```
## Installing R (and RStudio) ----

#download R here. http://cran.cnr.berkeley.edu/
#download RStudio here (bottom of page). https://www.rstudio.com/products/rstudio/download/


## Basics ----

# is a comment. Anything after the # is going to be ignored by R

# you can Run lines either with button in the right corner of this window,
# or by hitting cmd+enter or control+enter depending on your operating system.

# The usually math operations work in R
2+2
3^4

#storing variables is with <- not =
four <- 2+2
# you can look inside a variable with it's name
four

# Generally you will be working with vecotrs or data frames

#these are vectors
v1<- 1:10
v2<-  c(1,2,3,4)
v3<- c("one", "two", "three", "four")

# we can see the contents with either the name of the variable or in the environment window to the right.
v1
v2
v3

# we can get a specific element of a vector with [] (keep in mind that R starts indexing at 1, not 0)
v1[5]

#data frames are nice ways to mere columns together (but they have to have the same length)
data.frame(v2, v3)
# Whats better is we can names the columns with =
df <- data.frame(numbers= v2, words= v3)
# in R, we use <- for variable assignment and = when we are renaming something in function calls.
df
# we can get back columns using their names
df$numbers
# or using their index (keep in mind that R starts indexing at 1, not 0)
df[2]

#we can go deeper to access specific parts of a data frame where df[row, column]
df[1,2]

# but if a data frame only has one column we still need the , to tell R we only want one number
newdf <- data.frame(1:10)
newdf
newdf[1]
newdf[1,]

# when you start using alot data, the head, tail, and summary functions become very import
bigData <- data.frame(nums = 1:10000, vals = (1:10000)^2)
bigData
head(bigData)
tail(bigData)

summary(bigData)

# R has a base ploting library
plot(bigData$nums, bigData$vals)


## Plotting with Data given in R  ----

data(AirPassengers) #import data from R
AirPassengers #look at the data
summary(AirPassengers) #quick summary of data
plot(AirPassengers) # plot over time
abline(h=280) #add in a reference for the mean passengers

# fit a linear model
fit<- lm(AirPassengers ~ time(AirPassengers))
summary(fit)
abline(fit, lwd=3, col= "blue") # not that abline will only work with straight (linear) lines

# you can check out all of the data sets in R with the following
data()
```

```r
## Reading and Writing your own data in R ----

#data usually isnt in the form of that last example, so lets do some more realistic stuff
airData<- as.data.frame(AirPassengers)
airData # now this is just a bunch of numbers and we need to make it into a nice data frame

# the airPassengers data was from 1949 to 1960 with data for each month, so
months <- seq(1,12) # same as 1:12
months
length(1949:1960)# figure out how many times we should repeat the months
monthNum <- rep(months, 12) # this will repeat what ever you want n times (16 in this case)
monthNum

year <-  rep(1949:1960 , each= 12) # this will give us the corresponding years for each month we have
year

# we can then convert these to dates using this as.POSIXlt function
date<- as.POSIXlt(paste0( year,'/',monthNum,'/', 1, " ", 0,':', "00:00 EST") )
date

# We can also have another column describing the month's first letter
monthLetter <- c("J", "F", "M", "A", "M", "J", "J", "A", "S", "O", "N", "D")

#putting it all together we have,
df <- data.frame( airData , monthNum, year, date, monthLet =monthLetter[monthNum], fracTime= (year + monthNum/12 ))
head(df)

plot(df$fracTime, df$x) #see how this plot is different?
# the base plot function in R tries to guess what you want
# the last time we used it,
# it knew we wanted the plot to look like a time series plot because our data came from R

# Most of the time it is easier to use ggplot2 to get the exact plot you want

install.packages("ggplot2") # you only have to install a package once
library(ggplot2) # each time you use a package you have to tell R that you will use it

ggplot(data = df, aes(x=date, y=x)) + geom_point() + theme_bw() #thats a little better

#but we can do more
ggplot(data = df, aes(x=date, y=x)) + geom_point() + geom_line() + theme_bw()+
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1)

# and more
ggplot(data = df, aes(x=date, y=x)) + geom_point() + geom_line() + theme_bw()+
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1)+
  geom_text(aes(label=df$monthLet),hjust=-.5, vjust=0)

# if we wanted to only graph the points in January, we can also subset
df$monthNum== 1 # this gives us booleans that we can pull from our large data set
subsetdf<- df[df$monthNum== 1,] # this is all of the January dates
ggplot(data = subsetdf, aes(x=date, y=x)) + geom_point() + geom_line() + theme_bw()+
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1)+
  geom_text(aes(label=subsetdf$monthLet),hjust=-.5, vjust=0)

# or if we only want to see the last two years of data
subsetdf<- df[df$year > 1958,]
ggplot(data = subsetdf, aes(x=date, y=x)) + geom_point() + geom_line() + theme_bw()+
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1)+
  geom_text(aes(label=subsetdf$monthLet),hjust=-.5, vjust=0)


# finally we can save our data by writing it to a file
#but first we should navigate to our folder on our computer
getwd() #where we are currently
setwd("C:/Users/evan/Desktop/R workshop (spring)") # where we want to go
write.csv(df, "airData.csv") # write to our file

# then if we wanted to clean up our RStudio we can run
rm(list=ls()) # or click the broom above the enviroment panel

# note how all of the variable we have been working with are now gone.

# finally if we want to start data analysis again on another day, we can get our data we just wrote out with
data <- read.csv("airData.csv")
data

## More Resources ----

# A common R tutorial through is availible as a package.
# You can visit this link for instructions on how to use it https://swirlstats.com/students.html

# A very good tutorial on how to use ggplot2. https://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html
# Example code for ggplot2 plots. http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

# Handbook that covers statistical tests in R. http://rcompanion.org/handbook/index.html
```