

~ / > _

Plain-Text Tools

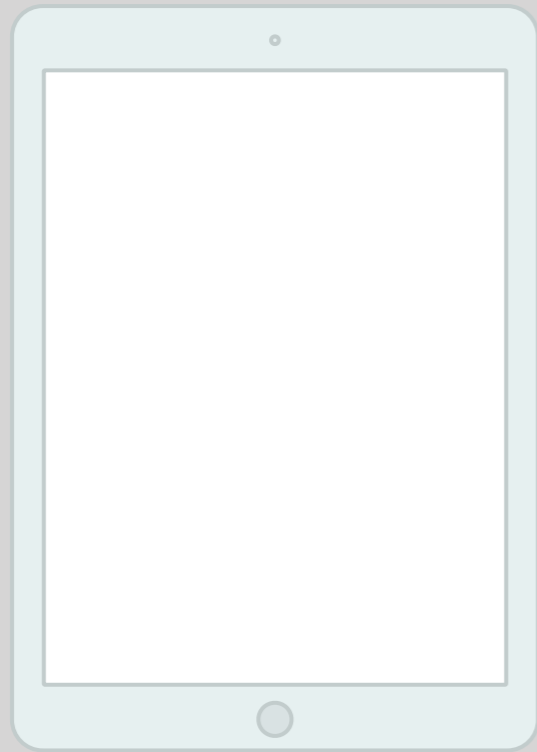
~ / > previously ...

Tools

R, R Studio

git, GitHub

bash, shell utilities



Two Revolutions

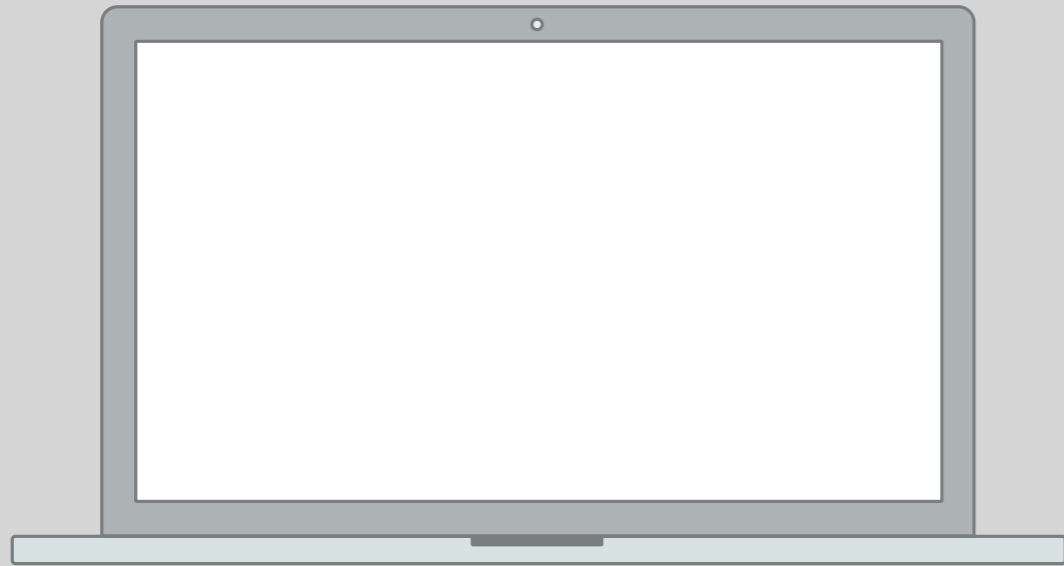


- **Touch-Based UI**
- **Foregrounds Single-Purpose Applications**
- **Dislikes Multi-Tasking***
- **Hides the File System**

Where the Action is

I mean, “Making applications work together in the service of a single goal”, not “Checking Facebook while also listening to a talk and keeping up with the latest score in the game.”

