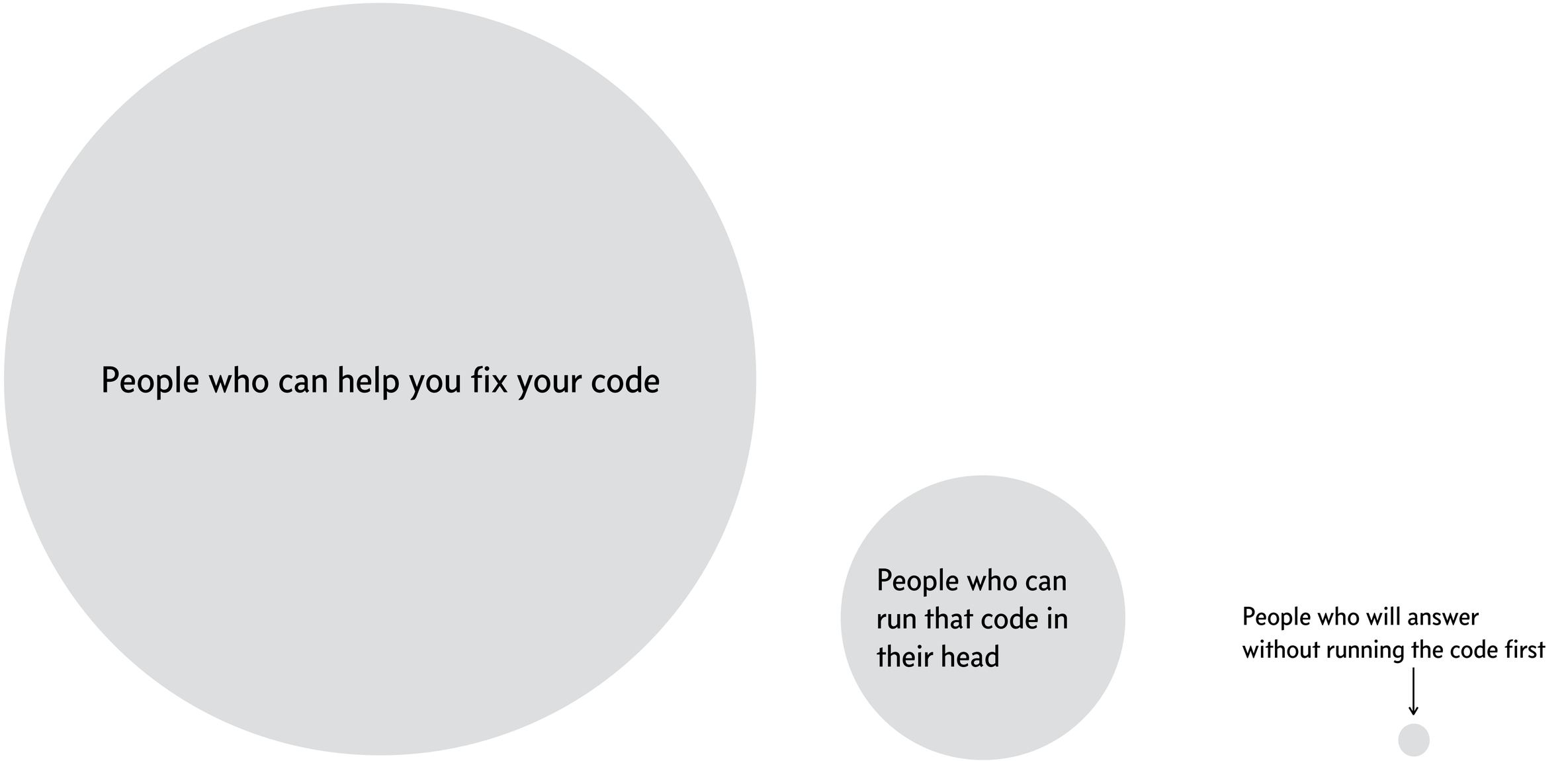
how i found your answer

Mara Averick (@dataandme)
Tidyverse Developer Advocate, RStudio
bit.ly/noreastr

by running your code...


