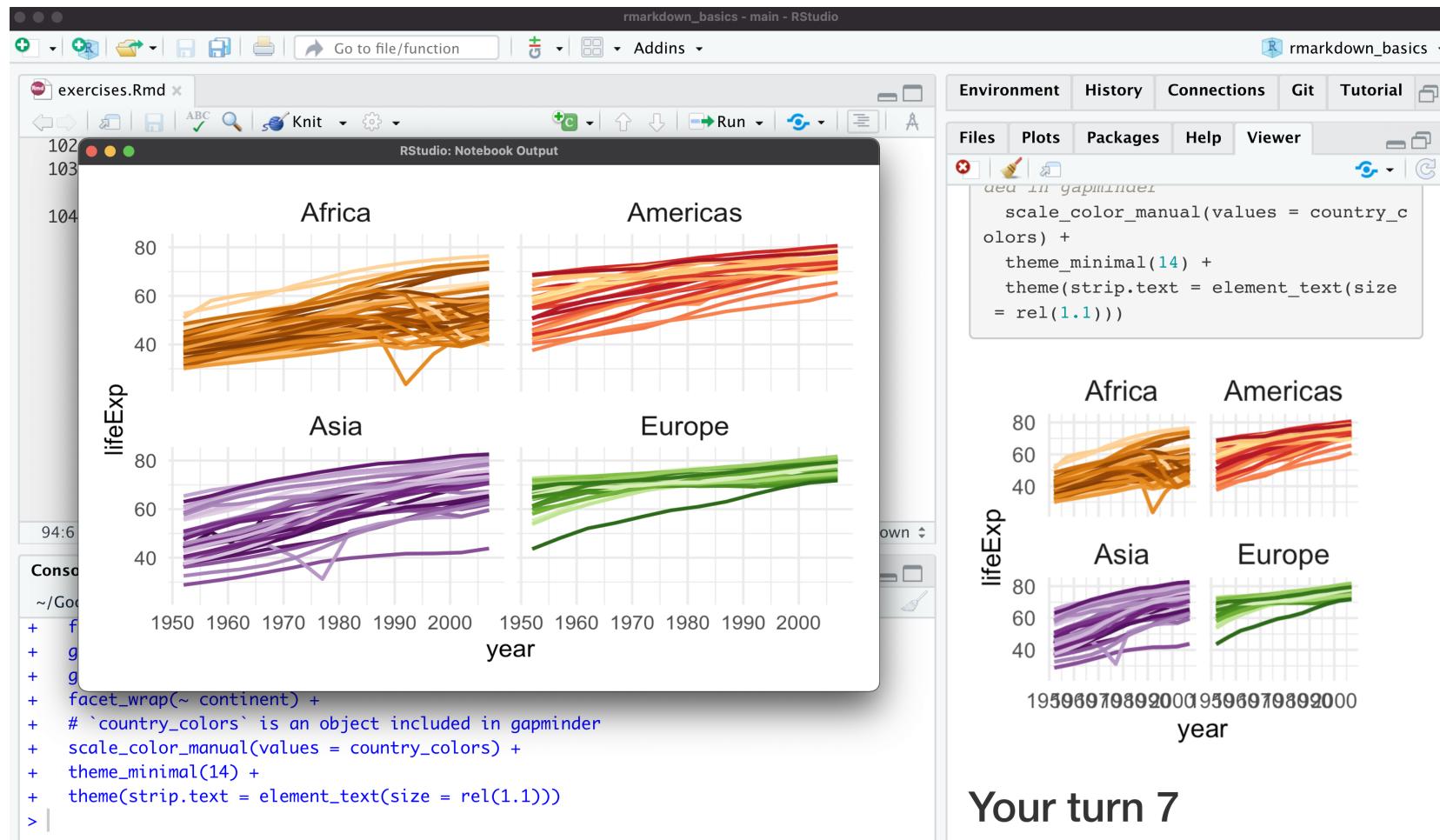


# Dynamic documents in R

## Making figures in R Markdown

2021-06-04



# What goes into a figure?

**Absolute size:** physical dimensions (inches, cm, etc)

**Pixel size:** no inherent size!

**Resolution:** pixels per inch (ppi) or dots per inch (dpi); links absolute & pixel size

**Pointsize:** absolute text size (1 pt = 1/72 inch)

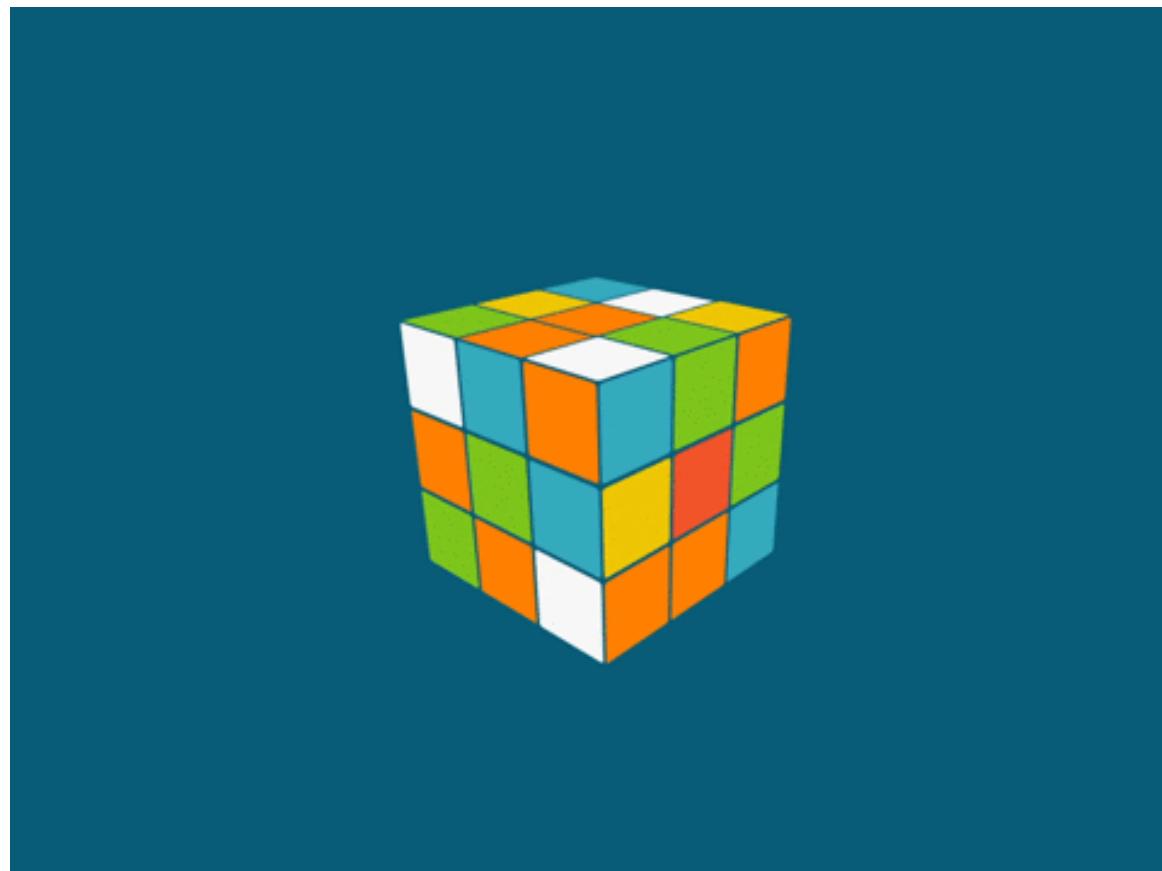
**Plot theming and aesthetics:** choices about text size, line size, margins, and so on.

# Essential options

Chunk Option	Controls
<code>fig.height</code>	Rendered figure height (in)
<code>fig.width</code>	Rendered figure width (in)
<code>fig.asp</code>	Rendered figure aspect ratio (use with ONE of height or width)
<code>dpi</code>	Resolution
<code>out.height</code>	Figure container height (in)
<code>out.width</code>	Figure container width (in)

See all of them at <http://yihui.name/knitr/options/>

