

Cornell Data Science

# Data Science Training Program

# About Me

Name: Dae Won Kim

Position: President (Supreme Leader) of CDS

Major: Operations Research

Fun Facts:

- 1) I was a freshman in 2010
- 2) I was in the Korean army but used VBA

[dk444@cornell.edu](mailto:dk444@cornell.edu)



# Teaching Associates



Amit Mizrahi

Comp Sci '19

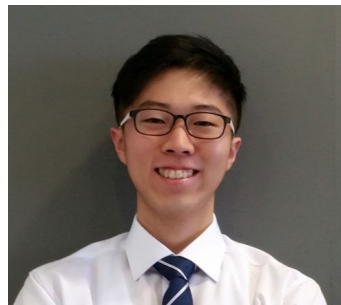
[am2269@cornell.edu](mailto:am2269@cornell.edu)



Chase Thomas

Info Sci '19

[cft32@cornell.edu](mailto:cft32@cornell.edu)



Jared Lim

Comp Sci '20

[jl3248@cornell.edu](mailto:jl3248@cornell.edu)



Kenta Takatsu

Comp Sci '19

[kt426@cornell.edu](mailto:kt426@cornell.edu)



# Goals

Comfort  
In Using R

Data  
Manipulation

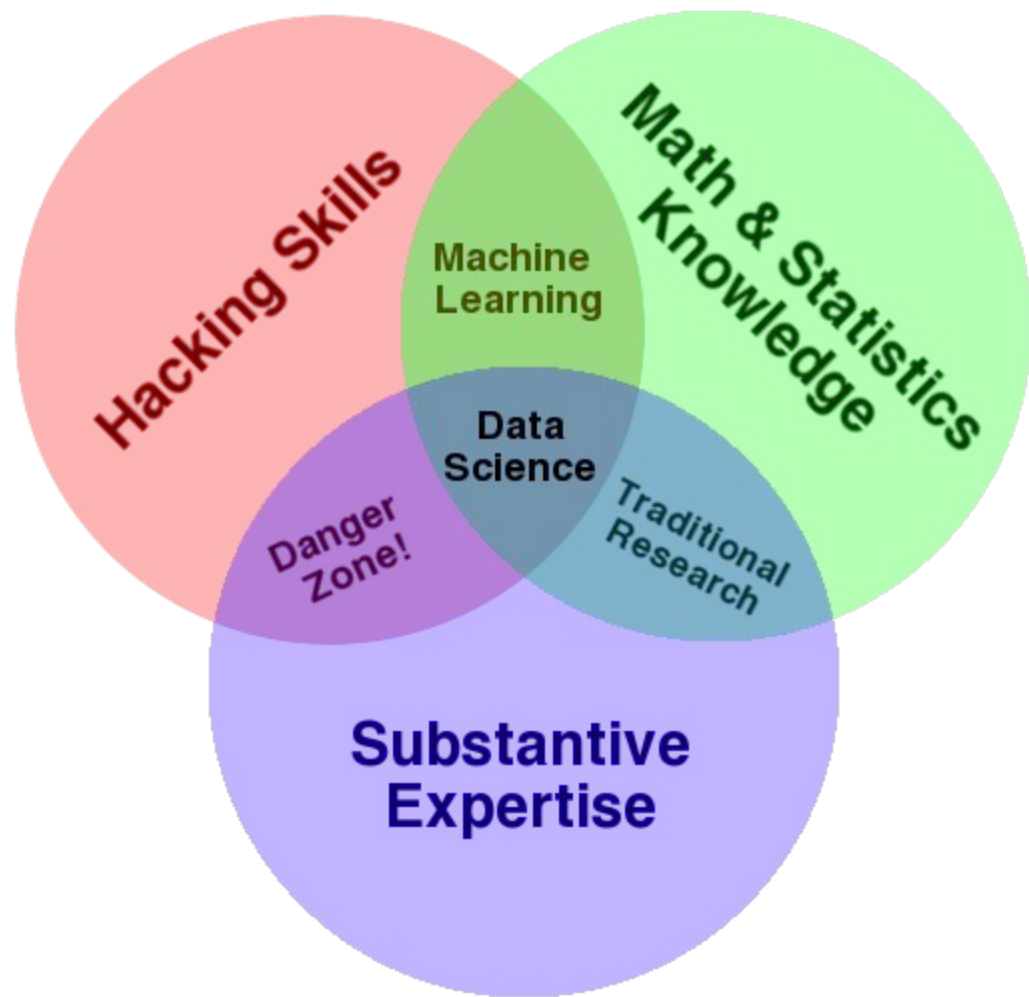
Data  
Visualization

ML  
Implementation

Model  
Optimization

Ensemble  
Implementation





Data can be...

**LARGE**

*fast*

*unStRUcTUReD*

**Volume**

**Velocity**

**Variety**



# Language Wars



# R: The Good

R has powerful **visual** tools.

**Most used** data science language.

**Concise** and powerful.

**Functional**-programming oriented.





# R: The Bad

**1-indexed** language.

Hard to write fast code.

