An Educator’s Guide to the Open Case Studies

January, 2022

Table of Contents

# About this guide

The [Open Case Studies project](https://www.opencasestudies.org), developed at the [Johns Hopkins Data Science Lab](https://jhudatascience.org/), is an education platform that provides self-contained, multimodal, peer-reviewed, and open-source guides for real-world examples for active experiences of complete data analyses.

Case study guides can be used 1) in the classroom (either onsite or online) by engaging students to actively participate in a hands-on problem-solving experience, and 2) outside of the classroom by providing an archive of stand-alone examples of best practices.

To help guide educators on how to most effectively use the Open Case Studies (either in or outside of the classroom), this guide documents examples of how to use the case studies, various entry points to using the case studies (including an R package to enable modular use of the case studies), how to modify and adapt components of the case studies for the classroom, and how to contribute new case studies.

# 1 Introduction

## 1.1 Motivation

The [Open Case Studies project](https://www.opencasestudies.org), developed at the [Johns Hopkins Data Science Lab](https://jhudatascience.org/), is an education platform that provides self-contained, multimodal, peer-reviewed, and open-source guides for real-world examples for active experiences of complete data analyses. The intention of this guide is to provide instructors more information about how to make the most of our case studies.

## 1.2 Target Audience

This guide is intended for educators who are interested in using the Open Case Studies for instruction.

Elements of the Case studies can be helpful for instructors who teach:

* High school students
* Undergraduate students
* Graduate students

Elements of the case studies can assist with teaching courses about the following topics:

* Data science
* Statistics
* Public health
* Programming
* Technical writing

… and more

## 1.3 Curriculum

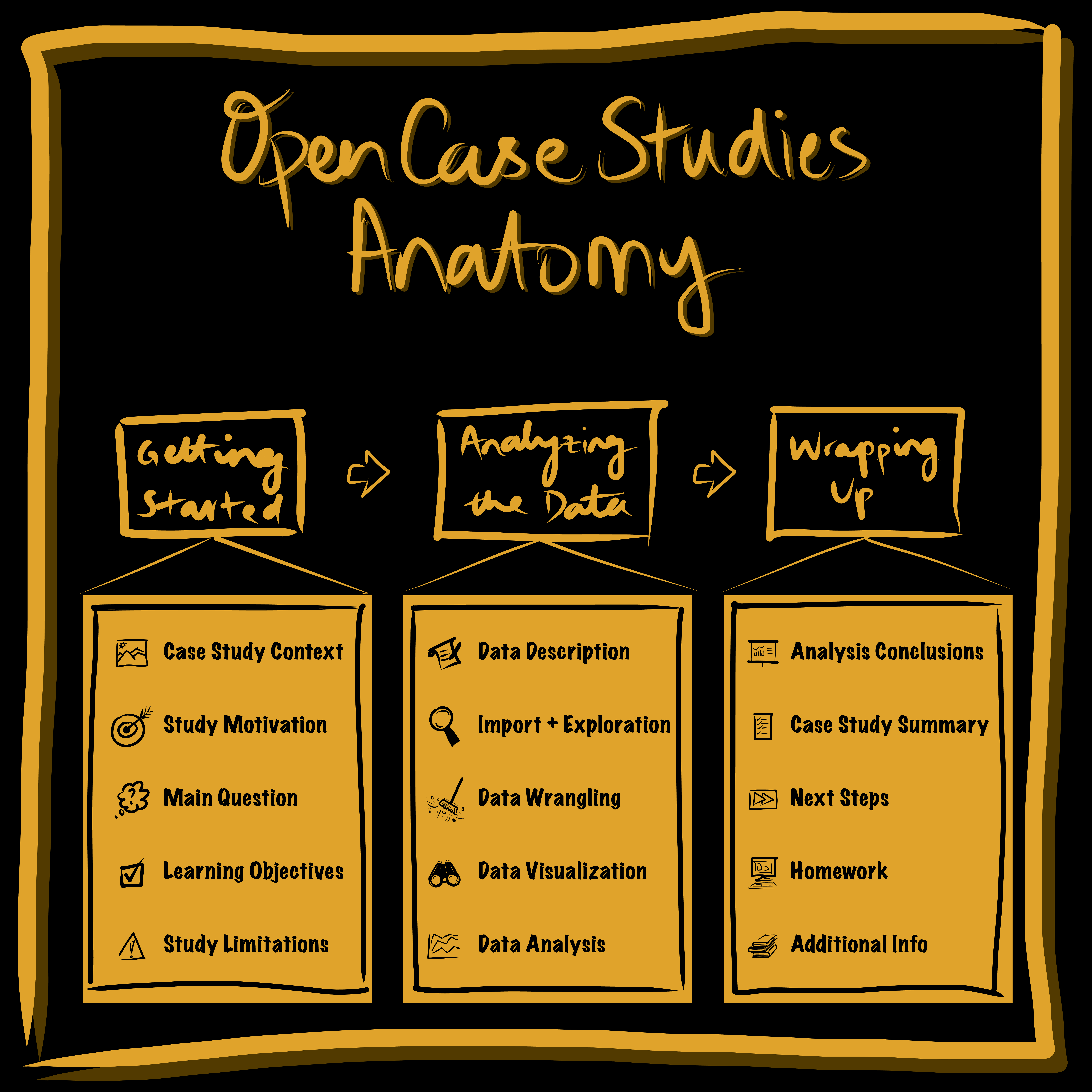
This guide documents:

* The open case studies philosophy
* The general structure of case studies
* Various entry points to using the case studies (including an R package to enable modular use of the case studies)
* Examples of how to use the case studies
* How to modify and adapt components of the case studies for the classroom
* How to contribute new case studies.

## 1.4 Open Case Studies Philosophy



## 1.5 Open Case Studies Anatomy



Case studies are designed to be self-contained examples of complete data analyses. They start with a subject area of interest and a problem definition.

Each case study is composed of three main stages. - Stage 1: Getting Started In the first stage, students learn about the subject area they will be investigating in their data analysis. This can include historical perspectives, previous literature, and current overarching questions in the field. Ideally, this will provide motivation as to why the data analysis they will be performing is interesting and useful. Students will then further refine questions of interest and define a specific question to be investigated by the data at hand. This question will define the statistical and data science learning objectives for the case study. Finally, students will explore the limitations of their investigation and discuss why their data analysis may or may not be able to fully answer the main question of interest. - Case study context - Study motivation - Main question - Learning objectives - Study limitations

* Stage 2: Analyzing the Data This stage includes the bulk of the data analysis. It begins with a detailed description of the data used in the case study, how it was generated, and where is was obtained. This sets the stage for a walk-through of the data import procedure and exploration of the available data. From there, the specific subsets of the data relevant to answering the main question of interest can be identified and the data can be cleaned and wrangled to give the data in the required format. The cleaned data can then be visualized and analyzed to answer the study question.
  + Data description
  + Import and exploration
  + Wrangling
  + Visualization
  + Analysis
* Stage 3: Wrapping-Up In the final stage of the case study, students synthesize conclusions to the main study question based on the results of their data analysis. They summarize the data exploration and analysis steps they took and discuss potential next steps they could take based on their results. Finally, additional analyses are suggested as homework and additional information relevant to the case study is provided.
  + Analysis conclusions
  + Case study summary
  + Next steps
  + Homework
  + Additional info

# 2 Open Case Study Infrastructure

## 2.1 Learning Objectives

In this chapter we will discuss the overall infrastructure of the Open Case Studies platform, which includes: \* our website \* an open case studies search tool \* our GitHub organization \* an R package called OCSdata

## 2.2 OCS Website

The [OpenCaseStudies website](https://www.opencasestudies.org/) describes the mission of the OpenCaseStudies project, the history of its inception, and current and previous members of the OCS team, among other information.

The website also includes a link to a [user experience survey](https://www.opencasestudies.org/) (takes up to 10 minutes to complete), which helps us learn more about how to improve the data science education experience. Part of this includes getting a better understanding of who is using our case studies and how so that we can better design our case studies. We would greatly appreciate you filling it out if you have the time!

Finally, the website includes the OCS case study search tool to aid instructors in finding appropriate case studies for their learning objectives. The search tool is described in more detail below.

## 2.3 OCS Case Study Search Tool

The [OCS case study search tool](https://www.opencasestudies.org/#searchtab) is designed to aid instructors in identifying appropriate case studies for their learning objectives. The tool consists of a table with searchable columns, where each row is an individual case study.

The main two columns likely to be helpful in identifying appropriate case studies are the “Packages” and “Objectives” columns. The “Packages” column details all the R packages used in the case study, and can help identify if a case study teaches a specific data import, wrangling, analysis, or visualization skill. The “Objectives” column details the learning objectives of each case study (e.g. importing data from PDF files, reshaping data, specific statistical analysis, etc.).

