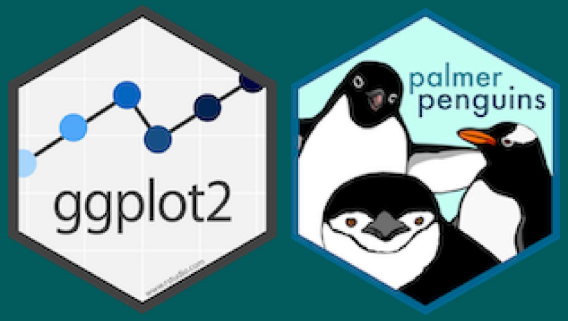
LT2010 - Data Science for Transport and Logistics Systems

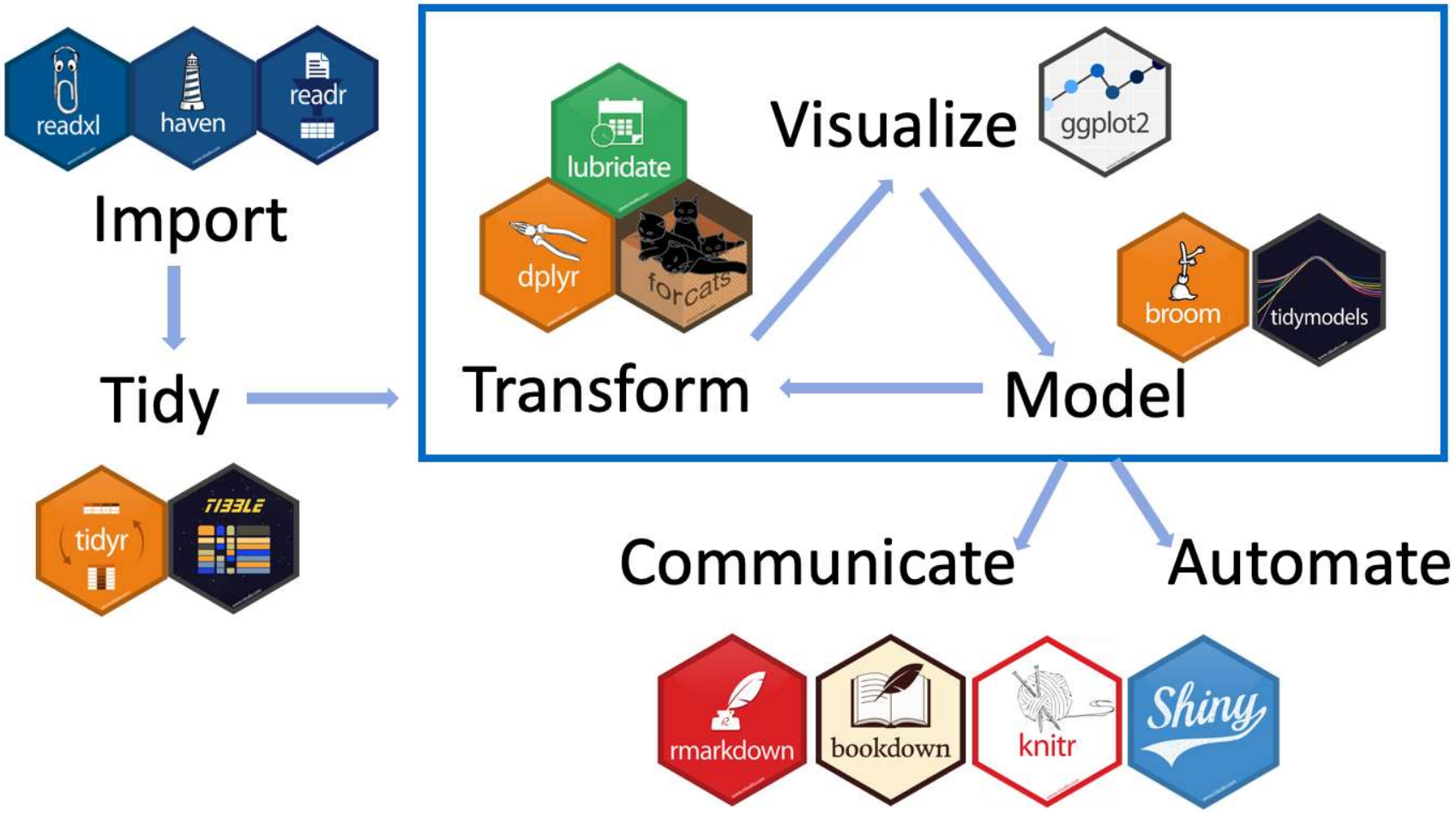
Data visualization with R

Dr. Priyanga D. Talagala

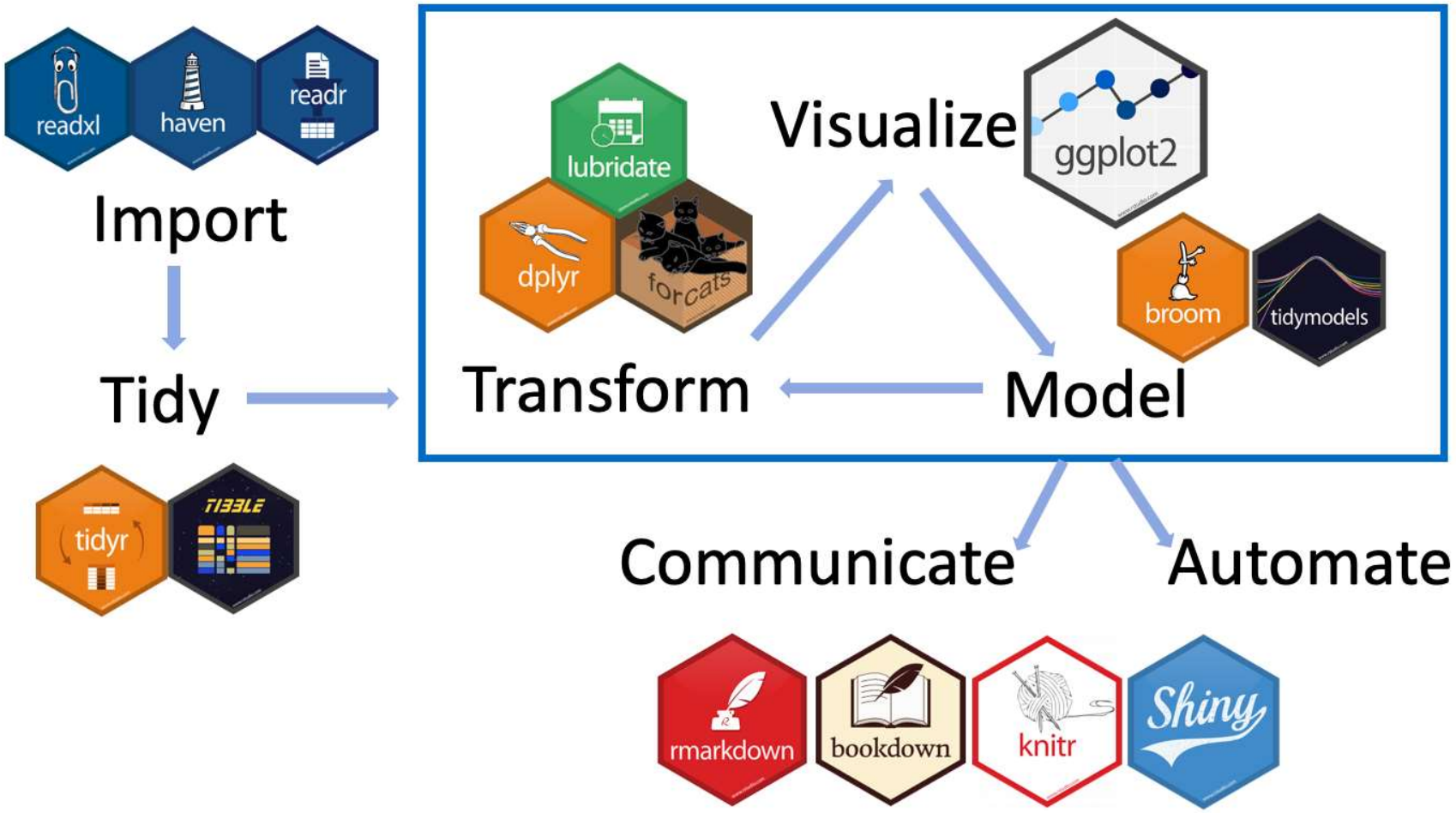
25/10/2023



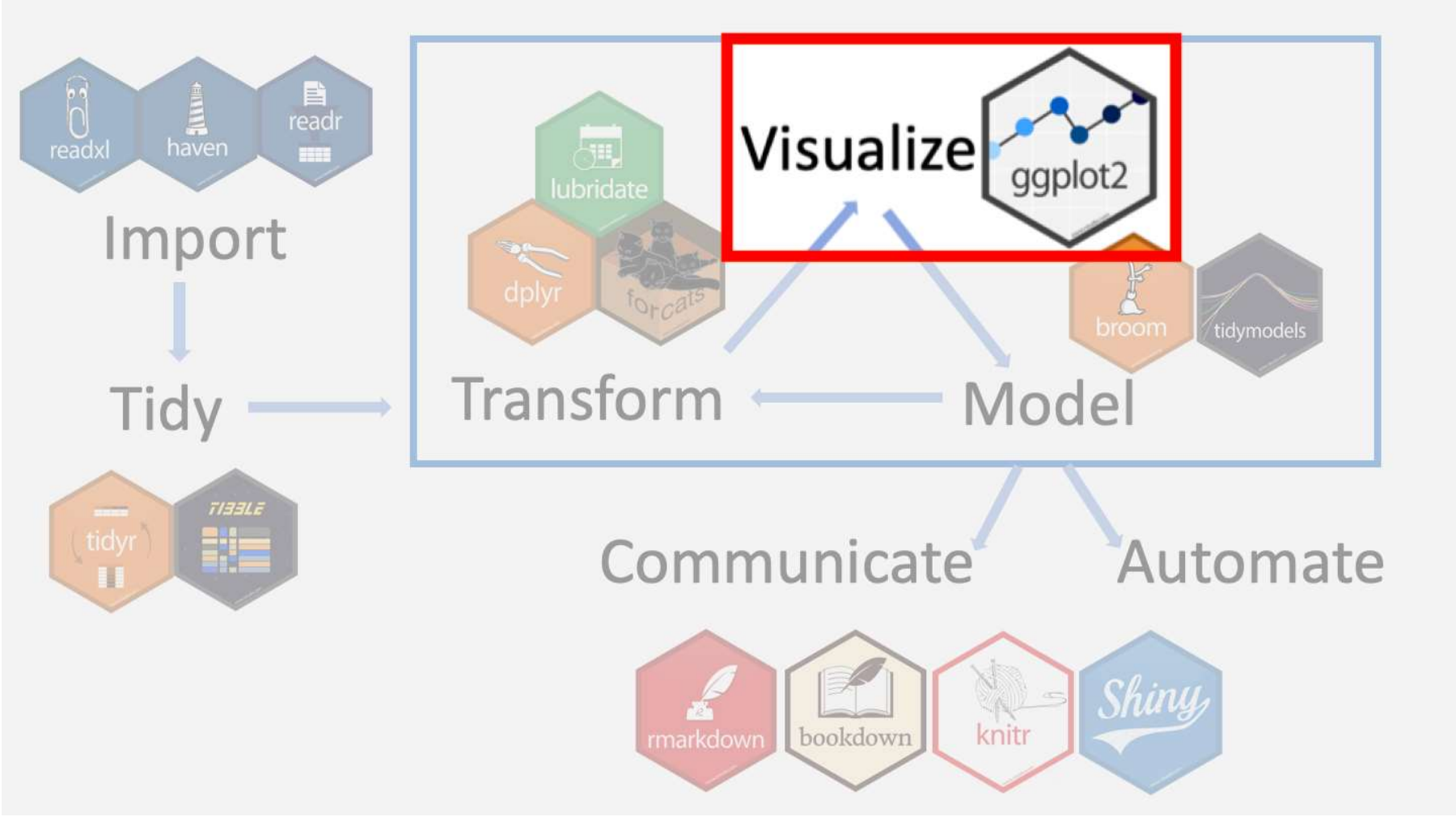
Tidy Workflow



Tidy Workflow



Tidy Workflow

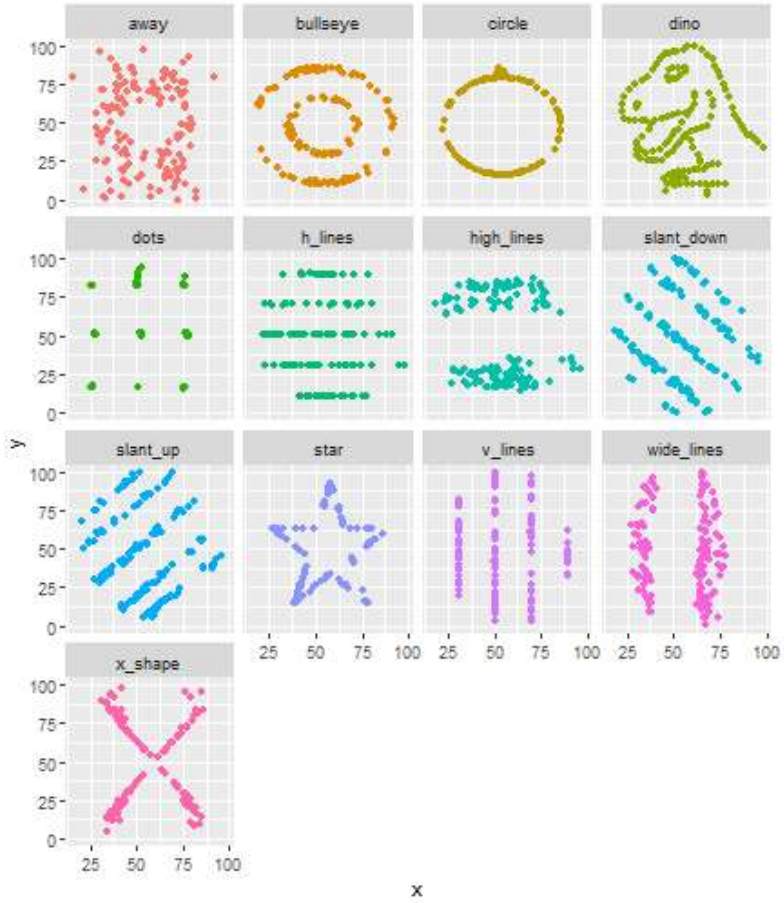


The Datasaurus Dozen

```
library(datasauRus)
library(ggplot2)

datasaurus_dozen |>
  ggplot(aes(x, y, color = dataset)) +
  geom_point(show.legend = FALSE) +
  facet_wrap(~dataset, ncol = 4)
```

```
head(datasaurus_dozen)
```

```
## # A tibble: 6 × 3
##   dataset     x     y
##   <chr>   <dbl> <dbl>
## 1 dino     55.4  97.2
## 2 dino     51.5  96.0
## 3 dino     46.2  94.5
## 4 dino     42.8  91.4
## 5 dino     40.8  88.3
## 6 dino     38.7  84.9
```

