01\_Class\_Activity

Bill Perry

# In class activity 1:

*Note: This activity is really in place of the outline above which you should have read before class.*



# Objectives and goals

* Can you identify inductive or deductive reasoning?
* How do you formulate a question?
* Can you develop a prediction?
* What are hypotheses - Null and Alternate?
* What is a replicate and how do we sample?
* How do we organize data?
* How do we graph the data we gather?

# Schedule:

* Head to the field outside of Swenson and go to pine stand
* Discuss approaches to science and how to make observations
  + Point out the North versus South side of a pine tree and how weather might affect the needled dimensions
* Return to laboratory and measure pine needles
* Orgnize data
* Make graphs to see what data looks like visually

# Inductive reasoning approach

### If you were a person that sees the world through an inductive reasoning approach what would you do right now?

* Answer
  + start measuring needles
  + then measure needles on another species
    - make a generalization about needles as affected by weather or wind
* make a generalization about the species of pine trees and needle lenghts

# Deductive reasoning approach

### If you were a person that sees the world through a deductive reasoning approach what would you do right now?

* Answer
  + note that there seems to be differences in pine needles on the different sides of trees
  + make generalization that weather affects pine needle length
  + start measuring pine needles
  + test if the pattern exists or not
* Then maybe test trees in sheltered versus sheltered areas or shaded and sunny areas

# How does pine needle length vary

* windward side of trees differ from the leeward side of trees
* what might we expect if there is:

### No effect of weather

* Then there would be no difference in needle length
* This is called the Null Hypothesis and is denoted Ho

### There is a difference in needle length

* Could be shorter or longer
* Then we reject the idea above that there is no effect
* Accept the idea that there is a difference
* This is called the Alternate Hypothesis
* Note – we do not say if it is shorter or longer – this then becomes a prediction

# How would we collect data to test this?

### Can we collect needles from this tree and do the test?

* NO – this is because this tree might have short needles – no judging…
* This is called **pseudo replication**

### We have to collect needles from many trees

* The trees are the unit of replication here
* We can collect needles from the same tree and take the average of them and use that as a single replicate

# So go out and collect needles

* 20 needles from the windward side
* 20 needles from the leeward side
* Note that you need to collect them the same way and get the very base of the needle…

### When done gather back here and we will head back to the laboratory.

# Back in the laboratory

### So before we begin – what are the steps we need to decide?

1. How we measure these using calipers?
   1. Are there any things we need to pay attention to?
2. What are the variables we have identified?
   1. **Note is Date the same thing as date?**
   2. date
   3. group
   4. n\_s – north or south or even the degrees
   5. wind - windward or leeward as it may change
   6. tree\_no – group # may work
   7. Pine needle length – len\_mm
      1. What is the name we use?
      2. What are the units?
      3. Can we name the variable for both?
3. What is meta data?
   1. List of variables, description, units, possible values
   2. Saved as an associated text file
4. Ok so take the measures and recode in the shared google spreadsheet

# One last thing – Lets Estimate Error

* Select 3 pine needles and number 1, 2, 3
* Open a separate shared spreadsheet
* Have everyone measure the 3 needles
* Enter their name
* Enter pine needle number
* Enter the length
* Record the length of each

### When done each person can export the shared google drive as a CSV or comma delimited file and as an XLSX or excel file

# Now Lets Open R Studio

1. Download the compressed file pine\_needles.zip
2. Unzip this file in both windows and mac as it wont work
3. Open the folder and look at what is in there
4. Copy the files you have entered data into the data sub-directory with the same names as you see…
5. Then open RStudio
6. The window will look below – what is all this stuff…

# RStudio



**Important parts of the screen**

* Console – this is really R running and R studio is the interface
* Terminal – this is the back end of your computer without windows
* Environment – this is where things in memory are stored like
  + Dataframes
  + Graphs Files – these are the files that are available… this now is your home directory Plots – the plots you create
* Help – some form of help for you

More later - In the files you will see folders and you can click on them to see what is in there…

**you can also click the dots next to the green arrow to go back up a level** 

# Now lets open the project that I created

1. click file - open project and select the 01\_homework.Rproj file
2. your screen will now change as RStudio knows where home is