The table also includes links to both the static and interactive versions of the case studies and their GitHub repositories.

(avocado Include example of one case study row)

## 2.4 GitHub Organization

The GitHub repository contains all the materials needed for the case study. This includes the case study text to be distributed to students, the data used in the case study (discussed below), additional documents and references, and brief guidelines on case study use.

Data included in the GitHub repository is available in multiple formats to facilitate modularization of the case studies as described below. To use the case study data, you can download the GitHub repository directly or use the OCSdata R package described below.

(Add figure 2.4 outlining repo structure)

## 2.5 OCSdata

To simplify the process of accessing the data required for each case study, we have created the OCSdata R package. Briefly, the OCSdata package creates a new folder called “OCSdata” where it downloads the data needed for a specific case study. Users can download the data in its original raw format or in various processed formats that correspond to different stages of data wrangling and cleaning. This allows users to perform the data exploration and wrangling or the data visualization and analysis sections of the case study without having to process the data from the raw files. For some of the case studies, the OCSdata package also downloads extra source data that is not used in the case study.

(Add table 2.1 summarizing data formats available for download via OCSdata)

### 2.5.1 Getting Started with OCSdata

The OCSdata package is available on the package repository CRAN and can be installed in R as follows:

[avocado R version requirement?]

install.packages("OCSdata") #only run once to install package  
library(OCSdata) #run every new R session to load package

### 2.5.2 Downloading raw data

The raw\_data function will download the raw data files that can be imported into R.

The first argument is the name of the case study. A list of case study names can be found in the package documentation [online](https://cran.r-project.org/web/packages/OCSdata/vignettes/instructions.html#casestudy) or by typing ?raw\_data in R.

The outpath argument is a string specifying the folder where the data should be downloaded. To download the data to a folder named “OCS\_data” in the current working directory, you can supply getwd() to the output argument. If nothing is provided for the argument, you will be prompted to enter 1, 2, or 3 to download the data into the current director, to specify the download path, or to cancel, respectively.

In the following example, we download the raw data for the “Opioids in the United States” case study to the current directory.

raw\_data("ocs-bp-opioid-rural-urban", outpath = getwd())

### 2.5.3 Downloading data in other formats

The OCSdata package can be used to download the data in various processed formats that may be helpful in skipping certain case study sections and focusing on data wrangling and/or analysis and visualization. All of the functions take the same arguments described above.

#### 2.5.3.1 Simpler import

The simpler\_import\_data function will download raw data files that have been converted to file formats that are easier to import into R, typically .csv. Some case studies offer this option when the original raw files require a more complicated import step.

simpler\_import\_data("ocs-bp-opioid-rural-urban", outpath = getwd())

#### 2.5.3.2 Importing data as R objects

The imported\_data function will download raw data files in .rda format. This means the data have already been imported into R objects. This can be used to skip the data import section and start directly with data wrangling. The R objects files can be imported into R by either double clicking on the files in Rstudio or using the load() function as follows.

imported\_data("ocs-bp-opioid-rural-urban", outpath = getwd()) #download data in .rda format   
file\_path = "~/OCS\_data/data/imported/land\_area.rda" #specify download directory   
load(file\_path) #load R objects

#### 2.5.3.3 Importing wrangled data

The following functions will download the data files that have already been wrangled and are ready to be analyzed. These come in both .csv and .rda formats.

Download as csv files:

wrangled\_csv("ocs-bp-opioid-rural-urban", outpath = getwd())

Downloading as R objects:

wrangled\_rda("ocs-bp-opioid-rural-urban", outpath = getwd())

### 2.5.4 Downloading extra data

Some case studies have extra data are not used in the case study but can be used to explore the case study subject from different perspectives. These data but can This data can be downloaded using the extra\_data function.

extra\_data("ocs-bp-opioid-rural-urban", outpath = getwd())

### 2.5.5 Downloading all case study data

The zip\_ocs function will download the all of the repository files in a .zip folder and unzip them into a specified directory. This includes the case study data in all the formats detailed above (raw, simpler\_import, imported, wrangled, and extra). It also includes the case study .Rmd file, which can be modified by instructors as needed.

zip\_ocs("ocs-bp-opioid-rural-urban", outpath = getwd())

