

# HW0 – Math 58B

**due: Thursday, January 22, 2026**

your name here

The syllabus is an agreement between the professor and students. It explains your responsibilities, lays out the structure, and gives you information on how best to achieve the course goals. Like any agreement, it must be read carefully and referenced frequently to answer questions. The fun activity below will highlight some important parts of the syllabus and give you a chance to try out R!<sup>1</sup>

## **Instructions**

Answer each question below by typing a response. Render the file to a compiled pdf, and submit it to Gradescope (via Canvas).

Hint: To compile to pdf, you might need to use the `tinytex` package (you'll know you need it if you can render to html but not to pdf). If that is the case, then in the console (below, do you see where it says "Console"? only one time ever, type the following. Ask me if it doesn't work automatically for you. (Don't spend hours figuring this part out, ask me immediately.)

```
install.packages('tinytex')
tinytex::install_tinytex()
```

<https://yihui.name/tinytex/>

## **Q1.**

On which dates are the exams?

After you've marked the exam dates in your calendar, answer the following question: what type of calendar do you keep? (e.g., Google calendar, outlook, paper journal, post-its all over your desk, etc.)

---

<sup>1</sup>adapted from David White at Denison University.

**Q2.**

Provide three pieces of information from the syllabus related to class participation.

**Q3.**

Will course notes be available and posted? If so, where?

**Q4.**

What is the software program we are using for the class? Ask one question that you have about R.

**Q5.**

Run the code below one line at a time. Provide a few words describing what each line of code is doing.

The words of explanation could come before or after the R chunk, just like any sentences written to a client describing the analysis.

```
mydata <- c(1:10) # the words of explanation could come after the hashtag
mydata
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
mydata^2
```

```
[1] 1 4 9 16 25 36 49 64 81 100
```

```
sample(mydata, size = 25, replace = TRUE)
```

```
[1] 7 2 1 1 7 10 2 6 8 2 3 10 2 10 8 4 8 1 1 6 8 4 1 7 2
```

```
mydata2 <- sample(mydata, size = 25, replace = TRUE)
mydata2
```

```
[1] 2 3 7 4 4 5 9 6 5 6 4 5 3 7 2 5 5 7 5 4 3 7 10 2 10
```

## Q6.

What are the reflection questions and ethics considerations? Where do you find them? What should you do with them?

## Q7.

How do you plan to use AI in Math 58B?

## Q8.

Nice job! Run the chunk of code below. You might need to install the **praise** package. (See above.)

```
praise()
```

```
[1] "You are terrific!"
```

Note: if you want any of your output to remain constant, use the **set.seed()** function. The function will control the randomness associated with the task you've asked of R. For example, you asked R to sample from some integers. Do you want the sample of integers to stay the same every time? Well, use **set.seed()**! The only argument you need for **set.seed()** is a single integer. You can choose *any* integer you want. And the function goes before (either right before or at the top of the file) the command where R is doing something random. Here is an example of some code (which won't be run because I set **eval = FALSE**) where the randomness is controlled. Try it yourself in your work above.

```
set.seed(47)
sample(mydata, size = 25, replace = TRUE)
```