People who can help you fix your code
People who can run that code in their head
People who will answer without running the code first

The art of the question

The most useless problem statement that one can face is "it doesn't work", yet we seem to get it far too often.

– Thiago Maciera

Maciera, Thiago. 2012. "The Art of Problem Solving." In Open Advice: FOSS: What We Wish We Had Known When We Started, edited by Lydia Pintscher, 55-61.

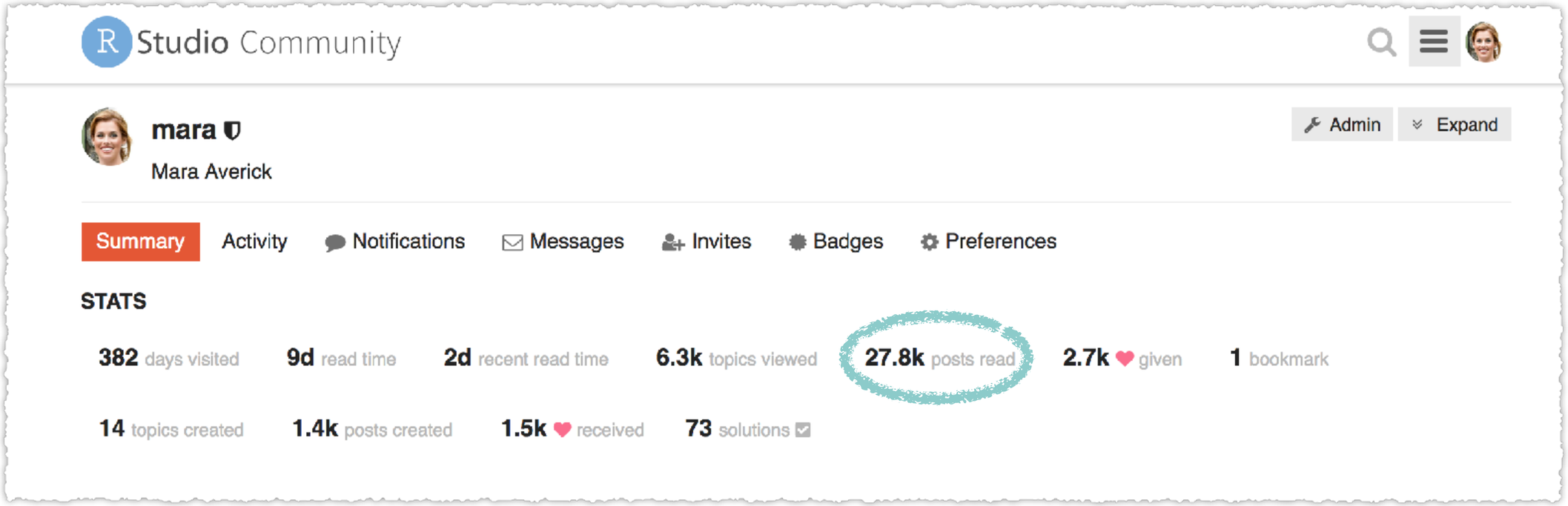
The newcomer's paradox...

When you ask for help, some friendly soul will no doubt tell you that "it's easy, just do foo, bar and baz." *Except for you, it is not easy, there may be no documentation for foo, bar is not doing what it is supposed to be doing and what is this baz thing anyway with its eight disambiguation entries on Wikipedia?*

— Leslie Hawthorne

"You'll Eventually Know Everything They've Forgotten." In *Open Advice: FOSS: What We Wish We Had Known When We Started*, edited by Lydia Pintscher, 29–32.

Lessons learned reading ~28k posts...



Source: mara profile summary on RStudio Community https://community.rstudio.com/u/mara/summary

Context is key

Where to ask?