### 2.5.6 Forking the case study repository

## R version 4.0.2 (2020-06-22)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 20.04.3 LTS  
##   
## Matrix products: default  
## BLAS/LAPACK: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.8.so  
##   
## locale:  
## [1] LC\_CTYPE=en\_US.UTF-8 LC\_NUMERIC=C   
## [3] LC\_TIME=en\_US.UTF-8 LC\_COLLATE=en\_US.UTF-8   
## [5] LC\_MONETARY=en\_US.UTF-8 LC\_MESSAGES=C   
## [7] LC\_PAPER=en\_US.UTF-8 LC\_NAME=C   
## [9] LC\_ADDRESS=C LC\_TELEPHONE=C   
## [11] LC\_MEASUREMENT=en\_US.UTF-8 LC\_IDENTIFICATION=C   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## loaded via a namespace (and not attached):  
## [1] bookdown\_0.24 crayon\_1.3.4 digest\_0.6.25 R6\_2.4.1   
## [5] lifecycle\_1.0.0 magrittr\_1.5 evaluate\_0.14 pillar\_1.4.6   
## [9] leanbuild\_0.1.2 rlang\_0.4.10 stringi\_1.5.3 fs\_1.5.0   
## [13] ellipsis\_0.3.1 vctrs\_0.3.4 rmarkdown\_2.10 tools\_4.0.2   
## [17] stringr\_1.4.0 readr\_1.4.0 hms\_0.5.3 xfun\_0.26   
## [21] yaml\_2.2.1 compiler\_4.0.2 pkgconfig\_2.0.3 htmltools\_0.5.0  
## [25] knitr\_1.33 tibble\_3.0.3

# 3 Use of Open Case Studies

## 3.1 Learning Objectives

This chapter will provide guidance on how to use Open Case Studies in classroom settings for beginner, intermediate, and advanced students. Example uses will be discussed as well as suggestions for crafting homework assignments based on the case studies.

### 3.1.1 Experience Level Descriptions

In this chapter we will discuss how we envisioned case studies to be used in the classroom. We will give a coarse guide on which case studies include material appropriate for beginner, intermediate, or advanced learners. We will present some examples of assignments that can accompany the case studies and the rubrics that they can be evaluated with. Furthermore, we provide examples of extensions to case studies that can serve as a template for homework assignments or independent student exploration.

The examples presented in this chapter are merely suggestions - modifications to the case studies to fit student needs are expected and encouraged! If you come up with a different way to use the case studies, please let us know what you come up with so that other educators may be inspired by your creativity.

| Experience Level | Description |
| --- | --- |
| Beginner | Little to no previous experience with coding and/or statistical analysis. |
| Intermediate | Familiar with at least one programming language and has experience working with data and statistics. |
| Advanced | Fluent in at least one programming language and likely familiar with more. Has a depth of experience working with data science projects. |

Typically, most middle/high school and first year undergraduate students will fit in the beginner category. Upperclassmen undergraduates and some graduate students are often at the intermediate level. Most advanced level students will be at the graduate level. However, this is a generalization, and a student may be considered beginner, intermediate, or advanced at any academic level depending on their independent studies and experiences.

## 3.2 Open Case Studies in the Classroom

The case studies are structured to support both partial and full use of a case study. Educators are also free to use case study data by itself.

### 3.2.1 Teaching a Full Case Study

The case studies are written to provide a cohesive story that simulates data science in the real world. Reading through an entire case study is an excellent exercise for students to experience a standard data science workflow and learn the best practices of today from start to finish. The following list provides a few examples of how educators have used a full case study in the past:

* Converted the case study into a slide deck (add link, source and example image of slide)
* Assigned students to read the case study and write a report as homework
* Suggested reading and supplemental material
* more?

### 3.2.2 Teaching Part of a Case Study

Some educators may find that only certain sections of a case study are relevant to their course learning objectives. For these educators the case studies are structured to allow for modular use. The case study ![GitHub repositories](data:text/html; charset=utf-8;base64,) provide the appropriate data files to be used at the start of each section. These data files can be downloaded directly from GitHub or with the ![OCSdata](data:text/html; charset=utf-8;base64,) package (see chapter 2.5). The following table describes which data folder contains the corresponding data for each case study section. An example use for each data type is provided as well.

