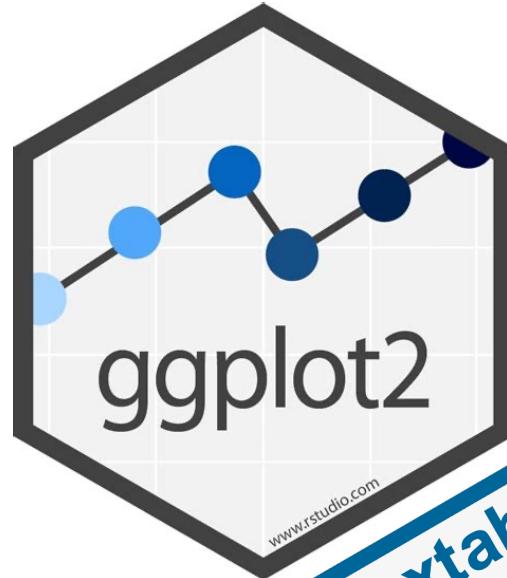
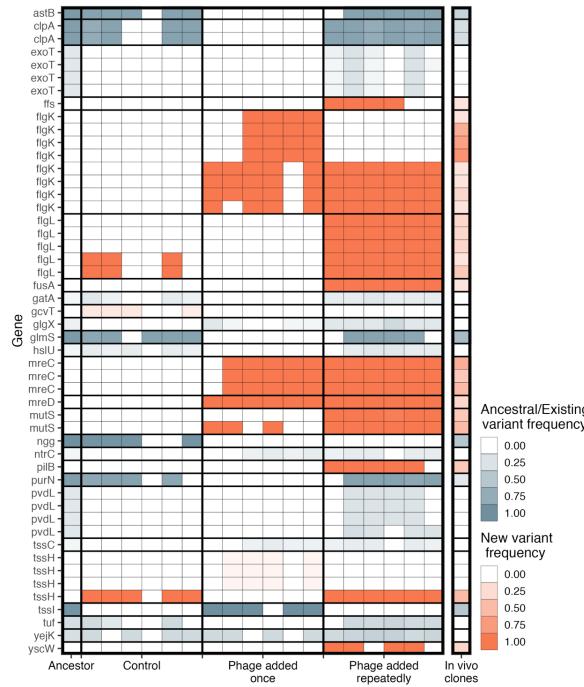


Reproducible publication-ready plots and tables with ggplot2 and flextable



Treatment	Time contrast	Odds-ratio	SE	z-ratio	p-value
Phage 1	1 / 2	0.0866	0.0579	-3.66	0.000747
Phage 1	1 / 3	0.255	0.173	-2.02	0.108
Phage 1	2 / 3	2.94	1.51	2.1	0.0899
Phage 2	1 / 2	0.00000000812	0.0000354	-0.00427	1
Phage 2	1 / 3	0.00000000571	0.0000249	-0.00435	1
Phage 2	2 / 3	0.703	0.432	-0.573	0.835
Phage 1 and 2	1 / 2	0.0648	0.0538	-3.3	0.00282
Phage 1 and 2	1 / 3	0.0187	0.0157	-4.76	0.00000583
Phage 1 and 2	2 / 3	0.289	0.124	-2.9	0.0103



Daniel Padfield

d.padfield@exeter.ac.uk

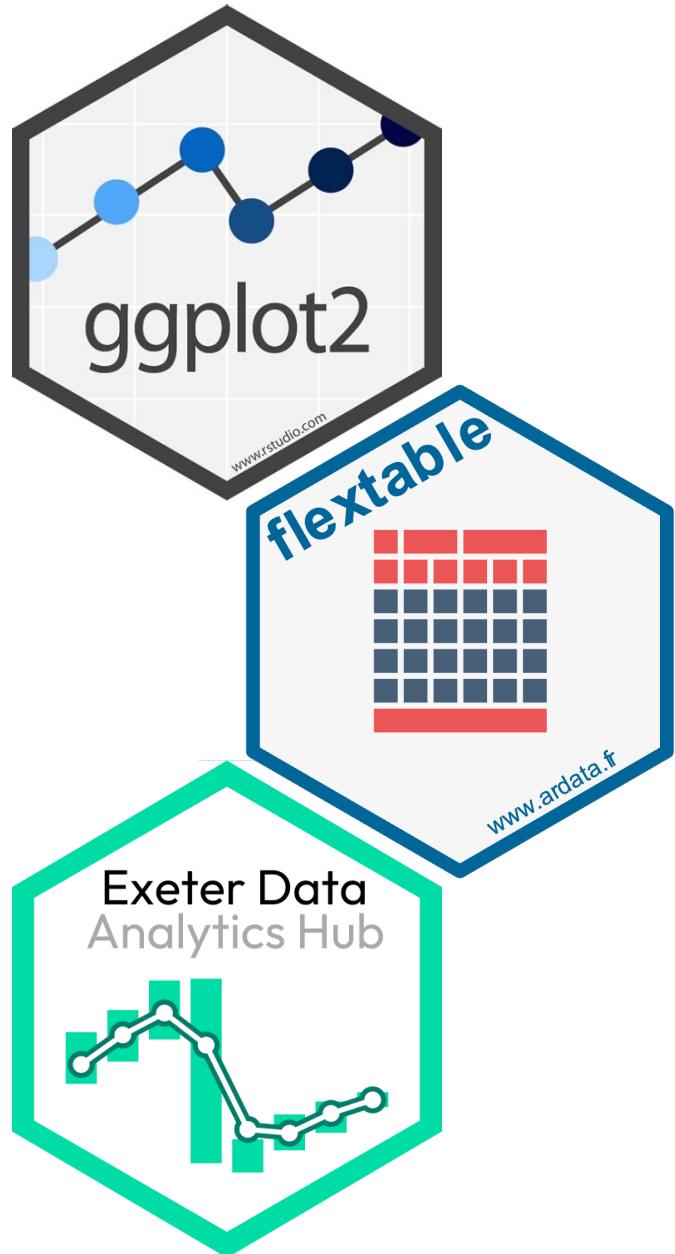
padpadpadpad



University
of Exeter

Housekeeping

- Have brought your laptop and a plot or table you want to make.
- Please interrupt me if you have questions (I may not have all the answers).
- You may know more than me! Please share your knowledge if you have it.
- Want to do something in **flextable** or **ggplot2** and do not know how?
 - Join the ExeDataAnalytics Team on Microsoft Teams and ask in the "Ask for Help" tab.
 - Reproducible examples using reprex get you kudos:
<https://reprex.tidyverse.org/>
- Introductory walkthroughs for both **ggplot2** and **flextable** are available on our webpage.

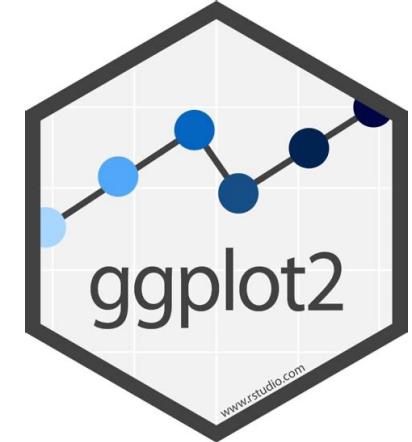


The importance of reproducible plots and tables

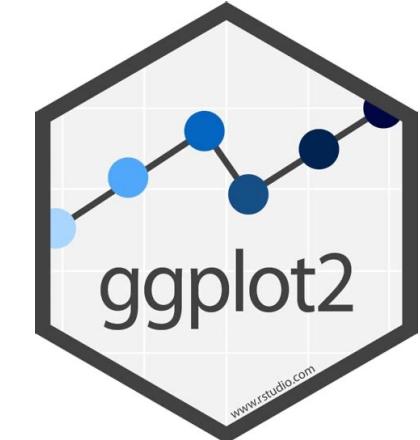
- Fewer errors
- Easier to update
- Easier to re-use
- **Can you think of any more benefits or why it is important?**

A brief introduction to ggplot2

- **ggplot2** is a system for creating graphics based on [The Grammar of Graphics](#).
- Part of the **tidyverse** packages alongside **dplyr**, **tidyr**, **purrr** etc
- One of the most popular R packages (downloaded >119m times)
- Used widely in academia, industry, and journalism.



A brief introduction to ggplot2

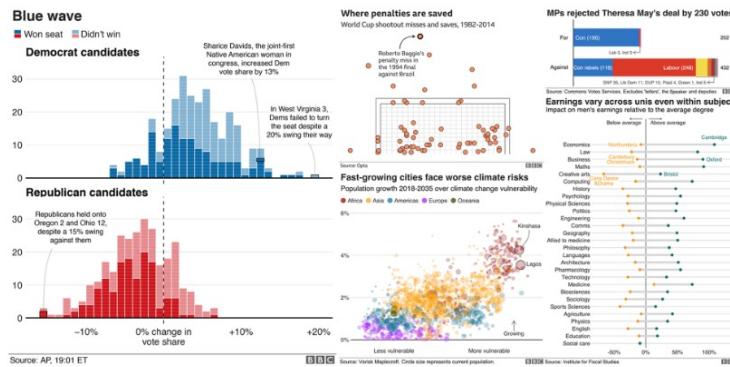


- **ggplot2** is a system for creating graphics based on [The Grammar of Graphics](#).
- Part of the **tidyverse** packages alongside **dplyr**, **tidyr**, **purrr** etc
- One of the most popular R packages (downloaded >119m times)
- Used widely in academia, industry, and journalism.

How to create BBC style graphics

At the BBC data team, we have developed an R package and an R cookbook to make the process of creating publication-ready graphics in our in-house style using R's ggplot2 library a more reproducible process, as well as making it easier for people new to R to create graphics.

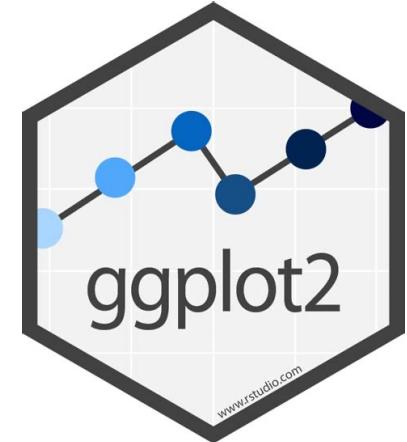
The cookbook below should hopefully help anyone who wants to make graphics like these:



We'll get to how you can put together the various elements of these graphics, but let's get the admin out of the way first...

<https://bbc.github.io/rcookbook/>

A brief introduction to ggplot2

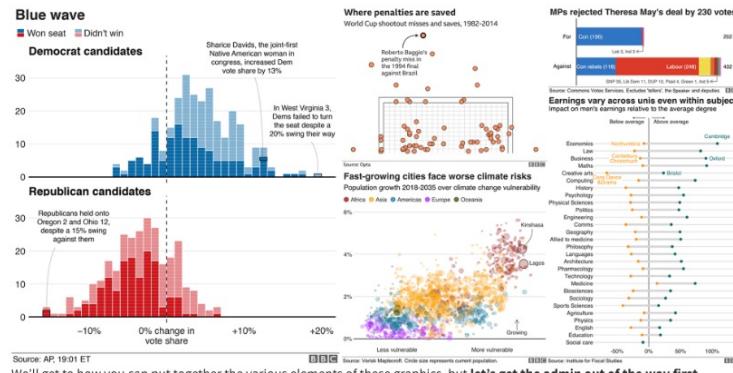


- **ggplot2** is a system for creating graphics based on [The Grammar of Graphics](#).
- Part of the **tidyverse** packages alongside **dplyr**, **tidyr**, **purrr** etc
- One of the most popular R packages (downloaded >119m times)
- Used widely in academia, industry, and journalism.

