# Building a DataCamp-course: Introduction to Social Statistics

*Joni Oksanen, Tuomo Nieminen,*

*20.01.2018*

## Contents



## Introduction

DataCamp is a web platform which enables building interactive R (and Python) courses. DC courses run inside users web browser and consist of small programming exercises that give users instant feedback and tips as well as reviews. DataCamp is also a company that offers free and paid R and Python courses. The purpose of this document is to demonstrate the structure, source code and functionality of DataCamp-course called *Helsinki Social Statistics* (HSS) which is in use in the course Introduction to Social Statistics (in Finnish: *Johdatus yhteiskuntatilastotieteeseen* or JYT) at the University of Helsinki.
You can find DC's own documentation about building courses at www.datacamp.com/teach/documentation. Building a DC-course requires knowledge of version control Git and GitHub. This document assumes that reader is familiar with basics of Git and GitHub.

### Introduction to Social Statistics (JYT)

Introduction to Social Statistics is an introduction course of applied statistics. Focus is more in the aspect of social science; however, the course is convenient to students of other fields of study. The course is divided in to two parts that are both worth of 5 ECTS credits, but together the courses form a module that covers the basics of statistical methods of social science. Part one's main theme is statistical literacy and part two focuses on the statistical inference. In the Faculty of Social Science the course has an established status as common method course in the early stages of studies. In 2015 the course expanded and it was organized as Finland's first statistics MOOC (Massive Open Online Course): anyone interested in statistics could participate in the course. Every autumn semester the course is completed by doing interactive, self-reviewing exercises that give users automatically feedback. The course also offers session that provides help with exercises. These sessions are led by the lecturer with the help of more experienced students. During the spring and summer semesters the course can be completed with online examinations that are administered by the Open University of Helsinki. The online examinations are also provide throughout the year.

### DataCamp learning platform (DC)

DataCamp (DC) is a web platform that enables users to build interactive R and Python courses. DataCamp's courses can be found at www.datacamp.com. DC courses run inside users web browser and consists of small programming exercises that give users automatically instant feedback and tips as well as reviews. Open DC-courses can be created by anyone. It can be attached as part of university courses when the course doesn't cost anything to students to participate (i.e. MOOCs).

## Helsinki Social Statistics Datacamp-course (HSS)

The university course JYT has its own DC-course called Helsinki Social Statistics (HSS). It can be found at www.datacamp.com/courses/helsinki-social-statistics. The course is open to everyone who has registered to DataCamp.

### Structure of HSS-course

DC-courses is made of chapters, that have exercises. There are three types of exercises avalaible: (1) programming , (2) multiple selection, (3) video. The majority of HSS exercises are R programming.

Programming exercise has four parts: (1) description/background information, (2) instructions, (3) editor, (4) console. Editor and console are from base R, they work the same way as in i.e. RStudio's editor and console. Editor is usually pre-filled with some example codes and user has to fill the provided code to complete the exercise.

### Source code of HSS-course

DC-courses are linked with direct connection to repository on GitHub. Repository contains the codes to build the web page for the course. In GitHub the DC-course has chapters, that are RMarkdown-files (.Rmd). Each DC-course has at least course.yml file that contains the course description and the information about the teachers of the course. Only certain named files affect the building of the course: repository can include files that do not affect to the view of the course in DataCamp at all. You can create a new DC-course (GitHub repository and web page) with few clicks at datacamp.com/teach. The source code of HSS-course can be found at github.com/TuomoNieminen/HelsinkiSocial-Statistics.

