Introduction to Data Science

T. ARNOLD

Course Wrap-Up

Today we are going to discuss a few final closing thoughts about the course:

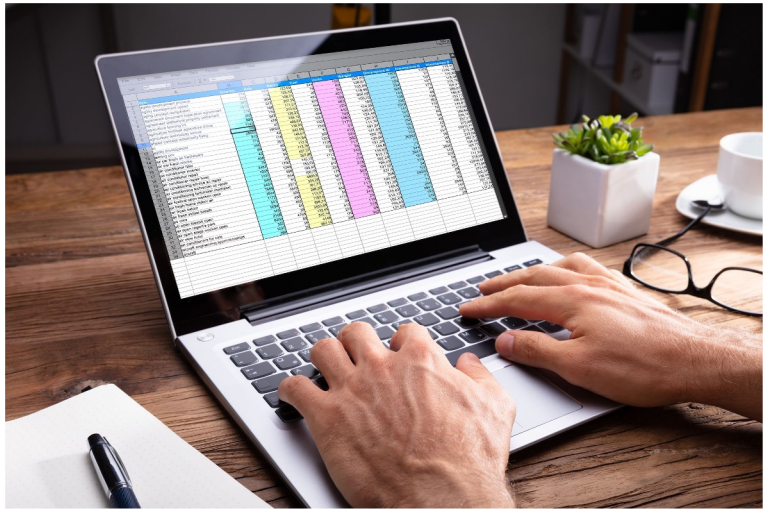
1. data science in the wild
2. data science minor
3. SQL and Python

I will be brief so that we have plenty of time to work on the final project.

1. Data Science in the Wild

Data Science in the Wild

I encourage you to use the techniques in this class in future endeavors. This may be as soon as a course next semester, or it may be a job several years down the road.



I suggest keeping two things in mind:

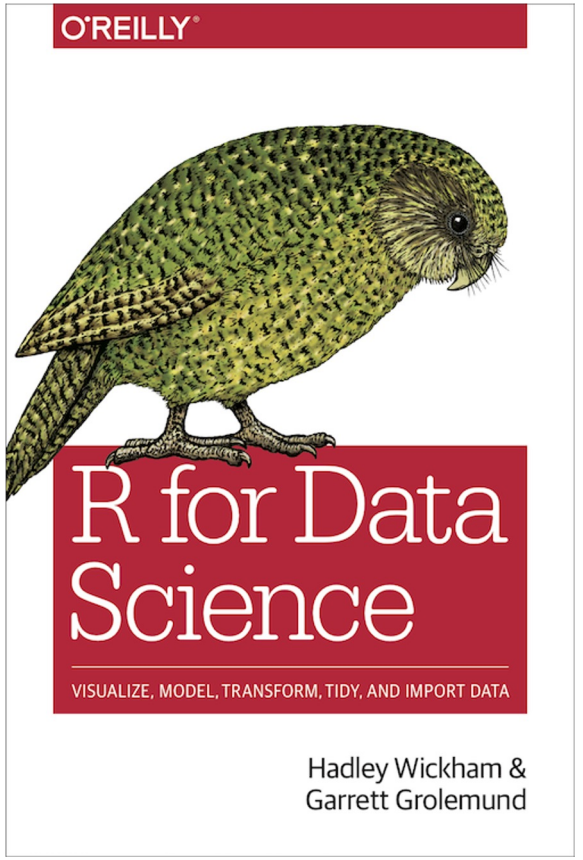
1. We covered the basics in the first 8 weeks, and these will cover most use-cases. Do not get overwhelmed by the more advanced stuff.

2. The most important thing to keep in mind are the class notes about creating data. If you make the basic data well, it is easy to use **ggplot2** + **dplyr**.

Getting Help

If you are trying to get help with R, or just data science in general, here are a few sources of help:

1. our course website (it should still be up)
2. The R for Data Science book
3. R package vignettes
4. GitHub issues
5. ROpenSci and the R Journal

