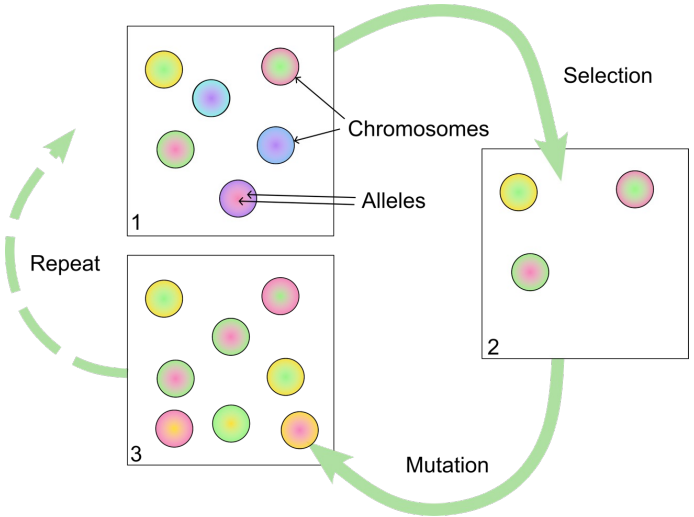
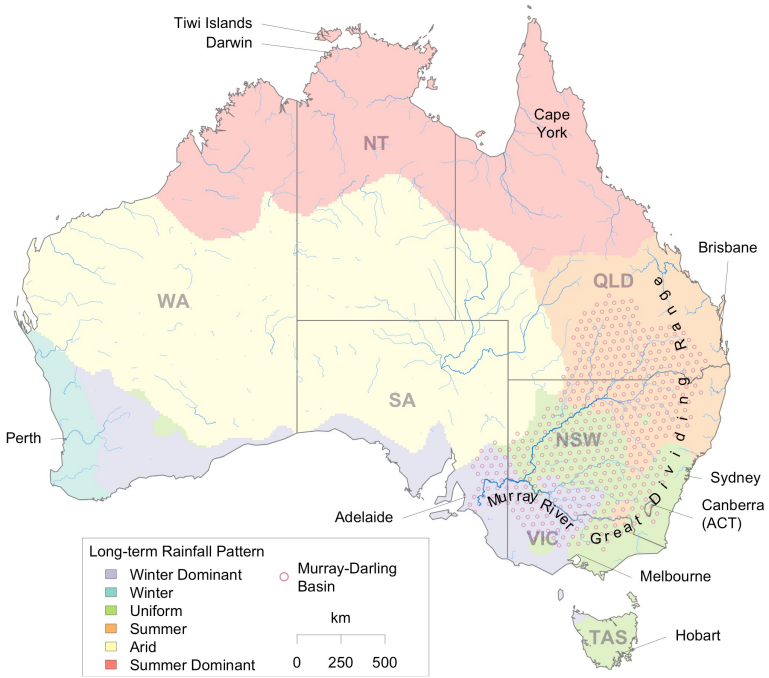
Visualising data
& other horror stories

with Lucy

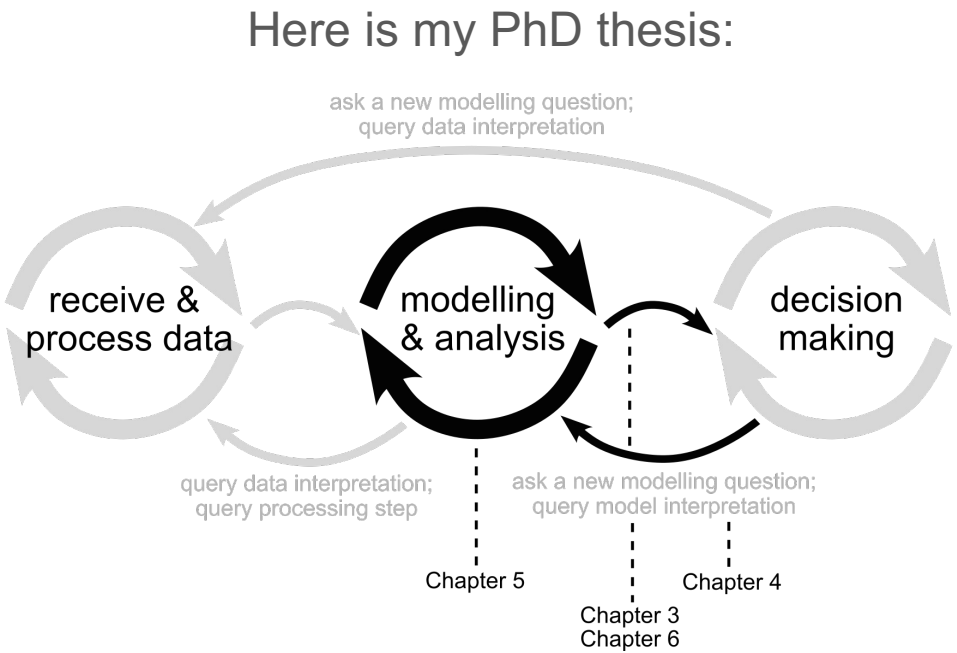
Preface

Visualisation is my favourite part of my job





Preface


Here is my PhD thesis:
ask a new modelling question;
query data interpretation
receive &
process data
modelling
& analysis
decision
making
query data interpretation;
query processing step
ask a new modelling question;
query model interpretation
Chapter 5
Chapter 3
Chapter 6
Chapter 4

This afternoon, we will:

1. Talk through some tips/tricks/rules of thumb for visualisation
2. Workshop the visualisations we're working on together

    Pick a visualisation from your manuscript, and we will peer review

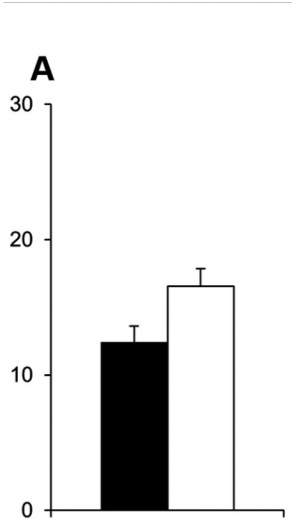
Resources

- Weissgerber et al. (2015). Beyond bar and line graphs: time for a new data presentation paradigm. *PLoS biology*, doi: 10.1371/journal. Pbio.1002128
- Weissgerber et al. (2019). Reveal, don't conceal: transforming data visualization to improve transparency. *Circulation*, doi: 10.1161/CIRCULATIONAHA.118.037777
- Midway (2020). Principles of effective data visualization. *Patterns*, doi: 10.1016/j.patter.2020.100141
- Hehman & Xie (2021). Doing better data visualization. *Advances in Methods and Practices in Psychological Science*, doi: 10.1177/25152459211045334.

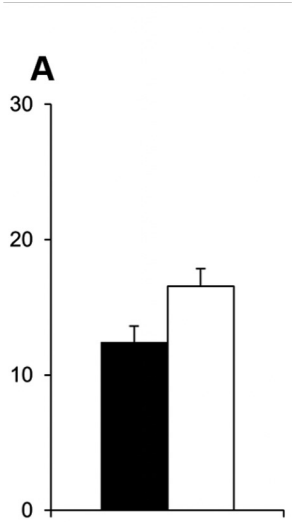
Some principles for visualising data …

1. Show the data
2. But don’t show too much
3. Uncertainty is important
4. And so is proportion
5. And colour
6. And finally, if in doubt … get another opinion!

