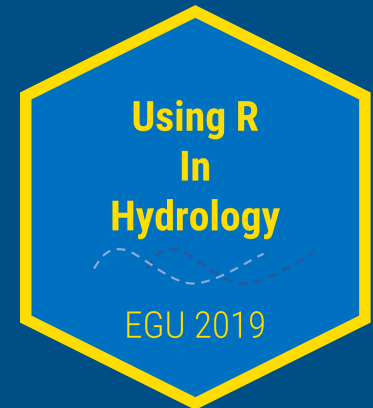


# Staying up-to date: automating tasks from downloading data to reporting

Alex Hurley  
*University of Birmingham*

🏠 [aglhurley.rbind.io](http://aglhurley.rbind.io)

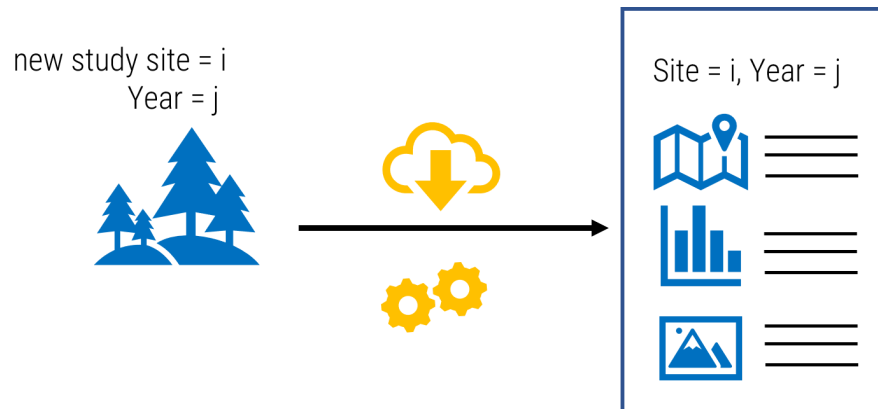
🐦 [aglhurley](https://twitter.com/aglhurley)



# Goals

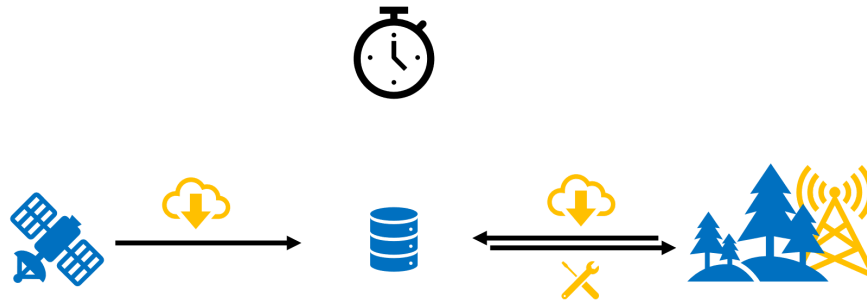
- ⚡ Discuss why and what to automate
- ♻️ Introduce task automation concepts
- 🔧 Highlight tools for each
- 💡 Use-case: Study Site Explorer with [R](#) and [TravisCI](#)

## Study Site Explorer?



# What and why?

- 🔄 recurring tasks: database updates, time series, ancillary data
- ⌚ time-consuming tasks: QA + QC, updating/creating reports
- 🔧 automated testing (package development)



Download on schedule  
Field station broken? Bring tools on next trip

# Concepts

## Storage

**local** 🖥️:

- software / triggers,
  - routines (e.g. R scripts),
  - outputs
- are **on your computer**

vs.

**hosted** ☁️:

- routines and outputs in a repository
  - software on a virtual machine
  - webservice schedules / triggers (webhooks)
- download** or view online

# Concepts

## Execution / Trigger

↺↻ recurring + ⌚ scheduled

vs.

⚡ event-based (e.g. on file change)

