# L06 Annotation & Positioning

Data Visualization (STAT 302)

Anna Wagman

## Contents



```
library(knitr)
opts_chunk$set(warning = FALSE, message = FALSE, comment = FALSE, dpi = 300)
```

### Overview

The goal of this lab is to explore methods for annotating and positioning with `ggplot2` plots. This lab also utilizes `scale_*` to a greater degree which is part of our next reading. In fact, students may find going through/reading chapter 11 Colour scales and legends useful.

### Datasets

We'll be using the `blue_jays.rda`, `titanic.rda`, `Aus_athletes.rda`, and `tech_stocks.rda` datasets.

```
# Load package(s)
library(tidyverse)
library(skimr)
library(ggplot2)
library(ggrepel)
library(cowplot)
library(lubridate)
library(patchwork)
library(scales)

# Load datasets
load(file = "data/blue_jays.rda")
load(file = "data/titanic.rda")
load(file = "data/Aus_athletes.rda")
load(file = "data/tech_stocks.rda")
load(file = "data/corruption.rda")
```

### Exercises

Complete the following exercises.

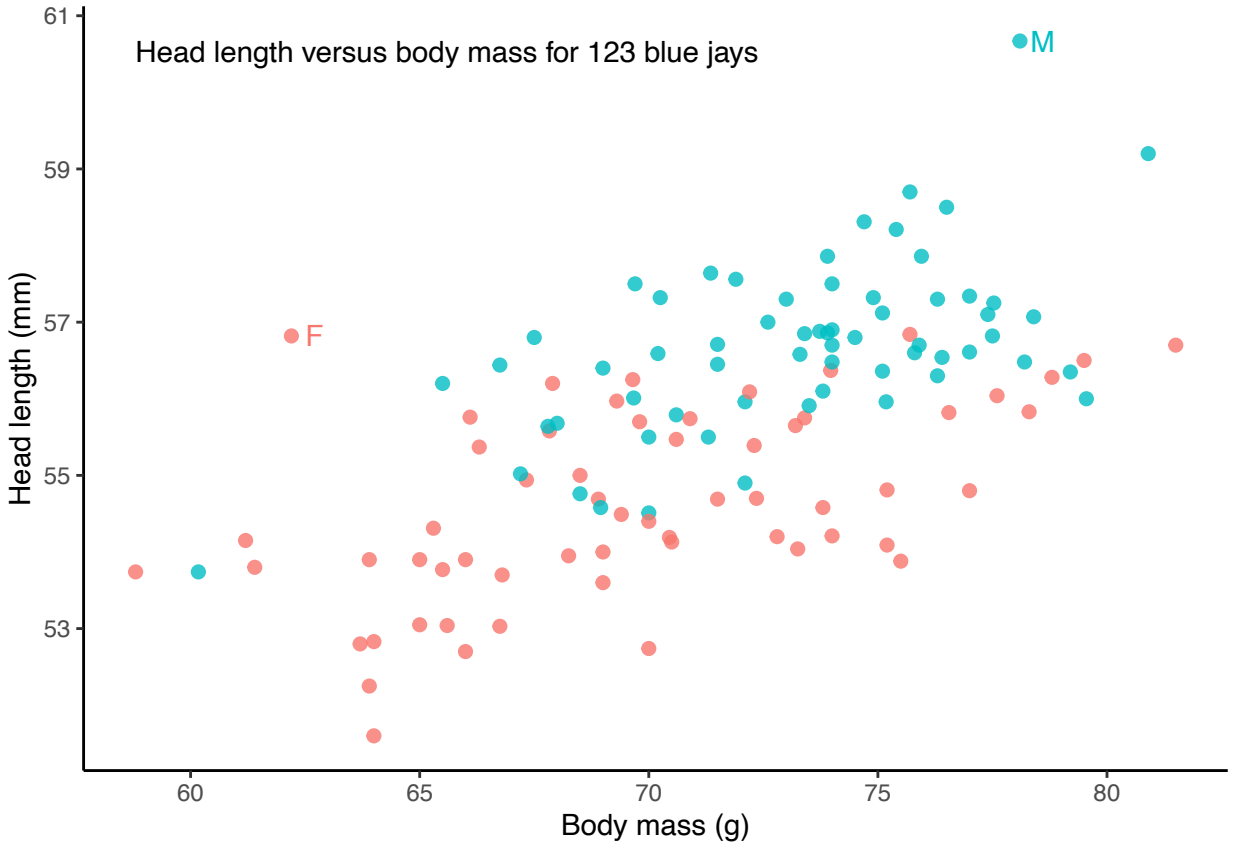
#### Exercise 1

Using `blue_jays.rda` dataset, recreate the following graphic as precisely as possible.

