R: [http://cran.r-project.org](http://cran.r-project.org/)

Rstudio: <https://www.rstudio.com/products/rstudio/download>

To see how to install R and Rstudio in windows, visit <https://www.youtube.com/watch?v=eD07NznguA4>

for Mac <https://www.youtube.com/watch?v=GFImMj1lMRI>

Using R as a Calculator

*# Add two numbers*

2 + 3

## [1] 5

*# Multiply two numbers*

2 \* 3

## [1] 6

*# Exponents*

2^3

## [1] 8

*# Be careful with parentheses*

2 + (3 \* 4)

## [1] 14

(2 + 3) \* 4

## [1] 20

*# Other built in functions*

exp(1)

## [1] 2.718282

log(10) *#Careful! This is natural log*

## [1] 2.302585

sin(2\*pi)

## [1] -2.449294e-16

floor(4.5)

## [1] 4

sqrt(9)

## [1] 3

factorial(4)

## [1] 24

Creating Variables

You can also store numbers in variables for later use.

Side note: Although R allows = as an assigmnet operator, I will usually use <-, as they do slightly different things.

*#Create a variable named x*

x <- 2

x

## [1] 2

*#Create a variable named y*

y <- x^2

y

## [1] 4

*#Create a variable named z*

z <- x + y

z

## [1] 6

The Curve Function

R is a very nice language for creating graphs. I highly recommend looking into ggplot2, as it produces beautiful/modern/publishable graphs quite easily. Base R has many simple to use functions for plotting such as, plot(), hist(), points(), lines() etc. For now though, I want us to become comfortable with the curve() function.

Lets try to plot the function

*f*(*x*)=2*x*2

*# Plot the function*

curve(2\*x^2) *#Plots from 0 to 1 by default*

*# Change the limits of the plot*

curve(2\*x^2, from=-4, to=4)

*# Make the plot pretty*

curve(2\*x^2, from=-4, to=4,

 col='blue', lwd=2, xlab='x', ylab='f(x)', main='This is a Title')

The Plot Function (Optional)

I chose to cover the curve function because of it’s simplicity. Another more versatile way of plotting is to use the plot() function.

*# Create a vector of x points to plot at*

x <- seq(-4, 4, by=1) *# Or equivalently, x <- -4:4*

*# Create vector of y points*

y <- 2\*x^2

*# Use the plot function*

plot(x, y, xlab='x', ylab='f(x)', main='Another Plot Title')

The plot function can take a type argument. Some of the options are

* ‘p’ for points (default)
* ‘l’ for lines
* ‘o’ for overlayed points and lines

We can also adjust the color, point type (pch), size (cex) etc. This stuff is all very google-able. (:

*# Make plot pretty*

plot(x, y, xlab='x', ylab='f(x)', main='Another Plot Title',

 col='blue', pch=21, bg='orange', type='o')

Consider the following function.

*f*(*x*)=1/1+exp(5−*x*)

f(x)=1/(1+x)