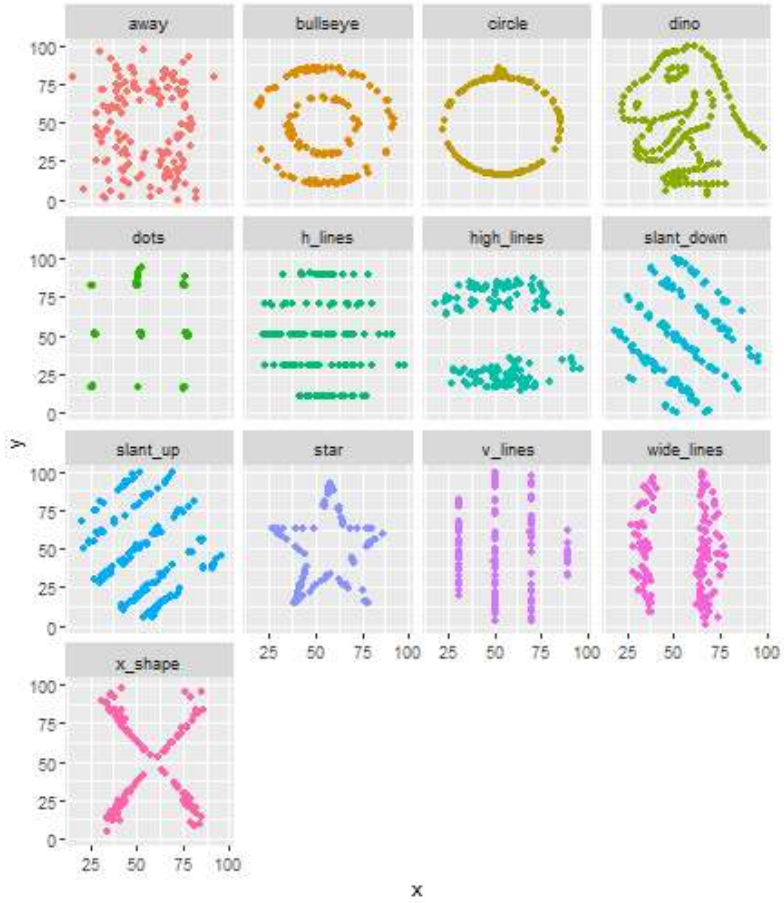


The Datasaurus Dozen

```
library(datasauRus)
library(ggplot2)

datasaurus_dozen |>
  ggplot(aes(x, y, color = dataset)) +
  geom_point(show.legend = FALSE) +
  facet_wrap(~dataset, ncol = 4)
```

Summary statistics	
X Mean	54.263
Y Mean	47.832
X SD	16.765
Y SD	26.935
Corr.	-0.064

The Datasaurus was created by Alberto Cairo



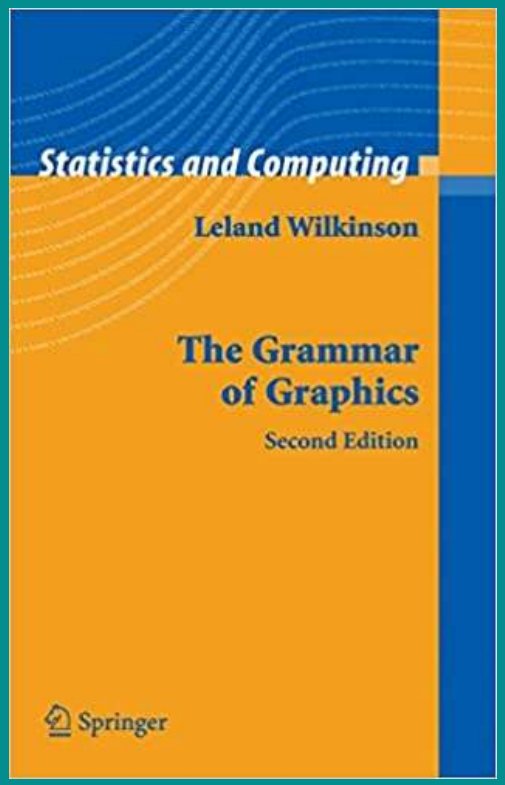
Never trust summary statistics ALONE

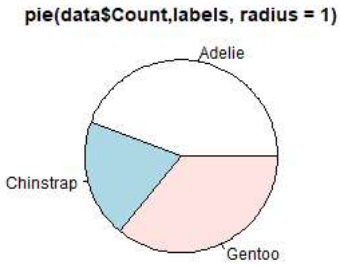
Always visualize your data

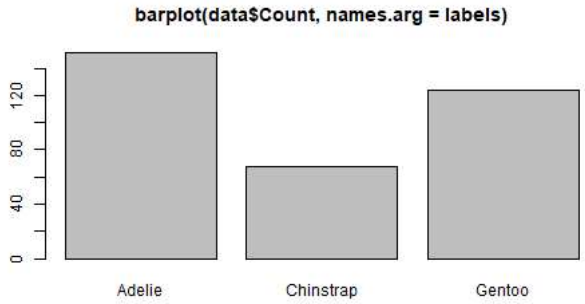
The Grammar of Graphics

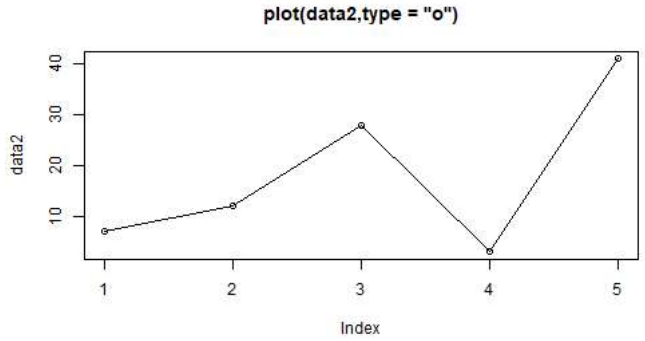
The Book

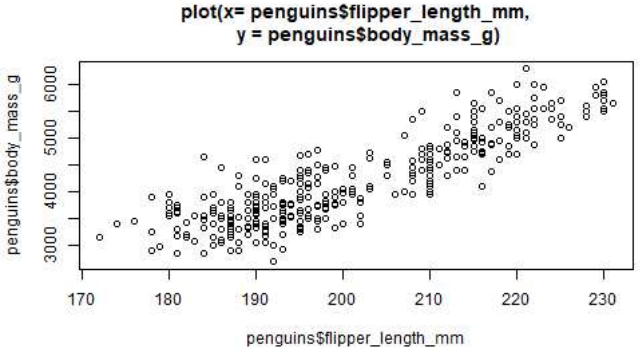
The Grammar of Graphics




pie(data$Count,labels, radius = 1)
Adelie
Chinstrap
Gentoo

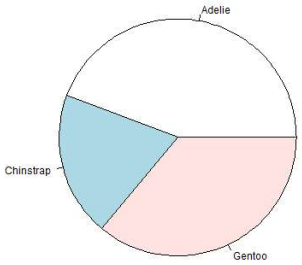

barplot(data$Count, names.arg = labels)
0
40
80
120
Adelie
Chinstrap
Gentoo


plot(data2,type = "o")
data2
10
20
30
40
1
2
3
4
5
Index

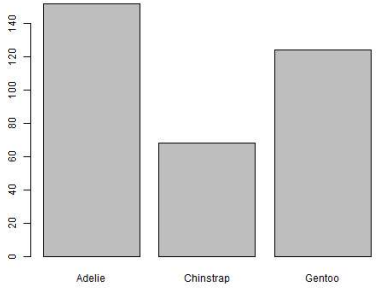

plot(x= penguins$flipper_length_mm,
y = penguins$body_mass_g)
penguins$body_mass_g
3000
4000
5000
6000
170
180
190
200
210
220
230
penguins$flipper_length_mm

The Grammar of Graphics

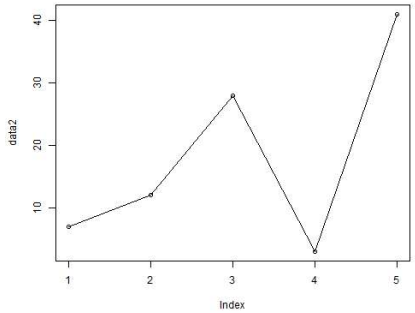
Pie Chart



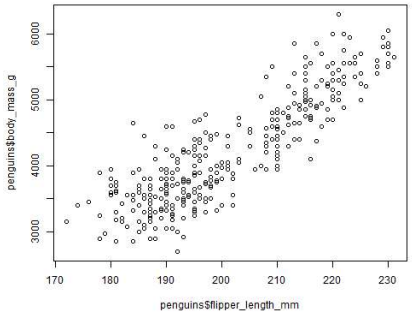
Bar Chart



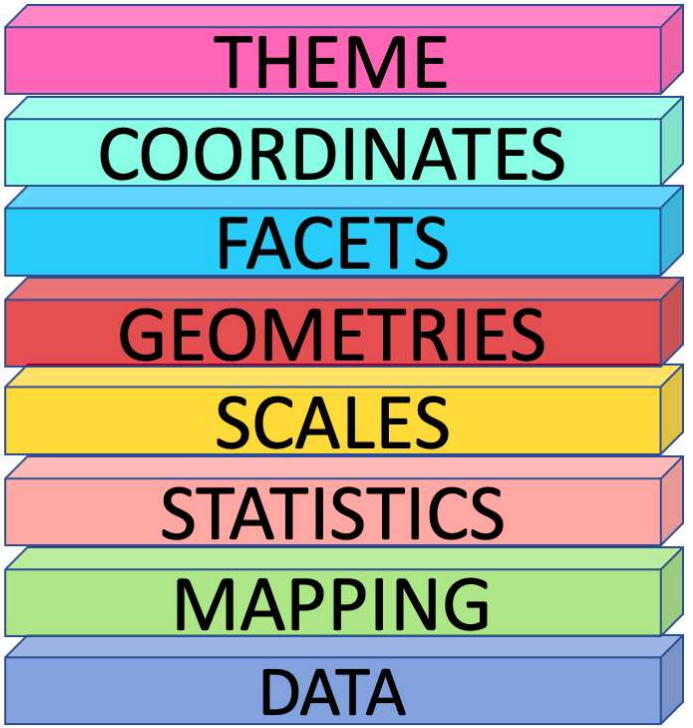
Line Chart



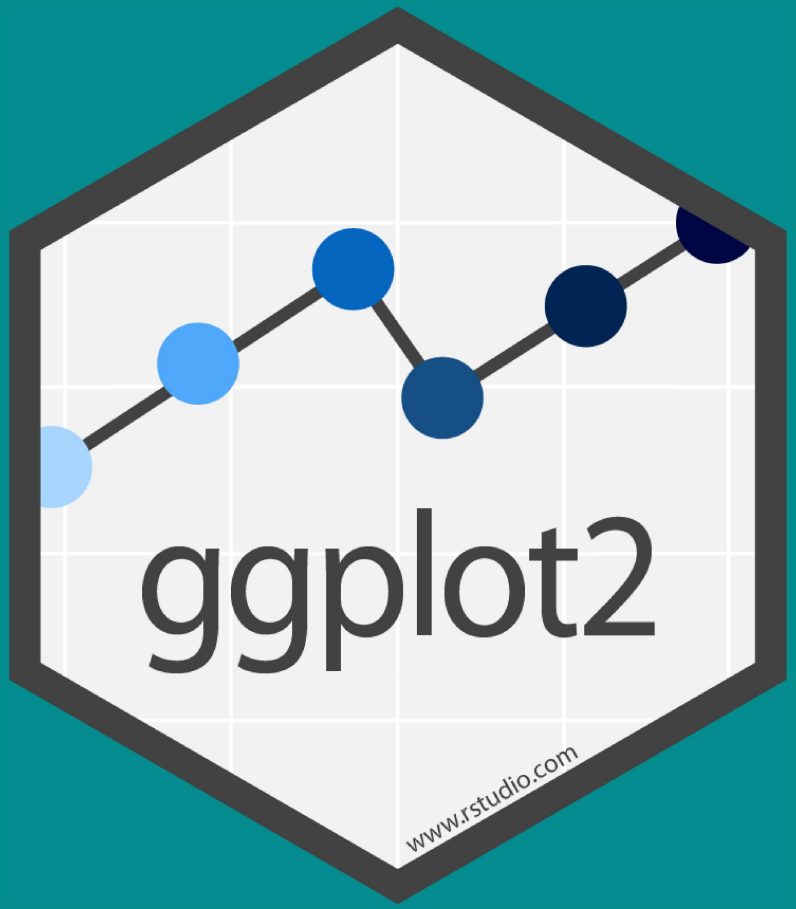
Scatterplot



VS



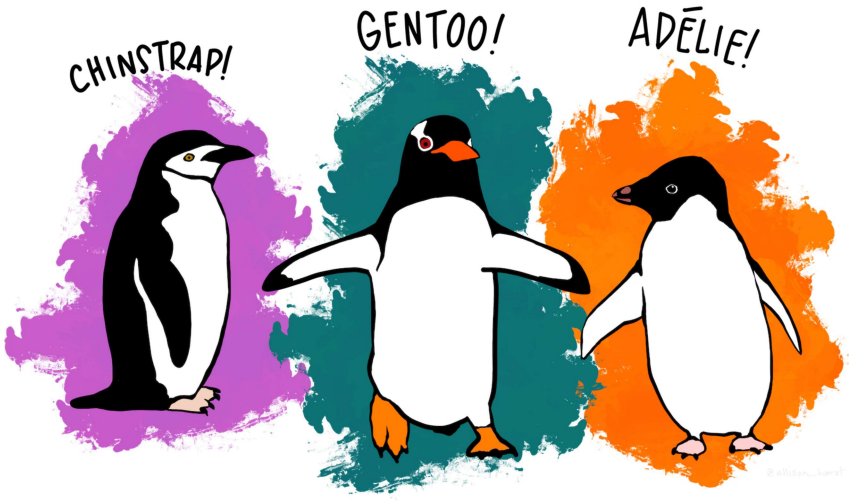
The ggplot2 API

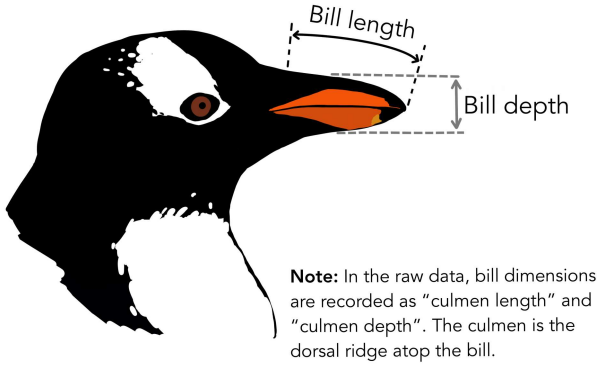


Which dataset to plot?

palmerpenguins data

The Palmer Archipelago penguins. Artwork by @allison_horst.





Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.


```
# A tibble: 6 × 8
  species island    bill_length_mm bill_depth_mm flipper_l…¹ body_…²
  <fct>   <fct>              <dbl>         <dbl>       <int>   <int>
1 Adelie  Torgersen           39.1          18.7         181    3750
2 Adelie  Torgersen           39.5          17.4         186    3800
3 Adelie  Torgersen           40.3          18           195    3250
4 Adelie  Torgersen           NA            NA            NA      NA
5 Adelie  Torgersen           36.7          19.3         193    3450
6 Adelie  Torgersen           39.3          20.6         190    3650
# … with abbreviated variable names ¹flipper_length_mm, ²body_mass_g

Rows: 344
Columns: 8
$ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Ade
$ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen,
$ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.
$ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.
$ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193,
$ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 467
$ sex               <fct> male, female, female, NA, female, male, fem
$ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2
```

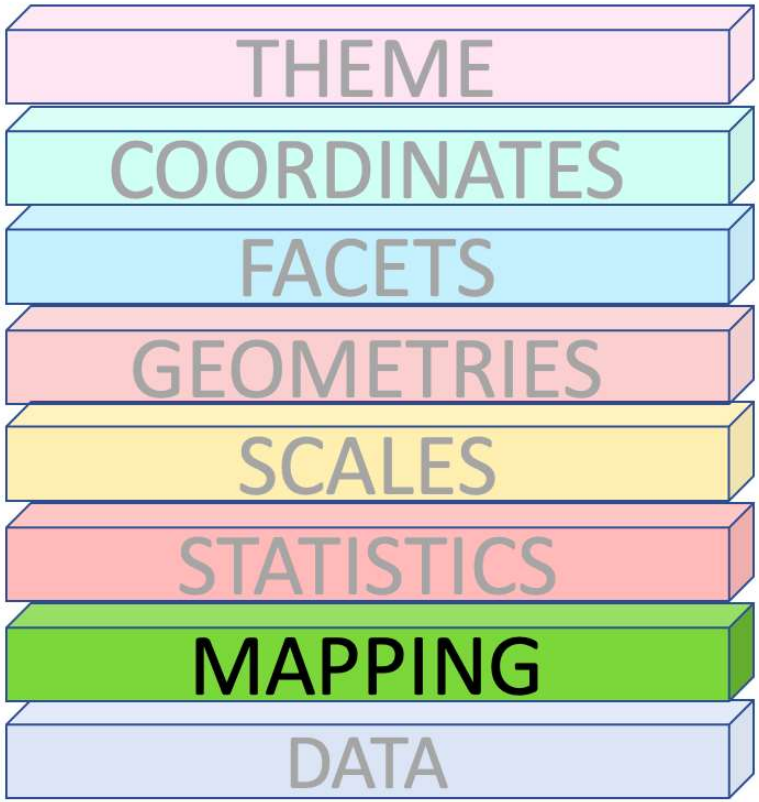
Which dataset to plot?

```
ggplot()
```

Which dataset to plot?