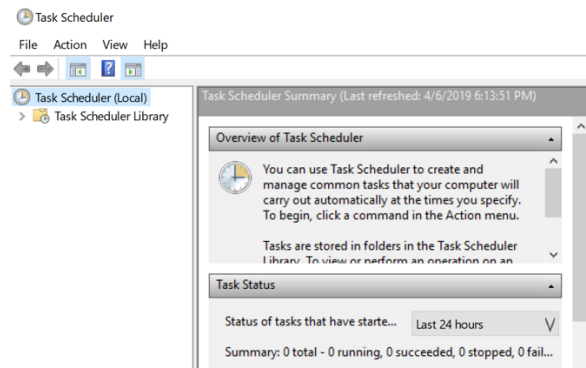
# Tools

# Tools

## Scheduling (local)

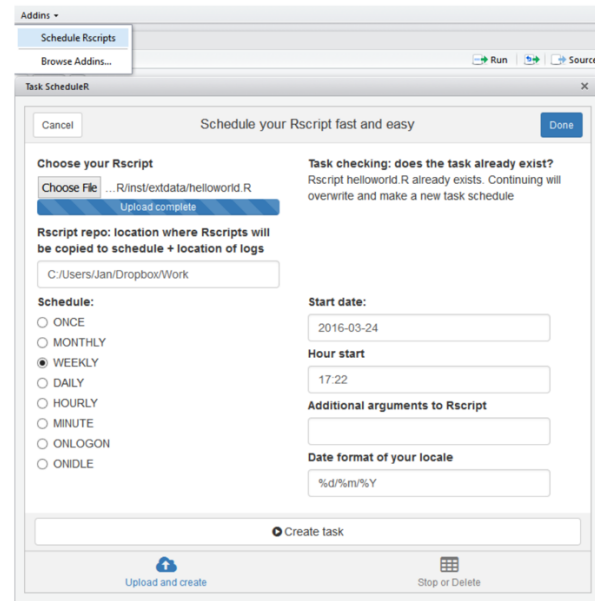
 taskschedulerR

Windows OS



taskschedulerR

```
install.packages('data.table')
install.packages('knitr')
install.packages('miniUI')
install.packages('shiny')
install.packages("taskschedulerR", repos = "http://www.datatailor.be/rcube", type = "source")
```



# Tools

## Scheduling (local)

🔔 Linux via shell scripts:

- 🕒 **cron** for recurring tasks
- 📅 **at** for one-of tasks



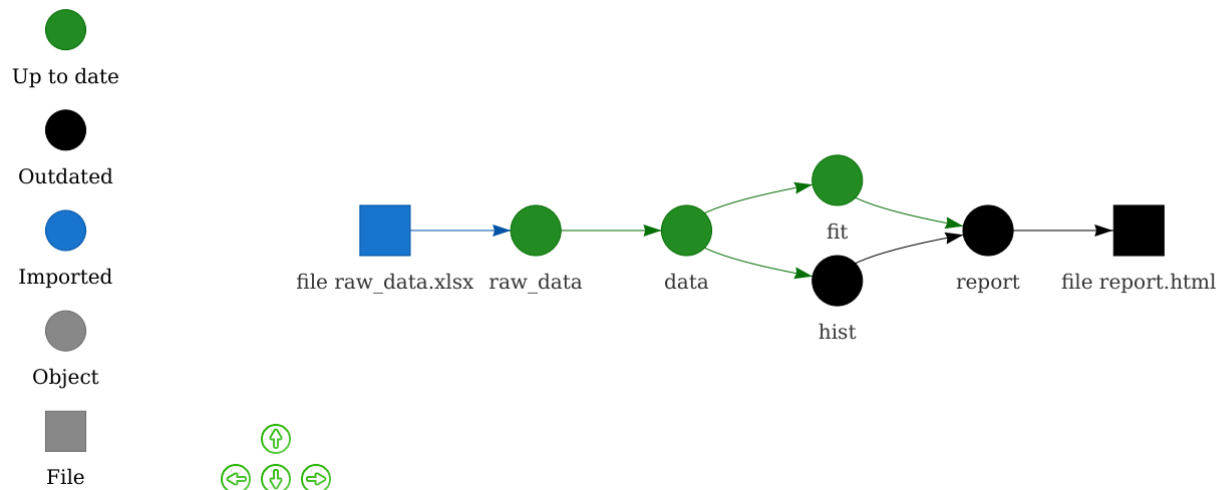
# Tools

## Event-based

rOpenSci  drake:

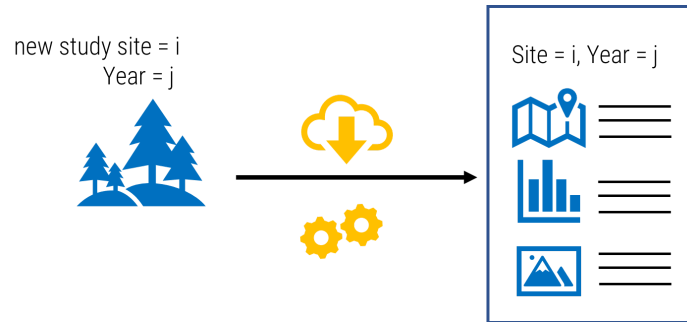
- semi-automated workflow manager
- monitors individual units/sections of analyses pipeline
- updates on change,
- but only parts of pipeline that require re-running

**Dependency graph**



# Study Site Explorer

# Study Site Explorer

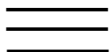
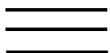
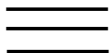


- R Markdown report with:
  - map
  - annual Precipitation + Temperature
  - 3D view of region
- Hosted online (collaborators can generate report)
- Triggered on file change (add new site)

# Study Site Explorer

Use parameterized report to define meta data!

