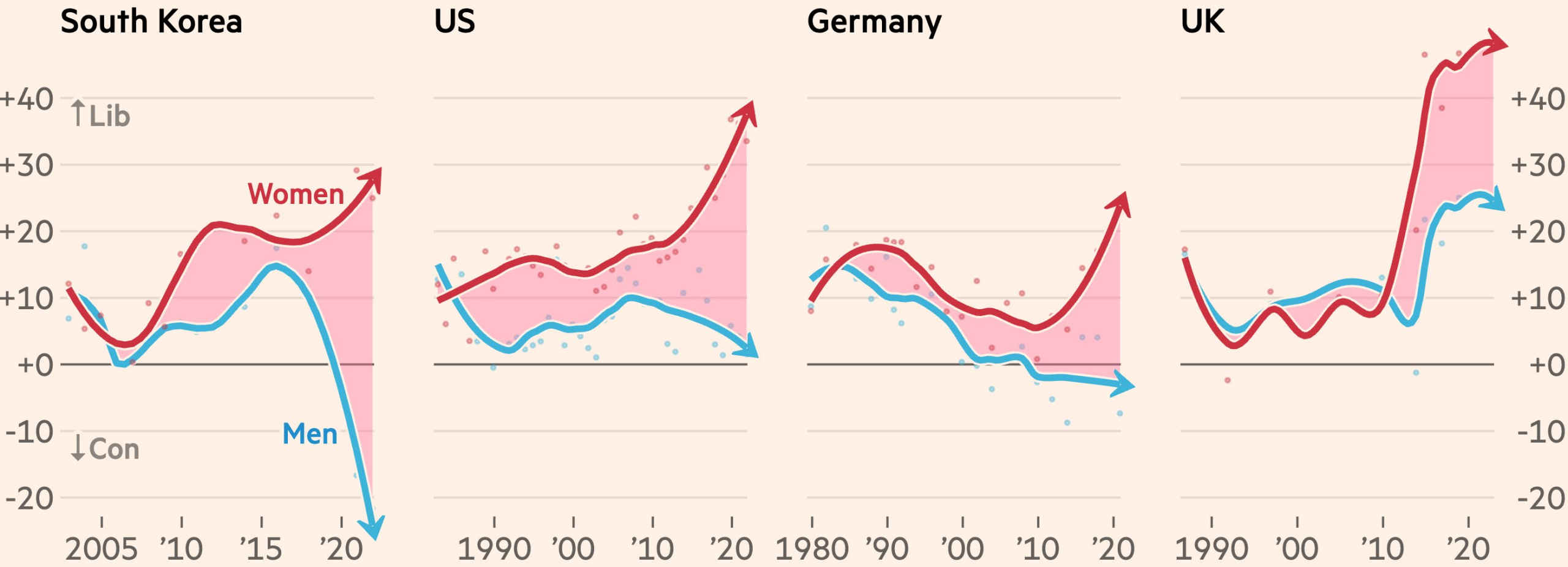
Comparing 2020 with 2015-2019 across selected countries. Countries are shown top left to bottom right ordered from highest to lowest average mortality rate in 2015-2019.



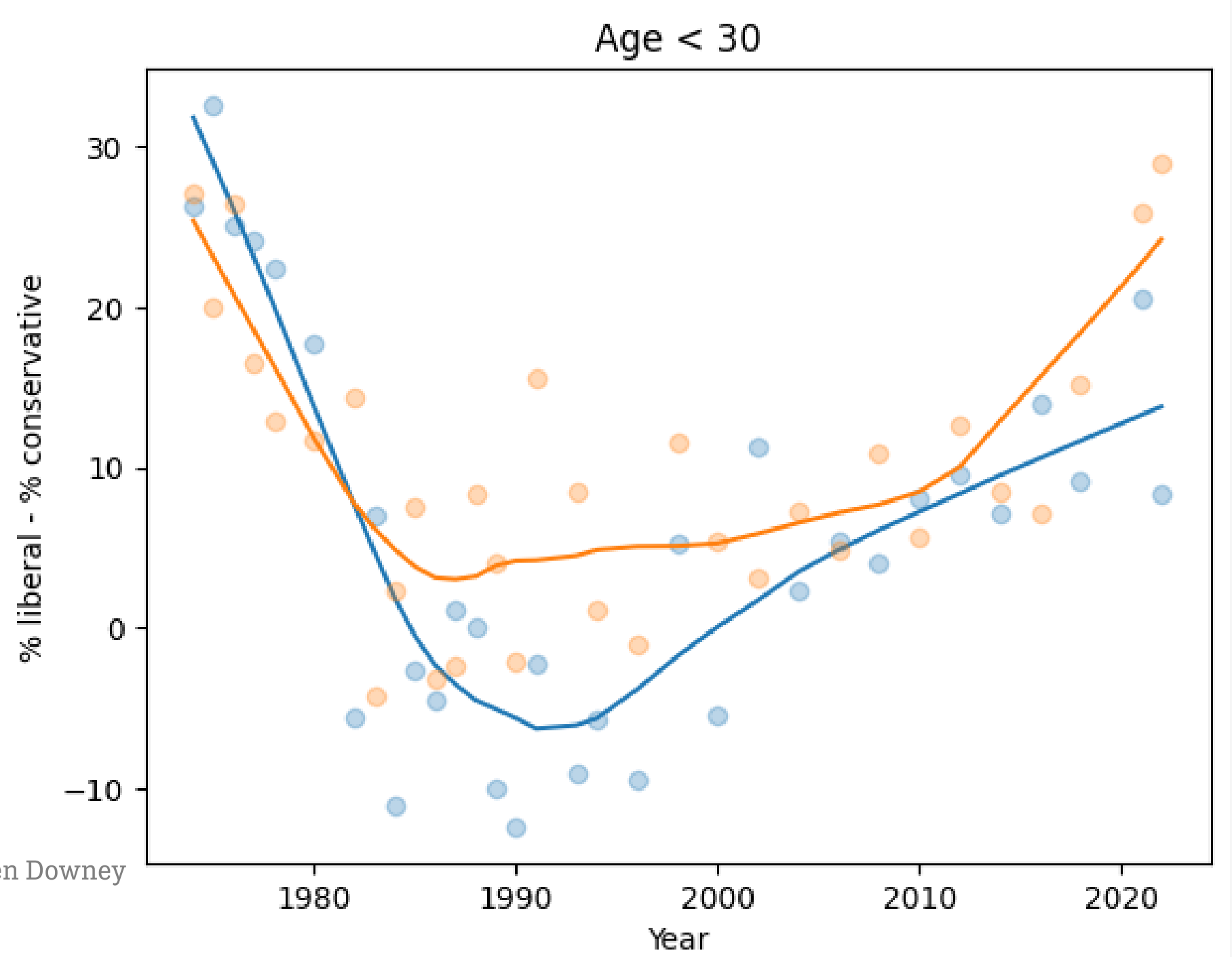
Graph: @kjhealy. Data: Human Mortality Database, mortality.org

