Tackling common plotting issues in `ggplot2` and `knitr`

Silvie Cinková

2025-08-01

Table of contents

1 How to use this file

Open the qmd source and view it in the Visual editor. Open the html rendering nearby and compare the outcomes.

2 Common issues

- Plot renders too small/narrow/low.
- Axis tick labels overlap.
- Axis ticks should be denser.
- Texts render too small.
- Overlap in text labels.
- The publisher wants it black and white.

3 Setup

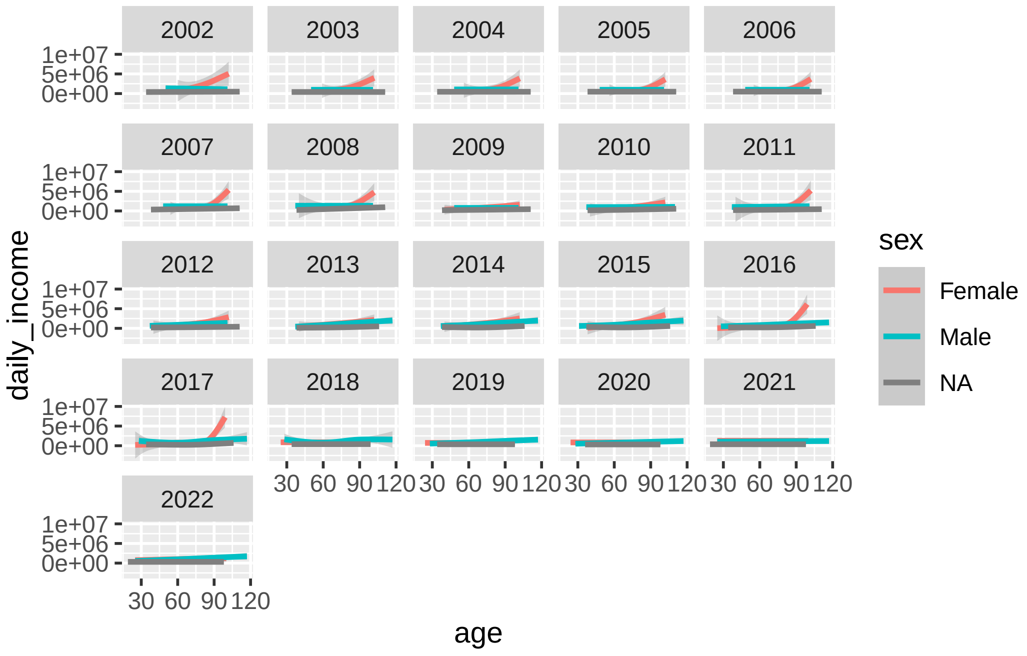
```
library(tidyverse, warn.conflicts = FALSE, quietly = TRUE, logical.return =
↪  FALSE )
library(glue, warn.conflicts = FALSE, quietly = TRUE)
library(grDevices, warn.conflicts = FALSE, quietly = TRUE) # mostly
↪  unnecesary
billionaires_df <- read_tsv(glue("~/R_BEGINNERS_SHORT/",

↪  "datasets_ATRIUM/billionaires_combined.tsv"),
                              show_col_types = FALSE)
```

When you load `tidyverse` , it loads some of them together, definitely `dplyr` and `ggplot`.

4 Problem: Plot too small

- mostly just pre-rendering and rendered OK, saved OK
- default: 178 x 178 mm (or equivalent in other units)

```
billionaires_df %>%
  ggplot() +
  geom_smooth(mapping = aes(x = age,
                            color = sex,
                             y = daily_income)) +
  facet_wrap(~ time)
```



5 Control dimensions when saving

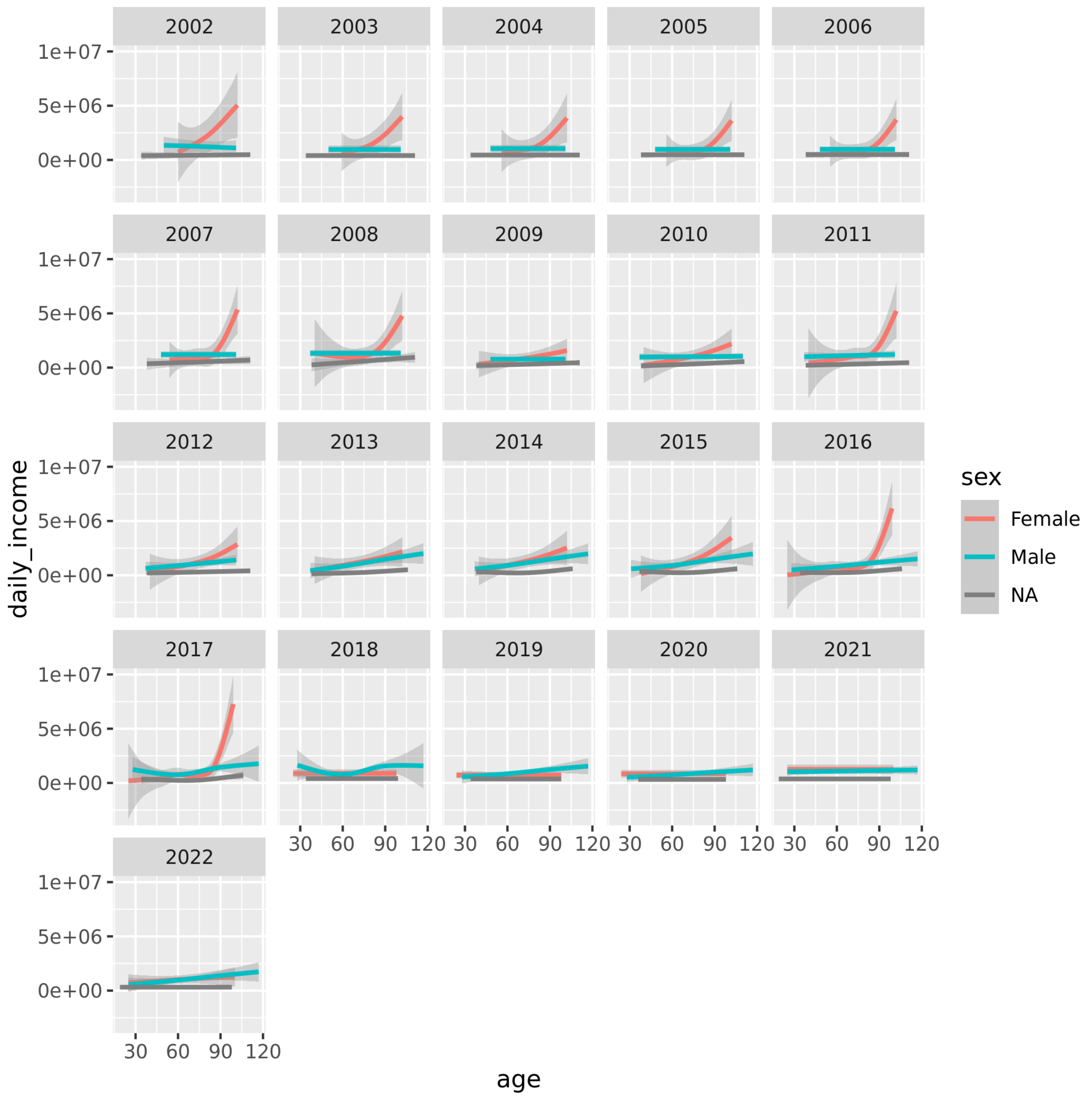
```
#### Erase #| eval: false when you want this script to run on Run or Render
ggsave(glue("my_output_files", "/10_ggplotOther_plot_too_small.png"),
       device = grDevices::png, units = "mm")
```

```
Saving 178 x 178 mm image
```

