SCS2211 - LABORATARY 02

PRACTICAL - 11-A

Instructions

- **Do the tasks given in the practical sheet and take screenshots of the output**
- **Create a report using the screenshots.**
- **Report must be in PDF format.**
- **Report name should be <Index number>.pdf (Eg: 2000000.pdf)**
- **Any form of plagiarism or collusion is not allowed**

Connecting RStudio with GitHub and Using Jupyter Notebook

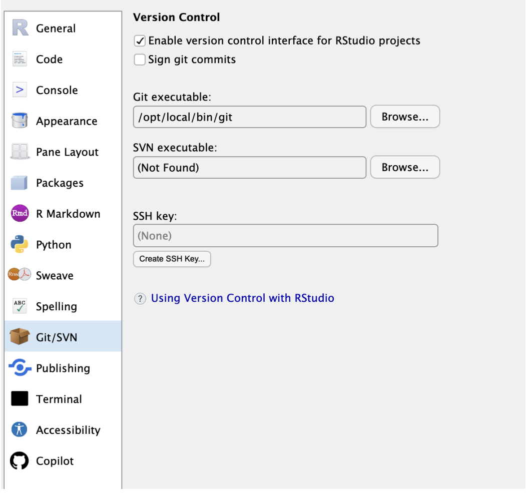
1. Connecting RStudio with GitHub

1.1 Prerequisites

- Install RStudio.
- Install Git.
- Create a GitHub account (GitHub Signup).
- Optional: Install GitHub Desktop for a GUI experience.

1.2 Setting up Git in RStudio

- Open RStudio and go to Tools > Global Options > Git/SVN.



Version Control

☑ Enable version control interface for RStudio projects

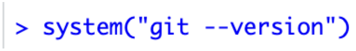
□ Sign git commits

Git executable: /opt/local/bin/git

SVN executable: (Not Found)

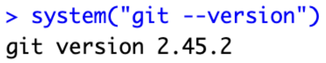
SSH key: (None)

- Check Enable version control interface for RStudio projects.
- Click Browse to locate the Git executable
- Test Git installation:
    - Open RStudio Console and type:



```
> system("git --version")
```

    - It should return your Git version.



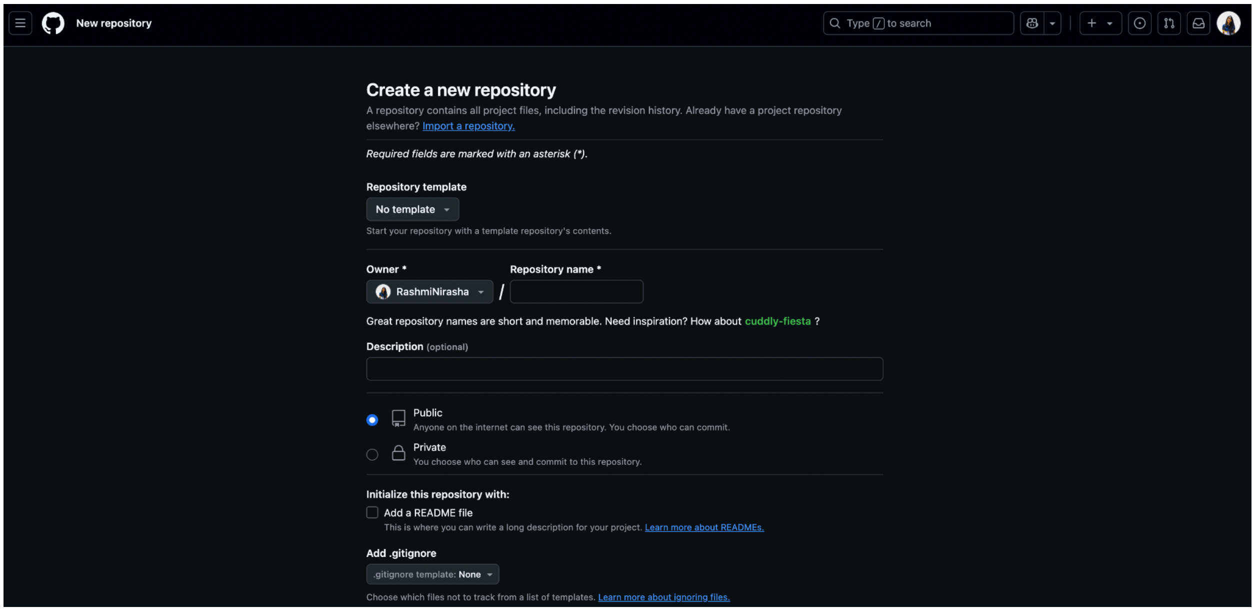
```
> system("git --version")
git version 2.45.2
```

