

# Colab + GitHub Quickstart

---

UoEO Computational Psychology Lab (dissertation support)

This guide shows you how to run the lab-starter notebooks in Google Colab and save your work back to your GitHub repo.

## What you need

- A GitHub account (free).
- A Google account to use Google Colab (free).
- A web browser (Chrome or Edge is safest for Colab).
- A fork of the lab-starter repo (see below).

## Key idea (how Colab and GitHub fit together)

Colab is where you run notebooks in the cloud. GitHub is where you store your dissertation artefacts (notebooks, data, saved figures, and short logs).

In this lab, your goal is always:

- run-all notebook → saved outputs → committed to your repo

## Step 1 — Fork lab-starter

Forking creates your own copy of the lab-starter repo under your GitHub account.

1. Open the lab-starter repo page in GitHub.
2. Click Fork (top right).
3. Name your fork (recommended): comp-psych-replication-<your-initials>.
4. Keep it Public unless your supervisor tells you otherwise. If Private, you must add your supervisor/marker as a collaborator.

## Step 2 — Open a notebook in Colab

Use the Open in Colab links in the repo README (recommended).

5. In your fork's README, click the Open in Colab badge for Notebook 01.
6. In Colab, check the notebook title includes your GitHub path (your username, not the lab org).
7. From the Colab menu, choose Runtime → Run all.

What you should see: plots appear below cells, and files are written into the repo folders (data/ and results/).

### The 3-cell 'run ritual' (what most notebooks do first)

Many lab notebooks begin with a small setup block like this:

- 1) Get the code into the Colab runtime (clone the repo)  

```
!git clone https://github.com/<YOUR-USERNAME>/<YOUR-REPO>.git
```

```
%cd <YOUR-REPO>
```
- 2) Install requirements (only if needed)  

```
!pip install -r requirements.txt
```
- 3) Make imports work (so you can import from src/)  

```
import sys
```

```
sys.path.append('.')
```

Do not worry if a given notebook does not include all three steps. The point is: clone → install (if needed) → run.

### Step 3 — Check outputs were created

By the end of each notebook, you should have outputs saved into the repo folders. For the starter demo:

- Notebook 01 should create data/dummy\_bandit.csv and 1–2 plots in results/figures/.
- Notebook 02 should create results/tables/model\_comparison.csv and a predicted\_vs\_observed plot in results/figures/.

In Colab, you can check saved files by clicking the folder icon on the left (Files).

## Step 4 — Save your work back to GitHub

Important: Colab does not automatically push your changes back to GitHub. You must save your notebooks and outputs into your fork.

### Option A (recommended) — Download from Colab, upload to GitHub

8. In Colab: File → Download → Download .ipynb (do this for each notebook you ran).
9. In Colab Files panel: download the outputs you created (CSV and PNG files).
10. In GitHub (your fork): open the target folder (e.g., notebooks/ or results/figures/) and click Add file → Upload files.
11. Commit changes (write a short commit message such as 'Run nb01 and save outputs').

### Option B — 'Save a copy in GitHub' (works for notebooks only)

Colab can save a copy of the notebook back to GitHub, but it does not reliably save extra output files. If you use this option, still upload your CSV/PNG outputs manually.

12. In Colab: File → Save a copy in GitHub.
13. Choose your forked repo and the correct path under notebooks/.
14. Commit with a clear message.

### Option C (advanced) — Push from Colab using git

This is optional. Only use it if you are comfortable with git. It requires a GitHub Personal Access Token (PAT). Never paste your PAT into screenshots or reports.

## Common problems

### 'ModuleNotFoundError' when importing from src/

Make sure you have cloned the repo and changed into it (`%cd <YOUR-REPO>`), then run `sys.path.append('.')`.

### No outputs saved into results/ or data/

Scroll to the end of the notebook and check the save cells ran. If needed, rerun the final 'save outputs' cell(s).

### I changed things but my GitHub repo looks unchanged

You have not uploaded/committed changes yet. Use Option A (download + upload) to commit your notebooks and outputs.

### My repo is private and Colab cannot access it

Either make your fork public for the replication task, or use authentication (advanced). For marking, your supervisor must be able to access the repo.

### The notebook takes too long to run

Restart runtime (Runtime → Restart runtime) and use the default coarse grid settings for fitting.

## Your Week 2 minimum (what supervisors look for)

A supervisor should be able to open your repo and reproduce your outputs quickly. Minimum deliverable:

- Your forked repo link plus a commit hash.
- Two notebooks saved back to notebooks/ (run-all).
- A dataset saved in data/ (dummy or real).
- At least 2 plots saved in results/figures/.
- At least 1 results table (CSV) saved in results/tables/.
- A short reports/reproducibility\_log.md stating what ran, what did not, and what you changed.

## One-minute glossary

**Fork:** A personal copy of a GitHub repo under your account.

**Commit:** A saved snapshot of changes in GitHub (with a message).

**.gitignore:** A file that tells Git which temporary files not to track (helpful for notebooks).

**Colab runtime:** The temporary computer in the cloud that runs your notebook. It resets when disconnected.