# Tweaking figure options



# Getting a figure to look good in RStudio, Word, and slides



# A few reasonable defaults

```
knitr::opts_chunk$set(  
  echo = FALSE,  
  dev = "ragg_png",  
  dpi = 320,  
  out.width = "80%",  
  fig.width = 6,  
  fig.asp = 0.618,  
  fig.retina = 2,  
  fig.align = "center",  
  fig.show = "hold"  
)
```

Inspired by [R for Data Science](#) and [Jumping Rivers](#)

# A few reasonable defaults

```
knitr::opts_chunk$set(  
  echo = FALSE,  
  dev = "ragg_png",  
  dpi = 320,  
  out.width = "80%",  
  fig.width = 6,  
  fig.asp = 0.618,  
  fig.retina = 2,  
  fig.align = "center",  
  fig.show = "hold"  
)
```

Inspired by [R for Data Science](#) and [Jumping Rivers](#)

# A few reasonable defaults

```
knitr::opts_chunk$set(  
  echo = FALSE,  
  dev = "ragg_png",  
  dpi = 320,  
  out.width = "80%",  
  fig.width = 6,  
  fig.asp = 0.618,  
  fig.retina = 2,  
  fig.align = "center",  
  fig.show = "hold"  
)
```

Inspired by [R for Data Science](#) and [Jumping Rivers](#)

