Memorial Sloan Kettering Cancer Center

Reproducible Workflows in R using {renv} and Project Templates

May 2, 2022

Caroline Kostrzewa, Epi-Bio
Daniel D. Sjoberg , Epi-Bio
Jessica Lavery , Epi-Bio
Karissa Whiting , Epi-Bio
Shannon Pileggi, PCCTC , Epi-Bio
Karolyn Ismay, Strategy & Innovation

Agenda

Workstation Setup

- Where to install R/RStudio
- Personal Access Token (PAT)
- Folder Organization

New Project Setup

- GitHub
- bstfun::create_bst_project()
- Symbolic Links

{renv}

- init(), snapshot(), restore()

Demo

Memorial Sloan Kettering
Cancer Center

Workstation Setup

Where to Install R/RStudio

- Depends on if you have admin rights on your computer

- If **YES** → install R/RStudio on your C: drive (C:\Program Files)
- If **NO** → install to the default location (on your C: drive)
  - should be something like C:\Users\username\AppData\Local\Programs
  - **If you want...**make a subfolder in your OneDrive called something like "Programs" – right click on this folder and *prevent it from syncing!*

Setting Up for Compiling Packages

- For Windows users: Download RTools
  - More Info: https://r-pkgs.org/setup.html#windows
  - There's a new version for R 4.2 – older versions will not work!
- Get a Personal Access Token (PAT)
  - Sometimes needed when restoring packages
  - Sometimes computer needs to be set up for development for this to happen – use `devtools::has_devel()` to check if in developer mode
  - Instructions for how to get a PAT: https://github.mskcc.org/pages/datadojo/mskRutils/articles/git_config.html#pat
  - Resource with even more details: https://happygitwithr.com/https-pat.html

Folder Organization

- For GitHub projects:
  - One Drive
    - GitHub
      - <PI name> <short description>
      - <PI name> <short description>

These are the local repositories cloned from GitHub

- For non-GitHub projects:
  - One Drive
    - Analytic Projects
      - <PI name> <short description>
      - <PI name> <short description>

Workstation Setup

- Where to install R/RStudio
- Personal Access Token (PAT)
- Folder Organization

Questions?

Memorial Sloan Kettering Cancer Center

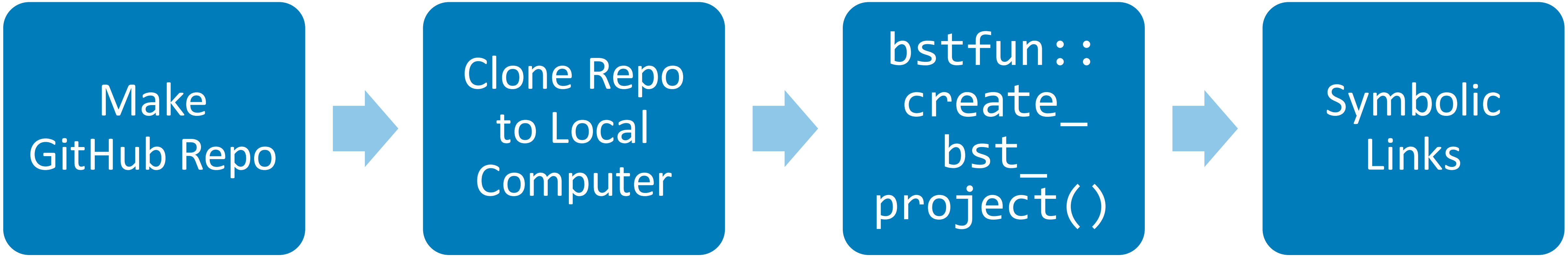
New Project Setup

“Happy ~~families~~ projects are all alike; every unhappy ~~family~~ project is unhappy in its own way.”

~~-Leo Tolstoy~~

Daniel Sjoberg

Recommended Workflow

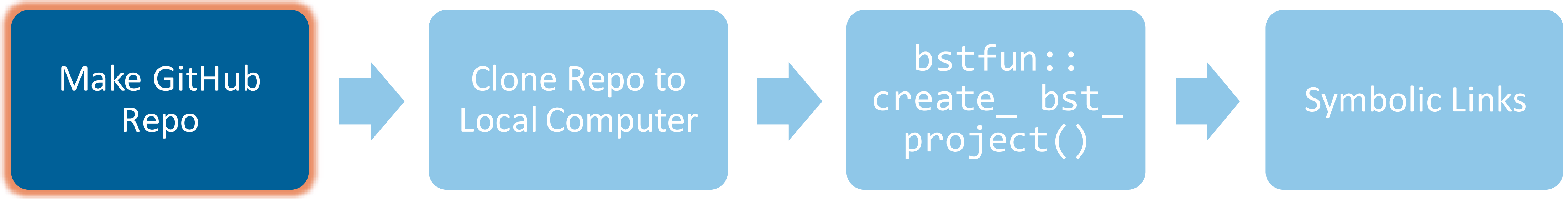


Make GitHub Repo

Clone Repo to Local Computer

`bstfun::create_bst_project()`

Symbolic Links



Make GitHub Repo

Clone Repo to Local Computer

`bstfun::create_bst_project()`

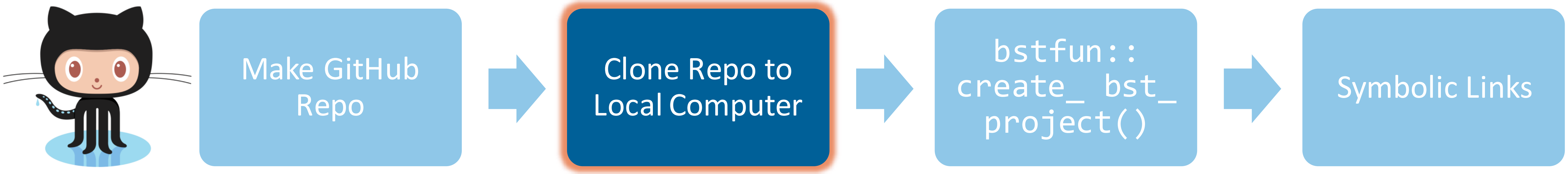
Symbolic Links

WHAT TO DO

- **Once you have completed your GitHub PHI Training…**
- Go to github.mskcc.org/Biostat-Analytic-Projects
- Create a new repository with an informative name
  - `<PI Last Name> <short description of project>`
  - Repo can be empty, or include a README.md file

WHY

- Version control, collaboration with GitHub
- Biostat-Analytic-Projects organization is set up for incidental PHI
- *Can access your repo from multiple locations (i.e. your desktop, your laptop, the computing cluster, etc!)*



