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:

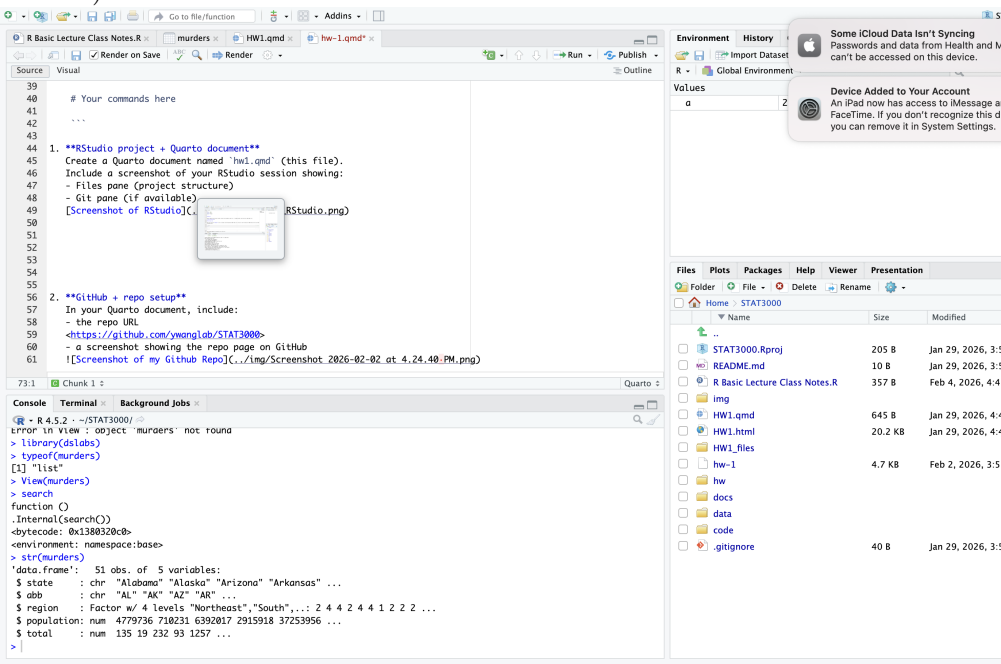
   - the Unix commands you used, or
   - the R commands you used

   ```
   # Linux command
   ```

   ```
   # Your commands here
   mkdir hw
   mkdir img
   mkdir data
   mkdir docs
   ```

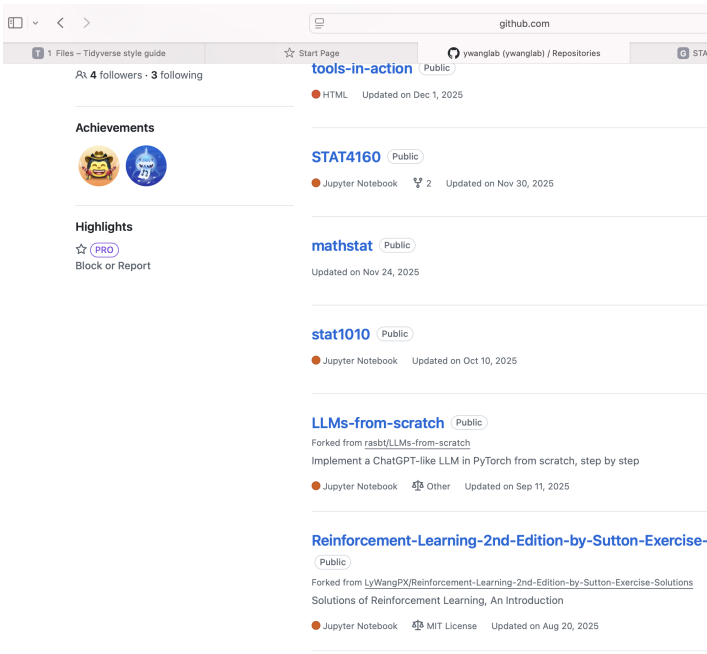
1. **RStudio project + Quarto document**
   Create a Quarto document named `hw1.qmd` (this file).
   Include a screenshot of your RStudio session showing:

   - Files pane (project structure)
   - Git pane (if available)



