Markdown and Network Plotting

## Introduction

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. The goal of RMarkdown Notebooks is to write documents and reports that intersperse text and code.

The text is written in [markdown](https://rmarkdown.rstudio.com/authoring_pandoc_markdown.html). It has lots of different formatting options. Just take a look at this document for example, such as the “Introduction header” or the links in this section.

## Making documents

You can either interact with RMarkdown files within RStudio or “knit” them into finished documents. To create a finished document, click the “Knit” button in RStudio.

### Exercise

Try knitting this document now. It should create a popup window that will be much prettier and easier to read.

### Excercise

Create a new paragraph explaining what you hope to get out of this class. Create a header (Hint: the header for this section is called ‘Making documents’). When you’re done, knit the document again so you can see what your paragraph looks like in the finished document.

## Excecuting Code

When you execute code within RStudio, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button  or by placing your cursor inside the code chunk it and pressing Ctrl+Shift+Enter.

### Excercise

*Run the code below in both ways*

x <- c(43,643,765)  
  
print(paste("The mean of x is", mean(x)))

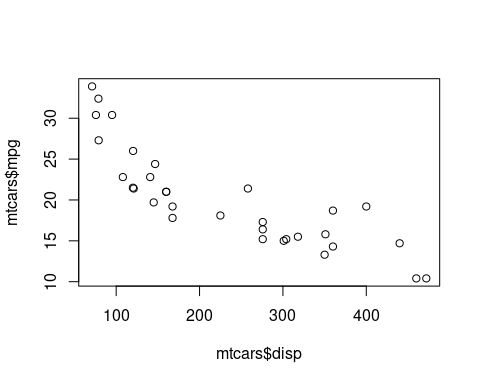
## [1] "The mean of x is 483.666666666667"

## Writing code

Code that will be run is marked with three backticks (`) followed by {r} and ends with three backticks. The simplest way to create a code block is to do Ctrl+Alt+I or to click the green Insert button  at the top of the Markdown window.

By default, the code and the output will appear in the created document, but you can set echo=FALSE and it will just show the output. Typically, you would want to do this when you’re creating a plot.

For example, the code below creates a scatterplot. For now, don’t worry about what all of the code means - we’ll come to that in later lessons.



### Exercise

Create your own code block below, and copy and paste the following code into it: hist(iris$Petal.Length)

If you do it correctly, you should see a histogram plot (a figure with a bunch of vertical gray bars)

## Packages

The last concept for today is “packages”. A package is a collection of functions and/or data. For example, igraph and tidygraph are two packages that make it much, much easier to work with network data.

There are two steps to using a package. First, you need to install it. This downloads the package to your computer. *You only need to do this once*.

If you want to use a package in your code, you will need to load it. This brings it into R so that R can use it. This is done with the library command.

### Exercise

Install the igraph package by copying and pasting the code below into the console (at the bottom left of RStudio) and hitting Enter

install.packages(“igraph”)

It should show a bunch of crazy messages while it installs, and take a couple of minutes. If it worked, then at the end it should say something like

DONE (igraph)

near the end, with no scary error messages.

### Exercise

Now that igraph is loaded, let’s create our first network graph!!

Create a new code block, and copy and paste the following into it, and then run it.

library(igraph)

G = erdos.renyi.game(10, .6) plot(G)