Weissgerber et al., 2015

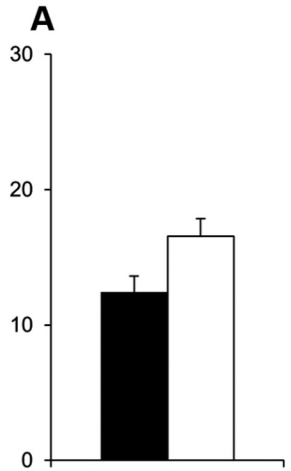


Weissgerber et al., 2015



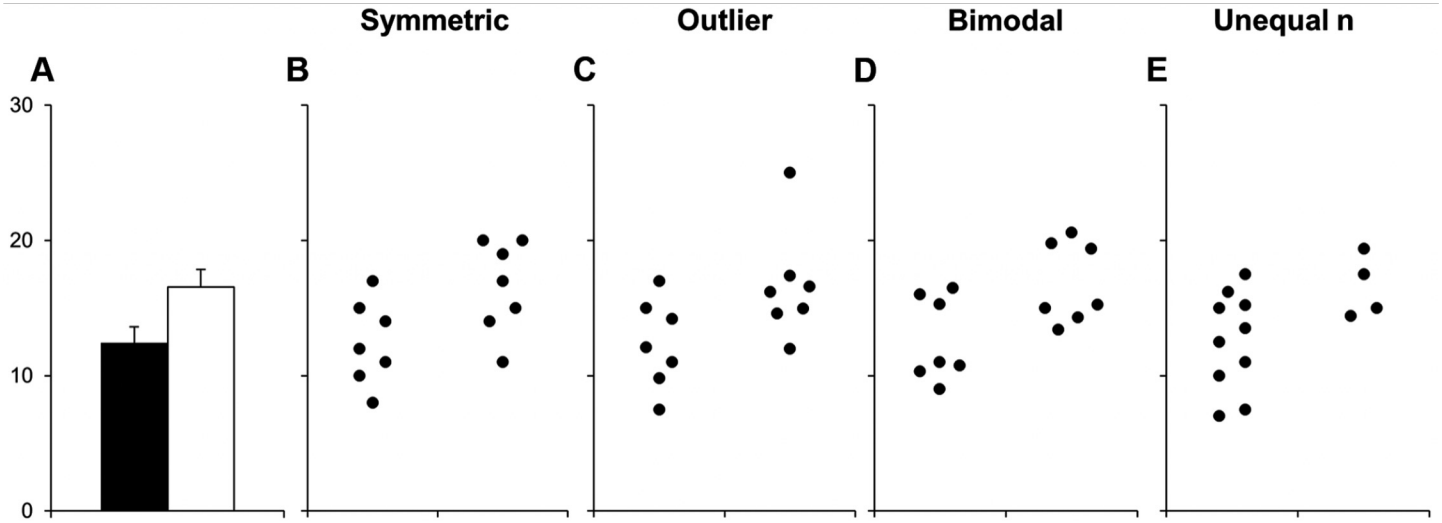
Show the data

Weissgerber
et al., 2015



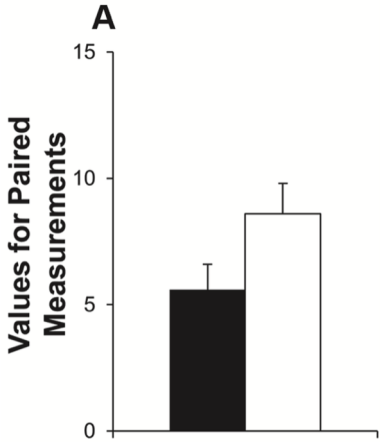
Show the data

Weissgerber et al., 2015



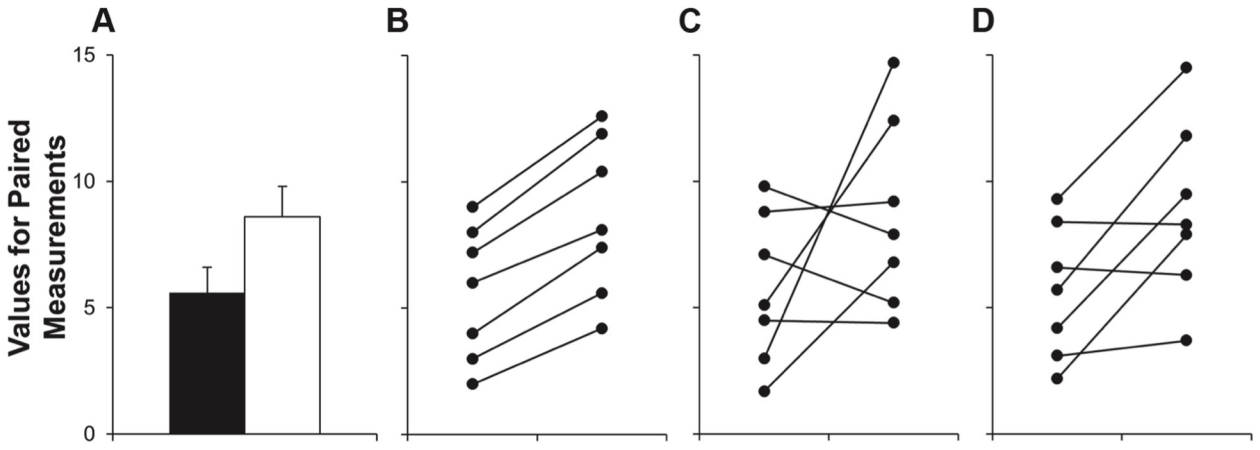
Show the data

Weissgerber
et al., 2015



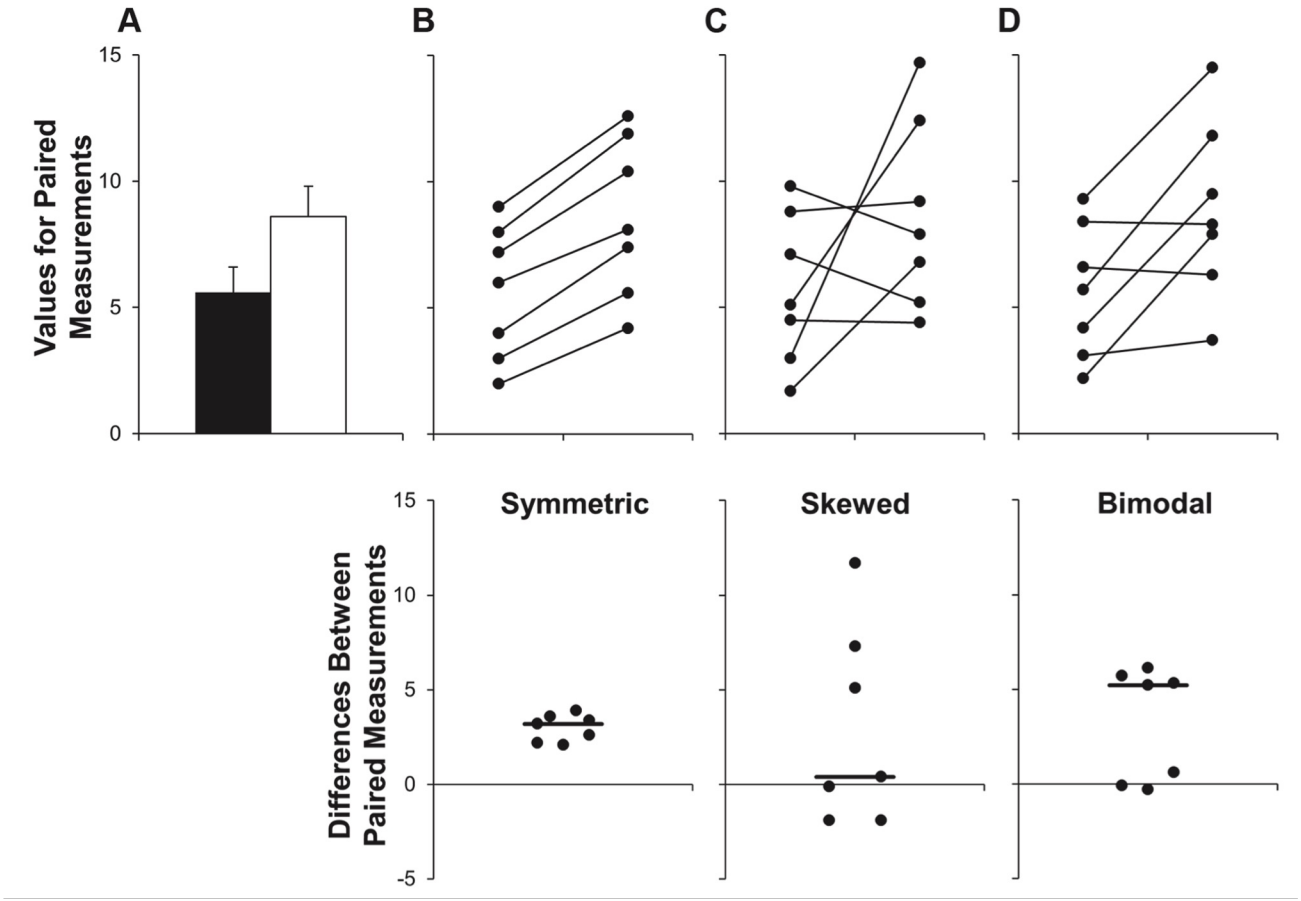
Show the data

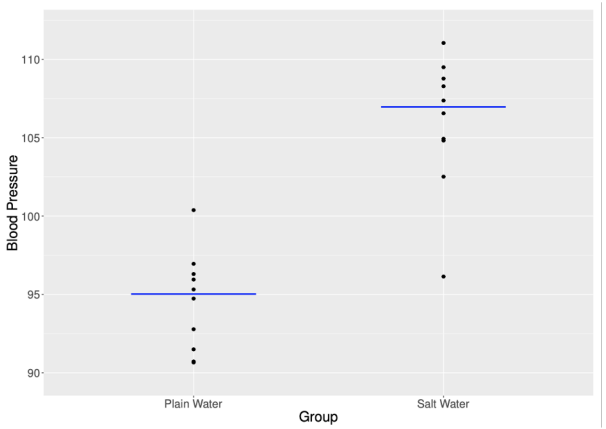
Weissgerber
et al., 2015

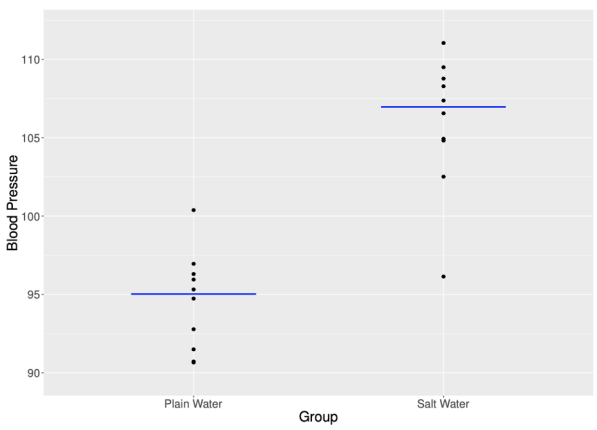


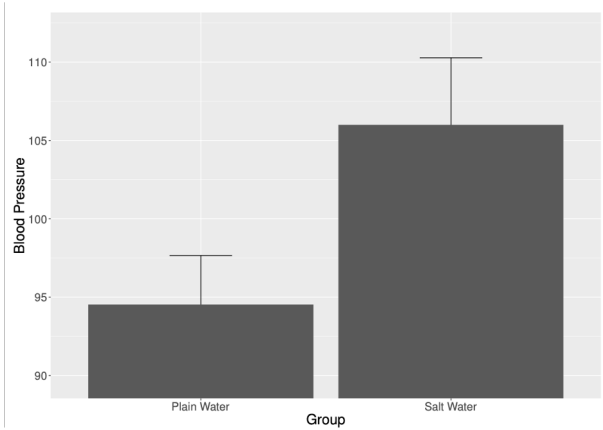
Show the data

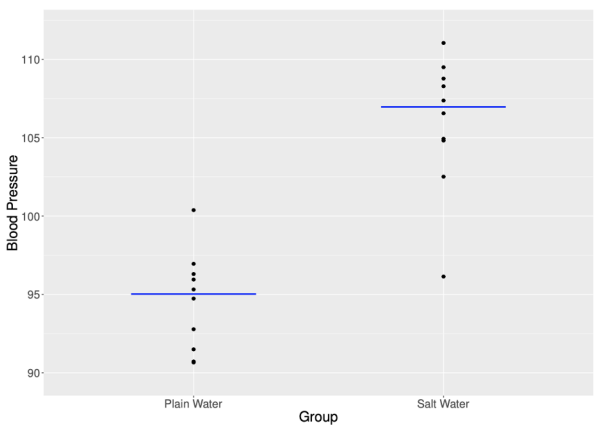
Weissgerber
et al., 2015

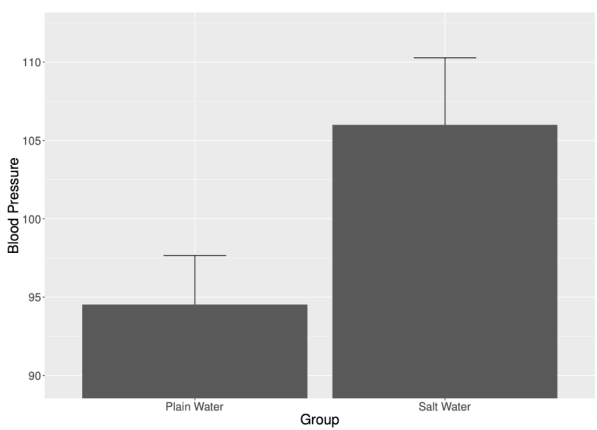


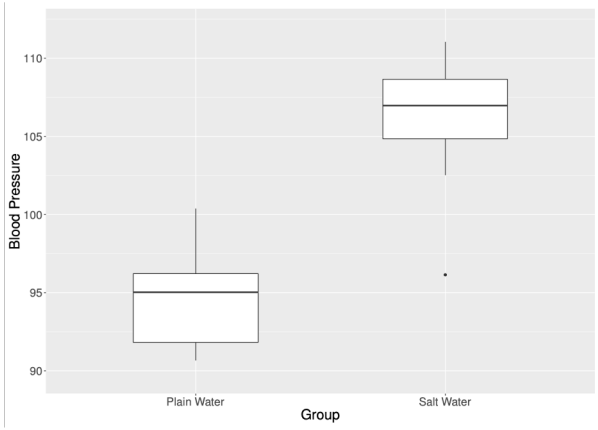

110
105
100
95
90
Blood Pressure
Plain Water
Salt Water
Group

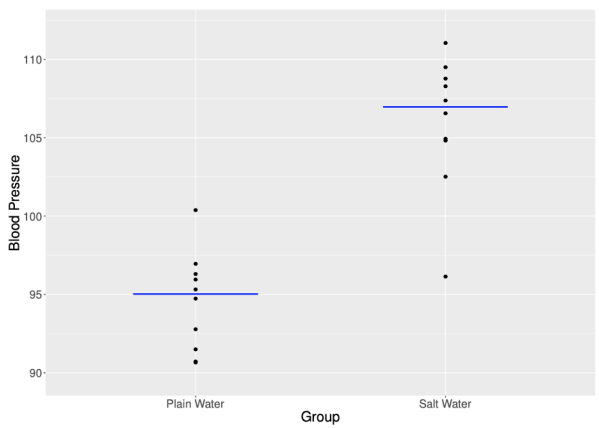

110
105
100
95
90
Blood Pressure
Plain Water
Salt Water
Group

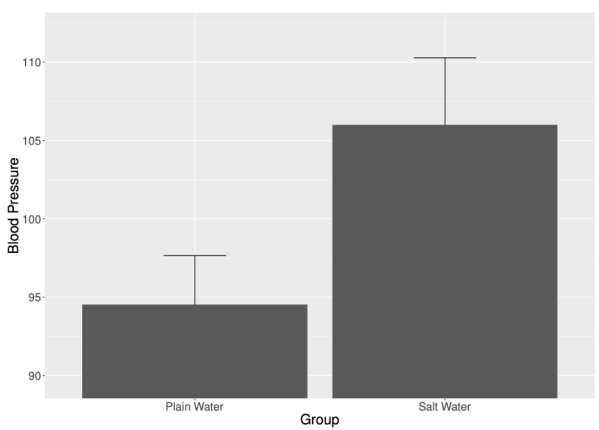

