Intro to NRES 710

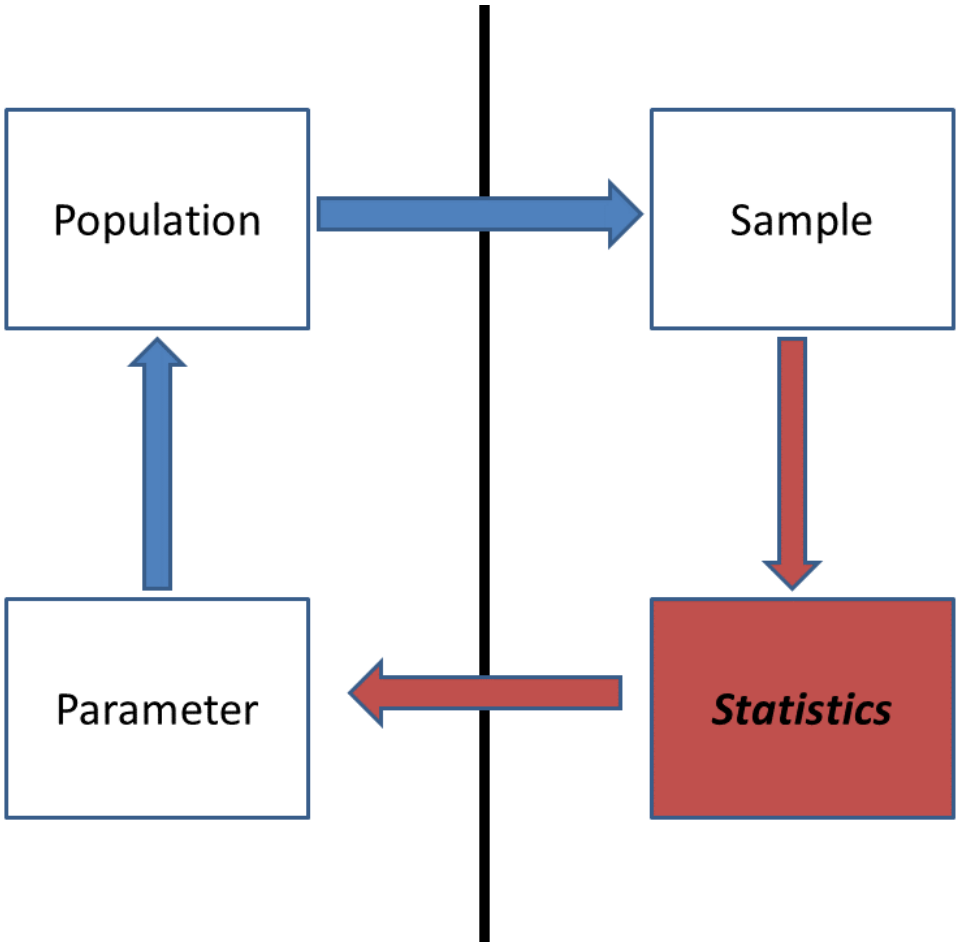
Welcome!

This course provides an introduction to both statistics and computer programming. Statistics and computer programming can in some ways be considered 'languages' because they help us to understand and communicate complex ideas (to other people and with computers!). But learning these languages is easier than learning a spoken language in many ways, because statistics and computer programming begin and end with pure logic! Statistics and computer programming are nothing if they don't make logical sense! Spoken languages on the other hand often don't follow completely logical structures- they don't always 'make sense'. So you can't 'get by' in most spoken languages simply by following a few basic rules. With statistics and computer programming, you can! With a solid grounding in basic rules of probability and statistics, and armed with some basic computer programming abilities (and some human creativity and ingenuity), you can go extremely far in making sense of data and communicating that understanding with others!!

Download the R code for this lecture!

To follow along with the R-based lessons and demos, right (or command) click on this link and save the script to your working directory

What is statistics?



Statistics is the process of making inference about a property of a *population* (a parameter) from a representative *sample*.

It's not just you. Many ecologists are fuzzy about the language, and even the concepts!

Why do we need stats class?

I tend to recommend that grad students take the least coursework possible- but I think stats is an exception. I think it's more efficient, fun, and rewarding to learn stats as a group rather than on your own. I hope you agree by the end of this class!

Types of data

Statistics is fundamentally about making sense of data. Therefore understanding statistics requires starting with an understanding types of data.

What types of data are there?
1. Categorical (Can be quantitative or qualitative) – represents characteristics, can be logical a. Nominal (labels, $>$ 2 categories) b. Dichotomous (2 categories) c. Ordinal (ordered units) 2. Quantitative a. Discrete: can be counted (number of individuals in a population) b. Continuous: not countable, but can be measured (total body length) i. Interval data: the difference between two values is meaningful but zero doesn't

mean the absence of a thing (e.g., temperature in Celsius) ii. Ratio: zero means complete absence (e.g., temperature in Kelvin). If zero means absence, then the ratio between two values is meaningful (something that weighs 4 g has twice the mass of something that weighs 2 g). Note that you can't say that for interval data (e.g., something with a temperature of 4 C doesn't have twice the heat of something at 2 C).