Many ways of doing the same thing.

The learning curve is a cliff.



# R: The Ugly

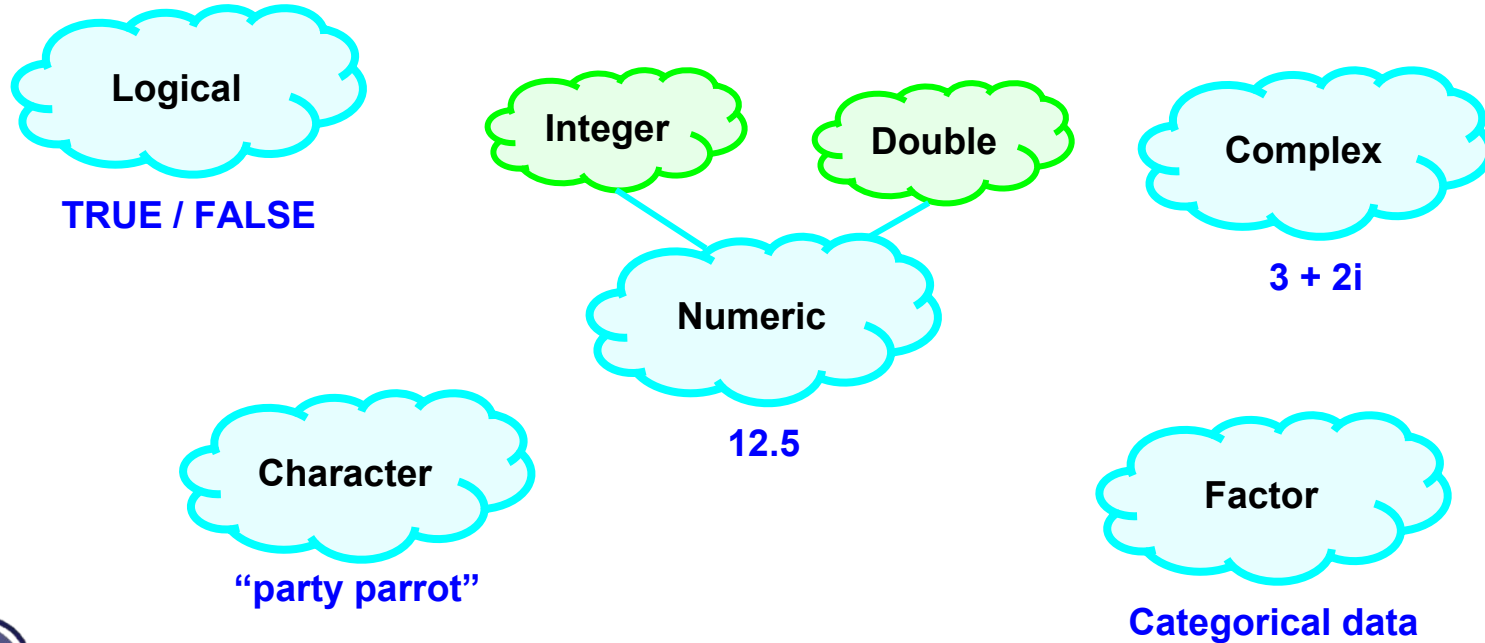
R objects are mostly **immutable**.

R relies primarily on memory.

Difficult to escape  
“**spreadsheet mentality**”.



# R Data Types



# Question:

What is the difference  
between categorical and  
continuous data?



# New Data Type: Factor

Used for handling **categorical variables**.

Factors take on only a limited number of values. Think `enum`.

Stored as a numeric, displayed as a character.

```
> gender <- c("male" "male" "male" "male" "female" "female")  
> gender <- as.factor(gender)
```



Internally, 1→female, 2→male (stores `gender` as two 1s, four 2s)

Alphabetically determined: 'f' before 'm'.

# Vector

```
> a <- c(1,2,5.3,6,-2,4) # numeric vector  
> b <- c("one","two","three") # character vector  
> c <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) # logical vector
```



# R Data Structures

*“Everything is a vector.”*

Types:

- **Matrix** - A vector with “row markers”, allows only one element type
- **List** - variable type, variable length
- **Data Frame** - variable type, same length



# Matrix



```
> matrix (data = c(1:10), nrow = 2)
```

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]
[1 ,]	1	2	3	4	5
[2 ,]	6	7	8	9	10