*Hints:*

- Transparency is 0.8
- Point size 2
- Create a `label_info` dataset that is a subset of original data, just with the 2 birds to be labeled
- Shift label text horizontally by 0.5
- See 8.3 building custom annotations
- Annotation size is 4
- Classic theme

```
label_info_ex1 <- blue_jays %>% #subset of original data
  select(Mass, Head, KnownSex) %>%
  arrange(desc(Head)) %>% #arrange by head size
  slice(n = c(1, 28))

caption <- 'Head length versus body mass for 123 blue jays'

xrange <- range(blue_jays$Mass)
yrange <- range(blue_jays$Head)

##plot label_info

ggplot(data = blue_jays,
       aes(x = Mass, y = Head, color = KnownSex)) +

  geom_point(size = 2, alpha = 0.8) + #given point size = 2 and transparency = .8
  labs(
    x = 'Body mass (g)',
    y = 'Head length (mm)') +

  theme_classic() + #theme is classic
  geom_text(data = label_info_ex1, label = c('M', 'F'),
            nudge_x = 0.5) + #shift horiz by .5
  annotate(geom = "text", x = xrange[1], y = yrange[2],
           label = caption, hjust = 0, vjust = 1, size = 4) +
  theme(legend.position = 'none')
```




**Exercise 2**

Using `tech_stocks` dataset, recreate the following graphics as precisely as possible.

