
○

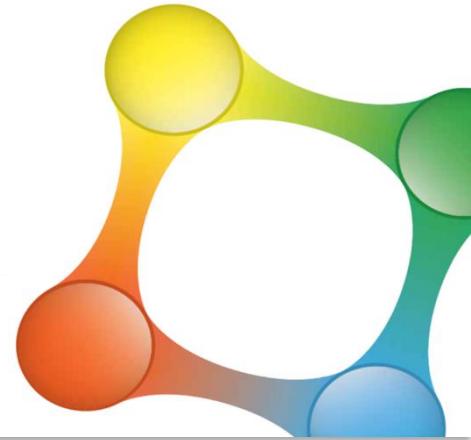
Training Data Research

Gemeente Weert

1. Data Research Toolbox

Versie 0.1 (In bewerking)

Claire Phillippeau



Onderwerpen

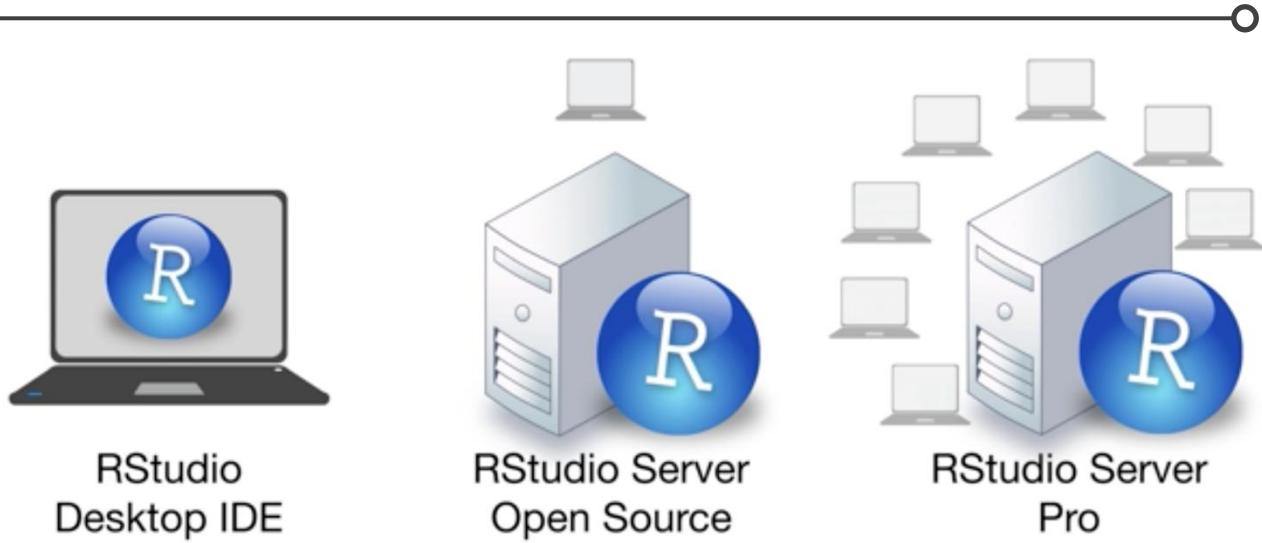
- Wat is R?
- RStudio tour
- R Packages en libraries
- Versiebeheer van scripts
 - Git
 - GitHub
 - Git binnen RStudio
 - RStudio koppelen met GitHub.com
- R leren met Swirl

Overview

- R is een open-source (gratis) programmeertaal die gebruikt wordt voor statistische berekeningen en het bewerken en visualiseren van data.
- Het is een simpele, meer gebruiksvriendelijke versie van de statistische programmeertaal ‘S’
- R wordt (net zoals Python) veel gebruikt in Data Science en voor data gerelateerde projecten.
- R is een ‘interpreter’. Elke regel wordt los door R geïnterpreteerd en uitgevoerd.
- Er hoeft dus ‘gecompileerd’ te worden. Hierbij wordt een compleet programma eerst overgezet naar een ‘machine-taal’ en dan pas door de computer uitgevoerd.

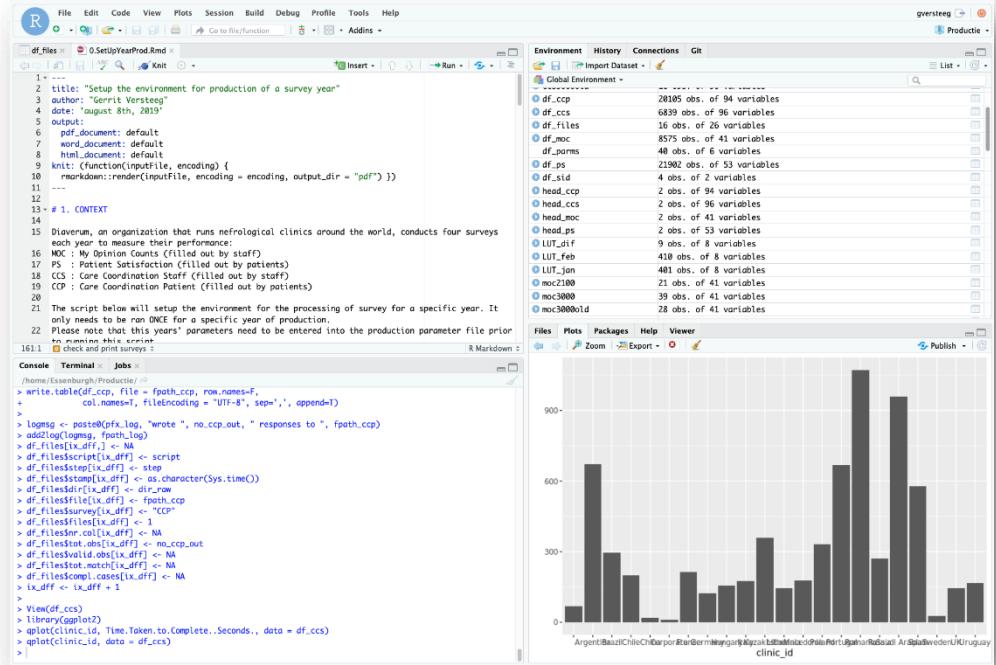


Overview



- Security
- Load Balancing
- Administrative tools
- Resource management
- Metrics and monitoring
- Multiple sessions
- Collaborative editing
- Easy R versioning
- Audit history

RStudio Server Overview

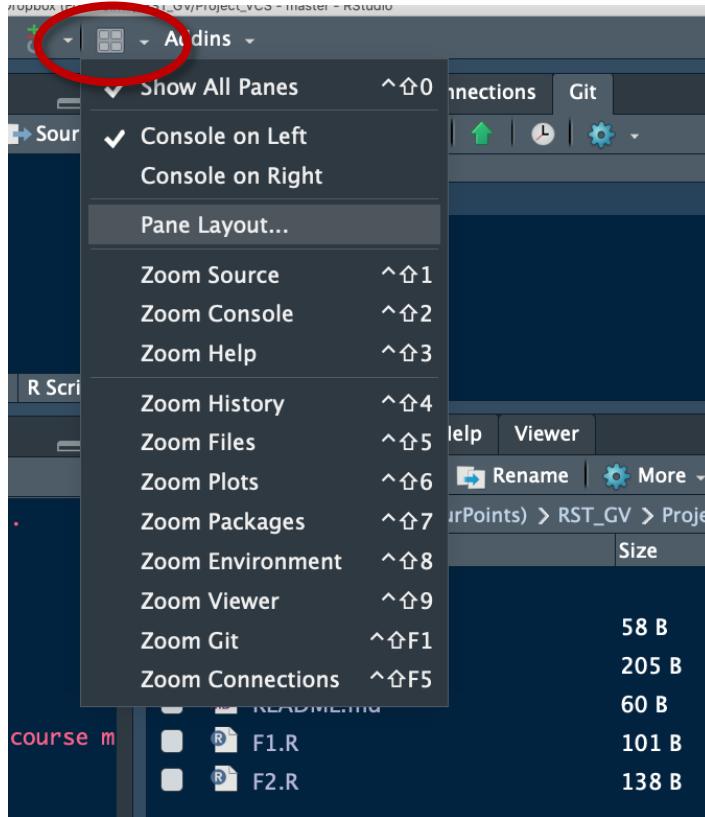


The screenshot shows the RStudio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Code Editor:** Shows a script named `SetupProd.Rmd` containing R code for setting up a survey environment, including file paths and data frame manipulations.
- Environment View:** Shows the global environment with various data frames like `df_ccp`, `df_moc`, and `df_ps`, each with their respective observation counts and variable counts.
- Plot Viewer:** Displays a histogram of clinic IDs, with the x-axis labeled "clinic_id" and the y-axis ranging from 0 to 900+.

- RStudio is een Integrated Development Environment (IDE).
- RStudio is ingericht om het schrijven van R-code te vergemakkelijken.
- Gedurende het programmeren krijg je suggesties rond functies en objecten die je mogelijk kan gebruiken ('context-aware suggestions').
- Syntax-fouten worden gesigneerd.
- Het scherm is standaard zo ingedeeld, dat je alle belangrijke zaken (script, console, variabelen, files, plots, etc.) overzichtelijk naast elkaar hebt staan.

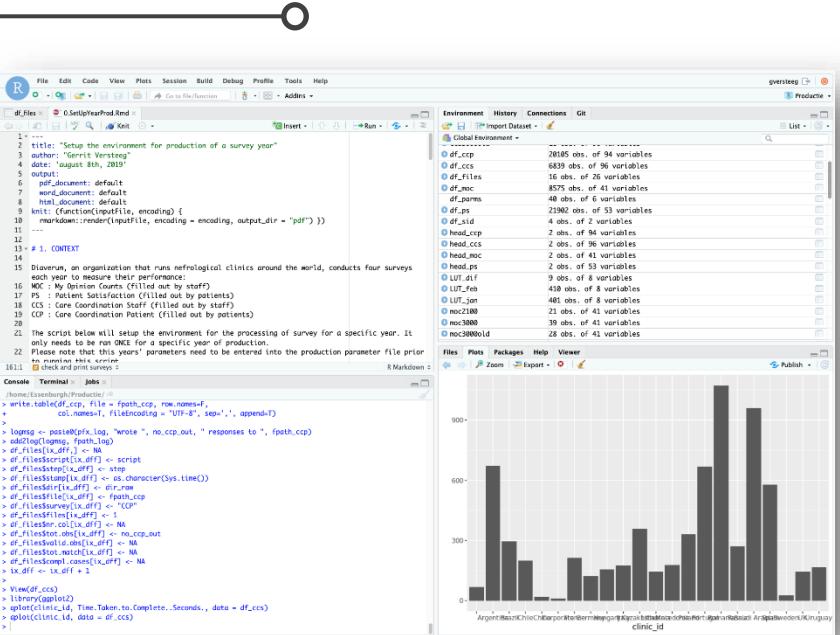
Tip: Shortcuts naar de tabs



- Tijdens het werken met RStudio, zul je vaak in verschillende 'panels' zijn (denk aan Console, Editor, View, Help, Files, etc.)
- Meestal gebruik je je keyboard in zo'n panel (Console, Editor, Help)
- In de praktijk komt het regelmatig voor dat je denkt dat je in een bepaald panel staat (bijv. Console), terwijl je in werkelijkheid in een ander panel staat (bijv. Editor).
- Een simpele klik in het panel is voldoende, maar dan moet je overschakelen naar je muis.
- Er zijn echter ook keyboard-shortcuts om van panel naar panel te gaan, zodat je niet naar je muis hoeft.

Demo: RStudio Tour

- Inloggen op RStudio Server Pro
- Opbouw met vier windows met tabs
- Korte uitleg van elke tab, langer stilstaan bij: script window, console, files, global environment, history, plots, help
- Demo van gebruik tijdens programmeren (een kort script schrijven). Uitleg bij de opkomende suggesties.
- Belangrijkste menu-items behandelen
- Projecten, working-directory, sessies



The screenshot shows the RStudio interface with several windows open:

- Script Window:** Displays R code for setting up survey environments and reading data from CSV files.
- Terminal Window:** Shows the command used to run the script.
- Global Environment:** Lists various objects and their characteristics (e.g., df_ccp, 20105 obs. of 94 variables).
- History:** Shows a list of previous commands entered.
- Plots:** A histogram showing the distribution of data across different categories.
- Files:** Shows the project structure and files.
- Packages:** Lists available packages.
- Help:** Provides help documentation for various functions.
- Viewer:** Shows the output of the histogram.

Handige cheatsheet(s)

RStudio IDE : CHEAT SHEET

Documents and Apps

Open Shiny, R Markdown, Latex, Rd files and more in Source pane

Check spelling, Render output, Choose output format, Choose output location, Insert code chunk

Jump to line, Run selected lines, Publish to server, Show file outline, Access markdown guide at Help > Markdown Quick Reference

Jump to chunk options, Set active chunk, Run this and all previous code chunks, Run this code chunk, Jump to its function in file, Change file type, Open or publish to view app, Publish to server or browser, Manage pushback accounts

RStudio mechanisms that files named `app.R`, `server.R`, `ui.R`, and `global.R` belong to a shiny App

Run app, Choose location to view app, Publish to shinyapps.io or browser, Manage pushback accounts

Write Code

Navigate tabs, Open in new window, Save, Find and replace, Compile as notebook, Run selected code

Import data with wizard, History of past commands to run history, Display R Help sidebar, File > New File > R Presentation

R Support

Good Start... (Current user), Load workspace, Save workspace with or without echo, Close file outline, Create all saved objects, Search code environments, Close environment to display from list of pasted environments, Load workspace, Save workspace, Close all saved objects, Display objects as list or grid

Code diagnostics that appear in the margin, Diagnose code diagnostic symbols for details, Syntax highlighting based on your R editor, Tab completion for blocks of code, Multi-language code editor, Expressions for quickly use different blocks of code

Change file type, Open or publish to view app, Publish to server or browser, Manage pushback accounts

Launch debugger mode when it encounters a breakpoint while executing code, Click next to line number to add/remove a breakpoint, Highlight line where execution has paused, Run commands in environment where execution has paused, Examine variables in executing environment, Select function in traceback to debug

Step through code one line at a time, Step into and out of functions to run, Resume execution, Quit debug mode

Version Control

Turn-on at Tools > Project Options > Git/SVN

Staged files, Show file, Commit, Push/Pull, View history, Add, Deleted, Modified, Untracked

Open traceback to explore the function call stack before the error occurred

Click to load package with `library()`, Unstick to detach package with `detach()`, Package version installed, Delete from library

Package Writing

File > New Project > New Directory > R Package, Turn project into package, Export R package documentation results, Tools > Project Options > Build Tools, Request code at Help > Roxygen Quick Reference

View(`data`) opens spreadsheet-like view of data set

9/29/2020

Allerlei handige cheatsheets rond R, worden beschikbaar gesteld op de RStudio-site:

<https://rstudio.com/resources/cheatsheets/>

R-bloggers Stackoverflow blogrstudio



GEMEENTI

W E E R T

R-bloggers

R news and tutorials contributed by hundreds of R bloggers

HOME ABOUT RSS ADD YOUR BLOG! LEARN R R JOBS CONTACT US

Tutorials for learning R

Posted on December 10, 2015 by Tal Galil in R guest posts | 0 Comments

For example, if you run `install.packages("gridExtra")` and then `gridExtra::grid.arrange()`, you will see this nicely arranged grid:

```
gridExtra::grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt)
```

Or if you run `gridExtra::grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb)`, you'll get this:

```
gridExtra::grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb)
```

Now, let's say you want to add some styling to the plot. You can do this by adding `gridExtra::grid.arrange(..., plot_bg = "red")`. This will change the background color of all plots to red. If you want to change the background color of only one plot, you can do this by adding `gridExtra::grid.arrange(..., plot_bg = "red", ..., plot_bg = "blue", ..., plot_bg = "green", ..., plot_bg = "yellow")`. The last argument is the current value, so it's changing along.

Instructions

- ① Import the dataset of the first two columns on the right. Hint: the `mtcars` dataset has 32 rows and 10 columns. You can either import it by changing the file path or using the `read_csv()` function.
- ② Create a scatter plot of the first two columns. Hint: use `mtcars %>% select(-c(3:10)) %>% ggplot(aes(x = mpg, y = cyl)) + geom_point()`. Once you have the plot, `grid.arrange` it with the first two columns. Hint: use `gridExtra::grid.arrange(mtcars$mpg, mtcars$cyl, plot_bg = "red")`.
- ③ Add labels to the x-axis and y-axis. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, plot_bg = "red", labels = c("Miles per gallon", "Cylinders"))`.
- ④ Add a title to the plot. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, main = "Cars Data")`.
- ⑤ Add a legend to the plot. Hint: follow the code in the `grid.arrange` section above. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, legend = TRUE)`.
- ⑥ Add the last two columns on the right. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, mtcars$vs, mtcars$am, plot_bg = "red")`.
- ⑦ Add labels to the x-axis and y-axis. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, mtcars$vs, mtcars$am, labels = c("Miles per gallon", "Cylinders", "Horsepower", "Weight", "VS", "AM"))`.
- ⑧ Add a title to the plot. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, mtcars$vs, mtcars$am, main = "Cars Data")`.
- ⑨ Add a legend to the plot. Hint: follow the code in the `grid.arrange` section above. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, mtcars$vs, mtcars$am, legend = TRUE)`.
- ⑩ Add the last two columns on the right. Hint: use `grid.arrange(mtcars$mpg, mtcars$cyl, mtcars$hp, mtcars$wt, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, mtcars$vs, mtcars$am, mtcars$gear, mtcars$carb, plot_bg = "red")`.