These are the dimensions of the file:

```
ggsave("my_output_files/10_ggplotOther_plot_too_small_height270.png",
       units = "mm", height = 270, device = grDevices::png)
```

```
Saving 178 x 270 mm image
```



True dimensions in the file:

or you can multiply both directions with something. Below is a very disproportionate plot just to make the effects very visible:

```
ggsave("my_output_files/10_ggplotOther_plot_too_small_scaleboth.png",
       units = "mm", height = 178 * 1.1, width = 178 * 0.7, device =
 ↪  grDevices::png)
```

6 You want to render it better

Sometimes you need to render a document with plots for sharing. One solution could be like the one above: you generate the plots and save them. Then you insert the files as images and prevent the code from running again. The cleaner option is that you fiddle with the dimensions in the `knitr` *chunk options.*

`knitr` is one of the libraries that allow Quarto to render markdown documents in HTML, PDF, and doc. It pays to have an idea about the existence of `knitr` *chunk options* - be it merely as key words for AI prompts!

Below you see activated a few of them: `fig.width`, etc. , if you look at this file in Quarto. You will not see them in the rendered documents of any format! They belong to the start of the chunk, and the code to control the rendered width and height would look like this:

~~~
```{r}
#| fig.width: 7
#| fig.height: 10
# your code or comments follow
```
~~~

The units are probably inches. The options are spelled in an older way using dots, but Help lists them with dashes instead (i.e., `fig-weight` instead of `fig.weight`. Our platform currently uses a slightly outdated version of RStudio from 2023 that ignores the new spelling (2023.06.1+524 "Mountain Hydrangea" Release (547dcf861cac0253a8abb52c135e44e02ba407a1, 2023-07-07) for Ubuntu Jammy), so I stick to the old spelling, but chances are that you will get a newer one to your computer that has switched the spelling to dash.

```r
# above are knitr chunk options invisible in html
billionaires_df %>% ggplot() +
  geom_smooth(mapping = aes(x = age, color = sex, y = daily_income)) +
  facet_wrap(~ time)
```

7 You want axis ticks labeled denser

This is the default.

```r
billionaires_df %>% ggplot() + geom_bar(aes(x = time, fill = sex))
```

You want ticks on the Y axis to mark each 500.

First, you need to know the right base R function to generate the sequence of the labels:

```
seq(from = 0, to = 3500, by = 500)
```

```
[1]    0  500 1000 1500 2000 2500 3000 3500
```

Then you tell ggplot2 that you want to modify the ticks on the Y-axis by calling the appropriate `scale_….` function. Each aesthetic scale has its own `scale_…` function with arguments that modify its behavior. Note that the `scale_` functions are divided into _discreet, _continuous, and sometimes _manual. You have to pick the type that corresponds to the type of the variable that you mapped on that aesthetic scale. The rule of the thumb is that categorical variables are always discrete and numeric variables are continuous. When you want a numeric variable to plot as discrete, you make it a factor before you pick the scale. In the plot, I also modify the X-axis.

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_y_continuous(breaks = seq(from = 0, to = 3500, by = 500)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 2 ))
```

Years are conceptually a discrete variable, but they are numbers, so it is a continuous scale.

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1))
```

8 Axis tick labels overlap

When axis tick labels overlap, the easiest workaround is to tilt the plot by 90 degrees with `coord_flip()`. This function belongs to the coordinates layer in `ggplot2`.

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1)) +
  coord_flip()
```

Sometimes this is all you have to do, but when you can't do it, you must delve into the `theme` layer, calling the appropriate `axis.text_…` argument (for X, for Y, or for both). This wants that you call a function called `element_text.` This function has arguments with which you can tilt the labels, adjust their size and their position around the corresponding axis tick, etc.

What often helps is tilting the labels:

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1)) +
  theme(axis.text.x = element_text(angle = 60))
```

Sometimes the labels bleed into the plot. This you can adjust with the vertical adjustment argument `vjust`. Experiment with the values. Normally you should need a small value like 0.1 or -0.1. Try `vjust = 0` to get a reference point.

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.7))
```

When you need to let the tick labels disappear, you do it like so:

```
billionaires_df %>% ggplot() +
  geom_boxplot(aes(y = time)) +
  scale_x_continuous(breaks = NULL)
```

9 Change the titles of aesthetic scales, e.g. axes

Two ways to access the axis titles

- `scale_…`
- `xlab`, `ylab`

9.1 `scale_…` to change the title of any aesthetic scale

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1),
                     name = "year") +
  scale_y_continuous(name = "number of billionaires listed in Forbes\n and
 ↪  other popular charts") +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.7)) +
  scale_fill_discrete(name = "gender", labels = c("women", "men", "diverse",
 ↪  "NA"))
```

9.2 `xlab`, `ylab` to change titles of the X, Y axes

These are shortcuts - probably written because these tasks are extremely common.

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.7)) +
  xlab(label = "year") +
  ylab(label = "number of billionaires listed in Forbes\n and other popular
↪   charts")
```

```
# Notice the \n to enforce a new line
```

10 Change things in color/fill with `scale_…`

- Stick to ready-made options whenever you can, it can easily get too complicated.
- Prefer to combine colors from color palettes designed by professionals. Some look awful but all guarantee enough contrast. Some of them even work for color-blind audience.
- Important things to consider

```
-   Discrete/ordinal or continuous variable? Some palettes work even only for binned variabl

-   Can values be distributed on a scale?

    -   diverging scale: two extremes (e.g., *totally agree - agree - don't know - disagree

    -   sequential: zero to extreme (e.g., *beginner - intermediate - advanced - expert* or

    -   qualitative: no scale, just contrast (e.g., *bugs, beetles, butterflies*)
```

10.1 `scale_fill_brewer`

- for discrete variables
- color palettes designed by cartographer Cynthia Brewer https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3
- The palettes are divided into three types: diverging (`div`), qualitative (`qual`) and sequential (`seq`)

```
# run repeatedly and explore the fill options with Help
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.7)) +
  scale_fill_brewer(name = "gender", type = "qual", palette = 8)
```

10.2 `scale_fill_manual`

- your manually picked colors in a character vector
- use either their names or HEX codes
- easy to pick here https://r-charts.com/colors/