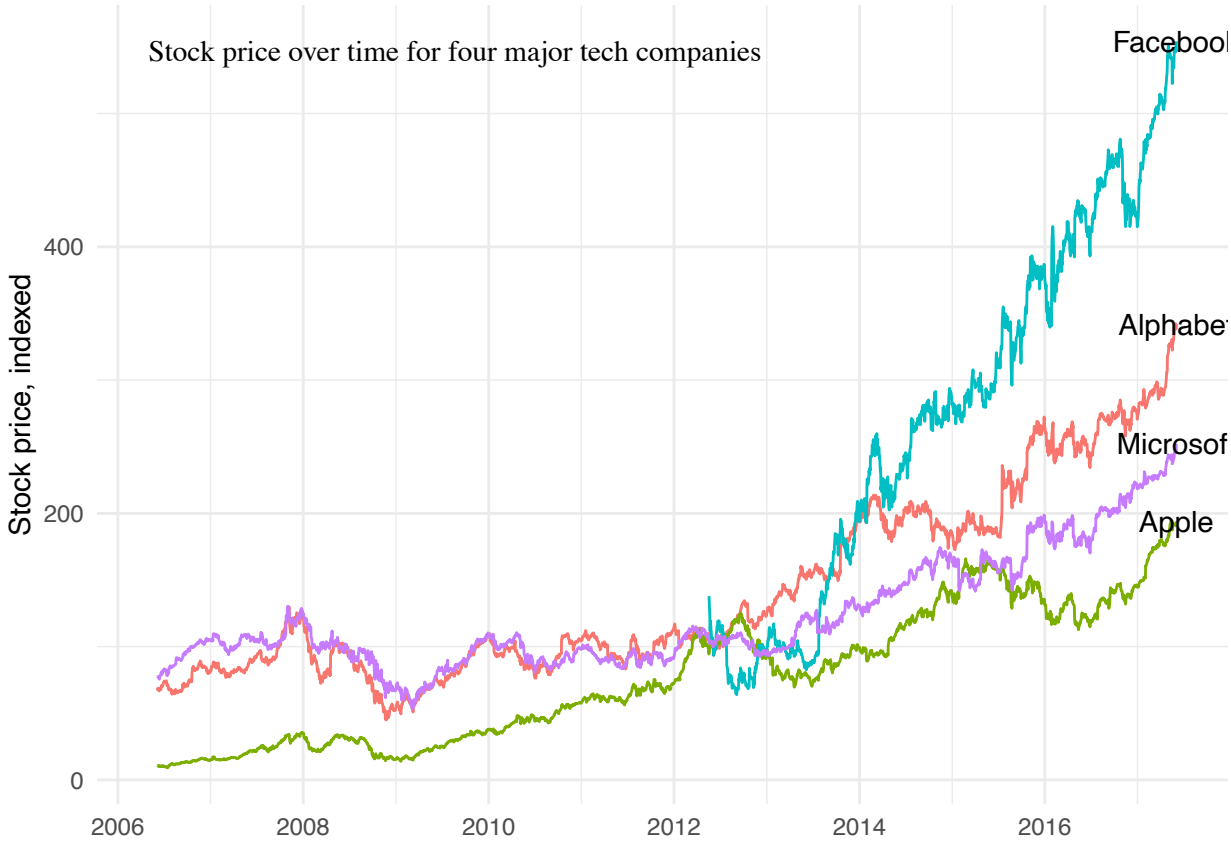
**Plot 1** *Hints:*

- Create a `label_info` dataset that is a subset of original data, just containing the last day's information for each of the 4 stocks
- serif font
- Annotation size is 4

PLOT 1:

```
label_info_ex2 <- tech_stocks %>%
  group_by(ticker) %>%
  filter(date == max(date))

caption_ex2 <- 'Stock price over time for four major tech companies'

#x and y range
xrange2 <- range(tech_stocks$date)
yrange2 <- range(tech_stocks$price_indexed)

tech_stocks %>%

#plot subset of data
  ggplot(aes(x = date, y = price_indexed)) +
  geom_line(aes(color = company)) +
```

```
  labs(x = NULL, y = 'Stock price, indexed') +
  geom_text(data = label_info_ex2, label = label_info_ex2$company) +
  annotate(geom = 'text', x = ymd('2006-05-01'), y = yrange2[2],
           label = caption_ex2, hjust = 0, vjust = 1, size = 4, family = 'serif') +

  theme_minimal() +
  theme(legend.position = 'none')
```

