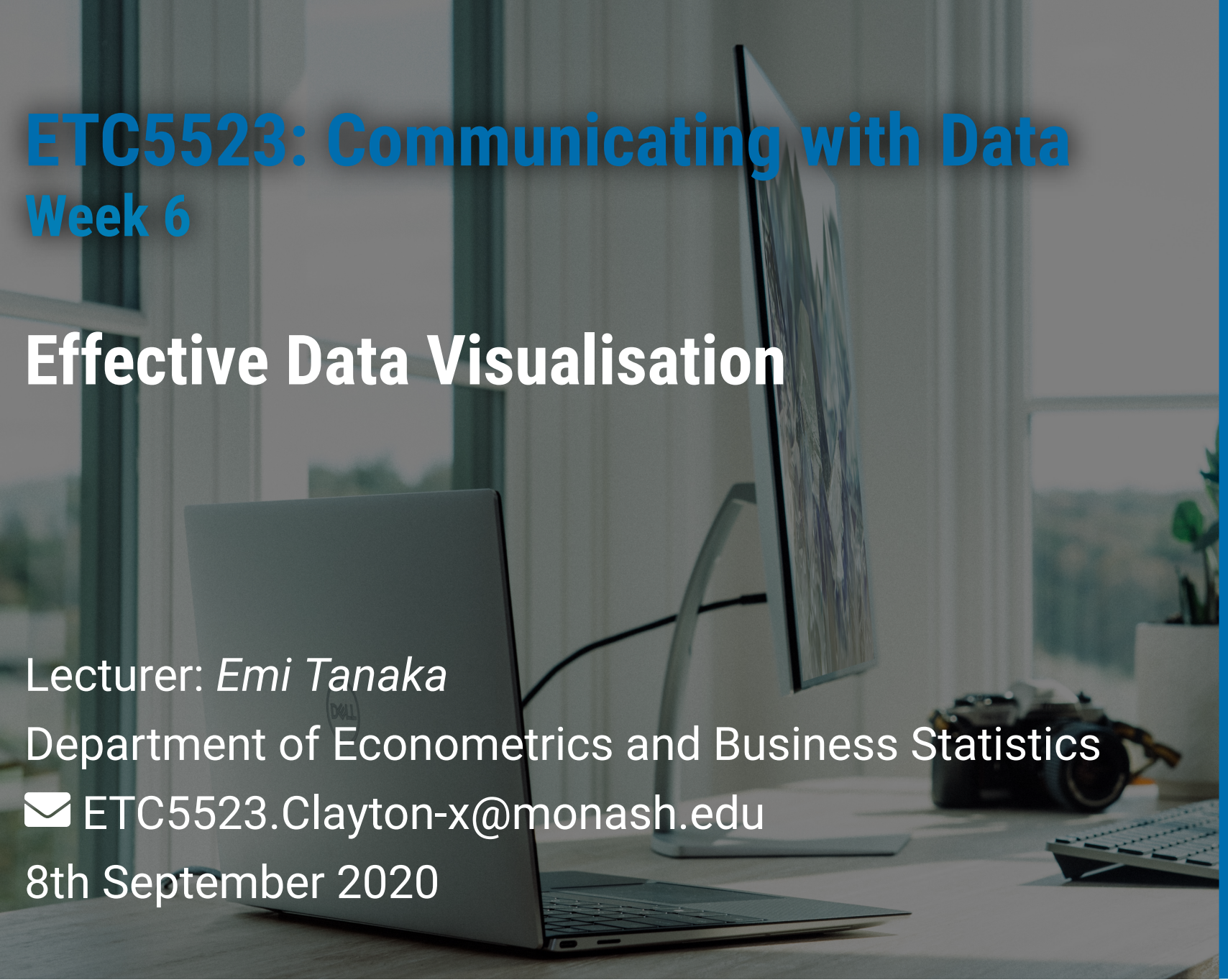


ETC5523: Communicating with Data

Week 6

Effective Data Visualisation

Lecturer: *Emi Tanaka*
Department of Econometrics and Business Statistics
ETC5523.Clayton-x@monash.edu
8th September 2020

Why make data plots?

> A picture is worth a thousand words.
>
> -Henrik Ibsen

- Data visualisation generally ***communicates information much quicker*** than numerical tables.
- Data visualisation can also ***reveal unexpected structures in data***; it is not surprising that data visualisation is one of the key tools in exploratory data analysis.
- Data plot is usually more ***eye-catching*** even if you lose accuracy of the information. And yes, right is still a data plot. It's called a waffle plot.

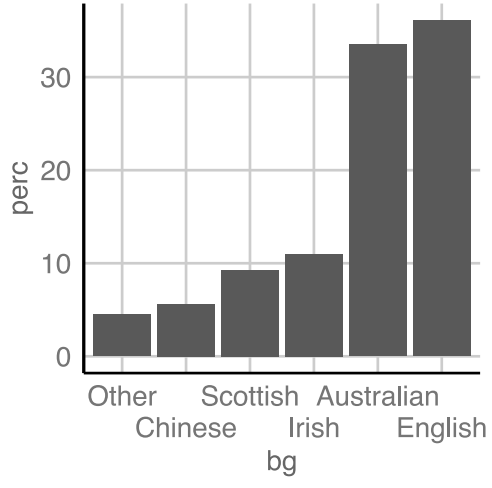
1 in 8 men are color blind.

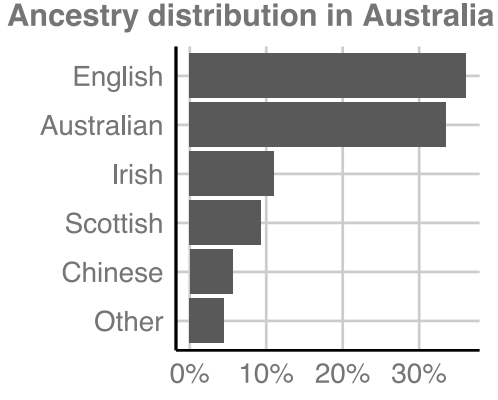
Communicating with data visualisation

- Effective data visualisation means to design your data plot to effectively use human visual system to improve cognition about a targeted information from the data.
- The ***design of your data plot*** depends on the ***usage plan*** is.
- Below is presenting percentage of the reported ancestories from the Australian Bureau of Statistics 2016 Census.

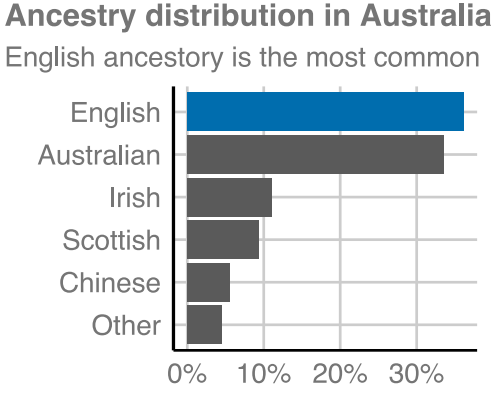
Table	Internal	Informative	Data storytelling

Ancestry	%
English	36.1
Australian	33.5
Irish	11.0
Scottish	9.3
Chinese	5.6
Other	4.5





Source: Australian Bureau of Statistics 2016 Census



Source: Australian Bureau of Statistics 2016 Census

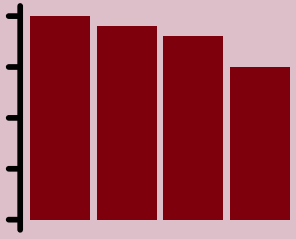
ℹ️ Is communication with data plot about infographics?

- ***Infographic*** is a collection of images, plots and text to present information on a topic in an easily understood manner.
- These days infographics are used for ***mass communication meant for the general public***, i.e. with assumption of little technical knowledge from the reader.
- These type of graphics may involve graphic designers, or at least special software for making infographics.
- ❗ Making sophisticated infographics like on the right will *not* be the focus of this lecture.

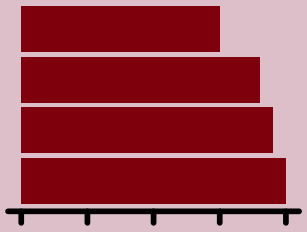
What type of plot should you use?

Data Visualisation Catalogue (non-exhaustive)

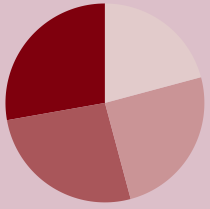
BARPLOT (HORIZONTAL)



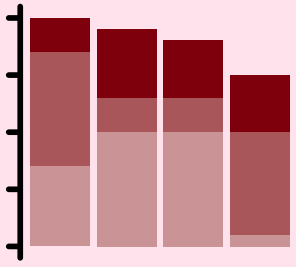
BARPLOT (VERTICAL)



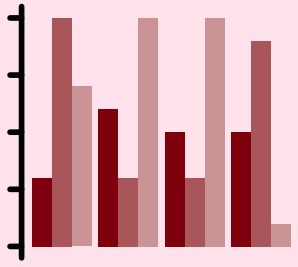
PIE CHART



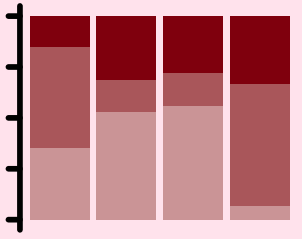
STACKED BARPLOT



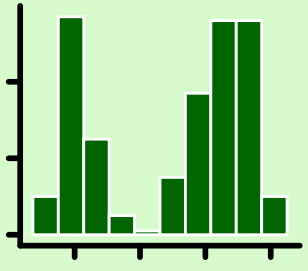
GROUPED BARPLOT



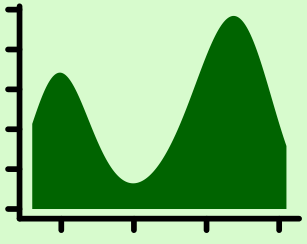
STACKED PERCENTAGE BARPLOT



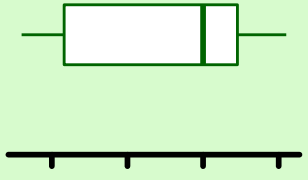
HISTOGRAM



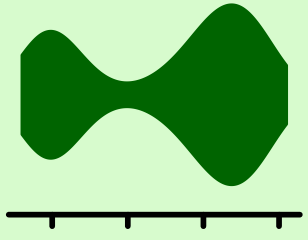
DENSITY PLOT (FILLED)