| Data Folder | Case Study Section | Example Use |
| --- | --- | --- |
| raw | Data Import | Assigning public health students to read through a case study starting from the beginning |
| imported | Data Exploration, Data Wrangling | Data science students practicing wrangling methods with in class exercises |
| wrangled | Data Visualization, Data Analysis | Statistics course practicing analysis methods with real data as a live lesson |
| simpler\_import | Data Import | Introductory data science instructor wants students to practice data import without over-complication |
| extra | Not Used in Case Study | Public health course assigns homework to practice case study methods with similar but new data |

*MB: Should we highlight the interactive case studies here as well?*

### 3.2.3 Teaching With Case Study Data Only

* Data available on GitHub
* Can use OCSdata package for streamlined data retrieval process
* Suggestions on how to use data?
* Look at case studies for exercise ideas
* more?

## 3.3 Case Study Recommendations

The following table lists a few example case studies that would be suitable for each experience level.

| Experience Level | Case Studies |
| --- | --- |
| Beginner | Exploring CO2 emissions across time, Vaping Behaviors in American Youth |
| Intermediate | Mental Health of American Youth, Opioids in United States |
| Advanced | Influence of Multicollinearity on Measured Impact of Right-to-Carry Gun Laws Part 1, Influence of Multicollinearity on Measured Impact of Right-to-Carry Gun Laws Part 2 |

## 3.4 Examples

## 3.5 Homework Assignments

======= ## Examples of assignments

In addition to helping students develop data analysis skills, we hope that the case studies can also help students develop their technical writing and communication skills. This includes but is not limited to data visualization and presentation, written reports, and oral presentations.

### 3.5.1 Data visualization

The data visualizations included in the case studies are not the only way to present the results of the analyses. Below is an example of an assignment to guide students in exploring different ways of visualizing the same results.

(avocado data visualization assignment guide, example, rubric)

### 3.5.2 Written report

Below is an example of a scientific-style paper written based on the [Opioid Use Case Study](https://www.opencasestudies.org/ocs-bp-opioid-rural-urban/#Main_Question). We also include an example rubric by which this paper can be evaluated adapted from [here](https://ocw.mit.edu/courses/biological-engineering/20-109-laboratory-fundamentals-in-biological-engineering-spring-2010/assignments/guidelines-for-writing-up-your-research/#Evaluation).

(avocado report assignment guide) [OCS Opioid Use in the US Example Report](https://raw.githubusercontent.com/opencasestudies/OCS_Guide/main/assets/OCS_Opioids_Example_Report.pdf) [OCS Opioid Use in the US Example Report Rubric](https://raw.githubusercontent.com/opencasestudies/OCS_Guide/main/assets/OCS_Opioids_Example_Report_Rubric.pdf)

### 3.5.3 Oral presentation

Below is an example of a presentation based on the [Vaping Behaviors in American Youth Case Study](https://www.opencasestudies.org/ocs-bp-vaping-case-study/). This presentation focuses on the context of the study and the methods used in the analysis. Assignments for presentations can be modified to emphasize presenting results and conclusions or to emphasize communication to different audiences (e.g. policy makers, other researchers, the public, etc. ).

(avocado insert presentation assignment guide, example, and grading rubric)

## R version 4.0.2 (2020-06-22)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 20.04.3 LTS  
##   
## Matrix products: default  
## BLAS/LAPACK: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.8.so  
##   
## locale:  
## [1] LC\_CTYPE=en\_US.UTF-8 LC\_NUMERIC=C   
## [3] LC\_TIME=en\_US.UTF-8 LC\_COLLATE=en\_US.UTF-8   
## [5] LC\_MONETARY=en\_US.UTF-8 LC\_MESSAGES=C   
## [7] LC\_PAPER=en\_US.UTF-8 LC\_NAME=C   
## [9] LC\_ADDRESS=C LC\_TELEPHONE=C   
## [11] LC\_MEASUREMENT=en\_US.UTF-8 LC\_IDENTIFICATION=C   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## loaded via a namespace (and not attached):  
## [1] bookdown\_0.24 crayon\_1.3.4 digest\_0.6.25 R6\_2.4.1   
## [5] lifecycle\_1.0.0 magrittr\_1.5 evaluate\_0.14 pillar\_1.4.6   
## [9] leanbuild\_0.1.2 rlang\_0.4.10 stringi\_1.5.3 fs\_1.5.0   
## [13] ellipsis\_0.3.1 vctrs\_0.3.4 rmarkdown\_2.10 tools\_4.0.2   
## [17] stringr\_1.4.0 readr\_1.4.0 hms\_0.5.3 xfun\_0.26   
## [21] yaml\_2.2.1 compiler\_4.0.2 pkgconfig\_2.0.3 htmltools\_0.5.0  
## [25] knitr\_1.33 tibble\_3.0.3