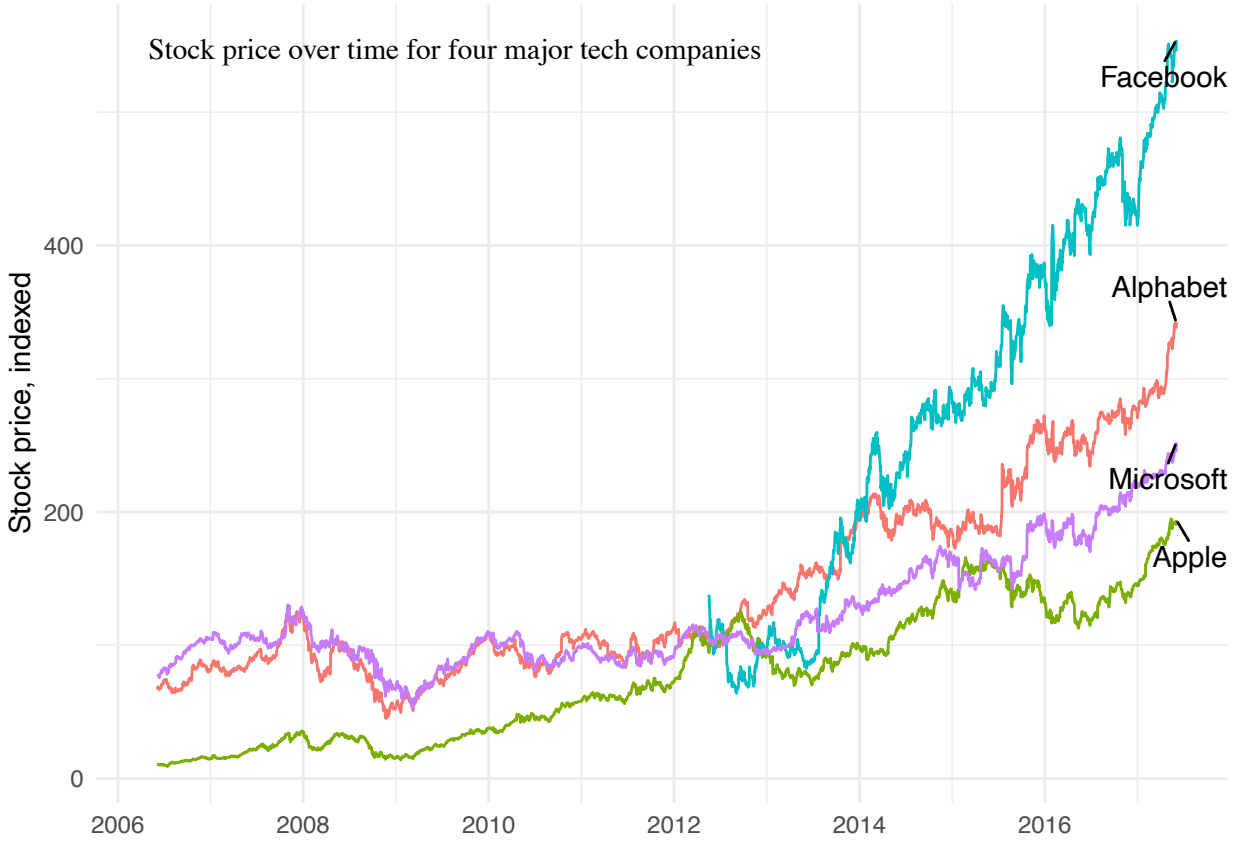



**Plot 2** *Hints:*

- Package `ggrepel`
- Annotation size is 4
- `box.padding` is 0.6
- Minimum segment length is 0
- Horizontal justification is to the right
- seed of 9876

```
tech_stocks %>%

  ggplot(aes(x = date, y = price_indexed)) +
  geom_line(aes(color = company)) +
  labs(x = NULL, y = 'Stock price, indexed') +
  theme_minimal() +
  theme(legend.position = 'none') +

  annotate(geom = 'text', x = ymd('2006-05-01'), y = yrange2[2],
           label = caption_ex2, hjust = 0, vjust = 1, size = 4, family = 'serif') +
```

```
##USE GGREPEL PACKAGE

geom_text_repel(data = label_info_ex2,
                aes(label = company),
                hjust = 'right',
                box.padding = 0.6,
                min.segment.length = 0,
                seed = 9876)
```



