Lecture 1.10 – Troubleshooting Problems

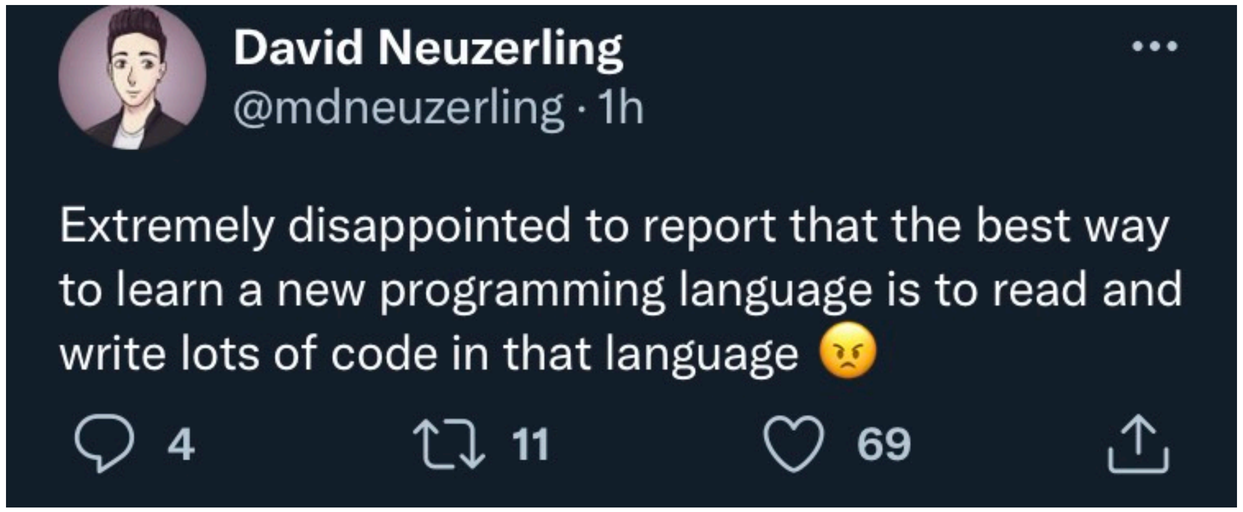
Specific Learning Objectives:

3.5.1 – Learn basic skills in debugging and troubleshooting error messages.

What happens when you get an error?

- Errors are quite normal. No one is perfect (far from it!), and mistakes slip through all the time.

- Learning a language (spoken or technical) involves making a lot of errors! You have to be comfortable being bad at something in order to practice and learn it.



David Neuzerling
@mdneuzerling · 1h

Extremely disappointed to report that the best way to learn a new programming language is to read and write lots of code in that language 😠

4 11 69

- Troubleshooting errors can be challenging and frustrating! But there is a solution.

- Today, we'll cover some strategies for tackling errors.

Strategies for Troubleshooting Code

- First, take a deep breath and remember you can fix it. And if you can't figure it out, someone can help you.

- Follow the simple steps below:

  1. Did you spell it correctly? *Are you sure?*
  2. Is the capitalization/punctuation correct?
  3. Is the syntax correct?
  4. Is the object in your environment?

1. Did you spell it correctly?

- About 75% of students who come for help with an error in the first half of this course have spelled something incorrectly.

- Remember that object and function names must be *spelled exactly correct*. There is no autocorrect, you have to get it right! (But there is actually autofill, which helps!)

Dave Hemprich-Bennett @ha... · 19h ···
Every day when I finish debugging

- What do errors look like when you spell things incorrectly? Example: `Loblolly` data set.

```
> View(Loblollly)
Error in View : object 'Loblollly' not found
> View(Lobolly)
Error in View : object 'Lobolly' not found
> View(Bloblolly)
Error in View : object 'Bloblolly' not found
```

R is looking for *exactly* what you tell it to look for!