How to create BBC style graphics

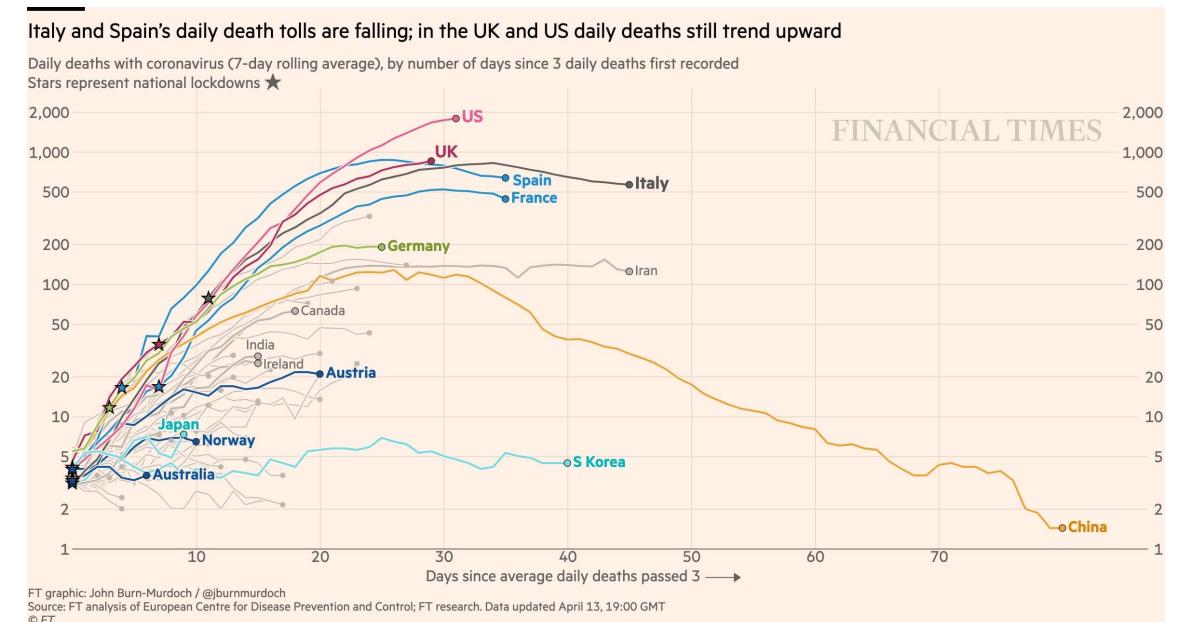
At the BBC data team, we have developed an R package and an R cookbook to make the process of creating publication-ready graphics in our in-house style using R's ggplot2 library a more reproducible process, as well as making it easier for people new to R to create graphics.

The cookbook below should hopefully help anyone who wants to make graphics like these:



We'll get to how you can put together the various elements of these graphics, but let's get the admin out of the way first...

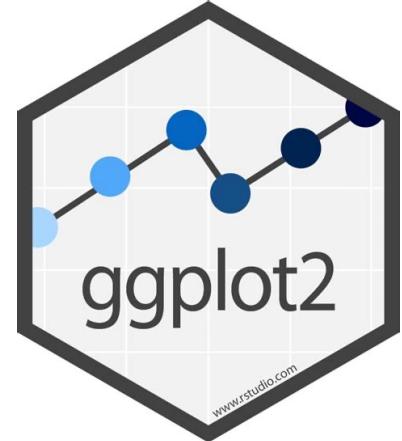
<https://bbc.github.io/rcookbook/>



John Burn Murdoch at the Financial Times

The basics of a ggplot2

Making a plot using **ggplot2** is like writing a recipe.



Ingredients

- flour
- eggs
- sugar
- butter
- chocolate

Recipe

1. Mix flour sugar eggs and butter
2. Put chocolate in
3. Bake



The basics of a ggplot2

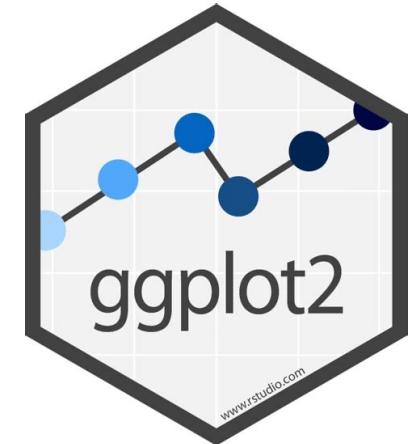
Making a plot using **ggplot2** is like writing a recipe.

Ingredients

- Empty plot
- Boxplot
- Raw data points
- Labels
- Theme
- Axis limits

Recipe

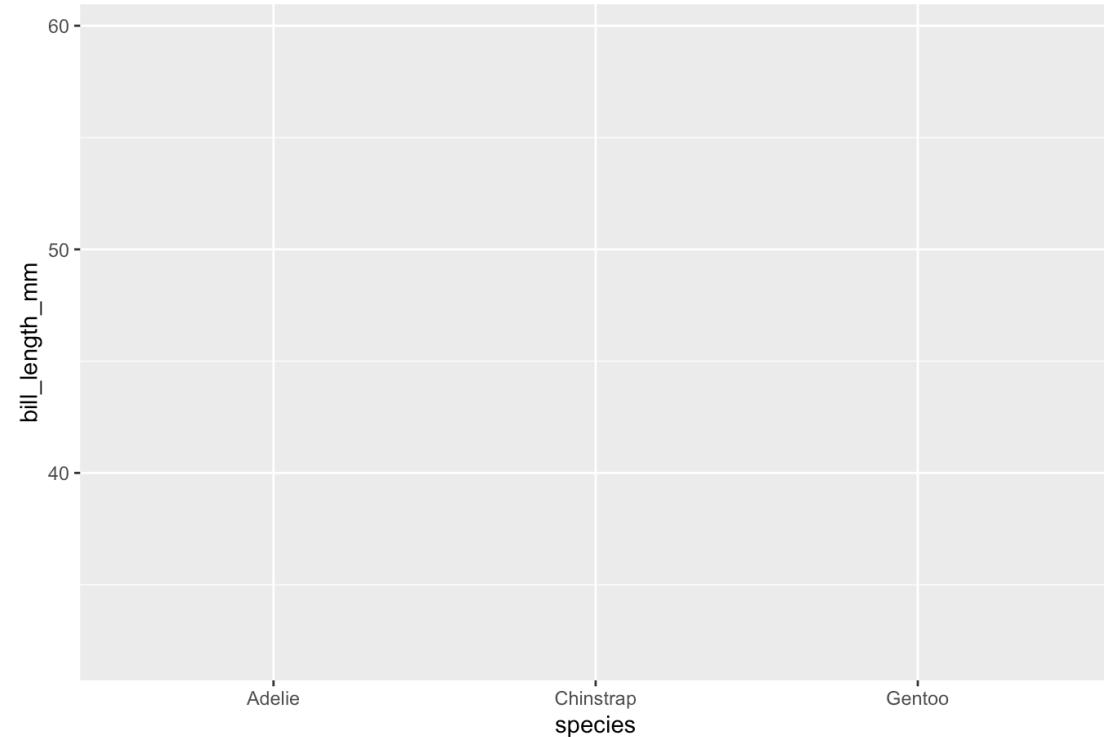
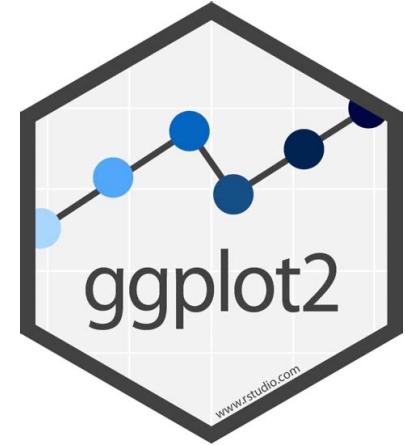
1. Start with an empty plot
2. Place the boxplot on top
3. Sprinkle over raw data points
4. Change the theme
5. Alter the axis limits



The basics of a ggplot2

Making a plot using **ggplot2** is like writing a recipe.

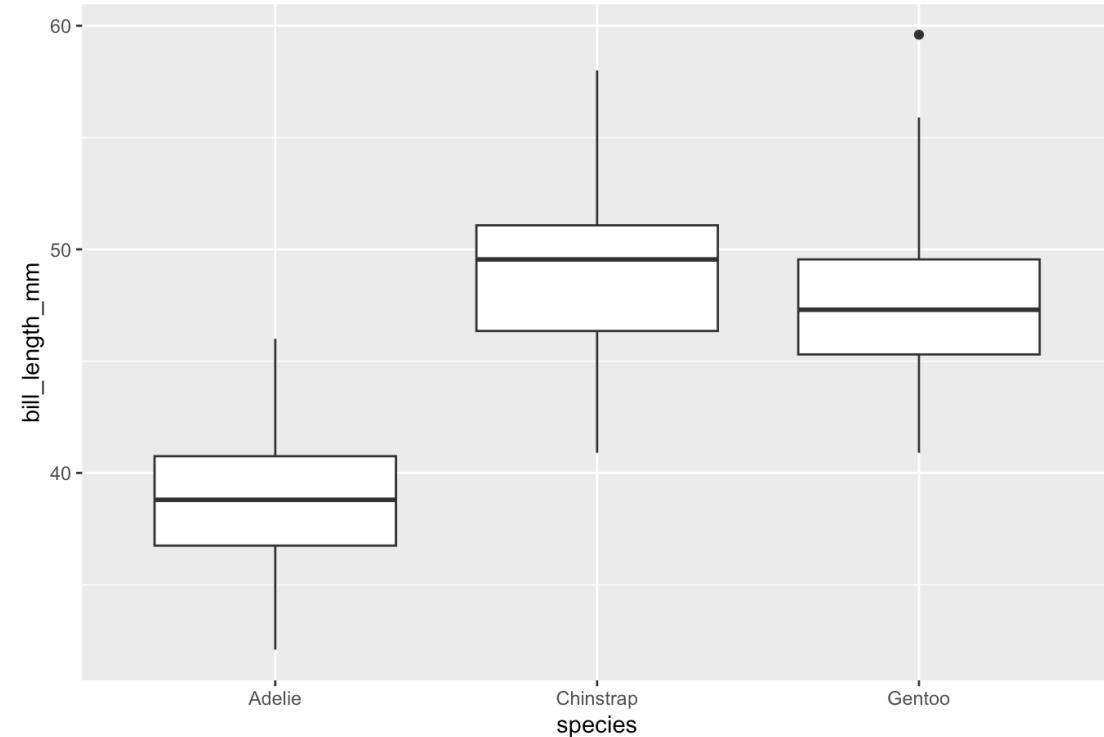
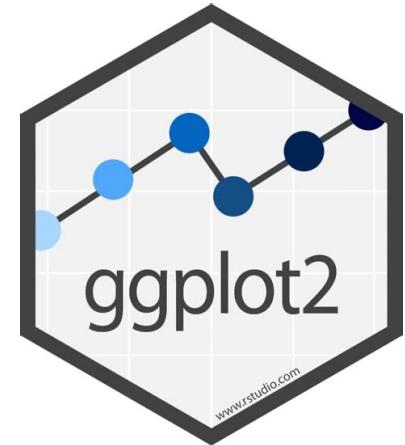
```
ggplot(penguins, aes(x=species, y=bill_length_mm))
```



The basics of a ggplot2

Making a plot using **ggplot2** is like writing a recipe.