Share **Tweet**

[social4i size="large" align="float-right"]

There are tons of resources to help you learn the different aspects of R, and as a beginner this can be overwhelming. It's also a dynamic language and rapidly changing, so it's important to keep up with the latest tools and technologies.

Search R-bloggers.. **Go**

Your e-mail here **Subscribe** 52793 readers

Follow @rbloggers 87.1K followers

R bloggers Like Page 79K likes

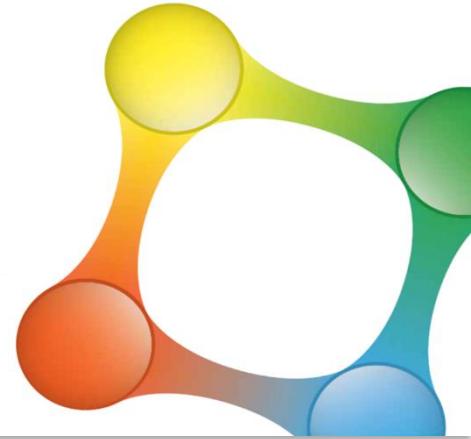
Be the first of your friends to like this

Most viewed posts (weekly)

- 5 Ways to Subset a Data Frame in R
- Cheesecake Diagrams: Pie Charts with a Different Flavour
- How to write the first for loop in R
- Date Formats in R
- In-depth introduction to machine learning in 15 hours of expert videos

<https://blog.rstudio.com/>

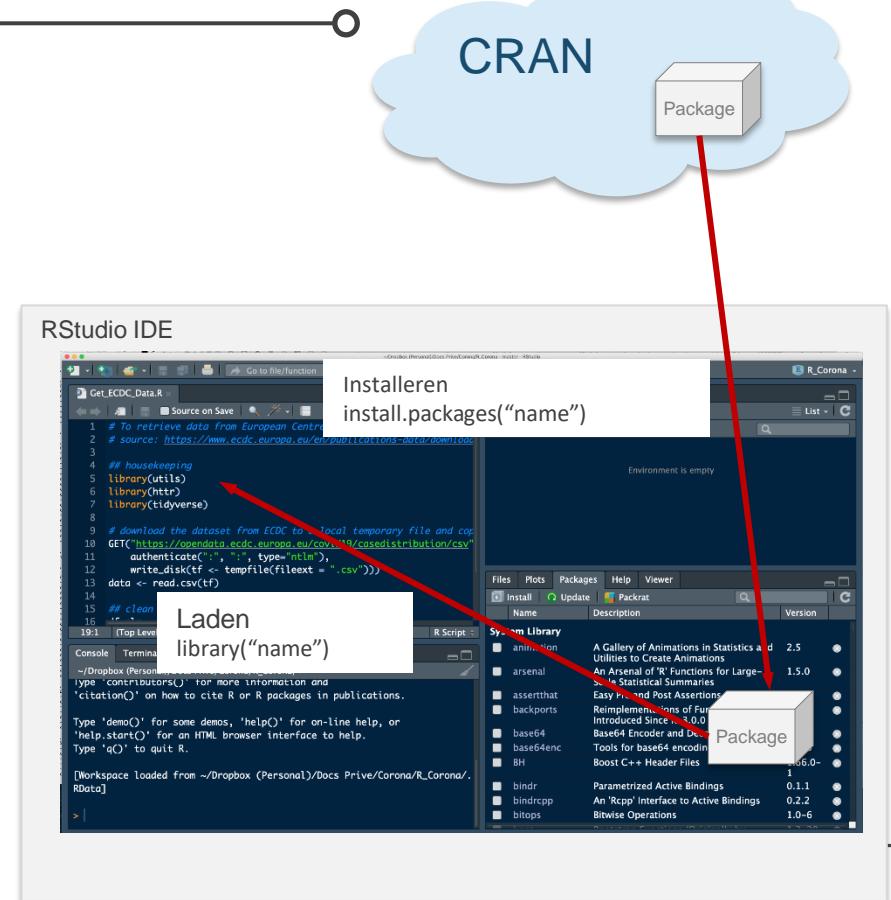
R-Packages / Libraries



Packages en Libraries

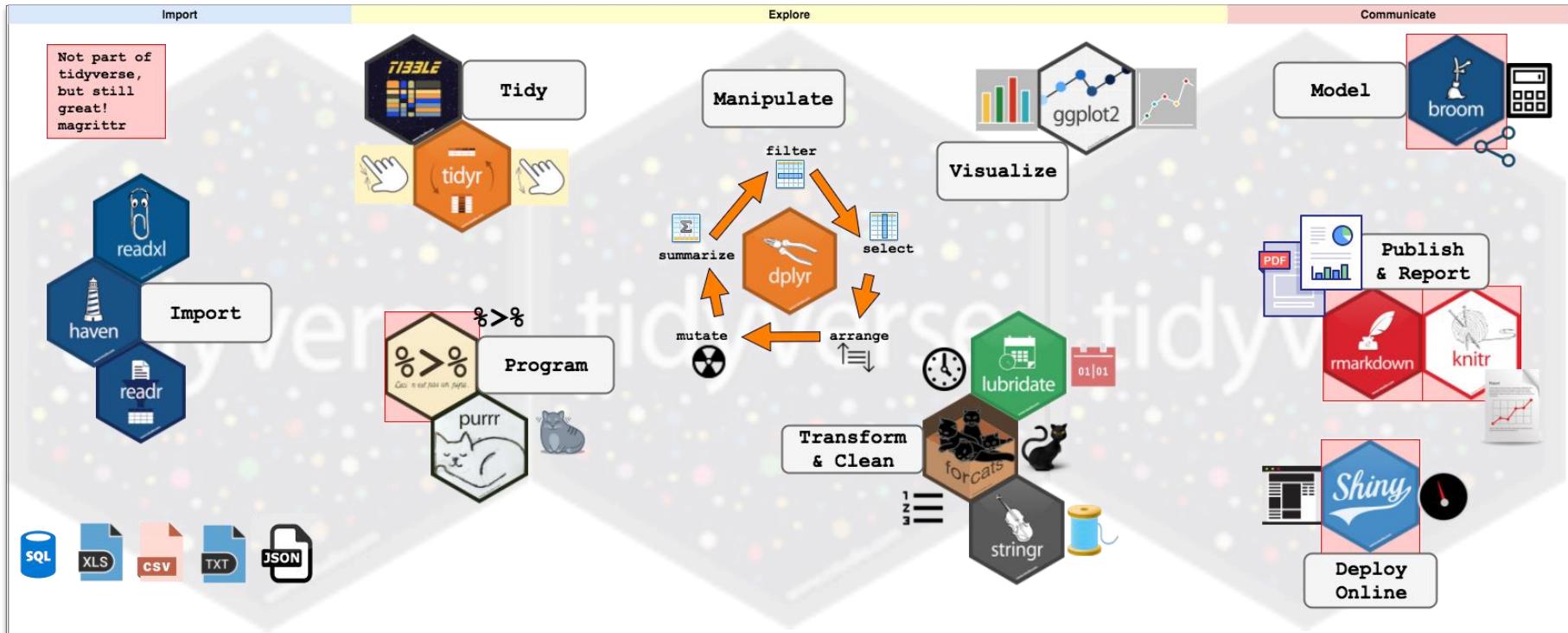
Overview

- Alle functies die je in R kunt gebruiken zijn afkomstig uit ‘packages’. Elk R-Package bevat een aantal R-functies (en soms ook voorbeeld-data), die je kunt gebruiken als je het ‘package’ hebt geladen.
- De standaard functies van R zitten in de package ‘BaseR’. Deze package wordt standaard geladen als je R (RStudio) opstart. Andere packages moet je zelf ‘laden’ voordat je de functies eruit kunt gebruiken.
- Er zijn bijvoorbeeld packages voor:
 - data-ontsluiting, -bewerking en -visualisatie
 - tekstverwerking (NLP)
 - statistische analyses
 - machine-learning
 - etc.
- De meest gebruikte ‘repository’ voor R-Packages is CRAN. Hier vanuit downloaden we onze packages en slaan deze dan gewoonlijk op in onze eigen persoonlijke package-‘library’ met het commando `install.packages("name")`. Als het package is geïnstalleerd dan is het beschikbaar in de ‘user-library’.
- Als we de functies uit het package willen gebruiken in ons R-script, dan moeten we het package eerst ‘laden’ met: `library("name")`



Packages en Libraries

Voorbeeld: Tidyverse



Packages en Libraries

Packages in R bekijken

Onder de drop-down van de knop ‘Global Environment’ vind je ook alle geladen R-packages.

Als je één van de R-functies in het R-package kiest, krijg je de source code van die functie te zien.

The screenshot shows the RStudio interface. On the left, the code editor displays the source code for the `fix_data_frame` function from the `broom` package. On the right, the Environment pane shows the global environment with various loaded packages. A dropdown menu in the Environment pane is set to `package:broom`, which is highlighted with a yellow arrow. Below this, the `Values` section lists several functions from the `broom` package, and the `Functions` section shows the definitions for `augment`, `fix_data_frame`, `glance`, and `tidy`.

Environment History

Import Dataset

Global Environment

Global Environment

package:mgcv

package:nlme

package:broom

package:dplyr

package:ggplot2

package:stats

package:graphics

package:grDevices

3988 obs.

2 obs. of

3 obs. of

2 obs. of

1

1

1

2

1-explore.R * mod3 * tidy * fix_data_frame * app.R *

Run

(Read-only)

Function: fix_data_frame (namespace:broom)

```
1 function (x, newnames = NULL, newcol = "term")
2 {
3   if (all(rownames(x) == seq_len(nrow(x)))) {
4     ret <- data.frame(x, stringsAsFactors = FALSE)
5     if (!is.null(newnames)) {
6       colnames(ret) <- newnames
7     }
8   } else {
9     ret <- data.frame(a = rownames(x), plyr::unrowname(x),
10   stringsAsFactors = FALSE)
11   colnames(ret)[1] <- newcol
12   if (!is.null(newnames)) {
13     colnames(ret)[-1] <- newnames
14   }
15 }
16 unrowname(ret)
17 }
18 }
```

Environment History

Import Dataset

package:broom

Values

augment_columns	<Promise>
bootstrap	<Promise>
confint_tidy	<Promise>
finish_glance	<Promise>
inflate	<Promise>
tidyMCMC	<Promise>

Functions

augment	function (x, ...)
fix_data_frame	function (x, newnames = NULL, newcol = "term")
glance	function (x, ...)
tidy	function (x, ...)

Files Plots Packages Help Viewer

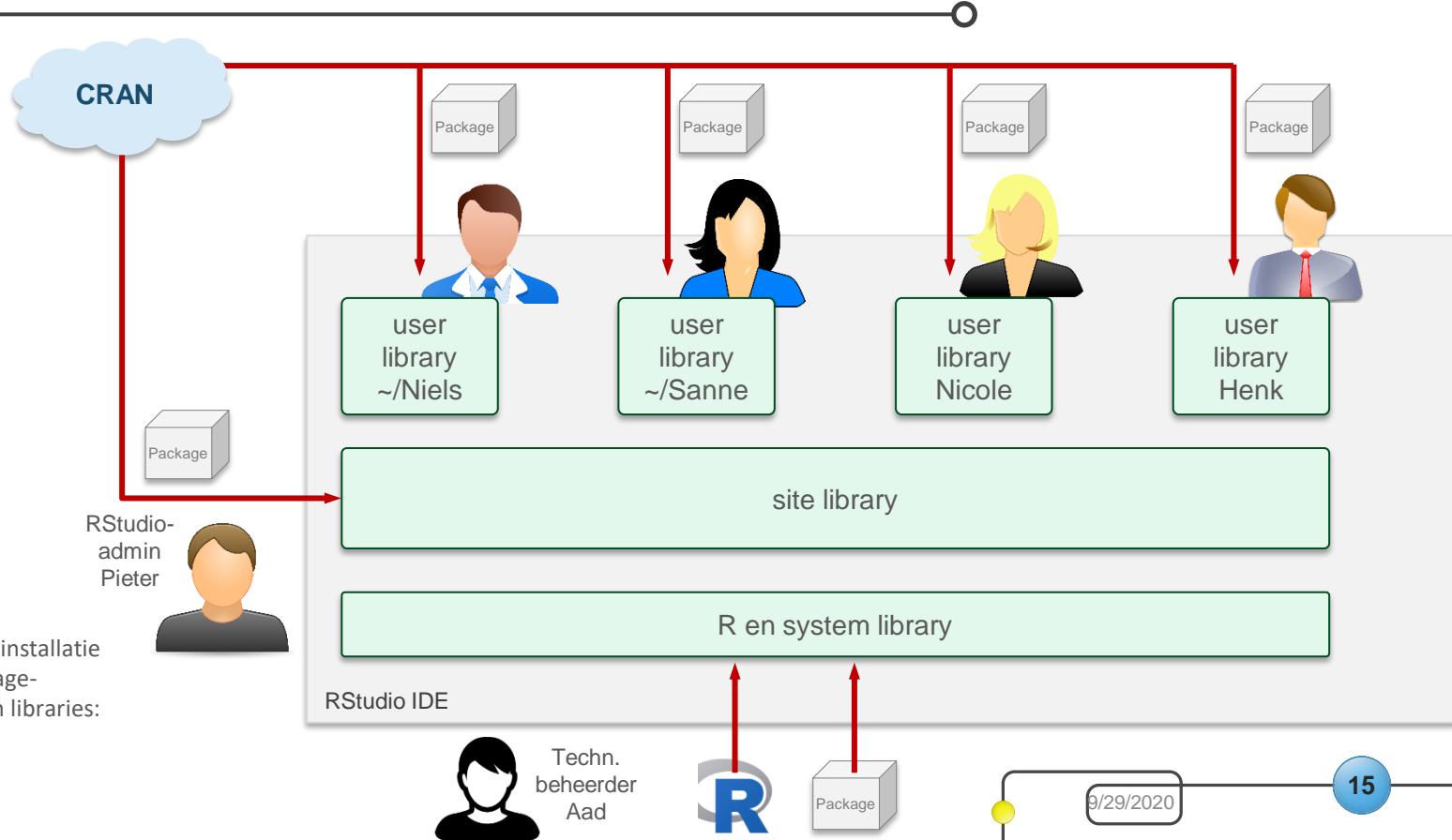
Zoom Export Publish

Package-libraries (drie typen)

- Een R-Package wordt bij installatie opgeslagen in een 'package-library'. Er zijn drie typen libraries:
 - System-library
 - Site-library
 - User-library
- System-library:** bevat de basis-packages voor R om te kunnen functioneren (bijv. base). Deze packages worden geïnstalleerd tijdens de installatie van een (nieuwe) R-versie en zijn daarna beschikbaar voor alle onderzoekers
- Site-library:** bevat R-Packages die door veel onderzoekers worden gebruikt. Deze packages worden door de RStudio-admin geïnstalleerd en zijn daarna beschikbaar voor alle onderzoekers
- User-library:** bevat R-Packages die door een individuele onderzoeker zijn geïnstalleerd. Zijn alleen beschikbaar voor die ene onderzoeker

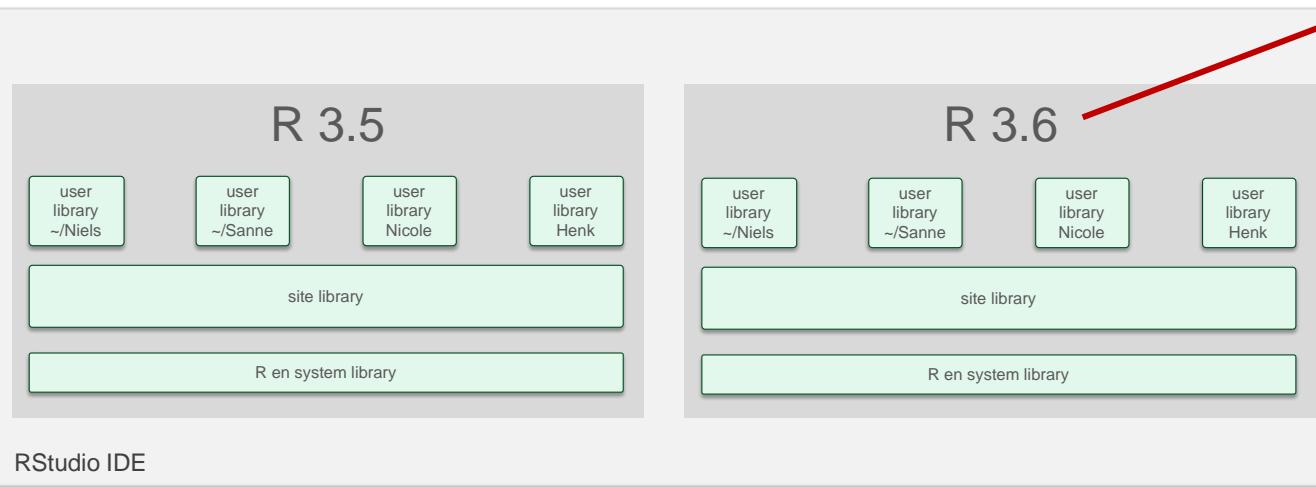
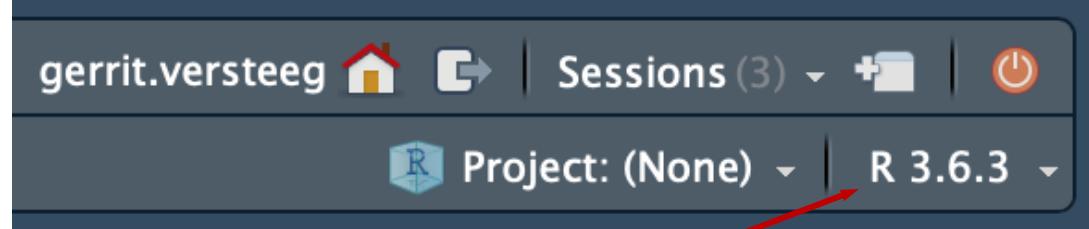
Libraries	System	Site	User
Bedoeld voor	Basis R-packages (bijv. base)	R-packages die door veel onderzoekers worden gebruikt (bijv. tidyverse)	R-packages die door één of slechts enkele onderzoekers worden gebruikt
Installatie door	Tech. applicatie-beheerder	RStudio-admin Functioneel applicatie-beheerder	Onderzoeker zelf
Locatie	Centrale folder	Centrale folder	Eigen home folder