# A few reasonable defaults

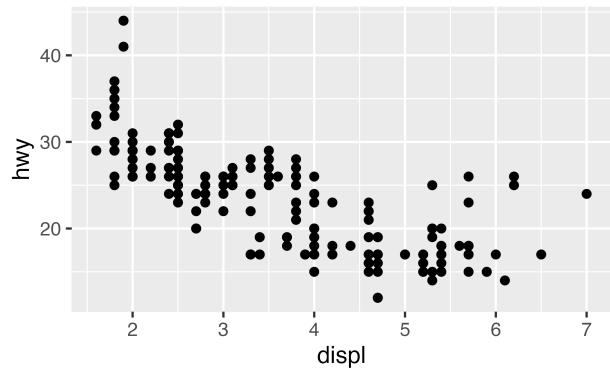
```
knitr::opts_chunk$set(  
  echo = FALSE,  
  dev = "ragg_png",  
  dpi = 320,  
  out.width = "80%",  
  fig.width = 6,  
  fig.asp = 0.618,  
  fig.retina = 2,  
  fig.align = "center",  
  fig.show = "hold"  
)
```

Inspired by [R for Data Science](#) and [Jumping Rivers](#)

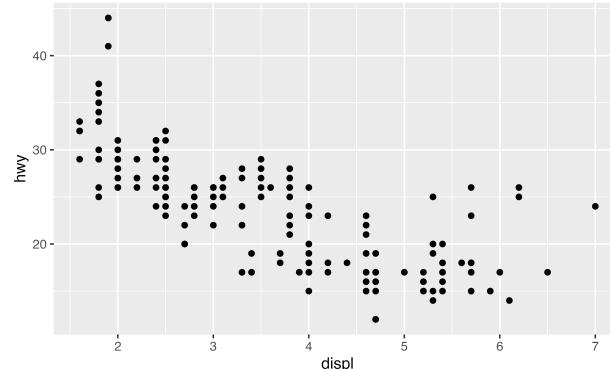
# Plot scaling

```
ggplot(mpg, aes(displ, hwy)) + geom_point()
```

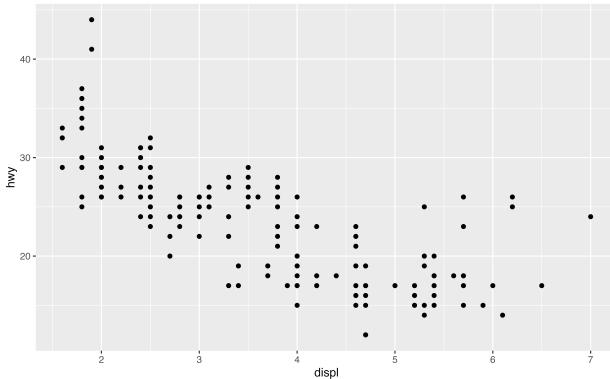
`figure.width = 4`



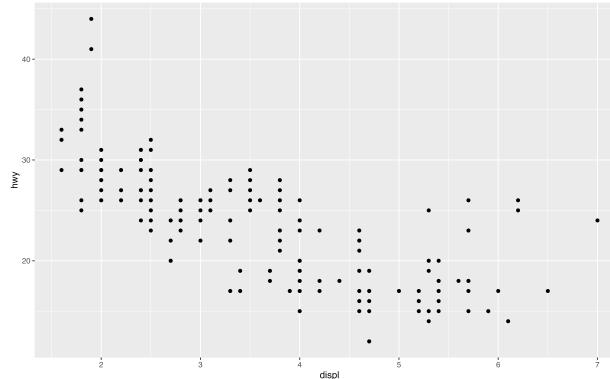
`figure.width = 6`



`figure.width = 8`



`figure.width = 10`



# Scaling saved files

`ggsave()`: Set the scale option

`ragg::agg_png()`: Set the scaling option

**Warning: these arguments work differently from one another!**

# A few reasonable defaults

```
knitr::opts_chunk$set(  
  echo = FALSE,  
  dev = "ragg_png",  
  dpi = 320,  
  out.width = "80%",  
  fig.width = 6,  
  fig.asp = 0.618,  
  fig.retina = 2,  
  fig.align = "center",  
  fig.show = "hold"  
)
```

Inspired by [R for Data Science](#) and [Jumping Rivers](#)