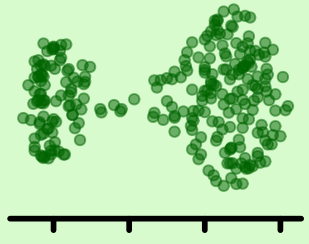
BOXPLOT



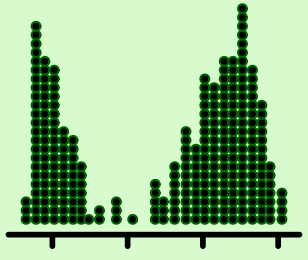
VIOLIN PLOT



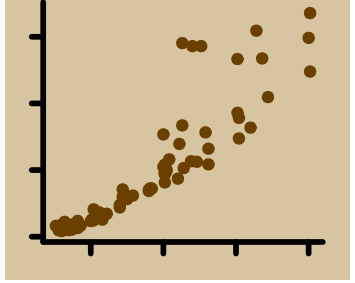
SINA PLOT



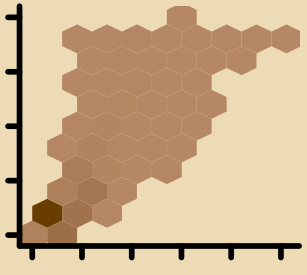
DOT PLOT



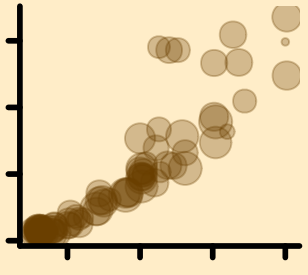
SCATTER PLOT



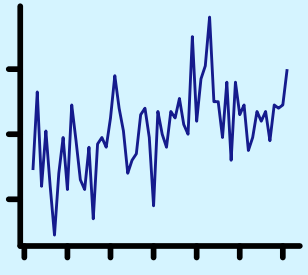
HEX PLOT



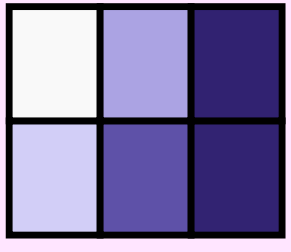
BUBBLE CHART



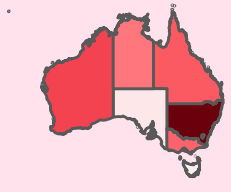
LINE PLOT



HEATMAP

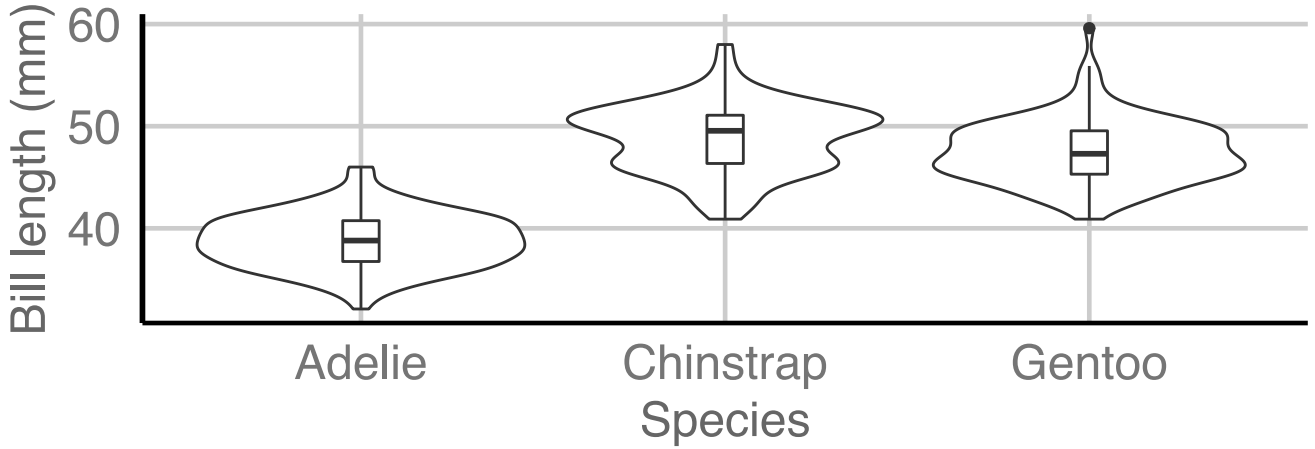


CHOROPLETH MAP

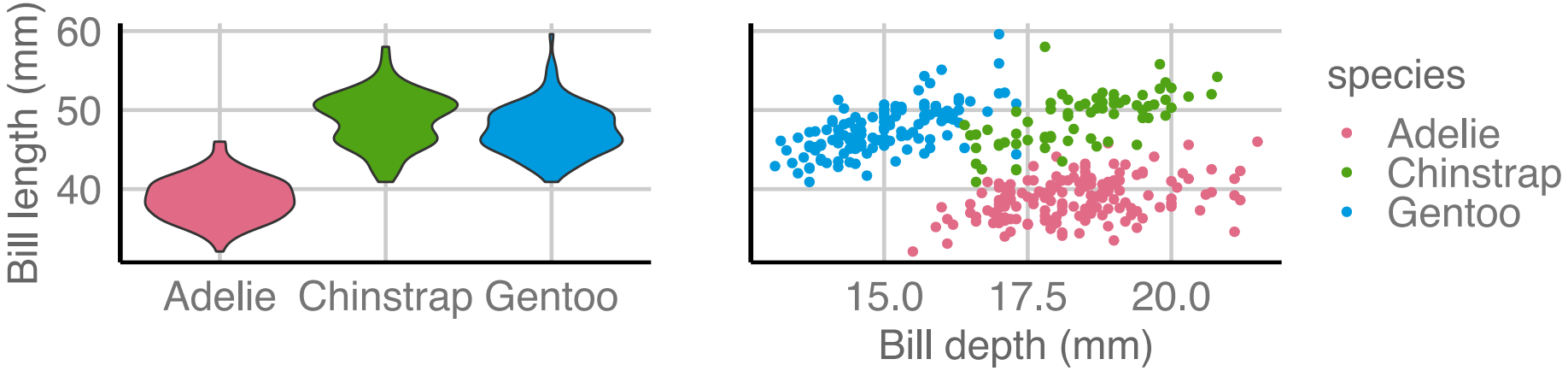


Composite plots

- Your data plot may be made from **multiple plot types**:



- Your data plot may be composed of **multiple subplots**:



How to make the data plots?

- There are many packages that make statistical packages in R.
- One of the most popular R-package is `ggplot2`.
- Why is it so popular?

The `ggplot` object

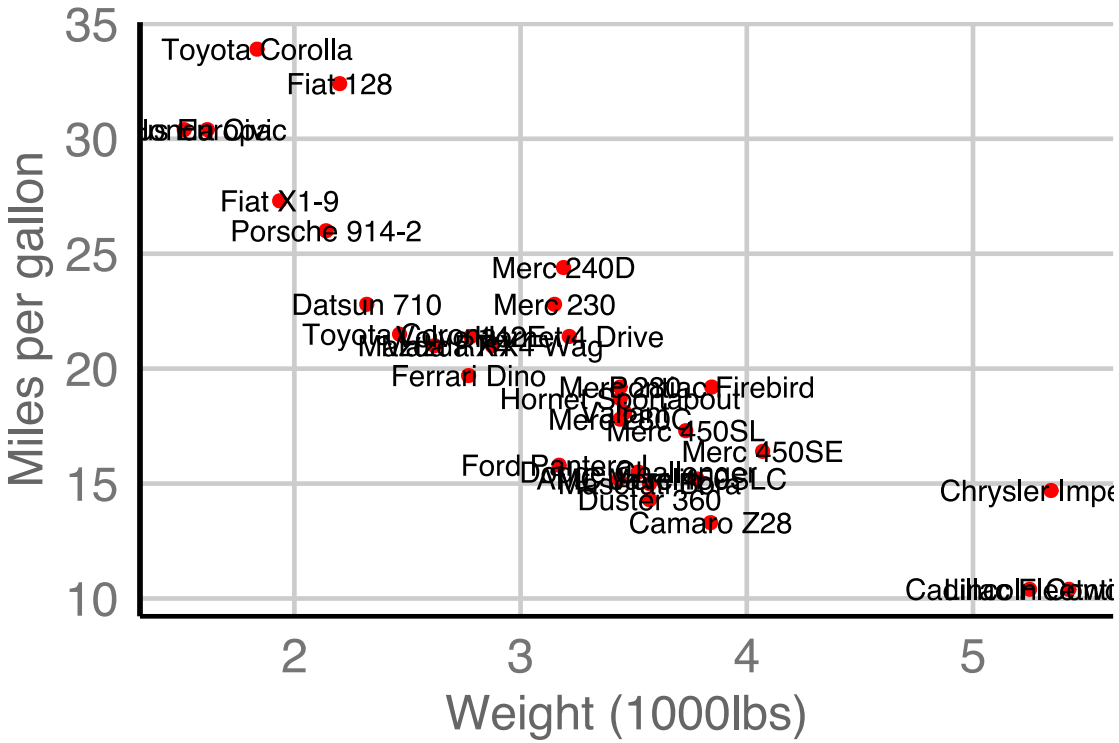
- **Modifiable**: `ggplot` object can be modified
- **Generalisable**: `ggplot2` uses a cohesive and complex system under the hood to make many kinds of plots
- **Extensible**: the system can be extended to make specialised plots or add more features if the same "grammar" is adopted

Most importantly, your graphical output is easily **reproducible**!