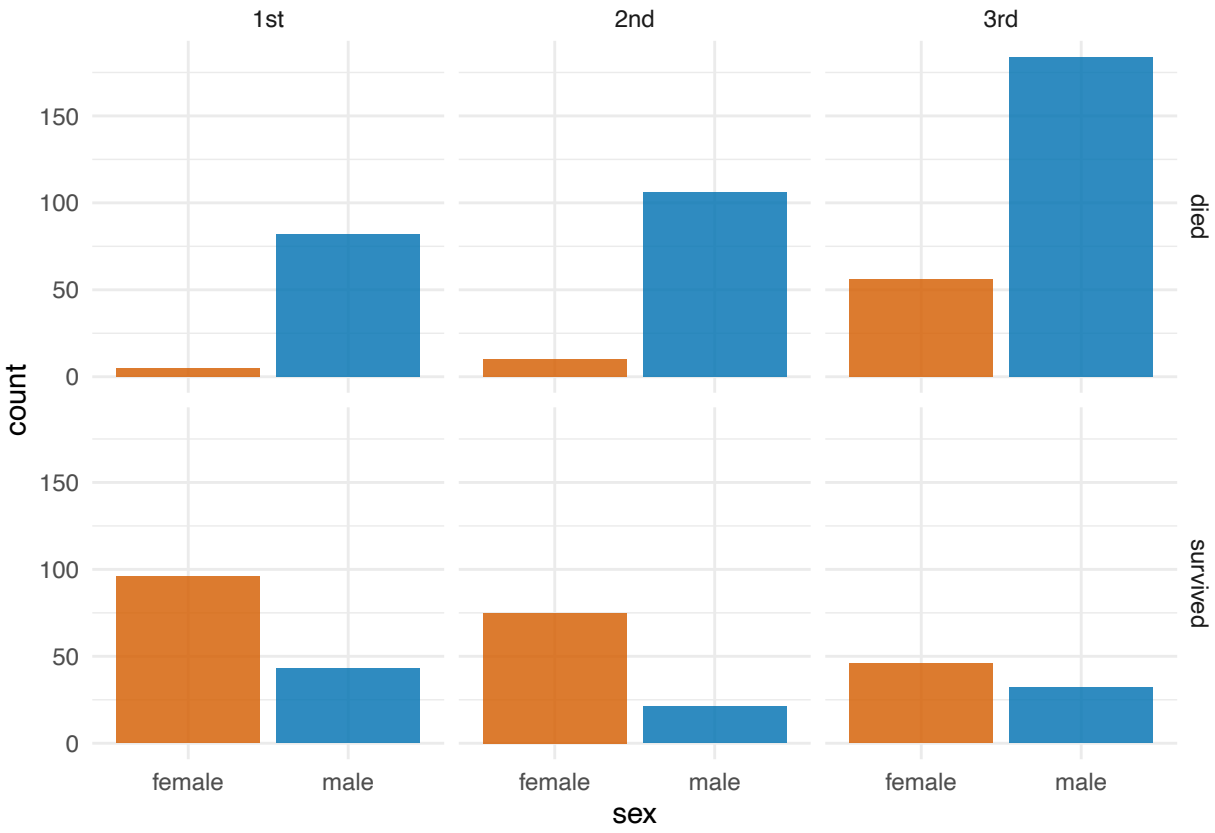

**Exercise 3**

Using the `titanic.rda` dataset, recreate the following graphic as precisely as possible.

*Hints:*

- Create new variable for that uses `died` and `survived` as levels/categories
- Hex colors: `#D55E00D0`, `#0072B2D0` (no alpha is being used)

```
ggplot(titanic, aes(sex, fill = sex)) +
  geom_bar() +
  facet_grid(
    factor(survived, labels = c("died", "survived"))
    ~
    class
  ) +
  scale_fill_manual(values = c("#D55E00D0", "#0072B2D0")) +
  theme_minimal() +
  theme(legend.position = "none")
```




**Exercise 4**

Use the `athletes_dat` dataset — extracted from `Aus_althetes.rda` — to recreate the following graphic as precisely as possible. Create the graphic twice: once using `patchwork` and once using `cowplot`.

```
# Get list of sports played by BOTH sexes
both_sports <- Aus_athletes %>%
  distinct(sex, sport) %>%
  count(sport) %>%
  filter(n == 2) %>%
  pull(sport)

# Process data
athletes_dat <- Aus_athletes %>%
  filter(sport %in% both_sports) %>%
  mutate(sport = case_when(
    sport == "track (400m)" ~ "track",
    sport == "track (sprint)" ~ "track",
    TRUE ~ sport
  ))
```

*Hints:*

- Build each plot separately
- Hex values for shading: `#D55E0040` and `#0072B240` (bottom plot), `#D55E00D0` & `#0072B2D0` (for top two plots) — no `alpha`
- Hex values for outline of boxplots: `#D55E00` and `#0072B2`
- Boxplots should be made narrower; 0.5
- Legend is in top-right corner of bottom plot

- Legend shading matches hex values for top two plots
- Bar plot lower limit 0, upper limit 95; shift bar labels by 5 units and top justify; label size is 5
- `rcc`: red blood cell count; `wcc`: white blood cell count
- Size 3 will be useful

```
##number on bar plot
bar_numb <- athletes_dat %>%
  count(sex)

plotA <- ggplot(athletes_dat, aes(sex, fill = sex)) +
  geom_bar(show.legend = FALSE)+
  scale_fill_manual(values = c("#D55E00D0", "#0072B2D0")) +
  scale_x_discrete(
      name = NULL
    , labels = c("female", "male")
  ) +
  scale_y_continuous(
      name = "number"
    , breaks = seq(0, 100, 25)
    , limits = c(0, 95)
  ) +
  theme_minimal() +
  geom_text(data = bar_numb, aes(y = n, label = n), size = 5, nudge_y = -5, vjust = 'top')
```