Make GitHub Repo

Clone Repo to Local Computer

`bstfun::create_ bst_ project()`

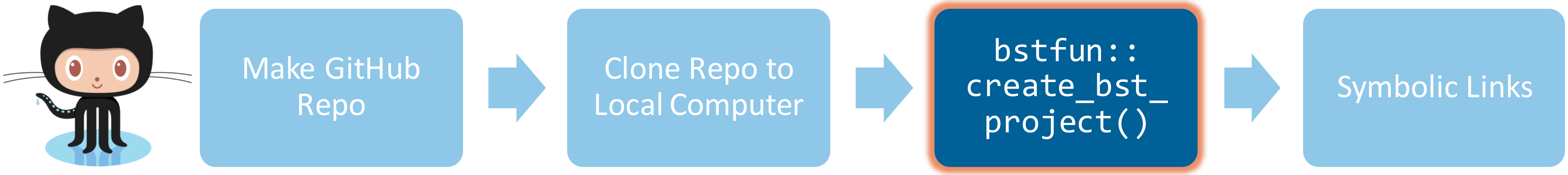
Symbolic Links

WHAT TO DO

- Clone the new repo to your local computer
  - Local repo should be in a folder called "GitHub" on your OneDrive (which lives on the C: drive)
- See previous GitHub trainings for more details

WHY

- We need the repo to be on the local computer in order to make edits and commit changes



Make GitHub Repo

Clone Repo to Local Computer

`bstfun::create_bst_project()`

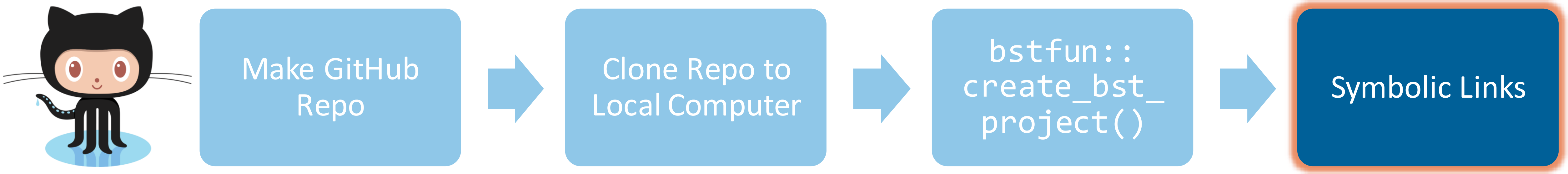
Symbolic Links

WHAT TO DO

- In RStudio, run `bstfun::create_bst_project()`
  - There will be some prompts in the RStudio console asking for preferences
- Path passed to the function should be to the cloned version of the Git repo
- Can also pass the path to the data
  - `path_data = “H:\...\Project Folder\secure_data”`

WHY

- Sets up a **quality project skeleton**
  - Separate scripts for setup, analysis, and report
  - SAP document shell
  - Labelled variables
- Automatically initializes `{renv}`
- Can detect if a folder is established on GitHub/will link the repo
- Loads `{biostatR}`



Make GitHub Repo

Clone Repo to Local Computer

`bstfun::create_bst_project()`

Symbolic Links

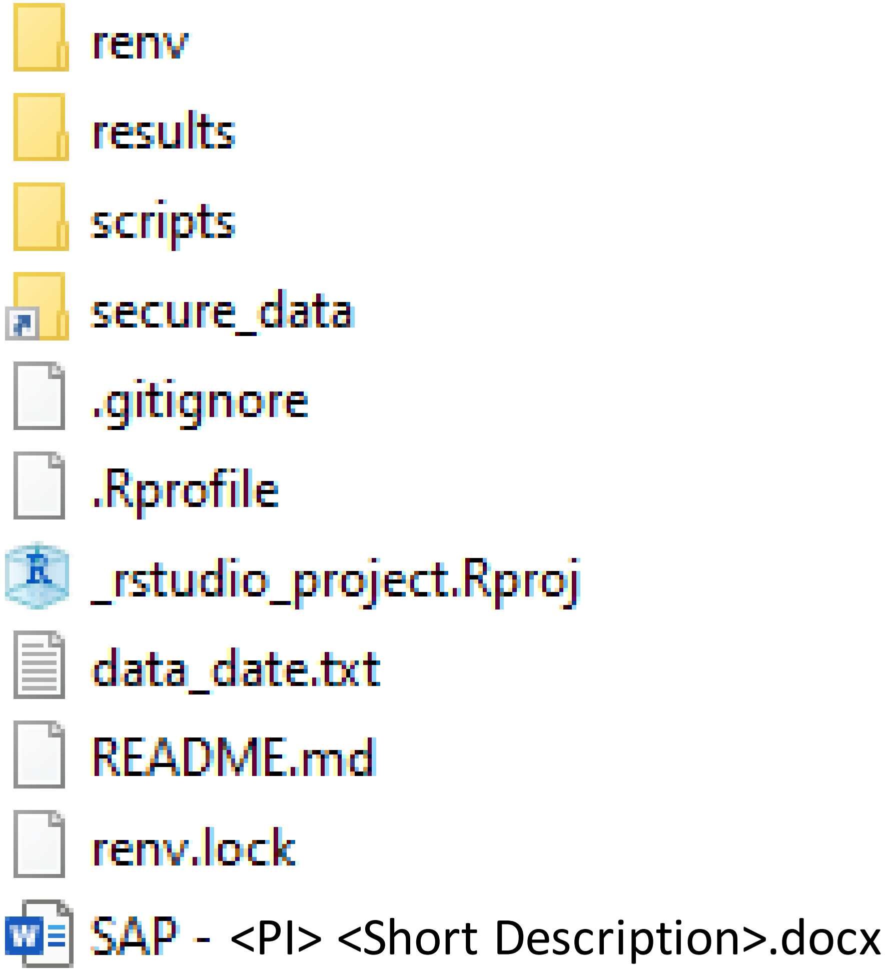
WHAT TO DO

- `bstfun::create_bst_project(…, path_data = <data path>)`
- `starter::create_symlink()`
- In your scripts, you can point to the data by using `bstfun::here_data()`

