Exploratory Data Analysis - hands-on!

YOUR NAME

2023-09-26

Outline for this document

- Quick recap
- EDA Part 1: mass and metabolic rate in *Artiodactyla*
- EDA Part 2: shrub volumes

Recap - what you've learned so far

- How to use R as a calculator.
- About data types of variables : factors, strings or characters, numeric, logical.
- About data structures: vectors, matrices, data.frames, and lists, arrays, and tibbles.
- How to create data structures and index their elements.
- How to create a basic R script with code and comments.
- How to read into data saved in a a tabular format (e.g. `.csv`)
- How to have a general look at the imported data and understand its structure.
- How to extract parts of the data for further analysis.
- About another type of script to write R code and text: R markdown
- How to handling missing data in some of the functions we've learned so far and how to eliminate NAs altogether.
- The ideal ways to save data files for analysis in R.
- A soft introduction to the Tidyverse suit of packages, focusing on the `dplyr` and `ggplot` packages.

Tasks

- Complete the Data Camp Activities set as due 09/21.
- Work on the hands-on in class project
- Work on or ask questions about LA1
- Upload to Moodle this file knitted into a PDF **AND** the final modified `.csv` you will create.

Data Camp activities

- Data manipulation with `dplyr`: Transforming Data with dplyr - covers `select`, `filter`, `mutate`, `arrange`.
- Data manipulation with `dplyr`: Aggregating data - covers `group_by`, `summarize`, etc
- Introduction to data visualization with `ggplot2` will give you many more tools to use with `ggplot2`.

Applying what you've learned

Today we will be using these eight functions:

```
library()
read_csv()
select()
filter()
mutate()
group_by()
summarise()
n()
ggplot()
```

We will also be working directly within this R Markdown document to help you get more familiar with it for your lab assignments.

Before we get started:

- This is an R markdown file, so for each exercise, write your code inside a code chunk. I've set these up for you but, as a reminder, you can create a code chunk by clicking at the "+C" green rectangle just above this area of your interface. Notice that I've given each chunk a label (something like `ex1`, etc) and set `echo=TRUE` (your code will appear in the PDF) and `eval=TRUE` (your code will be run when the PDF is compiled).
- Before you modify this file at all, go ahead and click "knit" in the button above and watch this file become a beautiful PDF! This is what you will do at the end when you're done, so make sure you knit this file often to pick up any errors as they come.
- Replace `YOUR NAME` in the header of this file with your actual name. Then knit the file again and see your name appear!
- I suggest that, after you make a plot, you save it with a descriptive name as a `.png` using `ggsave`. This will not be required but it's a good practice for you. It will save your plots into this workspace.
- Get in the habit of commenting your code.

Exercise 1: Mass vs Metabolism

The relationship between the body size of an organism and its metabolic rate is one of the most well studied and still most controversial areas of organismal physiology.

We want to graph this relationship in the *Artiodactyla* using a subset of data from a large compilation of body size data (Savage et al. 2004). The data are in the `mass_metabolism.csv` file in your workspace.

Make the following manipulations and plots with appropriate axis labels, or answer the questions.

Ex.1) Make sure `ggplot2` and `dplyr` are installed. The easiest way is to just use the `tidyverse` package, which installs those two and more for you:

```
#install the tidyverse package here:

#after you do this, you can comment the line so it doesn't re-install this every time you
↪  knit your document.
#then, load the tidyverse package here:
```

Ex.2) Read in the dataset using the `read_csv` function. Assign the data into an object called `mass_metabolism`. Have a look at the first 6 rows of the data. What kinf of data structure is this? How many columns are there?

```
# type your code here
```

Answer:

Ex.3) What are the dimensions (# rows, # columns)

```
# type your code here
```

Answer:

Ex.4) Check whether `mass_metabolism` is a tibble. Hint: you can use the `is_tibble` function. Hint: you will need to install and load the `tibble` package first.

```
# type your code here
```

Answer:

Ex.5) What are the classes of each variable in `mass_metabolism`?

```
# type your code here
```

Answer:

Ex.6) Convert the `family` column into the `factor` data type and assign that new object to `mass_metabolism2`. Use the function `mutate` to accomplish this change. What are the levels of the variable `family`?. Hint, you can figure this out with the command `levels(your_object_name)`. Save the levels into an object called `mass_metab_levels`

```
# type your code here
```

Answer:

Ex.7) Obtain a summary of each variable (column) in `mass_metabolism` and `mass_metabolism2`. What is the main difference between the two and why?

```
# type your code here
```

Answer:

Ex.8) Check whether the dataset has any missing data. Are there missing data? Save your answer into a variable called `mass_metab_na`, assigning the value `TRUE` is there are missing values and `FALSE` if not.

```
# type your code here
```

Answer:

Ex.9) Explore the variable body_mass: what is the range of the values? Save your answer into a variable called `range_mass`. The range is the difference between maximum and minimum values. Then, create a separate object called `range_mass_bovidae` containing the body mass range only for members of the Bovidae family. Look at both summaries and answer this: does it look like the Bovidae individuals are the ones making the range in body mass so huge?

```
# type your code here
```