1.3 Configuring Git

- Run the following in RStudio Console to set your GitHub identity:
    - system('git config --global user.name "Your Name"')
    - system('git config --global user.email "youremail@example.com"')

1.4 Connecting RStudio Project to GitHub

- Create a repository on GitHub:
    - Go to GitHub and click New Repository.
    - Name your repository and click Create Repository.
        - (Ex: Indexno_Rpractical)



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner * RashmiNirasha ▾ / **Repository name ***

Great repository names are short and memorable. Need inspiration? How about cuddly-fiesta ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

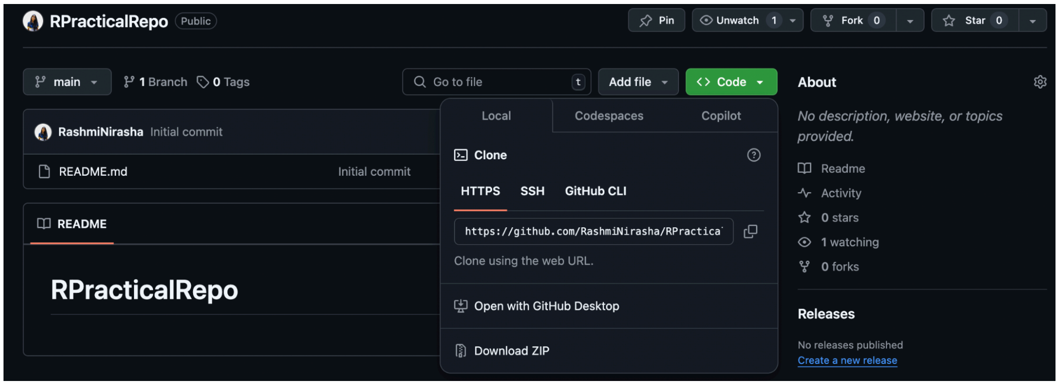
□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. Learn more about ignoring files.

- Clone the repository to RStudio:
  - In GitHub, copy the repository URL (HTTPS).

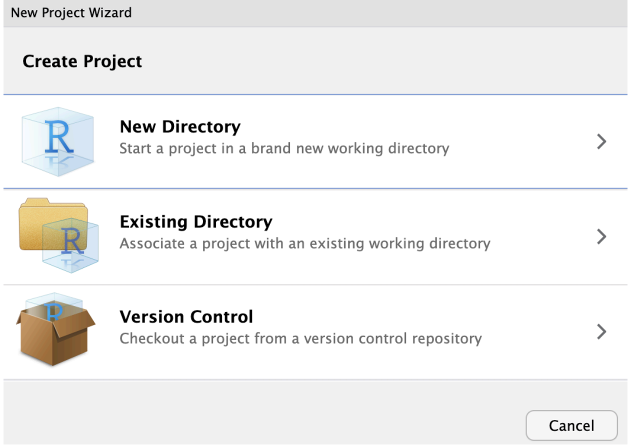


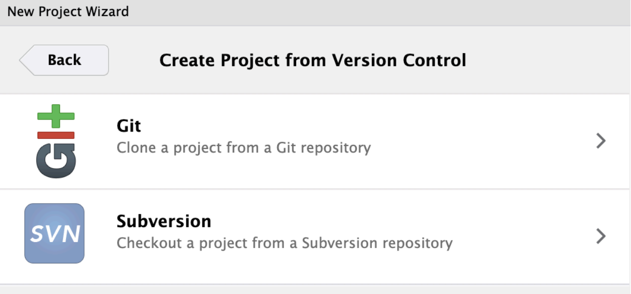
https://github.com/RashmiNirasha/RPractica

Clone using the web URL.

