

YOU WILL READ THIS FIRST.

And then you will read this line next.

You will go back to read this body copy if you want to know more. It takes the most effort to read because it has a lot of text in a small font in a light weight with tight line spacing. Many people will skip paragraphs like this unless if they aren't engaged right away. This is why it's important to draw attention to your message using visual hierarchy.

You'll probably
read this before
the paragraph.

1

YOU WILL READ THIS FIRST.

And then you will read this line next.

You will go back to read this body copy if you want to know more. It takes the most effort to read because it has a lot of text in a small font in a light weight with tight line spacing. Many people will skip paragraphs like this unless if they aren't engaged right away. This is why it's important to draw attention to your message using visual hierarchy.

You'll probably
read this before
the paragraph.

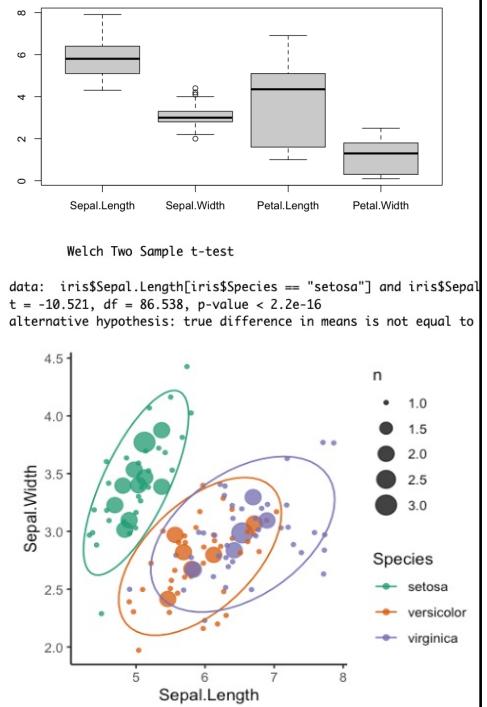
Download your Rmd file here: github.com/lauralwd/beyond_barcharts_workshop

2

Beyond bar charts

Data visualisation
workshop

Laura Dijkhuizen
Bioinformatics & Theoretical
Biology @UU



3

Challenger example

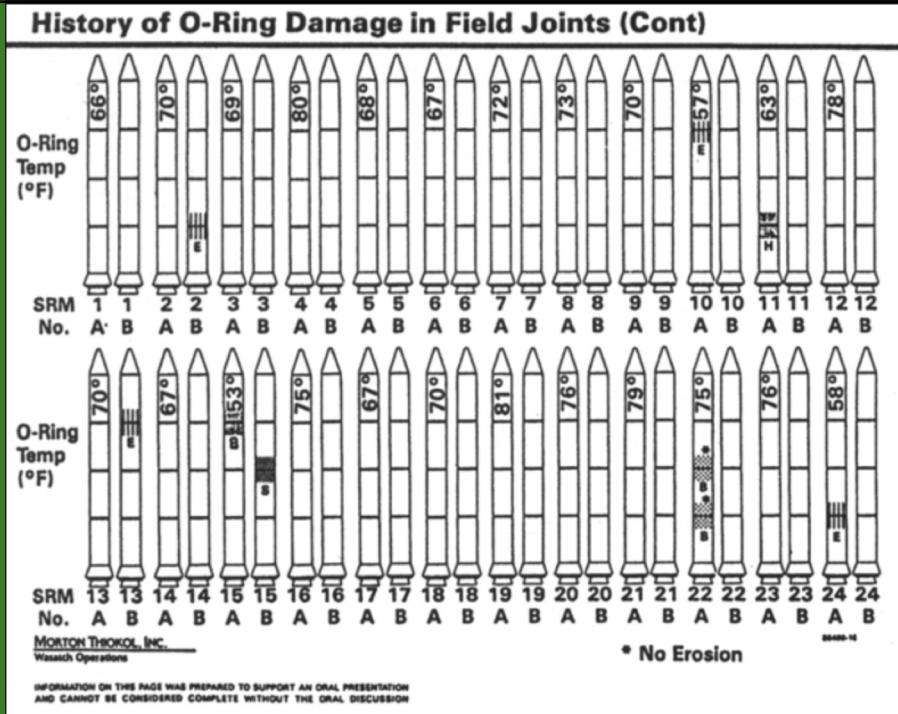
Challenger Space
Shuttle accident in
1986.



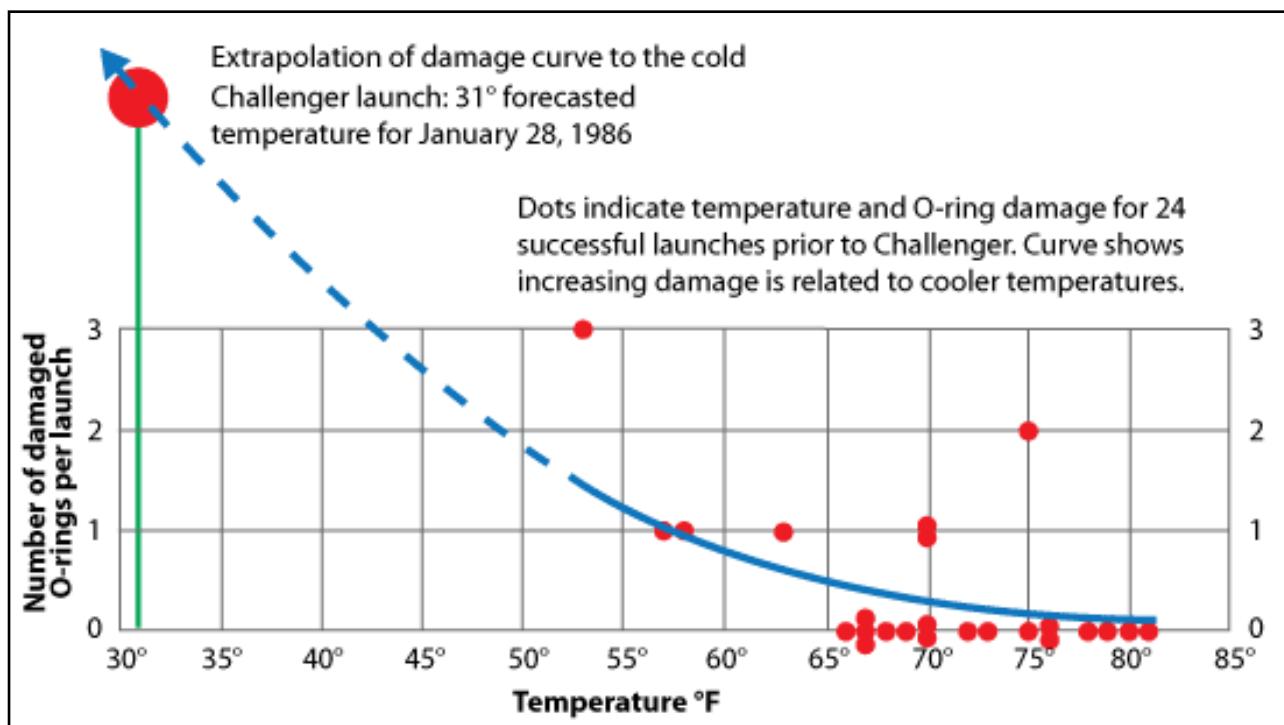
4

Challenger example:

Forecasted temperature:
31 °F



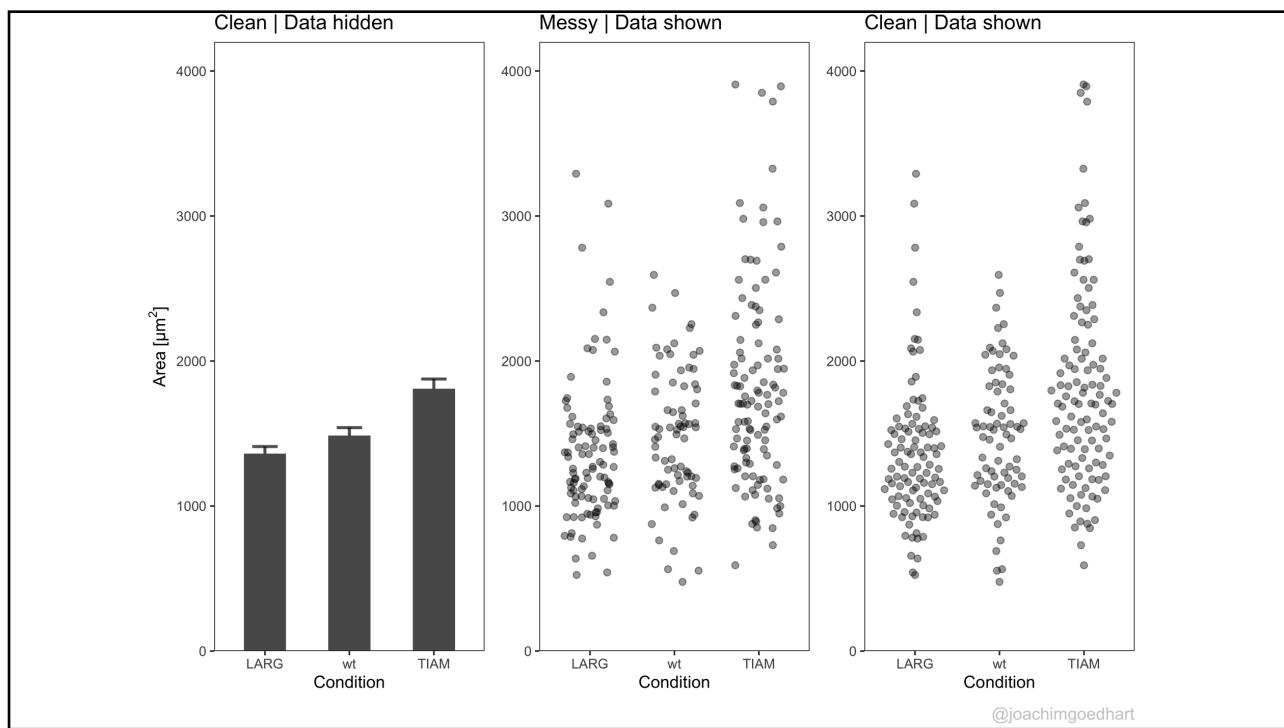
5



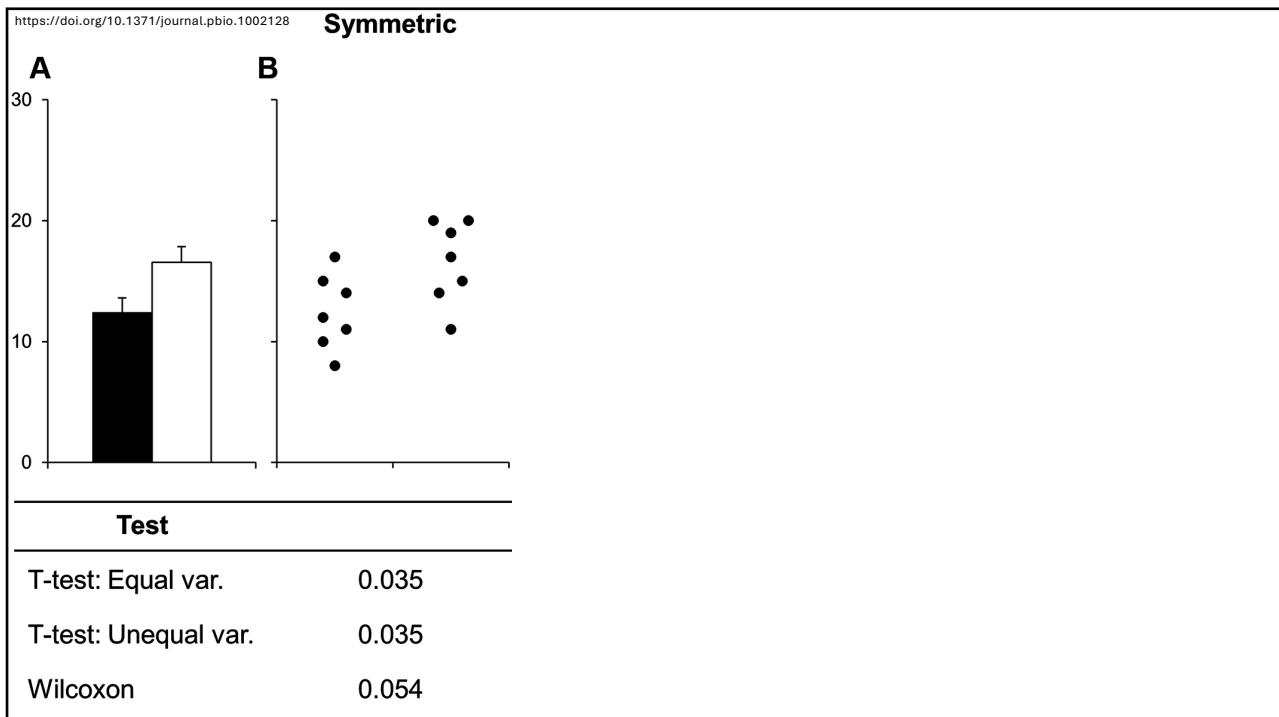
6

Let's look at some figures!

