Zehui Yin

First-year PhD Student in Geography at SEES

- **Email:** yinz39@mcmaster.ca

  - Please use your McMaster email address.

  - Please put the course code *ENVSOCTY 4GA3* in the subject line.

  - Please include your name and student number in the body of the email.

  - I will try to reply within 24 hours (please expect longer delays during weekends/holidays).

- **Personal Website:** zehuiyin.github.io

- **Research Interests:** Spatial Analysis, Transportation, Travel Behaviour, Public Transit, Shared Mobility

Agenda for today

- Introduction to basic concepts for coding in R

- Getting a flavor of R syntax and style

- Setting up R and reproducible environment on your personal computer

R and RStudio

- R is a free and open-source programming language for statistical computing and graphics.

- RStudio is an integrated development environment (IDE) for coding in R.

  - An IDE is a set of tools that helps you code.

- We use RStudio to write our R codes.

2025 Zehui Yin

R packages

- R packages are the fundamental units of reproducible R code.

- They can include functions, data, or both, along with documentation.

- Think of them as plug-ins that enhance the functionality of existing software.

- For example, web browser extensions like ad blockers add additional features that the original browser doesn't have.

Reproducible environment

- An environment is the system where a program is run, including hardware and software such as operating system dependencies, programming language, packages, their configuration, and versions.

- Just as running 1000 meters affects individuals differently, running code on different computers or with different package versions can produce varied results.

- A reproducible environment ensures that everyone gets the same result by keeping the environment consistent.

renv package
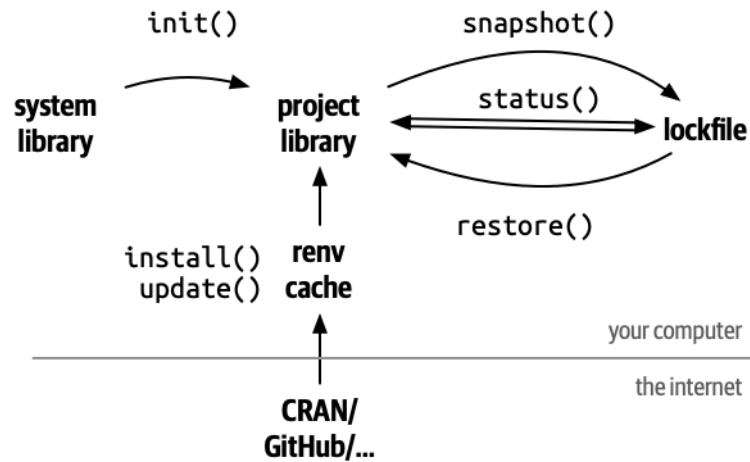
- **renv** is an R package that helps create reproducible environments for R projects.

- It records the R version and all R packages along with their versions in a lockfile.

  - A lockfile is a text file that stores all the environment information.

Code hosting and Github
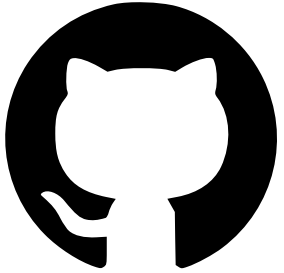
- Code hosting involves storing code online to facilitate sharing, management, and collaboration with others.

- One of the most popular code hosting platforms is GitHub (owned by Microsoft).

- Both the textbook and the companion R package used in this course are hosted on GitHub.

- GitHub is like a cloud drive (similar to OneDrive or Dropbox) but specialized for storing code, including R scripts.

