# Introduction

## Advanced Statistical Methods and Models in Experimental Design

Bartosz Maćkiewicz

2025-02-20

# Goals

1. Learn how to *read* experimental papers
2. Learn how to *choose* the right statistical methods
3. Learn how to *perform* statistical analyses in R
4. Learn how to *describe* their results in a professional manner

# Inferential statistics

In this course, we will focus on **inferential statistics**, with emphasis on **null hypothesis significance testing** (NHST).

## Inferential statistics

- drawing conclusions about the population
- estimating parameters of the population
- testing hypotheses about population(s)
- examples: statistical tests, confidence intervals, etc.

## Descriptive statistics

- describing a sample
- summarizing data
- plotting data
- examples: sample mean, sample standard deviation, boxplot, etc.

# Statistics and other disciplines

There are many similarities and overlaps between statistics and other disciplines such as Machine Learning and Data Science.

That's OK.

# Statistics and other disciplines

Let's consider linear models (gross simplification!):

## Machine Learning

- overall performance of the model
- accuracy of predictions
- features selection
- models comparison

## Statistics

- hypothesis testing
- explanation (% of variance accounted for, effect sizes)
- statistical significance of individual predictors

# Two types of students

1. “Statistical Sharks”
    - They have a solid understanding of the logic of hypothesis testing.
    - They are familiar with common statistical tests and models.
    - They have some experience with analyzing data using R or other similar software.
    - They have some programming skills (preferably with R).
2. “Squirrels of Statistics”
    - They have little to no previous exposure to inferential statistics or want to consolidate knowledge.
    - They have never analyzed “real” experimental data, only toy examples.
    - They do not feel very confident in their programming skills

Some of you are Sharks, and some of you are Squirrels.

Hard problem: how to accommodate both Sharks and Squirrels?

# Accommodating Sharks and Squirrels

1. “Statistical Sharks”
   - In the majority of assignments there will be more creative and slightly harder exercises that expand upon the knowledge of basics.
   - Sharks will be able to complete completely alternative assignments called “Shark Track” which will be much more complex and adequately test their skills.
2. “Squirrels of Statistics”
   - I will provide them with additional resources on learning R as a programming language (unfortunately some of them only in Polish).
   - Deadlines can be made more flexible for students who need more time to catch up with material.

# Overview

1. Review of classical statistical tests (2-3 weeks)
    - binomial test (as a “toy example”)
    - chi-square test of independence and chi-square goodness-of-fit test
    - one sample t-test and two sample t-test
    - test for correlation between paired samples (Pearson’s *r*, etc.)
    - power analysis and effect sizes
2. Generalized Linear Model (4-5 weeks)
    - linear regression
    - nominal predictors and contrasts
    - analysis of variance
    - GLM assumptions and data transformations
    - link functions
    - logistic regression
3. Complex experimental designs (3-4 weeks)
    - repeated measures and mixed designs
    - hierarchical models
4. Factor analysis (1-2 weeks)
5. Meta-analysis (1 week)

# Assignments and tests

## Weekly or once every two weeks

1. Reading (not obligatory but strongly recommended)
2. Assignment (0-5 points)
   - loading data & light “data wrangling”
   - testing some research hypothesis
   - writing up the results
   - objective: acquire practical skills working with “real” data

## Final project

- Task: write a report with a more advanced analysis of some real publicly available experimental data.

## Final test

- Only if necessary (probably not): a short quiz

# RMarkdown

A nice way to communicate complex ideas and analyses using combination of code, text and graphics.

This is an example of RMarkdown syntax:

```
# This is a first level header (or a new slide)
## This is a second level header

You can *also* format your **text** and make
* pretty
* lists
1. of different
2. types

'''{r}
 # wrong character here, on purpose! use backtick (`)
 hist(rnorm(100))
'''
```

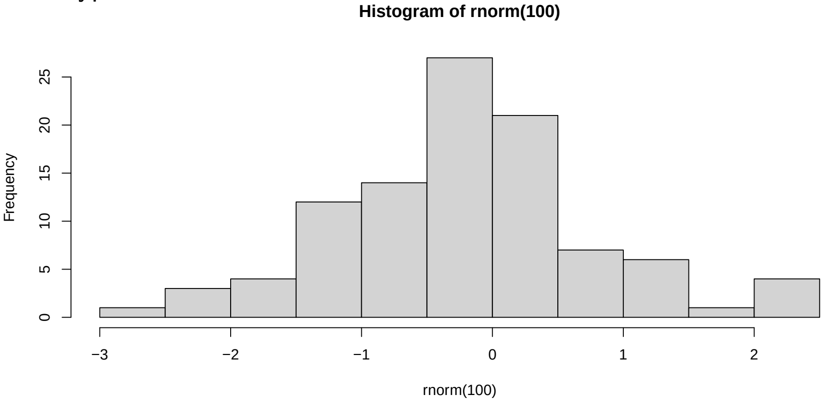
# This is a first level header (or a new slide)

