

L06 Annotation & Positioning

Data Visualization (STAT 302)

YOUR NAME

Contents

Overview	1
Datasets	1
Exercises	1

Overview

The goal of this lab is to develop 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 the updated `tech_stocks.rda` datasets.

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

# Load datasets
load('data/blue_jays.rda')
load('data/tech_stocks.rda')
load('data/titanic.rda')
load('data/Aus_athletes.rda')
```

Exercises

Complete the following exercises.

Exercise 1

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

Hints:

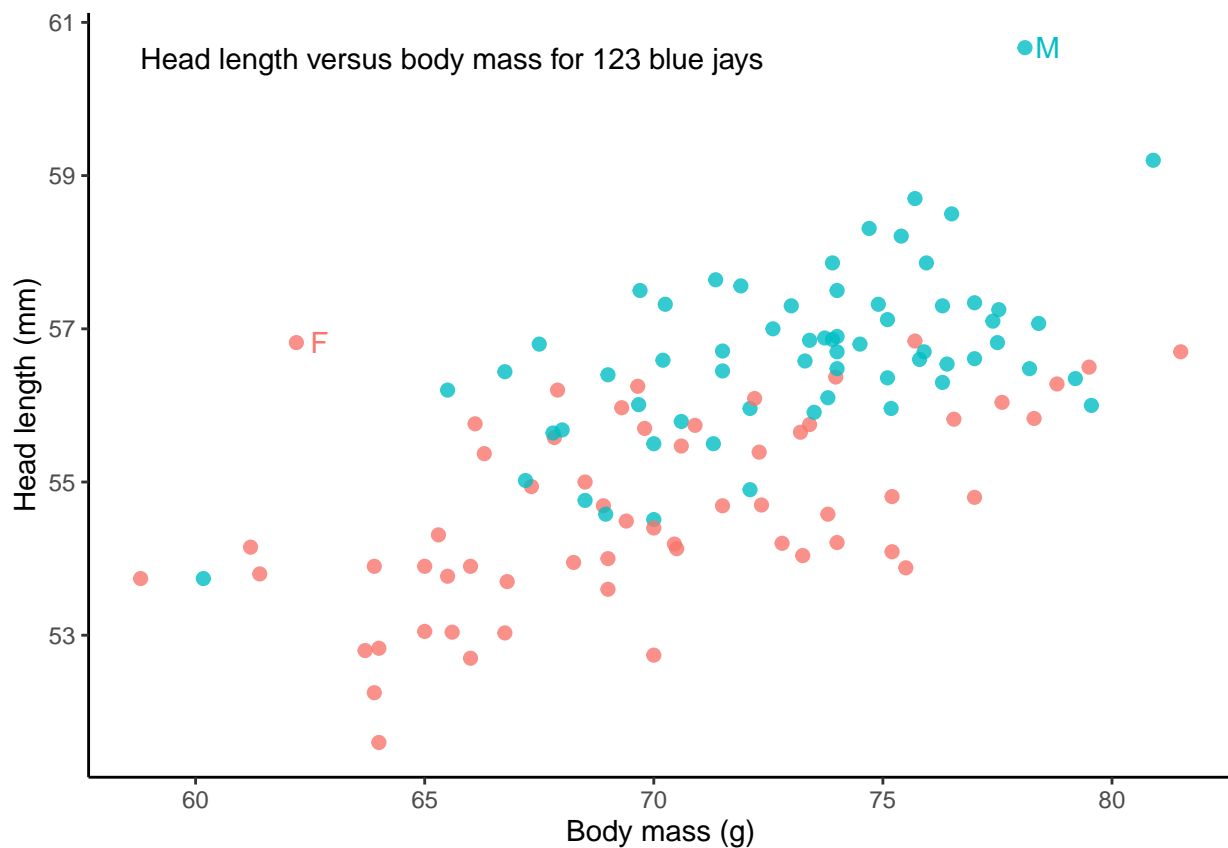
- Transparency is 0.8
- Point size 2
- Create a `label_info` dataset that is a subset of original data, just with the 2 birds to be labeled
- Shift label text horizontally by 0.5
- See 8.3 building custom annotations
- Annotation size is 4
- Classic theme

```
label_info <- blue_jays %>%
  select(Mass, Head, KnownSex) %>%
  arrange(desc(Head)) %>%
  slice(n = c(1, 28))