WHY

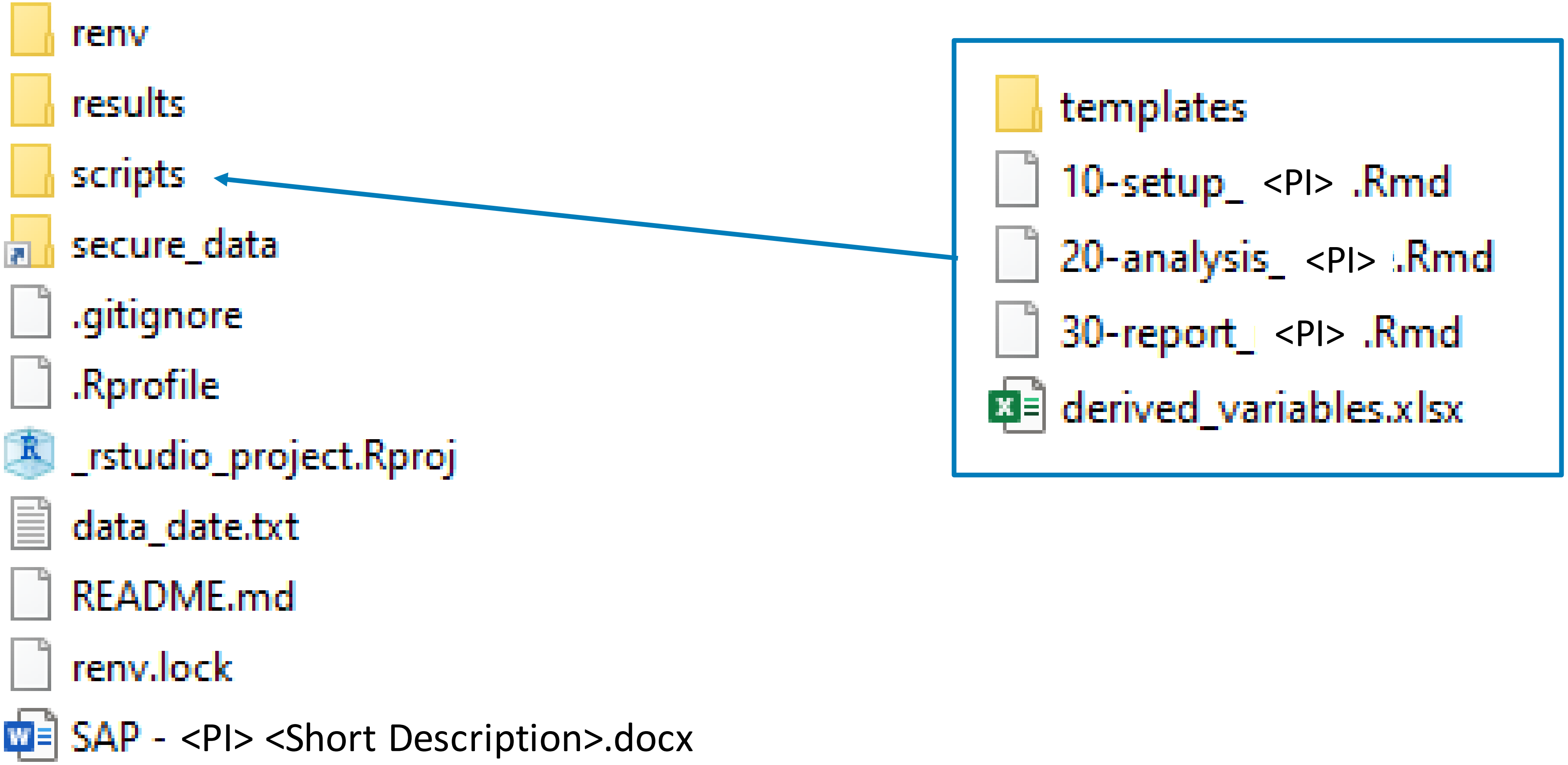
- Data should be saved on a network drive that is backed up and secure (for example, the H: drive)
- The symbolic link will put a shortcut in your project folder that links to the actual data location

New Project Shell



- renv
- results
- scripts
- secure_data
- .gitignore
- .Rprofile
- _rstudio_project.Rproj
- data_date.txt
- README.md
- renv.lock
- SAP - <PI> <Short Description>.docx

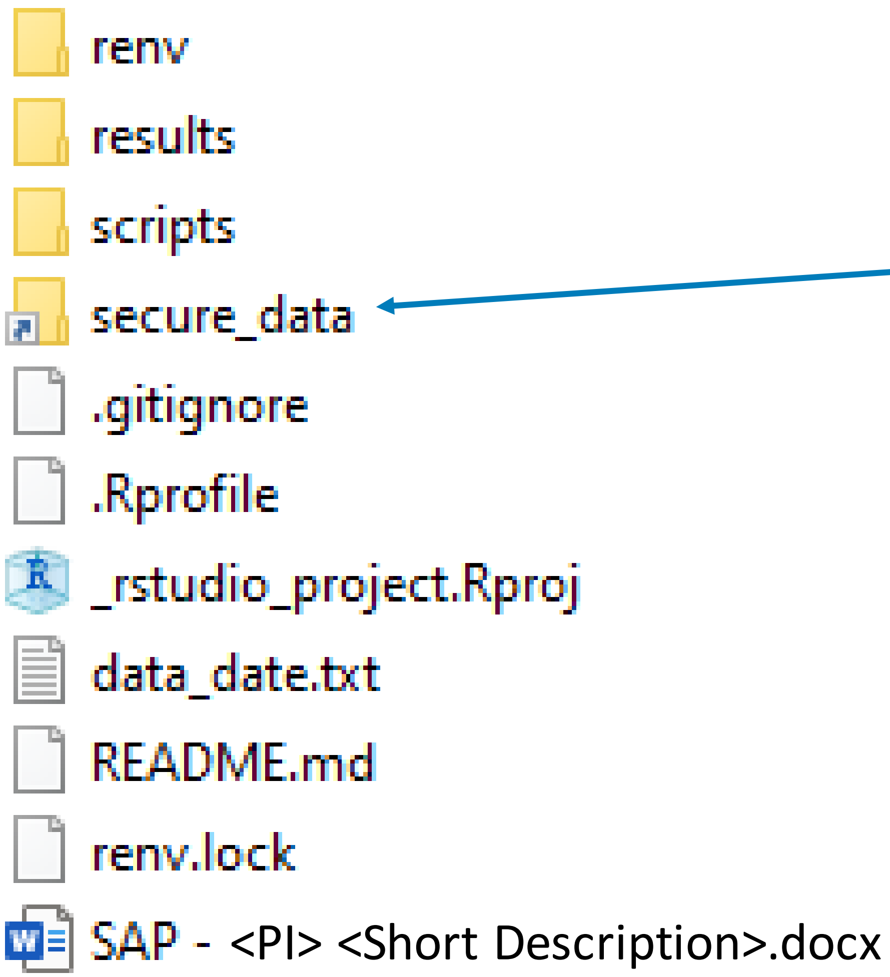
New Project Shell



renv
results
scripts
secure_data
.gitignore
.Rprofile
_rstudio_project.Rproj
data_date.txt
README.md
renv.lock
SAP - <PI> <Short Description>.docx

templates
10-setup_ <PI> .Rmd
20-analysis_ <PI> .Rmd
30-report_ <PI> .Rmd
derived_variables.xlsx