```
# run repeatedly and explore the fill options with Help
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.7)) +
  #scale_fill_discrete(type = c("red", "blue", "grey", "grey34"))
  scale_fill_discrete(type = c("#A52A2A", "#4F94CD", "snow3", "#8B5A2B"))
```

11 To become a fill/color guru

- read https://ggplot2-book.org/scales-colour#sec-colour-theory
- ggplot2 functions implement color theory
  - which colors match/contrast
  - how they are rendered in computers

12 Add title to the entire plot

- with `labs`
- with `ggtitle`

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.7)) +
  labs(title = "Count of world's billionaires over 2002-2022",
       subtitle = "broken by sex")
```

```
billionaires_df %>% ggplot() +
  geom_bar(aes(x = time, fill = sex)) +
  scale_x_continuous(breaks = seq(from = 2002, to = 2022, by = 1)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.7)) +
  ggtitle(label = "Count of world's billionaires over 2002-2022",
          subtitle = "broken by sex")
```

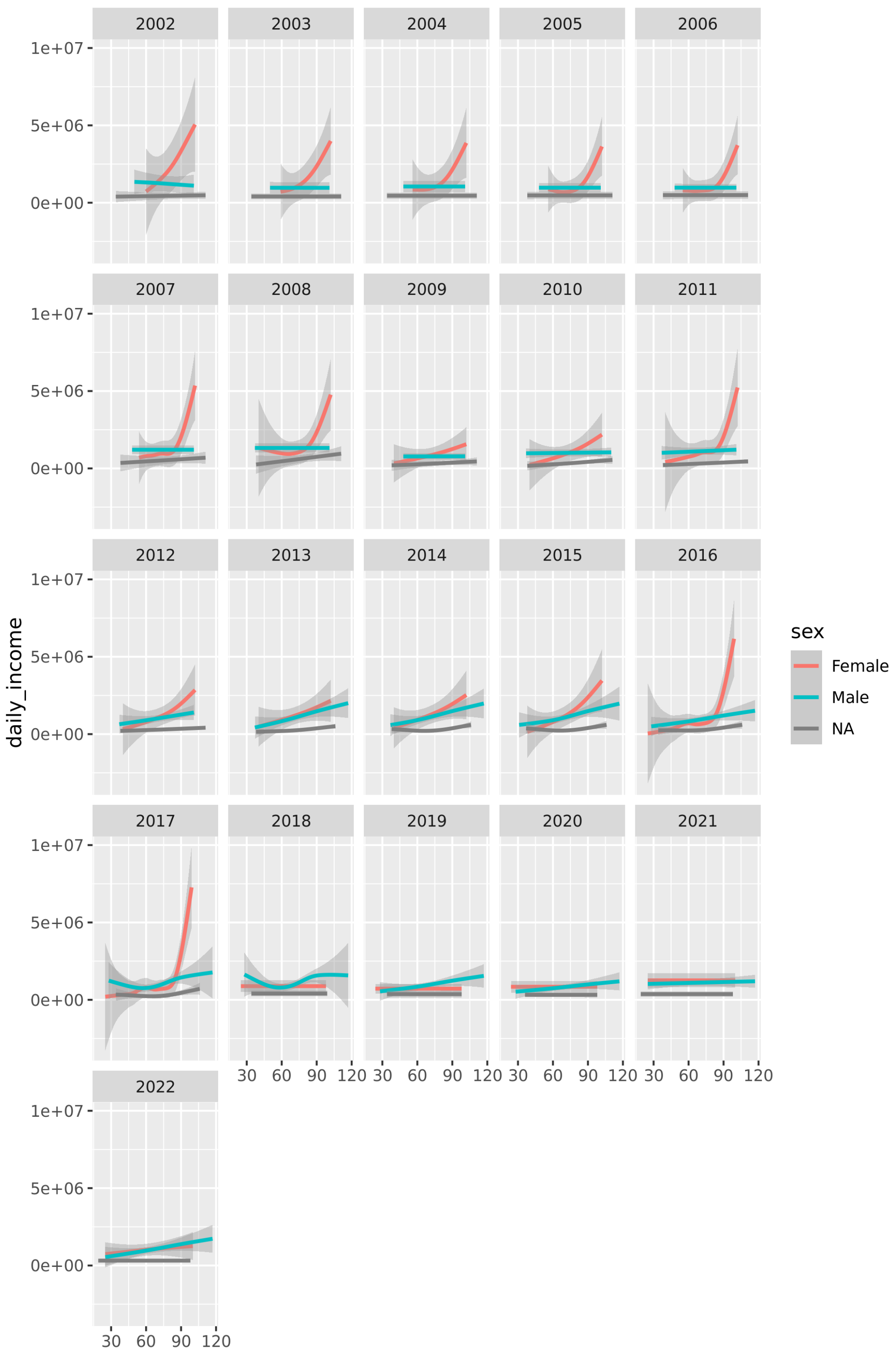


Figure 1: height increased to 270, units = mm

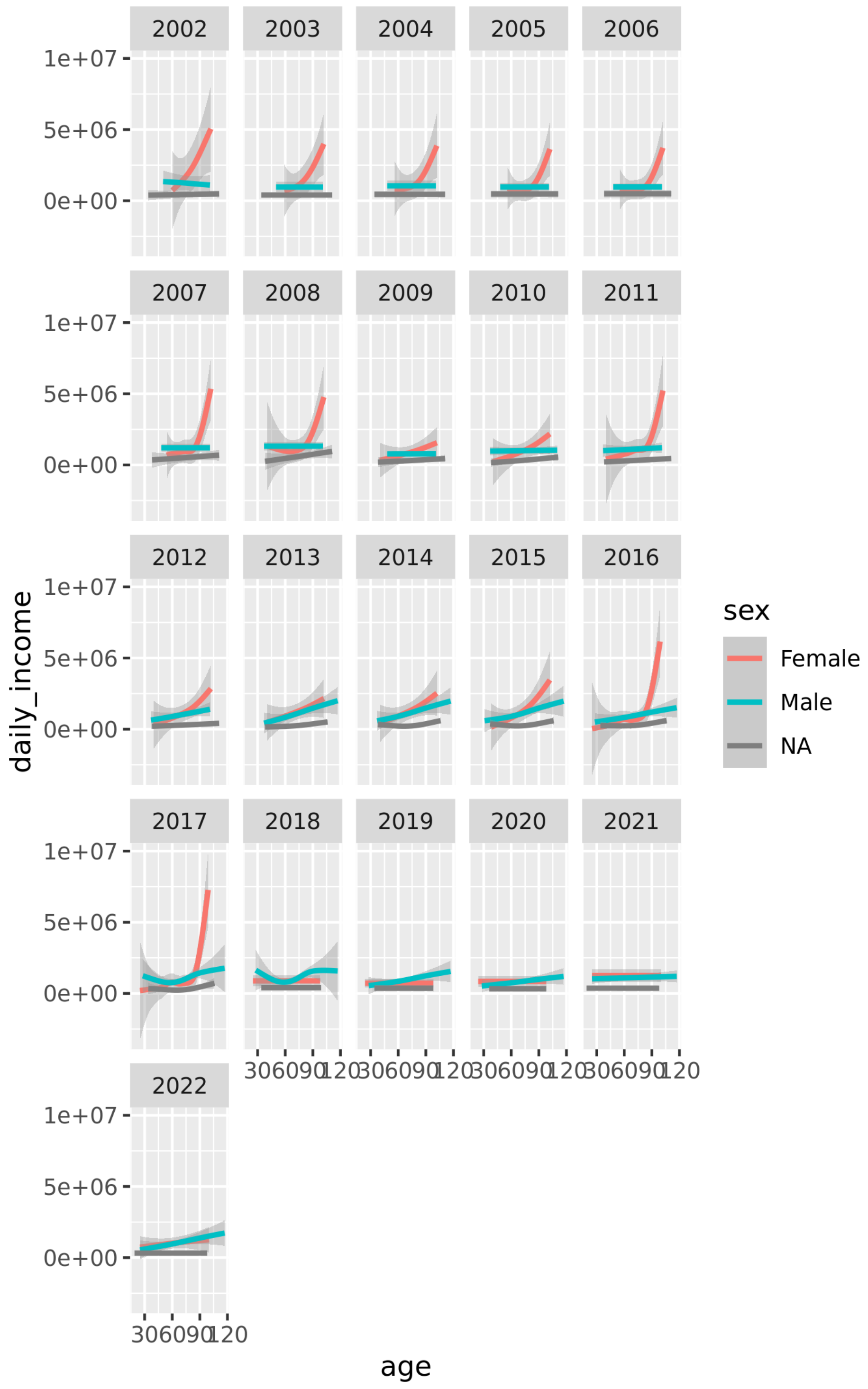
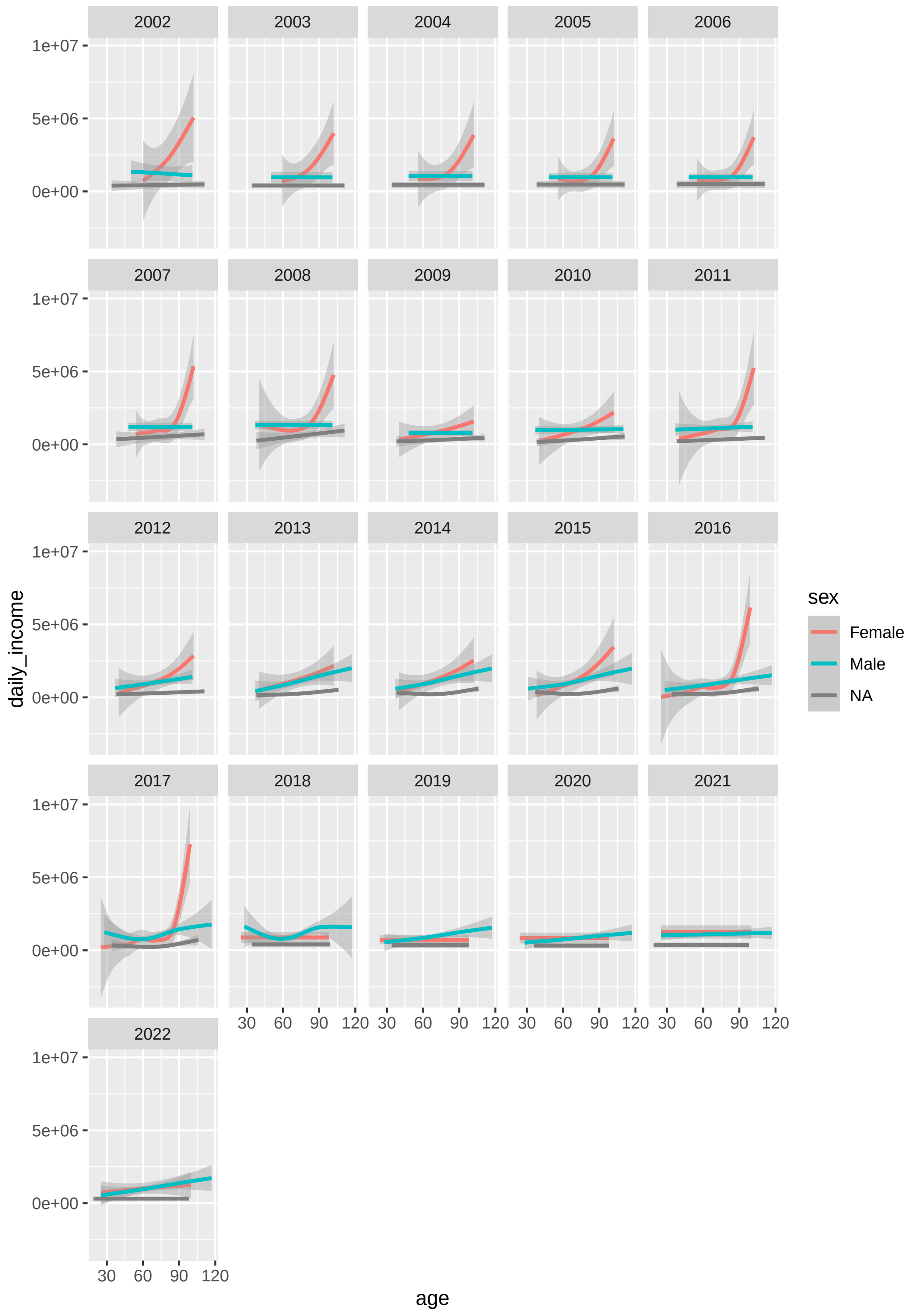
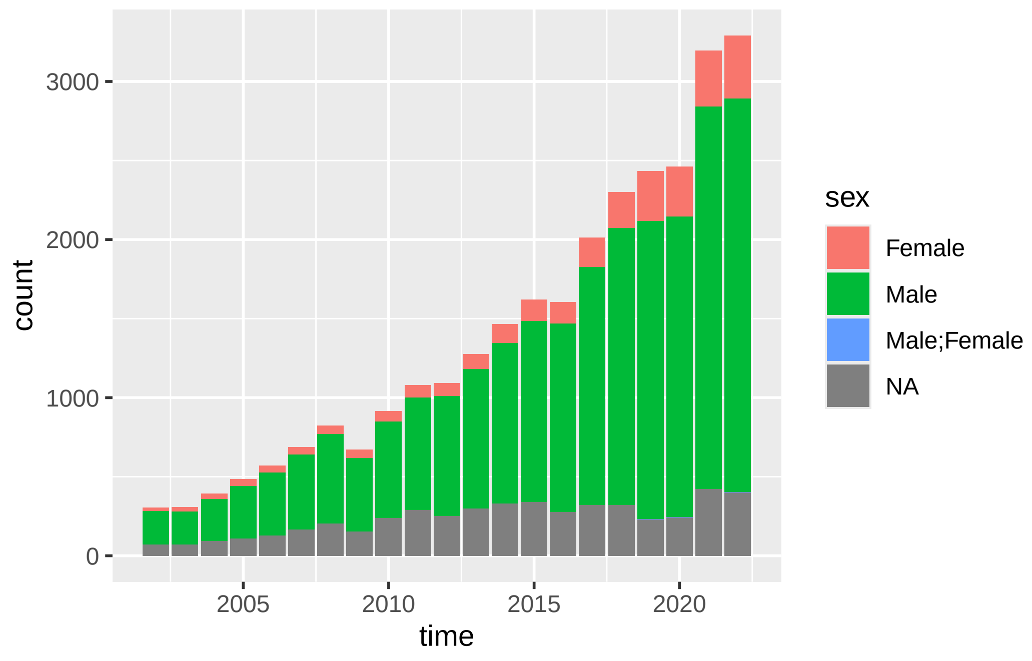
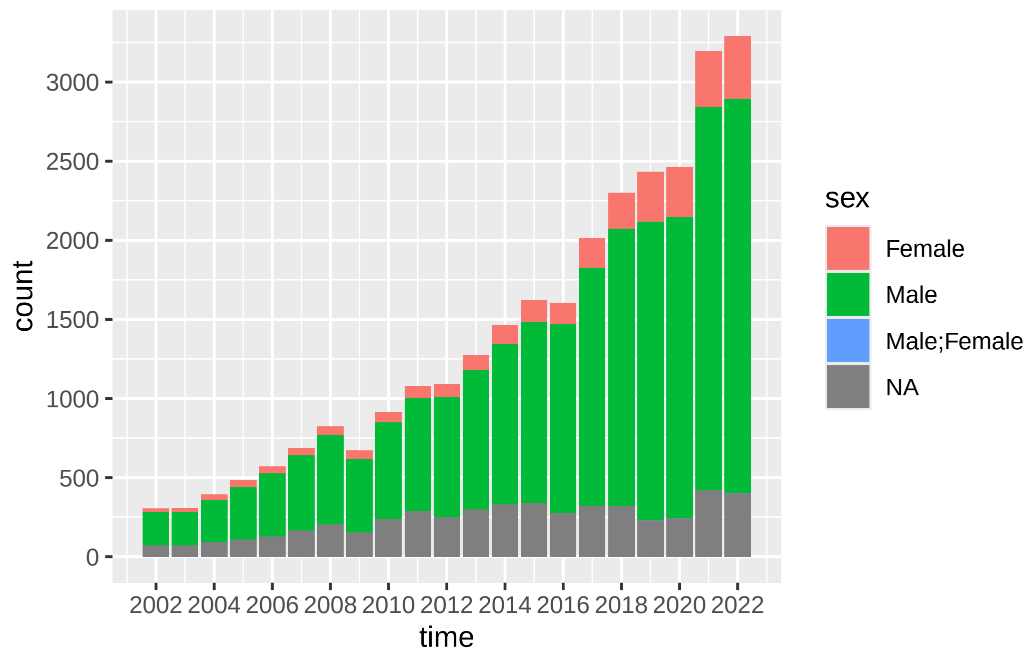


