Exercise 4

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`: Monthly 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

Multivariate Analysis: Regression Models

So far, we have looked at our data **one variable at a time** (univariate) and in **pairs** (bivariate).
In this sheet, we take the next step to **multivariate analysis**: we use **regression models** to study how several predictors together are related to an outcome.

In the context of our research story, our main outcomes of interest are still:

- overall life satisfaction (`bpsy01`), and
- having children (yes/no, `blcd06`).

Now we want to answer questions like:

- How is life satisfaction related to **several factors at once** (e.g. work–life compatibility, emotional support, burden to perish, self efficacy, income)?
- Does a predictor still matter for life satisfaction **after controlling for** other variables?
- How do different predictors relate to the **probability of having children**?

In this exercise, you will learn to:

- fit and interpret a **multiple linear regression** with `bpsy01` as the dependent variable,
- build and compare **nested linear models** (adding blocks of predictors and comparing model fit),
- fit and interpret a **logistic regression** with `blcd06` (children yes/no) as the dependent variable,
- report and interpret **regression coefficients**, **confidence intervals**, and (for logistic regression) **odds ratios**,
- summarize the main results in simple sentences (e.g. "Holding other variables constant, higher burden is associated with lower life satisfaction").

2.1. Set-up: Load packages and dataset

1. Install `tidyverse`, `haven`, `tidymodels` if not already installed and load these packages.

2. For this exercise, we will use the file *`04_qa_multi_data.sav`* in the *exercises* folder. Define the *exercise* folder as your working directory with the function `setwd("path")`.

3. Choose `mydata4` as a name for your data frame and import it with the following structure: `chosen_name <- read_sav("filename")`.

2.2. Converting categorical to factor variables

In preparation of the multiple linear regression, make sure the categorical variables are factors.

Use `mutate()` and `as_factor()` to transform all categorical variables (`adbi15, adem01, adem03, apar10, bdbi01, blcd01, blcd06`) into a factor variable. Save the result again as `mydata4`.

The code should look like this:

```
mydata4 <- mydata4 %>%
  mutate(
    across(
      c(adbi15, adem01, adem03, apar10, bdbi01, blcd01, blcd06),   # all cat. variables
      ~ haven::as_factor(.x)       # labels -> factor levels
    )
```

```
  )

glimpse(mydata4)
```

2.3. Multiple linear regression: life satisfaction <- PhD % work condition

We want to model **overall life satisfaction** (`bpsy01`) as a function of PhD characteristics and work conditions:

- `bdbi01`: PhD status (factor)
- `adbi15`: doctoral discipline (factor)
- `bdcd17`: – emotional support during PhD (5-point Likert scale, treated as metric)
- `bdwr12`: – burden index (5-point Likert scale, treated as metric)
- `bemp81`: – monthly gross income in €

We use **multiple linear regression** with `lm()` and **tidy the output** with *tidy*.

1. Create a cleaned data frame `mydata_lm_ex1` that only contains rows with non-missing values for the dependent variable and all predictors in this model.

The code should look like this:

```
mydata_lm_ex1 <- mydata4 %>%
  filter(
    !is.na(bpsy01),   # dependent variable: life satisfaction
    !is.na(bdbi01),
    !is.na(adbi15),
    !is.na(bdcd17),
    !is.na(bdwr12),
    !is.na(bemp81)
  )
```

The new data frame `mydata_lm_ex1` should now appear in your environment pane.

2. Fit a multiple linear regression with `bpsy01` as the dependent variable and `bdbi01`, `adbi15`, `bdcd17`, `bdwr12`, and `bemp81` as predictors.

