Lecture 20: Oct 22, 2018

Unstructured Data

- *Unstructured Data*
- *Text Representations*
- *String Operations*
  - *Length, Case, Concatenation, Substring, Split String*

James Balamuta
STAT 385 @ UIUC

Announcements

- **hw07** is due **Friday, Oct 26th, 2018** at **6:00 PM**
- **Office Hour Changes**
  - **John Lee's** are now from **4 - 5 PM** on **WF**
  - **Hassan Kamil's** are now from **2:30 - 3:30 PM** on **TR**
- **Quiz 08** covers Week 7 contents @ CBTF.
  - Window: Oct 16th - 18th
  - Sign up: https://cbtf.engr.illinois.edu/sched
- Want to review your homework or quiz grades?
  Schedule an appointment.

Lecture Objectives

- **Manipulate** unstructured data
- **Understand** where unstructured data is found
- **Differentiate** between character values

Unstructured Data

Previously

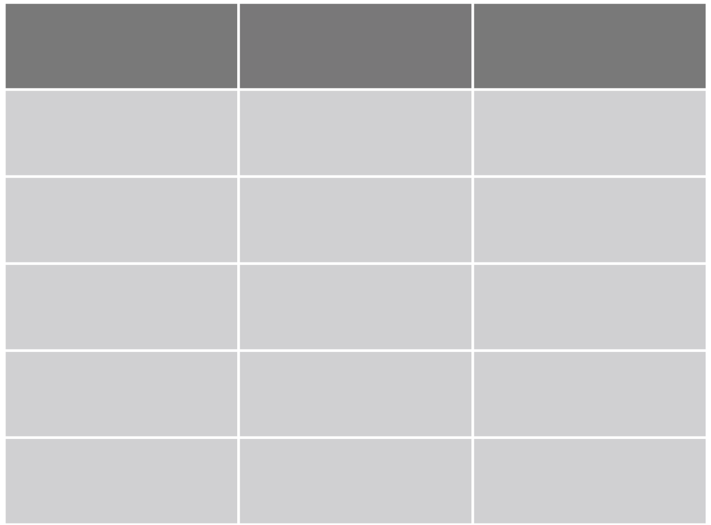
Structures of Data

… how data is shaped …

Structured*

Rectangular

~5 - 10%



Semistructured**

key: value

~5 - 10%

```
---
title: "Untitled"
author: "JJB"
date: "1/27/2018"
output: html_document
---
```

Unstructured***

?????????