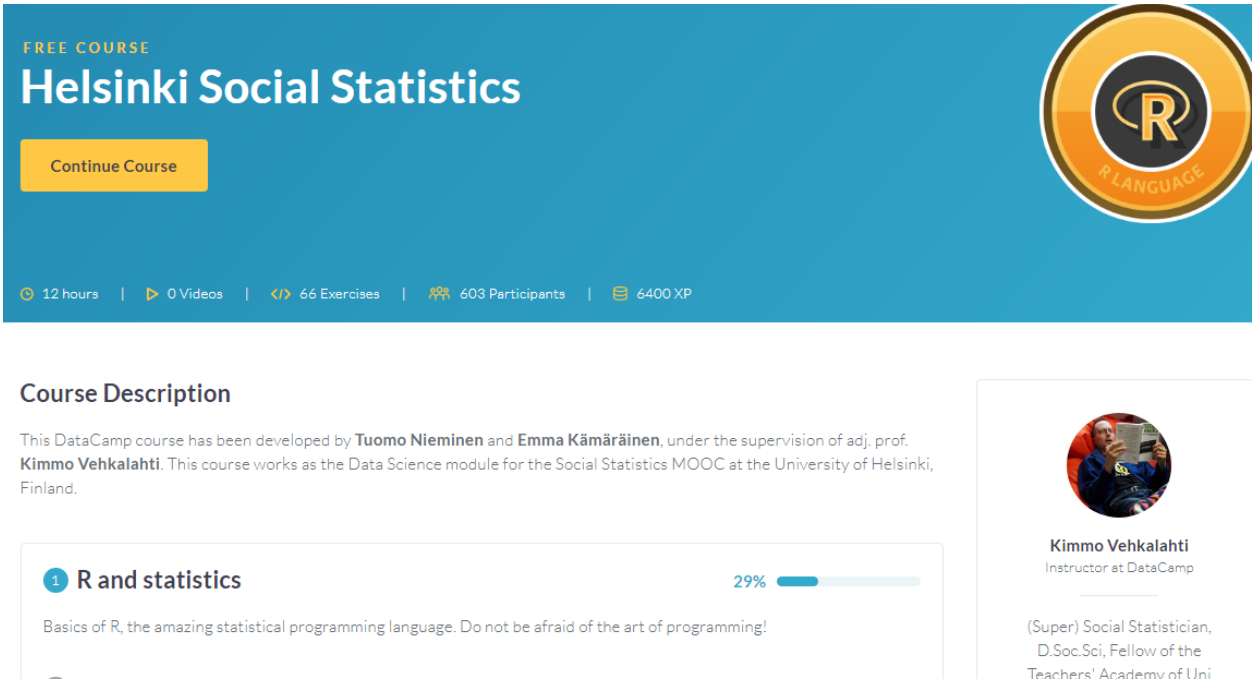


Figure 1: Front page of Helsinki social statistics DC-course

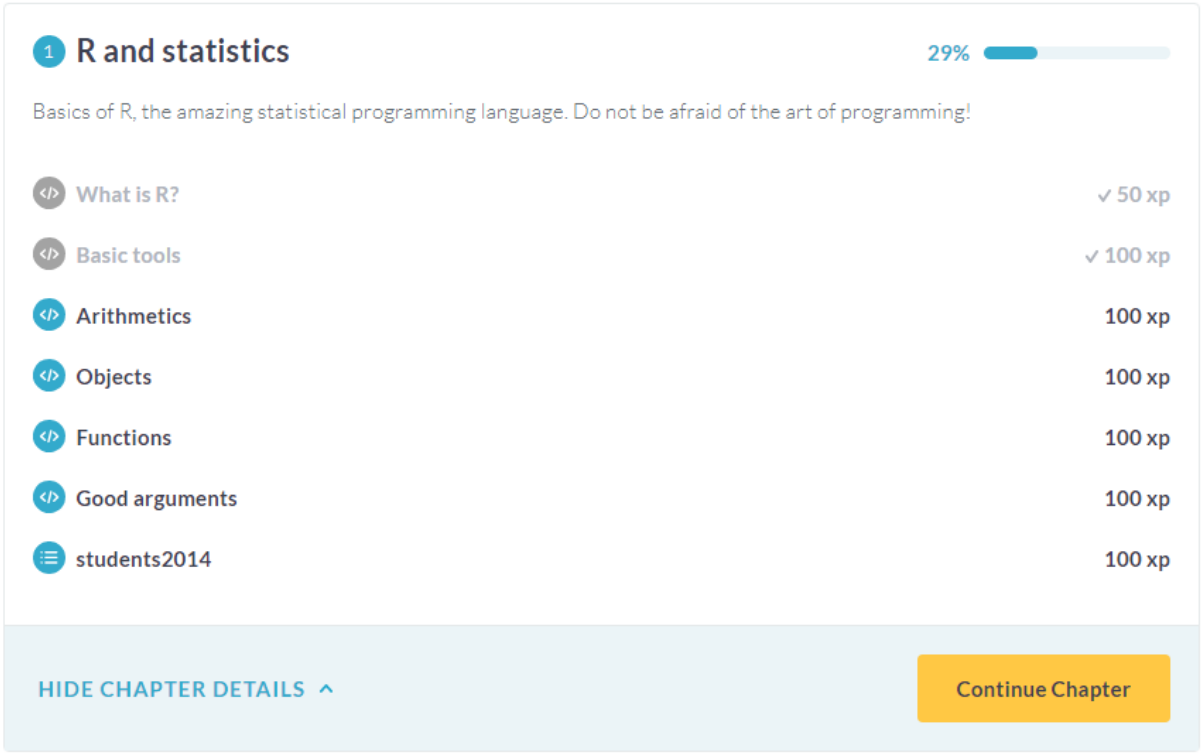


Figure 2: DC-course has chapters that consist of short exercises. User receives experience points (xp) from successfully completed exercises.