Site = i, Year = j



```
---  
author: "rHydro Demonstrator"  
date: '`r paste("generated at:", Sys.Date())`'  
output: html_document  
params:  
  location: "Mt St Helens"  
  year:  
    value: 2000  
title: "`r paste('Overview for:', params$location)``"  
---
```

# Study Site Explorer

## Define area of interest and make map:

```
# specify area of interest
aoi <- AOI::getAOI(clip = list(params$location,15 ,15), km = TRUE)

# pull and plot map
map <- OpenStreetMap::openmap(upperLeft = c(aoi@bbox[[4]], aoi@bbox[[1]]),
                              lowerRight = c(aoi@bbox[[2]], aoi@bbox[[3]]),
                              type = "osm",
                              minNumTiles = 12)
```

# Study Site Explorer

## Download climate data and plot:

```
daymet_data <- daymetr::download_daymet(lat = aoi@polygons[[1]]@labpt[2],
                                       lon = aoi@polygons[[1]]@labpt[1],
                                       start = params$year,
                                       end = params$year)

daymet_data$data <- dplyr::mutate(daymet_data$data,
                                tmean = (tmax..deg.c. + tmin..deg.c.)/2,
                                date = as.Date(paste(year, yday, sep = "-"),
                                                "%Y-%j"))

library(ggplot2)
ggplot(daymet_data$data,
       aes(x = date,
           y = prcp..mm.day.)) +

  geom_col(position = "dodge",
           color = "darkblue") +

  labs(x = "Date",
       y = "P (mm/day)",
       title = paste0(params$location, ": ",
                      params$year,
                      " - Precipitation")) +

  theme_bw()
```

# Study Site Explorer

## 3D-Viz of Site (full code available [here](#))

```
ned_aoi <- aoi %>% HydroData::findNED() # National DEM

# convert to matrix for rayshader
ned <- matrix(raster::extract(ned_aoi$NED,
                             raster::extent(ned_aoi$NED),
                             buffer=1000),
             nrow=ncol(ned_aoi$NED), ncol=nrow(ned_aoi$NED))

overlay <- create_overlay(prcp_raster, ned_aoi$NED)

library(rayshader)
ned %>%
  sphere_shade(texture = "imhof1") %>%
  add_water(detect_water(ned), color="desert") %>%
  add_overlay(overlay, alphacolor = NULL, alphaslayer = 0.8) %>%
  add_shadow(ray_shade(ned)) %>%
  add_shadow(ambient_shade(ned)) %>%
  plot_3d(heightmap = ned ,
          zscale = 1,
          # fov = 90,
          lineantialias = TRUE,
          theta = 15,
          phi = 85,
          zoom = 0.3)
render_snapshot()
rgl::rgl.close()
```

# Study Site Explorer

## Generate report

```
' Render Location and Climate report
#'
#' @param location Character, location passed to AOI::getAOI()
#' @param year Character, Year in YYYY
#'
#' @return Returns nothing, but writes a file to the reports directory
#' @export
#'
#' @examples
render_report = function(location, year) {
  # house keeping on names
  location_dir <- stringr::str_replace_all(location, pattern = "[.]", "-")
  location_dir <- stringr::str_replace_all(location_dir, pattern = "[ ]", "_")

  rmarkdown::render(
    "./templates/report_template.Rmd", <
    params = list(
      location = location,
      year = year
    ),
    output_dir = "./reports",
    output_file = paste0("Report-", location_dir, "-", year, ".html")
  )
}
```



# Study Site Explorer

## Overview for: Mt St Helens

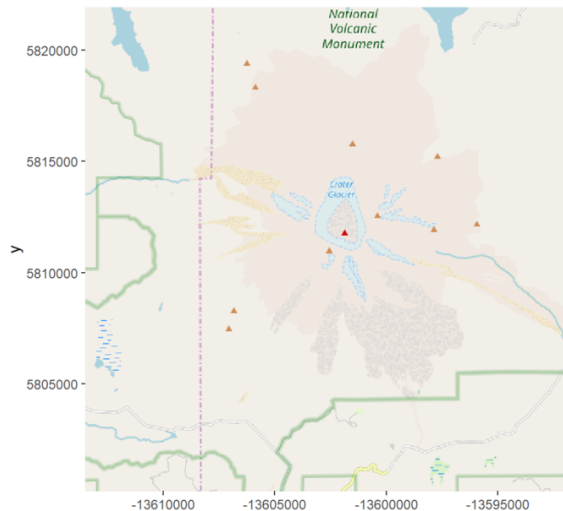
rHydro Demonstrator

generated at: 2019-04-06

## Overview

This report is for Mt St Helens and was generated for 2010.

