

Functions: Creatures who live in the Realm of R

Beau Benjamin Bruce, MD, PhD

Assistant Professor of Ophthalmology,
Neurology, and Epidemiology
Emory University

What is computer programming?

Essentially, programming is creating a set of instructions to complete a task that you would like to be carried out – sort of like a recipe.

Human vs. Computer

- Humans usually understand vague instructions
- Fill in the gaps (or at least attempt to)
- Prefer “interesting”, complex cognitive tasks
- Flexible
- Usually friendly
- When not, at least emotions can guide you
- Computers need *everything* spelled out in minute detail
- Do what you ask: nothing more, nothing less
- Good at the most boring tasks you can imagine
- Inflexible
- Rarely friendly
- Emotionless

Why this learning process is different?

- It is not about learning about facts (although you need to know some), it is about thinking
- It is not about learning a specific way of doing things that your teacher shows you and then repeating them in the same fashion (although there will be models), it is about creating your own way of doing things
- It is not about finished products, but about tools that you must then apply to make finished products

Analogies

- Learning a foreign language
- Cooking

Analogies

- Learning a foreign language
- Cooking

Both have different levels of skill, both start in a similar manner to programming, etc.

Learning a foreign language

- Very relevant since we will indeed be learning a new language. However, this language is NOT a human language, but a alien language – that of the computer
- The computer does not understand nuance
- The computer does not understand emotion
- The computer does not know how hard you are trying

Learning a foreign language

- When you are not understood by the computer it is always ALL your fault (unlike a human)
- The computer talks back in only three ways:
 1. Giving you the answer you want
 2. Giving you the wrong answer (a "bug")
 3. Giving you no answer (an error)
- Only you can differentiate 1 & 2

Learning a foreign language

- Just like learning a foreign language you start with only a few words and you use patterns or models rigidly
- As you advance, you realize that those patterns are generalizable allowing you to “abstract” the essence and apply it to new situations
- Unlike human languages though, those abstractions are less likely to have idioms

Learning a foreign language

Foreign language

- Grammar & Syntax
- Nouns
- Verbs

Computer language

- Grammar & Syntax
- Data
- Functions

Cooking

- When starting you use someone else's recipes
- Then, you start to try this or that, sometimes it tastes good (right answer), sometimes it doesn't taste good (bug), and sometimes it is a terrible mess (error)

Cooking

- You advance to be become a chef where you imagine what you want and make a completely new dish

Things to keep in mind

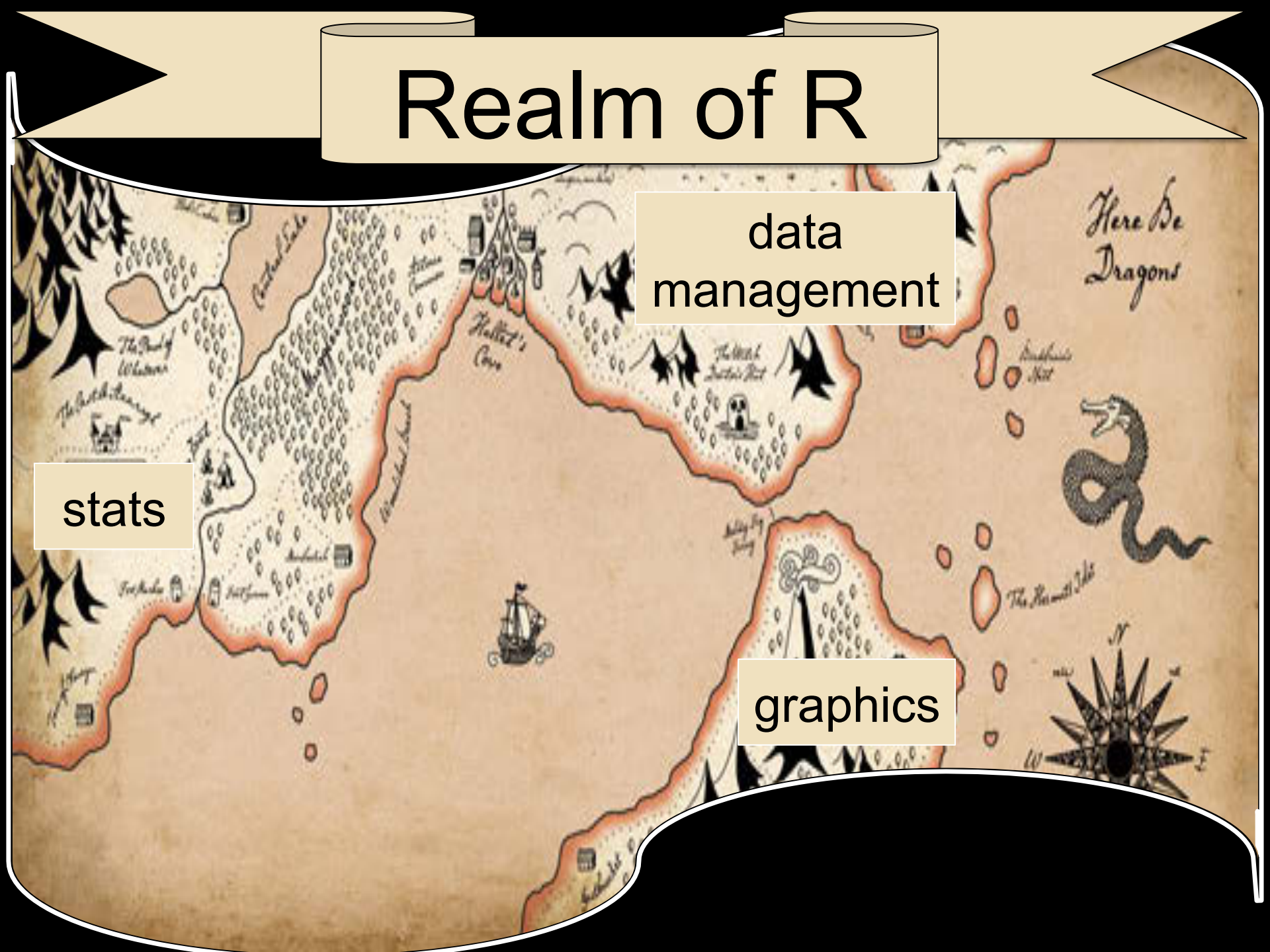
- Requires LOTS of trial and ERROR!
Especially when learning!
- Every capital letter, comma, parenthesis, & space likely matters
- In time you will become better if you keep working at it
- Indeed, there will be times that you feel that you are creating magical incantations

Realm of R

data
management

stats

graphics



Real

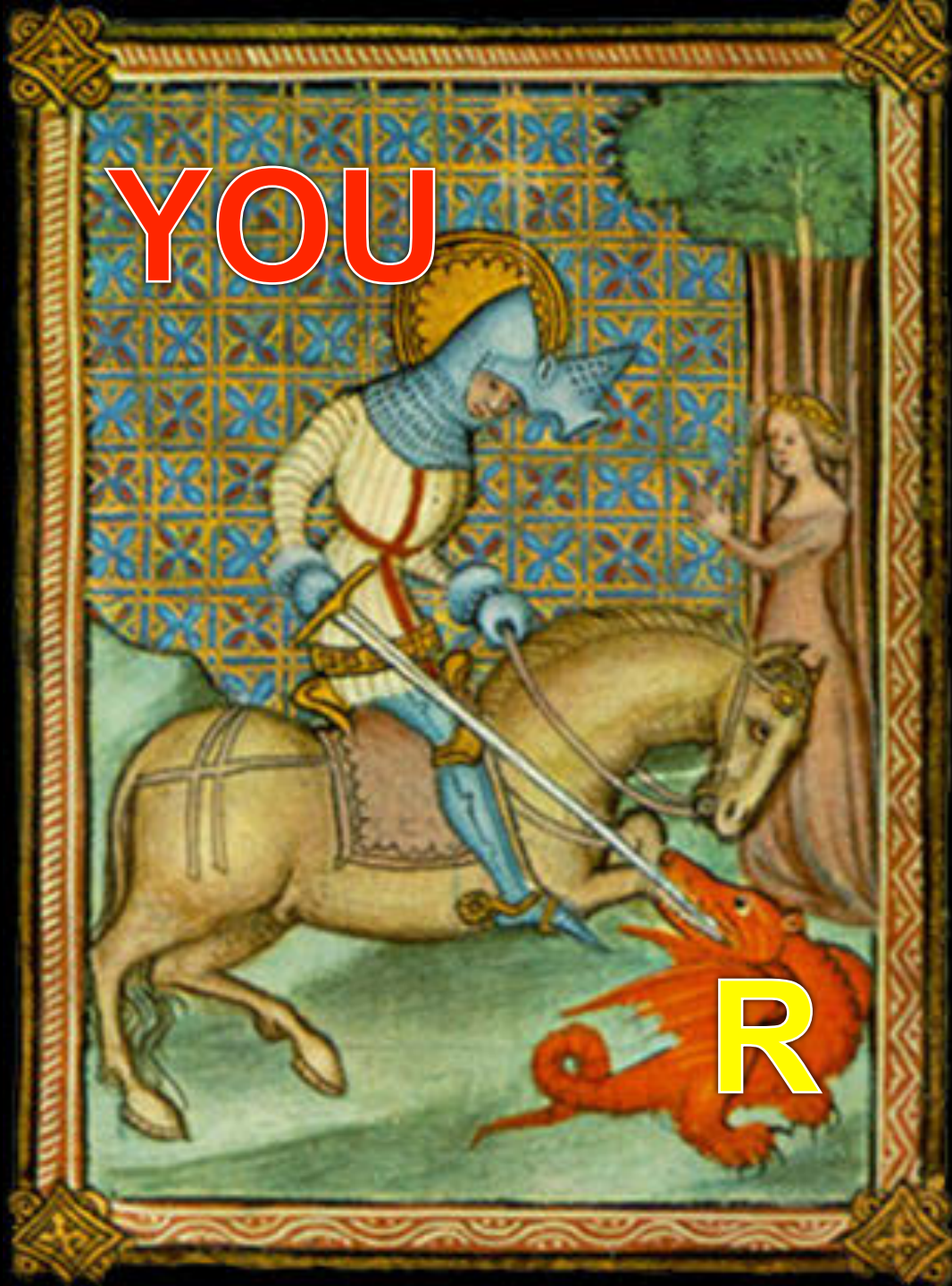
Here Be
Dragons

stats

Here Be
Dragons



YOU



R

Why R?

- Free!
- Exceptional (and easy to use) graphics
- Interactive and flexible
- Easy to modify and expand

Why R?

