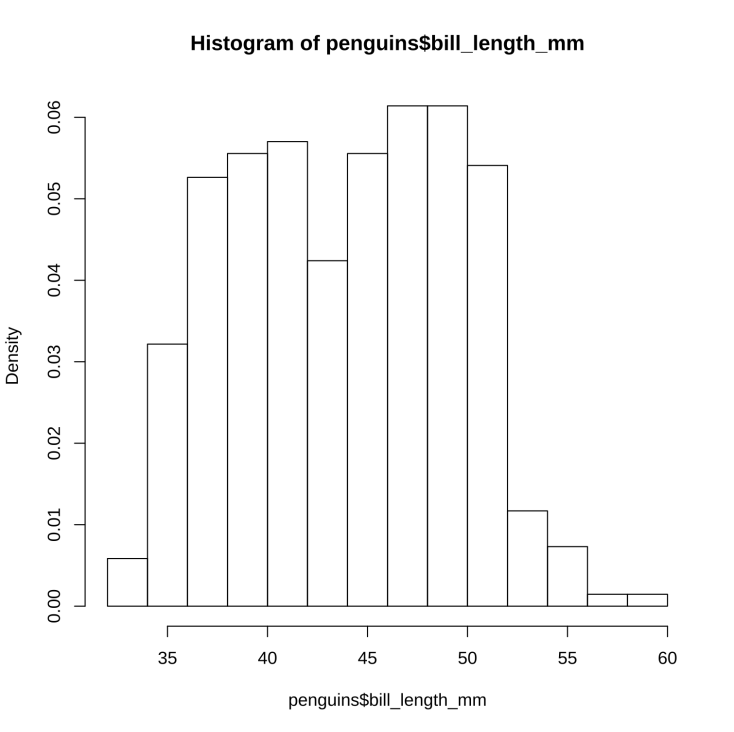
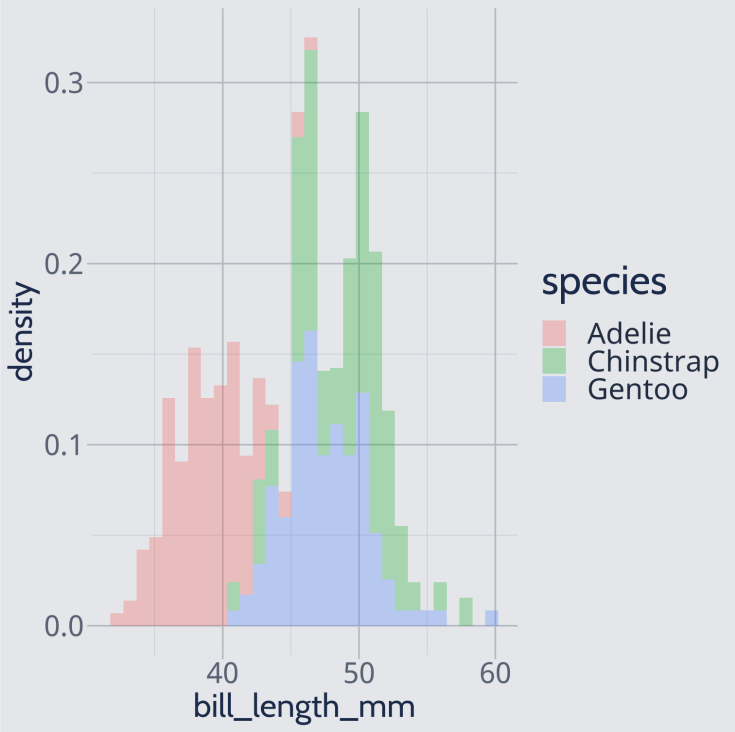
Data Visualisation with ggplot2

