Introduction to Math 377

Lt Col Ken Horton Professor Bradley Warner

06 May, 2020

Introduction to the Course

Welcome to Math 377 - Advanced Probability and Statistics. Notes for this course are provided in the form of pdf and html *knit*, generated in `R`, from Rmarkdown files. Don't worry this will make more sense later in the course. This document is only to supplement the course letter, so read the course letter for details on course structure.

The notes for the course are based on ideas from the following books:

1. Introductory Statistics with Randomization and Simulation, First Edition by Diez, Barr, and Cetinkaya-Rundel. This is an open source book and it is under the creative commons license, see http://www.openintro.org/perm/stat2nd_v1.txt. The book is available at https://www.openintro.org/book/isrs/. In fact block 1, relies heavily on this book.
2. Introduction to Probability and Statistics Using R by G. Jay Kerns, http://ipsur.r-forge.r-project.org/book/download/IPSUR.pdf

You can use these books as references for the course and they are open source and free.

This course focuses on balancing mathematics with computation. In many cases we will use both approaches. Mathematical solutions offer insights and prepare you for further graduate studies. However, in practice the ability to use computational solutions is more important. Computational solution often have less assumptions and adjust to a wider variety of problems.

Software

We have selected `R` for our course software. It is open source, free, it is used extensively in industry and academia, and it is a statistical package.

`R` is really a declarative language, or command line, meant to be executed step by step although you can do more sophisticated programming. For this course we will start simple and keep the programming overhead low but as the semester progresses we will add more sophisticated methods.

In using `R` ask yourself the following questions:

1. What do you want `R` to do?

This will generally determine which R function to use.

2. What must `R` know to do that?

This will determine the inputs to the function.

Early we will use the `mosaic` package which attempts to give `R` a common formula framework. The template for supplying information is as follows:

```
goal( y ~ x, data = MyData, ... ) # pseudo-code for the formula template
```

Our first lesson notes have more information on this.

As is true for most computer languages, `R` has to be used on its terms. `R` does not learn the personality and style of its users. Getting along with `R` is much easier if you keep in mind a few key features of the `R` language.

1. `R` is case-sensitive

If you mis-capitalize something in `R` it won't do what you want. Unfortunately, there is not a consistent convention about how capitalization should be used, so you just have to pay attention when encountering new functions and data sets.

2. Functions in `R` use the following syntax:

```
functionname( argument1, argument2, ... )
```

- The arguments are always surrounded by *(round) parentheses* and *separated by commas*. Some functions (like `data()`) have no required arguments, but you still need the parentheses.
- If you type a function name without the parentheses, you will see the *code* for that function (this generally isn't what you want unless you are curious about how something is implemented).

3. TAB completion and arrows can improve typing speed and accuracy.

If you begin a command and hit the TAB key, R and RStudio will show you a list of possible ways to complete the command. If you hit TAB after the opening parenthesis of a function, RStudio will display the list of arguments it expects. The up and down arrows can be used to retrieve past commands when working in the console.

4. If you see a + prompt, it means R is waiting for more input.

Often this means that you have forgotten a closing parenthesis or made some other syntax error. If you have messed up and just want to get back to the normal prompt, press the escape key and start the command fresh.

Probability and Statistics

This course is divided into four general blocks: data/descriptive summaries, probability, inference and regression. The first block, probability, is the study of stochastic (random) processes and their properties. Specifically, we will explore random experiments. As its name suggests, a random experiment is an experiment whose outcome is not predictable with exact certainty.

Even though an outcome is determined by chance, this does not mean that we know nothing about the random experiment. My favorite simple example is that of a coin flip. If I flip a coin, the possible outcomes are heads and tails. We don't know for sure what outcome will occur, but this doesn't mean we don't know anything about the experiment. If we assume the coin is fair, we know that each outcome is equally likely.

Also, we know that if we flip the coin 100 times (independently), we are likely to see around 50 heads, and very unlikely to see 10 heads or fewer.

It is important to distinguish probability from inference and modeling. In probability, we consider a known random experiment and answer questions about what we expect to see from this random experiment. In statistics (inference and modeling), we consider data (the results of a mysterious random experiment) and infer about the underlying process. For example, suppose we have a coin and we are unsure whether this coin is fair or unfair. We flipped it 20 times and it landed on heads 14 times. Inferential statistics will help us answer questions about the underlying process (could this coin be unfair?).

This first block (16 lessons or so) is devoted to the study of random experiments. First, we will explore simple experiments, counting rule problems, and conditional probability. Next, we will introduce the concept of a random variable and the properties of random variables. Following this, we will cover common distributions of discrete and continuous random variables. We will end the block on multivariate probability (joint distributions and covariance).