R Markdown vs. R
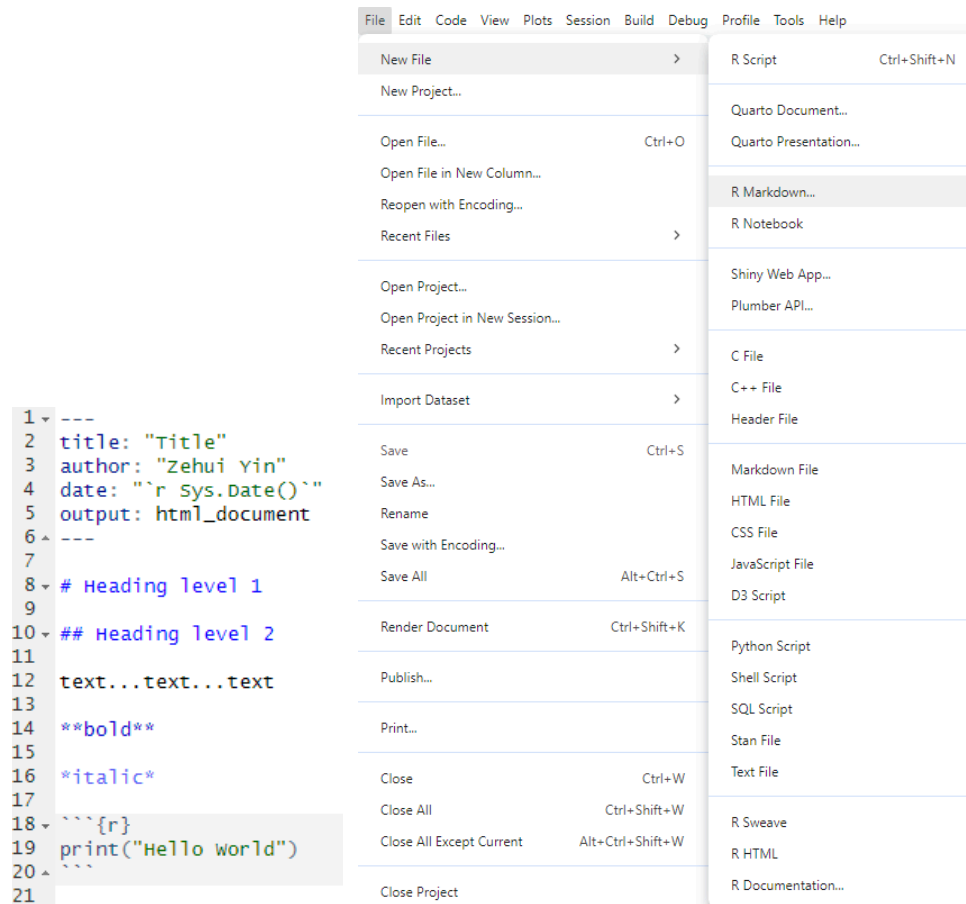
- R Markdown is a file format that combines R code, its results, and accompanying text.

- It uses the file extension `.Rmd`.

- R code chunks are created using ```` ```{r} … ``` ````.

- You can start by creating an `Rmd` file on the lab computer.

R basics: arthmatic operations

You can start by trying out R on the lab computer. Later, we'll set it up on your personal computer.

R can be used as a calculator, using intuitive symbols for these operations:

```
1  1 + 5
```

[1] 6

```
1  8 - 3
```

[1] 5

```
1  3 * 4
```

[1] 12

```
1  9 / 3
```

[1] 3

R basics: assigning values

One of the cornerstones of programming languages is assignment. You can assign a value/object to a name using `<-` (suggested R style) or `=` ("Python" style).

```r
1  a <- 1
2  b <- 3
3  a + b
```

[1] 4

```r
1  c = 7
2  d = 5
3  c * d
```

[1] 35

R basics: built-in functions

R comes with many built-in functions. The calling syntax is `function(parameter1, parameter2, ...)`. Additionally, with extra R packages, there are even more functions you can use.

```r
1 values <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2 sum(values)
```

```
[1] 55
```

```r
1 mean(values)
```

```
[1] 5.5
```

```r
1 library(MASS)
2 # integrate the sin function from 0 to pi.
3 area(sin, 0, pi)
```

```
[1] 2
```

R basics: flow control

Flow control is an important component of any programming language. In R, the `if-else` statement and loops work as follows:

```r
x <- 6
if (x > 5) {
  print("Greater than 5")
} else {
  print("Less or equal to 5")
}
```

```
[1] "Greater than 5"
```

```r
for (i in 1:3) {
  print(i)
}
```

```
[1] 1
[1] 2
[1] 3
```

R basics: custom functions

To define your own function in R, you can use the following syntax. Note that the last line of code is automatically returned by R, though a "Python" style return statement is also valid in R.

```r
1  add <- function(a, b) {
2    a + b
3  }
4
5  add(1, 4)
```

```
[1] 5
```

```r
1  add <- function(a, b) {
2    return(a + b)
3  }
4
5  add(1, 4)
```

```
[1] 5
```

Download R version 4.4.2

- https://mirror.csclub.uwaterloo.ca/CRAN/

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2024-10-31, Pile of Leaves) R-4.4.2.tar.gz, read what's new in the latest version.
- The CRAN directory src/base-prerelease contains R alpha, beta, and rc releases as daily snapshots in time periods before a planned release.
- Between releases, the same directory src/base-prerelease contains snapshots of current patched and development versions.
  Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Alternatively, daily snapshots are available here.
- Source code of older versions of R is available here.
- Contributed extension packages.

CRAN
Mirrors
What's new?
Search
CRAN Team

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Task Views
Other

Documentation
Manuals
FAQs
Contributed

Donations
Donate

Download RStudio

- https://posit.co/download/rstudio-desktop/

Restoring the environment

Download the `Applied-Spatial-Statistics` zip file from Avenue and unzip it. You should then have a folder with the following structure:

```
Applied-Spatial-Statistics/
├── renv/
├── .gitignore
├── .Rprofile
├── Applied-Spatial-Statistics.Rproj
├── README.md
├── README.Rmd
└── renv.lock
```

- Double-click the `Applied-Spatial-Statistics.Rproj` file to open the R project.