# Lists

```
> a <- list(1, "two", 5.3, FALSE, -2, 4)
```



# Data Frame

```
> iris
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa

# Packages

**Installation:** Use the `install.packages` function.

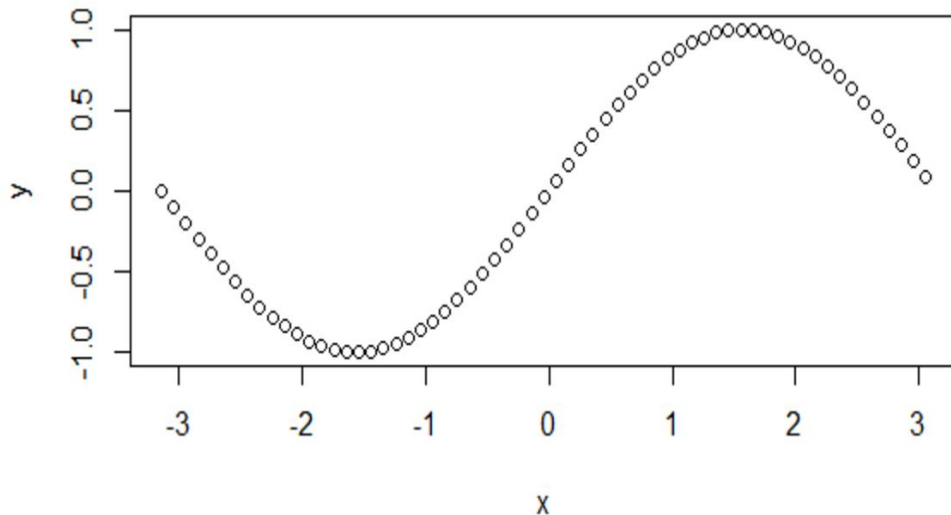
**Usage:** can use `library` or `require` (they are different!)



# Basic plotting functions

**plot** is the most used plotting function. Highly generic.

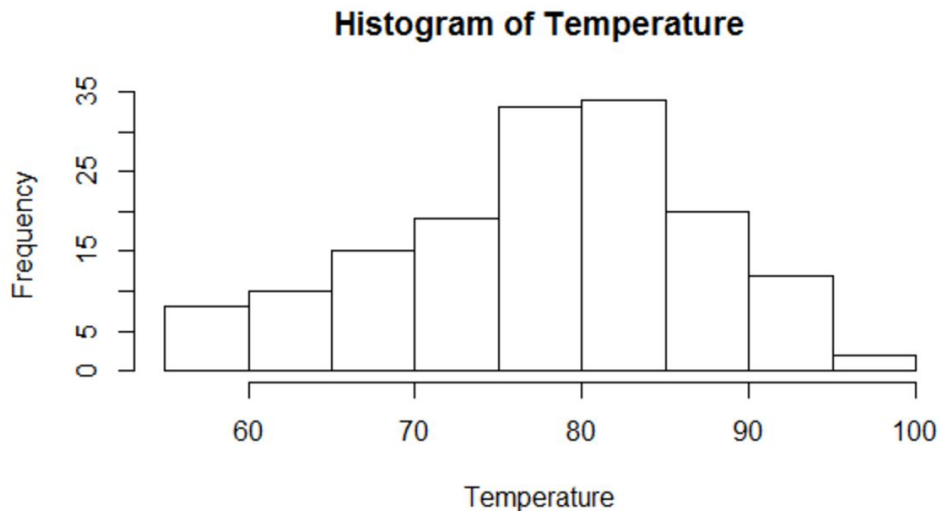
```
> x <- seq(-pi, pi, 0.1)  
> plot(x, sin(x))
```



# Basic plotting functions

**hist** makes a histogram of the vector you pass in.

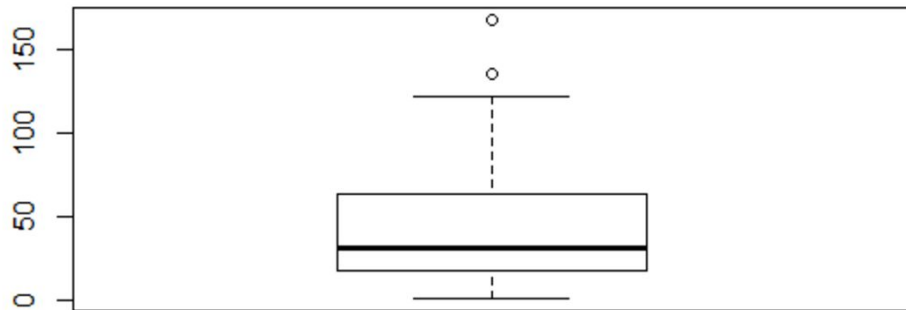
```
> temperature <- airquality$Temp  
> hist(temperature)
```



# Basic plotting functions

**boxplot** makes a box plot and can take list of numeric vectors.

```
> boxplot(airquality$Ozone)
```

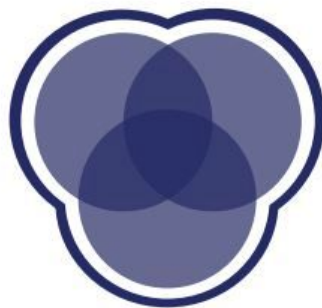


# Coming Up

**Your assignment:** Assignment 1

**Next week:** Becoming data manipulation masters

See you then!



## Helpful Links

Sign up for CMS here! [bit.ly/cornellcdscms](https://bit.ly/cornellcdscms)

^you must do this before submitting assignments^

Course website: [datascienceis.life](https://datascienceis.life)