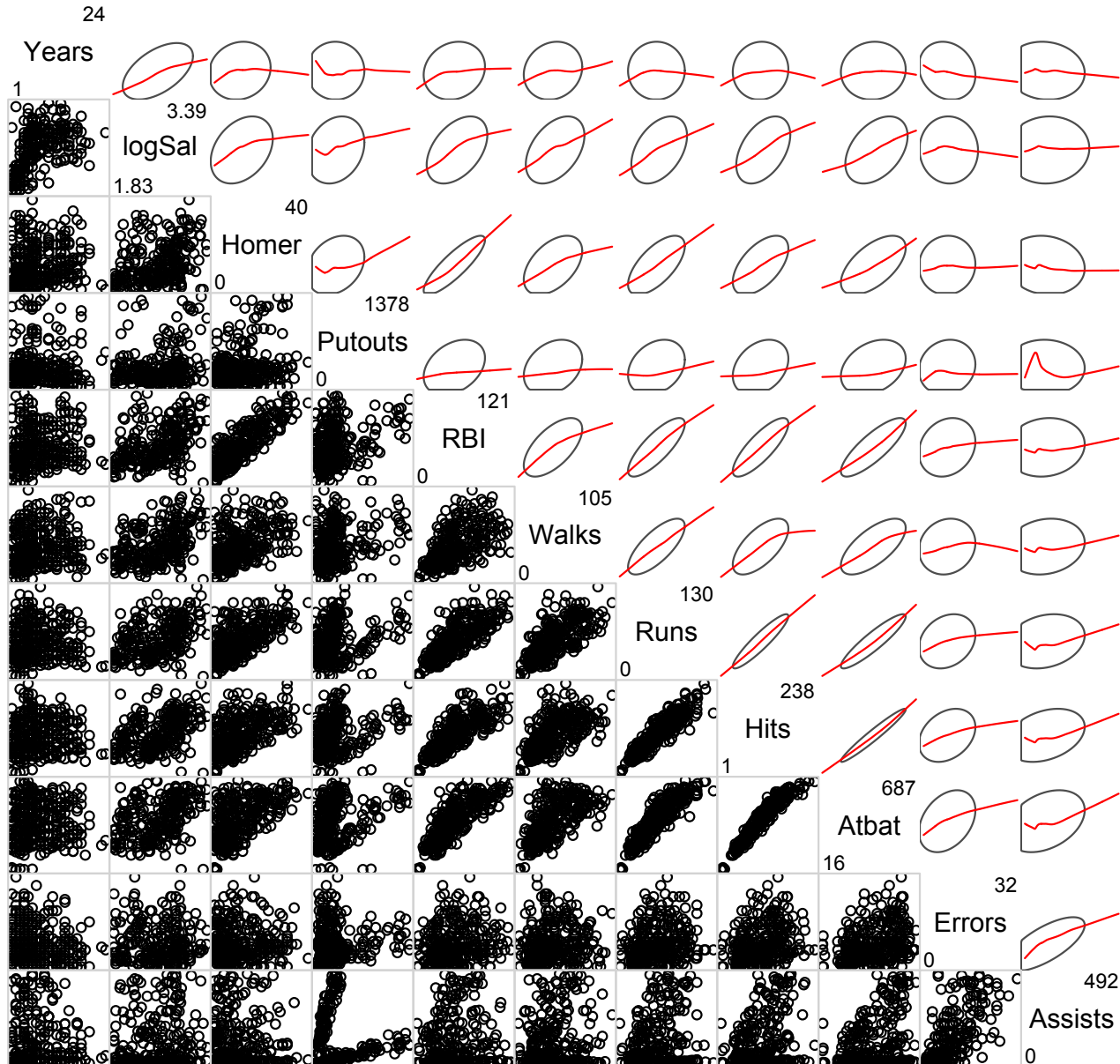
- Cutting edge methods available (usually before SAS, SPSS, or Stata)
- Interfaces easily with other software (databases, BUGS)

To boldly go...

- **Mission:**
Enable the best and most thorough data analysis possible
- **Prime Directive:**
The computations and the software for data analysis should be trustworthy
 - do what they claim
 - and be seen to do so

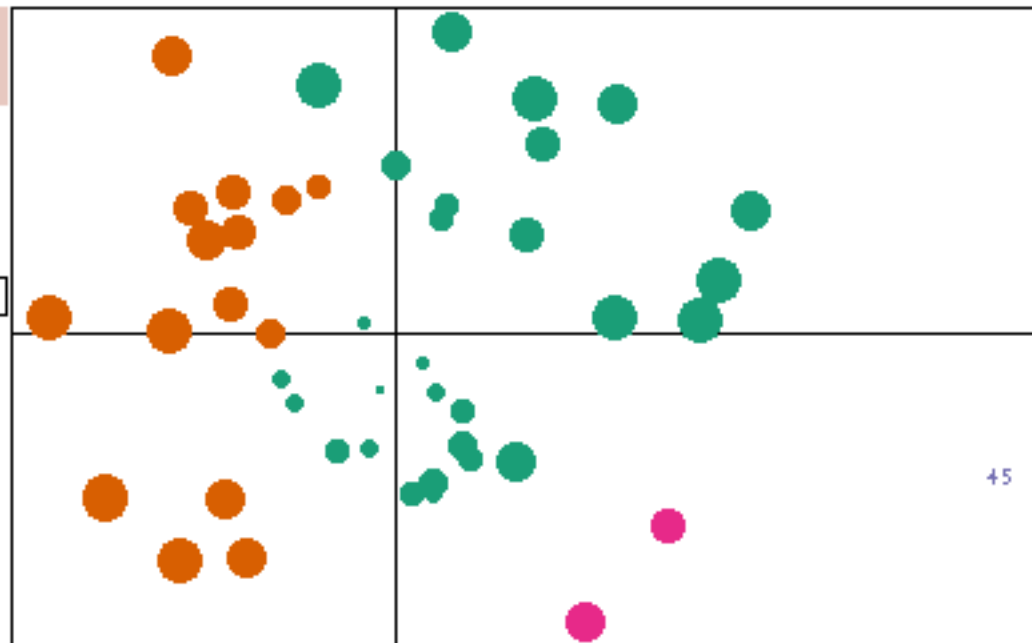
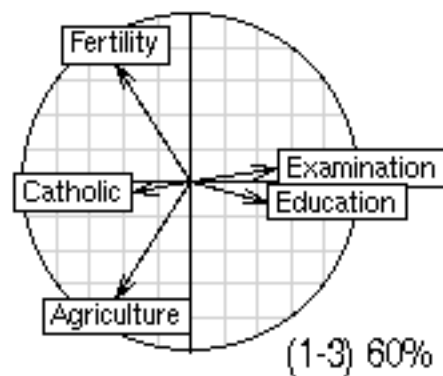


Baseball correlation ellipses



PCA 5 vars

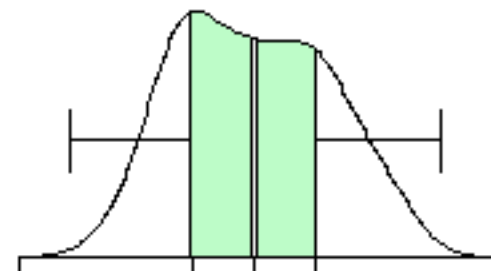
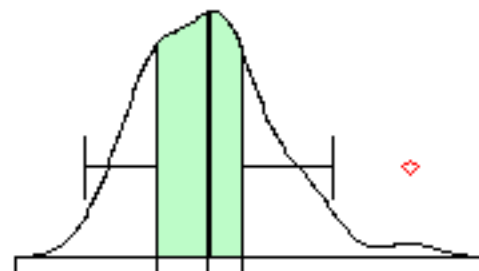
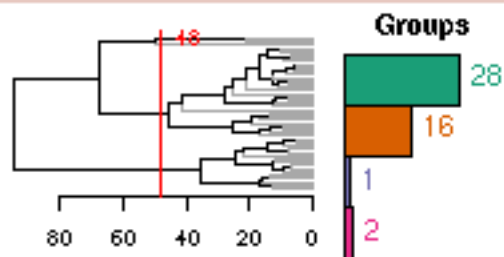
`princomp(x = data, cor = cor)`



Clustering 4 groups

Factor 1 [41%]

Factor 3 [19%]



Why not?

- Steep initial learning curve
- You get what you pay for, i.e., no commercial customer support
- Occasionally, an area lags behind other software (e.g., structural equation modeling)

Why not?

- Memory limits working with large datasets, but there are usually ways around it
- Sometimes your “mistakes” are silent and can make data cleaning and analysis error prone for the inexperienced

My advice

- My opinion concurs with the following quote:

“Life is short, use the command line”

– Michael Crawley

“Statistical Computing:

An Introduction to Data Analysis

Using S-Plus.”

My advice

- No matter what software you use you need to have a record of exactly what you did
- It can be extremely difficult to replicate what you do sometimes (maybe because you made a mistake somewhere!)

I want to point and click...

For novice users, there is no doubt that the graphical user interface (GUI, i.e., a windows based system) is much easier to use.

I want to point and click...

Even for advanced users, there are many tasks for which a GUI is more efficient (e.g., remembering infrequently used or obscure commands) than a command line interface (CLI, i.e., command prompt) or for mouse-keyboard two handed tasks (left hand for key combination, right for selecting/moving).

I want to point and click...

However, for common, repetitive tasks (e.g., cut, paste, save, etc.) the keyboard will save you time:

- moving from the keyboard to the mouse or vice versa (0.4 seconds)
- clicking a button with a mouse (2.5 seconds)
- using a keyboard shortcut (2.0 seconds)

I want to point and click...

So if you can keep your hands on the keyboard, you can save about a second per operation. Seconds become minutes, minutes become hours.

What learning R is like:



**But don't we all want to be
wizards?**



I went to Georgia Tech: I can do that...