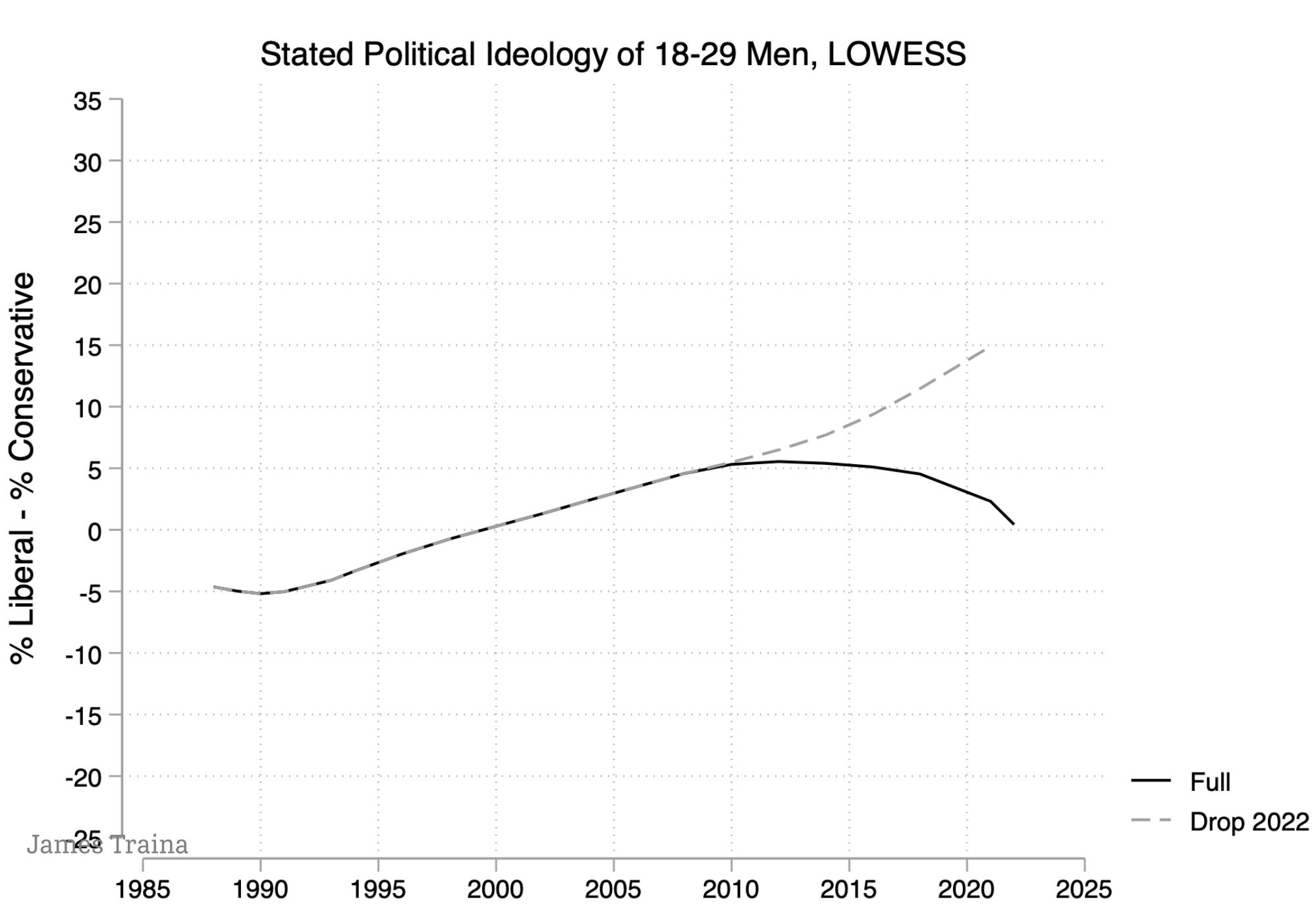
A wide ideology gap is opening up between young men and women in countries across the world

Political ideology of 18-29s (% liberal minus % conservative), by sex



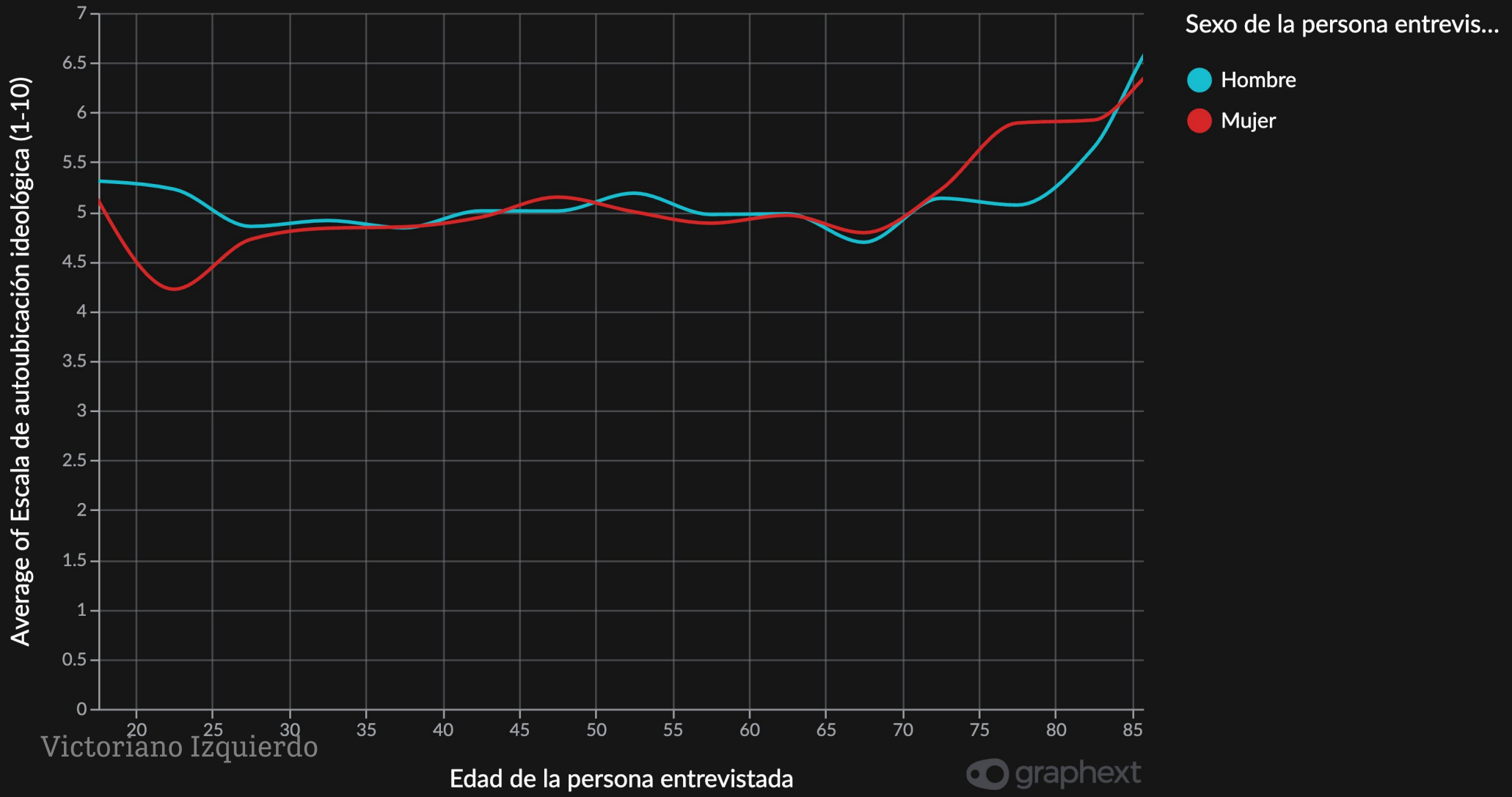
Sources: Daniel Cox, Survey Center on American Life; Gallup Poll Social Series; FT analysis of General Social Surveys of Korea, Germany & US and the British Election Study. US data is respondent's stated ideology. Other countries show support for liberal and conservative parties
All figures are adjusted for time trend in the overall population