Package-libraries (drie typen)



Package-libraries (per R-versie)

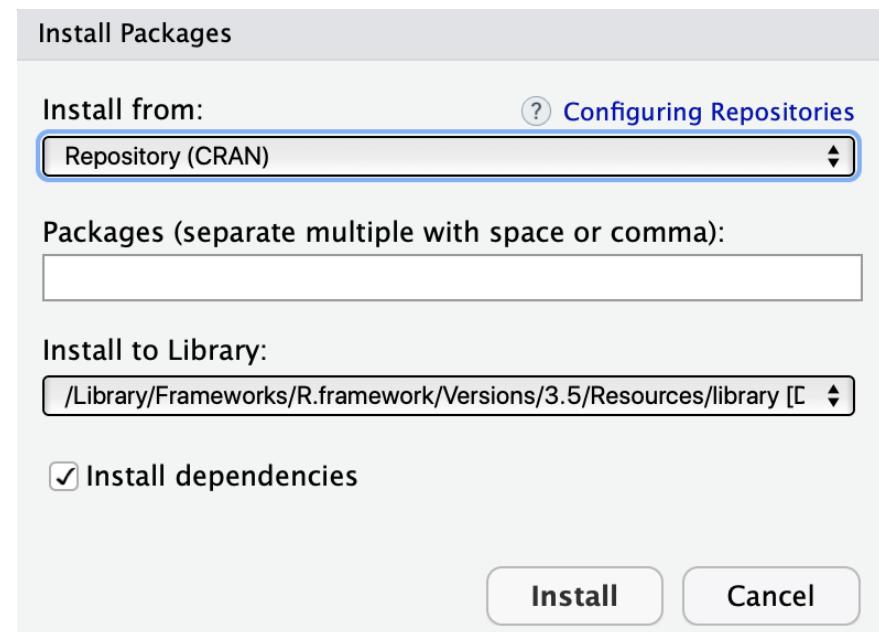
Elke versie van R heeft zijn eigen 'package-libraries'



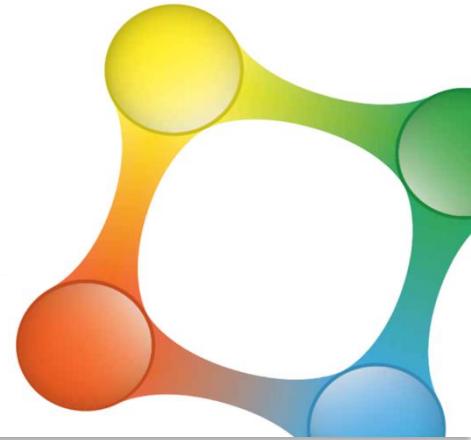
RStudio IDE

Demo: Het gebruik van R-Packages

- Selecteer twee veel-gebruikte packages + cars (als voorbeeld met data)
- Install Packages tab
- Install packages scherm
 - Install from Repo
 - Package-name
 - Install to Library
- Install dependencies
- Troubleshooting packages

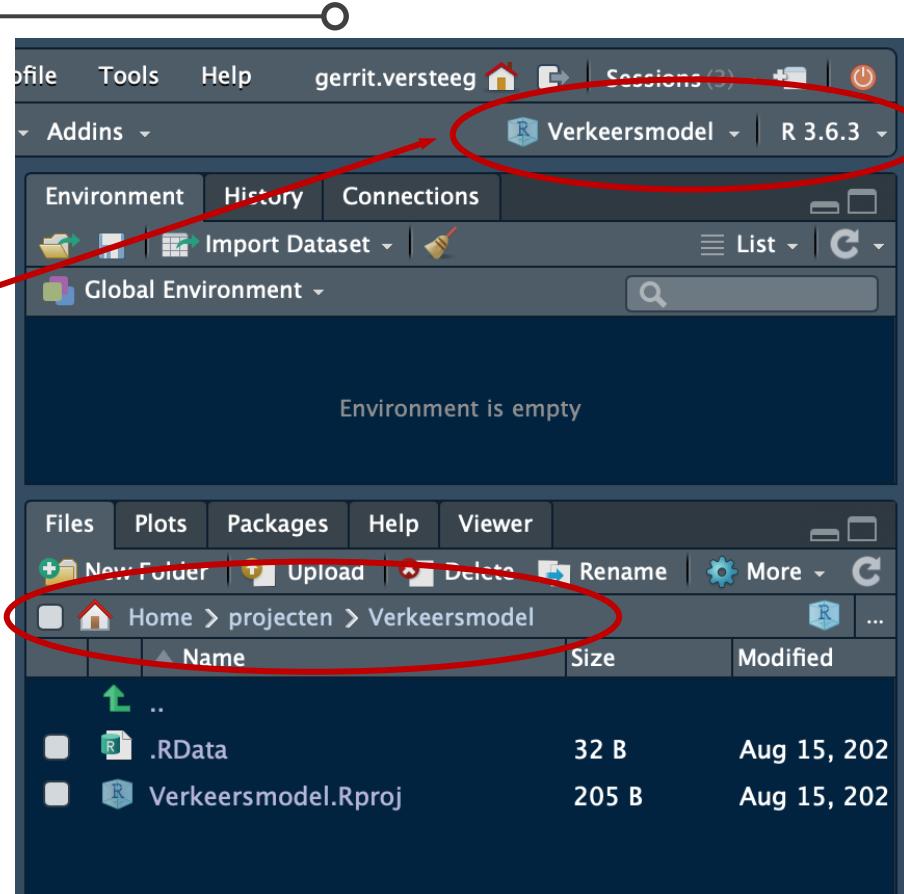


R-Projecten



Overview

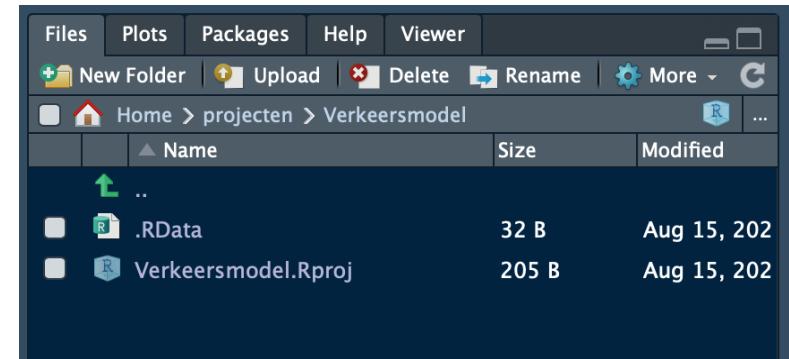
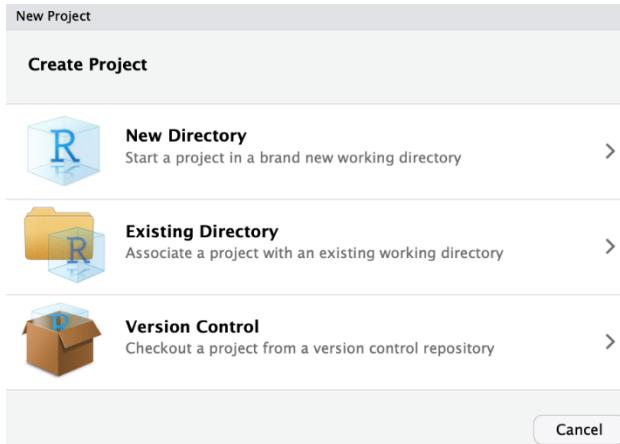
- Binnen RStudio kun je met 'projects' werken. Dit maakt het makkelijk om je werk op te delen in meerdere 'contexten' met elk een eigen:
 - working directory
 - environment
 - history
 - brondocumenten, etc.
- Als je verder wilt werken aan een specifiek project (bijv. een verkeersmodel) dan kun je dat project selecteren in de lijst van bestaande projecten.
- Daarmee schakelt RStudio om naar de context van dat project, en heb je situatie terug die je achterliet toen je voor het laatst met het project werkte.



Het aanmaken van een project

Bij een nieuwe project, zal RStudio o.a.:

- een project-bestand met de extensie .Rproj maken (bevat project-opties en dient als shortcut om het project te openen vanuit je file explorer)
- Het project laden in RStudio



Een RStudio project is altijd verbonden met een hoofd projectfolder

'R working directory'

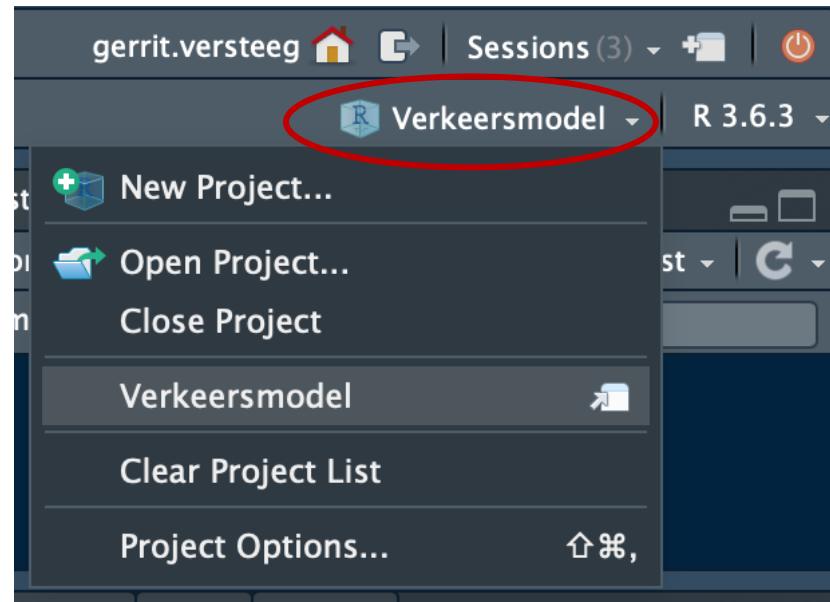
Je kunt een RStudio-project maken:

- in een nieuwe working directory
- in een bestaande working directory
- door een Git-repository te clonen in een nieuwe working directory

Het openen van een project

Bij het openen van een nieuwe project, zal RStudio:

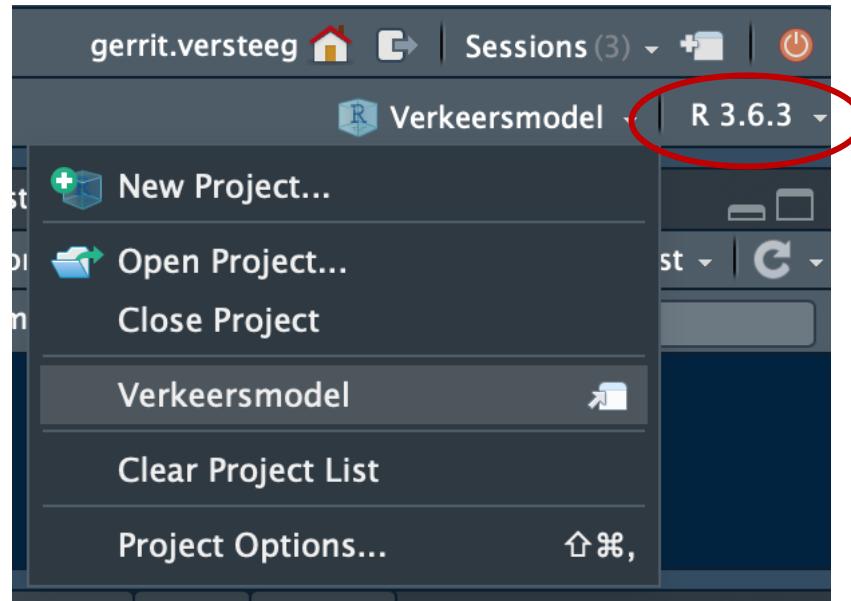
- Een nieuwe R sessie starten
- En dan vanuit de hoofd projectfolder:
 - Het .Rprofile bestand 'sourcen'
 - Het .RData bestand laden in de environment tab
 - Het .Rhistory bestand laden in de history tab
- R's 'current working directory' wordt gezet op de hoofd projectfolder
- Source documenten die ge-edit waren worden hersteld in de editor-tabs
- Alle editor settings (zoals actieve tabs) worden hersteld naar de laatste keer dat het project was gebruikt



De R-versie van een project

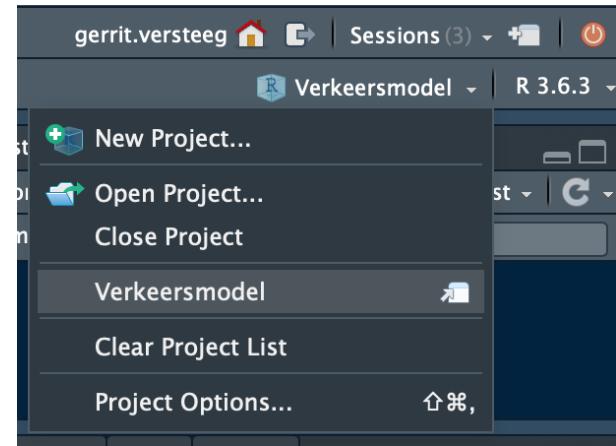
Het is mogelijk om bij een project de gebruikte versie van R te specificeren:

- Hiermee blijft je code werken, ook al zijn er nieuwe R-versies geïnstalleerd.
- Maar je kunt de R-versie van je project altijd wijzigen met behulp van de R-versie knop

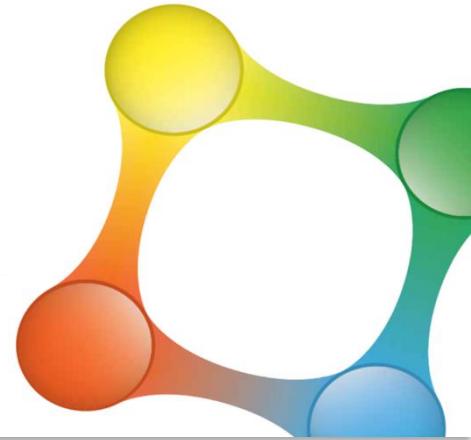


Demo: R-Projecten

- Tonen van project-context (wat hoort allemaal bij een R-Project)
- Maken van twee nieuwe project (zonder versie-beheer) vanuit een folder met gemixte bestanden
- Laten zien hoe de projecten-folder te gebruiken in je home-directory.
- Navigeren tussen projecten



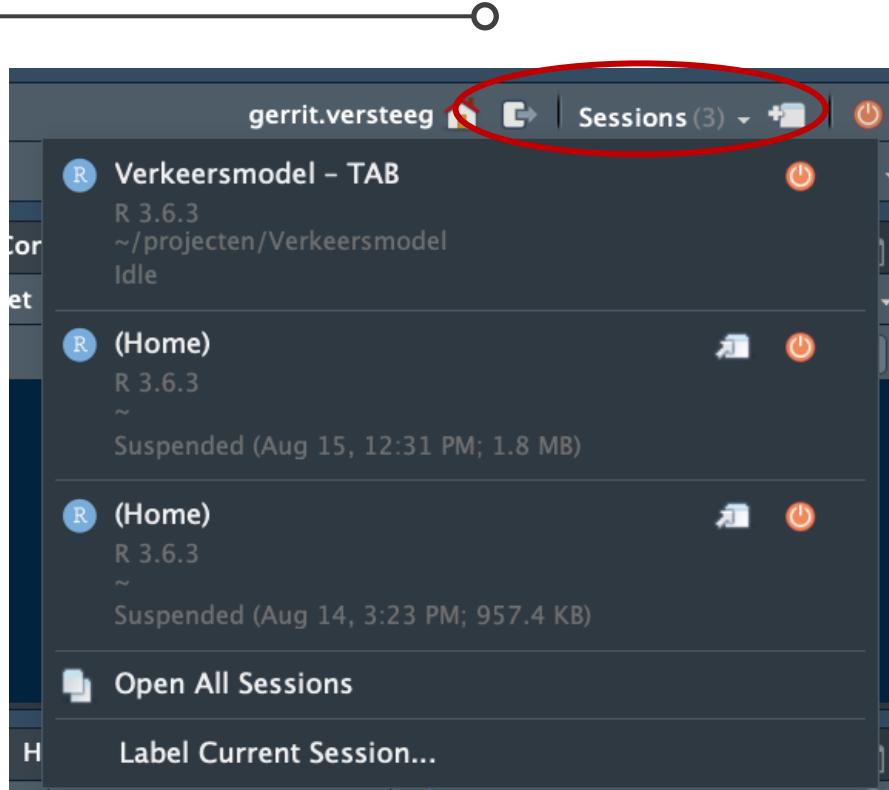
R-Sessies



Sessies

Overview

- Binnen RStudio kun je meerdere R-sessions tegelijkertijd open hebben staan.
- Dat is vooral handig als je een lange berekening in één sessie doet en dan een andere sessie opent om intussen verder te kunnen werken.
- Naast je gebruikersnaam is een drop-down menu met de sessies die je open hebt staan, zodat je makkelijk tussen sessies kan switchen

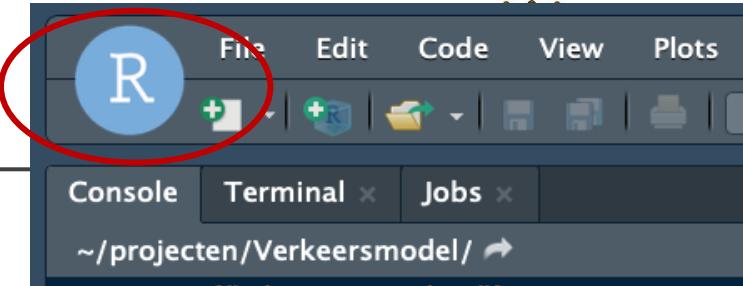


Het RStudio Overview scherm

Als je op de R home-knop drukt, opent het RStudio Overview scherm.