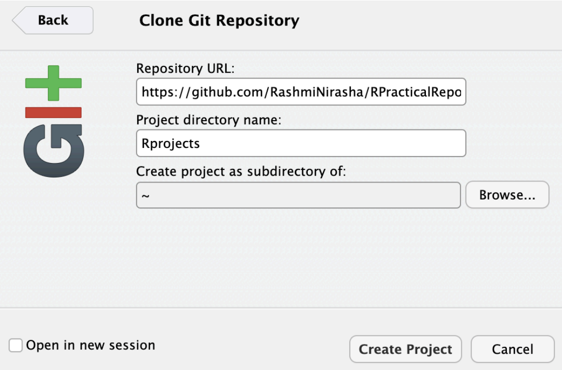
No description, website, or topics provided.

- In **RStudio**, go to File > New Project > Version Control > Git.


New Project Wizard
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel


New Project Wizard
Back
Create Project from Version Control
Git
Clone a project from a Git repository
Subversion
Checkout a project from a Subversion repository

  - Paste the repository URL and choose a local folder.
  - Click **create project**.



Repository URL: https://github.com/RashmiNirasha/RPracticalRepo

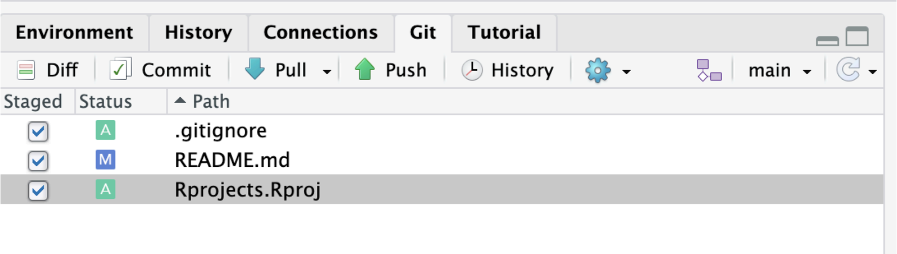
Project directory name: Rprojects

Create project as subdirectory of: ~

- Optional : Initialize Git in an existing RStudio project:
  - Go to Tools > Version Control > Project Setup.
  - Select Git.

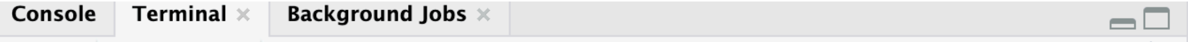
1.5 Common Git Commands in RStudio

- Stage Changes: Check files in the Git pane and click Stage.

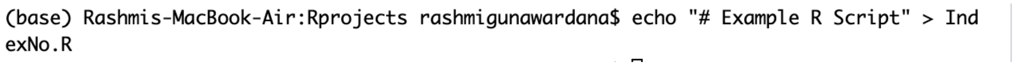


Staged	Status	Path
☑	A	.gitignore
☑	M	README.md
☑	A	Rprojects.Rproj

- Open the Terminal in RStudio:
  - Go to the **Tools** menu and select **Terminal** > **New Terminal**.
  - Alternatively, click on the **Terminal** tab at the bottom of RStudio.

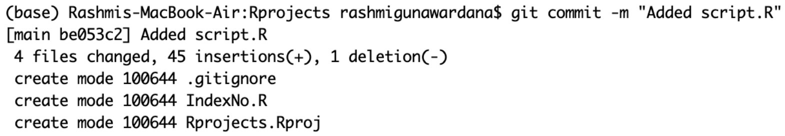


- Create new file using,
  - Enter command : `echo “#example R Script”> IndexNo.R`



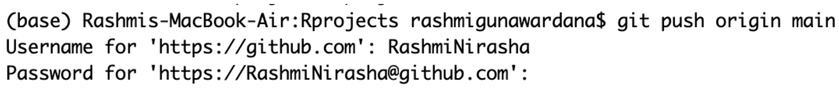
```
(base) Rashmis-MacBook-Air:Rprojects rashmigunawardana$ echo "# Example R Script" > Ind
exNo.R
```