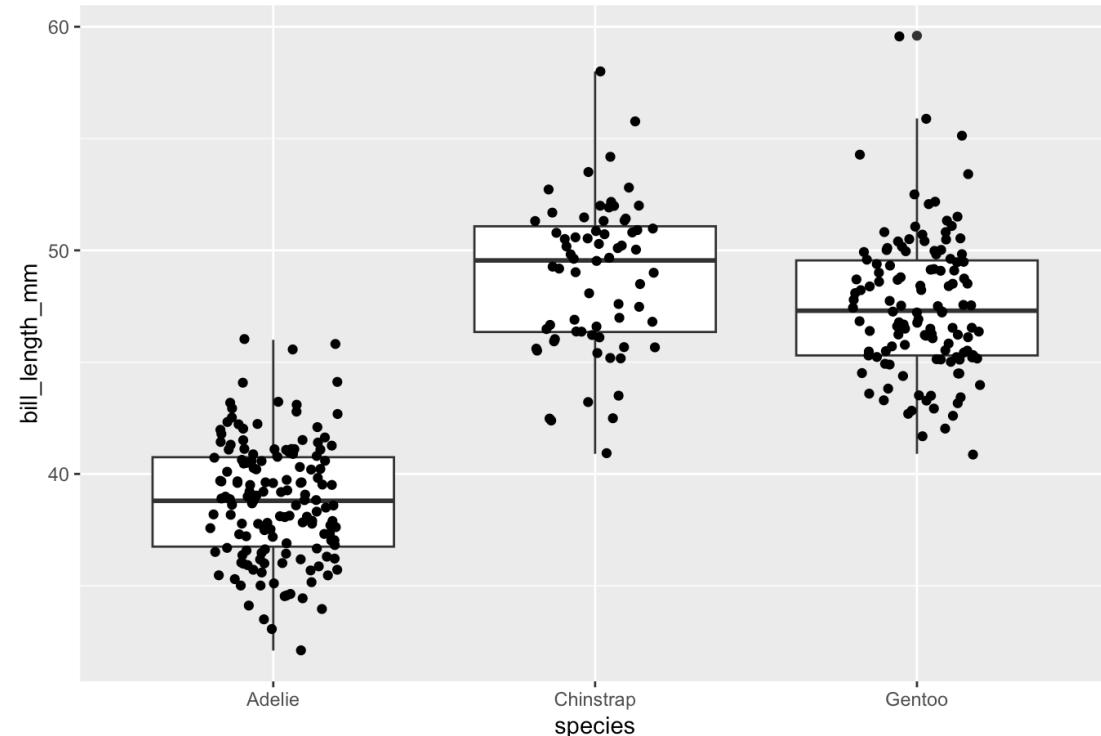
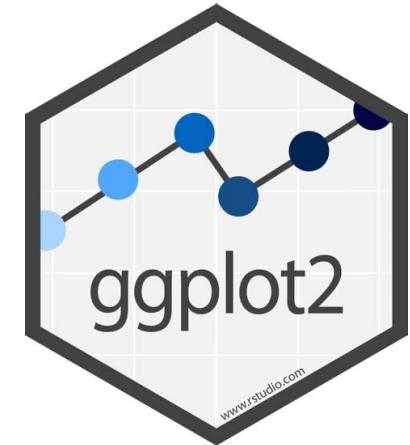
```
ggplot(penguins, aes(x=species, y=bill_length_mm)) +  
  geom_boxplot()
```



The basics of a ggplot2

Making a plot using **ggplot2** is like writing a recipe.

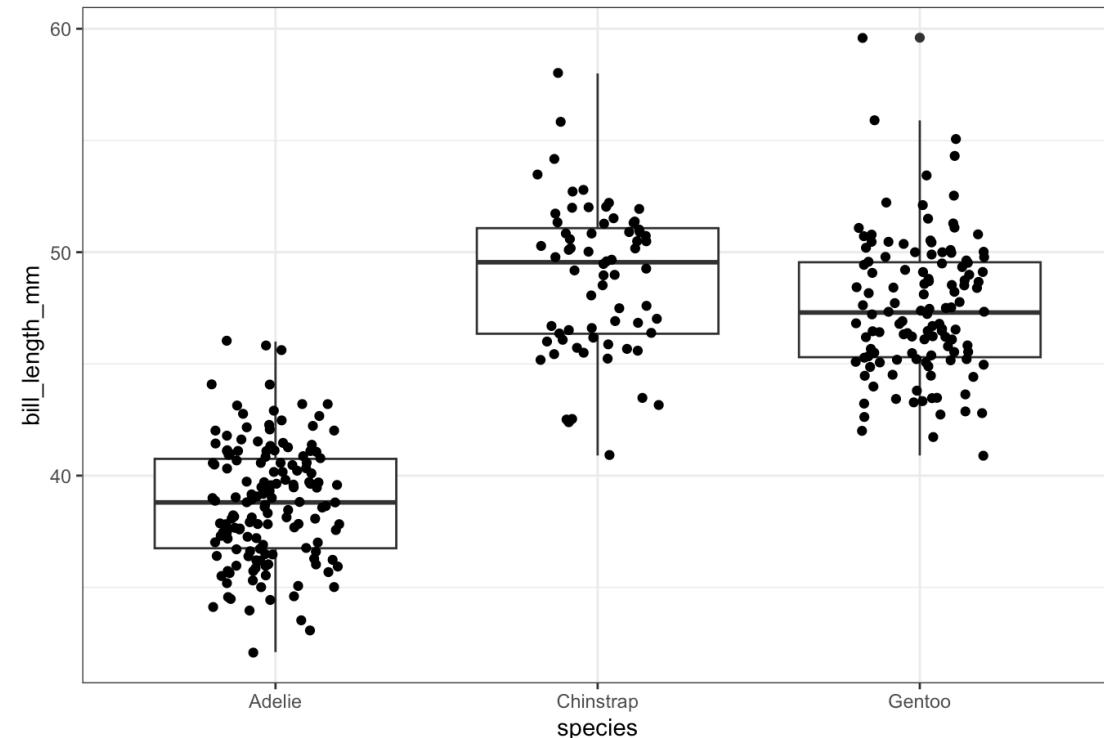
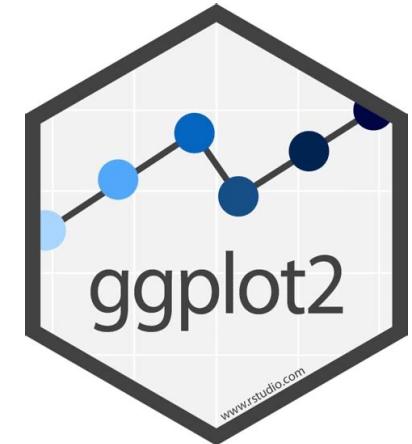
```
ggplot(penguins, aes(x=species, y=bill_length_mm)) +  
  geom_boxplot() +  
  geom_jitter(width = 0.2)
```



The basics of a ggplot2

Making a plot using **ggplot2** is like writing a recipe.

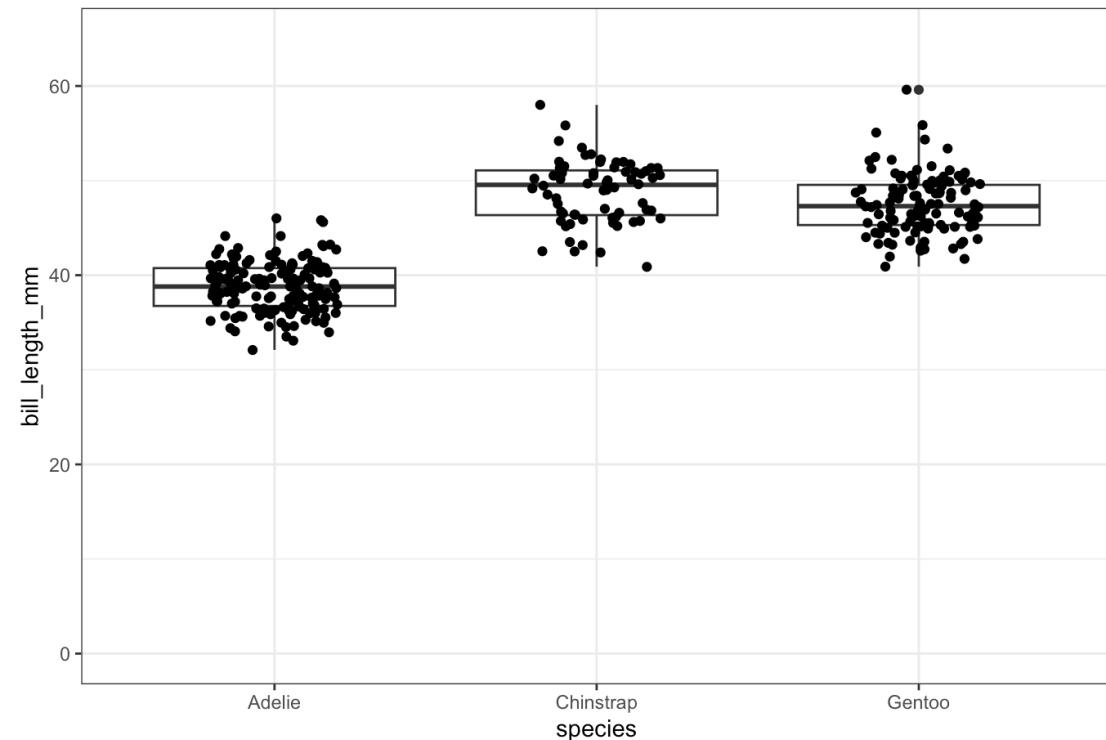
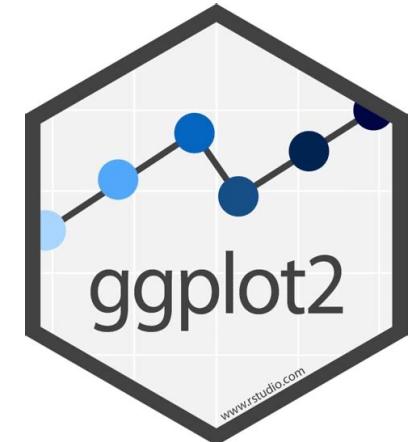
```
ggplot(penguins, aes(x=species, y=bill_length_mm)) +  
  geom_boxplot() +  
  geom_jitter(width = 0.2) +  
  theme_bw()
```



The basics of a ggplot2

Making a plot using **ggplot2** is like writing a recipe.

```
ggplot(penguins, aes(x=species, y=bill_length_mm)) +  
  geom_boxplot() +  
  geom_jitter(width = 0.2) +  
  theme_bw() +  
  ylim(c(0,65))
```



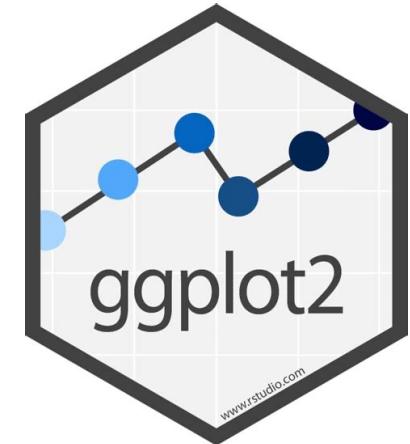
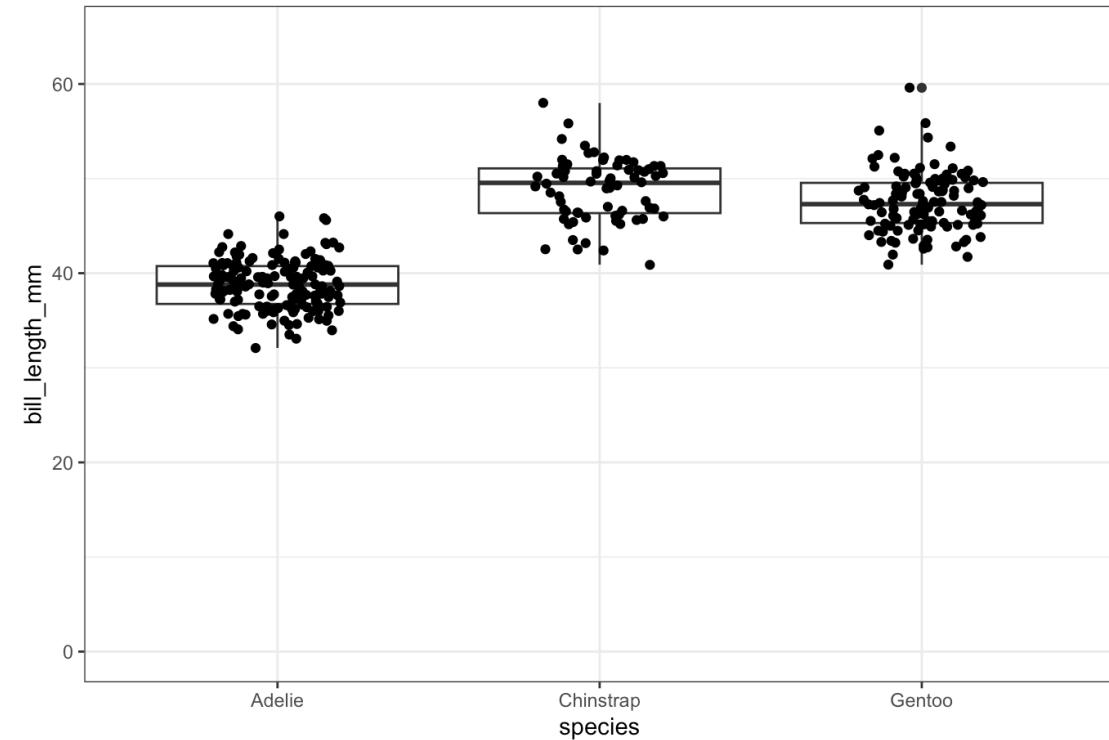
The basics of a ggplot2

Making a plot using **ggplot2** is like writing a recipe.

```
ggplot(penguins, aes(x=species, y=bill_length_mm)) +  
  geom_boxplot() +  
  geom_jitter(width = 0.2) +  
  theme_bw() +  
  ylim(c(0,65))
```

Use **aes()** when you want something in the plot to be dependent on the data.