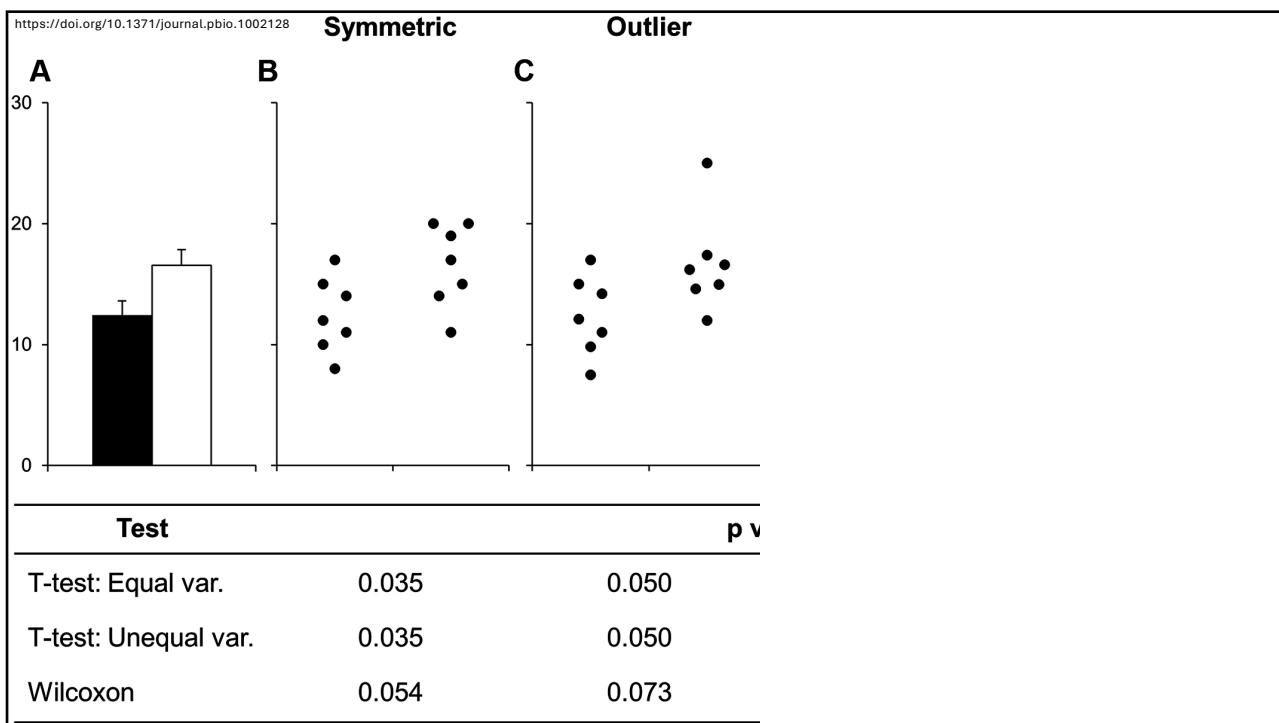
7



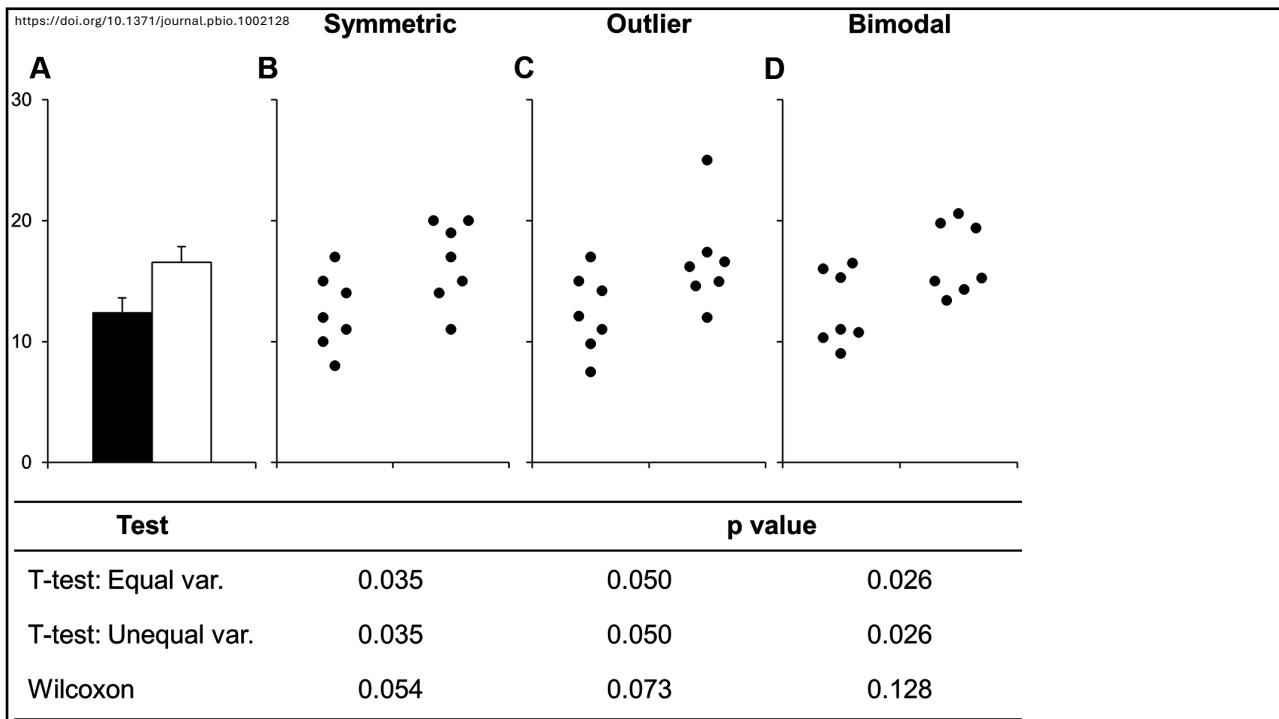
8



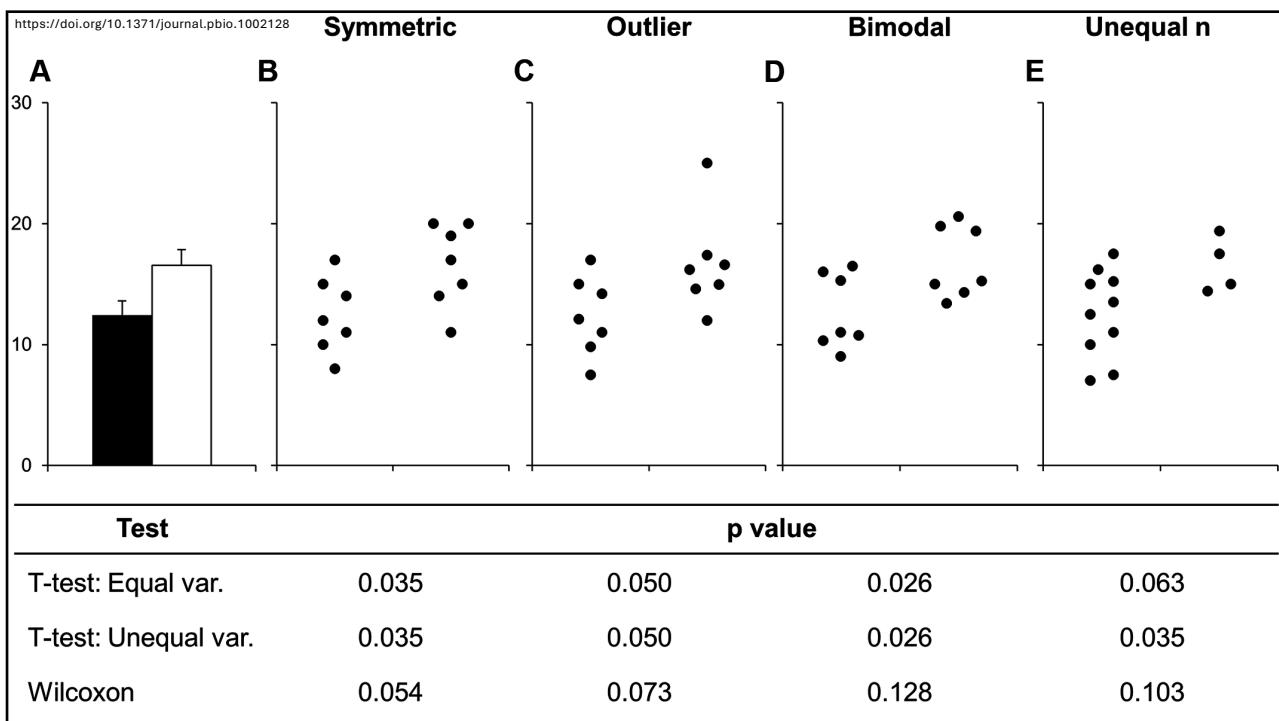
9



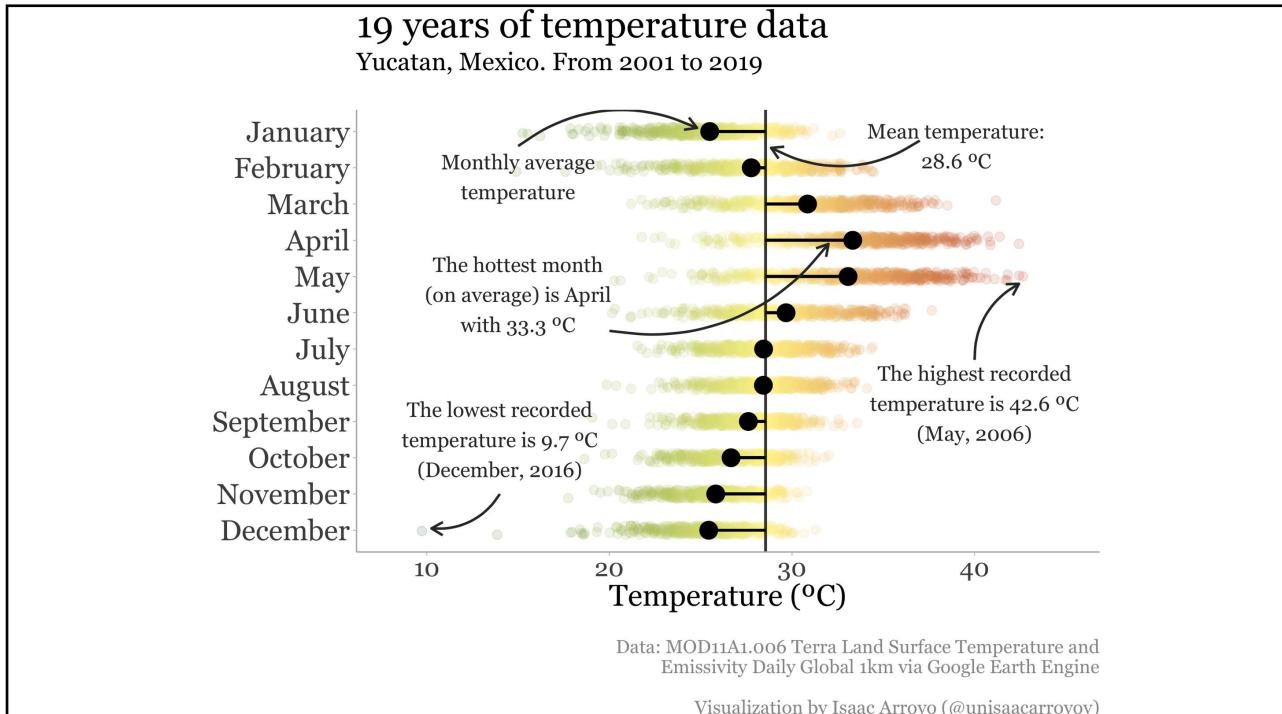
10



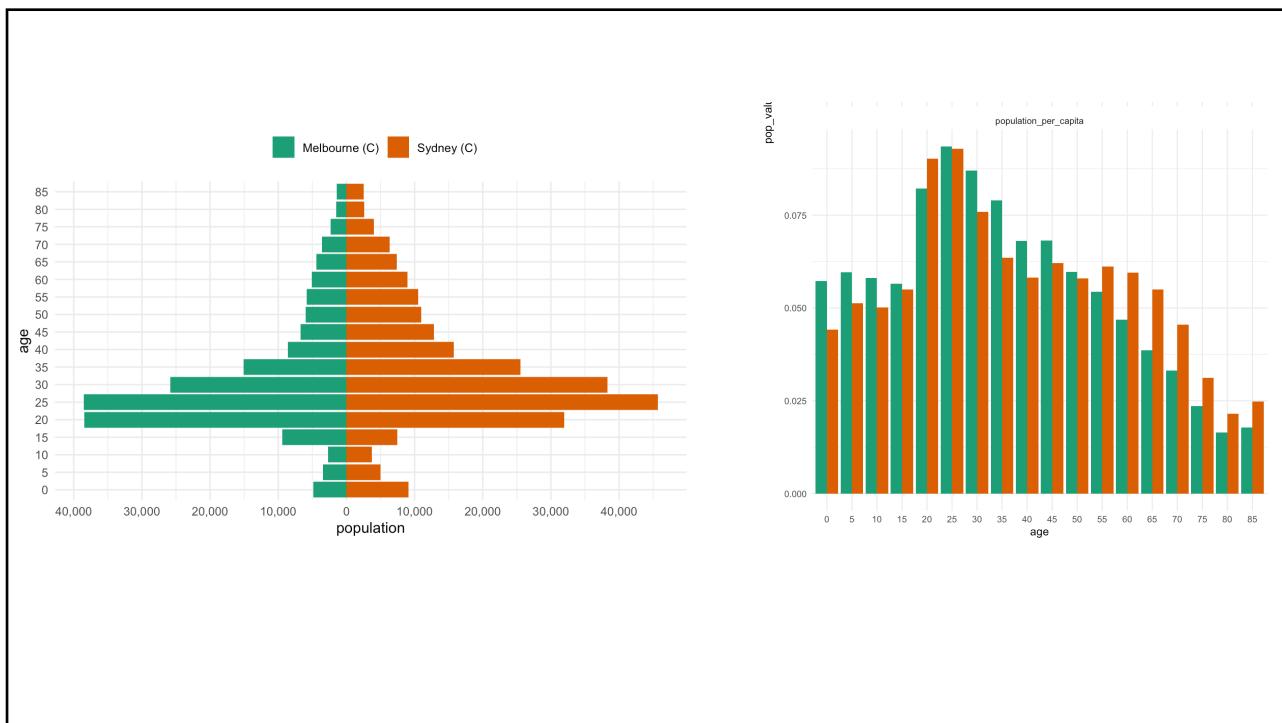
11



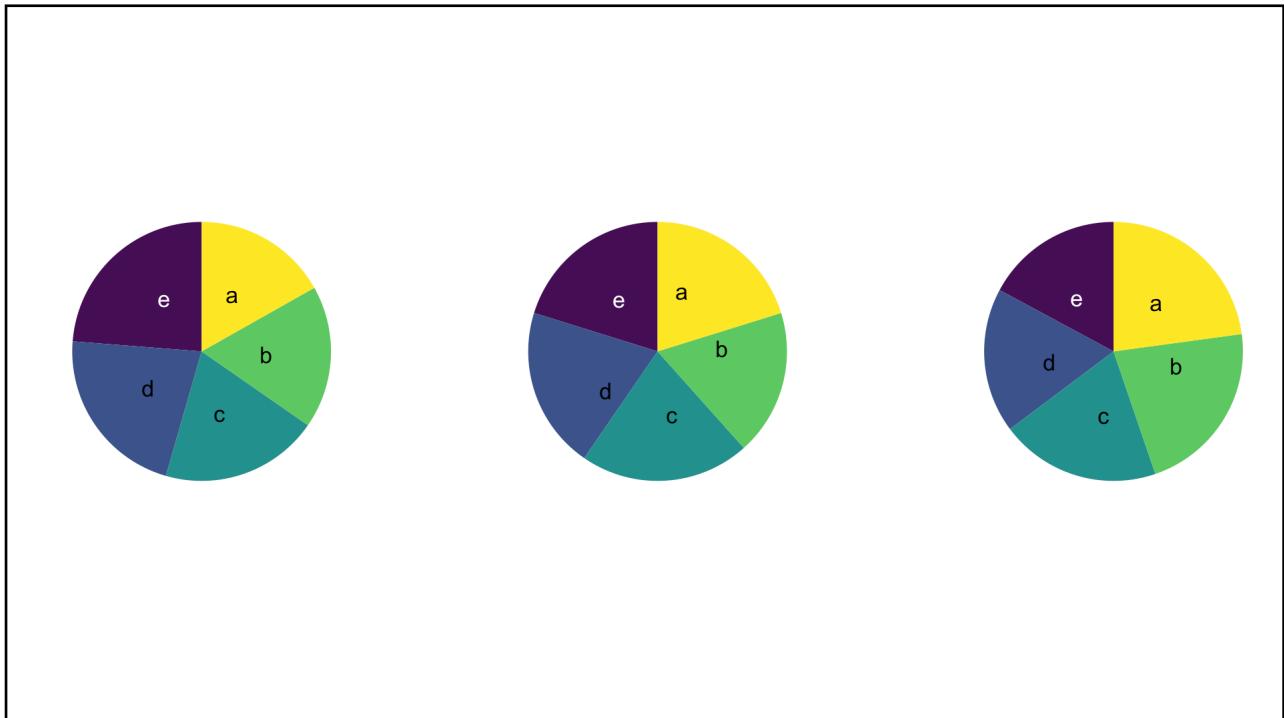