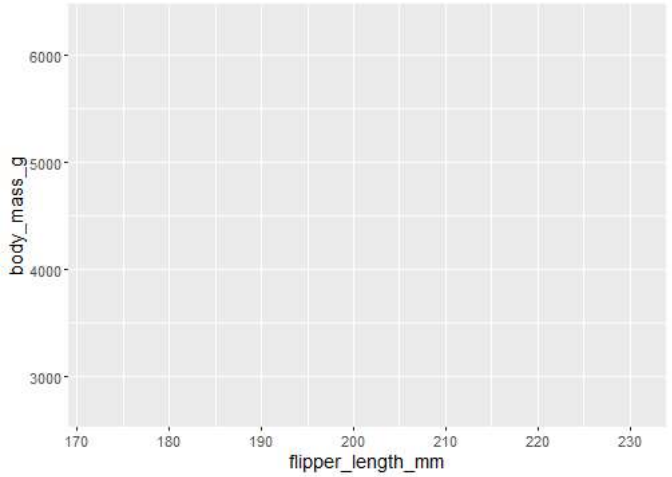
```
ggplot(data = penguins)
```

Mapping

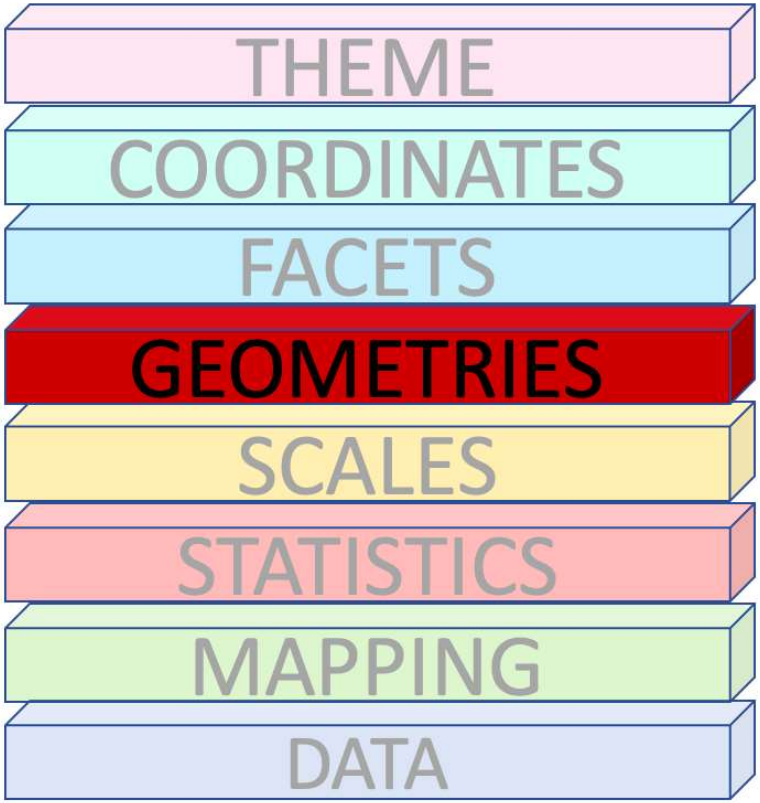

THEME
COORDINATES
FACETS
GEOMETRIES
SCALES
STATISTICS
MAPPING
DATA

Which columns to use for x and y?

```
ggplot(data = penguins,
       mapping = aes(x = flipper_length_mm,
                     y = body_mass_g))
```

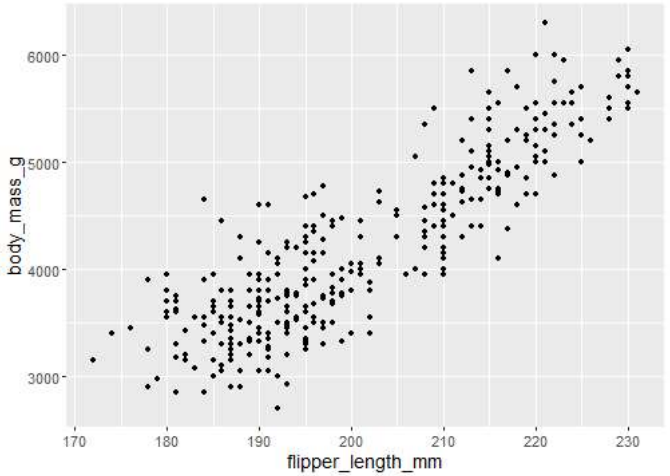


Geometries

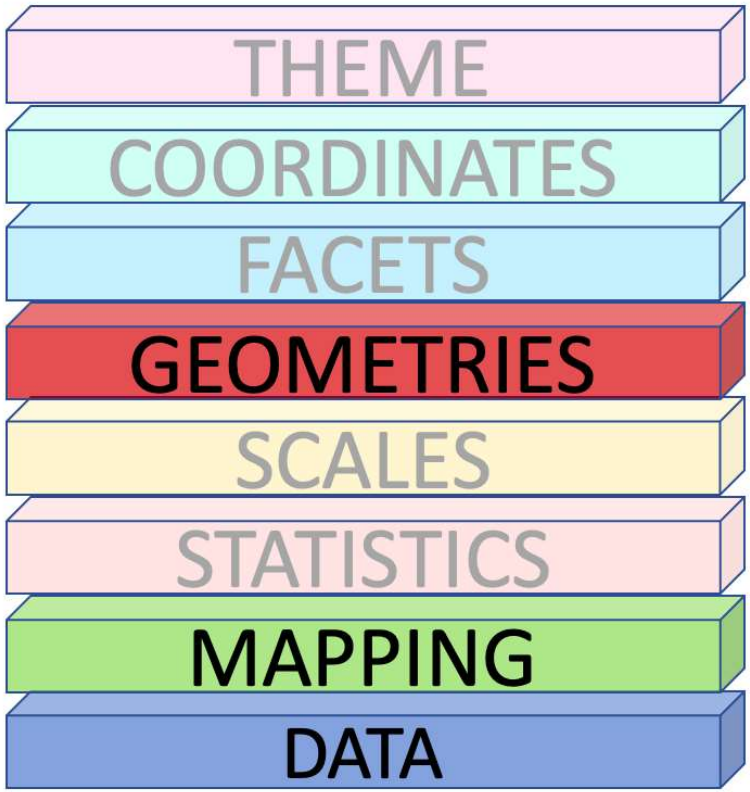

THEME
COORDINATES
FACETS
GEOMETRIES
SCALES
STATISTICS
MAPPING
DATA

How to draw the plot?

```
ggplot(data = penguins,
       mapping = aes(x = flipper_length_mm,
                     y = body_mass_g)) +
  geom_point()
```

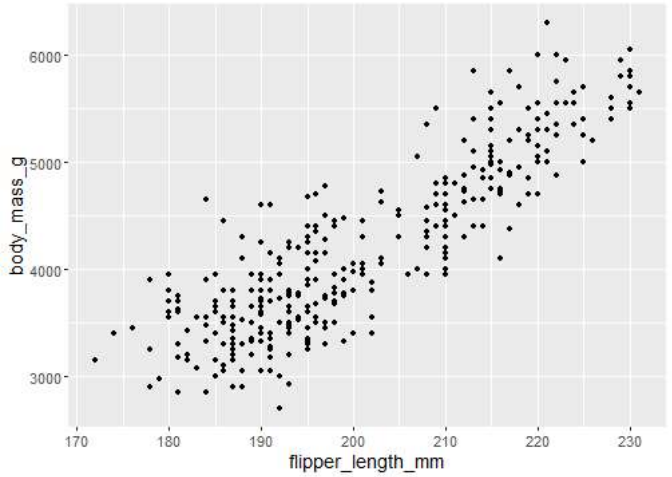


Data, Mapping and Geometries



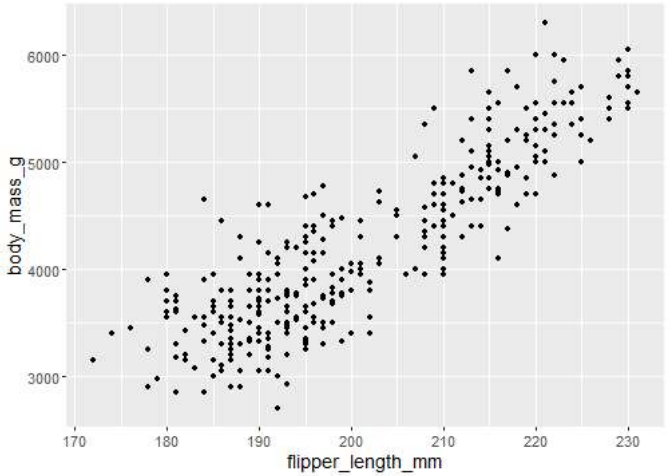
How to draw the plot?

```
ggplot(data = penguins) +
  geom_point(mapping = aes(x = flipper_length_mm,
                           y = body_mass_g))
```



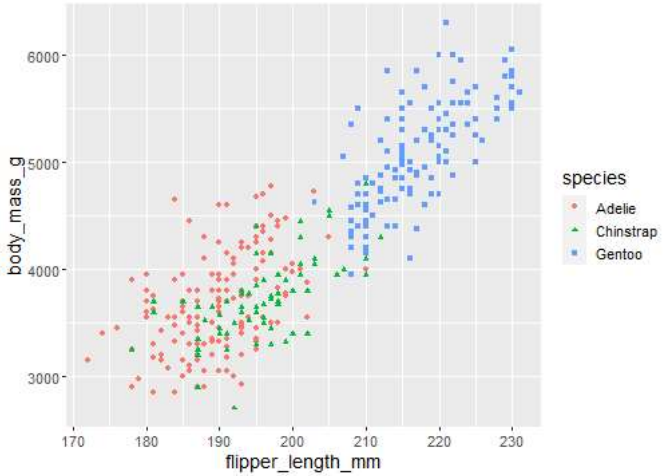
How to draw the plot?

```
ggplot() +
  geom_point(mapping = aes(x = flipper_length_mm,
                           y = body_mass_g),
             data = penguins)
```



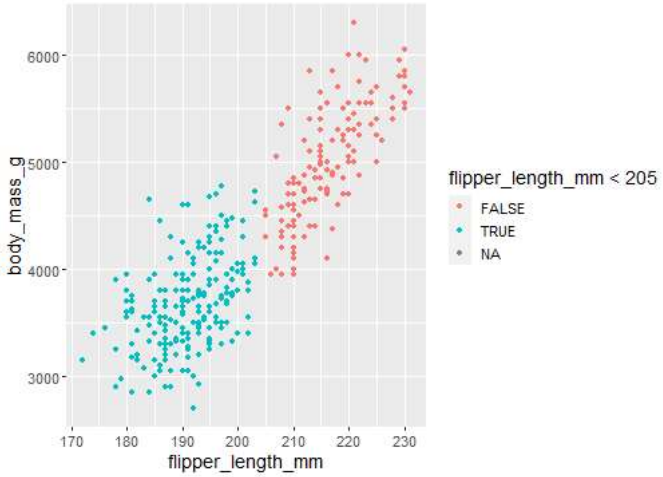
Mapping Colours

```
ggplot(penguins) +
  geom_point( aes(x = flipper_length_mm,
                  y = body_mass_g,
                  color = species,
                  shape = species))
```



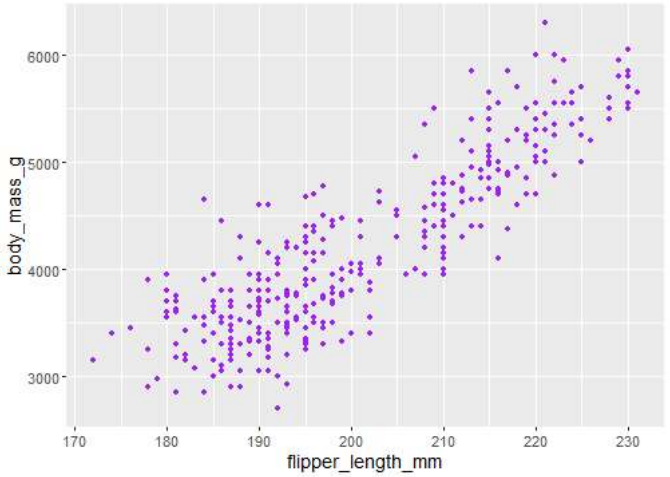
Mapping Colours

```
ggplot(penguins) +
  geom_point( aes(x = flipper_length_mm,
                  y = body_mass_g,
                  colour = flipper_length_mm < 205))
```

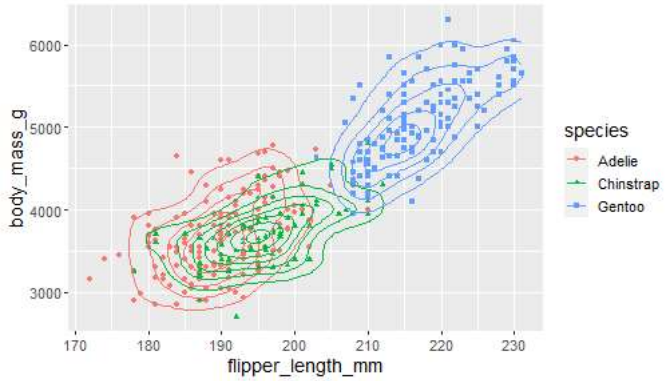


Setting Colours

```
ggplot(penguins) +
  geom_point( aes(x = flipper_length_mm,
                  y = body_mass_g),
                  colour = 'purple')
```

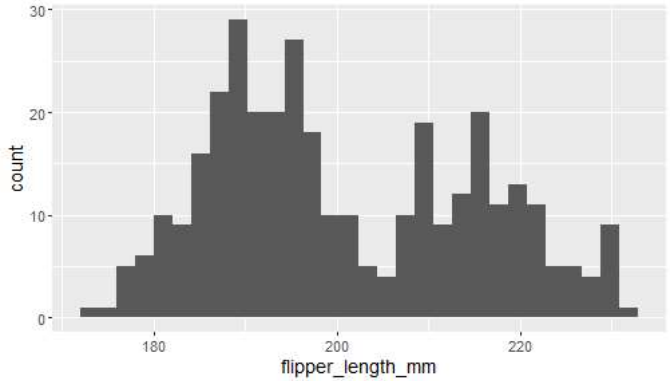


```
ggplot(penguins,
       aes(x = flipper_length_mm,
           y = body_mass_g,
           color = species,
           shape = species)) +
  geom_point() +
  geom_density_2d()
```