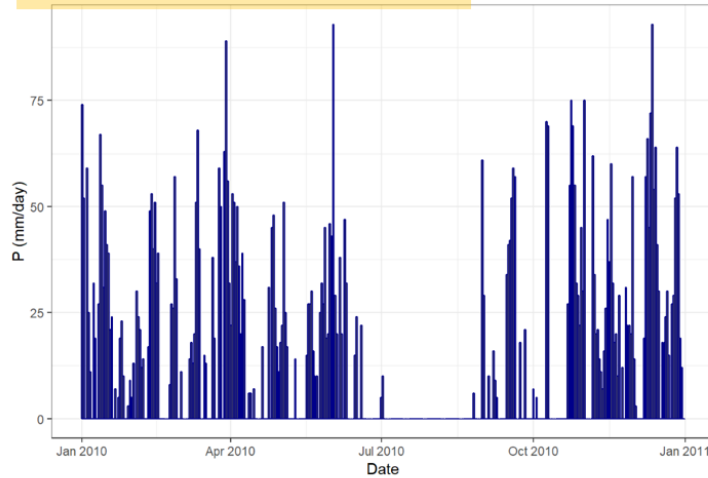
## Area overview



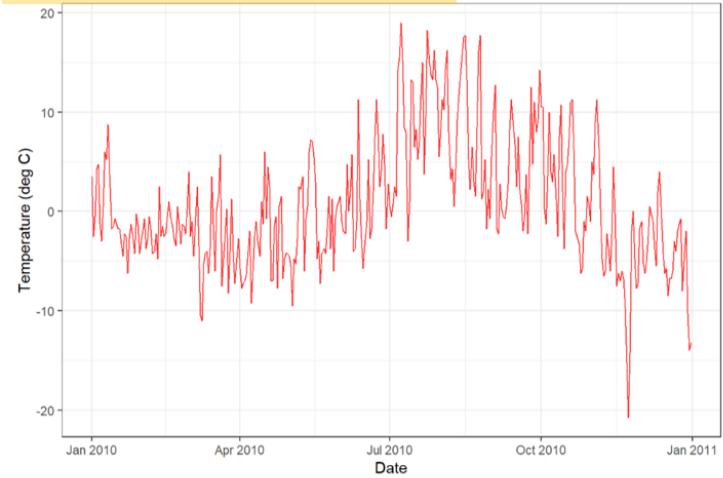
# Study Site Explorer

## Climate data

Mt St Helens: 2010 - Precipitation



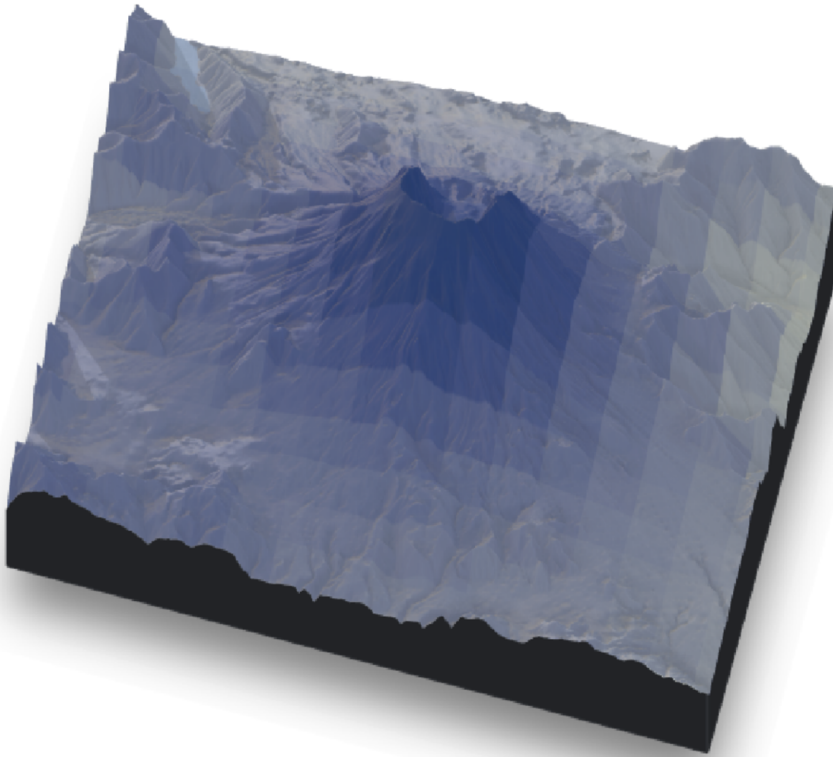
Mt St Helens: 2010 - Temperature



Annual total P for 2010 was: 6038 mm.

Annual mean T for 2010 was: 0.7582192 degC.

# Study Site Explorer



# Study Site Explorer

Generate report for several sites:

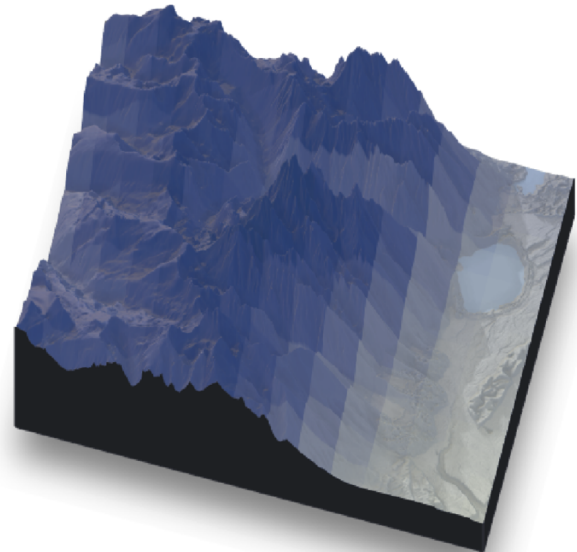
```
## Script executes report generation for all locations listed in sites.txt
source("R/render_report.R")

sites <- read.csv("sites.csv", header = TRUE, stringsAsFactors = FALSE)
for(site in sites$sites){
  render_report(site, year = 2010)
}
```

