Video 3: Getting Set up in R Studio

Stats 102A

Miles Chen, PhD

Programming: Set Up and Shortcuts

Tip: create a separate folder for each hw assignment

- Tip: Create a separate folder for each homework assignment.
- Tip: do not use spaces in folder names, use underscores or dashes instead
- Put all related files into the same folder

For HW1, create a folder on your computer called “HW1”.

Download the three files for the assignment (`102A_hw_01_instructions.pdf`, `102A_hw_01_output.Rmd`, and `month_names.txt`) and put them into the same HW1 folder.

Do not work directly from your downloads folder.

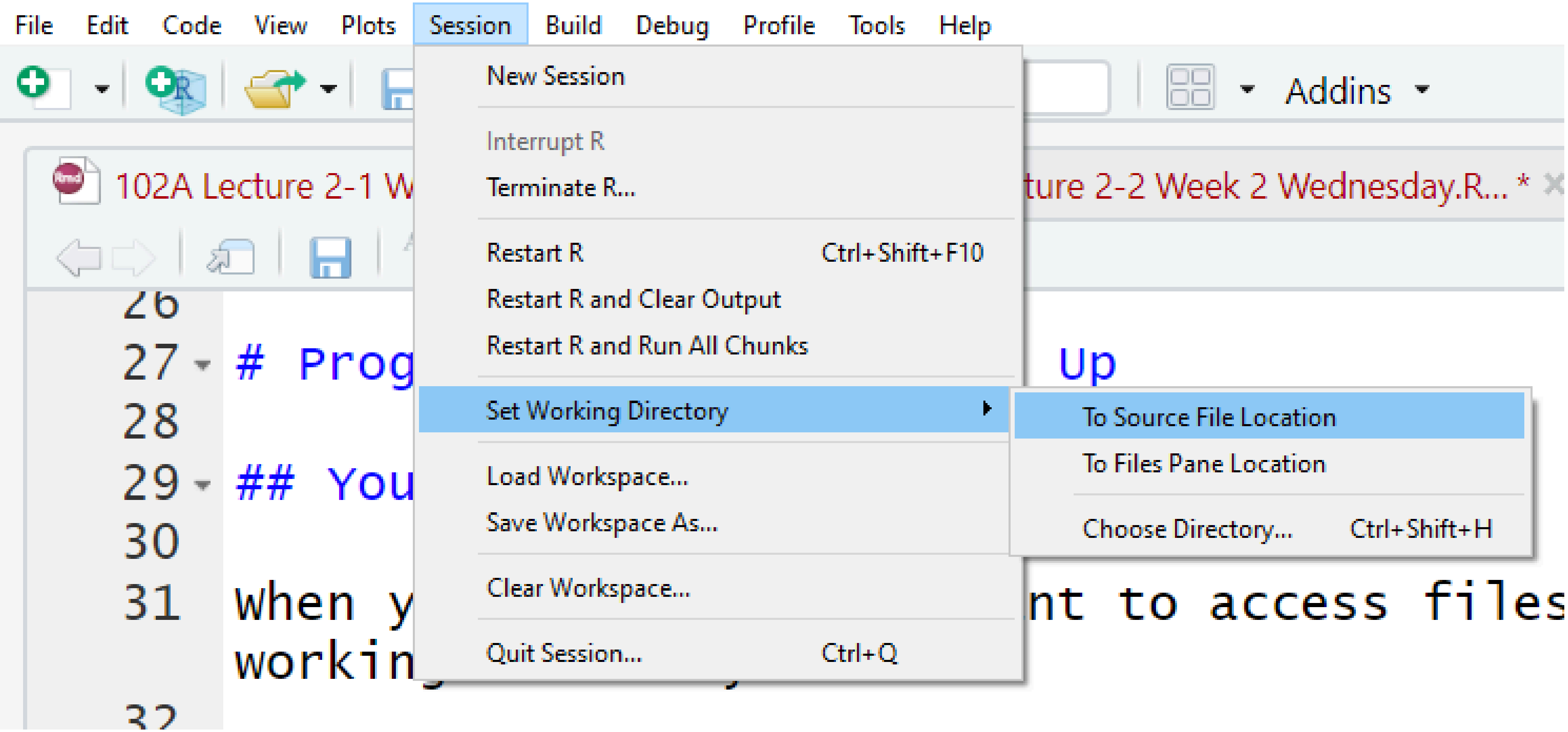
Navigate to your HW1 folder and open the Rmd file from there.

You will need to create a new file in the folder: `102a_hw_01_script_Joe_Bruin.R`

Your Working Directory

To make sure R can see the appropriate files, you need to set the working directory.

I use the menu commands to set my working directory: **Session** > **Set Working Directory** > **To Source File Location**.



Your Working Directory

At some point, you might encounter the following message or something like it:

```
Warning: cannot open file 'month_names.txt': No such file or directory
Error in file(file, "rt") : cannot open the connection
```

This means R is searching for a file “month_names.txt” in the working directory and it does not see it.