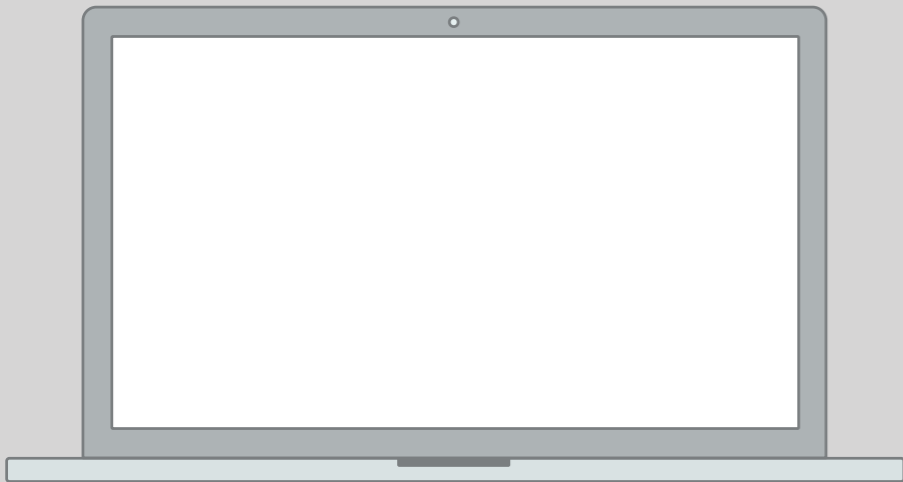
***Multitasking**



- **Windows and Pointers**
- **Multi-Tasking**
- **Exposes the File System**
- **Underneath, it's the 1970s, UNIX, and the command line**

We Still Live Here

- Better than they've ever been!
- Free! Open! Powerful!
- Improving continuously!
- But grounded in a UI paradigm that is increasingly far away from the everyday use of computing devices



Plain-Text Data Tools

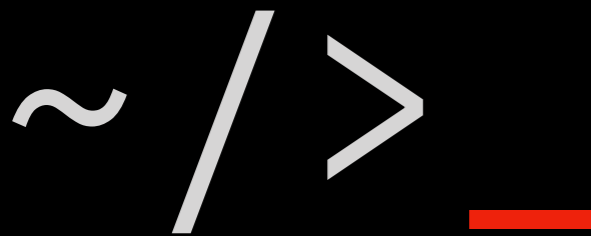
The Research Process

~ / > _

is intrinsically messy

“[T]he type of writing I typically do ... is loaded with facts. I am constantly referring to photographs, drawings, experimental test results, calculations, reports written by others, textbooks, journal articles, and so on. These are not distractions; they are essential to the writing process.

And it's not just reference material. Quite often I need to make my own graphs and drawings to include in a report. Because the text and the graphics are all part of a coherent whole, I need to go back and forth between the two; the words inform the pictures and the pictures inform the words. This is not the Platonic ideal of a clean writing environment—a cup of coffee on an empty desk in a white room—that you see in videos for distraction-free editors.



Some of the popularity of these editors is part of the backlash against multitasking, but people are confusing themselves with their computers. When I'm writing a report, that is my single task, and I bring to bear whatever tools are necessary to complete it. That my computer is multitasking by running many programs simultaneously isn't a source of confusion or distraction, it's the natural and efficient way for me to get my one task done.”

Two Models

~ / > _

of your Work

Office Model

- What is “Real” in your Project?
- What is the Final Output and how is it produced?
- How are changes managed?

Engineering Model

- A Ledger of your actions,**
- that can replay what you did,**
- keep track of what others did,**
- and help branch and merge code**

git: local use case

- **Your project lives on your Computer**
- **You add and commit changes as you go**
- **That's it.**
- **It's just a ledger**

```
~/> git add new_analysis.r
```

```
~/> git commit -m "Fixed MLE issue"
```

