THE UNIVERSITY OF BRITISH COLUMBIA

# Python Translation of DSCI 100: Introduction to Data Science
## Trevor Campbell (Stats), Joel Ostblom (Stats), Lindsey Heagy (EOAS)

## Project Overview

UBC's introductory data science course, DSCI 100, is currently offered only in the R programming language. While R is popular within statistics, Python is significantly more popular in a wide variety of fields in industry and academia.

### Contributions

A translation of the online DSCI 100 textbook and autograded Jupyter worksheets from R to Python



A new tool, *autotest*, for generating reproducible, reliable test code for autograded Jupyter worksheets

### ### AUTOTEST

## Our Team



**Trevor Campbell**
Associate Professor
Statistics

**Lindsey Heagy**
Assistant Professor
EOAS

**Joel Ostblom**
Assistant Professor of Teaching
Statistics

## Special Thanks



**T. Timbers, M. Lee**
Statistics
R version authors
(with T. Campbell)

**N. Dahiya, G. Ye**
MDS Students
original rough translation
of the textbook

**P. Austin**
EOAS
translation
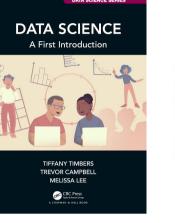guidance

## Textbook
https://python.datasciencebook.ca

**Major accomplishment:** both versions are **identical** in flow

Concepts:
- reading/loading data
- cleaning/wrangling data
- visualization
- classification (KNN)
- regression (KNN, linear)
- clustering (K-Means)
- inference (bootstrap)

Software Conversion:
- tidyverse → pandas
- tidymodels → sklearn, numpy
- ggplot → altair
- dbplyr → ibis
- rvest → requests, beautifulsoup
- git/GitHub, docker, jupyter notebooks



DATA SCIENCE SERIES
**DATA SCIENCE**
A First Introduction
TIFFANY TIMBERS
TREVOR CAMPBELL
MELISSA LEE
CRC Press

## Worksheets
https://worksheets.python.datasciencebook.ca

**Major accomplishment:** both versions are **identical** in flow

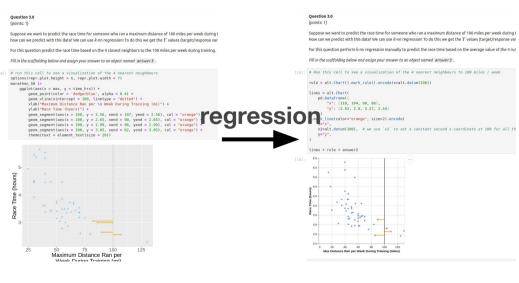### ### AUTOTEST
now included in nbgrader!
https://nbgrader.readthedocs.io/

For every question on every worksheet, we used to have to write test code…and if anything changed, we had to redo the work!!

```
test_6.0 <- function(){
    test_that('Did not create an object named marathon_split', {
        expect_true(exists('marathon_split'))
    })
    test_that('marathon_split should be rsplit (not a tibble)', {
        expect_true('rsplit' %in% class(marathon_split))
    })
    test_that('Did not create an object named marathon_training', {
        expect_true(exists('marathon_training'))
    })
    test_that('marathon_training does not contain 0.75 of the data.', {
        expect_equal(dim(marathon_training), c(696,13))
        expect_equal(digest(int_round(sum(marathon_training$age), 0)), 'fbf206d474a56038342d51f17b0ba4c8')
    })
    test_that('Did not create an object named marathon_testing', {
        expect_true(exists('marathon_testing'))
    })
    test_that('marathon_testing does not contain 0.25 of the data.', {
        expect_equal(dim(marathon_testing), c(233, 13))
        expect_equal(digest(int_round(sum(marathon_testing$age), 0)), '4a09fa6ca41f8e9bff7eea7fd4565399')
    })
    print("Success!")
}
```

**autotest** lets you specify only *what* to test
it figures out *how* to test it for you!

```
# HASHED AUTOTEST marathon_min is None; type(marathon_min); marathon_min;
# HASHED AUTOTEST marathon_best_RMSPE is None; marathon_best_RMSPE;
```
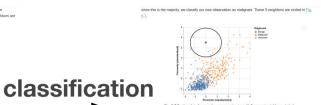
## Example

### R            ### Python



**classification** →

## Analytics
(Jan 1-22, 2024)

### R

| COUNTRY | USERS |
|---|---|
| Canada | 1.9K |
| United States | 262 |
| Brazil | 103 |
| India | 78 |
| China | 60 |
| Mexico | 54 |
| Peru | 50 |

### Python

| COUNTRY | USERS |
|---|---|
| Canada | 75 |
| United States | 26 |
| Brazil | 10 |
| China | 3 |
| India | 3 |
| Chile | 2 |
| Germany | 2 |



## Example

### R            ### Python



**regression** →

both books available from CRC Press
online version offered under a Creative Commons 4.0 license