Extra exercises for feedback

Pernille Brams

3/4/2024

Contents

Exercises for individual feedback

What is this?

This markdown has exercises covering some stuff from Methods 2 and a bit from Methods 1. The exercises herein are not part of any portfolio, but are just an opportunity for you to practice your stats/code and see how far you have come during Methods1/2.

You can complete some of / all the exercises depending on what you have time for, submit them to me (instructions for how to do this below), and then I'll give you all some brief feedback. If you do not want feedback, you are naturally also very welcome to just do some of / all of the exercises for yourself - it will still be a good marker to see how far you have come.

The purpose of the markdown is for you to hopefully feel motivated to complete some code-work by getting the chance to have your code reviewed.

What does the markdown include?

The markdown includes smaller exercises, some of which you have seen before if you've done class exercises, and some which are related to the new stuff we're learning in weeks 12-16 + some stuff from Methods 1 to recap as per your midterm responses indicating that wish.

The exercises are also of varying size. You do not have to complete them chronologically unless it's specified. There are also a few that are not code based.

Solutions will be uploaded after deadline for submission.

Can I complete the exercises in groups (any groups) or should I do it alone?

Completely up to you.

But consider if you want feedback on your individual progress rather than your progress as a group/pair/trio, as I will give feedback to you as a group if you submit as one.

How do I submit?

After completing however many of the exercises you want to, if you want feedback from me, please do the following two steps:

1) Submit an 1) .Rmd and 2) a knitted file with your answers in the Git Classroom link where you found the exercise markdown. If you have trouble with knitting, don't worry and just submit the .Rmd by itself.

2) When uploaded to git, send me a link to your git repo with your solutions via email + a note on if you completed in a group with group member names. If you did it in groups, one email with a link suffices and put your group members cc on the email so I can reply with feedback to all of you. If you collaborated on the exercises in your group, note that you did.

If you have any questions about anything, go in the anonymous Q/A, Github discussion forum, my email (au650502@uni.au.dk) or find me on messenger. If you message me on linkedin reg this i will grin

When do I submit?

Friday 19th is the deadline for submitting and getting feedback on all responses, but I will give feedback as submissions come in - so if you hand your submission in before this date, you'll likely also get feedback sooner. If you need an extension to the deadline, just shoot me an email :-)

Does completing and submitting these exercises to Pernille count towards my grade?

NOPE. :)

To be clear: completing this markdown in no way counts towards your final exam / portfolio grades for Methods 2 or any course. It is purely an opportunity for you to get some individual feedback from me on "grunt work" code (in contrast to the longer, more conceptual portfolios, which are the usual ones you get feedback on). We rarely focus on the grunt code review and that's actually pretty important :-)

Exercises

Section A

A1: Normal distribution: Explain what a normal distribution is and list at least two places where we have used it in Methods 1 or 2. [Write at least 3-4 lines]

A2: Experimental design: Describe the difference between within-subjects and between-subjects design. Why is it important to distinguish between these two types of designs in psychological research? [Write at least 3-4 lines]

A3: Simulating: In your own words, describe what it means when we sample from a distribution - and find and describe a case where we can use it. [Write at least 3-4 lines]

Section B

B1 [CHRONOLOGICAL]

B1.1: Choosing the right test: You are conducting an experiment where you measure the reaction times of 20 subjects before and after they drink a cup of coffee. Reaction time is being recorded based on the moment they reach out for a pen that is being dropped in front of them by the experimenter. You have to analyze it using a t-test. Describe the experimental design (is it within-subject or between-subject, and why?) and explain which type of t-test would be appropriate to analyze the data and why.

B1.2: Simulating data for the experiment: To see what these data might look like and avoid having to go out and collect actual data, we need to simulate some reaction times. Simulate two sets of data from the 20 subjects, one set acting as the before and the other set being the reaction times recorded after having coffee, assuming a slight decrease in reaction time for the latter set (meaning they are expected act quicker after having had coffee). Use rnorm() to simulate for both sets, and for the mean and standard deviation you need to set there, set some numbers you think make sense (e.g. time yourself a couple times on the task and set the mean of your milliseconds as the mean, or look to literature and see if a study or something has been done). Put references if you use lit.

B1.3: Run the right type of t-test you explained was appropriate on your data to see if there's an effect.

B1.4: Reporting: Report on the test formally.

Section C

C1: For loop: Write a for loop in R that calculates and prints the square of numbers from 1 to 25

C2: For loop: Write a for loop in R that calculates and prints the square of only odd numbers from 1 to 25

C3: Sampling and Plotting: Sample 10 points from a normal distribution with mean = 5 and sd = 2, and use ggplot2 to create a histogram of these samples.

C4: Data Plotting: Find a simple dataset (e.g. mtcars) with two columns you can plot against each other. Read in the data as a dataframe and call it df. Given the data frame df with columns X and Y (whatever they are), use ggplot2 to plot X on the x-axis and Y on the y-axis.

C5: If-else: Write an if-else statement in R that prints "positive" if a randomly sampled number from a normal distribution with mean 0 and standard deviation 1 is greater than 0, and "negative" otherwise.

C6: While loop: Write a while loop in R that continuously samples a number from a standard normal distribution and stops once it samples a number greater than 2.