Hierin kun je:

- al je open sessies zien en openen
- al je actieve projecten zien en openen
- nieuwe sessies of projecten starten

A screenshot of the RStudio Overview screen. At the top left is the "R Studio Server Pro" logo. On the right are "Logout" and "gerrit.versteeg". Below the logo are two main sections: "Sessions" and "Projects".

Sessions

- (Home)**
SUSPENDED R 3.6.3
(Home) CREATED: 14/08/2020 LAST USED: 14/08/2020
- (Home)**
SUSPENDED R 3.6.3
(Home) CREATED: 10/08/2020 LAST USED: 15/08/2020
- ~/projecten/Verkeersmodel**
IDLE R 3.6.3 TAB
~/projecten/Verkeersmodel CREATED: 30/07/2020 LAST USED: 17:37:20

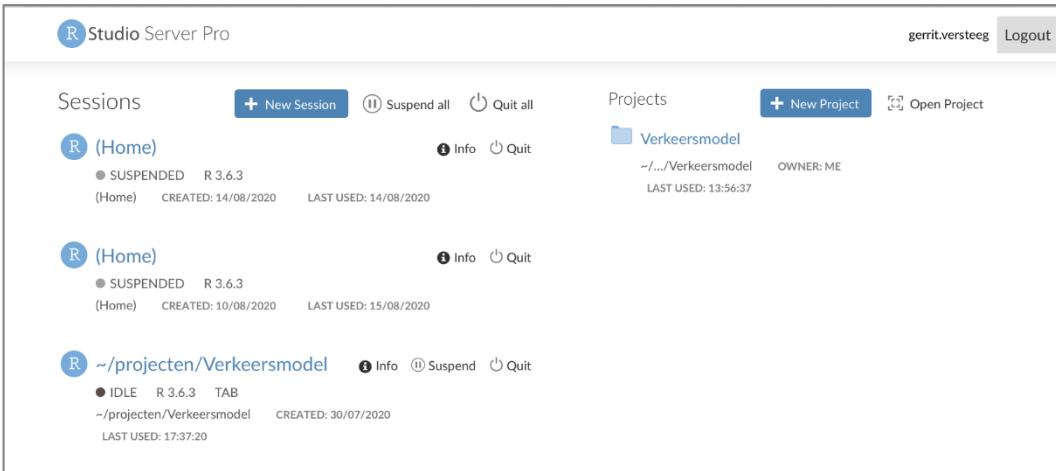
Projects

- Verkeersmodel**
~/.../Verkeersmodel OWNER: ME LAST USED: 13:56:37

Het openen van een nieuwe sessie

Of je kunt in het RStudio Overview scherm de '+ New Session' knop gebruiken.

Je kunt met + knop naast de 'Sessions' dropdown knop, een nieuwe sessie openen.



The screenshot shows the RStudio Server Pro interface. At the top, it displays the user 'gerrit.versteeg' and a 'Logout' button. Below this, there are two main sections: 'Sessions' and 'Projects'. The 'Sessions' section contains three entries:

- (Home) SUSPENDED R 3.6.3 (Created: 14/08/2020, Last Used: 14/08/2020)
- (Home) SUSPENDED R 3.6.3 (Created: 10/08/2020, Last Used: 15/08/2020)
- ~/projecten/Verkeersmodel IDLE R 3.6.3 TAB (Created: 30/07/2020, Last Used: 17:37:20)

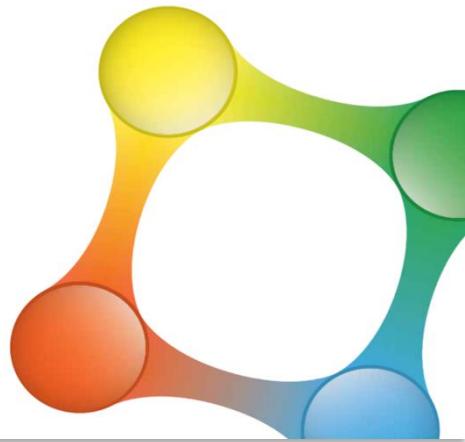
Each session entry has an 'Info', 'Suspend', and 'Quit' button. To the right of the sessions, there is a '+ New Session' button. The 'Projects' section shows a single project named 'Verkeersmodel' with details: OWNER: ME, LAST USED: 13:56:37, and an 'Open Project' button. To the right of the projects, there is a '+ New Project' button.

Demo: R-Sessions

- Geef in een bestaande R-Sessie het commando sys("Sleep 240").
- Laat zien in het RStudio Overview scherm zien, dat de sessie busy is
- Open een nieuwe sessie
- Laat zien in het RStudio Overview scherm zien, dat de tweede sessie aanwezig is
- Laat zien dat je kunt switchen tussen de sessies
- Laat zien dat je verder kunt werken in de idle session



Versiebeheer



Waarom versiebeheer

A STORY TOLD IN FILE NAMES:

Filename	Date Modified	Size	Type
data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
data_2010.05.28_huh???.dat	7:20 PM 5/28/2010	30 KB	DAT file
data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
data_2010.05.29_aaarrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
data_2010.05.29_#\$@*!&!.dat	2:40 AM 5/29/2010	0 KB	DAT file
data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
data_2010.05.29_woohoo!.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,894 KB	DAT file
analysis_graphs.xls	7:13 AM 5/29/2010	455 KB	XLS file
ThesisOutline!.doc	7:26 AM 5/29/2010	38 KB	DOC file
Notes_Meeting_with_ProfSmith.txt	11:38 AM 5/29/2010	1,673 KB	TXT file
JUNK...	2:45 PM 5/29/2010		Folder
data_2010.05.30_startingover.dat	8:37 AM 5/30/2010	420 KB	DAT file

Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com

Bedoeld voor:
Veiligheid &
Samenwerking

Versies op de oude manier

- script_V0.1.R
- script_V0.11.R
- script_V0.2.R
- script_V0.21.R
- script_V1.0.R
- script_V1.0_Final.R
- script_V1.0_FinalFixed.R
- script_V1.0_ReallyFinal.R

Waarom versiebeheer

Git en Github leren

- Is frustrerend in het begin en er zijn veel nieuwe begrippen en concepten. Maar het gebruik van RStudio helpt je hierbij.
- betaalt zich terug door een verhoging van je programmeersnelheid omdat je je minder zorgen hoeft te maken over vergissingen
- betaalt zich terug door jouw werk te delen met onderzoekers uit jouw organisatie zodat zij kunnen helpen om je code beter te maken

Wat wil je weten van een verandering:

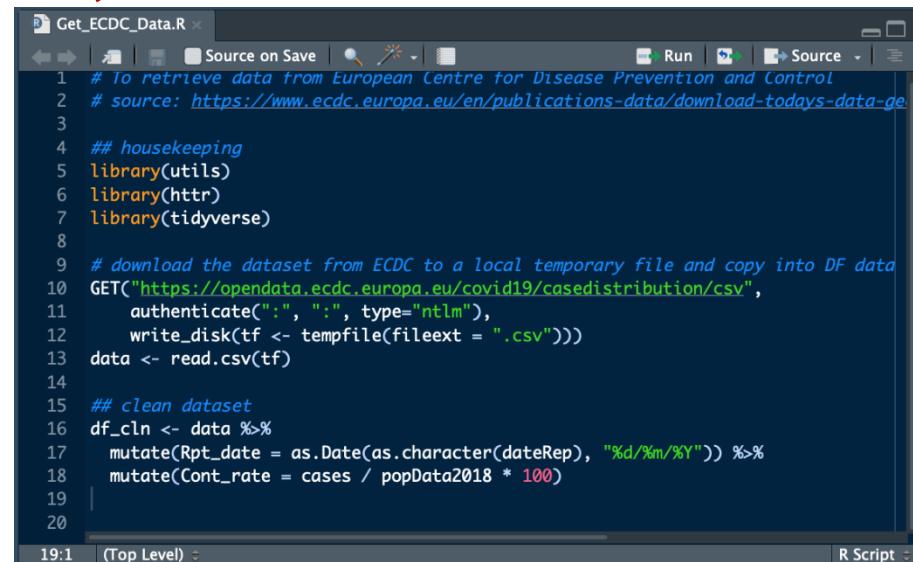
- Wat is er veranderd?
- Wanneer is het veranderd?
- Wie heeft het veranderd?
- Waarom is het veranderd?

Het is daarbij handig dat elke verandering een unieke identificatie heeft

Versiebeheer Overview

- Met R programmeer je:
 - los in de console
 - in een R-script
 - in een R-markdown script
 - in een ‘notebook’
- Als je in een script of een notebook programmeert is het verstandig om met ‘versies’ van je script te werken. Op die manier kun je altijd terug naar een vorige versie van je script-bestand.
- Versiebeheer gebruiken we **alleen** voor tekst-georiënteerde bestanden, zoals scripts of documentatie. Dus niet voor databestanden.
- Voor versiebeheer gebruiken we ‘GIT’. Dit is een Version Control System (VCS). Het VCS ‘GIT’ is door ons al op de research omgeving geïnstalleerd, samen met R en RStudio.
- MAAR we moeten wel eenmalig opgeven wie we zijn, zodat GIT weer wie de verandering heeft aangebracht:
 - git config --global user.name "YOUR FULL NAME"
 - git config --global user.email "YOUR EMAIL ADDRESS"
- Versies slaan we ook in de cloud op, namelijk in github.com. Hiermee hebben we een back-up en kunnen andere onderzoekers aan jouw scripts bijdragen of jouw scripts hergebruiken

Name	Size	Modified
..	46 B	Mar 30, 2020, 1:12 PM
.gitignore	2.5 KB	Jun 4, 2020, 5:23 PM
.RData	1.8 KB	Jun 16, 2020, 2:13 PM
.Rhistory		
data		
Get_ECDC_Data.R	687 B	Jun 4, 2020, 2:54 PM
R_Corona.Rproj	205 B	Aug 13, 2020, 11:13 AM
README.md	38 B	Mar 27, 2020, 2:14 PM



```
# To retrieve data from European Centre for Disease Prevention and Control
# source: https://www.ecdc.europa.eu/en/publications-data/download-todays-data-ge
## housekeeping
library(utils)
library(httr)
library(tidyverse)

# download the dataset from ECDC to a local temporary file and copy into DF data
GET("https://opendata.ecdc.europa.eu/covid19/casedistribution/csv",
  authenticate(":", "", type="ntlm"),
  write_disk(tf <- tempfile(fileext = ".csv")))
data <- read.csv(tf)

## clean dataset
df_cln <- data %>%
  mutate(Rpt_date = as.Date(as.character(dateRep), "%d/%m/%Y")) %>%
  mutate(Cont_rate = cases / popData2018 * 100)
```

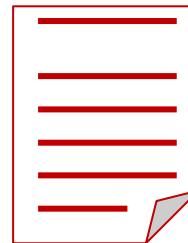
Voor extra leesmateriaal rond het gebruik van Git in RStudio:
<http://r-pkgs.had.co.nz/git.html>

GIT – Basis concepten

GIT is een version control system (VCS)
Een VCS maakt het mogelijk om veranderingen op
(tekst-gebaseerde) bestanden vast te leggen over
de tijd (door middel van ‘snapshots’ of ‘commits’)



Version 1



Version 2



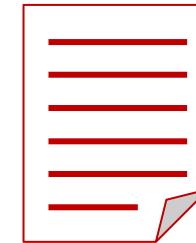
Version 3



Snapshots kun je gebruiken om bestanden te herstellen
naar voorgaande versies, als je dat nodig hebt



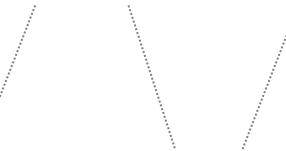
Version 1



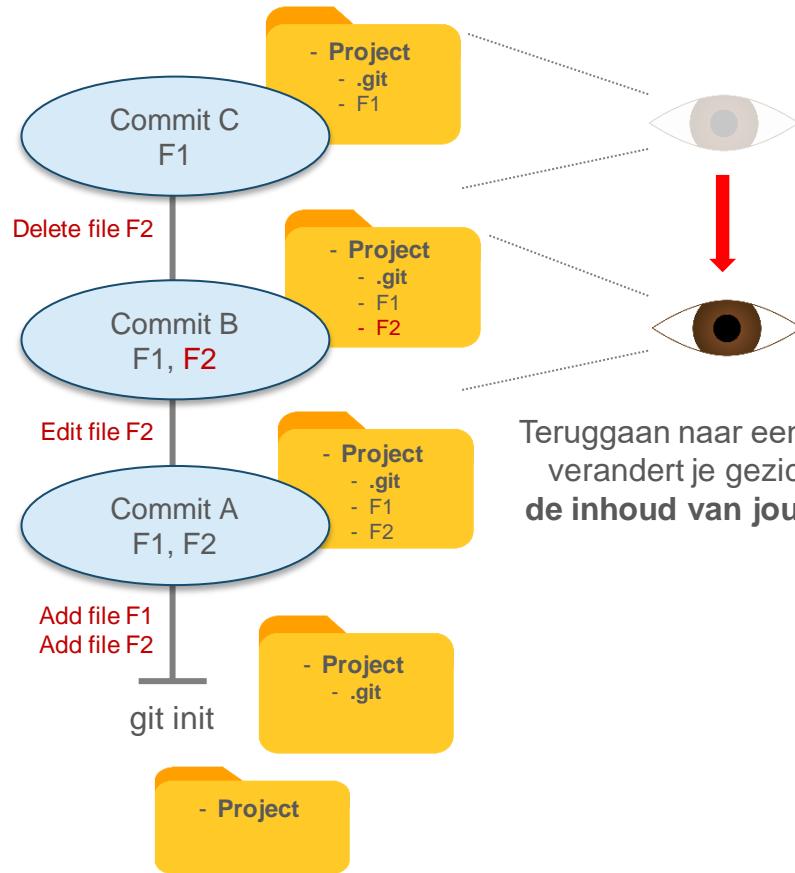
Version 2



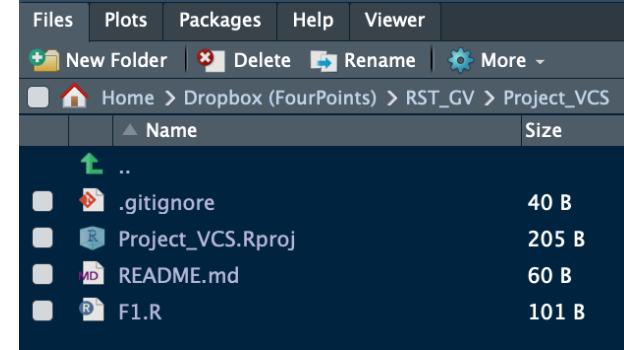
Version 3



GIT – Folders / Repositories

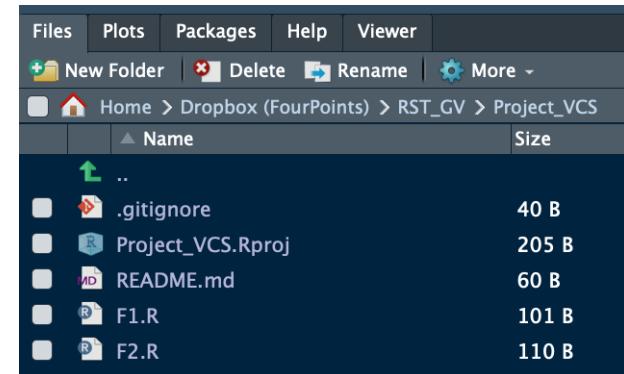


Teruggaan naar een vorige snapshot ('restore')
verandert je gezichtspunt én daarmee ook:
de inhoud van jouw werkelijke werkfolder !



A screenshot of a file browser interface showing the contents of a folder named "Project_VCS".

Name	Size
..	40 B
.gitignore	205 B
Project_VCS.Rproj	60 B
README.md	101 B
F1.R	



A screenshot of a file browser interface showing the contents of a folder named "Project_VCS".

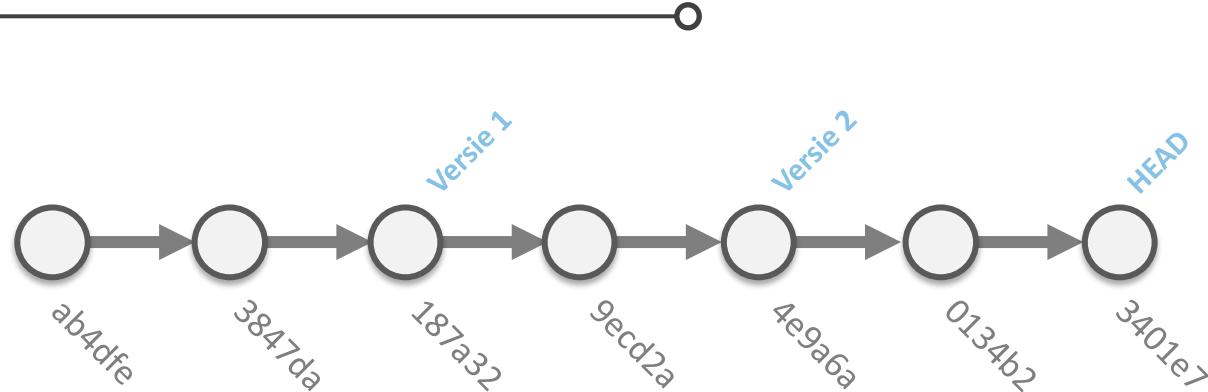
Name	Size
..	40 B
.gitignore	205 B
Project_VCS.Rproj	60 B
README.md	101 B
F1.R	
F2.R	110 B

GIT – Snapshots (Commits)

Een snapshot (commit) neem je als je een betekenisvolle stap hebt genomen in je code.