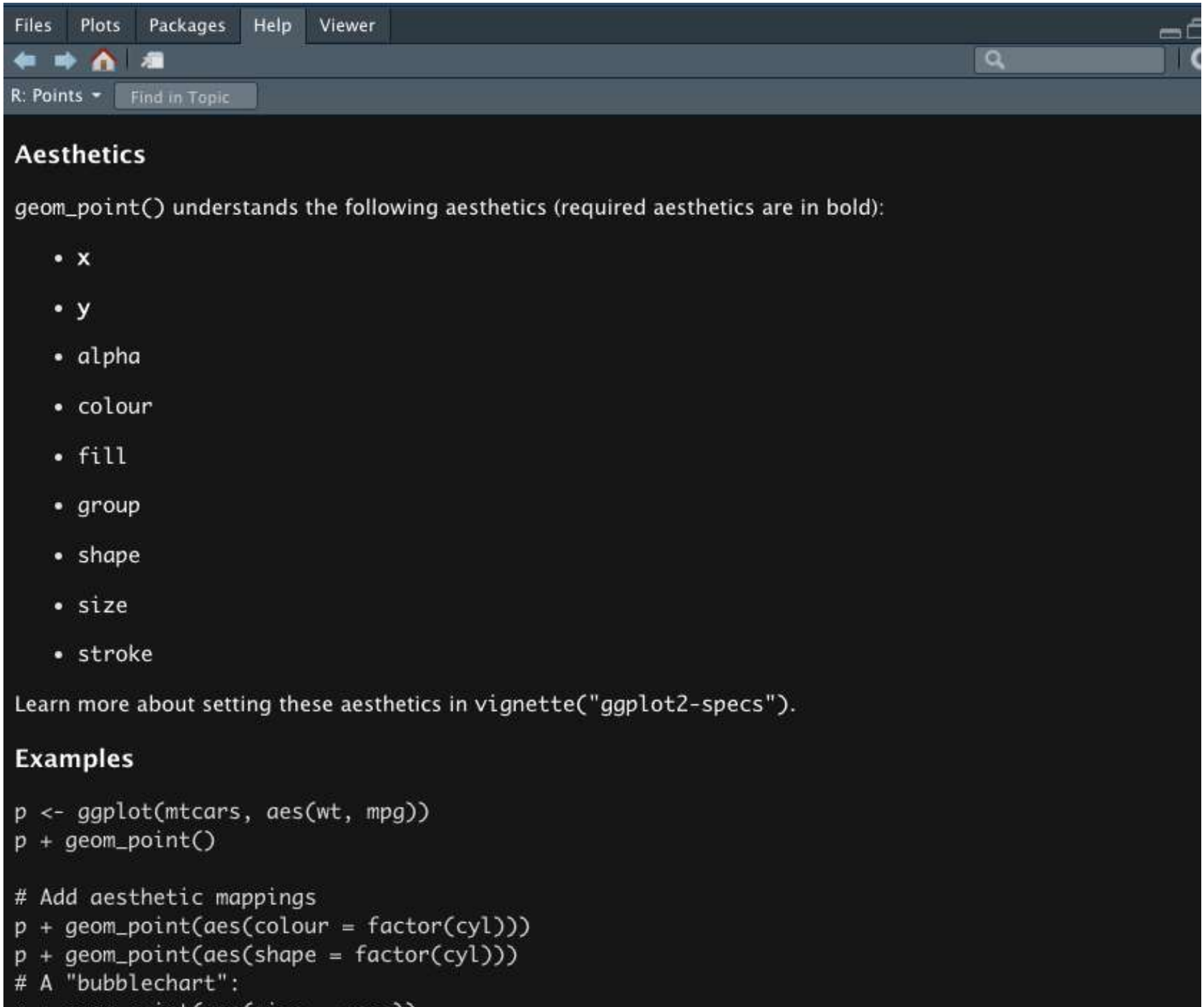
- Syntax starts with `geom_*`.
- eg: geom_histogram(), geom_bar(), geom_boxplot().
- Each shape has its own specific aesthetics arguments.

```
ggplot(penguins) +
  geom_histogram(
    aes(x = flipper_length_mm))
```

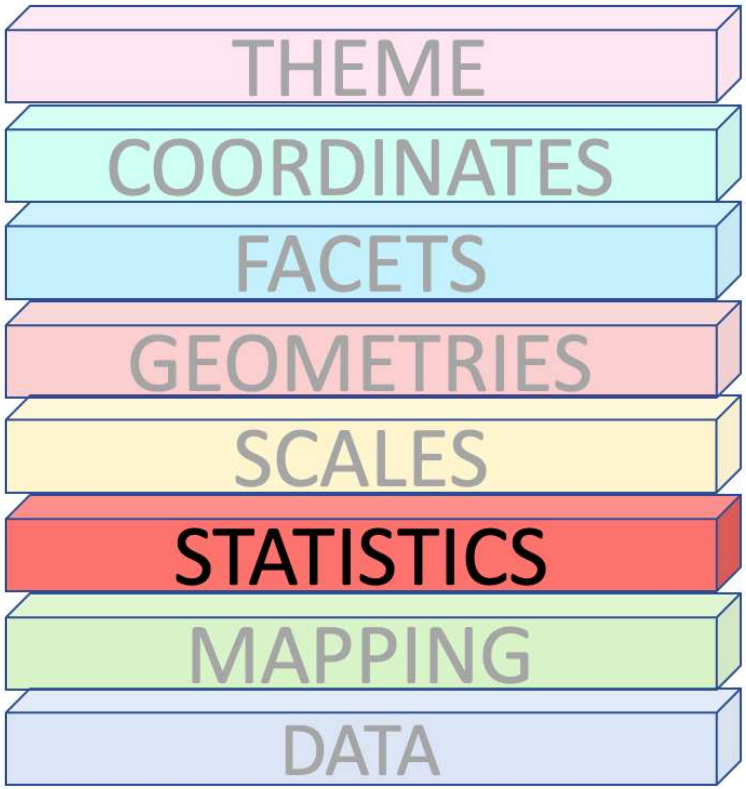


Each shape has its own specific aesthetics arguments.

`?geom_point`



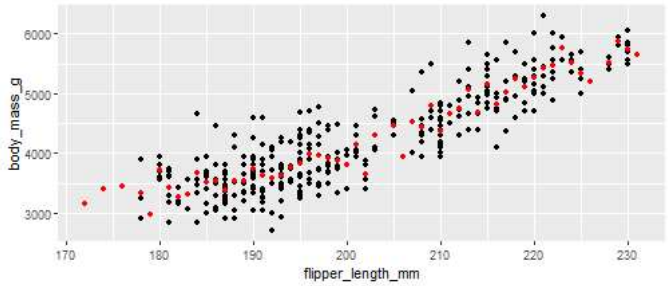
Statistics


THEME
COORDINATES
FACETS
GEOMETRIES
SCALES
STATISTICS
MAPPING
DATA

- There are two ways to use statistical functions.

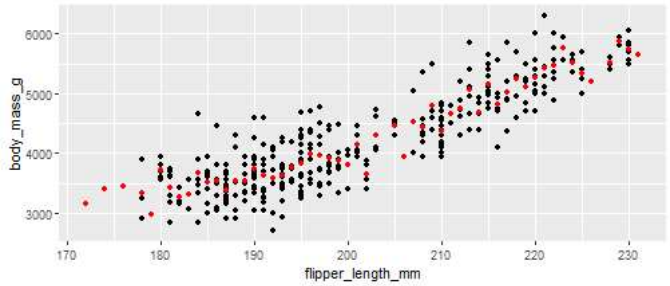
define `stat_*()` function and `geom` argument inside that function

```
ggplot(penguins,
       aes(x = flipper_length_mm,
           y = body_mass_g)) +
  geom_point() +
  stat_summary(
    geom ="point",
    fun.y ="mean",
    colour ="red")
```

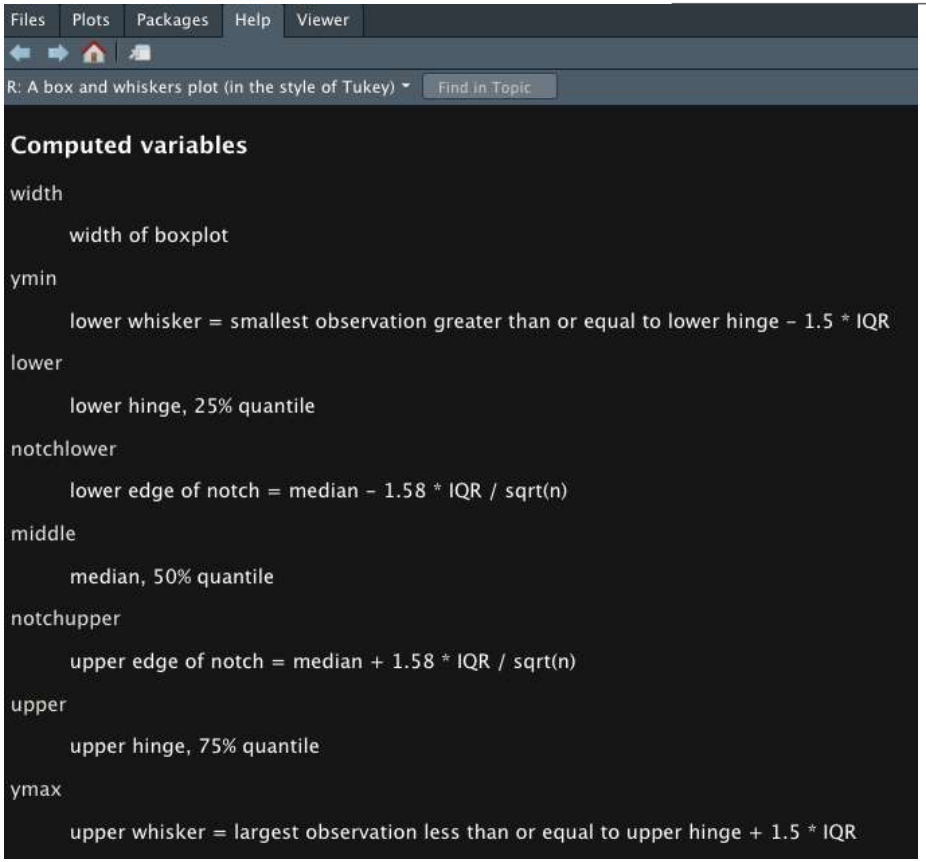


define `geom_*()` function and `stat` argument inside that function

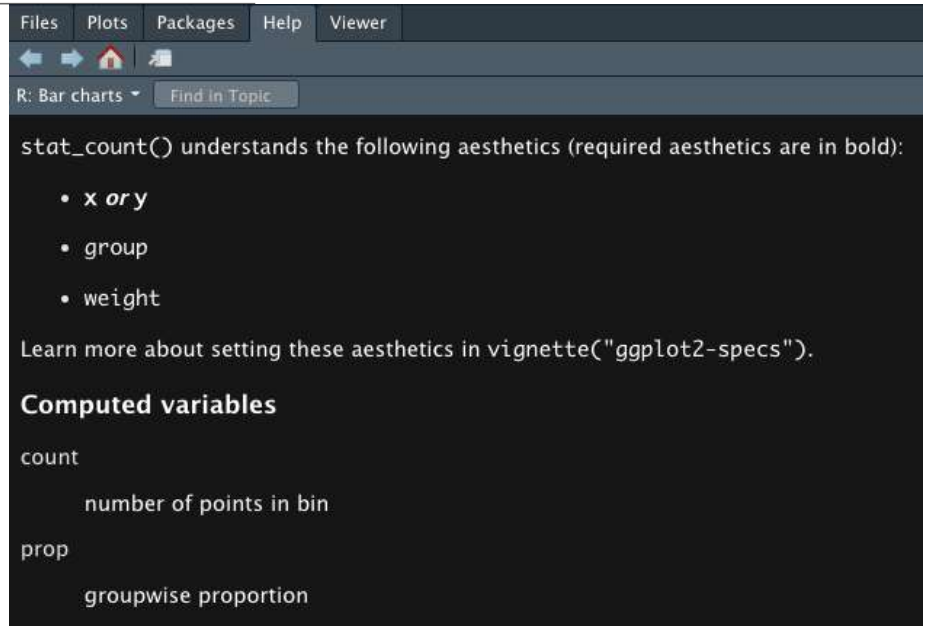
```
ggplot(penguins,
       aes(x = flipper_length_mm,
           y = body_mass_g)) +
  geom_point() +
  geom_point(
    stat ="summary",
    fun.y ="mean",
    colour ="red")
```



Statistics	Geometries
stat_count	geom_bar
stat_boxplot	geom_boxplot
stat_identity	geom_col
stat_bin	geom_bar, geom_histogram
stat_density	geom_density

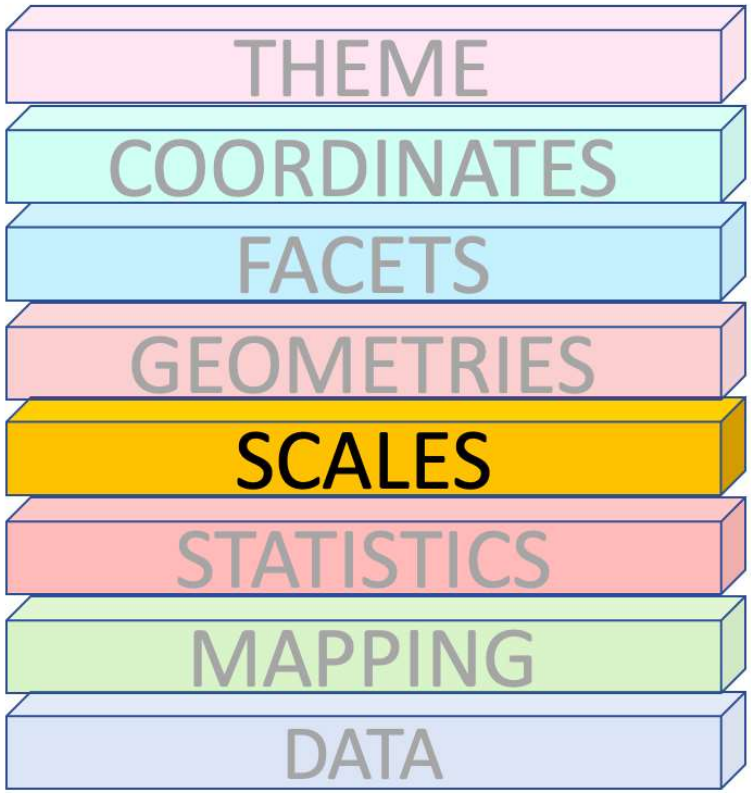


?geom_boxplot



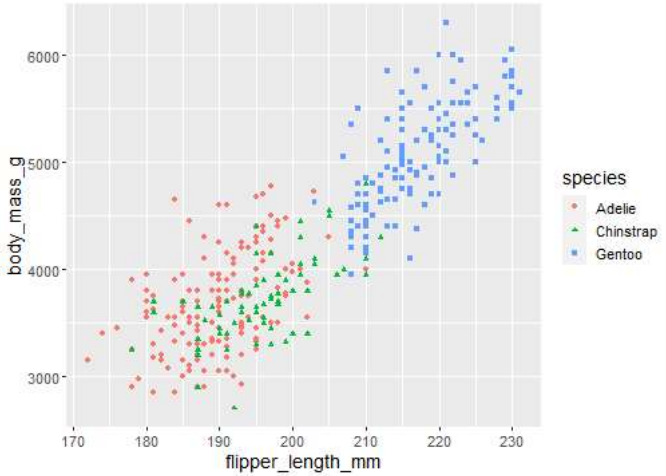
?geom_bar

Scales


THEME
COORDINATES
FACETS
GEOMETRIES
SCALES
STATISTICS
MAPPING
DATA

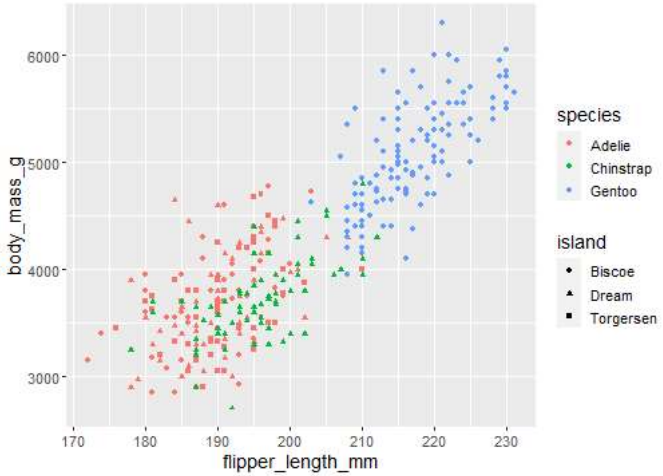
Scales

```
ggplot(penguins) +
  geom_point( aes(x = flipper_length_mm,
                  y = body_mass_g,
                  color = species,
                  shape = species))
```