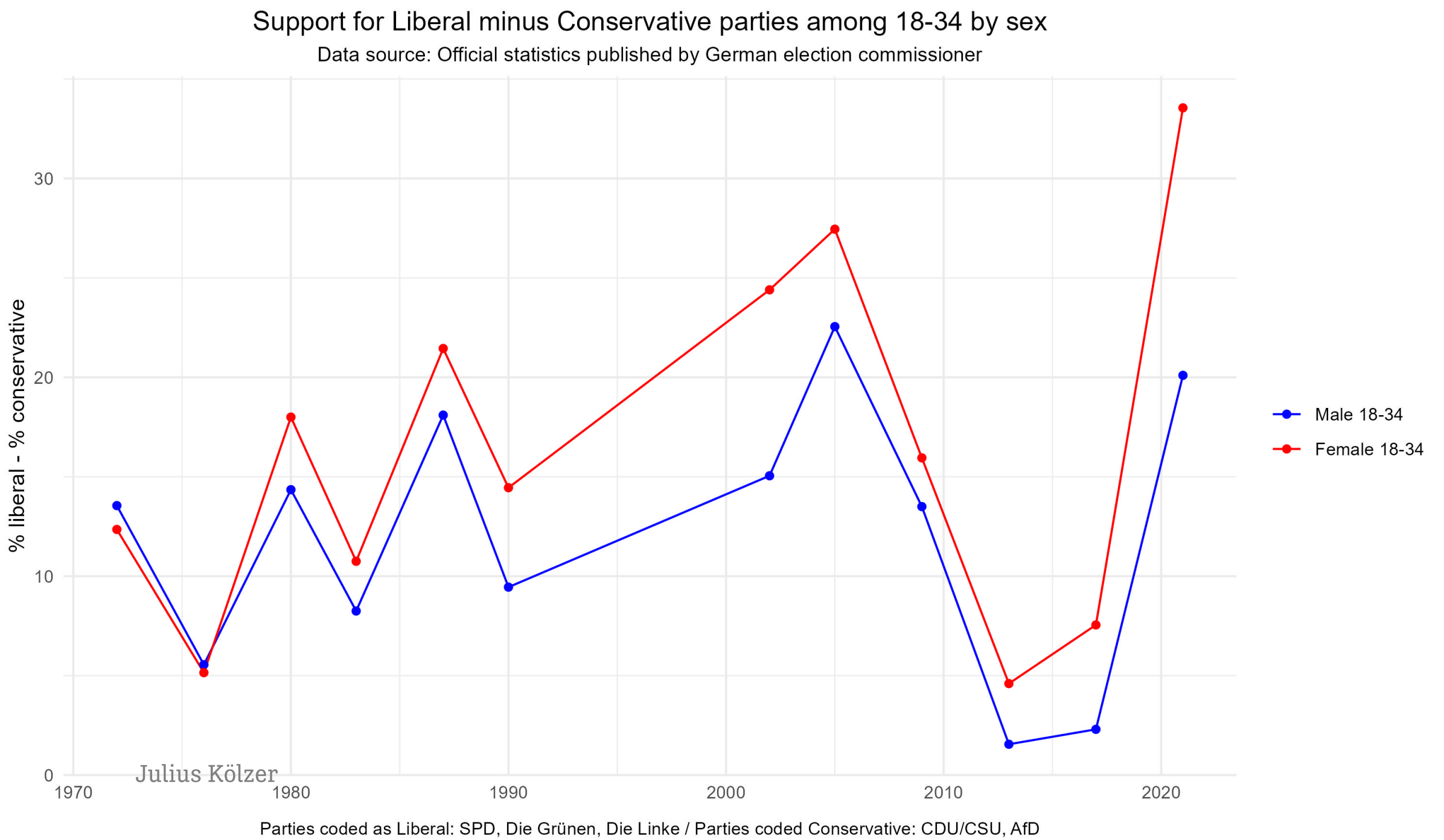

Age < 30
% liberal - % conservative
30
20
10
0
−10
1980
1990
2000
2010
2020
Year


Stated Political Ideology of 18-29 Men, LOWESS
% Liberal - % Conservative
35
30
25
20
15
10
5
0
-5
-10
-15
-20
-25
1985
1990
1995
2000
2005
2010
2015
2020
2025
Full
Drop 2022
James Traina

División ideológica en menores de 30 años

CIS Fusión Barómetros Enero-Marzo 2023. Graphext




Support for Liberal minus Conservative parties among 18-34 by sex
Data source: Official statistics published by German election commissioner
% liberal - % conservative
30
20
10
0
1970
1980
1990
2000
2010
2020
Julius Kölzer
Male 18-34
Female 18-34
Parties coded as Liberal: SPD, Die Grünen, Die Linke / Parties coded Conservative: CDU/CSU, AfD

Added fats and oils availability increased almost 30 pounds between 1970 and 2010

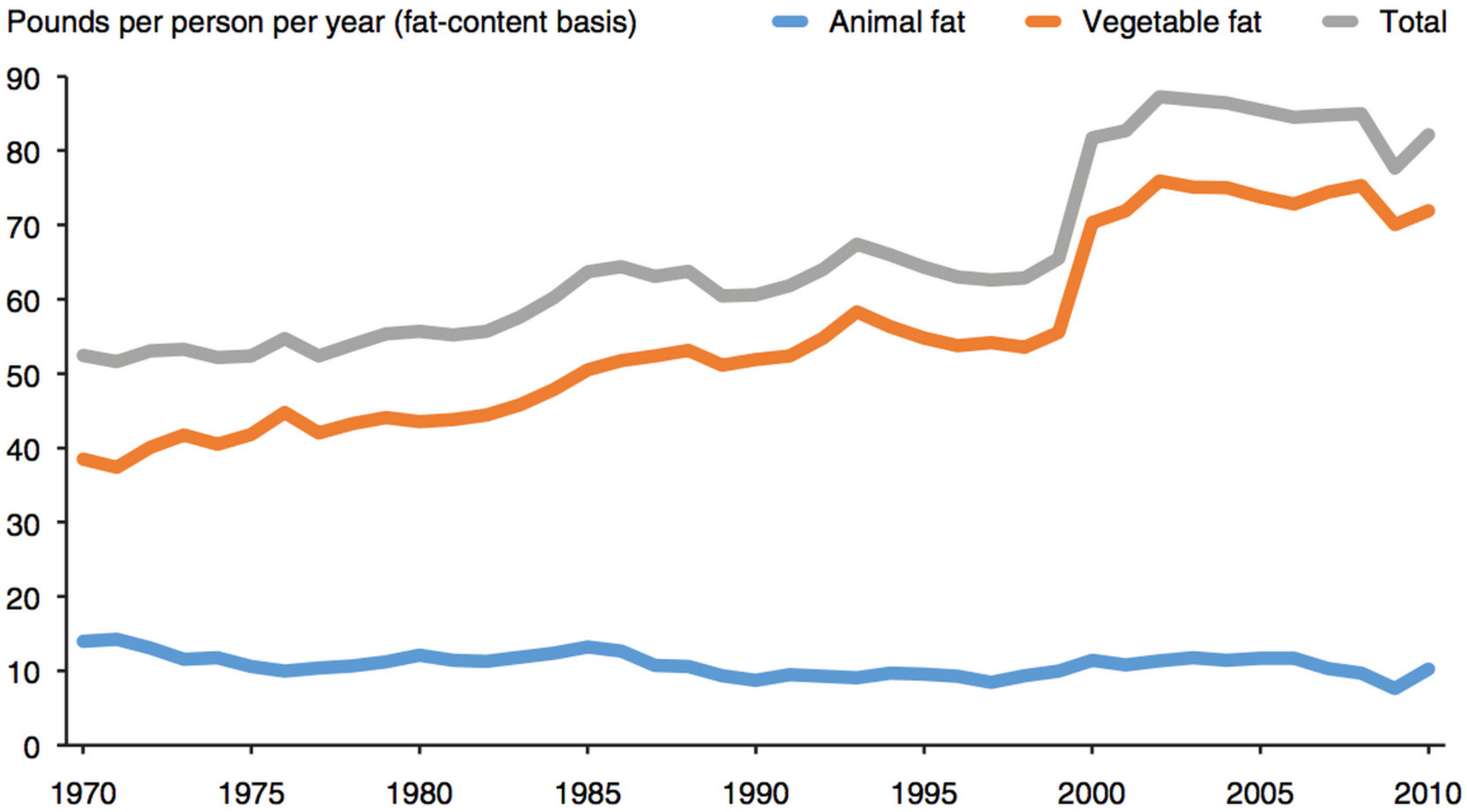


Table 9

Per capita availability of added fats and oils increased 57 percent between 1970 and 2010

Item	Food availability[1]		Change, 1970 to 2010		Loss-adjusted food availability[2]	
	1970	2010			1970	2010
	Pounds per person/ year		Pounds per year	Percent	Grams	
Animal fats	14.0	10.2	-3.7	-27	8.33	5.85
Butter	4.3	3.9	-0.4	-8	3.22	2.95
Lard (direct use)[3]	4.5	1.5	-2.9	-65	1.81	0.63
Edible beef tallow (direct use)[3]	NA	3.3	NA	NA	NA	1.33
Shortening	4.7	1.4	-3.3	-69	3.01	0.92
Margarine	0.5	0.0	-0.4	-92	0.29	0.03
Vegetable fats and oils[4]	38.5	71.9	33.4	87	27.85	57.14
Salad and cooking oils (olive and canola oil)	15.4	53.6	38.2	248	12.83	44.70
Shortening	12.6	13.9	1.3	10	8.06	8.86
Margarine	8.2	2.8	-5.4	-66	4.92	2.10
Other edible fats and oils[5]	2.3	1.7	-0.6	-27	2.04	1.48
Total added fats and oils[6]	52.5	82.2	29.7	57	36.18	62.99

Notes: NA = Not available. Because of rounding, calculations based on numbers in the table will not be accurate.

[1]Aggregate data, unadjusted for cooking losses, plate waste, and other losses. Fats and oils reported on a fat content basis.

[2]Adjusted for cooking losses, plate waste, and other losses. Fat content of butter and margarine calculated at 80 percent. One gram of fat equals 9 calories.

[3]Exlcudes use in margarine and shortening.