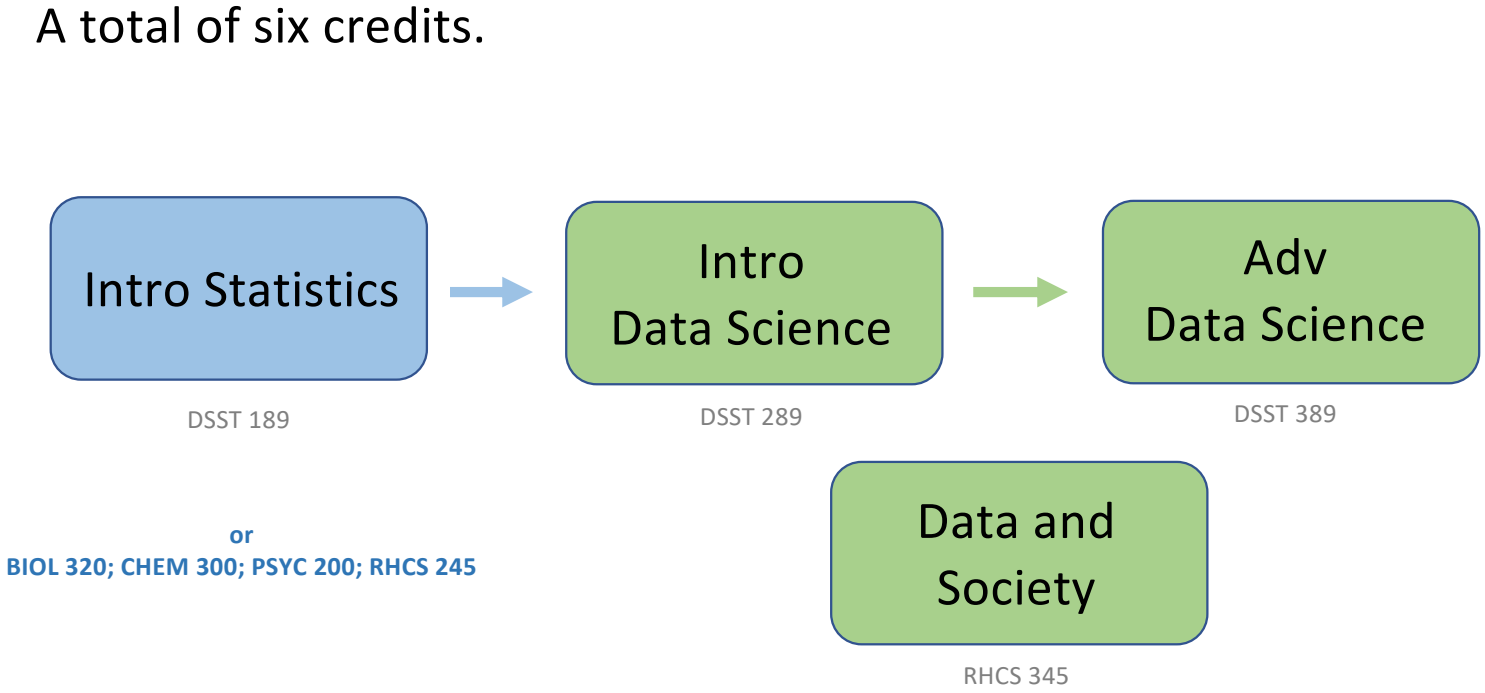


2. DS Minor

T. ARNOLD

Data Science and Statistics Minor: Structure

A total of six credits.



3. SQL and Python

T. ARNOLD

What We Have Learned

I warned you all at the start of the semester that this course can often feel like we are just learning the programming language R, while in fact we are learning more general concepts from various fields of data science.

As a way of giving a course recap and to illustrate this, let's look at two other popular languages for data analysis. This is not a complete introduction to them, but a good starting point.

Structured Query Language (SQL)

A language for manipulating data from a data base. Closely linked to the data verbs from **dplyr**.

```
SELECT calories, sugar FROM food WHERE food_group = "fruit" ORDER BY sugar ;
```

```
food %>%
  filter(food_group == "fruit") %>%
  select(calories, sugar) %>%
  arrange(sugar)
```

```
SELECT mu AS avg(calories) FROM food GROUP BY food_group ;
```

```
food %>%
  group_by(food_group) %>%
  summarize(mu = mean(calories))
```

Python

A language very similar to R, let's see it compares.

Python

A language very similar to R, particularly when you use the popular **pandas** library and Python notebooks.

```
import numpy as np
import pandas as pd
from plotnine import ggplot, geom_point, geom_text, aes

dt = pd.read_csv('notes/data/food.csv')
```

```
dt <- read_csv("notes/data/food.csv")
```

```
(ggplot(dt, aes('calories', 'sugar', color='factor(food_group)')) +
  geom_point())
```

```
dt %>% ggplot(aes(calories, sugar)) + geom_point()
```

Python (cont.)

```
dt[dt['food_group'] == 'vegetable']
```

```
filter(dt, food_group == 'vegetable')
```

```
dt['calories2'] = dt['calories'] * 2
```

```
mutate(dt, calories2 = calories * 2)
```

```
dt[['fiber', 'food_group']]
```

```
select(dt, fiber, food_group)
```

```
dt.sort_values('calories')
```

```
arrange(dt, calories)
```