<https://www.youtube.com/watch?v=98nNpzE6gls>

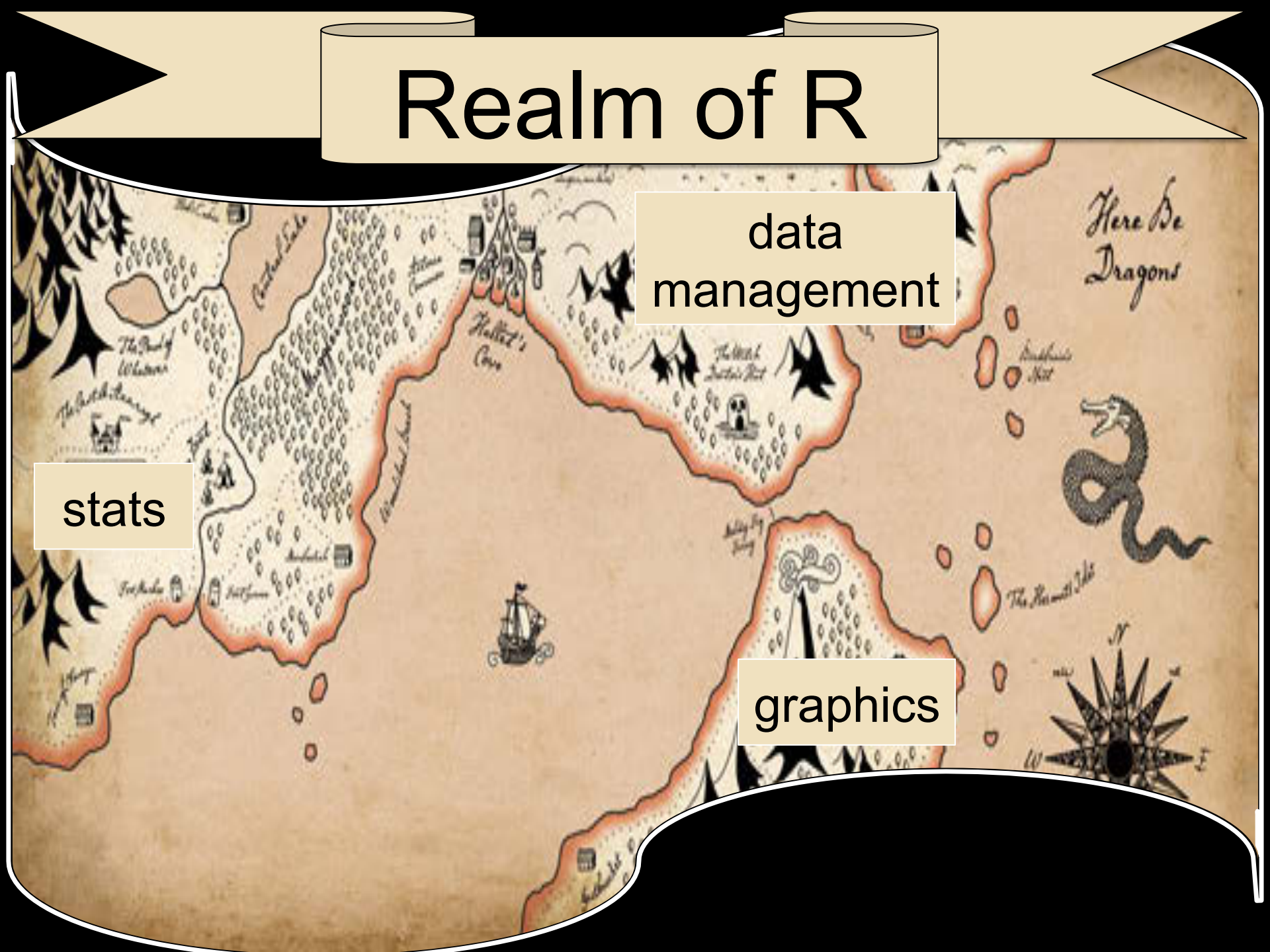


Realm of R

data
management

stats

graphics

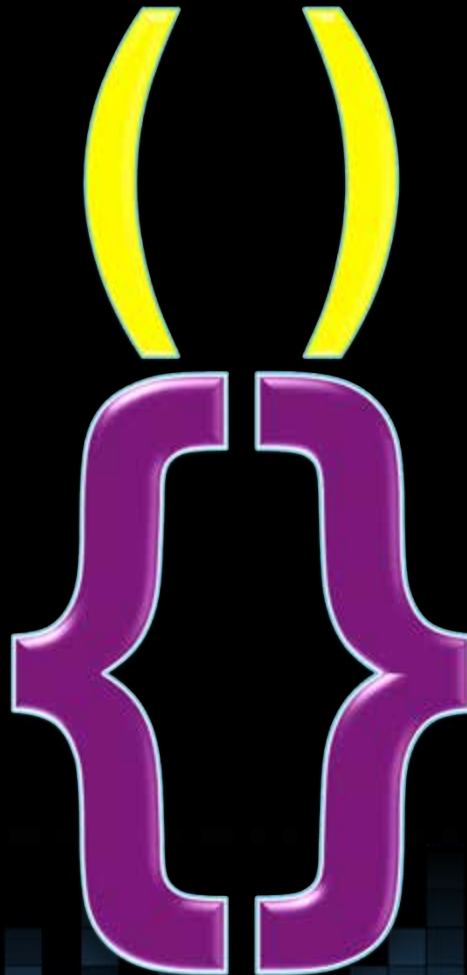


In the Realm of R

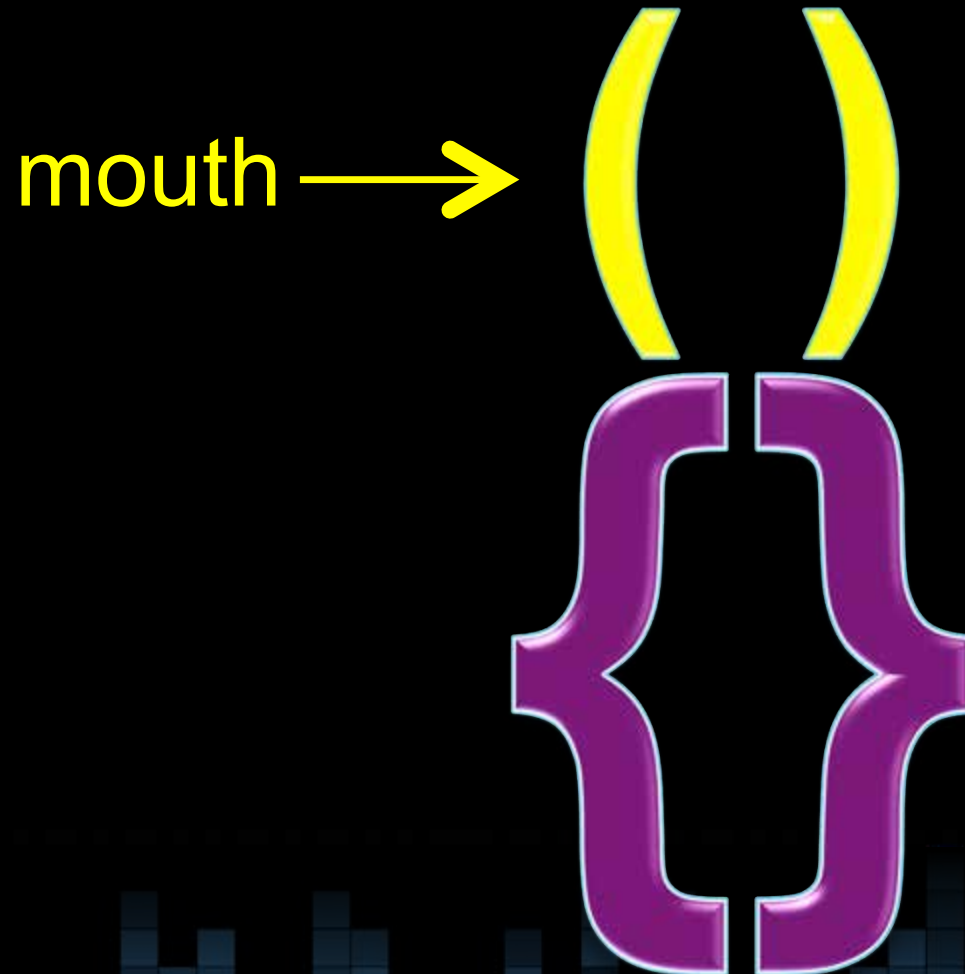
- You are the king and queen
- Whether you are also a wizard and dragon slayer is only a matter of effort and experience



Anatomy of Creatures in the Land of R: Functions



Anatomy of Creatures in the Land of R: Functions



Anatomy of Creatures in the Land of R: Functions

mouth → ()

body → { }

Anatomy of Creatures in the Land of R: Functions

name

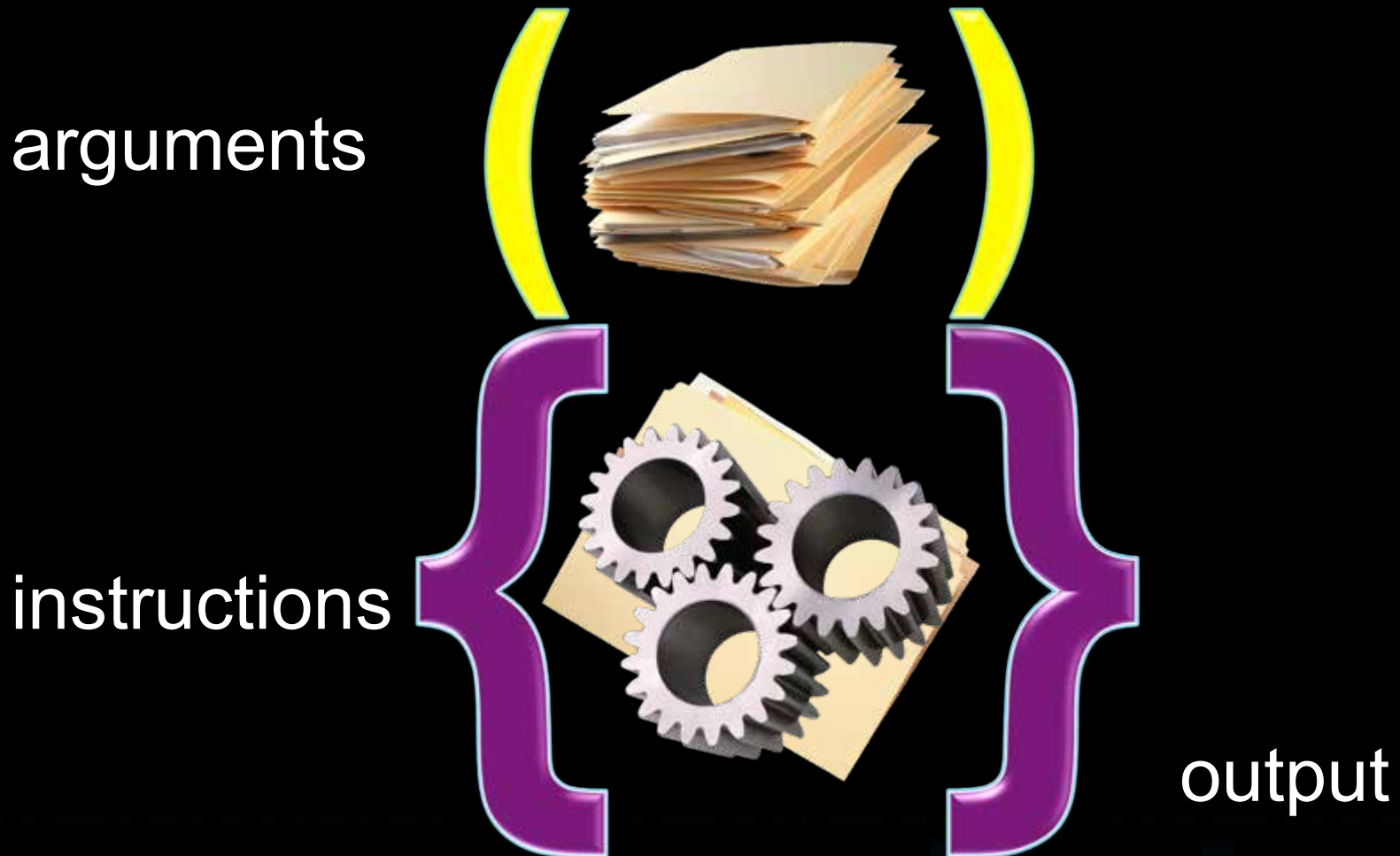
mouth →



body →

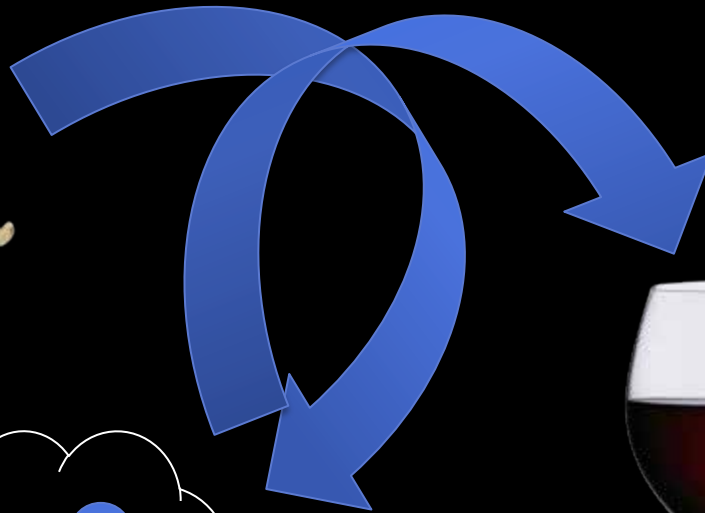
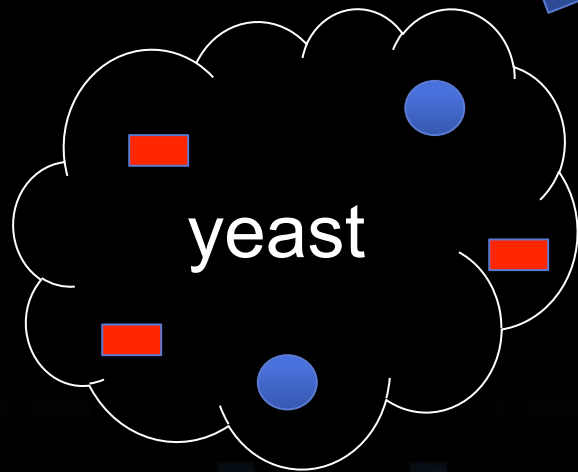


Anatomy of Creatures in the Land of R: Functions

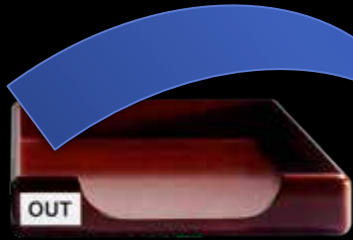




sugar



input
data



function



output
data

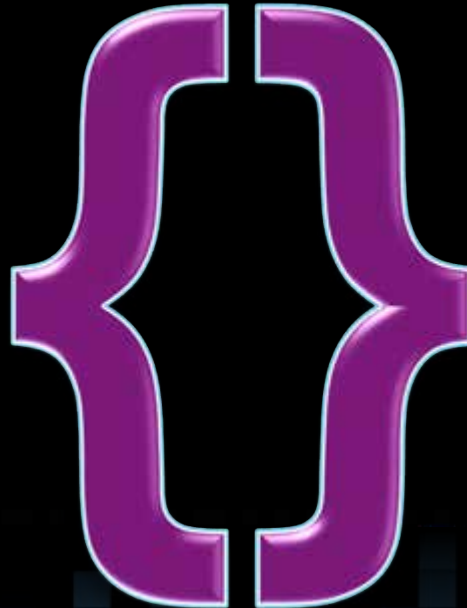
Anatomy of Creatures in the Land of R: Functions

name

mouth →



body →



shy!



**Wizards
in the Realm of R
you can even
create new creatures!!!**

Data: Food for functions in the Realm of R

Beau Benjamin Bruce, MD, PhD

Assistant Professor of Ophthalmology,
Neurology, and Epidemiology
Emory University

What do People eat?

- Simple foods
 - apple
 - beef
 - grapes
 - lettuce
- Composite foods
 - hamburger
 - pizza
 - soup

What do R Creatures (Functions) eat?