110
105
100
95
90
Blood Pressure
Plain Water
Salt Water
Group

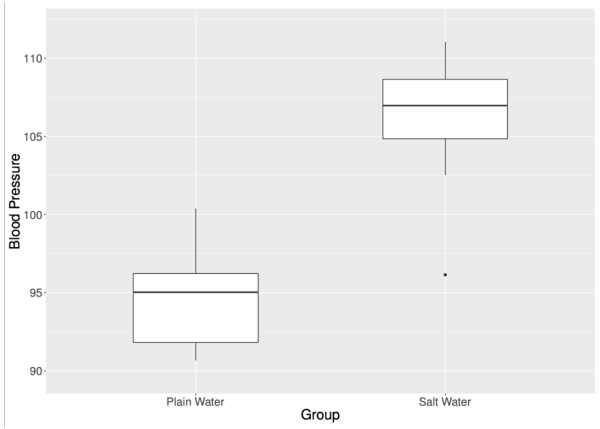

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

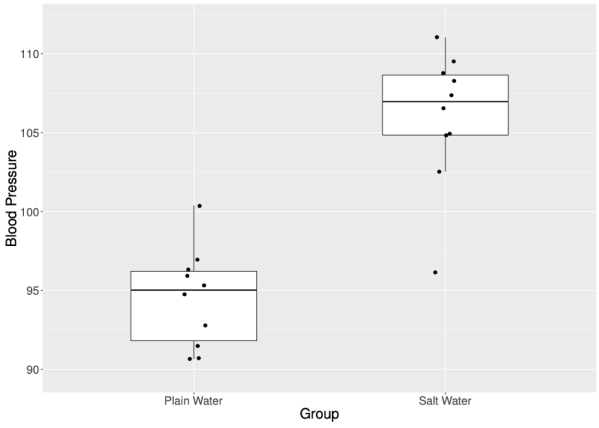

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

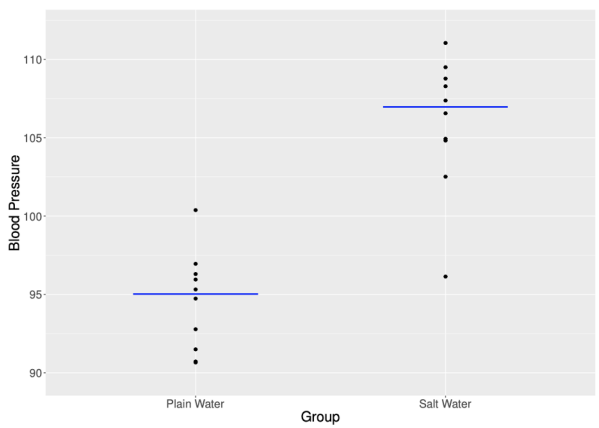

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

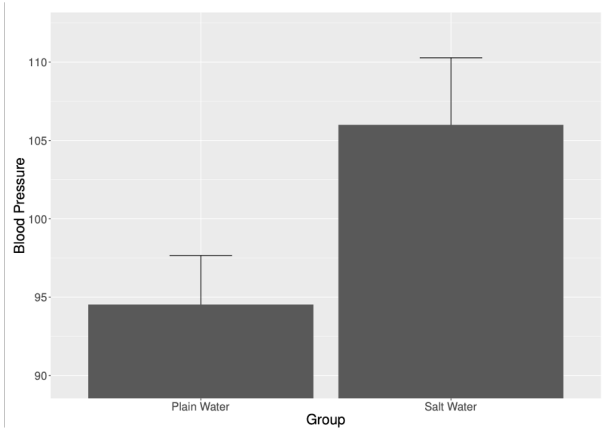

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

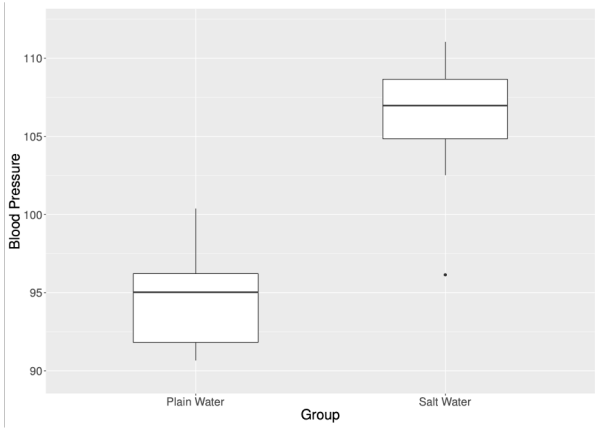

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

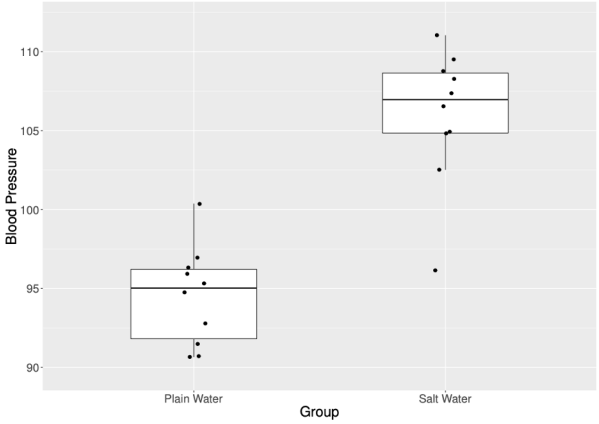

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

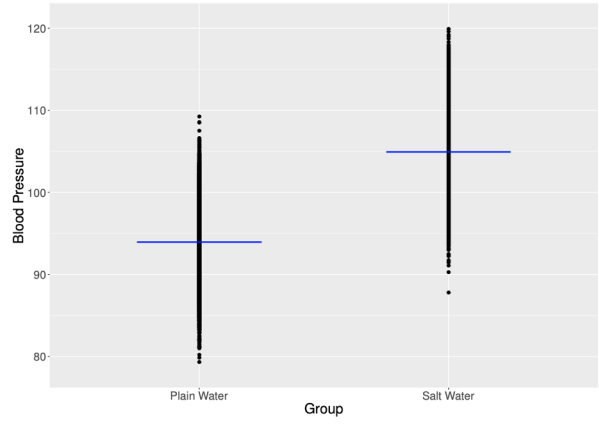

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

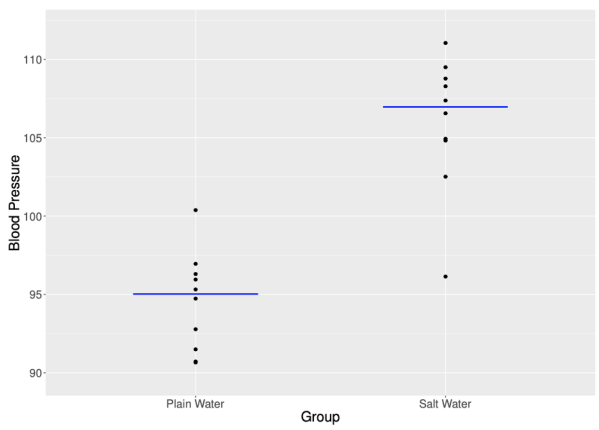

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

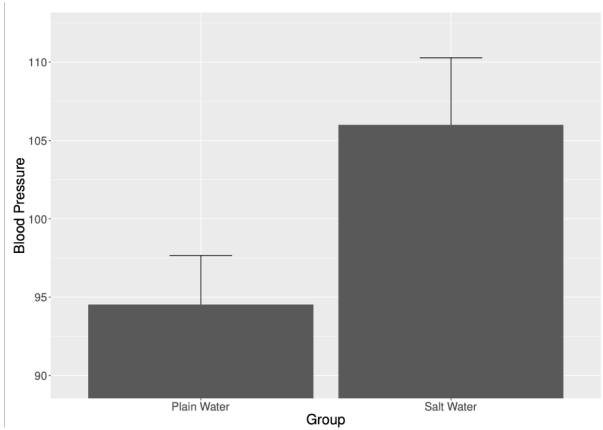

