

ESOC 2014 Introduction to Data Science

Adriana Picoral, PhD (she/her) adrianaps@email.arizona.edu The University of Arizona sits on the

2020-08-04

Contents

1	Syllabus	5
1.1	Course Description	5
1.2	A Few Words on R and Coding	6
1.3	A Few Words on Technology	6
1.4	Readings	7
1.5	Assignments with Grade Breakdown	7
1.6	Requirements for the Course	9
1.7	Course Schedule	9
1.8	Final Project	13
1.9	Honors Students' Requirements	13
1.10	Student Accommodations	13
1.11	Attendance, Due Dates, and Missing Work	14
1.12	Course Conduct and Campus Policies	14
1.13	Code of Conduct	15
1.14	How to Ask For Help	17
2	What's data science?	19
2.1	Before class #1	19
2.2	What's data science?	19
2.3	What does a data scientist do?	20
2.4	Data Science Workflow	20
2.5	Before class #2	21
2.6	What's data?	21
2.7	What does data analysis look like?	29
3	Exploring our IDE (Rstudio)	31
3.1	Before class	31
3.2	Create an R Project	32
3.3	Operations and Objects	33
4	R Basics	35
4.1	Before Class	35
4.2	Dataframes	35
4.3	Slicing you dataframe	36

4.4	Adding new variables (i.e., columns) to your dataframe	37
4.5	Descriptive stats on dataframes	38
4.6	Note on coding style	38
4.7	Organizing Data Science Projects	39
5	Tidyverse	41
5.1	Before Class	41
5.2	Installing Packages	41
5.3	Load Data with Tidyverse	42
6	Data Tidying	43
7	Descriptive Statistics	45
7.1	Before class	45
8	Data Visualization	47
9	Data Case Study 1	49
10	Data Case Study 2	51
11	Getting Data	53
12	Data Case Study 3	55
13	Markdown	57
14	Data Case Study 4	59
15	Analysis Reporting	61

Chapter 1

Syllabus

Tuesday & Thursday 12:30pm - 1:45pm

- Final Exam Date: December 16 (Wednesday) 1:00pm - 3:00pm

Tuesday & Thursday 2:00pm - 3:15pm

- Final Exam Date: December 14 (Monday) 3:30pm - 5:30pm

Office Hours/Free help session/Work time

- Tuesday 9:30am - 11:00am & Wednesday 1:00pm - 2:30pm

1.1 Course Description

This course provides an introduction to the various skills and considerations required for data management and analysis in business, education, and science. Particular attention will be given to learning how to use the free and open-source computing environment R, as well as one of its data visualization packages, ggplot2.

Upon completion of this course, students should be able to:

1. Understand data science as a process
2. Think critically about how data are stored, used, managed, analyzed, and presented
3. Engage in computational thinking and modeling
4. Experiment with skills required for large-scale data analysis

This course is designed to be interactive and hands-on.

1.2 A Few Words on R and Coding

This course will be based around the programming language R which we will use within the integrated development environment (IDE) R Studio. For many of you this will be the first time programming, **AND THAT'S OK! This course is intended for beginners, and we will actively focus on building up your R skills over the course of the semester.** Of course, there will still be challenges along the way, but you will rapidly figure out how to solve your own problems as well as to apply your current knowledge to new and exciting questions. If you are struggling I highly encourage you to take advantage of my free help sessions (see times above). Of course, Google is always a super helpful way to get insight into coding problems. Our class Slack channel will also be there so you can help each other out. You might want to watch Roger Peng's video on how to get help, which contains guidelines on what information to provide when asking a question in a public forum.

I also want to note that I highly encourage you to help each other, as data scientists are rarely working in isolation. **This does not mean you can directly share code associated with an assignment (this is a violation of UA's Code of Academic Integrity).** What it does mean is that it is helpful to talk to each other about problems you encountered, resources you found, or provide helpful tips.

Learning to code nowadays is much easier, since a simple Google search will research in a huge amount of code that can solve any number of problems. You may use online resources (e.g., StackOverflow), but we will go over the syntax needed to solve all assignments in class. If you do use any external resources, you must explicitly cite where the code was obtained in your comments (add a direct link to the resource). I'll be checking for recycled code, and any code you re-used without a proper citation will be treated as plagiarism.

1.3 A Few Words on Technology

1. YOU MUST HAVE ACCESS TO A COMPUTER YOU CAN CODE WITH IN EVERY CLASS! We will be actively coding in R on a daily basis, and not being able to follow along will severely hamper your learning. If you do not have a laptop or yours had troubles at some point during the semester, the library offers fast and free rentals of both macs and PCs: <https://new.library.arizona.edu/tech/borrow>. You can also take advantage of the multiple computer labs on campus: <https://it.arizona.edu/service/oscr-computer-labs>
2. You will have access to and will be required to retrieve all course materials from the course page on GitHub.