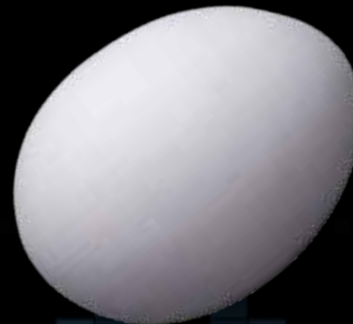
Data are the food for functions

Two types:

- atomic (simple foods)
- composite (composite foods)

Atomic Data/Food

- Numeric
- Character
- Logical



Numeric

1

3.14

-3245

Character

- Text
- Also called strings or character strings

"Hello"

"How are you?"

"I'm fine"

'Have a good day'

'Beau'

"Beau"

'I'm fine' => ERROR

Logical

- The simplest answer a computer can give

TRUE

FALSE

True => ERROR

TRUE = yes; FALSE = no

Creatures that tell you if what they are eating is a certain kind of food

- `is.numeric`, `is.logical`, `is.character`
- feed them something:
- `is.numeric(1) ==> TRUE`

Break it down

`is.numeric(1)` => TRUE

- name: `is.numeric`
- mouth: `()`
- food: `1`
- body: can't see (remember - shy!)
- output: TRUE

"food(s)" when in the mouth is/are also called "argument(s)"

Creatures that tell you if what they are eating is a certain type

- `is.numeric`, `is.logical`, `is.character`

- `is.numeric(1)` \Rightarrow `TRUE`
- `is.numeric("A")` \Rightarrow `FALSE`
- `is.logical(TRUE)` \Rightarrow `TRUE`
- `is.logical(FALSE)` \Rightarrow `??`
- `is.character(1)` \Rightarrow `??`
- `is.character("1")` \Rightarrow `??`

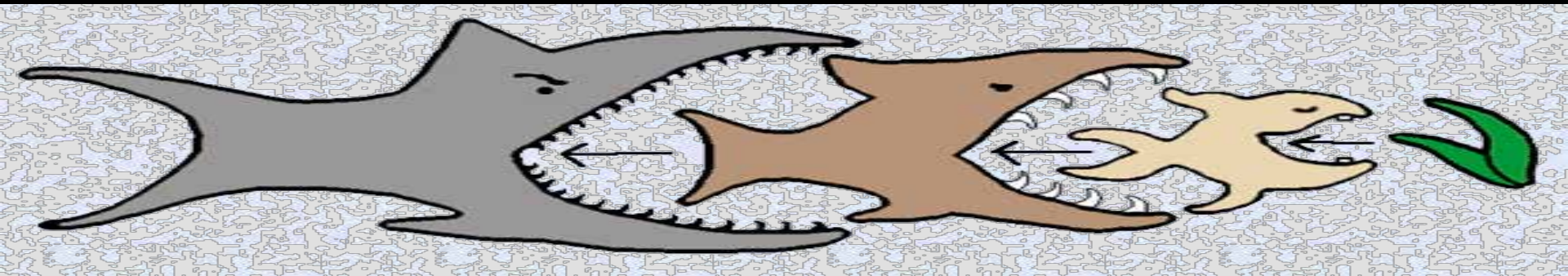
Composite Data/Food

- Many, many types!
- Vectors, matrixes, arrays, dates, lists, data frames...



Composite Data/Food

- Many, many types!
- Vectors, matrixes, arrays, dates, lists, data frames...
- Everything in the land of R can be "food/data" even the "creatures/functions"



Vectors

- The creature that makes vectors is named `c`
- Separate food items with commas (,)
- All food must be of same *atomic* type for a vector

`c(1,2,3,4)`

`c(TRUE,FALSE,TRUE)`

`c(1,TRUE,2,"A") => BUG*`

* Will become same as `c("1","TRUE","2","A")`

Working with vectors

- `is.numeric`, `is.logical`, `is.character` like to eat vectors too:

`is.numeric(c(1,2,3,4))` \Rightarrow TRUE

`is.numeric(c(TRUE,FALSE))` \Rightarrow ??

`is.logical(c(TRUE,FALSE))` \Rightarrow ??

Working with vectors

- Use various functions, e.g.:

+ => add

- => subtract

* => multiply

/ => divide

exp => e to the power of

`exp(c(1,2))` => 2.718282 7.389056

`c(1,2,3,4) * 2` => 2 4 6 8

**How is * a function –
where is its mouth???**

HERE BE DRAGONS!



How is * a function – where is its mouth???

- Some R creatures are more shy and they don't normally even show you their mouths!
- However, if you use their name in a special way they will open up a little:

How is * a function – where is its mouth???

`c(1,2,3,4) * 2` => 2 4 6 8

``*(c(1,2,3,4),2)` => 2 4 6 8

- ``` is called a backtick, it is NOT a single quote `'`
(see the subtle difference?)
- Obviously it is easier to use `*` the first way; this is one of the “idioms” of the R language

Naming vectors (or anything in R)

- Typing a long vector over and over gets old
- Best way to hold onto it during your R session is to name it
- You use the special function: `<-`
 - again one that is too shy to show its mouth
 - but now you know how you could get it to show its mouth

```
myvec <- c(1,2,3,4)
```

```
myvec * 2    => 2 4 6 8
```

Naming vectors (or anything in R)

`<-` is probably in the class of the MOST shy R creatures: you don't see it's mouth, body, or output!

`myvec <- c(1,2,3,4)` \Rightarrow no output shown

but again ask nicely and it will show you:

`(myvec <- c(1,2,3,4))` \Rightarrow 1 2 3 4

this a common R "idiom" to use for functions that hide their output when you want to see it

Can also use <- to change a value

```
myvec <- c(1,2,3,4)
```

```
myvec          => 1 2 3 4
```

```
myvec <- c(2,3,4)
```

```
myvec          => 2 3 4
```

```
myvec[2] <- 5
```

```
myvec          => 2 5 4
```

Taking apart vectors

- Sometimes you want to get the pieces (elements) of a vector back out in order to use them individually or small parts of a vector:

```
myvec <- c("a","b","c","d")
```

```
myvec[1]           => "a"
```

```
myvec[1,3]         => ERROR
```

```
myvec[c(1,3)]      => "a" "c"
```

Taking apart vectors

- There is another very shy function that can be helpful when selecting a section of a vector

1:3 => 1 2 3

myvec[1:3] => "a" "b" "c"

c(1:3,8:10) => 1 2 3 8 9 10

New Food: Factor

A special vector that lets R know it should be considered categorical.
Created by the creature named factor:

```
> (a <- c(1,1,2,2))  
[1] 1 1 2 2  
> summary(a)  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
  1.0    1.0    1.5    1.5    2.0    2.0   
> summary(factor(a))  
1 2  
2 2
```

New Food: Factor

Usually want better names:

```
> factor(a, labels=c("M", "F"))  
[1] M M F F  
Levels: M F
```

Note that one of the foods is given a name in the mouth of factor, i.e., labels. Why?

Aside: more about the care and feeding of Creatures in the Land of R

- It turns out all foods can be named in the mouth of the creatures, but you only have to use the names in certain circumstances
- Let's look at the help file (the ultimate field guide for all the creatures and natural foods in the land of R) for factor

type: ?factor -or- help("factor")

Usage:

```
factor(x = character(), levels, labels = levels,  
       exclude = NA, ordered = is.ordered(x), nmax = NA)
```

Arguments:

x: a vector of data, usually taking a small number of distinct values.

levels: an optional vector of the values that 'x' might have taken. The default is the unique set of values taken by 'as.character(x)', sorted into increasing order _of 'x'_. Note that this set can be specified as smaller than 'sort(unique(x))'.

labels: _either_ an optional vector of labels for the levels (in the same order as 'levels' after removing those in 'exclude'), _or_ a character string of length 1.

Aside: more about the care and feeding of Creatures in the Land of R

```
factor(x=a, labels=c("M","F"))  
factor(a, labels=c("M","F"))  
factor(labels=c("M","F"), x=a)
```

- All produce the same output
- Must use name if skip one of the arguments (in this case you are skipping levels)
- Must use name if use out of order (but don't do this)
- *These principles apply to all functions*

New creature: data.frame

- A special list of vectors all with same length (i.e., the number of observations)

```
data.frame(age=c(3,2,3,3,1,2,4,4),  
           sorethroat=factor(c("y","y","y",  
                               "y","n","n","n","n"))))
```

two variables, 8 observations

New creature: data.frame

- Usually will create data.frames by importing them from other files (e.g., Excel as we did earlier)
- List variable names:
 - names(iih_data)
- Access variable by name with \$:
 - (a very shy creature)
 - lih_data\$sex

New creature: data.frame

- Access row by number:
 - `iih_data[3,]`
- Access column by number:
 - `iih_data[, 3]`
- Access rows by test:
 - `iih_data[sex == "M",]`
- Change a value
 - `iih_data$sex[3] <- "F"`

Special human food that is hated by the creatures who live in the land of R

- Called comments
- Start with the character #
- Anything after a # is spit out and ignored by R creatures:

```
1 + 1 + 1 # + 1 + 1
```

```
[1] 3
```

Comments

Very useful for humans (often yourself!)
reading your code

Use in order to remember why you did
something the way you did

Use it to document what you are doing
for other humans





Data Types

Heather E. Moss, MD, PHD

Beau B. Bruce, MD, PhD

Levels of Measurement

- What type of data do you have?
- Important for choosing the right statistical test
- Important for knowing how to treat the variable in software

A framework valuable for statistics

- Nominal
- Ordinal
- Continuous

Nominal

- Nom = name
- For categories with no inherent order
- Examples:
 - Race and ethnicity
 - Sex

Ordinal

- Ord = order
- For categories with an inherent order
- Examples:
 - Cancer stage
 - Opinion on a 5-point scale

Continuous

- Contin = continuity, no gaps
- For values that could you could measure with arbitrary precision (assuming you could build the device)
- Distance between values has meaning

Continuous

- Examples:
 - Age
 - Weight
- Often some (real) confusion between ordinal and continuous

Ordinal vs. continuous

- Ordinal no clear amount of distance between categories – is stage IV cancer “double” stage II?
- Continuous: my height is 67.7632... inches

Ordinal vs. continuous

- Can sometimes treat ordinal as continuous:
 - A large number of closely spaced ordinal values
 - A smaller number of regularly spaced values

Other terms relevant for choosing statistics

- Categorical (at least nominal) or qualitative
 - Dichotomous – two categories
 - Polytomous – three or more categories
- Quantitative
 - Continuous

Representations in R

Level of measurement	Representation of type in R
nominal	factor
ordinal	numeric -or- factor(..., ordered = TRUE)
continuous	numeric

Nominal, ordinal, or continuous?

- Patient age
- Patient sex
- Photographic quality (5 point scale)
- Eye (right vs. left)
- Number of photographs

VA: options for representation

- Continuous
 - Log MAR
- Ordinal
 - Quartiles
 - Clinically relevant groups
- Nominal
 - Quartiles
 - Clinically relevant groups
 - Dichotomous



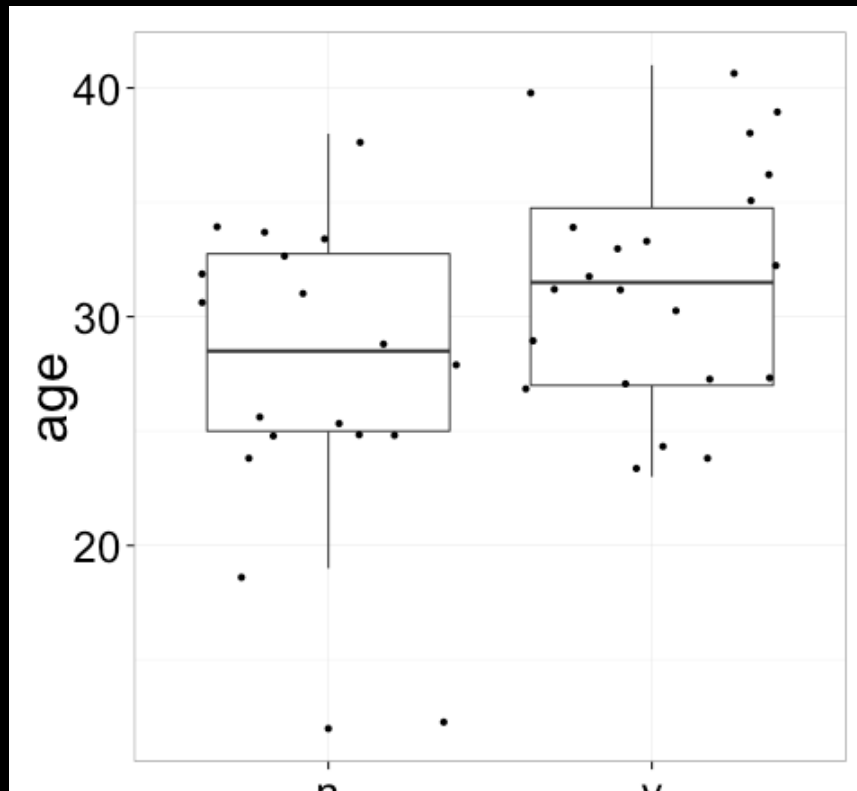
Comparing means / locations

Beau B. Bruce, MD, PhD

Assistant Professor of Ophthalmology,
Neurology, and Epidemiology
Emory University

Study Question

- Is the age of our subjects different at baseline by treatment status?

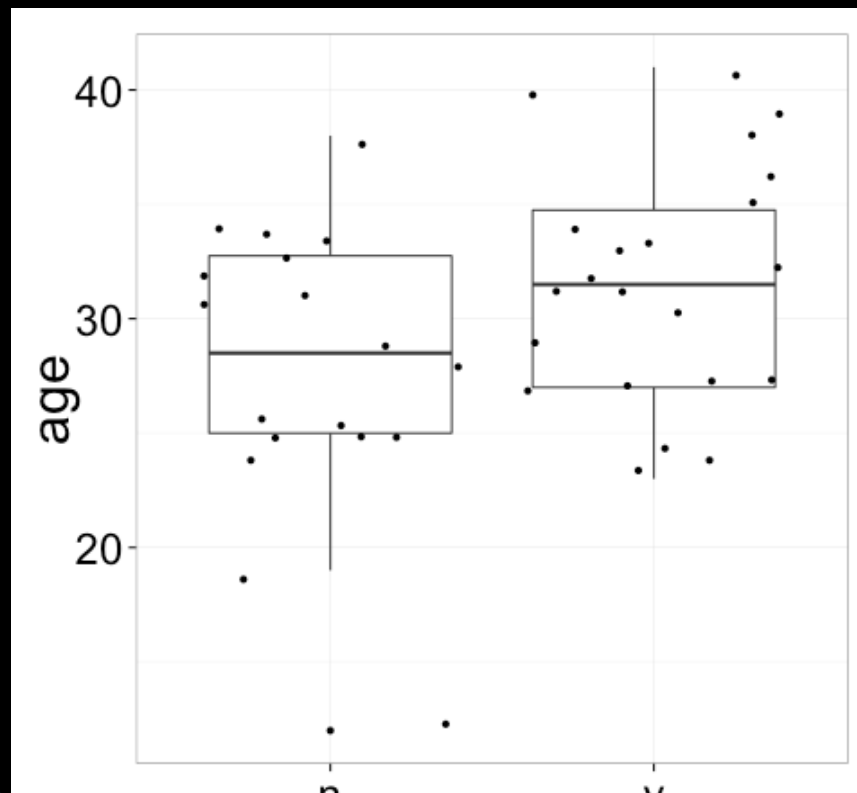


What do you mean by different?

- Group level – estimation