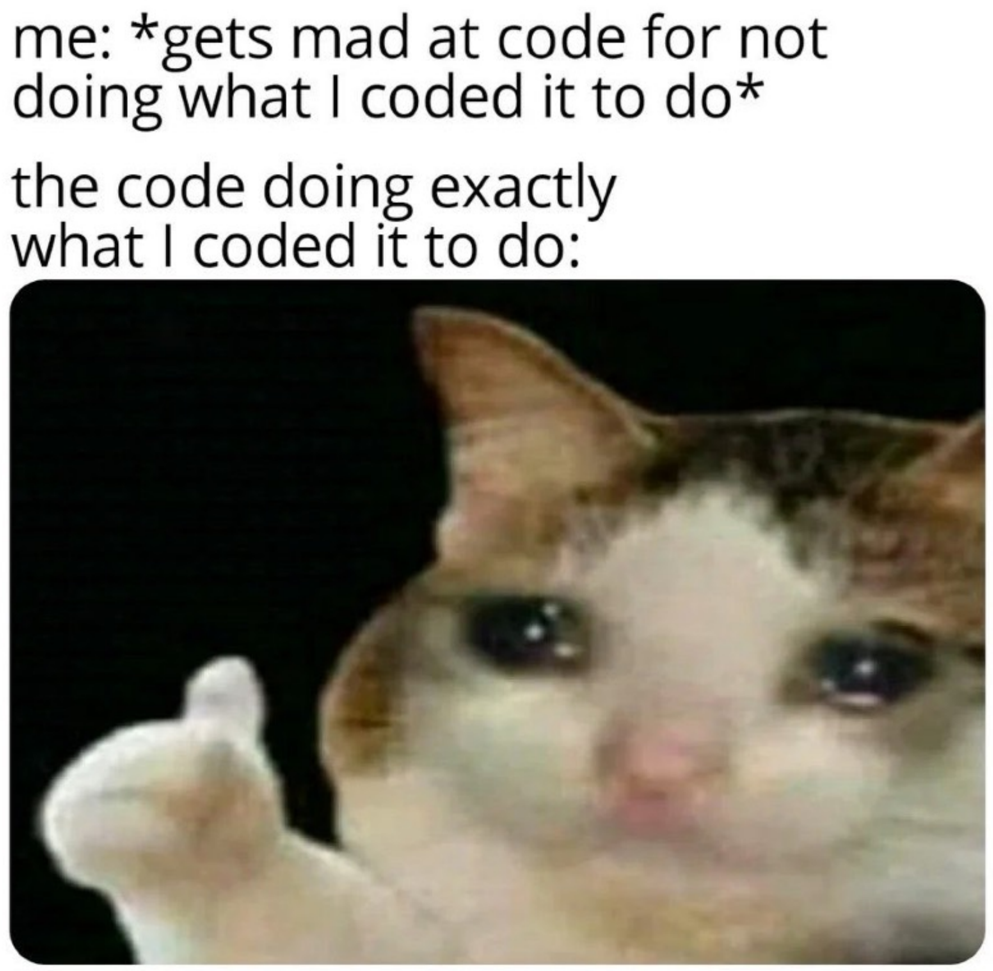
Please check your spelling before panicking about errors!

2. Is the capitalization/punctuation correct?

- If things are spelled correctly, make sure the punctuation and capitalization is correct.

- Remember, *R* is case-sensitive (like many passwords): Loblolly is different than loblolly.

- Similarly, punctuation has to be correct as well. `dat.model` is not the same as `dat_model`.