3. You will need to have R and R Studio installed and functioning by the second day of class. We will go over what these programs are and how to install them in the first week of class.
4. Slack participation is critical! If you are having a coding issue, first try and solve it on your own. If you're still struggling, then post it to our Slack. Essentially, if you are about to email me with a homework/class/coding question, post it to Slack first. I'm not doing this to save me time, but rather because virtually all programmers/coders solve problems by helping each other, and thus I want you to do the same! Please register for our Slack channel.

1.4 Readings

There is no required textbook for this class. A few times we will use the book “R for Data Science” by Hadley Wickham and Garret Golemund. This book covers how to create full data science pipelines in R (more than we'll be doing here) and is available free here: <https://r4ds.had.co.nz/>.

Aside from this book, there will be other required readings. I will link these readings for you on this bookdown. Some come from academic journals, and others are news articles that appear in many of the newspapers you read in print and online. For each reading, a word count and an approximate reading time will be provided. Please adjust these approximations to your own reading time, so you can plan accordingly.

It is crucial that you read all assigned readings to do well in this class. Anyone who has not done the reading will simply not be able to participate.

1.5 Assignments with Grade Breakdown

Activity	Total Percent	Unit Percent	Description
Final Project	30%	5% Project Proposal 15% Write-up 10% Oral presentation	This will be a full data science project, complete with formal write-up and presentation to the class
Midterm	20%	20%	
Sharing Code	10%	1%	
(10)			

Activity	Total Percent	Unit Percent	Description
Data Challenges (9)	28%	3.5%	Lowest will be dropped. All assignments must completed by the date and time provided in the assignment instructions
Class Participation	10%		Participation includes both in-class and message board questions, engagement. To get full credit I should see your name or hear you in class once a week.
Intro and exit surveys	2%	1%	

Late assignments within 24 hours of due date and time will get a 20% grade penalty. Assignments submitted 24 hours after the due date and time will not get any credit.

If you are unable to complete an assignment on the due date due to an illness or another personal problem, please contact me as soon as possible so we can talk about ways to help you complete that assignment.

Grade Distribution:

90-100% = A “exemplary, far beyond reqs/expectations”

80-89% = B “exceeds requirements/expectations”

70-79% = C “meets requirements/expectations”

60-69% = D “falls short of requirements/expectations”

< 60% = E “repeat of course needed”

A Note About Final Grades

I do not modify final grades. I have designed this course to be highly passable for the new learner assuming they do the modest homework assignments, come to class, and participate. I’m not a difficult grader, and I build in extensive

opportunities for ‘easy points.’ Given all this, please do not try and ask for a higher grade when end of semester rolls around.

1.6 Requirements for the Course

To succeed in this course, 2-3 hours of study time per hour of formal class time (or per unit) are required. This means that in addition to our three hours of formal class meeting time, 6-9 hours a week of study time are needed in order to meet course expectations. These hours should be spent on reading texts, working on your data challenges, researching for new information, or thinking about course content.

It’s important to mention that each lesson builds upon the previous, and thus staying on top of the material is critical to your success. As mentioned before, this class is built specifically for beginners, and plenty of students who have never coded before have done extremely well. But, the reason they did is that they came to class consistently, asked questions when they had an issue, and completed their data challenges. If you miss a class, come to office hours to make up what you missed. I will do everything possible to make sure you succeeded assuming you’re willing to put in the work!

1.7 Course Schedule

Here is the tentative course schedule. **Data challenges are always due before the start of class on the associated due date.** There will sometimes be other short readings and assignments. These will be posted on D2L directly after the class period in which they are assigned.

Week	Date	Goals	Assignment
Week 01	2020-08-25	Introductions Syllabus	
	2020-08-27	Intro do Data Science Data Science workflow	Reading: What’s data science? (20 min) YouTube video Angry Hiring Manager Panel (6.5 min) Survey 1 (10 min)

Week	Date	Goals	Assignment
Week 02	2020-09-01	What's Data? What does data analysis look like? IDE overview How and Why to Start a Project	Reading: Data Science examples (8 min), Data Intake (12 min) Install R and RStudio
	2020-09-03	Basics of R - Basics of R - basic operations - objects - data types	
Week 03	2020-09-08	Basics of R - data frames - inspecting data - slicing your data	Read A Million Lines of Bad Code (549 words, 5 min)
	2020-09-10	Submitting assignments through GitHub	
Week 04	2020-09-15	Installing R Packages Intro to Tidyverse	Read Advice to Young (and Old) Programmers: A Conversation with Hadley Wickham (2,599 words, 10 min)
	2020-09-17	Tidyverse	Data Challenge 1
Week 05	2020-09-22	Data Tidying	
	2020-09-24	Data Tidying	
Week 06	2020-09-29	Intro to Descriptive Statistics	Data Challenge 2
	2020-10-01	Descriptive Statistics	Read What is Statistics Good For? (398 words, 3 min)
Week 07	2020-10-06	Intro to Data Visualization	Data Challenge 3
	2020-10-08	Data Visualization	
Week 08	2020-10-13	MIDTERM	Deadline to meet about final project