Scales

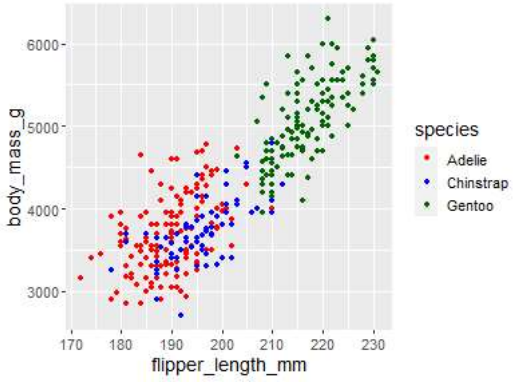
```
ggplot(penguins) +
  geom_point( aes(x = flipper_length_mm,
                  y = body_mass_g,
                  color = species,
                  shape = island))
```



Scales manual

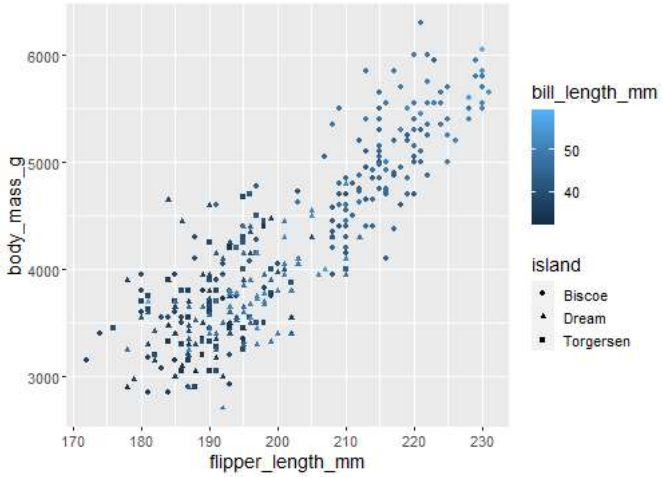
- It's recommended to use a named vector

```
cols <- c("Adelie" = "red", "Chinstrap" = "blue", "Gentoo" = "darkgreen")

ggplot(penguins) +
  geom_point( aes(x = flipper_length_mm,
                  y = body_mass_g,
                  color = species)) +
  scale_colour_manual(values = cols)
```



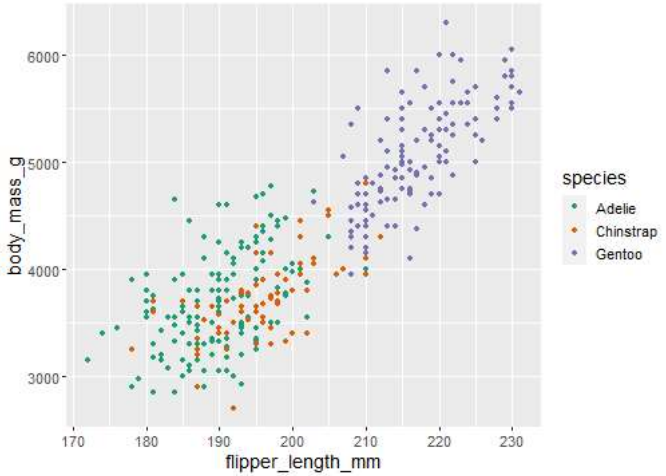
Scales

```
ggplot(penguins) +
  geom_point( aes(x = flipper_length_mm,
                  y = body_mass_g,
                  color = bill_length_mm,
                  shape = island))
```



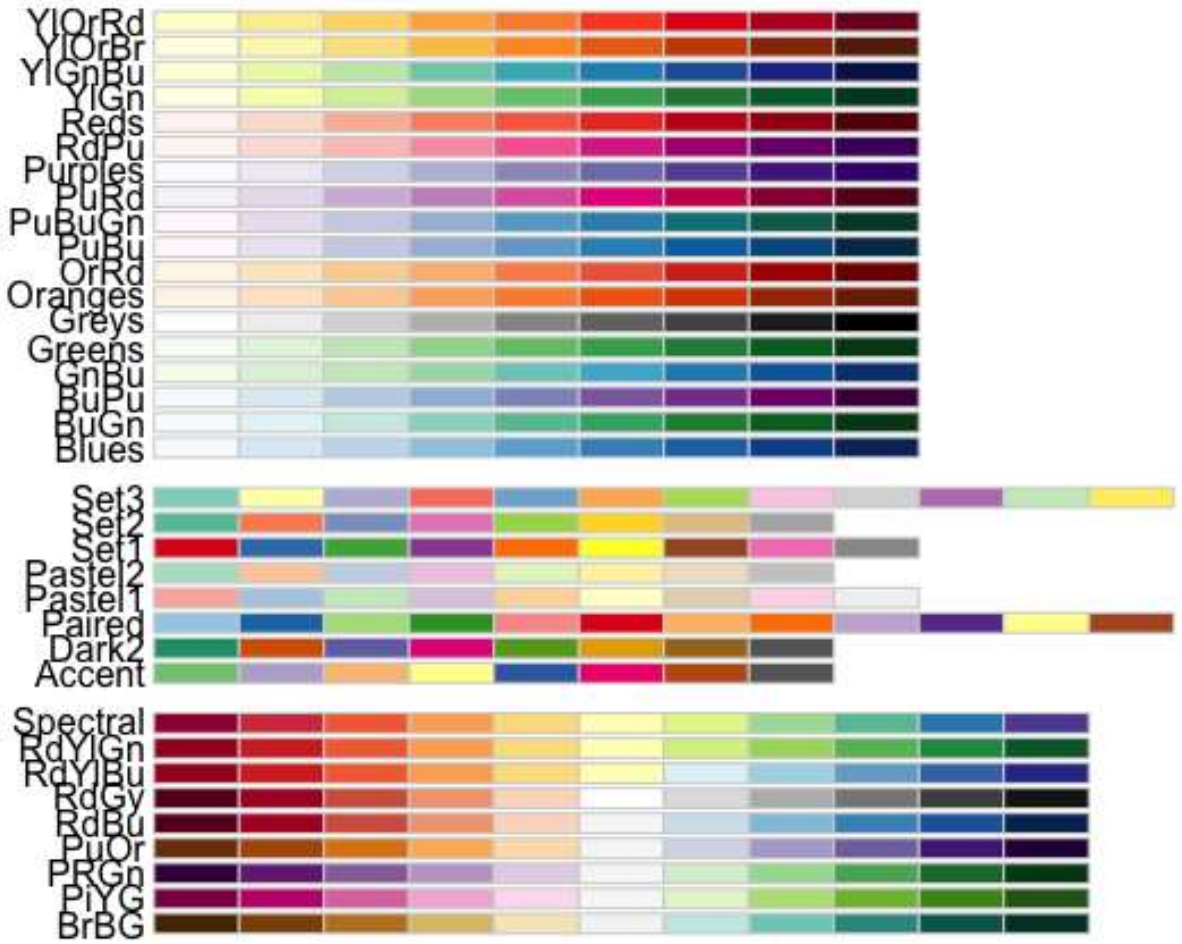
Scales

```
ggplot(penguins) +
  geom_point(aes(x = flipper_length_mm,
                 y = body_mass_g,
                 color = species)) +
  scale_color_brewer(type = 'qual',
                     palette =  'Dark2')
```

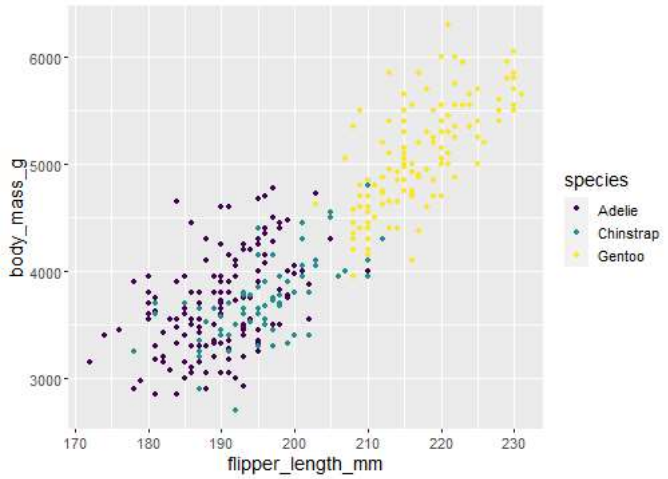


- scale_<aesthetic>_<type>

```
RColorBrewer::display.brewer.all()
```

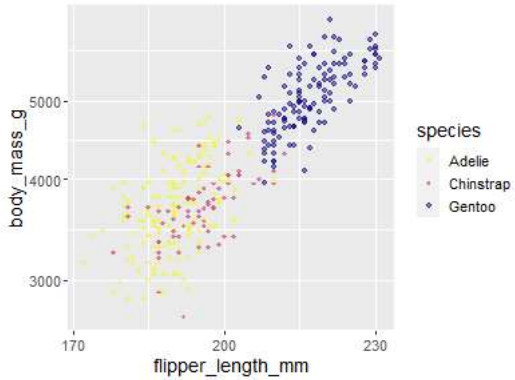


```
ggplot(penguins) +
  geom_point(aes(x = flipper_length_mm,
                 y = body_mass_g,
                 color = species)) +
  scale_color_viridis_d()
```

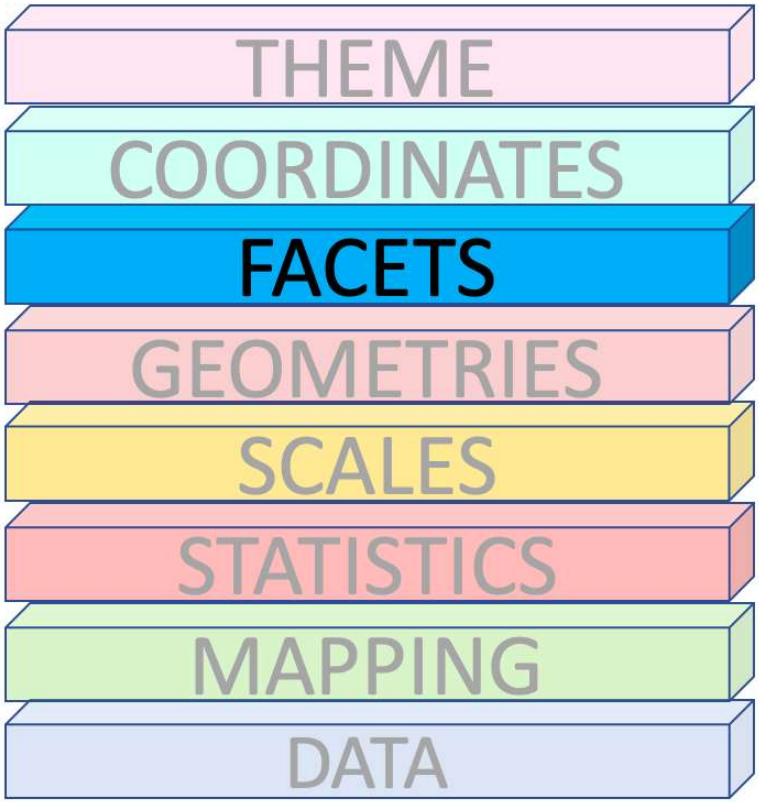


- `viridis` and `RColorBrewer` provide different color scales that are robust to color-blindness.
- For details and an interactive palette selection tools see http://colorbrewer.org

```r
ggplot(penguins) +
  geom_point(aes(x = flipper_length_mm,
                 y = body_mass_g,
                 color = species,
                 shape = species,
                 alpha = species)) +
  scale_x_continuous( breaks = c(170,200,230)) +
  scale_y_log10() +
  scale_colour_viridis_d(direction = -1, option= 'plasma') +
  scale_shape_manual( values = c(17,18,19)) +
  scale_alpha_manual( values = c( "Adelie" = 0.6, "Gentoo" = 0.5, #
                                  "Chinstrap" = 0.7))
```

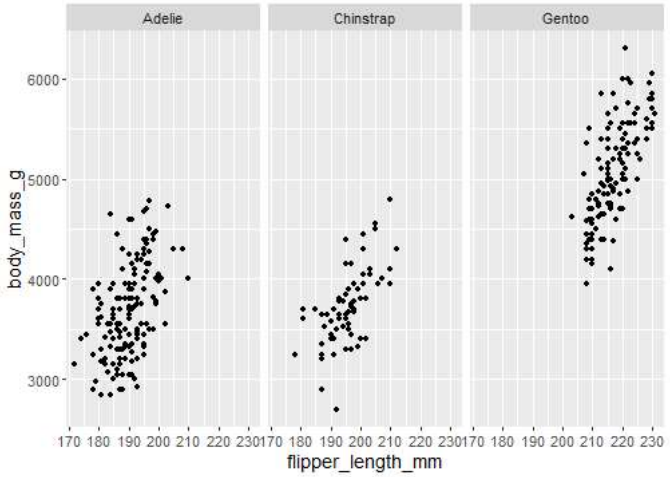


Facets


THEME
COORDINATES
FACETS
GEOMETRIES
SCALES
STATISTICS
MAPPING
DATA

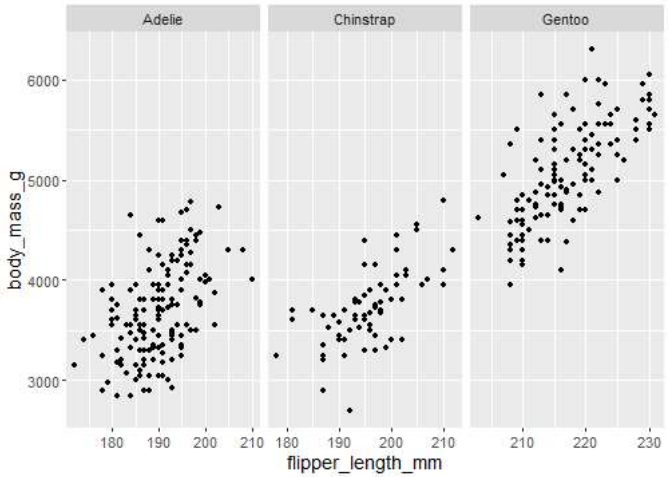
facet_wrap()

```
ggplot(penguins) +
  geom_point(aes(
    x = flipper_length_mm,
    y = body_mass_g)) +
  facet_wrap(vars(species))
```



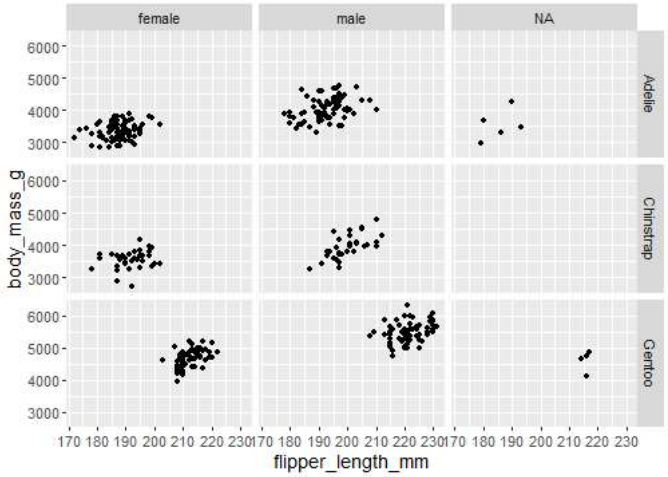
facet_wrap()

```
ggplot(penguins) +
  geom_point(aes(
    x = flipper_length_mm,
    y = body_mass_g)) +
  facet_wrap(vars(species),
             scales = "free_x")
```