Why do we need to know about data types? We have to deal with these data types differently!

For example:

- Nominal data: frequencies, pie charts
- Continuous data: calculate percentiles, mean and variance.

Messing around with data types in R:

Let's explore some existing data that is easily obtained in R: iris, mtcars, titanic (install titanic package first)

We will also simulate some data on our own!

Notes about working in R:

R is open source, so it is 'messy'. Packages are being added all the time, there are millions of ways to accomplish the same task.

R is incredibly powerful and feature rich- you are NOT expected to memorize much, just know that the answer is always a few clicks away!

When in doubt, Google it!!

Here is a base R cheat sheet (from RStudio)- this is a great reference for most of the basic tasks you will need to perform in R. Keep it handy!

Learn to use R scripts- and save the R script frequently!! This is the primary record of what you've done and allows you and other scientists to reproduce your models, analyses and visualizations.

First R demo!

NOTE: for those wishing to follow along with the R-based demo in class, click here for an R-script that contains all of the code blocks in this web-based lecture.

All of you should have R and RStudio installed on your computers. See the links page for some useful references. I will be leading "bootcamps" on using R in two upcoming Saturday afternoons (Sep 12 and 19- virtual of course)- please consider attending if you don't have much experience in R!

First of all, R can be used as a calculator. Try it!

```
2+2                # use R as a calculator
```

```
## [1] 4
```

```
four <- 2+2        # define your first variable!
four
```

```
## [1] 4
```

```
five<-2+2          # you can make mistakes and define misleading labels- R will let you!
five<-2+3
five<-four+1       # you can use variables to define new ones
```

Q: what are those hashtags doing in the above code block? [these are 'comments' and they are super helpful- use them early and often!]

Note: Be careful about creating object names. Try to avoid any names that might already be defined as a function- like 'sum'.

Use variable names that will not confuse you.

Use Rstudio's autofill feature to avoid typos!

Variable names can not start with a number - e.g., 5pt

Explore R's existing datasets

```
# data()     # 'uncomment' this command and run it to explore built-in datasets
```

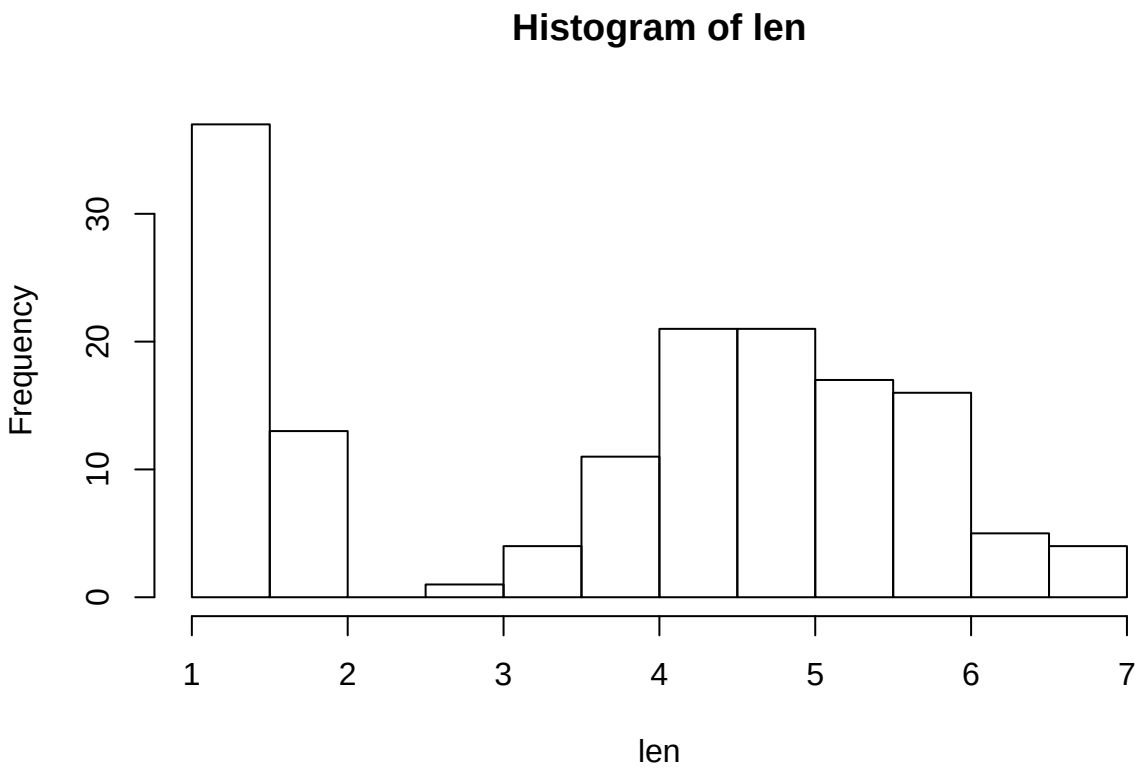
Let's start by working with the famous iris dataset:

```
#iris                    # this is a data frame- the basic data storage type in R
head(iris)              # [add your own comment here!]
tail(iris)

#  ?iris                # uncomment this to learn more about the iris dataset
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
len<-iris$Petal.Length
hist(len)              # what does this do? How could you learn more about this 'hist' function?
```




```
# Q: what kind of data are petal lengths?
```