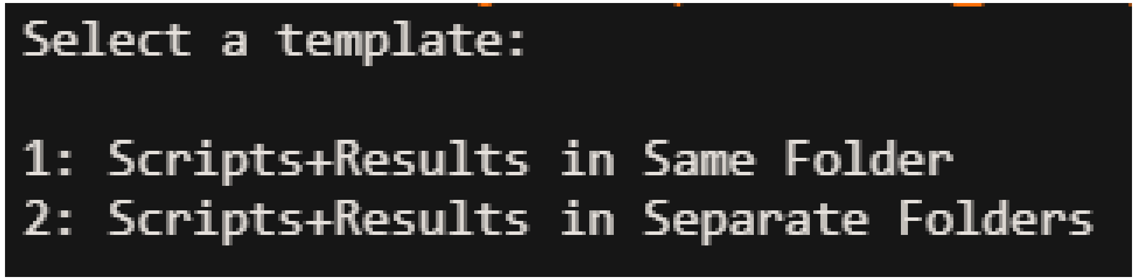
New Project Shell



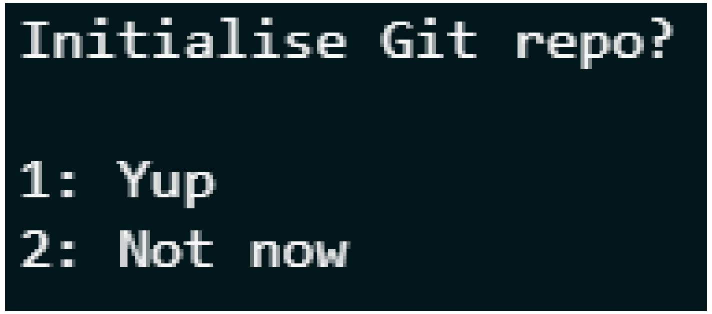
renv
results
scripts
secure_data
.gitignore
.Rprofile
_rstudio_project.Rproj
data_date.txt
README.md
renv.lock
SAP - <PI> <Short Description>.docx

This is the symbolic link. If you click on this folder, you will be taken to the data (with a different path)

When running `bstfun::create_bst_project()`, you’ll get some prompts:

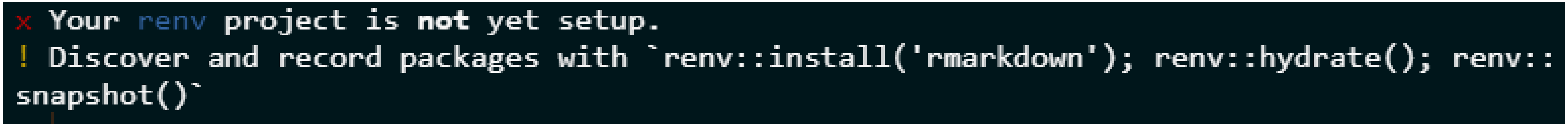


```
Select a template:

1: Scripts+Results in Same Folder
2: Scripts+Results in Separate Folders
```



```
Initialise Git repo?

1: Yup
2: Not now
```

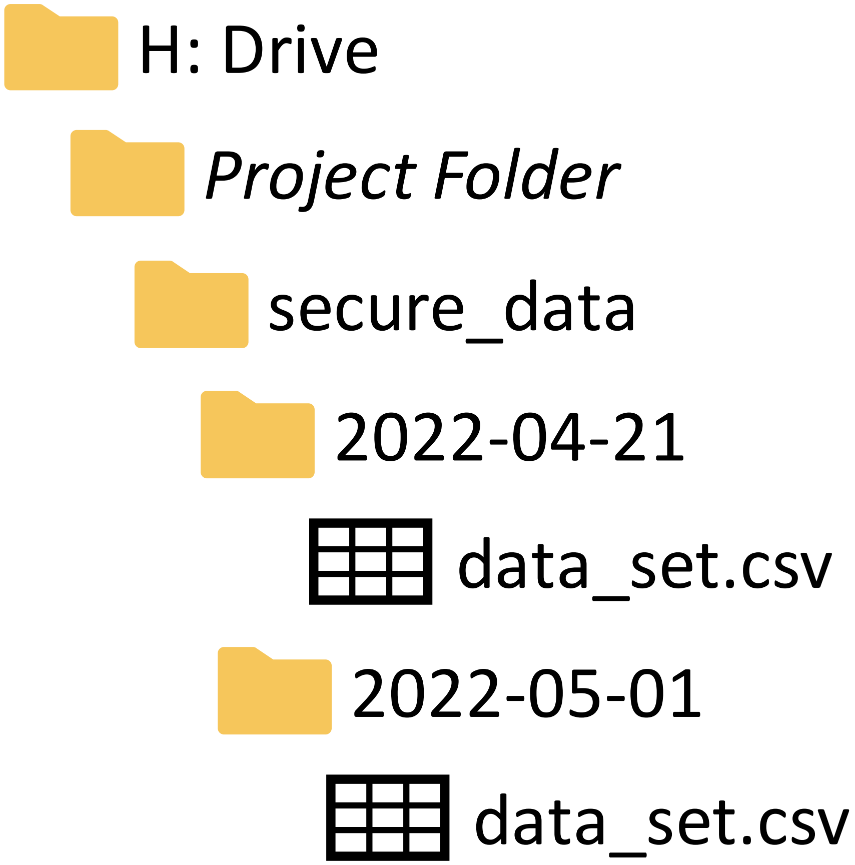
Your new R project will open automatically and will prompt you about `{renv}` tasks:



```
x Your renv project is not yet setup.
! Discover and record packages with `renv::install('rmarkdown'); renv::hydrate(); renv::
snapshot()`
```

bstfun::here_data()

- All data should be stored in a secure folder on a network drive with a subfolder indicating the date the data was received
- Every time new data is received, make a new date folder under **secure_data**
- The file **data_date.txt** stores the date of the current data – update this file when you get new data.
- `bstfun::here_data()` locates the current data folder based on **data_date.txt**



H: Drive

Project Folder

secure_data

2022-04-21

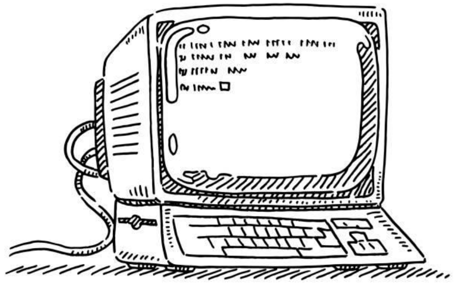
data_set.csv

2022-05-01

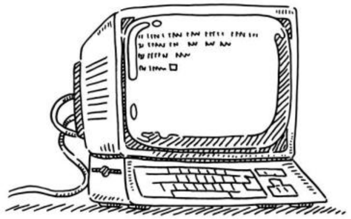
data_set.csv

```
> bstfun::here_data()
“C:/Users/username/OneDrive/GitHub/<PI> <short description>/secure_data/2022-05-01”
```

Recommended Workflow For **Non-GitHub Projects/Users**



```
bstfun::create_bst_project()
```



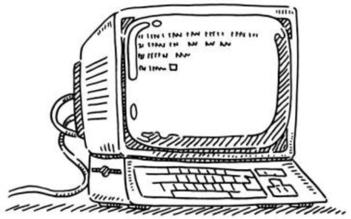
bstfun::create_bst_project()

WHAT TO DO

- In RStudio, run `bstfun::create_bst_project()`
    - There will be some prompts in the Rstudio console asking for preferences
- Path passed to the function should be to the project folder in the “Analytic Projects” folder on the OneDrive

WHY

- Sets up a **quality project skeleton**
    - Separate scripts for setup, analysis, and report
    - SAP document shell
    - Labelled variables
- Automatically initializes `{renv}`
- Loads `{biostatR}`



`bstfun::create_bst_project()`

- Follow the correct folder organization!
- Not necessary to store data on a network drive, but you could
  - Regardless, functions like `here_data()` can work if you have a **data_date.txt** file, and name your data folder **secure_data**

New Project Setup

- GitHub
- bstfun::create_bst_project()
- Symbolic Links

Questions?

