DSCI 100: Introduction to Data Science

Course Design Intent & Lessons Learned

Tiffany A. Timbers (@TiffanyTimbers)

University of British Columbia

2019/05/24 (updated: 2019-05-27)

DSCI 100 - Introduction to Data Science

Use of data science tools to summarize, visualize, and analyze data. Sensible workflows and clear interpretations are emphasized

Links:

- UBC course calendar
- Course repository on GitHub

Design Intent

Design Intent

1. Introduce Data Science to undergraduates in an authentic way
2. Use pedagogical best practices
3. Reduce barriers to entry and success

Learner Personas

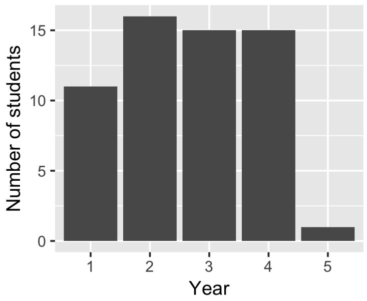
intended audience

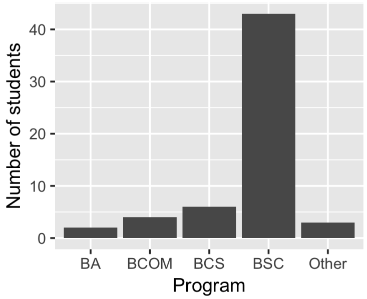
prerequisite: MATH 12

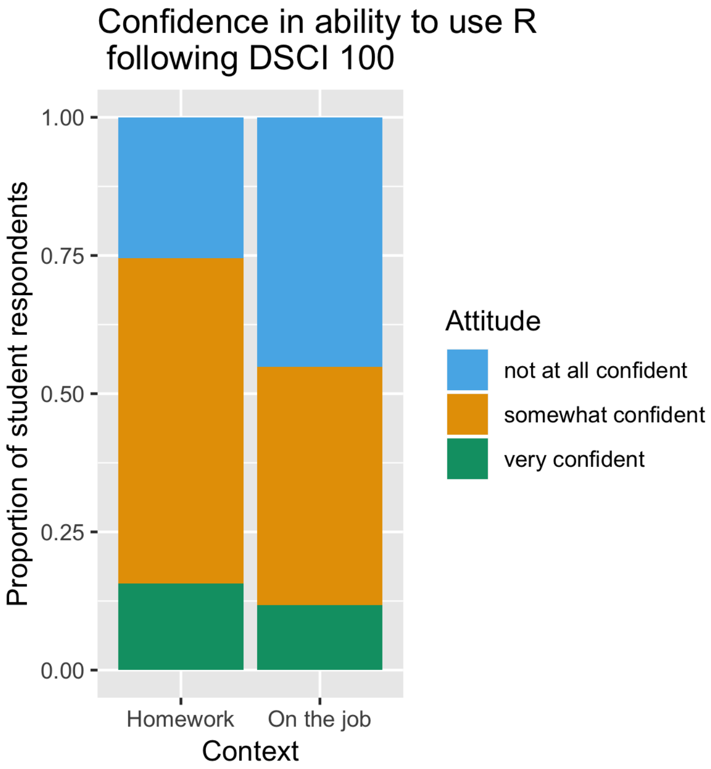
Emily is a first year undergraduate who is thinking of becoming a data analyst after she graduates. She only has grade 12 Math. She has heard of the R programming language and knows it is used at her university in many statistics courses (which she plans to take) but never used it (or any other programming language) before. She is finding university very financially challenging and can only afford a Chrome book.

Mohammed is fourth year Biology undergraduate who plans to attend graduate school next year. His undergraduate honours thesis project requires analysis of data sets that are too big to be opened in JMP, the only data analysis software he has ever used. His thesis supervisor suggested he learn R to do the analysis. The supervisor gave him a book to read to do this, but Mohammed is having a hard time staying motivated doing this on his own without any feedback.