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

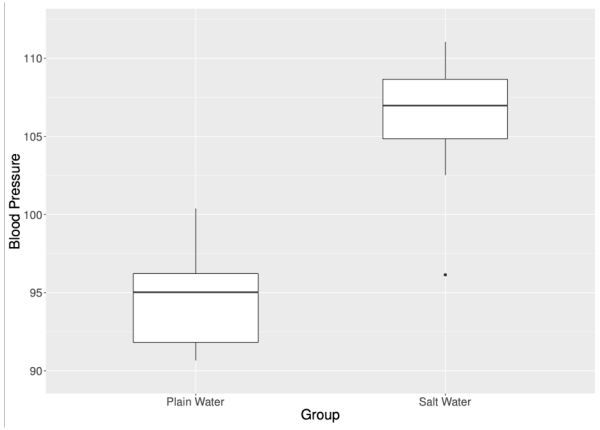

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

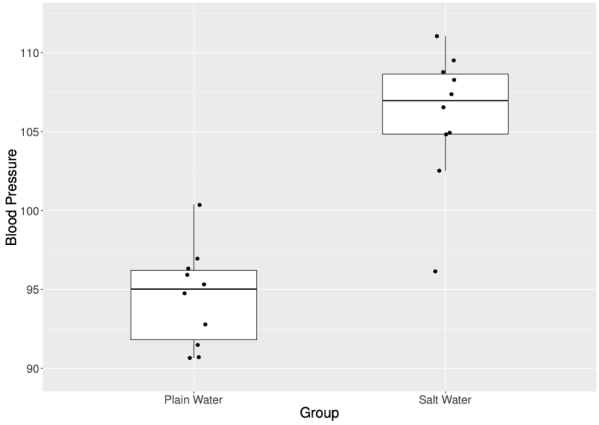

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

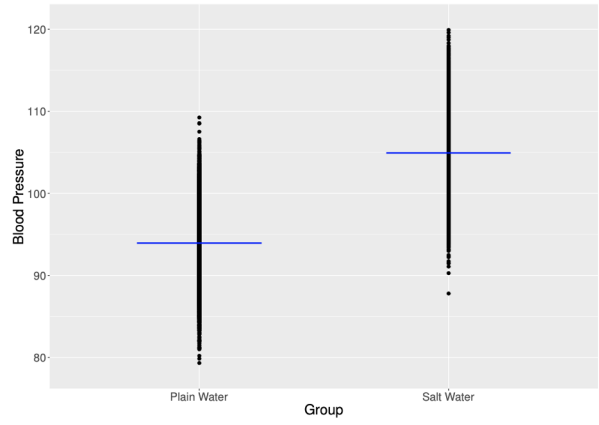

Blood Pressure
120
110
100
90
80
Plain Water
Salt Water
Group

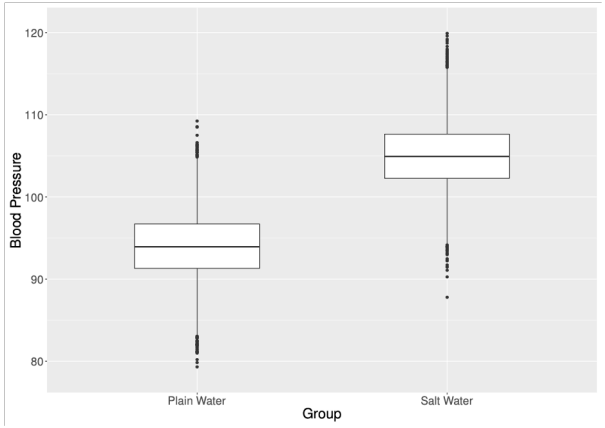

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group


Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

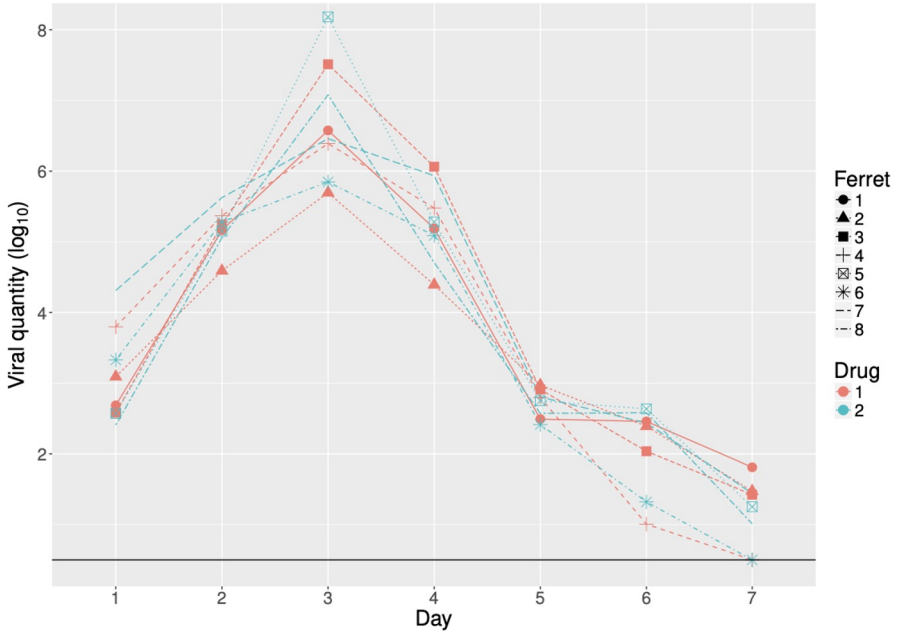

Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group


Blood Pressure
110
105
100
95
90
Plain Water
Salt Water
Group

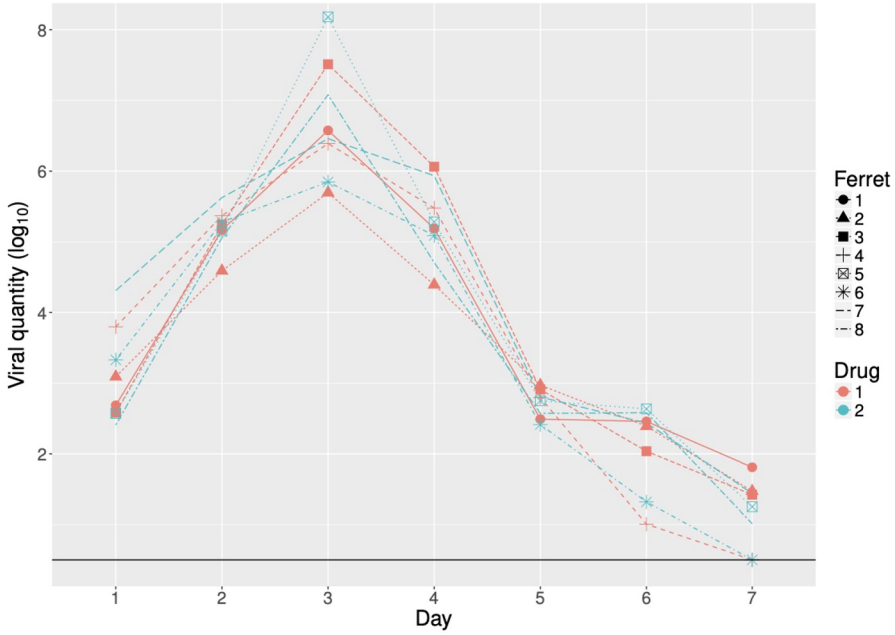

Blood Pressure
120
110
100
90
80
Plain Water
Salt Water
Group

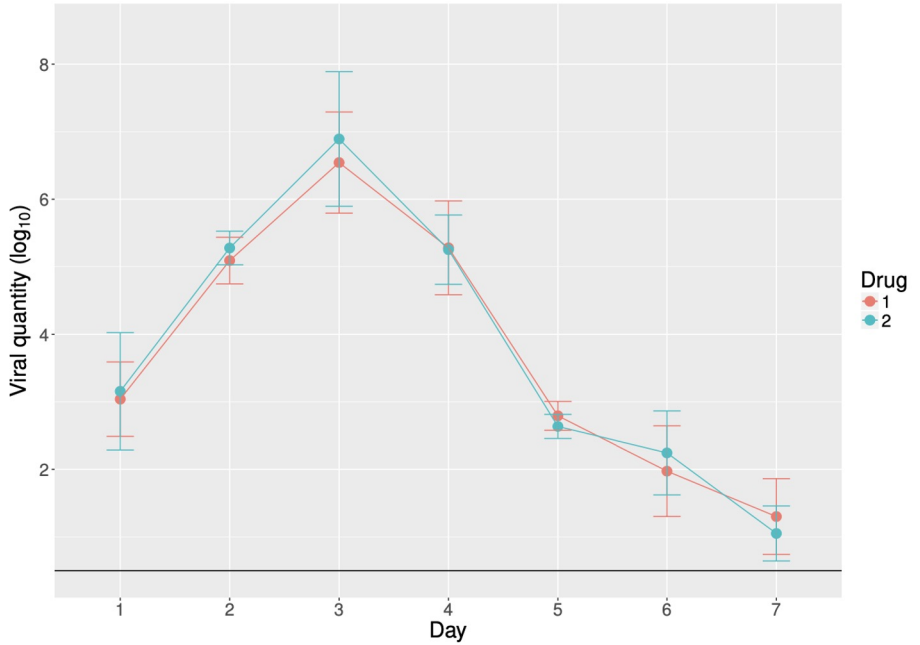

Blood Pressure
120
110
100
90
80
Plain Water
Salt Water
Group

Less may be more

