BSc CogSci - 2. Sem - Methods 2 - Review Exercises

aiswary-a

2024-04-19

Contents

Exercises

```r
# loading libraries

library(pacman)
pacman::p_load(tidyverse,
               matlib, # for work with matrices
               lme4) # for work with linear models
```

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.

- **What:** The normal distribution (also known as the Gaussian distribution) is a probability distribution (depicting continuous data) that is symmetric around the mean and thus also unimodal, which results in a bell shape. The x-axis on which it rests, represents the 'spread' of the displayed data, and, often, a greater SD indicates a greater spread of data and thus a flatter histogram.
- **Where:** Checking / assessing the normality of data sets the foundation for many statistical tests (e.g. parametric t-test, ANOVA). Furthermore, as the maths involved in deriving the test statistics is simplified by normality, *replicability* and ease of interpretation are enhanced. This distribution also serves as a fundamental assumption in linear regression models, where residuals are assumed to follow a normal distribution.

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?

- **Within-Subject Design:** All participants go through every condition.
- **Between-Subject Design:** Different groups of participants are assigned to different experimental conditions.
- **Importance:** In differentiating between the two experimental designs, it is vital to understand that each type of design has its own advantages and disadvantages.
  - A *within-subject* design may be beneficial to consider if you want to control for individual differences between the participants. Furthermore, this approach may be better if you have a smaller sample / number of participants. However, the order of going through the conditions will likely have an effect on the observations. Additionally, participants will be aware of the experimental manipulations which may bias their behaviour.
  - On the other hand, a *between-subject* design may be advantageous in that participants are not necessarily made aware of the experimental manipulation as they all only go through a single condition. Simultaneously, however, it may be difficult to achieve fully equivalent participant groups with respect to all variables except the manipulation. This approach is likely to work better if your sample / number of participants is greater.

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.

- **What:** To sample from a distribution means to select a particular number of points generated by some theoretical distribution. When you sample from theoretical distribution, the points that are selected conform to the parameters of that particular distribution.
- **Example:** In the case of a normal distribution, which often mirrors human behaviour and various natural phenomena, we can generate random samples from it using specified parameters (mean and standard deviation) with functions like **rnorm() in R**. We can thus simulate datasets that resemble real-world observations, and test statistical hypotheses as well as make predictions based on the data.

Section B

B1

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.

- **Experimental Design:** The experimental design being implemented in this example is a **within-subject** design, as all 20 participants go through the same condition; for *all* we measure the reaction time before they drink the cup of coffee AND after.
- **t-Test Type:** The type of t-test that should be used to analyse the data is the paired t-test. This is clear because: **1)** there is a continuous measurement being taken (reaction time); **2)** the goal is to determine if the difference between paired measurements (reaction time before and after participants drink a cup of coffee) is 0 or not i.e. if the reaction time before vs. after is different or not.

```r
set.seed(711) # setting seed for replicability

# using results in literature: "Caffeine improved reaction time (from 0.42 ± 0.05 to 0.37 ± 0.07 s)"
# (Santos et al., 2014).
## though the studies are not identical in task, there is a similar focus
## on the impact of caffeine on physical activity

sim_coffeeBefore <-
  rnorm(# 20 reaction times for the pre-coffee condition, assumed to be slower than
        # post-coffee condition
    20, mean = 42, sd = 5) # mean and sd in milliseconds)

sim_coffeeAfter <-
  rnorm(# 20 reaction times for the post-coffee condition, assumed to be faster than
        #pre-coffee condition
    20, mean = 37, sd = 7) # mean and sd in milliseconds)
```

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.

```r
# running a paired t-test on the above data
stat_tCoffee <- t.test(sim_coffeeBefore,
                       sim_coffeeAfter,
                       paired = TRUE,
                       alternative = "two.sided")
# viewing results
stat_tCoffee
```

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

```
##
##  Paired t-test
##
## data:  sim_coffeeBefore and sim_coffeeAfter
## t = 2.9416, df = 19, p-value = 0.008376
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##   1.847841 10.963529
## sample estimates:
## mean difference
##        6.405685
```

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

- Consistent with the assumption made, a cup of coffee improved reaction time of participants (N = 20) in the pen-reaching task, t(19) = 2.94, $p = 0.008376 < 0.05$, 95% CI[1.85, 10.96], MD = 6.40 milliseconds.