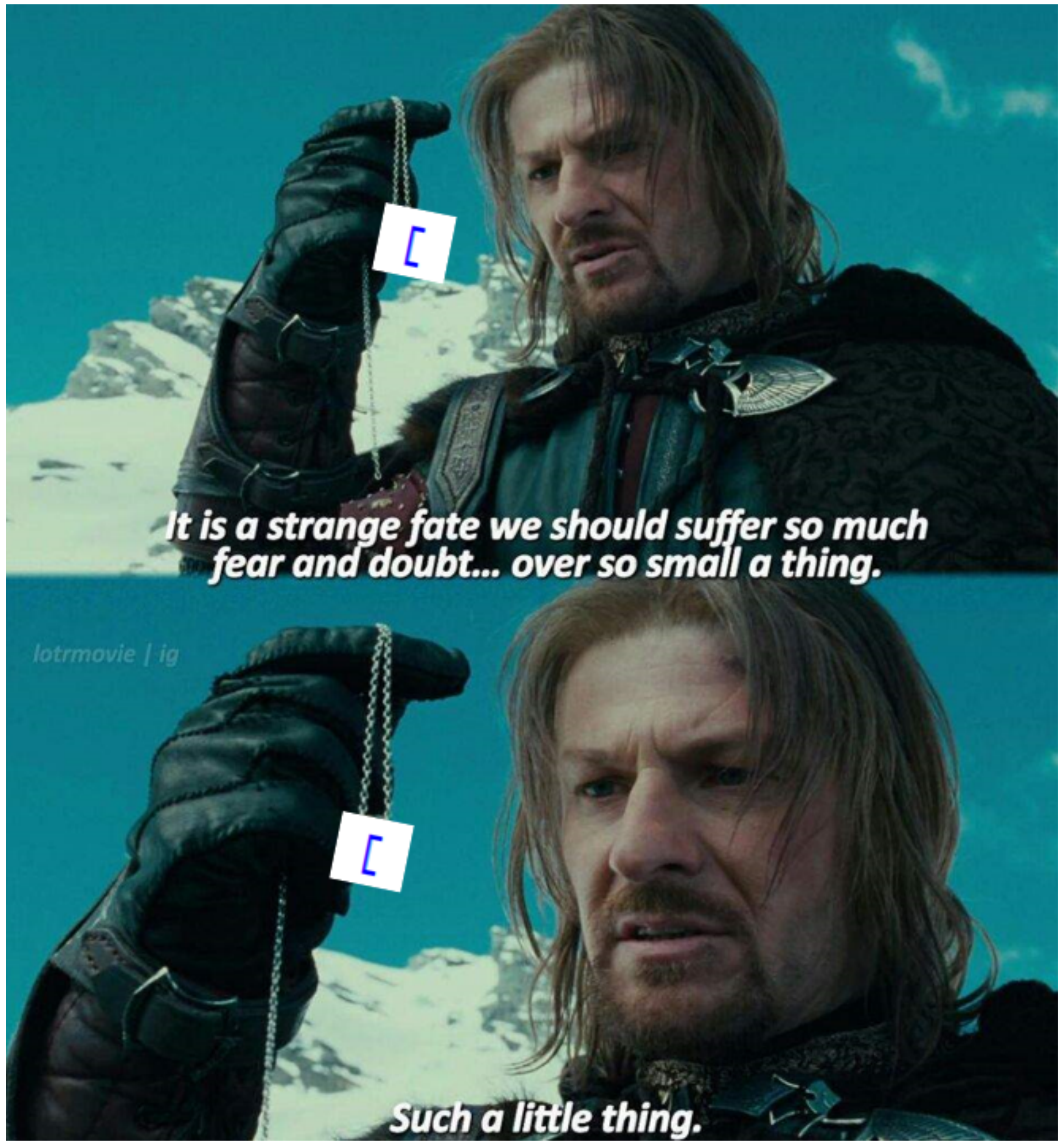

me: *gets mad at code for not doing what I coded it to do*
the code doing exactly what I coded it to do:

- Errors will look the same as misspellings: *R* will tell you it can't find an object!

Please check your capitalization before panicking about errors!

3. Is the syntax correct?

- Syntax errors are also common for both beginning and experienced programmers! (They can be harder for beginners to find.)
- Remember, syntax is incredibly important and the most difficult part of learning a language.
- What will errors look like? They range from very straightforward to very unclear.


[
It is a strange fate we should suffer so much fear and doubt... over so small a thing.
lotrmovie | ig
[
Such a little thing.

Unmatched square bracket

```
> d[ <- 3
Error: unexpected assignment in "d[ <-"
```

Unmatched parenthesis

```
> d <- c(3, 5, 5
+
```

(R is waiting for you to close this!!)

Can't use minus sign for object name!

```
> pop-v <- 4
Error in pop - v <- 4 : object 'pop' not found
```

Can't start an object name with a number!

```
> 3flips <- 3
Error: unexpected symbol in "3flips"
```

Syntax tips

- Object naming rules:

  - Object names must begin with a letter.

    Correct: `flip3`, `flip.3`, `Flip_3`

    Incorrect: `3flip`, `_flip3`, `=flip3`

  - Object names must not contain special characters or spaces (stick with . and _).

    Correct: `flip3`, `flip.3`, `Flip_3`

    Incorrect: `flip-3`, `flip#3`, `Flips@3`

  - Avoid renaming already existing objects

    Correct: `dat`, `t`, `name1`

    Incorrect: `data`, `T`, `names`

Syntax tips

- Avoid problems with brackets and parentheses by using whitespace! Making code more human-readable will help you find errors faster.