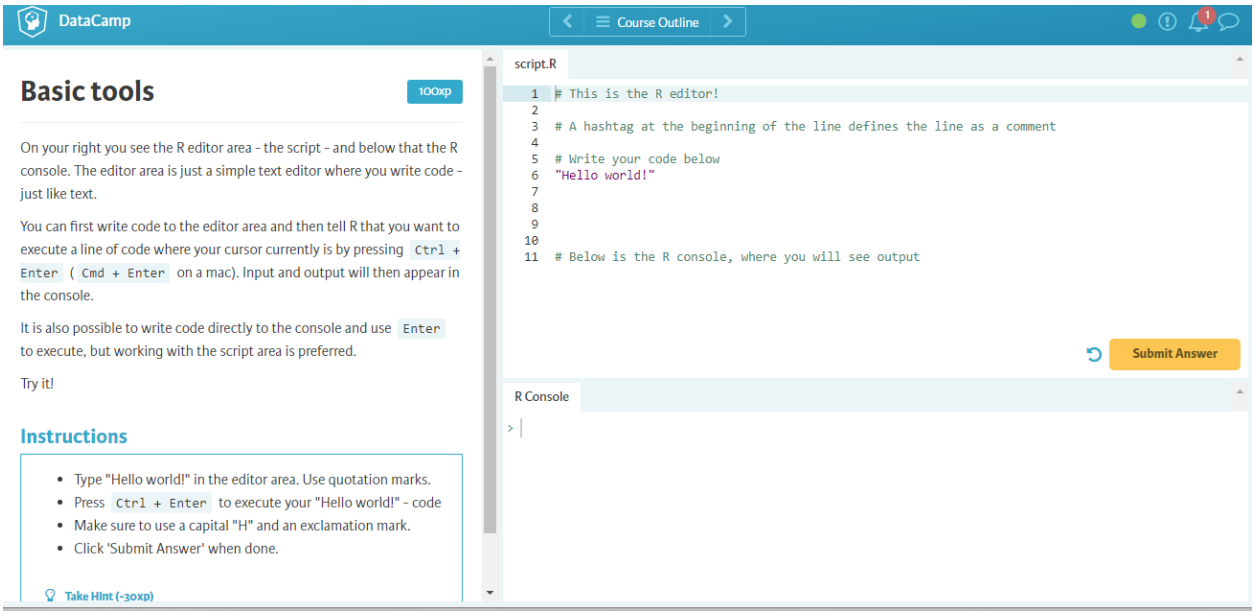


Figure 3: Example of a DC exercise: Second exercises of HSS. Exercise has four parts: 1) description/background information (upper left), (2) instructions (lower left), (3) editor (upper right), (4) console (lower right).