 - Average / mean
 - Median
 - Variance
- Individual level - prediction

Study Question

- Is the average (mean) age of our subjects different at baseline?



Two sample t-test

- Used to compare whether the means of two samples are different
- Assumptions:
 - Samples are *independent* of each other
 - Each sample is drawn from a *normally distributed population* or each sample is of a "large" size

Independent

- Says that seeing one value tells you nothing about another value
- Assumption broken in situations where you have repeated or longitudinal measurements on the same individual

Independent

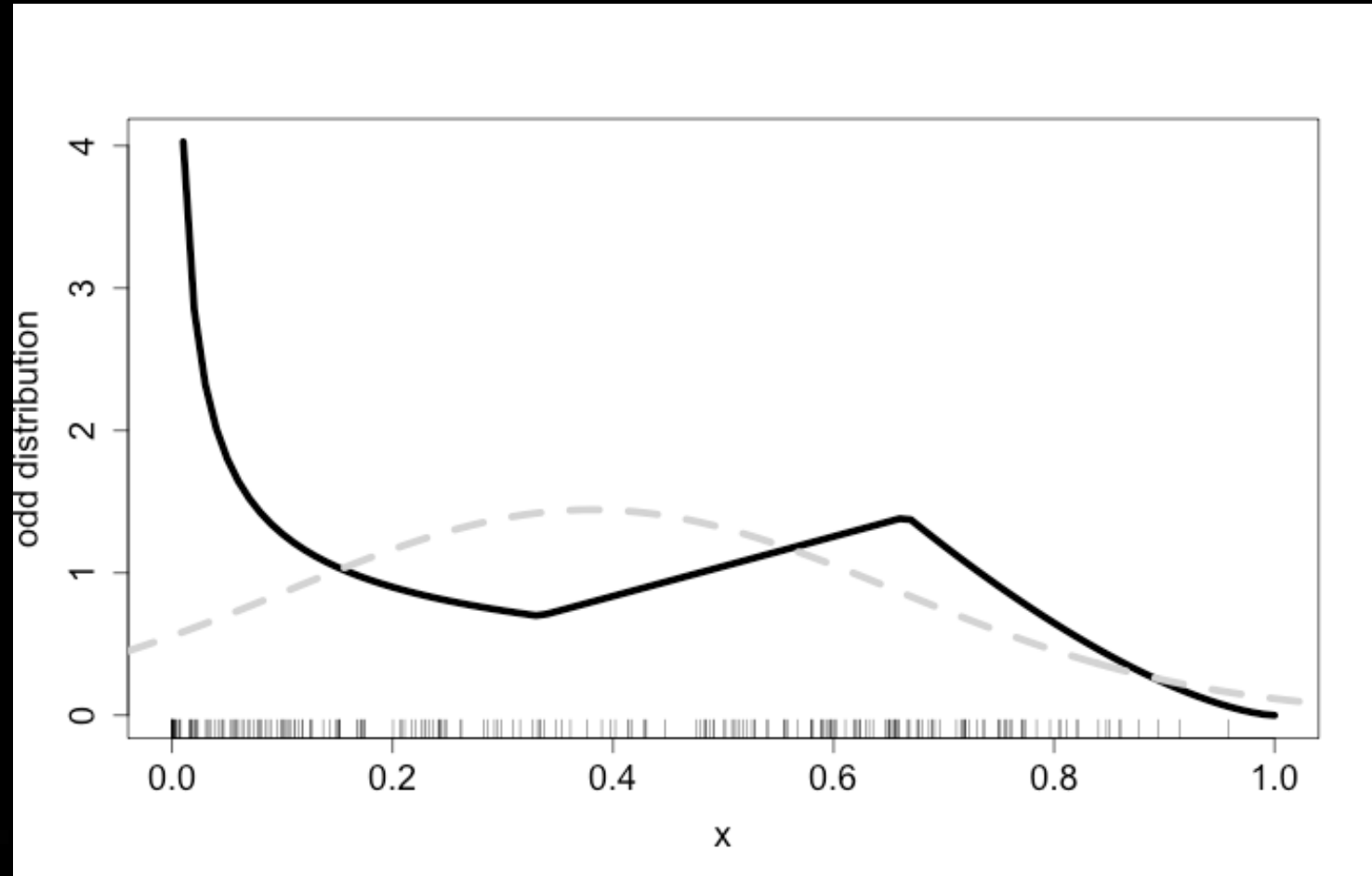
- My blood pressure today is more similar to my blood pressure tomorrow than it is to a random person

"Large" sample size

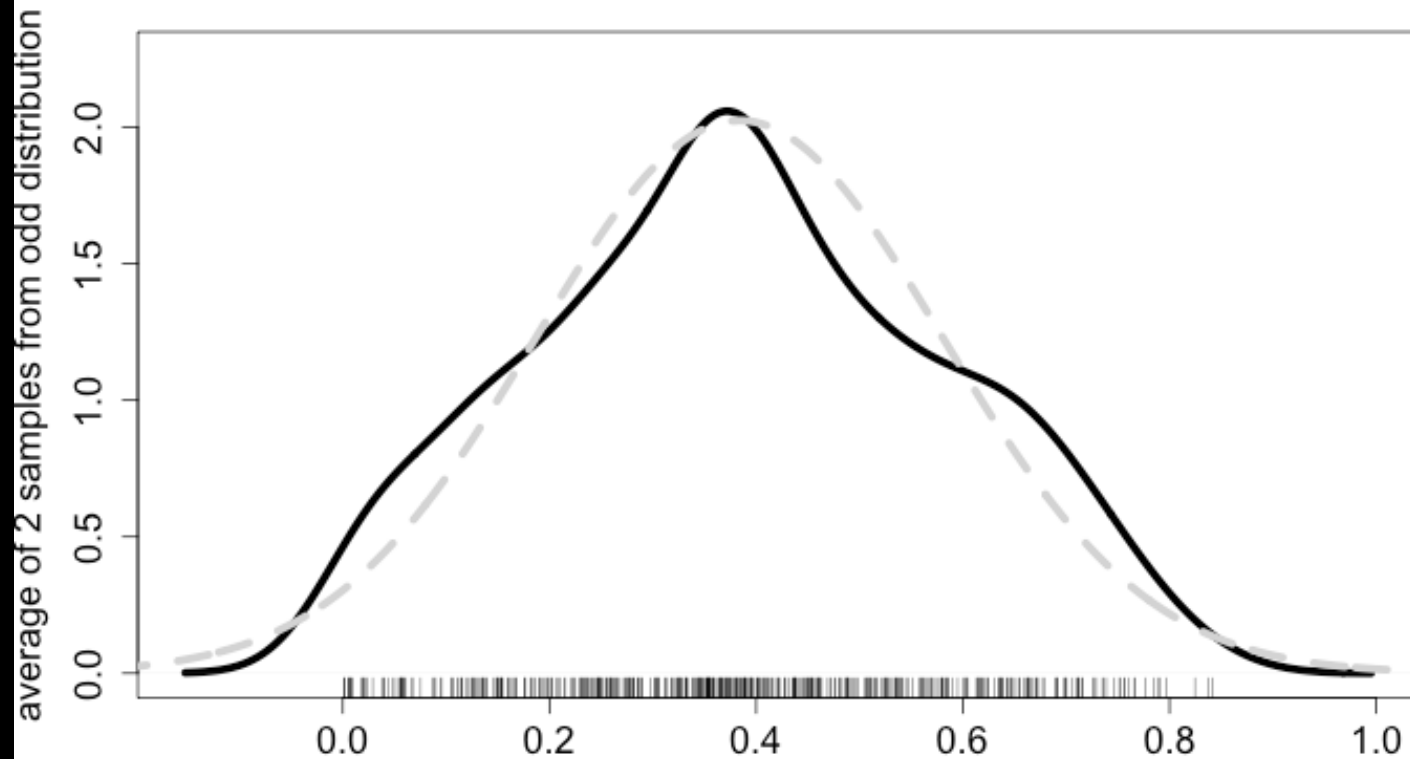
- The central limit theorem says that the *mean* of a "large" number of independent samples from *any shaped distribution* will be normally distributed!



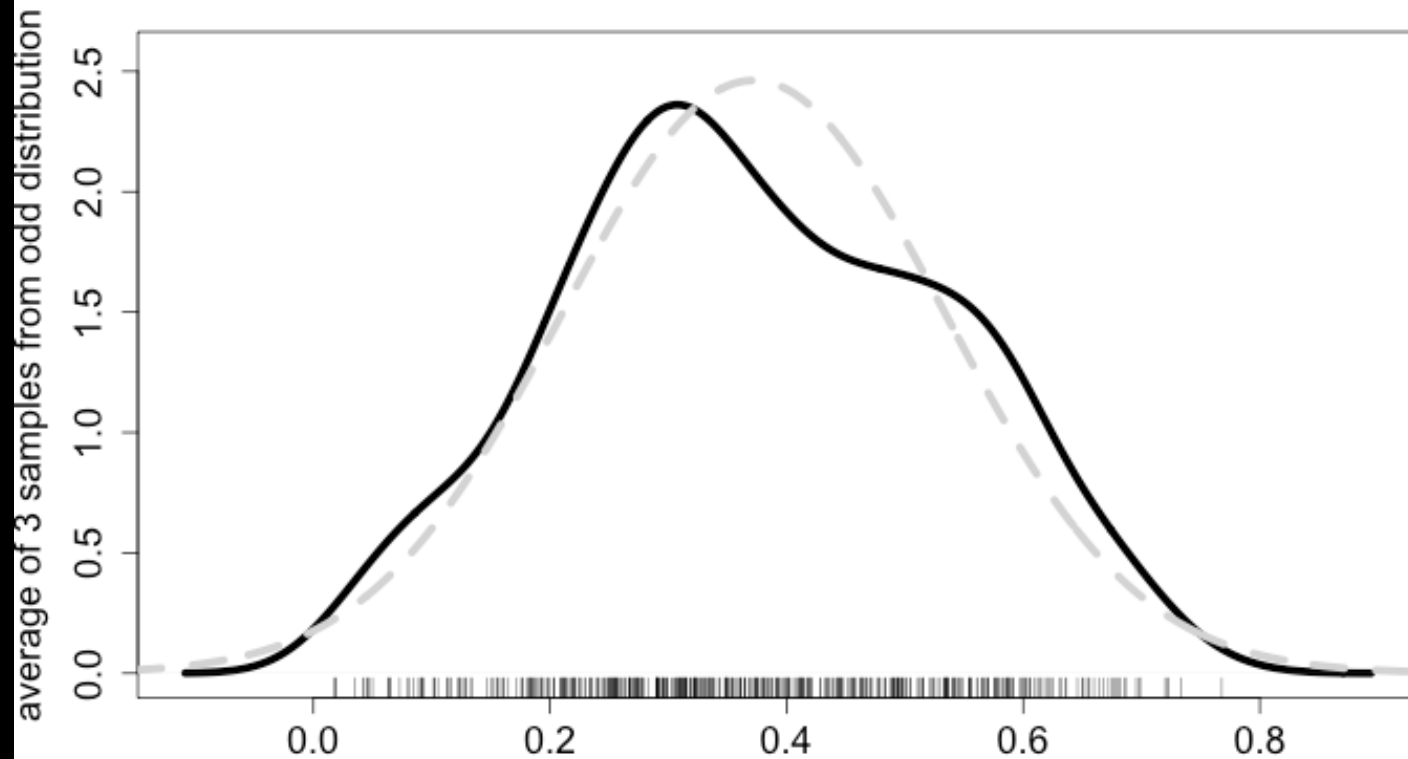
Odd distribution



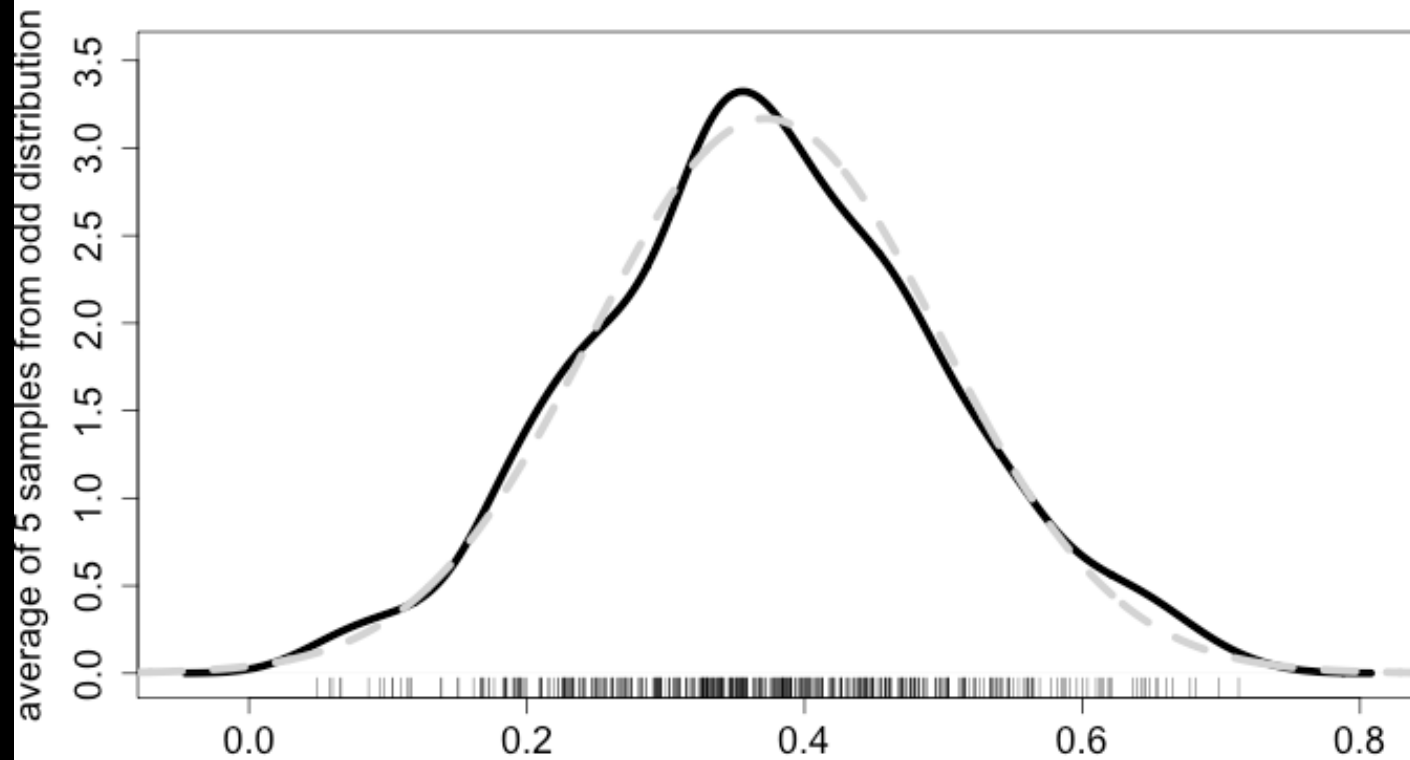
Averages of samples of size 2 from odd dist.



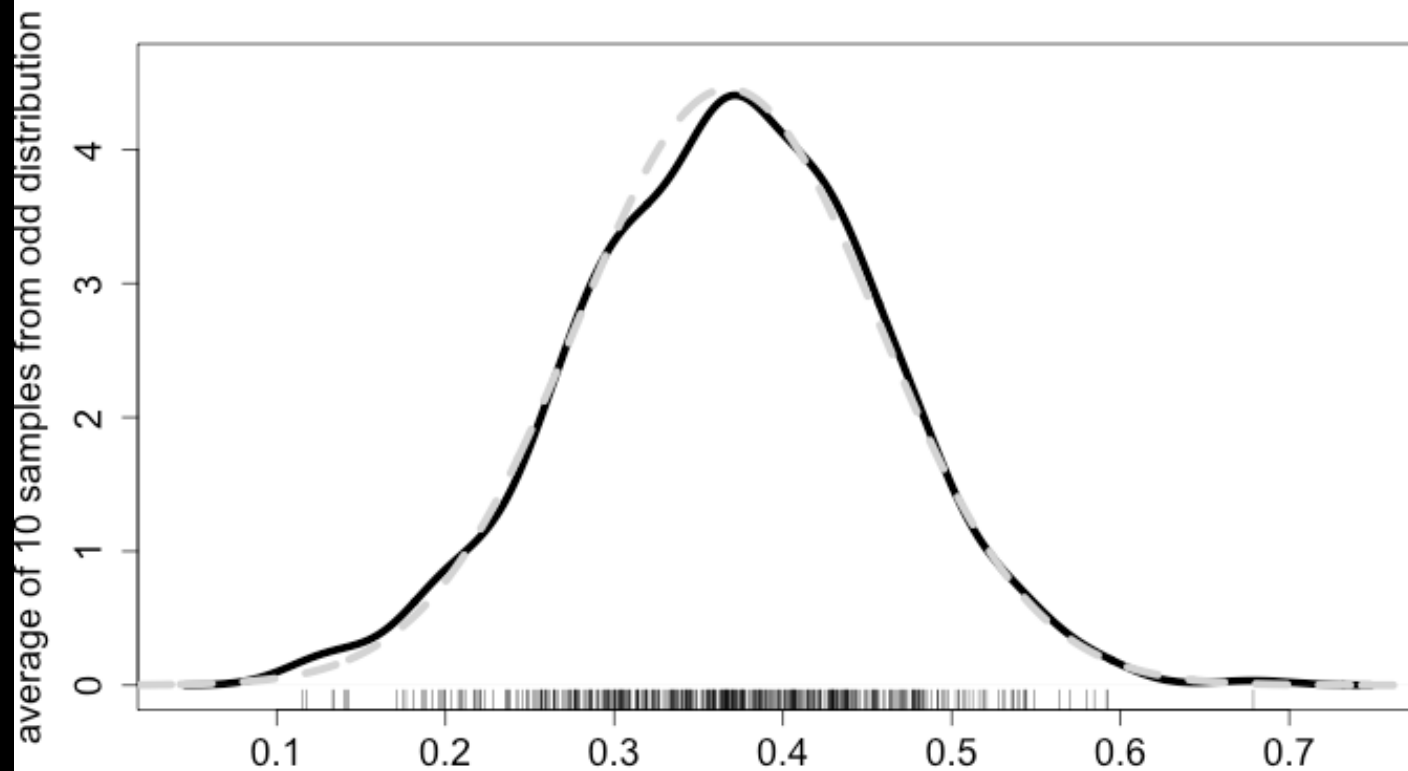
Averages of samples of size 3 from odd dist.



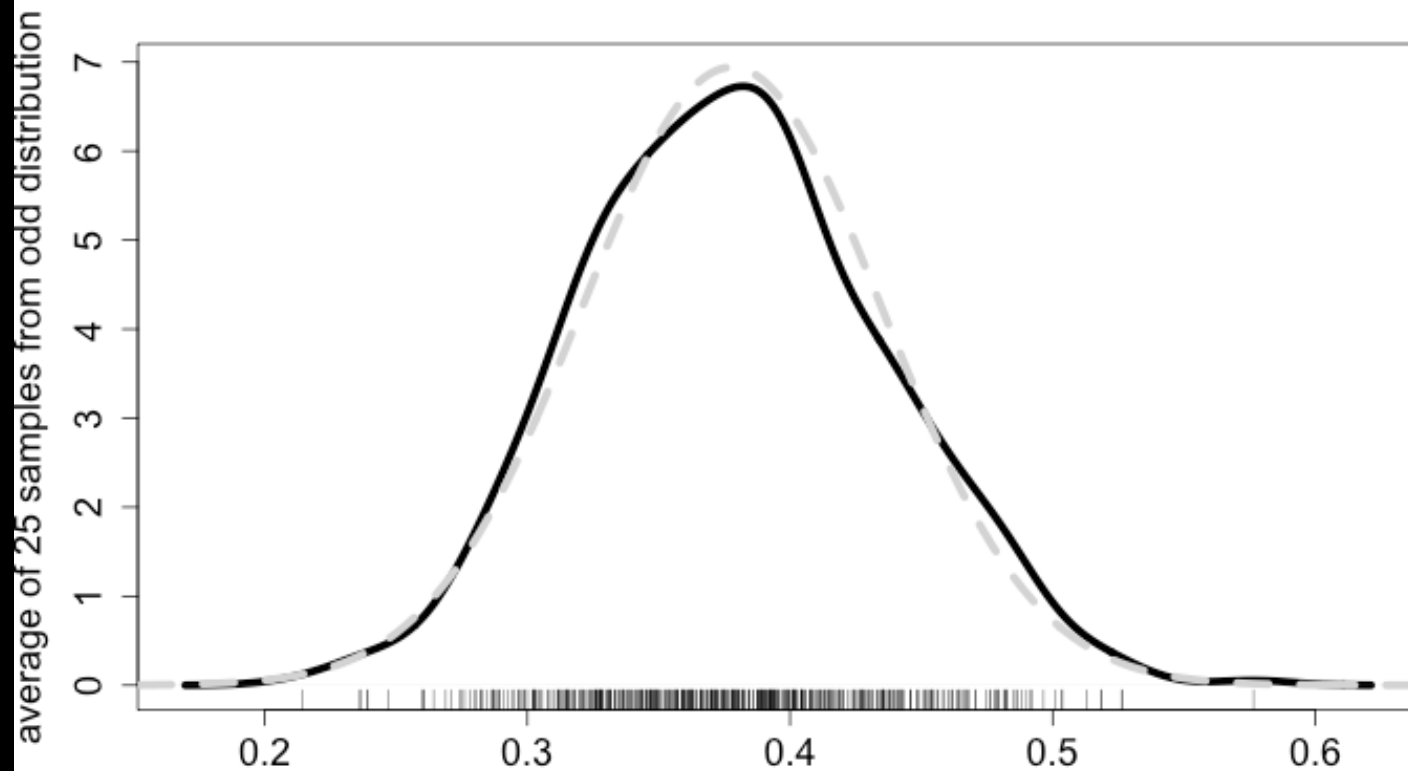
Averages of samples of size 5 from odd dist.



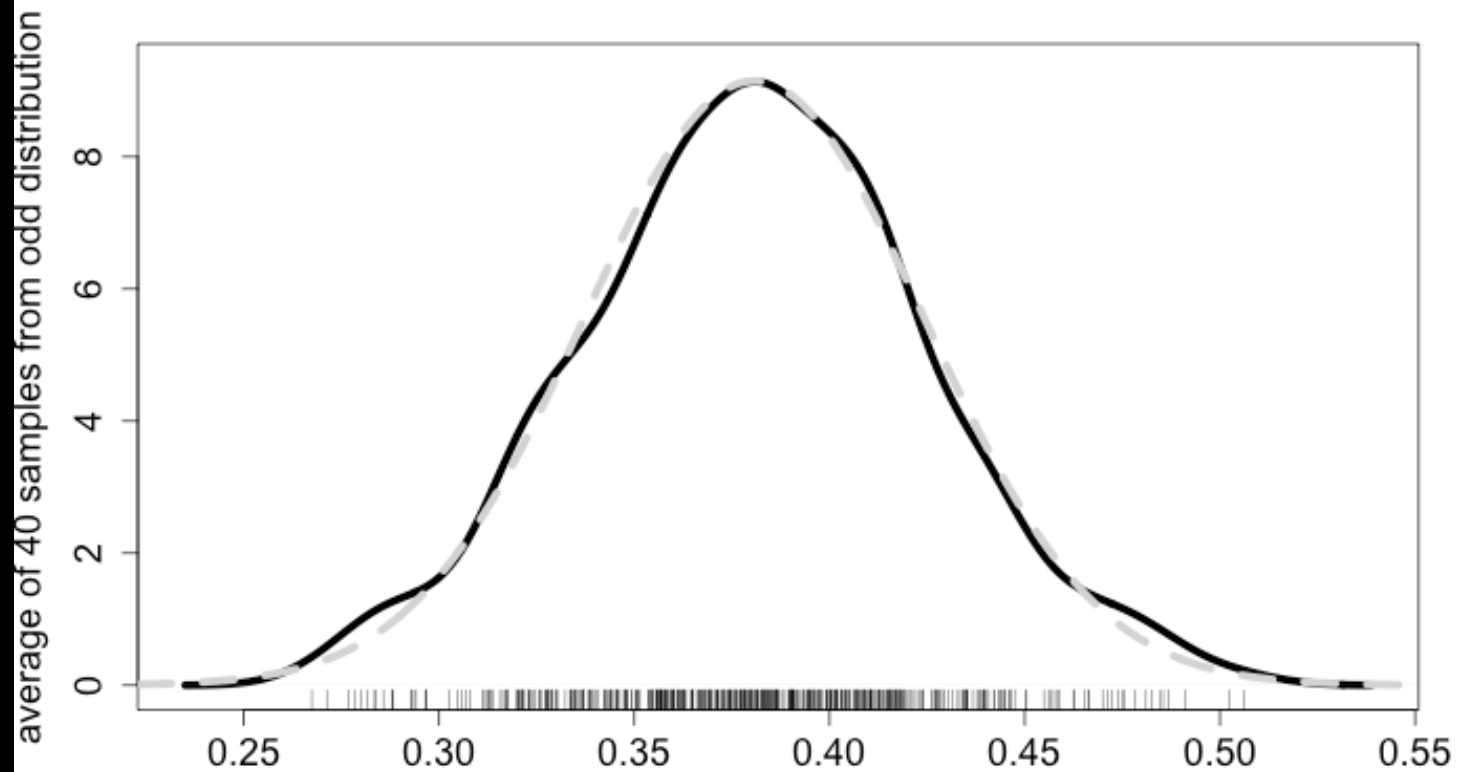
Averages of samples of size 10 from odd dist.



Averages of samples of size 25 from odd dist.



Averages of samples of size 40 from odd dist.



Two sample t-test

- Used to compare whether the means of two samples are different
- Assumptions:
 - Samples are *independent* of each other
 - Each sample is drawn from a *normally distributed population* or each sample is of a "large" size ($n=25$)

Conceptually

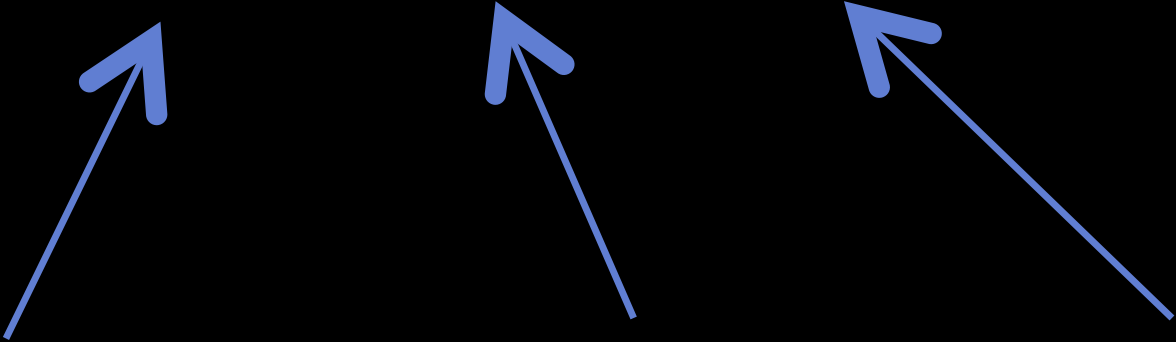
- Could the means in the two groups be randomly generated from a common normal distribution (null distribution)?
- What is the probability of observing the difference in the means we see or more extreme if they did (p-value)?

Mathematically

Assume Equal Variances	Assume Unequal Variances
$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{X_1 X_2} \cdot \sqrt{\frac{1}{n}}}$	$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{X_1 X_2} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$
$s_{X_1 X_2} = \sqrt{s_{X_1}^2 + s_{X_2}^2}$	$s_{X_1 X_2} = \sqrt{\frac{(n_1 - 1)s_{X_1}^2 + (n_2 - 1)s_{X_2}^2}{n_1 + n_2 - 2}}$

Two sample t-test in R