    - R is not sensitive to whitespace, so use spaces and tabs!

```
#Bad
df<-data.frame(x=c(9,2,54,1,39,99,29,40,80,2,68,3,34),y=c(T,F,T,T,T,F,T,F,F,T,F,F,T))
```

Is the y value that corresponds to x = 39 True or False?

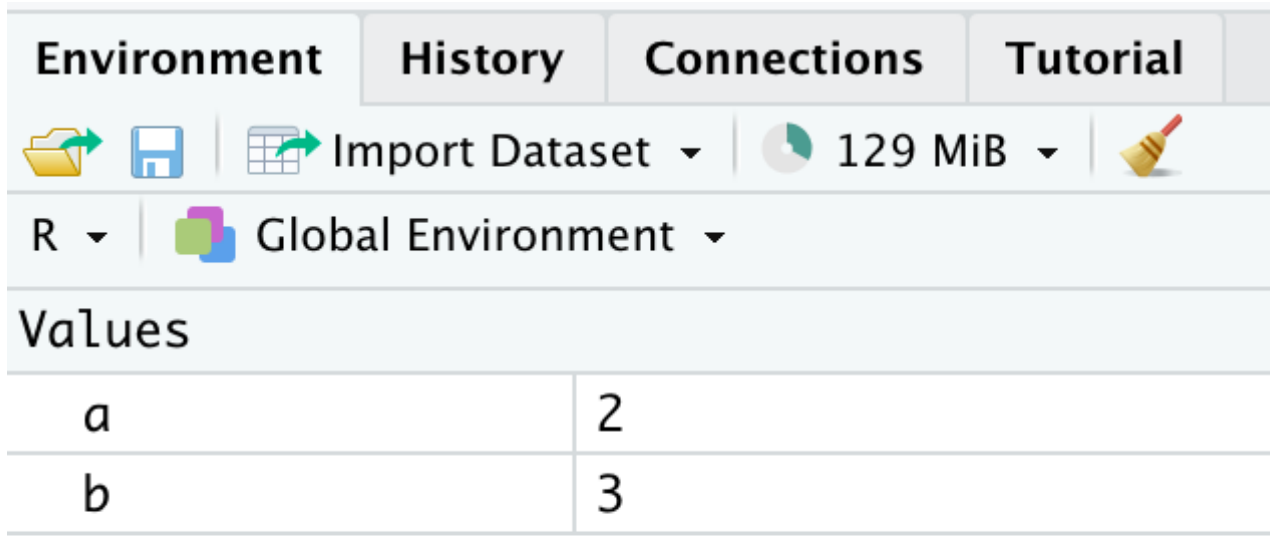
```
# Good
df <- data.frame(x = c(9, 2, 54, 1, 39, 99, 29, 40, 80, 2, 68, 3),
                 y = c(T, F,  T, T,  T,  F,  T,  F,  F, T,  F, F))
```

TRUE!

4. Is the object in your environment?

- The environment lists all objects in R's memory.

- It will be empty every time R starts or restarts.



