# Welcome to Data Science!

Today, we’ll be working on getting you set up with the tools you will need for this class. Once you are set up, we’ll do what we’re here to do: analyze data!

Here’s what we need to get done today:

1. Introductions
2. Installing R
3. Installing Rstudio
4. Learning the basic verbs of data analysis

## Introductions

We need two basic sets of tools for this class. We will need R to analyze data. We will need RStudio to help us interface with R and to produce documentation of our results.

## Installing R

R is going to be the only programming language we will use. R is an extensible statistical programming environment that can handle all of the main tasks that we’ll need to cover this semester: getting data, analyzing data and communicating data analysis.

If you haven’t already, you need to download R here: <https://cran.r-project.org/>.

## Installing Rstudio

When we work with R, we communicate via the command line. To help automate this process, we can write scripts, which contain all of the commands to be executed. These scripts generate various kinds of output, like numbers on the screen, graphics or reports in common formats (pdf, word). Most programming languages have several **I** ntegrated **D** evelopment **E** nvironments (IDEs) that encompass all of these elements (scripts, command line interface, output). The primary IDE for R is Rstudio.

If you haven’t already, you need to download Rstudio here: <https://www.rstudio.com/products/rstudio/download2/>. You need the free Rstudio desktop version.

## .Rmd files

Open the 01-Intro.Rmd file. In Rstudio, go to File–>Open, then find the 01-Intro.Rmd file in the directory.

.Rmd files will be the only file format we work in this class. .Rmd files contain three basic elements:

1. Script that can be interpreted by R.
2. Output generated by R, including tables and figures.
3. Text that can be read by humans.

From a .Rmd file you can generate html documents, pdf documents, word documents, slides . . . lots of stuff. All class notes will be in .Rmd. All assignments will be turned in as .Rmd files, and your final project? You guessed it, .Rmd.

In the 01-Intro.Rmd file you’ll notice that there are three open single quotes in a row, like so: ``` This indicates the start of a “code chunk” in our file. The first code chunk that we load will include a set of programs that we will need all semester long.

## R Packages

When we say that R is extensible, we mean that people in the community can write programs that everyone else can use. These are called “packages.” In these first few lines of code, I load a set of packages using the library command in R. The set of packages, called tidyverse were written by Hadley Wickham and play a key role in his book. To install this set of packages, simply type in install.packages("tidyverse") at the R command prompt.

To run the code below in R, you can:

* Press the “play” button next to the code chunk
* In OS X, place the cursor in the code chunk and hit cmd+shift+enter
* In Windows, place the cursor in the code chunk and hit ctrl+shefit+enter

# This section prints the number 4

2+2

## [1] 4

## Clear environment  
rm(list=ls())  
## Get necessary libraries-- won't work the first time, because you need to install them!  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages ----------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.1  
## v tidyr 0.8.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'tibble' was built under R version 3.5.3

## Warning: package 'tidyr' was built under R version 3.5.3

## Warning: package 'purrr' was built under R version 3.5.3

## Warning: package 'dplyr' was built under R version 3.5.3

## Warning: package 'stringr' was built under R version 3.5.3

## Warning: package 'forcats' was built under R version 3.5.3

## -- Conflicts -------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

Now we’re ready to load in data. The data frame will be our basic way of interacting with everything in this class. The sc data frame contains information from the college scorecard on 127 different colleges and univeristies.

However, we first need to make sure that R is looking in the right place. When you opened up your project, Rstudio automagically took you to the directory for that project. But because we keep lessons in a separate directory, we need to

## Load in the data  
load("college.Rdata")

Here are the variables in the college.Rdata dataset:

*Variable Name* :*Definition* unitid: Unit ID

instnm: Institution Name

stabbr: State Abbreviation

year: Year

control: control of institution, 1=public, 2= private non-profit, 3=private for-profit

preddeg: predominant degree, 1= certificate, 2= associates, 3= bachelor’s, 4=graduate

adm\_rate: Proportion of Applicants Admitted

sat\_avg: Midpoint of entrance exam scores, on SAT scale, math and verbal only

costt\_4a: Average cost of attendance (tuition and room and board less all grant aid)

debt\_mdn: Median debt of graduates

md\_earn\_ne\_pg: Earnings of graduates who are not enrolled in higher education, six years after graduation

*Looking at datasets*

We can look at the first few rows and columns of sc by typing in the data name.

We can look at the whole dataset using View.