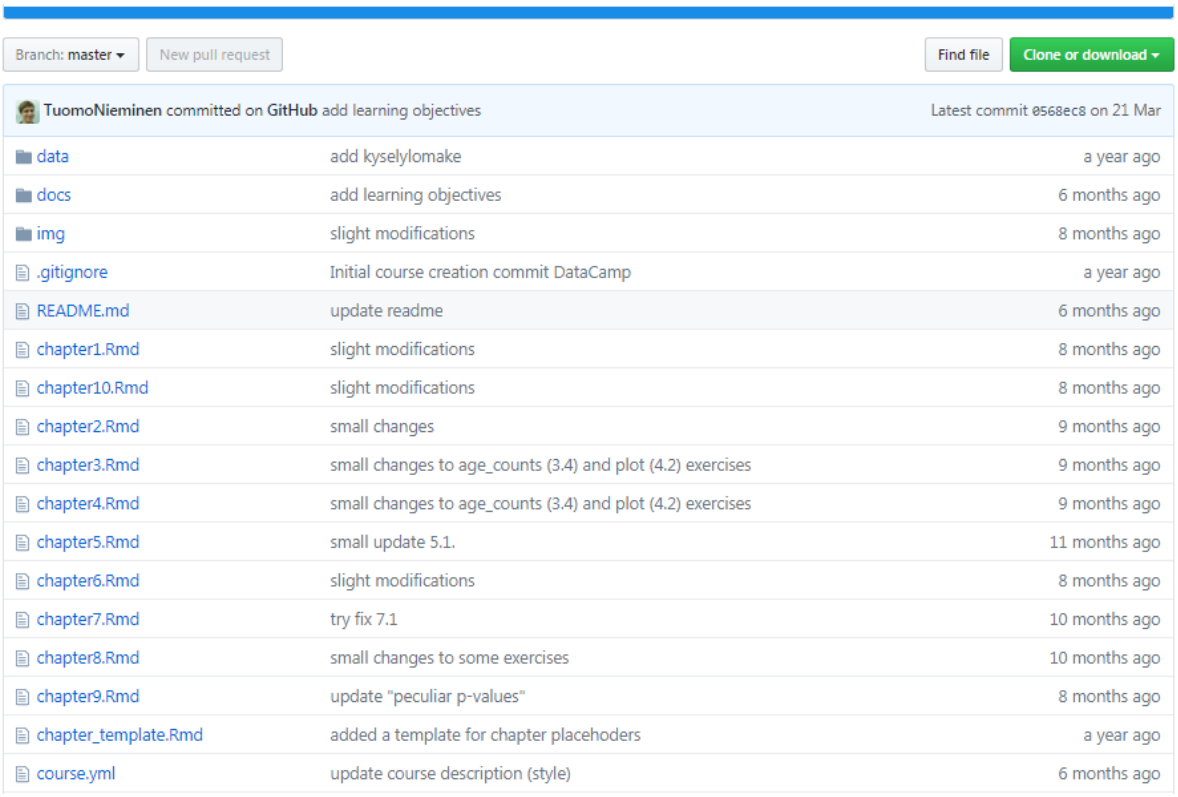


Figure 4: Source of a DC-course is at GitHub. Source has chapter in RMarkdown format (.Rmd), course.yml file and possibly other files.