# Study Site Explorer

Generate report for several sites:

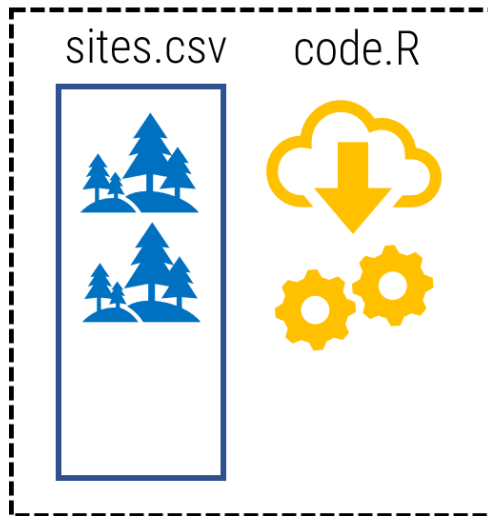
```
+-- reports
|   +-- Report-Mt_Baldy-2010.html
|   +-- Report-Mt_St_Helens-2010.html
|   +-- Report-Grand_Teton_Mountain-2010.html
|   \-- Report-El_Capitan_Yosemite-2010.html
```



# Study Site Explorer

Continuous integration with Travis

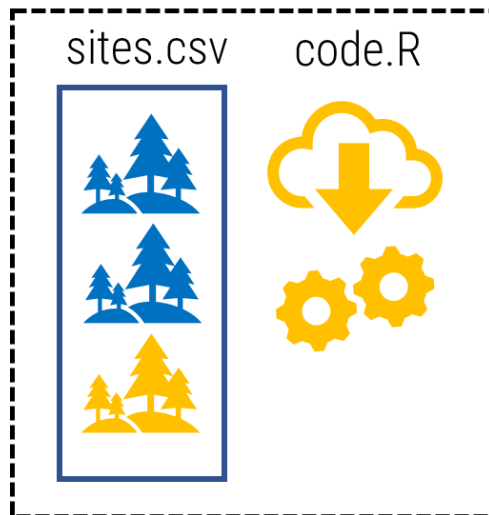
**GitHub**



# Study Site Explorer

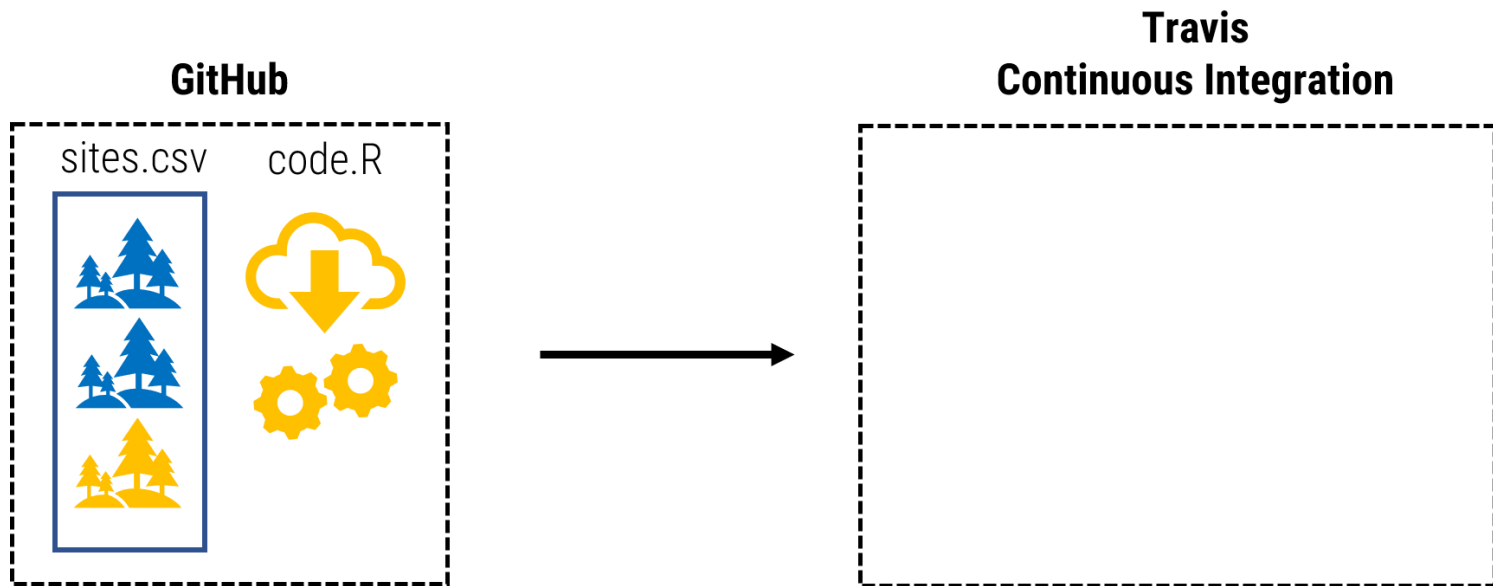