```
t.test(age ~ treat, iih_data)
```



Continuous
variable to
test

Grouping
variable

Dataset

Study Question

- Is the average (mean) age of our subjects different at baseline?
- Treated group mean age (n=22): 31.5
- Untreated group mean age (n=18): 28.0
- Two-sample t-test: 0.0655

Two sample t-test

- Used to compare whether the means of two samples are different
- Assumptions:
 - Samples are *independent* of each other
 - ~~Each sample is drawn from a normally distributed population or each sample is of a "large" size~~

What if cannot assume normal distribution?

- Wilcoxon rank-sum test, also called:
 - Mann-Whitney U test
 - Mann-Whitney-Wilcoxon test
 - Wilcoxon-Mann-Whitney test
- Assumptions:
 - Independent samples
 - At least ordinal

Wilcoxon rank-sum test

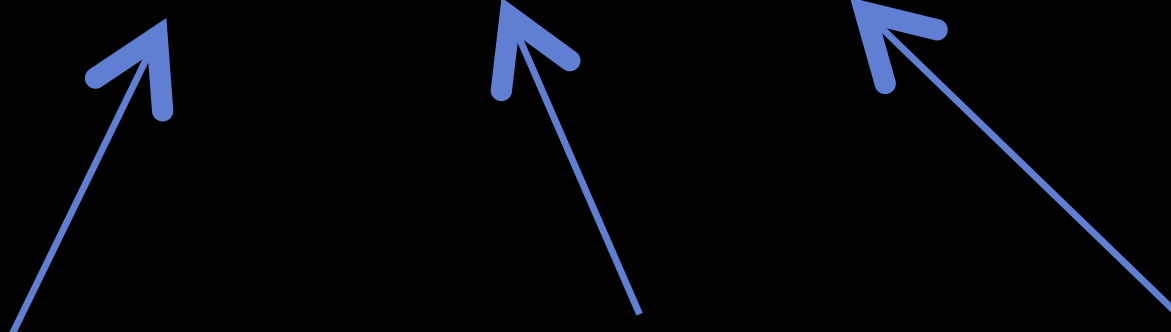
- If your variable is *continuous* (e.g., age), then you can interpret significant test as showing a difference in medians

Conceptually

- Rank order all the observations
- How often does one group “beat” the other group in the rankings?

Wilcoxon rank-sum test in R

```
library(coin) # run once per session  
wilcox_test(age ~ treat, iih_data)
```



Continuous
variable to
test

The diagram consists of three blue arrows pointing upwards from the labels below to the code above. The first arrow points from 'Continuous variable to test' to 'age'. The second arrow points from 'Grouping variable' to 'treat'. The third arrow points from 'Dataset' to 'iih_data'.

Grouping
variable

Dataset

Two sample t-test

- Used to compare whether the means of two samples are different
- Assumptions:
 - ~~Samples are independent of each other~~
 - Each sample is drawn from a *normally distributed population* or each sample is of a "large" size

What if cannot assume independence?

If you want to compare two measurements on the same subject (e.g., before vs. after), you need a *paired* t-test

Conceptually

- Accounts for the similarity within individuals
- More powerful (i.e., more likely to be significant if a difference exists)

Paired t-test in R