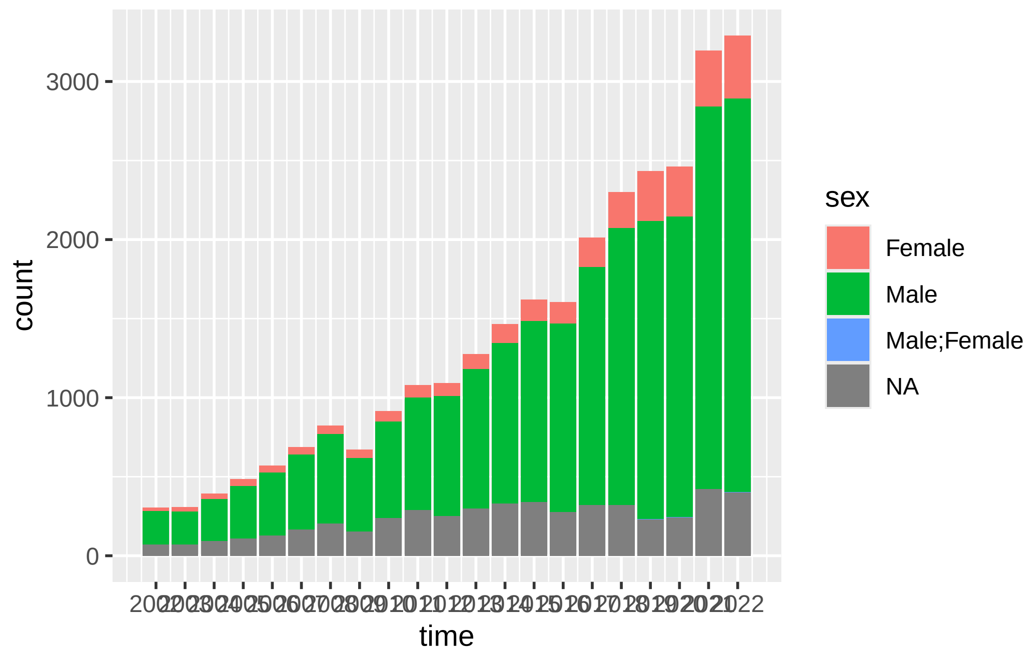
Figure 2: A monstrous plot produced by manual scaling of width and height