## This is a second level header

You can *also* format your **text** and make

- pretty
- list

1. of different
2. Types



# Webpage

https://adv-stat.kursy.bartoszmackiewicz.pl

Yeah, the address is bad. Get over it. You can bookmark it.

The page is currently **under construction** but I will put there:

1. Additional resources (books, articles, tutorials, etc.)
2. Assignments
3. Datasets that we'll be working with during classes

You will need to **create an account**

- You may use a nickname as long as you somehow identify yourself at the end of the course.
- Do not use your "real" password – come up with something new that will be used only on our platform.
- You may use a fake e-mail address if you wish, I don't care.

# Policy on Academic Integrity and Use of AI Tools

## Prohibition of AI Assistance

- **ChatGPT trivializes a significant portion of homework assignments**, making it tempting to bypass the learning process.
- **Using ChatGPT for homework is strictly forbidden** as it:
  - Undermines the foundational learning of beginners who need to master the basics.
  - Prevents the evaluation of students' actual understanding and skills, which is crucial at the university level.
  - Makes the review of homework completed by ChatGPT dehumanizing for the instructor.

## Consequences of Violation

- **Immediate failure in the course** for using ChatGPT or similar AI tools.
- **Notification to academic institutions** about the breach of academic integrity.

# Policy on Academic Integrity and Use of AI Tools

## Rationale Behind the Policy

- Large Language Models (LLMs) like ChatGPT may excel in tasks such as writing JavaScript widgets or centering DIVs, but they often fail to produce correct/good code for more complex and non-standard tasks.
- **Critical skills development is hindered** by relying on AI as a co-pilot, including:
  - Reading documentation.
  - Debugging.
  - Interpreting complex analysis results.
- **Using LLMs prevents the acquisition of these essential skills**, especially for beginners who are just learning to program.

# Policy on Academic Integrity and Use of AI Tools

## Our Agreement

- **Avoid using ChatGPT as a co-pilot** in your learning journey.
- **In return, I will not resort to unpleasant quizzes or exams** that are ineffective for programming courses.
- I acknowledge that LLMs can be useful in many aspects of life, including programming, **provided that one understands the basics and has practical skills**.

# Discord server

From my experience Discord server greatly improves communication so I will create one for us!

- Not obligatory, important information will be still emailed via USOSMail.
- Less formal way to ask a quick question or two or discuss something.
- It aims to reduce ‘email anxiety’
- I try to put there at least one funny picture (a meme if you wish) a week that is related to R or statistics.

https://discord.gg/rSJFUGYypp

# R

- We will be using R (programming language + programming environment) combined with RStudio (Integrated Development Environment, IDE)
    - but if you are more comfortable with some other tools (Jupyter Notebook with IRkernel, VS Code, etc.) feel free to use them!
- We will be trying to use cool libraries from `tidyverse` meta-package (`dplyr`, `tidyr`, `ggplot2`, etc.) so you need to install them on your PC (`install.packages("tidyverse")`)

# Expectations

1. You will have a solid understanding of basic concepts related to working with files such as:
   - the working directory (how to set it? how to change it?)
   - file types, the hierarchy of directories (what is in my file? where my file is stored?)
2. You will install R and RStudio on your own PC (or PC that you can conveniently use):
   - order: first install R, then RStudio
   - RTools may be needed to build some packages (and some compilers and build tools if you are using Linux)
   - Windows & MacOS: download them from the official repositories
   - Linux: install R using your package manager (`apt`, `yum`, `pacman`), install RStudio from the official repository (I can help!)
   - R: https://cloud.r-project.org/index.html
   - RStudio: https://www.rstudio.com/products/rstudio/download/#download

# Logic of NHST

1. Begin with a **research hypothesis**.
2. Set up the **null hypothesis**.
3. Construct the **sampling distribution** of the particular statistic on the assumption that $H_0$ is true.
4. **Collect** some data.
5. **Compare** the sample statistic to that distribution.
6. Decide to reject or retain $H_0$ based on the probability, under $H_0$, of observing a sample statistic as extreme as the one obtained."