Values

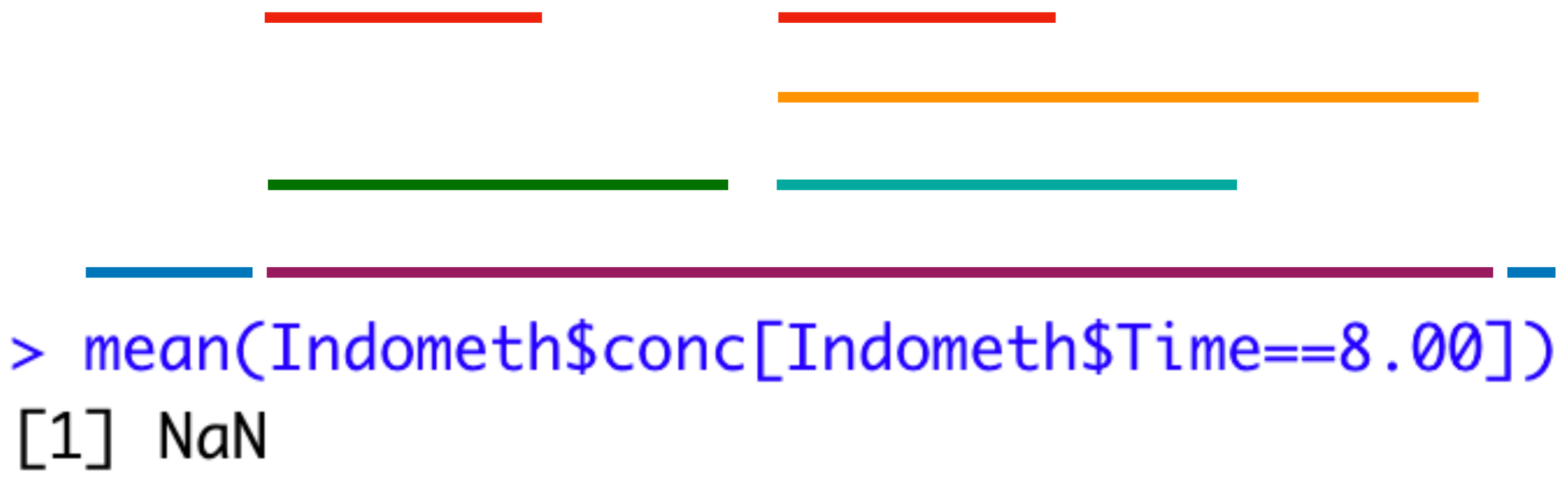
a	2
b	3

- Sometimes, you'll assume objects are in the environment and they won't be there (or they won't be the same). This will result in an error where there was not before!

- Tips for avoiding this:
  - Write all your code in scripts, in the order in which each line should be run.
  - Check the environment in RStudio to make sure it's there.
  - Frequently restart R or clear your environment to make sure your script runs cleanly!

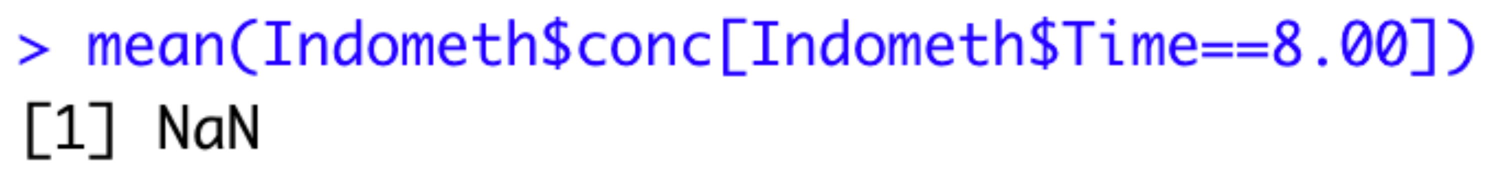
Troubleshooting Strategy: Splitting

- What do you do if you don't know which part of the code is the problem?
- Try splitting the line into the smallest elements. Run each independently to try and pinpoint the problem.
- Example: many ways to split the statement.

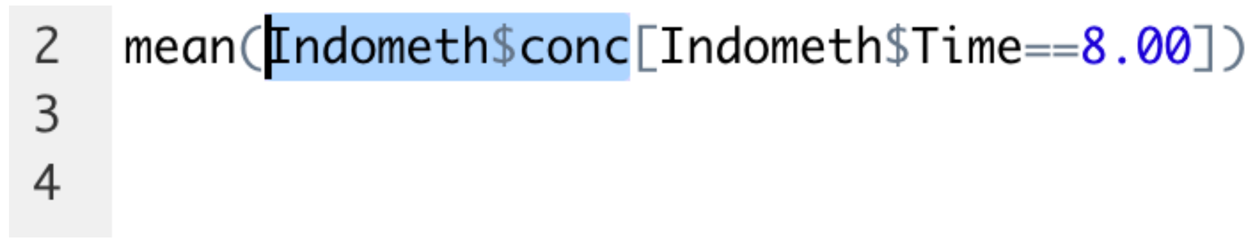


```
> mean(Indometh$conc[Indometh$Time==8.00])
[1] NaN
```