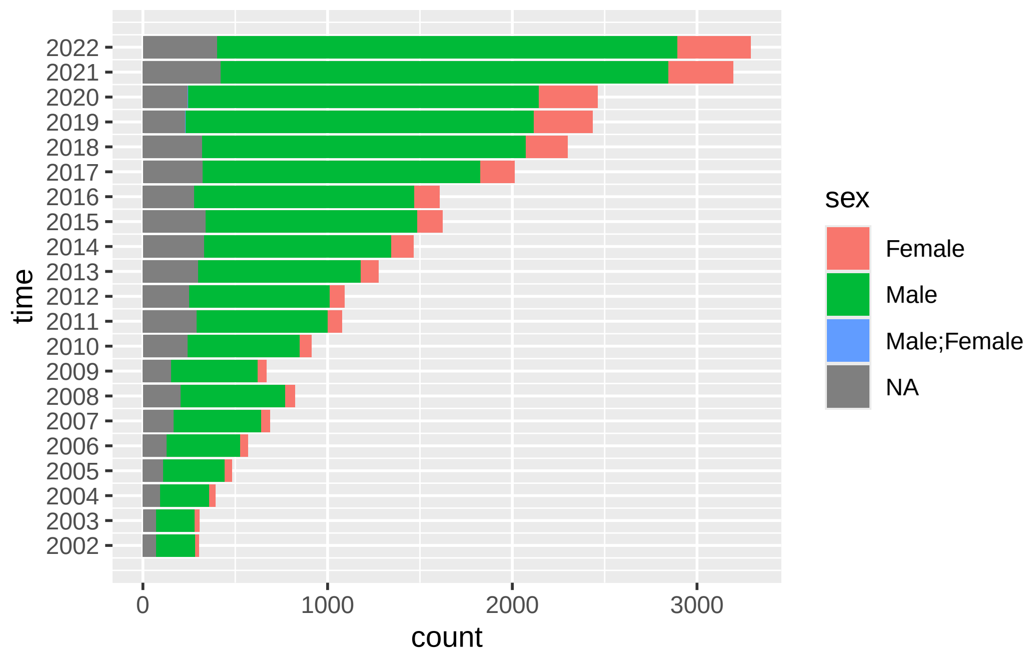


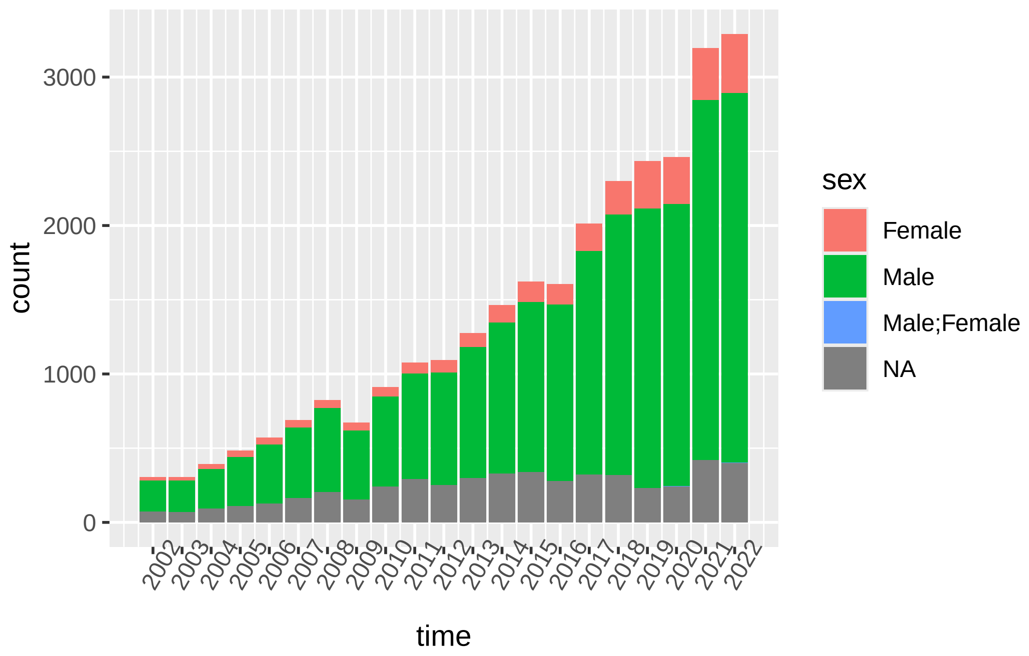
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
1e+07
5e+06
0e+00
daily_income
30
60
90
120
age
sex
Female
Male
NA

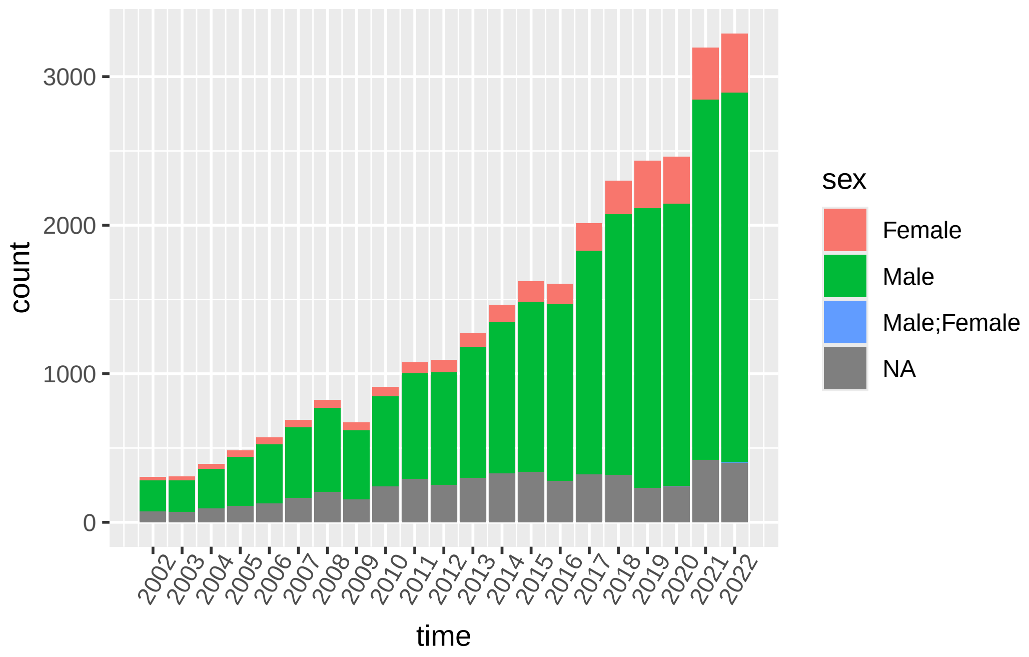

count
3000
2000
1000
0
2005
2010
2015
2020
time
sex
Female
Male
Male;Female
NA

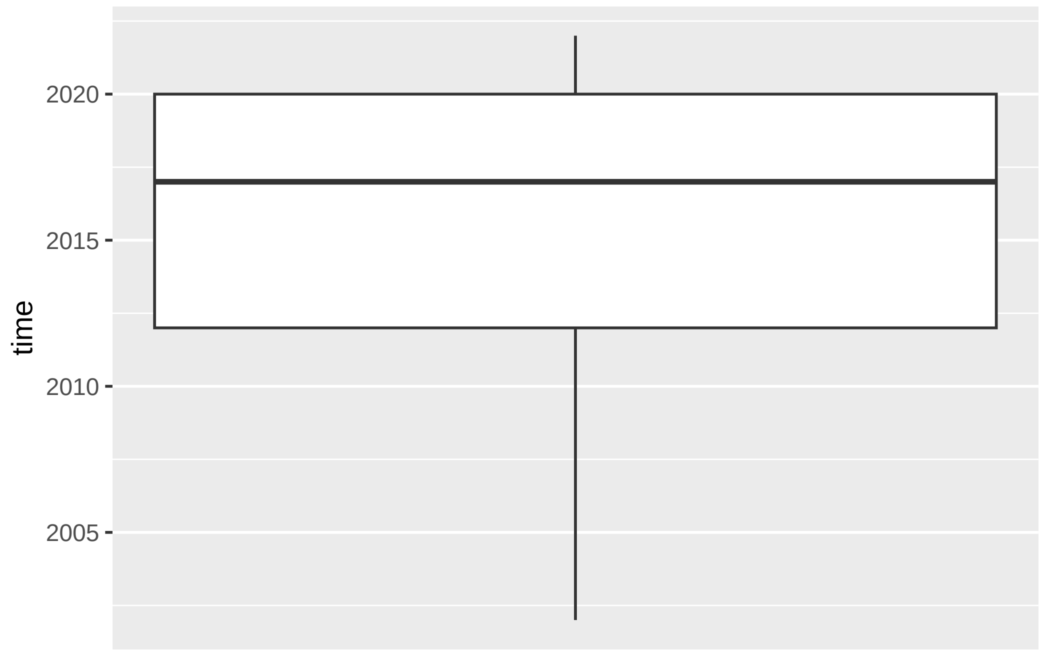

count
3000
2500
2000
1500
1000
500
0
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
time
sex
Female
Male
Male;Female
NA


3000
2000
1000
0
count
2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022
time
sex
Female
Male
Male;Female
NA