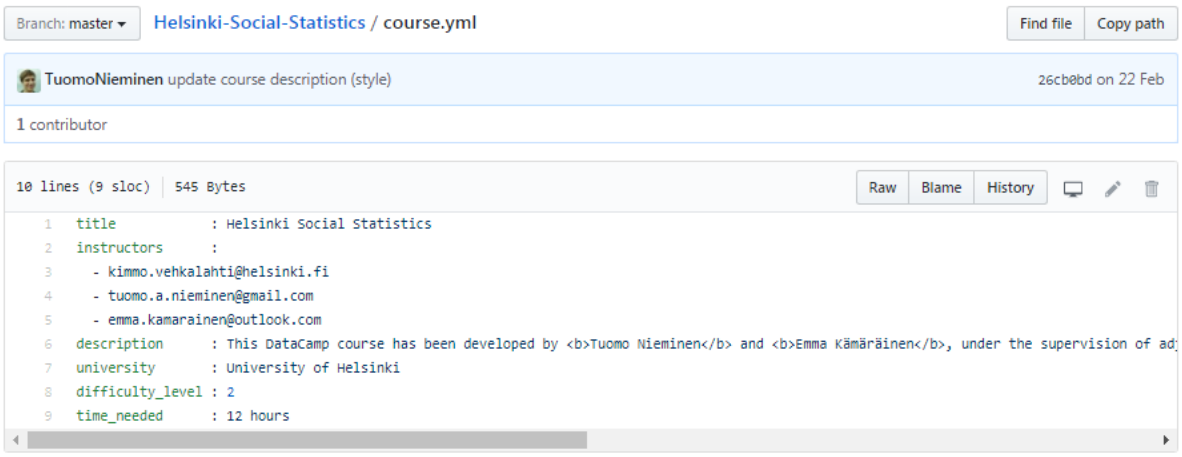


Figure 5: Contents of course.yml-file of Helsinki social statistics

## Editing the HSS-course

Changes in DC-course's source code at GitHub affect immediately to the course on datacamp.com. Every change triggers automatic build of the course on DataCamp's servers. The building process and different versions can be viewed with administrative Dashboard

### Editing rights

Editors of the course must have rights to edit the course repository at GitHub, i.e. with *collaborator* status on GitHub. In addition the emails of admin DC-accounts must be listed in course.yml-file.

### Dashboard

Metafiles concerning the build process can be monitored in DataCamp. Problems might occur in DC if there are errors in the course code. Dashboard for the monitoring can be accessed by clicks in Teach Dashboard-link in courses GitHub's README-part or at datacamp.com/teach.

### Development version of HSS-course

The main branch of HSS-courses repository defines the content of the published version of HSS atwww.datacamp.com/courses/helsinki-social-statistics. Every new branch triggers separate version of the course to DataCamp. The course can be developed further within its own i.e. development branch.

Link to all course versions is found in the Dashboard. HSS dev-version is located at www.datacamp.com/courses/1935. This dev versio is testground for the changes in the course assignments. When the testing of the new elements is finished, you can merge this branch to the main branch. That way each version of the course is available to public.