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

xrng <- range(blue_jays$Mass)
yrng <- range(blue_jays$Head)

ggplot(data = blue_jays, aes(x = Mass, y = Head, color = KnownSex)) +
  geom_point(size = 2, alpha = 0.8) +
  labs(x = 'Body mass (g)', y = 'Head length (mm)') +
  theme_classic() +
  geom_text(data = label_info, label = c('M', 'F'), nudge_x = 0.5) +
  annotate(geom = "text", x = xrng[1], y = yrng[2],
    label = caption, hjust = 0, vjust = 1, size = 4) +
  theme(legend.position = 'none')
```



Exercise 2

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

Plot 1 *Hints:*

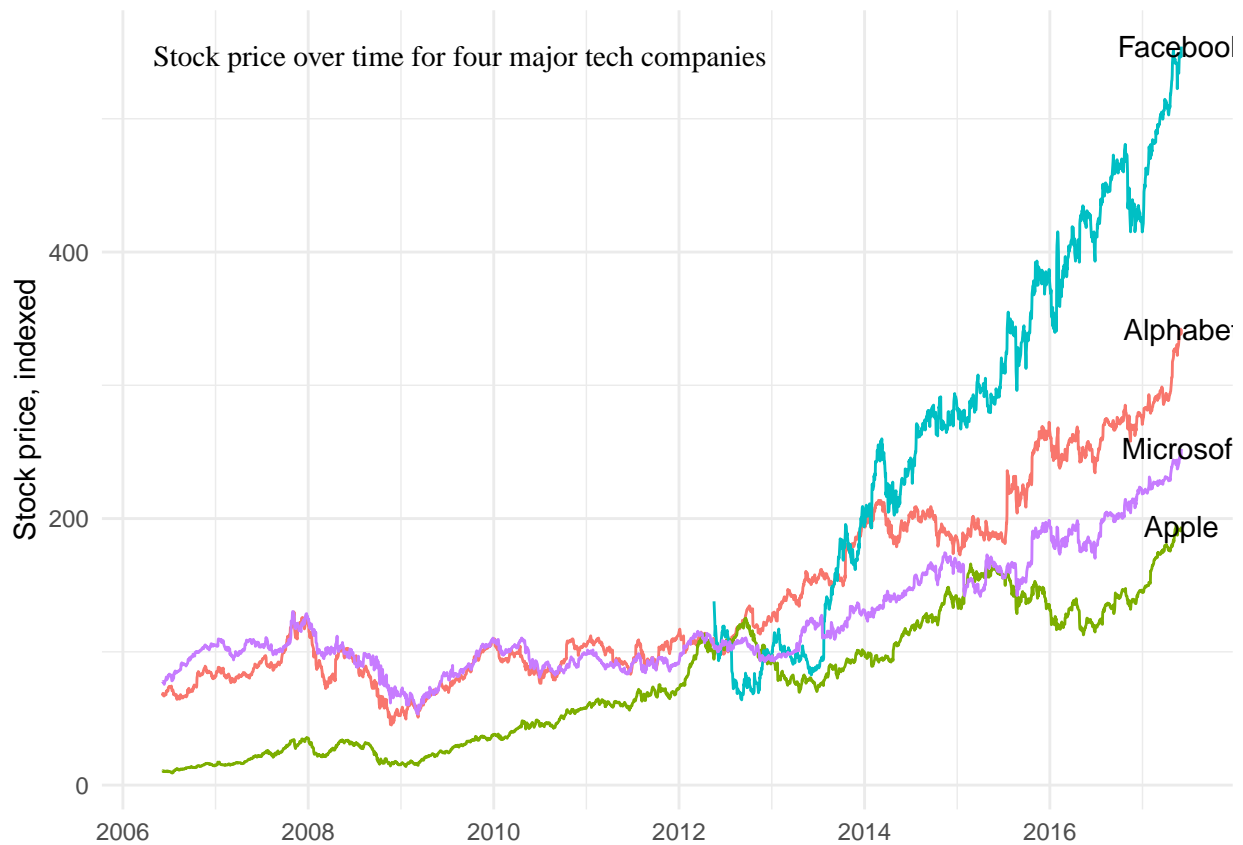
- Create a `label_info` dataset that is a subset of original data, just containing the last day's information for each of the 4 stocks
- serif font
- Annotation size is 4