# ragg: AGG Graphic Devices



# **ragg: AGG Graphic Devices**



**Faster than grDevices or Cairo**

# ragg: AGG Graphic Devices



Faster than grDevices or Cairo

**Better system font access and text rendering**

# **ragg: AGG Graphic Devices**

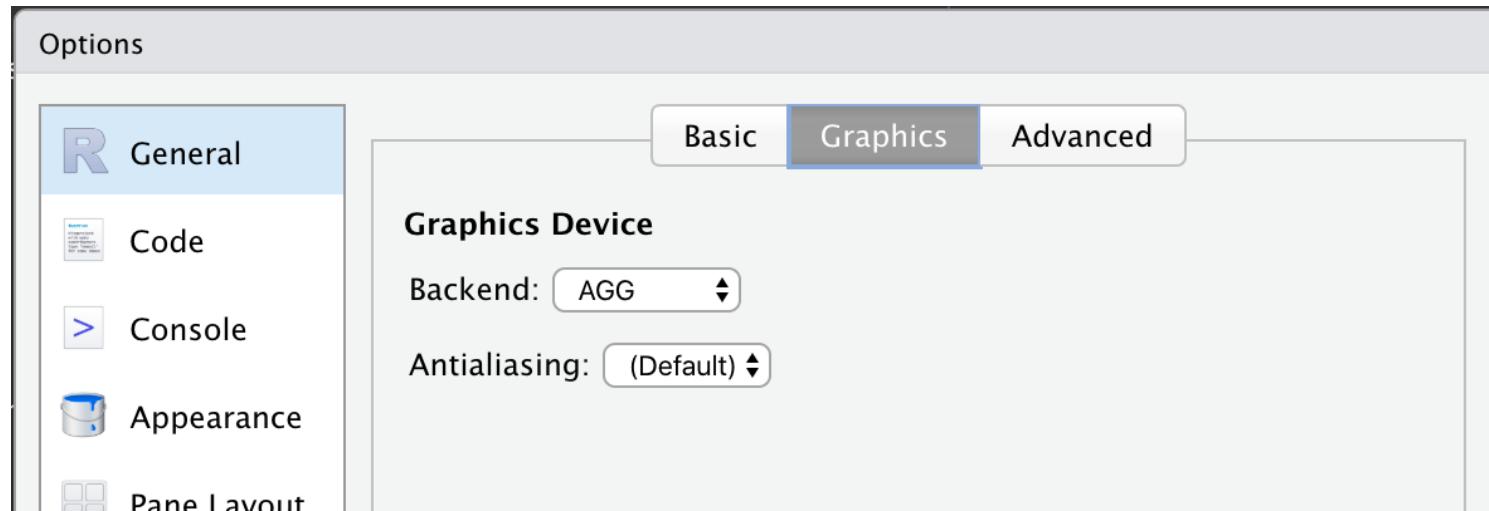


Faster than grDevices or Cairo

Better system font access and text rendering

**System independent rendering**

# Setting ragg as your default in RStudio



This sets the default for the **viewer**, not R Markdown

## Your Turn 1

**Using the chunk option defaults we discussed, set the global chunk options using  
knitr::opts\_chunk\$set()**