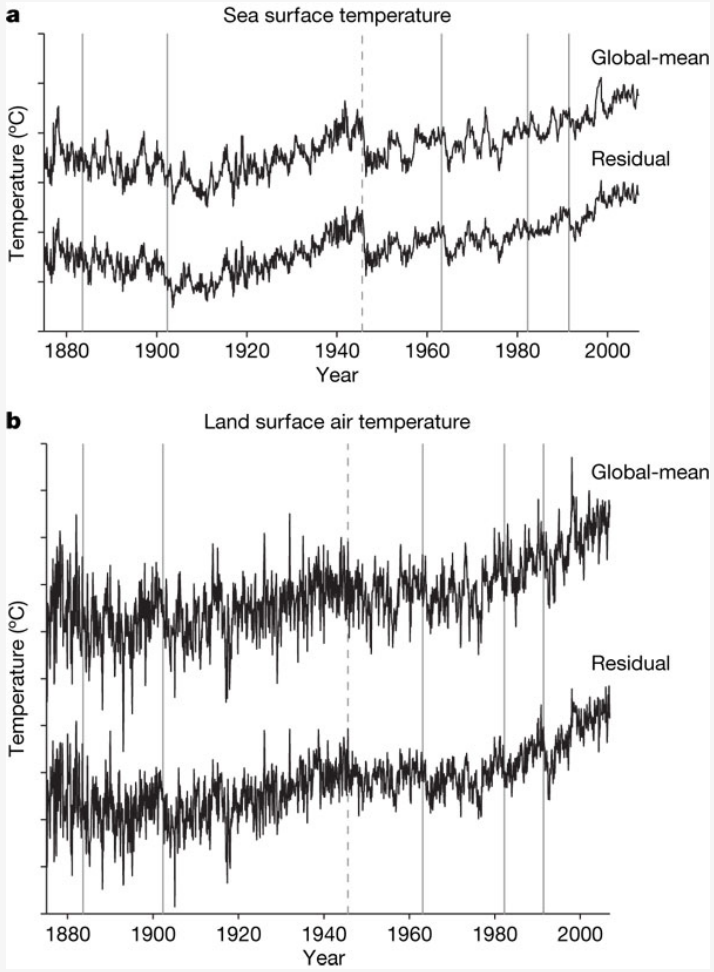
[4]In 2000, the number of firms reporting vegetable oil production increased.

[5]Specialty fats used mainly in confections and nondairy creamers.

[6]Excludes most naturally occurring fat, such as meats, beverage milks, nuts, and avocados.

Source: USDA, Economic Research Service, Food Availability data and Loss-Adjusted Food Availability data.

Global Sea Surface Temperature



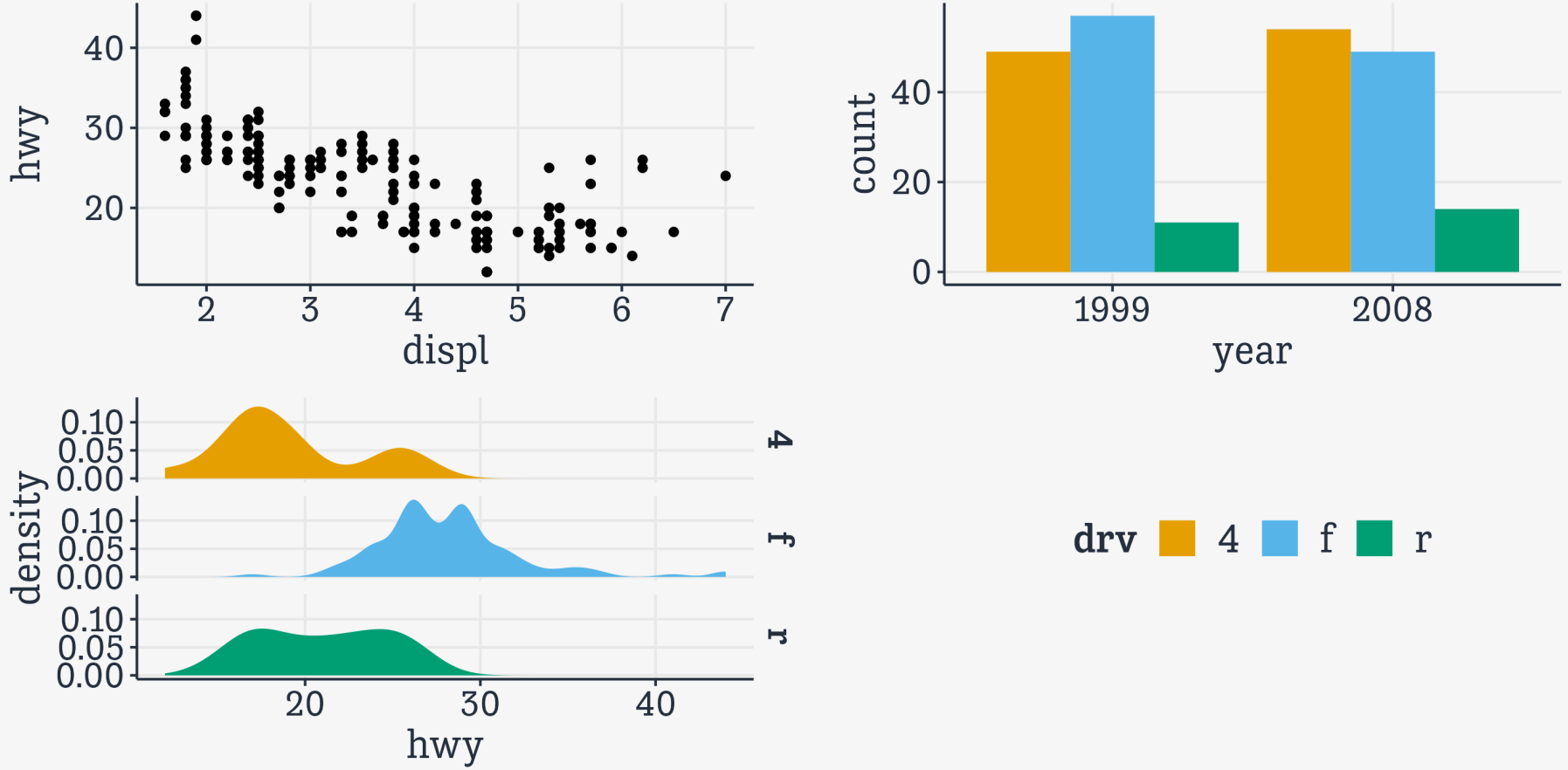
Patchwork

Patchwork

```
p1 <- ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy))

p2 <- ggplot(mpg) +
  geom_bar(aes(x = as.character(year), fill = drv), position = "dodge") +
  labs(x = "year")

p3 <- ggplot(mpg) +
  geom_density(aes(x = hwy, fill = drv), colour = NA) +
  facet_grid(rows = vars(drv))

p4 <- ggplot(mpg) +
  stat_summary(aes(x = drv, y = hwy, fill = drv), geom = "col", fun.data = mean_se) +
  stat_summary(aes(x = drv, y = hwy), geom = "errorbar", fun.data = mean_se, width = 0.5)
```

Patchwork

```
p1 + p2 + p3 + guide_area() + plot_layout(ncol = 2, guides = "collect")
```



US COVID Mortality in 2020

```
nchs_wdc
```

```
# A tibble: 347,706 × 7
   jurisdiction  year  week week_ending_date cause_detailed                n cause
   <chr>        <dbl> <dbl> <date>           <chr>                    <dbl> <chr>
 1 Alabama       2014     1 2014-01-04       All Cause                 1057 All …
 2 Alabama       2014     1 2014-01-04       Alzheimer disease (G30)     38 Alzh…
 3 Alabama       2014     1 2014-01-04       Cerebrovascular diseas…     51 Cere…
 4 Alabama       2014     1 2014-01-04       Chronic lower respirat…     66 Chro…
 5 Alabama       2014     1 2014-01-04       Diabetes mellitus (E10…     29 Diab…
 6 Alabama       2014     1 2014-01-04       Diseases of heart (I00…    264 Dise…
 7 Alabama       2014     1 2014-01-04       Influenza and pneumoni…     38 Infl…
 8 Alabama       2014     1 2014-01-04       Malignant neoplasms (C…    196 Canc…
 9 Alabama       2014     1 2014-01-04       Natural Cause              992 Natu…
10 Alabama       2014     1 2014-01-04       Nephritis, nephrotic s…     21 Kidn…
# ℹ 347,696 more rows
```

US COVID Mortality in 2020

```
start_week ← 1
end_week ← 53

states ← nchs_wdc ▷
  select(jurisdiction) ▷
  unique()

states
```

```
# A tibble: 54 × 1
   jurisdiction
   <chr>
 1 Alabama
 2 Alaska
 3 Arizona
 4 Arkansas
 5 California
 6 Colorado
 7 Connecticut
 8 Delaware
 9 District of Columbia
10 Florida
# ℹ 44 more rows
```