Felix Zaussinger

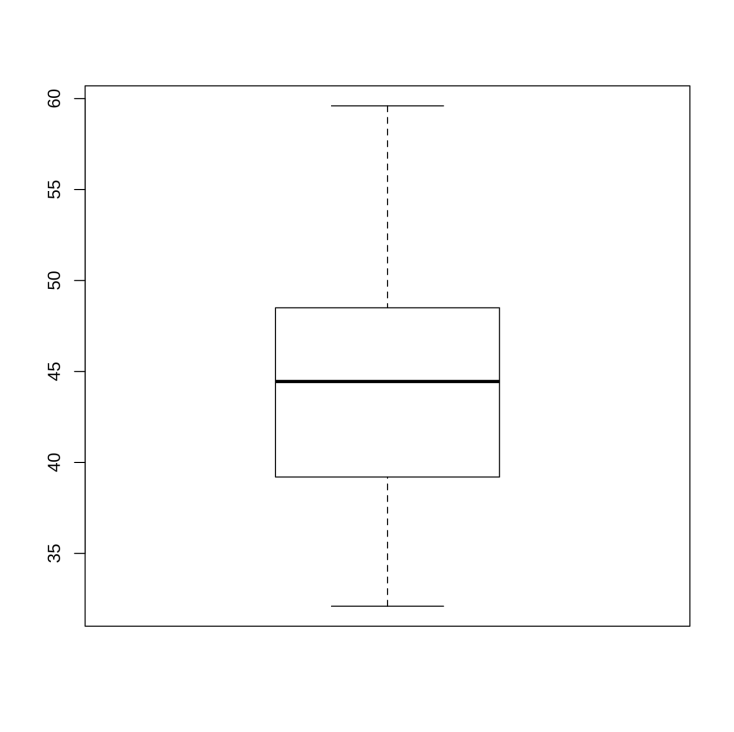
10.09.2020

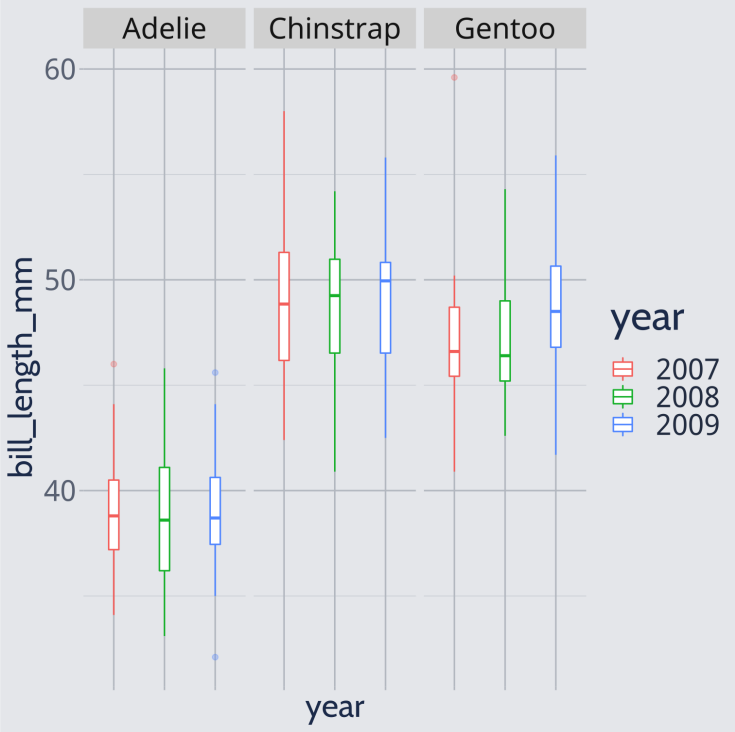
Motivation


Histogram of penguins$bill_length_mm
Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
35
40
45
50
55
60
penguins$bill_length_mm


density
0.0
0.1
0.2
0.3
40
50
60
bill_length_mm
species
Adelie
Chinstrap
Gentoo

Motivation


60
55
50
45
40
35


Adelie
Chinstrap
Gentoo
60
50
40
bill_length_mm
year
year
2007
2008
2009

ggplot2

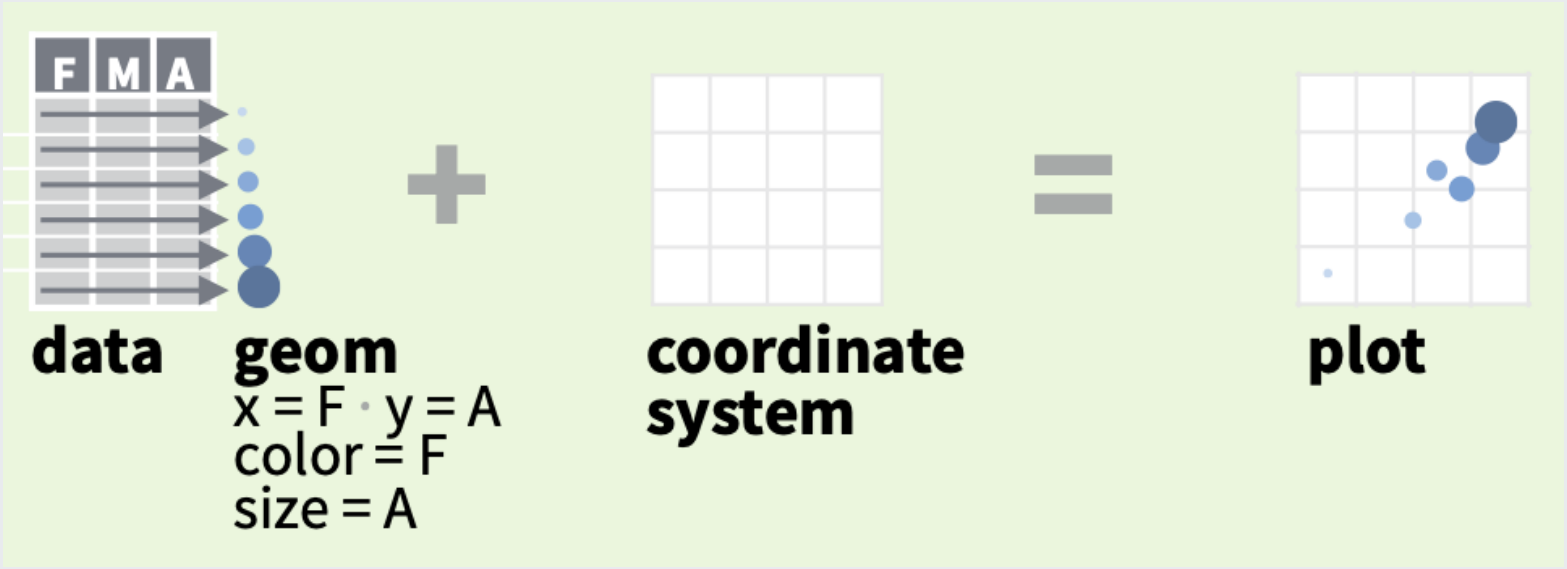
"The grammar of graphics" -> 3 components make a graph