Week	Date	Goals	Assignment
Week 09	2020-10-15	Data analysis case study 1	Data Challenge 4
	2020-10-20	Data analysis case study 1	
	2020-10-22	Data analysis case study 2	Data Challenge 5
Week 10	2020-10-27	Data analysis case study 2	
Week 11	2020-10-29	Getting Data	Data Challenge 6
	2020-11-03	Getting Data	
	2020-11-05	Data analysis case study 3	Project Proposal
Week 12	2020-11-10	Data analysis case study 3	
Week 13	2020-11-12	Markdown	Data Challenge 7
	2020-11-17	Markdown	
	2020-11-19	Full data analysis case study 4	Data Challenge 8
Week 14	2020-11-24	Full data analysis case study 4	
Week 15	2020-11-26	Happy Thanksgiving!	
	2020-12-01	Written and Oral Communication in Data Science	Data Challenge 9
	2020-12-03	Written and Oral Communication in Data Science	
Week 16	2020-12-08	Preparing for Final Presentations Wrap-up	

For more information about dates including holidays, check UArizona's Academic Calendar.

Why am I using YYYY-MM-DD date format?

PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS ***THE*** CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27


THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13

20130227 2013.02.27 27.02.13 27-02-13

27.2.13 2013.II.27. $2\frac{7}{2}$ -13 2013.158904109

MMXIII-II-XXVII MMXIII $\frac{LVII}{CCCLXV}$ 1330300800

$((3+3) \times (111+1) - 1) \times 3 / 3 - 1 / 3^3$ ~~2013~~  Hissss

10/11011/1101 02/27/20/13 $\begin{matrix} 2 & 3 & 1 & 4 \\ 0 & 1 & 2 & 3 & 7 \\ 5 & 6 & 7 & 8 \end{matrix}$

1.8 Final Project

There is a final project in place of a final exam for this class. You will find your own dataset that helps you answer a question that you're interested in. You'll bring these data into R, explore it, clean it, make features, and run an analysis that allows you to answer your question. You will be graded on the completed R script as well as your presentation of the data.

The presentation will last 3-4 minutes, and will take place on the day of the final exam (in place of the exam). University policy on final examinations can be found here: <https://www.registrar.arizona.edu/courses/final-examination-regulations-and-information>

1.9 Honors Students' Requirements

Students wishing to take this course for Honors Credit should email me to set up an appointment to discuss the terms of the contract and to sign the Honors Course Contract Request Form. The form is available at <https://www.honors.arizona.edu/sample-contracts>. Students earning credit with the University of Arizona Honors College will be held to the following enhancements:

1. Honors students will be required to create an academic poster based on their final project, and then present this poster at the iSchool's iShowcase at the end of the semester. Creating a poster will require extra work to ensure clarity of logic, having a well-defined question and approach, and the creation of quality visuals. Guidelines on how to create an engaging academic poster can be found here: <https://guides.nyu.edu/posters>. Note: The iShowcase is at the end of the semester, but before finals when the regular class will have the project due. Thus, you will have to be ahead of schedule a bit to meet your honors requirement.
2. Honors students will also be expected to informally 'journal' about the course each week. Each week, that is, students will be required to write a five-sentence paragraph reflecting on some issue or moment that has arisen in our readings or discussions (e.g., the problem with particular terms or some philosophical or practical dilemma). Ultimately, if offering a paragraph each week, honors students will have written roughly 15 reflective paragraphs for the semester. This must be emailed directly to me by Sunday 5pm each week.

1.10 Student Accommodations

It is the University's goal that learning experiences be as accessible as possible. If you anticipate or experience physical or academic barriers based on disability

or pregnancy, please let me know immediately so that we can discuss options. You are also welcome to contact Disability Resources (520-621-3268) to establish reasonable accommodations. For additional information on Disability Resources and reasonable accommodations, please visit <http://drc.arizona.edu/>.

1.11 Attendance, Due Dates, and Missing Work

1. Missed class assignments or exams cannot be made up without a well-documented, verifiable, excuse (for example, a physician's medical excuse). Indeed, due dates are firm, and late work will be accepted only with a verifiable and valid excuse.
2. The UA policy regarding absences for any sincerely held religious belief, observance or practice will be accommodated where reasonable, <http://policy.arizona.edu/human-resources/religious-accommodation-policy>.
3. Absences pre-approved by the UA Dean of Students (or Dean designee) will be honored. <https://deanofstudents.arizona.edu/absences>
4. Arriving late and leaving early is extremely disruptive to others in the class. Please avoid this kind of disruption.
5. The UA's policy concerning Class Attendance and Administrative Drops is available at: <https://catalog.arizona.edu/policy/class-attendance-participation-and-administrative-drop>

1.12 Course Conduct and Campus Policies

It's important to be familiar with all campus policies.