git

~/ > _

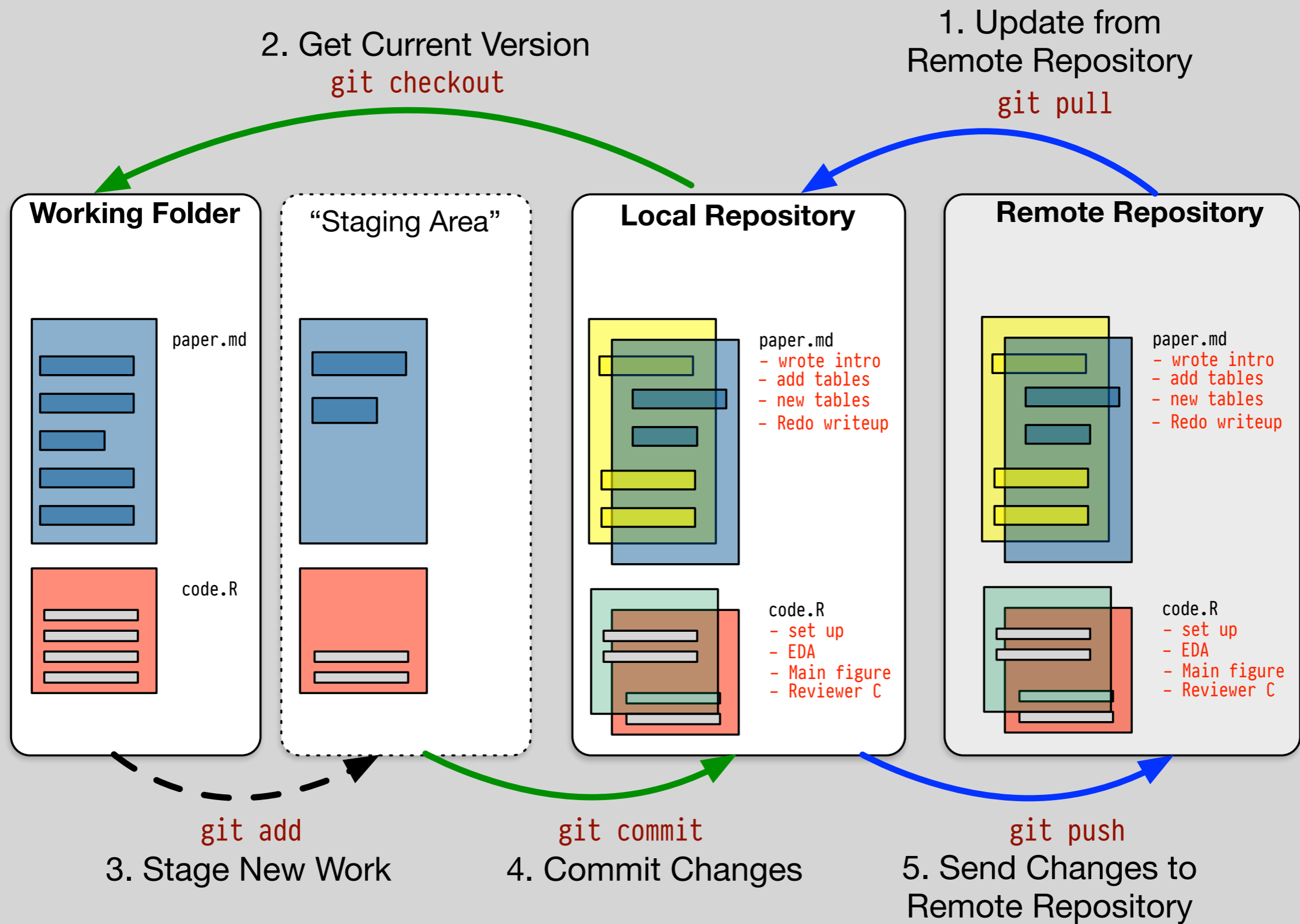
A too-rapid tour

I'm assuming you did this

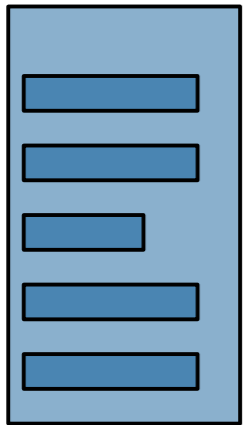
<https://happygitwithr.com>

git: GitHub use case

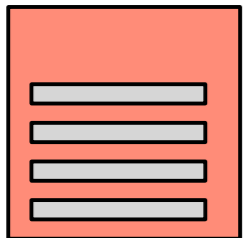
- **Your project lives on GitHub**
- **There's a working copy on your Computer**
- **You add and commit changes locally**
- **You push the changes to GitHub**