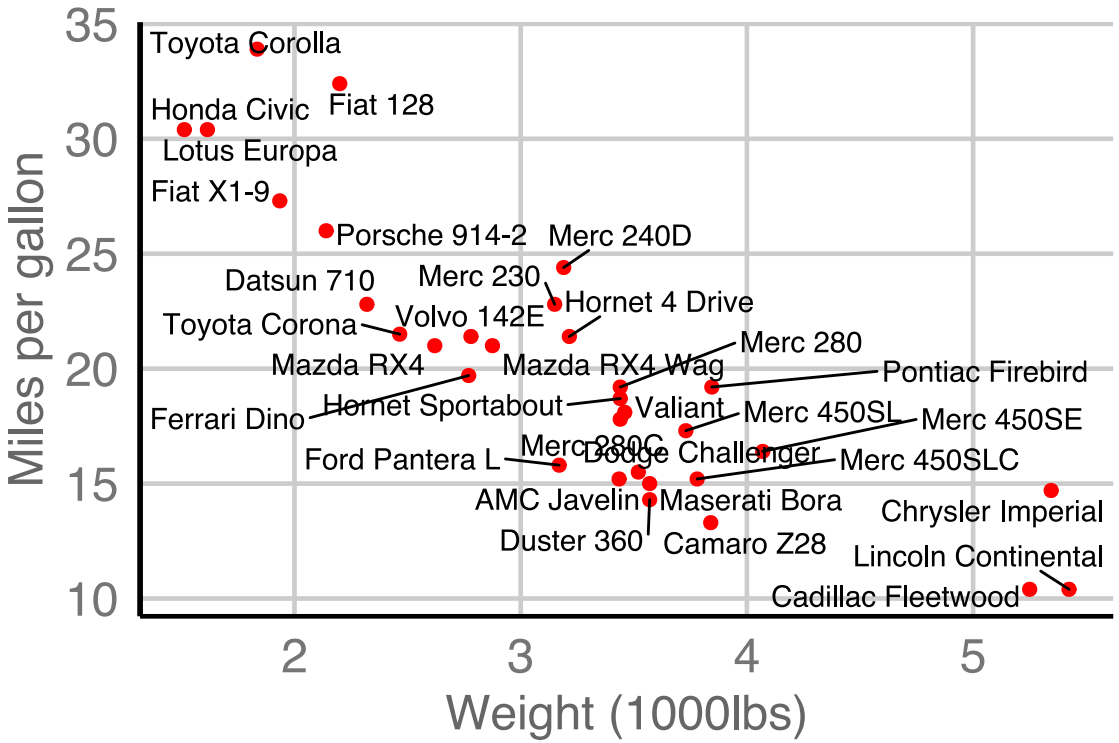
Imagine all the time spent manually making graphs (via photoshop or otherwise) only to repeat it again when data are updated. Not only are you less prone to error, you save time!

ggrepel  one of my favorites!

geom_text()



geom_text_repel()

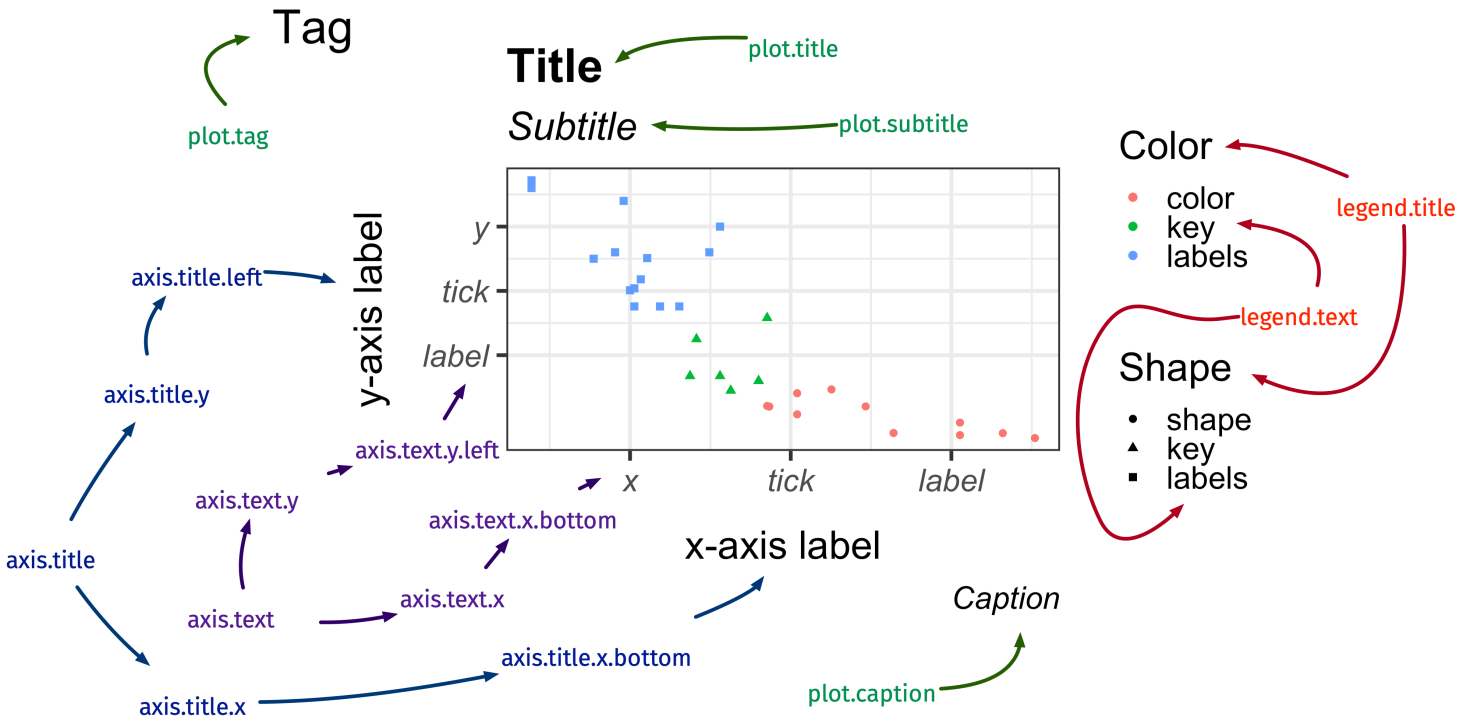


There are many extension packages!

Kamil Slowikowski (2020). ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'. R package version 0.8.2. https://CRAN.R-project.org/package=ggrepel

Some tips about `theme` in `ggplot2`

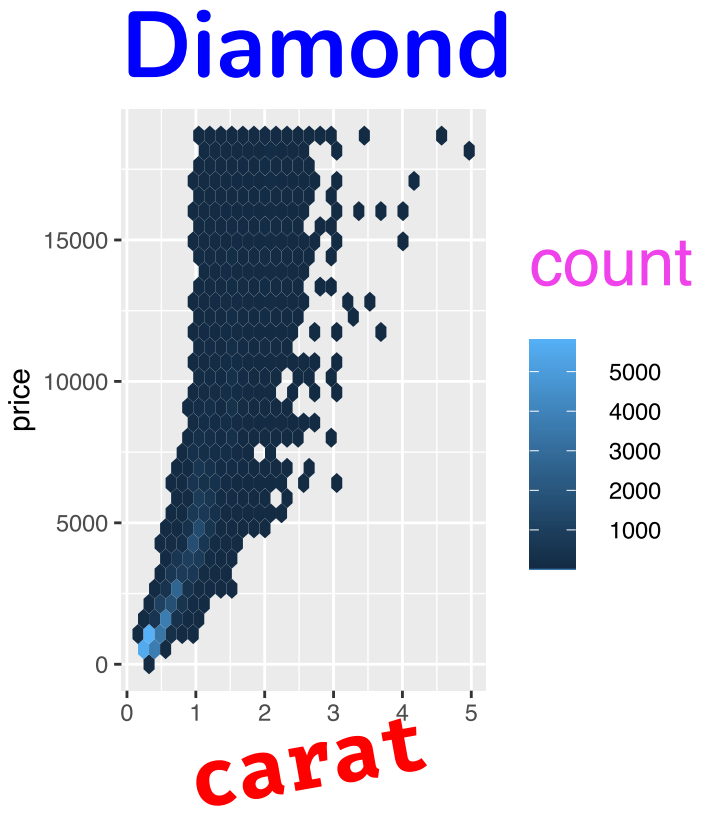
theme: modify the *look* of texts



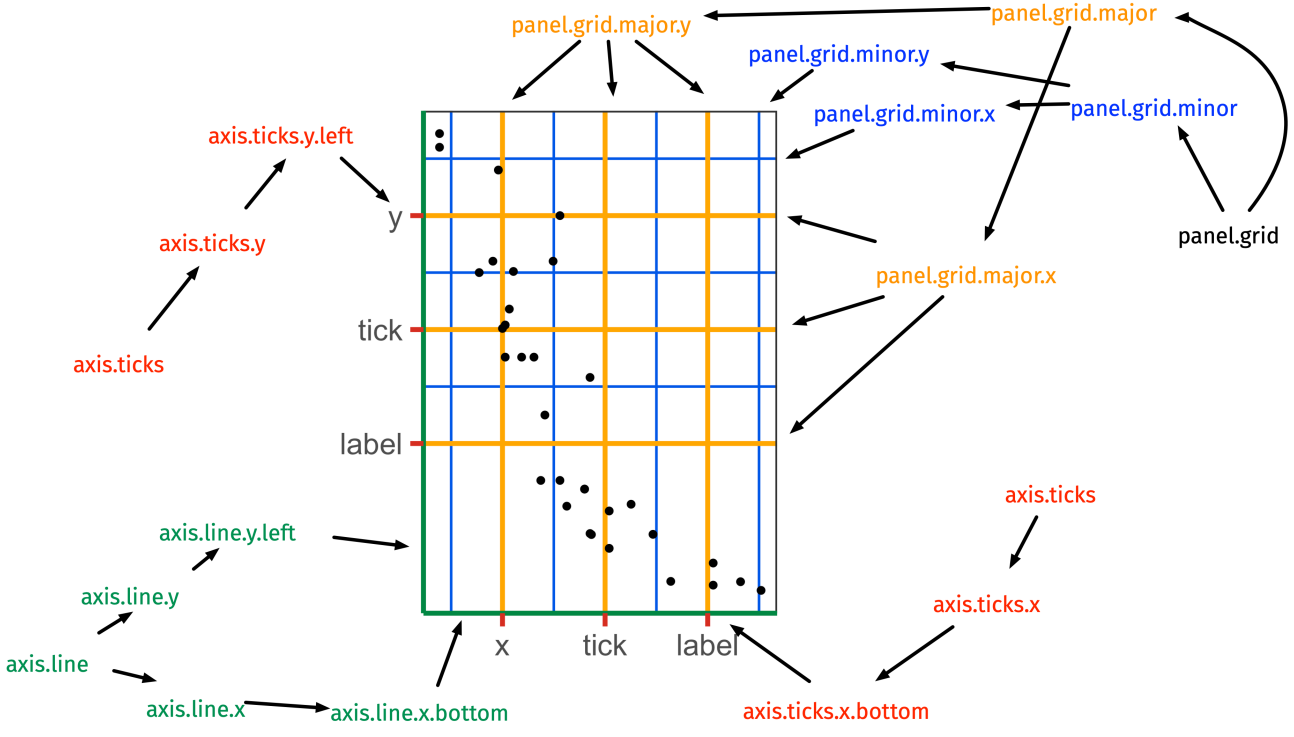
`element_text()`

element_text()

```
ggplot(diamonds, aes(carat, price)) +
  geom_hex() +
 labs(title = "Diamond") +
 theme(axis.title.x = element_text(size = 30,
                                   color = "red",
                                   face = "bold",
                                   angle = 10,
                                   family = "Fira Code"),
       legend.title = element_text(size = 25,
                                   color = "#ef42eb",
                                   margin = margin(b = 5)),
       plot.title = element_text(size = 35,
                                   face = "bold",
                                   family = "Nunito",
                                   color = "blue"
                                  ))
```



