



# CMSC 320

## INTRODUCTION TO DATA SCIENCE



AKILESH PRAVEEN  
PROF. JOHN DICKERSON • FALL 2020 • UNIVERSITY OF MARYLAND  
<https://cmsc320.github.io>

---

Last Revision: April 28, 2020

### Table of Contents

<b>1</b>	<b>Notes &amp; Preface</b>	<b>2</b>
<b>2</b>	<b>Lecture 1</b>	<b>2</b>
	What is Data Science? . . . . .	2
	Topics . . . . .	2
	Tools . . . . .	2
<b>3</b>	<b>Test Section</b>	<b>3</b>
	Lower Level . . . . .	3
<b>4</b>	<b>Closing Thoughts</b>	<b>3</b>

---

# 1 Notes & Preface

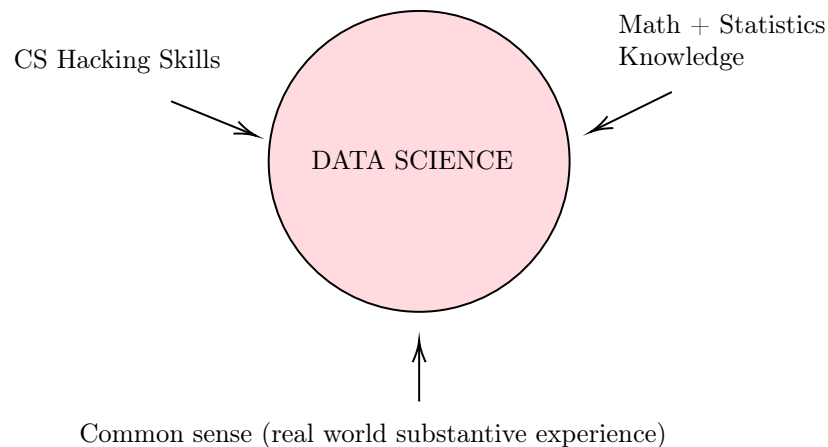
Course notes for CMSC320, under Prof. John Dickerson. Notes collected from previous and current lectures.

## 2 Lecture 1

### What is Data Science?

Data Science is the application of computation and statistical techniques to address or gain insight. It's the intersection of statistics and Computer Science. Based on what I've learned thus far, learning to do data science is like learning how to use a TI-84 in statistics class. You're simply learning how to leverage programming tools in order to perform advanced, complex, and meaningful data-related operations.

It's the use of statistics and computer science in order to find real-world insights.



### Topics

Here are the general topics that this class will cover.

- Processing data
- Visualizing data
- Understanding data
- Communicating data
- Extracting value from data

### Tools

Here are some tools commonly employed by data scientists. We'll try to cover how to use most of them here.

- Python
- Scikit-Learn
- Docker
- PANDAS
- Spark
- TensorFlow

## Conda

Conda is a package and environment manager for python that we can use with the command line. It can create multiple environments for us and install separate packages in each of them. This will be highly useful to us, as we sometimes want to consolidate the tools we use into separate environments.

## 3 Test Section

### Lower Level

Here's a cool code example.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double value;
6
7     printf("Enter a number: ");
8     scanf("%lf", &value);    /* Notice the use of %lf */
9
10    printf("sqrt %f: \n", sqrt(value));
11    printf("power of 2: %f\n", pow(value, 2));
12    printf("sin: %f\n", sin(value));
13
14    return 0;
15 }
```

Remember that the `-lm` flag essentially enables us to use the math library. In other words, if you want to compile the above file and have it work properly, (let's assume it's called `math_example.c`) then you'll want to compile it using the following command.

```
gcc -lm math_example.c
```

## 4 Closing Thoughts

End of notes.