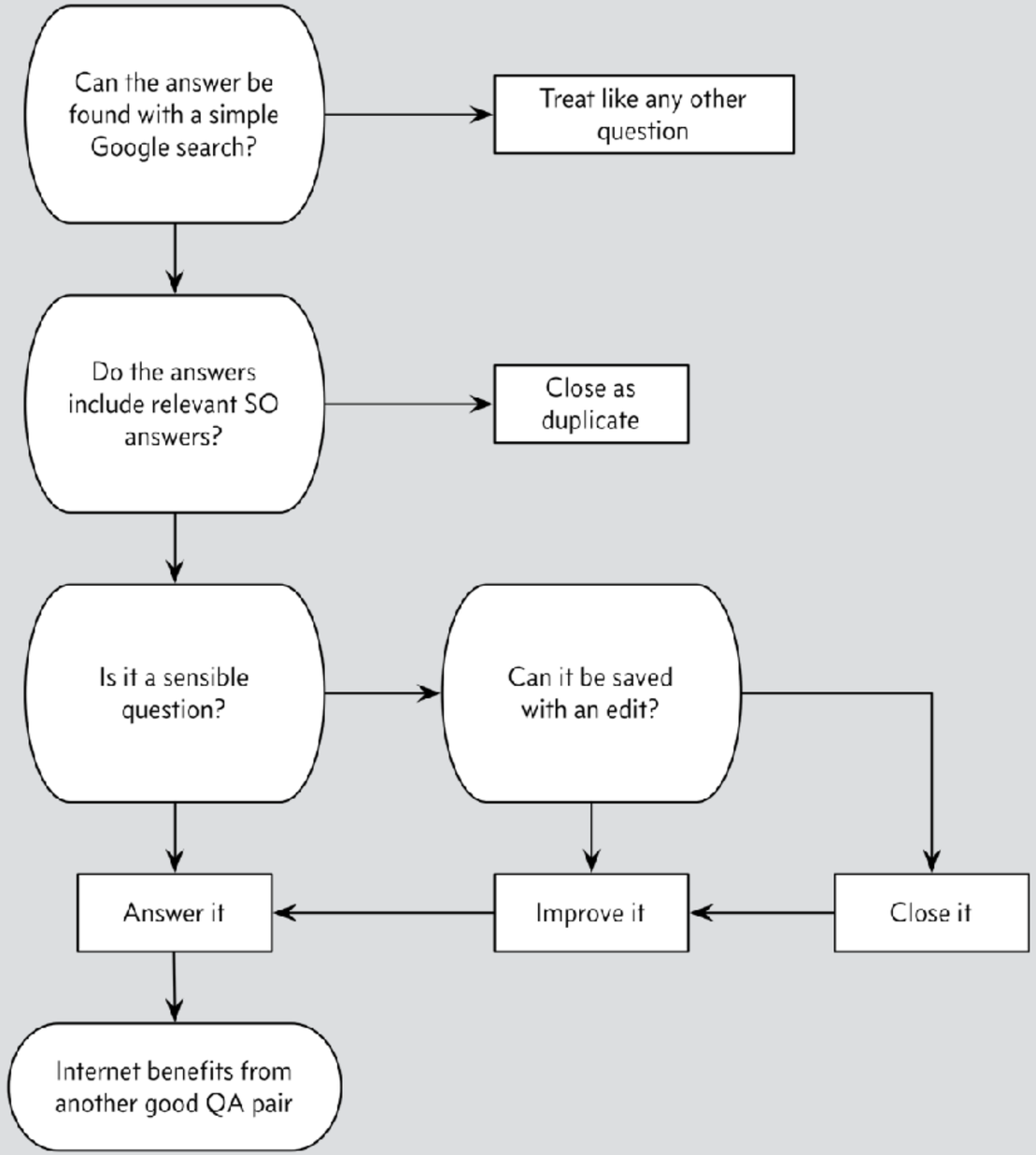
Twitter

RStudio Community

StackOverflow