  - **Run Git Commands in the Terminal**: Enter your Git commands in the terminal. For example:
    - ➔ `git add IndexNo.R / or git add .`
    - ➔ `git commit -m "Added script.R"`
    - ➔ `git push origin main`



```
(base) Rashmis-MacBook-Air:Rprojects rashmigunawardana$ git commit -m "Added script.R"
[main be053c2] Added script.R
 4 files changed, 45 insertions(+), 1 deletion(-)
 create mode 100644 .gitignore
 create mode 100644 IndexNo.R
 create mode 100644 Rprojects.Rproj
```

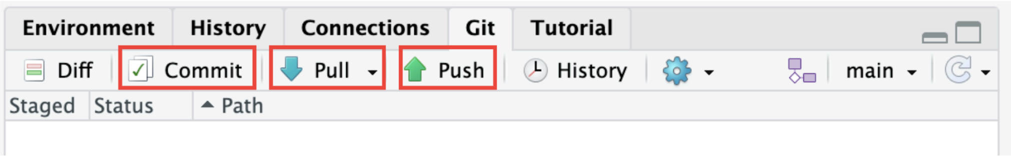
- ○ **When pushing, you might have to enter your username and password again**:



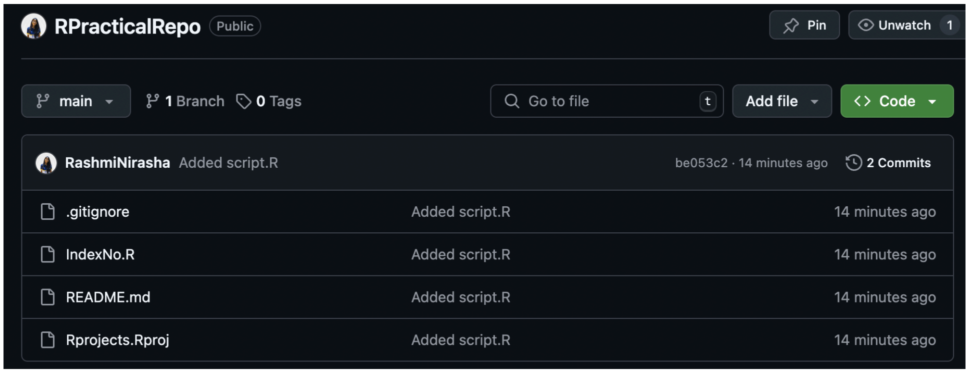
```
(base) Rashmis-MacBook-Air:Rprojects rashmigunawardana$ git push origin main
Username for 'https://github.com': RashmiNirasha
Password for 'https://RashmiNirasha@github.com':
```

- Optional: If You Need to Use R Console:
  - ○ If you want to run Git commands from the R console, you can use the `system()` function in R:
    - ➔ `system("git add script.R")`
    - ➔ `system("git commit -m 'Added script.R'")`
    - ➔ `system("git push origin main")`

- Optional: If You Need to Use Git Pane in RStudio: (**Recomended** option if you have trouble navigation terminal)
  - ○ Go to the **Git Pane** in RStudio.
  - ○ Stage files by checking the boxes next to the files.
  - ○ Add a commit message and click **Commit**.
  - ○ Use the **Push** and **Pull** buttons for remote operations.



- Check GitHub repository if all the filed have been pushed.
  - ○ How it should look.



RashmiNirasha Added script.R be053c2 · 14 minutes ago 2 Commits

.gitignore	Added script.R	14 minutes ago
IndexNo.R	Added script.R	14 minutes ago
README.md	Added script.R	14 minutes ago
Rprojects.Rproj	Added script.R	14 minutes ago

