Applied Quantitative Methods for the Social Sciences II
MA in Social Sciences, UC3M, Spring 2026

Assignment 1: Setting Up Workflow

Instructions:

- Deadline: **February 12, before class**
- This assignment walks you through setting up Git and GitHub
- You will use this setup to submit all assignments throughout the course
- Complete all the tasks below and send me the link to your GitHub repository

Contents

1 Task 1: Create a GitHub Account

If you don't already have a GitHub account, create one at https://github.com.

1. Go to https://github.com and click "Sign up"
2. Choose a **professional username**—you may use this for years in your academic career
3. Use your university email or a professional email address
4. Complete the verification process

---

2 Task 2: Create a Repository for This Course

Create a new **public** repository. Name it something like `aqmss2` or `quant-methods-2026`.

2.1 Option A: Using the GitHub Web Interface

1. Click the "+" icon in the top-right corner of GitHub
2. Select "New repository"
3. Fill in the form:
   - **Repository name**: `aqmss2` (or similar)
   - **Description**: "Assignments for AQMSS II, Spring 2026"
   - **Visibility**: Select **Public**
   - Check the box "Add a README file"
4. Click "Create repository"

2.2 Option B: Using the Command Line

First, make sure Git is installed on your computer. Then run:

```
# Create a new directory and initialize Git
mkdir aqmss2
cd aqmss2
git init

# Create a README file
echo "# AQMSS II - Problem Sets" > README.md

# Stage and commit the file
git add README.md
git commit -m "Initial commit"

# Set up the remote repository (create it on GitHub first, without README)
git branch -M main
git remote add origin https://github.com/YOUR-USERNAME/aqmss2.git
git push -u origin main
```

Note: Replace YOUR-USERNAME with your actual GitHub username.

2.3 Option C: Using RStudio

1. First, create an empty repository on GitHub (without README)
2. In RStudio: File → New Project → Version Control → Git
3. Paste your repository URL: `https://github.com/YOUR-USERNAME/aqmss2.git`
4. Choose a location on your computer
5. Click "Create Project"

---

3 Task 3: Edit the README File

Your README is the "front page" of your repository. Edit it to include:

- Your name
- A brief description (e.g., "Assignments for AQMSS II, Spring 2026")
- Optionally, a list of what will be in the repository

3.1 Option A: On the Web

1. Click on `README.md` in your repository
2. Click the pencil icon (edit) in the top-right of the file view
3. Make your changes using Markdown syntax
4. Scroll down and click "Commit changes"
5. Add a commit message like "Update README with my info"

3.2 Option B: Command Line

```
# Edit README.md with any text editor, then:
git add README.md
git commit -m "Update README with my info"
git push
```

3.3 Option C: RStudio

1. Edit the `README.md` file in the Files pane
2. Go to the Git pane (usually top-right)
3. Check the box next to `README.md` to stage it
4. Click "Commit"
5. Write a commit message and click "Commit"
6. Click "Push" to upload to GitHub

---

4 Task 4: Create a Folder and R File

Create a folder called `assignment1` and add your first R file.

4.1 Option A: On the Web

1. Click "Add file" → "Create new file"
2. In the filename box, type: `assignment1/assign1.R`
   - This creates the folder and file at once
3. Add the following content:

```
# Assignment 1
# AQMSS II, Spring 2026
# [Your Name]

# This file will contain my solutions for Assignment 1
print("Hello, Git!")
```

4. Scroll down and commit with message "Add assign1.R"

4.2 Option B: Command Line

```
# Create the folder
mkdir assignment1

# Create the R file (use any text editor)
# Then stage, commit, and push:
git add assignment1/assign1.R
git commit -m "Add assign1.R"
git push
```

4.3 Option C: RStudio

1. Create a new folder `assignment1` in the Files pane
2. Create a new R script: File → New File → R Script
3. Add the header content and save as `assignment1/assign1.R`
4. In the Git pane, stage the new file, commit, and push

---

5 Task 5: Load a Dataset and Make a Plot

In this task, you will download a public dataset, add it to your repository, load it in R, and create a simple plot.

5.1 Download the data

We will use the Gapminder dataset, which contains country-level data on life expectancy, GDP per capita, and population from 1952 to 2007.

1. Create a `data/` folder in your repository
2. In R, run the following to download the data and save it as a CSV file:

   ```
   install.packages("gapminder")
   library(gapminder)
   write.csv(gapminder, "data/gapminder.csv", row.names = FALSE)
   ```

3. Verify that the file `data/gapminder.csv` exists in your repository folder

5.2 Load the data and make a plot

Create a new R script called `assignment1/ass1_plot.R` with the following steps:

1. Load the data from the CSV file:

   ```
   library(ggplot2)
   gapminder <- read.csv("data/gapminder.csv")
   ```

2. Explore the data briefly:

   ```
   head(gapminder)
   str(gapminder)
   ```

3. Create a plot showing how life expectancy has changed over time for a few countries of your choice. For example:

   ```
   countries <- c("Spain", "France", "Germany", "United Kingdom")
   df <- gapminder[gapminder$country %in% countries, ]

   ggplot(df, aes(x = year, y = lifeExp, color = country)) +
     geom_line() +
     geom_point() +
     labs(x = "Year", y = "Life expectancy",
          title = "Life expectancy over time") +
     theme_minimal()

   ggsave("assignment1/ass1_plot.png", width = 7, height = 5)
   ```

4. You can choose different countries or a different variable (e.g., `gdpPercap` or `pop`)

5.3 Commit everything

Stage and commit the data file, the R script, and the plot:

```
git add data/gapminder.csv assignment1/ass1_plot.R assignment1/ass1_plot.png
git commit -m "Add gapminder data and first plot"
git push
```

6 Task 6: View Your Commit History

Check that your commits were recorded properly.

6.1 Option A: On the Web

1. Go to your repository page on GitHub
2. Click on “Commits” (or the clock icon with a number)
3. You should see your commits listed with messages and timestamps

6.2 Option B: Command Line

```
git log --oneline
```

This shows a compact list of your commits.

6.3 Option C: RStudio

1. In the Git pane, click “History” (clock icon)
2. Browse through your commits

Take a screenshot of your commit history showing at least 3–4 commits.

7 Submission

1. Send me an email with the URL of your GitHub repository (e.g., `https://github.com/username/aqmss2`)
2. Requirements:

- Your repository is **public**
- It contains a README with your name
- It has a `data/` folder with the Gapminder CSV file
- It has a `assignment1/` folder with your R files and a saved plot