Homework 23\_1

For all of the following tasks MATLAB code has to be written. Do not use the workspace to solve the tasks. Please submit all code in *one* script, named “lastname\_homework\_23\_1.m”.

Use “%%” to separate your code for the different Questions. Comment your code where appropriate.

You can work in groups of up to 4 people. Each group should submit just one script (you can list names as comments).

Submit your finished code in StudIP in the module *6.02.240\_1 Introduction to scientific programming I* under **files** in the folder **homework\_1\_submission.**

The data that must be used is stored in StudIP in the HW1 folder. Download the data and load the mat-files were appropriate.

Before you start writing, make sure you understand all tasks. If you have problems understanding the question, please use the Forum in StudIP to ask questions.

You might have to use functions unknown to you. Use the **doc fname** or **help fname** command in MATLAB to find out how they work.

**Section 1: Basics (55 P)**

**Q1: Logical comparisons** (15 P)

(a) Use Matlab to answer the question, “Is 5 smaller than 10?”. Store the answer as a variable. (2 P)

(b) Use Matlab to answer the question, “Is 7 smaller than 10 or is it positive?”. Use a comment to explain the answer. (4 P)

(c) Use Matlab to answer the question, “Is 7 bigger than 5 and is it even?”. Use a comment to explain the answer. (5 P)

(d) Use Matlab to answer the question, “Is the word ‘duck’ *not* the same as the word ‘goose’ ?”. Use a comment to explain the answer. (4 P)

**Q2: Conditionals** (25 P)

If the temperature in Celsius is *below* 0 degrees, a puddle of water will freeze.

(a) Create a variable that stores today’s temperature. (Enter the temperature yourself.) (1 P)

(b) Create a conditional. If today’s temperature will cause the puddle to freeze, it should display ‘The puddle will freeze.’ Else, it should display, ‘The puddle will not freeze.’ (5 P)

(c) Copy your code from part b to answer this question. Extend your conditional to also show what the temperature is *in a full sentence.* For example, ‘The puddle will freeze as the temperature today is -25 degrees Celsius.’ (8 P)

(d) Copy your code from part c to answer this question. It may rain or snow depending on the temperature *if* the weather is bad. Create a variable that stores whether the weather today is bad. Now extend your code to show whether it will rain, snow, or neither. (11 P)

**Q2: Logical Bonus** (5 P)

Create a random vector X with randi(10, 100,1). If we want to count all the elements which have a value between 5 and 8. The following options give you to different results:

A: elem\_btw\_5\_8=sum(X>5&X<8);

B: elem\_btw\_5\_8=sum(5<X<8);

Explain shortly in comments, which version is correct, and what the difference is.

**Q3: Matrices, subscript, and linear indexing** (10 P)

(a) Create a 9X9 matrix of zeros. Let’s call this **square**. (1 P)

(b) Subscript index to the 2nd row, 2nd column of **square**. Linearly index to the same element. (2 P)

(c) Replace the element at the center with a 5. (1 P)

(d) Create a 2 row X 9 column matrix of ones. Let’s call this **rectangle**. (2 P)

(e) Replace the first two *columns* of **square** with **rectangle**. (2 P)

(f) Multiply the first *rows* of **square** with **rectangle** *element-wise*. The result should be 2 rows X 9 columns. (2 P)

**Q4: Interpreting code/Using doc** (5 P)

You will have to read the documentation to help you with this one.

(a) Explain as comments what the following lines do in as much detail as possible: (3 P)

ToSearch = [1:4;5:8];

LinInds = find(ToSearch > 4);

(b) Now explain what the following lines do differently: (2 P)

ToSearch = [1:4;5:8];

[rows,columns] = find(ToSearch > 4);

**Section 2: Indexing, Loops and data types (30 P)**

**Q5: Divide by 2 (10 P)**

(a) Create a variable that starts off being equal to 1000. (1 P)

(b) How many times do you think you can halve this variable while it remains positive? Answer this as a comment before you create a loop. (1 P)

(c) Test your hypothesis with a loop. Add a counter variable within your loop to check how many times the loop runs. (5 P)

(d) Are you surprised by the result? Explain why you got this result in a comment. (1 P)

(e) Now repeat the steps a-c, but make the variable a double instead of a single. Copy-paste if you need to - don’t overwrite your previous answers!  *Briefly* explain why your loop ran a different number of times than before. (3 P)

**Q6: Reaction time (20 P)**

Load the data homework1.mat by using the full path to its location on the hard drive.

Pet\_data is a 101x2x5 matrix with the following data:

|  |  |
| --- | --- |
| Stimuli/Trigger time (ms) | Response time (ms) |
| 100 | 524 |
| 500 | 632 |
| ... | ... |

Stimuli time is the time the stimuli was presented to the participants (in relation to 0, when the experiment starts). Response time is the time the participant hit the response button (again in relation to 0), delivering the answers to the system.