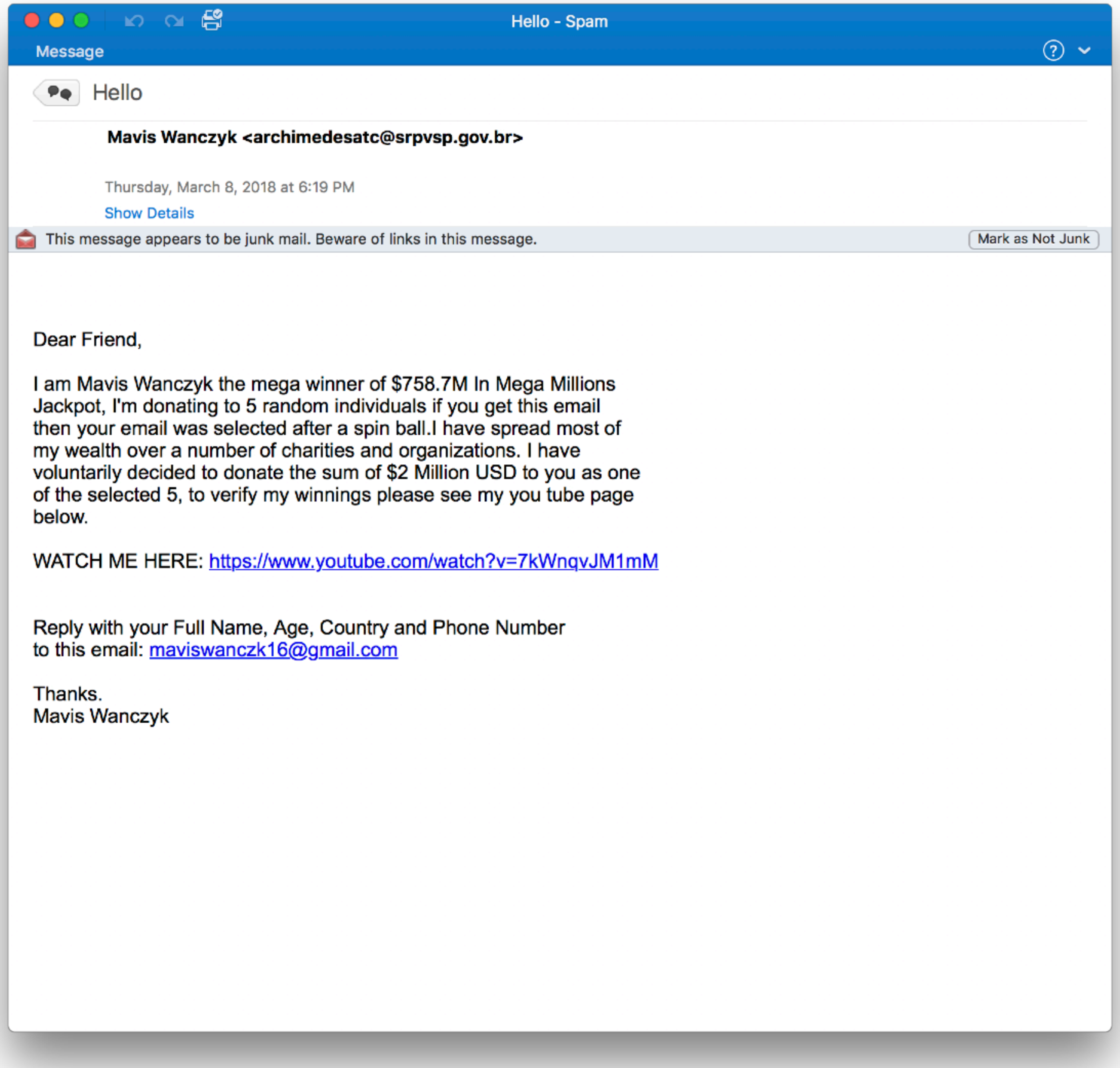
~80 - 90%

Pinky said,
"Gee, Brain. What are we going to do tonight?"
The Brain replied, "The same thing we do every night, Pinky. Try to take over the world."

* Typical form for scientific experiments and company databases
** *RMarkdown* Document Properties (YAML), JavaScript Object Notation (JSON), XML
*** Pure text documents, images, social media posts, and so on. No visible relationship.

E-mails and Spam

… common unstructured data problems …



Hello

Mavis Wanczyk <archimedesatc@srpvsp.gov.br>

Thursday, March 8, 2018 at 6:19 PM

This message appears to be junk mail. Beware of links in this message.

Dear Friend,

I am Mavis Wanczyk the mega winner of $758.7M In Mega Millions Jackpot, I'm donating to 5 random individuals if you get this email then your email was selected after a spin ball.I have spread most of my wealth over a number of charities and organizations. I have voluntarily decided to donate the sum of $2 Million USD to you as one of the selected 5, to verify my winnings please see my you tube page below.

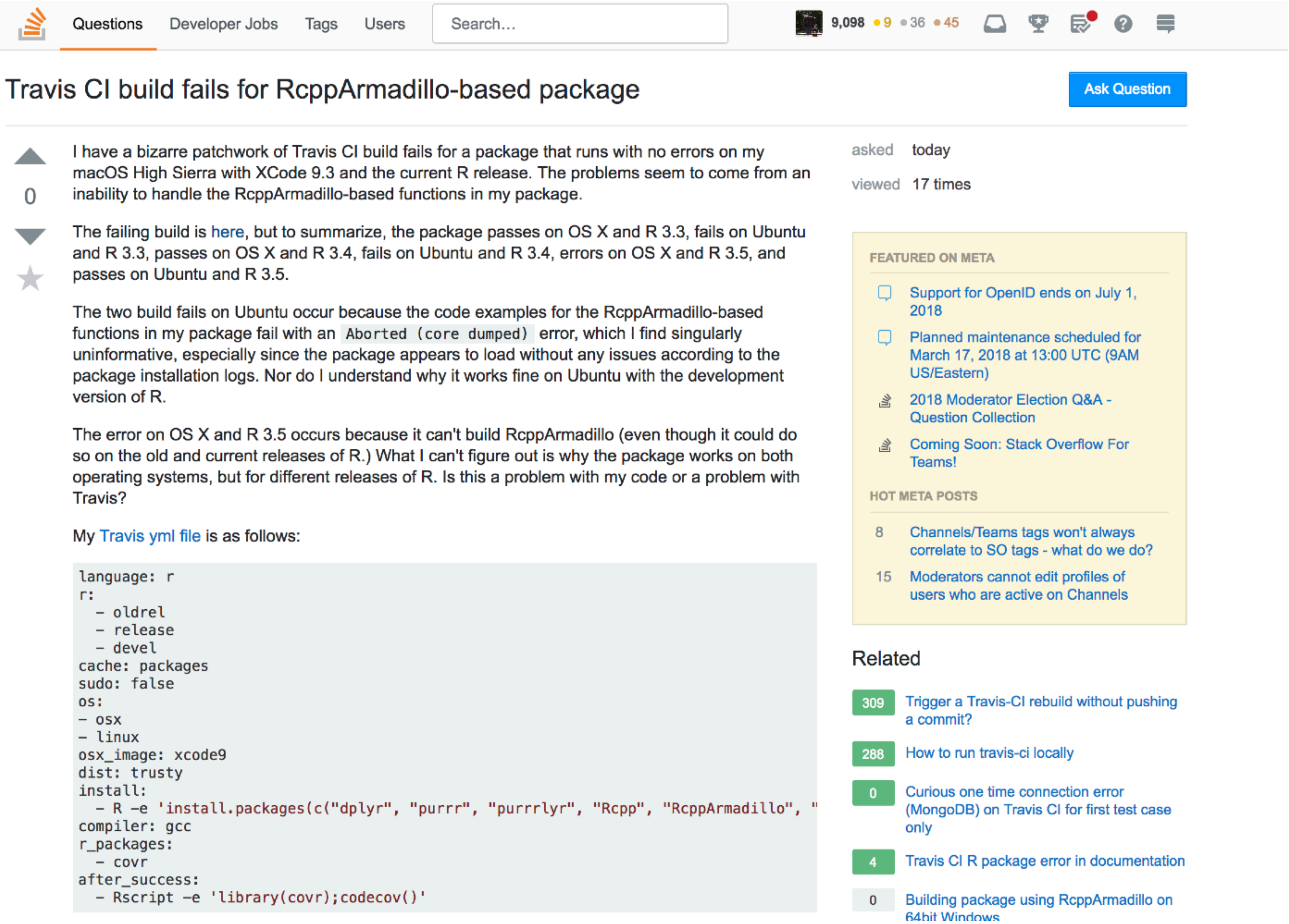
WATCH ME HERE: https://www.youtube.com/watch?v=7kWnqvJM1mM