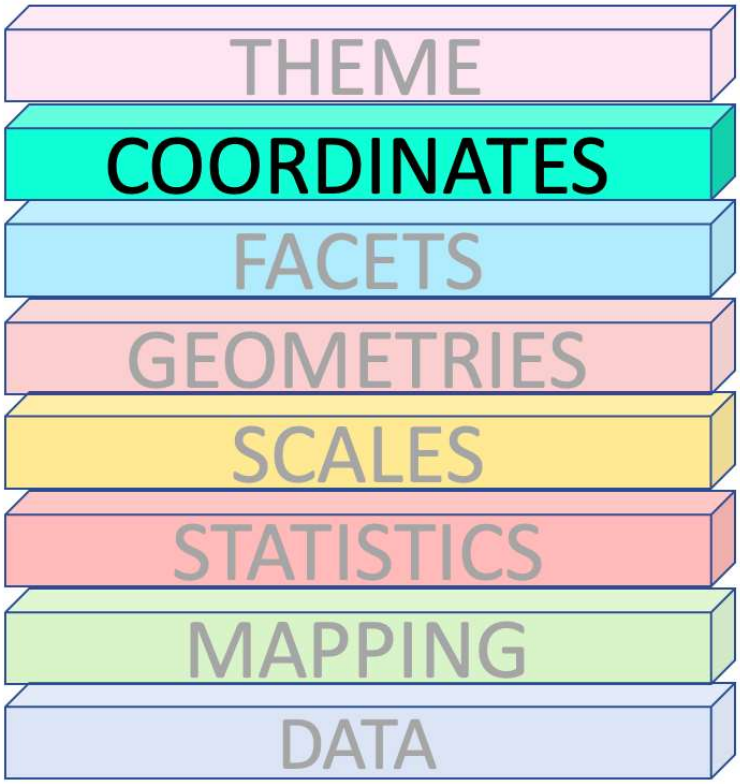


facet_grid()

```
ggplot(penguins) +
  geom_point(aes(
    x = flipper_length_mm,
    y = body_mass_g)) +
  facet_grid( vars(species), vars(sex))
```

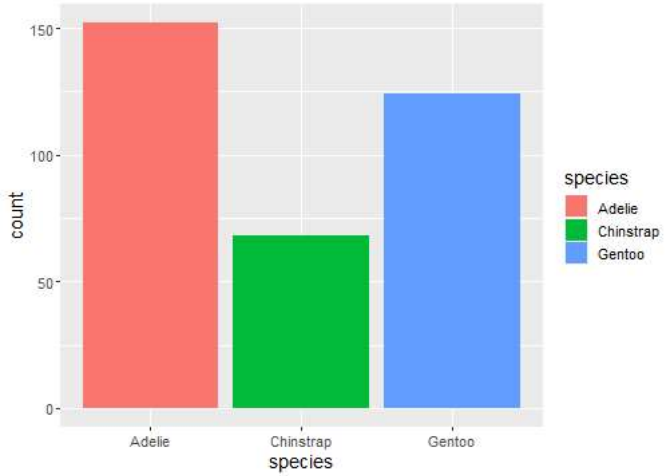


Coordinates

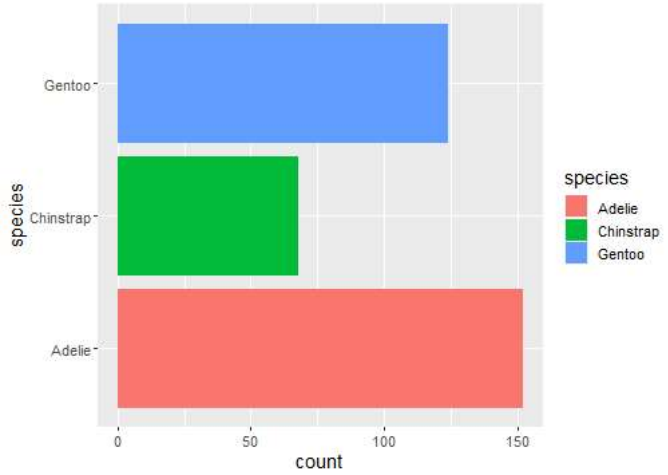


Coordinates

```
ggplot(penguins) +
  geom_bar(aes(x= species, fill = species))
```

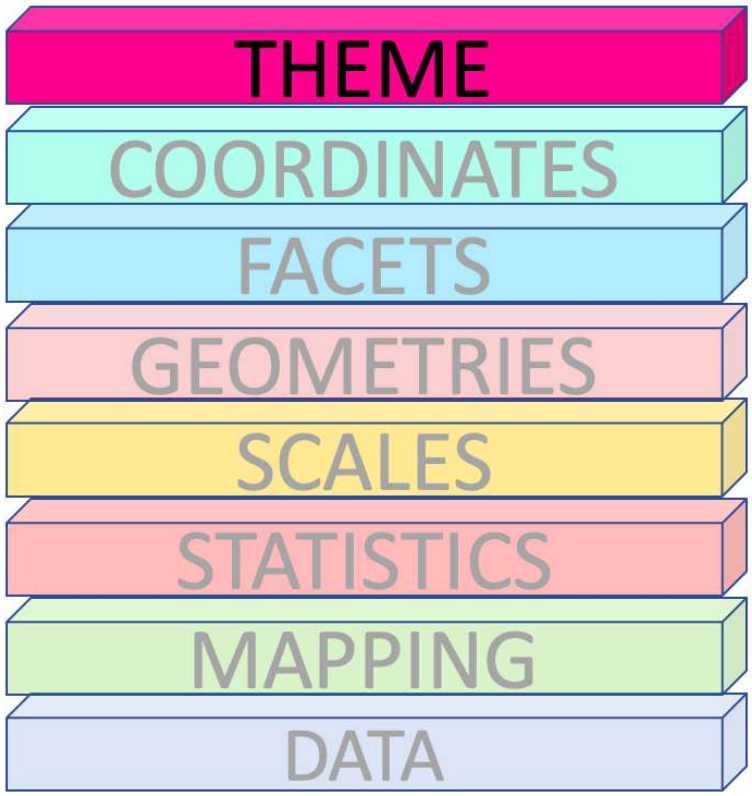


```
ggplot(penguins) +
  geom_bar(aes(x= species, fill = species)) +
  coord_flip()
```



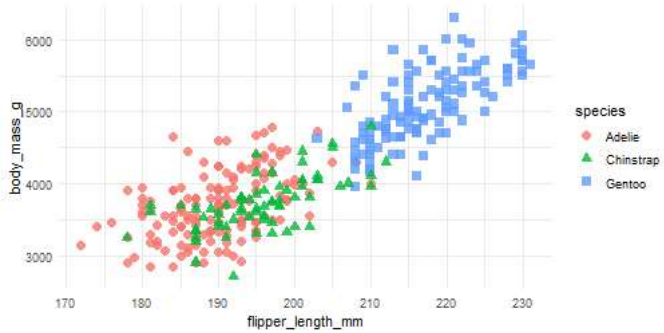
- There are two types of coordinate systems:
  - Linear coordinate systems
  - Non-linear coordinate systems
- Linear coordinate systems : `coord_cartesian()`, `coord_flip()`, `coord_fixed()`
- Non-linear coordinate systems : eg : `coord_map()`, `coord_quickmap()`, `coord_sf()`, `coord_polar()`, `coord_trans()`

Themes

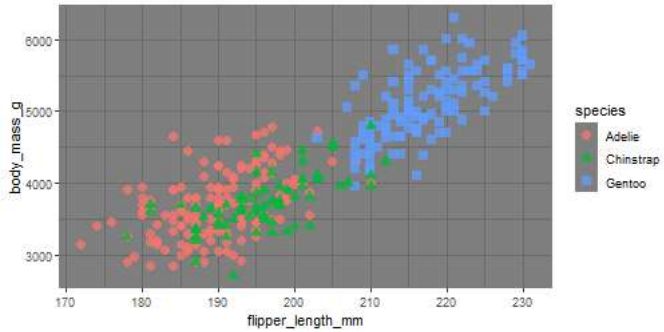

THEME
COORDINATES
FACETS
GEOMETRIES
SCALES
STATISTICS
MAPPING
DATA

These are complete themes which control all non-data display.

```
ggplot(data = penguins,
       aes(x = flipper_length_mm,
           y = body_mass_g)) +
  geom_point(aes(
    color = species,
    shape = species),
    size = 3,
    alpha = 0.8) +
  theme_minimal()
```

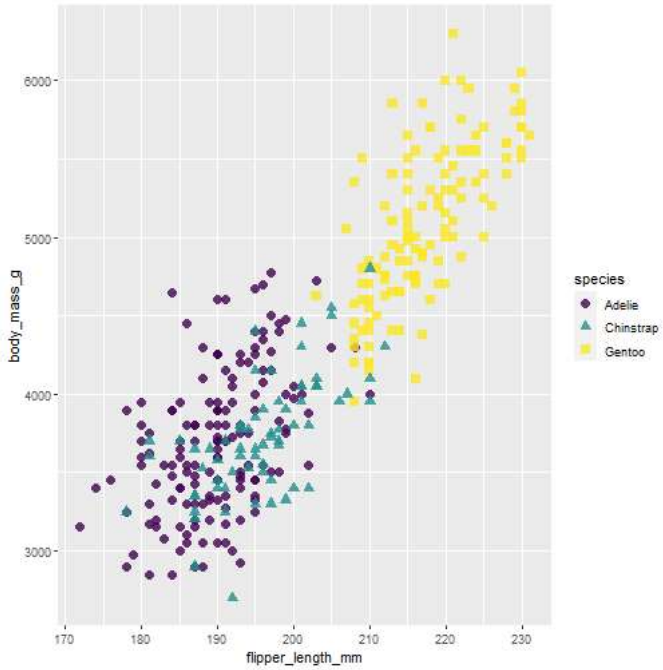


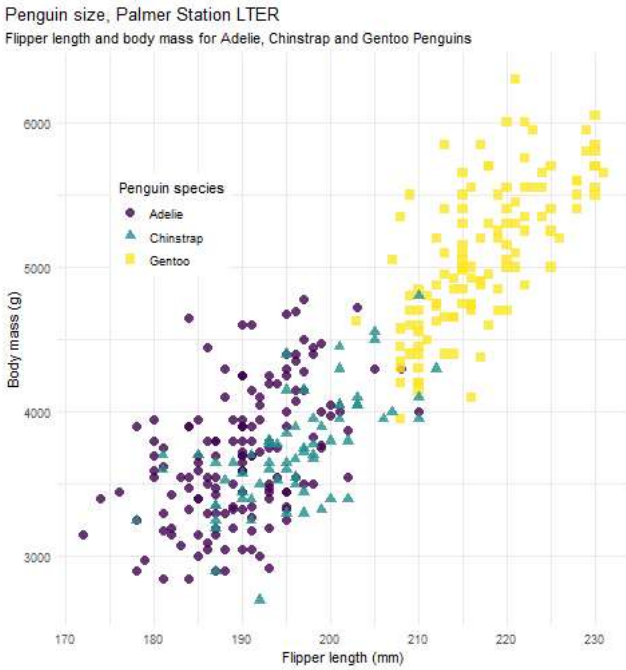

```
ggplot(data = penguins,
       aes(x = flipper_length_mm,
           y = body_mass_g)) +
  geom_point(aes(
    color = species,
    shape = species),
    size = 3,
    alpha = 0.8) +
  theme_dark()
```