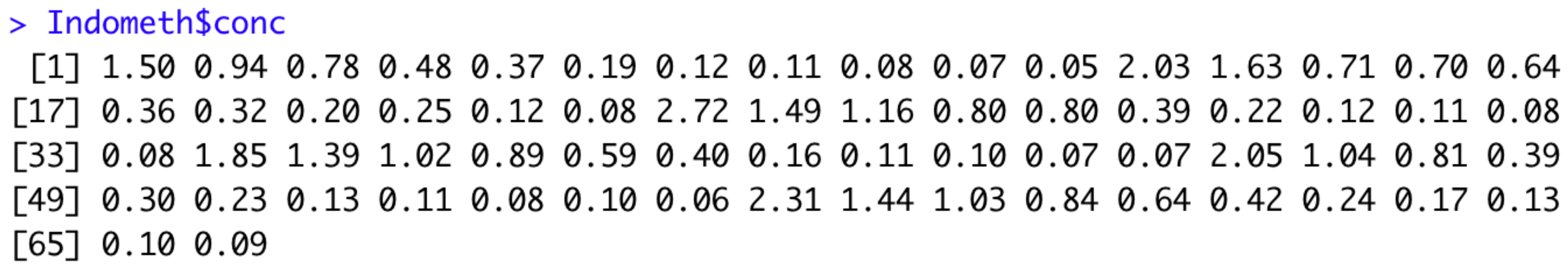
Troubleshooting Strategy: Splitting



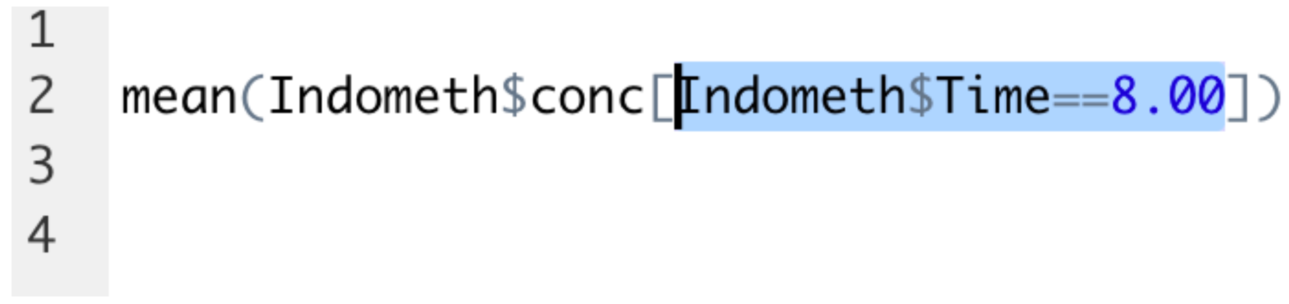
```
> mean(Indometh$conc[Indometh$Time==8.00])
[1] NaN
```



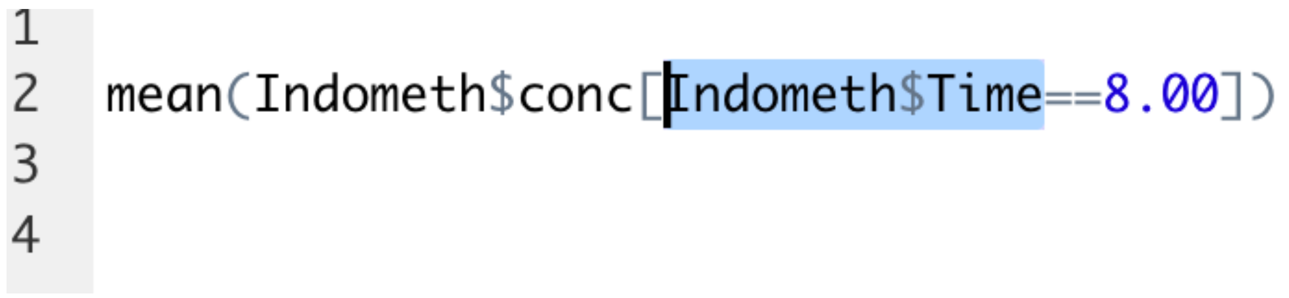
```
mean(Indometh$conc[Indometh$Time==8.00])
```