Our actual learners







High-level learning goals of this course

At the end of the course, students will know how to:

- Use modern reproducible tools (Jupyter notebooks, R, `tidyverse` & `caret` R packages) to do data analysis
- Recognize when a data science problem can be solved by classification, regression* or clustering (* *in a predictive context only*)
- Use R to solve classification, regression and clustering problems, and correctly interpret and communicate the results

Course organization and mechanics

- Two 90 min meetings a week (lecture + tutorial)
- 3/4 flipped classroom
- paperless course
- ~ 60% of assessments were autograded

Deliverable	% grade
Lecture worksheets	5
Tutorial problem sets	15
Group project	20
Two quizzes/exams	20
Final exam	40

Introduce Data Science to undergraduates in an authentic way

- Practice what you preach (e.g., teaching materials & tools)
- Give students many opportunities to learn and practice Data Science tools and workflows

Textbook/readings

No modern yet accessible textbooks available that are suitable for our target learners... at least that I am aware of...

... so we wrote our own using the `bookdown` R package!

An Introduction to Data Science

by Tiffany Timbers, Melissa Lee & Samuel Hinshaw

https://ubc-dsci.github.io/introduction-to-datascience/

- still a work in progress (planned edits/updates happening this summer)
- open source and licensed CC BY 2.5 CA

Lecture worksheets & tutorial homework

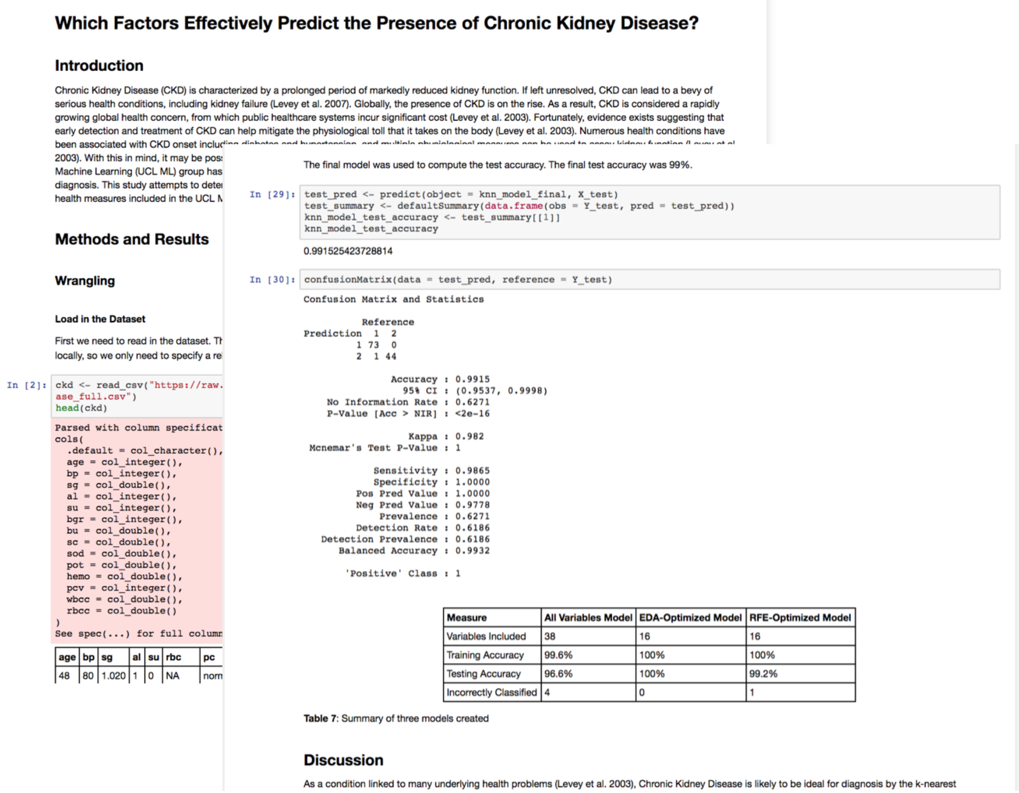
Examples of DSCI 100 worksheets:

- worksheet_01
- worksheet_08

- Jupyter notebooks are literate code documents similar to R Markdown
- Markdown and LaTeX rendering in developing environment makes them easier to read while editing
- notebooks can be manually or autograded using an open source tool, nbgrader

Group project

End product is a self-contained reproducible data analysis and report inside a Jupyter notebook



Which Factors Effectively Predict the Presence of Chronic Kidney Disease?