```
t.test(iih_data$pmd_6m,      # second  
       iih_data$pmd_bl,     # first  
       paired = TRUE)
```

Two sample t-test

- Used to compare whether the means of two samples are different
- Assumptions:
 - ~~Samples are independent of each other~~
 - ~~Each sample is drawn from a normally distributed population or each sample is of a "large" size~~

What if cannot assume normality or independence?

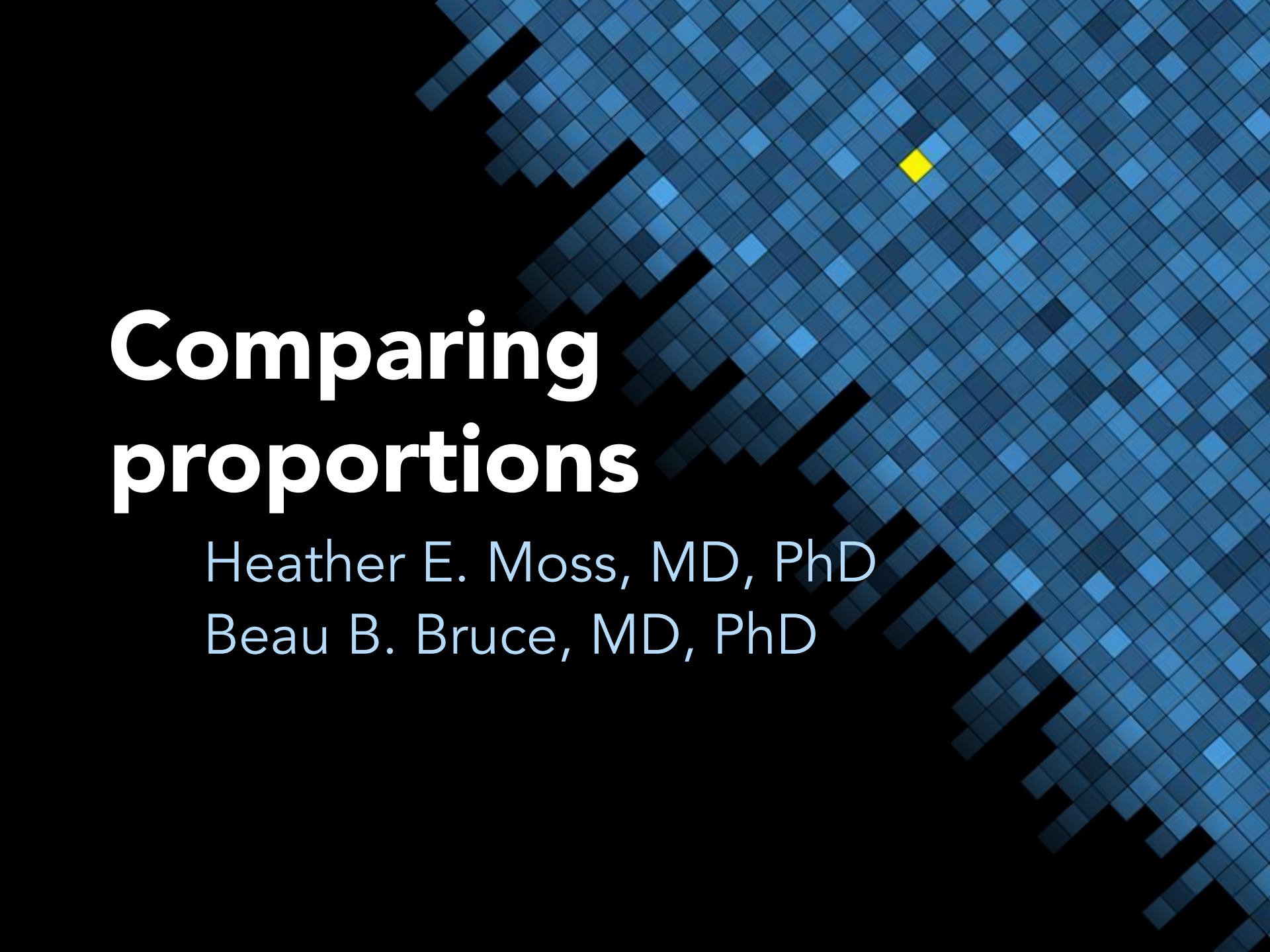
If you want to compare two measurements on the same subject (e.g., before vs. after), you need a *Wilcoxon signed rank test*

Conceptually

- For each pair was there a positive or negative change
- Are there more +’s (or –’s) than there should be based on chance?

Wilcoxon signed-rank test in R

```
wilcoxsign_test(  
  pap_6m ~ pap_bl,    # second, first  
  iih_data,           # dataset  
  zero.method = "Pratt") # what to  
                        # do with  
                        # zeros/ties
```

The background of the slide is a pixelated pattern of blue and black squares. A single yellow pixel is located in the upper right quadrant of the image.

Comparing proportions

Heather E. Moss, MD, PhD
Beau B. Bruce, MD, PhD

Study Questions

- Comparing 2 proportions:
 - Are proportions of women in treated and untreated groups different?
- Comparing proportion to a value:
 - Is proportion of women in our study different from 50%?

Binomial distribution

- Characterizes the count of events generated by flipping a (possibly biased) coin (heads/tails, disease/no disease, male/female) occurring with some probability

Binomial distribution

- Fair coin flip has probability of
 - 0.5 head
 - 0.5 tail
- If I flip a fair coin 2 times, probabilities are:
 - 0.5 1 head/1 tail
 - 0.25 2 heads
 - 0.25 2 tails

Binomial distribution

- Question: if I flip a fair coin 5 times what is the probability I get 4 heads?
- Clinical question: If the risk of disease is 10% in a population, what is the probability that if I sample 100 people I will have 15 with disease?

Exact binomial test

- Compares a observed count to an expected count based on the assumed probability
- If this coin is fair (probability 50%), what is the chance that I would have gotten this 4 heads (or more) in 5 flips?

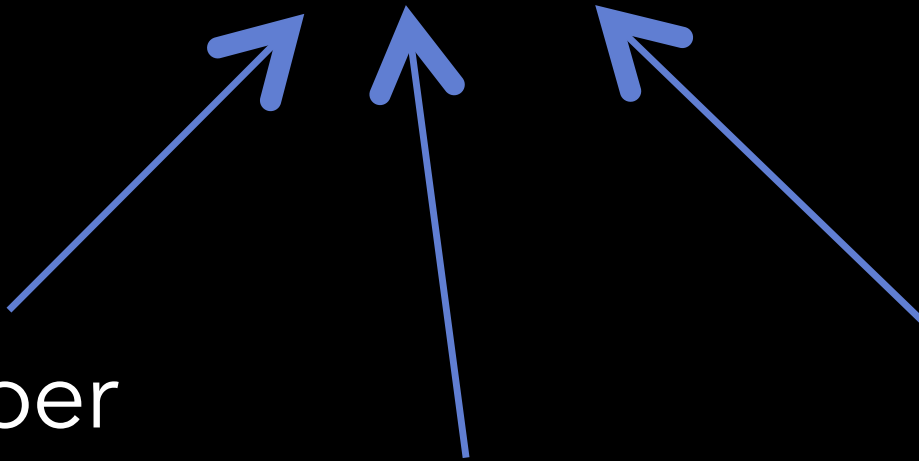
Exact binomial test in R

```
binom.test(4, 5, 0.5)
```

Number
observed

Number of
trials

Expected
probability



Exact binomial test in R

Question: If disease is equally common in men & women, what is the chance that we observed our sample characteristics?

```
table(iih_data$sex)
```

```
binom.test(table(iih_data$sex))
```

Exact binomial test in R

number of successes = 4, number of trials = 40,
observed probability = 0.1

Null hypothesis: true probability of success is 0.5

Alternative hypothesis: true probability of success
is not equal to 0.5

p-value = 1.857e-07

95 percent confidence interval:

0.02792542, 0.23663740

Exact binomial test results

p-value = $1.857\text{e-}07$

Probability is 0.0000001857 of observing 4/40 males given true probability of 0.5

Exact binomial test results

Observed proportion = 0.1

95 percent confidence interval:
0.02792542, 0.23663740

We are 95% confident that 0.02-0.24
contains the true proportion

Normal approximation of binomial

- Can use the central limit theorem to approximate the binomial with the normal when the sample size is “large”

```
binom.test(table(iih_data$sex))
```



```
prop.test(table(iih_data$sex))
```

Two proportions test (equivalent to 2 x 2 χ^2 test)

- Are the proportions of headache different in the treated and untreated groups?
- Here no need to specify an expected probability since you are comparing the two groups to each other
 - Calculate expected values

Two proportions test in R

- `table(iih_data$treat,
 iih_data$ha_bl)`
- `prop.test(table(iih_data$treat,
 iih_data$ha_bl))`
- `chisq.test(table(iih_data$treat,
 iih_data$ha_bl))`

Chi square (χ^2) test

- Used to compare the distribution of a categorical variable (with two or more categories) between two or more groups
- Assumptions:
 - Table axes are independent
 - Measurements on independent subjects
 - Large sample:
 - smallest expected cell value 5 or more

Expected value

Start with number observed

	Group 1	Group 2
A	10	20
B	15	15
C	35	5

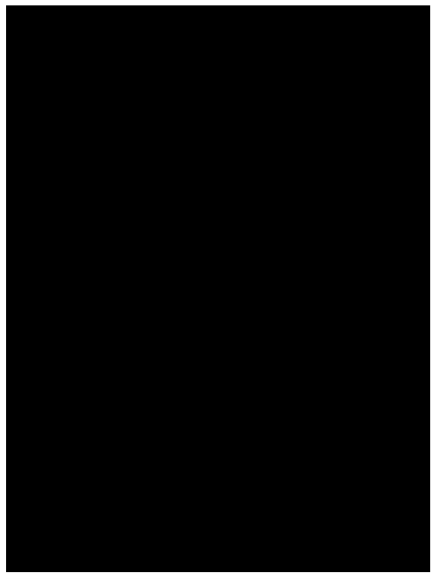
Expected value

Calculate the margins of the table

	Group 1	Group 2	
A	10	20	30 Total A
B	15	15	30 Total B
C	35	5	40 Total C
	60	40	100

Expected value

Erase the inside

	Group 1	Group 2	
A			30
B			30
C			40
	60	40	100

Expected value

Given the margins what would you expect there in each cell?

A	18	30	
B		30	
C		40	
	60	40	100

$$\frac{30 \times 60}{100} = 18$$

Expected value

Given the margins what would you expect there in each cell?

A	18	12	30
B	18	12	30
C	24	16	40
	60	40	100

Conceptually

- Compare the values you actually observed to the values you would expect given column/row totals
- What is the chance that the observed counts occurred due to random error?

Compare observed with expected

Observed

Group 1 Group 2

A	10	20
B	15	15
C	35	5

Expected

Group 1 Group 2

A	18	12	30
B	18	12	30
C	24	16	40
	60	40	100

Example question

- Does the distribution of race (white, black, other) differ in the two treatment categories?

Chi square (χ^2) test in R

- `table(iih_data$race,
 iih_data$treat)`
- `chisq.test(table(iih_data$race,
 iih_data$treat))`
- P value is probability of observing observed proportions or more extreme if actual proportions are not different between groups

Fisher exact test

- Used when the large sample assumption of the χ^2 test is not met
- No need to worry about how that is determined as R will warn you

Fisher exact test in R

- `prop.test(table(iih_data$sex,
 iih_data$treat))`
- `fisher.test(table(iih_data$sex,
 iih_data$treat))`
- `fisher.test(table(iih_data$race,
 iih_data$treat))`

McNemar test

- Used when the axes of the table are *not independent*, but rather repeat / longitudinal measurements of the same thing on independent subjects
- Example: did the proportion of people reporting headache at baseline change at 6 months?

Conceptually

- For each subject
 - $HA \rightarrow HA$ no change
 - $No\ HA \rightarrow no\ HA$ no change
 - $HA \rightarrow no\ HA$ improved
 - $No\ HA \rightarrow HA$ worse
- Only the people who change provide information about this question

Conceptually

- Comparing the number who changed “for the better” to those who changed “for the worse”
- Similar to comparing heads versus tails in “coin flip” analogy
 - (probability of 0.5)

McNemar test in R

- `table(iih_data$ha_bl,
 iih_data$ha_6m)`
- `mcnemar.test(table(iih_data$ha_bl,
 iih_data$ha_6m))`

Where to now?

Beau B. Bruce, MD, PhD

Assistant Professor of Ophthalmology,
Neurology, and Epidemiology
Emory University

Learning more

- Read a basic tutorial
 - Kickstarting R
<http://cran.r-project.org/doc/contrib/Lemon-kickstart/index.html>
 - R for Beginners
https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
 - Or another tutorial (in 19 languages)
<https://cran.r-project.org/other-docs.html>

Learning more

- Take a MOOC (massive open online course)
 - Coursera – Johns Hopkins University – Data Science Specialization
<https://www.coursera.org/specializations/jhu-data-science>

Learning more

- Get some slightly more advanced resources
 - Analysis of Epidemiological Data using R and Epicalc
http://cran.r-project.org/doc/contrib/Epicalc_Book.pdf
 - A Little Book of R for Biomedical Statistics
<http://a-little-book-of-r-for-biomedical-statistics.readthedocs.org/en/latest/>

How to get help

- Try Google, but <http://www.rseek.org/> usually works better
- Within R
 - Use ?<command> to get help for a specific command
 - Use help.search("...") or RSiteSearch("...") to find something you do not know

How to get help

- Ask someone you know who uses R
- Search and then ask if no one has asked before on StackExchange
<http://stackexchange.com/>
- Ask me (but please try the others first!)

Getting Started

"You can get help from teachers, but you are going to have to learn a lot by yourself, sitting alone in a room."

- Dr. Seuss
"On Becoming a Writer"
The New York Times
May 21, 1986



Experienced programmers

- Even experienced programmers not infrequently want to:
 - Pull their hair out
 - Throw the computer out the window
 - Take a sledgehammer to the monitor
- Take a deep breath, a long break, or come back to it another day

Surgeon General's Warning

"Using R is a bit akin to smoking.

Beginnings are difficult, one may get headaches, and even gag on the first experiences. But in the long run, it becomes pleasurable, and even addictive.

Yet, deep down, for those willing to be honest, there is something not fully healthy in it."

- François Pinard
Aug 20, 2007 on R-help



Photo by Simon Law