- data
- coordinate system
- geometries ("geoms"): visual marks representing data points

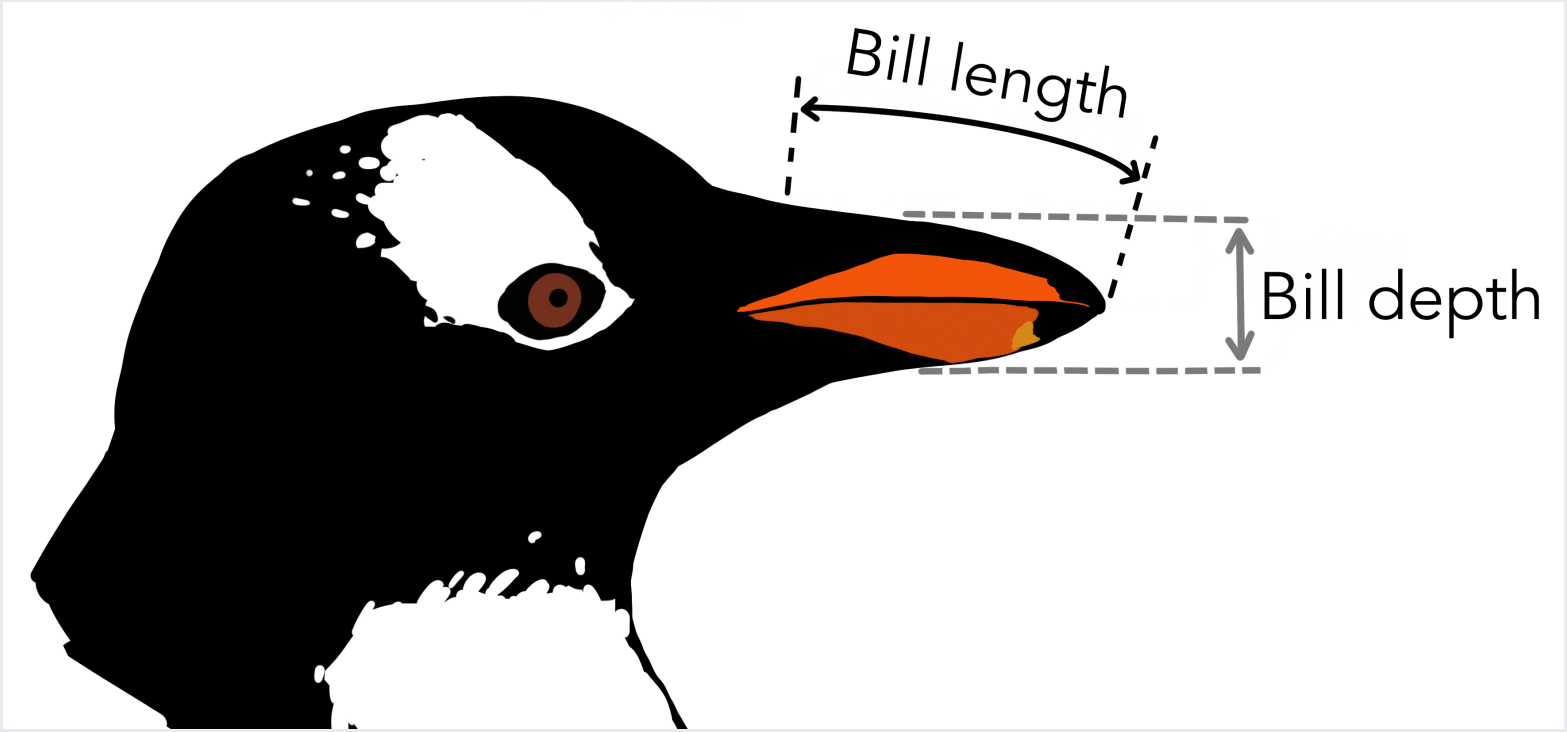
ggplot2

geom's have properties -> "aesthetics"

- x, y
- color
- size



Visualisation practice



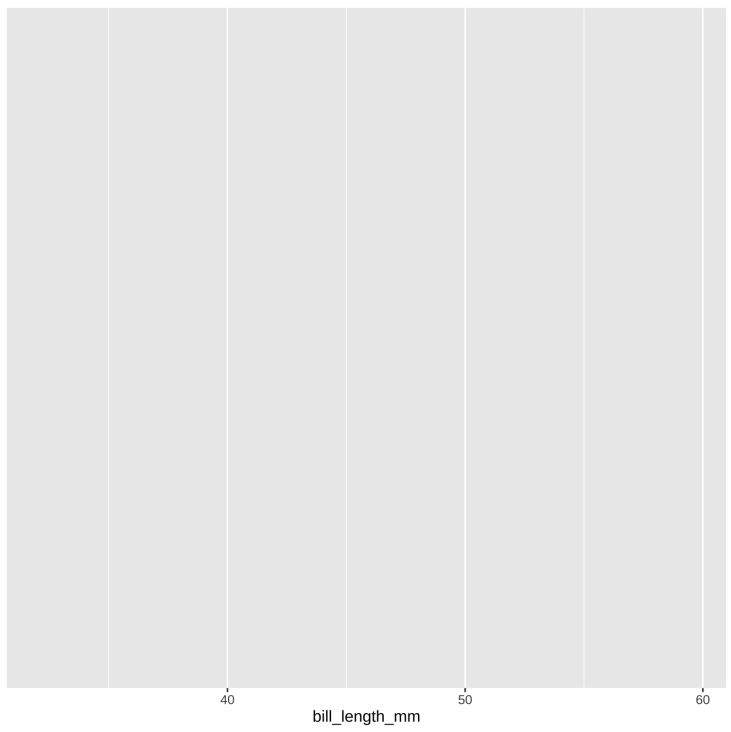
(Artwork by @allison_horst, Data from https://github.com/allisonhorst/palmerpenguins)