Reply with your Full Name, Age, Country and Phone Number
to this email: maviswanczk16@gmail.com

Thanks.
Mavis Wanczyk

Text on Web Pages

… common unstructured data problems …



Travis CI build fails for RcppArmadillo-based package

I have a bizarre patchwork of Travis CI build fails for a package that runs with no errors on my macOS High Sierra with XCode 9.3 and the current R release. The problems seem to come from an inability to handle the RcppArmadillo-based functions in my package.

The failing build is here, but to summarize, the package passes on OS X and R 3.3, fails on Ubuntu and R 3.3, passes on OS X and R 3.4, fails on Ubuntu and R 3.4, errors on OS X and R 3.5, and passes on Ubuntu and R 3.5.

The two build fails on Ubuntu occur because the code examples for the RcppArmadillo-based functions in my package fail with an `Aborted (core dumped)` error, which I find singularly uninformative, especially since the package appears to load without any issues according to the package installation logs. Nor do I understand why it works fine on Ubuntu with the development version of R.

The error on OS X and R 3.5 occurs because it can't build RcppArmadillo (even though it could do so on the old and current releases of R.) What I can't figure out is why the package works on both operating systems, but for different releases of R. Is this a problem with my code or a problem with Travis?

My Travis yml file is as follows:

```
language: r
r:
  - oldrel
  - release
  - devel
cache: packages
sudo: false
os:
- osx
- linux
osx_image: xcode9
dist: trusty
install:
  - R -e 'install.packages(c("dplyr", "purrr", "purrrlyr", "Rcpp", "RcppArmadillo", "
compiler: gcc
r_packages:
  - covr
after_success:
  - Rscript -e 'library(covr);codecov()'
```

asked today

viewed 17 times

FEATURED ON META

- Support for OpenID ends on July 1, 2018
- Planned maintenance scheduled for March 17, 2018 at 13:00 UTC (9AM US/Eastern)
- 2018 Moderator Election Q&A - Question Collection
- Coming Soon: Stack Overflow For Teams!

HOT META POSTS

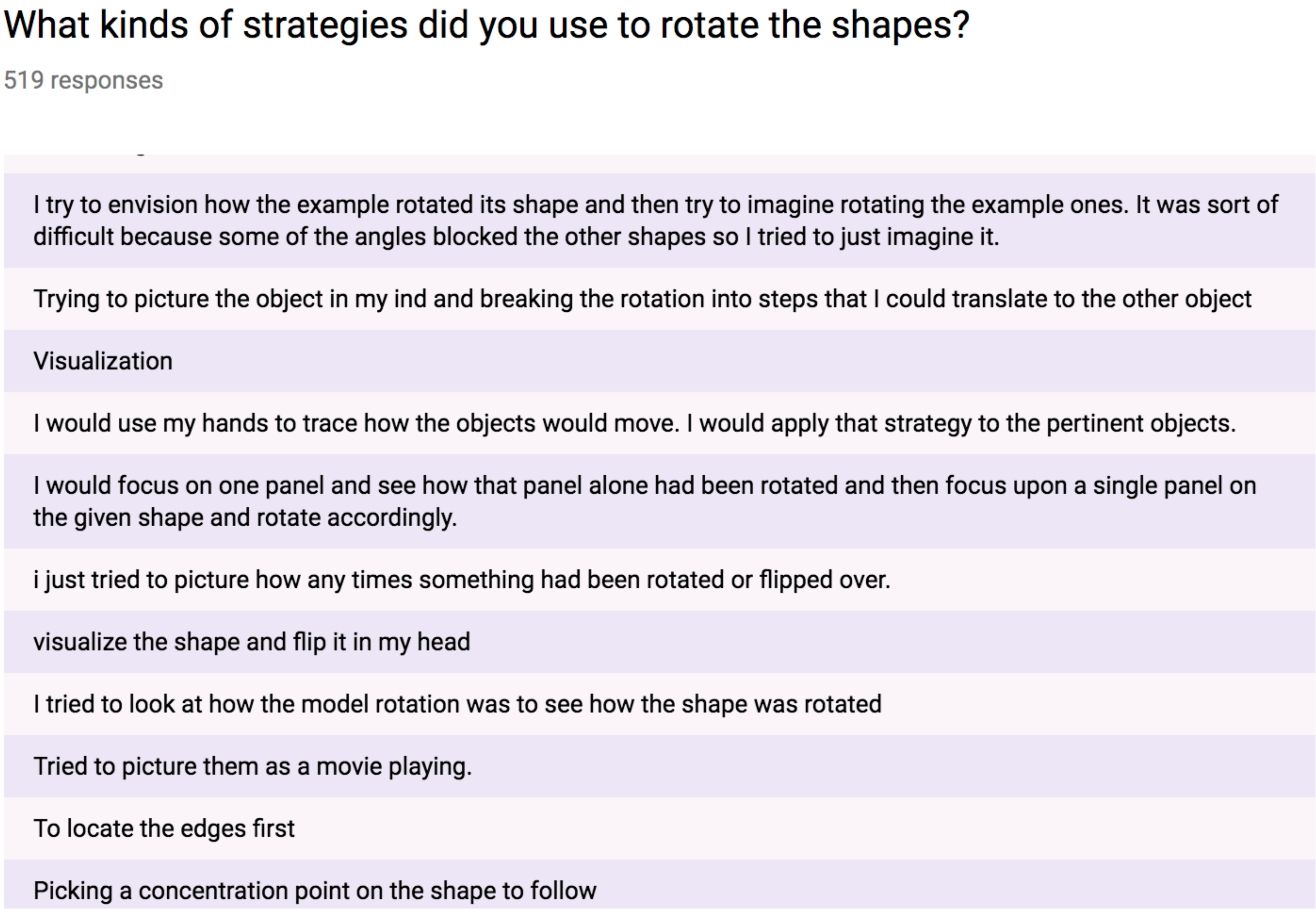
- 8 Channels/Teams tags won't always correlate to SO tags - what do we do?
- 15 Moderators cannot edit profiles of users who are active on Channels