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

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



- 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()
```

```
pdf
  2
```

```
png("../img/plot.png", width = 800, height = 500)
plot(cars)
dev.off()
```

```
pdf
  2
```

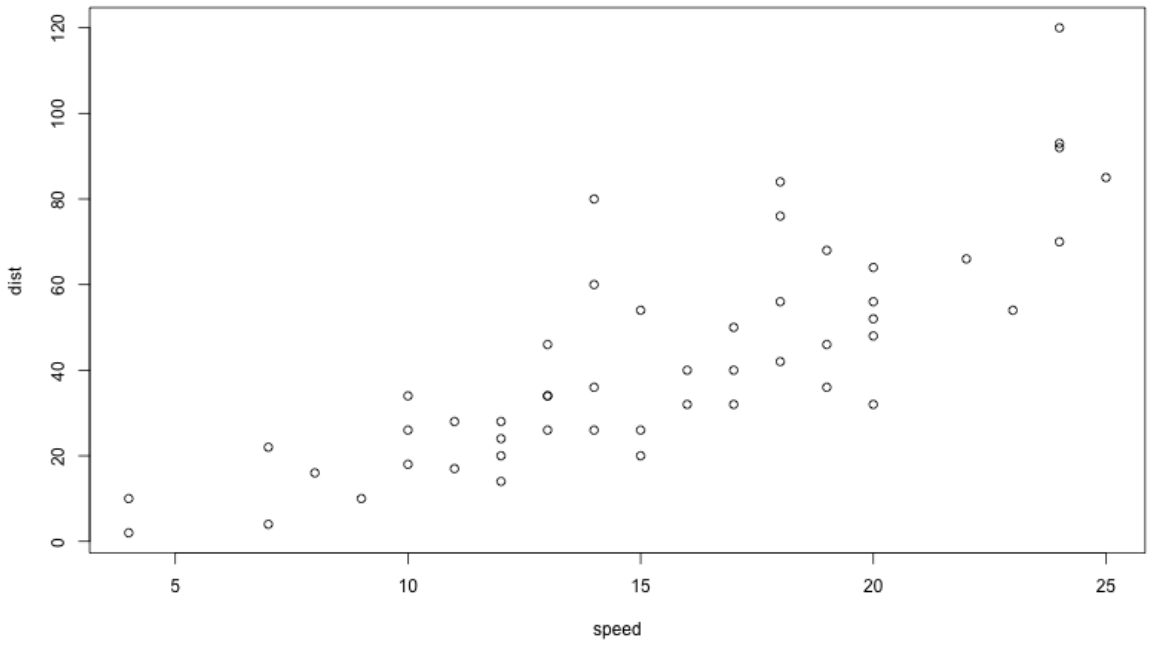


Figure 1: Simple cars plot

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"`.

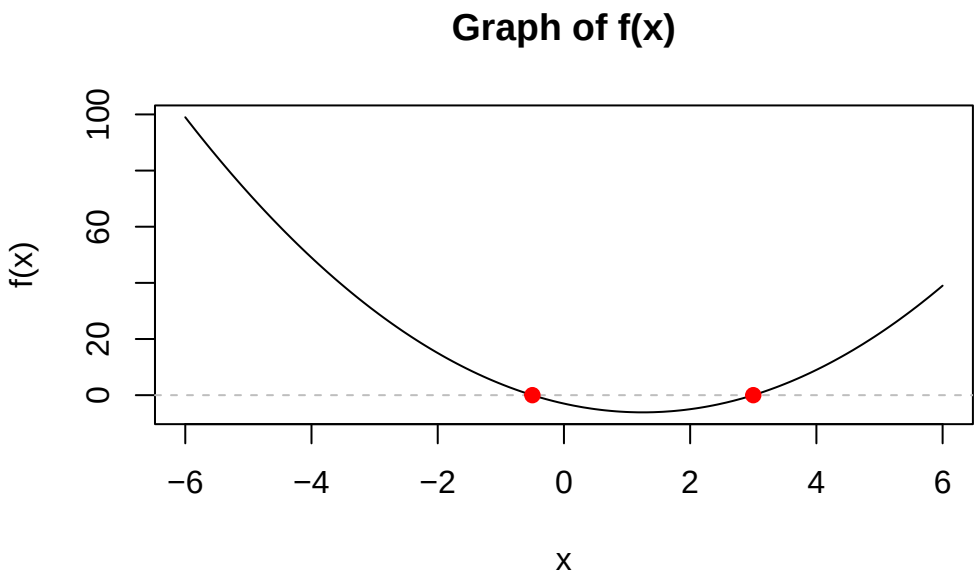
```
  a <- 2
b <- -5
   c <- -3
   #
   disc <- b^2 - 4*a*c
   disc
```