# 4 Modifying and creating open case studies

## 4.1 Learning Objectives

\*Every chapter also needs Learning objectives that will look like this:

This chapter will cover:

* {You can use <https://tips.uark.edu/using-blooms-taxonomy/> to define some learning objectives here}
* {Another learning objective}

## 4.2 Libraries

For this chapter, we’ll need the following packages attached:

\*Remember to add [any additional packages you need to your course’s own docker image](https://github.com/jhudsl/OTTR_Template/wiki/Using-Docker#starting-a-new-docker-image).

library(magrittr)

# 5 Topic of Section

You can write all your text in sections like this!

## 5.1 Subtopic

Here’s a subheading and some text in this subsection!

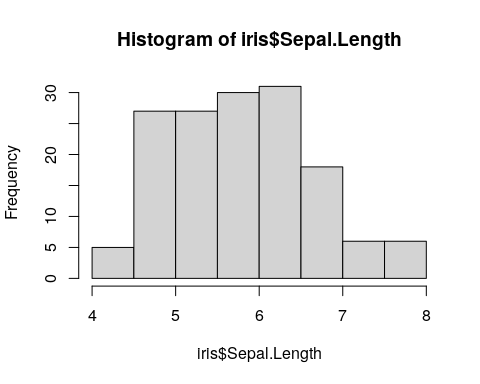
### 5.1.1 Code examples

You can demonstrate code like this:

output\_dir <- file.path("resources", "code\_output")  
if (!dir.exists(output\_dir)) {  
 dir.create(output\_dir)  
}

And make plots too:

hist\_plot <- hist(iris$Sepal.Length)



You can also save these plots to file:

png(file.path(output\_dir, "test\_plot.png"))  
hist\_plot

## $breaks  
## [1] 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0  
##   
## $counts  
## [1] 5 27 27 30 31 18 6 6  
##   
## $density  
## [1] 0.06666667 0.36000000 0.36000000 0.40000000 0.41333333 0.24000000 0.08000000  
## [8] 0.08000000  
##   
## $mids  
## [1] 4.25 4.75 5.25 5.75 6.25 6.75 7.25 7.75  
##   
## $xname  
## [1] "iris$Sepal.Length"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

dev.off()

## png   
## 2

### 5.1.2 Image example

How to include a Google slide. It’s simplest to use the leanbuild package:



But if you have the slide or some other image locally downloaded you can also use html like this:

### 5.1.3 Video examples

You can use knitr::include\_url() like this:

knitr::include\_url("https://www.youtube.com/embed/VOCYL-FNbr0")

## PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

OR this works:

### 5.1.4 Links to files

This works:

Or this:

[This works](https://www.messiah.edu/download/downloads/id/921/Microaggressions_in_the_Classroom.pdf).

Or this:

### 5.1.5 Links to websites

Examples of including a website link.

This works:

knitr::include\_url("https://yihui.org")

OR this:

![Another link](data:text/html; charset=utf-8;base64,)

Figure : Another link

OR this:

### 5.1.6 Citation examples

We can put citations at the end of a sentence like this ([Allaire et al. 2021](#ref-rmarkdown2021)). Or multiple citations Xie, Allaire, and Grolemund ([2018](#ref-Xie2018)).

but they need a ; separator ([Allaire et al. 2021](#ref-rmarkdown2021); [Xie, Allaire, and Grolemund 2018](#ref-Xie2018)).

In text, we can put citations like this Allaire et al. ([2021](#ref-rmarkdown2021)).