Section C

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

```
# creating a for loop
for (x in 1:25) { # for range 1 to 25
  print(x ^ 2) # print x squared
}
```

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
## [1] 36
## [1] 49
## [1] 64
## [1] 81
## [1] 100
## [1] 121
## [1] 144
## [1] 169
## [1] 196
## [1] 225
## [1] 256
## [1] 289
## [1] 324
## [1] 361
## [1] 400
## [1] 441
## [1] 484
## [1] 529
## [1] 576
## [1] 625
```

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

```
# creating a for loop
for (x in 1:25) { # for range 1 to 25
  if (x %% 2 != 0) { # for x that cannot be divided by 2 with no remainder
    print(x ^ 2) # print x squared
  }
}
```

```
## [1] 1
## [1] 9
## [1] 25
## [1] 49
## [1] 81
## [1] 121
## [1] 169
## [1] 225
## [1] 289
## [1] 361
## [1] 441
## [1] 529
## [1] 625
```

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.

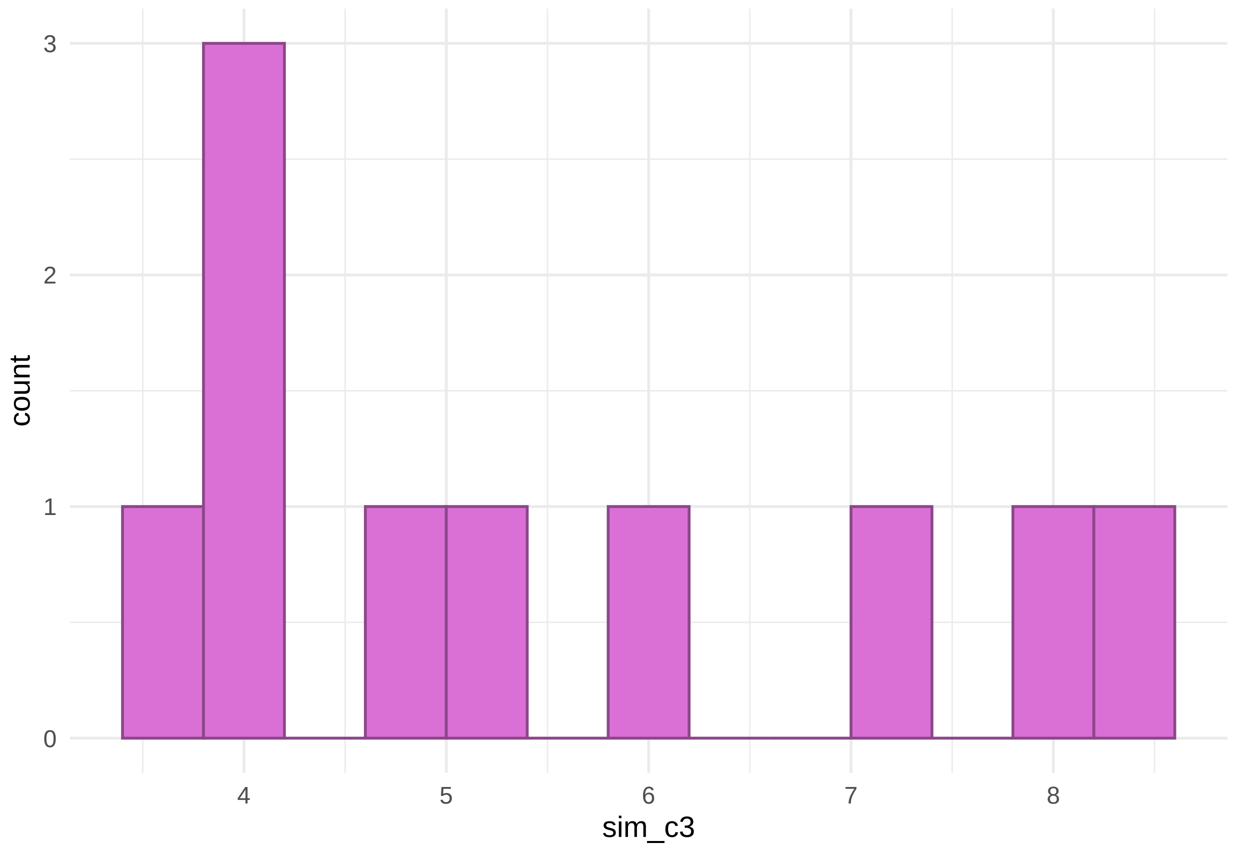
```r
set.seed(711) # for replicability