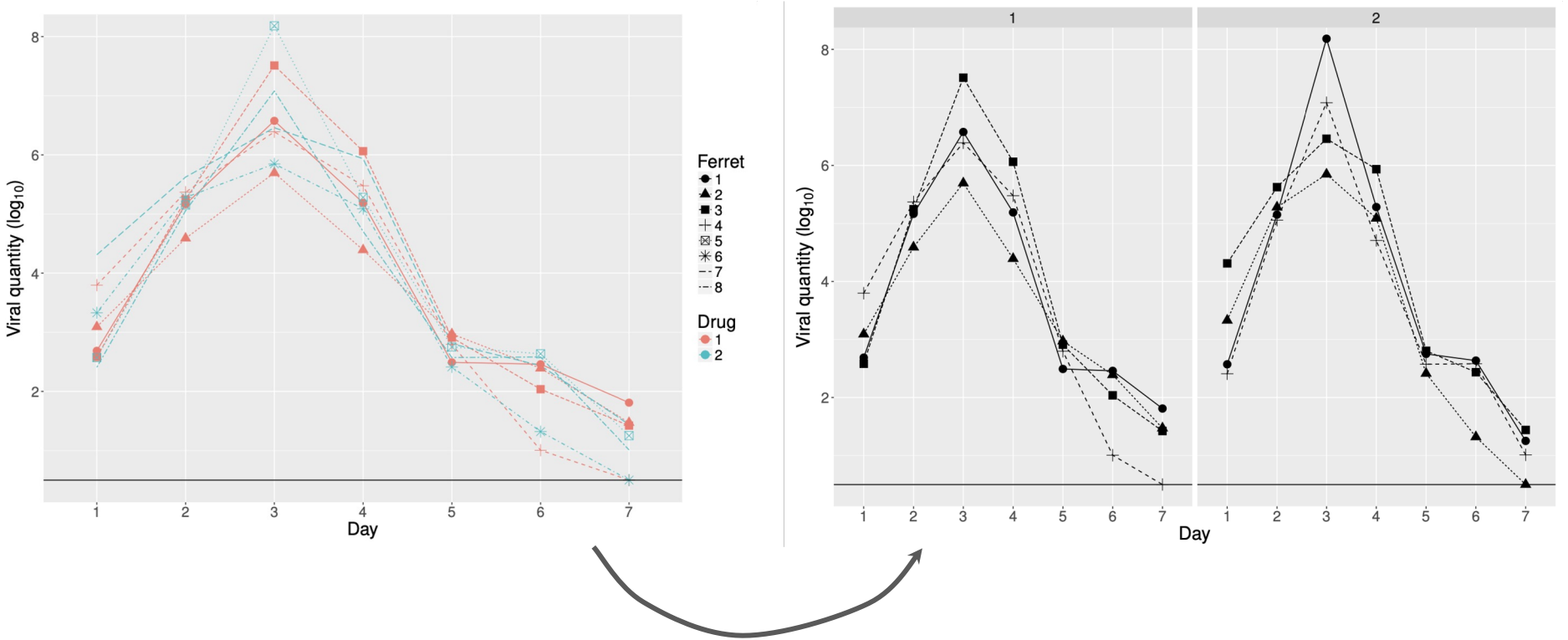


Less may be more





Less may be more



Summary statistics can be misleading

Hehman & Xie, 2021

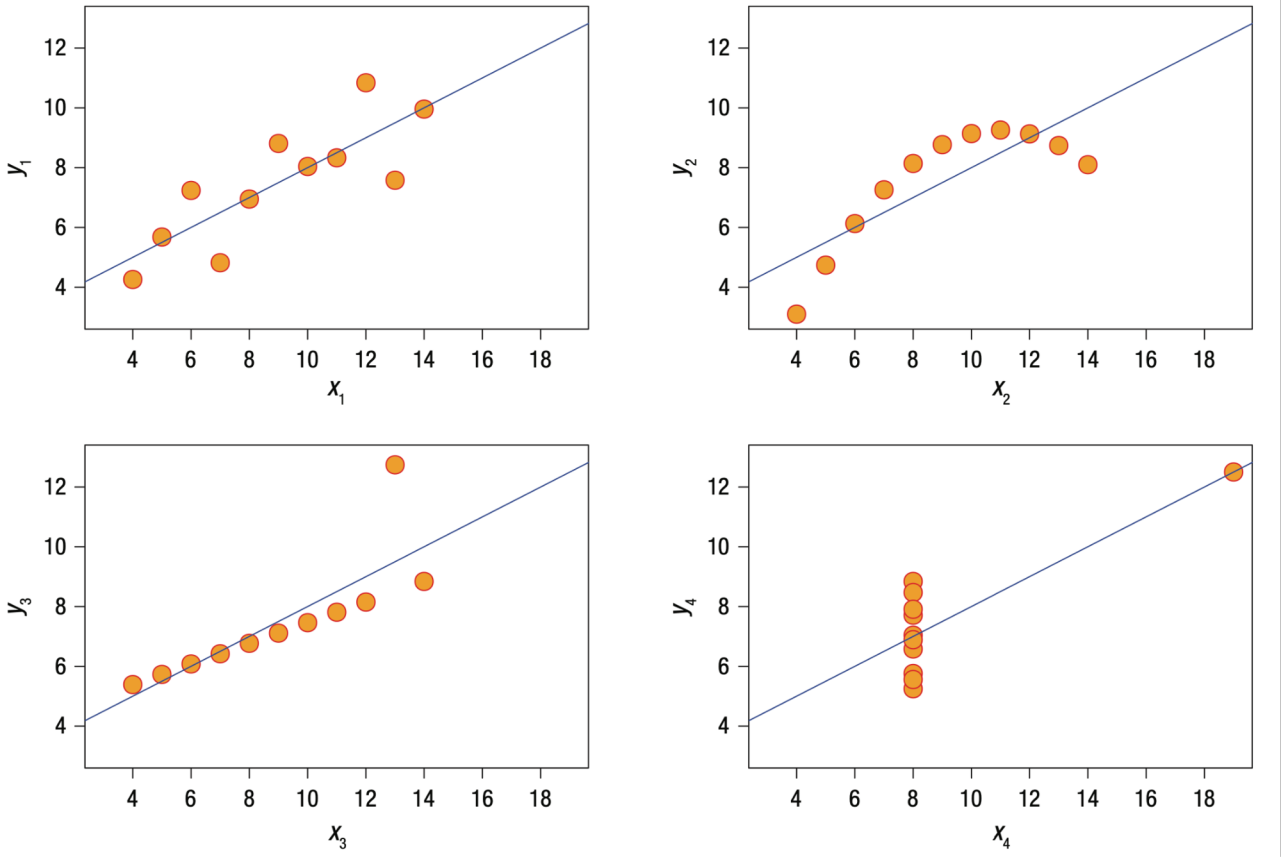
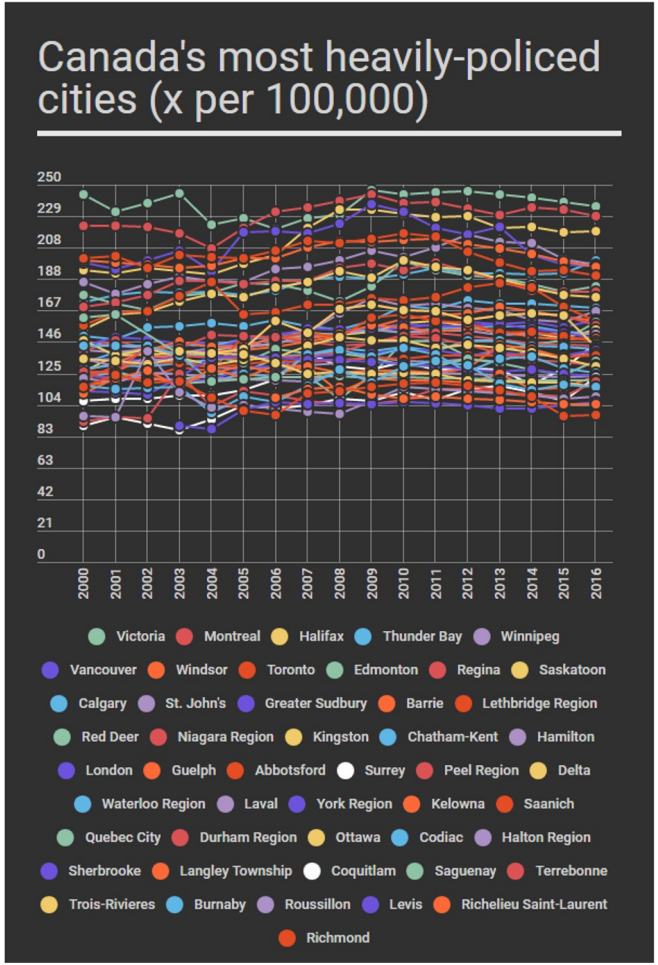


Fig. 1. Anscombe's quartet. In all four data sets depicted, the mean of x is 9, the variance of x is 11, the mean of y is 7.5, the variance of y is 4.12, and the correlation between x and y is .82. Important features of the data are hidden unless the individual observations are visualized.

Summarising your data (e.g. “detonator plots”) can remove information

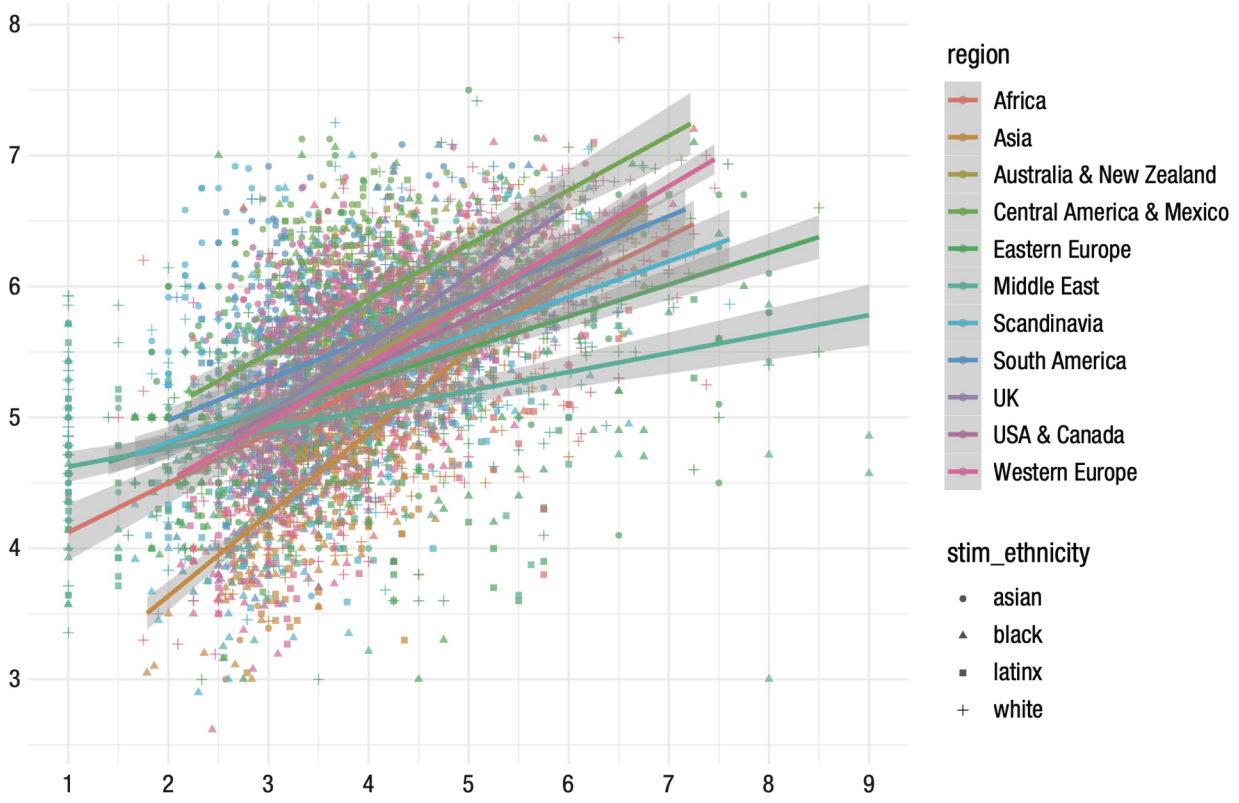
Keep it simple



(i found this on reddit)

Keep it simple

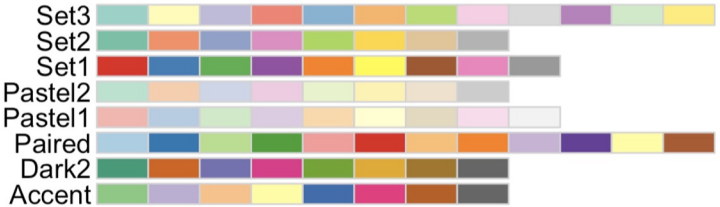
Hehman & Xie, 2021



Colour is important!

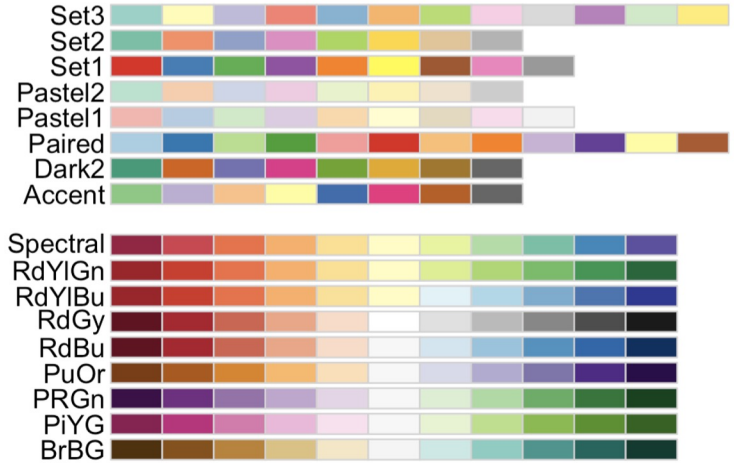
Colour is important!

- Categorical / qualitative



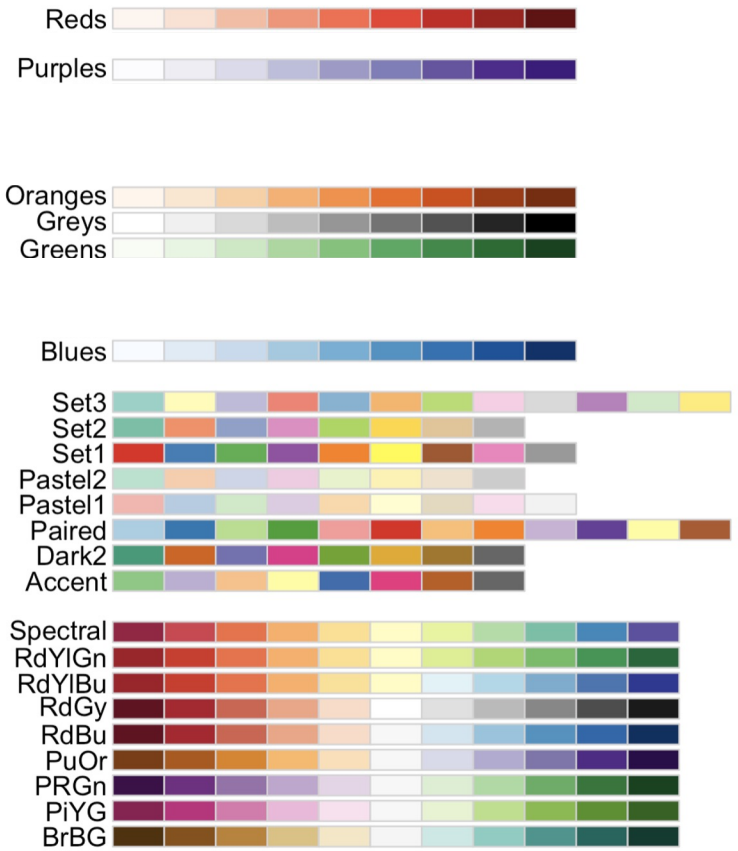
Colour is important!