Related

309	Trigger a Travis-CI rebuild without pushing a commit?
288	How to run travis-ci locally
0	Curious one time connection error (MongoDB) on Travis CI for first test case only
4	Travis CI R package error in documentation
0	Building package using RcppArmadillo on 64bit Windows

Free Response in Surveys

… common unstructured data problems …



What kinds of strategies did you use to rotate the shapes?

519 responses

I try to envision how the example rotated its shape and then try to imagine rotating the example ones. It was sort of difficult because some of the angles blocked the other shapes so I tried to just imagine it.

Trying to picture the object in my ind and breaking the rotation into steps that I could translate to the other object

Visualization

I would use my hands to trace how the objects would move. I would apply that strategy to the pertinent objects.

I would focus on one panel and see how that panel alone had been rotated and then focus upon a single panel on the given shape and rotate accordingly.

i just tried to picture how any times something had been rotated or flipped over.

visualize the shape and flip it in my head

I tried to look at how the model rotation was to see how the shape was rotated

Tried to picture them as a movie playing.

To locate the edges first

Picking a concentration point on the shape to follow

Text Representations

Definition:

Character is a *single* symbol that is displayed

'a' 'b' 'c' 'D' 'E' 'F'

'1' '2' '3' '4'

' ' '*' ',' '"'

Definition:
String multiple *characters* combined together.

'UIUC' 'STAT' 'Chambana' 'Chicago' 'Illinois'

Character Representation

```
class("S")
# [1] "character"

class("STAT 385")
# [1] "character"
```

Characters Welcome

… constructing strings …

```
double_quote = "Hello World!"

single_quote = 'Hello World!'

complex_string = "It's happening!"

escape_string = 'It\'s happening!'

white_space = " "

empty_string = ""
```

Escape Characters

… using special characters …

Symbol	Description
\n	*newline*
\r	carriage return
\t	tab
\b	backspace
\a	alert (bell)
\f	form feed
\v	vertical tab

Symbol	Description
\\	*backslash* \
\'	ASCII apostrophe '
\"	ASCII quotation mark "
\`	ASCII grave accent (backtick) `
\nnn	character with given octal code (1, 2 or 3 digits)
\xnn	character with given hex code (1 or 2 hex digits)
\unnnn	Unicode character with given code (1--4 hex digits)

Your Turn

Construct a string that includes the following quote in *R*

> “Actually, I see it as part of my job to inflict R on people who are perfectly happy to have never heard of it. Happiness doesn't equal proficient and efficient. In some cases the proficiency of a person serves a greater good than their momentary happiness.”

– Patrick Burns, R-help (2005)

