**Работа с GitHub в RStudio: Пошаговое руководство**

Использование Git и GitHub позволит вам отслеживать все изменения в файлах проекта (скриптах, данных, документации), легко возвращаться к предыдущим версиям, а также, в будущем, делиться кодом или работать над ним совместно с кем-то еще.

**Шаг 1: Первоначальная настройка (выполняется один раз)**

Прежде чем мы начнем, нужно убедиться, что все необходимые инструменты установлены и настроены.

1. **Установка Git:** Если Git еще не установлен на вашем компьютере, его необходимо скачать и установить с официального сайта: <https://git-scm.com/downloads>. В процессе установки просто следуйте инструкциям, оставляя настройки по умолчанию.
2. **Создание аккаунта на GitHub:** Если у вас его еще нет, зарегистрируйтесь на <https://github.com>. Это бесплатно.
3. **Настройка Git в RStudio:**
   * Откройте RStudio.
   * Перейдите в меню Tools -> Global Options....
   * В открывшемся окне выберите раздел Git/SVN.
   * Убедитесь, что RStudio "видит" установленный Git. В поле Git executable: должен быть указан путь к git.exe. Если поле пустое, нажмите Browse... и найдите его (обычно он находится в C:/Program Files/Git/bin/git.exe).
   * Нажмите OK.
4. **Настройка вашего имени и email в Git:** Это важно, чтобы GitHub знал, кто вносит изменения.
   * Откройте RStudio.
   * Перейдите во вкладку Terminal (обычно она находится рядом с Console).
   * Введите и выполните поочередно две команды, заменив "Ваше Имя" и "ваш@email.com" на те, что вы использовали при регистрации на GitHub:

Bash

git config --global user.name "Ваше Имя"

git config --global user.email "ваш@email.com"

**Шаг 2: Создание проекта в RStudio, связанного с GitHub**

Теперь давайте создадим новый RStudio-проект на основе репозитория (хранилища) на GitHub.

1. **Создание нового репозитория на GitHub:**
   * Зайдите в свой аккаунт на GitHub.
   * Нажмите на значок + в правом верхнем углу и выберите New repository.
   * **Repository name:** Дайте вашему проекту понятное имя, например, hotspots-analysis-voronezh.
   * **Description:** Добавьте краткое описание, например, "Анализ влияния климатических и антропогенных факторов на возникновение природных пожаров в Воронежской области".
   * **Public/Private:** Выберите Public, если хотите, чтобы проект был виден всем, или Private, если он только для вас.
   * **ВАЖНО:** Поставьте галочку напротив Add a README file. Это создаст главный файл с описанием проекта.
   * Нажмите Create repository.
2. **Клонирование репозитория в RStudio:**
   * На странице вашего нового репозитория на GitHub нажмите на зеленую кнопку <> Code.
   * Убедитесь, что выбрана вкладка HTTPS, и скопируйте URL-адрес (он будет выглядеть примерно так: https://github.com/ваше\_имя/hotspots-analysis-voronezh.git).
   * Теперь вернитесь в RStudio.
   * Перейдите в меню File -> New Project....
   * В окне выберите Version Control.
   * Затем выберите Git.
   * В поле Repository URL: вставьте скопированный адрес с GitHub.
   * В поле Project directory name: RStudio автоматически подставит имя вашего репозитория.
   * В поле Create project as subdirectory of: выберите папку на вашем компьютере, где будет храниться проект. **Рекомендую выбрать новую, пустую папку, например, B:/R\_Projects/, чтобы RStudio создал внутри нее папку hotspots-analysis-voronezh**.
   * Нажмите Create Project.

RStudio автоматически скачает (клонирует) ваш репозиторий с GitHub, и вы увидите в окне Files файл README.md и .gitignore. Поздравляю, ваш проект теперь связан с GitHub!

**Шаг 3: Рабочий процесс "Коммит и Пуш"**

Теперь, когда вы работаете над проектом, ваш основной цикл действий будет таким: **поработали -> сохранили изменения (коммит) -> отправили на GitHub (пуш)**.

1. **Внесение изменений:**
   * Скопируйте ваши существующие скрипты и данные в папку проекта, которую вы только что создали (B:/R\_Projects/hotspots-analysis-voronezh). Можете организовать их по подпапкам, например, scripts/, data/, docs/.
   * Откройте один из скриптов и поработайте с ним. Сохраните файл (Ctrl+S).
2. **Отслеживание изменений (Commit):**
   * В RStudio откройте вкладку Git (она находится в правом верхнем окне, рядом с Environment и History).
   * Вы увидите список всех новых или измененных файлов.
   * Поставьте галочки (Stage) напротив тех файлов, изменения в которых вы хотите сохранить. Это могут быть один или несколько связанных файлов.
   * Нажмите кнопку Commit.
   * В открывшемся окне в поле Commit message напишите краткое, но понятное описание того, **что именно** вы сделали. Например: "Добавлен скрипт предварительной обработки данных OSM" или "Исправлена ошибка в расчете лагов для осадков". Хорошие комментарии — залог порядка в проекте.
   * Нажмите кнопку Commit. Вы только что "сфотографировали" состояние вашего проекта.
3. **Отправка изменений на GitHub (Push):**
   * После того как вы сделали коммит, нажмите на зеленую стрелку вверх — Push.
   * RStudio отправит все ваши "сфотографированные" коммиты в ваш репозиторий на GitHub. Если вы зайдете на страницу вашего проекта на GitHub, вы увидите там все загруженные файлы и историю коммитов.

**Рекомендации по документации**

Я внимательно ознакомился с вашей диссертацией и описаниями скриптов. Это отличная основа! Давайте используем ее для создания качественной документации прямо в проекте.

* **README.md:** Это главный файл вашего проекта. Отредактируйте его прямо в RStudio. В нем стоит описать:
  + **Цель проекта:** Кратко, о чем этот анализ.
  + **Структура проекта:** Опишите, что лежит в каждой папке (scripts, data, output и т.д.).
  + **Порядок запуска скриптов:** Укажите, в какой последовательности нужно запускать скрипты для получения финального результата (например, 1\_prepare\_data.R -> 2\_monthly\_raster\_generation.R и т.д.).
  + **Источники данных:** Укажите, откуда были взяты данные (FIRMS, ERA5, OSM), как это сделано в диссертации.
* **Комментирование кода:**
  + Внутри каждого R-скрипта используйте комментарии (#) для объяснения логики.
  + Перед каждой логической секцией (как у вас в описаниях) или перед каждой функцией добавляйте блок комментариев, объясняющий, **что делает** этот блок кода.