- Categorical / qualitative
- Continuous
  - Diverging



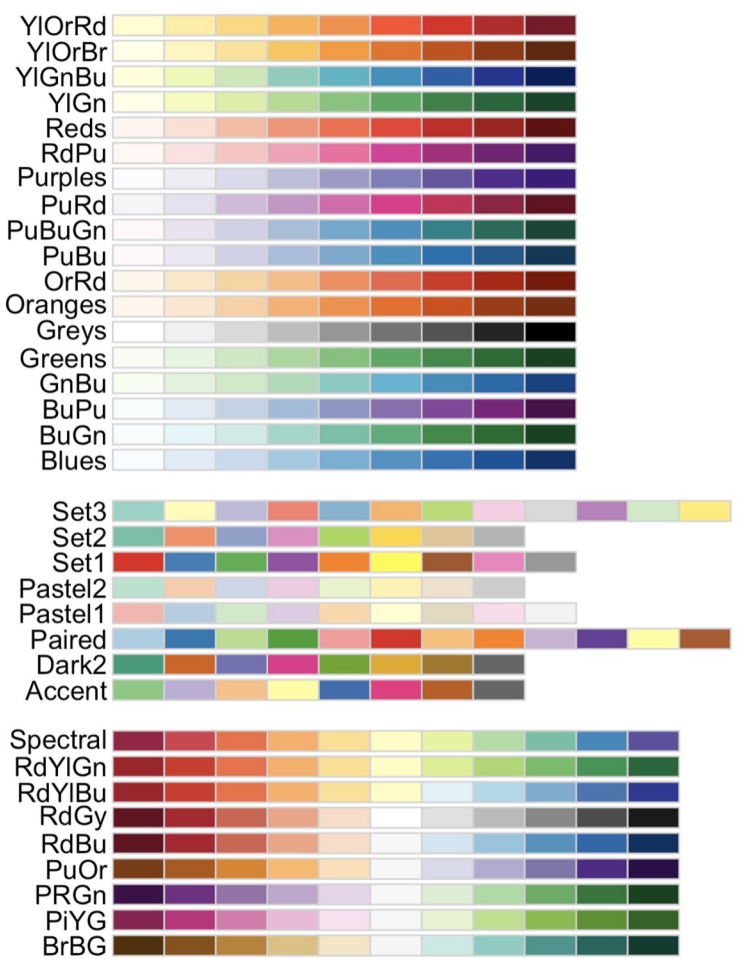
Colour is important!

- Categorical / qualitative
- Continuous
  - Diverging
  - Sequential
    - Single hue



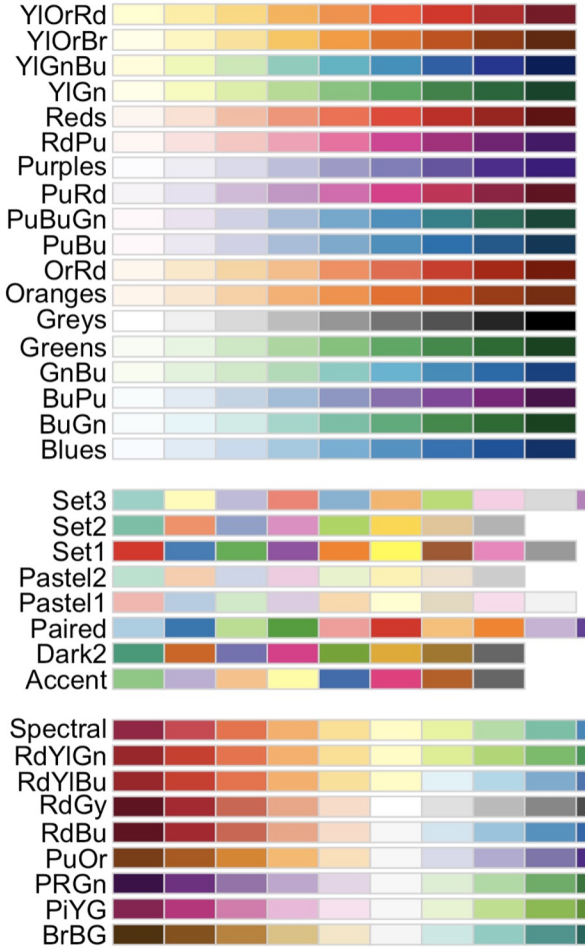
Colour is important!

- Categorical / qualitative
- Continuous
  - Diverging
  - Sequential
    - Single hue
    - Multi-hue



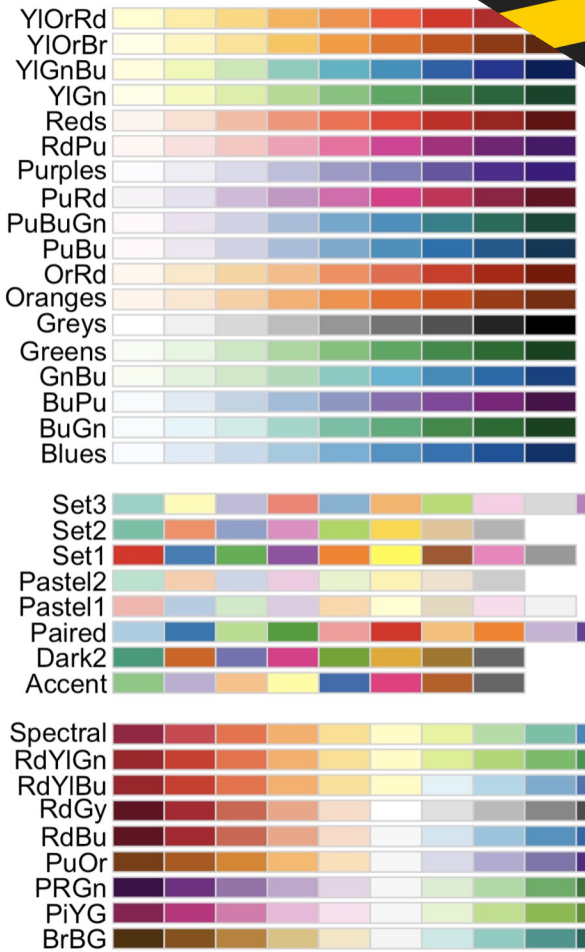
Colour is important!

Bonus: Viridis
https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html



Colour is important!

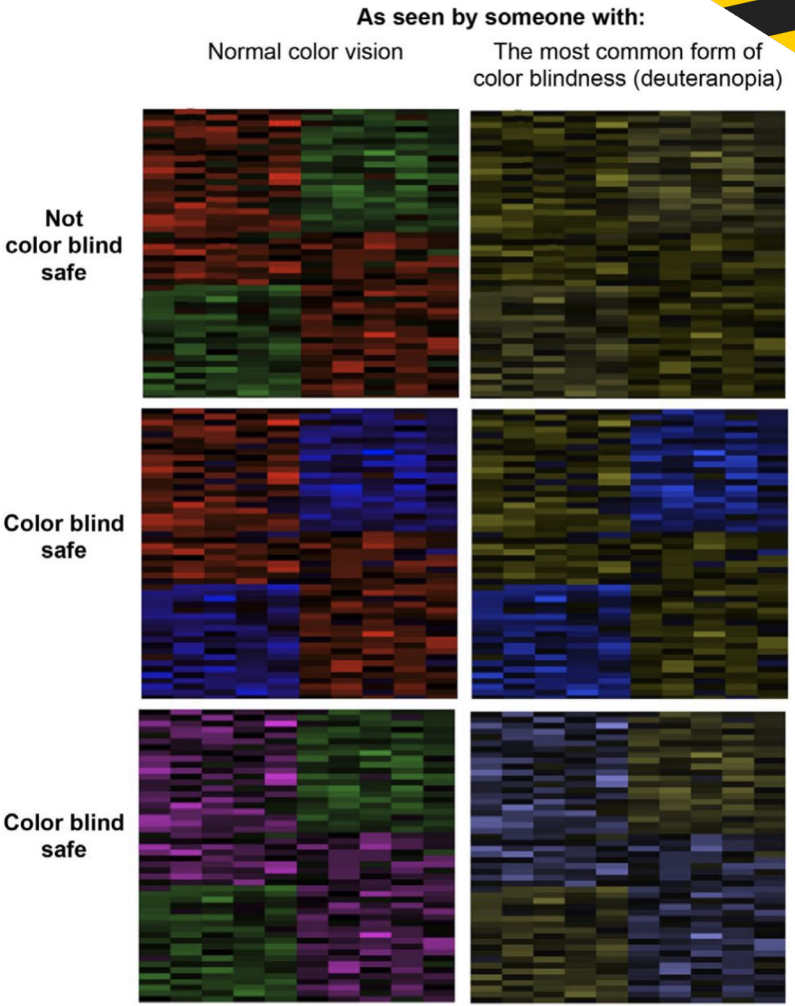
Bonus: Viridis
https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html



Colour is important!

Weissgerber et al., 2019
(using Color Oracle)
https://colororacle.org

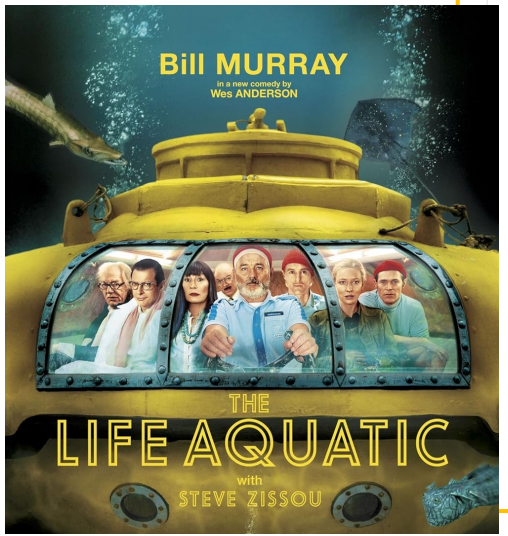
Color brewer:
https://colorbrewer2.org



Colour is important!