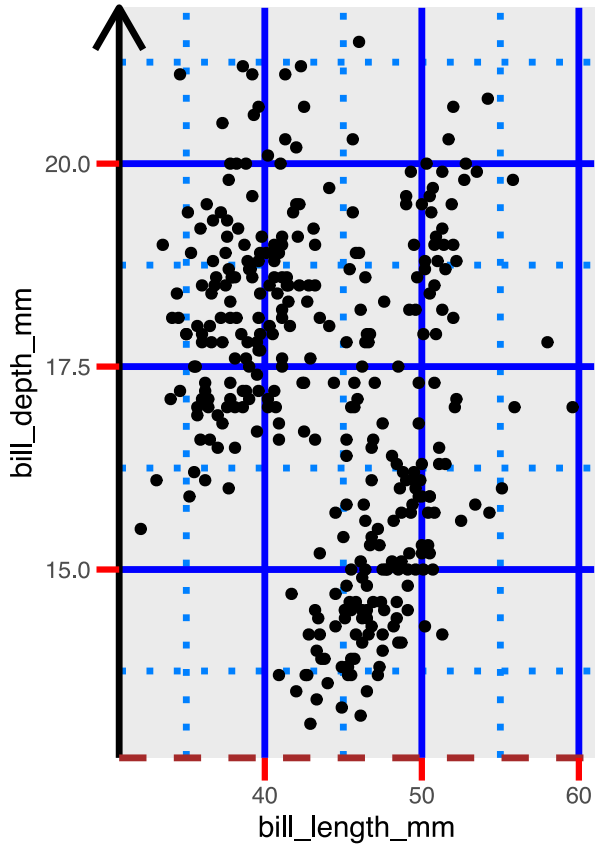
`theme`: modify the *look* of the lines



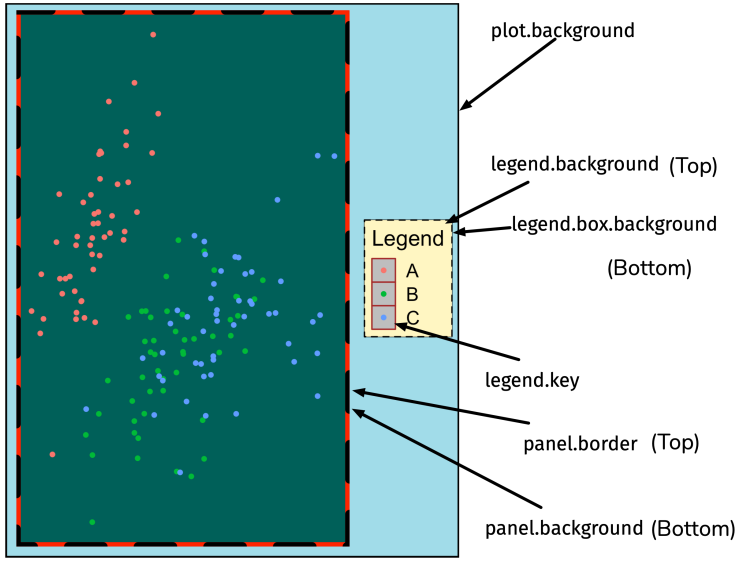
`element_line()`

element_line()

```
ggplot(penguins, aes(bill_length_mm, bill_depth_mm)) +
  geom_point() +
  theme(axis.line.y = element_line(color = "black",
                                   size = 1.2,
                                   arrow = grid::arrow()),
        axis.line.x = element_line(linetype = "dashed",
                                   color = "brown",
                                   size = 1.2),
        axis.ticks = element_line(color = "red", size = 1.1),
        axis.ticks.length = unit(3, "mm"),
        panel.grid.major = element_line(color = "blue",
                                        size = 1.2),
        panel.grid.minor = element_line(color = "#0080ff",
                                        size = 1.2,
                                        linetype = "dotted"))
```



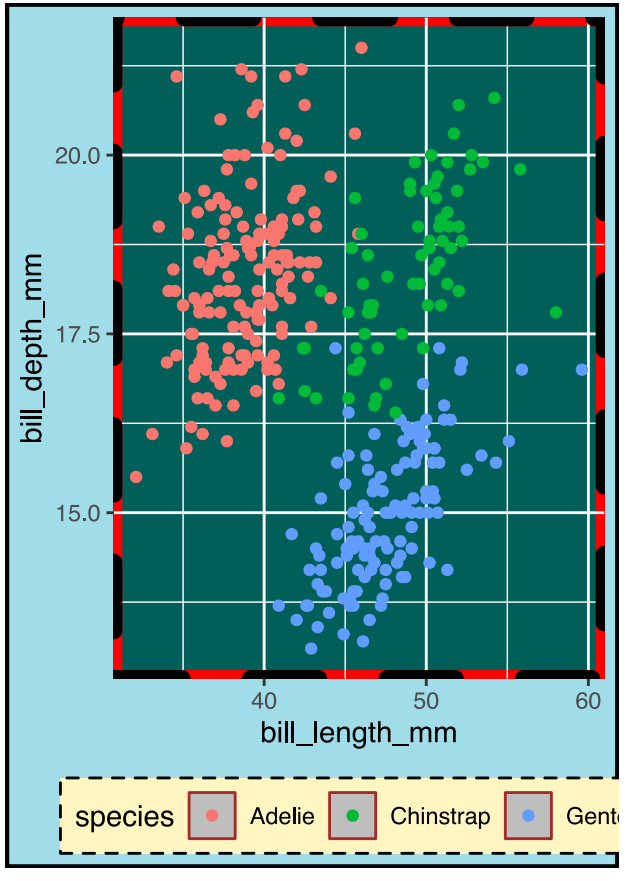
`theme`: modify the *look* of the rectangular regions



`element_rect()`

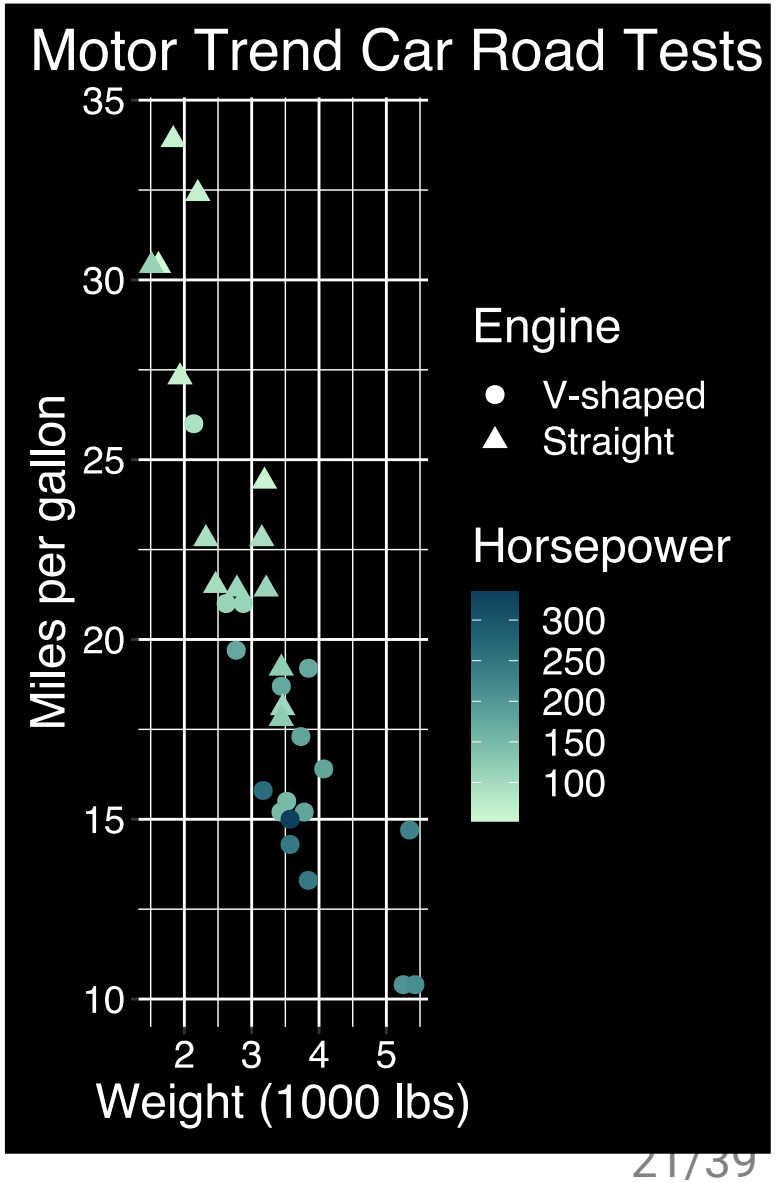
element_rect()

```
ggplot(penguins, aes(bill_length_mm, bill_depth_mm)) +
 geom_point(aes(color = species)) +
 theme(
  legend.background = element_rect(fill = "#fff6c2",
                                   color = "black",
                                  linetype = "dashed"),
  legend.key = element_rect(fill = "grey", color = "brown"),
  panel.background = element_rect(fill = "#005F59",
                                  color = "red", size = 3),
  panel.border = element_rect(color = "black",
                              fill = "transparent",
                     linetype = "dashed", size = 3),
  plot.background = element_rect(fill = "#a1dce9",
                                 color = "black",
                                  size = 1.3),
  legend.position = "bottom")
```