12



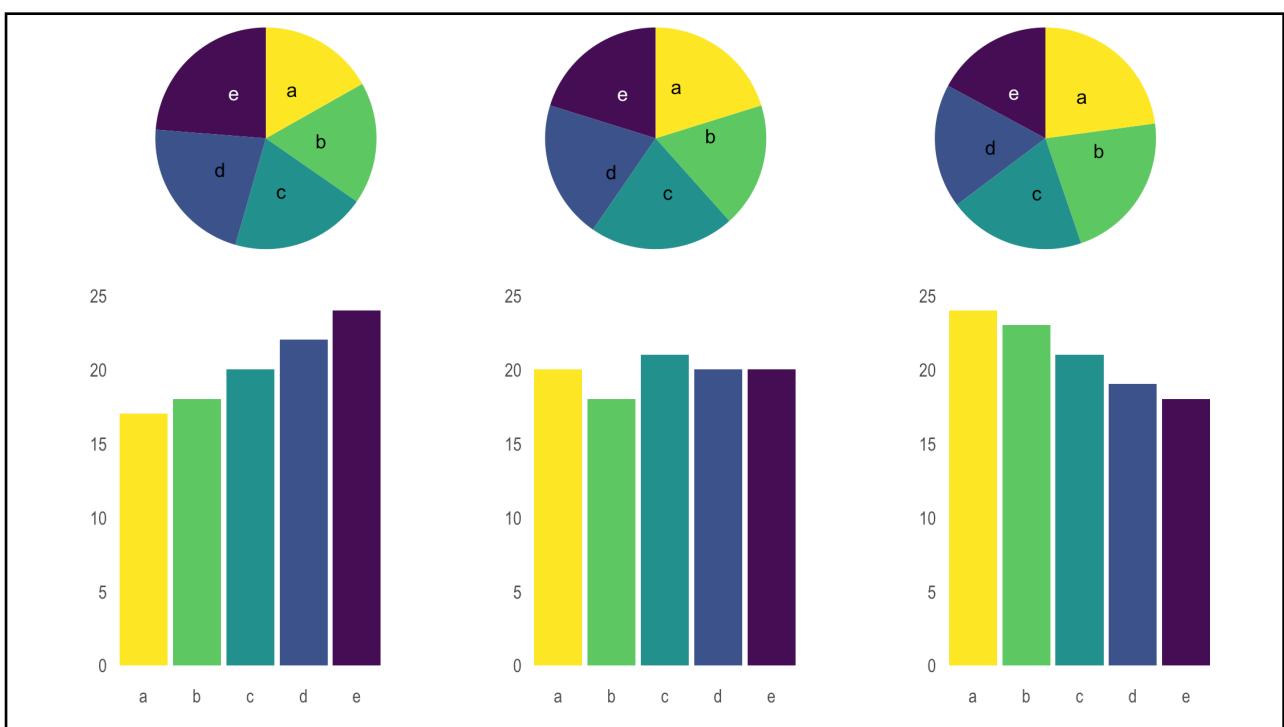
13



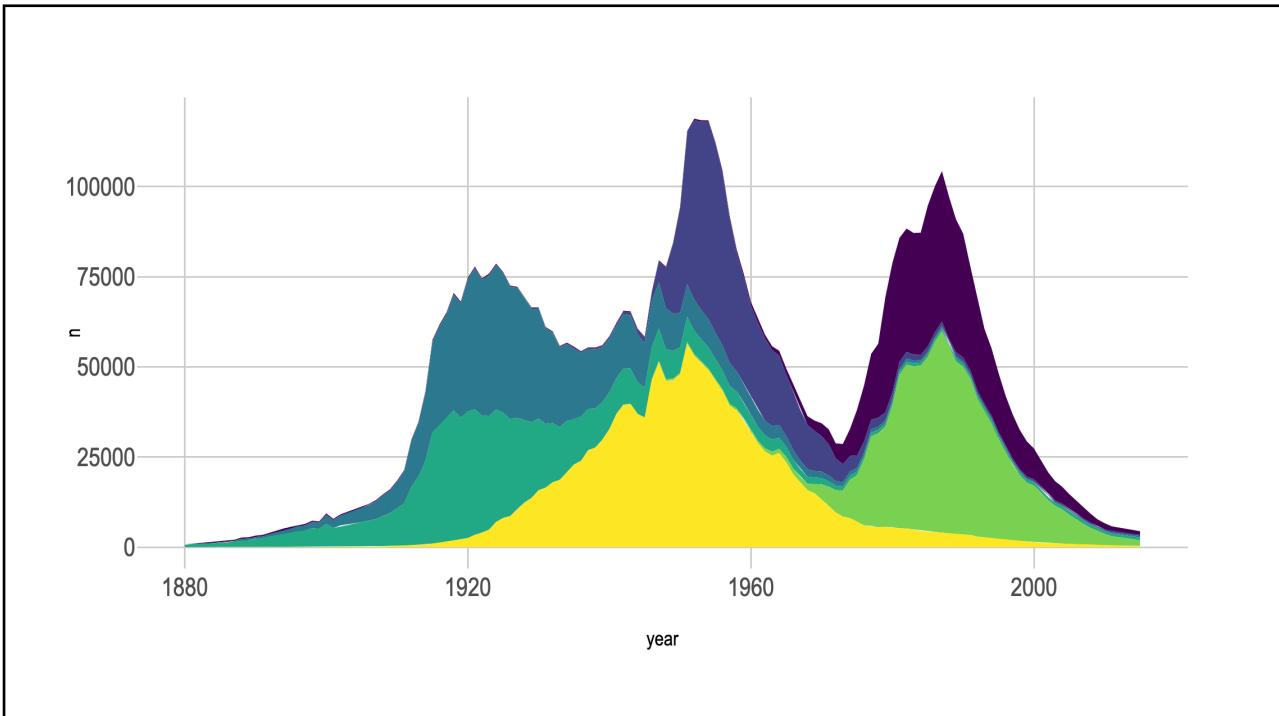
14



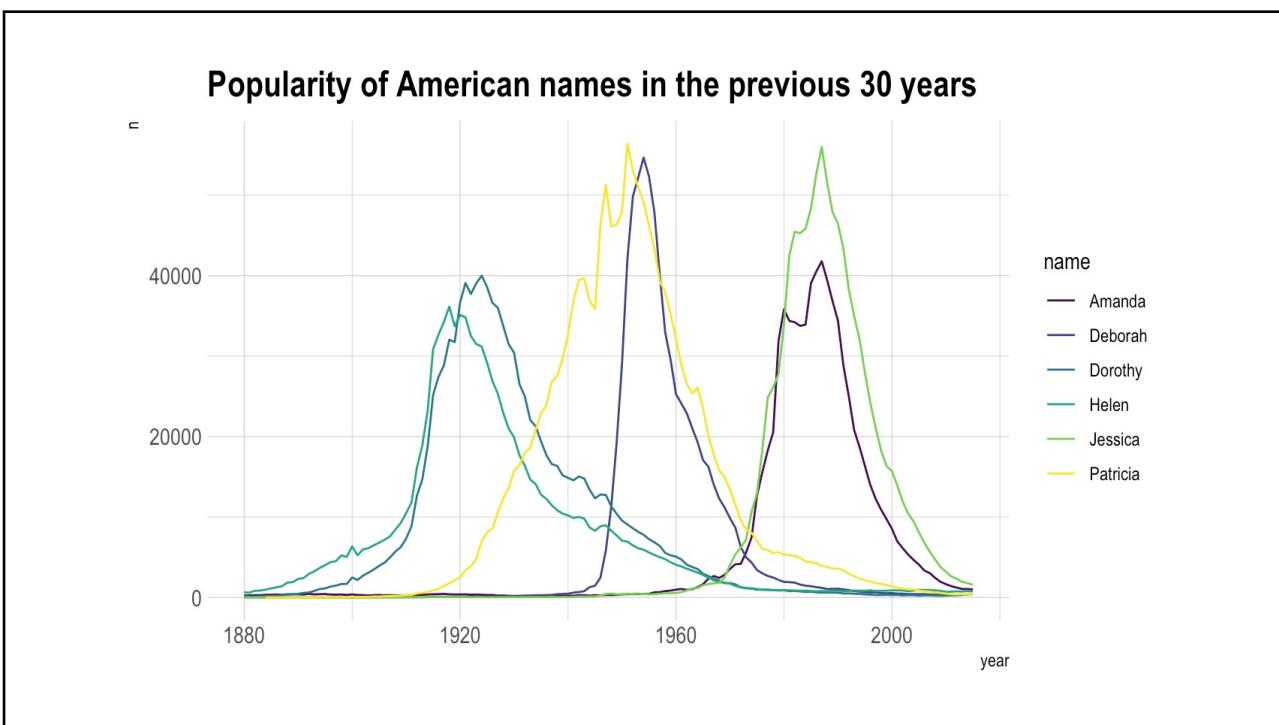
15



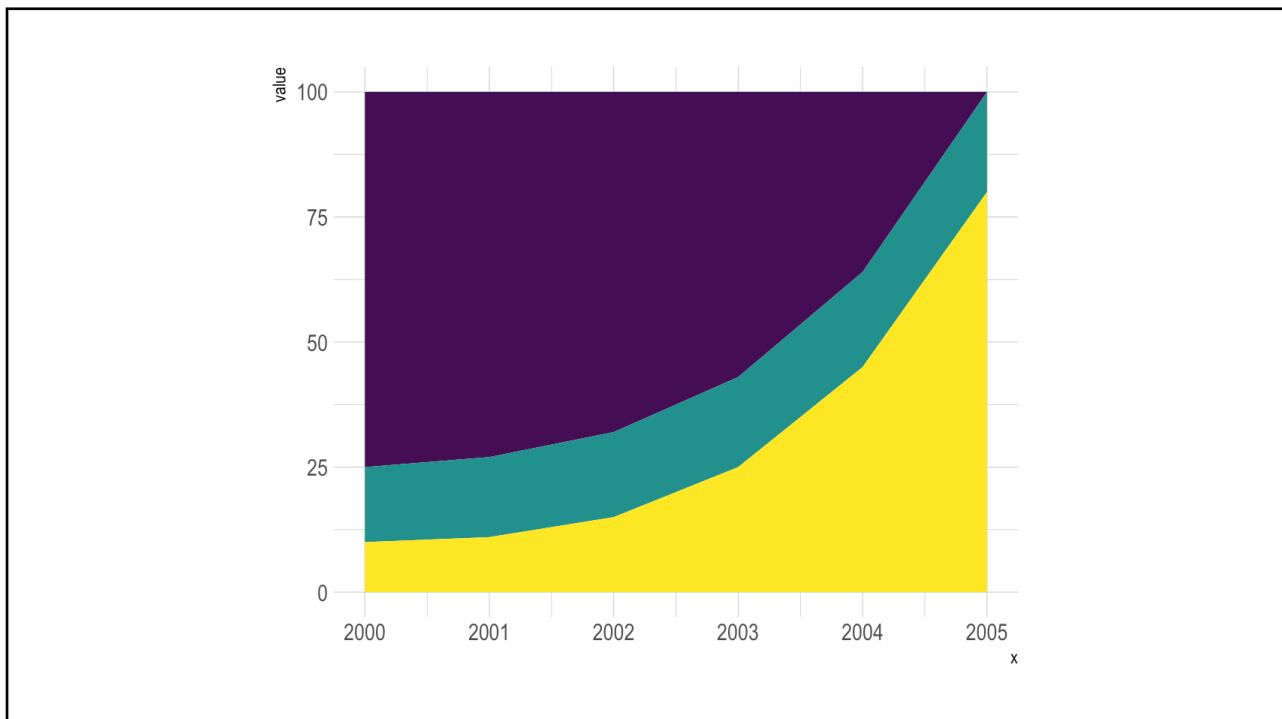
16



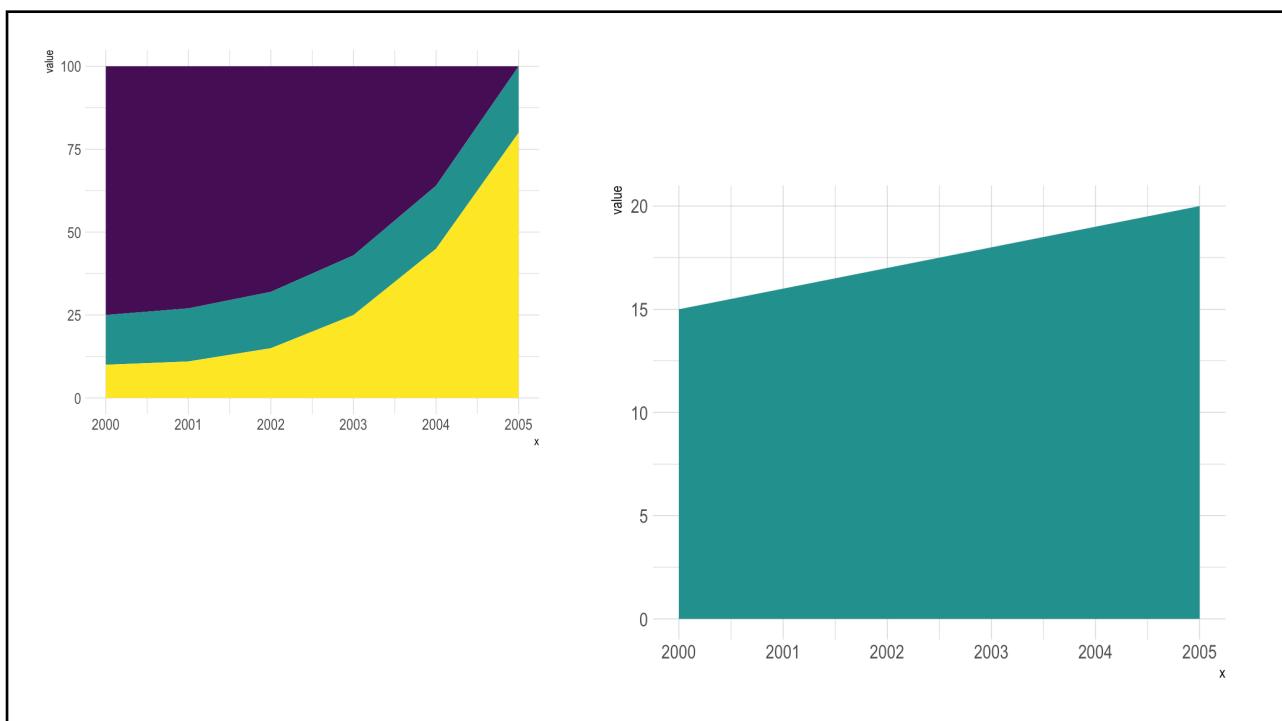
17