Either:

- the file is saved somewhere else on your computer (not in the same directory as the homework Rmd file)
- your working directory is set to the wrong folder location

Fix the problem and try again.

Checking and Clearing the environment

As you work, your environment, or workspace, will fill with objects. You can check the environment pane in R Studio to see what exists. You may also get a list of the contents with `ls()`. You can remove individual elements with `rm()`. `rm(x)` will remove the variable `x`.

You should get in the habit of clearing out your environment whenever you start working on a new file or project. Keep in mind that whenever you render/knit a markdown file, R begins with an empty environment.

You can clear the environment this by clicking the Broom Icon in the environment pane.

Checking and Clearing the environment

Another quick command to clear the workspace is to type:

```r
rm(list = ls())

# ls() provides a vector of all the object names in the environment.
# rm() goes through and removes all of those names
```

Problem: “My code runs in R, but when I try to knit, I get an error.”

“My code runs in R, but when I try to knit, I get an error.”

This is one of the most common errors students encounter. There are many possible reason this happens, but one of the most common scenarios is this:

1. Student creates an object in R.
2. Student modifies or deletes the line of code that creates the original object.
3. The object still exists in the Environment, so new lines of code work
4. When it comes time to knit, the line to create said object doesn’t exist or creates a different version of the object. R is not able to execute the other lines of code.

Resolving the Issue

1. Save your files: Rmd file, R script, etc.
2. Clear your environment.
3. Starting at the beginning of your Rmd file, run each code chunk one at a time.
    - Most likely, you will encounter an error message.
    - See which line caused the error. This will provide some insight into which object is missing or needs to be fixed.
4. If you can clear the environment and run through every single code chunk with no errors, then your file should knit without problems.

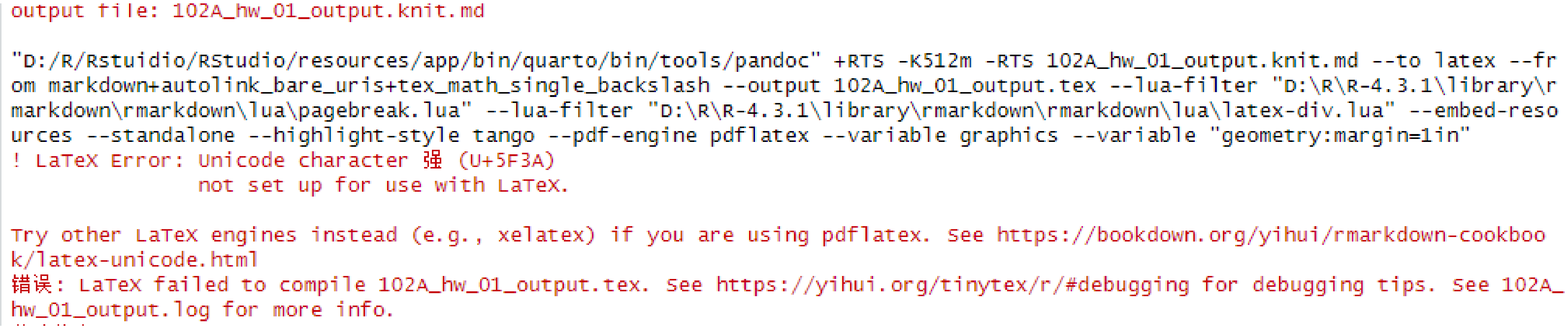
Resolving Really Stubborn Issues

If the steps in the previous slide do not work, you can attempt the following to try to resolve really stubborn issues. This is much more work, so I only recommend it when absolutely necessary.

1. Save your files: the current Rmd file that you can’t knit, R script, etc. I’ll refer to the rmd file with errors as ProblemRmd.
2. Clear your environment.
3. Create a new Rmd file and save it to the same Homework folder. I will call the FreshRmd.
4. Knit the FreshRmd file. Because it is brand new, it should knit with no problems. Delete any unnecessary stuff from FreshRmd.
5. Copy one paragraph or one code chunk from ProblemRmd to FreshRmd. After copying, knit FreshRmd. It should knit.
6. Continue copying one paragraph or one code chunk at a time from ProblemRmd to FreshRmd and knit FreshRmd each time.
7. When you copy something from ProblemRmd to FreshRmd and FreshRmd encounters and error when knitting, you know the problem exists in the most recently copied chunk. You can event try deleting that chunk and knitting to verify that it is indeed that piece.
8. Resolve the issue.

Tip: Set your language to English

If your language is not English, error messages will contain foreign characters. LaTeX is not able to handle these characters and will result in an error like the following.