```
label_info_2 <- tech_stocks %>%
  group_by(ticker) %>%
  filter(date == max(date))

caption_2 <- 'Stock price over time for four major tech companies'

xrng2 <- range(tech_stocks$date)
yrng2 <- range(tech_stocks$price_indexed)

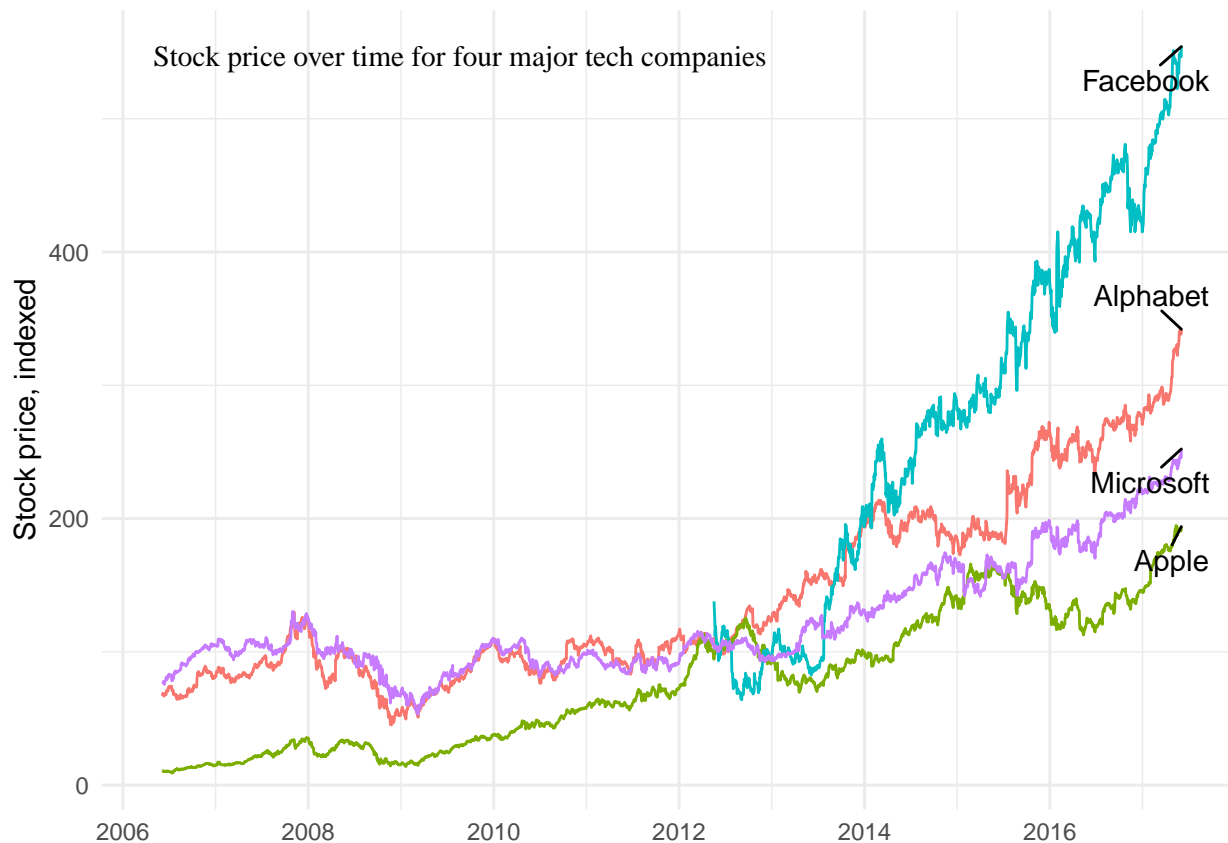
tech_stocks %>%
  ggplot(aes(x = date, y = price_indexed)) +
  geom_line(aes(color = company)) +
  labs(y = 'Stock price, indexed', x = NULL) +
  geom_text(data = label_info_2, label = label_info_2$company) +
  annotate(geom = 'text', x = ymd('2006-05-01'), y = yrng2[2],
    label = caption_2, hjust = 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(y = 'Stock price, indexed', x = NULL) +