Add a new layer using '+' and start a new code line to make it easy to use.



ggplot2 tips and tricks

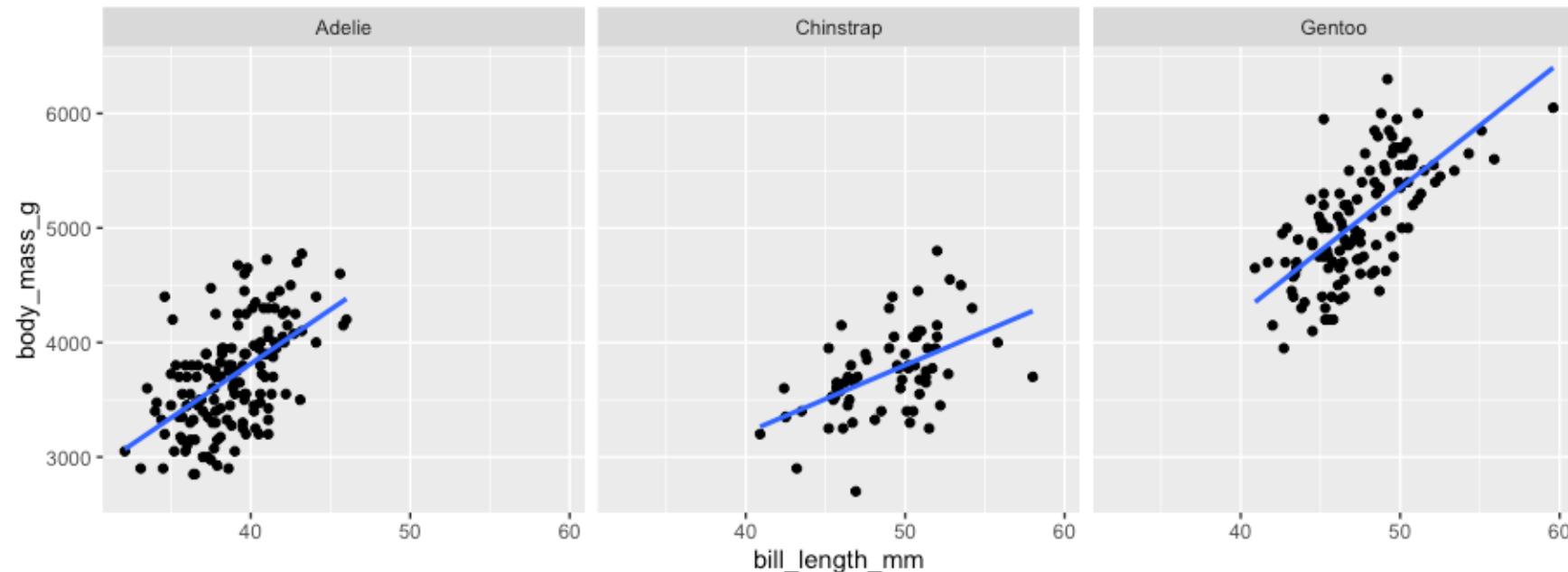
Automatically label facets with letters.

```
ggplot(penguins, aes(bill_length_mm, body_mass_g)) +  
  geom_point() +  
  geom_smooth(method = 'lm', se = FALSE) +  
  facet_wrap(~species)
```

ggplot2 tips and tricks

Automatically label facets with letters.

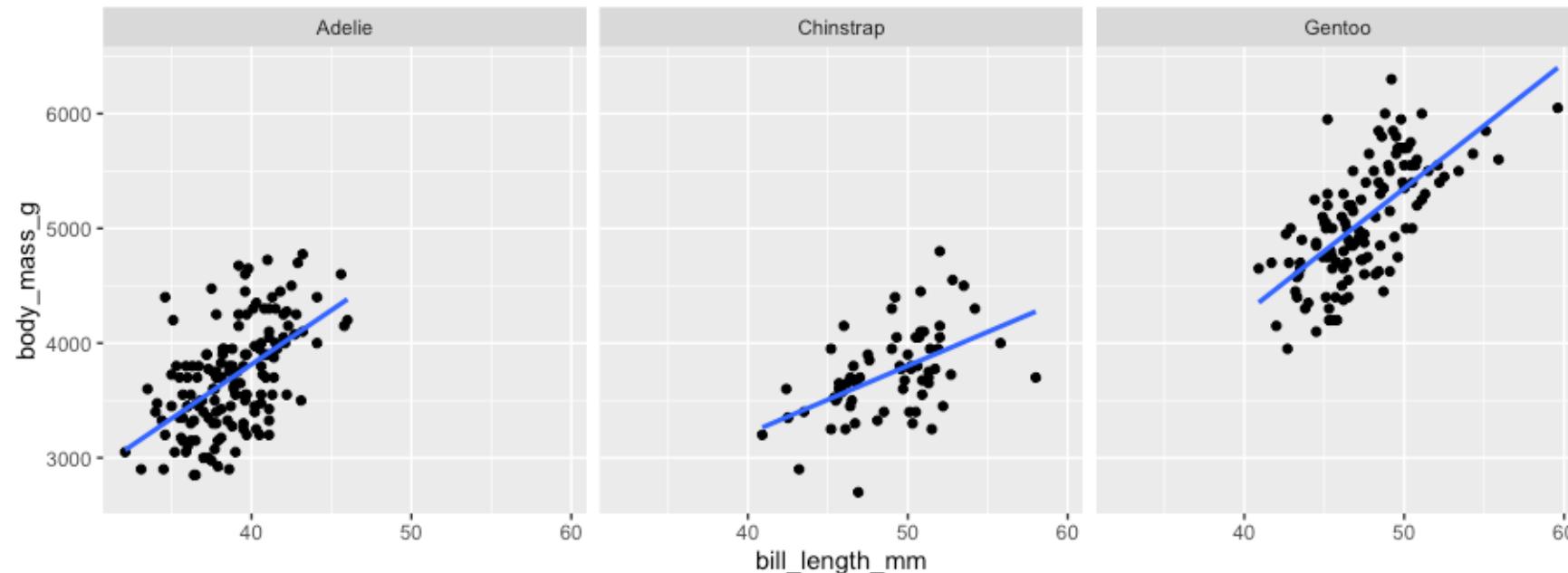
```
ggplot(penguins, aes(bill_length_mm, body_mass_g)) +  
  geom_point() +  
  geom_smooth(method = 'lm', se = FALSE) +  
  facet_wrap(~species)
```



ggplot2 tips and tricks

Automatically label facets with letters.

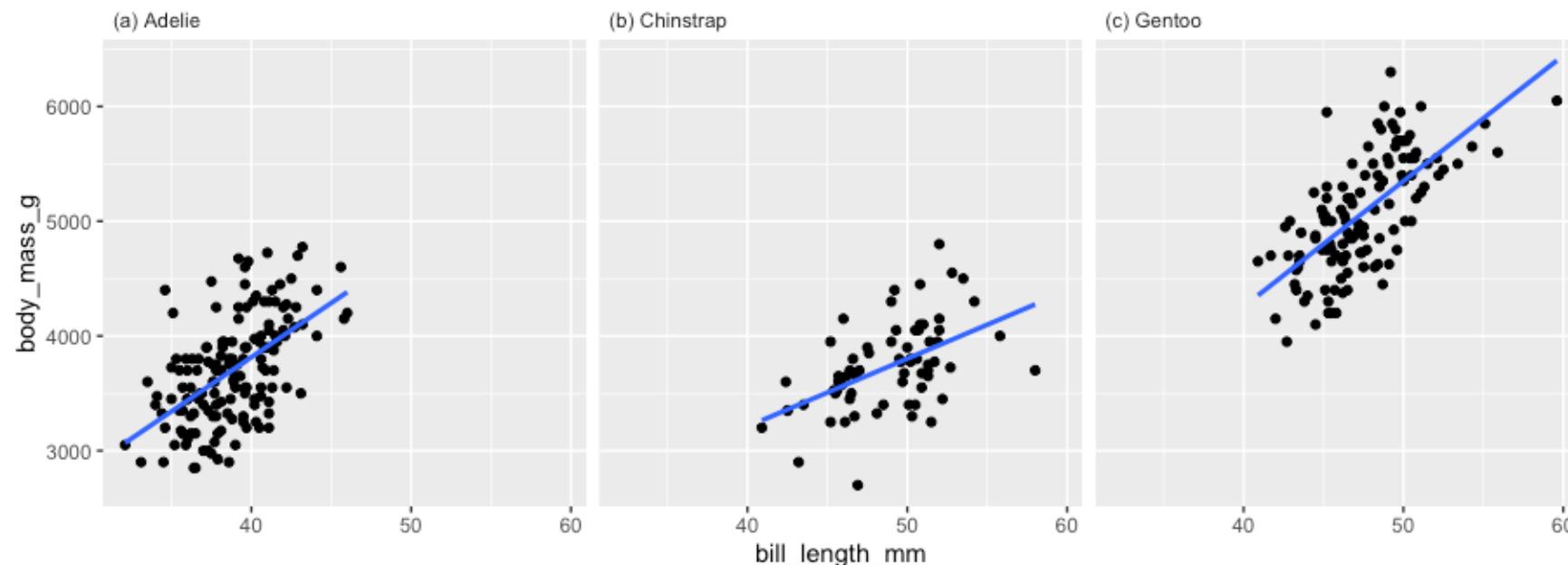
```
ggplot(penguins, aes(bill_length_mm, body_mass_g)) +  
  geom_point() +  
  geom_smooth(method = 'lm', se = FALSE) +  
  facet_wrap(~species, labeller = labeller(species = letter_facets)) +  
  theme(strip.background = element_blank(),  
        strip.text = element_text(hjust = 0))
```



ggplot2 tips and tricks

Automatically label facets with letters.