1. Note that in the upper right you will see 01\_homework so you know you are in the right spot

# How does RStudio work with R - <-

We need to cover a bit of syntax in R

### in the console

* r is case sensitive so X is different than x
* the <- is the assignment operator
* it stores whatever is on the right in a name that you have on the right
* try typing x<-7 then return
  + this will store a new object in the environment that is an x and in that is 7
* now type x and hit return and see what happens
  + a 7 will appear
* now type y <- 2 and enter
* now type x \* y and return
  + you should see that it multiplied the x by the y variable and you get 14
* THIS WILL GET WERIOUSLY OLD IF YOU HAVE TO RETYPE ALL THE TIME!!!
* So we make command files or scirpts

### now in a r script file

* lets click file and New File ... and R file
* this will open a new script file that you can write code and run it with CTRL and return or on mac command + return
* r is case sensitive so X is different than x
* the <- is the assignment operator
* it stores whatever is on the right in a name that you have on the right
* try typing x<-7 then return
  + this will store a new object in the environment that is an x and in that is 7
* now type x and hit command + return and see what happens in the console
  + a 7 will appear
* now type y <- 2 and enter
* now type x \* y and command + return
  + you should see that it multiplied the x by the y variable and you get 14
* This is one version - the other is a quarto markdown file we will see next

### Quarto Files - how we will roll in the class

* Now click the scripts folder where the scripts are stored
* Open - 01\_plain\_r\_script\_for\_pine\_needles\_blank.R by clicking on it
* Then follow along…
  + one tricky note - when you open these they think this is home and all files are here or under them

# How does RStudio work with R

* we could do all of our work with R or base R
* people have written a lot of helper functions called libraries stored in packages
* we use these a lot of these
* you install the package one time - a lot like buying a light bulb and screwing it in… you do that once
* then you can load the libraries stored in the package each time you use a library
* lets see how it works in a script I have made

# Here is the script I provided to work on:

### I have provided a lot of details here so you can see what is going on

You should have installed packages which is done below The # is a comment and allows you to write whatever you want and it won’ run

# install packages -----  
# install.packages("readxl")  
# install.packages("tidyverse")

Each script you run from then on you will load the libraries from within the package.

# Load the libraries ----  
library(readxl) # allows to read in excel files  
library(tidyverse) # provides utilities seen in console

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
✔ purrr 1.0.4   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(here)

here() starts at /Users/wlperry/Desktop/www\_umd\_biostats

# Loading files

Now like we did before with x and y we will do this with a spreadsheet from a CSV file or excel file

This will import the excel file

# Visualize data

Use GGPlot to graph the data

the line below loads the dataframe and what the aesthetics are

it does not tell ggplot how to add a layer of the geometry to show the data

## Tapestry Plot ——

knitr::opts\_chunk$set(  
 comment = '', fig.width = 4, fig.height = 3)  
  
ggplot(data = p\_df, aes(x=wind, y=len\_mm))



## XY Plot —–

notice the points are layered on top but some overlap

knitr::opts\_chunk$set(  
 comment = '', fig.width = 4, fig.height = 3)  
   
ggplot(data = p\_df, aes(x=wind, y=len\_mm)) +   
 geom\_point()



## XY Plot with dodged points ——

knitr::opts\_chunk$set(  
 comment = '', fig.width = 4, fig.height = 3)  
  
ggplot(data = p\_df, aes(x=wind, y=len\_mm)) +   
 geom\_point(position = position\_dodge2(width=0.2) )



# this dodges the points # position\_dodge2 or can use position\_dodge depending on grouping

What are the other ways to display the data?

## Histogram —–

knitr::opts\_chunk$set(  
 comment = '', fig.width = 4, fig.height = 3)  
  
ggplot(data = p\_df, aes(x=len\_mm)) +   
 geom\_histogram()

`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Note we really want to see the histograms colored by wind direction

We can map the wind aesthetic to a fill in the histogram

## Histogram Colors —–

knitr::opts\_chunk$set(  
 comment = '', fig.width = 4, fig.height = 3)  
  
ggplot(data = p\_df, aes(x=len\_mm, fill = wind)) + geom\_histogram( position = position\_dodge2(width = 0.5))

`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Histogram Bins —–

knitr::opts\_chunk$set(  
 comment = '', fig.width = 4, fig.height = 3)  
  
ggplot(data = p\_df, aes(x=len\_mm, fill = wind)) +  
 geom\_histogram( binwidth = 2,   
# sets the width in units of the bins - try different nubmers  
 position = position\_dodge2(width = 0.5))



## Other Plots if time

## Box and Whisker Plots

knitr::opts\_chunk$set(  
 comment = '', fig.width = 4, fig.height = 3)  
  
ggplot(data = p\_df, aes(x=wind, y=len\_mm, fill = wind)) + geom\_boxplot()