  theme_minimal() +
  theme(legend.position = 'none') +
  annotate(geom = 'text', x = ymd('2006-05-01'), y = yrng2[2],
          label = caption_2, hjust = 0, vjust = 1, size = 4, family = 'serif') +
  geom_text_repel(data = label_info_2, aes(label = company), hjust = 'right',
                  box.padding = 0.6, min.segment.length = 0, seed = 9876)
```



Exercise 3

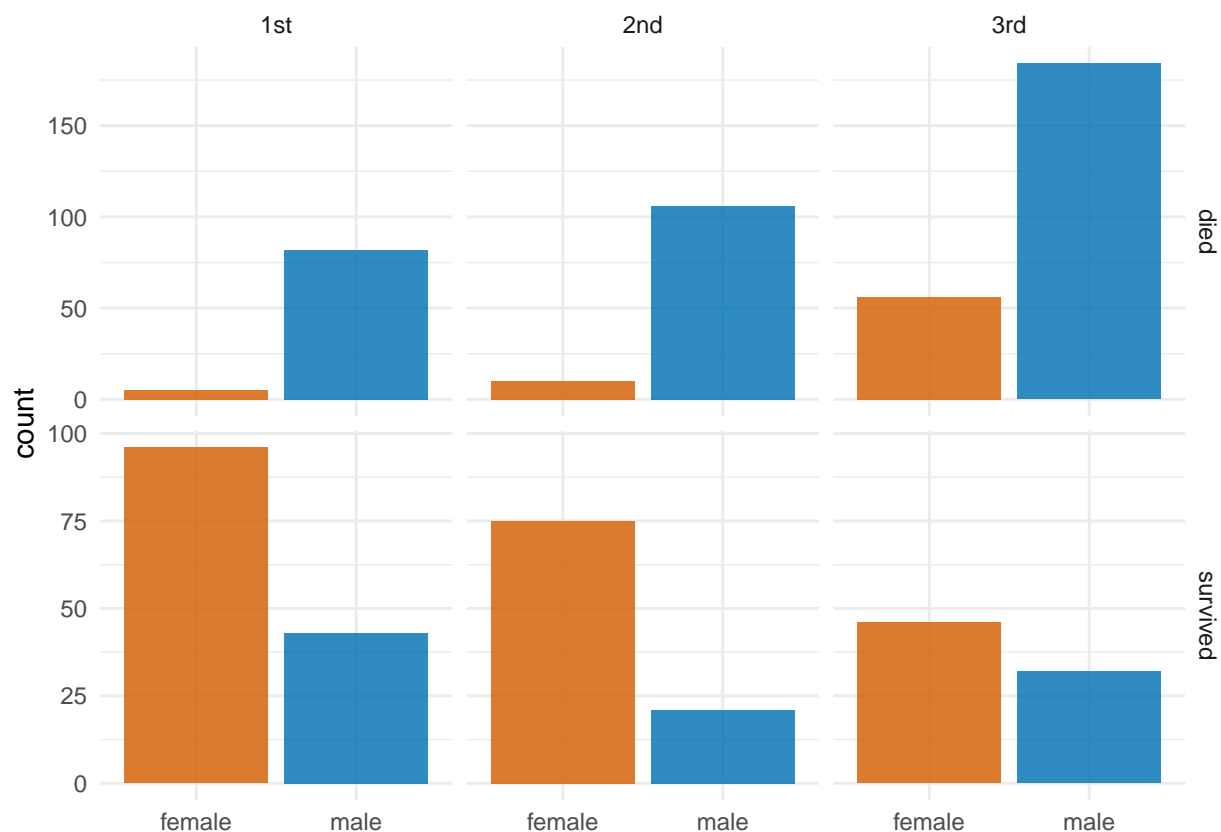
Using the `titanic.rda` dataset, recreate the following graphic as precisely as possible.

Hints:

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