sim_c3 <- rnorm(10, mean = 5, sd = 2) # sampling 10 points from normal dist.

df_c3 <- data.frame(sim_c3) # creating a data frame with sampled points

# creating a histogram with sampled points
histo_c3 <- df_c3 %>%
  ggplot(aes(x = sim_c3)) +
  geom_histogram(binwidth = 0.4,
                 color = "orchid4",
                 fill = "orchid") +
  theme_minimal()

# viewing the histogram
histo_c3
```



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.

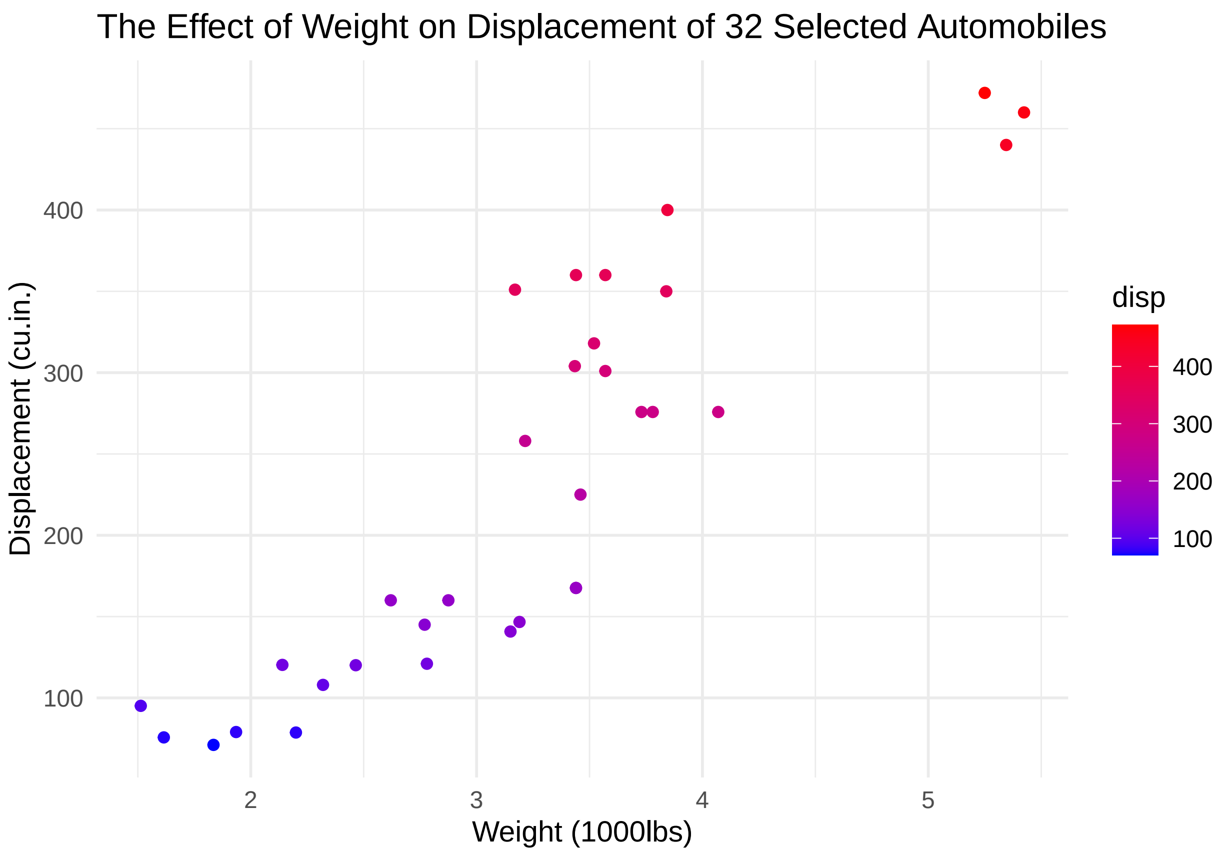
```
# accessing and (pre-)viewing data
data(mtcars)
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
# converting data to dataframe
df_mtCars <- data.frame(mtcars)