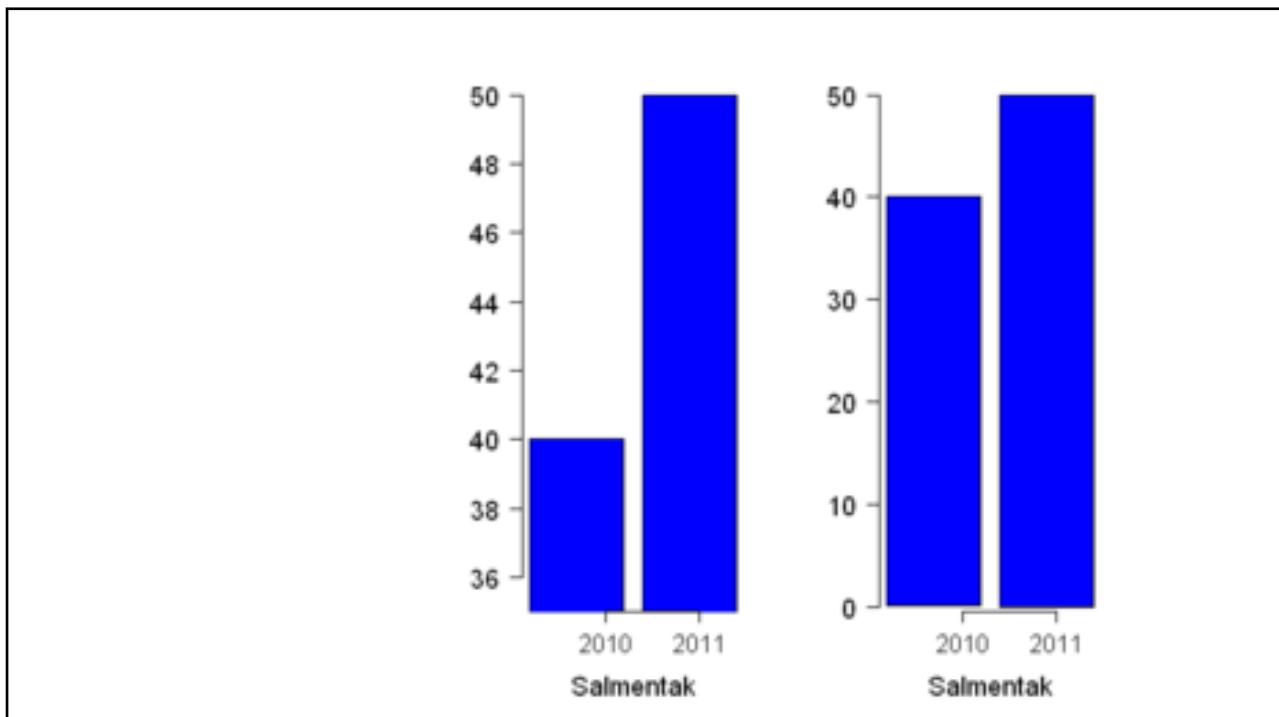
18



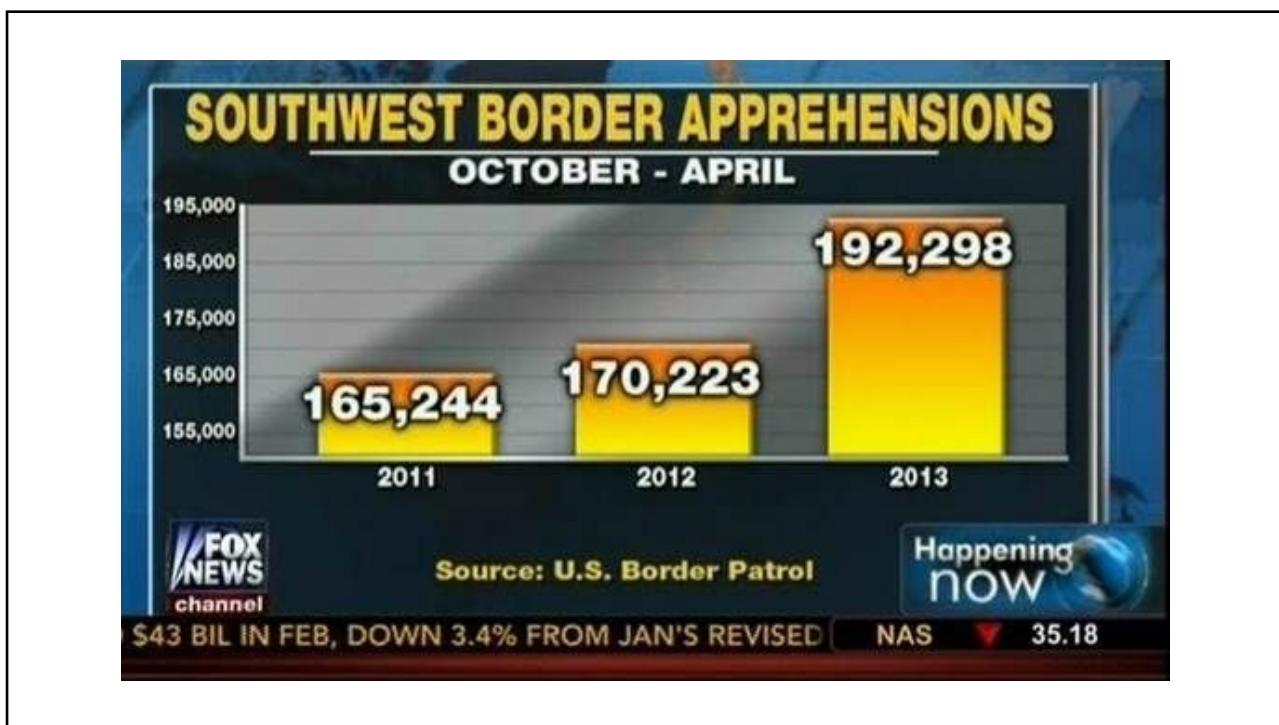
19



20



21



22

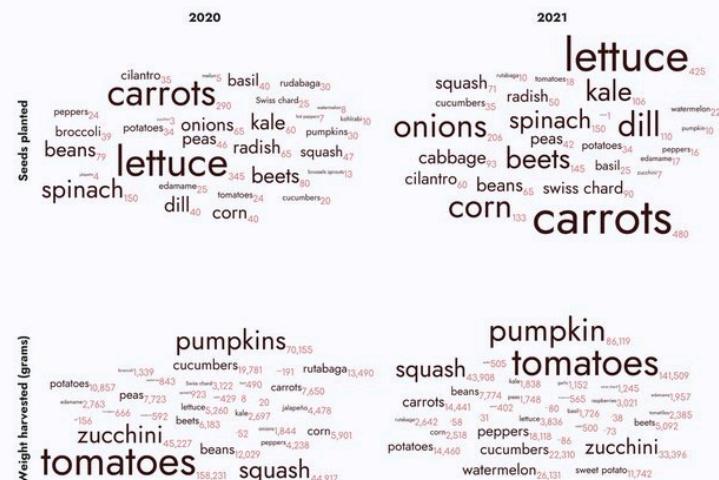
Take home 1:

There are many factually correct ways to visualise your data, but few truly effective ones.