1. Students are encouraged to share intellectual views and discuss freely the principles and applications of course materials. However, graded work/exercises must be the product of independent effort unless otherwise instructed. Students are expected to adhere to the UA Code of Academic Integrity as described in the UA General Catalog. See: <http://deanofstudents.arizona.edu/academic-integrity/students/academic-integrity>.
2. It is the University's goal that learning experiences be as accessible as possible. If you anticipate or experience physical or academic barriers based on disability or pregnancy, please let me know immediately so that we can discuss options. You are also welcome to contact Disability Resources (520-621-3268) to establish reasonable accommodations. For additional information on Disability Resources and reasonable accommodations, please visit <http://drc.arizona.edu/>.
3. The UA Threatening Behavior by Students Policy prohibits threats of physical harm to any member of the University community, including to

oneself. See <http://policy.arizona.edu/education-and-student-affairs/threatening-behavior-students>.

4. All student records will be managed and held confidentially. <http://www.registrar.arizona.edu/personal-information/family-educational-rights-and-privacy-act-1974-ferpa?topic=ferpa>
5. The University is committed to creating and maintaining an environment free of discrimination; see <http://policy.arizona.edu/human-resources/no-discrimination-and-anti-harassment-policy>.
6. Information contained in this syllabus, other than the grade and absence policy, may be subject to change without advance notice as deemed appropriate by the instructor.

1.13 Code of Conduct

This code of conduct is based on GitHub Community Guidelines. One of the goals of this course is to get you familiar with the data science community, and how people work and learn better together. This is a community we build together, and we need everybody's help to make it better each day.

1.13.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as instructor and students pledge to making participation in our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

- **Be welcoming and open-minded.** Although this is an intro course, like in any other learning setting, we have people at different levels of experience. Other people may not have the same experience level or background as you, but that doesn't mean they don't have good ideas to contribute. I encourage you to be welcoming to everyone, from more advanced coders to those just getting started. We can all learn from each other.
- **Respect each other.** Nothing sabotages healthy conversation like rudeness. Be civil and professional, and don't post or say anything that a reasonable person would consider offensive, abusive, or hate speech. Don't harass or grief anyone. Treat each other with dignity and consideration in all interactions.

You may wish to respond to something by disagreeing with it. That's fine. But remember to criticize ideas, not people. Avoid name-calling, ad hominem

attacks, responding to a post's tone instead of its actual content, and knee-jerk reactions. Instead, provide reasoned counter-arguments that improve the conversation.

- **Communicate with empathy.** Disagreements or differences of opinion are a fact of life. Being part of a community means interacting with people from a variety of backgrounds and perspectives (and we are all better because of this variety), many of which may not be your own. If you disagree with someone, try to understand and share their feelings before you address them. This will promote a respectful and friendly atmosphere where people feel comfortable asking questions, participating in discussions, and making contributions.
- **Be clear and stay on topic.** The goal of this course is to learn about data science and how to do data science with R. Off-topic comments are a distraction (sometimes welcome, but usually not) from getting work done and being productive. Staying on topic helps produce positive and productive discussions.

Additionally, as this class will be conducted online, you might not have met each other in person. Communicating on the internet can be awkward, even when you already know people. It's hard to convey or read tone, and sarcasm is frequently misunderstood. Try to use clear language, and think about how it will be received by the other person.

1.13.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission

- Other conduct which could reasonably be considered inappropriate in a professional setting

1.13.3 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting your instructor at adrianaps@email.arizona. Your instructor will review and investigate all complaints, and will respond in a way that it deems appropriate to the circumstances. Your instructor is obligated to maintain confidentiality with regard to the reporter of an incident.

1.13.4 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at <http://contributor-covenant.org/version/1/4>

1.14 How to Ask For Help

We'll see in this course that a key skill that you should develop as a data science is the ability to find solutions to problems. Knowing how to get help is part of that skill.

1.14.1 Before You ask for help

- **Check for typos.** One of the most common causes of errors are typos, which usually throw an error such as Error in _____ : could not find function "_____" due to a function being misspelled
- **Check loaded packages.** You also get errors like Error in data %>% summary() : could not find function "%>%" when you failed to load a package.
- **Read the error message.** Don't ignore what R is telling you. Be aware that red text that appears in your console is not always indication of errors. Sometimes it's just a warning.
- **Google is your friend.** Copy and paste the exact error message on a Google search. (this step also includes **read the documentation** on the package you're trying to use).
- If you are still stuck, you can always try **rubber duck debugging**. Describe the problem aloud, explaining it line-by-line, to a rubber duck or another person (who might not have any experience with programming of

data science). This is also a good preparation step to asking other people for help (next section).

1.14.2 Ask other people for help

Like mentioned before, you should ask your peers for help before you ask your instructor. Relying on a single person to solve all of your problems is dangerous, because that person won't be available throughout your career as a data scientist.

- Check **our Slack** to see if someone else has asked a question similar to yours, and whether there's a solution posted for it.
- **Be precise and informative.** The more context you can provide about what you're trying to do and what errors you're getting, the better. Also describe the steps you took to try to solve the problem yourself.