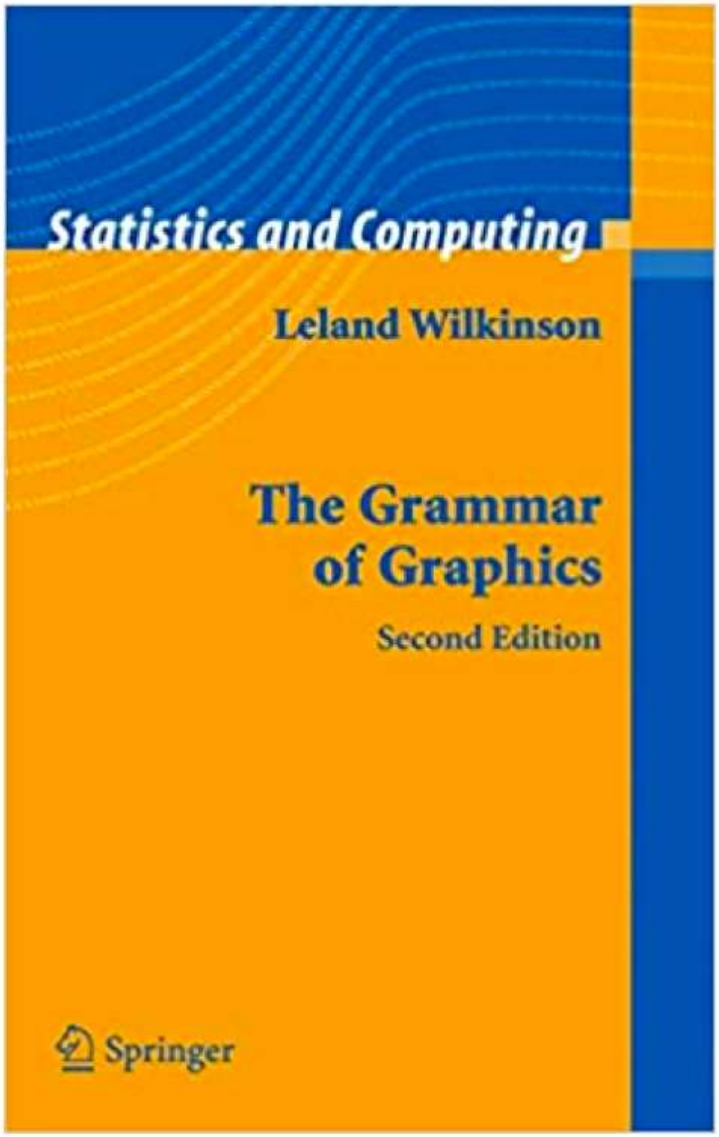


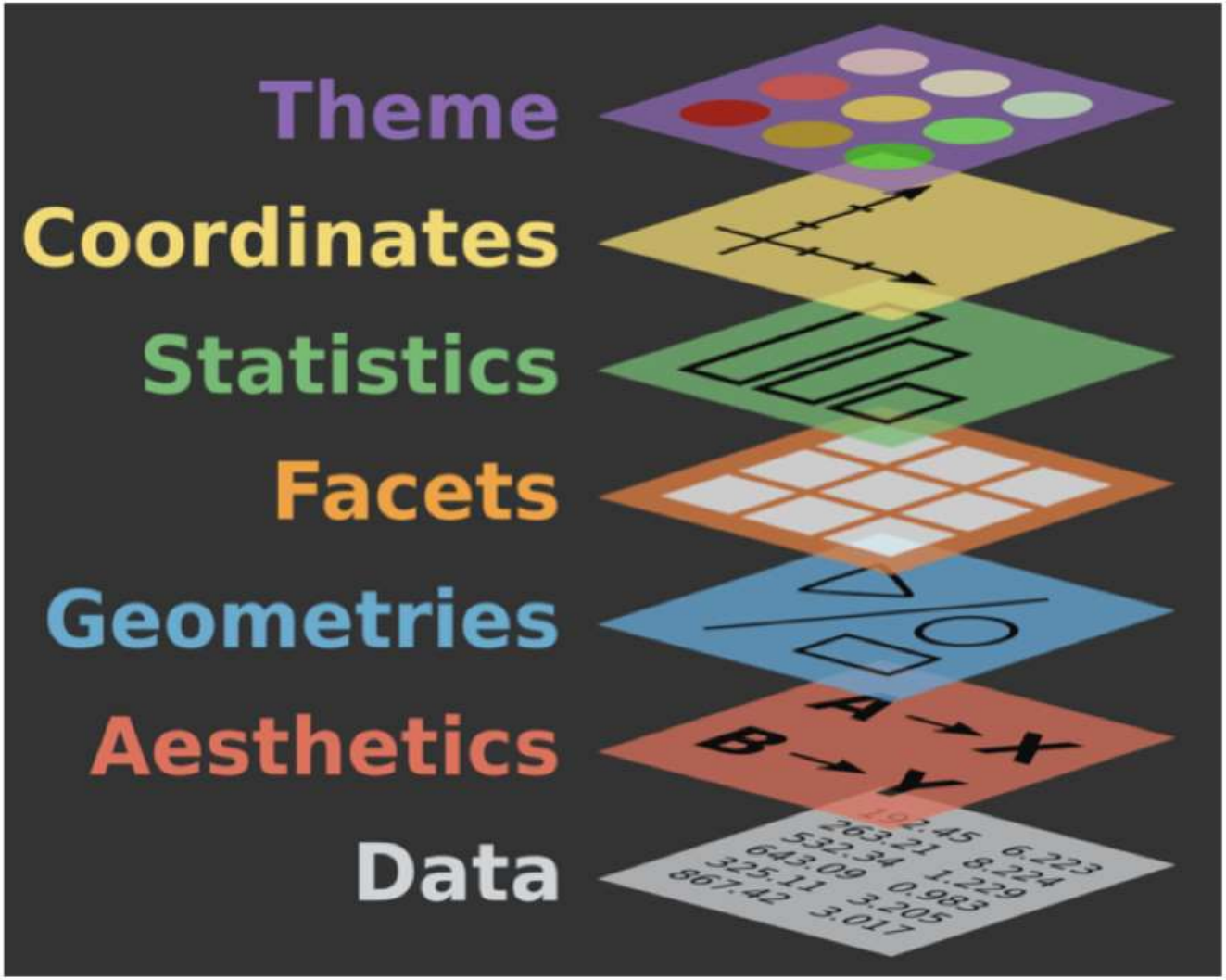
Create custom themes in ggplot.

```
ggplot(penguins,
       aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(color = species, shape = species), size = 3, alpha = 0.8) +
  scale_color_viridis_d() +
  theme_minimal() +
  labs(
    title = "Penguin size, Palmer Station LTER",
    subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
    x = "Flipper length (mm)", y = "Body mass (g)",
    color = "Penguin species", shape = "Penguin species") +
  theme(
    aspect.ratio = 1, legend.position = c(0.2, 0.7),
    legend.background =
      element_rect(
        fill = "white",
        color = NA),
    plot.title.position = "plot",
    plot.caption =
      element_text(
        hjust = 0,
        face= "italic"),
    plot.caption.position = "plot")
```


6000
5000
4000
3000
body_mass_g
170
180
190
200
210
220
230
flipper_length_mm
species
Adelie
Chinstrap
Gentoo


Penguin size, Palmer Station LTER
Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins
6000
5000
4000
3000
Body mass (g)
170
180
190
200
210
220
230
Flipper length (mm)
Penguin species
Adelie
Chinstrap
Gentoo

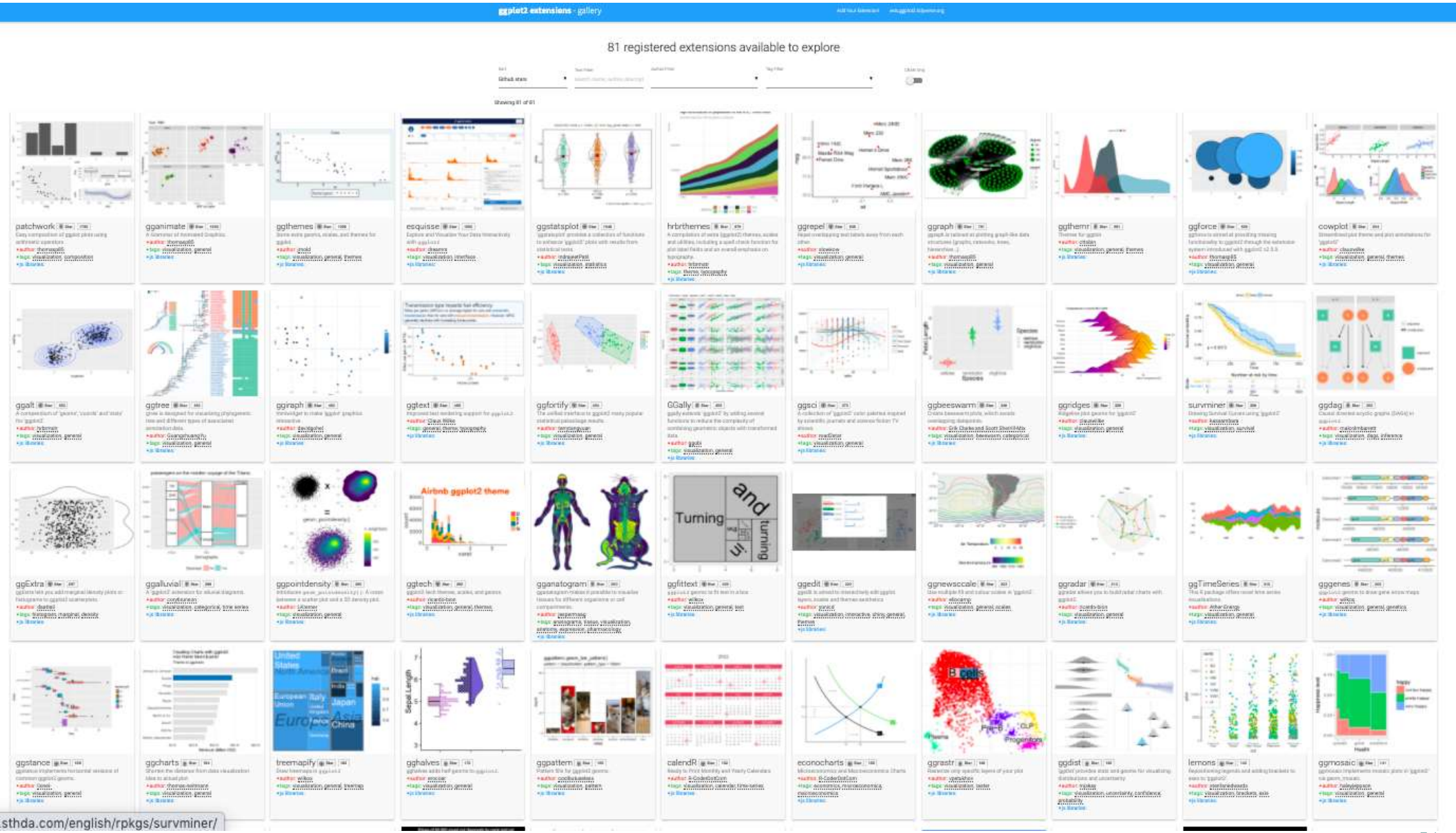




https://www2.stat.duke.edu/courses/Spring20/sta199.002/slides/02-data-and-viz.html#1

ggplot2 extensions

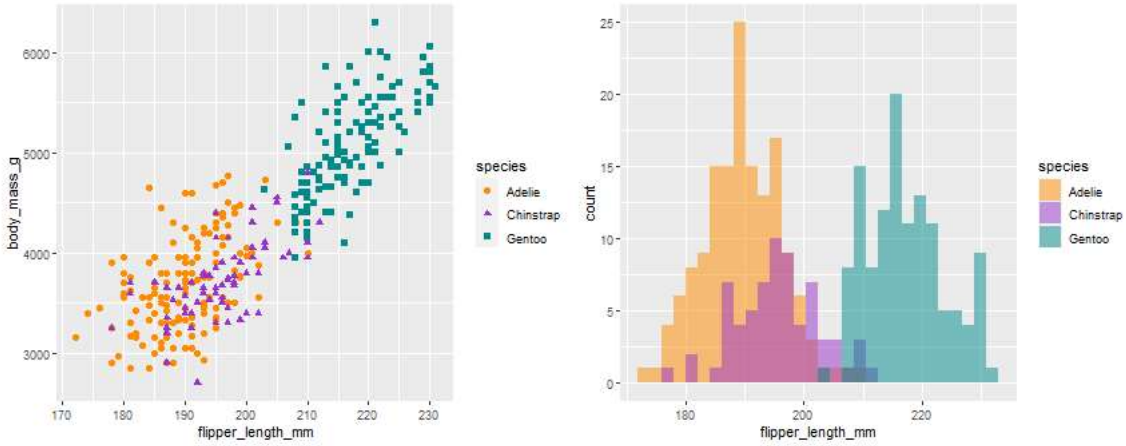
ggplot2 extensions: https://exts.ggplot2.tidyverse.org/



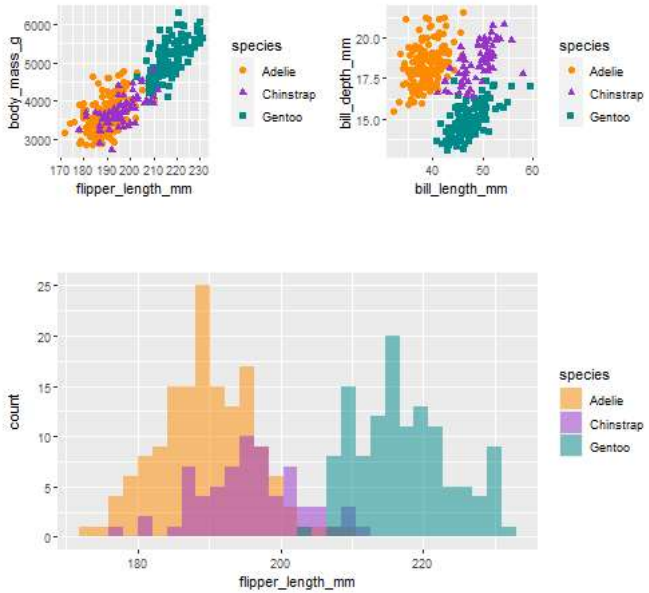
1. patchwork for plot composition

```r
p1 <-  ggplot(data = penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(color = species, shape = species), size = 2) +
  scale_color_manual(values = c("darkorange","darkorchid","cyan4")) +
  theme(aspect.ratio = 1)

p2 <-  ggplot(data = penguins, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point(aes(color = species, shape = species), size = 2)  +
  scale_color_manual(values = c("darkorange","darkorchid","cyan4")) +
    theme(aspect.ratio = 1)

p3 <- ggplot(data = penguins, aes(x = flipper_length_mm)) +
  geom_histogram(aes(fill = species), alpha = 0.5, position = "identity") +
  scale_fill_manual(values = c("darkorange","darkorchid","cyan4"))
```

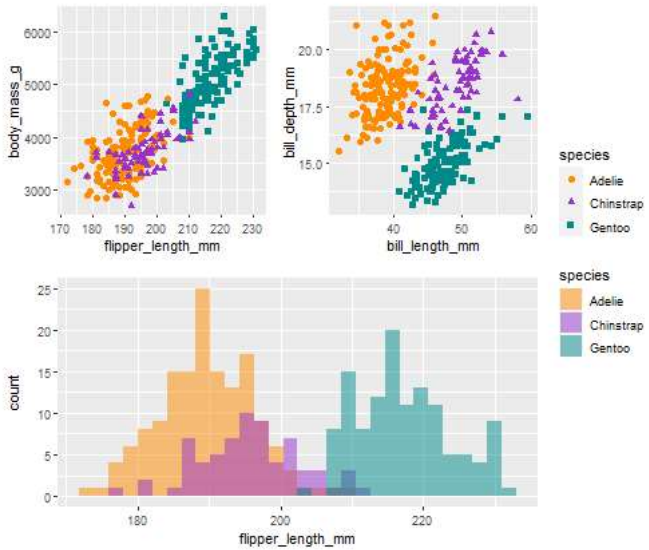
```
library(patchwork)
p1 + p3
```



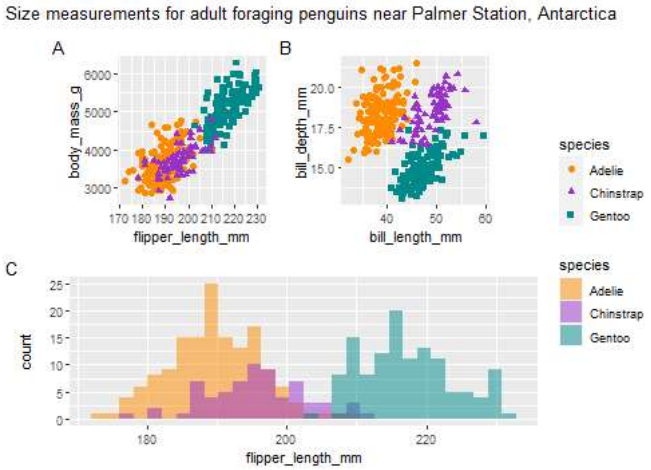
```
library(patchwork)
(p1 | p2) / p3
```



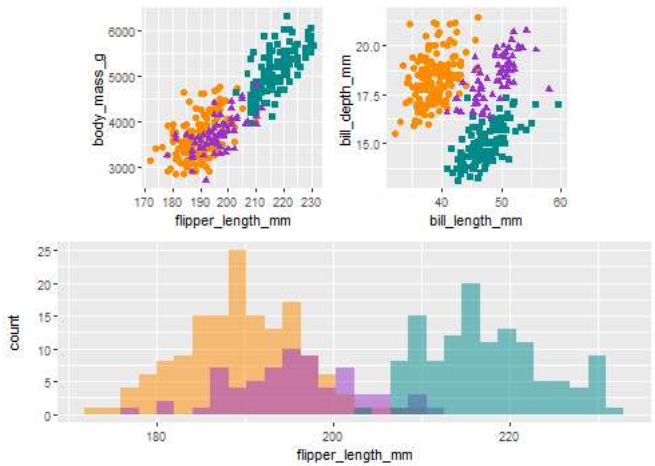
```
library(patchwork)
p <- (p1 | p2) / p3

p + plot_layout(guide = 'collect')
```



```r
library(patchwork)
p <- (p1 | p2) / p3

p +
  plot_layout(guide = 'collect') +
  plot_annotation(
    title = 'Size measurements for adult foraging penguins near Palmer Station, Antarctica',
    tag_levels = 'A')
```



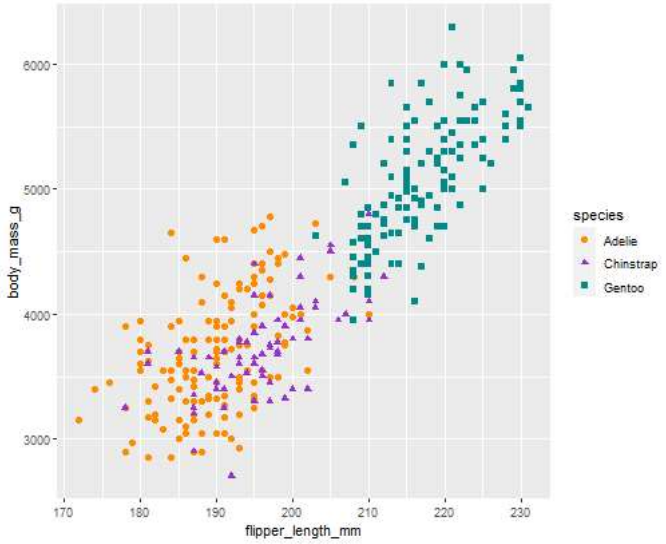
```r
library(patchwork)
p <- (p1 | p2) / p3

p &
  theme(legend.position =  'none')
```



