# starting with data

# the survey dataset

- One row per individual

| Column | Description |
|---|---|
| record_id | Unique id for the observation |
| month | month of observation |
| day | day of observation |
| year | year of observation |
| plot_id | ID of a particular plot |
| species_id | 2-letter code |
| sex | sex of animal (“M”, “F”) |
| hindfoot_length | length of the hindfoot in mm |
| weight | weight of the animal in grams |
| genus | genus of animal |
| species | species of animal |
| taxon | e.g. Rodent, Reptile, Bird, Rabbit |
| plot_type | type of plot |

# downloading the dataset

We are going to download the file to our `data_raw` sub folder:

```
download.file(url = "https://ndownloader.figshare.com/files/2292169",
              destfile = "data_raw/portal_data_joined.csv")
```

We can read it using function `read.csv()`

```
surveys <- read.csv("data_raw/portal_data_joined.csv")
```

# reading files into R

Functions to read data are key to any project.
for dataframes: `read.csv()`, `read.delim()`

```
surveys <- read.csv("data_raw/portal_data_joined.csv")
surveys_check <- read.table(file = "data_raw/portal_data_joined.csv",
                            sep = ",",
                            header = TRUE)
identical(surveys, surveys_check)
```

```
## [1] TRUE
```

There are **many other ways** to read data into R, some are specific for the type of data (GIS shapefiles or raster, and specific packages may come with their own reader functions)

# inspecting `data.frame` objects

Based on the output of `str(surveys)`, can you answer the following questions?

- What is the class of the object surveys?
- How many rows and how many columns are in this object?
- How many species have been recorded during these surveys?
    - `data.frame`
    - 34786 rows and 13 columns
    - Number of species?

# indexing and subsetting data frames

- a vector has only one dimension, so:
  - `length()` refers to number of **elements**
  - `dim()`
  - selection between brackets `[]`
- a data.frame has **two** dimensions, so `dim()`, `ncol()`, `nrow()` and selection between brackets `[]` BUT with the two dimensions separated by a comma: `[rows, columns]`

# indexing and subsetting data frames

```
# first element in the first column of the data frame (as a vector)

# first element in the 6th column (as a vector)

# first column of the data frame (as a vector)

# first column of the data frame (as a data.frame)

# first three elements in the 7th column (as a vector)

# the 3rd row of the data frame (as a data.frame)

# equivalent to head_surveys <- head(surveys)
```

# indexing and subsetting data frames

```
# first element in the first column of the data frame (as a vector)
surveys[1, 1]
# first element in the 6th column (as a vector)
surveys[1, 6]
# first column of the data frame (as a vector)
surveys[, 1]
# first column of the data frame (as a data.frame)
surveys[1]
# first three elements in the 7th column (as a vector)
surveys[1:3, 7]
# the 3rd row of the data frame (as a data.frame)
surveys[3, ]
# equivalent to head_surveys <- head(surveys)
head_surveys <- surveys[1:6, ]
```

# indexing and subsetting data frames

- minus sign to **remove** the indexed column or row

```
# The whole data frame, except the first column
surveys[, -1]
surveys[-(7:34786), ] # Equivalent to head(surveys)
```

# subsetting by name

```
surveys["species_id"]       # Result is a data.frame
surveys[, "species_id"]     # Result is a vector
surveys[["species_id"]]     # Result is a vector
surveys$species_id          # Result is a vector
```

- R has several ways to do some things

# challenge

- Create a data.frame (`surveys_200`) containing only the data in row 200 of the `surveys` dataset
- Notice how `nrow()` gave you the number of rows in a data.frame? Use that number to pull out just that last row in the data frame
- Compare that with what you see as the last row using `tail()` to make sure it's meeting expectations
- Pull out that last row using `nrow()` instead of the row number.
- Create a new data frame (`surveys_last`) from that last row.
- Use `nrow()` to extract the row that is in the middle of the data frame. Store the content of this row in an object named `surveys_middle`.
- Combine `nrow()` with the - notation above to reproduce the behavior of `head(surveys)`, keeping just the first through 6th rows of the surveys dataset.

