Problem set 1

2026-02-02

Instructions/Setup

Download source

- Make sure you have a GitHub account. In your GitHub account, create a GitHub repository named `STAT3000`.
- Create an RStudio Project, e.g., `stat3000`, connected to the GitHub repo (clone or connect existing).
- Answer each item **inside this Quarto file**. Show commands/code where requested.

Tasks

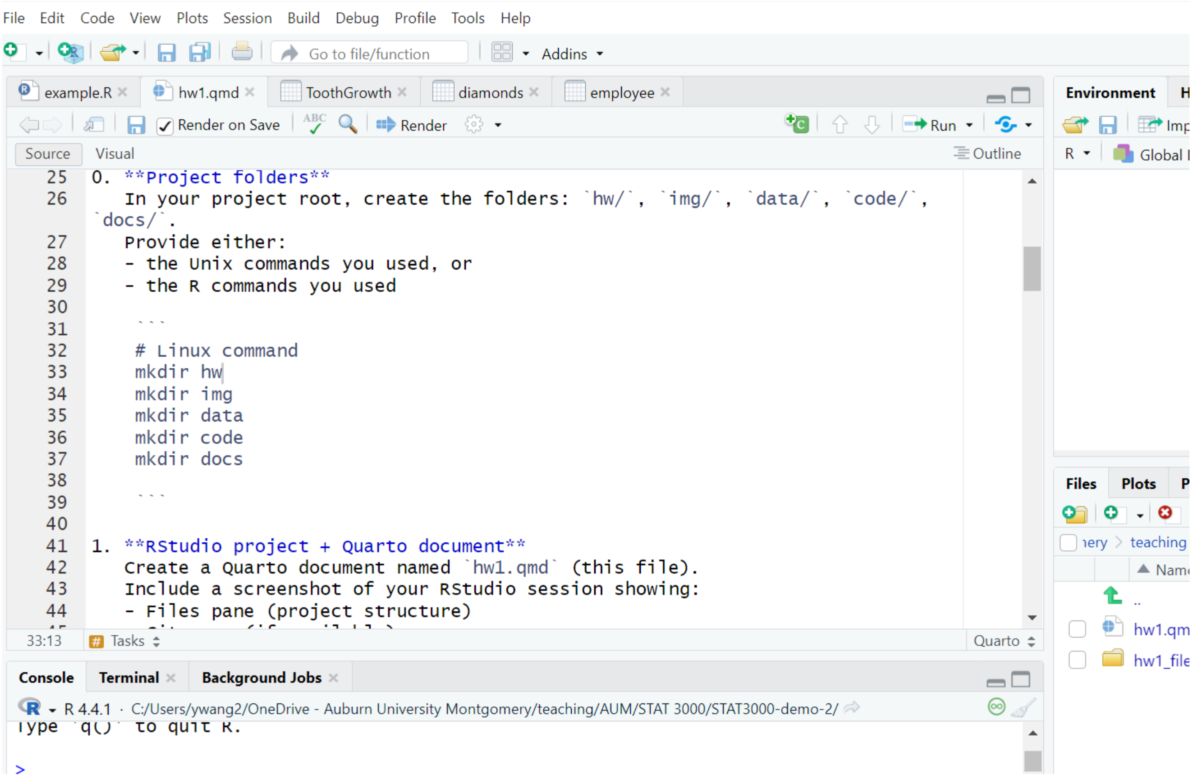
0. **Project folders**
   In your project root, create the folders: `hw/`, `img/`, `data/`, `code/`, `docs/`.
   Provide either:

   - the Unix commands you used, or
   - the R commands you used

   ```
   # Linux command
   mkdir hw
   mkdir img
   mkdir data
   mkdir code
   mkdir docs
   ```

1. **RStudio project + Quarto document**
   Create a Quarto document named `hw1.qmd` (this file).
   Include a screenshot of your RStudio session showing:

   - Files pane (project structure)



```
0. **Project folders**
   In your project root, create the folders: `hw/`, `img/`, `data/`, `code/`,
`docs/`.
   Provide either:
   - the Unix commands you used, or
   - the R commands you used

    ```
    # Linux command
    mkdir hw
    mkdir img
    mkdir data
    mkdir code
    mkdir docs

    ```


1. **RStudio project + Quarto document**
   Create a Quarto document named `hw1.qmd` (this file).
   Include a screenshot of your RStudio session showing:
   - Files pane (project structure)
   - Git pane (if available)
```