Memorial Sloan Kettering
Cancer Center

{renv}

What is {renv}?

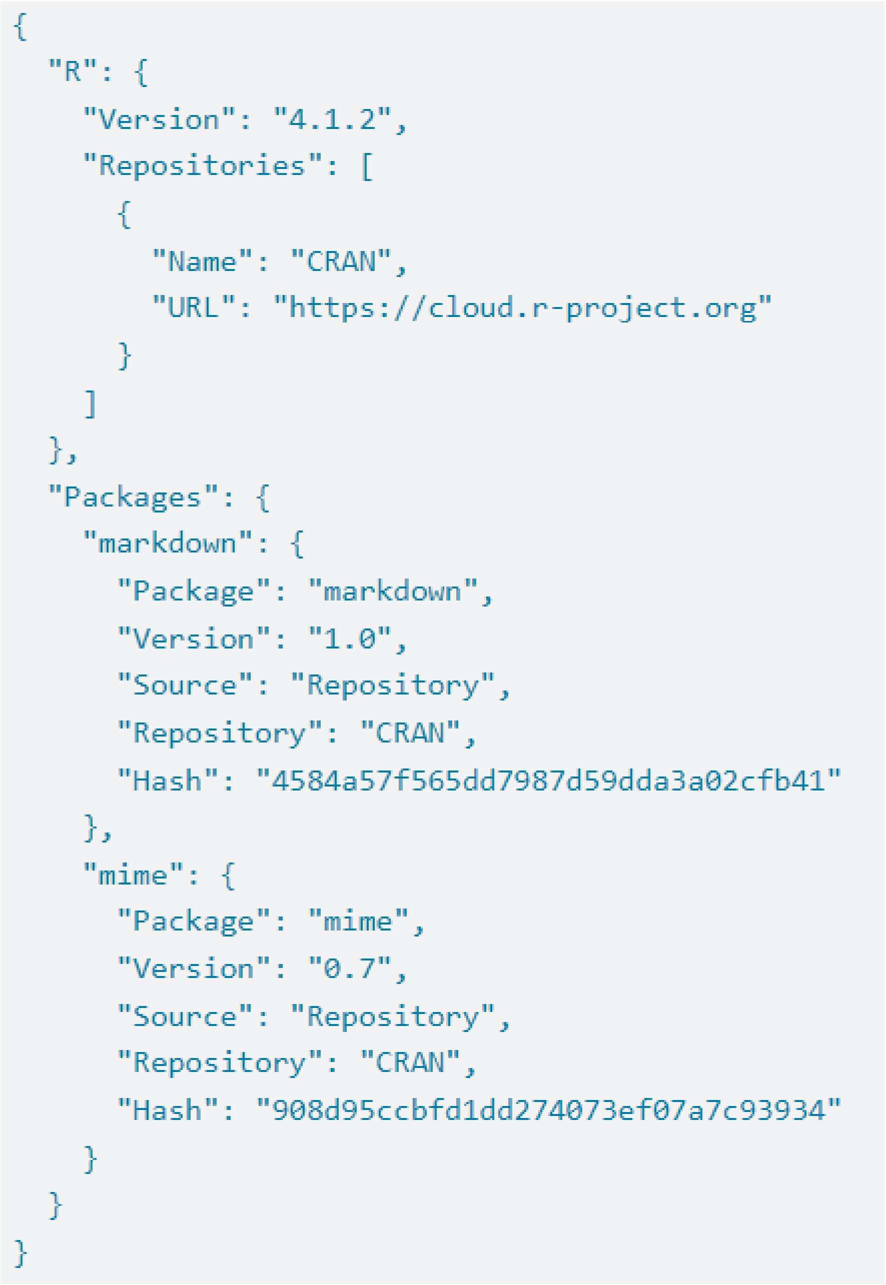
- Package to create **r**eproducible **env**ironments for your R projects
  - **Isolated**: installing a package in one project won’t break code in other projects
  - **Portable**: easily install the packages your projects depends on to another computer
  - **Reproducible**: ensure exact versions get installed

Why use {renv}?

- Can safely upgrade packages without breaking code in other projects
- Easy collaboration tool – collaborators can install exact packages/versions needed for a project easily from the lockfile
- Can create a “time capsule” for when you don’t touch a project for a while

How does {renv} work?

- Sets up a private library for each R project
- Creates a lockfile called **renv.lock**
  - Stores R version, renv version, package versions, and more



```
{
  "R": {
    "Version": "4.1.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "markdown": {
      "Package": "markdown",
      "Version": "1.0",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash": "4584a57f565dd7987d59dda3a02cfb41"
    },
    "mime": {
      "Package": "mime",
      "Version": "0.7",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash": "908d95ccbfd1dd274073ef07a7c93934"
    }
  }
}
```

How do I use {renv} in my projects?

- `renv::init()` – initializing a new renv
  - Creates a folder *specific to this project* to stores packages in/loads packages from
- `renv::snapshot()` – takes a “snapshot” of the current packages/versions used in the project to store in the lockfile
- `renv::restore()` – loads packages and versions as recorded in the lockfile into the project library
- `renv::status()` – reports differences between the lockfile and the project library

How does {renv} fit into our recommended workflow?

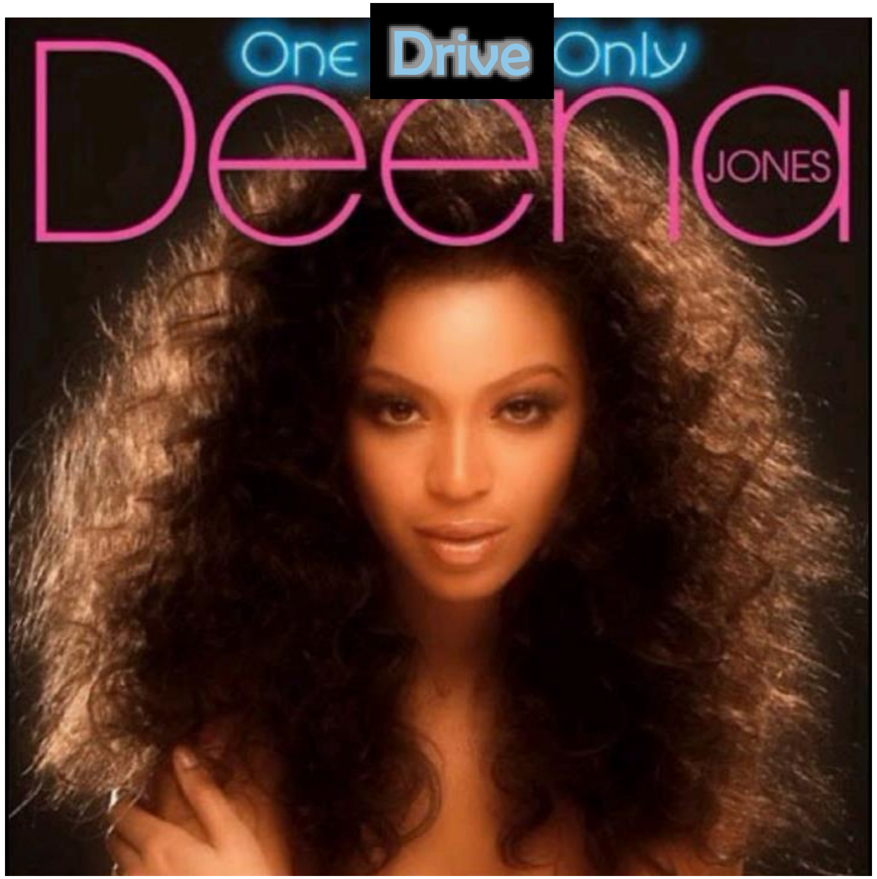
- `bstfun::create_bst_project()` will automatically initialize renv/start a lockfile
- While working on your project, periodically run `renv::snapshot()` to update the lockfile, especially after installing/loading/using a new package
- If returning to an old project, run `renv::restore()` to ensure the packages are consistent with the lockfile