1) Data

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

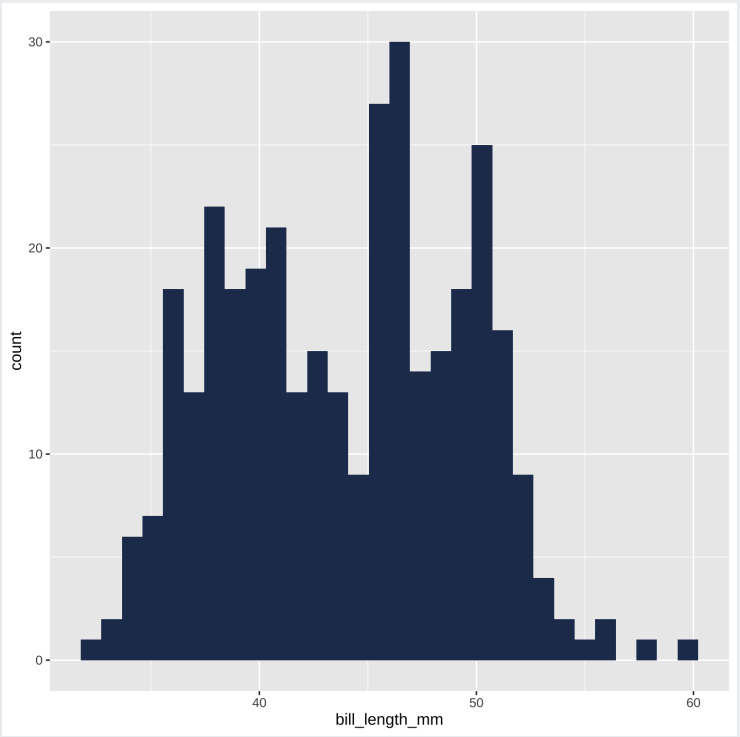
2) Coordinate System

```
ggplot(data=penguins) +
  aes(x=bill_length_mm)
```



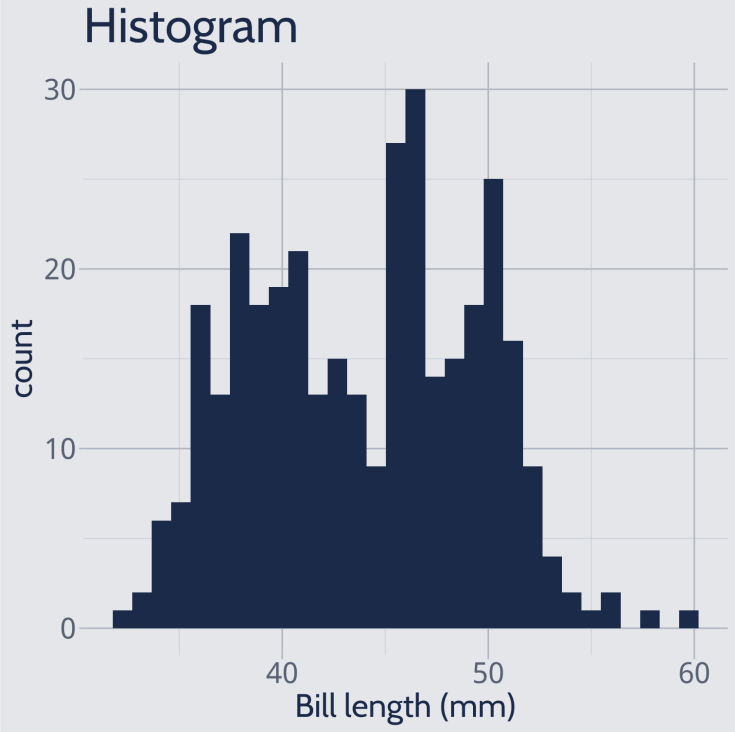
3) Geometry

```
ggplot(data=penguins) +
  aes(bill_length_mm) +
  geom_histogram()
```



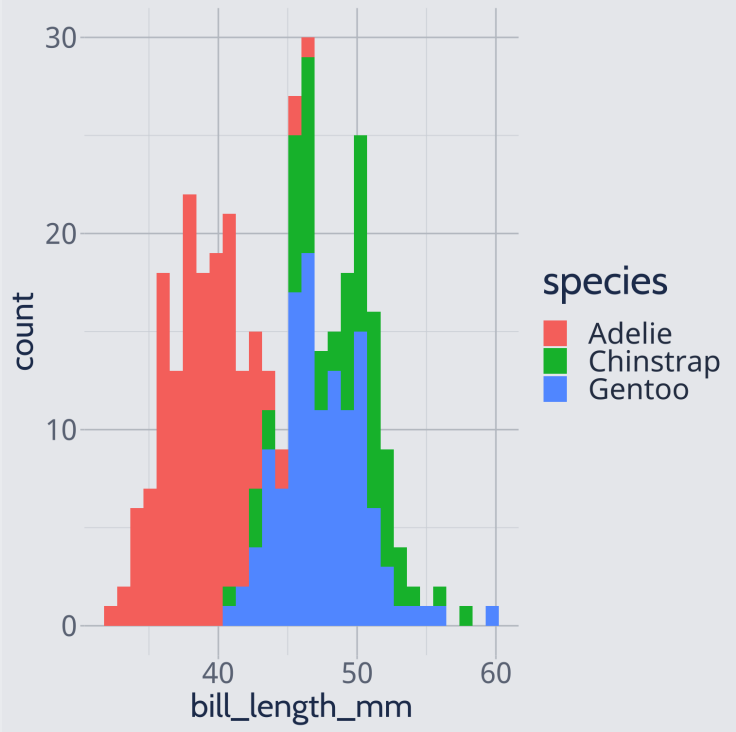
... labeling

```
ggplot(data=penguins) +
  aes(bill_length_mm) +
  geom_histogram() +
  labs(x="Bill length (mm)",
       title="Histogram") +
  theme_xaringan()
```