**Knit this document, and take a look at the first three figures. Do you like how they look?**

**Modify fig.width for each chunk until you're satisfied**

# What affects ggplot2 sizing?

- 1 geoms
- 2 themes
- 3 scales and axes
- 4 clipping

## Theme sizing

**ggplot2 themes all have a `base_size` argument, e.g.**

```
theme_minimal(base_size = 14)
```

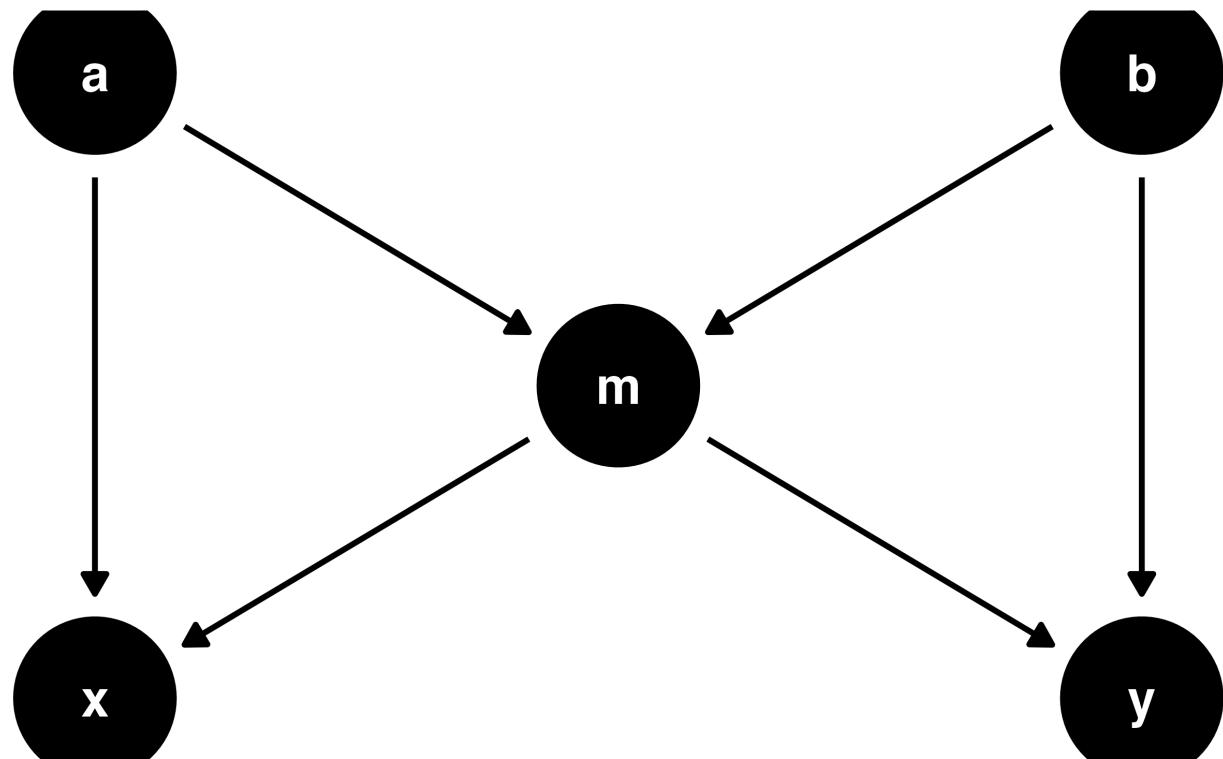
# Theme sizing

**ggplot2 themes all have a base\_size argument, e.g.**

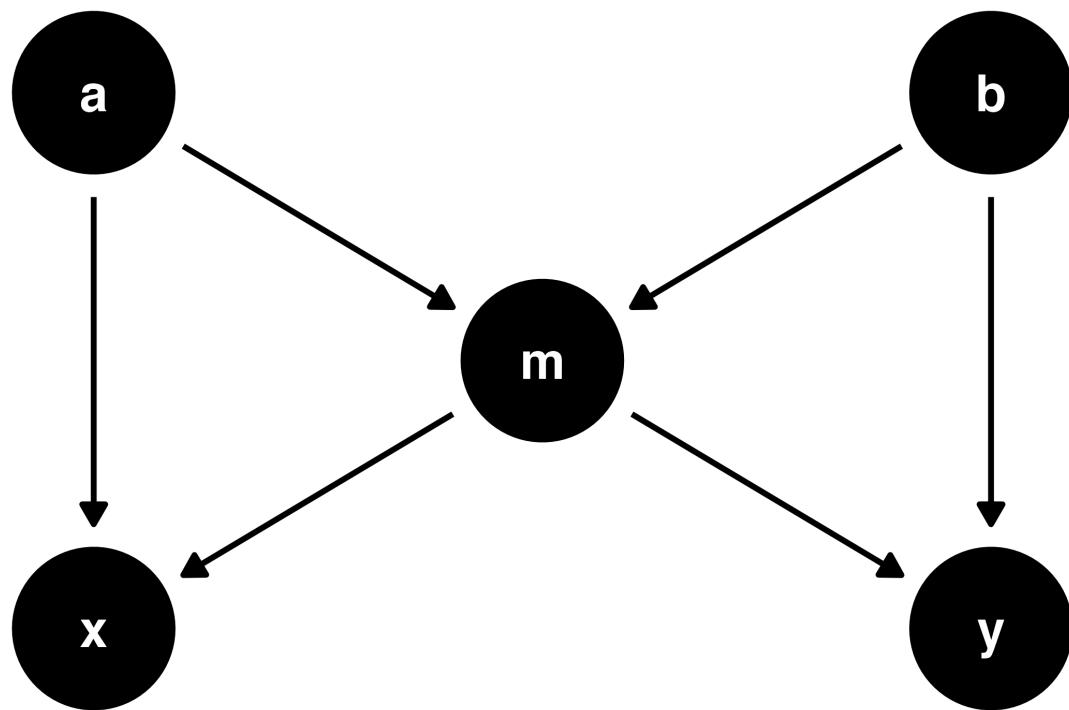
```
theme_minimal(base_size = 14)
```

**Consider well-proportioned cowplot themes, e.g. `theme_minimal_grid()`**

# Expanding scales (fig.width = 4)

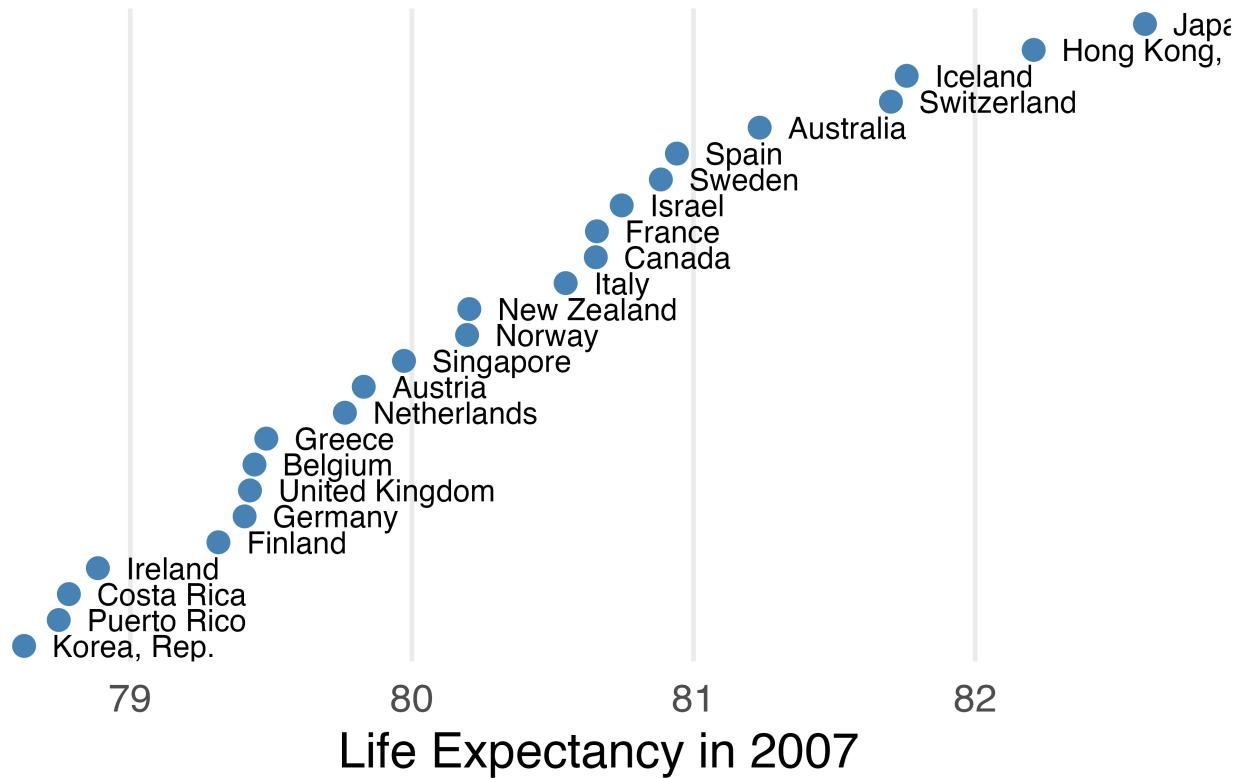


```
library(ggdag, warn.conflicts = FALSE)