Ex.10) Make a plot for body mass for members of the `Bovidae` family only. What does this distribution look like? Hint: Use `filter` to obtain the data points of interest and then use `ggplot` to plot the distribution.

```
#type your code here
```

Ex.11) Explore the variable `metabolic_rate`. What is the range of the values? Save your answer into a variable called `range_metab`. Then, create a separate object called `range_metab_bovidae` containing the metabolic rate range only for members of the Bovidae family. Look at both summaries and answer this: does it look like the Bovidae individuals are the ones making the range in body mass so huge?

```
#type your code here
```

Answer:

Ex.12) Make a plot for metabolic rate for members of the `Bovidae` family only. What does this distribution look like? Hint: Use `filter` to obtain the data points of interest and then use `ggplot` to plot the distribution.

```
#type your code here
```

Answer:

Ex.13) Using the entire dataset, make a plot of metabolic rate vs. body mass: which variable should be the response variable and which should be the explanatory variable?

```
#type your code here
```

Answer:

Ex.14) Make a plot of body mass vs. metabolic rate, with log10 scaled axes (this stretches the axis, but keeps the numbers on the original scale), and the point size set to 3. Hint: use `scale_y_continuous(trans='log10')` and `scale_x_continuous(trans='log10')`. You may need to read a little about what these do.

```
#type your code here
```

Ex.15) Make the same plot as in Ex. 14, but with the different families indicated using color.

```
#type your code here
```

Ex.16) Make the same plot as in Ex.14, but with the different families each in their own subplot. Hint: to achieve this, use `facet_wrap`.

```
#type your code here
```

Ex.17) Using `group_by`, `summarise`, and `group_by`, calculate the range of body mass for each family as well as the number of rows for each family. Save the output to an object called `body_mass_summary`. Why do you think some ranges are zero?

```
#type your code here
```

Exercise 2: Carbon storage in shrubs

Dr. Granger is interested in studying the factors controlling the size and carbon storage of shrubs. She has conducted an experiment looking at the effect of three different treatments on shrub volume at four different locations. The data are in the file `shrub-volume-data.csv` in your workspace files.

Ex.18) Using the previous activity (Ex. 2-6) as guide, read in the data and assing it to an object called `shrubs`. Have a first look at the data, look at it's class, dimensions, structure (class of each column), etc. Do the data types for `site` and `experiment` make sense to you? If not, change them to something that does make sense and save the new object to `shrubs2`. Then, run summary on both `shrubs` and `shrubs2`

```
#type your code here
```

Answer:

Ex.19) Before moving further, look at the output from the previous question. Do you see NA values? If so, filter them out and save the clean object into `shrubs3`.

Next, explore the variables `length`, `width`, `height` one at a time. Make a histogram for each. Note: use 10 bins in each histogram.

```
#create shrubs3

#length

#width

# height
```

Ex.20) Using `group_by`, `summarise`, `n()`, create an object containing the range of values for `length`, `width`, `height` and the number of rows per experiment.

```
#type your code here
```

Ex.21) Add a new column named `area` containing the area of the shrub, which is the `length` times the `width` (using `mutate`). Then, filter the data to include only plants with heights greater than 5 (using `filter`). Finally, sort the data by length (using `arrange`). Save the result into object `shrubs4`.

```
#type your code here
```

Ex.22) Filter the data in `shrubs3` to include only plants with heights greater than 4 and widths greater than 2 (using `,` or `&` to include two conditions). Save the result into object `shrubs5`.

```
#type your code here
```

Ex23) Filter the data in `shrubs2` to include only plants from experiment 1 or experiment 3 (using `|` for `or`). Save the result into object `shrubs6`.

```
#type your code here
```

Ex.24) Filter the data in `shrubs` to remove rows with null values in the height column (using `!is.na`).

```
#type your code here
```

Ex.25) Create a new data frame called `shrub_volumes` that includes all of the original data (before any filtering), plus the `area` column you created above, and a new column containing the volumes (length * width * height), and display it. Assign it to an object `shrub_volumes`.

```
#type your code here
```

Ex.26) Save your new data frame as a cvs file called `shrub_volumes.csv`. Now you have a modified data file without having modified the raw data! Remember: always keep the original raw data file unmodified. You can create as many modified files as you want, but don't mess with that one!

The end!

- Knit your document into a PDF using the button above this text editor and upload the PDF into Moodle.
- Also upload your `shrub_volumes.csv` file.

References

Savage, Van M., et al. "The predominance of quarter-power scaling in biology." Functional Ecology 18.2 (2004): 257-282.

Data Carpentry for Biologists. https://datacarpentry.org/semester-biology/exercises/Dplyr-shrub-volume-data-basics-R/

Data Carpentry for Biologists. https://datacarpentry.org/semester-biology/exercises/Graphing-mass-vs-metabolism-R/

Stack Overflow. In R markdown in RStudio, how can I prevent the source code from running off a pdf page?