# Checklist to validate repo before publishing to GitHub

The process to bring an internal repo to external GitHub requires the authors (lead analysts) to review the code quality and to ensure the lack of sensitive information.

For code quality, we recommend the analysts applying good engineering practices for building their code. Some of these materials can be found in our Data Science [rap-community-of-practice](https://nhsd-git.digital.nhs.uk/data-services/analytics-service/iuod/rap-community-of-practice.git) and NHSD Software Engineering [data-quality-framework](https://github.com/NHSDigital/software-engineering-quality-framework). The below is only concerned with checks to ensure there is no sensitive information present, and that the code is safe to publish externally.

RED Not fit for publishing - requires immediate attention

AMBER Not fit for publishing - will require attention before external review

GREEN Fit for publishing

### Repository to be published on Github: <insert Github repo link here>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Checklist items** | **Internal check** | **Int. Comments & Suggestions** | **Confirm suggestions have been implemented** | **External check** | **Ext. Comments & Suggestions** | **RAG** |
| * Confirm code is fit for purpose   (All outputs replicate what has been published. Code adheres to standards for clarity, commenting, style, etc. Testing applied.) |  |  |  |  |  |  |
| * + Confirm code has been peer reviewed.   + Confirm code is tested and runs in a different environment. |  |  |  |  |  |  |
| * Identify clearly who owns the code and how others can use it |  |  |  |  |  |  |
| * + Obtain approval from the owner of the designed products or services in this repo to publish code.   + Assign the person (or team) with responsibility for ongoing support and communications for the code.   + Make sure the code does not include any unreleased policy or sensitive algorithm(e.g. fraud detection) |  |  |  |  |  |  |
| * Secret & credentials scanning |  |  |  |  |  |  |
| * + Remove all passwords, IP addresses, AWS secret, and identification info.   + Remove Git history to prevent leakage of credentials or secrets in past commits (in the fit-for-publishing process).   + Code should be put through a build test to ensure that it will run in another environment. E.g. promote code to test env, install and run. Or CI pipeline in gitlab using docker image. |  |  |  |  |  |  |
| * Confirm no data stored in the Git repo |  |  |  |  |  |  |
| * + Ensure no Jupyter notebooks exist in the repo.   + Keep any binaries, build artIfacts/package files (e.g. wheel, egg) etc out of the repo.   + Logs   + code comments in the script   + outputs (terminal, files, and database) or no credentials or PII data printed to screen |  |  |  |  |  |  |
| * Documentation |  |  |  |  |  |  |
| * + README should contain clear and key info of the repo ([guide](https://www.makeareadme.com/)).   + Select an [appropriate licence and copyright](https://nhsd-confluence.digital.nhs.uk/display/DAT/DS_216%3A+Select+the+right+license+and+copyright+for+your+project) for the repo.   + Add contact details (e.g. email) so users can request additional information or improvements.   + Share the location of any available synthetic data or the metadata (if possible).   + Link with the corresponding publication report |  |  |  |  |  |  |