23

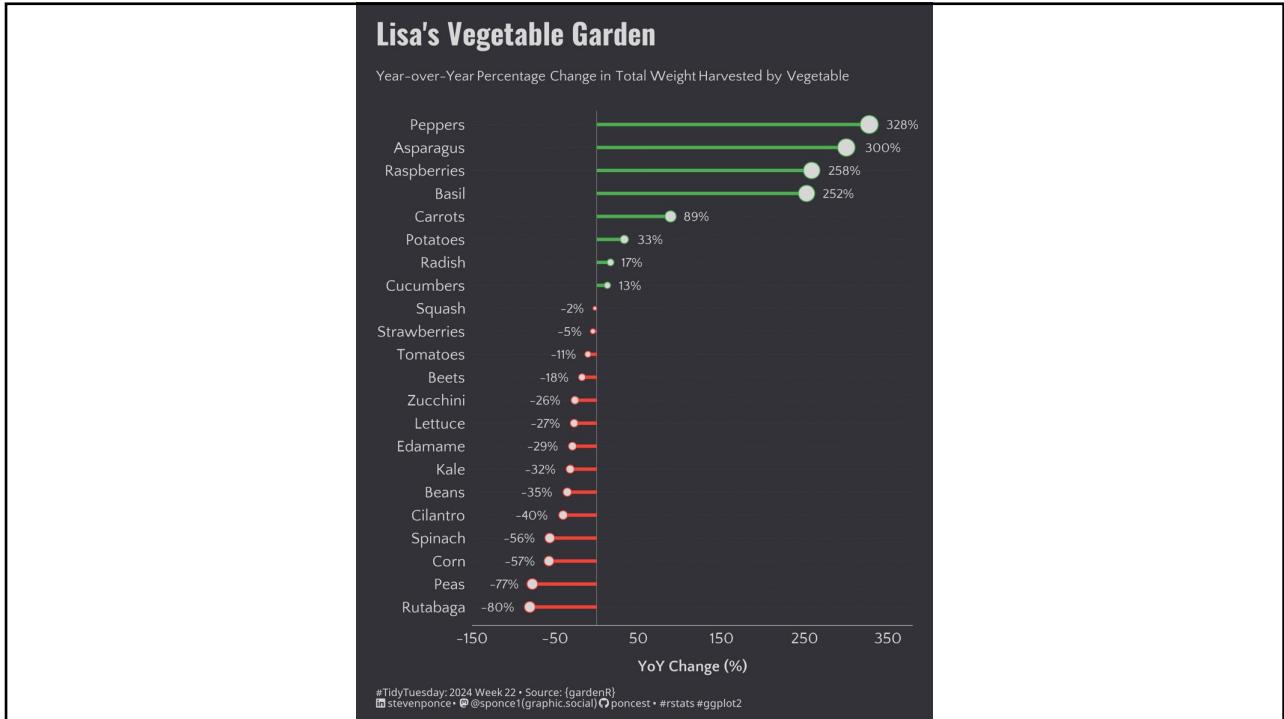
Lisa's Vegetable Garden Data

Number of Vegetable Seeds Planted and Harvested Weight (in grams) from Lisa's Garden.