Working Folder

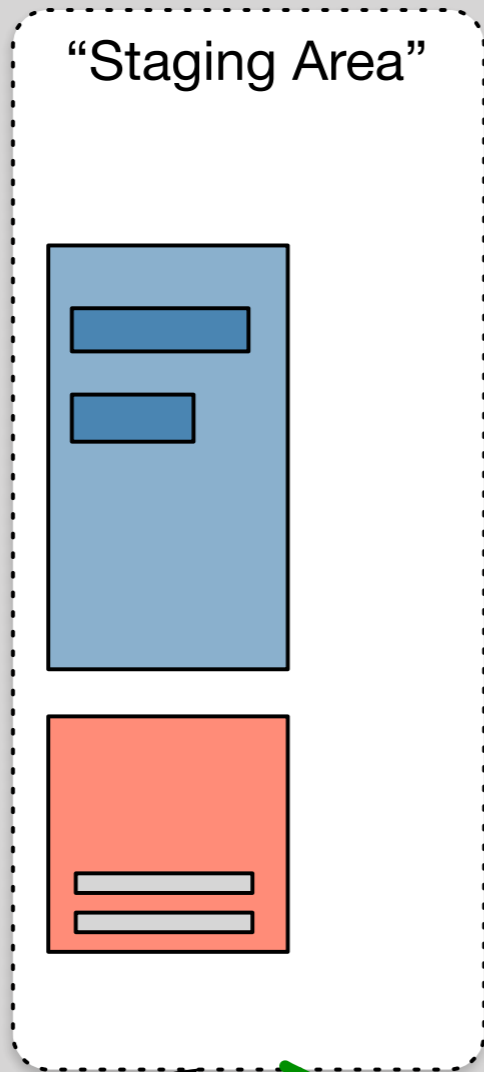
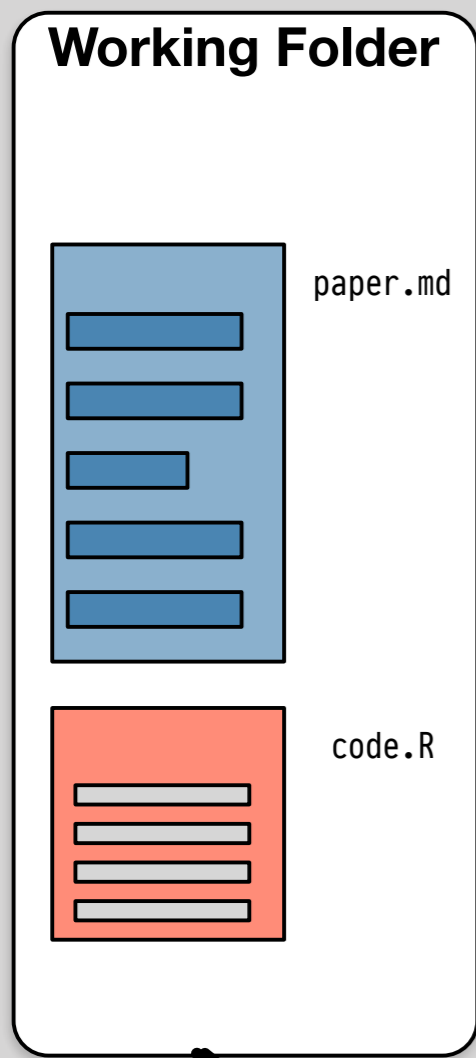