## 5.2 Print out session info

sessionInfo()

## R version 4.0.2 (2020-06-22)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 20.04.3 LTS  
##   
## Matrix products: default  
## BLAS/LAPACK: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.8.so  
##   
## locale:  
## [1] LC\_CTYPE=en\_US.UTF-8 LC\_NUMERIC=C   
## [3] LC\_TIME=en\_US.UTF-8 LC\_COLLATE=en\_US.UTF-8   
## [5] LC\_MONETARY=en\_US.UTF-8 LC\_MESSAGES=C   
## [7] LC\_PAPER=en\_US.UTF-8 LC\_NAME=C   
## [9] LC\_ADDRESS=C LC\_TELEPHONE=C   
## [11] LC\_MEASUREMENT=en\_US.UTF-8 LC\_IDENTIFICATION=C   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] magrittr\_1.5  
##   
## loaded via a namespace (and not attached):  
## [1] knitr\_1.33 hms\_0.5.3 R6\_2.4.1 rlang\_0.4.10   
## [5] stringr\_1.4.0 highr\_0.8 httr\_1.4.2 tools\_4.0.2   
## [9] webshot\_0.5.2 xfun\_0.26 htmltools\_0.5.0 ellipsis\_0.3.1   
## [13] yaml\_2.2.1 leanbuild\_0.1.2 digest\_0.6.25 tibble\_3.0.3   
## [17] lifecycle\_1.0.0 crayon\_1.3.4 bookdown\_0.24 readr\_1.4.0   
## [21] vctrs\_0.3.4 fs\_1.5.0 curl\_4.3 evaluate\_0.14   
## [25] rmarkdown\_2.10 stringi\_1.5.3 compiler\_4.0.2 pillar\_1.4.6   
## [29] pkgconfig\_2.0.3

# 6 Contribution guidelines

In this chapter we will discuss the guidelines for creating new case studies as well as how to publish your own new case study as part of our project.

## 6.1 Learning Objectives

\*Every chapter also needs Learning objectives that will look like this:

This chapter will cover:

* {You can use <https://tips.uark.edu/using-blooms-taxonomy/> to define some learning objectives here}
* {Another learning objective}

## 6.2 Libraries

For this chapter, we’ll need the following packages attached:

\*Remember to add [any additional packages you need to your course’s own docker image](https://github.com/jhudsl/OTTR_Template/wiki/Using-Docker#starting-a-new-docker-image).

library(magrittr)

# 7 Topic of Section

You can write all your text in sections like this!

## 7.1 Subtopic

Here’s a subheading and some text in this subsection!

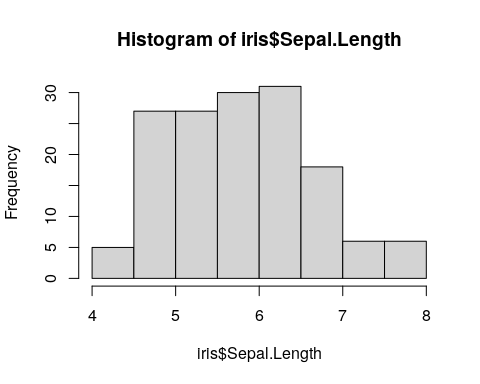
### 7.1.1 Code examples

You can demonstrate code like this:

output\_dir <- file.path("resources", "code\_output")  
if (!dir.exists(output\_dir)) {  
 dir.create(output\_dir)  
}

And make plots too:

hist\_plot <- hist(iris$Sepal.Length)



You can also save these plots to file:

png(file.path(output\_dir, "test\_plot.png"))  
hist\_plot

## $breaks  
## [1] 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0  
##   
## $counts  
## [1] 5 27 27 30 31 18 6 6  
##   
## $density  
## [1] 0.06666667 0.36000000 0.36000000 0.40000000 0.41333333 0.24000000 0.08000000  
## [8] 0.08000000  
##   
## $mids  
## [1] 4.25 4.75 5.25 5.75 6.25 6.75 7.25 7.75  
##   
## $xname  
## [1] "iris$Sepal.Length"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

dev.off()

## png   
## 2

### 7.1.2 Image example

How to include a Google slide. It’s simplest to use the leanbuild package:



But if you have the slide or some other image locally downloaded you can also use html like this:

### 7.1.3 Video examples

You can use knitr::include\_url() like this:

knitr::include\_url("https://www.youtube.com/embed/VOCYL-FNbr0")

## PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

OR this works:

### 7.1.4 Links to files

This works:

Or this:

[This works](https://www.messiah.edu/download/downloads/id/921/Microaggressions_in_the_Classroom.pdf).

Or this:

### 7.1.5 Links to websites

Examples of including a website link.

This works:

knitr::include\_url("https://yihui.org")

OR this:

![Another link](data:text/html; charset=utf-8;base64,)

Figure : Another link

OR this:

### 7.1.6 Citation examples

We can put citations at the end of a sentence like this ([Allaire et al. 2021](#ref-rmarkdown2021)). Or multiple citations Xie, Allaire, and Grolemund ([2018](#ref-Xie2018)).

but they need a ; separator ([Allaire et al. 2021](#ref-rmarkdown2021); [Xie, Allaire, and Grolemund 2018](#ref-Xie2018)).

