

Decision Making Under Uncertainty, MATH 20011

Homework 1: Introduction to R and R Markdown

Christian Auman

2026-01-28

Instructions

- Replace “Your Name Here” in the author field above with your name.
- Follow the prompts in each part of this assignment and edit this .Rmd file directly. There are two parts to this assignment.
- As you provide your answers, notice how different sections of the document are formatted. Pay attention to features such as lists, **bold** and *italic* text, equations within text, links, and code chunks.
- Knit your finished document to PDF and submit the PDF to Canvas.

PART A

In this part of the assignment, you will learn how to:

- Use basic Markdown formatting (lists, **bold**, *italics*)
- Create links and block quotes
- Embed and evaluate R code chunks

Problem 1: Update Your Information (10 marks)

Replace the placeholder items below with your own information:

- Name: **Christian Auman**
- Field of study: **Computer Science**
- Two courses you are taking this semester
 - **Decision Making Under Uncertainty**
 - **Operating Systems**
- Email: **cauman@kent.edu**

Problem 2: Text Formatting: Bold, Italics, Links and Quotes (10 marks)

Hi, my name is **Christian Auman**. My favorite movie of all time is *The Maze Runner*. My favorite online learning medium is Udemy. My favorite quote from a book, movie, or person is

One small step for man, one giant leap for mankind

Problem 3: Inline Code (10 marks)

Complete the inline R code below to compute:

- The sum of 12 and 13 is 25
- The difference between 19 and 7 is 12

Problem 4: Quadratic Function (15 marks)

Recall the quadratic function we wrote in class, we will output the function in an R pdf and use it to solve two quadratic problems. The quadratic function is provided in the code chunk below:

```
quad <- function(a, b, c) {  
  x1 <- (-b + sqrt(b^2 - 4 * a * c)) / (2 * a)  
  x2 <- (-b - sqrt(b^2 - 4 * a * c)) / (2 * a)  
  c(x1, x2)  
}
```

Notice that we use `echo = TRUE` to show the function. Explain what that does and what would have happened if we `echo = FALSE`.

Answer: Echo will echo the function definition whenever we run the code which defines it. If it was not set, there would be no output in the terminal

1. Suppose we want to solve the quadratic problem,

$$x^2 - 5x + 6 = 0$$

- . In the code chunk named `quadratic1` below, provide the code that will solve this problem using the function provided in the code chunk above.

```
# Call Quadratic_fn() here with appropriate arguments  
quad(1, -5, 6)
```

```
## [1] 3 2
```

2. Suppose we want to solve another quadratic problem, $x^2 - 2x + 1 = 0$. In the code chunk named `quadratic2` below, provide the code that will solve this problem using the function provided in the code chunk above.

The second quadratic problem is

```
# Call Quadratic_fn() here with appropriate arguments  
quad(1, -2, 1)
```

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

PART B

In this part of the assignment, you will complete several tasks in R and provide your results in the corresponding code chunks. Some tasks may use functions that are new to you. Remember, you can always use R's built-in help system, either in the RStudio editor or the console, to learn more about unfamiliar functions.

Problem 1: Basic Arithmetic in R (10 marks)

R can be used as a simple calculator. Perform the following calculations directly in your R Markdown file and output the result.

Calculate the sum of 45 and 73, then subtract 18 from the result.

```
(45 + 73) - 18
```

```
## [1] 100
```

Calculate the sum of 15 and 5, then multiply the result by 5.

```
(15 + 5) * 5
```

```
## [1] 100
```

Problem 2: Assigning Variables (10 marks)

In R, you can store values in variables for later use. Assign the value 50 to a variable called `x`, and then assign the value of `x + 50` to a new variable called `y`. Output the resulting value of `y`.

```
x <- 50  
y <- x + 50  
y
```

```
## [1] 100
```

Problem 3: Vectors (10 marks)

The code block below creates a vector `v1` that contains the numbers 5 to 15.

```
v1 <- c(5:15)  
v1
```

```
## [1] 5 6 7 8 9 10 11 12 13 14 15
```

Use the `str()` function to verify the data type vector `v1`.

```
str(v1)
```

```
## int [1:11] 5 6 7 8 9 10 11 12 13 14 ...
```

Use the `mean()` function to calculate the average of the numbers in `v1`.

```
mean(v1)
```

```
## [1] 10
```

Problem 4: Vectors (10 marks)

The below code block assigns a character value to the variable `favorite_color` and outputs the result.

```
# Example Code
```

```
(favorite_color <- "blue")
```

```
## [1] "blue"
```

Use the `c()` function to create and output a character vector with 3 different colors as the elements. By “wrapping” your element in quotations, R will automatically detect the data type.

```
v2 <- c("red", "green", favorite_color)  
v2
```

```
## [1] "red"   "green" "blue"
```

Use the `str()` function to verify the data type of your vector.

```
str(v2)
```

```
##  chr [1:3] "red" "green" "blue"
```

Problem 5: Creating a Data Frame (15 marks)

A data frame with two columns is created below: `x_values` containing the numbers 1 to 5, and `y_values` containing the squares of those numbers.

```
# Example Code  
df1 <- data.frame(  
  x_values = c(1, 2, 3, 4, 5),  
  y_values = c(1, 4, 9, 16, 25)  
)  
  
df1 # This outputs the data frame
```

```
##   x_values y_values  
## 1       1       1  
## 2       2       4  
## 3       3       9  
## 4       4      16  
## 5       5      25
```

Suppose you have taken a survey of 4 of your closest friends. You asked them what their favorite sport is and how many siblings they have. Complete the code block below so that the data frame represents the information from your survey (you can make up how many siblings each person has). The dataframe will have three columns: `friend_name`, `favorite_sport` and `siblings_count`.

```
df2 <- data.frame(  
  name = c("Nathan", "Jack", "Xavier", "Jake"),  
  favorite_sport = c("WWE", "Soccer", "Football", "Football"),  
  siblings_count = c(1, 2, 0, 0)  
)  
  
df2  
  
##      name favorite_sport siblings_count  
## 1 Nathan          WWE             1  
## 2 Jack            Soccer           2  
## 3 Xavier          Football         0  
## 4 Jake            Football         0
```

Use the `str()` function to check the data type of each column in your data frame.

```
str(df2)
```

```
## 'data.frame': 4 obs. of 3 variables:  
## $ name : chr "Nathan" "Jack" "Xavier" "Jake"
```

```
## $ frvorite_sport: chr  "WWE" "Soccer" "Football" "Football"  
## $ siblings_count: num  1 2 0 0
```