# creating a scatter plot
scatter_mtCars <- df_mtCars %>% # accessing df
  ggplot(aes(x = wt, y = disp, color = disp)) + # x = weight, y = displacement
  geom_point() +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "The Effect of Weight on Displacement of 32 Selected Automobiles", x = "Weight (1000lbs)
  theme_minimal()

# viewing scatter plot
scatter_mtCars
```



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.

```
set.seed(711) # for replicability

norm_c5 <- rnorm(1000, mean = 0, sd = 1) # randomly generating normal dist. with 1000 points

# creating if-else statement
result_c5 <- ifelse(norm_c5 > 0, "positive", "negative")
head(result_c5) # pre-view of results
```

```
## [1] "negative" "negative" "negative" "negative" "positive" "negative"
```

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.

```
set.seed(711) # for replicability

while (TRUE) {
  norm_c6 <- rnorm(1, mean = 1, sd = 2)  # sample number from normal dist.
  print(norm_c6) # print sampled number

  if (norm_c6 > 2) { # check if sampled number is greater than 2
    break # if yes, break out of the loop
  }
}
```

```
## [1] -0.108868
## [1] 0.09360558
## [1] 0.7229539
## [1] 0.02856451
## [1] 1.358995
## [1] -0.3072199
## [1] 3.287043
```

Section D

- for D1 – D4 please look at the JPG in my repo! :))

D5. Invert this 2x2 matrix using R:

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

```
#creating matrix P
P <- matrix(c(0, 8,
              1, -1), nrow = 2, byrow = TRUE)

# generating inverse of matrix P
inv_P <- matlib::inv(P)
inv_P # viewing inverse of P
```

```
##       [,1] [,2]
## [1,] 0.125    1
## [2,] 0.125    0
```

D6. Inversion for larger matrices (Using Gauss-Jordan)

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

```r
# Step 1: creating matrix T!
T <- matrix(c(1,-1, 2, 1,
              1, 0, -1, 2,
              2, 1, 0, -1,
              -1, 2, 1, 0), nrow = 4, byrow = TRUE)

# Step 2: creating a function to perform Gauss-Jordan elimination method!
gaussjordan <- function(X) {
  n <- nrow(X)

  # augmenting matrix T with identity matrix of same size (i.e. 4 x 4)
  aug_matrix <- cbind(X, diag(4))

  # Gauss-Jordan elimination!
  for (i in 1:n) {
    aug_matrix[i, ] <- aug_matrix[i, ] / aug_matrix[i, i]

    # subtracting multiples of the current row from other rows!
    for (j in 1:n) {
      if (i != j) {
        aug_matrix[j, ] <- aug_matrix[j, ] - aug_matrix[j, i] * aug_matrix[i, ]
      }
    }
  }