Section D

D1. Calculate the determinant of this 2x2 matrix (either in R or in hand - if done in hand, attach a picture of your calculation as a pdf / jpg in your repo)

$$D = \begin{pmatrix} 0 & 8 \\ 1 & -1 \end{pmatrix}$$

D2. Calculate the determinant of this 3x3 matrix (either in R or in hand - if done in hand, attach a picture of your calculation as a pdf / jpg in your repo)

$$H = \begin{pmatrix} 2 & 5 & 7 \\ 6 & 3 & 4 \\ 5 & -2 & -3 \end{pmatrix}$$

D3. What does it mean when a vector is orthogonal to something? [1-2 lines minimum]

D4. Find a vector orthogonal to each of the following: (either in R or in hand - if done in hand, attach a picture of your calculation as a pdf / jpg in your repo)

$$\mathbf{v}_7 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \mathbf{v}_8 = \begin{pmatrix} 4 \\ -5 \\ 6 \end{pmatrix}$$

D5. Invert this 2x2 matrix using R:

$$P = \begin{pmatrix} 0 & 8 \\ 1 & -1 \end{pmatrix}$$

D6. Inversion for larger matrices (Using Gauss-Jordan) (either in R or in hand - if done in hand, attach a picture of your calculation as a pdf / jpg in your repo)

Perform matrix inversion using either the Gauss-Jordan elimination method (if not sure, try youtubing!) in hand or solve() in R on:

$$T = \begin{pmatrix} 0 & -1 & 2 & 1 \\ 1 & 0 & -1 & 2 \\ 2 & 1 & 0 & -1 \\ -1 & 2 & 1 & 0 \end{pmatrix}$$

D7: Determining the Dimensions of the Product Matrix

Consider two matrices,

$$A$$

and

$$B$$

, defined as follows:

$$A = \begin{pmatrix} 0 & -1 & 2 & 1 \\ 1 & 0 & -1 & 2 \\ 2 & 1 & 0 & -1 \\ -1 & 2 & 1 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

Given these matrices, if we were to multiply them, what will the dimensions of the resulting product matrix be?

D8: Calculate the product of matrix A and B from D7 (either in R or in hand - if done in hand, attach a picture of your calculation as a pdf / jpg in your repo)

D9: Checking Compatibility for Matrix Multiplication

Given matrices

$$X$$

and

$$Y$$

below, use R to check if these matrices are compatible for multiplication. Explain your reasoning.

$$X = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 & -1 & 2 & 1 \\ 1 & 0 & -1 & 2 \end{pmatrix}$$

Section E

E1: Creating a ggplot with Aesthetics

Run the following R code to simulate data for this exercise. Your task is to create a ggplot that visualizes this data with the following specifications:

- Plot the individual data points as dots.
- Use a nice color scheme with a color palette of your choice.
- Add a line representing the mean value of y across GROUPS of x
- Add titles/subtitles to explain what we're plotting + what the lines are

```
# Code for simulating
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
set.seed(42)
df <- data.frame(
  x = rep(1:10, each = 5),
  y = rnorm(50, rep(1:10, each = 5), 2)
) %>%
  mutate(group = ifelse(x <= 5, "A", "B"))

# Calculate mean y value by group
means_group <- df %>%
  group_by(group) %>%
  summarise(mean_y = mean(y))

# your plot code here
library(ggplot2)
```

E2: Checking for Normality

Use the below R code to simulate a dataset. Your task is to perform a normality test on the dataset and interpret the results. Discuss whether the data appears to follow a normal distribution based on the test outcome.

```r
# Simulate data
set.seed(123)
data <- rnorm(100, mean = 50, sd = 10)

# your normality testing code here
```

E3: Running and Interpreting an lmer Model

Run the following R code to simulate some data for a repeated measures experiment. Your task is to fit a linear mixed-effects model (lmer()) predicting y from x while accounting for repeated measures within subjects (subj). Interpret the model output, focusing on the fixed effects and their significance.

```r
# Load the required package
if(!require(lme4)){install.packages("lme4")}
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(lme4)

# Simulate data
set.seed(456)
df <- expand.grid(subj = factor(1:20), x = factor(1:5))
df$y <- rnorm(nrow(df), mean = as.numeric(df$x) * 2, sd = 1)

# your lmer code here
```

E4: A study example

Imagine you are conducting a longitudinal study on the effect of a new educational technique on student performance. You collect data on student test scores at multiple points throughout the academic year from several different schools. Due to the nature of your data—multiple observations from the same students and inherent differences between schools—you decide to analyze the data using a linear mixed-effects model (lmer()) and include random intercepts for students and schools.

Question: What assumption of linear mixed-effects models is addressed by incorporating random intercepts for both students and schools? Explain how using random intercepts in the model helps to mitigate the violation of this assumption.

Section F:

F1: What is the ordinary least squares method and what do we use it for? [~2 lines minimum]

F2: Optional extra task: Find a dataset (e.g. mtcars) and fit a model using stan_glm with a prior

```r
# First load rstanarm
if (!requireNamespace("rstanarm", quietly = TRUE)) {
  install.packages("rstanarm")
```

```
}
library(rstanarm)
```

```
## Loading required package: Rcpp
## This is rstanarm version 2.32.1
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
## - For execution on a local, multicore CPU with excess RAM we recommend calling
##   options(mc.cores = parallel::detectCores())
```

```
# your bayesian code here
```