Distinguishing species via colors

```
ggplot(data=penguins) +
  aes(bill_length_mm) +
  geom_histogram(
    aes(fill = species)
    ) +
  theme_xaringan()
```



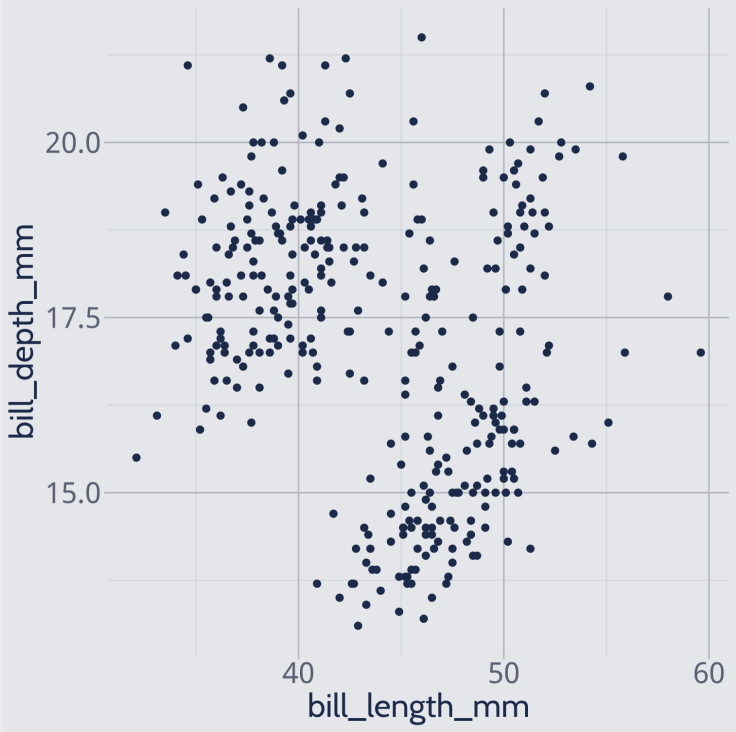
Recap 1

- a trio of **data** + **coordinate system** + **geometries** makes a ggplot
- certain **properties** can be assigned to **geometries** via **aes()**
- we can create plots through applying a **logical sequence of commands connected by + signs**
- Histograms are created with via *geom_histogram*
- Labels can be assigned with via *labs*
- within *aes()*, the *fill* property can be used to **distinguish different categories in your data set**

Any questions so far?

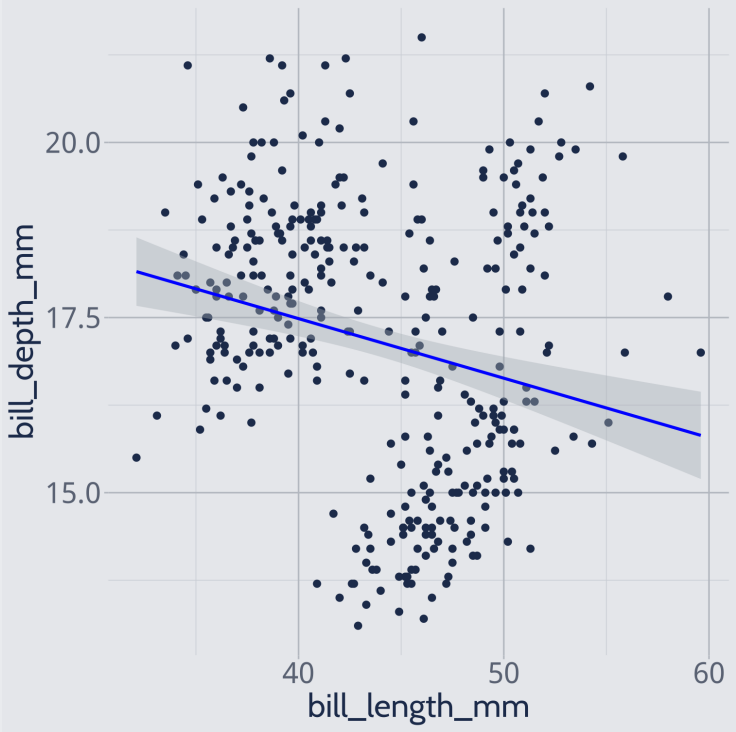
Scatterplot

```
ggplot(data = penguins) +
  aes(x = bill_length_mm,
      y = bill_depth_mm) +
  geom_point(size = 2) +
  theme_xaringan()
```