Restoring the environment
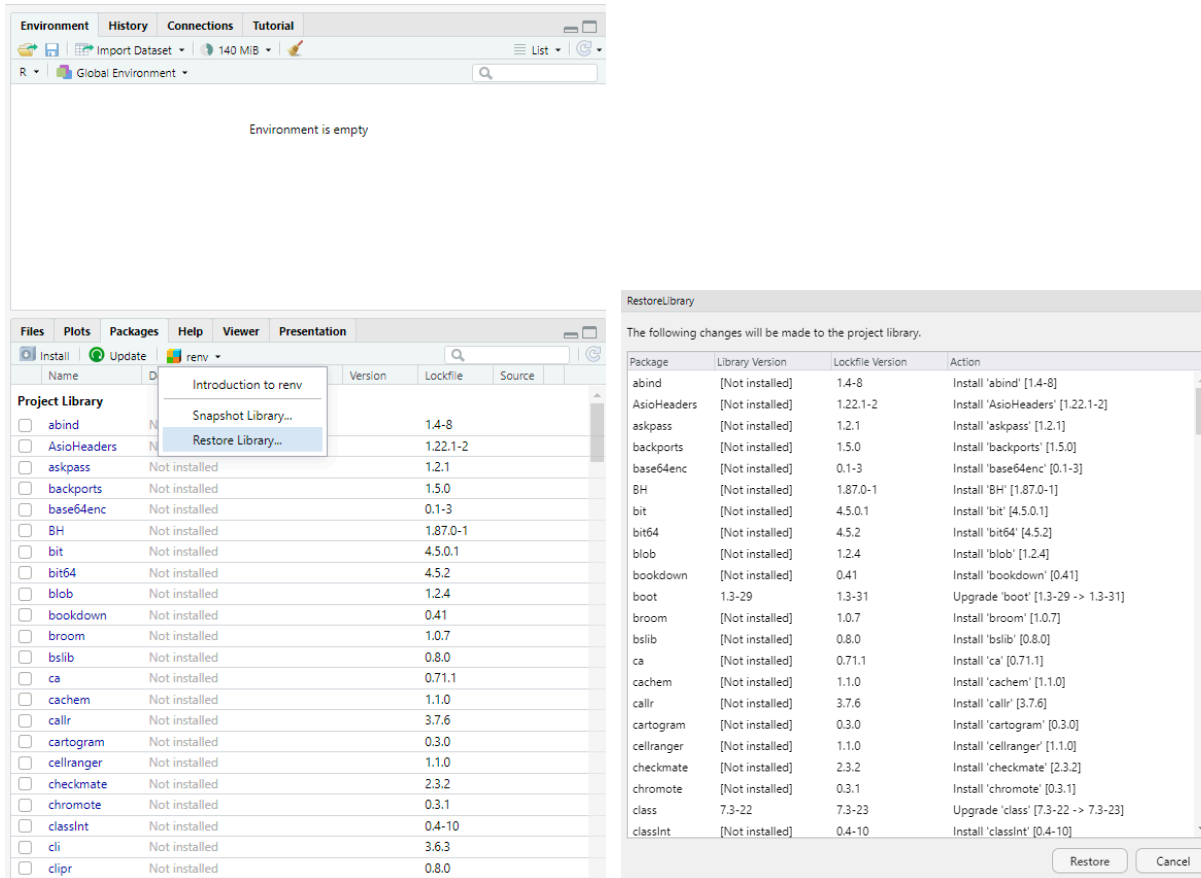
- Navigate to the bottom right panel.

- Click the Packages tab, then the renv button, and finally select the Restore Library option.

- Click the Restore button in the pop-up panel.

Install $\LaTeX$

$\LaTeX$ is a high-quality typesetting system. While it may seem as a language, understanding it isn't necessary for our purposes. We will use it to export `Rmd` files with results into PDF files.

If you already use $\LaTeX$ and have it installed through `MiKTeX` or `TeX Live`, you can skip this step.

If you are unfamiliar with $\LaTeX$ and don't have it installed yet, simply run the following R code to install it:

```r
tinytex::install_tinytex()
```

Lab slides

- You can access all the lab slides by scanning the QR code or by visiting the URL directly.

- zehuiyin.github.io/ENVSOCTY4GA3

2025 Zehui Yin

References

- https://r-pkgs.org/

- https://book.the-turing-way.org/reproducible-research/renv.html

- https://rstudio.github.io/renv/articles/renv.html