## About me

* 11 years at Cartegraph
* java script (knockout) application with a c# backend
* I think data science is important and it will become more important
* no real qualifications for this presentation other than taking the coursera data science specialization

## Why this is important

## Why Cont.

* Their theory was sound
* the source data was good
* Humans are poor fortune tellers
* implement all the good process we as software engineers have discovered along the way:
* Pair programming, code reviews, unit tests, test driven development, open source software -> open source equations
* Off my soap box and on to the good stuff

## What is R

* Good for newbie programmers, but can be harder for others
* Apple OS does have some functional differences than when used with Windows
* There is even an IDE for Android

## A little History

* Mathematics of New Zealand
* Mathematics not computer scientists
* Not written as a programming language first, but an interactive environment first.
* this history give us context as we try to figure out how to use the system

## The Tools

* Show R and Rstudio

## Variable assignment

* highlight influenced by mathematics

## Two more ways to assign variables

## Auto print

## Vectors

## Vectors cont.

* complex = imaginary numbers
* list objects = lists are collections of vectors where objects don't have to be the same type
* Factors are not first class citizens in R
* A "factor" is a vector whose elements can take on one of a specific set of values. For example, "gender" will usually take on only the values "M" or "F," whereas "Name" will generally have lots of possibilities. The set of values that the elements of a factor can take are called its levels.

## Vectors

## Vector Math

* since in R we are generally always working with a set of data, it becomes truly awesome that vectors are the simplest object
* note on notation:
  + 1:20 give me the range between 1 and 20 and it gives me random 5 objects from that sample
  + to get random numbers use: rnorm(n, mean = 0, sd = 1)
  + rnorm = set of random numbers from the normal distribution by default with a mean of 0 and standard deviation of 1

## Vector Math

* The process of centering then scaling the data is called "normalizing" the data
* normalized data has a mean of 0 and a standard deviation of 1
* Normalize data to bring variables in portion to one another
  + Example comparing GPA that are one a 5 point scale and ones that are on a 4 point scale

## Data frames - The most useful object

* Mtcars is the name of the dataset

## Data frames

## Data frames

* one way of setting the optional parameter, could also have done tail(mtcars, n=4)

## Data frames

* the meta data for the columns
* 11 columns and 32 rows (observations = row, variables = columns)

## Data frames

* word wrapping

## Data frames

* obvious min and max
* 2nd most important thing I have learned. R is written assuming you are a fluent statistician. (I am not so there is a lot of learning)

## Data frames

* point out the items numbers in the row wrapping [15], 10.4 is the 15th element

## Data frames

## Packages

* multiple packages do the same thing. There are 3 packages to read xls files (2 of them suck)

## NA

## NA

* not that the sum of the variables is 32, 5 objects but removing the na takes it down to 4 (32/4 = 8)

## A real Data set

26,086 rows

1. set the working directory
2. point out: escape back slashes with double slashes like in c# or use the /
3. read the data from the file - read.csv is the preferred method. Read.table is slower but useful if you have other delimiters
4. show the environment variables
5. str shows a summary of the structure of an object
6. show the quick text with tab
7. factors
8. packages
9. help – can also use help(“searchtext”)
10. ... parameter : these are the list of parameters that the function summarize needs
11. optional parameters , similar to .Net, they have default values and access by referencing the parameter name

## Reusable scripts

1. comments
2. in a rerun script use function require
3. method returns
4. looping
5. sourcing the script
6. functions are treated like any other object

## lapply, sapply, tapply

* sapply returns a vector, lapply returns a list
* demo

## Scoping and free variables in R

* demo

## Scoping and free variables in R

## Unit Testing in R

* demo
* continuous integration is available for packages
  + Travis is a service that offers CI on top of github.
  + R is not (yet) a first class citizen on travis-ci

## Want to get started?

* up for the challenge

## Questions?

R coming to SQL

[R](http://mran.revolutionanalytics.com/documents/what-is-r/) is coming to SQL Server. SQL Server 2016 ([which will be in public preview this summer](http://blogs.technet.com/b/dataplatforminsider/archive/2015/05/04/sql-server-2016-public-preview-coming-this-summer.aspx)) will include new real-time analytics, automatic data encryption, and the ability to run R within the database itself:

<http://blog.revolutionanalytics.com/2015/05/r-in-sql-server.html>

Edward Tufty: <http://www.edwardtufte.com/tufte/courses>