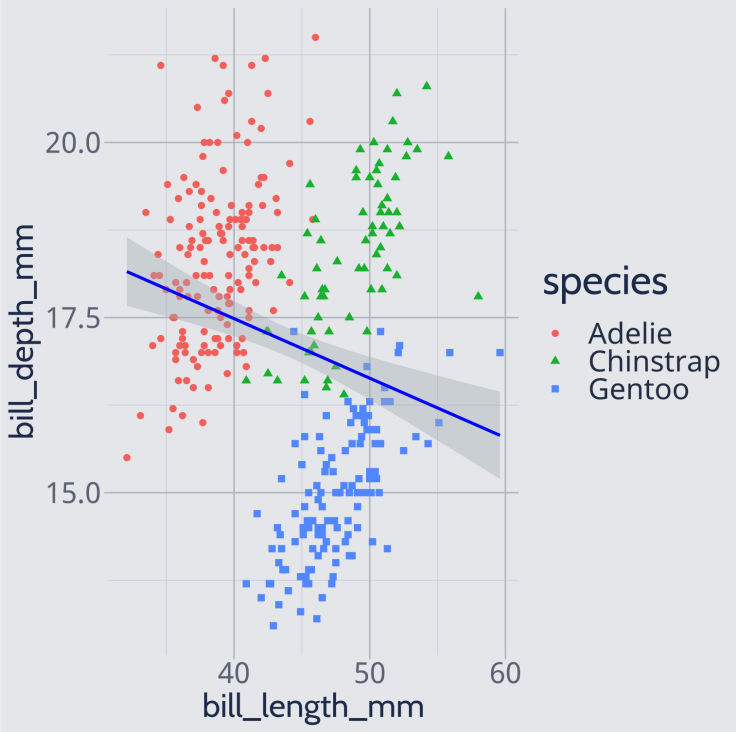
Add a linear regression line

```
ggplot(data = penguins) +
  aes(x = bill_length_mm,
      y = bill_depth_mm) +
  geom_point(size = 2) +
  geom_smooth(method="lm",
              color="blue") +
  theme_xaringan()
```



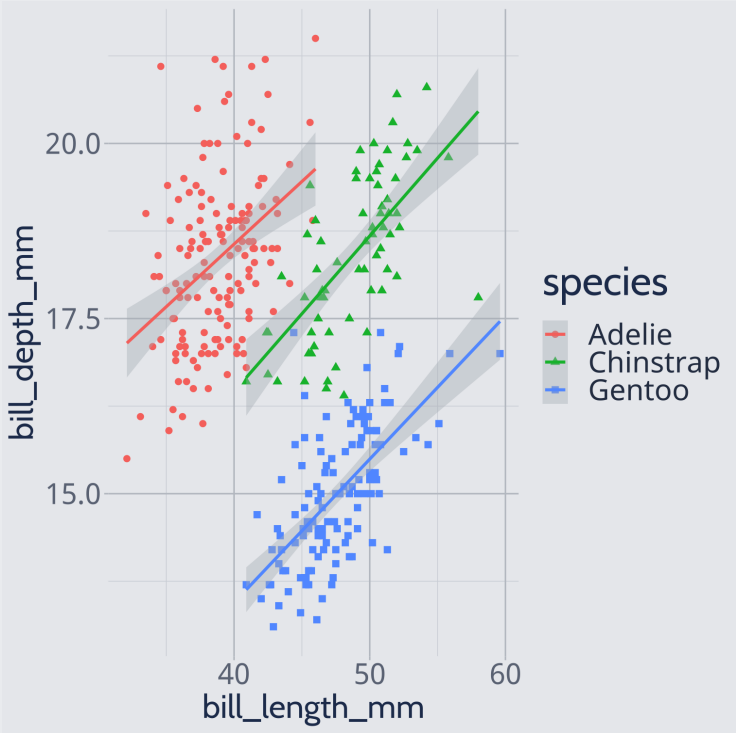
Distinguish species with colors

```
ggplot(data = penguins) +
  aes(x = bill_length_mm,
      y = bill_depth_mm) +
  geom_point(
    aes(color = species,
        shape = species),
    size = 2) +
  geom_smooth(method="lm",
              color="blue") +
  theme_xaringan()
```



Category-specific regression lines

```
ggplot(data = penguins) +
  aes(x = bill_length_mm,
      y = bill_depth_mm) +
  geom_point(
    aes(color = species,
        shape = species),
    size = 2
    ) +
  geom_smooth(
    method = "lm",
    se = TRUE,
    aes(color = species)
    ) +
  theme_xaringan()
```