  # Step 3: extracting the inverted matrix
  inversed <- aug_matrix[, (n + 1):(2 * n)]
  return(inversed)
}

# Step 4: using the function to invert matrix T using Gauss-Jordan method!
inv_T <- gaussjordan(T)
inv_T # viewing results!
```

```
##            [,1]       [,2]          [,3]       [,4]
## [1,]  0.1111111  0.1111111  3.333333e-01 -0.1111111
## [2,] -0.1111111  0.1388889  1.666667e-01  0.3611111
## [3,]  0.3333333 -0.1666667 -2.775558e-17  0.1666667
## [4,]  0.1111111  0.3611111 -1.666667e-01  0.1388889
```

Section E

E1: Creating a ggplot with Aesthetics

- 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

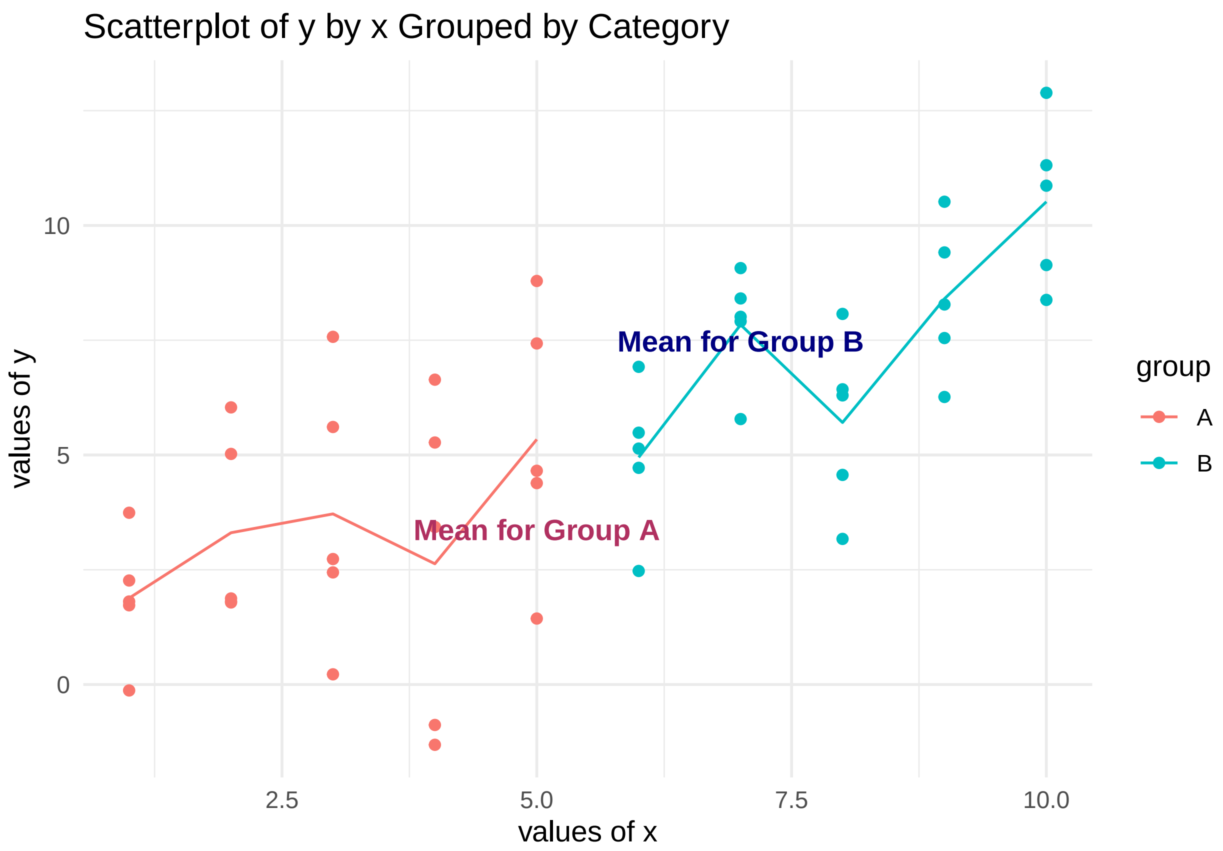
```r
# Code for simulating
set.seed(42)
df <- data.frame(
  x = rep(1:10, each = 5),
  y = rnorm(50, rep(1:10, each = 5), 2)
) %>%
  mutate(group = ifelse(x <= 5, "A", "B"))