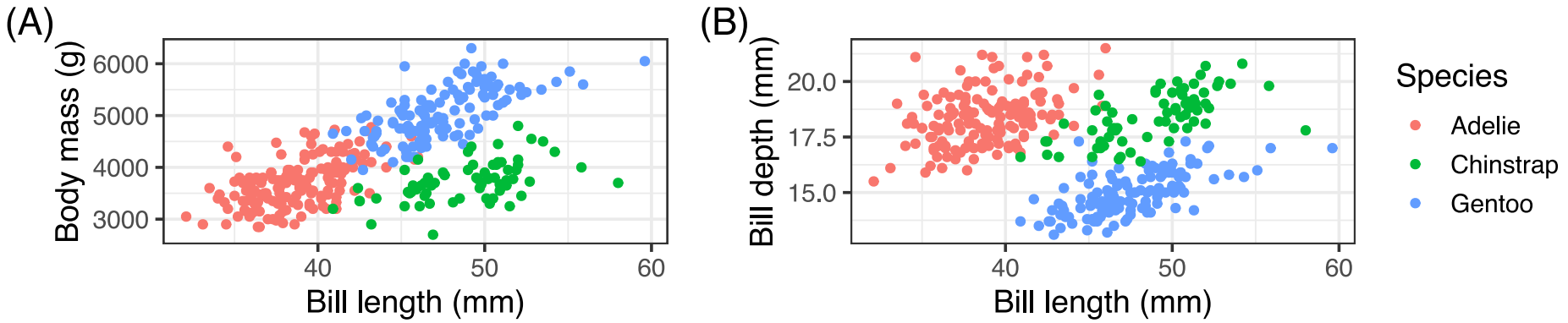
Reproducible publication ready plots

```
ggplot(mtcars_df,
       aes(wt, mpg, shape = factor(vs), color = hp)) +
 geom_point(size = 3) +
 scale_color_continuous_sequential(palette = "Dark Mint") +
 scale_shape_discrete(labels = c("V-shaped", "Straight")) +
 labs(x = "Weight (1000 lbs)", y = "Miles per gallon",
      title = "Motor Trend Car Road Tests",
      shape = "Engine", color = "Horsepower") +
 theme(text = element_text(size = 18, color = "white"),
       rect = element_rect(fill = "black"),
       panel.background = element_rect(fill = "black"),
       legend.key = element_rect(fill = "black"),
       axis.text = element_text(color = "white"),
       plot.title.position = "plot",
       plot.margin = margin(10, 10, 10, 10)) +
 guides(shape =
      guide_legend(override.aes = list(color = "white")))
```



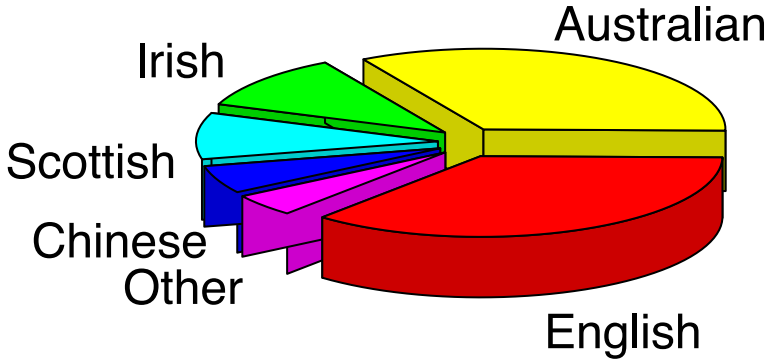
Combining plots with patchwork 📦

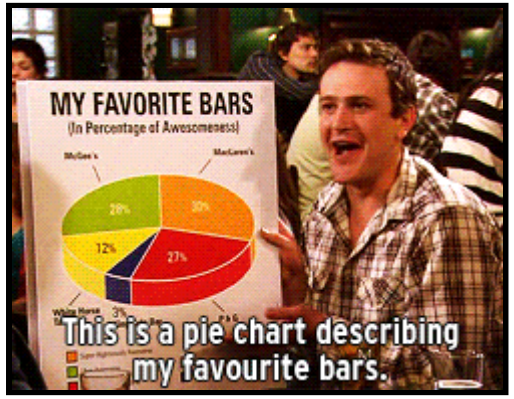
```
library(patchwork)
g1 <- ggplot(penguins, aes(bill_length_mm, body_mass_g, color = species))
  geom_point() + theme_bw(base_size = 14) +
  labs(tag = "(A)", x = "Bill length (mm)", y = "Body mass (g)", color =
g2 <- ggplot(penguins, aes(bill_length_mm, bill_depth_mm, color = species
  geom_point() + theme_bw(base_size = 14) +
  labs(tag = "(B)", x = "Bill length (mm)", y = "Bill depth (mm)", color =
g1 + g2 + plot_layout(guides = "collect")
```



Thomas Lin Pedersen (2020). patchwork: The Composer of Plots. R package version 1.0.1. https://CRAN.R-project.org/package=patchwork

Why is 3D pie chart considered a "bad plot"?



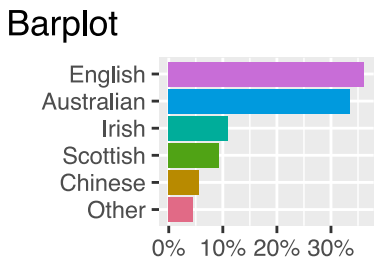


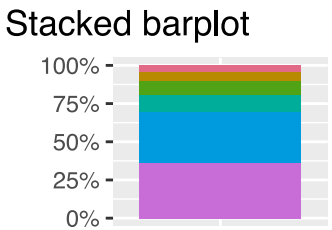
What about 2D pie charts?

```
help("pie")
```

> Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data.

- This comes from empirical research of Cleveland & McGill (1984) among others.





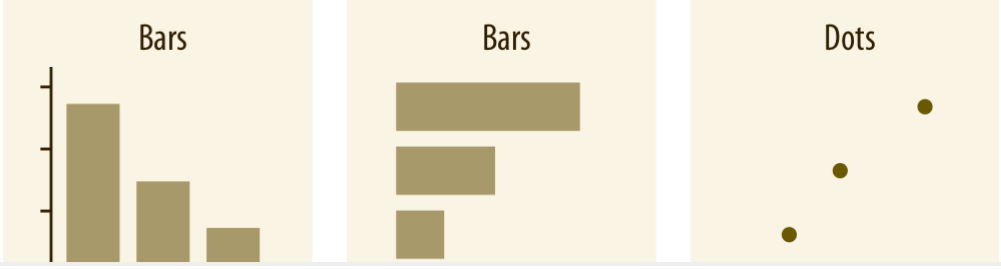
Cleveland, William S., and Robert Mc Gill. (1984) “Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods.” https://www.jstor.org/stable/pdf/2288400.pdf.

https://clauswilke.com/dataviz/directory-of-visualizations.html

5 Directory of visualizations

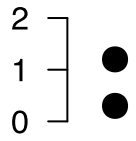
This chapter provides a quick visual overview of the various plots and charts that are commonly used to visualize data. It is meant both to serve as a table of contents, in case you are looking for a particular visualization whose name you may not know, and as a source of inspiration, if you need to find alternatives to the figures you routinely make.

5.1 Amounts

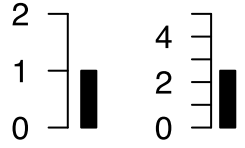


Elementary Perceptual Tasks (non-exhaustive)

POSITION
COMMON SCALE



POSITION
NON-ALIGNED SCALE

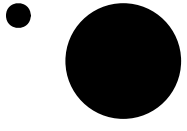


LENGTH

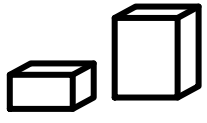
DIRECTION

ANGLE

AREA

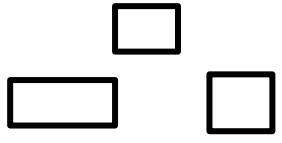


VOLUME

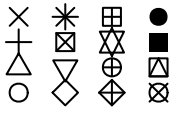


CURVATURE

TEXTURE

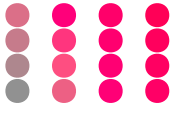


SHAPE



COLOR HUE

COLOR CHROMA



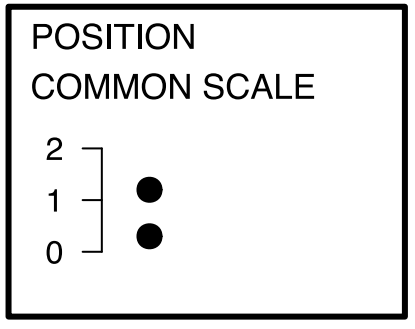
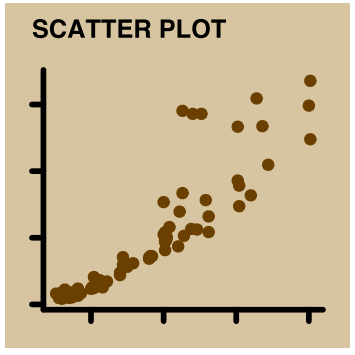
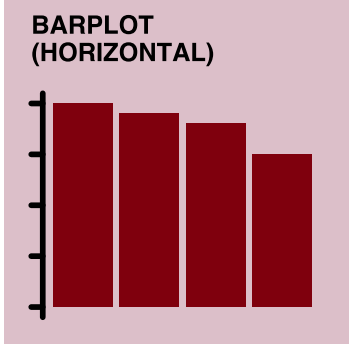
COLOR LUMINANCE