US COVID Mortality in 2020

```
dat ← nchs_wdc ▷
  filter(year > 2014, year < 2021) ▷
  mutate(month_label = lubridate::month(week_ending_date, label = TRUE))

dat
```

```
# A tibble: 225,450 × 8
   jurisdiction  year  week week_ending_date cause_detailed              n cause
   <chr>        <dbl> <dbl> <date>           <chr>                 <dbl> <chr>
 1 Alabama       2015     1 2015-01-10       All Cause              1139 All …
 2 Alabama       2015     1 2015-01-10       Alzheimer disease (G30)   59 Alzh…
 3 Alabama       2015     1 2015-01-10       Cerebrovascular diseas…   48 Cere…
 4 Alabama       2015     1 2015-01-10       Chronic lower respirat…   73 Chro…
 5 Alabama       2015     1 2015-01-10       Diabetes mellitus (E10…   36 Diab…
 6 Alabama       2015     1 2015-01-10       Diseases of heart (I00…  273 Dise…
 7 Alabama       2015     1 2015-01-10       Influenza and pneumoni…   48 Infl…
 8 Alabama       2015     1 2015-01-10       Malignant neoplasms (C…  200 Canc…
 9 Alabama       2015     1 2015-01-10       Natural Cause          1069 Natu…
10 Alabama       2015     1 2015-01-10       Nephritis, nephrotic s…   26 Kidn…
# ℹ 225,440 more rows
# ℹ 1 more variable: month_label <ord>
```

US COVID Mortality in 2020

```
average_deaths ← nchs_wdc ▷
  filter(year %in% c(2015:2019)) ▷
  group_by(jurisdiction, week, cause) ▷
  summarize(average_wk_deaths = mean(n, na.rm = TRUE))

average_deaths
```

```
# A tibble: 36,504 × 4
# Groups:   jurisdiction, week [2,808]
   jurisdiction  week cause                                 average_wk_deaths
   <chr>        <dbl> <chr>                                             <dbl>
 1 Alabama          1 All Cause                                        1119.
 2 Alabama          1 Alzheimer's                                        56.2
 3 Alabama          1 Cancer                                            206
 4 Alabama          1 Cerebrovascular Diseases                           55.2
 5 Alabama          1 Chronic Lower Respiratory Diseases                 76.6
 6 Alabama          1 Diabetes                                           29.8
 7 Alabama          1 Diseases of the Heart                             282.
 8 Alabama          1 Influenza and Pneumonia                            33.6
 9 Alabama          1 Kidney Diseases                                    20.4
10 Alabama          1 Natural Causes                                   1042.
# ℹ 36,494 more rows
```

US COVID Mortality in 2020

```
df ← left_join(dat, average_deaths) ▷
  select(everything(), n, average_wk_deaths) ▷
  mutate(n_diff = n - average_wk_deaths,
         pct_diff = (n_diff / n)*100) ▷
  filter(cause %nin% c("Natural Causes", "Other"))

df
```

```
# A tibble: 191,646 × 11
   jurisdiction  year  week week_ending_date cause_detailed              n cause
   <chr>        <dbl> <dbl> <date>           <chr>                   <dbl> <chr>
 1 Alabama       2015     1 2015-01-10       All Cause                1139 All …
 2 Alabama       2015     1 2015-01-10       Alzheimer disease (G30)    59 Alzh…
 3 Alabama       2015     1 2015-01-10       Cerebrovascular diseas…    48 Cere…
 4 Alabama       2015     1 2015-01-10       Chronic lower respirat…    73 Chro…
 5 Alabama       2015     1 2015-01-10       Diabetes mellitus (E10…    36 Diab…
 6 Alabama       2015     1 2015-01-10       Diseases of heart (I00…   273 Dise…
 7 Alabama       2015     1 2015-01-10       Influenza and pneumoni…    48 Infl…
 8 Alabama       2015     1 2015-01-10       Malignant neoplasms (C…   200 Canc…
 9 Alabama       2015     1 2015-01-10       Nephritis, nephrotic s…    26 Kidn…
10 Alabama       2015     1 2015-01-10       Other diseases of resp…    30 Othe…
# ℹ 191,636 more rows
# ℹ 4 more variables: month_label <ord>, average_wk_deaths <dbl>, n_diff <dbl>,
#   pct_diff <dbl>
```

Getting set up

```
nwks ← 53

season_label ← tibble(wk_num = lubridate::epiweek(as.Date(c("2020-03-01",
                                                            "2020-06-01",
                                                            "2020-09-01",
                                                            "2020-12-01"))),
                      season_lab = c("Spring", "Summer", "Autumn", "Winter"))

season_label
```

```
# A tibble: 4 × 2
  wk_num season_lab
   <dbl> <chr>
1     10 Spring
2     23 Summer
3     36 Autumn
4     49 Winter
```

Getting set up

```
month_label ← tibble(wk_num = lubridate::epiweek(lubridate::ymd("2020-01-15") + months(0:11)),
                     month_lab = as.character(lubridate::month(lubridate::ymd("2020-01-15") +
                                                               months(0:11), label = TRUE)))

month_label
```

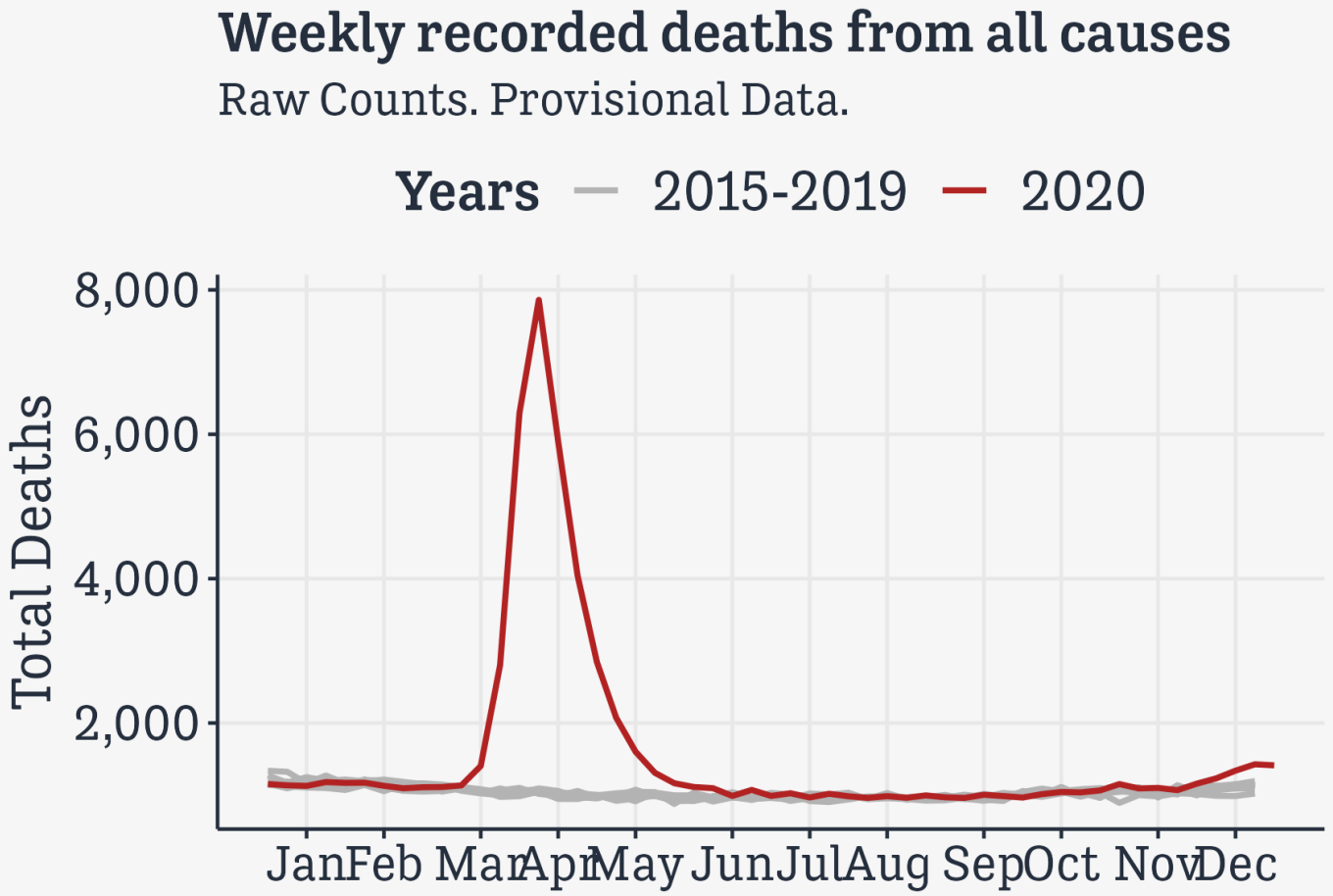
```
# A tibble: 12 × 2
   wk_num month_lab
    <dbl> <chr>
 1      3 Jan
 2      7 Feb
 3     12 Mar
 4     16 Apr
 5     20 May
 6     25 Jun
 7     29 Jul
 8     33 Aug
 9     38 Sep
10     42 Oct
11     47 Nov
12     51 Dec
```