String Operators

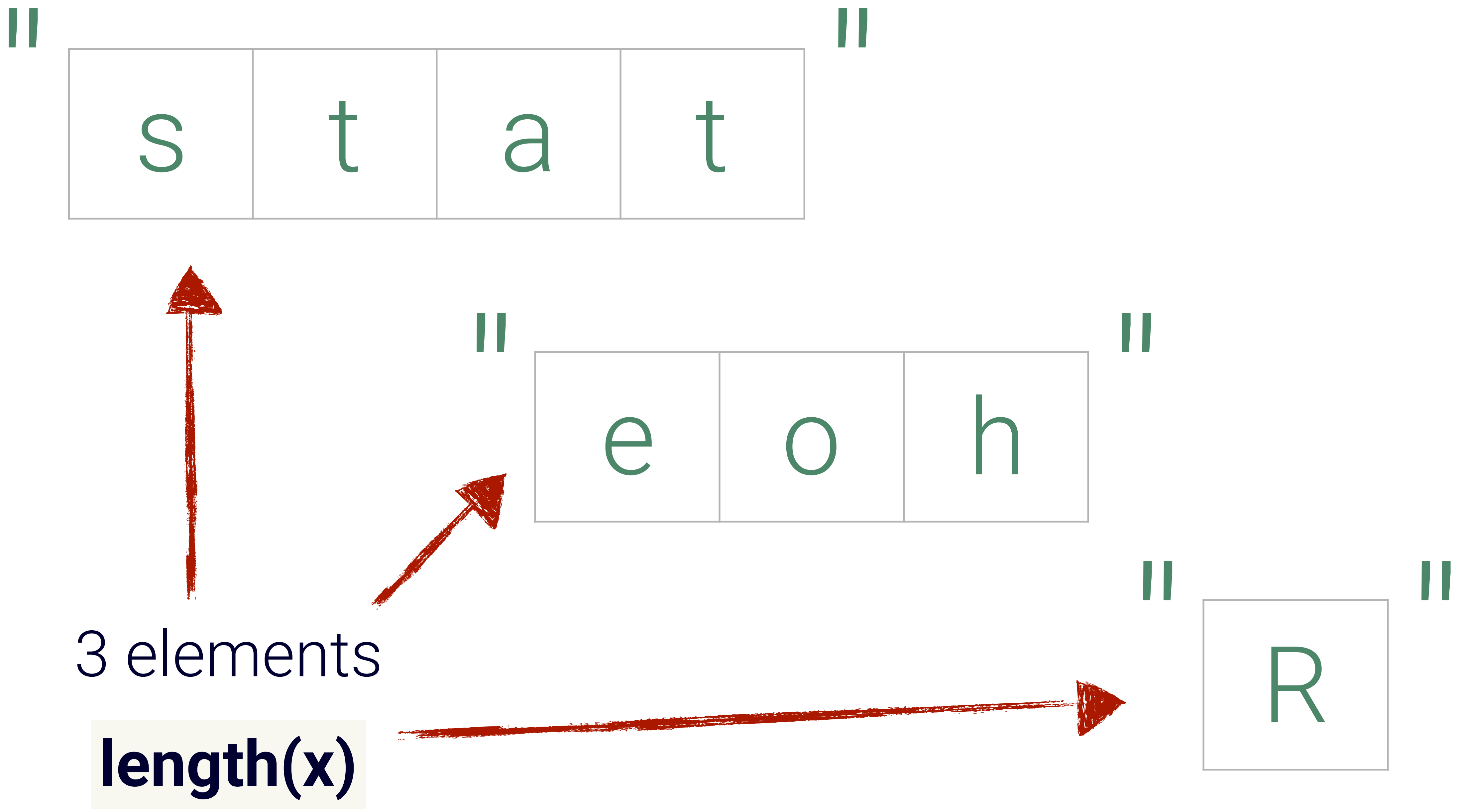
Length

Determining the character count of a string

```
# Total number of elements
length("stat")
# [1] 1

# How many letters per element?
nchar("stat")
# [1] 4

# Example String Vector
ex_string = c("stat", "eoh", "r")

length(ex_string)
# [1] 3

nchar(ex_string)
# [1] 4 3 1
```

Behind Length

... determining size ...