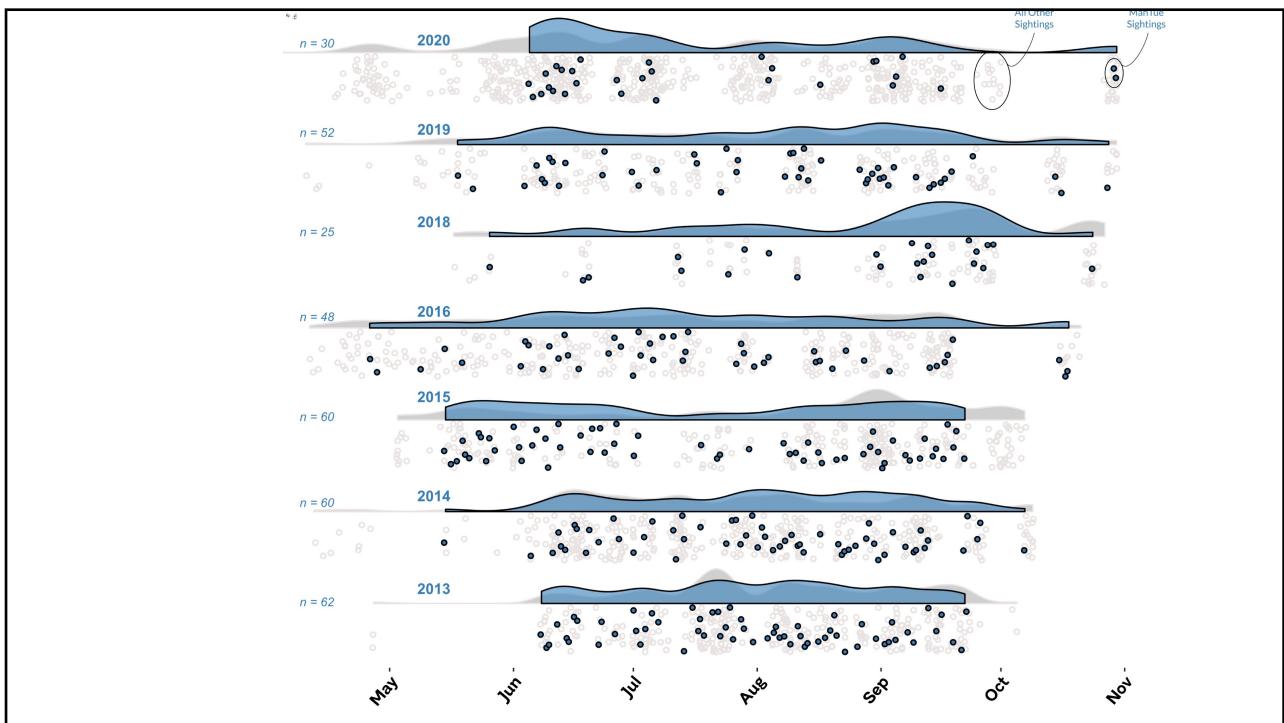


Source: [\[gardenR\] package](#) | Graphic: Nikolaos Pechlivanis

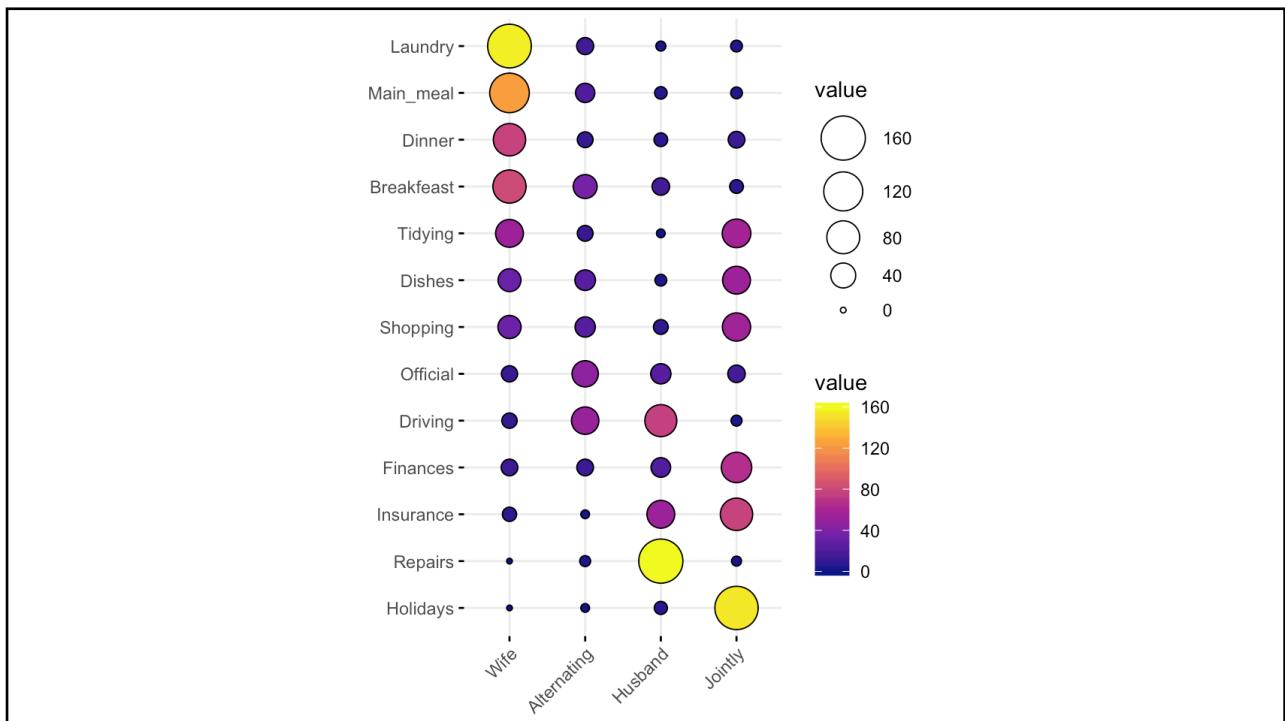
24



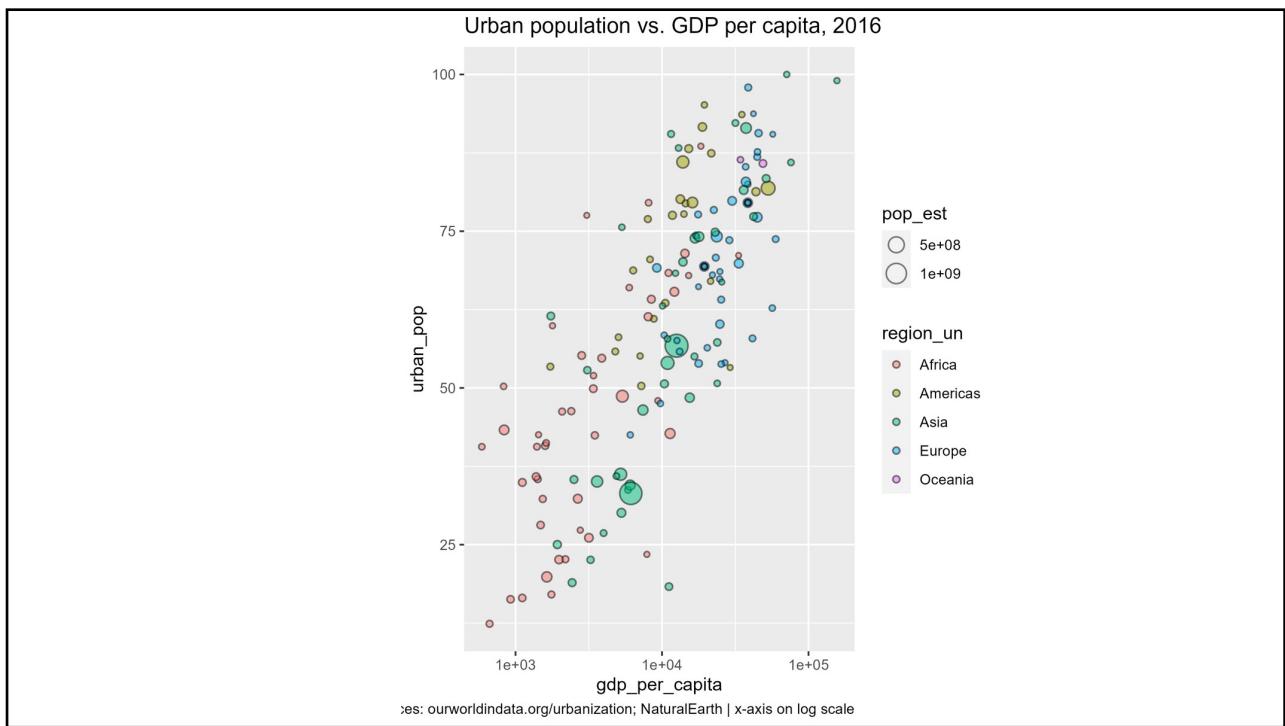
25



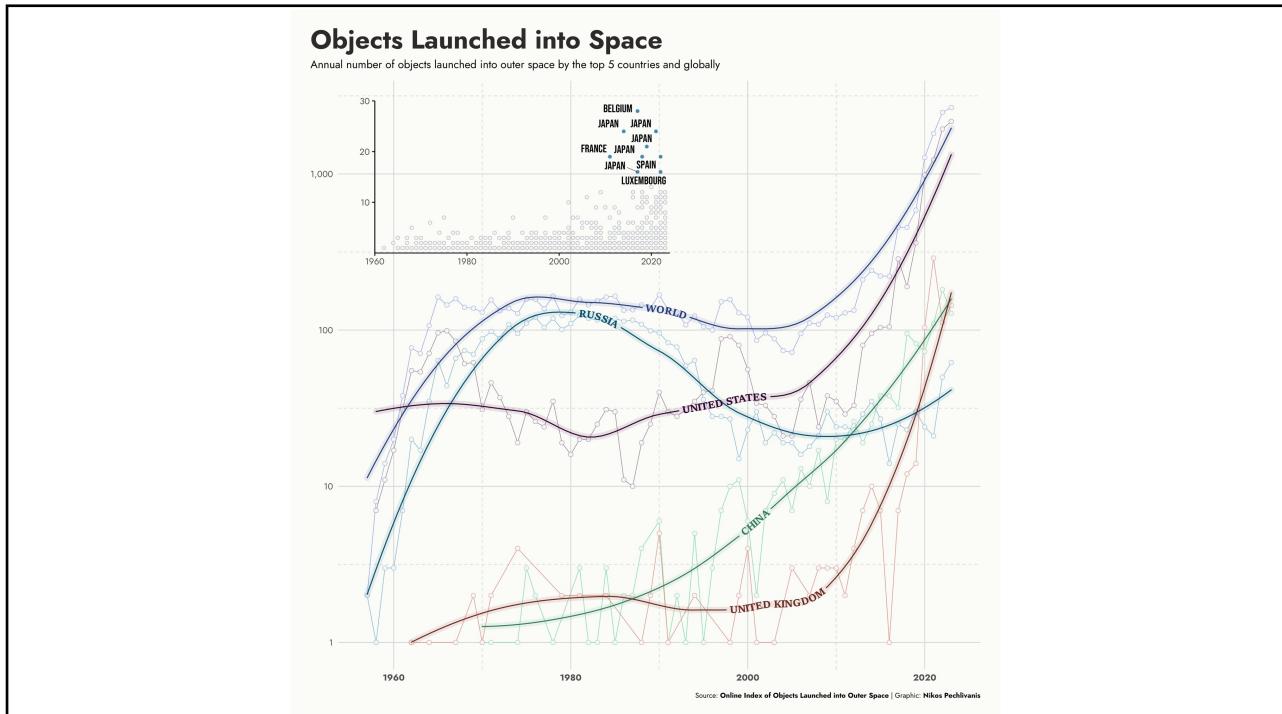
26



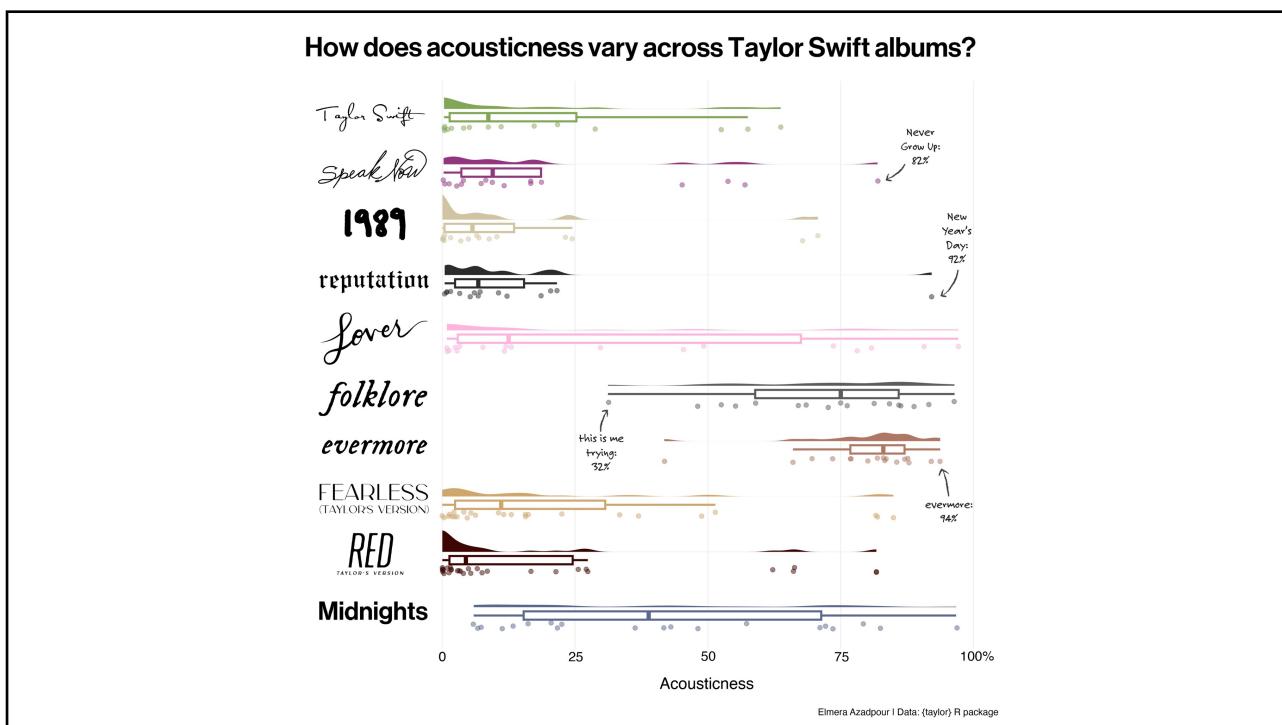
27



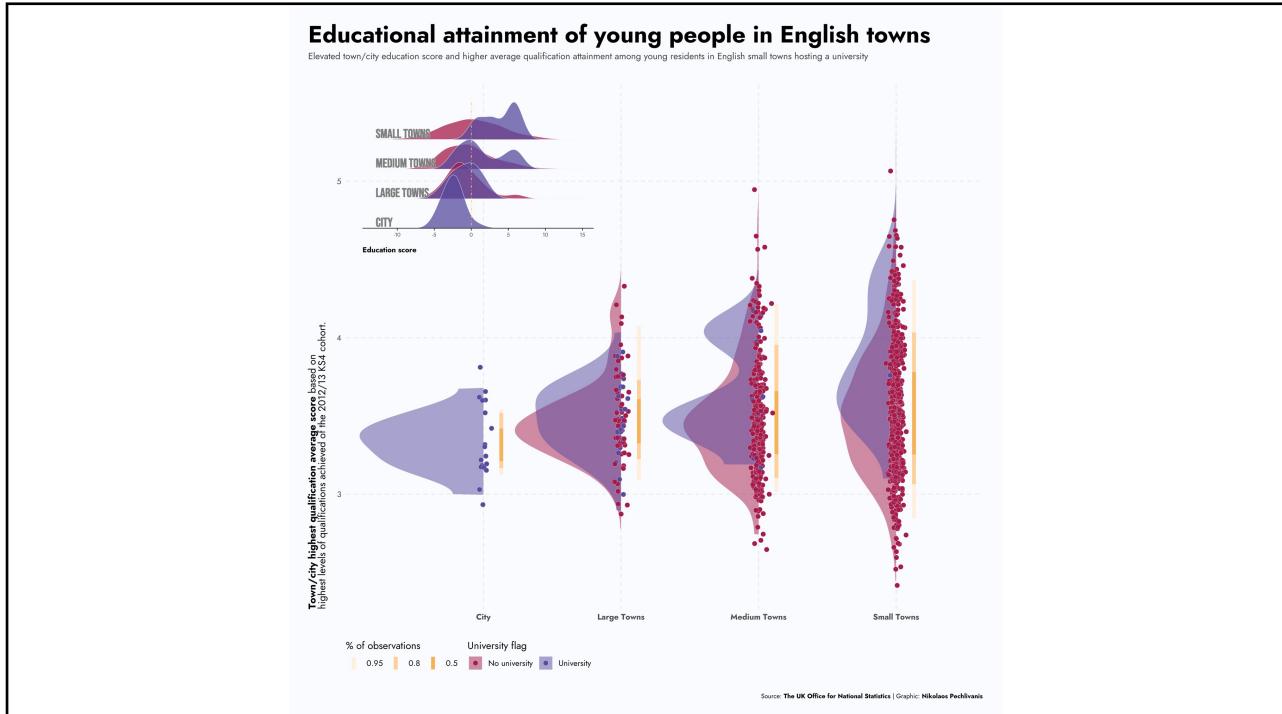
28



29



30



31

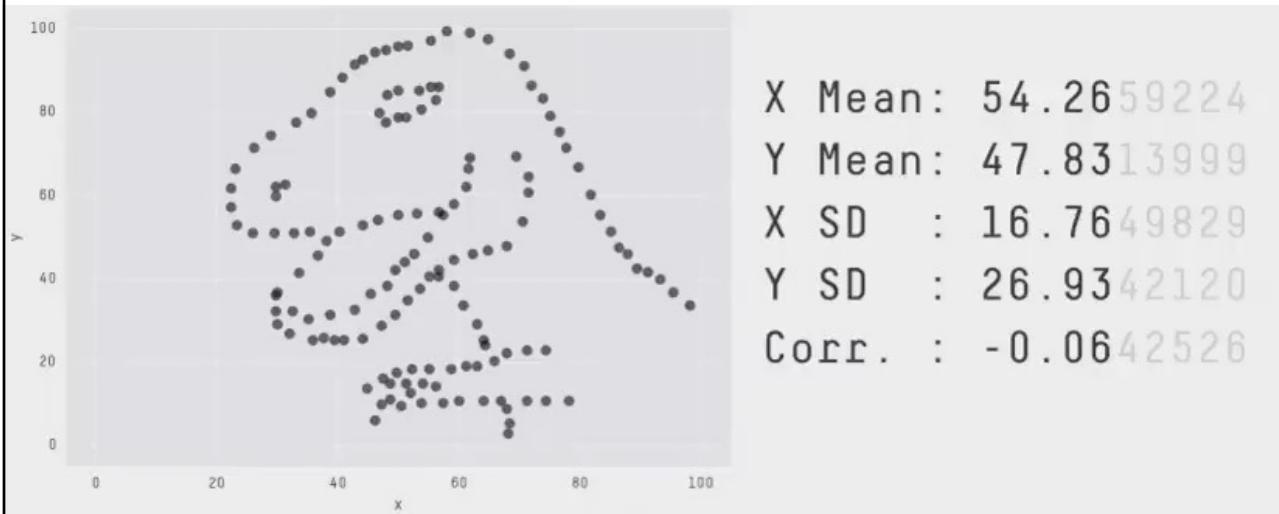
Take home 2:

**A good figure reveals patterns in data
that you as a researcher want to share.**

**A reader shouldn't need to look for that pattern.
Ideally, it jumps off the screen.**

32

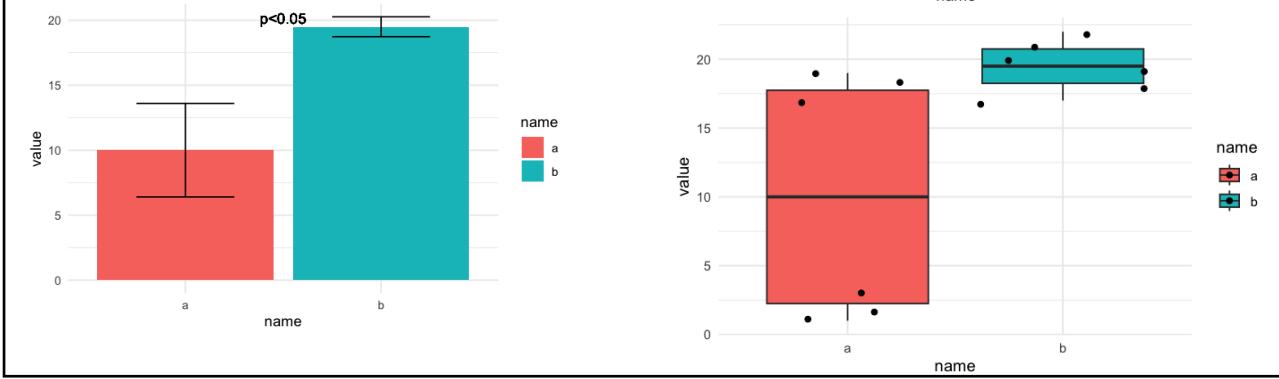
Explore Plot your data



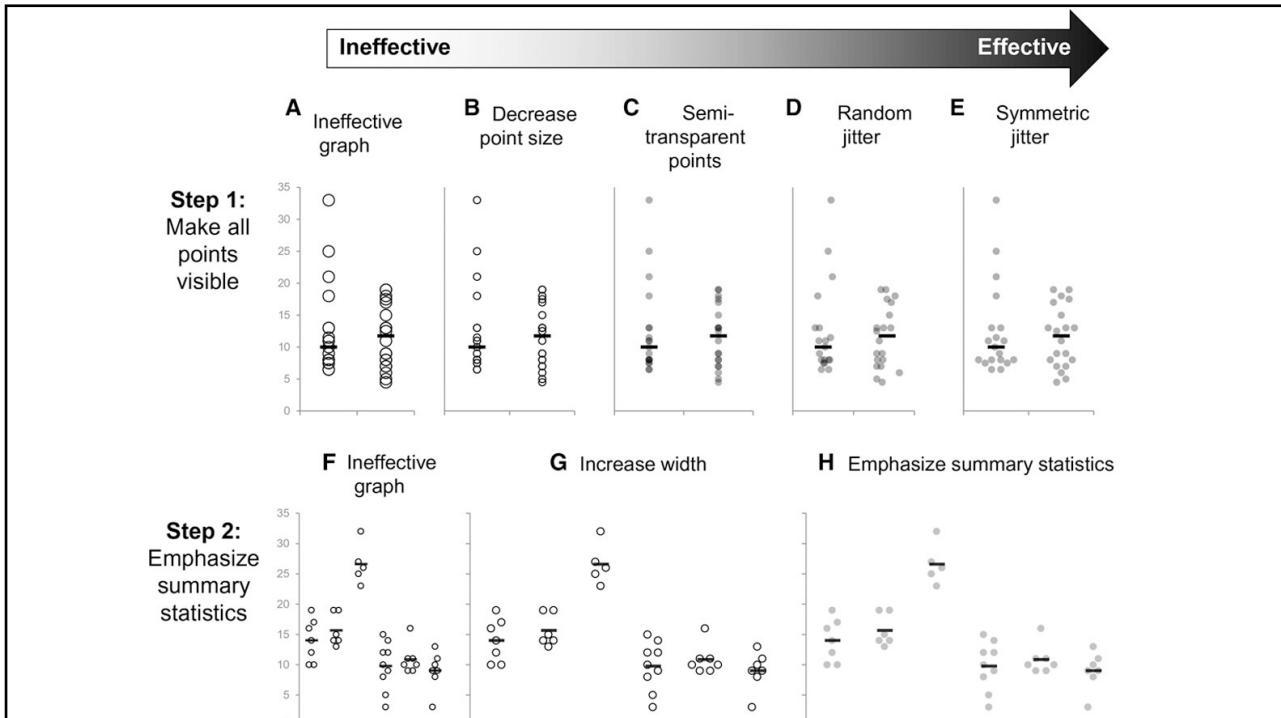
33

How to fraud with barcharts

Would you believe this barchart in a paper?



34



35

VISUALIZING DATA

Error bars hide:

- Outliers
- Distribution shape
- Bi- or multi-modal distributions
- Asymmetric distributions

<https://twitter.com/sketchscience>

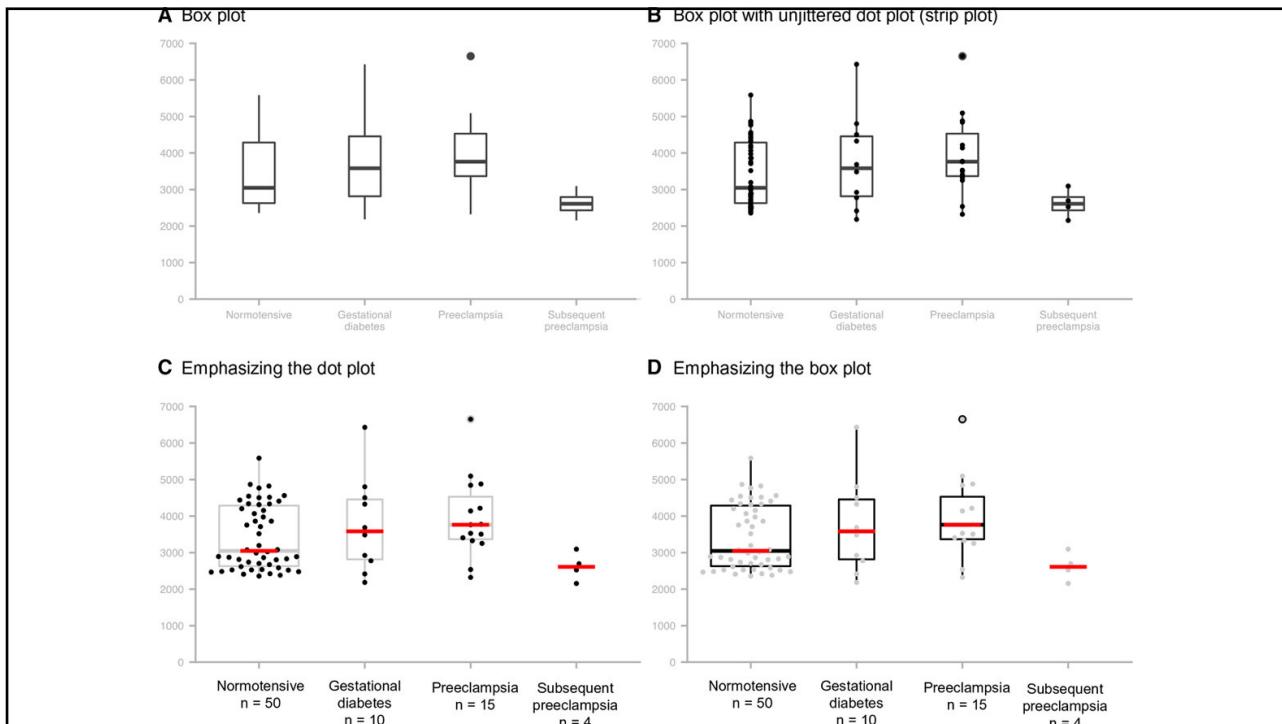
Control Experiment 1 Experiment 2

Treatment	Mean Response	Min	Q1	Median	Q3	Max
Control	5.0	4.0	4.5	4.8	5.2	6.5
Experiment 1	4.8	3.5	4.0	4.5	5.0	6.0
Experiment 2	6.5	5.5	6.0	6.2	6.8	8.0

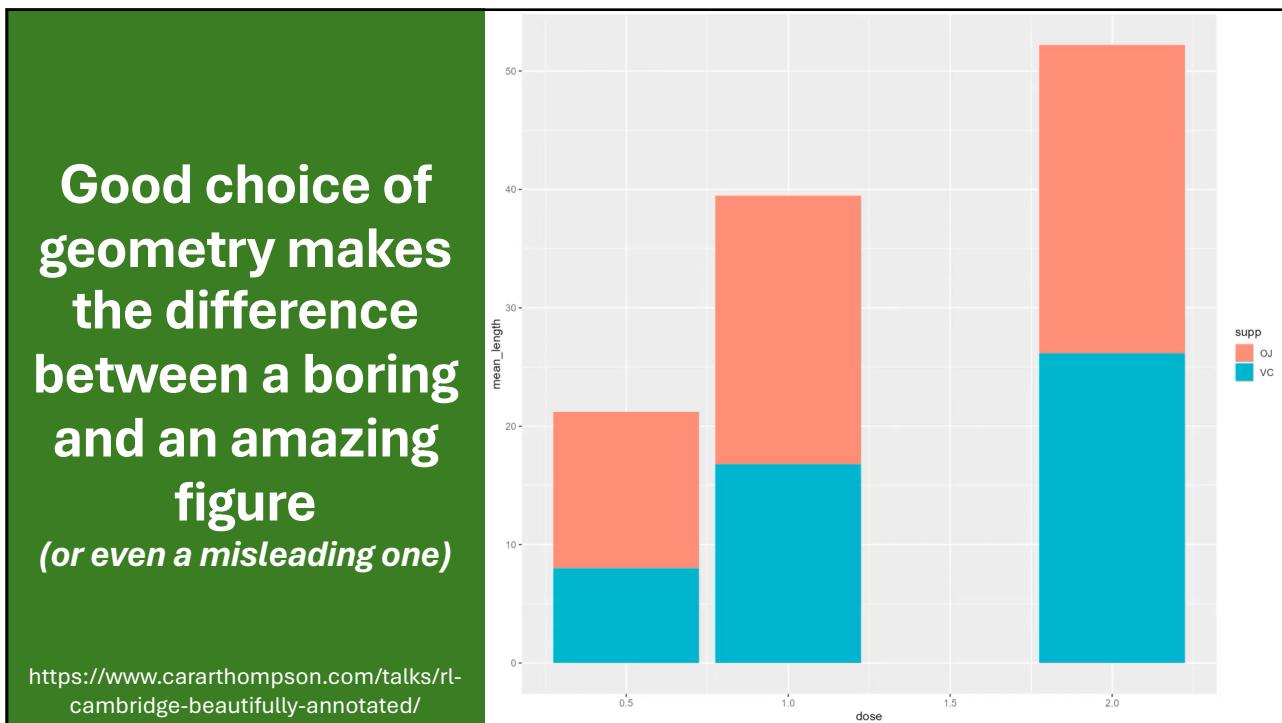
Control Experiment 1 Experiment 2

Sketching Science

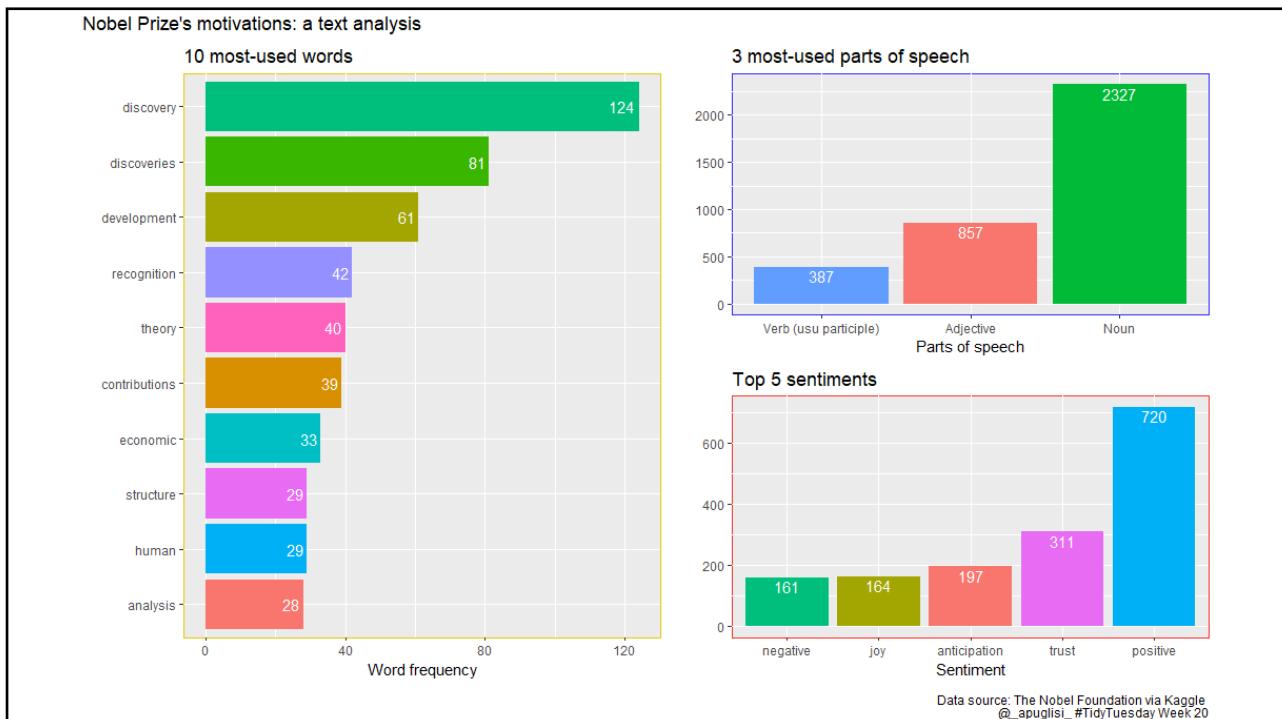
36



37



38



39

Take home 3:

Find a visualisation that shows:

Firstly, your main pattern effectively

Secondly, as much of the raw data as the image can handle

40

GGplot in a nutshell

Guides & Legends

Themes

Facets

Coordinate system & zoom

Statistics

Geometries

Scales

Mapping / aesthetics

Data



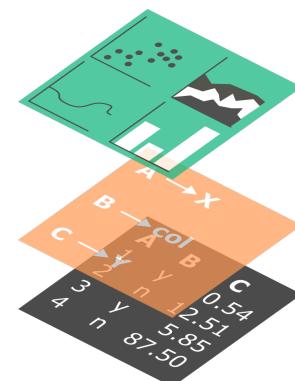
41

GGplot in a nutshell

Geometries

Mapping / aesthetics

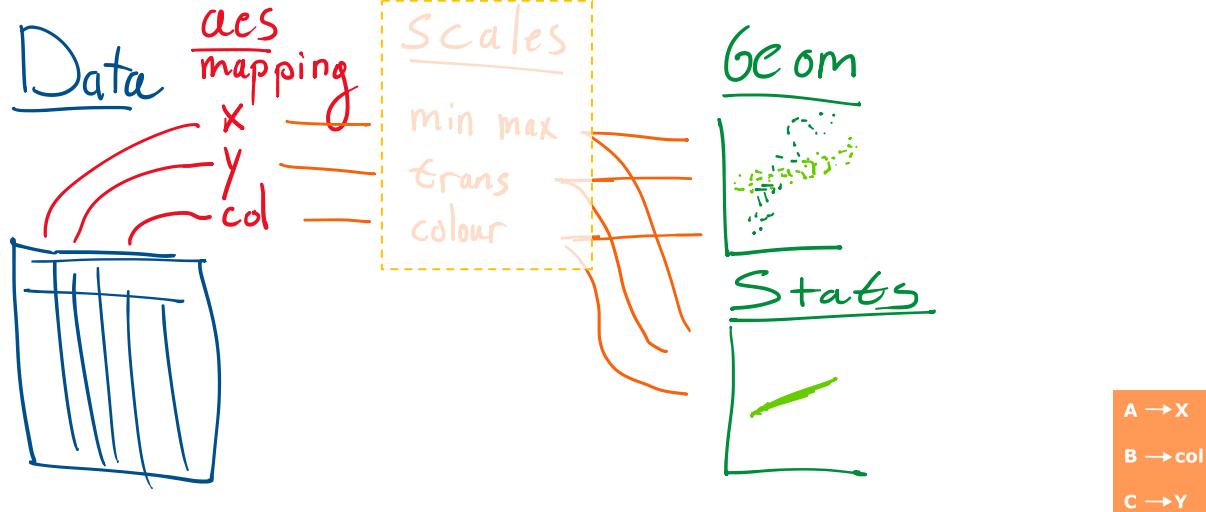
data



42

21

Aesthetic mapping links your data to your plot



43

Use the cheatsheet to keep an overview

Data visualization with ggplot2 :: CHEAT SHEET



Basics

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same components: a data set, a coordinate system, and geoms—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like size, color, and x and y locations.