Introduction

Chronic Kidney Disease (CKD) is characterized by a prolonged period of markedly reduced kidney function. If left unresolved, CKD can lead to a bevy of serious health conditions, including kidney failure (Levey et al. 2007). Globally, the presence of CKD is on the rise. As a result, CKD is considered a rapidly growing global health concern, from which public healthcare systems incur significant cost (Levey et al. 2003). Fortunately, evidence exists suggesting that early detection and treatment of CKD can help mitigate the physiological toll that it takes on the body (Levey et al. 2003). Numerous health conditions have been associated with CKD onset incluc
2003). With this in mind, it may be pos
Machine Learning (UCL ML) group has
diagnosis. This study attempts to dete
health measures included in the UCL M

Methods and Results

Wrangling

Load in the Dataset

First we need to read in the dataset. Th
locally, so we only need to specify a re

```
In [2]: ckd <- read_csv("https://raw.
        ase_full.csv")
        head(ckd)
```

```
Parsed with column specificat
cols(
  .default = col_character(),
  age = col_integer(),
  bp = col_integer(),
  sg = col_double(),
  al = col_integer(),
  su = col_integer(),
  bgr = col_integer(),
  bu = col_double(),
  sc = col_double(),
  sod = col_double(),
  pot = col_double(),
  hemo = col_double(),
  pcv = col_integer(),
  wbcc = col_double(),
  rbcc = col_double()
)
See spec(...) for full column
```

age	bp	sg	al	su	rbc	pc
48	80	1.020	1	0	NA	norn

The final model was used to compute the test accuracy. The final test accuracy was 99%.

```
In [29]: test_pred <- predict(object = knn_model_final, X_test)
         test_summary <- defaultSummary(data.frame(obs = Y_test, pred = test_pred))
         knn_model_test_accuracy <- test_summary[[1]]
         knn_model_test_accuracy
```

```
0.991525423728814
```

```
In [30]: confusionMatrix(data = test_pred, reference = Y_test)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2
         1 73  0
         2  1 44

               Accuracy : 0.9915
                 95% CI : (0.9537, 0.9998)
    No Information Rate : 0.6271
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.982
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9865
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9778
             Prevalence : 0.6271
         Detection Rate : 0.6186
   Detection Prevalence : 0.6186
      Balanced Accuracy : 0.9932

       'Positive' Class : 1
```

Measure	All Variables Model	EDA-Optimized Model	RFE-Optimized Model
Variables Included	38	16	16
Training Accuracy	99.6%	100%	100%
Testing Accuracy	96.6%	100%	99.2%
Incorrectly Classified	4	0	1

Table 7: Summary of three models created

Discussion

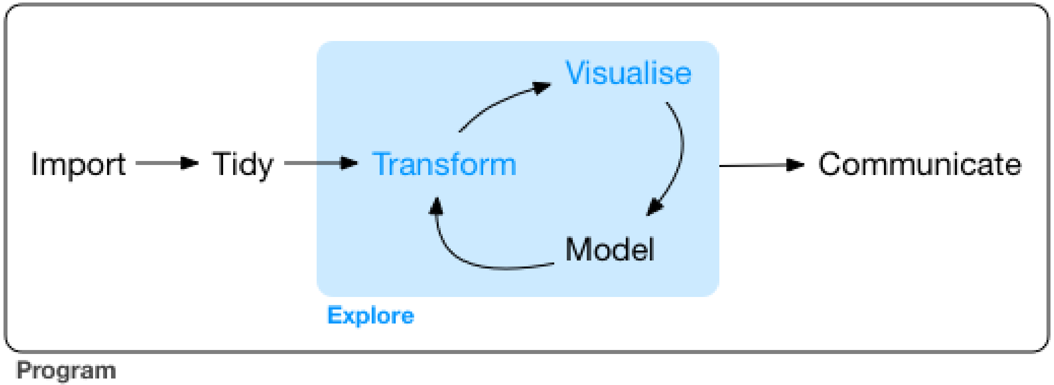
As a condition linked to many underlying health problems (Levey et al. 2003), Chronic Kidney Disease is likely to be ideal for diagnosis by the k-nearest

Use pedagogical best practices

- Case studies
- Practice with immediate feedback
- Active learning activities

Case studies

Each topic was taught within the context of Data Science case studies:

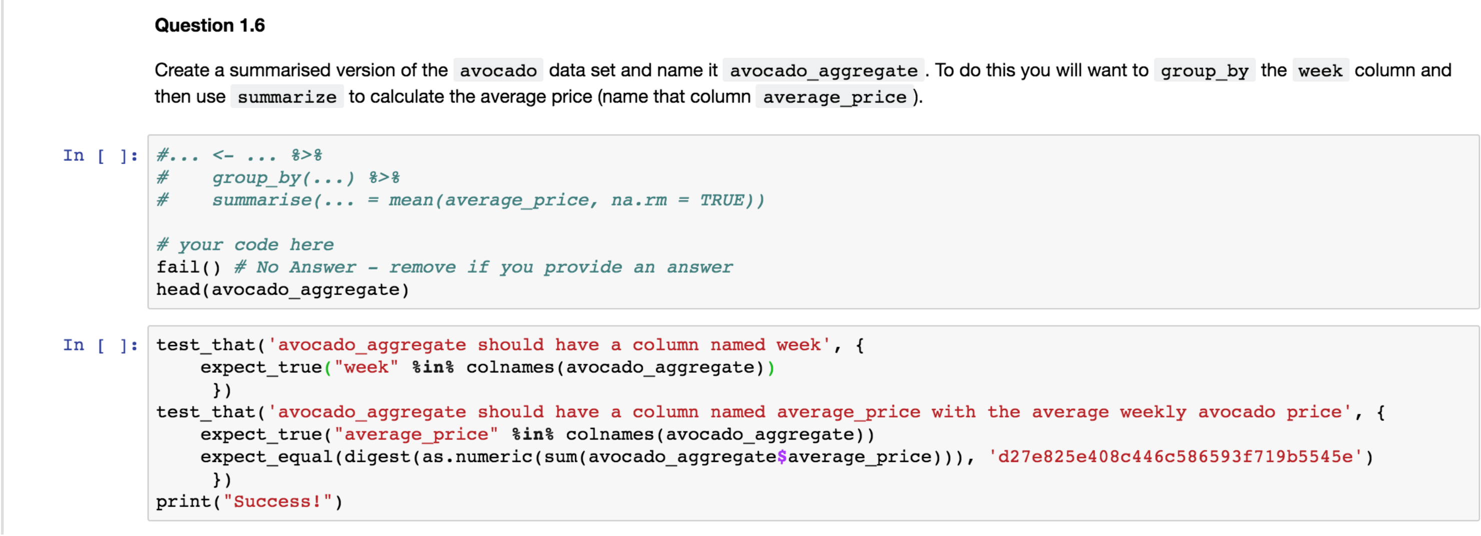


source: R for Data Science by Garrett Grolemund & Hadley Wickham

Practice with immediate feedback

- autograding via nbgrader has the added advantage that students can run the test to assess if their answer is correct
- *they also get to learn about tests as a side-affect*
- we used 3 types of exercises:
  - multiple choice questions
  - code & run with fill in the blanks
  - classic code & run

Code and run examples



Question 1.6

Create a summarised version of the `avocado` data set and name it `avocado_aggregate`. To do this you will want to `group_by` the `week` column and then use `summarize` to calculate the average price (name that column `average_price`).

In [ ]:

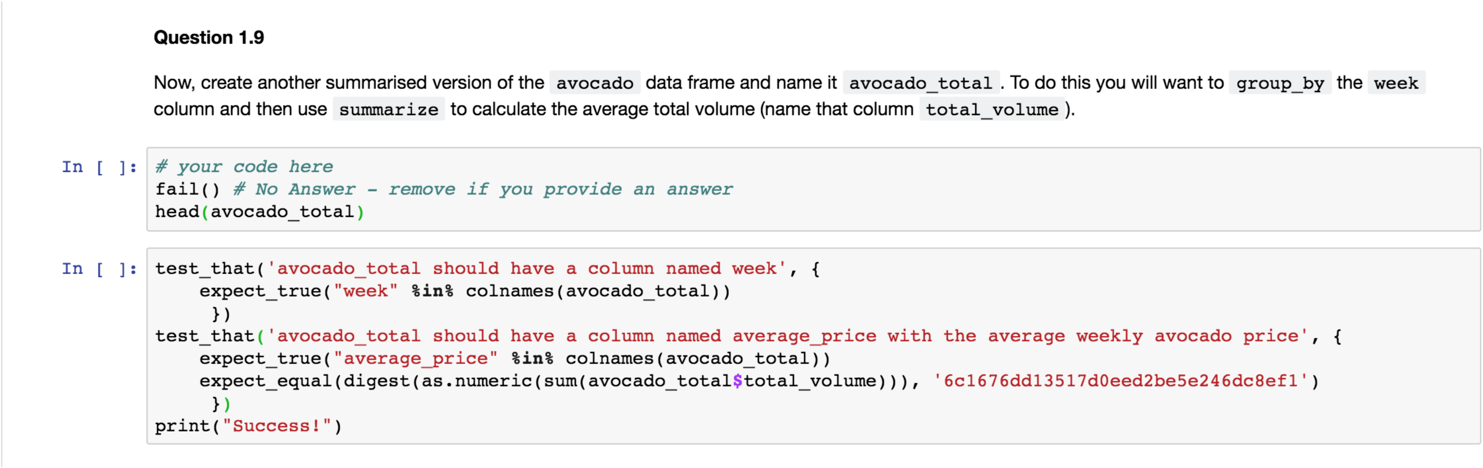
```
#... <- ... %>%
#    group_by(...) %>%
#    summarise(... = mean(average_price, na.rm = TRUE))

# your code here
fail() # No Answer - remove if you provide an answer
head(avocado_aggregate)
```

In [ ]:

```
test_that('avocado_aggregate should have a column named week', {
    expect_true("week" %in% colnames(avocado_aggregate))
     })
test_that('avocado_aggregate should have a column named average_price with the average weekly avocado price', {
    expect_true("average_price" %in% colnames(avocado_aggregate))
    expect_equal(digest(as.numeric(sum(avocado_aggregate$average_price))), 'd27e825e408c446c586593f719b5545e')
     })
print("Success!")
```

Code and run examples



Question 1.9

Now, create another summarised version of the `avocado` data frame and name it `avocado_total`. To do this you will want to `group_by` the `week` column and then use `summarize` to calculate the average total volume (name that column `total_volume`).

In [ ]:

```
# your code here
fail() # No Answer - remove if you provide an answer
head(avocado_total)
```

In [ ]:

```
test_that('avocado_total should have a column named week', {
    expect_true("week" %in% colnames(avocado_total))
     })
test_that('avocado_total should have a column named average_price with the average weekly avocado price', {
    expect_true("average_price" %in% colnames(avocado_total))
    expect_equal(digest(as.numeric(sum(avocado_total$total_volume))), '6c1676dd13517d0eed2be5e246dc8ef1')
     })
print("Success!")
```

Active learning activites

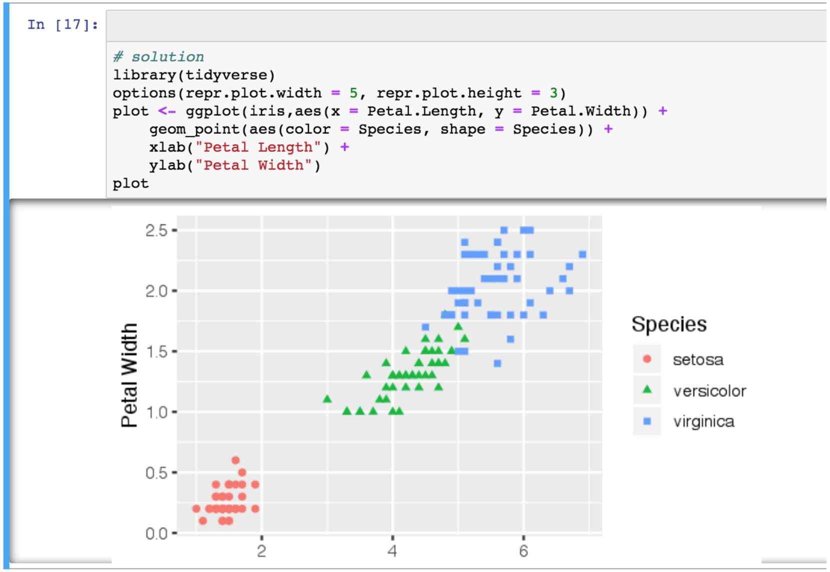
Most lectures and some tutorials had in-class activities, examples include:

- pair/group programming challenges
- tactile exercises
- dance/role-playing

Example pair/group programming challenge

Instructions to students: Can petal length and width be used to separate the Iris flower species? Create an effective plot to answer this question! When you are done, share your code on the class forum.