- Binnen Git heeft elke commit een unieke identificatie: een SHA (eigenlijk een lange reeks van hex getallen, maar meestal afgekort tot de zes eerste daarvan)
- De SHA is een sleutel in een database van commits met de auteur, datum en een beschrijving
- Je kunt een commit ook een eigen naam geven:

```
git tag "Versie 1" 187a32
```



GIT – De git-repository

Een git-repository is:
een werkfolder met daarin een
speciale (verborgen) **.git** subfolder,
waarin alle snapshots
worden opgeslagen

Alle normale bestanden in
jouw project folder

Work-directory

F1
F2
F3

speciale .git subfolder

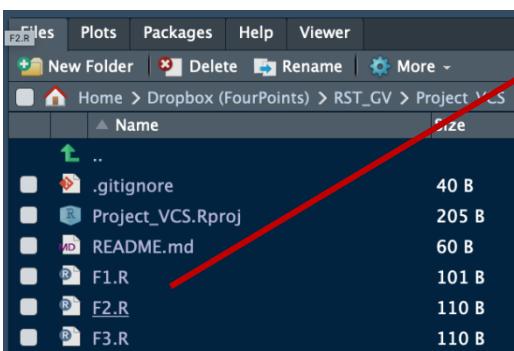
Staging area (index)

History

Commit B
F1, F2, F3

Commit A

git init

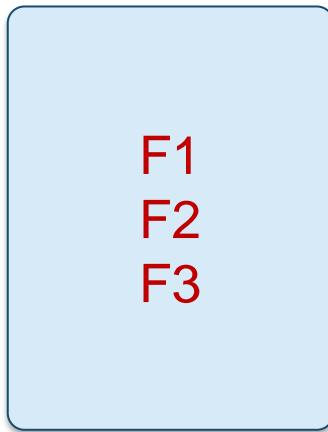


	Name	Size
..	..	40 B
■	.gitignore	40 B
■	Project_VCS.Rproj	205 B
■	README.md	60 B
■	F1.R	101 B
■	F2.R	110 B
■	F3.R	110 B

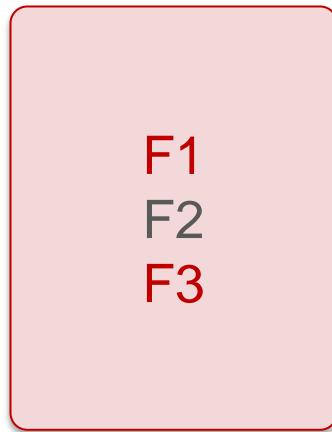


GIT – Stage (selecteer de inhoud van je snapshot)

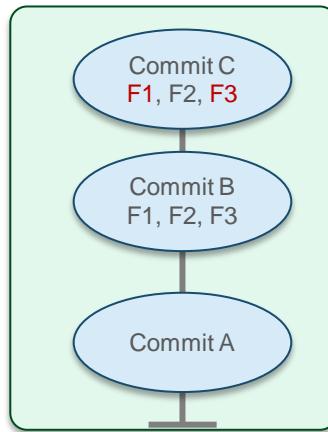
Working tree



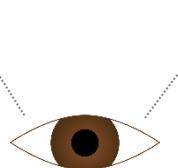
Staging area (index)



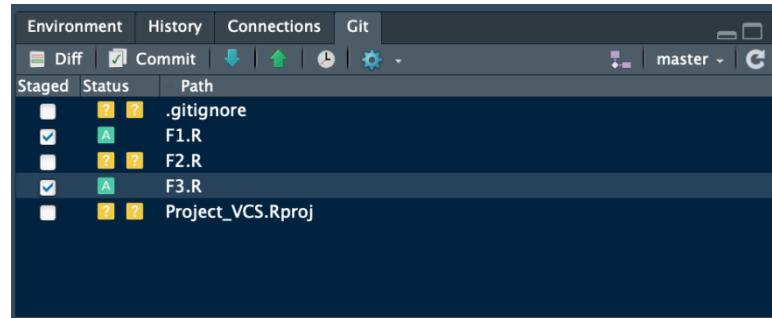
History



git init

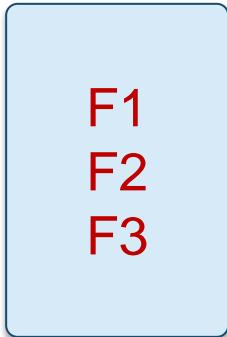


Selecteer alleen F1 en F3 voor de snapshot, door de nieuwe versies van alleen deze twee files, in de staging area te plaatsen



GIT – Je eerste snapshot ('commit')

Working tree



Voegt F1 en F2 toe aan de staging area, waardoor die nu onder beheer vallen ('tracked')

`git add F1, F3`

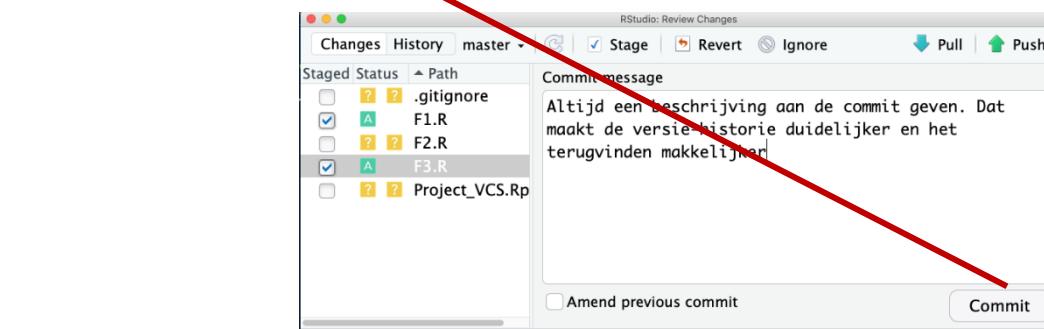
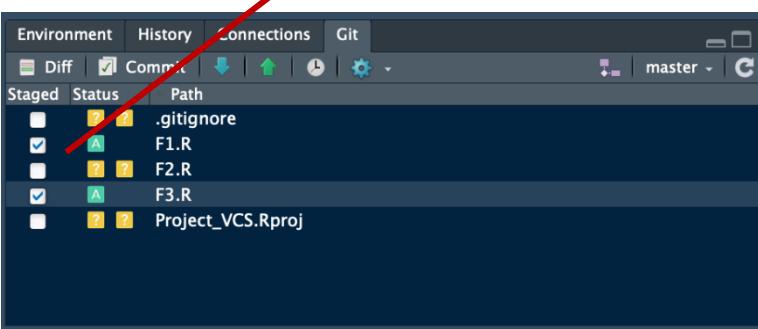
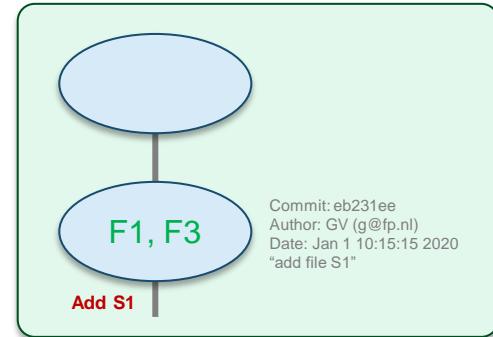
Staging area



Maakt een snapshot ('commit') van alle bestanden in staging naar de 'history'

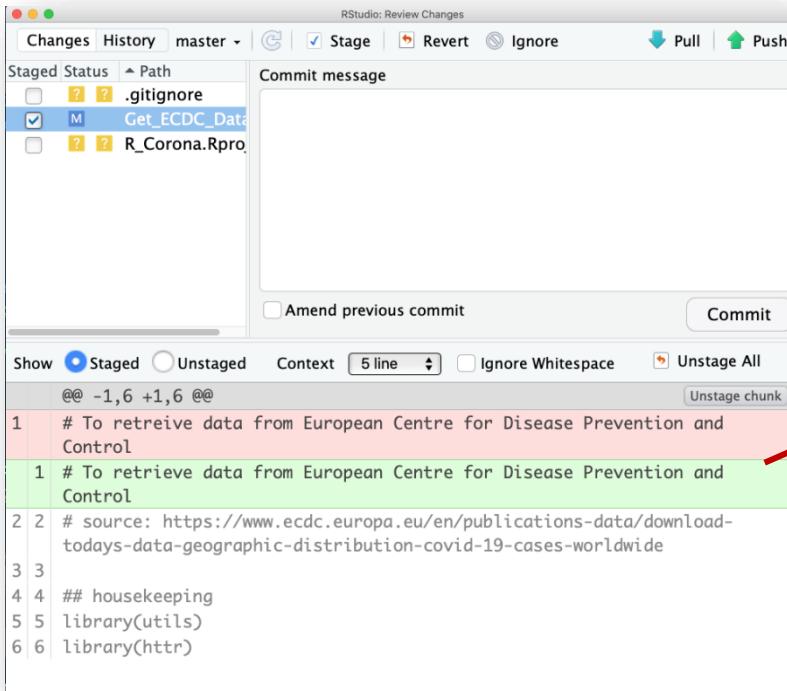
`git commit -m "toevoegen F1 en F3"`

History

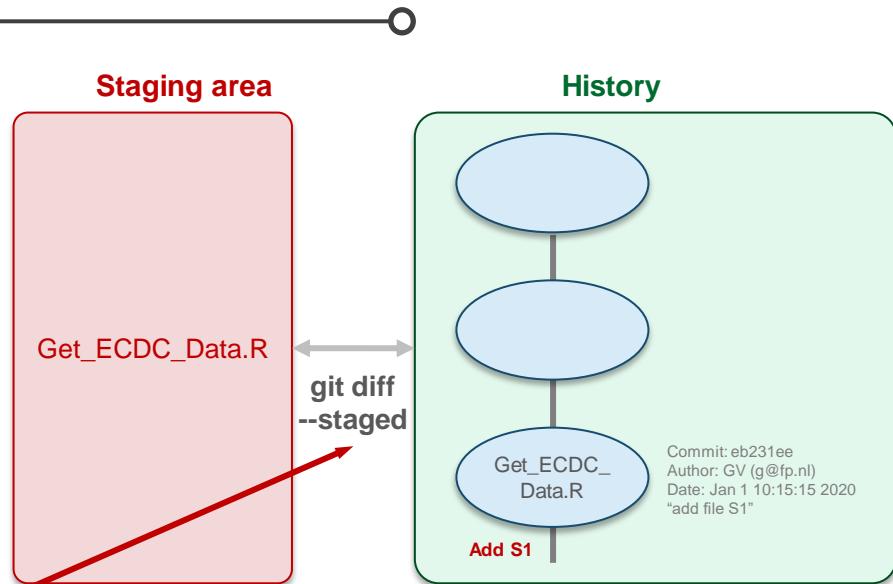


GIT – Verschillen met vorige versie

Note. ander voorbeeld gebruikt dan voorgaande sheets



```
@@ -1,6 +1,6 @@
1 # To retrieve data from European Centre for Disease Prevention and
Control
1 # To retrieve data from European Centre for Disease Prevention and
Control
2 # source: https://www.ecdc.europa.eu/en/publications-data/download-
todays-data-geographic-distribution-covid-19-cases-worldwide
3
4 ## housekeeping
5 library(utils)
6 library(httr)
```



GIT – Verschillen met vorige versie

RStudio: Review Changes

Subject	Author	Date	SHA
HEAD -> refs/heads/master	Gerrit Versteeg <gversteeg@fo	2020-06-04	db031637
Commit data-retrieval script	Gerrit Versteeg <gversteeg@fo	2020-03-27	0f75be22
Initial commit	Gerrit Versteeg <gversteeg@fo	2020-03-27	9f764e07

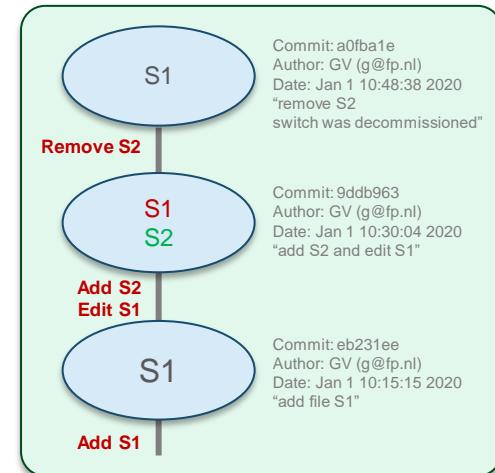
Commits 1-3 of 3

```

SHA db031637
Author Gerrit Versteeg <gversteeg@fourpoints.nl>
Date 2020-06-04 12:49
Subject first test
Parent 0f75be22
@ Get_ECDC_Data.R
@ Get_ECDC_Data.R View file @ db031637
@@ -1,17 +1,19 @@
1 # To retrieve data from European Centre for Disease Prevention and Control
2 # source:
3 # https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-
distribution-covid-19-cases-worldwide
2 # source: https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-
distribution-covid-19-cases-worldwide
4 ## housekeeping
5 library(utils)
6 library(httr)
7 library(tidyverse)
8
8 # download the dataset from the ECDC website to a local temporary file
9 # download the dataset from ECDC to a local temporary file and copy into DF data
9 10 GET("https://opendata.ecdc.europa.eu/covid19/casedistribution/csv",
10 11 authenticate(":", ":", type="ntlm"),
11 12 write_disk(tf <- tempfile(fileext = ".csv")))

```

History



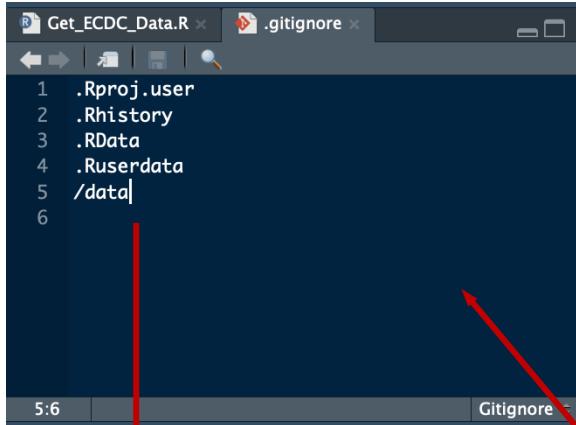
Note. commits komen niet overeen.
Bedoeld als voorbeeld van mogelijke commits

Roze : verwijderde regel
 Groen : toegevoegde regel
 Wit : geen verandering

GIT – .gitignore (pas aan in de editor)

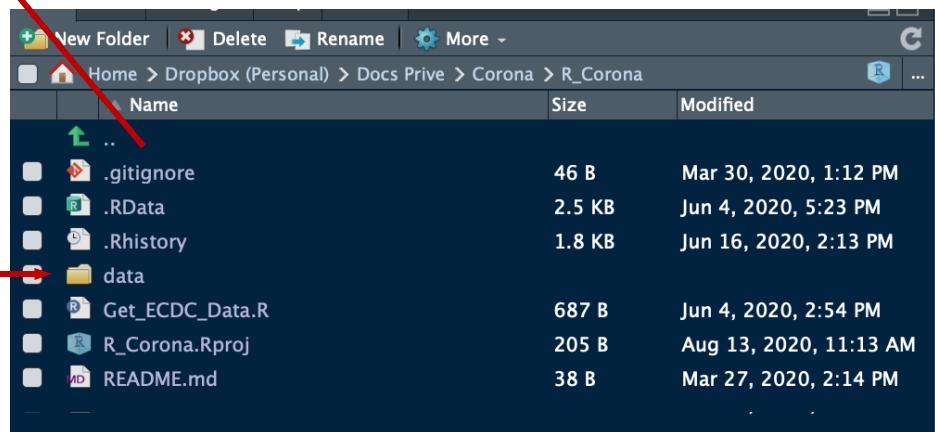
Workdirectory

```
data  
Get_ECDC_Data.R  
R_Corona.Rproj  
README.md  
.gitignore
```



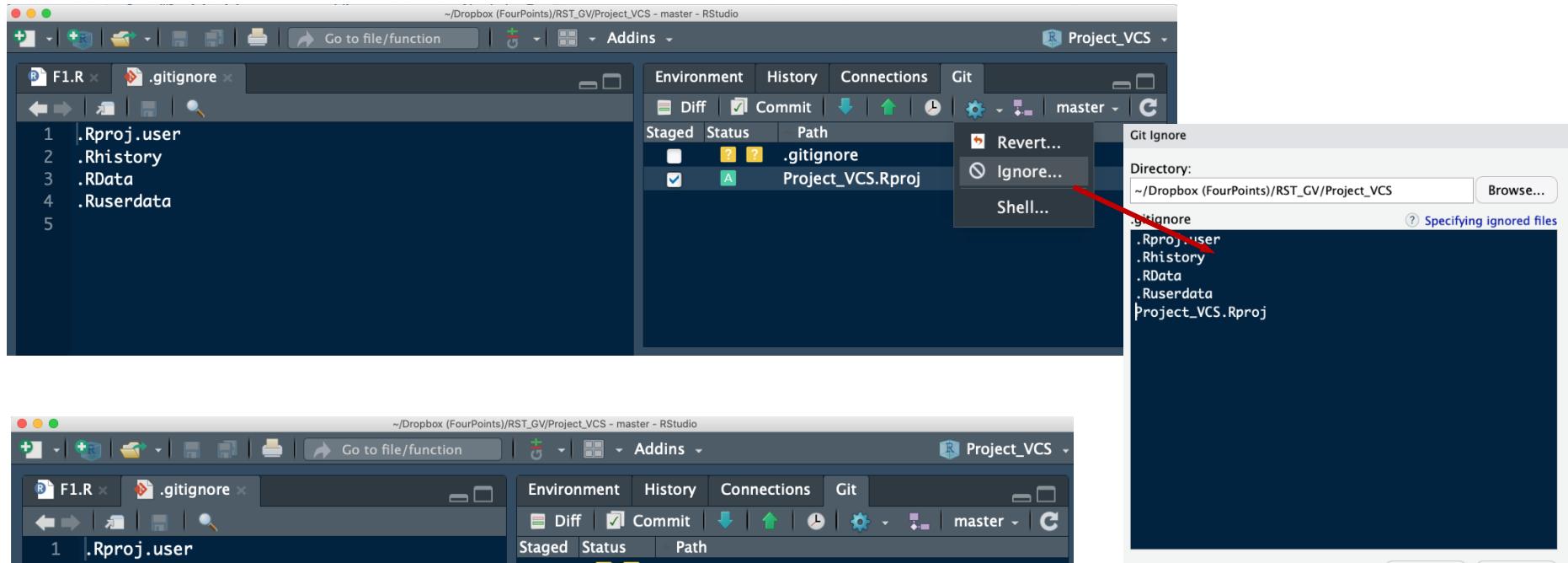
```
1 .Rproj.user  
2 .Rhistory  
3 .RData  
4 .Ruserdata  
5 /data|  
6
```

- Bepaalde objecten in je werkfolder kun je structureel buiten versiebeheer houden. Je hoeft dan niet steeds op te letten welke bestanden je selecteert (staged) voor de snapshot (commit).
- Daar is een speciaal bestand voor: .gitignore
- In .gitignore kun je de namen van bestanden en hele folders opnemen, zodat deze door Git worden genegeerd voor versiebeheer.
- De data die we gebruiken, zetten we gewoonlijk NIET in versiebeheer. Daarom is het handig om deze in een aparte folder te plaatsen.