Simpson's paradox: https://en.wikipedia.org/wiki/Simpson%27s_paradox

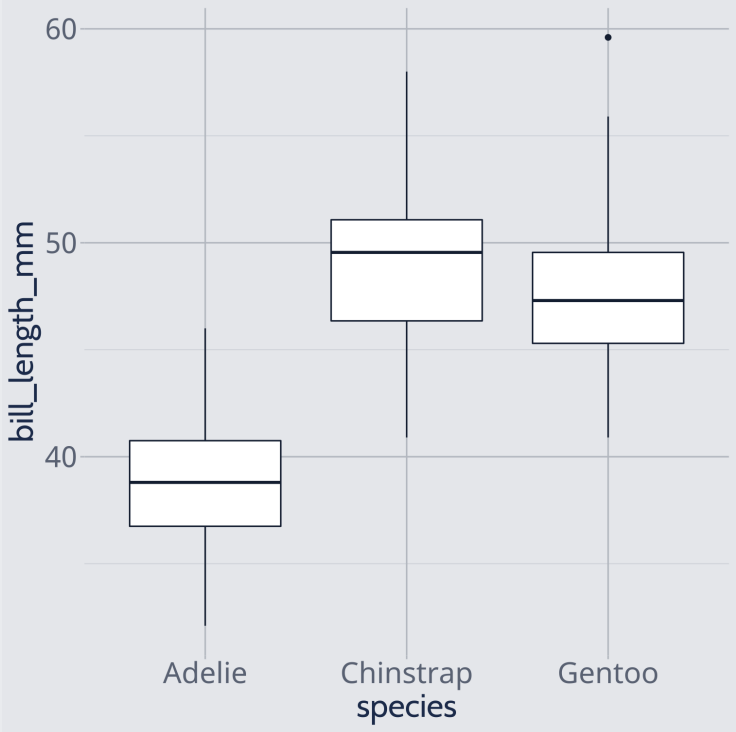
Recap 2

- **Scatterplots** are created via *geom_point*
- **Regression lines** can be fit to the data points via *geom_smooth*. We learned about the *'lm'* method, but many other (non-linear) methods are available.
- Within *aes()*, the *color* and *shape* properties can be used to distinguish categories in your data
- **Category-specific regression lines** can be fitted by specifying the category in *aes()*
- A bit off-topic, but important: Unraveling **categorical clusters** in your data is crucial for gaining valid insights (*Simpson's paradox*)

Questions?

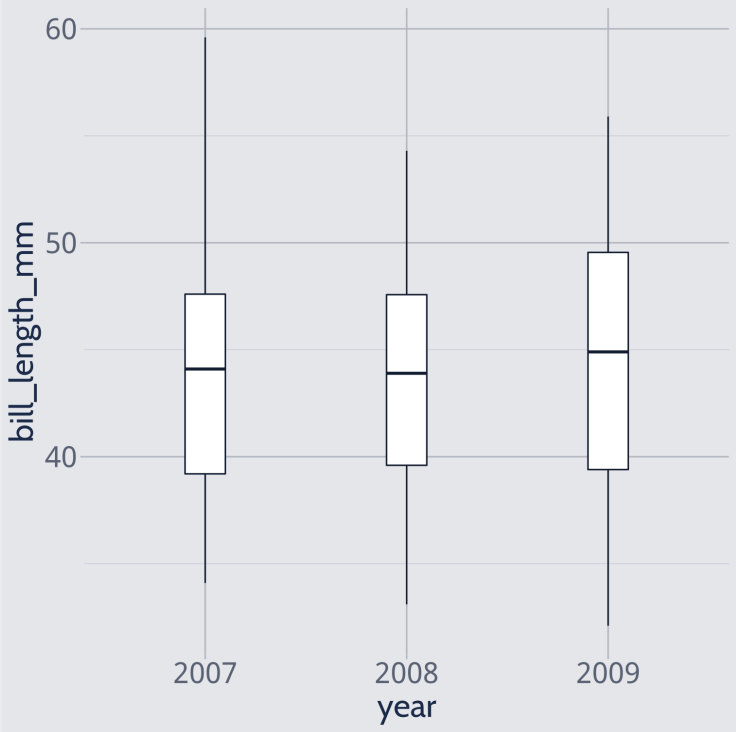
Boxplot: x = species

```
ggplot(data = penguins) +
  aes(x = species,
      y = bill_length_mm) +
  geom_boxplot() +
  theme_xaringan()
```



Boxplot: x = year

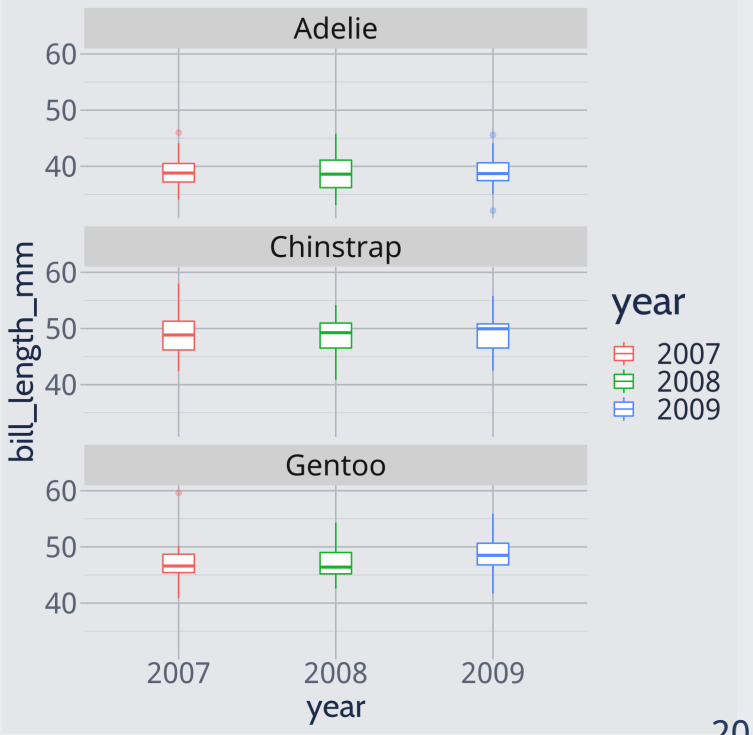
```
ggplot(data = penguins) +
  aes(x = year,
      y = bill_length_mm) +
  geom_boxplot(
    aes(group=year),
    width=0.2,
    outlier.alpha = 0.3) +
  theme_xaringan()
```



What if we want to visualise both?