Now let's switch to the 'titanic' dataset. To get this dataset you need to install an R package!

```
#install.packages("titanic")     # uncomment this command to install the package- you only need to inst

library(titanic)              # this 'loads' the package and needs to be done every time you run this s

data("titanic_train")
head(titanic_train)
# ?titanic_train              # uncomment and run to learn more about the data

# Q: What kind of data are those in the "Embarked" column?
# Q: What kind of data are those in "Pclass?"
```

We can even make our own dataset!

```
# Make our own data
# lets pull 15 numbers from the standard normal distribution

a <- rnorm(15)
a <- rnorm(15,mean=2,sd=0.5)
a
```

```
##  [1] 1.8621334 1.8944754 2.6791275 2.4098508 2.9629858 1.6358907 2.1033763
```

```
##  [8] 1.2499076 1.6026846 2.1053473 2.5132814 0.9364408 1.2709293 2.3474297
## [15] 2.1201997
```

```
# want to be able to repeat this? set the "seed."

set.seed(1234)
a <- rnorm(15)
a
```

```
##  [1] -1.20706575  0.27742924  1.08444118 -2.34569770  0.42912469
##  [6]  0.50605589 -0.57473996 -0.54663186 -0.56445200 -0.89003783
## [11] -0.47719270 -0.99838644 -0.77625389  0.06445882  0.95949406
```

```
# let's pull 15 numbers from the binomial distribution
b<- rbinom(15, size=1, prob=0.2) # we could "weight the coin"
b
```

```
##  [1] 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
```

```
# we can create categories:
unit<-rep(c("Control","+N","+P","+NP"),each=20)
unit
```

```
##  [1] "Control" "Control" "Control" "Control" "Control" "Control" "Control"
##  [8] "Control" "Control" "Control" "Control" "Control" "Control" "Control"
## [15] "Control" "Control" "Control" "Control" "Control" "Control" "+N"
## [22] "+N"      "+N"      "+N"      "+N"      "+N"      "+N"      "+N"
## [29] "+N"      "+N"      "+N"      "+N"      "+N"      "+N"      "+N"
## [36] "+N"      "+N"      "+N"      "+N"      "+N"      "+P"      "+P"
## [43] "+P"      "+P"      "+P"      "+P"      "+P"      "+P"      "+P"
## [50] "+P"      "+P"      "+P"      "+P"      "+P"      "+P"      "+P"
## [57] "+P"      "+P"      "+P"      "+P"      "+NP"     "+NP"     "+NP"
## [64] "+NP"     "+NP"     "+NP"     "+NP"     "+NP"     "+NP"     "+NP"
## [71] "+NP"     "+NP"     "+NP"     "+NP"     "+NP"     "+NP"     "+NP"
## [78] "+NP"     "+NP"     "+NP"
```

```
# we can even create a whole dataframe
my.data <- data.frame(
  Obs.Id = 1:100,
  Treatment = rep(c("A","B","C","D","E"),each=20),
  Block = rep(1:20,times=5),
  Germination = rpois(100,lambda=rep(c(1,5,4,7,1),each=20)),
  AvgHeight = rnorm(100,mean=rep(c(10,30,31,25,35,7),each=20))
)
head(my.data)
```

We can also import data from files stored on our computers (or even directly from the web)

```
# Then, we can import data.

# Don't forget to set your working directory (or just make sure you're using an Rstudio Project). I did
```

```
# setwd("~/Desktop")       # uncomment and run if you want to set the desktop as your working directory

#Read in the data. Note that the file needs to be in csv format, the name must be in quotes, and the na

# Pleach<-read.csv("PbyTime_Bio.csv", header=T) # obvs, this won't work for you because you don't have

# Use your own file to try it out. This is an example!
```

Note: I recommend always using Rstudio Projects. This reduces the hassle of setting working directories. By default the project directory becomes your working directory!

math/bio "stats chats"

Paul Hurtado (math/stats dept), Ken Nussear (geography) and I host informal sessions for grad students in EECB, NRES, Geography etc (and faculty) to discuss data analysis questions. This can be a good opportunity to ask questions, find out more about what types of data and questions your peers are working with, and contribute some insights! We haven't figured out a time yet for this semester, but I'll keep you posted.

Remember, I need your frequent and honest feedback in order to make sure that the level is appropriate and the topics are useful!

–go to next lecture–