Continuous integration with Travis

**GitHub**



# Study Site Explorer

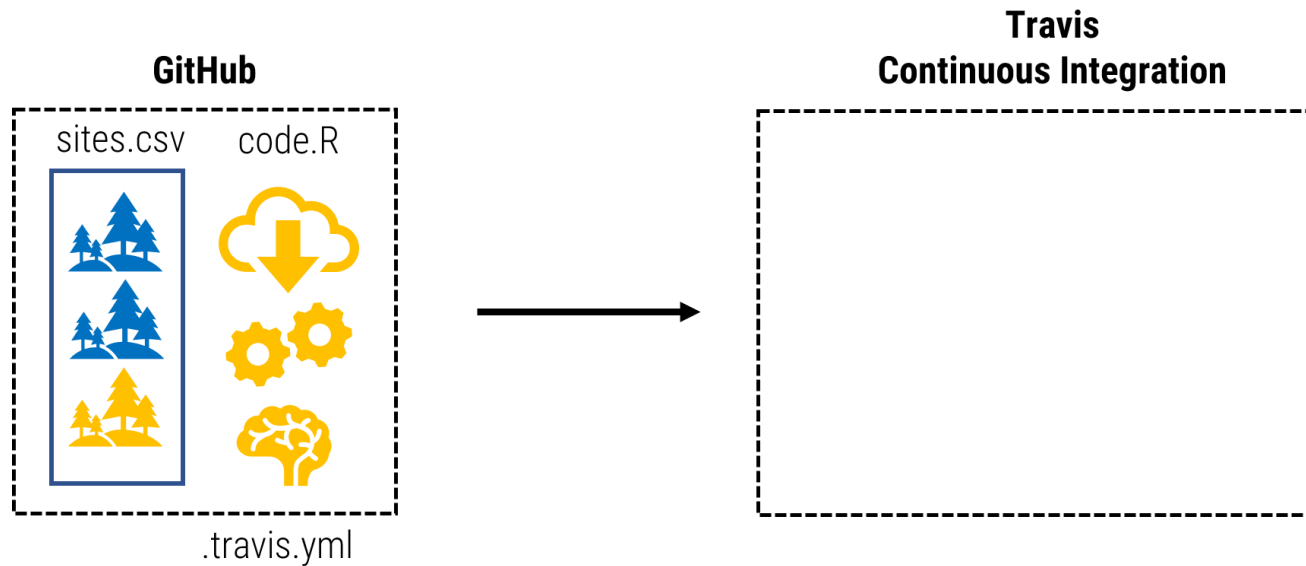
## Continuous integration with Travis





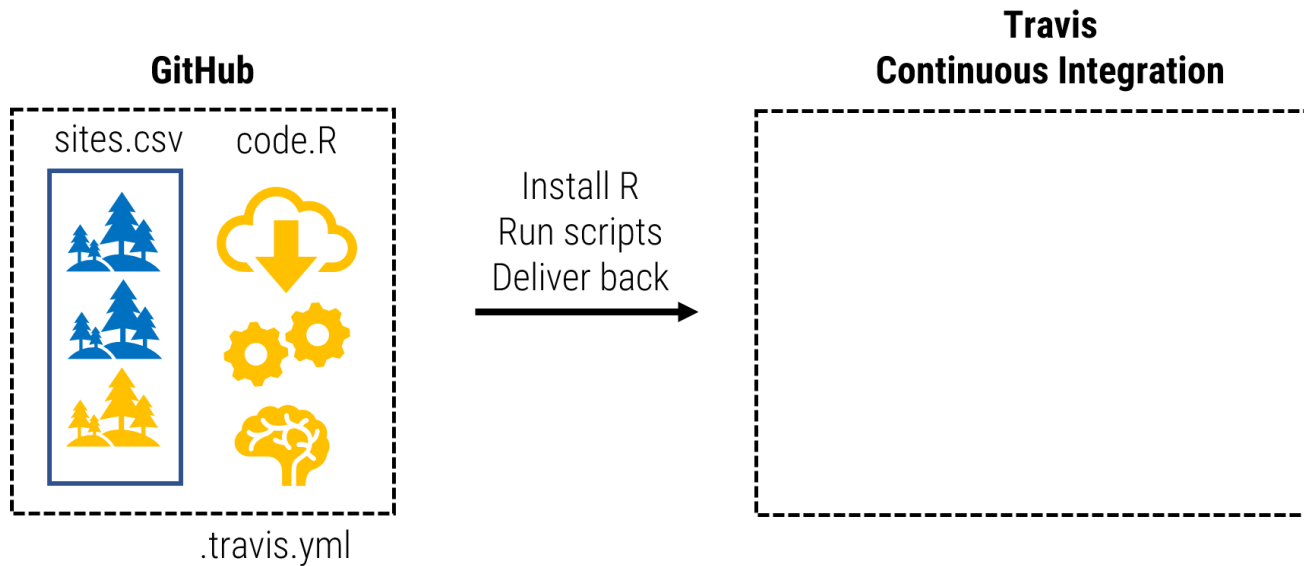
# Study Site Explorer

## Continuous integration with Travis



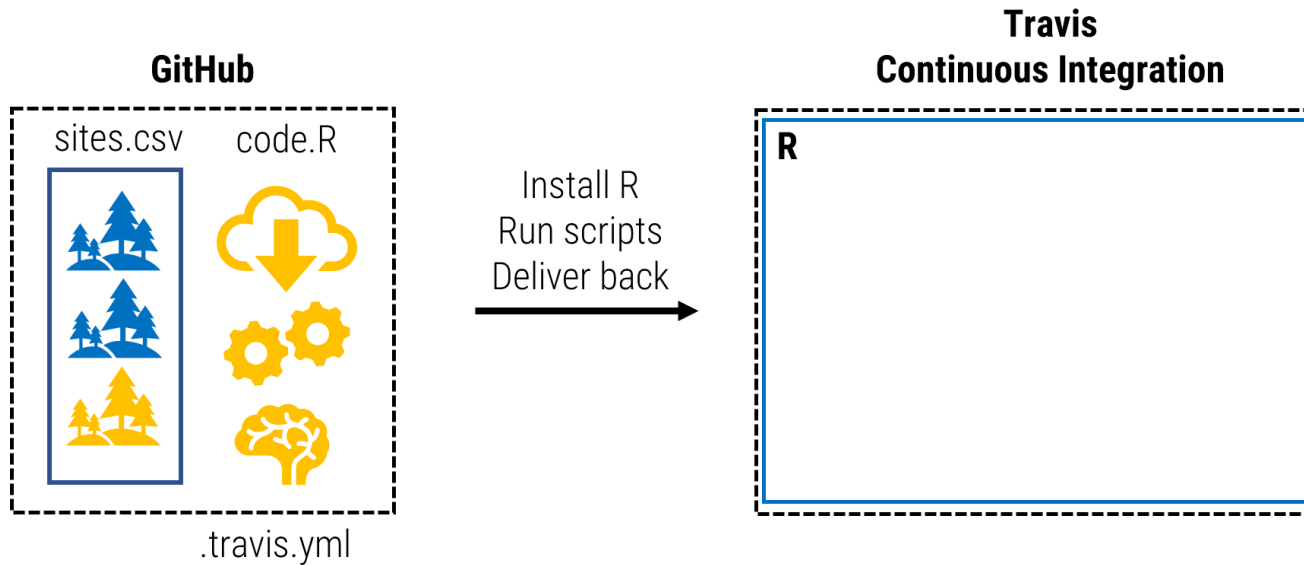
# Study Site Explorer

## Continuous integration with Travis



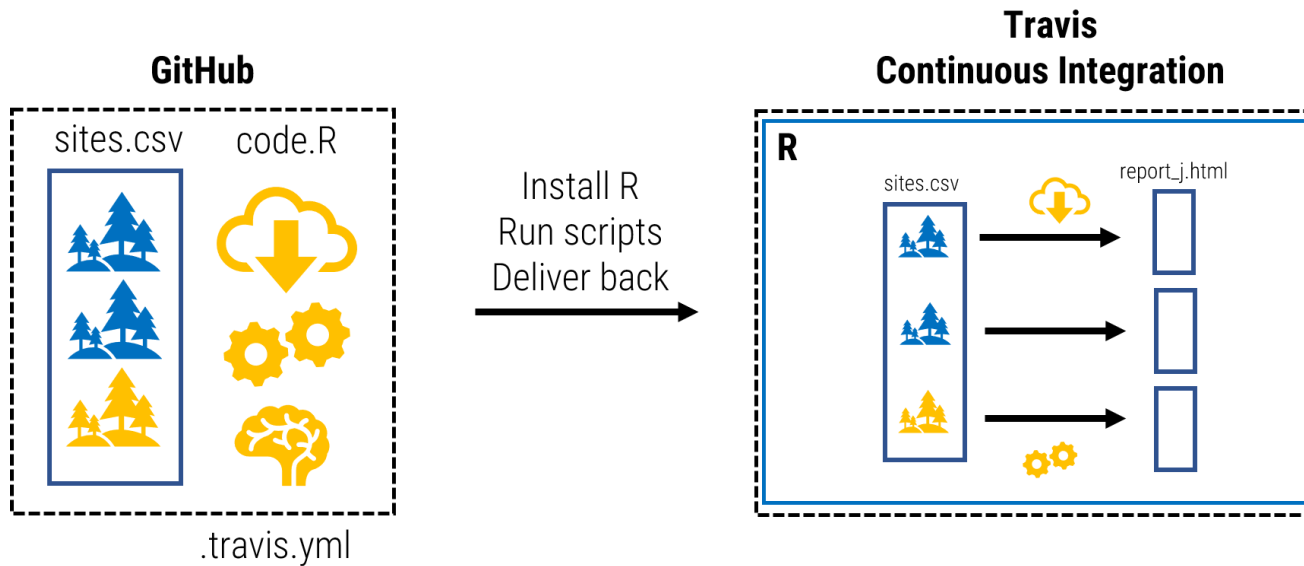
# Study Site Explorer

## Continuous integration with Travis



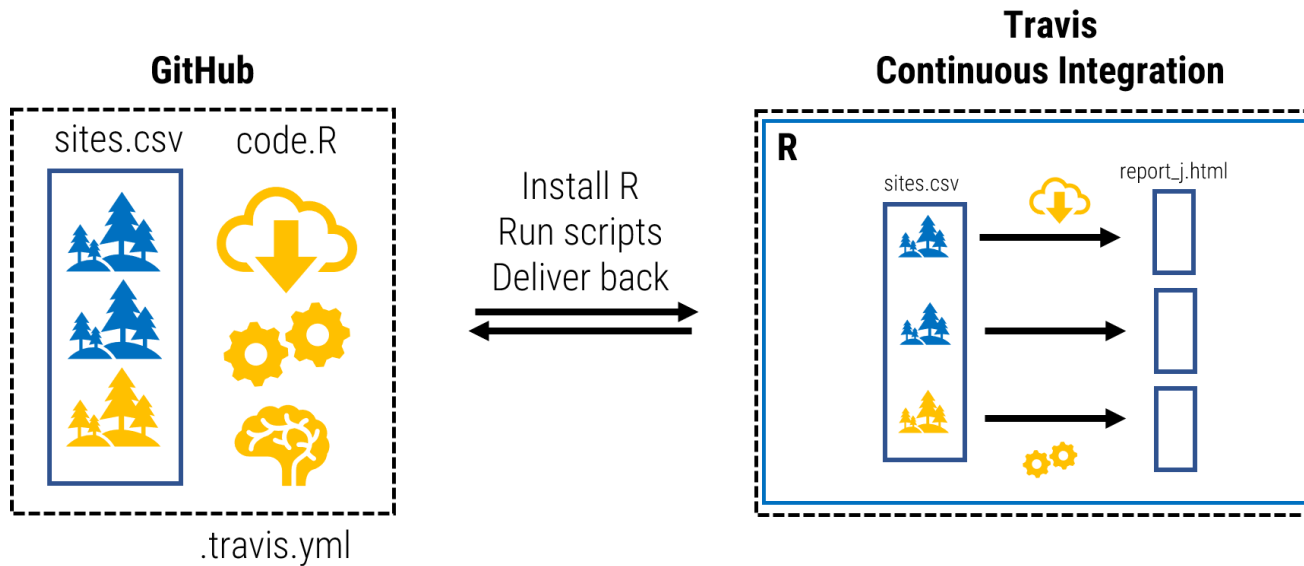