1.14.3 List of Resources

- Getting Help with R
- Roger Peng's How To Get Help video
- Rubber Duck Debugging

Chapter 2

What's data science?

2.1 Before class #1

Required external reading for this module: What's data science? (4,660 words, approx. 20 minutes of reading time)

Watch the YouTube video Angry Hiring Manager Panel from 10:18 to 16:48 (6.5 minutes) and list the skills they mention as important to have in a data science position.

Fill out Survey 1 (10 min)

2.2 What's data science?

Data science is one of the fields with the highest demand, with prospects of increased demand for the next decade (Kross et al., 2020; Hadavand et al., 2018). Interestingly, the data scientist title was invented in 2008, and the median base salary for a data scientist surpassed \$100,000 in the United States in 2019 (Robinson and Nolis, 2020).

Based on your own experience and on your reading for this module, in your groups discuss the following question:

- What is data science?

2.3 What does a data scientist do?

Data science is an interdisciplinary field, and as such data scientists hold jobs with a broad range of skills, from statistics to communication. A quick Google search for data science jobs reveals this long list of skills. However, no single data scientist has all skills listed for different data science jobs. Instead, each data scientist specializes in different skills (Robinson and Nolis, 2020).

Make a list of skills listed on data science job announcements and in the video you just watched. Based on these, discuss the following questions in your group:

- 1) Which skills do you already have? At what level of proficiency?
- 2) Which skills are you interested in developing further?
- 3) Based on the skills you already have, and the skills you want to acquire, what type of job in data science would you be interested in?

2.4 Data Science Workflow

The basic data science workflow involve three main parts:

- 1) Data Acquisition: data file, database, or web API
- 2) Data Wrangling: import + tidy data + transform (Grolemund and Wickham, 2018)
- 3) Data Exploration: transform + visualize + model + repeat (Grolemund and Wickham, 2018)
- 4) Results Communication: visualize + write + knit (Grolemund and Wickham, 2018)

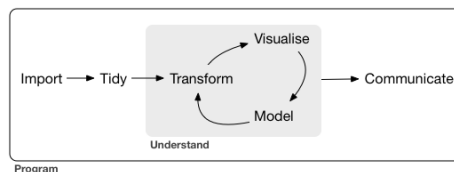


Figure 2.1: typical data science project model (Grolemund and Wickham, 2018)

In your groups, based on your own intuition and experience, and based on the Introduction to R for Data Science book (Grolemund and Wickham, 2018), summarize what each of the following steps means:

- 1) Tidy
- 2) Transform

- 3) Visualize
- 4) Model
- 5) Communicate

We will approach the entire data science workflow in this course (but not necessarily every step listed), not in this order. We start with step 2 (Data Wrangling) and 3 (Data Exploration), before we address step 1 (Data Acquisition) and step 4 (Communication)

Go back to the list of skills and job positions we discussed (based on the reading and the video):

- 1) Which steps in the data science workflow correspond to the job skills we talked about?

2.5 Before class #2

Please fill out Survey 1 (10 min).

Reading: Data Science examples (1,0333 words, 8 min)

Reading: Data Intake (1,686 words, 12 min)

2.6 What's data?

In your small group, discuss the examples provided in the excerpt from “Executive Data Science” (Caffo et al., 2016).

- 1) How is this definition of Data Science different from other definitions we discussed so far?
- 2) Why did Netflix end up not implementing the best solution from the Netflix prize challenge?
- 3) What data was used in each of the examples provided?
- 4) What is data? (come up with a definition).

Examples of what data might look like.

- Structured data (rare):

State

School Year

Average Tuition

Nevada

2004-05

3621.392

Nevada

2005-06

3687.290

Florida

2004-05

3848.201

Florida

2007-08

3879.416

Florida

2006-07

3887.656

Florida

2005-06

3924.234

Wyoming

2008-09

3928.671

Wyoming

2007-08

4071.898

Wyoming

2004-05

4086.351

Wyoming

2006-07

4122.205

- Structured, but messy data (more common):

