Exercise 2

No Fear of Numbers: Introduction to Quantitative Data Analysis in R

Dr. Susanne de Vogel, Data Science Center, University of Bremen

2 December 2025

Research Question: Is there a Life beyond the PhD?

In this workshop, we will work with one coherent example throughout all four parts.We focus on two outcomes of interest:

- `bpsy01`: overall life satisfaction (10 point scale)
- `blcd06`: Children (yes/no)

We will examine how these outcomes are related to different aspects of doctoral researchers' lives and backgrounds:

PhD and Work conditions

- `adbi01`: Status of the doctorate
- `adbi15`: Discipline
- `bdcd17`: Emotional Support during PhD
- `bwdr12`: Perceived scientific pressure
- `bemp81`: Gross income

Attitudes and Well-Being

- `bldc12`: Satisfaction with work-life balance
- `apar15b`: Relationship with parents
- `bpsy05`: Self-Efficacy

Demographics

- `adem01`: Gender
- `adem02`: Age in years
- `apar10`: Highest vocational degree of parents
- `adem03`: Country of birth
- `blcd01`: Relationship status

Univariate Analysis

In the previous exercise, we got to know our dataset and the main variables. In this sheet, we focus on univariate analysis looking at **one variable at a time**. Univariate analysis is the starting point of any data analysis. It helps us **understand distributions**, typical values, and strange values and is the basis for all further steps: bivariate relationships and regression models.

In the context of our research story, we are interested in **overall life satisfaction** (`blcd01`) and **having children (yes/no)** (from `blcd06`). Before we relate these outcomes to other factors, we first want to understand

- How are these variables distributed in our sample?
- How are other key variables (e.g. gender, age, burden, work–life compatibility) distributed?

In this exercise, you will:

- transform categorial variables to factors
- describe nominal variables using absolute and relative frequencies, mode, and bar charts,
- describe ordinal variables using cumulative frequencies, median, percentiles,
- describe metric variables using mean, standard deviation, percentiles, boxplots, histograms, skewness and kurtosis, standard error, confidence intervals

2.1. Set-up: Load packages and dataset

First, we (install and) load the packages `tidyverse` and `haven`, as well as `janitor`, `sjmisc`, `scales`, `skimr` and `moments`. These packages are tidyverse-friendly but not part of the core tidyverse packages, so they need to be installed and loaded separately.

1. Install `tidyverse`, `haven`, `janitor`, `sjmisc`, `scales`, `skimr` and `moments` if not already installed and load these packages.

2. For this exercise, we will use the file *`02_qa_uni_data.sav`* in the *exercises* folder. Define the *exercise* folder as your working directory with the function `setwd("path")`.

3. Choose `mydata2` as a name for your data frame and import it with the following structure: `chosen_name <- read_sav("filename")`.

2.2. Nominal variables: Children and gender

1. Calculate absolute and relative frequencies for `blcd06` children (1 = yes/ 2 = no).

Use the function `tabyl()` from the *janitor* package to create a frequency table for `blcd06`. Look at the counts and percentages in the output, what does it tell you?

2. Alternative: Calculate Absolute and relative frequencies for `adem01` gender *with labels*

`tabyl(adem01)` will show the **numeric codes**, which is not very intuitive. To make the table easier to read, we want to display the **variable and value labels** instead.

2a. To do so, first convert `adem01` into a *factor* with labels.

Use `mutate()` and `adem01 = as_factor(adem01, levels="labels")` to transform `adem01` into a factor variable. Save the result again as `mydata2`. The code should look like this:

```r
# convert the labelled variable into a factor whose levels are the value labels (e.g. "female",
↪   "male", …)
mydata2 <- mydata2 %>%
  mutate(
    adem01 = as_factor(adem01, levels="labels")
  )
```

2b. Create a frequency table for `adem01` using `tabyl()`.

Compare this to the table for `blcd06` with numeric codes. Decide which version is easier to read.

3. Identify the *mode* of `blcd06`and `adem01`

The modus is the category that appears most often. As there are variables with little categories, we can easily identify it by looking at the frquency tables we created with `tabyl()`. For variables with many categories, you can identify the mode by sorting the table so that the category with the highest `n` comes first.

For gender `adem01` the categories are accidentally sorted from largest to smallest. The most frequent category is `female`.

For `alcd06`, create a `%>% pipe` where you first create a table with the `tabyl()` function for `alcd06` and then sort this table by `n` (from largest to smallest) using the `arrange(desc(n))` function. The code should look like this:

```r
# identify mode
mydata2 %>%
  tabyl(blcd06) %>% # frequency table
  arrange(desc(n)) # sort in descending order
```

What is the most frequent category of `blcd06`?

4. Visualize the distribution of `blcd09` children (yes/no) in a bar chart.

Draw a bar chart with `blcd06` on the x-axis using `ggplot()` function. The height of the bars should show how many people are in each category (1/2). The code should look like this:

```r
mydata2 %>% # take mydata2 as input
  ggplot(aes(x = blcd06)) + # put blcd06 on x-axis
  geom_bar() + # draw bars, counting tows per category
  labs( # add labels and title
    x = "Children (1 = yes, 2 = no)", # label for x-axis
    y = "Number of respondents", # label for y-axis
    title = "Distribution of having children (yes/no)" # main title of the plot
  )
```

What do you notice?