Then bring everyone together to discuss a solution from the class forum:



In [17]:

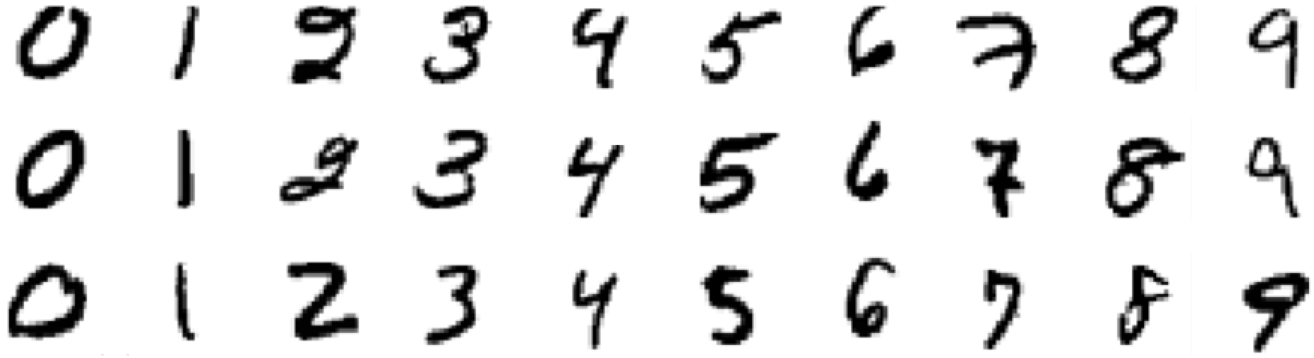
```
# solution
library(tidyverse)
options(repr.plot.width = 5, repr.plot.height = 3)
plot <- ggplot(iris,aes(x = Petal.Length, y = Petal.Width)) +
    geom_point(aes(color = Species, shape = Species)) +
    xlab("Petal Length") +
    ylab("Petal Width")
plot
```

Example pair/group programming challenge

The screens in the DSCI 100 classroom greatly facilitated these activites:

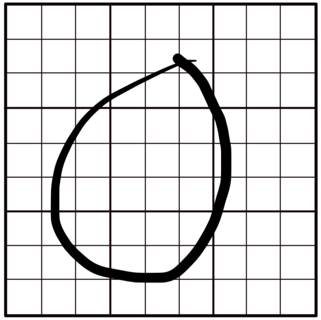
Example tactile exercise

At the beginning of a tutorial on classification of the handwritten digits data set it is very hard for students to understand how you go from images like those shown below to tabular data for k-nn classification.

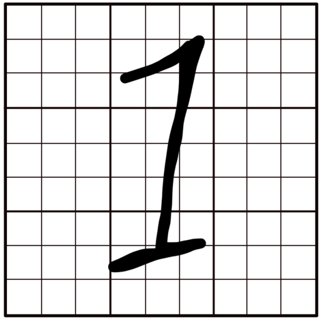


Example tactile exercise

Grid 1



Grid 2



Materials: scissors, tape & printed images (above)

Exercise: For each grid:

- give each square an estimated value for pixel intensity (0 for completely white, 1 for completely black)
- use the scissors to cut the printed images into rows
- with tape, paste the rows together to get a single row

Example of dance/role-playing

To reinforce the algorithm behind Kmeans clustering:

+

Set-up:

- TA's wear coloured t-shirts & students have a post it of each colour
- one post-it note for each student has an asterisk (random)

Exercise:

1. students hold up post-it with asterisk
2. TA's move the center of the students holding up the colour corresponding to their shirt
3. students look for the closest TA and update the colour of the post-it they are holding to match
4. iterate over 2 & 3 until things stop changing

Reducing barriers to entry and success

- Gender and cultural minorities are under represented in STEM
- Aim: remove as many barriers as possible for entry & success in DSCI 100

How?

