



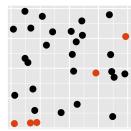
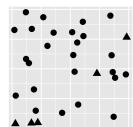
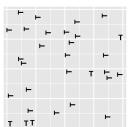
## Effective data visualisation

## Graphical perception

1. Preattentive processing
2. Proximity
3. Position vs angle
4. Colour matters

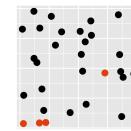
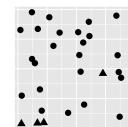
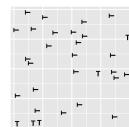
2 / 43

### Preattentive processing



3 / 43

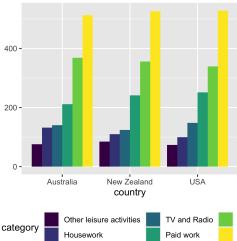
### Preattentive processing colour > form (shape)



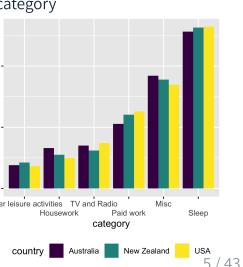
4 / 43

## Proximity Make easy comparisons by grouping elements together

- compare time use by categories within each country

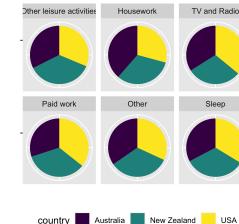
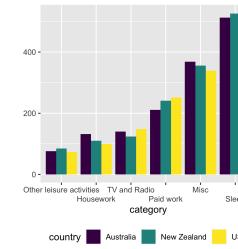


- compare time use by countries within each category



5 / 43

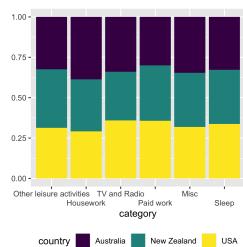
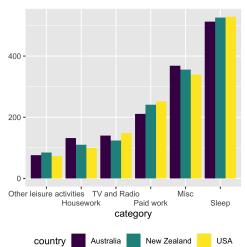
## Position vs angle position > angle



Pie charts are BAD!!

6 / 43

## Absolute vs relative positions absolute > relative



7 / 43

## Colour matters

- Colour spaces
- Colour scales
- Colour blindness

8 / 43

### 3 ways to represent colour spaces

1. RGB
2. HSV/HSL
3. HCL for humans

9 / 43

### RGB

- Red (0-255): amount of red light
- Green (0-255): amount of green light
- Blue (0-255): amount of blue light

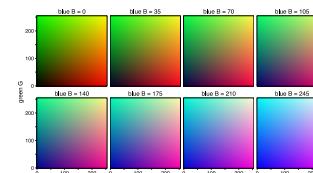


image credit: Claus O. Wilke

10 / 43

### HSV

- Hue (0-360): hue of the colour
- Saturation (0-1): colourfulness relative to the brightness of the colour
- Value (0-1): subjective perception of amount of light emitted

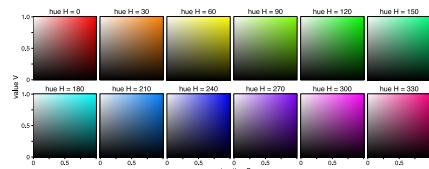


image credit: Claus O. Wilke

11 / 43

### HSL

- Hue (0-360): hue of the colour
- Lightness (0-1): brightness relative to the brightness of a illuminated white
- Saturation (0-1): colourfulness relative to the brightness of the colour

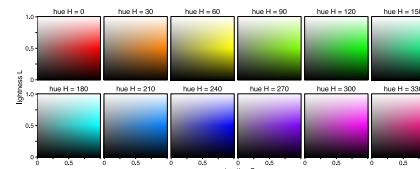


image credit: Claus O. Wilke

12 / 43

## HCL aka polar LUV

- › Hue (0-360): hue of the colour
- › Chroma (0-180): degree of vividness of a colour
- › Luminance (0-100): amount of light emitted

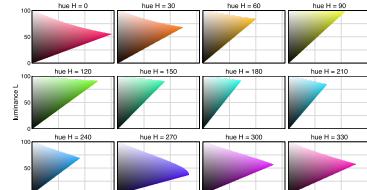
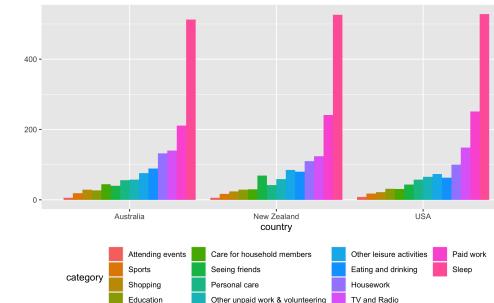


image credit: Claus O. Wilke

13 / 43

## Encoding too much



14 / 43

## Colour scales

3 fundamental use cases for colours in data visualisations:

1. use colour to distinguish groups of data from each other
2. use colour to represent data values
3. use colour to highlight

3 types of colour palettes ColorBrewer

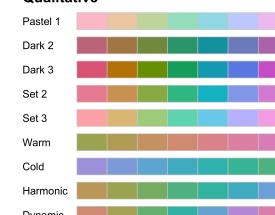
1. Qualitative
2. Sequential
3. Diverging

15 / 43

## Qualitative palettes for categorical data with no intrinsic ordering

```
colorspace::hcl_palettes("Qualitative", plot = TRUE, n = 7)
```

### Qualitative



16 / 43

## Sequential palettes for ordered data from high to low

```
colorspace::hcl_palettes("Sequential", plot = TRUE, n = 7)
```



17 / 43

## Diverging palettes for mid-range values and extremes at both ends

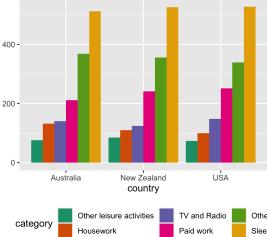
```
colorspace::hcl_palettes("Diverging", plot = TRUE, n = 7)
```



18 / 43

## Use colour palettes

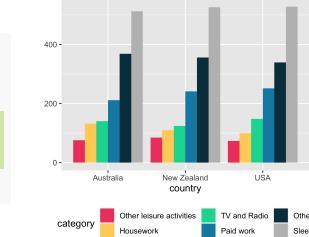
```
time_use %>%
  ggplot(aes(country, time_minutes)) +
  geom_col(
    aes(fill = category),
    position = "dodge") +
  scale_fill_brewer(palette = "Dark2") +
  labs(y = "") +
  theme(legend.position = "bottom")
```



19 / 43

## Set custom colours

```
time_use %>%
  ggplot(aes(country, time_minutes)) +
  geom_col(
    aes(fill = category),
    position = "dodge") +
  scale_fill_manual(
    values = c("#EFA76F", "#FFD166",
               "#06D6A0", "#118AB2",
               "#073B4C", "grey")) +
  labs(y = "") +
  theme(legend.position = "bottom")
```



20 / 43

## Colour-vision deficiency



➤ Red-green colour-vision deficiency (deutanomaly & protanomaly) is the most common.



➤ Blue-green colour-vision deficiency (tritanomaly) is rare but does occur.

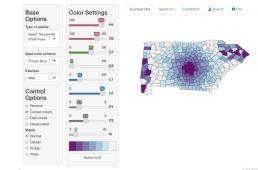
💡 Approximately 8% of males and 0.5% of females suffer from some sort of color-vision deficiency.

reference: Claus O. Wilke [Fundamentals of Data Visualization](#)

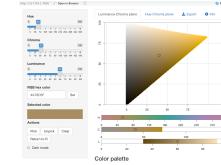
21 / 43

## Choose colours using {colorspace}

➤ `colorspace::hclwizard()`



➤ `colorspace::hcl_color_picker()`



22 / 43

## Scales

- Control how data is mapped to perceptual properties, and produce guides (axes and legends) which allow us to read the plot.
- Important arguments: `breaks`, `labels`, and `limits`.
- Naming scheme: `scale_[aes]_[datatype]()`

23 / 43

scale	Description
<code>scale_alpha</code> , <code>scale_alpha_continuous</code> , <code>scale_alpha_binned</code> , <code>scale_alpha_discrete</code>	Alpha transparency scales
<code>scale_x_binned</code> , <code>scale_y_binned</code>	Positional scales for binning continuous data (x & y)
<code>scale_colour_brewer</code> , <code>scale_fill_brewer</code> , <code>scale_colour_distiller</code> , <code>scale_fill_distiller</code>	Sequential, diverging and qualitative colour scales from colorbrewer.org
<code>scale_colour_continuous</code> , <code>scale_fill_continuous</code> , <code>scale_colour_binned</code> , <code>scale_fill_binned</code>	Continuous and binned colour scales
<code>scale_colour_discrete</code> , <code>scale_fill_discrete</code>	Discrete colour scales

Previous 1 2 3 4 Next  
24 / 43



## Publication-ready visualisation

25 / 43

## Towards publication-ready visualisation



26 / 43

## Exploratory data visualisation

- For **internal use** only. Need to be able to create rapidly because your first attempt will never be the most revealing.
- Iteration is crucial for developing multiple displays of your data.