Panel 1

```
out ← df ▷
    filter(year < 2021, jurisdiction %in% "New York City", cause == "All Cause") ▷
    group_by(year, week) ▷
    mutate(yr_ind = year %in% 2020) ▷
    filter(!(year == 2020 & week > nwks)) ▷
    ggplot(aes(x = week, y = n, color = yr_ind, group = year)) +
    geom_line(size = 0.9) +
    scale_color_manual(values = c("gray70", "firebrick"), labels = c("2015-2019", "2020")) +
    scale_x_continuous(breaks = month_label$wk_num, labels = month_label$month_lab) +
    scale_y_continuous(labels = scales::comma) +
    labs(x = NULL,
         y = "Total Deaths",
         color = "Years",
         title = "Weekly recorded deaths from all causes",
         subtitle = "Raw Counts. Provisional Data.")
```

Panel 1

```
out
```

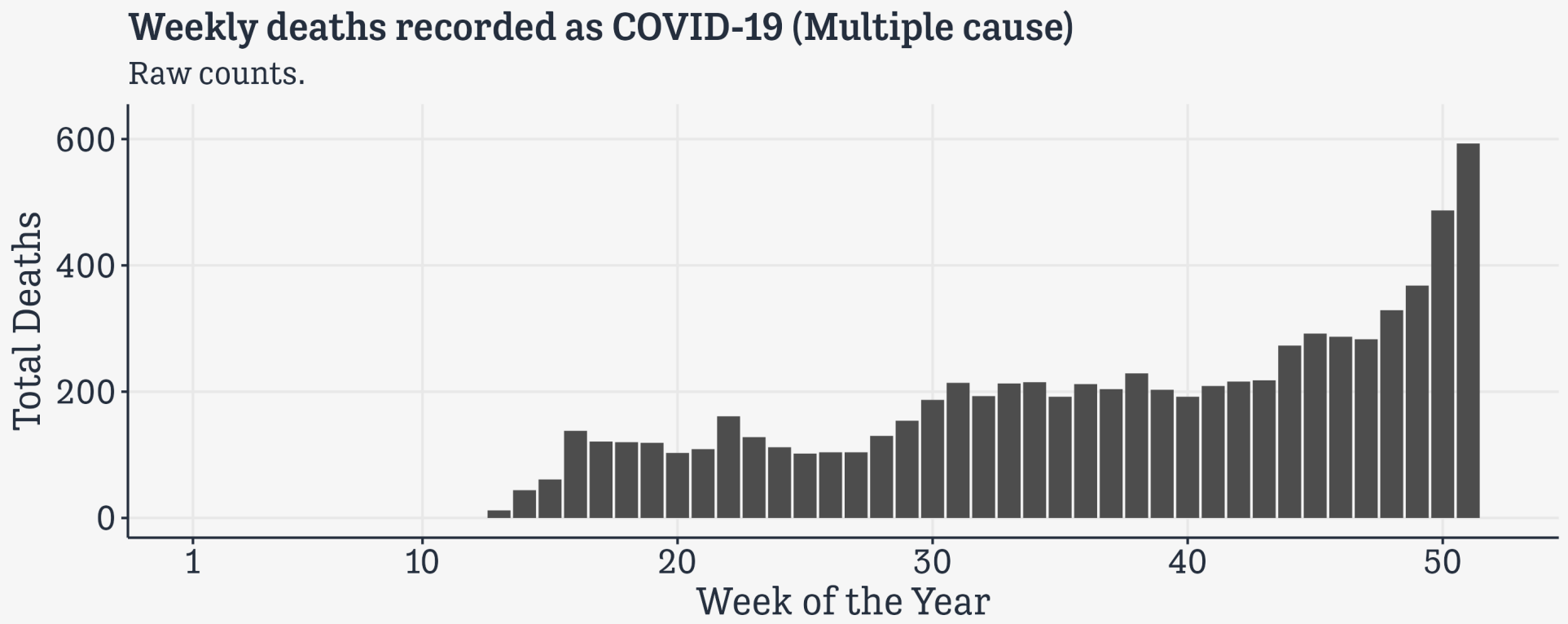

Weekly recorded deaths from all causes
Raw Counts. Provisional Data.
Years
2015-2019
2020
Total Deaths
8,000
6,000
4,000
2,000
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Panel 2

```
out ← df ▷
  filter(year < 2021) ▷
  filter(jurisdiction %in% "North Carolina", cause %in% c("COVID-19 Multiple cause")) ▷
  group_by(year, week) ▷
  mutate(yr_ind = year %in% 2020) ▷
  filter(year == 2020) ▷
  ggplot(aes(x = week, y = n, group = year)) +
  geom_col(fill = "gray30") +
  scale_x_continuous(breaks = c(1, 10, 20, 30, 40, 50),
                     labels = as.character(c(1, 10, 20, 30, 40, 50)),
                     limits = c(1, 52)) +
  scale_y_continuous(labels = scales::comma) +
  labs(x = "Week of the Year",
       y = "Total Deaths",
       color = "Years",
       subtitle = "Raw counts.",
       title = "Weekly deaths recorded as COVID-19 (Multiple cause)")
```

Panel 2

```
out
```

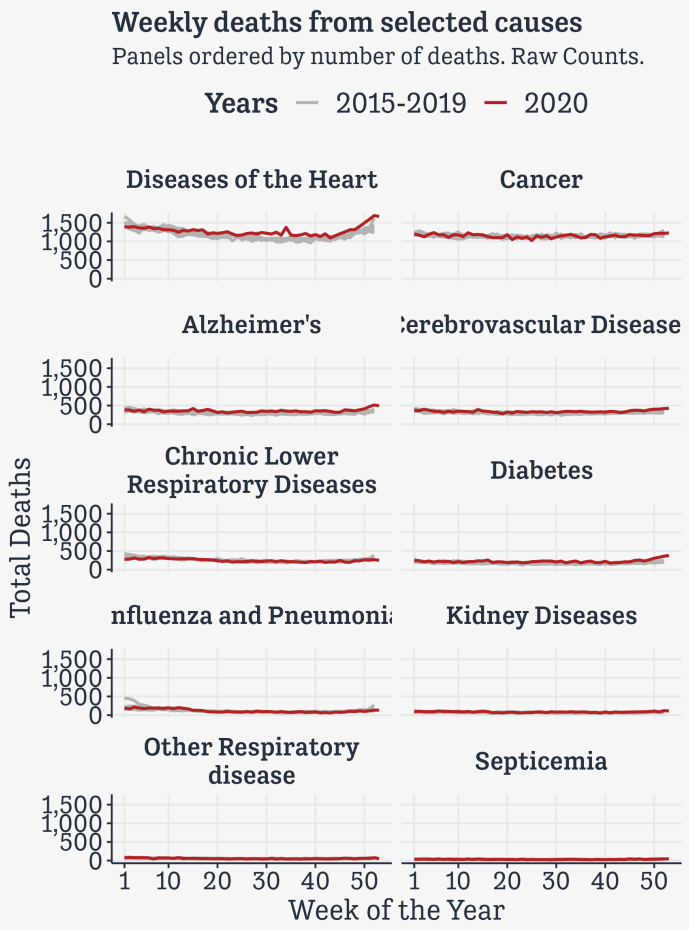

Weekly deaths recorded as COVID-19 (Multiple cause)
Raw counts.
Total Deaths
600
400
200
0
1
10
20
30
40
50
Week of the Year