- Minimal pre-requisites (MATH 12)
- Anonymous class discussion forum (Piazza)
- Formal and public course code of conduct
- **Web server to provide access to homework via the course learning management system (LMS)!**

A JupyterHub server accessed via the course LMS

- Did not want to relying on computer labs and/or managing student installation of course software
- Students access homework and necessary software via a button inside course LMS (e.g., Canvas, EdX)

Demonstration time!

Experiment with JupyterHub yourself!

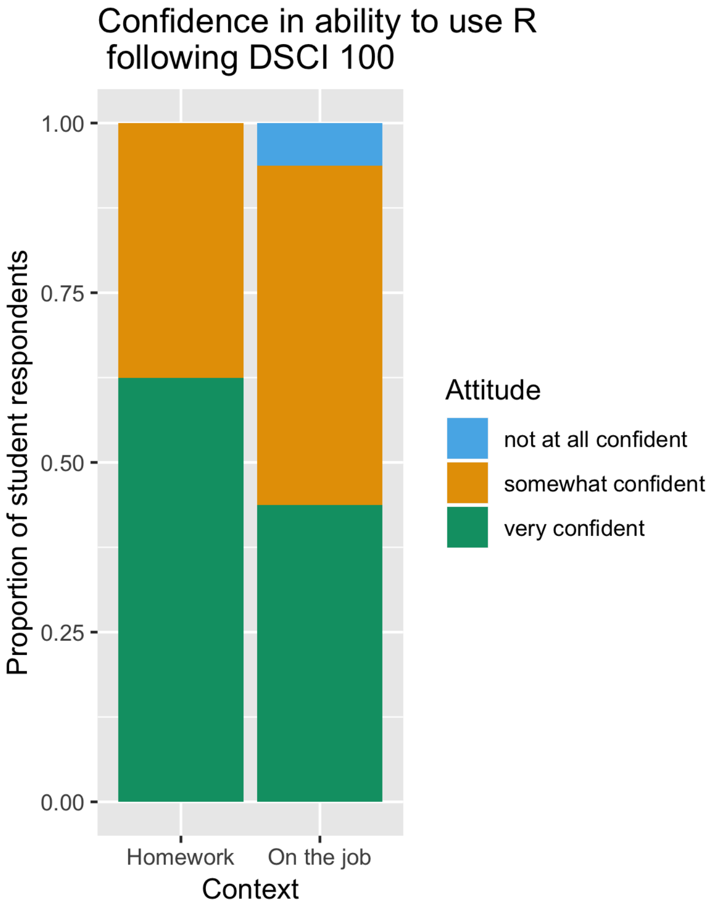
The below links let you access the homework for DSCI 100 via a JupyterHub server that uses Google authentication (and therefore open to everyone):

- worksheet_01
- worksheet_08

Exit survey results

16 students have completed our exit survey, so far…

- **94%** student respondents reported they are **more interested in taking additional Data Science courses**
- **69%** of students said they perceive **every topic in the course as valuable**



Exit survey results

Topics reported as most valuable

- Data Wrangling
- Data visualization
- Modelling/prediction/machine learning
- Practical/hands on work with data
- R programming language
- Working in a group
- Jupyter notebooks
- Git & GitHub
- Scraping data off the web

bolded terms do not intersect these two lists

Topics reported as least valuable

- R programming language
- Git & GitHub
- **Visualizing high dimensional data visualization**
- Scraping data off the web

Lessons learned

Many, but here are the big 3:

1. May not have enough time and learner prior knowledge & motivation to teach Git & GitHub
2. Assessing incoming and exiting knowledge is difficult
3. Have many eyeballs review autograded questions

Acknowledgements

DSCI 100 Development:

- Paul Gustafson
- Matias Salibian-Barrera
- Will Welch
- Nancy Heckman
- Tiffany Timbers
- Melissa Lee
- Samuel Hinshaw
- Melissa Guzman
- Harmeet Gill
- Ian Flores Siaca

DSCI 100 Infrastructure:

- Ian Allison
- Samuel Hinshaw
- The Ha
- Calvin Leung
- Yuvi Pandas

DSCI 100 Teaching Team:

- Tiffany Timbers
- Aaron Quinton
- Harmeet Gill
- Ian Flores-Siaca

Questions?