{biostatR}

- `{renv}` isolates your project, `{biostatR}` brings needed packages to your project:
    - `{ragg}`, `{flextable}`, `{ftExtra}`, `{styler}`, `{remedy}`
- Loading `{biostatR}` messages you if your R installation is more than 2 versions behind.
    - More than 2 versions behind denies you access to pre-compiled builds on CRAN, and will sometimes install old compiled versions if you choose not to compile yourself. (It doesn’t warn about installing a very old version!)
- Notifies you of out-of-date packages
- Confirms your RSPM is set up correctly

Why OneDrive Only?

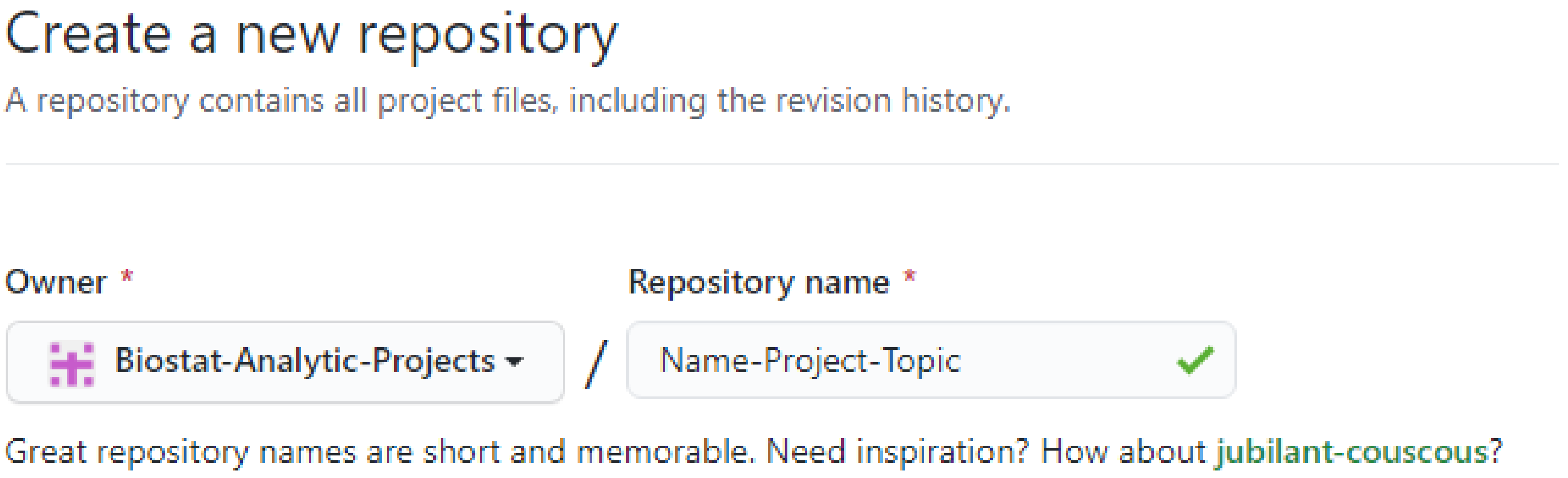
- If your projects (i.e. R projects) are not on the same storage drive as your renv cache…
  - Projects will take forever to load (**literally** 1.5 hours for basic project)
  - {renv} will need to reinstall every single package for each new project
- Saving everything to the OneDrive solves these problems!



Memorial Sloan Kettering Cancer Center

Demo: Starting a New Project

1. Create new repository on github.mskcc.org/Biostat-Analytic-Projects



Create a new repository

A repository contains all project files, including the revision history.

Owner *

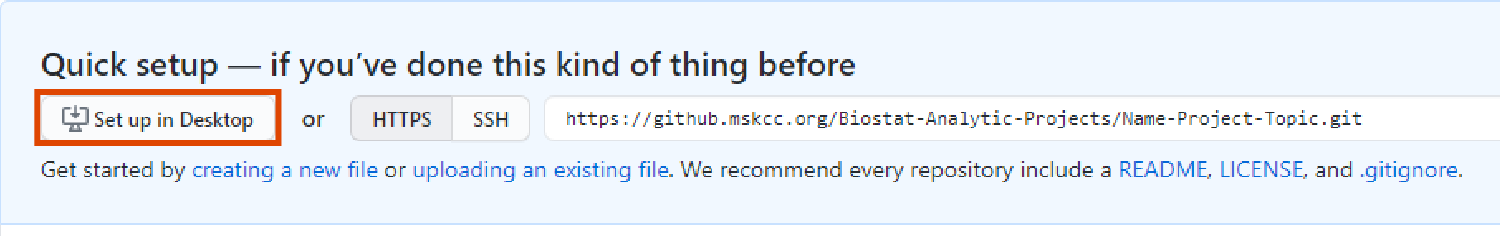
Biostat-Analytic-Projects ▾ / Repository name * Name-Project-Topic ✓

Great repository names are short and memorable. Need inspiration? How about jubilant-couscous?