# Study Site Explorer

## Continuous integration with Travis



# Study Site Explorer

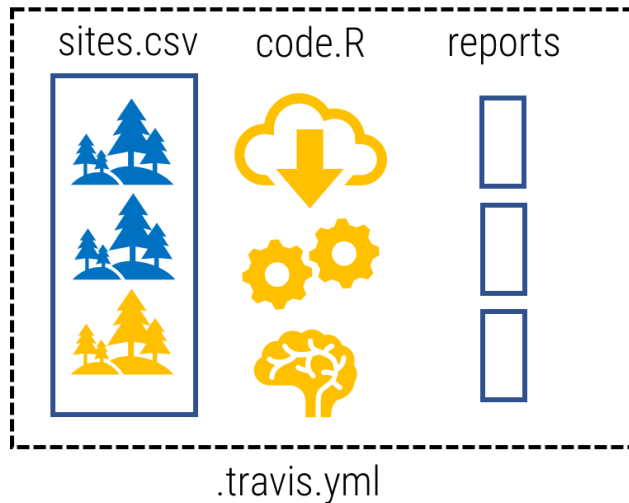
## Continuous integration with Travis



# Study Site Explorer

## Continuous integration with Travis

### GitHub



```
sudo: required
language: r
cran: https://cran.rstudio.com/
cache: packages
before_install:
after_success:
install:
  - Rscript install_packages.R
script:
  - Rscript generate_parameter_reports.
  - Rscript push_back.R
env:
  global:
    secure: <GitHub_Access_Token>
```

# Summary:

## Notes:

- Requires GitHub account
- Linked to TravisCI (CircleCI as alternative)
- Build a config `.travis.yml`
- See in action: [github.com/the-hull/02\\_task\\_automation](https://github.com/the-hull/02_task_automation)

## Task Automation

- Useful locally or hosted
- Frees up time
- Builds up and checks data sets
- When hosted, allows collaborators to produce standardized outputs