```
titanic <- titanic %>%
  mutate(state = ifelse(survived == 1, 'survived', 'died')) %>%
  mutate(state = as.factor(state))

ggplot(titanic, aes(sex)) +
  geom_bar(aes(fill = sex)) +
  facet_grid(state ~ class, scales = "free") +
  scale_fill_manual(values = c("#D55E00D0", "#0072B2D0")) +
  labs(x = NULL) +
  theme_minimal() +
  theme(legend.position = 'none')
```



Exercise 4

Use the `athletes_dat` dataset — extracted from `Aus_athletes.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

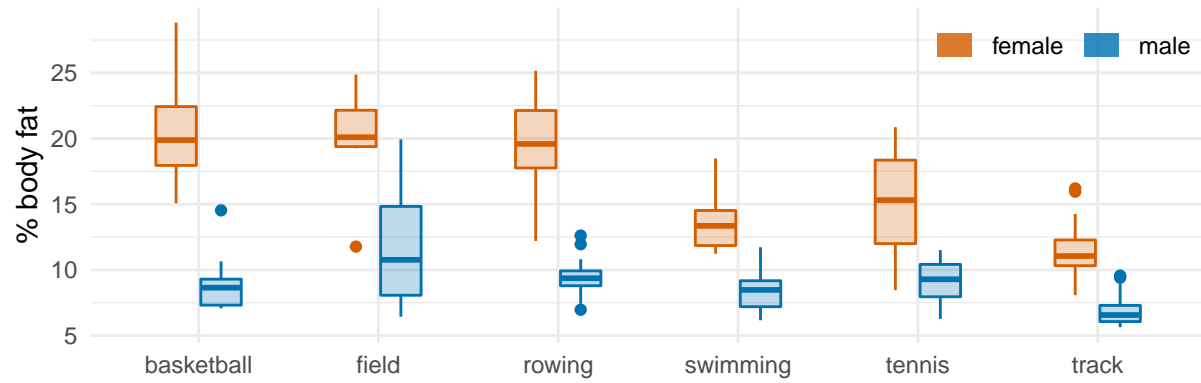
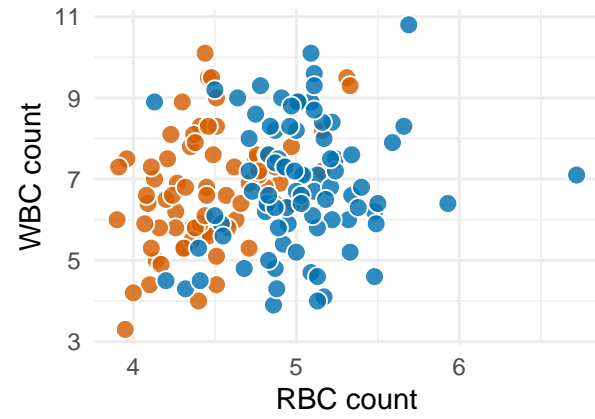
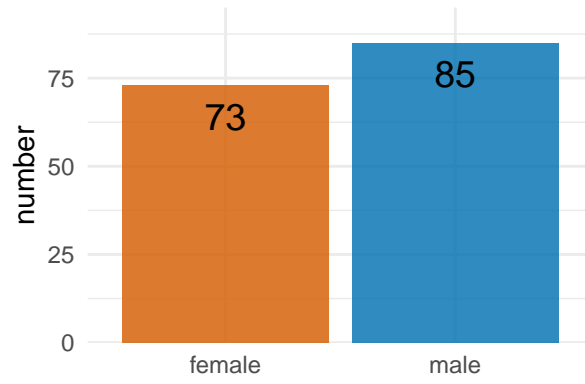
```
#Bar Plot
bar_labels <- athletes_dat %>%
  count(sex)

p1 <- athletes_dat %>%
  ggplot(aes(x = sex)) +
  geom_bar(aes(fill = sex)) +
  geom_text(data = bar_labels, aes(y = n, label = n),
            size = 5, nudge_y = -5, vjust = 'top') +
  scale_x_discrete(labels = c('female', 'male')) +
  scale_y_continuous(limits = c(0, 95),
                    expand = c(0, 0)) +
  scale_fill_manual(values = c('#D55E00D0', '#0072B2D0'),
                   guide = 'none') +
  theme_minimal() +
  labs(x = NULL, y = 'number')

#Scatter Plot
p2 <- athletes_dat %>%
  ggplot(aes(rcc, wcc)) +
  geom_point(aes(fill = sex), size = 3, shape = 21, color = 'white') +
  theme_minimal() +
  labs(x = 'RBC count', y = 'WBC count') +
  scale_fill_manual(values = c('#D55E00D0', '#0072B2D0'), guide = 'none')