State

2004-05

2005-06

2006-07

2007-08

2008-09

2009-10

2010-11

2011-12

2012-13

2013-14

2014-15

2015-16

Alabama

5682.838

5840.550

5753.496

6008.169

6475.092

7188.954

8071.134

8451.902

9098.069

9358.929

9496.084

9751.101

Alaska

4328.281

4632.623

4918.501

5069.822

5075.482

5454.607

5759.153

5762.421

6026.143

6012.445

6148.808

6571.340

Arizona

5138.495

5415.516

5481.419

5681.638

6058.464

7263.204

8839.605

9966.716

10133.503

10296.200

10413.844

10646.278

Arkansas

5772.302

6082.379

6231.977

6414.900

6416.503

6627.092

6900.912

7028.991

7286.580

7408.495

7606.410

7867.297

California

5285.921

5527.881

5334.826

5672.472

5897.888

7258.771

8193.739

9436.426

9360.574

9274.193

9186.824

9269.844

Colorado

4703.777

5406.967

5596.348

6227.002

6284.137

6948.473

7748.201

8315.632

8792.856

9292.954

9298.599

9748.188

Connecticut

7983.695

8249.074

8367.549

8677.702

8720.976

9371.019

9827.013

9736.431

10036.627

10453.110

10663.995

11397.337

Delaware

8352.890

8610.597

8681.846

8945.801

8995.473

9987.183

10534.181

11026.241

11362.690

11502.524

11514.660

11676.216

Florida

3848.201

3924.234

3887.656

3879.416

4150.004

4783.032

5510.659

5940.945

6494.901

6451.664

6345.000

6360.159

Georgia

4298.040

4492.167

4584.268

4790.266

4831.365

5549.913

6428.007

7709.284

7853.257

7992.390

8063.014

8446.961

user__id

screen__name

text

reply__to__screen__name

6.331283e+07

blagogirl

? The illiterate calling Iran out? 80 million bounty on Trumps head?

realTuckFrumper

6.331283e+07

blagogirl

? Iran does NOT fear Trump. They realize what OUR country is dealing with.
“The White House is inflicted with mental retardation”

JonHutson

1.125104e+18

dl_kirkwood

I'm afraid 11 soldiers had to be shipped out from the Iran hit after all with
traumatic brain injuries. Seems the Military does not notify homeland unless
a soldiers is shipped out for the injury. So, Trump did not know for a week.
<https://t.co/HdBNbKClBl>

NA

2.820552e+07

djbarro

? Are you going to carry a sign supporting the women in Iran brave enough to
remove their hijabs and go to prison?

GloriaAllred

1.506314e+08

kizu91

US...Special...Representative...Hold...Press...Briefing...Situation in...Iran...Video...first...week...January...saw...
Trump...order...assassination...elite...Quds...Force...commander...Qasem...Soleimani...Iraq

NA

1.506314e+08

kizu91

crash...land...collide...plane...aircraft...all...176...people...on board...Iran...missile...attack...US...base...Iraq...ro

NA

1.506314e+08

kizu91

Iran...MP...Urge...Gov't...Expel...UK...Envoy...Consider...Downgrading...Diplomatic...Ties...Alleged...Meddli...
Macaire...detained...days...ago...alleged...participation...unsanctioned...protest...Tehran...down...Ukraine...Bo

NA

1.506314e+08

kizu91

Government...Supporter...Gather...Tehran...13...Friday...Prayer...Video...Iran...gather...rally...commemorate

NA

1.506314e+08

kizu91

British...Treasury...Expand...Hezbollah...Asset...Freeze...UK...government...approved...measure...follow...heat...conflict...U
States...Islamic...Republic...Iran...Trump...Administration...target...assassination...high-
profile...military...general...early...January...film

NA

7.297365e+17

SwmpladySH

Hackers Are Coming for the 2020 Election — And We're Not Ready <https://t.co/q82kNu9gMd> via ?

NA

- Textual Data (always messy):

```
## [1] "CHAPTER I"
## [2] ""
## [3] ""
## [4] "Emma Woodhouse, handsome, clever, and rich, with a comfortable home"
## [5] "and happy disposition, seemed to unite some of the best blessings of"
## [6] "existence; and had lived nearly twenty-one years in the world with very"
## [7] "little to distress or vex her."
## [8] ""
## [9] "She was the youngest of the two daughters of a most affectionate,"
## [10] "indulgent father; and had, in consequence of her sister's marriage, been"
```

What data formats are out there in the world. Create a list based on your experience and the excerpt from “Modern Data Science with R” (Baumer et al., 2017).

2.7 What does data analysis look like?

The way you communicate your data analysis will depend on what question you're trying to answer and who your audience is. Here are some of my favorite data analysis reports:

- Whose (coffee) beans reign supreme? A #tidytuesday static image
- Women in Space A #tidytuesday static image
- Which city is faster? A City Cycle Race Shinny app

- The Physical Traits that Define Men and Women in Literature An interactive website

Chapter 3

Exploring our IDE (Rstudio)

3.1 Before class

Install R and RStudio.

We are using RStudio as our IDE for this course. If you are running your R code in your computer, you need to install both R and RStudio. Alternatively, you can create a free account at <http://rstudio.cloud> and run your R code in the cloud. Either way, we will be using the same IDE (i.e., RStudio).

What's an **IDE**? IDE stands for **i**ntegrated **d**evelopment **e**nvironment, and its goal is to facilitate coding by integrating a **t**ext **e**ditor, a **c**onsole and other tools into one window.

3.1.1 I've never installed R and RStudio in my computer OR I'm not sure I have R and RStudio installed in my computer

1. Download and install R from <https://cran.r-project.org> (If you are a Windows user, first determine if you are running the 32 or the 64 bit version)
2. Download and install RStudio from <https://rstudio.com/products/rstudio/download/#download>

3.1.2 I already have R and RStudio installed

1. Open RStudio

2. Check your R version by entering `sessionInfo()` on your console.
3. The latest release for R was June 22, 2020 (R version 4.0.2 Taking Off Again). If your R version is older than the most recent version, please follow step 1 in the previous section to update R.
4. Check your RStudio version, if your version is older than Version 1.3.x, please follow step 2 in the previous section to update RStudio.