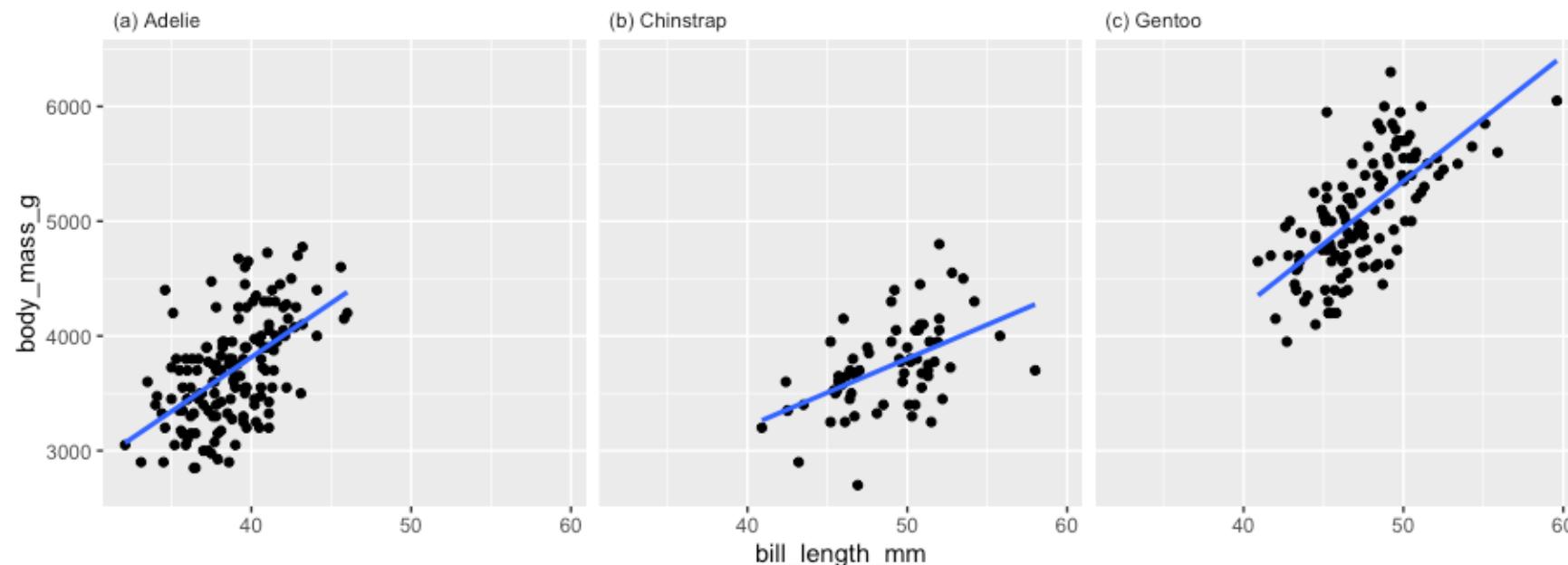
```
ggplot(penguins, aes(bill_length_mm, body_mass_g)) +  
  geom_point() +  
  geom_smooth(method = 'lm', se = FALSE) +  
  facet_wrap(~species, labeller = labeller(species = letter_facets)) +  
  theme(strip.background = element_blank(),  
        strip.text = element_text(hjust = 0))
```



ggplot2 tips and tricks

Automatically label facets with letters.

```
ggplot(penguins, aes(bill_length_mm, body_mass_g)) +  
  geom_point() +  
  geom_smooth(method = 'lm', se = FALSE) +  
  facet_wrap(~species, labeller = labeller(species = letter_facets)) +  
  theme(strip.background = element_blank(),  
        strip.text = element_text(hjust = 0))
```



```
# create function to add letter to facet labels  
letter_facets <- function(string){  
  len <- length(string)  
  string = paste("(", letters[1:len], ") ", string, sep = "")  
  return(string)  
}
```

ggplot2 tips and tricks

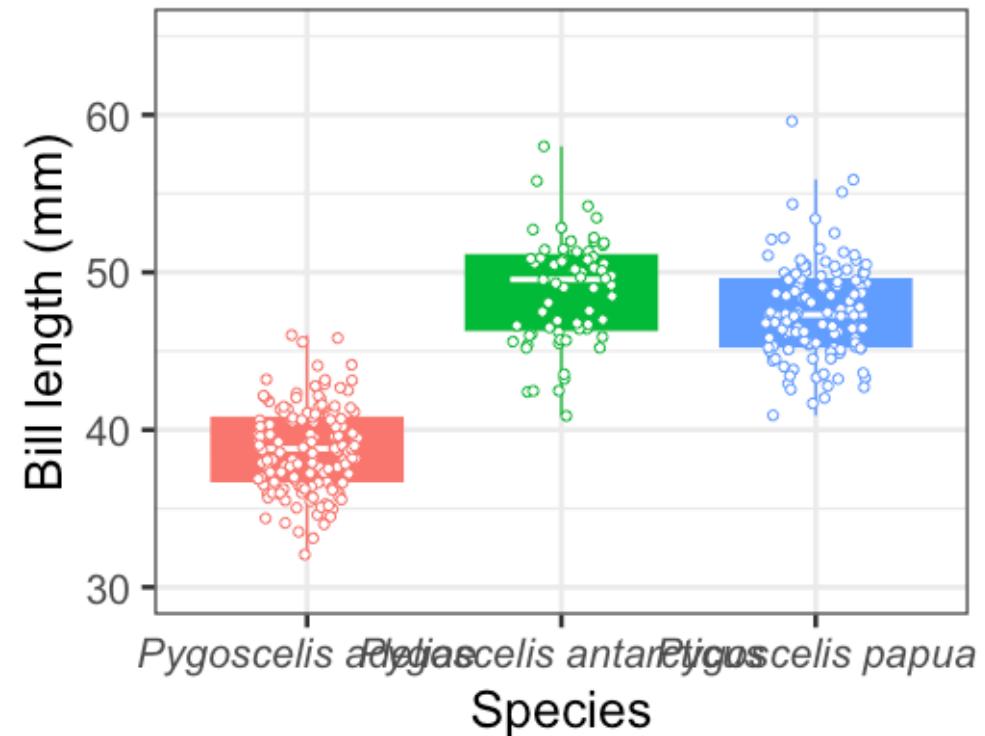
Soft wrap or stagger text labels to stop them overlapping.

```
ggplot(d, aes(latin_name, bill_length_mm, col = species, fill = species)) +  
  geom_boxplot(outlier.shape = NA) +  
  stat_summary(geom = "crossbar", fatten = 2, color = "white", width = 0.4,  
               fun.data = function(x){return(c(y = stats::median(x), ymin = stats::median(x), ymax = stats::median(x))))}) +  
  geom_jitter(width = 0.2, shape = 21, fill = 'white') +  
  theme_bw(base_size = 20) +  
  labs(x = 'Species',  
       y = 'Bill length (mm)') +  
  ylim(c(30,65)) + guides(col = 'none',  
                           fill = 'none') +  
  theme(axis.text.x = element_text(face = "italic"))
```

ggplot2 tips and tricks

Soft wrap or stagger text labels to stop them overlapping.

```
ggplot(d, aes(latin_name, bill_length_mm, col = species, fill = species)) +  
  geom_boxplot(outlier.shape = NA) +  
  stat_summary(geom = "crossbar", fatten = 2, color = "white", width = 0.4,  
               fun.data = function(x){return(c(y = stats::median(x), ymin = stats::median(x), ymax = stats::median(x))))}) +  
  geom_jitter(width = 0.2, shape = 21, fill = 'white') +  
  theme_bw(base_size = 20) +  
  labs(x = 'Species',  
       y = 'Bill length (mm)') +  
  ylim(c(30,65)) + guides(col = 'none',  
                           fill = 'none') +  
  theme(axis.text.x = element_text(face = "italic"))
```

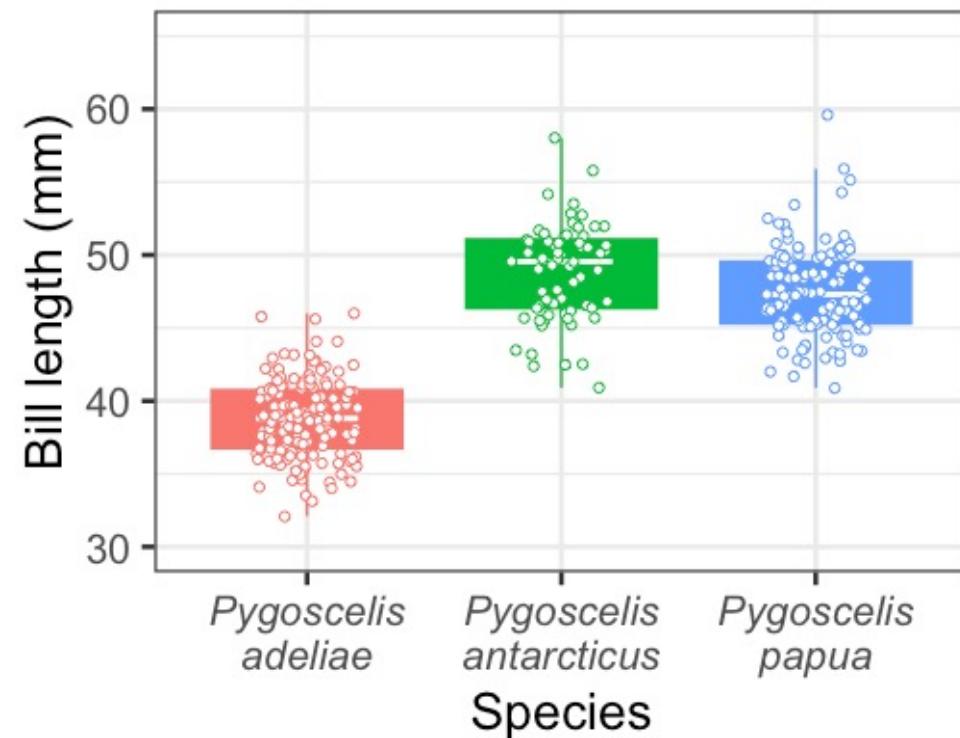


ggplot2 tips and tricks

Soft wrap or stagger text labels to stop them overlapping.

```
ggplot(d, aes(latin_name, bill_length_mm, col = species, fill = species)) +  
  geom_boxplot(outlier.shape = NA) +  
  stat_summary(geom = "crossbar", fatten = 2, color = "white", width = 0.4,  
               fun.data = function(x){return(c(y = stats::median(x), ymin = stats::median(x), ymax = stats::median(x))))}) +  
  geom_jitter(width = 0.2, shape = 21, fill = 'white') +  
  theme_bw(base_size = 20) +  
  labs(x = 'Species',  
       y = 'Bill length (mm)') +  
  ylim(c(30,65)) + guides(col = 'none',  
                           fill = 'none') +  
  theme(axis.text.x = element_text(face = "italic")) +  
  scale_x_discrete(labels = scales::label_wrap(10))
```

Automatic wrap using **scales::label_wrap()**.

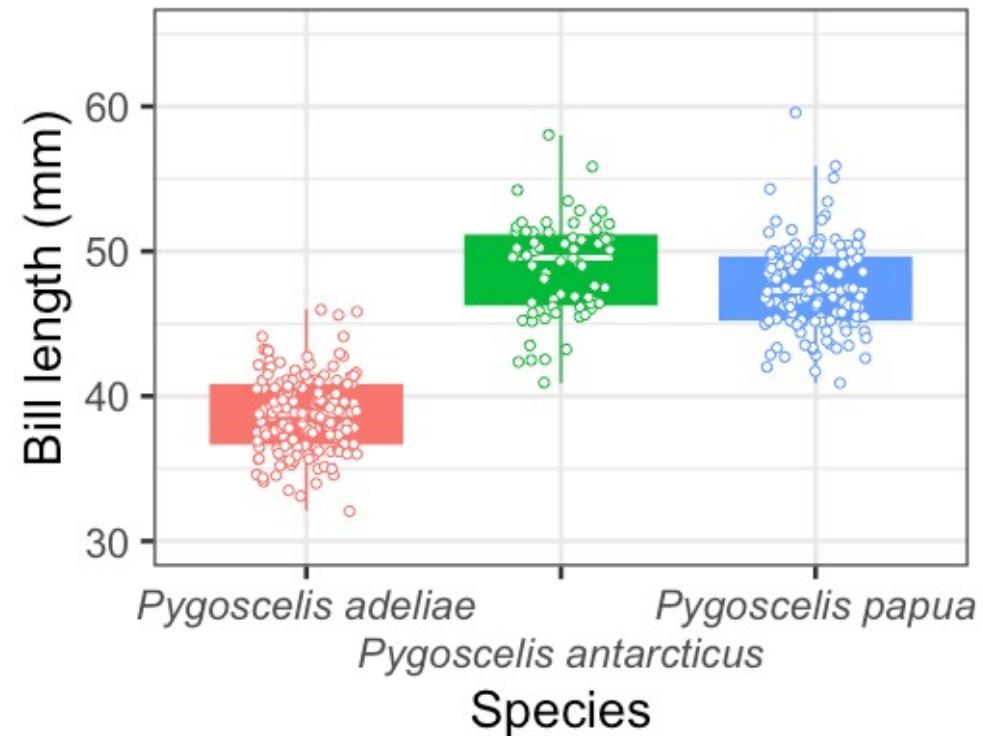


ggplot2 tips and tricks

Soft wrap or stagger text labels to stop them overlapping.

```
ggplot(d, aes(latin_name, bill_length_mm, col = species, fill = species)) +  
  geom_boxplot(outlier.shape = NA) +  
  stat_summary(geom = "crossbar", fatten = 2, color = "white", width = 0.4,  
               fun.data = function(x){return(c(y = stats::median(x), ymin = stats::median(x), ymax = stats::median(x))))}) +  
  geom_jitter(width = 0.2, shape = 21, fill = 'white') +  
  theme_bw(base_size = 20) +  
  labs(x = 'Species',  
       y = 'Bill length (mm)') +  
  ylim(c(30,65)) + guides(col = 'none',  
                           fill = 'none') +  
  theme(axis.text.x = element_text(face = "italic")) +  
  scale_x_discrete(guide = guide_axis(n.dodge = 2))
```