```
[1] 49
```

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

```r
   # Create x values and compute f(x)
x <- seq(-6, 6, length.out = 400)
fx <- a * x^2 + b * x + c

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

# Create x values and compute f(x)
x <- seq(-6, 6, length.out = 400)
fx <- a * x^2 + b * x + c
```

```r
# Plot f(x)
plot(x, fx, type = "l",
     xlab = "x", ylab = "f(x)",
     main = "Graph of f(x)")

# Horizontal line at y = 0
abline(h = 0, col = "gray", lty = 2)

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

# Plot real roots if they exist
 if (disc >= 0)
# roots <- # Plot real roots if they exist
if (disc >= 0) {
  roots <- c((-b + sqrt(disc)) / (2*a),
             (-b - sqrt(disc)) / (2*a))
  points(roots, rep(0, length(roots)), col = "red", pch = 19)
}
```




```
# 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)
```

6. **Write coefficients to a text file using Unix** Use Unix to create a file `data/coefs.txt` containing a single line with: mkdir -p data echo "2 -5 -3" > data/coefs.txt

   ```
   2 -5 -3
   ```

   Show the Unix commands you used:

   ```
   mkdir -p data
   echo "2 -5 -3" > data/coefs.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):

   ```
   mkdir -p code
   cp 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`).

```r
coefs <- scan("../data/coefs.txt")

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

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

```
[1] 49
```

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.

   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:

    ```
    quarto render hw/hw-1.qmd
    ```

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`

```
git add .
git commit -m "Implement quadratic discriminant and root calculations"
```

- `Add rendering output to docs`

```
git add docs/
git commit -m "Render PDF output to docs folder"
```

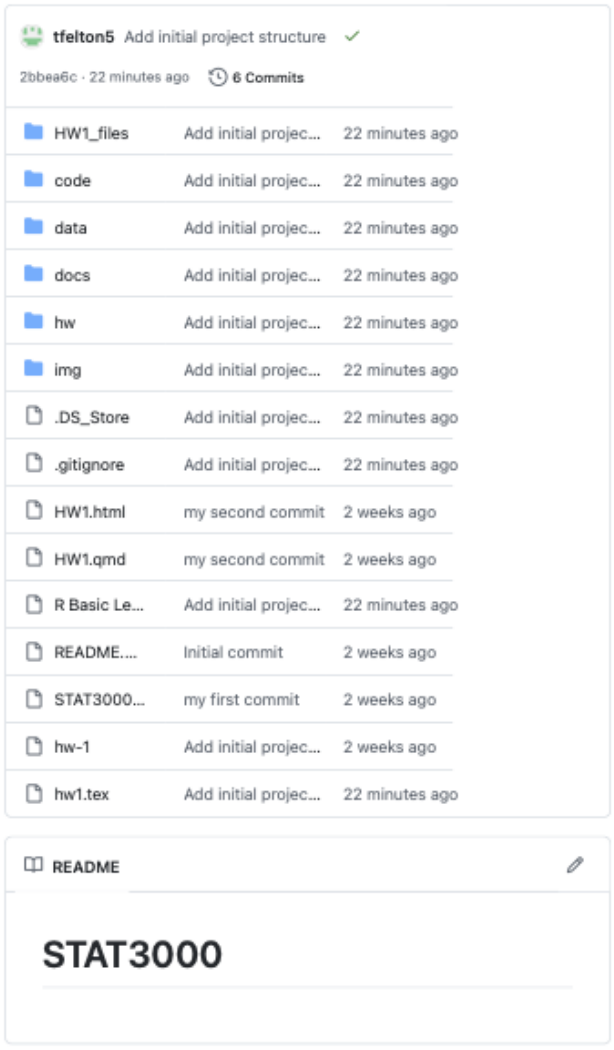
In this document, paste the output of:

```
2bbea6c (HEAD -> main, origin/main, origin/HEAD) Add initial project structure
cf6b925 Add initial project structure
7584aec my second commit
3d17757 update docs
6dcdc65 my first commit
```

```
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