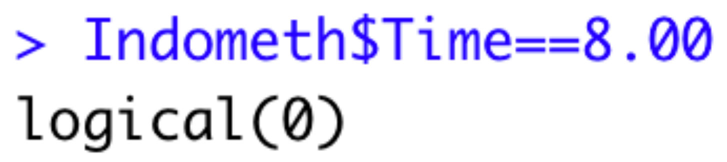
```
> Indometh$conc
 [1] 1.50 0.94 0.78 0.48 0.37 0.19 0.12 0.11 0.08 0.07 0.05 2.03 1.63 0.71 0.70 0.64
[17] 0.36 0.32 0.20 0.25 0.12 0.08 2.72 1.49 1.16 0.80 0.80 0.39 0.22 0.12 0.11 0.08
[33] 0.08 1.85 1.39 1.02 0.89 0.59 0.40 0.16 0.11 0.10 0.07 0.07 2.05 1.04 0.81 0.39
[49] 0.30 0.23 0.13 0.11 0.08 0.10 0.06 2.31 1.44 1.03 0.84 0.64 0.42 0.24 0.17 0.13
[65] 0.10 0.09
```



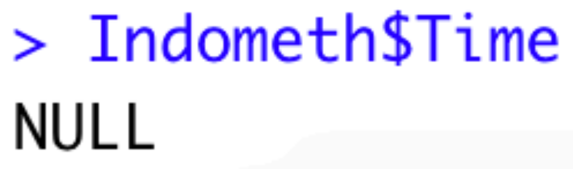
```
mean(Indometh$conc[Indometh$Time==8.00])
```



```
mean(Indometh$conc[Indometh$Time==8.00])
```



```
> Indometh$Time==8.00
logical(0)
```

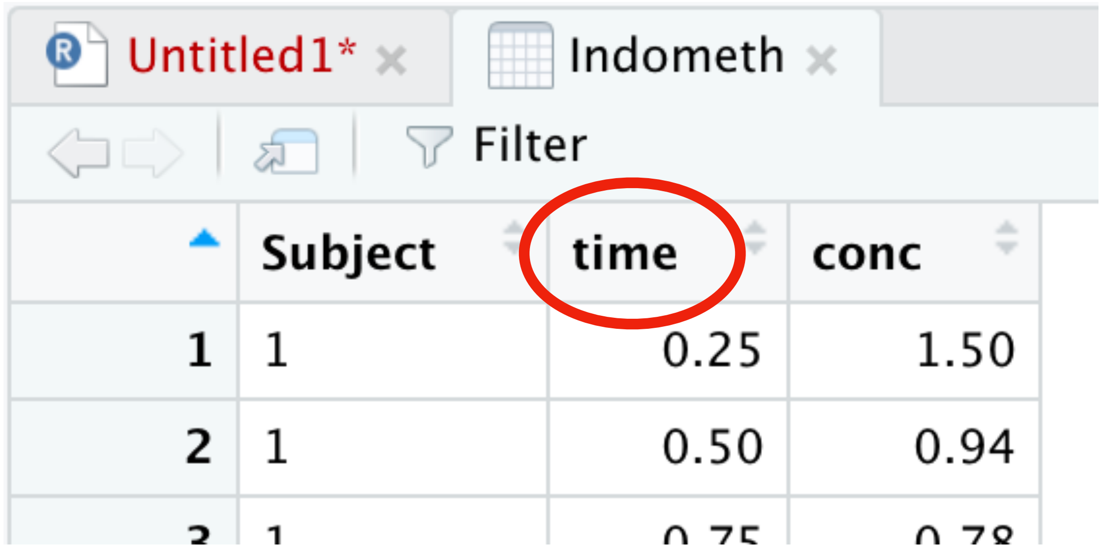


```
> Indometh$Time
NULL
```

Troubleshooting Strategy: Splitting

```
> mean(Indometh$conc[Indometh$Time==8.00])
[1] NaN
```

```
> View(Indometh)
```