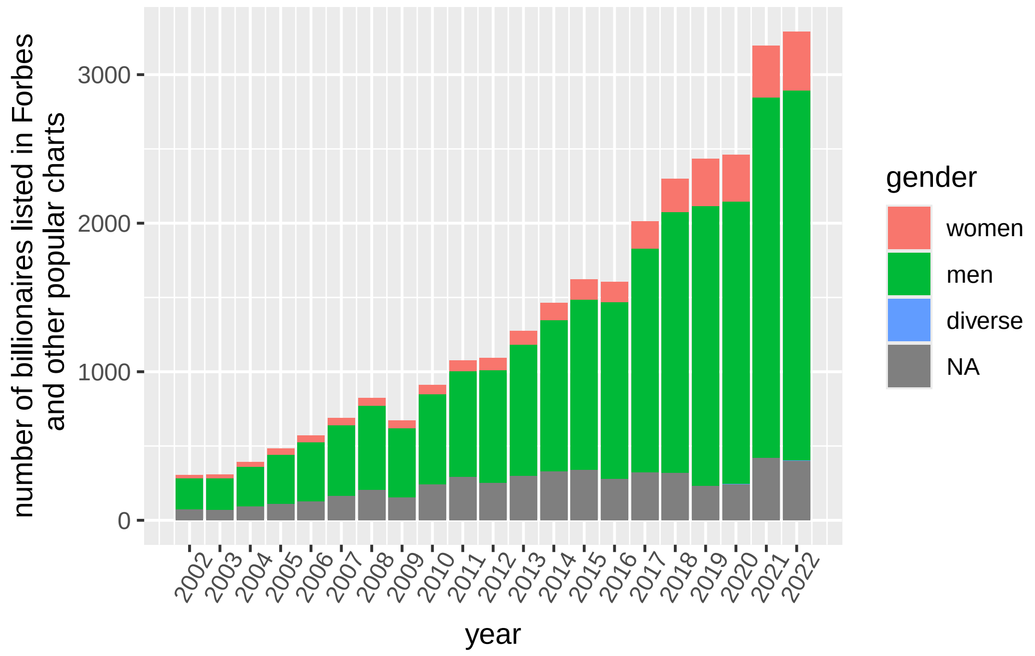

2022
2021
2020
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
2006
2005
2004
2003
2002
time
0
1000
2000
3000
count
sex
Female
Male
Male;Female
NA


count
3000
2000
1000
0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
time
sex
Female
Male
Male;Female
NA

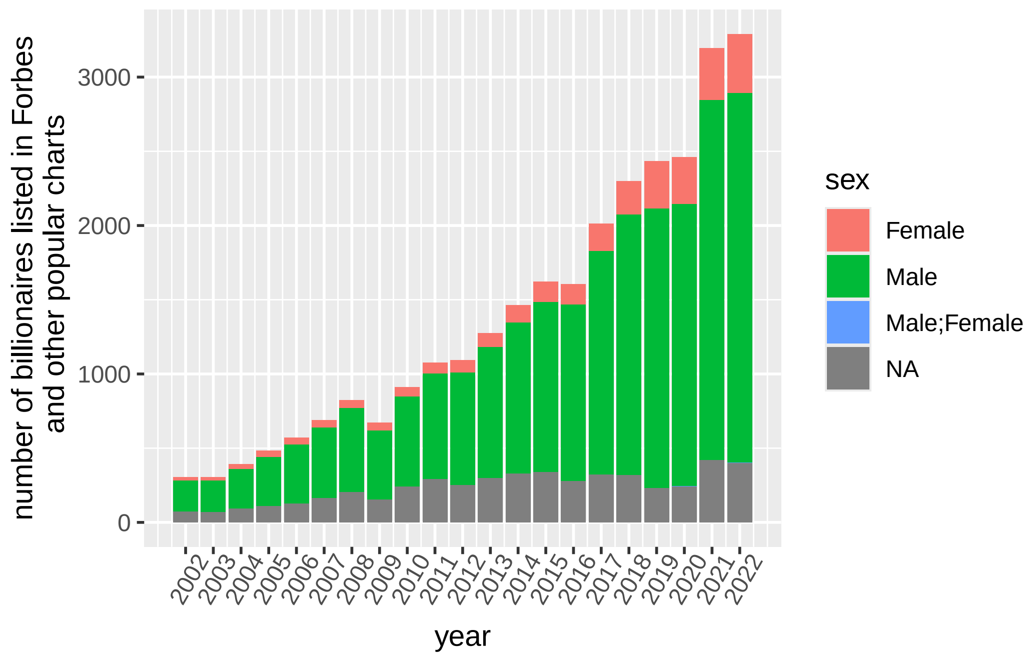

count
3000
2000
1000
0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
time
sex
Female
Male
Male;Female
NA