How often should I update R and RStudio? Always make sure that you have the latest version of R, RStudio, and the packages you're using in your code to ensure you are not running into bugs that are caused by having older versions installed in your computer.

When asked, Jenny Bryan summarizes the importance of keeping your system up-to-date saying that “You will always eventually have a reason that you must update. So you can either do that very infrequently, suffer with old versions in the middle, and experience great pain at update. Or admit that maintaining your system is a normal ongoing activity, and do it more often.”

You can ensure your packages are also up-to-date by clicking on “Tools” on your RStudio top menu bar, and selecting “Check for Packages Updates...”

3.2 Create an R Project

In today's class, we will focus on situating ourselves around our IDE. For every lesson, we will either start a new R project or open an R project we've been working on.

Why create a RStudio project? RStudio projects make it easier to keep your projects organized, since each project has their own working directory, workspace, history, and source documents. In other words, it's much easier to open an R project and not have to worry about setting your working directory than to try to hunt down your files.

Here are the steps we are starting with today:

1. Start a new R project
2. Create a new R script
3. Save that R script as 01-class_one

Take a moment to look around your IDE. What are the main panes on the RStudio interface. What are the 4 main areas of the interface? Can you guess what each area is for?

3.3 Operations and Objects

Let's start by using R as a calculator. On your **console** type `3 + 3` and hit enter.

```
3 + 3
```

```
## [1] 6
```

What symbols do we use for all basic operations (addition, subtraction, multiplication, and division)? What happens if you type `3 +`?

Let's save our calculation into an object, by using the assignment symbol `<-`.

```
sum_result <- 3 + 3
```

Take a moment to look around your IDE once again. What has changed?

Now, let's use this new object in our calculation

```
sum_result + 3
```

```
## [1] 9
```

Take a moment to look around your IDE once again. Has anything changed?

What else can we do with an object?

```
class(sum_result)
```

```
## [1] "numeric"
```

R is primarily a functional programming language. That means that there pre-programmed functions in base R such as `class()` and that you can also write your own functions (more on that later).

Type `?class` in your console and hit enter to get more information about this function.

CHALLENGE

Create an object called `daisys_age` that holds the number 8. Multiply `daisys_age` by 4 and save the results in another object called `daisys_human_age`

Imagine I had multiple pets (unfortunately, that is not true, Daisy is my only pet). I can create a **vector** to hold multiple numbers representing the age of each of my pets.

```
my_pets_ages <- c(8, 2, 6, 3, 1)
```

Take a moment to look around your IDE once again. What has changed?

What is the class of the object `my_pets_ages`?

Now let's multiply this vector by 4.

```
my_pets_ages * 4
```

```
## [1] 32  8 24 12  4
```

Errors are pretty common when writing code in any programming language, so be ready to read error messages and debug your code. Let's insert a typing error in our previous code:

```
my_pets_ages <- c(8, 2, 6, '3', 1)
```

CHALLENGE

Try to multiply `my_pets_ages` by 4. What happens? How can we debug our code to find out what is causing the error?

Chapter 4

R Basics

4.1 Before Class

Read A Million Lines of Bad Code a blog post by David Robinson. (549 words, 5 minutes)

4.2 Dataframes

You will rarely work with individual numeric values, or even individual numeric vectors. Often, we have information organized in dataframes, which is R's version of a spreadsheet.

Let's go back to my imaginary pet's ages (make sure you have the correct vector in your global environment).

```
my_pets_ages <- c(8, 2, 6, '3', 1)
my_pets_ages <- as.numeric(my_pets_ages)
```

We will now create a vector of strings or characters that holds my imaginary pets' names (we have to be careful to keep the same order then the `my_pets_ages` vector).

```
my_pets_names <- c('Daisy', 'Violet', 'Lily', 'Iris', 'Poppy')
```

Let's now create a dataframe that contains info about my pets.

```
# create dataframe
my_pets <- data.frame(name = my_pets_names, age = my_pets_ages)
```

```
# print out dataframe
my_pets
```

```
##      name age
## 1  Daisy   8
## 2 Violet   2
## 3   Lily   6
## 4   Iris   3
## 5  Poppy   1
```

CHALLENGE

There's a number of functions you can run on dataframes. Try running the following functions on `my_pets`:

- `summary()`
- `nrow()`
- `ncol()`
- `dim()`

What other functions can/do you think/know of?

4.3 Slicing your dataframe

There are different ways you can slice or subset your dataframe.