ggdag(butterfly_bias()) +
  theme_dag() +
  scale_x_continuous(expand = expansion(.2)) +
  scale_y_continuous(expand = expansion(.2))
```



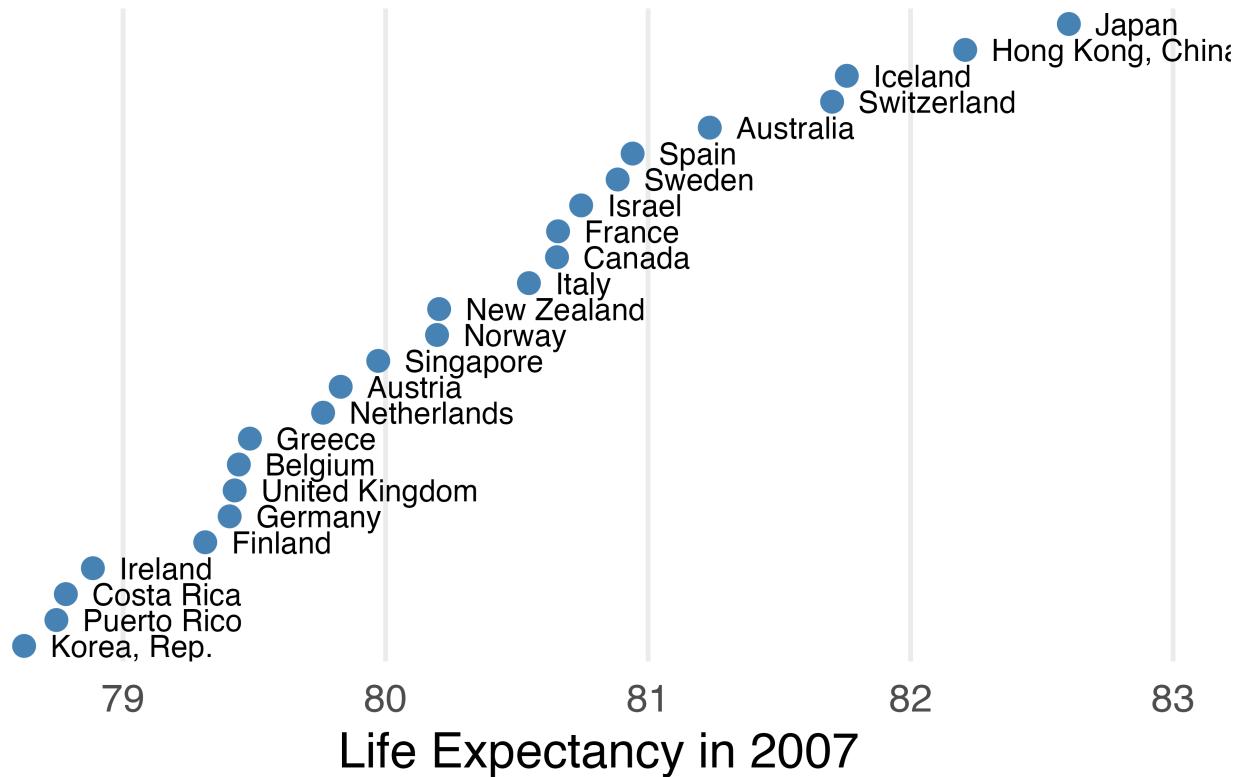
```
p <- gapminder %>%
  filter(year == 2007) %>%
  slice_max(lifeExp, n = 25) %>%
  mutate(country = fct_rev(fct_inorder(fct_drop(country)))) %>%
  ggplot(aes(lifeExp, country)) +
  geom_point(size = 3, color = "steelblue") +
  geom_text(aes(label = country), hjust = 0, nudge_x = .1, size = 3.5) +
  theme_minimal(16) +
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_blank(),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank()
  ) +
  xlab("Life Expectancy in 2007")
```