## Communication graphics

- When you **communicate** your findings, you need to spend much time polishing your graphics to eliminate distractions and focus on the storytelling.
- Iteration is crucial to ensure all the bits and pieces works well: labels, color choices, tick marks...

27 / 43

## Case study: COVID-19

```
covid19 <- read_csv("data/covid19-daily-cases.csv")
covid19

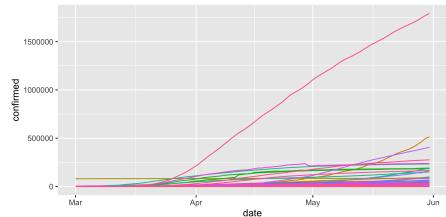
#> # A tibble: 15,677 x 3
#>   country_region date       confirmed
#>   <chr>          <date>     <dbl>
#> 1 Afghanistan   2020-03-01      1
#> 2 Afghanistan   2020-03-02      1
#> 3 Afghanistan   2020-03-03      2
#> 4 Afghanistan   2020-03-04      4
#> 5 Afghanistan   2020-03-05      4
#> 6 Afghanistan   2020-03-06      4
#> # ... with 15,671 more rows
```

28 / 43

## COVID-19 -scale-y

### Data as is

[Plot](#) [Code](#)

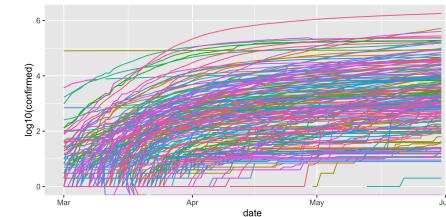


29 / 43

## COVID-19 -scale-y

### Logarithmic scale

[Plot](#) [Code](#)

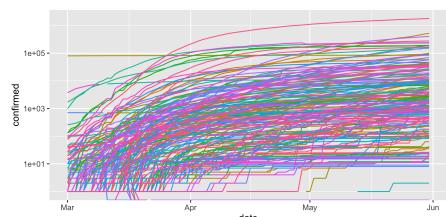


30 / 43

## COVID-19 -scale-y

### Logarithmic scale

[Plot](#) [Code](#)



31 / 43

## COVID-19 -scale-y -scale-x

```
covid19_rel <- covid19 %>%
  group_by(country_region) %>%
  mutate(days = as.numeric(date - min(date))) %>%
  ungroup()
covid19_rel
```

```
#> # A tibble: 15,677 x 4
#>   country_region date       confirmed days
#>   <chr>        <date>     <dbl>    <dbl>
#> 1 Afghanistan  2020-03-01     1      0
#> 2 Afghanistan  2020-03-02     1      1
#> 3 Afghanistan  2020-03-03     2      2
#> 4 Afghanistan  2020-03-04     4      3
#> 5 Afghanistan  2020-03-05     4      4
#> 6 Afghanistan  2020-03-06     4      5
#> # ... with 15,671 more rows
```

32 / 43

Rob J Hyndman's blog post on [Why log ratios are useful for tracking COVID-19](#)