Helpful Hints:

- Check out the previous trainings for more specifics on how to create the remote repo
- Name your project something meaningful, such as <PI name> <short description>

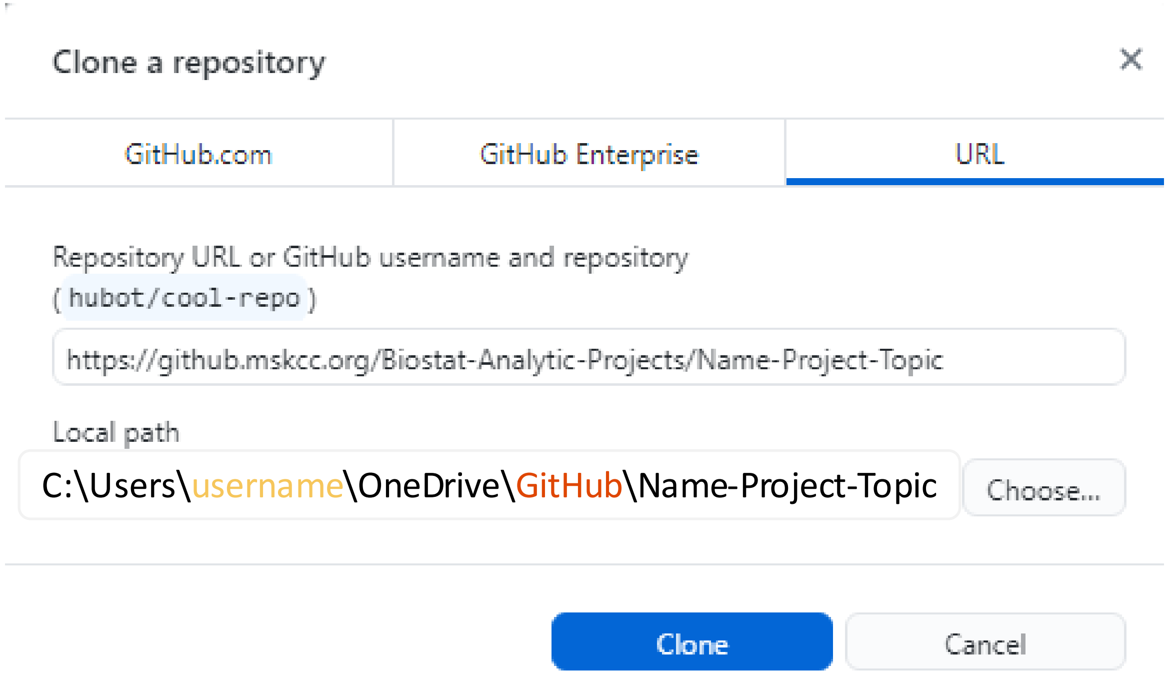
2. Clone new repo to local computer



Quick setup — if you've done this kind of thing before

https://github.mskcc.org/Biostat-Analytic-Projects/Name-Project-Topic.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.



Clone a repository

Repository URL or GitHub username and repository (hubot/cool-repo)

https://github.mskcc.org/Biostat-Analytic-Projects/Name-Project-Topic

Local path

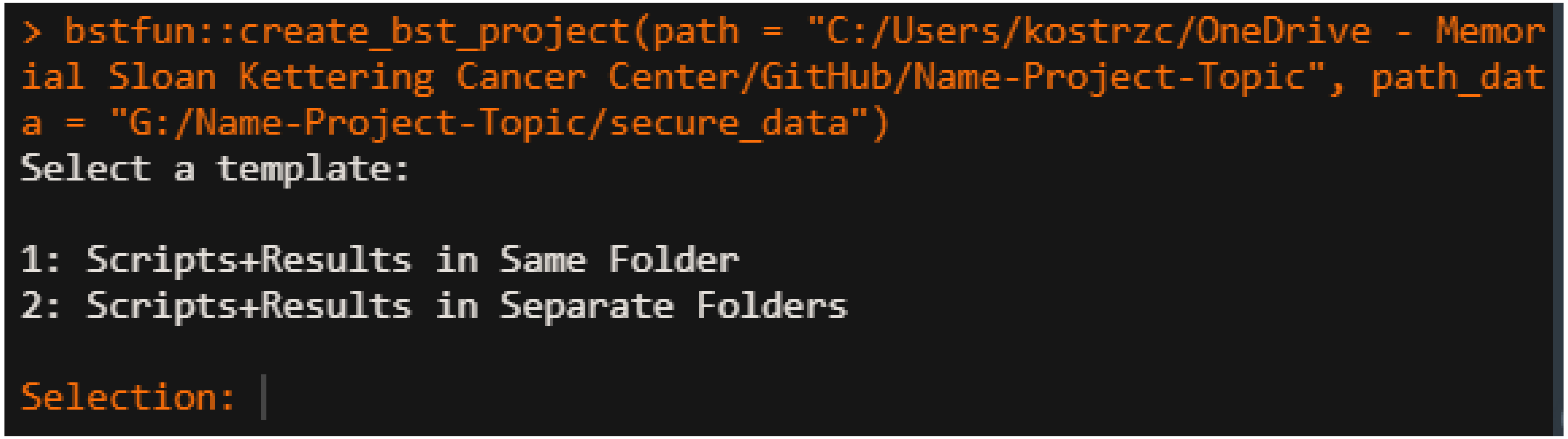
C:\Users\username\OneDrive\GitHub\Name-Project-Topic

Helpful Hints:

- The "Set up in Desktop" button looks slightly different (i.e. not green) when you initialize an empty repo
- Remember to change the local path to the recommended path! (That is, a folder called "GitHub" that lives on your OneDrive.)

3. Open RStudio and run

```
bstfun::create_bst_project(
    path = <path to cloned repo>,
    path_data = <path to secure data>
)
```

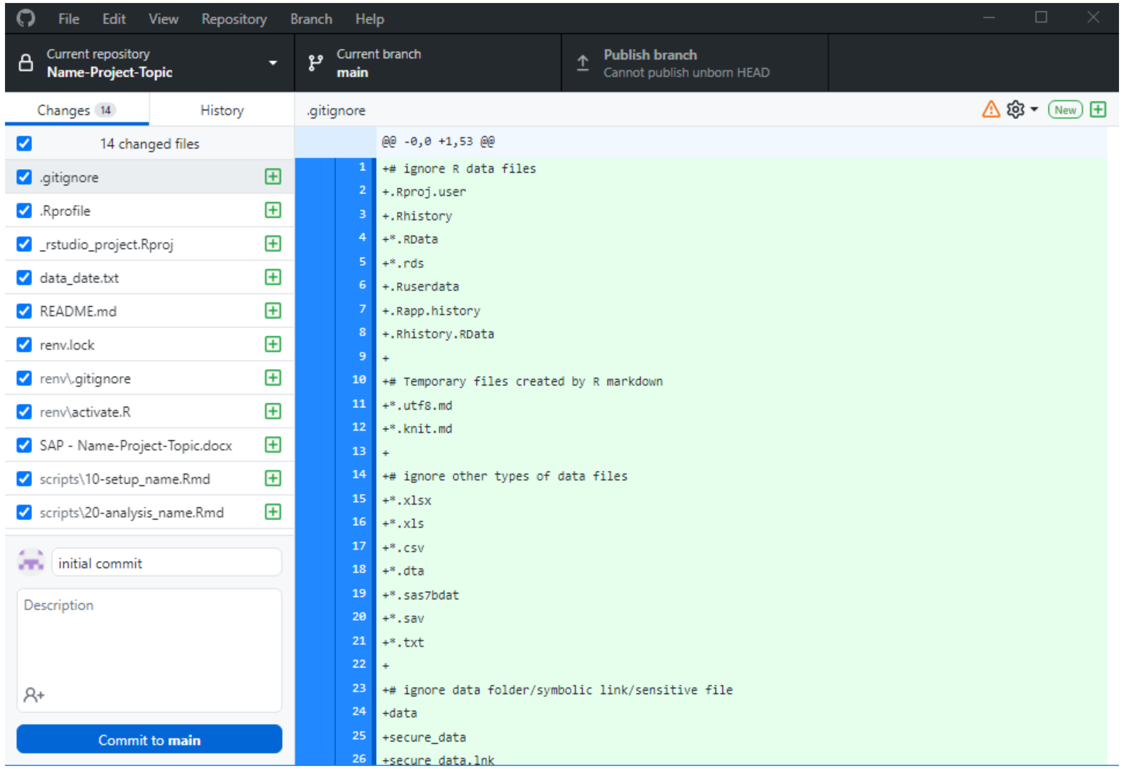


```
> bstfun::create_bst_project(path = "C:/Users/kostrzc/OneDrive - Memor
ial Sloan Kettering Cancer Center/GitHub/Name-Project-Topic", path_dat
a = "G:/Name-Project-Topic/secure_data")
Select a template:

1: Scripts+Results in Same Folder
2: Scripts+Results in Separate Folders

Selection: 
```

