Coding together week 3

Warm-up

- Open RStudio and open the project for week 2
- What is in your environment?
- What packages are loaded?
- What directory is RStudio in?
- Please draw a picture or write a description of where you think your project exists? e.g. a flowchart or an analogy.

Data wrangling I

An intro to dplyr:

Transforming tables: filter() picks cases based on their values. arrange() changes the ordering of the rows. select() picks variables based on their names. mutate() adds new variables that are functions of existing variables summarise() reduces multiple values down to a single summary.

Recap: arrange() and filter()

select:

Create an object called surveys small that filters weight less than 5 and selects the columns species_id, sex and weight. Use the pipe.

```
surveys_sml <- surveys %>%
  filter(weight < 5) %>%
  select(species_id, sex, weight)

surveys_sml
```

mutate:

Use mutate to first create a weight_kg variable and then create another variable weight_lb using weight_kg multiplied by 2.2. You don't need to create an object.

```
surveys %>%
  mutate(weight_kg = weight / 1000,
         weight_lb = weight_kg * 2.2)
```

summarise:

Use filter with is.na() to remove the NA values from the weight variable, the use summarise to create mean_weight and min_weight variables, using mean() and min() functions.

```
surveys %>%
  filter(!is.na(weight)) %>%
  summarize(mean_weight = mean(weight),
  min_weight = min(weight))
```

group_by:

Group the surveys data by sex and then use summarise with the n() function to create a count variable, that gives the number of male and female animals.

```
surveys %>%
    group_by(sex) %>%
    summarise(count = n())
```

Use surveys_mutated to group_by rodent_type and then summarrise, we should have 8 species of 2 types.

```
surveys_mutated %>% group_by(rodent_type) %>% summarise()
```

Summative exercise

By semester from 1980 to 2000.

```
surveys %>%
filter(plot_id %in% exp_plots,
year >= 1980 & year <= 2000) %>%
mutate(rodent_type = case_when(
         species_id == "DM" ~ "Kangaroo Rat",
         species_id == "DO" ~ "Kangaroo Rat",
         species_id == "DS" ~ "Kangaroo Rat",
         species_id == "PP" ~ "Granivore",
         species_id == "PF" ~ "Granivore",
         species_id == "PE" ~ "Granivore",
         species_id == "PM" ~ "Granivore",
         species_id == "RM" ~ "Granivore",
         TRUE ~ "Other"),
         date = make_date(day = day, month = month, year = year),
               semester = semester(date, with_year = TRUE)) %>%
               group_by(rodent_type,plot_type,semester) %>%
        summarise(captures = n()/2) %>%
        filter(rodent_type != "Other") %>%
        ggplot(aes(x=semester,y=captures,colour=rodent_type)) +
  geom_line() +
  geom_point() +
  facet_wrap(~ plot_type) +
        theme(legend.position = "bottom") +
        ggtitle("How does excluding Kangeroo Rats effect Granivore populations?",
                subtitle = "Mean half yearly observations")
```

Data wrangling II

This lesson covers:

An intro to tidyr

Pivoting changes the representation of a rectangular dataset, without changing the data inside of it. pivot_longer() Pivot data from wide to long pivot_wider() Pivot data from long to wide

Manipulating character vectors to unite and separate variables: unite() Unite multiple columns into one by pasting strings together separate() Separate a character column into multiple columns using a regular expression separator

Missing values: complete() Complete a data frame with missing combinations of data drop_na() Drop rows containing missing values replace_na() Replace missing values

Joining tables: bind_rows() bind_cols() inner_join() left_join() right_join()

Extras

The `dslabs` package contains a variety of interesting data.

For example there are two tables, one called `murders` containing the number of homicides in 2010 for each state in the USA, and another called `results_us_election_2016` containing the US presidential election results for 2016 for each state.

Let's take a `glimpse()` at these tables:

```
{r dslabs-murders-elections} # Number of homicides in 2010 for each state in the US
glimpse(murders) # US presidential election results for 2016 for each state glimpse(results_us_election
```

Although these are tables of different data, they both have a `state` variable, which means we can use that to join them together and combine the datasets.

For example, in the USA the president is elected not through a popular vote, but via a process called the electoral college by which electoral vote allocation is based upon the US census and the population size in each state such that more populated states have more votes than less populated ones.

We can look at this relationship by joining these two tables, as the `murders` table contains `population` information for each state and the `results_us_election_2016` contains the `electoral_votes` for each state.

```
library(scales)
library(ggrepel)
murders %>% inner_join(results_us_election_2016, by = "state") %>%
        ggplot(aes(x = population, y = electoral_votes,label = abb)) +
        geom_point() +
        geom_text_repel() +
        geom_smooth(method = "lm",se = FALSE) +
        scale_x_log10(labels = comma_format()) +
        scale_y_log10()
```