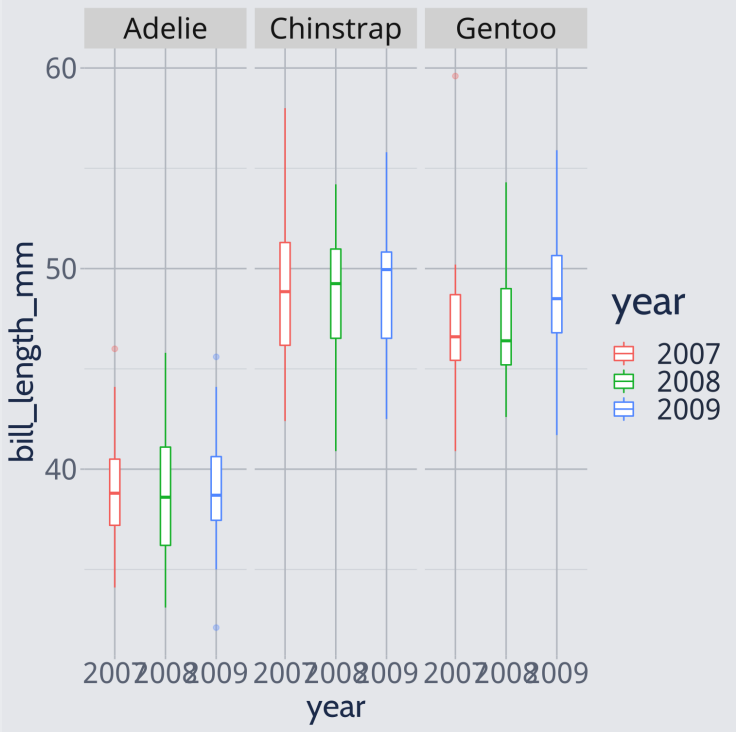
Facetting: building multi-panel plots via *facet_wrap*

```
ggplot(data = penguins) +
  aes(x = year,
      y = bill_length_mm) +
  geom_boxplot(
    aes(group=year,
        color=year),
    width=0.2,
    outlier.alpha = 0.3) +
  facet_wrap(
    vars(species),
    ncol=1) +
  theme_xaringan()
```



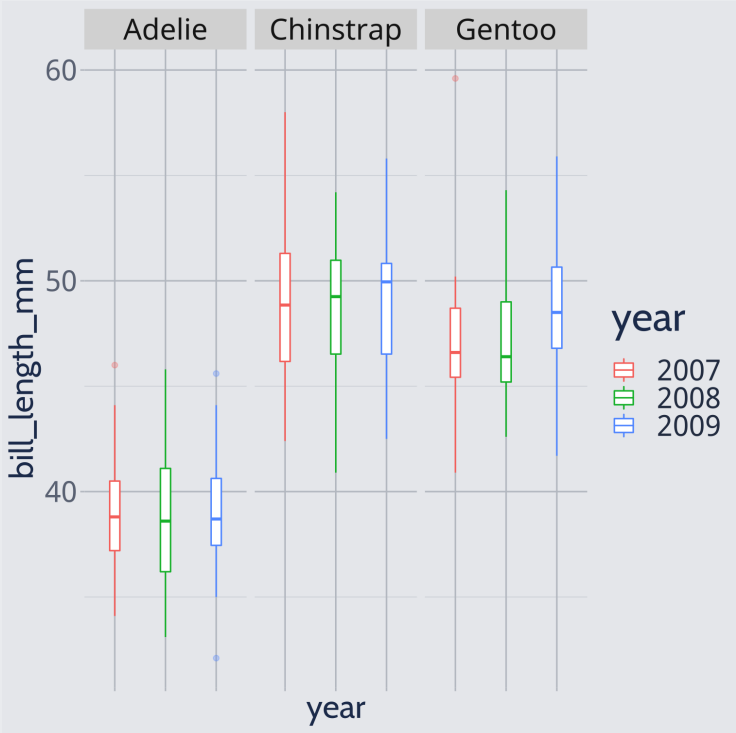
facet_wrap with 3 columns

```
ggplot(data = penguins) +
  aes(x = year,
      y = bill_length_mm) +
  geom_boxplot(
    aes(group=year,
        color=year),
    width=0.2,
    outlier.alpha = 0.3) +
  facet_wrap(
    vars(species),
    ncol=3) +
  theme_xaringan()
```



Removing x-labels for beautification

```
ggplot(data = penguins) +
  aes(x = year,
      y = bill_length_mm) +
  geom_boxplot(
    aes(group=year,
        color=year),
    width=0.2,
    outlier.alpha = 0.3) +
  facet_wrap(
    vars(species),
    ncol=3) +
  theme_xaringan() +
  theme(axis.text.x
        =element_blank())
```



Recap 3

- **Boxplots** are created via *geom_boxplot*. We need to specify *x* and *y* within *aes()* for R to know which data to plot
- We can change the width of a geometry via *width* and change the opacity of outliers in *geom_boxplot* via *outlier.alpha*
- **facet_wrap** is a powerful command that let's us create multiple panels called *facets* for different units in a category
- We can hide the x-labels of a plot by calling *theme(axis.text.x=element_blank())*.

Questions?

Enough said...