# Coding together week 4

## Warm-up

- Complete the questionaire and quiz
- Open RStudio and open the project for week 2
- How many distinct taxa are there in the surveys dataset?
- How would we do the grouped summary we did at the end of week three, but only for the rodents?

## Data wrangling II

### 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.

### 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.

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

## Summative exercise

Repeat the analysis by semester from 1980 to 2000.

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

### pivot_longer()

Using an example from the tidyr website:

relig_income is a dataset that comes with tidyr that contains observations of the income bands for 18 religons.

Inspect it by typing: relig_income and pressing enter

Let's consider the income bands that are column headings as a variable. Pivot them into a single variable called "income" with the values as a variable called "count". Don't use the religon variable.

```
relig_income %>%
        pivot_longer(names_to = "income", values_to = "count" , -religion)
```

## pivot_wider()

Using an example from the tidyr website:

fish_encounters is another dataset that come with tidyr contributed by Myfanwy Johnston, describes when fish swimming down a river are detected by automatic monitoring stations.

It only records when fish are detected, and not when they aren't. It has three variables: fish, station and seen

Pivot the table wider such that the station become variable names and the values in the new columns are from seen.

Pivot the table again, but this time fill in the missing values with zeros. Use ?pivot_wider to find out how to do this. Or google.

```
fish_encounters %>%
        pivot_wider(names_from = station, values_from = seen)
```

```
fish_encounters %>% pivot_wider(
  names_from = station,
  values_from = seen,
  values_fill = list(seen = 0)
)
```

## Missing values

## Joins