# factors

- **factors**: vectors (one-dimensional) representing **categorical variables** and thus having **levels**. ordered or unordered (`c("low", "medium", "high")`

- R < 4.0 had a default behavior `stringsAsFactors = TRUE` so any character column was transformed into a factor

```
`?read.csv()`
?default.stringsAsFactors
```

# today if we want factors we have to transform the vectors

# factors

```
## Compare the difference between our data read as
#`factor` vs `character`.
surveys <- read.csv("data_raw/portal_data_joined.csv",
                    stringsAsFactors = TRUE)
str(surveys)
surveys <- read.csv("data_raw/portal_data_joined.csv",
                    stringsAsFactors = FALSE)
str(surveys)
## Convert the column "plot_type" and sex into a factor
surveys$plot_type <- factor(surveys$plot_type)
surveys$sex <- factor(surveys$sex)
```

# working with factors

```
sex <- factor(c("male", "female", "female", "male"))
levels(sex) # in alphabetical order!
nlevels(sex)
sex
sex <- factor(sex, levels = c("male", "female"))
sex # after re-ordering
as.character(sex)

year_fct <- factor(c(1990, 1983, 1977, 1998, 1990))
as.numeric(year_fct)                # Wrong! And there is no warning...
as.numeric(as.character(year_fct)) # Works...
as.numeric(levels(year_fct))
as.numeric(levels(year_fct))[year_fct]    # The recommended way.
```

# let's make a plot of a factor variable

```
plot(as.factor(surveys$sex))
```

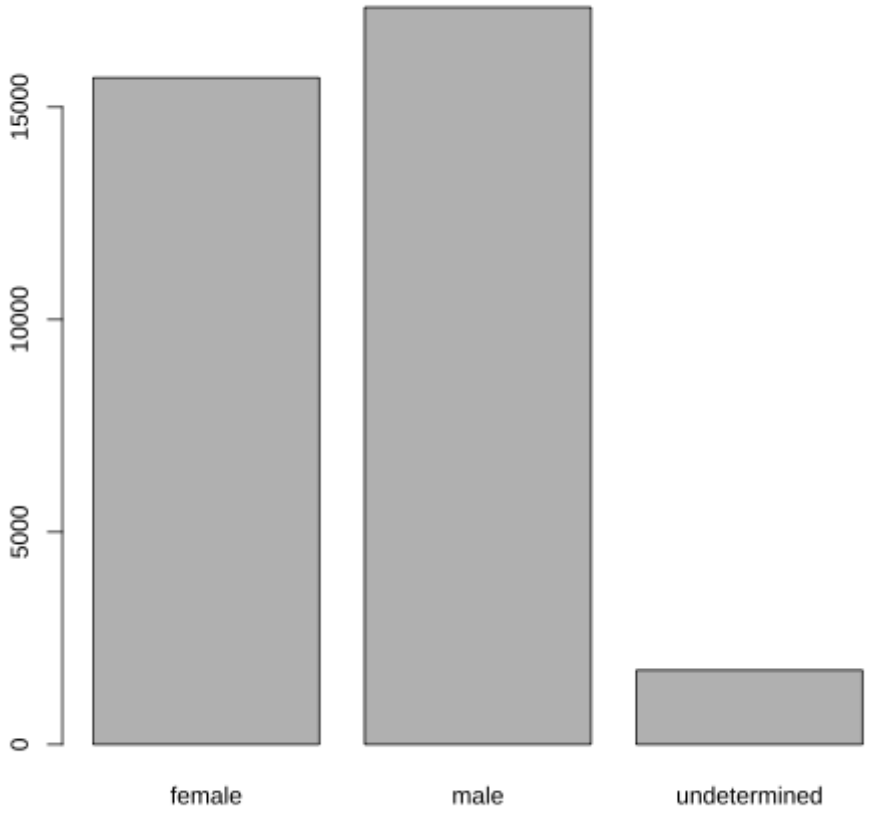
let's rename this label

# let's make a plot of a factor variable

```
plot(sex)
```

let's rename this label

# challenge




- Rename “F” and “M” to “female” and “male” respectively.
- Now that we have renamed the factor level to “undetermined”, can you recreate the barplot such that “undetermined” is last (after “male”)?