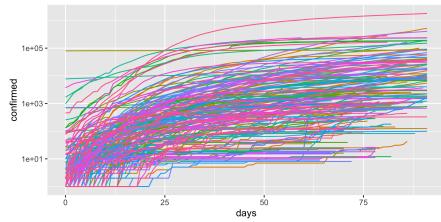
### COVID-19

- scale-y
- scale-x

### Relative days

> Plot

> Code



33 / 43

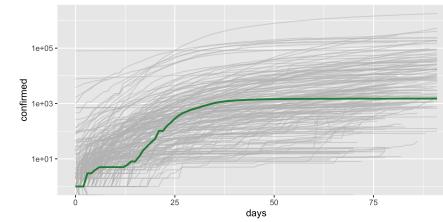
### COVID-19

- scale-y
- scale-x
- highlight

### Highlight New Zealand

> Plot

> Code



34 / 43

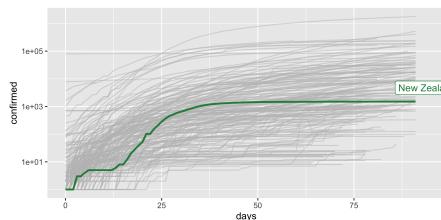
### COVID-19

- scale-y
- scale-x
- highlight
- annotate

### Label New Zealand

> Plot

> Code



35 / 43

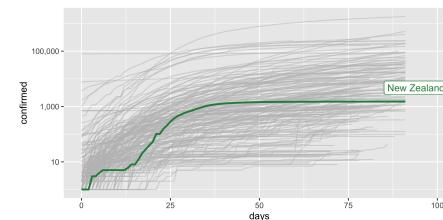
### COVID-19

- scale-y
- scale-x
- highlight
- annotate
- limits

### Expand limits

> Plot

> Code



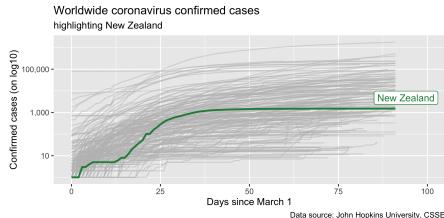
36 / 43

## COVID-19

- scale-y
- scale-x
- highlight
- annotate
- limits
- labels

### Every figure needs the title

[Plot](#) [Code](#)



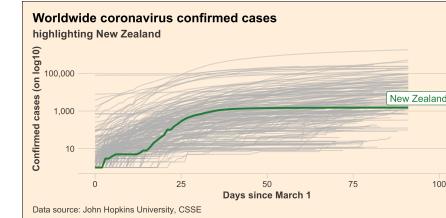
37 / 43

## COVID-19

- scale-y
- scale-x
- highlight
- annotate
- limits
- labels
- theme

### Apply themes

[Plot](#) [Code](#)



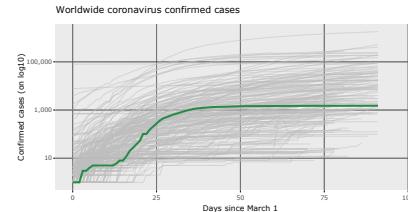
38 / 43

## Interactive graphics

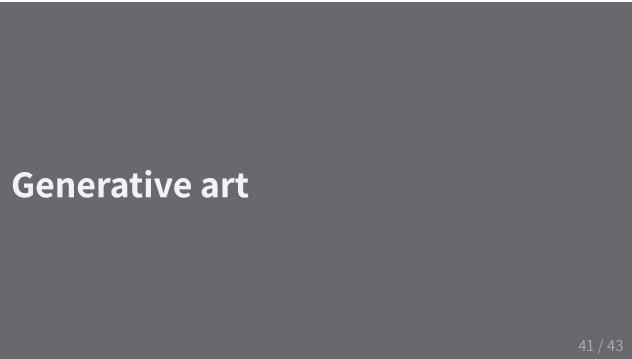
39 / 43

### Easily turn ggplot2 into plotly

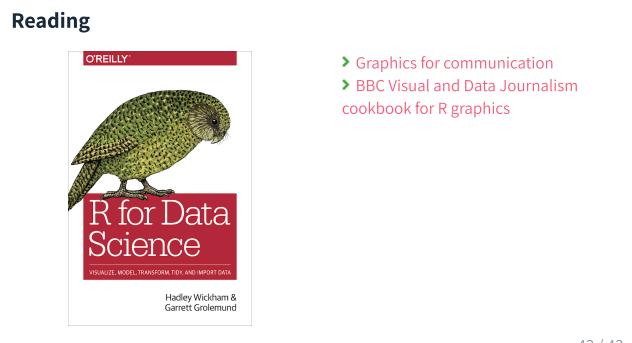
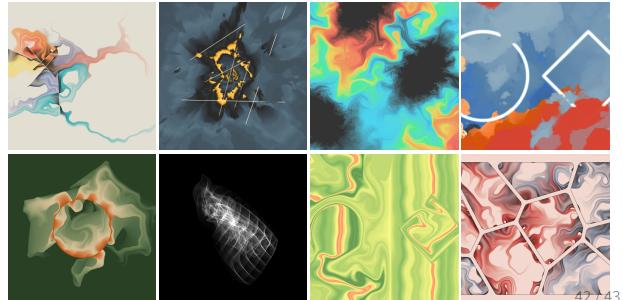
```
library(plotly)  
ggplotly(p_nz)
```



40 / 43



A peek at generative art works by Thomas Lin Pedersen



- Graphics for communication
- BBC Visual and Data Journalism cookbook for R graphics