Panel 3

```
panel_ordering ← order_panels(st = "California")

out ← df ▷
    filter(year < 2021,
           jurisdiction == "California",
           cause %nin% c("All Cause", "COVID-19 Multiple cause", "COVID-19 Underlying")) ▷
    group_by(cause, year, week) ▷
    summarize(deaths = sum(n, na.rm = TRUE), .groups = "drop") ▷
    mutate(yr_ind = year %in% 2020) ▷
    filter(!(year == 2020 & week > nwks)) ▷
    left_join(panel_ordering, by = "cause") ▷
    ggplot(aes(x = week, y = deaths, color = yr_ind)) +
    geom_line(size = 0.9, mapping = aes(group = year)) +
    scale_color_manual(values = c("gray70", "firebrick"), labels = c("2015-2019", "2020")) +
    scale_x_continuous(breaks = c(1, 10, 20, 30, 40, 50), labels = as.character(c(1, 10, 20, 30, 40, 50))) +
    scale_y_continuous(labels = scales::comma) +
    facet_wrap(~ cause_ord, ncol = 2, labeller = label_wrap_gen(25)) +
    labs(x = "Week of the Year",
         y = "Total Deaths",
         color = "Years",
         title = "Weekly deaths from selected causes",
         subtitle = "Panels ordered by number of deaths. Raw Counts.")
```

Panel 3

```
out
```

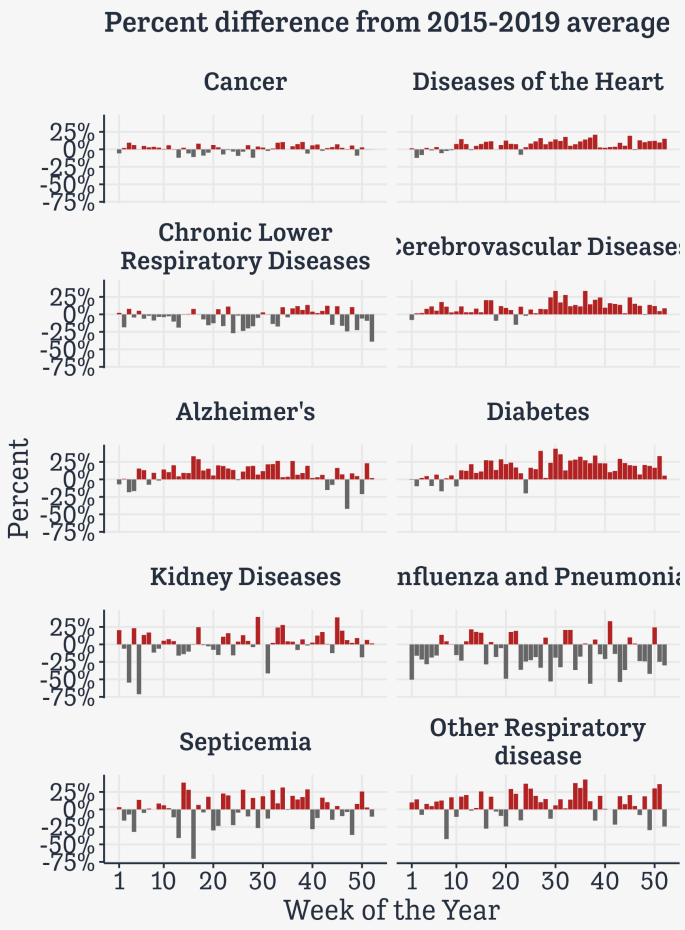

Weekly deaths from selected causes
Panels ordered by number of deaths. Raw Counts.
Years 2015-2019 2020
Diseases of the Heart
Cancer
Alzheimer's
Cerebrovascular Disease
Chronic Lower Respiratory Diseases
Diabetes
nfluenza and Pneumoni
Kidney Diseases
Other Respiratory disease
Septicemia
Total Deaths
1,500
1,000
500
0
1 10 20 30 40 50
Week of the Year

Panel 4

```
panel_ordering ← order_panels(st = "North Carolina")

out ← df ▷
    filter(year < 2021) ▷
    filter(jurisdiction %in% "North Carolina",
           year == 2020,
           cause %nin% c("All Cause", "COVID-19 Multiple cause",
                         "COVID-19 Underlying"), !is.na(pct_diff)) ▷
    group_by(week) ▷
    filter(!(week > nwks)) ▷
    mutate(ov_un = pct_diff > 0) ▷
    left_join(panel_ordering, by = "cause") ▷
    ggplot(aes(x = week, y = pct_diff/100, fill = ov_un)) +
    geom_col() +
    scale_x_continuous(breaks = c(1, seq(10, nwks, 10)), labels = as.character(c(1, seq(10, nwks, 10)))) +
    scale_y_continuous(labels = scales::percent) +
    scale_fill_manual(values = c("gray40", "firebrick")) +
    guides(fill = "none") +
    facet_wrap(~ cause_ord, ncol = 2, labeller = label_wrap_gen(25)) +
    labs(x = "Week of the Year",
         y = "Percent",
         title = "Percent difference from 2015-2019 average")
```

Panel 4

```
out
```


Percent difference from 2015-2019 average
Cancer
Diseases of the Heart
Chronic Lower Respiratory Diseases
Cerebrovascular Disease
Alzheimer's
Diabetes
Kidney Diseases
nfluenza and Pneumonia
Septicemia
Other Respiratory disease
Percent
25%
0%
-25%
-50%
-75%
1
10
20
30
40
50
Week of the Year

Now the fun part

```
patch_state_count ← function(state) {

  out ← df ▷
    filter(year < 2021, jurisdiction %in% state, cause == "All Cause") ▷
    group_by(year, week) ▷
    mutate(yr_ind = year %in% 2020) ▷
    filter(!(year == 2020 & week > nwks)) ▷
    ggplot(aes(x = week, y = n, color = yr_ind, group = year)) +
    geom_line(size = 0.9) +
    scale_color_manual(values = c("gray70", "firebrick"), labels = c("2015-2019", "2020")) +
    scale_x_continuous(breaks = month_label$wk_num, labels = month_label$month_lab) +
    scale_y_continuous(labels = scales::comma) +
    labs(x = NULL,
         y = "Total Deaths",
         color = "Years",
         title = "Weekly recorded deaths from all causes",
         subtitle = "Raw Counts. Provisional Data.")

  out

}
```

Wrap each panel into a function that creates it

```
patch_state_covid ← function(state) {

  out ← df ▷
    filter(year < 2021) ▷
    filter(jurisdiction %in% state, cause %in% c("COVID-19 Multiple cause")) ▷
    group_by(year, week) ▷
    mutate(yr_ind = year %in% 2020) ▷
    filter(year == 2020) ▷
    ggplot(aes(x = week, y = n, group = year)) +
    geom_col(fill = "gray30") +
    scale_x_continuous(breaks = c(1, 10, 20, 30, 40, 50),
                       labels = as.character(c(1, 10, 20, 30, 40, 50)),
                       limits = c(1, 52)) +
    scale_y_continuous(labels = scales::comma) +
    labs(x = "Week of the Year",
         y = "Total Deaths",
         color = "Years",
         subtitle = "Raw counts.",
         title = "Weekly deaths recorded as COVID-19 (Multiple cause)")

  out

}
```

Wrap each panel into a function that creates it

```
patch_state_cause ← function(state, nc = 2) {

  panel_ordering ← order_panels(st = state)

  out ← df ▷
    filter(year < 2021,
           jurisdiction == state,
           cause %nin% c("All Cause", "COVID-19 Multiple cause", "COVID-19 Underlying")) ▷
    group_by(cause, year, week) ▷
    summarize(deaths = sum(n, na.rm = TRUE), .groups = "drop") ▷
    mutate(yr_ind = year %in% 2020) ▷
    filter(!(year == 2020 & week > nwks)) ▷
    left_join(panel_ordering, by = "cause") ▷
    ggplot(aes(x = week, y = deaths, color = yr_ind)) +
    geom_line(size = 0.9, mapping = aes(group = year)) +
    scale_color_manual(values = c("gray70", "firebrick"), labels = c("2015-2019", "2020")) +
    scale_x_continuous(breaks = c(1, 10, 20, 30, 40, 50), labels = as.character(c(1, 10, 20, 30, 40, 50))) +
    scale_y_continuous(labels = scales::comma) +
    facet_wrap(~ cause_ord, ncol = nc, labeller = label_wrap_gen(25)) +
    labs(x = "Week of the Year",
         y = "Total Deaths",
         color = "Years",
         title = "Weekly deaths from selected causes",
         subtitle = "Panels ordered by number of deaths. Raw Counts.")

  out
}
```