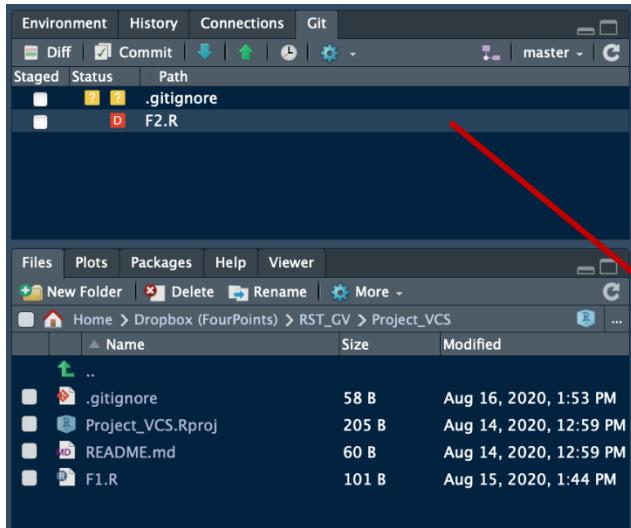
Name	Size	Modified
..	46 B	Mar 30, 2020, 1:12 PM
.gitignore	2.5 KB	Jun 4, 2020, 5:23 PM
.RData	1.8 KB	Jun 16, 2020, 2:13 PM
.Rhistory		
data		
Get_ECDC_Data.R	687 B	Jun 4, 2020, 2:54 PM
R_Corona.Rproj	205 B	Aug 13, 2020, 11:13 AM
README.md	38 B	Mar 27, 2020, 2:14 PM

GIT – .gitignore (gebruik RStudio ignore)



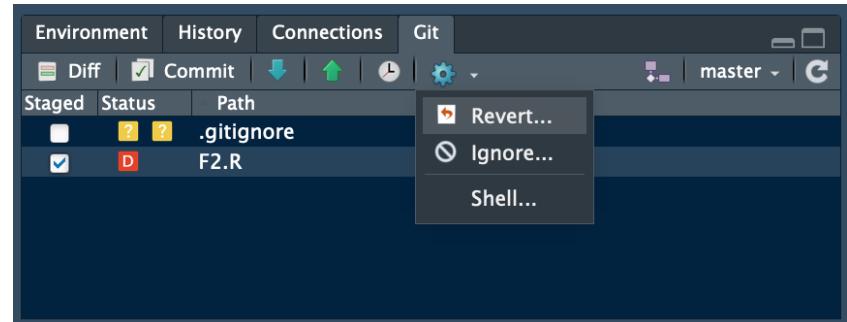
GIT – Terug naar een vorige versie (1)

Stel dat je F2.R terug wilt, maar je hebt het script verwijderd (maar nog niet ge-commit)



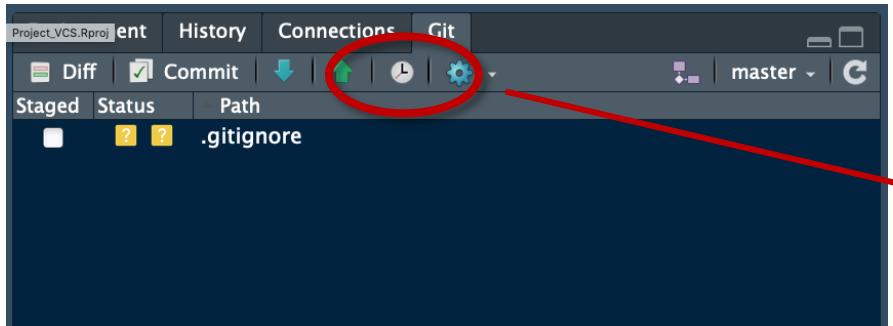
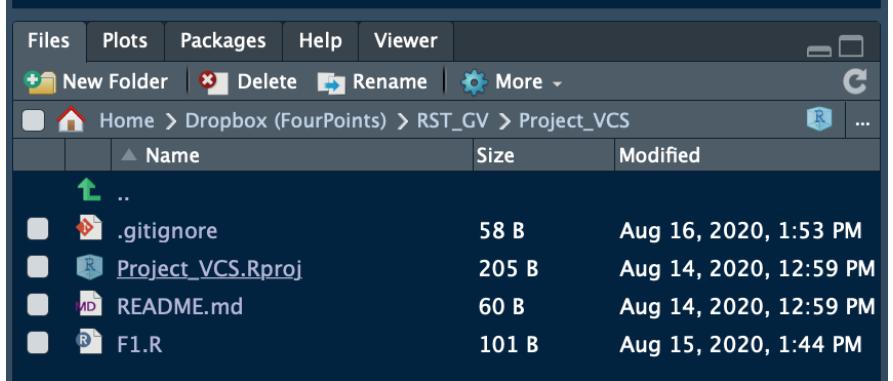
Environment History Connections Git			
Diff	Commit	Path	master
Staged	Status	Path	
?	?	.gitignore	
?	D	F2.R	

Files	Plots	Packages	Help	Viewer
New Folder	Delete	Rename	More	C
Home > Dropbox (FourPoints) > RST_GV > Project_VCS				
Name	Size	Modified		
..				
.gitignore	58 B	Aug 16, 2020, 1:53 PM		
Project_VCS.Rproj	205 B	Aug 14, 2020, 12:59 PM		
README.md	60 B	Aug 14, 2020, 12:59 PM		
F1.R	101 B	Aug 15, 2020, 1:44 PM		
F2.R	138 B	Aug 16, 2020, 2:28 PM		

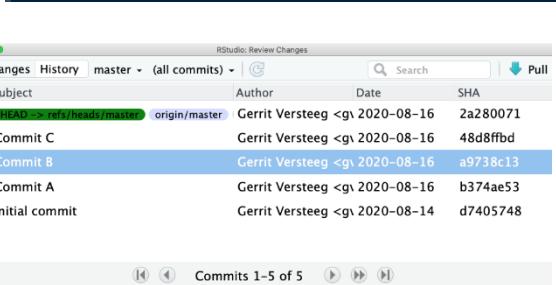


GIT – Terug naar een vorige versie (2)

Stel dat je F2.R terug wilt, maar je hebt het script verwijderd en ook al ge-commit

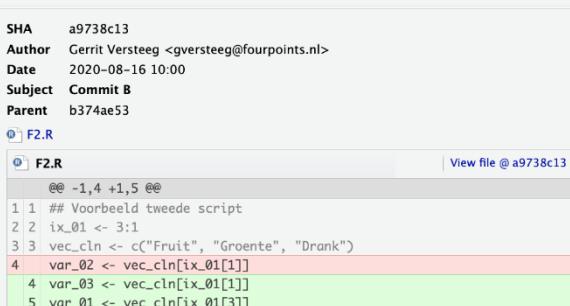



The screenshot shows the RStudio interface with the 'Git' tab selected. A red circle highlights the gear icon in the top right corner of the Git panel. Below the panel, the file list shows several files including '.gitignore', 'Project_VCS.Rproj', 'README.md', 'F1.R', and 'F2.R'. The file '.gitignore' is currently selected.



The screenshot shows the RStudio interface with the 'Files' tab selected. A red arrow points from the previous screenshot to this one, indicating the selection of 'F2.R'. The file list shows the same files as the previous screenshot, with 'F2.R' now selected.

- Kies klok in GIT-panel
- Zoek een commit waarin F2.R bestaat met de gewenste inhoud
- Klik op 'View file @ commit-id'
- Klik op Save file



The screenshot shows the RStudio interface with the 'Changes' tab selected. It displays the commit history for the 'master' branch. The commits are listed as follows:

Subject	Author	Date	SHA
origin/master	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-16 10:00	2a280071
Commit C	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-16	48d8ffbd
Commit B	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-16	a9738c13
Commit A	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-16	b374ae53
Initial commit	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-14	d7405748

Below the commit history, the content of 'F2.R' is shown at the commit 'a9738c13'. The code is as follows:

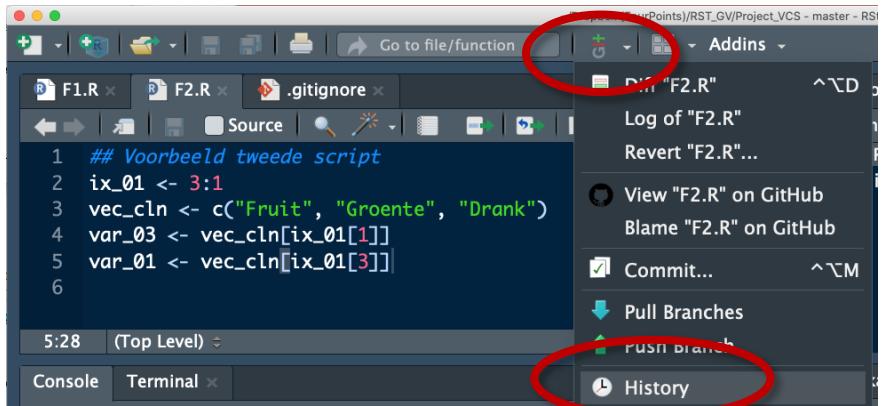
```

@@ -1,4 +1,5 @@
1 ## Voorbeeld tweede script
2 ix_01 <- 3:1
3 vec_cln <- c("Fruit", "Groente", "Drank")
4 var_02 <- vec_cln[ix_01[1]]
5 var_03 <- vec_cln[ix_01[1]]
6 var_01 <- vec_cln[ix_01[3]]

```

GIT – Terug naar een vorige versie (3)

Stel dat je terug wilt naar een versie van een nog bestaande F2.R uit een vorige commit



- Open F2.R in de editor.
- Klik op GIT in menu bar.
- Selecteer History

- Selecteer commits tot je de versie hebt die je wilt gebruiken
- Klik op 'View file @ commit-id'
- Klik op Save file en replace de bestaande versie met de oude

Subject	Author	Date	SHA
HEAD => refs/heads/master	origin/master	Gerrit Versteeg <gversteeg@fourpoints.nl>	2a280071
Commit C	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-16 10:00	48d8ffbd
Commit B	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-16 10:00	a9738c13
Commit A	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-16 10:00	b374ae53
Initial commit	Gerrit Versteeg <gversteeg@fourpoints.nl>	2020-08-14 10:00	d7405748

```

## Voorbeeld tweede script
ix_01 <- 3:1
vec_cln <- c("Fruit", "Groente", "Drank")
var_03 <- vec_cln[ix_01[1]]
var_01 <- vec_cln[ix_01[3]]

```

@@ -1,4 +1,5 @@

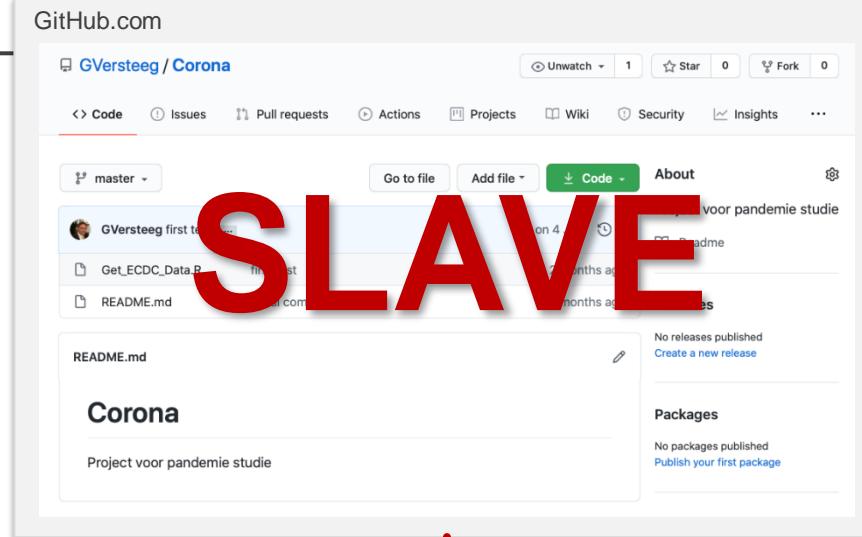
```

1 1 ## Voorbeeld tweede script
2 2 ix_01 <- 3:1
3 3 vec_cln <- c("Fruit", "Groente", "Drank")
4 4 var_02 <- vec_cln[ix_01[1]]
4 var_03 <- vec_cln[ix_01[3]]
5 var_01 <- vec_cln[ix_01[3]]
```

Name	Size	Modified
..		
.gitignore	58 B	Aug 16, 2020, 1:53 PM
Project_VCS.Rproj	205 B	Aug 14, 2020, 12:59 PM
README.md	60 B	Aug 14, 2020, 12:59 PM
F1.R	101 B	Aug 15, 2020, 1:44 PM
F2.R	110 B	Aug 16, 2020, 2:43 PM

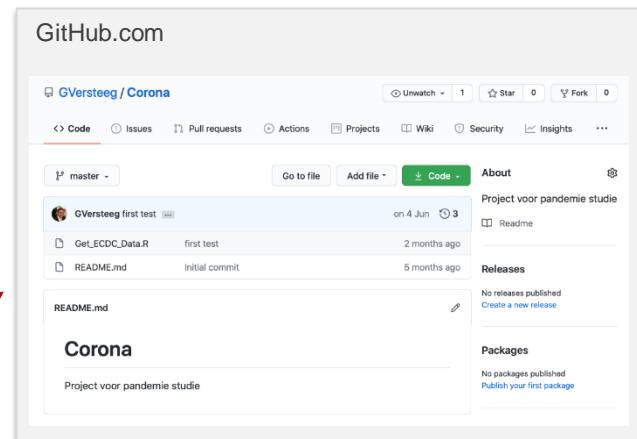
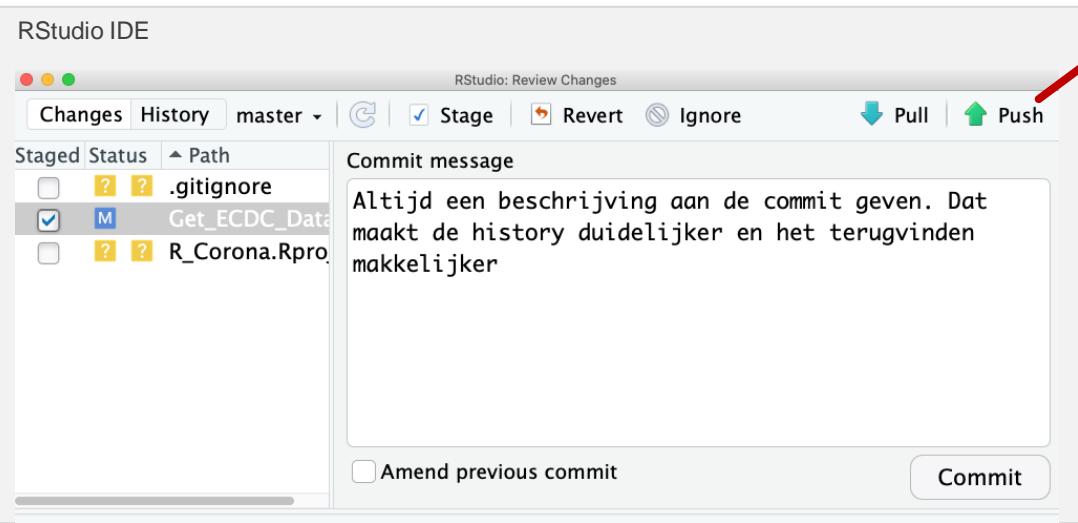
GitHub.com - Concept