In text, we can put citations like this Allaire et al. ([2021](#ref-rmarkdown2021)).

## 7.2 Print out session info

sessionInfo()

## R version 4.0.2 (2020-06-22)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 20.04.3 LTS  
##   
## Matrix products: default  
## BLAS/LAPACK: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.8.so  
##   
## locale:  
## [1] LC\_CTYPE=en\_US.UTF-8 LC\_NUMERIC=C   
## [3] LC\_TIME=en\_US.UTF-8 LC\_COLLATE=en\_US.UTF-8   
## [5] LC\_MONETARY=en\_US.UTF-8 LC\_MESSAGES=C   
## [7] LC\_PAPER=en\_US.UTF-8 LC\_NAME=C   
## [9] LC\_ADDRESS=C LC\_TELEPHONE=C   
## [11] LC\_MEASUREMENT=en\_US.UTF-8 LC\_IDENTIFICATION=C   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] magrittr\_1.5  
##   
## loaded via a namespace (and not attached):  
## [1] knitr\_1.33 hms\_0.5.3 R6\_2.4.1 rlang\_0.4.10   
## [5] stringr\_1.4.0 highr\_0.8 httr\_1.4.2 tools\_4.0.2   
## [9] webshot\_0.5.2 xfun\_0.26 htmltools\_0.5.0 ellipsis\_0.3.1   
## [13] yaml\_2.2.1 leanbuild\_0.1.2 digest\_0.6.25 tibble\_3.0.3   
## [17] lifecycle\_1.0.0 crayon\_1.3.4 bookdown\_0.24 readr\_1.4.0   
## [21] vctrs\_0.3.4 fs\_1.5.0 curl\_4.3 evaluate\_0.14   
## [25] rmarkdown\_2.10 stringi\_1.5.3 compiler\_4.0.2 pillar\_1.4.6   
## [29] pkgconfig\_2.0.3

# About the Authors

These credits are based on our [course contributors table guidelines](https://github.com/jhudsl/OTTR_Template/wiki/How-to-give-credits).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [FirstName LastName](link%20to%20personal%20website) |
| Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | Delivered the course in some way - video or audio |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([Leanbuild](https://github.com/jhudsl/leanbuild)) | [John Muschelli](https://johnmuschelli.com/), [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| **Art and Design** |  |
| Illustrator(s) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.3 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2022-01-19   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.3)   
## backports 1.1.10 2020-09-15 [1] RSPM (R 4.0.2)   
## bookdown 0.24 2022-01-19 [1] Github (rstudio/bookdown@88bc4ea)   
## callr 3.4.4 2020-09-07 [1] RSPM (R 4.0.2)   
## cli 2.0.2 2020-02-28 [1] RSPM (R 4.0.0)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.14 2019-05-28 [1] RSPM (R 4.0.3)   
## fansi 0.4.1 2020-01-08 [1] RSPM (R 4.0.0)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.6.0 2021-12-17 [1] CRAN (R 4.0.2)   
## htmltools 0.5.0 2020-06-16 [1] RSPM (R 4.0.1)   
## knitr 1.33 2022-01-19 [1] Github (yihui/knitr@a1052d1)   
## lifecycle 1.0.0 2021-02-15 [1] CRAN (R 4.0.2)   
## magrittr 1.5 2014-11-22 [1] RSPM (R 4.0.0)   
## memoise 1.1.0 2017-04-21 [1] RSPM (R 4.0.0)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.3.4 2020-08-11 [1] RSPM (R 4.0.2)   
## purrr 0.3.4 2020-04-17 [1] RSPM (R 4.0.3)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 0.4.10 2022-01-19 [1] Github (r-lib/rlang@f0c9be5)   
## rmarkdown 2.10 2022-01-19 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 1.3-2 2018-01-03 [1] RSPM (R 4.0.0)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2022-01-19 [1] Github (R-lib/testthat@e99155a)   
## usethis 2.1.5.9000 2022-01-19 [1] Github (r-lib/usethis@3c4ab66)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2022-01-19 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2021. *Rmarkdown: Dynamic Documents for r*. <https://github.com/rstudio/rmarkdown>.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown>.