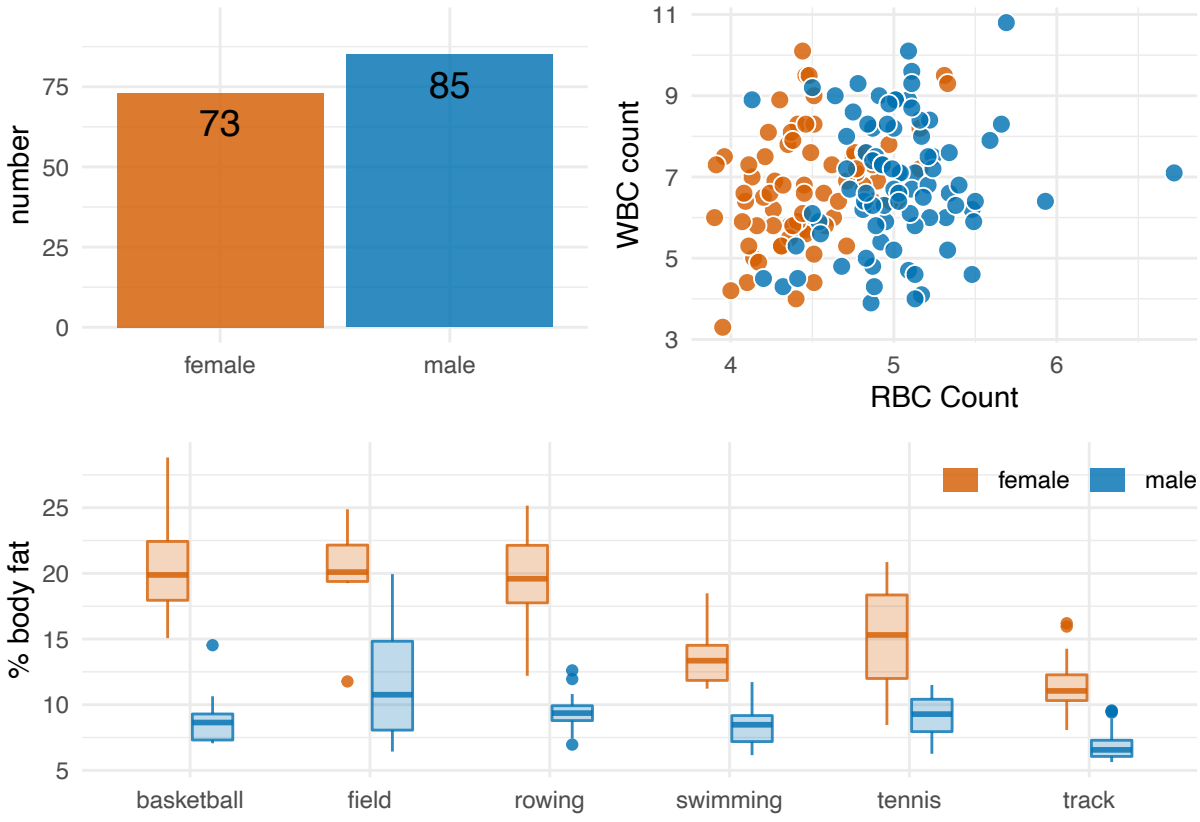
```
plotB <- ggplot(athletes_dat, aes(rcc, wcc)) +
  geom_point(
      aes(fill = sex)
    , shape = 21
    , color = "white"
    , size = 3
    , show.legend = FALSE
    ) +
  scale_fill_manual(values = c("#D55E00D0", "#0072B2D0")) +
  scale_x_continuous(name = "RBC Count") +
  scale_y_continuous(name = "WBC count") +
  theme_minimal()
```

```
sex_labs <- c("female", "male")

plotC <- ggplot(athletes_dat, aes(sport, pcBfat)) +
  geom_boxplot(
      aes(color = sex, fill = sex)
    , width = 0.5
    ) +
  scale_fill_manual(
      name = NULL
    , labels = sex_labs
    , values = c("#D55E0040", "#0072B240")) +
  scale_color_manual(
      name = NULL
    , labels = sex_labs
```

```
   , values = c("#D55E00D0", "#0072B2D0")) +
guides(fill = guide_legend(
    ncol = 2
  , override.aes = list(
      fill = c("#D55E00D0", "#0072B2D0")
    , color = "transparent"
    )
  )
) +
xlab(NULL) +
ylab("% body fat") +
theme_minimal() +
theme(
    legend.position = c(1, 1)
  , legend.justification = c(1, 1)
  #set top legend to zero
  , legend.margin = margin(t = 0)
)
```

**Using `patchwork`** ##USE PATCHWORK:

```
(plotA + plotB) / plotC + plot_layout()
```




**Using `cowplot`** Use `cowplot::plot_grid()` to combine them.

**Exercise 5**

Create the following graphic using `patchwork`.

*Hints:*

- Use plots created in exercise 4
- Useful values: 0, 0.45, 0.75, 1
- inset theme is classic

```
plotB +
  inset_element(plotA + theme_classic(),
                left = 0.75, bottom = 0, right = 1, top = 0.45) +

  plot_annotation(tag_levels = c('A', 'B'))
```