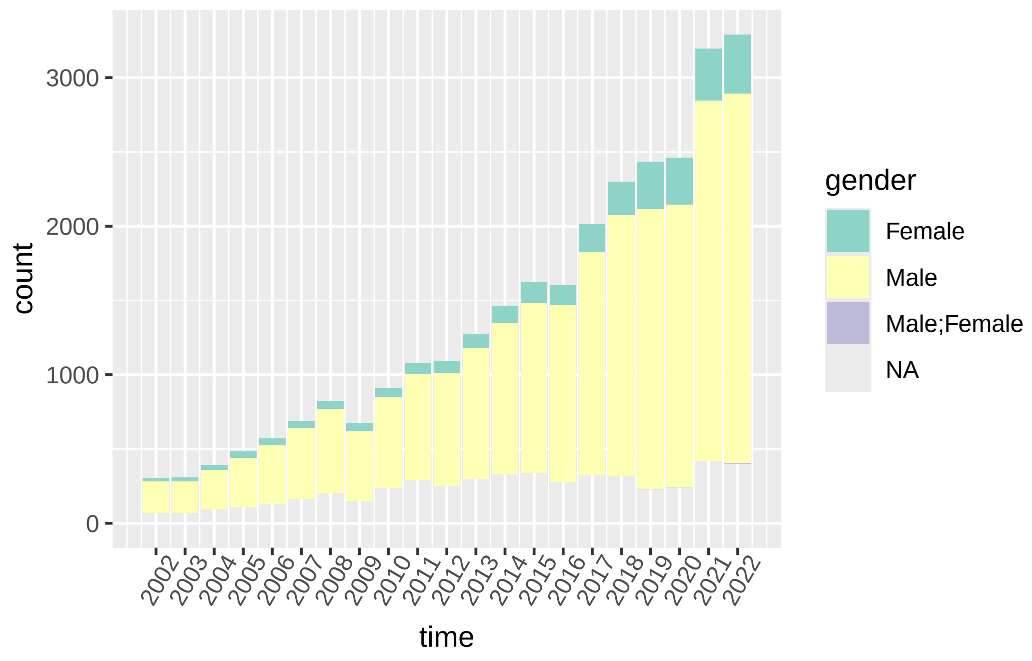


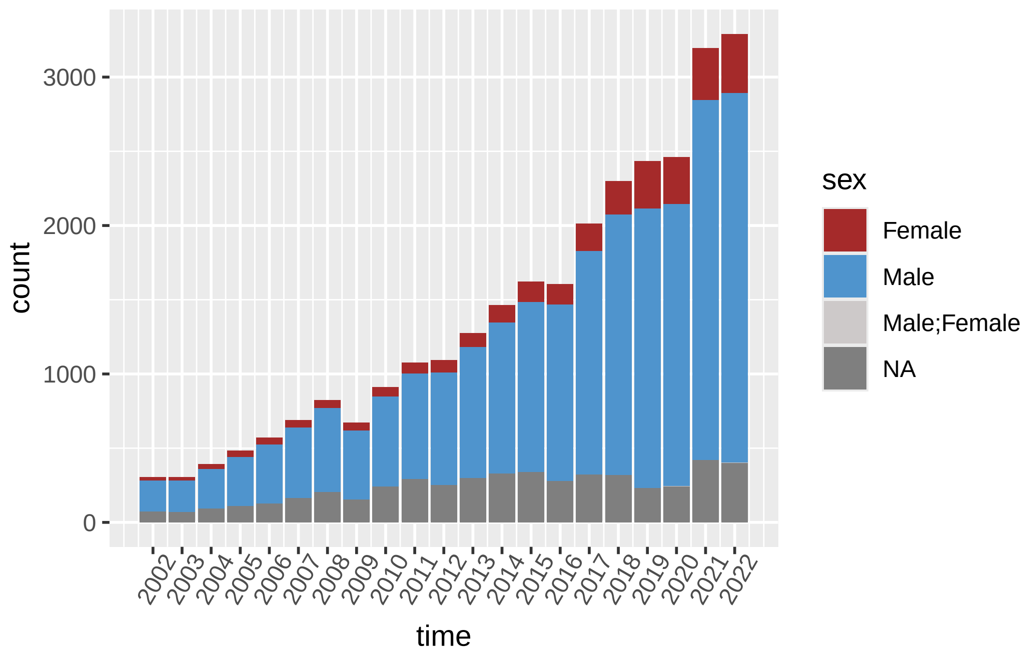
time
2020
2015
2010
2005

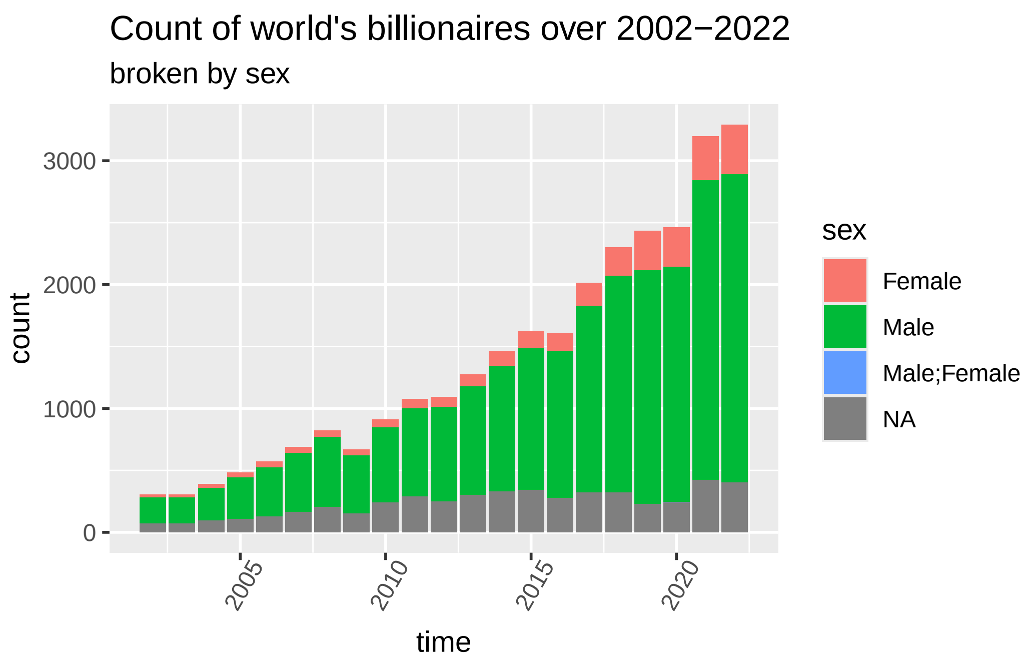


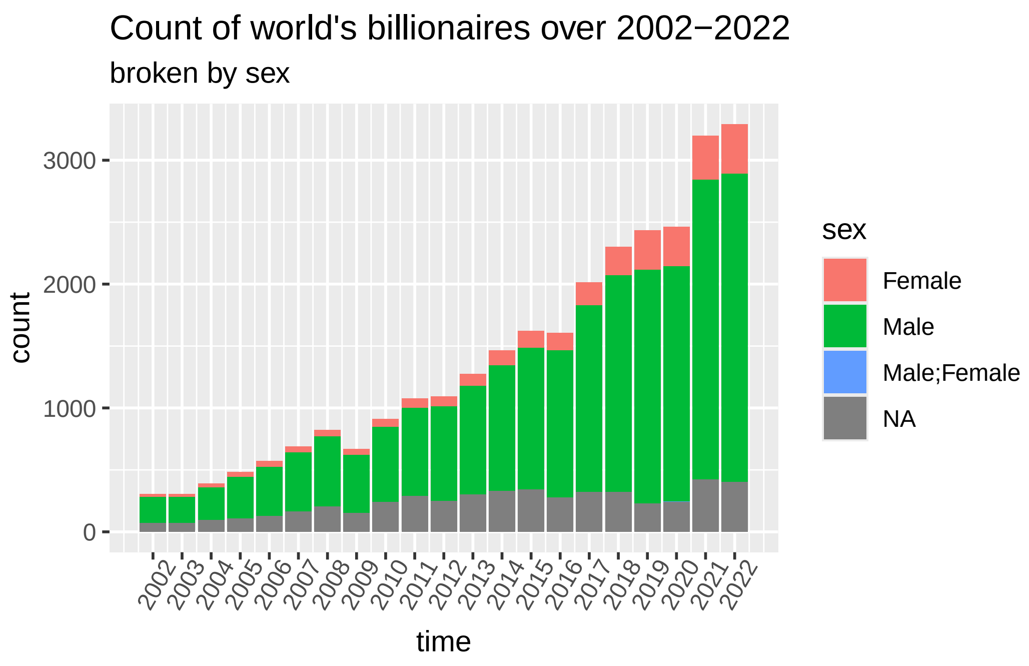
number of billionaires listed in Forbes and other popular charts
3000
2000
1000
0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
year
gender
women
men
diverse
NA


number of billionaires listed in Forbes and other popular charts
0
1000
2000
3000
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
year
sex
Female
Male
Male;Female
NA


count
0
1000
2000
3000
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
time
gender
Female
Male
Male;Female
NA


count
3000
2000
1000
0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
time
sex
Female
Male
Male;Female
NA


Count of world's billionaires over 2002−2022
broken by sex
count
3000
2000
1000
0
2005
2010
2015
2020
time
sex
Female
Male
Male;Female
NA


Count of world's billionaires over 2002−2022
broken by sex
count
3000
2000
1000
0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
time
sex
Female
Male
Male;Female
NA