Chapter 6: The bigger picture: contextualizing statistical methods in psychology

1. Assume that a certain hormone is known to be normally distributed in the *healthy population*, with:

    - *Population mean* = 50
    - *Population standard deviation* = 10

    A sample of *10 patients* with a certain (unspecified) condition had their hormone levels measured. Their values are:

    ```
    patient_values <- c(70, 76, 65, 52, 59, 71, 73, 86, 58, 81)
    ```

    - Using the population parameters for healthy individuals, compute a 95 per cent prediction interval for individual hormone levels.
      Hint: Assume a normal distribution and use the standard normal quantile (i.e. qnorm(0.975) = 1.96).
    - How many of the 10 patient observations fall outside this prediction interval? What might this suggest?
    - Compute the sample mean and standard deviation of the patient group. Then compute a 95 per cent confidence interval for the group mean.
      Hint: Use the formula $\mathrm{CI} = \mathrm{mean} \pm 1.96 \times \frac{s}{\sqrt{n}}$ *where s is the sample standard deviation, and n is the number of observations.*
    - Is the patient group's mean hormone level compatible with the healthy population mean of 50? What does the confidence interval tell you?
2. In the exercises of Chapter 4, you examined a dataset from Sternberg et al. (1982) measuring dopamine beta-hydroxylase (DBH) activity in two groups: individuals with schizophrenia showing psychotic features, and those without. You previously used a linear model to compare the two groups. Load the same data again (using `read.csv("dbh_data.csv")`) and compute and interpret the standardized effect of the difference in DBH activity between the two groups. Specifically:

    - Compute Cohen's d and its standard error for the difference in DBH activity between the two groups.
    - Using the standard error, compute a 95 per cent confidence interval for Cohen's d.
      Hint: Use the standard normal critical value: 1.96 × SE.
    - How large is the standardized effect? Would you describe it as *small*, *medium*, or *large*?
    - How does the effect size compare to the group difference estimated in the original linear model? What extra information does it provide?
3. In this exercise, you will perform a power analysis to determine how many participants would be needed to replicate the DBH group difference with at least 80 per cent power. use the standardized effect size (Cohen's d) that you

obtained from answering Question 2 above.
Your task is to determine the minimum sample size required to reliably replicate this result, using both an analytic and simulation-based approach:

- *Using analytic power analysis:* use the `pwr` package to compute the sample size needed to detect an effect size of $d = 0.8$ with power = 0.8 and $\alpha = 0.05$, for a two-sided t-test.
  Hint: Use the `pwr.t.test()` function.
- *Using simulation:* now approach the same problem via a simple simulation.
  - Assume that DBH activity in both groups is normally distributed with the same standard deviation, and a mean difference consistent with $d = 0.8$
  - Simulate multiple datasets for different sample sizes (e.g. 20, 25, ..., 100 participants total), and compute the proportion of simulations that return a statistically significant difference between groups using either a t-test or a linear regression with categorical predictor.
  - Store and plot the estimated power for each sample size.
  - Use this plot to estimate the smallest sample size at which power exceeds 80 per cent.
    Hint: You can simulate two samples with `rnorm(n, mean, sd)` and use `t.test()` to check for significance. A loop can help you repeat this many times per sample size.
- *Compare results:* Compare the minimum sample size estimated by the simulation to that from the `pwr` package. Are they consistent? If they differ slightly, why might that be?

4. For this exercise we will revisit the `bechdel` dataset introduced in Chapter 5. This dataset contains information about thousands of movies, including their release year, budget, and domestic gross revenue (adjusted for inflation), as well as whether they passed the *Bechdel test* — a simple metric of female representation in film.
   We will use this data to practice a powerful visual storytelling strategy: building a plot gradually, in steps, to enhance clarity and keep an audience engaged. This approach is especially useful when presenting results with large or overlapping datasets.

   Start by loading and subsetting the data as we did in Chapter 5:

```
d <- fivethirtyeight::bechdel
d <- d[c("title", "year", "binary", "budget_2013", "domgross_2013")]
```

   Next, log-transform the budget and gross values. This helps make patterns in the data easier to interpret visually. Finally, make a series of three plots using base R plotting functions or ggplot2. These plots should be suitable for building up a narrative across presentation slides:

- Plot 1: An empty plot with appropriate axis limits – no data yet, just axes.
- Plot 2: Add points representing movies that pass the Bechdel test.
- Plot 3: Add points for all movies, using different colours for those that pass vs. those that don't.

Use the `xlim` and `ylim` arguments to ensure that all three plots share the same axis limits — this makes the transition between slides clearer and easier to follow.

5. For this exercise, we will work with the `air_passenger_numbers.csv` dataset. These data give monthly totals of international airline passengers from January 1949 to December 1960. The counts are in thousands. The goal is to create a plot that reveals seasonal trends and how they evolve across years.

   Produce a plot where the x-axis shows the month of the year (from 1 to 12, or January to December), the y-axis shows the number of air passengers (in thousands), and a separate line for each year, drawn in a different colour.

   - Load the data from `air_passenger_numbers.csv`. The file should be structured with months as rows and years as columns.
   - Set up an empty plot using the `plot()` function – without drawing any data yet – but specifying the correct axis limits using `xlim` and `ylim` (e.g. using `xlim = c(1, 12)` and setting `ylim` based on the minimum and maximum of all passenger counts across all months and years). This ensures that the axis limits stay fixed regardless of the number of lines added.
   - Generate a set of colours – one for each year – using a colourblind-friendly palette such as those in the `viridis` library.
   - Use a `for` loop to add one line at a time to the plot using the `lines()` function.