- There are lots of palettes out there, once you've had enough of Colour Brewer + Viridis:

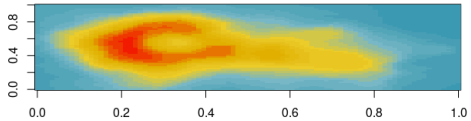
https://github.com/karthik/wesanderson



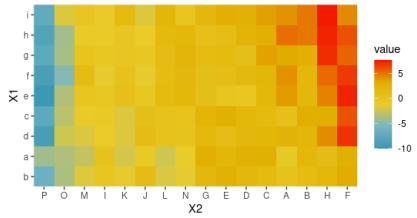
The Life Aquatic with Steve Zissou (2004)

```
wes_palette("Zissou1")
```


```
pal <- wes_palette("Zissou1", 21, type = "continuous")
image(volcano, col = pal)
```

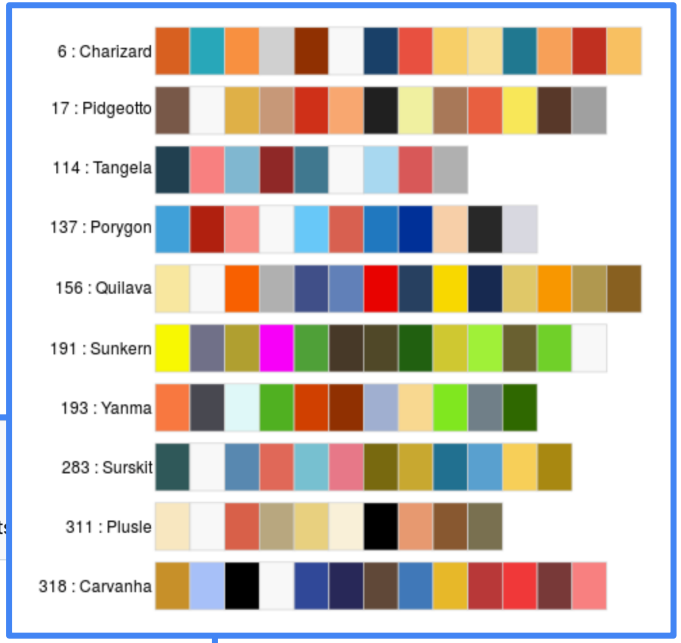


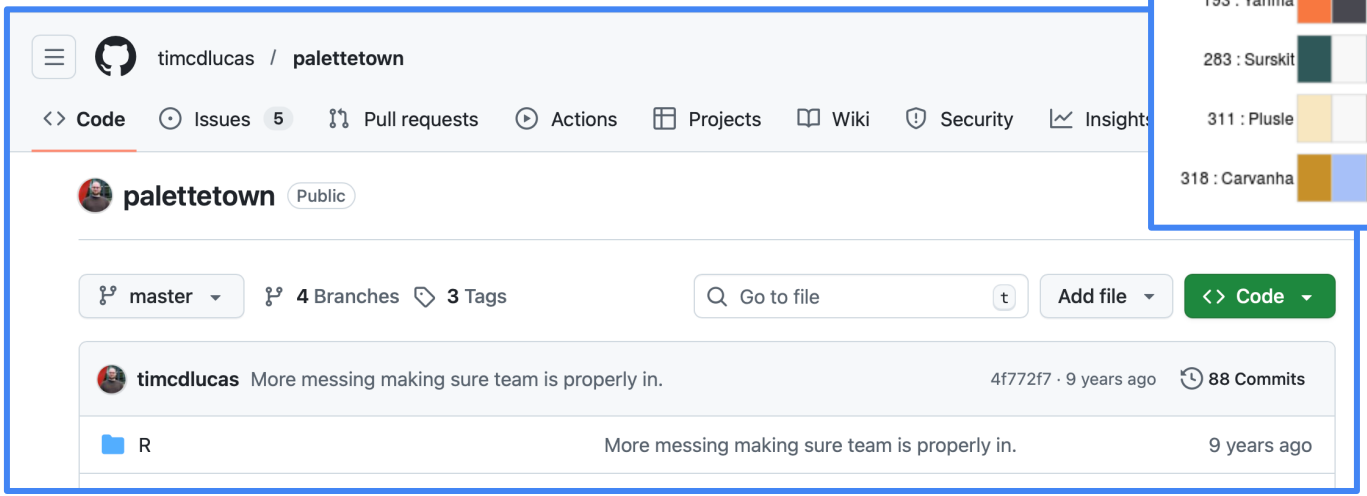

```
pal <- wes_palette("Zissou1", 100, type = "continuous")
# heatmap is a local dataset
ggplot(heatmap, aes(x = X2, y = X1, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colours = pal) +
  scale_x_discrete(expand = c(0, 0)) +
  scale_y_discrete(expand = c(0, 0)) +
  coord_equal()
```



Colour is important!

- There are lots of palettes out there, once you've had enough of Colour Brewer + Viridis:





https://github.com/timcdlucas/palettetown

Colour is important!

- If you can't find the colour palette you're looking for, make up your own!

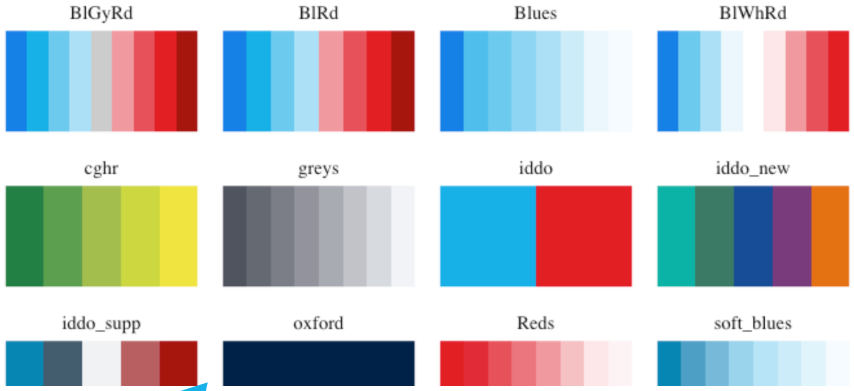
README CC0-1.0 license

iddoPal

Colour palettes for R inspired by the Infectious Diseases Data Observatory.

This package was written using the `palettes` package, using the accompanying guide. See also Gerry Ryan's `idpalette`.

Included palettes



https://mccarthy-m-g.github.io/palettes/articles/creating-packages.html

Colour is important!

- If you can't find the colour palette you're looking for, make up your own!

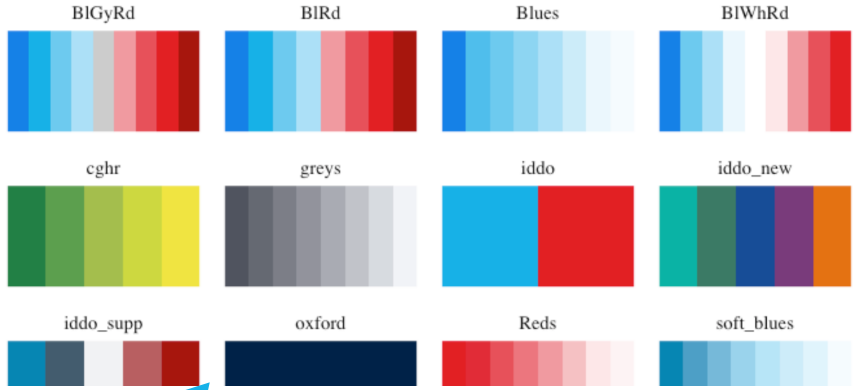
README CC0-1.0 license

iddoPal

Colour palettes for R inspired by the Infectious Diseases Data Observatory.

This package was written using the `palettes` package, using the accompanying guide. See also Gerry Ryan's `idpalette`.

Included palettes

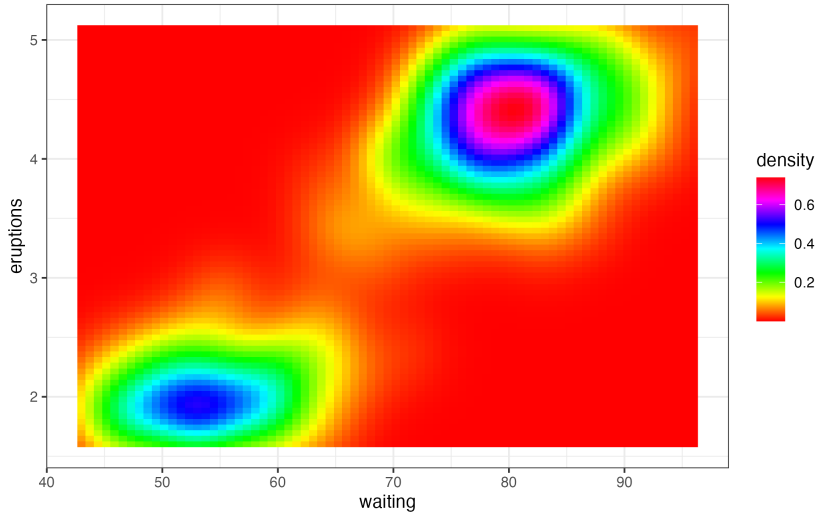


A little diversion: I love maps

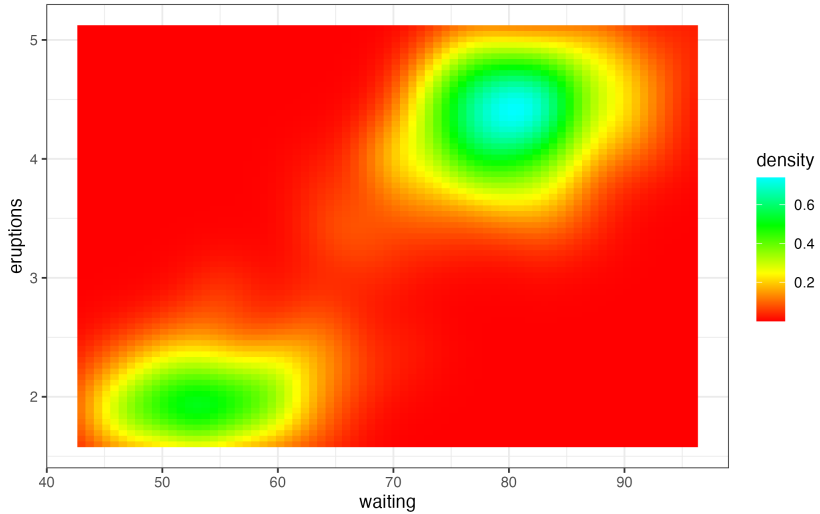
- Background should be distinct
- Beware blue
- Consider detail carefully
- Keep track of scale, projection + aggregation
- And all of the other rules apply too

```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))
```

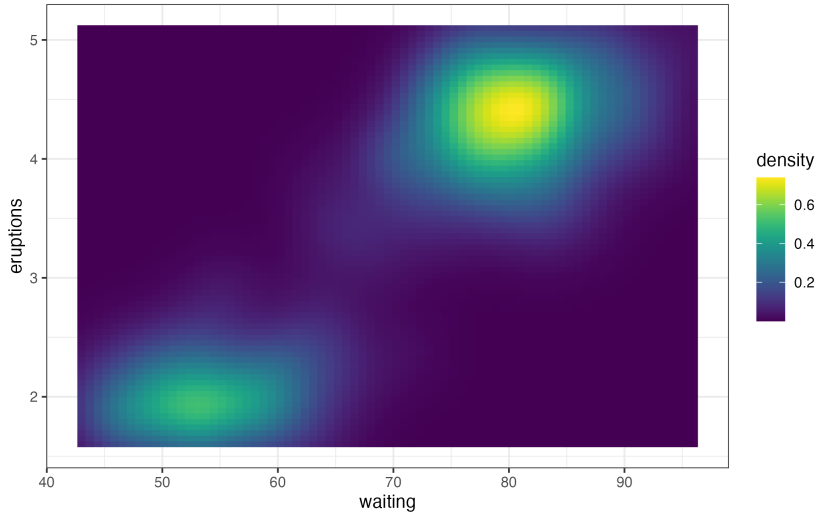
```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))

# this palette has very similar extremes:
p + scale_fill_gradientn(colours = rainbow(100))
```



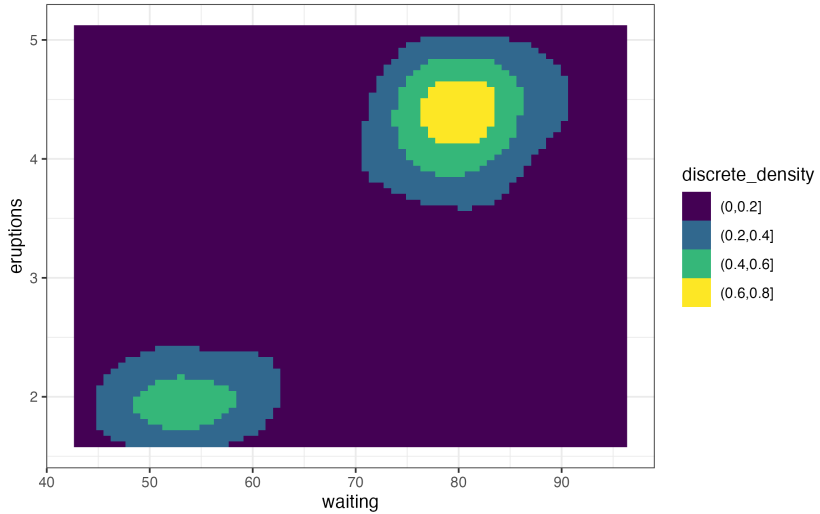
```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))

# this palette has very similar extremes:
p + scale_fill_gradientn(colours = rainbow(100))
# although even if we chop off extremes, there are some RdGn issues:
p + scale_fill_gradientn(colours = rainbow(100, end = 0.5))
```