# Example

Here we will present an example based on James Bond who insisted that martinis should be shaken rather than stirred.

Let's consider a hypothetical experiment to determine whether Mr. Bond can tell the difference between a shaken and a stirred martini. Suppose we gave Mr. Bond a series of 16 taste tests.

In each test, we flipped a fair coin to determine whether to stir or shake the martini. Then we presented the martini to Mr. Bond and asked him to decide whether it was shaken or stirred. Let's say Mr. Bond was correct on 13 of the 16 taste tests.

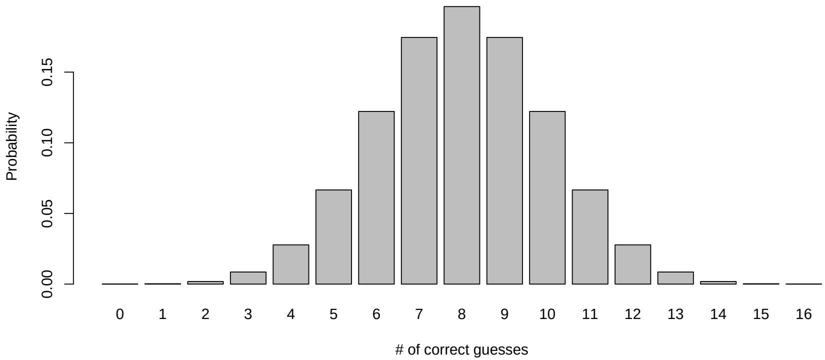
Does this prove that Mr. Bond has at least some ability to tell whether the martini was shaken or stirred?

# NHST in action

1. Begin with a **research hypothesis**.
   - Mr. Bond has some ability to tell whether the martini was shaken or stirred.
2. Set up the **null hypothesis**.
   - Mr. Bond is no better than chance ($H_0$: $Pr = 0.5$)
3. Construct the **sampling distribution** of the particular statistic on the assumption that $H_0$ is true.
   - binomial distribution: $B(16, 0.5)$
4. Collect some data.
   - on 13 taste tests Mr. Bond was correct
5. Compare the sample statistic to that distribution.
   - what is $Pr(13, 5, 0.5)$?
6. Reject or retain $H_0$, depending on the probability, under $H_0$, of a sample statistic as extreme as the one we have obtained.
   - R to the rescue!

# R in action

```r
barplot(dbinom(0:16, 16, 0.5),
        names.arg = 0:16,
        ylab = "Probability",
        xlab=  "# of correct guesses"
        )
```



# R in action

```
# using density function
sum(dbinom(13:16, 16, 0.5))
```

```
## [1] 0.01063538
```

```
# using cumulative distribution function
pbinom(13-1, 16, 0.5, lower.tail = F)
```

```
## [1] 0.01063538
```

# Type I and Type II errors

## Type I error

The first kind of error is the mistaken rejection of a null hypothesis as the result of a test procedure. This kind of error is called a type I error (false positive) and is sometimes called an error of the first kind.

In the context of the Mr. Bond example, a Type I error would occur if we incorrectly conclude that he can distinguish between stirred and shaken martinis when, in fact, he cannot.

## Type II error

The second kind of error is the mistaken acceptance of the null hypothesis as the result of a test procedure. This sort of error is called a type II error (false negative) and is also referred to as an error of the second kind.

In the context of the Mr. Bond example, a Type II error would occur if we fail to recognize his ability to distinguish between stirred and shaken martinis when he actually possesses this ability

# Loading data into the R environment

## Important functions:

- `setwd(path)` - sets the working directory to `path`
  - `path` may be *relative* (`data`, `../bartosz/data`) or *absolute* (`/home/bartosz/data/`, `C:/Users/bartosz/data`, `/Users/bartosz/data`)
- `getwd()` - gets the current working directory
- `read.{csv|csv2|delim|table|...}` - creates data frame from the file

# Exercise: load some data!

1. Download datasets from the webpage.
2. Load it into the R environment using the appropriate function.

# Exercise: first look at the worksheets!

1. Download the worksheet from the webpage.
2. Fill in the blanks with your solutions.
3. Upload the worksheet.