p

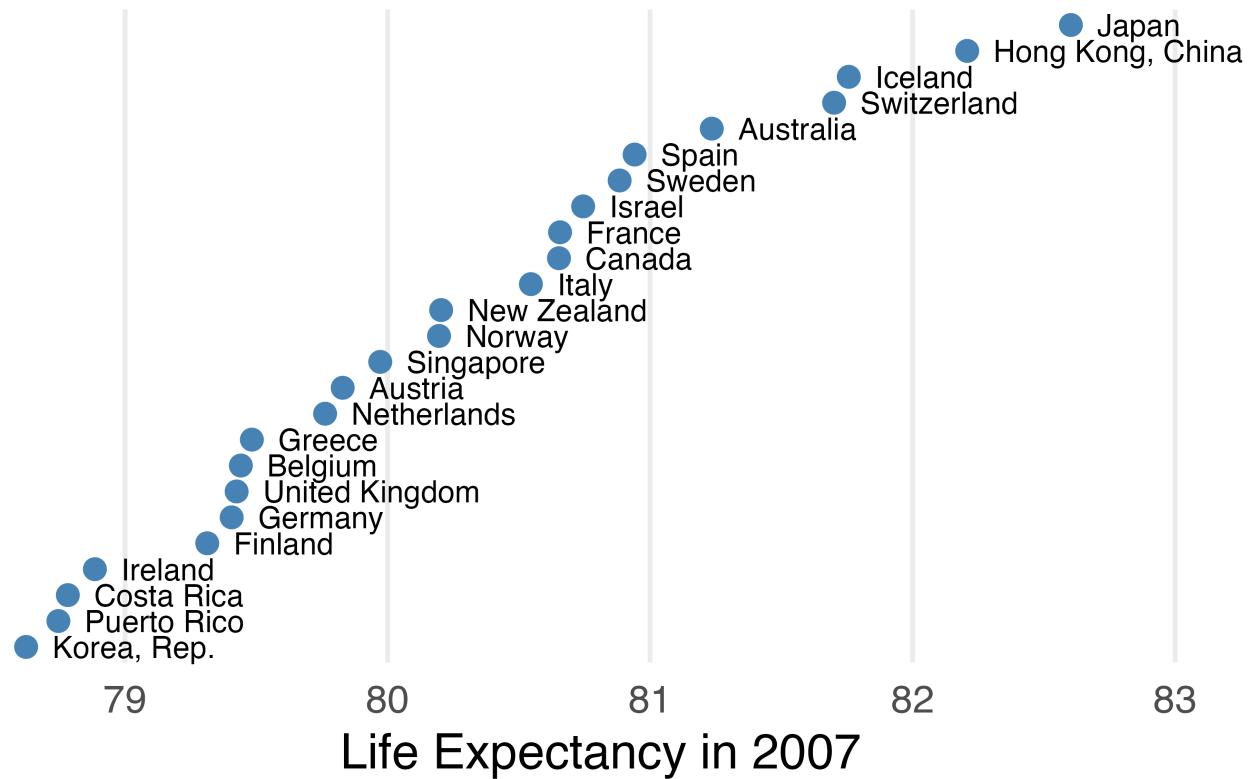


p +

xlim(NA, 83)



```
p +  
  xlim(NA, 83) +  
  coord_cartesian(clip = "off")
```



# Specify where R Markdown writes figures

```
fig.path = "folder/prefix-"
```

# Specify where R Markdown writes figures

```
fig.path = "folder/prefix-"
```

**Use with chunk names!**

here: find your PATH!



## Detour: The here package

Find files from the root up, particularly  
with **RStudio projects**

```
here("data", "file.csv")
```

# Detour: The here package

Find files from the root up, particularly  
with RStudio projects

```
here("data", "file.csv")
```

**Really convenient with Rmd, which sets  
a local directory**

See [Why should I use the here package when I'm already using projects?](#)