Each row in every page is a trial. The third dimension represents the participant number (each page is 1 participant)

For participant 1 (page 1).

1. Calculate the reaction time by subtracting the stimuli onset time (first column) from the response time (second column). (1)
2. Remove any trials where responses happen before triggers (so either response time < reaction time or the reaction from task 1 is smaller than zero). (2)
3. Remove trials with improbable reaction time ( < 250ms and > 450ms). (5)
4. Find the average reaction of participant 1 over all trials (2)

For 5 participants (you need to use a for-loop here)

1. Repeat tasks A to D for all 5 participants (save each participant to a new variable) (10)

**Section 3: Data Operations and plotting (15 P +(10 Bonus P))**

**Q7: Blood pressure recording**

In BP\_1.mat, you can find the recording of a patient’s blood pressure over the course of a day in the variable BP. The BP was measured every minute, that means you have 1440 data points over a day. We want to analyze this data-set and look for patterns and anomalies in the data.

Blood pressure is usually measured in two quantities for a given timeframe: The systolic (higher) and the diastolic (lower) value. If you inspect the variable BP, you’ll see that it’s a 2x1440 <single array> (Blood pressure type X Minutes)

1. We are not interested in the differences between systolic and diastolic pressure. Calculate the mean over both values for every timepoint. We will continue with this **mean\_pressure**. (3 P)
2. Plot the time course of **mean\_pressure** against the *hours* of the day. Plot no solid line but circular-markers. Add a title and axis labels. (7 P)
3. Find the timepoints (hours), where the **mean\_pressure** is exceptionally high (at least 3 standard deviations above the average value). (5 P)
4. (Bonus) With the xline-function, you can draw a vertical line at a specific x-position in your plot. Figure out, how you can use this function to mark all the timepoints, with the high pressure you found in (c.) Your plot could look like Fig.1 in the end. (10 P)

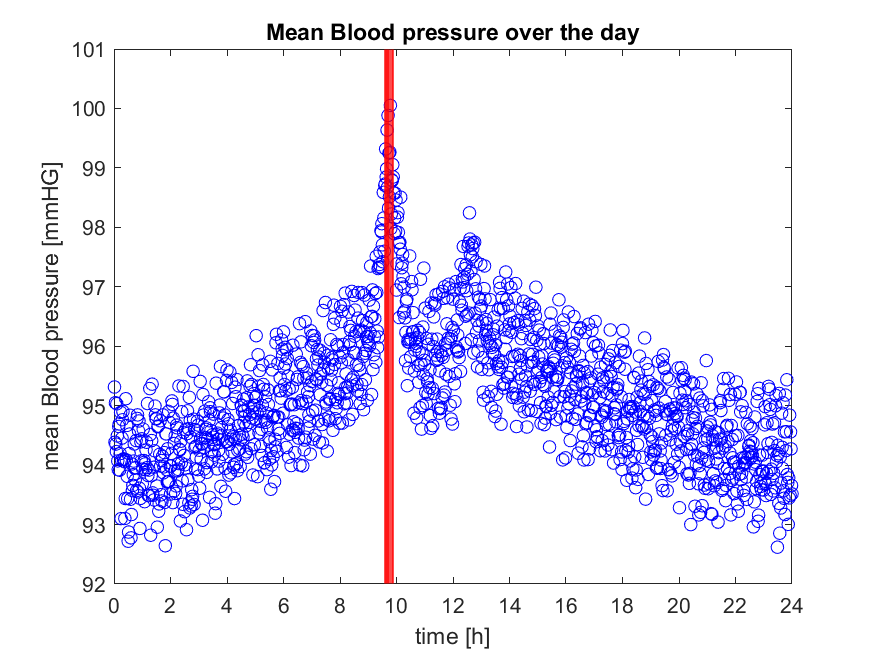


Figure 1: Exemplary figure for blood pressure question

**Bonus section: Decipher a secret message (13 Bonus Points)**

Have you ever tried to decipher a secret message? After question 6 (on reaction times) you should have a variable called **secret\_message** in your workspace. This message was enciphered using a cousin of the rail-fence cipher. The original, readable message was split in 2. The characters from each half were alternated to end up with **secret\_message**.

(a) Index to the first element in **secret\_message**. Index to the second element. What do you get (write comments)? Do blank spaces count as characters? (3 P)

(b) Create a vector which starts with 1 and goes until the length of **secret\_message** in steps of 1. Let’s say you call this vector **key**. (2 P)

(c) *Use logical indexing* to extract the odd elements of key and store them in a vector. Do the same thing for the even elements. (3 P)

(d) You can use these odd and even vectors to index elements out of **secret\_message**. Save these character vectors. (1 P)

(e) Concatenate the two character vectors (odd before even) to get the deciphered message. (1P )

(f) Now that you have been guided through all the steps to decipher **secret\_message**, try to encipher your own message by reversing the procedure. Do this using Matlab! Share your message with your classmates and see whether they can decipher it. (**doc ceil** may help.) (5 P)