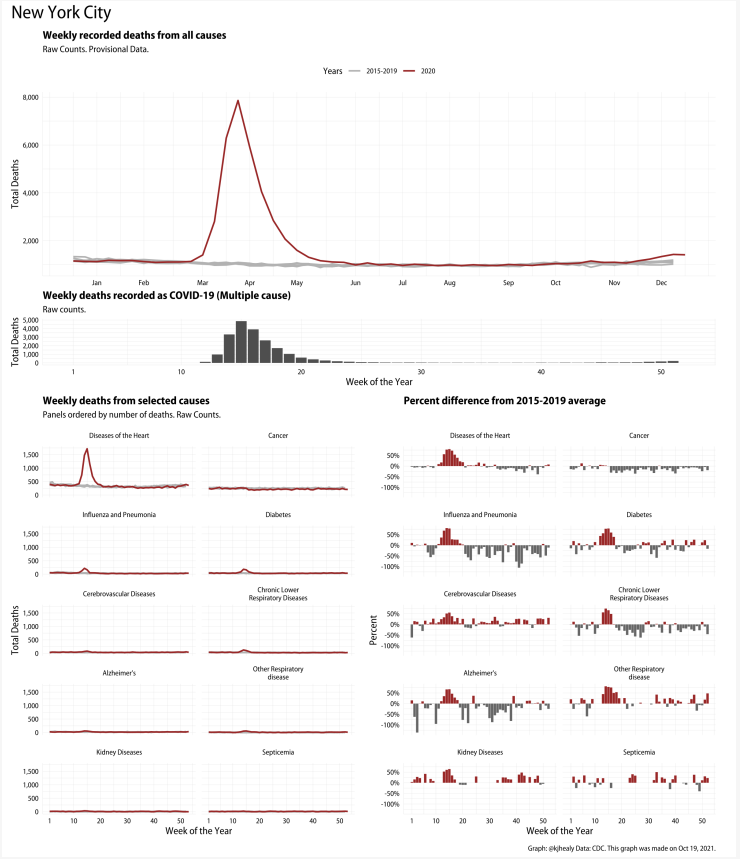
Wrap each panel into a function that creates it

Now wrap those functions up, too

```
make_patchplot ← function(state){

  if(state == "New York")  {
    state_title ← paste(state, "(Excluding New York City)")
  } else
  {
    state_title ← state
  }

  timestamp ←  lubridate::stamp("March 1, 1999", "%B %d, %Y")(lubridate::ymd(Sys.Date()))


  (patch_state_count(state) + theme(plot.margin = unit(c(5,0,0,0), "pt"))) /
    patch_state_covid(state) /
    (patch_state_cause(state) + (patch_state_percent(state))) +
    plot_layout(heights = c(2, 0.5, 4), guides = 'collect') +
    plot_annotation(
      title = state_title,
      caption = paste0("Graph: @kjhealy Data: CDC. This graph was made on ", timestamp, "."),
      theme = theme(plot.title = element_text(size = rel(2), hjust = 0, face = "plain")))
}
```

And now we can do this

```
p_out ← make_patchplot("New York City")
```



:scale 100%

But we're not done yet!

```
states ← states ▷
  mutate(fname = paste0(here::here("figures", tolower(jurisdiction)), "_patch"),
         fname = stringr::str_replace_all(fname, " ", "_"))

states
```

```
# A tibble: 54 × 2
   jurisdiction         fname
   <chr>                <chr>
 1 Alabama              /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 2 Alaska               /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 3 Arizona              /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 4 Arkansas             /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 5 California           /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 6 Colorado             /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 7 Connecticut          /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 8 Delaware             /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
 9 District of Columbia /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
10 Florida              /Users/kjhealy/Documents/talks/mcr-sta-313-guest-2024/f…
# ℹ 44 more rows
```

Map the function to the jurisdictions

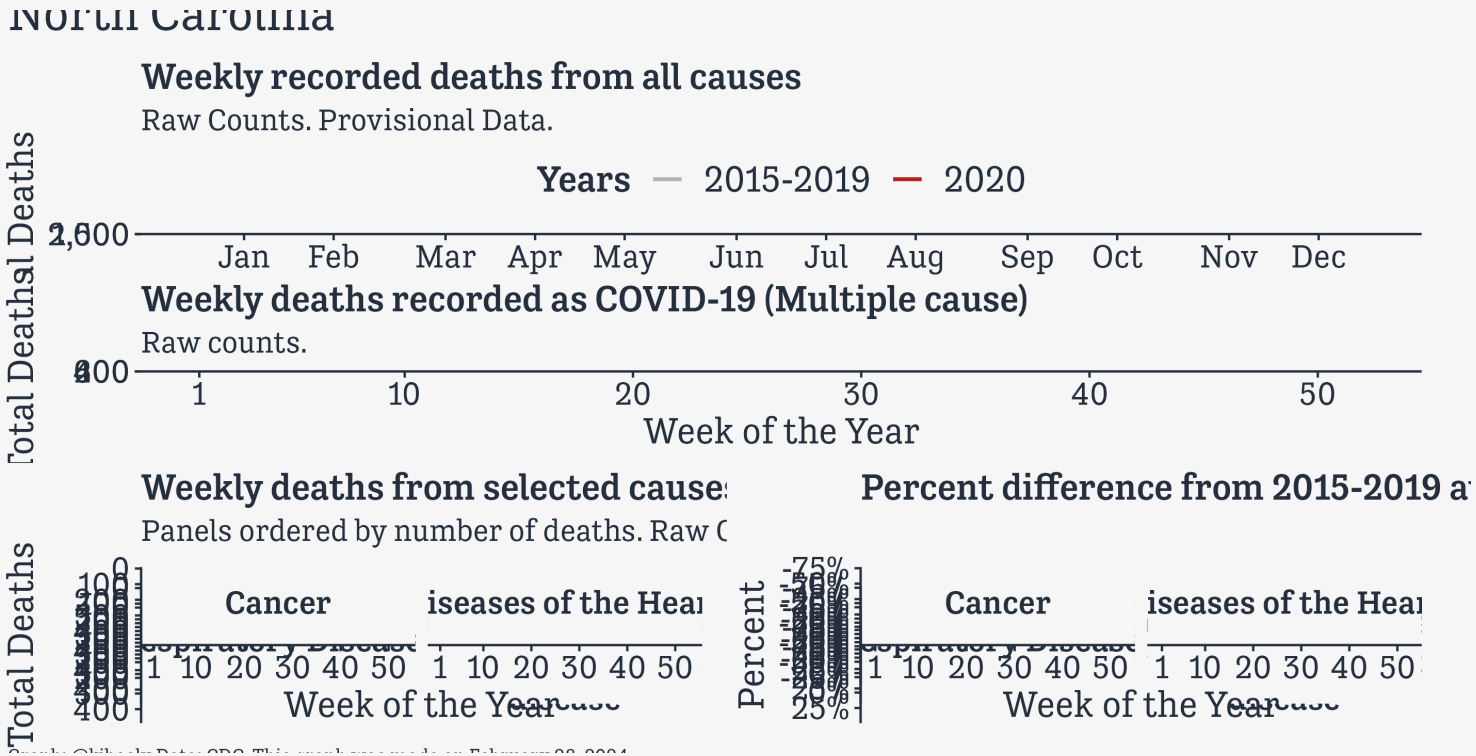
```
out_patch ← states ▷
  mutate(patch_plot = map(jurisdiction, make_patchplot))

out_patch
```

```
# A tibble: 54 × 3
   jurisdiction         fname                                          patch_plot
   <chr>                <chr>                                          <list>
 1 Alabama              /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 2 Alaska               /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 3 Arizona              /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 4 Arkansas             /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 5 California           /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 6 Colorado             /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 7 Connecticut          /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 8 Delaware             /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
 9 District of Columbia /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
10 Florida              /Users/kjhealy/Documents/talks/mcr-sta-313-g… <patchwrk>
# ℹ 44 more rows
```

Map the function to the jurisdictions

```
out_patch ▷
  filter(jurisdiction == "North Carolina") ▷
  pull(patch_plot)
```

```
[[1]]
```



Walk out the results to a folder

```
## Be patient
walk2(paste0(out_patch$fname, ".png"),
      out_patch$patch_plot, ggsave, height = 16, width = 9, dpi = 200)
```

Et voila

```
fs::dir_tree("figures")

figures
├── alabama_patch.png
├── alaska_patch.png
├── arizona_patch.png
├── arkansas_patch.png
├── california_patch.png
├── colorado_patch.png
├── connecticut_patch.png
├── delaware_patch.png
├── district_of_columbia_patch.png
├── florida_patch.png
├── georgia_patch.png
├── hawaii_patch.png
├── idaho_patch.png
├── illinois_patch.png
```