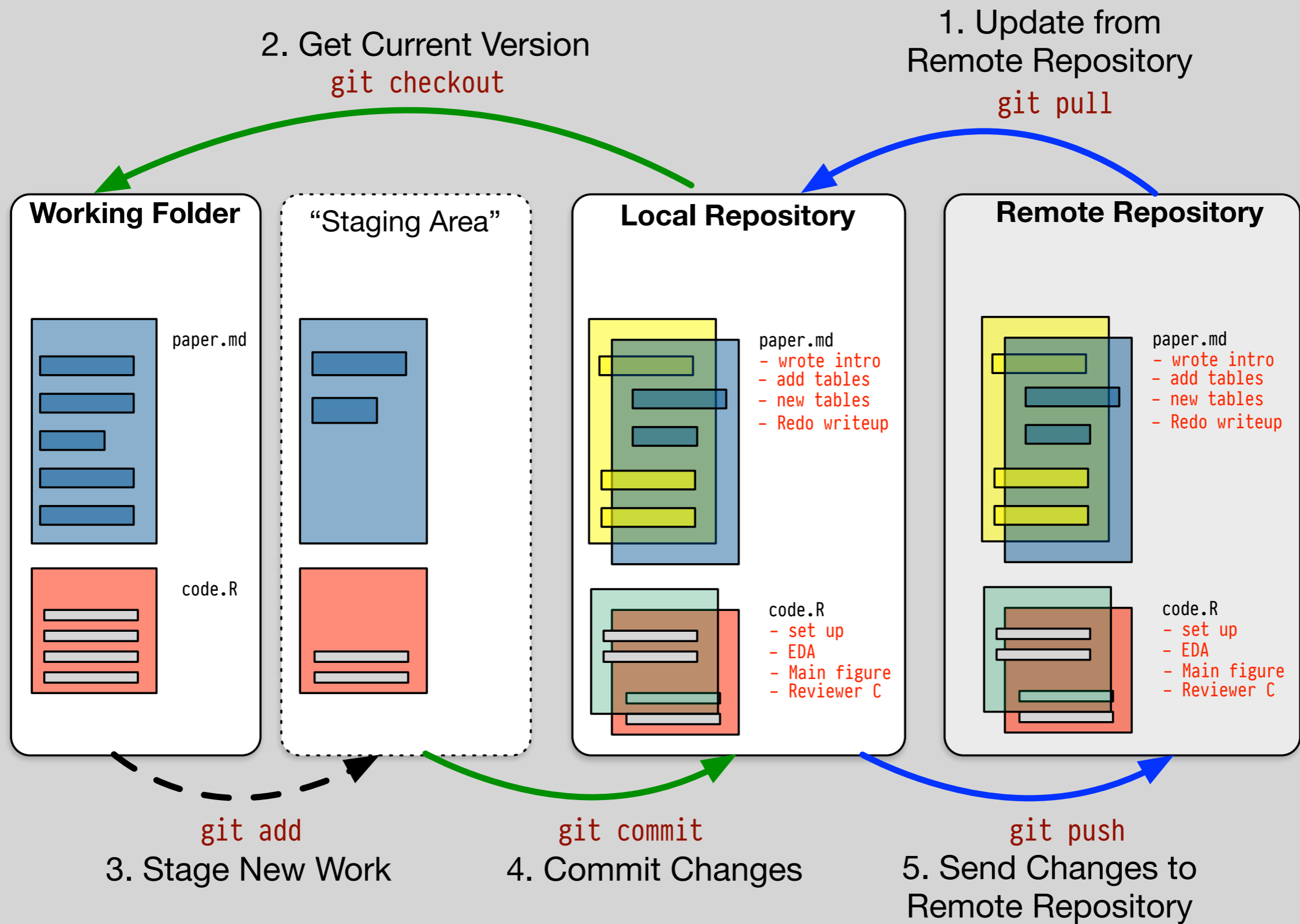
paper.md



code.R







git: other use cases

- You are experimenting, or collaborating
- You can make **branches** of your project
- You add and commit to the branch
- You merge changes into the **master** branch