- Je eigen repository (project) dat in je werkfolder staat, kun je bewaren in een 'remote' repository op GitHub.com in de cloud.
- Git noemt deze remote repository: origin
- Commando's:
 - Push : de veranderingen in je lokale repo naar de remote repo kopiëren
 - Pull : de veranderingen in de remote repo naar je eigen repo kopiëren
 - Clone : een complete remote repo als nieuw project in RStudio kopiëren



GitHub.com - PUSH

Alle veranderingen die heb ge-'commit'
(m.a.w. waarvan je een snapshot hebt gemaakt)
kun je in jouw remote GitHub-repository kopiëren
met een PUSH



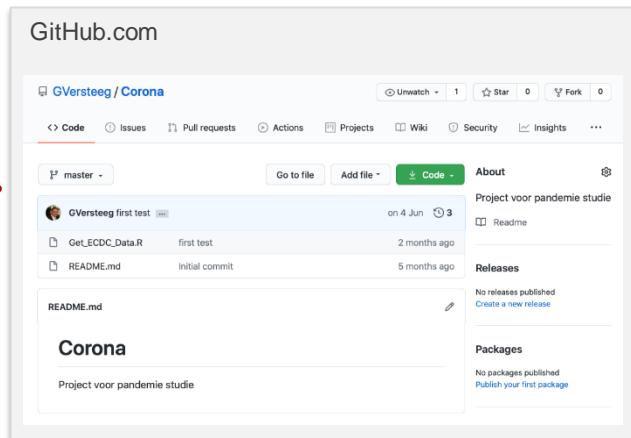


GitHub.com - PULL

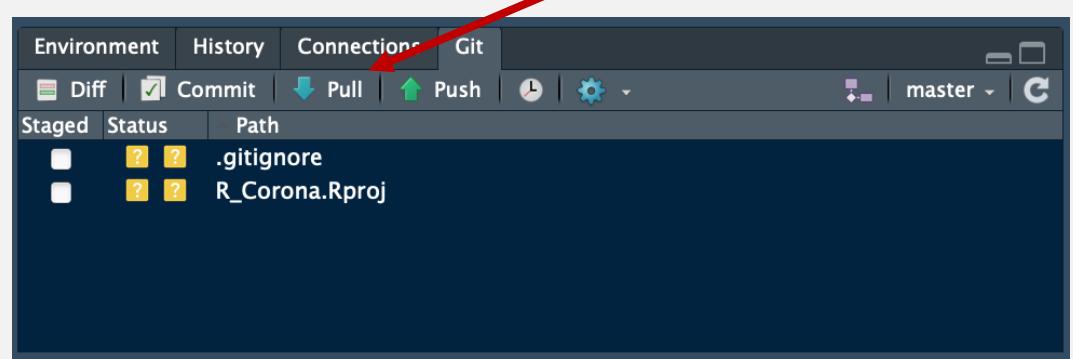
Als je dat toestaat, kunnen andere onderzoekers met jou meewerken en veranderingen aanbrengen op jouw remote repo.

Je kunt deze veranderingen in jouw lokale repo kopiëren door een **PULL**

PULL



RStudio IDE



GitHub.com - PULL

id van de vorige commit

id van de laatste commit

het pull commando

het adres van de remote repo

Git Pull

Close

```
>>> /usr/local/git/bin/git pull
From github.com:GVersteeg/Corona
 97b73b0..d28444e master      -> origin/master
Updating 97b73b0..d28444e
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

master: lokale branche

origin/master: remote branche

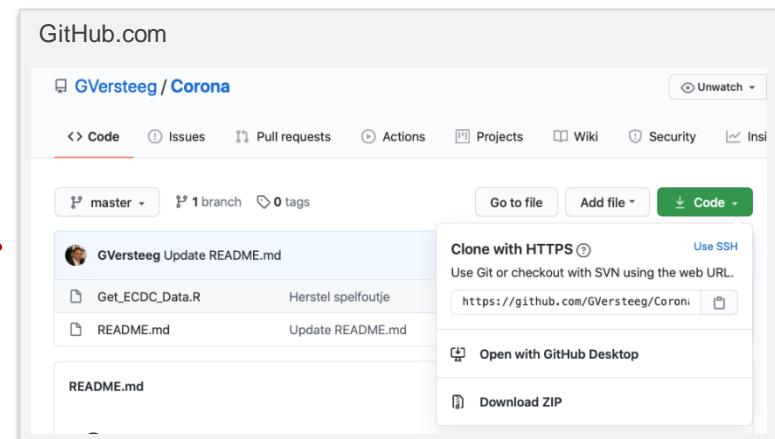
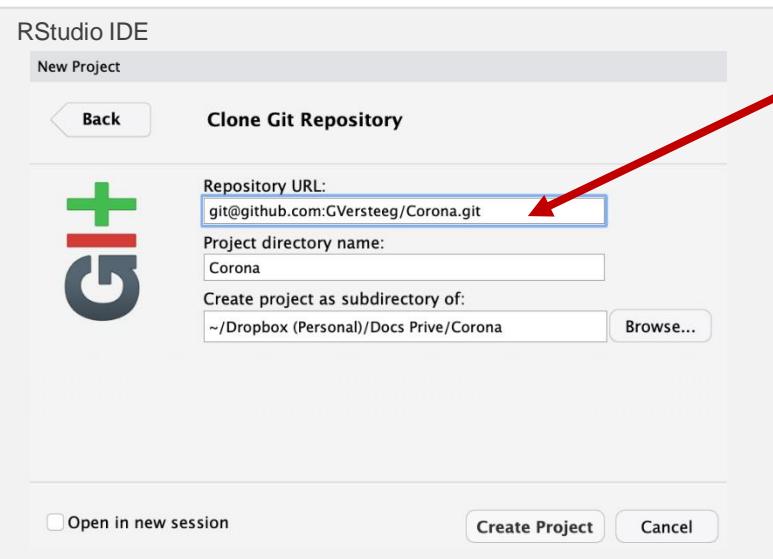
Alleen readme.md blijkt aangepast

type merge: fast-forward
(geen conflicten, zelfde branche)

GitHub.com - CLONE

Als je bij de start van een nieuw project een verse repo in GitHub hebt gemaakt of als je een repo van een ander wilt gebruiken,

dan kun je die hele repo in een eigen project kopiëren binnen RStudio
door een **CLONE**

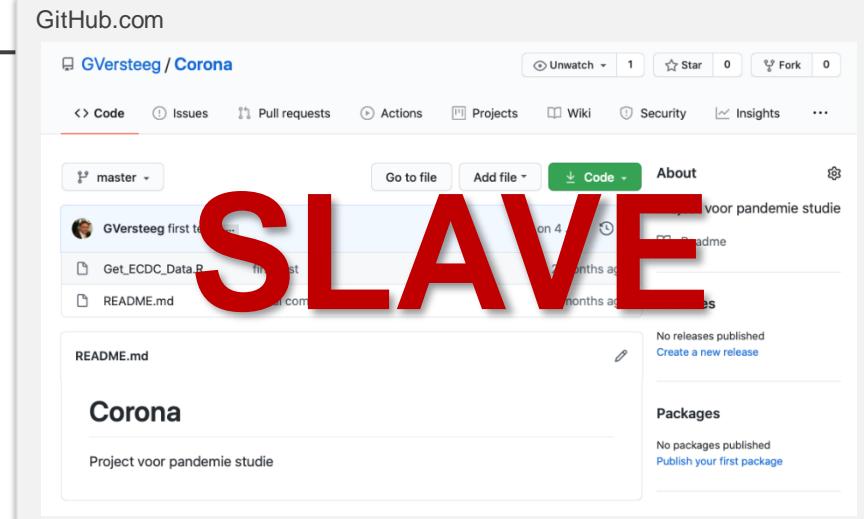
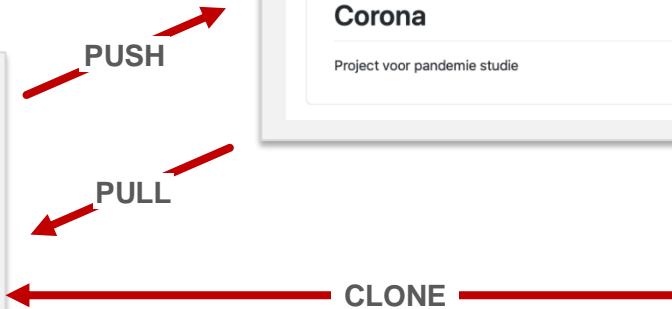


The screenshot shows the GitHub.com repository page for 'GVersteeg / Corona'. It displays the repository's structure with files like 'Get_ECDC_Data.R', 'README.md', and 'Update README.md'. A red arrow points from the word 'CLONE' in the previous slide to the 'Clone with HTTPS' button, which has the URL 'https://github.com/GVersteeg/Corona' listed next to it.

GitHub.com - Concept

Er zijn veel manieren om een ontwikkel-workflow in te richten. In de dagelijkse praktijk van een beginnende data-researchers adviseren we om:

- Gelijk altijd te starten met versiebeheer. Dus elk project heeft een repo in RStudio, gekoppeld aan een private repo in GitHub.
- Voorlopig alleen een ‘**master**’ branche te gebruiken
- De lokale repo in RStudio, te zien als de **MASTER**
- De remote repo in GitHub, te zien als de **SLAVE**

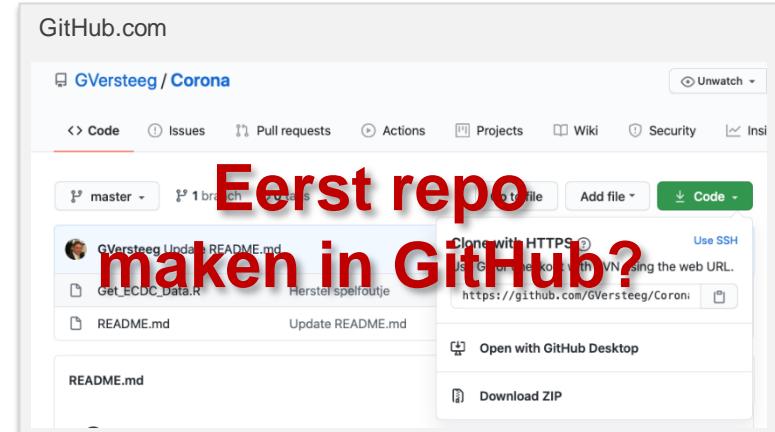
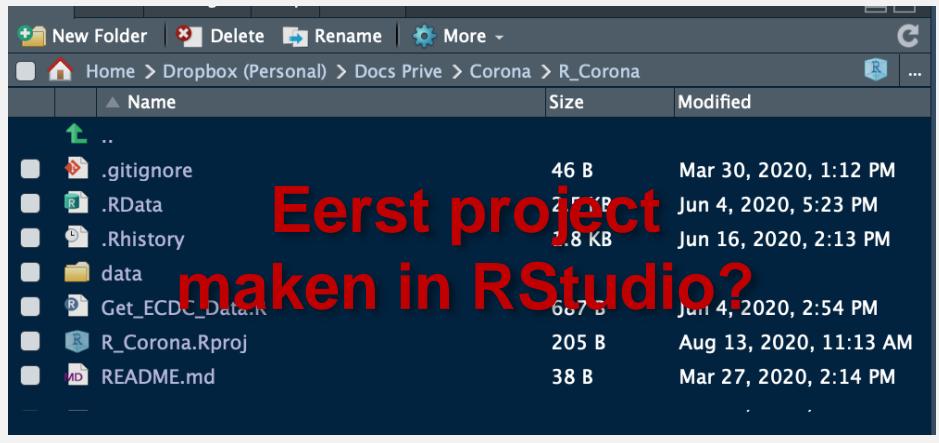


Waar beginnen we een nieuw project?

Als we een nieuw onderzoeksproject beginnen, waar gaan we dan starten?

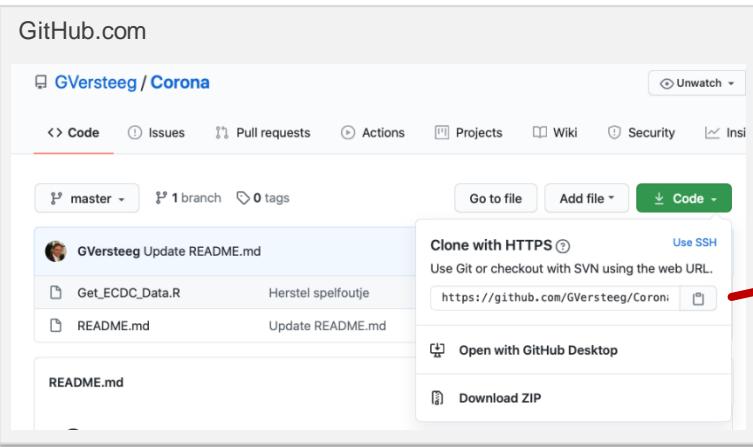
In RStudio of in GitHub?

RStudio IDE



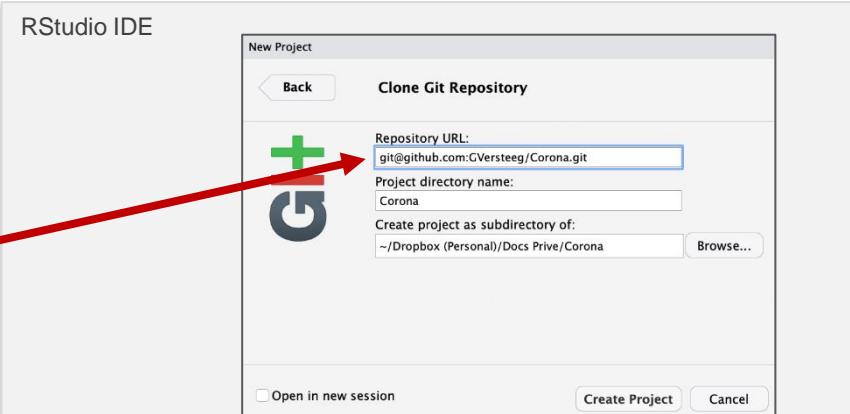
Het starten van een project

GitHub.com



URL

RStudio IDE



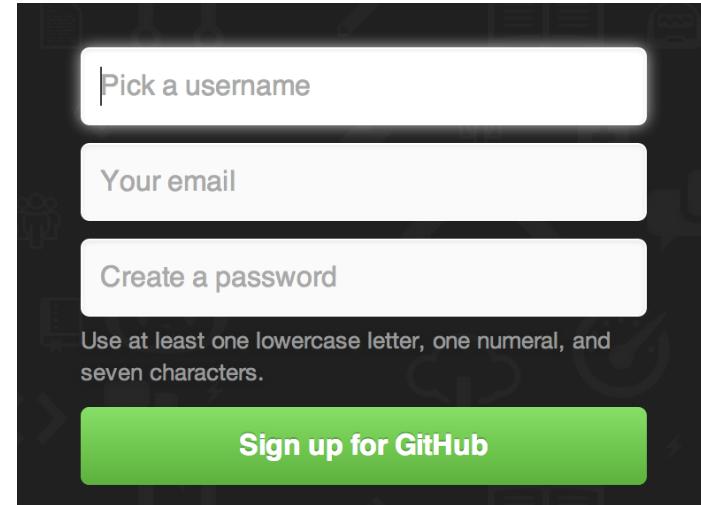
Eerst een lege repo maken in GitHub (met ReadMe-file)
let op: gebruik 'private repo'

Name	Size	Modified
..	46 B	Mar 30, 2020, 1:12 PM
.gitignore	2.5 KB	Jun 4, 2020, 5:23 PM
.RData	1.8 KB	Jun 16, 2020, 2:13 PM
.Rhistory		
data		
Get_ECDC_Data.R	687 B	Jun 4, 2020, 2:54 PM
R_Corona.Rproj	205 B	Aug 13, 2020, 11:13 AM
README.md	38 B	Mar 27, 2020, 2:14 PM

Daarna de URL van de repo gebruiken om een project onder versiebeheer aan te maken in RStudio

Stap 1: Maak je GitHub-account

1. Ga naar <https://github.com>
2. Selecteer 'sign up' en kies een gebruikersnaam die niet al bestaat
3. Geef je zakelijk email-adres en bedenk een wachtwoord
4. Druk op de Sign up for GitHub knop
5. Kies vooralsnog niet voor een betaald plan, dat wordt later geregeld als je opgenomen wordt in de organisatie
6. GitHub stuurt je een verificatie-mail toe
7. Maak je registratie af door middel van de instructies in de eMail.



Opzetten versiebeheer

Stap 2: GIT configureren in RStudio

1. Login op je RStudio-account (https://immd1_dl.nl) en controleer of het pad naar git wordt gevonden door RStudio:
 - a. Menu: Tools/Global Options
 - b. Selecteer links: Git/SVN
 - 1 c. Check of het pad naar git (/usr/bin/git) is gevonden
 - d. Klik op de knop 'OK'

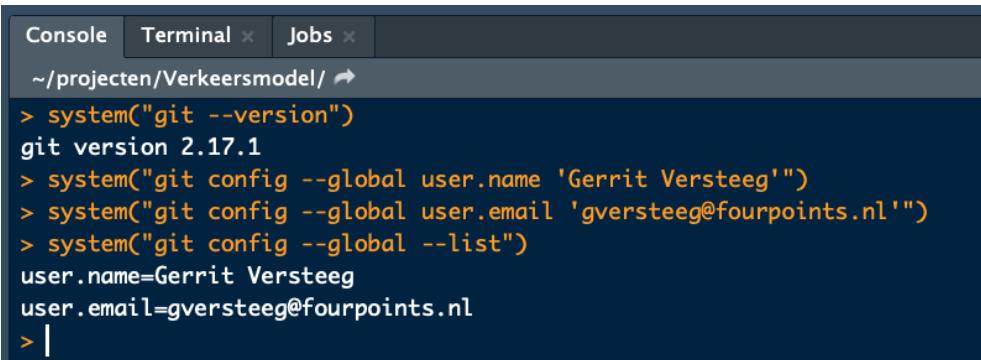
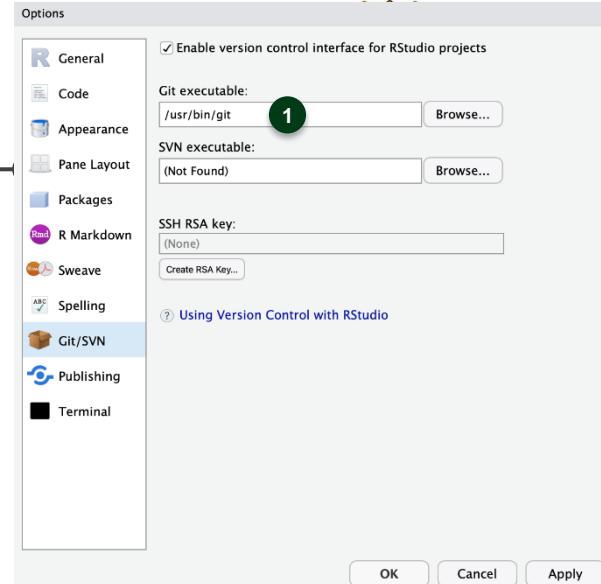
2. Ga naar de console in het hoofdscherm van RStudio:
 - 2 a. Om te controleren of git werkt, vraag de versie van git op:

```
system("git --version")
```
 - b. Voer je volledige naam in met:

```
system("git config --global user.name 'FULL NAME'")
```
 - b. Voer hetzelfde eMail adres in dat je gebruikt bij je GitHub-account met:

```
system("git config --global user.email 'EMAIL ADDRESS'")
```
 - b. Controleer de instellingen met:

```
system("git config --global --list")
```

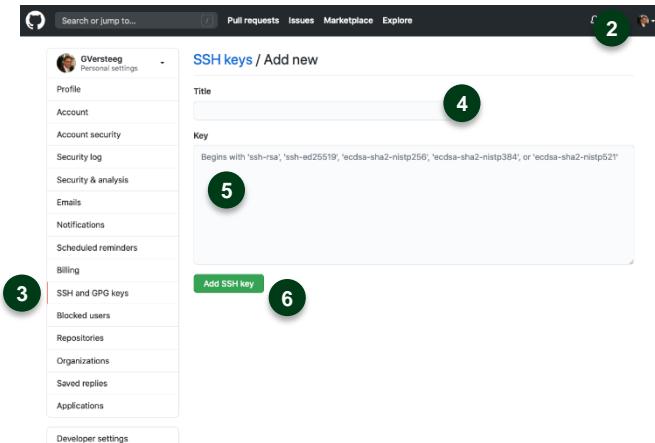
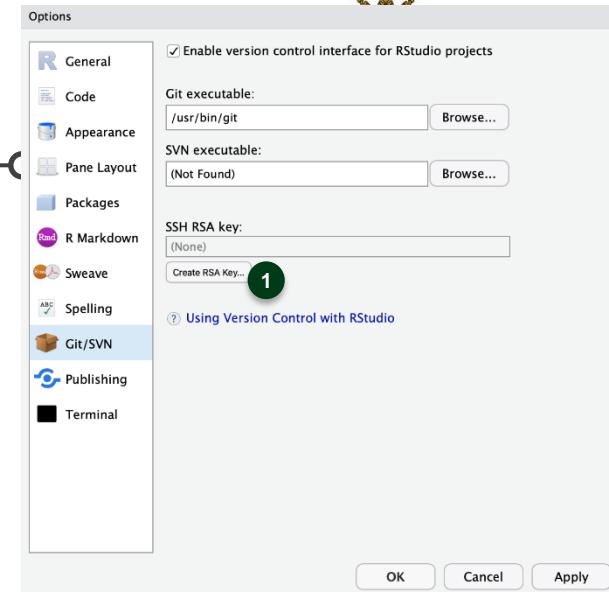


```
~/projecten/Verkeersmodel/ ➜
> system("git --version")
git version 2.17.1
> system("git config --global user.name 'Gerrit Versteeg'")
> system("git config --global user.email 'gversteeg@fourpoints.nl'")
> system("git config --global --list")
user.name=Gerrit Versteeg
user.email=gversteeg@fourpoints.nl
> |
```

Opzetten versiebeheer

Stap 3: RStudio koppelen aan GitHub

1. Om te RStudio te koppelen met GitHub, maken we een sleutelpaar aan:
 - a. Menu: Tools/Global Options
 - b. Selecteer links: Git/SVN
 - 1 c. In hetzelfde scherm, klik op de “Create RSA Key...” knop
 - d. Je kunt een ‘passphrase’ invullen als je dit wilt voor extra veiligheid maar het is optioneel. Klik op de ‘Create’ knop
 - e. Na de creatie van het sleutelpaar, klik ‘close’ om het scherm te sluiten
 - f. In het global options/Git-SVN scherm is er nu een blauwe link verschenen met ‘View public key’. Klik op deze link.
 - g. In de pop-up zie je je nieuwe public key. Gebruik cmd-c of ctl-c om de key te kopiëren op je clipboard. Klik dan op ‘Close’.
2. Nu moeten we deze key nog uploaden naar je GitHub account:
 - 2 a. Login bij je github-account, klik rechtsboven op je avatar en selecteer ‘Settings’ in het pull-down menu
 - 3 b. Selecteer uit de settings-gebieden in de linkerkolom: ‘SSH- and GPG-keys’
 - c. Klik op de groene ‘New SSH key’.
 - 4 d. Vul een titel in voor de key zodat je weet wie je toegang geeft: bijv. ‘RStudio Server DRO-Weert’
 - 5 e. Plak de public key vanuit je clipboard (cmd-v, ctl-v) in het ‘key’-veld
 - 6 f. Klik op de groene knop ‘Add SSH key’



Aanmaken RStudio project

Stap 1: Maak project-repo in GitHub

1. We beginnen altijd in GitHub, dat is het meest simpel:
 - a. Login op je GitHub-account en selecteer eventueel je organisatie
 - b. Klik op knop "Create new repository". Een nieuwe repository scherm opent zich
 - c. Voer een repo-naam in. Kies verstandig, dit wordt de algemene naam van je project
 - d. Voer een korte beschrijving van het project in onder 'Description'
 - e. Kies altijd voor 'Private' tenzij je aan een 'open source' project werkt
 - f. Neem altijd een README bestand op
 - g. De andere opties kun je default laten
 - h. Druk op de groene knop 'Create repository' om het scherm te sluiten
2. Nu moeten we de URL van deze nieuwe repo kopiëren:
 - a. Je bent nu in het hoofd-scherm voor de nieuwe repo
 - b. Klik op de groene knop 'Code'
 - c. Alleen als in het popup-scherm een blauwe tekst met 'Use SSH' staat, klik dan op die tekst.
 - d. Kopieer de URL voor deze repo door op het knopje met het clipboard (rechts van de URL) te drukken

Create a new repository

A repository contains all project files, including the revision history. Already have elsewhere? [Import a repository](#).

Owner * / Repository name * 1

Great repository names are short and memorable. Need inspiration? How about abc123?

Description (optional)
Project voor de uitleg van GIT i.c.m. RStudio 2

Public
Anyone on the internet can see this repository. You choose who can commit.

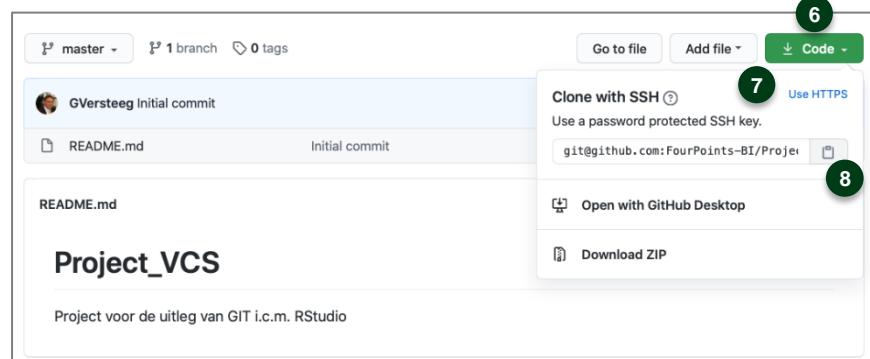
Private
You choose who can see and commit to this repository. 3

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None Add a license: None

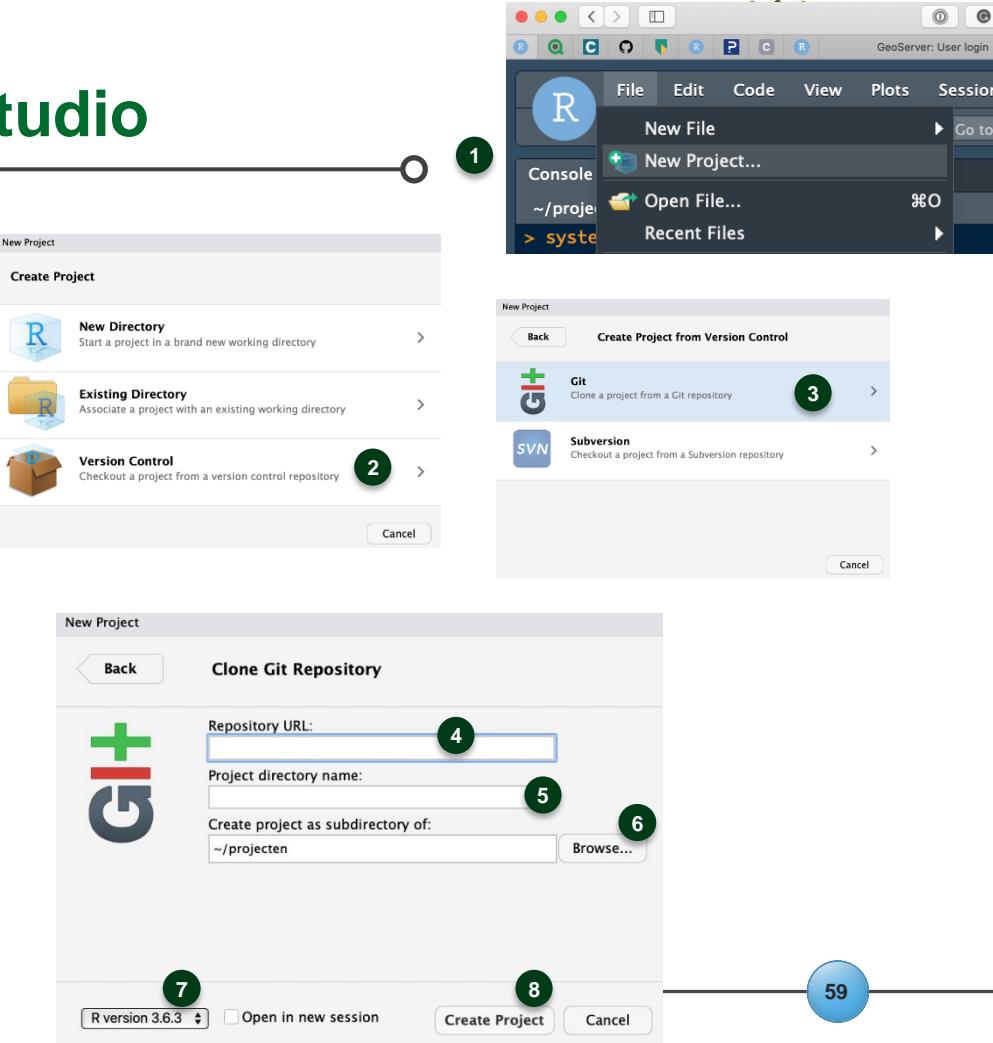
Create repository 5



Aanmaken RStudio project

Stap 2: Clone project in RStudio

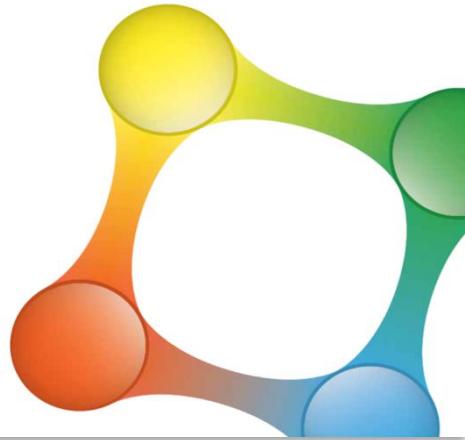
1. Direct na de opvoer van de repo in GitHub gaan we het project clonen in RStudio:
 - a. Login op je RStudio-account
 - 1 b. Kies in het menu: File/New Project...
 - 2 c. In het 'New Project' scherm, kies: 'Version Control'
 - 3 d. In het volgende 'New Project" scherm kies: GIT
 - 4 e. In het volgende 'New Project" scherm, paste je vanuit je clipboard met cmd-v of ctr-v de gekopieerde URL in het veld 'Repository URL'
 - 5 f. Als je in dat veld op TAB drukt, vult RStudio automatisch de repo-naam in het veld 'Project directory name'
 - 6 g. Klik op de knop 'Browse...' rechts naast het veld 'Create project as subdirectory of'
 - h. Navigeer naar je home-folder en dan naar de sub-folder 'projecten'. Kies de folder 'projecten'. Hierin komen al je projecten te staan.
 - 7 i. Kies linksonder in het scherm de door jou gewenste versie van R
 - 8 j. Druk op de knop 'Create project' om het scherm te sluiten en RStudio kloont de Git-repo te clonen in je project-folder.



Demo: Git, GitHub, RStudio

- Setup GitHub
 - Aanmaak GitHub-account
 - Kopiëren github email
- Configureren RStudio voor git
 - Check git vindbaar door RStudio
 - Configureren git (git config --global)
- Aanmaken eerste project
 - Maak Project-repo eerst in GitHub
 - Kopieren repo-URL naar RStudio
 - Clone de GitHub-repo in een nieuw VS-Project in RStudio
- Maken van een tweede commit in RStudio
 - Verander de readme en voeg een script toe
 - Laat git-window zien en wijs op changed files
 - Stage alle aangepaste files en commit met een message
 - Laat in GitHub zien dat er nog niets is veranderd
 - Push de commit naar remote (GitHub.com)
 - Laat in GitHub zien dat het nu aangepast is
- De .ignore file in RStudio
 - Maak een data-folder aan
 - Laat de nieuwe data-folder in de git-window zien
 - Pas de .ignore file aan, zodat data-folder genegeerd wordt.
 - Laat zien dat de nieuwe data-folder uit het git-window is verdwenen en dus niet meer onder versiebeheer valt.

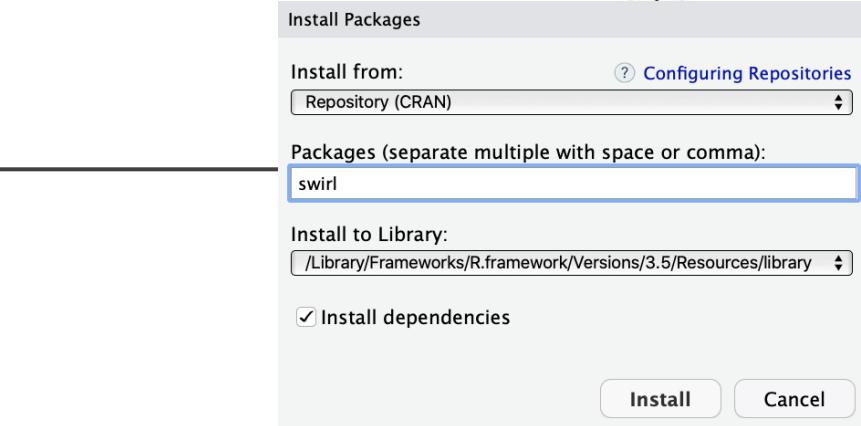
Opdracht



Opdracht SWIRL

- Installeer het R-Package “swirl” in je RStudio account. Laad daarna ‘swirl’
- Start swirl met `swirl()`
- In leerpakket 1 kun je starten met allerlei R-programming oefeningen.
- Voer van R Programming uit:
 - 1: Basic Building Blocks
 - 2: Workspace and Files
 - 3: Sequences of numbers

```
| When you are at the R prompt (>):  
| -- Typing skip() allows you to skip the current question.  
| -- Typing play() lets you experiment with R on your own; swirl will ignore  
| what you do...  
| -- UNTIL you type nxt() which will regain swirl's attention.  
| -- Typing bye() causes swirl to exit. Your progress will be saved.  
| -- Typing main() returns you to swirl's main menu.  
| -- Typing info() displays these options again.
```



1: R Programming: The basics of programming in R
2: Regression Models: The basics of regression modeling in R
3: Statistical Inference: The basics of statistical inference in R
4: Exploratory Data Analysis: The basics of exploring data in R
5: Don't install anything for me. I'll do it myself.

| Please choose a lesson, or type 0 to return to course menu.

1: Basic Building Blocks	2: Workspace and Files
3: Sequences of Numbers	4: Vectors
5: Missing Values	6: Subsetting Vectors
7: Matrices and Data Frames	8: Logic
9: Functions	10: lapply and sapply
11: vapply and tapply	12: Looking at Data
13: Simulation	14: Dates and Times
15: Base Graphics	

9/29/2020