COLOR SATURATION

COLOR SHADE

Retrieving information from graphs

Of the 10 elementary perception tasks, Cleveland & McGill (1984) found the accuracy ranked as followed:

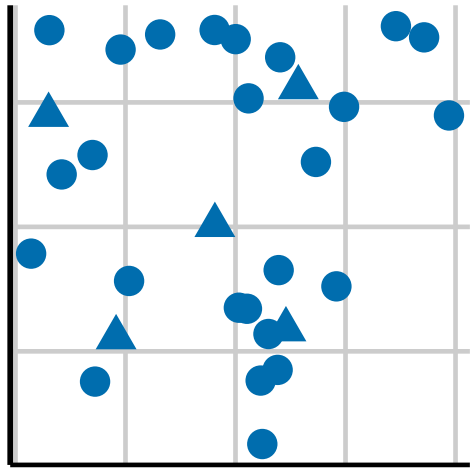
Accuracy Order (from most to least)	Elementary perception tasks	Example plots (plots can have multiple elementary perception tasks)
1		 

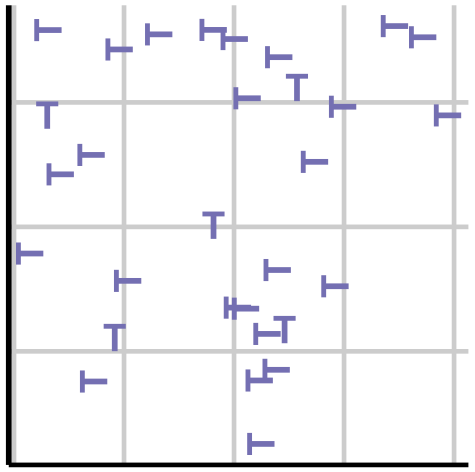
scroll

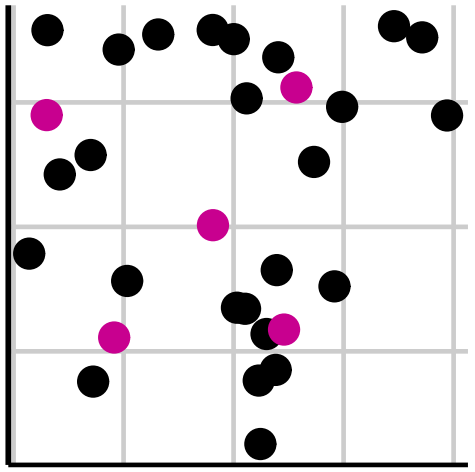
This and next few slides inspired by data visualisation workshop with Di Cook ✨

Preattentive processing

- Viewers can notice certain features are absent or present without focussing their attention on particular regions.
- Which plot helps you to distinguish the data points?



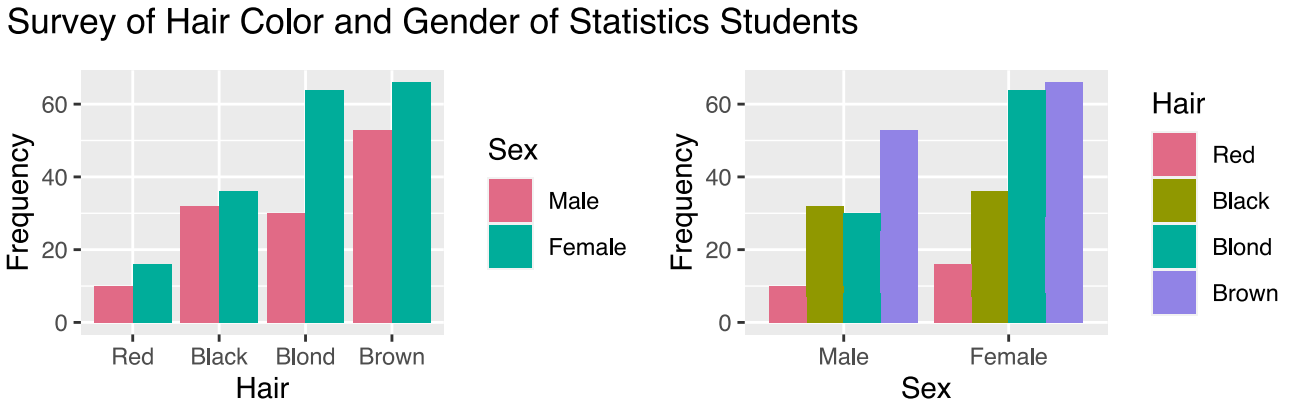




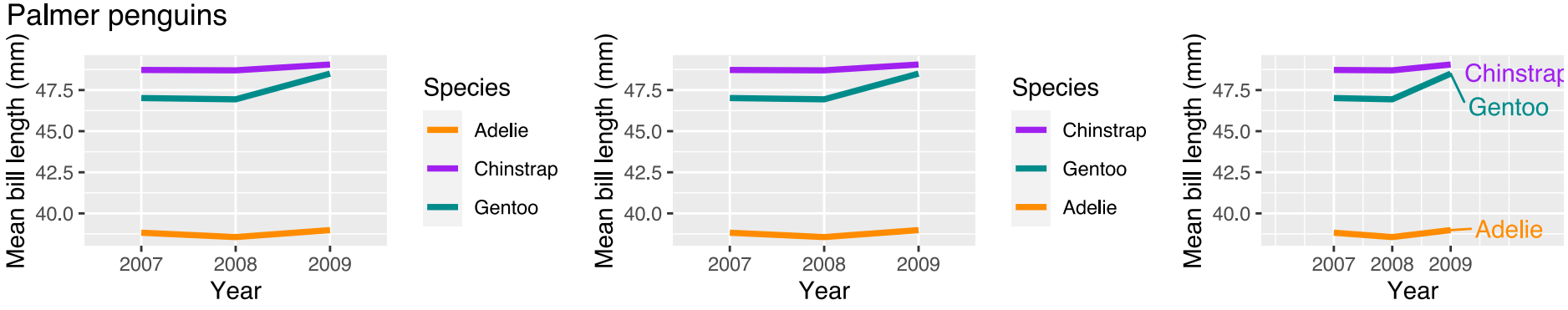
Healey, Christopher G., and James T. Enns (2012) “Attention and Visual Memory in Visualization and Computer Graphics.” IEEE Transactions on Visualization and Computer Graphics 18 (7): 1170–88.

Proximity

- By placing elements closer together, it makes it easier for you to group elements together as well as make comparisons.
- Which plot is better to compare gender frequency by hair color?



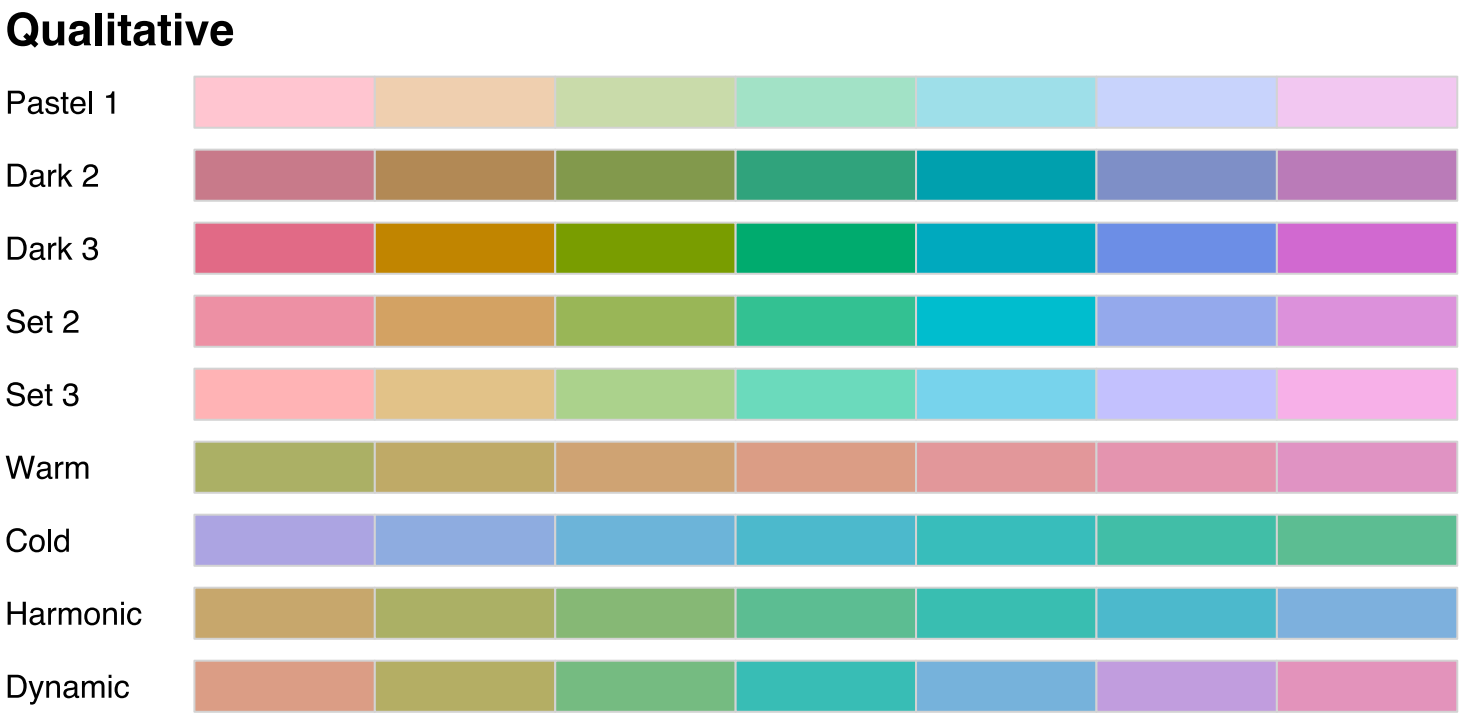
- Which plot is easier to understand the color labels?



