Section 1

Pipelines and Project Components

Overview

It is useful to break down a project into its most basic parts, which we call **pipeline** and **components**.

- A pipeline refers to the **steps that are necessary to build a project** (e.g., prepare dataset, run model, produce tables and figures), and
- Components refer to **a project's most nuclear building blocks** (e.g., data, source code, and generated temporary and/or output files).

Later on, you will see that such a structure enables you to work on your project using multiple computers (e.g., your workstation, your laptop, a computer in the cloud), or with various collaborators/co-authors.

Project Pipelines

In a research project, one typically has several tasks to accomplish, such as preparing the data, analyzing the data, writing the paper and producing a set of slides. **This is what we call a "project pipeline."**

A typical pipeline for an academic paper may look like this:

!!! example "Typical pipeline for an academic paper" - Prepare dataset for analysis - Run model on a dataset - Produce tables and figures for the paper

Over time, your pipeline will grow increasingly complex. For example, the pipeline above recently "matured" into this one:

!!! example "More complex pipeline for an academic paper" - Download public datasets from the U.N. (to be used for some control variables) - Have an RA code some auxiliary variables - files to be delivered in Excel - Merge your primary data set with control variables from previous steps; generate to "derivative" data sets

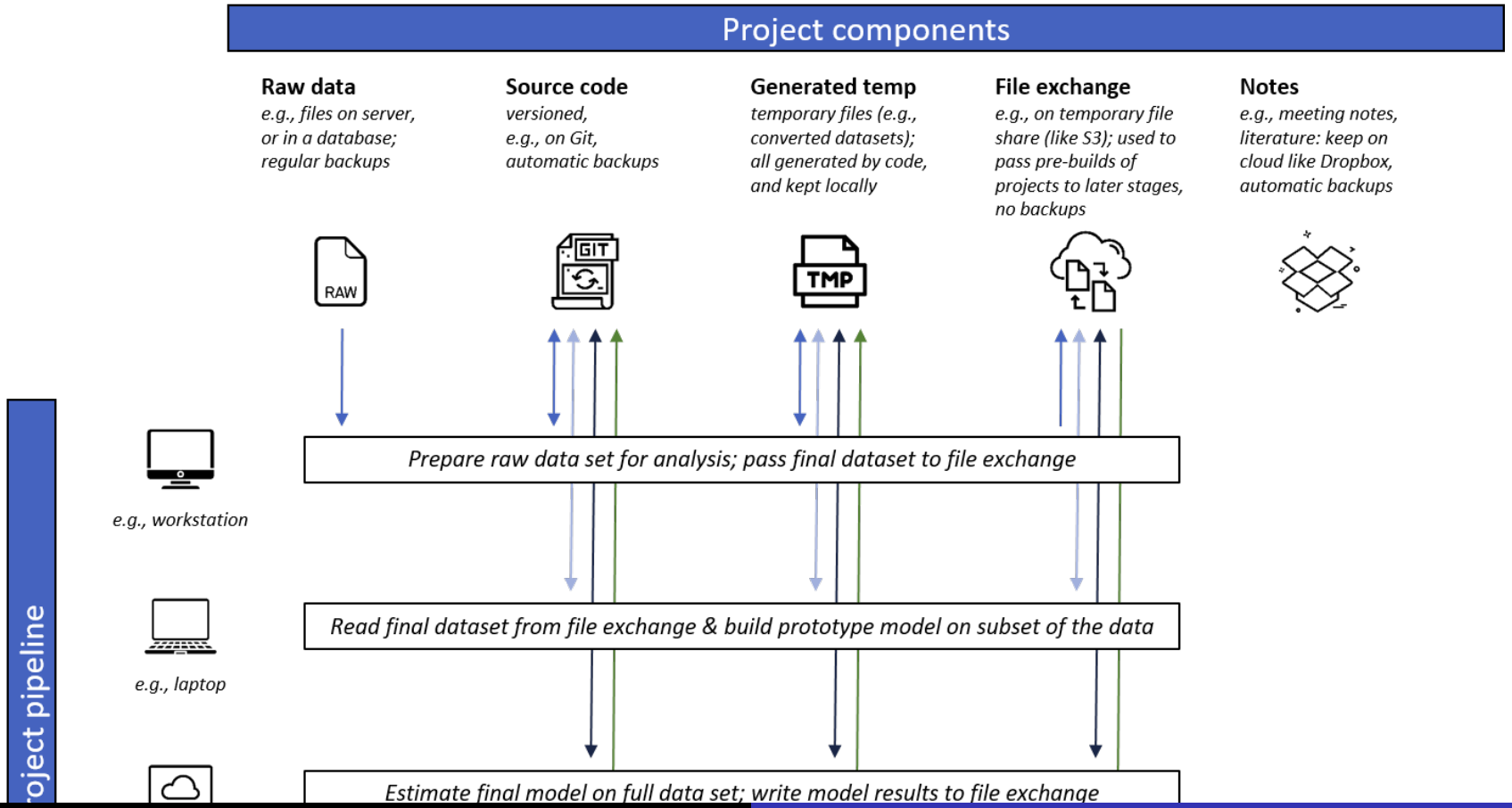
Project Components

Now that we've covered what a pipeline is, let's draw our attention to **project components**, which are the most nuclear building blocks of a project. It's useful to think of these components as *separate entities of your project* because their nature allows you to apply different data management policies.

- For example, we probably all agree it's desirable to *roll back to previous versions of a project* (e.g., an earlier version of a prepped dataset). But - if you work on large datasets - it may probably be too burdensome to store each version of your **generated files** (e.g., in one of the projects we've been working on, the generated (cleaned) data sets were a 500 GB, and we've created probably close to 50 versions = 25 TB).
- If you think about this a bit more, you may discover that storing these different data sets is completely inefficient - as the combination of raw data and *versioned source code* will be able to "re-cast" any data set version you have ever worked on.

Putting it all together...

It's time to finally build your project! See our overview chart below, which illustrates how different stages of your pipeline use different project components.



Summary

!!! summary

You've just learnt about two essential ways to look at a pr

1. The **pipeline** defines the logical steps in which your
is built (such as prepping data, analyzing it, and creating
2. The **components** refer to the most nuclear units in a
of raw data, source code, generated temporary/output files
and lastly a collection of notes/other documents.

The power of setting up your project that way lies in:

- **full portability**
    - Because of the modular nature of the project, each
    can essentially be executed on different computing sy
    handy if you have a powerful workstation in your offi
    some large data sets), but would like to work on your