5. Create a similar bar chart for the distribution of gender `adem01`. What do you notice?

6. Modify the graph so 1.) missings do not appear and 2.) percentages are plotted.

To do so, you need to remove missing values out of `adem01`, count the cases in the category and use these information for plotting the bar chart. Modify the `ggplot()` function like this:

```r
mydata2 %>%
  filter(!is.na(adem01)) %>%              # remove missing values of adem01
  count(adem01) %>%                       # count how many cases in each category
  mutate(prop = n / sum(n)) %>%           # compute proportion (relative frequency)
  ggplot(aes(x = adem01, y = prop)) +     # map category to x-axis, proportion to y-axis
```

```
  geom_col() +                             # draw bars with given heights (prop)
  scale_y_continuous(labels = percent) +   # scales package: show y-axis as percentages
  labs(
    x = "Gender",                          # x-axis label
    y = "Percent of respondents",          # y-axis label
    title = "Share of genders in the sample"
  )
```

2.3. Ordinal variable: Highest vocational degree of parents

Variable `apar10` measures the **highest vocational degree of the respondents' parents**. It is a categoricial variable with three categories.

1. Use `mutate()` and `apar10 = as_factor(adem01, levels="labels")` to transform `apar10`into a factor variable. Save the result again as `mydata2`.

2. Use the `frq()` function from the *sjmisc* package to calculate a table with *absolute, relative and cumulative frequencies*. Check also the *percentiles*.

Note: For ordinal variables, percentiles are categories, not exact numeric values in between categories. As you can see, `frq()` also calculates a mean and standard deviation. However, these values do not really make sense for categorical variables.

3. Create a bar chart showing the *distribution* of `apar10`, without missings and percentages plotted on the y-axis.

2.4. Metric variable: Overall life-satisfaction

Our outcome variable `bpsy01` measures the **overall life satisfaction** rated on a scale from 0 to 10. Technically, it is **an ordinal variable**, but in research practices, scales like this are often **treated as metric**. And this is what we are going to do.

1. Look at the distribution of `bpsy01`. Calculate the *absolute, relative and cumulative frequencies* and look at the *mean and standard distribution* using the `frq()` function.

2. Examine `bpsy01` using the `summary` and the `skim()` function from the *skimr* package. What can you see here? How do they differ?

3. To visualize the distribution of `bpsy01`, draw a boxplot with `ggplot`.

The code goes like this:

```
mydata2 %>%
  filter(!is.na(bpsy01)) %>%                # remove missings
  ggplot(aes(x = "", y = bpsy01)) +         # bpsy01 on y-axis
  geom_boxplot() +                          # draw boxplot
  labs(
    x = "",
    y = "bpsy01 (0-10 scale)",
    title = "Distribution of bpsy01 (boxplot)"
  )
```

What does the boxplot show you?

4. To see the overall shape of the distribution of `bpsy01`, create a *histogram* using `ggplot()`. Use `bpsy01` on the x-axis and choose a bin width of 1 to show the 0–10 scale clearly.

The code goes like this:

```r
# histogram for bpsy01
mydata2 %>%
  filter(!is.na(bpsy01)) %>%                # remove missings
  ggplot(aes(x = bpsy01)) +                 # bpsy01 on x-axis
  geom_histogram(binwidth = 1) +            # bins of width 1 (0,1,2,...,10)
  labs(
    x = "bpsy01 (0-10 scale)",
    y = "Number of respondents",
    title = "Distribution of bpsy01 (histogram)"
  )
```

What do you see regarding the skewness of `bpsy01`?

5. Let's look at inference statistics for `bpsy01`. Pick the functions from the table to calculate *standard errors* and both *95 % confidence intervals.*

Overview: Descriptive statistics, e.g. for variable `bpsy01`

What we compute	R command (inside `summarise()`)	Explanation
Number of valid cases (n)	`n = sum(!is.na(bpsy01))`	Counts all non-missing values of `bpsy01`.
Mean	`mean_val = mean(bpsy01, na.rm = TRUE)`	Average value of `bpsy01`.
Standard deviation (SD)	`sd_val = sd(bpsy01, na.rm = TRUE)`	How much values vary around the mean.
Variance	`var_val = var(bpsy01, na.rm = TRUE)`	SD squared.
Standard error of the mean (SE)	`se_val = sd_val / sqrt(n)`	Uncertainty of the sample mean.
95% CI – lower bound	`ci_lower = mean_val - 1.96 * se_val`	Lower limit of the 95% confidence interval.
95% CI – upper bound	`ci_upper = mean_val + 1.96 * se_val`	Upper limit of the 95% confidence interval.

Take Home checklist: Univariate analysis in R

Step	Question / task	Useful functions / tools
1	Do your categorical survey variables have readable categories (factors)?	Convert labelled variables with `as_factor()`, then use `tabyl()` or `ggplot()` on the factors.
2	Are missing values coded correctly?	Recode special codes to `NA` (e.g. `mutate(across(..., ~ ifelse(.x < 0, NA, .x)))`)
3	For nominal variables: how often does each category occur?	`tabyl()`, `count()`, bar charts with `ggplot(aes(x = var)) + geom_bar() / geom_col()`
5	For ordinal variables: how are responses distributed along the scale?	`frq()`, cumulative % and percentiles from the table
6	For metric variables: what are central tendency and spread and shape?	`mean()`, `sd()`, `var()`, `quantile()` `skimr()`, SE and CI via `summarise()`, box plot with `geom_boxplot()`, histogram with `geom_histogram()`
7	Are there obvious problems in the distributions?	Check tables, boxplots and histograms for impossible values, extreme outliers, very strong skewness etc.