Dodge labels using `guide_axis()`.



ggplot2 tips and tricks

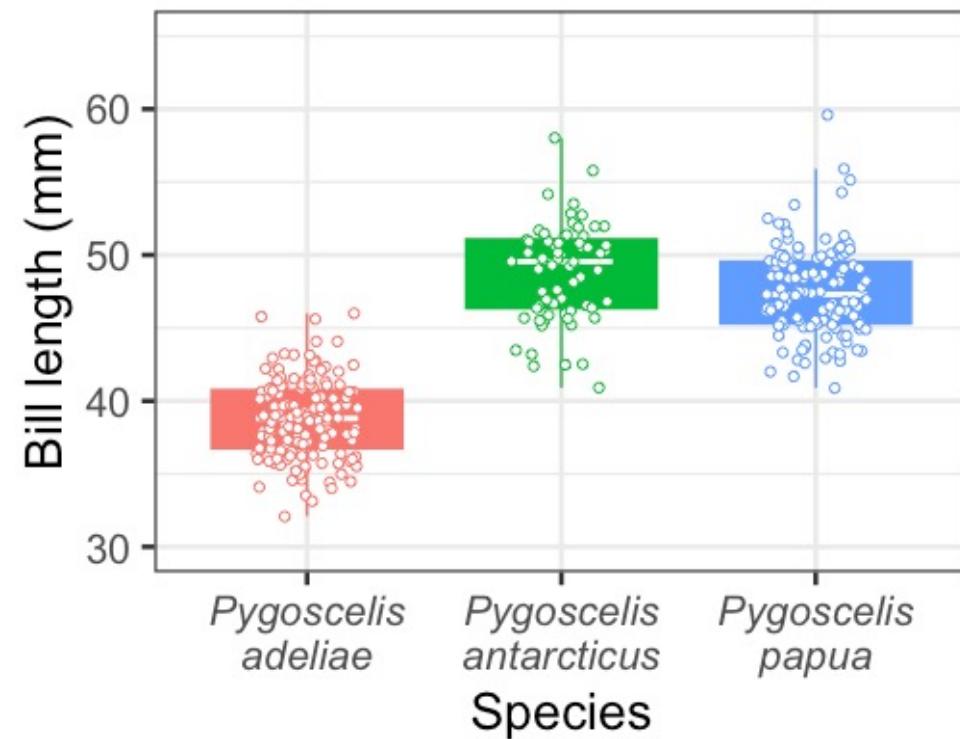
Add a final NULL layer to allow you to comment out layers.

```
ggplot(d, aes(latin_name, bill_length_mm, col = species, fill = species)) +  
  geom_boxplot(outlier.shape = NA) +  
  stat_summary(geom = "crossbar", fatten = 2, color = "white", width = 0.4,  
               fun.data = function(x){return(c(y = stats::median(x), ymin = stats::median(x), ymax = stats::median(x))))}) +  
  geom_jitter(width = 0.2, shape = 21, fill = 'white') +  
  theme_bw(base_size = 20) +  
  labs(x = 'Species',  
       y = 'Bill length (mm)') +  
  ylim(c(30,65)) +  
  guides(col = 'none',  
         fill = 'none') # turn off legend  
  theme(axis.text.x = element_text(face = "italic")) +  
  scale_x_discrete(labels = scales::label_wrap(10)) +  
  #scale_x_discrete(guide = guide_axis(n.dodge = 2)) +  
  NULL
```

ggplot2 tips and tricks

Add a final NULL layer to allow you to comment out layers.

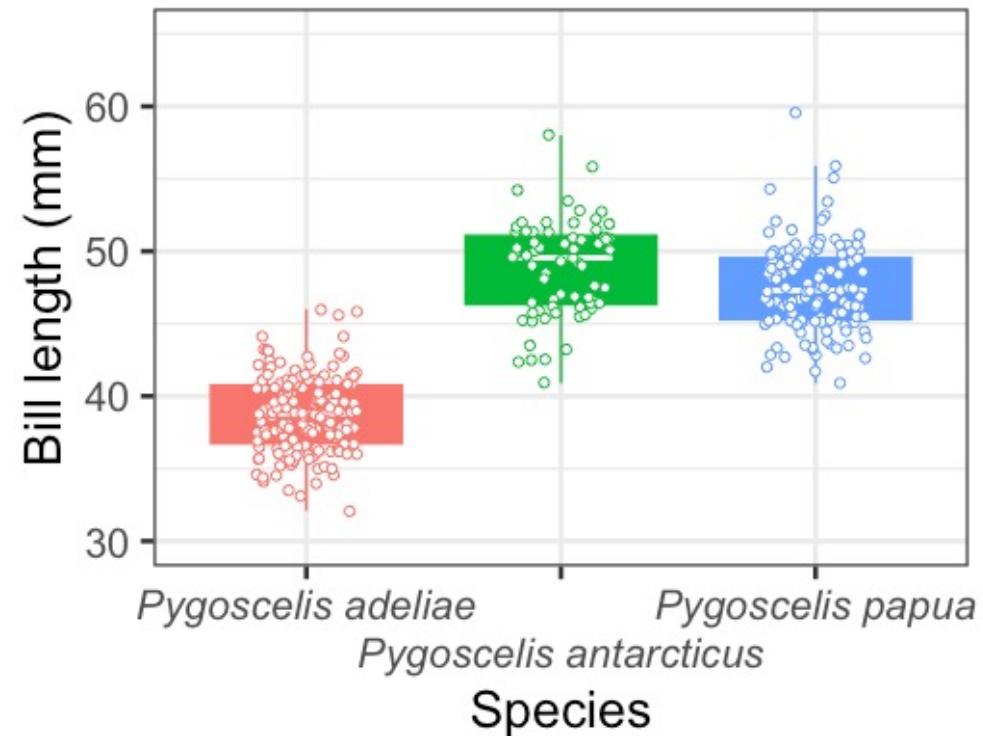
```
ggplot(d, aes(latin_name, bill_length_mm, col = species, fill = species)) +  
  geom_boxplot(outlier.shape = NA) +  
  stat_summary(geom = "crossbar", fatten = 2, color = "white", width = 0.4,  
               fun.data = function(x){return(c(y = stats::median(x), ymin = stats::median(x), ymax = stats::median(x))))}) +  
  geom_jitter(width = 0.2, shape = 21, fill = 'white') +  
  theme_bw(base_size = 20) +  
  labs(x = 'Species',  
       y = 'Bill length (mm)') +  
  ylim(c(30,65)) +  
  guides(col = 'none',  
         fill = 'none') + # turn off legend  
  theme(axis.text.x = element_text(face = "italic")) +  
  scale_x_discrete(labels = scales::label_wrap(10)) +  
  #scale_x_discrete(guide = guide_axis(n.dodge = 2)) +  
  NULL
```



ggplot2 tips and tricks

Add a final NULL layer to allow you to comment out layers.

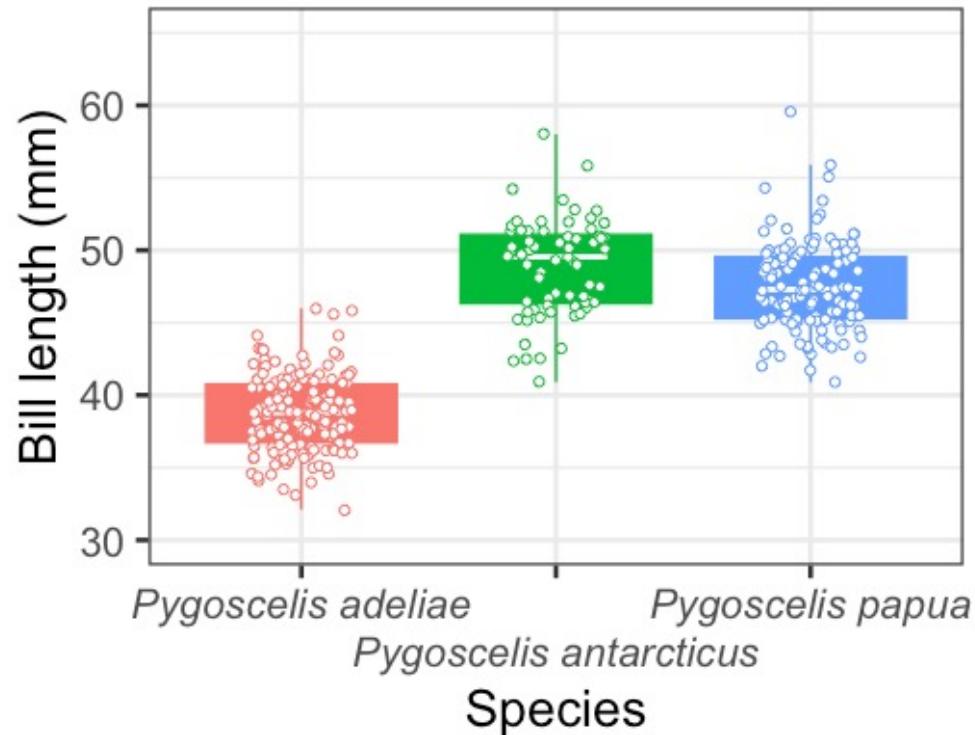
```
ggplot(d, aes(latin_name, bill_length_mm, col = species, fill = species)) +  
  geom_boxplot(outlier.shape = NA) +  
  stat_summary(geom = "crossbar", fatten = 2, color = "white", width = 0.4,  
               fun.data = function(x){return(c(y = stats::median(x), ymin = stats::median(x), ymax = stats::median(x))))}) +  
  geom_jitter(width = 0.2, shape = 21, fill = 'white') +  
  theme_bw(base_size = 20) +  
  labs(x = 'Species',  
       y = 'Bill length (mm)') +  
  ylim(c(30,65)) +  
  guides(col = 'none',  
         fill = 'none') + # turn off legend  
  theme(axis.text.x = element_text(face = "italic")) +  
  #scale_x_discrete(labels = scales::label_wrap(10)) +  
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +  
  NULL
```



ggplot2 tips and tricks

Save out plot using ggsave().

```
ggsave('plots/penguins.pdf', last_plot(), width = 7, height = 5)  
ggsave('plots/penguins.png', last_plot(), width = 7, height = 5)
```



ggplot2 extensions

ggplot2 extensions

Patchwork for aligning plots.