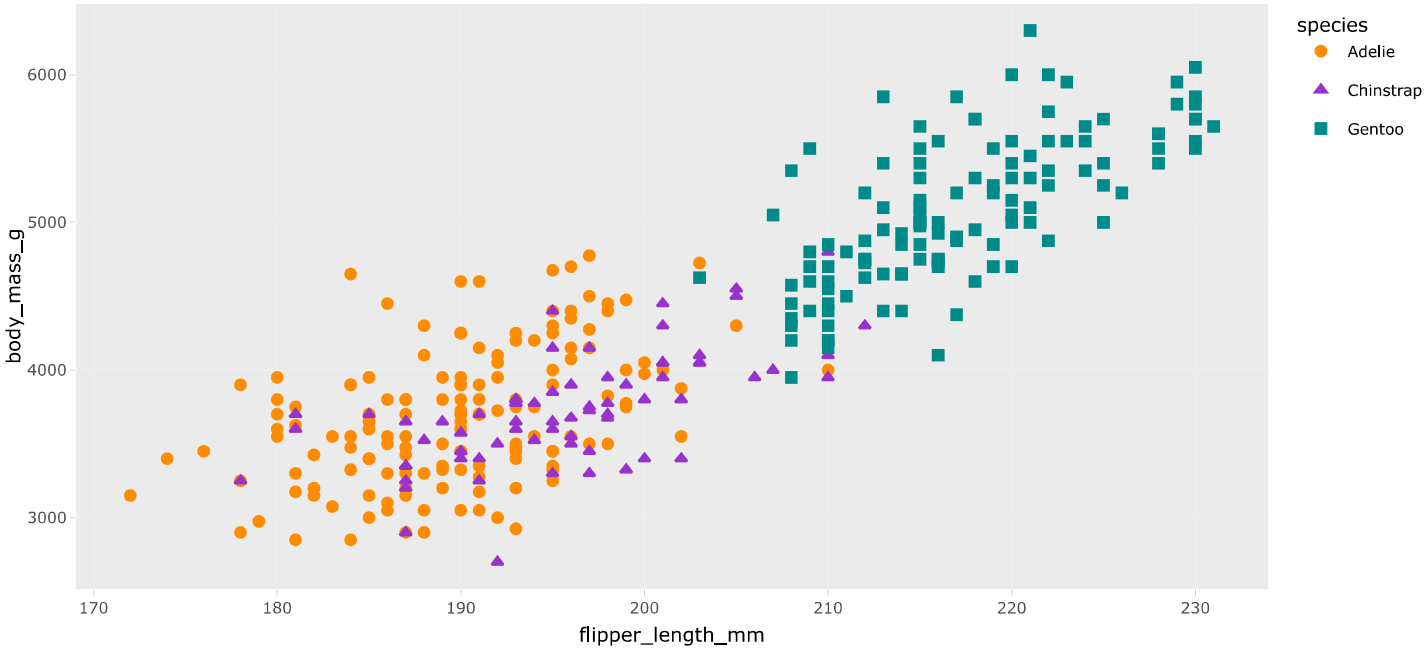
2. plotly

An R package for creating **interactive web graphics** via the open source JavaScript graphing library plotly.js.

```
p1 ## a ggplot object
```

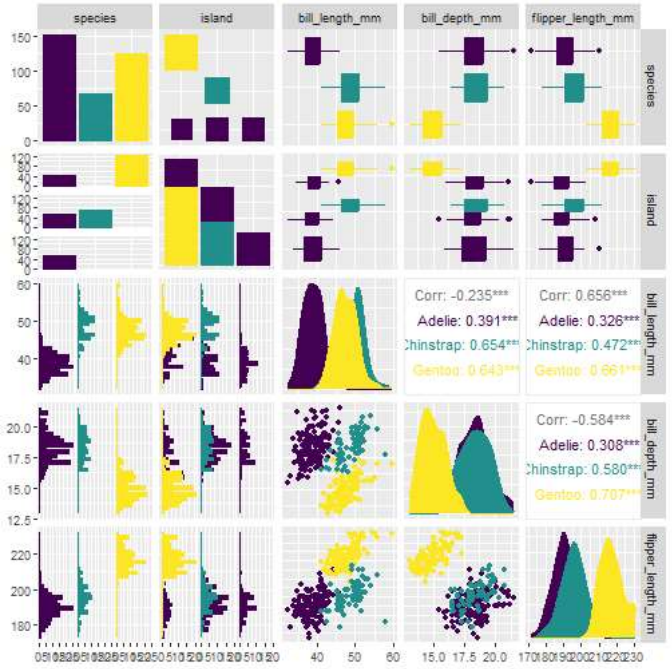


```
plotly::ggplotly(p1)
```

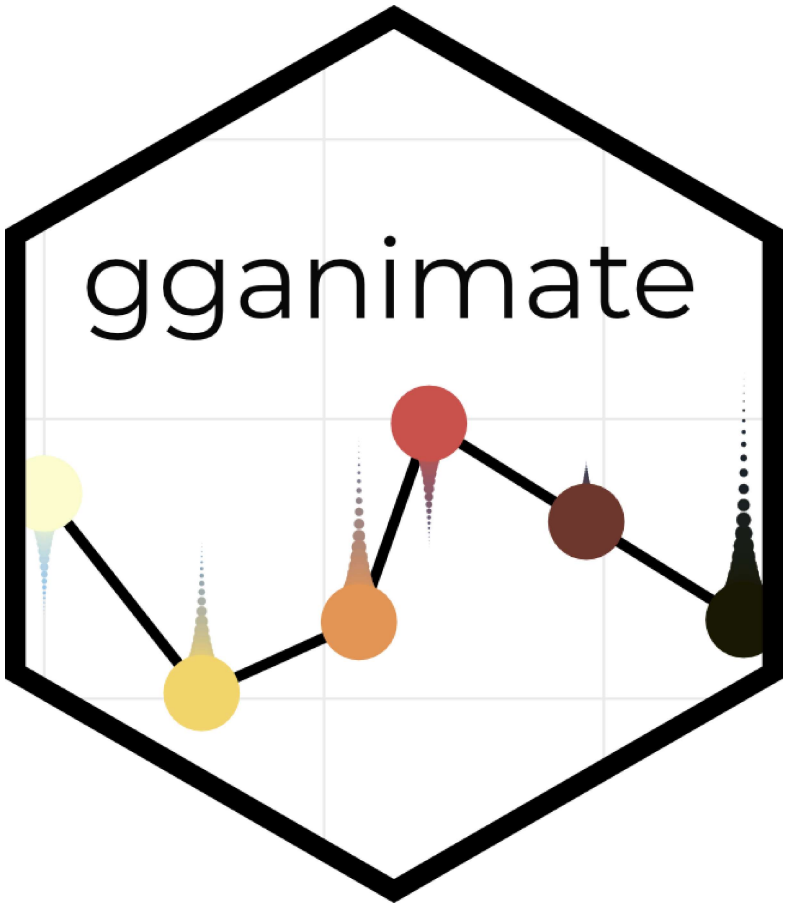


3. GGally

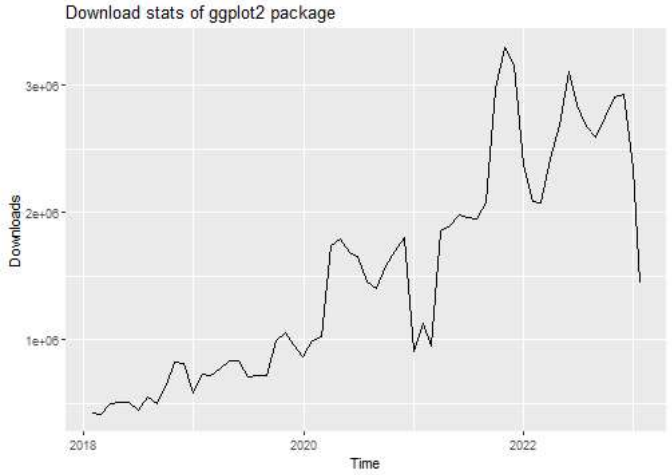
```
GGally::ggpairs(penguins[, 1:5], aes(color = species, fill = species))+
  scale_color_viridis_d() +
  scale_fill_viridis_d()
```



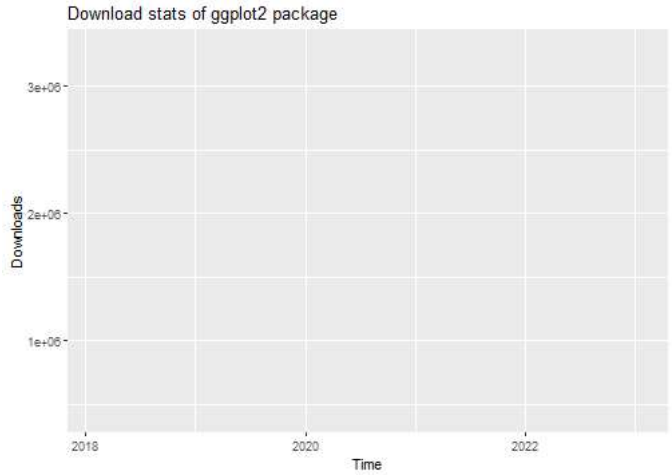
4. gganimate



```r
library("ggplot2")
library("dlstats")

data <- cran_stats("ggplot2")

p <- ggplot(data, aes(x= end, y = downloads)) +
  geom_line() +
  labs(title = "Download stats of ggplot2 package", x = "Time", y = "Downloads")

p
```

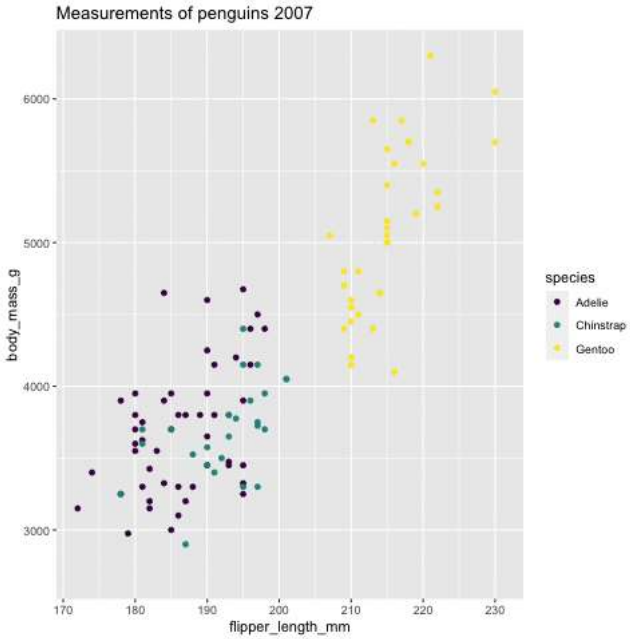


```
library(gganimate)
p +
  transition_reveal(along = end)
```



- Sometimes you might need to install the `png` and `gifski` packages and restart the R-Studio.

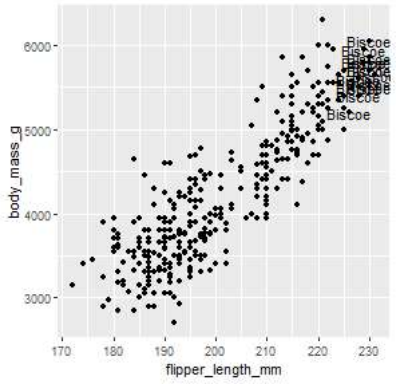
```
p <- ggplot(penguins, aes(flipper_length_mm, body_mass_g , color = species)) +
  geom_point() + scale_color_viridis_d() +
  labs(title = "Measurements of penguins {closest_state}")+
  transition_states(states = year) + enter_grow() + exit_fade()
p
```



5. ggrepel

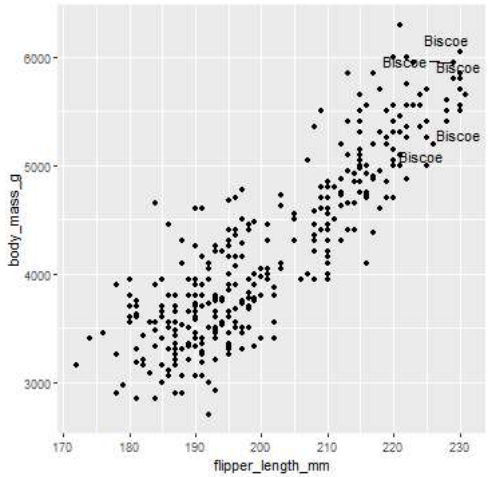
Text annotation

```
df <-  penguins |>
  filter( flipper_length_mm > 225 )

ggplot(penguins, aes(x=flipper_length_mm, y= body_mass_g))+
  geom_point()+
  theme(aspect.ratio = 1) +
  geom_text(data= df,
            aes(x=flipper_length_mm, y= body_mass_g, label= island))
```



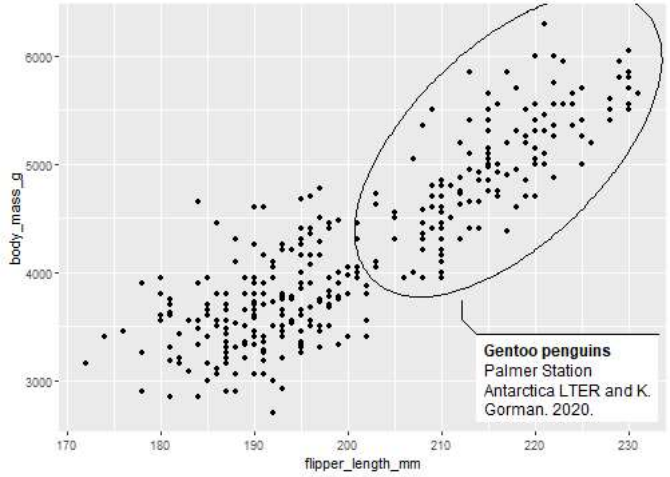
Text annotation

```
ggplot(penguins, aes(x=flipper_length_mm, y= body_mass_g))+
  geom_point()+
  theme(aspect.ratio = 1) +
  ggrepel::geom_text_repel(data= df,
            aes(x=flipper_length_mm, y= body_mass_g, label= island))
```

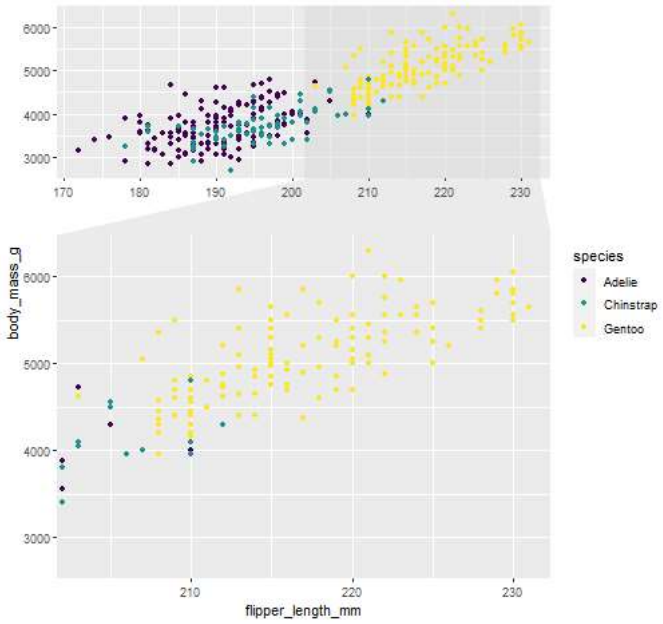


6. ggforce

```r
library(ggforce)

penguins <- penguins |> drop_na()
p <- ggplot(penguins, aes(x=flipper_length_mm, y= body_mass_g))+
  geom_mark_ellipse(aes(
    filter = species == "Gentoo",
    label = 'Gentoo penguins'),
    description = 'Palmer Station Antarctica LTER and K. Gorman. 2020.') +
  geom_point()

p
```



```
library(ggforce)
ggplot(penguins, aes(x=flipper_length_mm, y= body_mass_g, color = species)) +
    geom_point() +
  scale_color_viridis_d() +
    facet_zoom(x = species == "Gentoo")
```



Key References

- ggplot2: Elegant Graphics for Data Analysis https://ggplot2-book.org/
- ggplot2 workshop by Thomas Lin Pedersen https://www.youtube.com/watch?v=h29g21z0a68