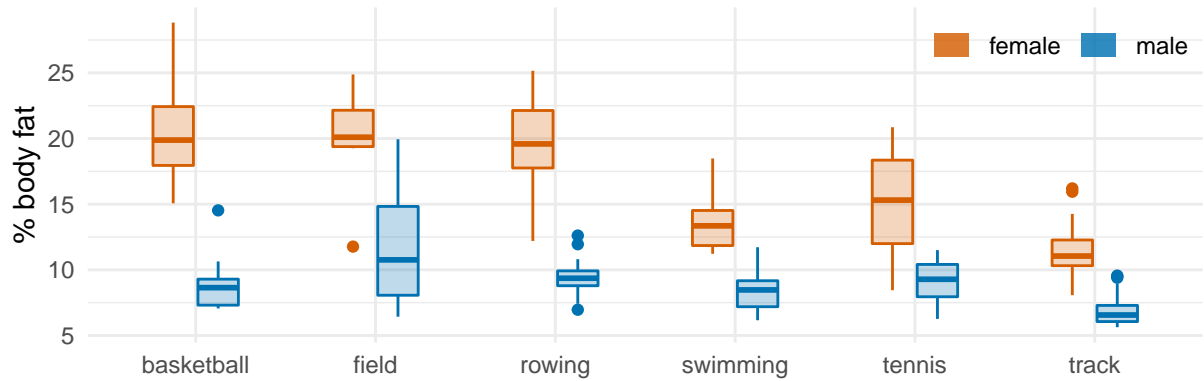
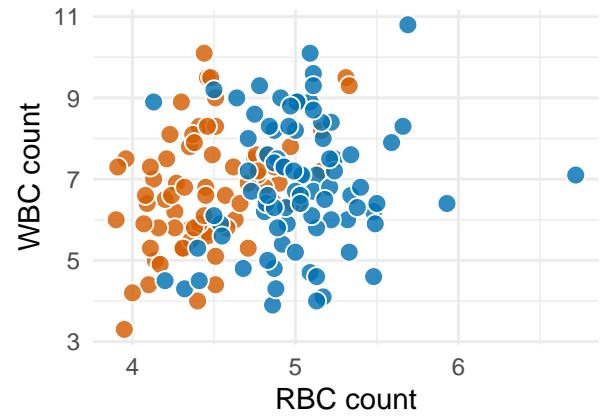
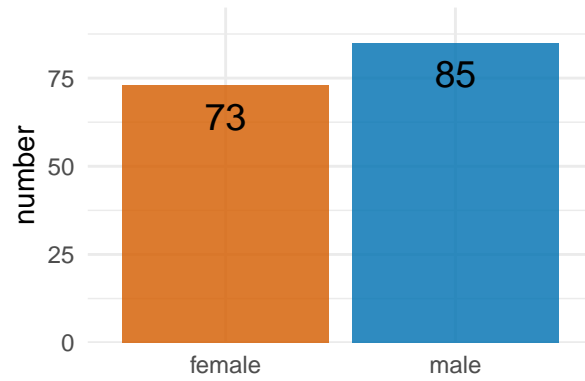
#Box Plot
p3 <- athletes_dat %>%
  ggplot(aes(x = sport, y = pcBfat)) +
  geom_boxplot(aes(fill = sex, colour = sex), width = 0.5,) +
  scale_fill_manual(values = c("#D55E0040", "#0072B240"),
                  name = NULL,
                  labels = c('female', 'male')) +
  scale_color_manual(values = c("#D55E00", "#0072B2"),
                   guide = 'none',
                   name = NULL,
                   labels = c('female', 'male')) +
  labs(x = NULL, y = '% body fat') +
  theme_minimal() +
  theme(legend.justification = c(1, 1),
        legend.position = c(1, 1),
        legend.direction = 'horizontal') +
  guides(color = guide_legend(override.aes = list(color = NA,
                                                  fill = c("#D55E00D0", "#0072B2D0"))))

#Patchwork
(p1 + p2) / p3 + plot_layout()
```



Using patchwork

```
plot_grid((p1 + p2) / p3)
```

Using cowplot

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

```
p2 +
  inset_element(p1 + theme_classic(),
    left = 0.75, bottom = 0, right = 1, top = 0.45) +
  plot_annotation(tag_levels = c('A', 'B'))
```