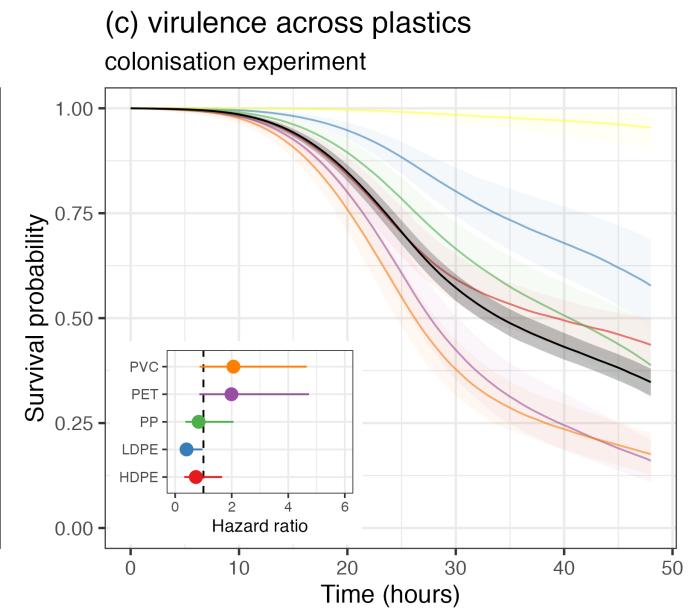
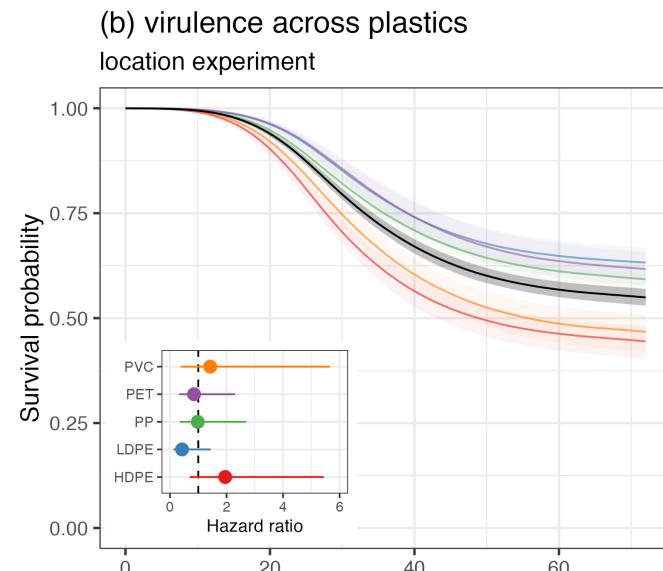
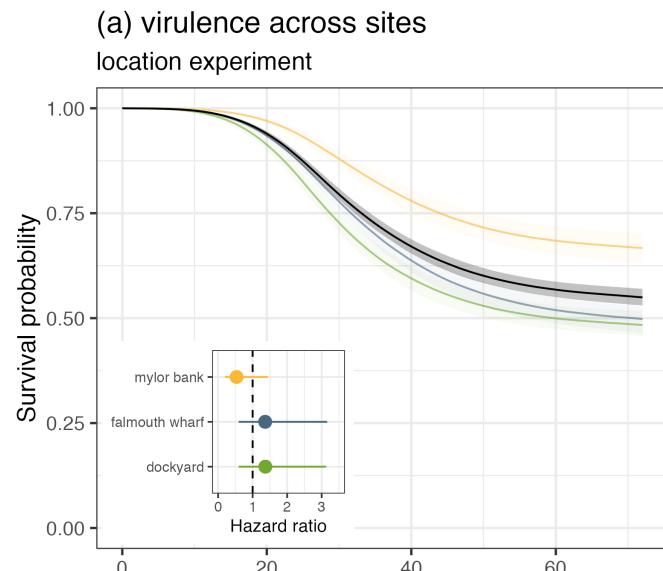
```
plot1 + plot2 + plot_layout(widths = c(0.5,0.5))
```



ggplot2 extensions

Patchwork for aligning plots.

```
plot1 + plot2 + plot_layout(widths = c(0.5,0.5))
```



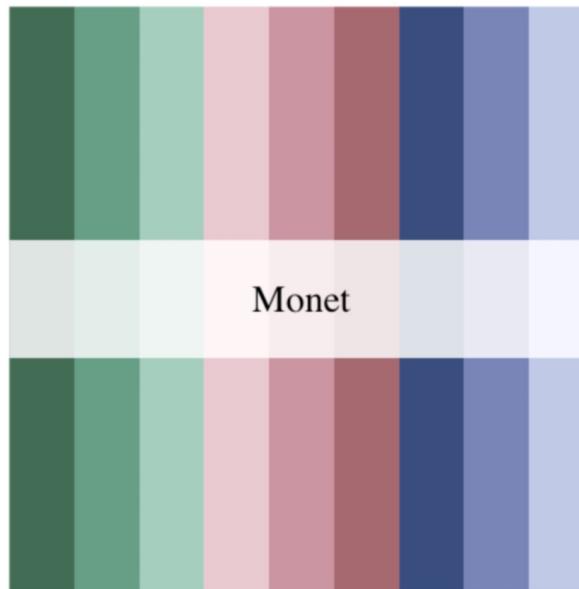
ggplot2 extensions

MetBrewer for beautiful colour schemes.

Adds to **scale_color_manual()** and **scale_fill_manual()**



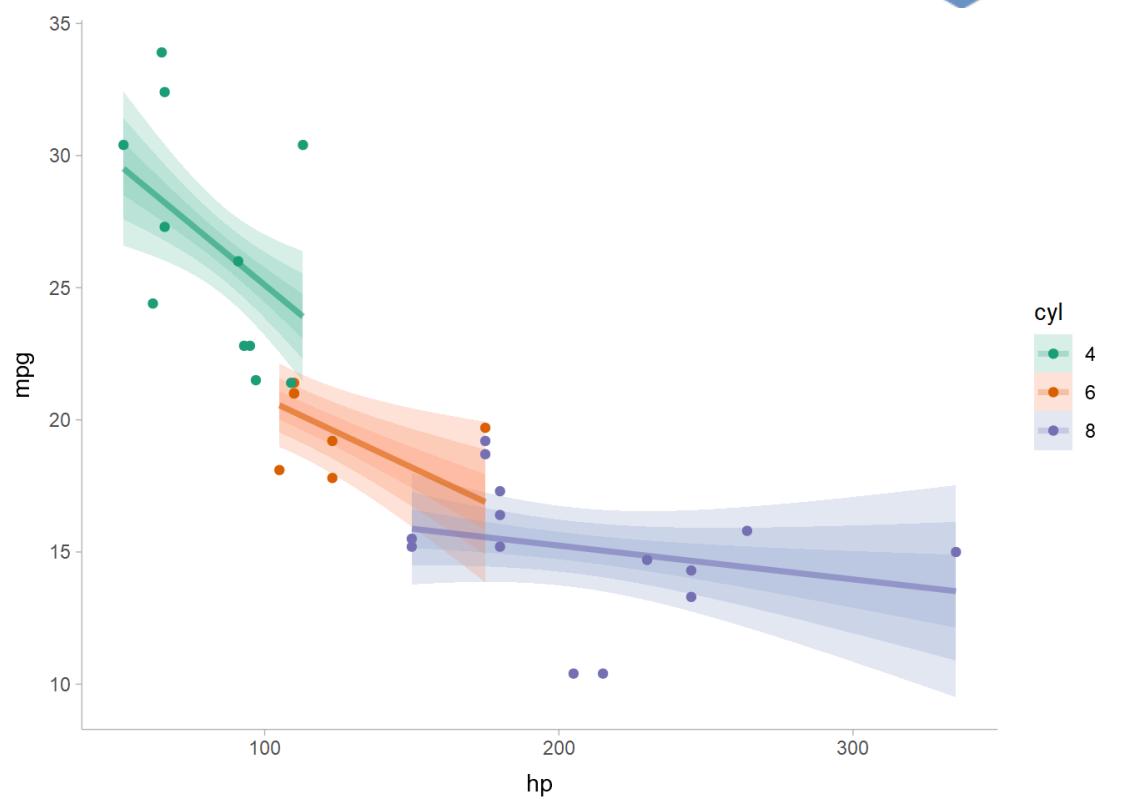
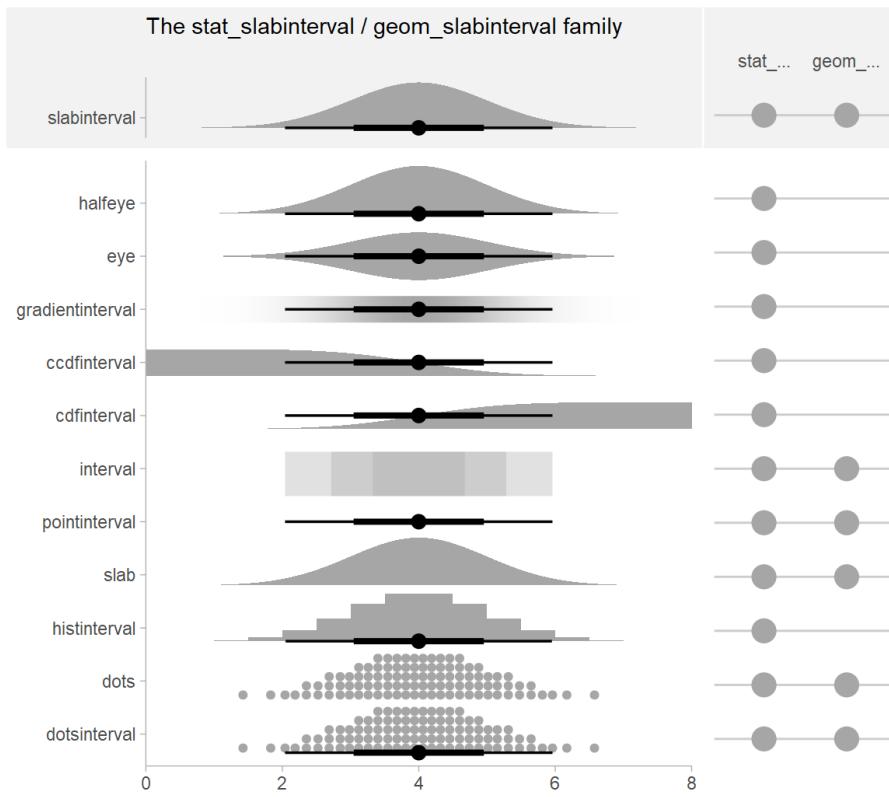
Bridge over a Pond of Water Lilies, 1899, Claude Monet



<https://github.com/BlakeRMills/MetBrewer>

ggplot2 extensions

ggdist for visualising distributions and uncertainty.



ggplot2 extensions

ggtree for making phylogenetic trees.

ggridge for making animations.

ggnewscale for adding new (colour) scales.

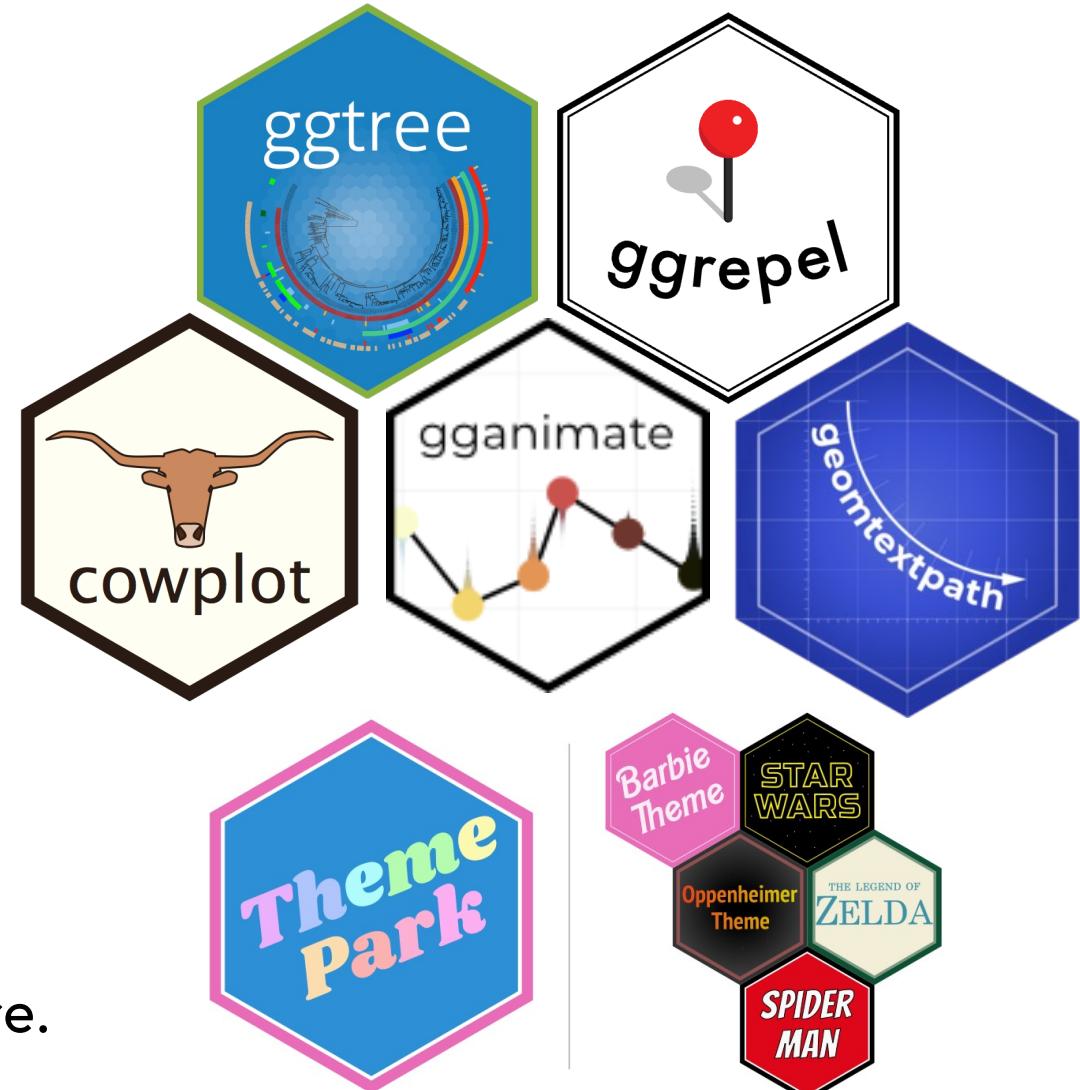
palettetown for Pokemon inspired colour palettes.

cowplot for aligning and labelling plots.

geomtextpath for curved text annotations.

ggrepel for better spaced text annotations.

themepark for themes based around popular culture.