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

# my plot code here! :D
scatter_df <- df %>%
  ggplot(aes(x = x, y = y, color = group)) + # grouping dots by 'group' variable!
  geom_point() +
  stat_summary(geom = "line", fun = mean, aes(group = group)) + # line representing the mean value of y
  labs(title = "Scatterplot of y by x Grouped by Category", x = "values of x", y = "values of y") +

  # perhaps not the prettiest or most optimal way to do this..
  annotate("text", x = 5, y = mean(df$y[df$group == "A"]), label = "Mean for Group A",
           color = "maroon", size = 4, fontface = "bold") +  # Subtitle for group A
  annotate("text", x = 7, y = mean(df$y[df$group == "B"]), label = "Mean for Group B",
           color = "navy", size = 4, fontface = "bold") + # Subtitle for group B
  theme_minimal()

# viwing the plot!
scatter_df
```



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)

# my normality testing code here!
normTest_data <- shapiro.test(data)

ifelse(
  normTest_data$p.value > 0.05,
  "Fail to reject null hypothesis",
  "Reject the null hypothesis"
)
```

```
## [1] "Fail to reject null hypothesis"
```

- Based on the Shapiro-Wilk test of normality, the data appears to be normal , $p = 0.9349 > 0.05$. It can thus be said, there is insufficient evidence to reject the null hypothesis that the data is normally distributed.

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).

```
# 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
model_df <- lme4::lmer(y ~ x + (1 | subj), data = df) # lmer accounting for repeated measures within su
summary(model_df)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 | subj)
##    Data: df
##
## REML criterion at convergence: 279
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.95391 -0.58745  0.08268  0.50048  1.91991
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subj     (Intercept) 0.08301  0.2881
##  Residual             0.87216  0.9339
## Number of obs: 100, groups:  subj, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   2.4830     0.2185  11.362
## x2            1.2764     0.2953   4.322
## x3            3.8563     0.2953  13.058
## x4            5.3101     0.2953  17.981
## x5            7.7449     0.2953  26.225
##
## Correlation of Fixed Effects:
##    (Intr) x2     x3     x4
## x2 -0.676
## x3 -0.676  0.500
## x4 -0.676  0.500  0.500
## x5 -0.676  0.500  0.500  0.500
```

- The results for the fixed effects suggest that as the levels of x increase from 2 to 5, there is a significant linear increase in the outcome variable y. Specifically, for each unit increase in x, y is expected to increase by 1.28 (95% CI: [0.69, 1.87]), 3.86 (95% CI: [3.26, 4.45]), 5.31 (95% CI: [4.71, 5.91]), and 7.74 (95% CI: [7.14, 8.33]) units, respectively.

Section F

F1: What is the ordinary least squares method and what do we use it for?

- **OLS and its Application:** Ordinary least squares (OLS) is a type of linear regression which involves the minimisation of residual squares i.e. making the squares values of the error as small as possible. Its application is limited to linear relations and it relies on a number of assumptions (fx. homoscedasticity, normality). However, it is easier to interpret, though it may not capture complex patterns accurately.

Bibliography

Santos, V. G. F., Santos, V. R. F., Felippe, L. C., De Almeida, J. W. L., De Moraes Bertuzzi, R. C., Kiss, M. a. P. D., & Lima-Silva, A. E. (2014). Caffeine reduces reaction time and improves performance in Simulated-Contest of taekwondo. Nutrients, 6(2), 637–649. https://doi.org/10.3390/nu6020637