2. Using Jupyter Notebook for R

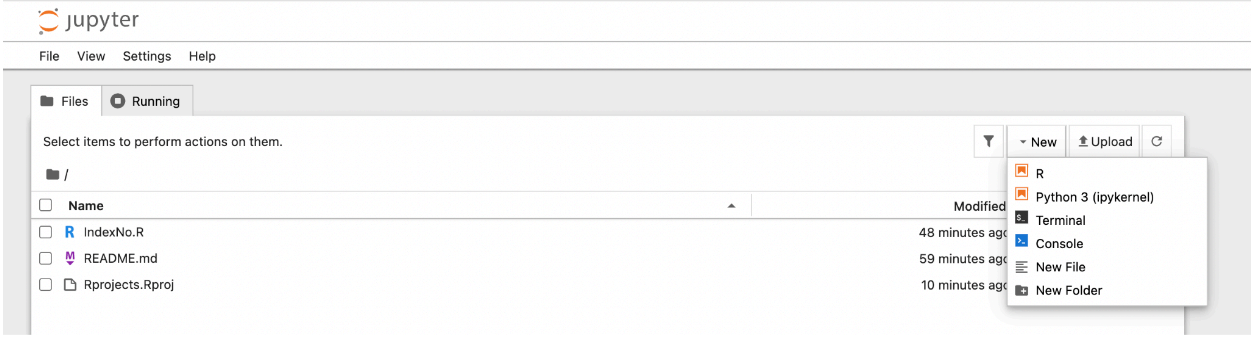
2.1 Installing Jupyter Notebook

- Install Python and Jupyter:
  - These commands are executed in the **R console** (inside RStudio or any R environment):
    - `install.packages("reticulate")`
    - `reticulate::install_miniconda()`
    - `reticulate::conda_install("r-reticulate", "jupyter")`

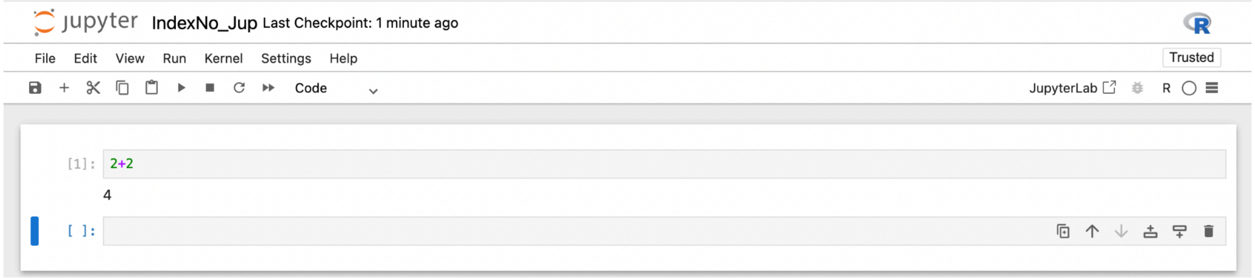
- Add R kernel to Jupyter:
  - `install.packages("IRkernel")`
  - `IRkernel::installspec()`

2.2 Launching Jupyter Notebook

- Start Jupyter Notebook:
  - `jupyter notebook`
- In the Jupyter interface, select New > R to create an R notebook.



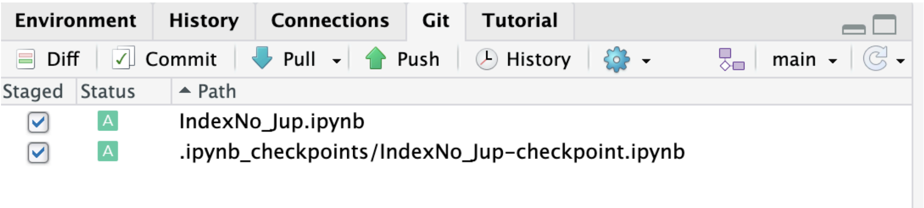
Name	Modified
IndexNo.R	48 minutes ago
README.md	59 minutes ago
Rprojects.Rproj	10 minutes ago


jupyter IndexNo_Jup Last Checkpoint: 1 minute ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
R

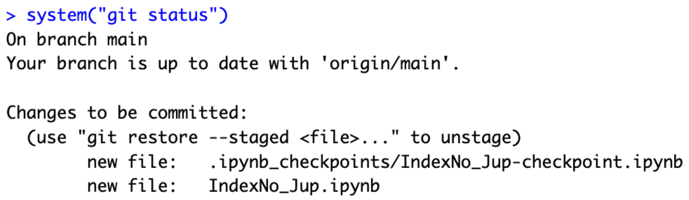
```
[1]: 2+2
     4
[ ]:
```

2.3 Syncing Jupyter Notebooks with GitHub

- Save your notebook (.ipynb) file in the RStudio Git project directory.