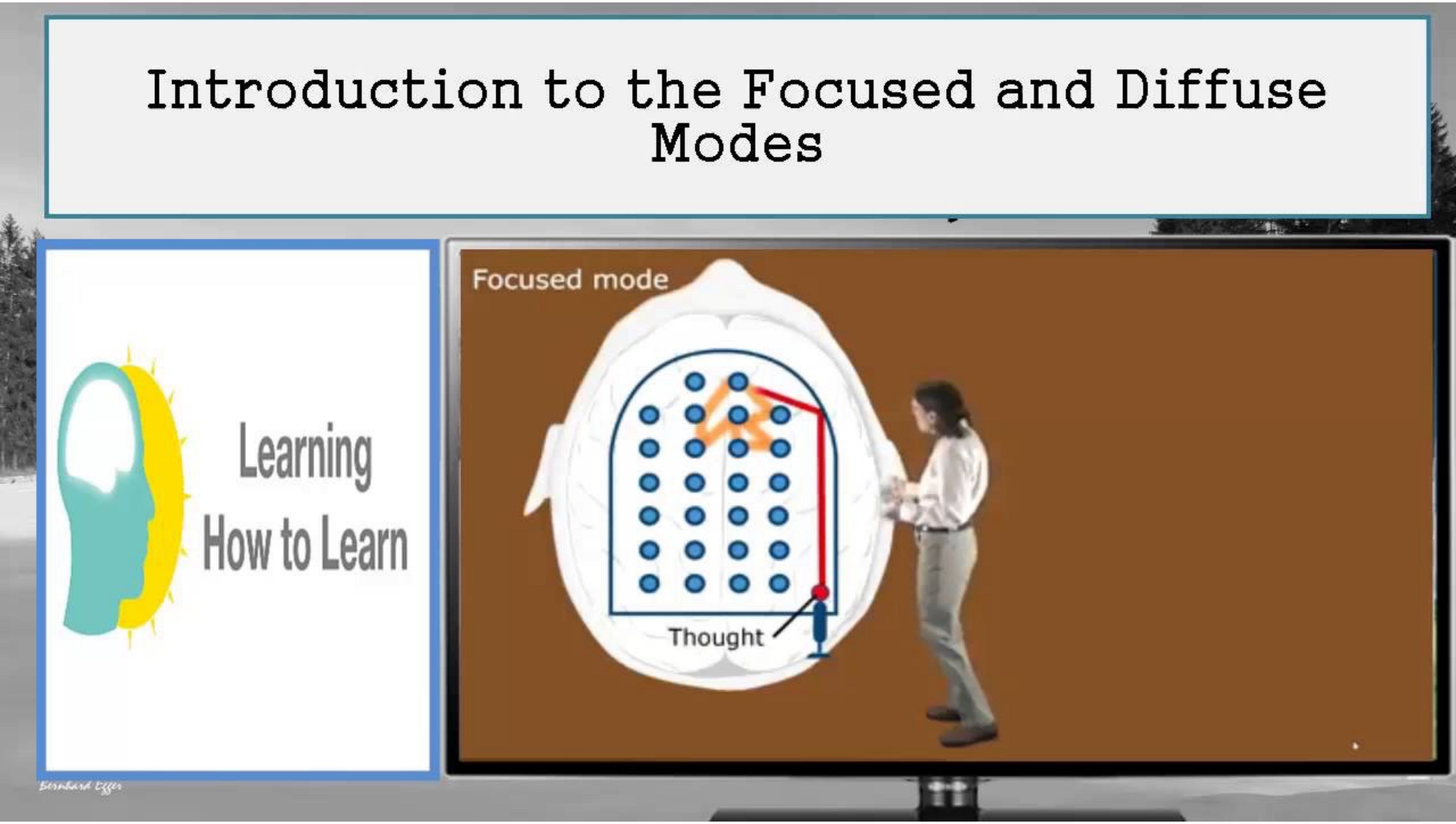
Untitled1* Indometh Filter

	Subject	time	conc
1	1	0.25	1.50
2	1	0.50	0.94

Fix it:

```
> mean(Indometh$conc[Indometh$time==8.00])
[1] 0.07166667
```

If you are stuck, try taking a break!



https://www.youtube.com/watch?v=lJtUg-3DfUk

In Summary

- Errors are quite normal for both beginning and advanced programmers. Practice will help, there are no shortcuts!

- First rule: **don't panic!** You'll be able to figure it out.

- Follow the four debugging steps:
  1. Did you spell it correctly?
  2. Is the capitalization/punctuation correct?
  3. Is the syntax correct?
  4. Is the object in your environment?

- Don't forget to **split** the problem.

- **Don't be afraid to ask your peers or instructors for help!!**

Action Items

1. Complete Assignments 1.14.

2. Prepare for your first Skill Check!