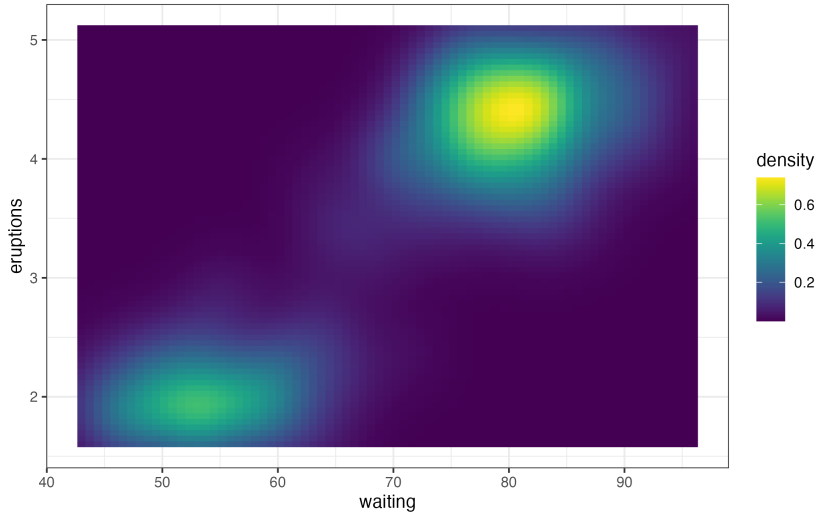
```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))

# this palette has very similar extremes:
p + scale_fill_gradientn(colours = rainbow(100))
# although even if we chop off extremes, there are some RdGn issues:
p + scale_fill_gradientn(colours = rainbow(100, end = 0.5))
# viridis probably gets us there:
p + scale_fill_gradientn(colours = viridis(100))
```



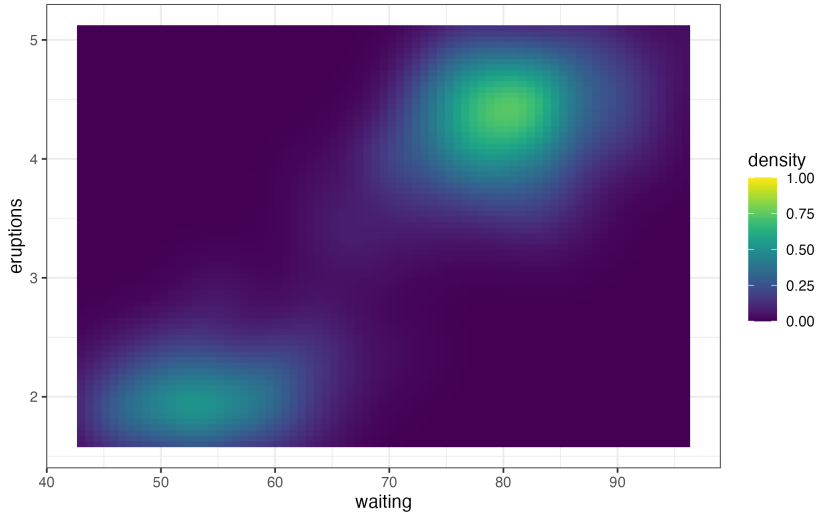
```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))

# this palette has very similar extremes:
p + scale_fill_gradientn(colours = rainbow(100))
# although even if we chop off extremes, there are some RdGn issues:
p + scale_fill_gradientn(colours = rainbow(100, end = 0.5))
# viridis probably gets us there:
p + scale_fill_gradientn(colours = viridis(100))

# Don't be afraid to contour!
ggplot(dat %>%
         mutate(discrete_density = cut(density, breaks = seq(0, 0.8, 0.2))),
       aes(waiting, eruptions)) +
  geom_raster(aes(fill = discrete_density)) +
  scale_fill_manual(values = viridis(4))
```



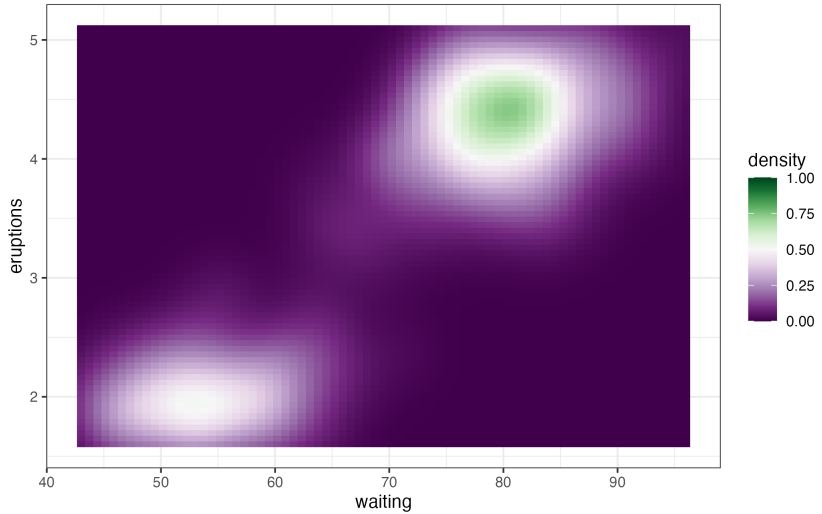
```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))

# this palette has very similar extremes:
p + scale_fill_gradientn(colours = rainbow(100))
# although even if we chop off extremes, there are some RdGn issues:
p + scale_fill_gradientn(colours = rainbow(100, end = 0.5))
# viridis probably gets us there:
p + scale_fill_gradientn(colours = viridis(100))

# Don't be afraid to contour!
ggplot(dat %>%
         mutate(discrete_density = cut(density, breaks = seq(0, 0.8, 0.2))),
       aes(waiting, eruptions)) +
  geom_raster(aes(fill = discrete_density)) +
  scale_fill_manual(values = viridis(4))


# viridis is fine,
p + scale_fill_gradientn(colours = viridis(100))
```



```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))

# this palette has very similar extremes:
p + scale_fill_gradientn(colours = rainbow(100))
# although even if we chop off extremes, there are some RdGn issues:
p + scale_fill_gradientn(colours = rainbow(100, end = 0.5))
# viridis probably gets us there:
p + scale_fill_gradientn(colours = viridis(100))

# Don't be afraid to contour!
ggplot(dat %>%
         mutate(discrete_density = cut(density, breaks = seq(0, 0.8, 0.2))),
       aes(waiting, eruptions)) +
  geom_raster(aes(fill = discrete_density)) +
  scale_fill_manual(values = viridis(4))


# viridis is fine,
p + scale_fill_gradientn(colours = viridis(100))
# but what if extremes are important?
p + scale_fill_gradientn(colours = viridis(100), limits = c(0, 1))
```



```r
# some examples: maps and colour
library(RColorBrewer)
library(grDevices)

dat <- faithfuld %>%
  mutate(density = density - min(density) + 1e-6,
         density = density / 0.05)

p <- ggplot(dat, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density))

# this palette has very similar extremes:
p + scale_fill_gradientn(colours = rainbow(100))
# although even if we chop off extremes, there are some RdGn issues:
p + scale_fill_gradientn(colours = rainbow(100, end = 0.5))
# viridis probably gets us there:
p + scale_fill_gradientn(colours = viridis(100))

# Don't be afraid to contour!
ggplot(dat %>%
         mutate(discrete_density = cut(density, breaks = seq(0, 0.8, 0.2))),
       aes(waiting, eruptions)) +
  geom_raster(aes(fill = discrete_density)) +
  scale_fill_manual(values = viridis(4))


# viridis is fine,
p + scale_fill_gradientn(colours = viridis(100))
# but what if extremes are important?
p + scale_fill_gradientn(colours = viridis(100), limits = c(0, 1))
# and so is the middle:
p + scale_fill_gradientn(colours = brewer.pal(11, "PRGn"), limits = c(0, 1))

```

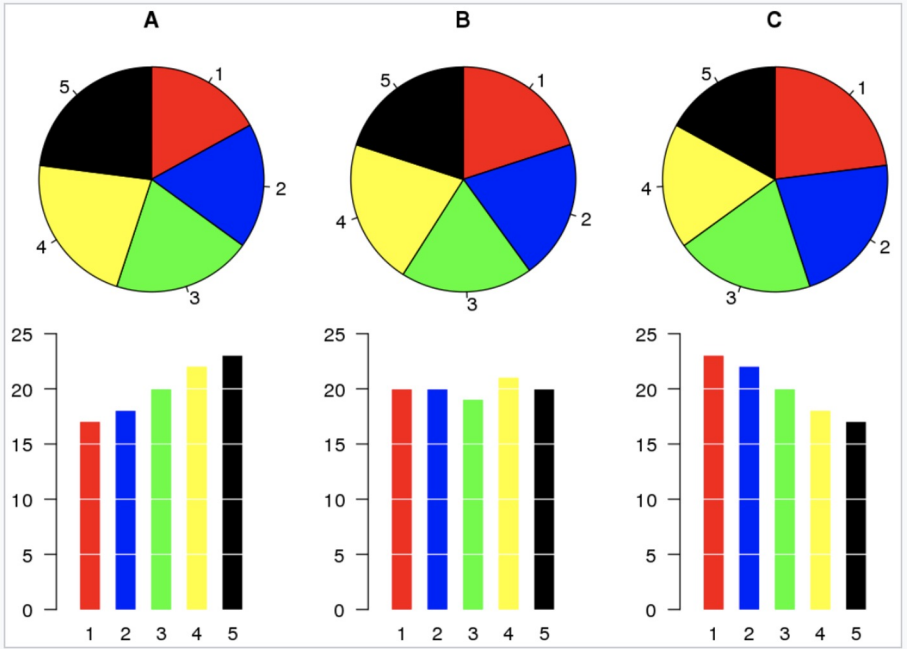


Finally, if in doubt …

Finally, if in doubt …

… get another opinion!

… and reconsider that pie chart!



Three sets of percentages, plotted as both piecharts and barcharts. Comparing the data on barcharts is generally much easier.

Hill, Andrew. "Are pie charts evil? An assessment of the value of pie and donut charts compared to bar charts." *Information Visualization* 24, no. 1 (2025): 3-23.

Bertini, Enrico, Niklas Elmqvist, and Thomas Wischgoll. "Judgment error in pie chart variations." In *Proceedings of the Eurographics/IEEE VGTC conference on visualization: Short papers*, pp. 91-95. 2016.

Siirtola, Harri. "The cost of pie charts." In *2019 23rd International Conference Information Visualisation (IV)*, pp. 151-156. IEEE, 2019.

Time for a quiz!

https://www.matplotlib-journey.com/bonus/design-principles

In summary ..

1. Show the data
2. But don’t show too much
3. Uncertainty is important
4. And so is proportion
5. And colour
6. And finally, if in doubt … get another opinion!

~ Let's compare notes ~