You can use indices for rows and columns.

```
my_pets[1,]
```

```
##      name age
## 1  Daisy   8
```

```
my_pets[, 1]
```

```
## [1] "Daisy" "Violet" "Lily"   "Iris"   "Poppy"
```

```
my_pets[1, 1]
```

```
## [1] "Daisy"
```

You can use a column name or a row name instead of an index.

```
my_pets[, 'age']
```

```
## [1] 8 2 6 3 1
```

```
my_pets['1', ]
```

4.4. ADDING NEW VARIABLES (I.E., COLUMNS) TO YOUR DATAFRAME³⁷

```
##      name age
## 1 Daisy   8
my_pets['1', 'age']
```

```
## [1] 8
```

Or you can use `$` to retrieve values from a column.

```
my_pets$age
```

```
## [1] 8 2 6 3 1
```

```
my_pets$age[1]
```

```
## [1] 8
```

You can also use comparisons to filter your dataframe

```
# get index with which() function
which(my_pets$age == 8)
```

```
## [1] 1
```

```
# use which() inside dataframe indexing my_pets[row_number, column_number]
my_pets[which(my_pets$age == 8),]
```

```
##      name age
## 1 Daisy   8
```

```
my_pets[which(my_pets$age == 8), 1]
```

```
## [1] "Daisy"
```

```
my_pets[which(my_pets$age == 8), 'name']
```

```
## [1] "Daisy"
```

```
my_pets[which(my_pets$age == 8),]$name
```

```
## [1] "Daisy"
```

CHALLENGE

Print out a list of pet names that are older than 3.

4.4 Adding new variables (i.e., columns) to your dataframe

So far the `my_pets` dataframe has two columns: name and age.

Let's add a third column with the pets' ages in human years. For that, we are going to use `$` on with a variable (or column) name that does not exist in our dataframe yet. We will then assign to this variable the value in the `age` column multiplied by 4.

```
# create new column called human_years
my_pets$human_years <- my_pets$age * 4

# print dataframe
my_pets
```

```
##      name age human_years
## 1  Daisy   8           32
## 2 Violet   2            8
## 3  Lily    6           24
## 4  Iris    3           12
## 5  Poppy    1            4
```

Inspect the new `my_pets` dataframe. What dimensions does it have now? How could you get a list of just the human years values in the data frame?

4.5 Descriptive stats on dataframes

Let's explore some functions for descriptive statistics.

CHALLENGE

Try running the following functions on `my_pets$age` and `my_pets$human_years`:

- `mean()`
- `sd()`
- `median()`
- `max()`
- `min()`
- `range()`

What other functions can/do you think/know of?

4.6 Note on coding style

Coding style refers to how you name your objects and functions, how you comment your code, how you use spacing throughout your code, etc. If your coding

style is consistent, your code is easier to read and easier to debug as a result. Here's some guides, so you can develop your own coding style:

- The tidyverse style guide
 - Hadley Wickham's Advance R coding style
 - Google's R Style Guide

4.7 Organizing Data Science Projects

Chapter 5

Tidyverse

5.1 Before Class

Read Advice to Young (and Old) Programmers: A Conversation with Hadley Wickham by Philip Waggoner (2,599 words, 10 minutes)

5.2 Installing Packages

There are a lot of R packages out here (check the Comprehensive R Archive Network, i.e., CRAN, for a full list). That is one of the beautiful things about R, anyone can create an R package to share their code.

The function to install packages in R is `install.packages()`. We will be working with TidyVerse extensively in this course, which is a collection of R packages carefully designed for data science.

Open your RStudio. In your console, enter the following to install tidyverse (this may take a while).

```
install.packages("tidyverse")
```

You need to install any package only once (remember to check for new package versions and to keep your packages updated). However, with every new R session, you need to load the packages you are going to use by using the `library()` function.

```
library(tidyverse)
```

Note that when calling the `install.packages()` function you need to enter the package name between quotation marks (e.g., “tidyverse”). When you call the

`library()` function, you don't use quotation marks (e.g., `tidyverse`).

5.3 Load Data with Tidyverse

Chapter 6

Data Tidying

Chapter 7

Descriptive Statistics

7.1 Before class

Reading: What is Statistics Good For? (398 words, 3 min)

Chapter 8

Data Visualization

Chapter 9

Data Case Study 1

Chapter 10

Data Case Study 2

Chapter 11

Getting Data

Chapter 12

Data Case Study 3

Chapter 13

Markdown

Chapter 14

Data Case Study 4

Chapter 15

Analysis Reporting

Bibliography

- Baumer, B. S., Kaplan, D. T., and Horton, N. J. (2017). *Modern data science with R*. CRC Press.
- Caffo, B., Peng, R. D., and Leek, R. H. (2016). *Executive data science: A guide to training and managing the best data scientists*. Leanpub.
- Grolemund, G. and Wickham, H. (2018). *R for data science*. O'Reilly.
- Hadavand, A., Gooding, I., and Leek, J. T. (2018). Can mooc programs improve student employment prospects? *Available at SSRN 3260695*.
- Kross, S., Peng, R. D., Caffo, B. S., Gooding, I., and Leek, J. T. (2020). The democratization of data science education. *The American Statistician*, 74(1):1–7.
- Robinson, E. and Nolis, J. (2020). *Build a career in data science*. Manning.