# **How do I create an RStudio Project again?**

**In RStudio: File > New Project**

# How do I create an RStudio Project again?

In RStudio: File > New Project

Or, in the console:

```
usethis::create_project("path/to/project")
```

```
my_project
| -- data
|   | -- data.csv
| -- figures
|   | -- figure1.png
| -- reports
|   | -- manuscript.Rmd
| -- R
|   | -- read_data.R
| -- my_project.Rproj
```

```
my_project
| -- data
|   | -- data.csv
| -- figures
|   | -- figure1.png
| -- reports
|   | -- manuscript.Rmd
| -- R
|   | -- read_data.R
| -- my_project.Rproj
```

```
source("../R/read_data.R")
```

```
read_csv("data/data.csv")
```

```
ggsave("../figures/figure1.png")
```

```
my_project
| -- data
|   | -- data.csv
| -- figures
|   | -- figure1.png
| -- reports
|   | -- manuscript.Rmd
| -- R
|   | -- read_data.R
| -- my_project.Rproj
```

```
source(here("R", "read_data.R"))
```

```
read_csv(here("data", "data.csv"))
```

```
ggsave(here("figures", "figure1.png"))
```

**Why here?**

**Works from the project up**

# Why here?

Works from the project up

**Robust to other ways people open and run your code**

# Why here?

Works from the project up

Robust to other ways people open and run your code

**Writes paths safely across operating systems**

## Your Turn 2

**Load the here package in the setup chunk. In the setup chunk, globally set fig.path to "figures/figure-" using here(). This will tell knitr to create figures in the "figures" folder with a prefix of "figures-".**

**Knit this document and take a look at the images in the figures folder.**

# Cross-referencing figures

A bookdown output format, e.g.  
`bookdown::html_document2`

A figure caption (`fig.cap = "Plot title"`)

A named code chunk (`{r chunk-name}`)

# Cross-referencing figures

A bookdown output format, e.g.  
bookdown::html\_document2

A figure caption (fig.cap = "Plot title")

A named code chunk ({r chunk-name})

Reference with \@ref(fig:chunk-name)

# Cross-referencing figures

A bookdown output format, e.g.  
bookdown::html\_document2

A figure caption (`fig.cap = "Plot title"`)

A named code chunk (`{r chunk-name}`)

Reference with `\@ref(fig:chunk-name)`

Also sets `fig.alt = fig.cap` (See Writing Alt Text for Data Visualization)

## Your Turn 3

Change the output type to use bookdown and cross-reference one of the figures above.

# Including external images

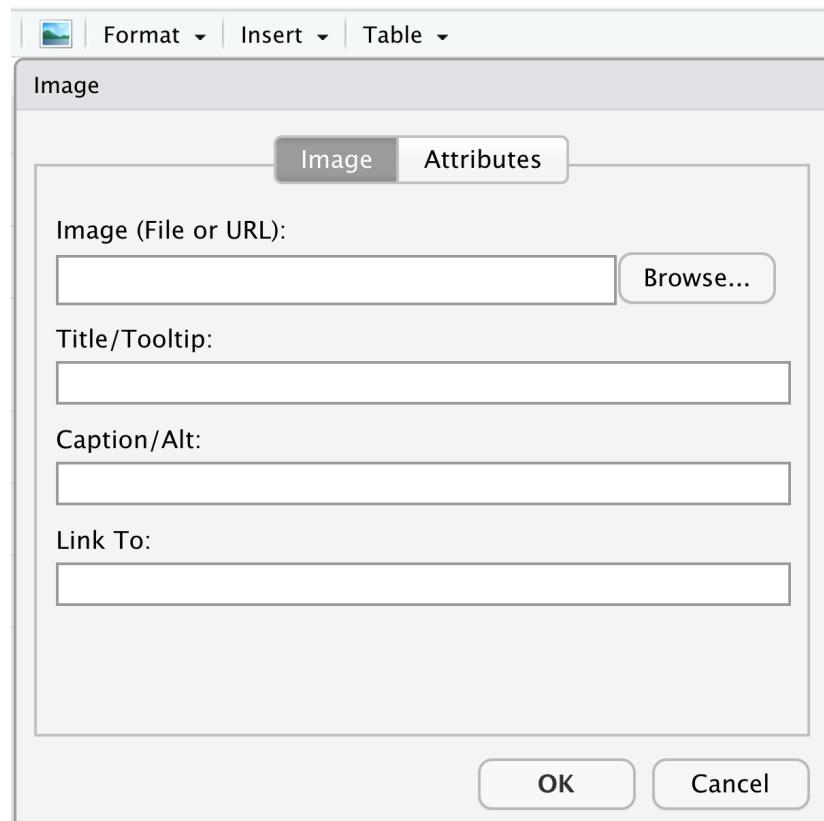
```
knitr::include_graphics("path/to/image")  
+ out.width
```

# Including external images

```
knitr::include_graphics("path/to/image")  
+ out.width
```

**include\_graphics() also accepts URLs**

# Including external images



![Alt text](path/to/image)

## Your Turn 4

**Include external\_img/r\_rollercoaster.png in the code chunk below.**

**Knit**

**Let's change a few chunk options: 1) Add a chunk name 2) Set fig.alt describing the image 3) Modify out.width to use a different percentage than the default.**

**Knit again**

# Resources

**R Markdown Cookbook:** Includes various recipes for figures

**Jumping Rivers Blog Series:** A blog series on images in R Markdown

**Taking Control of Plot Scaling:** A detailed blog on understanding scaling