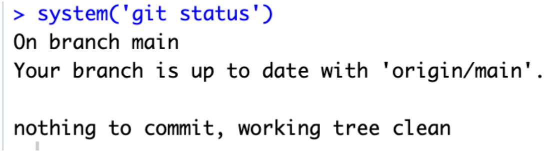


Staged	Status	Path
☑	A	IndexNo_Jup.ipynb
☑	A	.ipynb_checkpoints/IndexNo_Jup-checkpoint.ipynb



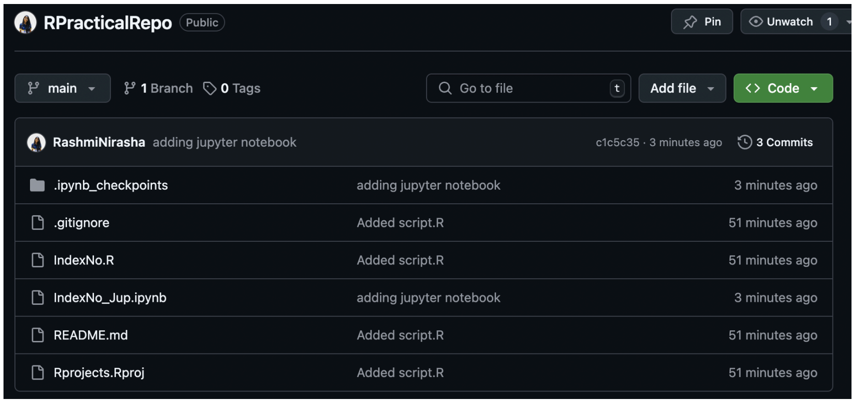
```
> system("git status")
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .ipynb_checkpoints/IndexNo_Jup-checkpoint.ipynb
        new file:   IndexNo_Jup.ipynb
```

- Use the same Git commands (commit, push, pull) to manage version control.
  - system("git add .")
  - system('git commit -m "adding jupyter notebook"')
  - system("git push origin main")



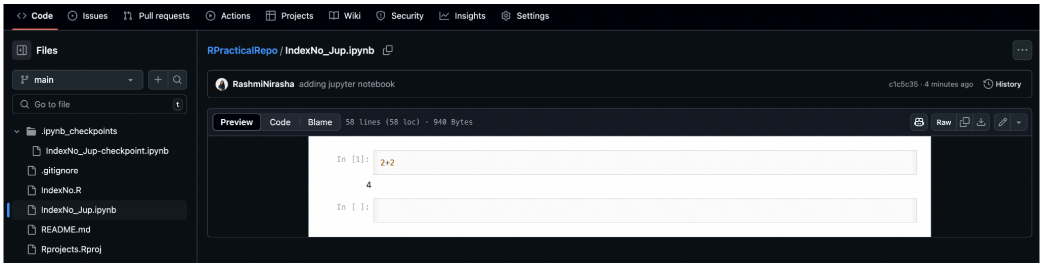
```
> system('git status')
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

- Check GitHub repository. The Jupitor Notebook should be pushed to it.



RashmiNirasha adding jupyter notebook c1c5c35 · 3 minutes ago 3 Commits

.ipynb_checkpoints	adding jupyter notebook	3 minutes ago
.gitignore	Added script.R	51 minutes ago
IndexNo.R	Added script.R	51 minutes ago
IndexNo_Jup.ipynb	adding jupyter notebook	3 minutes ago
README.md	Added script.R	51 minutes ago
Rprojects.Rproj	Added script.R	51 minutes ago



RPracticalRepo / IndexNo_Jup.ipynb

RashmiNirasha adding jupyter notebook c1c5c35 · 4 minutes ago

58 lines (58 loc) · 940 Bytes

```
In [1]: 2+2

        4

In [ ]:
```

3. Practical Tasks

Task 1: Create and Push an R Script to GitHub Using Rstudio.

a. Create a new R script in RStudio.
b. Write a simple function:

```
hello <- function() {

  print("Hello, GitHub!")

}
```

c. Stage, commit, and push the script to GitHub.

Task 2: Use Jupyter Notebook for Data Visualization

1. Load the ggplot2 library: install.packages("ggplot2")

```
library(ggplot2)

ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point()
```

2. Save the notebook, then commit and push to GitHub.

Example Git Repository : https://github.com/RashmiNirasha/RPracticalRepo