Helpful Hints:

- `path_data` should end in the **secure_data** folder (data should be stored on a network drive)
- Read through the output from the function to learn more

4. Make initial commit of new project files, push/publish to remote



☑ 14 changed files

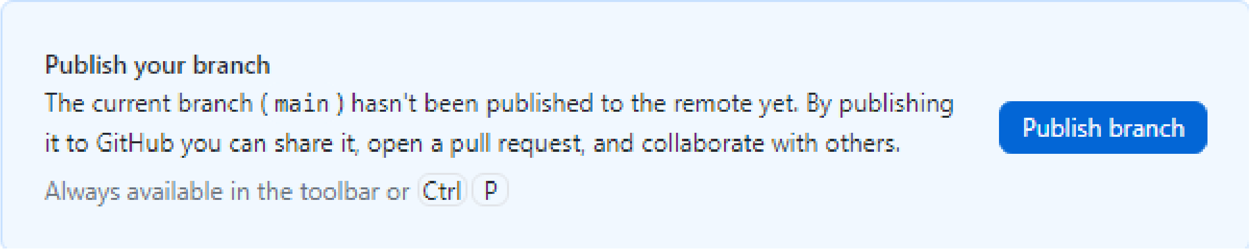
- ☑ .gitignore
- ☑ .Rprofile
- ☑ _rstudio_project.Rproj
- ☑ data_date.txt
- ☑ README.md
- ☑ renv.lock
- ☑ renv\.gitignore
- ☑ renv\activate.R
- ☑ SAP - Name-Project-Topic.docx
- ☑ scripts\10-setup_name.Rmd
- ☑ scripts\20-analysis_name.Rmd

initial commit

Description

Commit to **main**

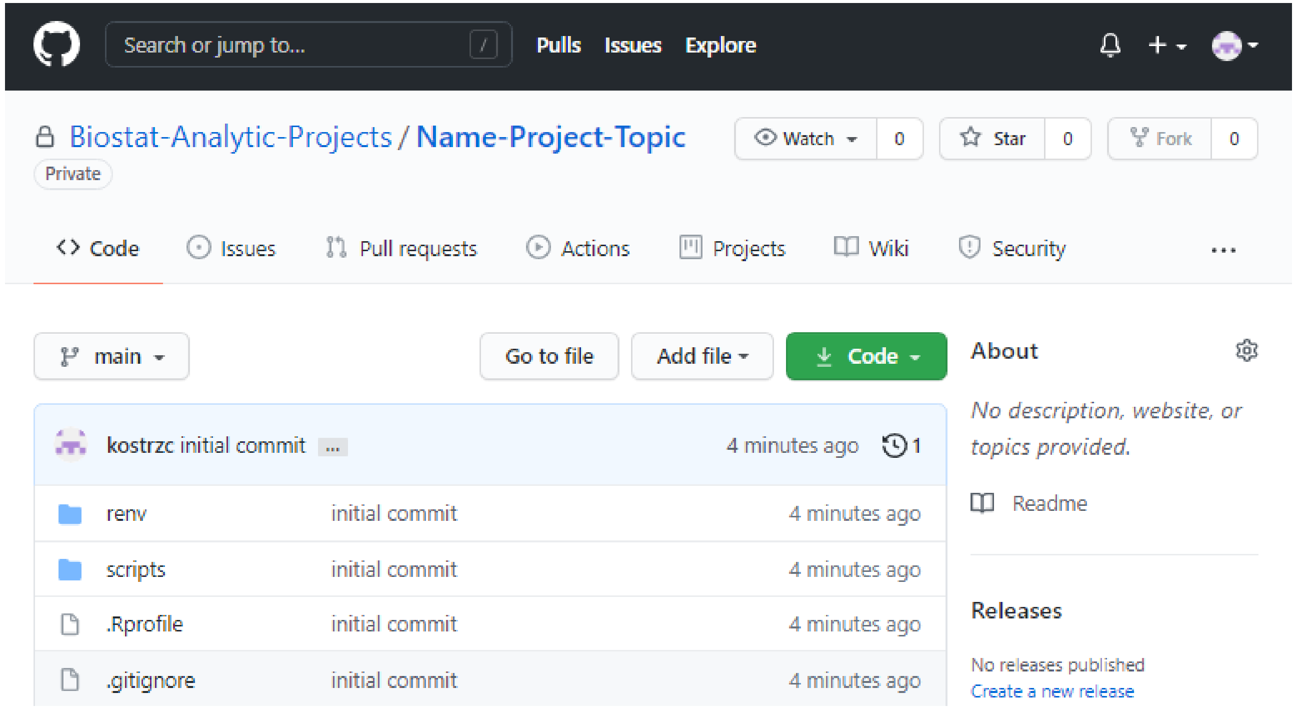
```
@@ -0,0 +1,53 @@
+# ignore R data files
+.Rproj.user
+.Rhistory
+*.RData
+*.rds
+.Ruserdata
+.Rapp.history
+.Rhistory.RData
+
+# Temporary files created by R markdown
+*.utf8.md
+*.knit.md
+
+# ignore other types of data files
+*.xlsx
+*.xls
+*.csv
+*.dta
+*.sas7bdat
+*.sav
+*.txt
+
+# ignore data folder/symbolic link/sensitive file
+data
+secure_data
+secure_data.lnk
```



Publish your branch

The current branch (main) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.

Always available in the toolbar or Ctrl P



kostrzc initial commit ... 4 minutes ago 1

renv	initial commit	4 minutes ago
scripts	initial commit	4 minutes ago
.Rprofile	initial commit	4 minutes ago
.gitignore	initial commit	4 minutes ago

No description, website, or topics provided.

No releases published
Create a new release

Helpful Hints:

- Revisit previous GitHub trainings for more specifics about committing/pushing/etc.
- **Git won't commit an empty folder**, so if you commit everything before editing the scripts, your results folder will not appear on the remote repo/won't be pulled if you switch computers.
- You can check the remote git repo to see what was committed