The code should look like this:

```
lm_phd <- lm(  # save results in data frame lm_phd
  bpsy01 ~ bdbi01 + adbi15 + bdcd17 + bdwr12 + bemp81, # bpsy01 as dependent var, others as
 ↪  predictors
  data = mydata_lm_ex1 # use data frame mydata_lm_phd
)
```

Tipp: By default, R uses the first factor level (often the lowest numeric code) as the reference category.You can change the reference directly inside the regression call using `relevel()` in the formula. For example like this:

```
lm_phd <- lm(
  bpsy01 ~
    bdbi01 +
    relevel(adbi15, ref = "mathematics, natural sciences") +  # change reference category here
    bdcd17 +
```

```
    bdwr12 +
    bemp81,
  data = mydata_lm_ex1
)
```

3. Create a tidy table of coefficients and check the results.

Do it like this:

```
# use tidy data frame lm_phd, save results in data frame lm_phd_results and display 95% confidence
↪  intervals
lm_phd_results <- tidy(lm_phd, conf.int = TRUE) %>%
  mutate(across(where(is.numeric), ~ round(.x, 3)) # optional: round output to three decimals
  )

lm_phd_results
```

You get the following information:

- *term*: variable name (intercept and predictors)
- *estimate*: regression coefficient (change in bpsy01 if the predictor increases by 1, holding - other variables constant; for factors: difference to the reference category, this is always the first)
- *std.error, statistic, p.value* = test statistics
- *conf.low, conf.high* = lower and upper bound of the 95% CI

How do you interpret the results?

4. Use the `glance()` function to get an overview of the model fit.

```
lm_phd_fit <- glance(lm_phd)
lm_phd_fit
```

Important values:

- *r.squared*: proportion of variance in life satisfaction explained by the model
- *adj.r.squared*: adjusted R^2 (corrected for number of predictors)
- *p.value* (for the overall F-test): tests if the model explains more variance than a model with no predictors*

How would you interpret the results?

2.4. Multiple linear regression: life satisfaction, nested models

In the next step, we build a **set of nested regression models**. This means that each new model contains all predictors from the previous model and then adds one block of additional variables. By comparing these models, we can see **how the explained variance changes** and whether certain predictors remain important **after controlling for** other factors.

Model 1: PhD and work conditions

In Model 1, we predict life satisfaction (`bpsy01`) from PhD-related characteristics and work conditions. The predictors are PhD status (`bdbi01`), doctoral discipline (`adbi15`), emotional support during the PhD (`bdcd17`), scientific pressure (`bdwr12`), and monthly income (`bemp81`). This model answers the question:

How are life satisfaction and PhD/work conditions related, without controlling for attitudes, well-being or demographics?

Model 2: + attitudes and well-being

Model 2 extends Model 1 by adding attitudes and well-being variables: satisfaction with work–life balance (`blcd12`), relationship with parents (`apar15b`), and self-efficacy (`bpsy05`). All predictors from Model 1 remain in the model. This model asks: *Do PhD/work conditions still matter for life satisfaction after we also account for attitudes and well-being?*

Model 3: Full model with demographics

Model 3 is the full model. It includes all predictors from Model 2 and adds demographic controls: gender (`adem01`), age in years (`adem02`), parents' highest vocational degree (`apar10`), country of birth (`adem03`), and relationship status (`blcd01`). This model answers: *Which predictors are associated with life satisfaction when we simultaneously control for PhD/work conditions, attitudes and well-being, and basic demographic characteristics?*

1. Create a cleaned data frame `mydata_lm_ex2` that that has no missing values on any variable used in the three models. This way all models are based on the same set of cases.

2. First, fit a simple model with *only PhD and work condition predictors* like in exercise 2.3.

3. Next, extend the model by adding *attitudes and well-being* variables. This model contains all predictors from Model 1 *plus* the new ones.

What changes compared to Model 1? What do the new predictors in Model 2 tell us?

4. Finally, build a *full model* that also includes *demographic controls*. Again, we keep all predictors from Model 2 and add the demographic variables.

What **changes compared** to Model 2? Is there an association of the demographic variables and overall life satisfaction?

5. Compute which model fits best.

Do it like this:

```
model_summaries <- bind_rows(
  glance(lm_ex2_m1) %>% mutate(model = "Model 1"),
  glance(lm_ex2_m2) %>% mutate(model = "Model 2"),
  glance(lm_ex2_m3) %>% mutate(model = "Model 3")
) %>%
  mutate(across(where(is.numeric), ~ round(.x, 3)))

model_summaries
```

2.5. Multiple logistic regression: probability of having children <- full model

We want to predict the probability of **having children (yes/no)** (`blcd06`) as a function of PhD and work conditions, attitudes and well-being and demographics.