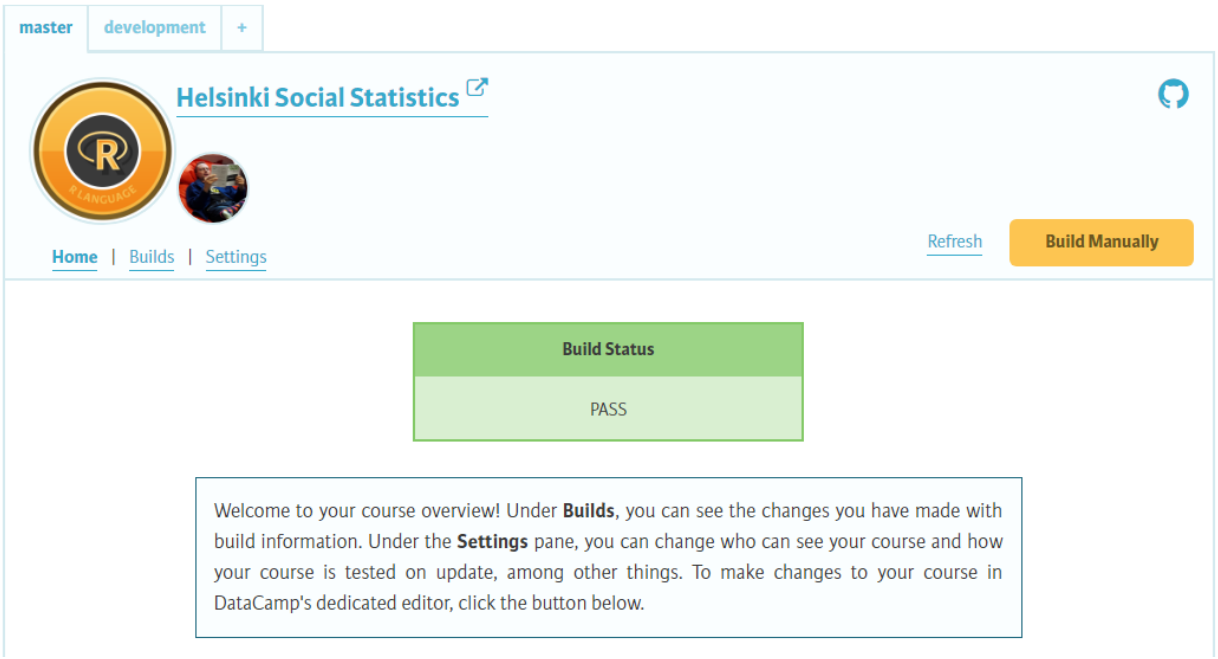


Figure 6: Dashboard of DC course in datacamp.tcom/teach. Dashboard has the different versions and logs of the builds (or build attempts). You can also start to build the course manually from the Dashboard.

## DataCamp-exercises

DC-assigments are short and have one main focus at the main. The goal is that student survives each exercise within 5 to 10 minutes. Especially in HSS exercises try to cheer students into the world of stastical programming, that's why there are not very challenging exercises.

Syntax of the exercises is described below. Each exercise has test which are described separately.

## DC-exercise syntax

As previosly stated, the source code of is divided into chapters (i.e. chapter1.RMD). Each chapter has exercises, that are defined by DC's own, RMarkdown-ish syntax. Below is an example of the syntax, that contains (1) meta, (2) heading, (3) general info, (4) hints, (5) tips, (6) the pre-runned R-code, (7) example code, (8) solution code, (9) tests.

‘‘‘

— type:NormalExercise lang:r xp:100 skills:1
## Heading

General info

*** =instructions
- instruction for the exercise

*** =hint
- tips

*** =pre_exercise_code

```
#   initialisation
```

*** =sample_code

```
# example code (pre-filled'editor')
```

*** =solution

```
# solution
```

*** =sct

```
# tests for the correct answer
```

‘‘‘ A complete template for new programming exercises can be found in HSS-course repository on GitHub

## DC-exercise test functions

Crucial part of the exercises are submission correctness tests (sct), which test the submission of students. Tests must be "approved" to be completed fully. Test should test each step of the exercise and give informative messages when the test is not passed.

With the help of tests, the produced objects/function calls can be compared to the solution. Tests are written with R-package 'testwhat's functions: https://github.com/datacamp/testwhat

As creating exercises it is useful to think how the testing will work. Most of the programming exercises require only three different test functions to create versatile exercises. Three the most useful tests are:

- test_output_contains("output"): Tests the submitted code's output. The wanted output is the character "*output*".
- test_object("object_name"): Tests whether the submitted code has produced object called correctly and is it identical with the solution.
- test_function("function_name", args=c("arg1")): Tests whether submitted code calls the function *function_name* correctly and does it use given argument *arg1* the same way as in examplecodes.

Each test function has argument inccorect_msg, which provides help if the test is not passed. Other aguments can be checked with code i.e. ?testwhat::test_function. Testfunction give students with default-messages even when the arguments are not defined.

## DataCamp groups

A course needs its own group in DC's server. Group gathers the gradebook for the course. Assignments can be set with DC Groups and teacher and assistants can follow the progress of group members with Dashboard.

More information at DC's site: https://www.datacamp.com/groups/education

## DataCamp and Moodle

DC-course is integrated with LTI plug-in (Learning Tools Interoprability) straight to University of Helsinki's MOOC server's course platform (Moodle rooms). The connection automatically logs in to DC with same user as in MOOC and the results are added automatically to MOOC's gradebook.