Learn more about ggplot2

- **ggplot2** has a very well documented [website](#).
- There is a whole [book](#) on how to use **ggplot2**.
- We have written a [walkthrough](#) of our favourite tips and tricks.
- I end up on Cedric Scherer's [blog post](#) a lot.
- StackOverflow, ChatGPT, #TidyTuesday all good places to get help and inspiration.

A brief introduction to **flextable**

- **flextable** makes it easy to create table for reporting and publications.
- Can embed tables into Rmarkdown, pdf, Word, pdfs. I like saving mine as images.
- Downloaded 1.5 million times!



A brief introduction to flextable



- **flextable** makes it easy to create table for reporting and publications.
- Can embed tables into Rmarkdown, pdf, Word, pdfs. I like saving mine as images.
- Downloaded 1.5 million times!

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap -32.4 67.5 339 -0.480  0.8807
Adelie - Gentoo   -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001
```

P value adjustment: tukey method for comparing a family of 3 estimates

.

A brief introduction to flextable



- **flextable** makes it easy to create table for reporting and publications.
- Can embed tables into Rmarkdown, pdf, Word, pdfs. I like saving mine as images.
- Downloaded 1.5 million times!

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap -32.4 67.5 339 -0.480  0.8807
Adelie - Gentoo   -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001
```

P value adjustment: tukey method for comparing a family of 3 estimates

Contrast	estimate	s.e.	df.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

A brief introduction to flextable



- **flextable** makes it easy to create table for reporting and publications.
- Can embed tables into Rmarkdown, pdf, Word, pdfs. I like saving mine as images.
- Downloaded 1.5 million times!

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap -32.4 67.5 339 -0.480  0.8807
Adelie - Gentoo   -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001
```

P value adjustment: tukey method for comparing a family of 3 estimates

Contrast	estimate	s.e.	df.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

isolate	AMR gene
1	multidrug efflux RND transporter permease subunit SdeB class C beta-lactamase
2	multidrug efflux RND transporter permease subunit SdeY multidrug efflux RND transporter permease subunit SdeB class C beta-lactamase SRT-3 tetracycline efflux MFS transporter Tet(41) multidrug efflux MFS transporter SmfY aminoglycoside 6-N-acetyltransferase
3	class A beta-lactamase BlaP
4	multidrug efflux MFS transporter SmfY tetracycline efflux MFS transporter Tet(41) class C beta-lactamase SRT-3 multidrug efflux RND transporter permease subunit SdeB multidrug efflux RND transporter permease subunit SdeY aminoglycoside 6-N-acetyltransferase
5	multidrug efflux RND transporter permease subunit SdeB class A beta-lactamase
6	class A beta-lactamase BlaP
7	cephalosporin-hydrolyzing class C beta-lactamase MIR-16 multidrug efflux RND transporter permease subunit OqxB9 fosfomycin resistance glutathione transferase FosA
8	class A beta-lactamase BlaP
9	cephalosporin-hydrolyzing class C beta-lactamase MIR-16 multidrug efflux RND transporter permease subunit OqxB9 fosfomycin resistance glutathione transferase FosA
10	multidrug efflux MFS transporter SmfY tetracycline efflux MFS transporter Tet(41) class C beta-lactamase SRT-3 multidrug efflux RND transporter permease subunit SdeB multidrug efflux RND transporter permease subunit SdeY aminoglycoside 6-N-acetyltransferase
11	class A beta-lactamase FONA-4
12	trimethoprim-resistant dihydrofolate reductase DfrE ABC-F type ribosomal protection protein LsrfA
13	multidrug efflux MFS transporter SmfY tetracycline efflux MFS transporter Tet(41) class C beta-lactamase SRT-3 multidrug efflux RND transporter permease subunit SdeB multidrug efflux RND transporter permease subunit SdeY aminoglycoside 6-N-acetyltransferase
14	multidrug efflux RND transporter permease subunit SdeY multidrug efflux RND transporter permease subunit SdeB class C beta-lactamase SRT-3 tetracycline efflux MFS transporter Tet(41) multidrug efflux MFS transporter SmfY aminoglycoside 6-N-acetyltransferase
15	class A beta-lactamase BlaP
16	multidrug efflux RND transporter permease subunit SdeB multidrug efflux RND transporter permease subunit SdeY multidrug efflux RND transporter permease subunit OqxB9 phosphoethanolamine-lipid A transferase MCR-10.1 multidrug efflux MFS transporter SmfY tetracycline efflux MFS transporter Tet(41) class C beta-lactamase SRT-3 class A extended-spectrum beta-lactamase OXY-6-2 cephalosporin-hydrolyzing class C beta-lactamase MIR-12 aminoglycoside 6-N-acetyltransferase
17	multidrug efflux MFS transporter SmfY tetracycline efflux MFS transporter Tet(41) class C beta-lactamase SRT-3 multidrug efflux RND transporter permease subunit SdeB multidrug efflux RND transporter permease subunit SdeY aminoglycoside 6-N-acetyltransferase class A beta-lactamase FONA-6
18	multidrug efflux MFS transporter SmfY tetracycline efflux MFS transporter Tet(41) class C beta-lactamase SRT-3 multidrug efflux RND transporter permease subunit SdeB multidrug efflux RND transporter permease subunit SdeY aminoglycoside 6-N-acetyltransferase
19	class C beta-lactamase multidrug efflux RND transporter permease subunit SdeB
20	class C beta-lactamase multidrug efflux RND transporter permease subunit SdeB

The basics of flextable

```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable()
```



The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable()
```

contrast	estimate	SE	df	t.ratio	p.value
Adelie - Chinstrap	-32.42598	67.51168	339	-0.4803018	0.88
Adelie - Gentoo	-1,375.35401	56.14797	339	-24.4951686	<0.0001
Chinstrap - Gentoo	-1,342.92802	69.85693	339	-19.2239776	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value")
```

Contrast	estimate	s.e.	d.f.	t-ratio	p value
Adelie - Chinstrap	-32.42598	67.51168	339	-0.4803018	0.88
Adelie - Gentoo	-1,375.35401	56.14797	339	-24.4951686	<0.0001
Chinstrap - Gentoo	-1,342.92802	69.85693	339	-19.2239776	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header')
```

Contrast	estimate	s.e.	d.f.	t-ratio	p value
Adelie - Chinstrap	-32.42598	67.51168	339	-0.4803018	0.88
Adelie - Gentoo	-1,375.35401	56.14797	339	-24.4951686	<0.0001
Chinstrap - Gentoo	-1,342.92802	69.85693	339	-19.2239776	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001
P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2)
```

Contrast	estimate	s.e.	d.f.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2) %>%
  align(align = 'center', part = 'header')
```

Contrast	estimate	s.e.	d.f.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001
P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2) %>%
  align(align = 'center', part = 'header') %>%
  align(align = 'left', part = 'body')
```

Contrast	estimate	s.e.	d.f.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2) %>%
  align(align = 'center', part = 'header') %>%
  align(align = 'left', part = 'body') %>%
  font(fontname = 'Times', part = 'all')
```

Contrast	estimate	s.e.	df.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2) %>%
  align(align = 'center', part = 'header') %>%
  align(align = 'left', part = 'body') %>%
  font(fontname = 'Times', part = 'all') %>%
  bold(~p.value == "<0.0001", j = "p.value")
```

Contrast	estimate	s.e.	df.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE  df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2) %>%
  align(align = 'center', part = 'header') %>%
  align(align = 'left', part = 'body') %>%
  font(fontname = 'Times', part = 'all') %>%
  bold(~p.value == "<0.0001", j = "p.value") %>%
  fontsize(size = 12, part = 'all')
```

Contrast	estimate	s.e.	df	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate   SE df t.ratio p.value
Adelie - Chinstrap    -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2) %>%
  align(align = 'center', part = 'header') %>%
  align(align = 'left', part = 'body') %>%
  font(fontname = 'Times', part = 'all') %>%
  bold(~p.value == "<0.0001", j = "p.value") %>%
  fontsize(size = 12, part = 'all') %>%
  autofit()
```

Contrast	estimate	s.e.	d.f.	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

The basics of flextable



```
mod1 <- lm(body_mass_g ~ species, penguins)
contrasts <- emmeans::emmeans(mod1, pairwise ~ species)
contrasts$contrasts
```

```
contrast      estimate    SE   df t.ratio p.value
Adelie - Chinstrap   -32.4 67.5 339  -0.480  0.8807
Adelie - Gentoo     -1375.4 56.1 339 -24.495 <.0001
Chinstrap - Gentoo -1342.9 69.9 339 -19.224 <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

Uses the pipe operator '%>%' to start a new layer

```
contrasts$contrasts %>%
  data.frame() %>%
  mutate(p.value = ifelse(p.value < 0.0001, "<0.0001", as.character(round(p.value, 2)))) %>%
  flextable() %>%
  set_header_labels(contrast = 'Contrast',
                    emmean = 'Estimate',
                    SE = 's.e.',
                    df = 'd.f.',
                    t.ratio = "t-ratio",
                    p.value = "p value") %>%
  italic(j = c(3:6), part = 'header') %>%
  colformat_double(j = c(2:6), digits = 2) %>%
  align(align = 'center', part = 'header') %>%
  align(align = 'left', part = 'body') %>%
  font(fontname = 'Times', part = 'all') %>%
  bold(~p.value == "<0.0001", j = "p.value") %>%
  fontsize(size = 12, part = 'all') %>%
  autofit()
```

Contrast	estimate	s.e.	df	t-ratio	p value
Adelie - Chinstrap	-32.43	67.51	339.00	-0.48	0.88
Adelie - Gentoo	-1,375.35	56.15	339.00	-24.50	<0.0001
Chinstrap - Gentoo	-1,342.93	69.86	339.00	-19.22	<0.0001

Save out table using `save_as_image()`

Learn more about flextable

- **flextable** has a very well documented [website](#).
- There is a whole online [book](#) on how to use **flextable**.
- We have written a [walkthrough](#) of our favourite tips and tricks.
- Post on the Teams Channel and we can learn **together!**
- Look up other people's code on GitHub.



Time to have some fun!

ggplot2

1. Beautify a plot you have already started.
2. Make your first **ggplot** using **palmerpenguins** and change some of the defaults colours/shapes/themes
3. Change the font used in a **ggplot2**
4. Align multiple plots using **patchwork** or **cowplot**
5. Make an animation using **ggridge**

flextable

1. Make a table you need to make
2. Make your first **flextable** using **palmerpenguins** and try rename some column names and round numbers to two decimal places
3. Merge some cells vertically that contain the same value
4. Make some cells bold



Share it with us on the Teams Channel!