Complete the template below to build a graph.

ggplot (data = **DATA**, aes = **MAPPING**, stat = **STATS**, position = **POSITIONS**) +

+**COORDINATE FUNCTIONS**

+**SCALE FUNCTIONS**

+**THEME FUNCTIONS**

ggplot (data = **DATA**, aes = **MAPPING**, stat = **STATS**, position = **POSITIONS**) +

geom_point(mapping = aes(x = **x**, y = **y**)) begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggplot("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values:
color and fill—string ("red", "#RRGGBB")
linetype—integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotteddash", 5 = "longdash", 6 = "mediumdash", 7 = "longdashdot", 8 = "dashdot", 9 = "dashdotdot", 10 = "lineend" string ("round", "butt", or "square")
linejoin—string ("round", "mitre", or "bevel")
size—integer (line width in mm)
shape—integer/shape name or a single character ("*")



Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))

b <- ggplot(economics, aes(x = long, y = a))

Ensure limits interact well across all plots.

b + geom_curve(aes(xend = lat + 1, yend = long + 1), curve_type = "bezier", ncp = 1)

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1), fill = "white", alpha = 0.5, color = "black", linetype = "solid", size = 1)

b + geom_polygon(aes(alpha = 0.5) -> x, alpha, color, fill, group, subgroup, linetype, size)

b + geom_rect(aes(ymin = unemploy - 900, ymax = unemploy + 900) -> x, alpha, ymin, alpha, color, fill, group, linetype, size)

b + geom_text(aes(label = "b"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "b")

x, y, alpha, color, line, size

b + geom_text(aes(label = "c"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "c")

x, y, alpha, color, line, size

b + geom_text(aes(label = "d"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "d")

x, y, alpha, color, line, size

b + geom_text(aes(label = "e"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "e")

x, y, alpha, color, line, size

b + geom_text(aes(label = "f"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "f")

x, y, alpha, color, line, size

b + geom_text(aes(label = "g"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "g")

x, y, alpha, color, line, size

b + geom_text(aes(label = "h"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "h")

x, y, alpha, color, line, size

b + geom_text(aes(label = "i"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "i")

x, y, alpha, color, line, size

b + geom_text(aes(label = "j"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "j")

x, y, alpha, color, line, size

b + geom_text(aes(label = "k"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "k")

x, y, alpha, color, line, size

b + geom_text(aes(label = "l"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "l")

x, y, alpha, color, line, size

b + geom_text(aes(label = "m"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "m")

x, y, alpha, color, line, size

b + geom_text(aes(label = "n"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "n")

x, y, alpha, color, line, size

b + geom_text(aes(label = "o"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "o")

x, y, alpha, color, line, size

b + geom_text(aes(label = "p"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "p")

x, y, alpha, color, line, size

b + geom_text(aes(label = "q"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "q")

x, y, alpha, color, line, size

b + geom_text(aes(label = "r"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "r")

x, y, alpha, color, line, size

b + geom_text(aes(label = "s"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "s")

x, y, alpha, color, line, size

b + geom_text(aes(label = "t"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "t")

x, y, alpha, color, line, size

b + geom_text(aes(label = "u"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "u")

x, y, alpha, color, line, size

b + geom_text(aes(label = "v"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "v")

x, y, alpha, color, line, size

b + geom_text(aes(label = "w"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "w")

x, y, alpha, color, line, size

b + geom_text(aes(label = "x"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "x")

x, y, alpha, color, line, size

b + geom_text(aes(label = "y"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "y")

x, y, alpha, color, line, size

b + geom_text(aes(label = "z"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "z")

x, y, alpha, color, line, size

b + geom_text(aes(label = "A"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "A")

x, y, alpha, color, line, size

b + geom_text(aes(label = "B"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "B")

x, y, alpha, color, line, size

b + geom_text(aes(label = "C"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "C")

x, y, alpha, color, line, size

b + geom_text(aes(label = "D"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

b + geom_rug(size = "D")

x, y, alpha, color, line, size

b + geom_text(aes(label = "E"))

nudge_x = 1, nudge_y = 1, label, alpha, angle, color, family, fontface, fontstyle, fontweight, size, vjust

b + geom_smooth(method = lm)

x, y, alpha, color, fill, group, method, size, weight

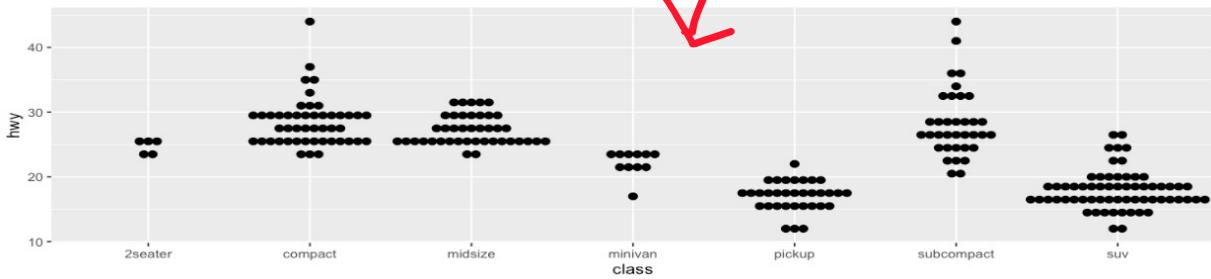
b + geom_rug(size = "E")

x, y, alpha, color, line, size

Cheat-sheet has example-code to get started

```
```{r}
f <- ggplot(mpg, aes(class, hwy))

f + geom_dotplot(binaxis = 'y', stackdir = 'center')
```
```



one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill, group

f + geom_violin(scale = "area")
x, y, alpha, color, fill, group, linetype, size, weight

45

To R studio!

Get your exercise file at
github.com/lauralwd/beyond_barcharts_workshop

46

When making a figure

Ask yourself, and your colleagues, often: **What do I want my reader to take from this figure**

And can you plot that in a more direct way?

47

What's not in this workshop

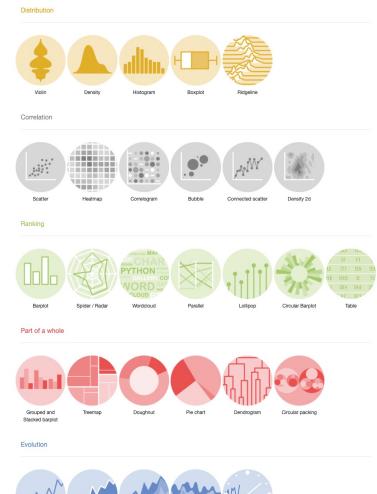
- clear labels and legends
 - colour usage
 - scales
 - themes
 - exporting specifically for publication
- and much more

--> Next ggplot course for beginners and novices in November!

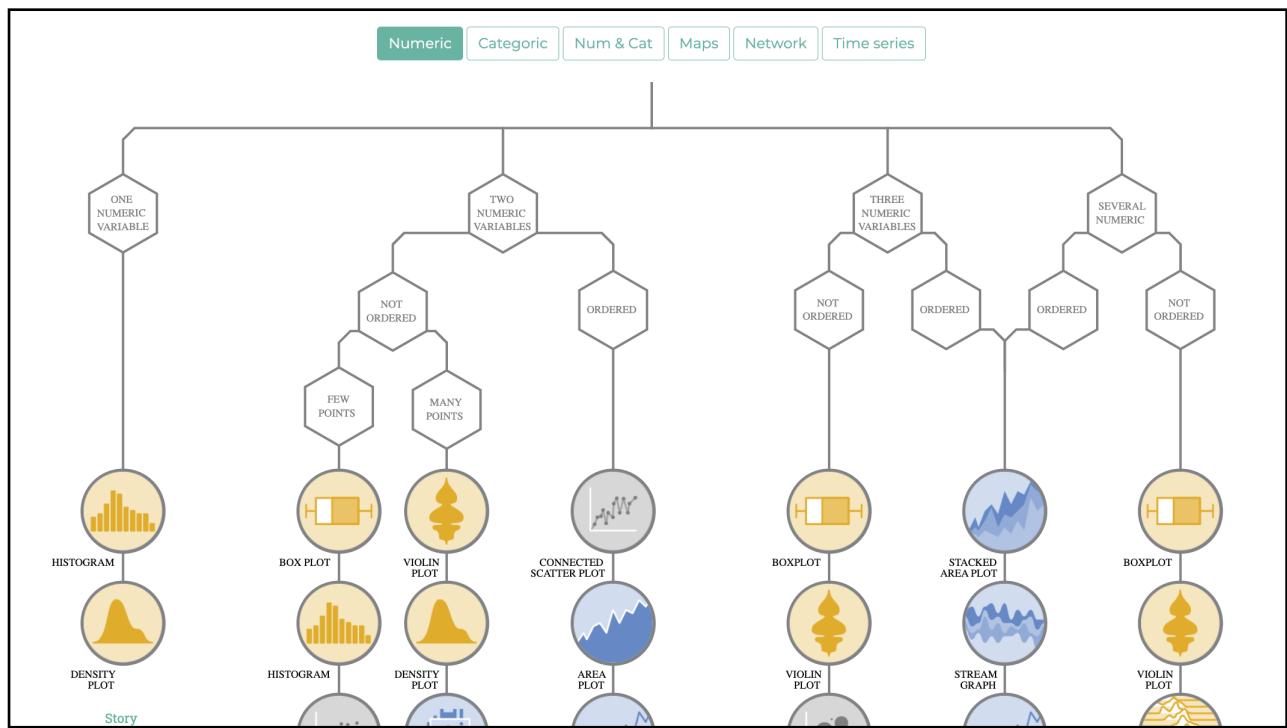
48

Choosing a geometry or statistic

- <https://r-graph-gallery.com/>
- <https://ggplot2.tidyverse.org/>
- <https://www.data-to-viz.com>
- <https://datavizcatalogue.com>



49



50

BOXPLOT

Summarize the distribution of numeric variables

About

A `boxplot` gives a nice summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

Common Mistakes

- Boxplot hides the sample size of each group, [show it with annotation or box width](#).
- Boxplot [hides the underlying distribution](#). Use jitter if low number of data points, or use violin with bigger data.
- [Order your boxplot by median](#) can make it more insightful.

Code

R graph gallery Python D3.js React

NOT ORDERED ORDERED ORDERED NOT ORDERED

BOXPLOT STACKED AREA PLOT BOXPLOT VIOLIN PLOT

VIOLIN PLOT STREAM GRAPH BUBBLE PLOT RIDGE LINE

BUBBLE PLOT LINE PLOT 3D SCATTER OR SURFACE AREA (SM)

3D SCATTER OR SURFACE Story Story PCA

51

And then, something completely different: Get GitHub co-pilot in your R-studio for free

1. Apply for a github account
 - [Github.com](https://github.com)
2. Apply for the education discount as a teacher at an educational institute
 - <https://docs.github.com/en/education/explore-the-benefits-of-teaching-and-learning-with-github-education/github-education-for-teachers/apply-to-github-education-as-a-teacher>
3. Connect your Rstudio to Github
 - <https://docs.posit.co/ide/user/ide/guide/tools/copilot.html>

To qualify for teacher benefits, you must:

- Have a GitHub account.
- Be an educator, faculty member, or researcher at a recognized educational institution.
- Be able to provide documentation from your school which demonstrates your current employment status.

52