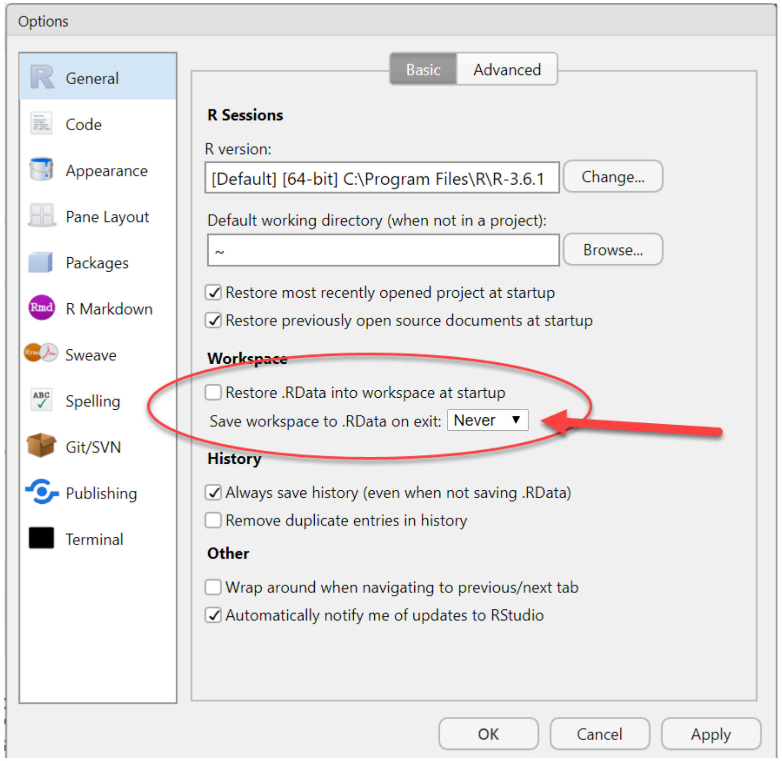
```
output file: 102A_hw_01_output.knit.md

"D:/R/Rstuidio/RStudio/resources/app/bin/quarto/bin/tools/pandoc" +RTS -K512m -RTS 102A_hw_01_output.knit.md --to latex --fr
om markdown+autolink_bare_uris+tex_math_single_backslash --output 102A_hw_01_output.tex --lua-filter "D:\R\R-4.3.1\library\r
markdown\rmarkdown\lua\pagebreak.lua" --lua-filter "D:\R\R-4.3.1\library\rmarkdown\rmarkdown\lua\latex-div.lua" --embed-reso
urces --standalone --highlight-style tango --pdf-engine pdflatex --variable graphics --variable "geometry:margin=1in"
! LaTeX Error: Unicode character 强 (U+5F3A)
               not set up for use with LaTeX.

Try other LaTeX engines instead (e.g., xelatex) if you are using pdflatex. See https://bookdown.org/yihui/rmarkdown-cookboo
k/latex-unicode.html
错误: LaTeX failed to compile 102A_hw_01_output.tex. See https://yihui.org/tinytex/r/#debugging for debugging tips. See 102A_
hw_01_output.log for more info.
```

How to set your language to English:

- For Windows: https://stackoverflow.com/a/26674852
- For Mac: https://stackoverflow.com/a/28353147

Tip: Don’t save your workspace when you exit RStudio

Tip: Change the following option: Tools > Global Options … On the General tab, uncheck “Restore .RData into workspace at startup” and change the setting ‘Save Worskspace to .Rdata on Exit’ to ‘Never’



R Sessions

R version:
[Default] [64-bit] C:\Program Files\R\R-3.6.1

Default working directory (when not in a project):
~

☑ Restore most recently opened project at startup
☑ Restore previously open source documents at startup

Workspace

□ Restore .RData into workspace at startup
Save workspace to .RData on exit: Never ▾

History

☑ Always save history (even when not saving .RData)
□ Remove duplicate entries in history

Other

□ Wrap around when navigating to previous/next tab
☑ Automatically notify me of updates to RStudio

Tip: Don’t save your workspace when you exit RStudio

Not saving on exit forces you to have good save habits.

It also prevents issues where your script receives old and incorrect values when it refers to common names such as `x` or `results`. If you don’t save your workspace on exit, those old values will simply not exist when you start RStudio the next time.

If you ever want to save your workspace, you can do so explicitly and you will have the workspace available as a labeled file that you can transport.

Toggle Comments in your code

You already know you can make a comment in your code with #

Commenting can be useful method to quickly disable lines of code.

In R Studio, the keyboard shortcut **Ctrl/Cmd + Shift + C** will toggle a line to be a comment. You can toggle entire blocks of code this way as well.

Writing Scripts in RStudio

Quickly create a new R Script file, use the keyboard shortcut **Ctrl/Cmd + Shift + N**

This will create a blank plain text document (with extension .R)

Each line or a block of highlighted text can then be executed with **Ctrl/Cmd + Enter**.

If you want to run the entire script, you can source the document: `source("script.R")`

Keyboard shortcut to source the entire script: **Ctrl/Cmd + Shift + S**

Other useful shortcuts:

Clear the console: **Ctrl/Cmd + L**

Restart your R Session: **Ctrl/Cmd + Shift + F10**

Knit your current Rmd File: **Ctrl/Cmd + Shift + K**