Qualitative palettes

designed for categorical variable with no particular ordering

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



Sequential palettes

designed for ordered categorical variable or number going from low to high (or vice-versa)

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

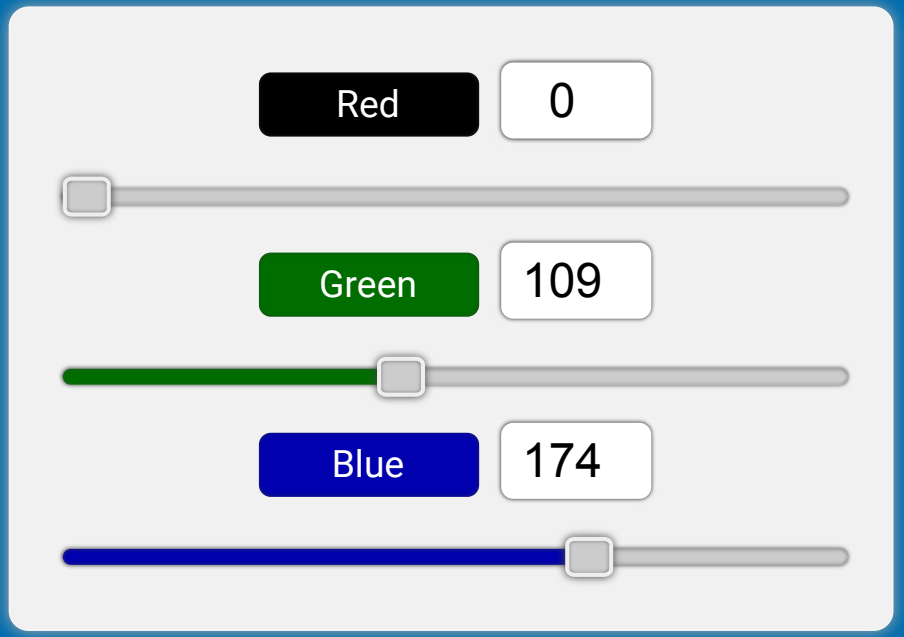
Diverging palettes

designed for ordered categorical variable or number going from low to high (or vice-versa) with a neutral value in between

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

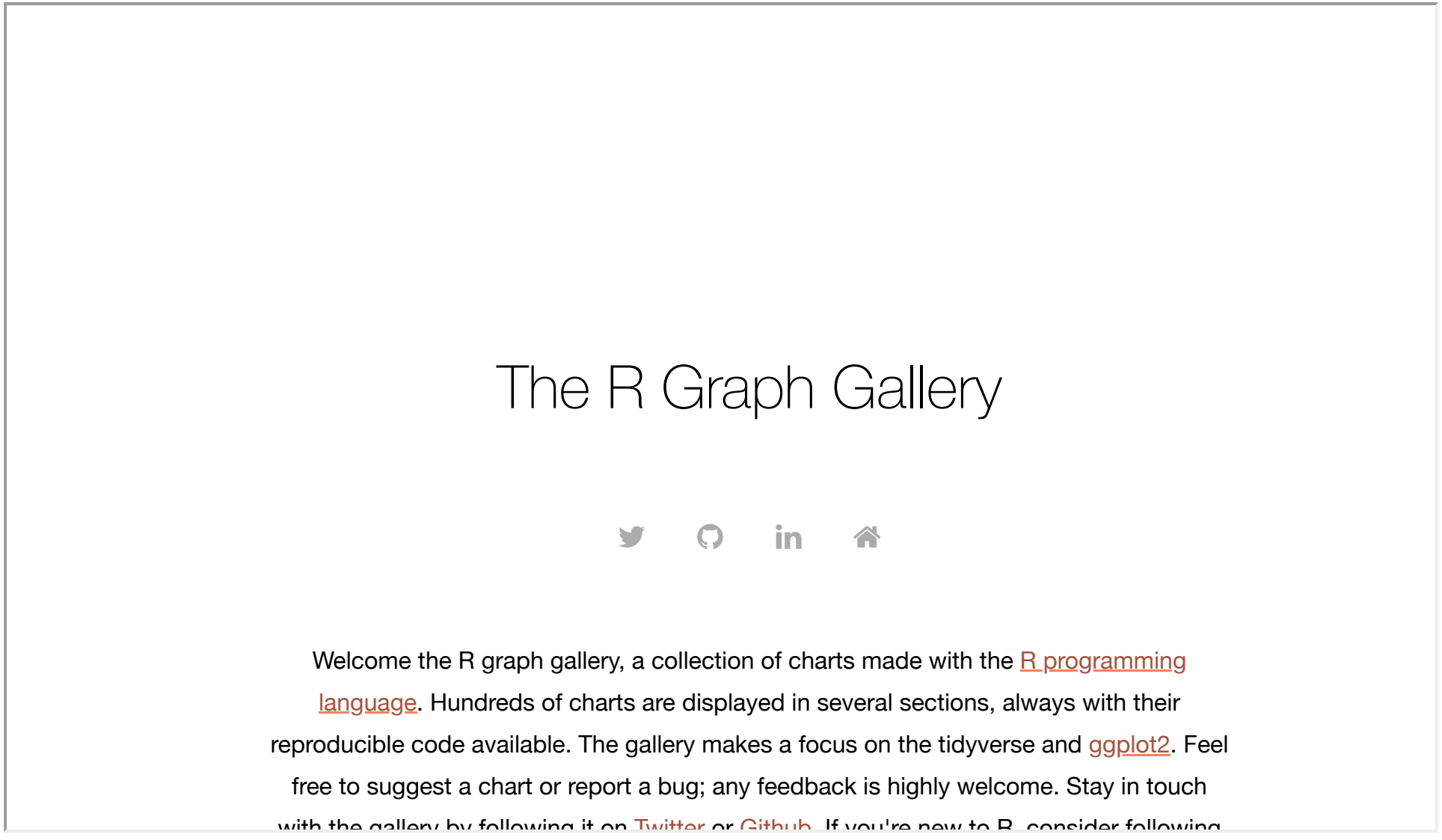
RGB color space

made for screen projection



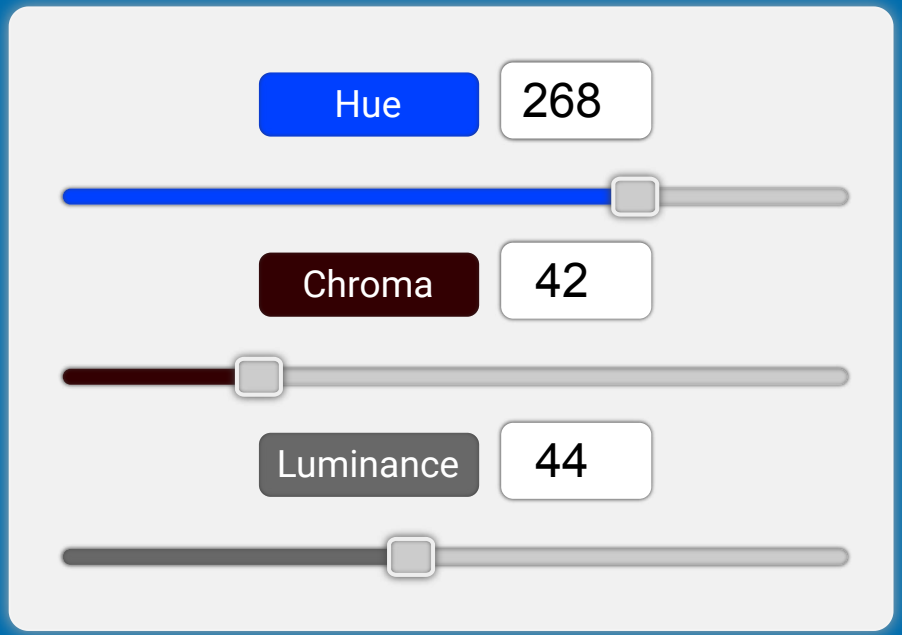
Code adapted from https://github.com/Golobro/rgbcolorslider

https://www.r-graph-gallery.com/



HCL color space

made for human visual system

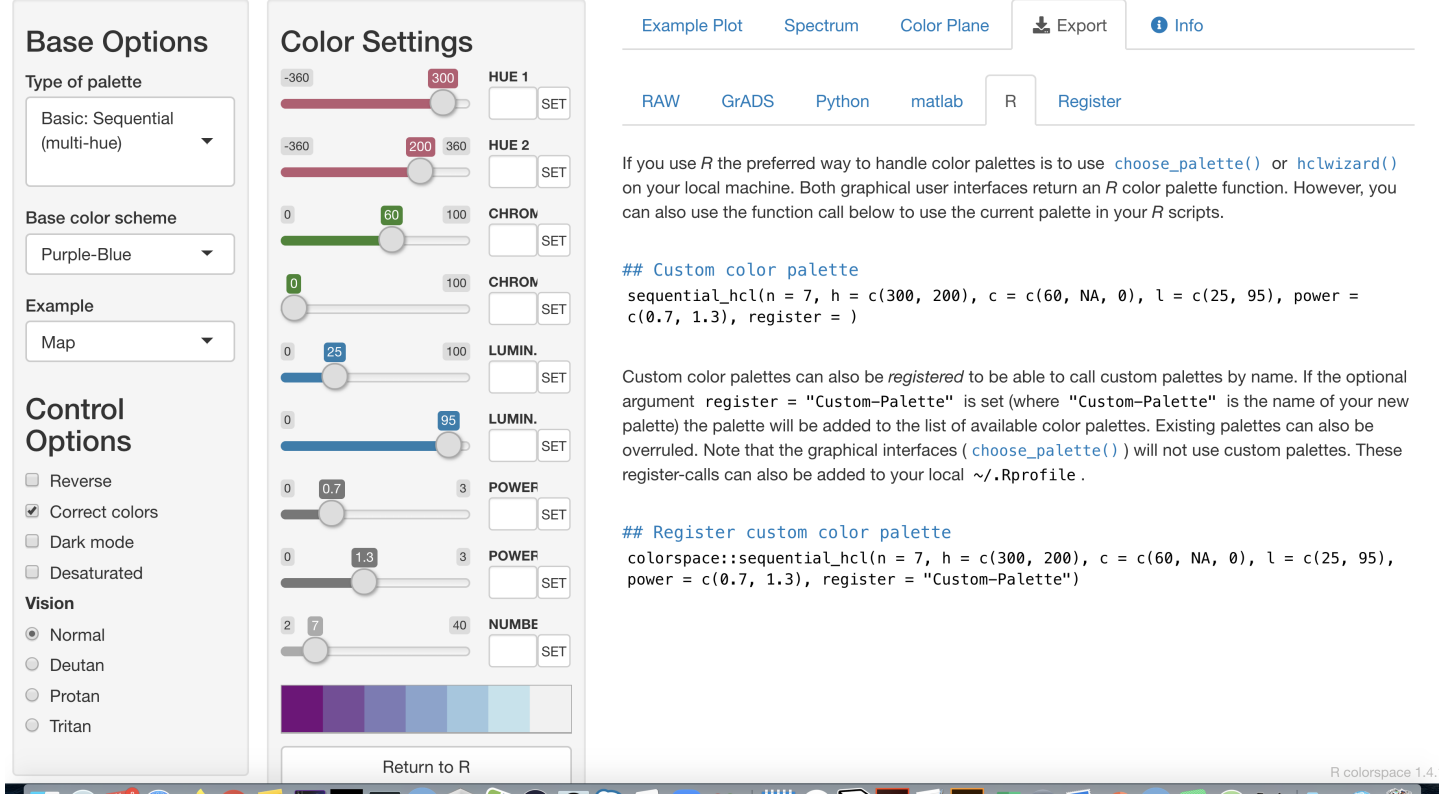


Color conversion using https://github.com/gka/chroma.js

colorspace LIVE DEMO

Interactively choose/create a palette using the HCL color space.

```
library(colorspace)
hcl_wizard() # OR choose_palette()
```