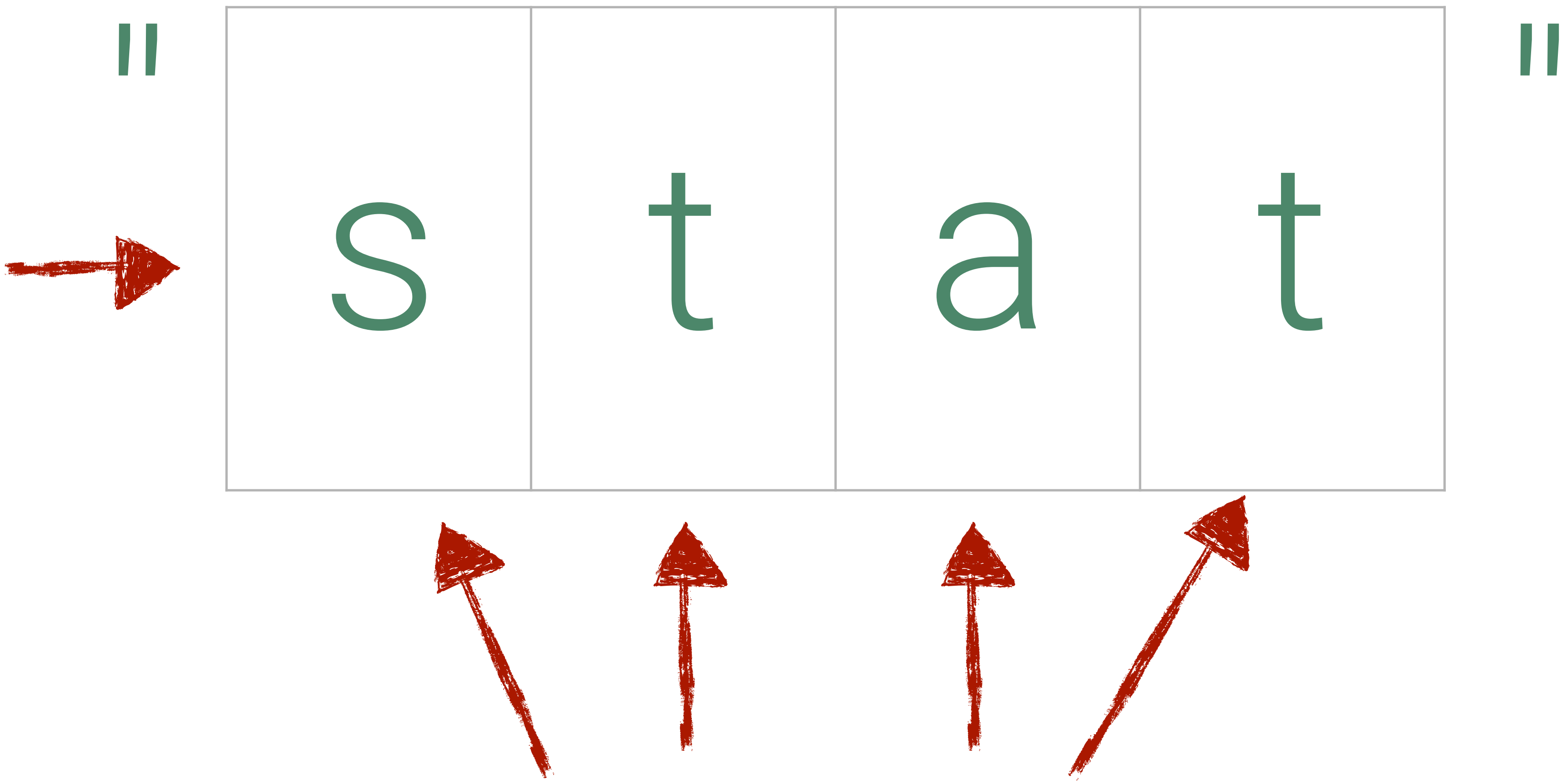


3 elements

length(x)

Behind Length

… determining size …



1 element

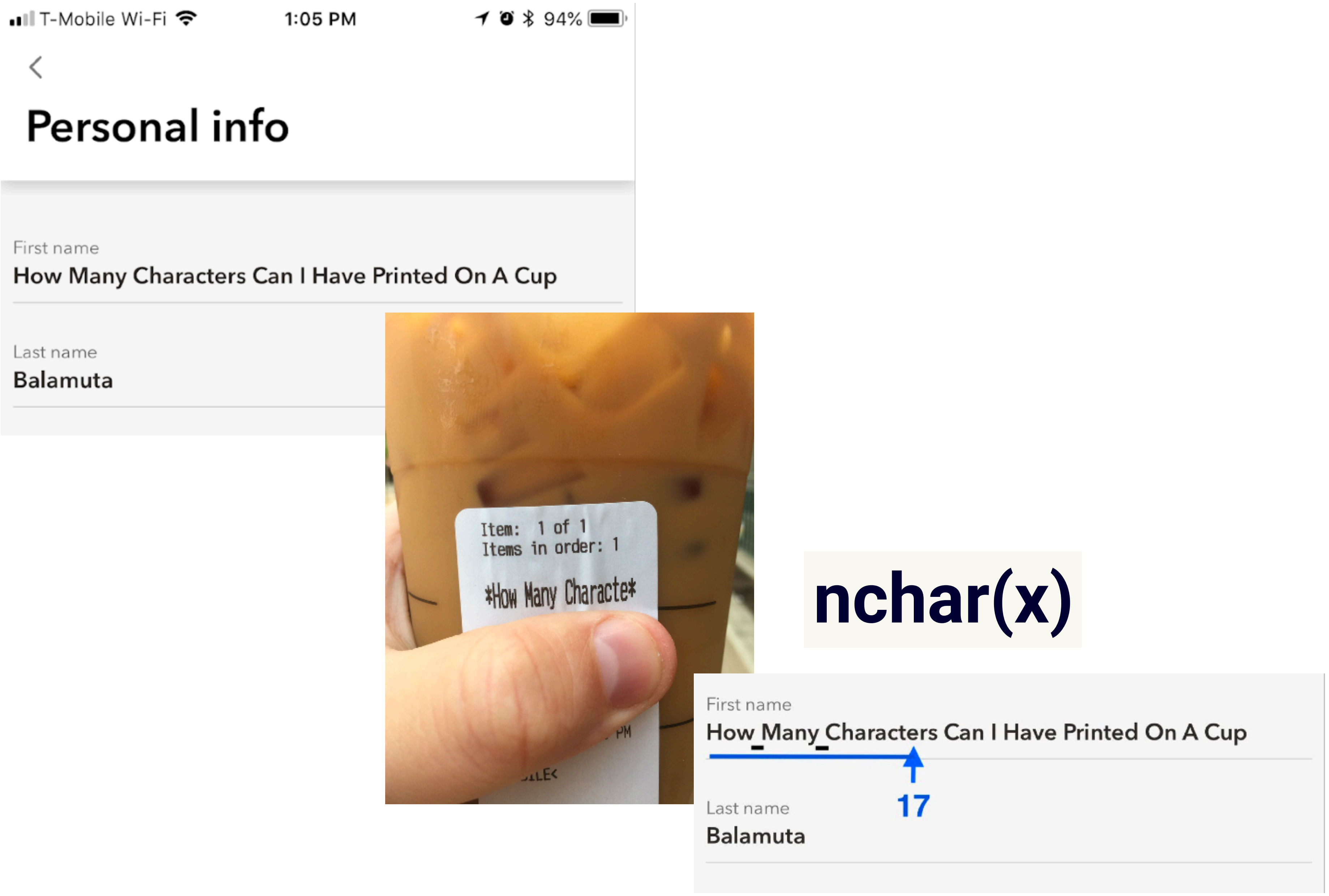
length(x)

4 characters

nchar(x)