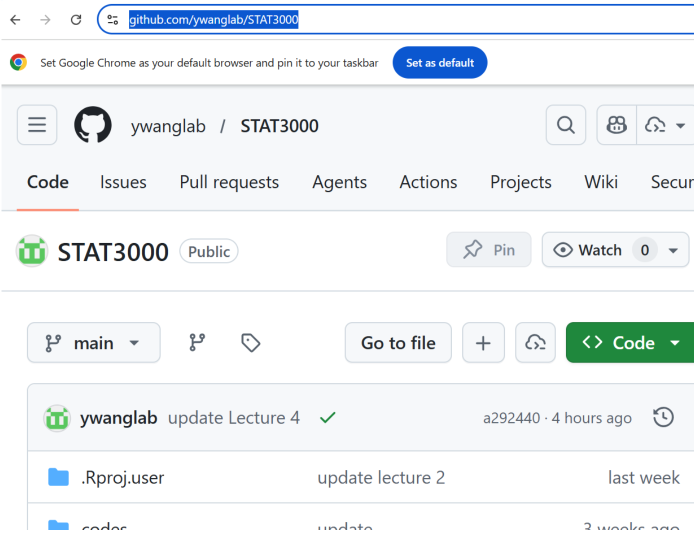
```
Type 'q()' to quit R.

>
```

Figure 1: Screenshot of RStudio

- Git pane (if available)

2. **GitHub + repo setup**
   In your Quarto document, include:

   - the repo URL https://github.com/ywanglab/STAT3000



ywanglab update Lecture 4 ✓ a292440 · 4 hours ago

.Rproj.user update lecture 2 last week

codes update 3 weeks ago

   - a screenshot showing the repo page on GitHub

3. **Create and include a plot image**
   Create a simple plot in R (any plot you like), save it as `img/plot.png`, and include it below.

```
# Example idea (you can change it):
png("../img/cars.png", width = 800, height = 500) #open png device
plot(cars)
dev.off() # close the current device and fall backs to "null derice 1" (no device open)
```

```
pdf
  2
```

4. **Define a function and compute real roots** In this document, define coefficients $a = 2$, $b = -5$, $c = -3$. Consider $f(x) = ax^2 + bx + c$. Print the **real** solutions of $f(x) = 0$ (if any).

   - If the discriminant is negative, print a message like `"No real roots"`.

```
  a <- 2
  b <- -5
  c <- -3
  #
```

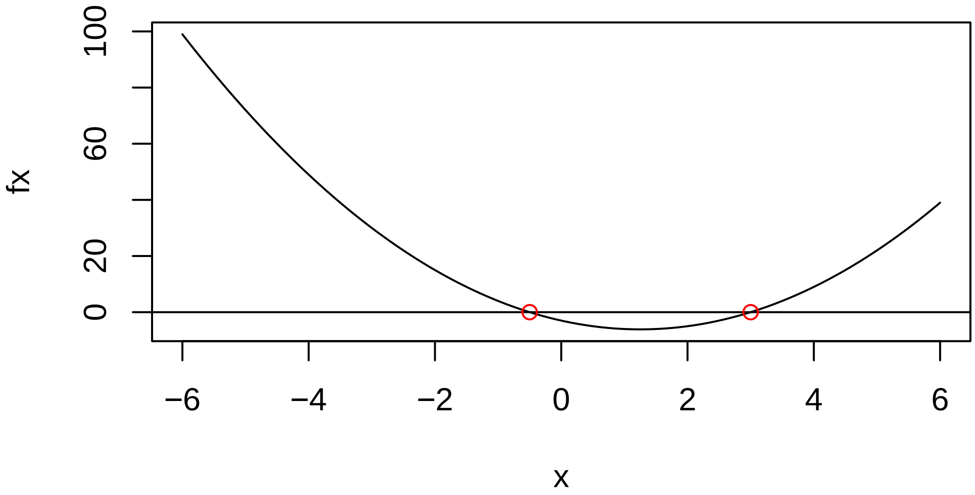
```
      disc <- b^2 - 4*a*c


      # Your code here to print only real roots
     if (disc <0 ) {
       cat("The discriminant is negative, hence no real roots")
       } else {
        x1<- 1/(2*a)*(-b+sqrt(disc))
        x2 <- 1/(2*a)*(-b- sqrt(disc))
        cat("The roots are x1 =", x1, ", x2 = ", x2)
      }
```

```
The roots are x1 = 3 , x2 =  -0.5
```

5. **Graph the quadratic on a specified interval** Make a graph of $f(x)$ versus x for $x \in (-6, 6)$. Add:

    - a horizontal line at $y = 0$
    - points for the real roots (if they exist)

```
    x <- seq(-6, 6, length.out = 400)
    fx <- a*x^2 + b*x + c

    # Hint: use plot(x, fx, type="l") and abline(h=0)
    plot(x, fx, type="l")
    abline(h=0) #h=0: horizontal y-inercept; v=t: vertical line with x-intercept t; abline(a
    points(c(x1,x2), c(0,0), col="red") # add ponts to an exsiting plot
```



6. **Write coefficients to a text file using Unix** Use Unix to create a file `data/coefs.txt` containing a single line with:

   ```
   2 -5 -3
   ```

   Show the Unix commands you used:

   ```
   # Your commands here
   printf "2 -5 -3" > ../data/coefs.txt
   # or
   # echo "2 -5 -3" > ../data/ceofs.txt
   ```

7. **Copy the Quarto file using Unix** Use Unix to copy `hw1.qmd` into `code/` and name it `quadratic.qmd`. Show the Unix command(s):

   ```
   # Your commands here (assumng your curretn directory is the parent director of hw/ and c
   cp hw/hw1.qmd code/quadratic.qmd
   ```

8. **Read coefficients from file using a relative path**

   Edit `code/quadratic.qmd` so it reads `a`, `b`, `c` from `data/coefs.txt` using a **relative path**. In *this* document, show the R code you used (it should also appear in `quadratic.qmd`).

```
    # Your code here (example outline):

    coefs <- scan("../data/coefs.txt") #read numbers/string and return a vector

    a <- coefs[1]; b <- coefs[2]; c <- coefs[3]

    # then compute real roots again

      #
      disc <- b^2 - 4*a*c


      # Your code here to print only real roots
     if (disc <0 ) {
       cat("The discriminant is negative, hence no real roots")
       } else {
        x1<- 1/(2*a)*(-b+sqrt(disc))
        x2 <- 1/(2*a)*(-b- sqrt(disc))
        cat("The roots are x1 =", x1, ", x2 = ", x2)
      }
```

```
The roots are x1 = 3 , x2 =  -0.5
```

9. **Switch to an absolute path and test portability**

   In `code/quadratic.qmd`, replace the relative path with: `file.path(getwd(), "data/coefs.txt")` Render it once to confirm it works.

```
    # Your code here (example outline):

    coefs <- scan(file.path(getwd(), "../data/coefs.txt")) #
# getwd(): get the abs. path of the current working directory
# file.path(): build full abs path that works in any os. Second argularment is relative to th

    a <- coefs[1]; b <- coefs[2]; c <- coefs[3]

    # then compute real roots again

      #
      disc <- b^2 - 4*a*c
      # Your code here to print only real roots
     if (disc <0 ) {
       cat("The discriminant is negative, hence no real roots")
       } else {
        x1<- 1/(2*a)*(-b+sqrt(disc))
        x2 <- 1/(2*a)*(-b- sqrt(disc))
        cat("The roots are x1 =", x1, ", x2 = ", x2)
      }
```

```
The roots are x1 = 3 , x2 =  -0.5
```

Then **move the entire project folder** to a new folder name (e.g., `RtmpXXXX`). Re-render `code/quadratic.qmd`. Does it still render?

Briefly explain what happened, then switch back to a relative path and confirm it renders again.

10. **Render to PDF and publish to GitHub Pages-style docs folder**

    Render a PDF version of `hw1.qmd` (or `code/quadratic.qmd`) and place the PDF into `docs/`. Show the terminal command(s) you typed:

    ```
    # Your command here, e.g.
    # quarto render hw1.qmd --to pdf --output-dir docs
    ```

11. **Git workflow** Make at least **three commits** with meaningful messages, such as:

- `Add initial project structure`

```
git add -A
git commit -m "Add initial project structure"
git push
```

- `Add quadratic root computation`
- `Add rendering output to docs`

  In this document, paste the output of:

  ```
  git log --oneline --decorate -n 5
  ```

12. **Push to GitHub** Push your work to GitHub. In this document, include:

- a screenshot of your GitHub repo showing the folders `img/`, `data/`, `code/`, `docs/`
- the link to the rendered PDF file in your repo (if applicable)

Submission checklist

□ Repo exists and is named `HW1`
□ `hw1.qmd` renders without errors
□ `img/plot.png` exists and is included in the document
□ `data/coefs.txt` exists and is read using a relative path
□ `docs/` contains the rendered PDF
□ At least three commits + pushed to GitHub