Choose your palette > Export > R > Copy the command

Registering your own palette

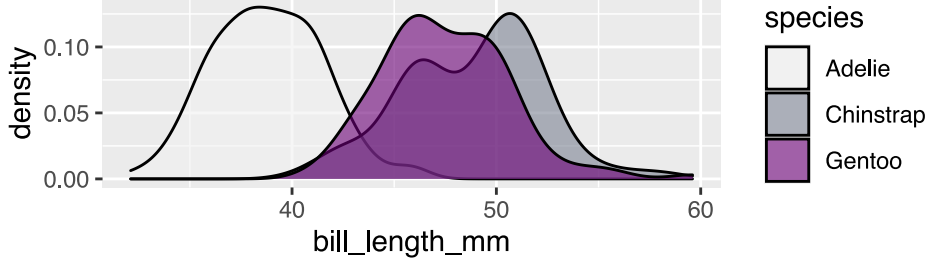
```
library(colorspace)
# register your palette
sequential_hcl(n = 7,
               h = c(300, 200),
               c = c(60, 0),
               l = c(25, 95),
               power = c(2.1, 0.8),
               register = "my-set")
# now generate from your palette
sequential_hcl(n = 3,
               palette = "my-set")

## [1] "#6B0077" "#7C8393" "#F1F1F1"
```

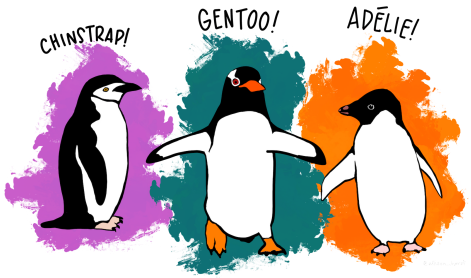
```
hcl_palettes(n = 5, palette = "my-set", plot = T)
```

Combining with ggplot:

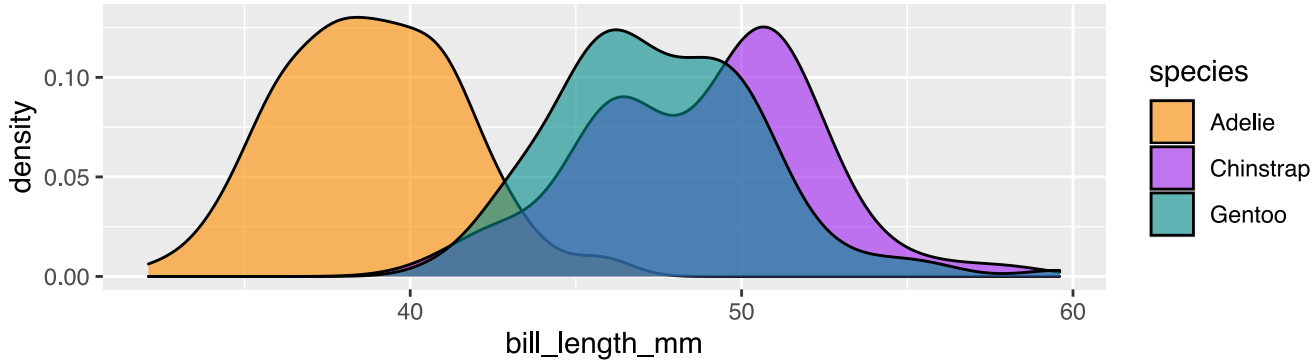
```
ggplot(penguins,
       aes(bill_length_mm, fill = species)) +
 geom_density(alpha = 0.6) +
  # notice here you don't need to specify the n!
 scale_fill_discrete_sequential(palette = "my-set")
```



Match your color with the story




```
g <- ggplot(penguins, aes(bill_length_mm, fill = species)) +
  geom_density(alpha = 0.6) +
  scale_fill_manual(
   breaks = c("Adelie", "Chinstrap", "Gentoo"), # optional but makes it mo
   values = c("darkorange", "purple", "cyan4"))
g
```

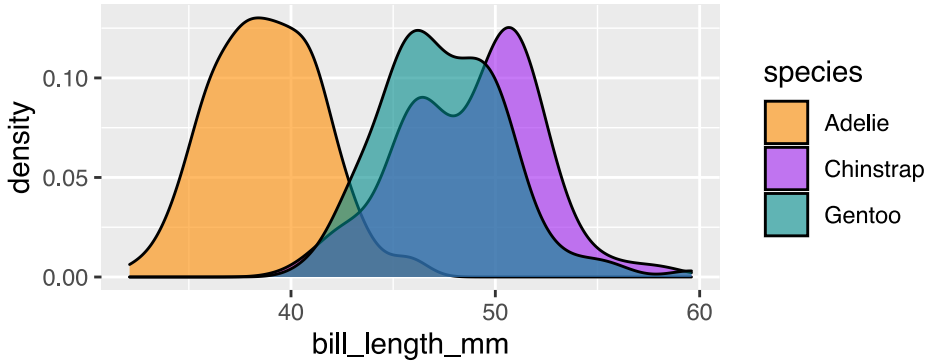


Art by Allison Horst.

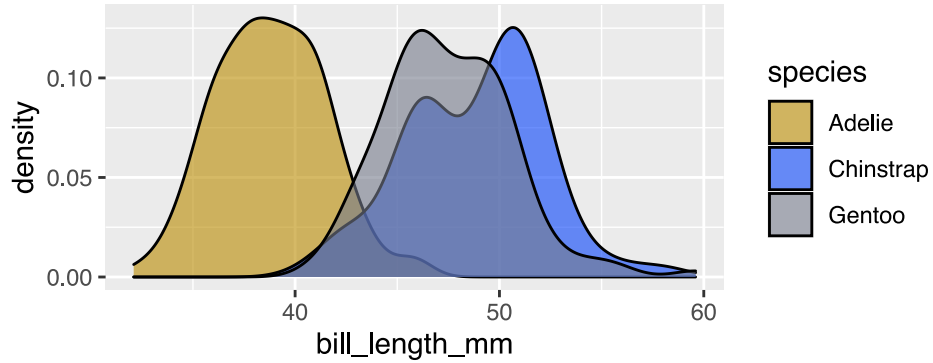
Check that it's colour blind friendly!

```
cols <- c("darkorange", "purple", "cyan4")
```

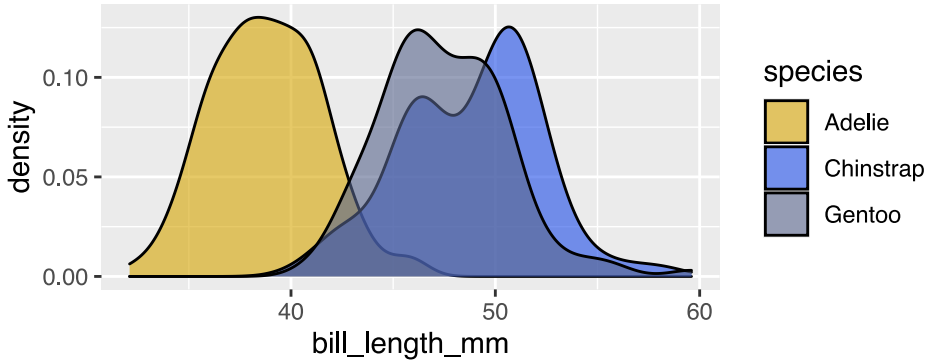
```
g # original
```



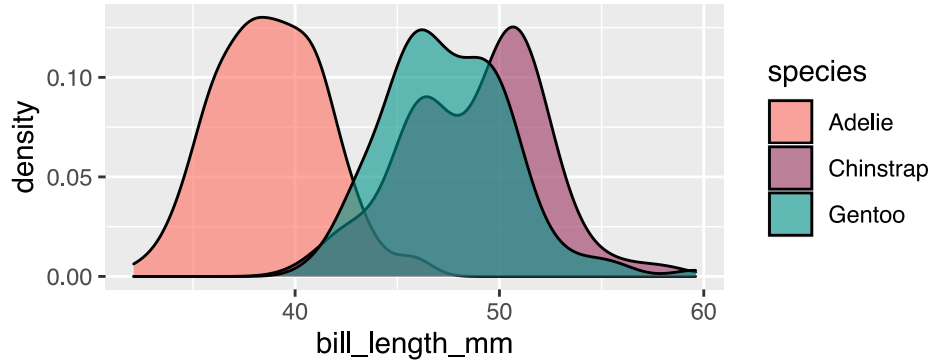

```
g + scale_fill_manual(values = protan(cols))
```




```
g + scale_fill_manual(values = deutan(cols))
```




```
g + scale_fill_manual(values = tritan(cols))
```



That's it!

Lecturer: Emi Tanaka

Department of Econometrics and Business Statistics

ETC5523.Clayton-x@monash.edu

https://www.data-to-viz.com/