## What does this data look like? Look at the first few rows, first few variables  
sc

## # A tibble: 125 x 12  
## unitid instnm stabbr year control preddeg adm\_rate sat\_avg costt4\_a  
## <int> <chr> <chr> <dbl> <int> <int> <dbl> <dbl> <int>  
## 1 446048 Ave M~ FL 2009 2 3 0.374 1104 29200  
## 2 443410 DigiP~ WA 2009 3 3 0.326 1194 23969  
## 3 111081 Calif~ CA 2009 2 3 0.283 NA 48784  
## 4 112260 Clare~ CA 2009 2 3 0.163 1389 50990  
## 5 113537 Dell'~ CA 2009 2 1 0 NA NA  
## 6 404338 Schil~ FL 2009 3 2 0.158 NA 35408  
## 7 117928 Argos~ CA 2009 3 3 0.323 NA 35858  
## 8 120537 Hope ~ CA 2009 2 3 0.386 975 33366  
## 9 119544 The N~ CA 2009 3 3 0.261 NA 19135  
## 10 107071 Hende~ AR 2009 1 3 0.313 1048 14629  
## # ... with 115 more rows, and 3 more variables: debt\_mdn <dbl>,  
## # md\_earn\_wne\_p6 <int>, ugds <int>

## This is commented as you might not always want to run it. Delete the "#" sign below in order to run the command.   
  
View(sc)

*Filter, Select, Arrange*

In exploring data, many times we want to look at smaller parts of the dataset. There are three commands we’ll use today that help with this.

-filter selects only those cases or rows that meet some logical criteria.

-select selects only those variables or columns that meet some criteria

-arrange arranges the rows of a dataset in the way we want.

For more on these, please see this [vignette](https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html).

Let’s grab just the data for Vanderbilt, then look only at the average test scores and admit rate.

## Where are we?  
sc%>%filter(instnm=="Vanderbilt University")

## # A tibble: 1 x 12  
## unitid instnm stabbr year control preddeg adm\_rate sat\_avg costt4\_a  
## <int> <chr> <chr> <dbl> <int> <int> <dbl> <dbl> <int>  
## 1 221999 Vande~ TN 2009 2 3 0.202 1430 52303  
## # ... with 3 more variables: debt\_mdn <dbl>, md\_earn\_wne\_p6 <int>,  
## # ugds <int>

sc%>%filter(instnm=="Vanderbilt University")%>%select(instnm,adm\_rate,sat\_avg )

## # A tibble: 1 x 3  
## instnm adm\_rate sat\_avg  
## <chr> <dbl> <dbl>  
## 1 Vanderbilt University 0.202 1430

This is actually pretty powerful- we can quickly learn a lot just by looking at individuals. We can also get groups of institutions by specifying the characteristics of institutions we’d like to look at. For instance, let’s look at institutions with admission rates of less than 10 percent. We’ll then select three characteristics: name, admit rate and average SAT scores. Then we’ll order the result by SAT scores.

## Just colleges with low admit rates: order by sat scores (- sat\_avg gives descending)  
sc%>%filter(adm\_rate<.1)%>%select(instnm,adm\_rate,sat\_avg)%>%  
 arrange(-sat\_avg)

## # A tibble: 6 x 3  
## instnm adm\_rate sat\_avg  
## <chr> <dbl> <dbl>  
## 1 Yale University 0.0856 1475  
## 2 Harvard University 0.0719 1468  
## 3 Stanford University 0.0797 1436  
## 4 Cooper Union for the Advancement of Science and Art 0.0735 1336  
## 5 Dell'Arte International School of Physical Theatre 0 NA  
## 6 The Juilliard School 0.0711 NA

*Summarizing Data*

We can summarize data by giving commands to the summarize operator. For instance, what if we wanted to know what average debt levels looked like across these colleges?

## What's the average median debt?  
sc%>%summarize(mean\_debt=mean(debt\_mdn,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_debt  
## <dbl>  
## 1 11277.

*Combining Commands* We can also combine commands, so that summaries are done on only a part of the dataset. Below, I summarize median debt for selective schools, and not very selective schools.

## What's the average median debt for very selective schools?  
sc%>%filter(adm\_rate<.1)%>%summarize(mean\_debt=mean(debt\_mdn,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_debt  
## <dbl>  
## 1 9336.

## And for not very selective schools?  
sc%>%filter(adm\_rate>.3)%>%summarize(mean\_debt=mean(debt\_mdn,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_debt  
## <dbl>  
## 1 11684.

## What's the average earnings for very selective schools?  
sc%>%filter(adm\_rate<.1)%>%summarize(mean\_earnings=mean(md\_earn\_wne\_p6,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_earnings  
## <dbl>  
## 1 53500

## And for not very selective schools?  
sc%>%filter(adm\_rate>.3)%>%summarize(mean\_earnings=mean(md\_earn\_wne\_p6,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_earnings  
## <dbl>  
## 1 34747.

*Quick Exercise: What is the admission rate at UNC Chapel Hill?*