We estimate the model with the `glm()` function and use `tidy()` to get odds ratios.

1. Create a cleaned data frame `mydata_lm_ex3` that only contains rows with non-missing values for the dependent variable and all predictors in this model.

2. Fit the full logistic regression model on having children (yes/no) `blcd06`.

To model the log-odds of `blcd06` for "yes" with reference category "no", we have to relevel our dependent variable with `relevel()`.

The code goes like this:

```
glm_full <- glm(
  relevel(blcd06, ref = "no") ~                     # change reference category
    bdbi01 + adbi15 + bdcd17 + bdwr12 + bemp81 + # predictors
    blcd12 + apar15b + bpsy05 + bpsy01 +
    adem01 + adem02 + apar10 + adem03 + blcd01,
  data    = mydata_glm_ex3,                         # take dataset without missings
  family = binomial(link = "logit")                 # logistic regression
)
```

3. Tidy the table and display odds ratios.

The code should look like this:

```
glm_ex3_results <- tidy(
  glm_full,
  exponentiate = TRUE, # exp(coef) = odds ratios instead of log-odds
  conf.int     = TRUE  # 95% confidence intervals
) %>%
  mutate(across(where(is.numeric), ~ round(.x, 3))) # round numeric values to three decimals for
  ↪  readability


glm_ex3_results

print(glm_ex3_results, n = Inf)  # show all rows
```

In the output table, estimate now shows **odds ratios** (because we used exponentiate = TRUE).

Odds ratio > 1: the predictor is associated with higher odds of having children (blcd06 = "yes").

Odds ratio < 1: the predictor is associated with lower odds of having children.

The 95% confidence interval shows the uncertainty around each odds ratio; if it includes 1, the effect is not statistically significant at the 5% level.

How do you interpret the results?

4. Finally, calculate the overall model fit including McFadden R^2.

The code should look like this:

```
glm_ex3_fit <- glance(glm_full)

glm_ex3_fit %>%
  mutate(
    mcfadden_r2 = 1 - deviance / null.deviance
  )
```

How do you interpret the model fit?

Take Home checklist: Multivariate regression in R

#	Question / task	Useful functions / tools
1	What is your **outcome variable** and which type of regression do you need?	Check scale & distribution with `str()`, `summary()`, `frq()`, `skimr::skim()`, histograms / bar charts via `ggplot()`
2	Are predictors coded correctly (metric vs. factor, reference categories, NAs)?	Recode with `mutate()`, `across()`, `case_when()`, set NAs; convert labelled vars with `haven::as_factor()`, set factor order / reference with `forcats::fct_relevel()` or `relevel()`
3	Fit a **baseline model (Model 1)** with a small set of predictors	Linear: `lm(outcome ~ x1 + x2, data = ...)`; Logistic: `glm(I(outcome == "yes") ~ ..., family = binomial)`; get coefficients with `broom::tidy(model, conf.int = TRUE)`; model fit with `broom::glance(model)`
4	Build **extended / nested models** (Model 2, Model 3 …) by adding predictor blocks	Specify new formulas in `lm()` / `glm()`; summarize and compare fit with `bind_rows(glance(m1), glance(m2), glance(m3)) %>% mutate(across(where(is.numeric), round, 3))`
5	How do you **interpret coefficients**?	Linear: change in outcome per 1-unit change in predictor, "holding other variables constant"; Logistic: use `tidy(glm_model, exponentiate = TRUE, conf.int = TRUE)` for **odds ratios** and 95% CIs, check if CI includes 1
6	How well does the **model fit** the data?	For linear: use `glance()` → `r.squared`, `adj.r.squared`; for logistic: `glance()` → McFadden pseudo-R^2: `mutate(mcfadden_r2 = 1 - deviance / null.deviance)`
7	Can you **communicate the results** in simple sentences?	From `tidy()` and `glance()`: report direction and size of effects, p-values and confidence intervals, plus overall fit (R^2 / pseudo-R^2); use phrases like "holding other variables constant…" and focus on **substantive** importance, not only significance