Real-World Example

… giving a name for a mobile order at Starbucks …



Personal info

First name
How Many Characters Can I Have Printed On A Cup

Last name
Balamuta

Item: 1 of 1
Items in order: 1

How Many Characte

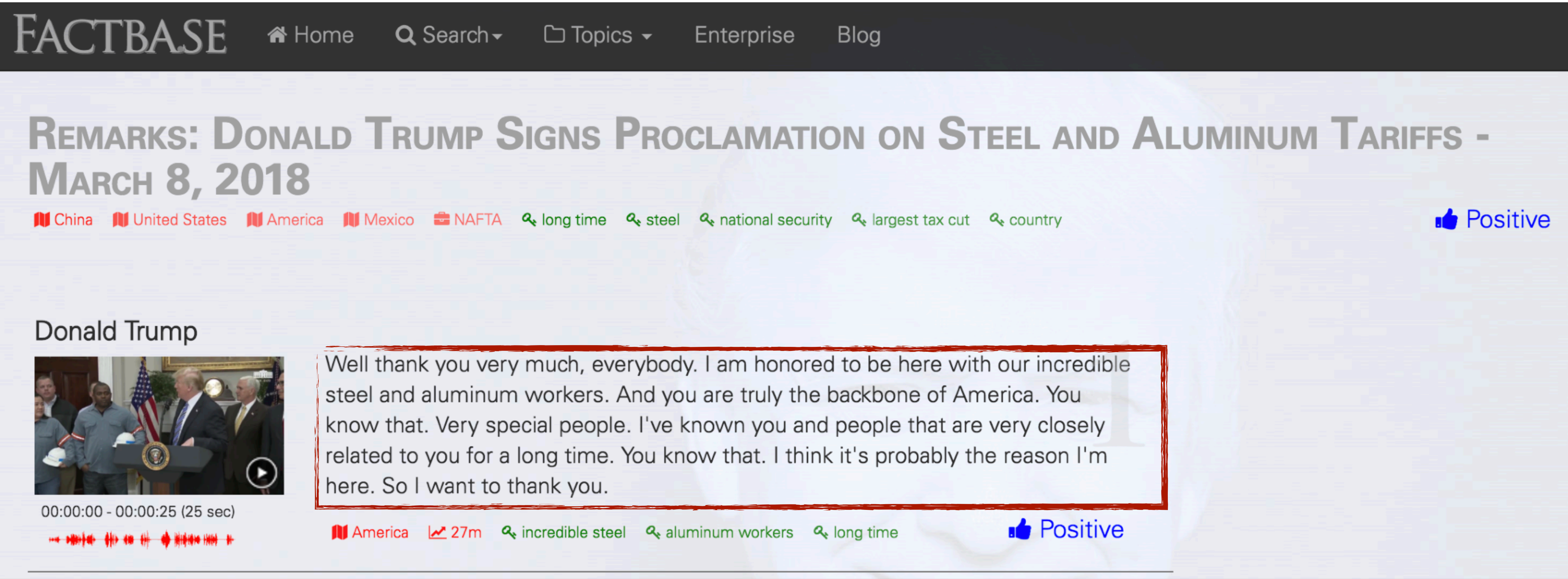
nchar(x)

First name
How Many Characters Can I Have Printed On A Cup

Last name
Balamuta

Your Turn

Count the number of characters used in this sound clip:



Remarks: Donald Trump Signs Proclamation on Steel and Aluminum Tariffs - March 8, 2018

China · United States · America · Mexico · NAFTA · long time · steel · national security · largest tax cut · country · 👍 Positive

Donald Trump

00:00:00 - 00:00:25 (25 sec)

Well thank you very much, everybody. I am honored to be here with our incredible steel and aluminum workers. And you are truly the backbone of America. You know that. Very special people. I've known you and people that are very closely related to you for a long time. You know that. I think it's probably the reason I'm here. So I want to thank you.

America · 27m · incredible steel · aluminum workers · long time · 👍 Positive

https://factba.se/transcript/donald-trump-remarks-steel-aluminum-tariffs-march-8-2018

Modifying Case

… UPPER to lower or lower to UPPER …

```
# Convert all letters to lower case
tolower("sTaT 385 at UiUc")
# [1] "stat 385 at uiuc"

# Convert all letters to upper case
toupper("sTaT 385 at UiUc")
# [1] "STAT 385 AT UIUC"
```

Concatenating Strings

… merging strings together …

```
your_name = "James"
paste("Hello World to you", your_name, "!") # Add white space
# [1] "Hello World to you James !"

paste0("Hello World to you", your_name, "!") # Omits white space
# [1] "Hello World to youJames!"

paste("Hello World to you", your_name, "!", sep = "--") # Control separator
# [1] "Hello World to you--James--!"

paste("Hello World to you", your_name, "!", sep = "") # Mimic paste0
# [1] "Hello World to youJames!"
```

Vectorized Concatenation

… merging strings together …

```
subject_ids = seq_len(5)

paste0("S", subject_ids)                # Create Subject IDs
# [1] "S1" "S2" "S3" "S4" "S5"

paste0("S", subject_ids, sep = "-")     # Specify a separator
# [1] "S1-" "S2-" "S3-" "S4-" "S5-"

paste0("S", subject_ids, collapse = "") # Reduce entries to a single value
# [1] "S1S2S3S4S5"
```

Your Turn

Form the following string:

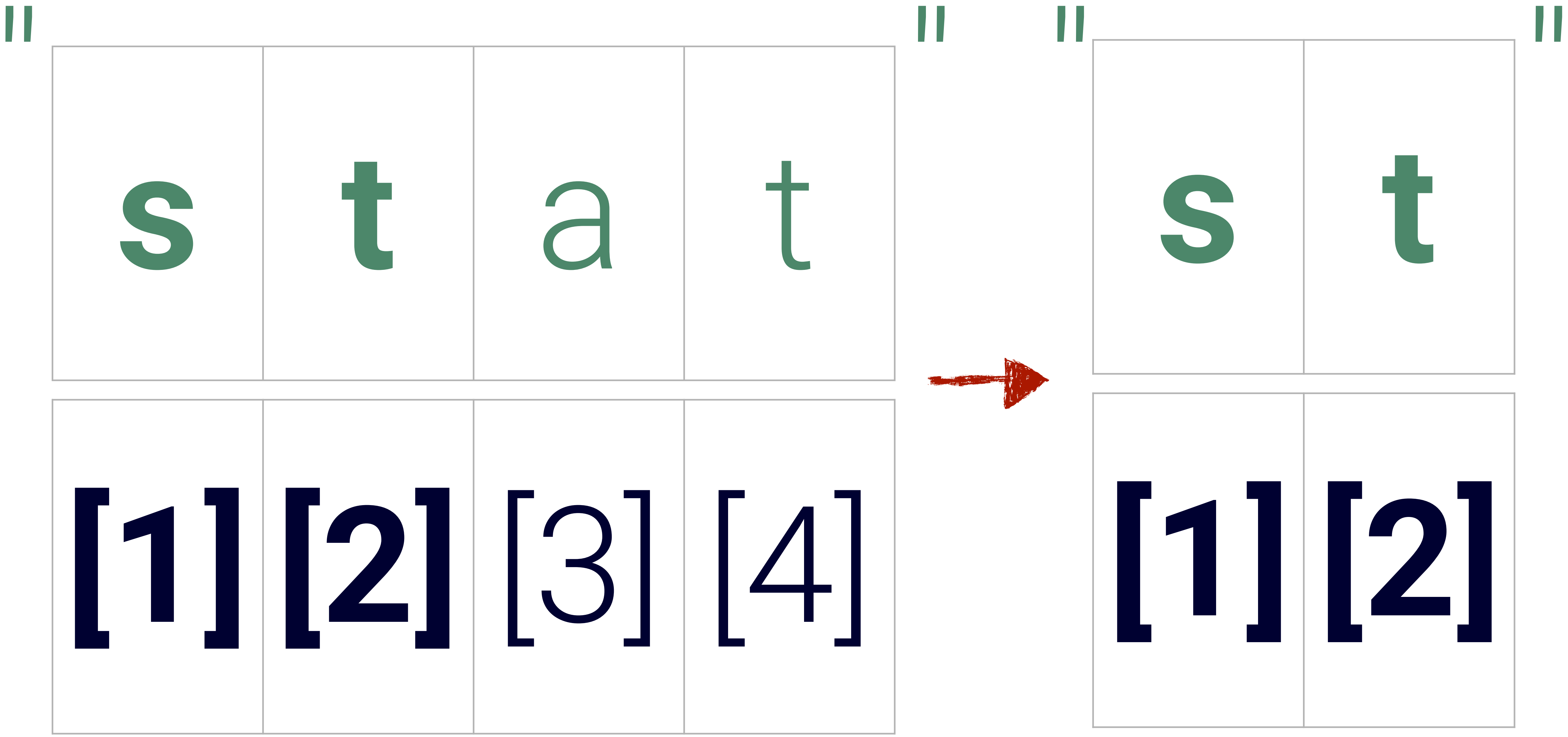
"Dividing {{x}} by {{mod}} gives a remainder of {{remainder}}"

To concatenate the following expressions:

```r
x = seq_len(5)
mod = 2

remainder = x %% mod
```

Substring

… extracting characters from a string …



"

s	t	a	t
[1]	[2]	[3]	[4]

"

"

s	t
[1]	[2]

"

Substring

… extracting characters from a string …

String Data
Strings to extract data from

Start Positional Index
Value to begin at inside the string

End Positional Index
Value to go up to in the string

```
substr(x = <data>, start = <begin-loc>, stop = <end-loc>)
```

Substrings

… cutting up a string into smaller strings …

```
substr("stat", 1, 2)
# [1] "st"

substr("Illinois", 4, 8)
# [1] "inois"

substr("coding", 7, 10)
# [1] ""

substr(c("stat", "Illinois"), 1:2, 3:4)
# [1] "sta" "lli"
```

Your Turn

Convert the start of all elements in the following vector to having a capital letter

```
x = c("mumford", "female", "male", "joe", "pete")
```

Splitting a String

… breaking a string in half …

String Data
Text data to be split apart

Pattern
Value to split on

```
strsplit(x = <data>, split = <pattern>)
```

Splitting a String

… breaking a string in half …

```
dishes = c("Spaghetti and Meatballs", "French Onion Soup")

strsplit(dishes, " ")
# [[1]]
# [1] "Spaghetti" "and"       "Meatballs"
#
# [[2]]
# [1] "French" "Onion"  "Soup"
```

Recap

- **Unstructured Data**
  - Text data such as e-mails, help posts, free response
- **Text Representation**
  - How *R* thinks about strings
- **String Operators**
  - Ways to modify strings in *R*