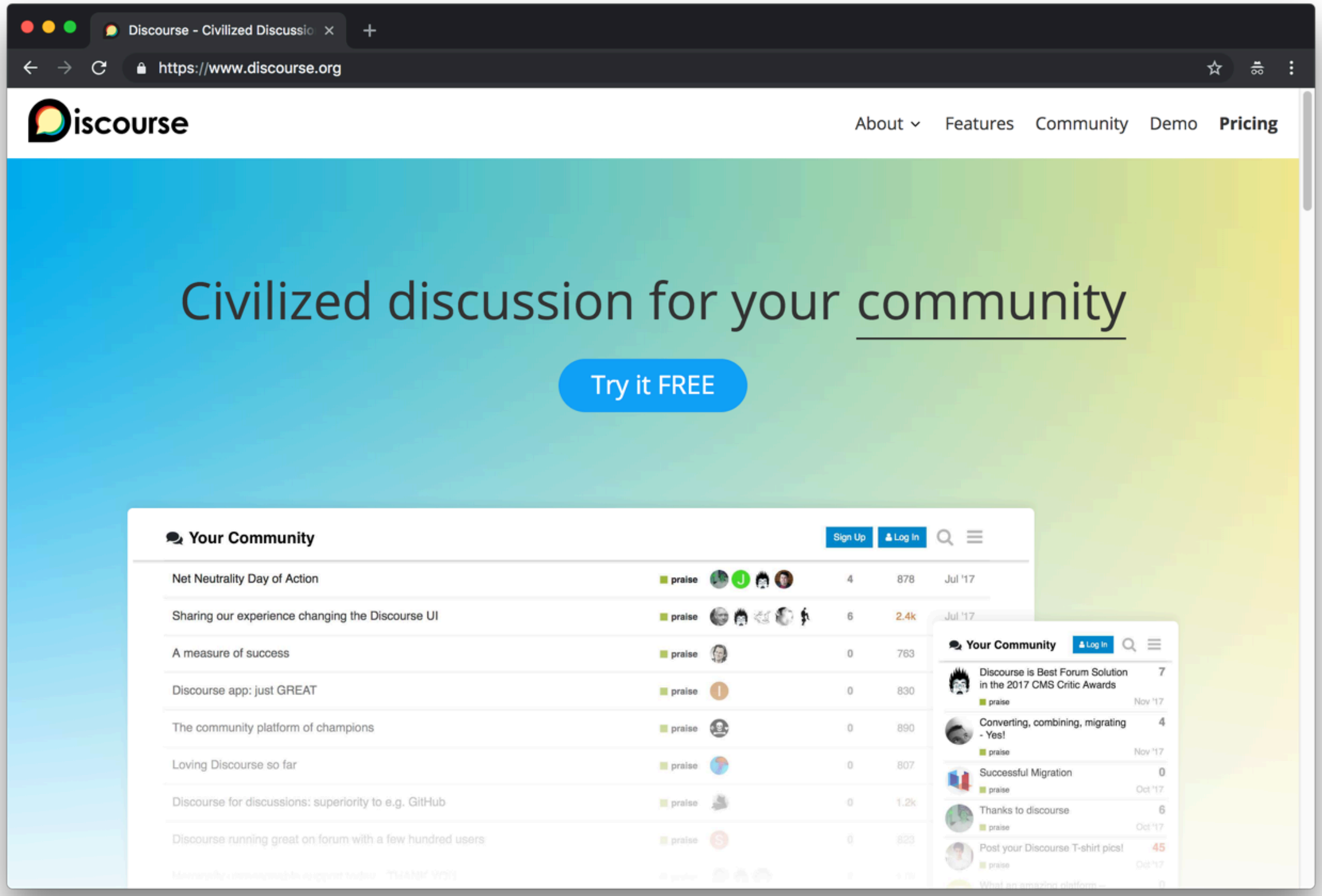
How should you respond to RTFM questions?

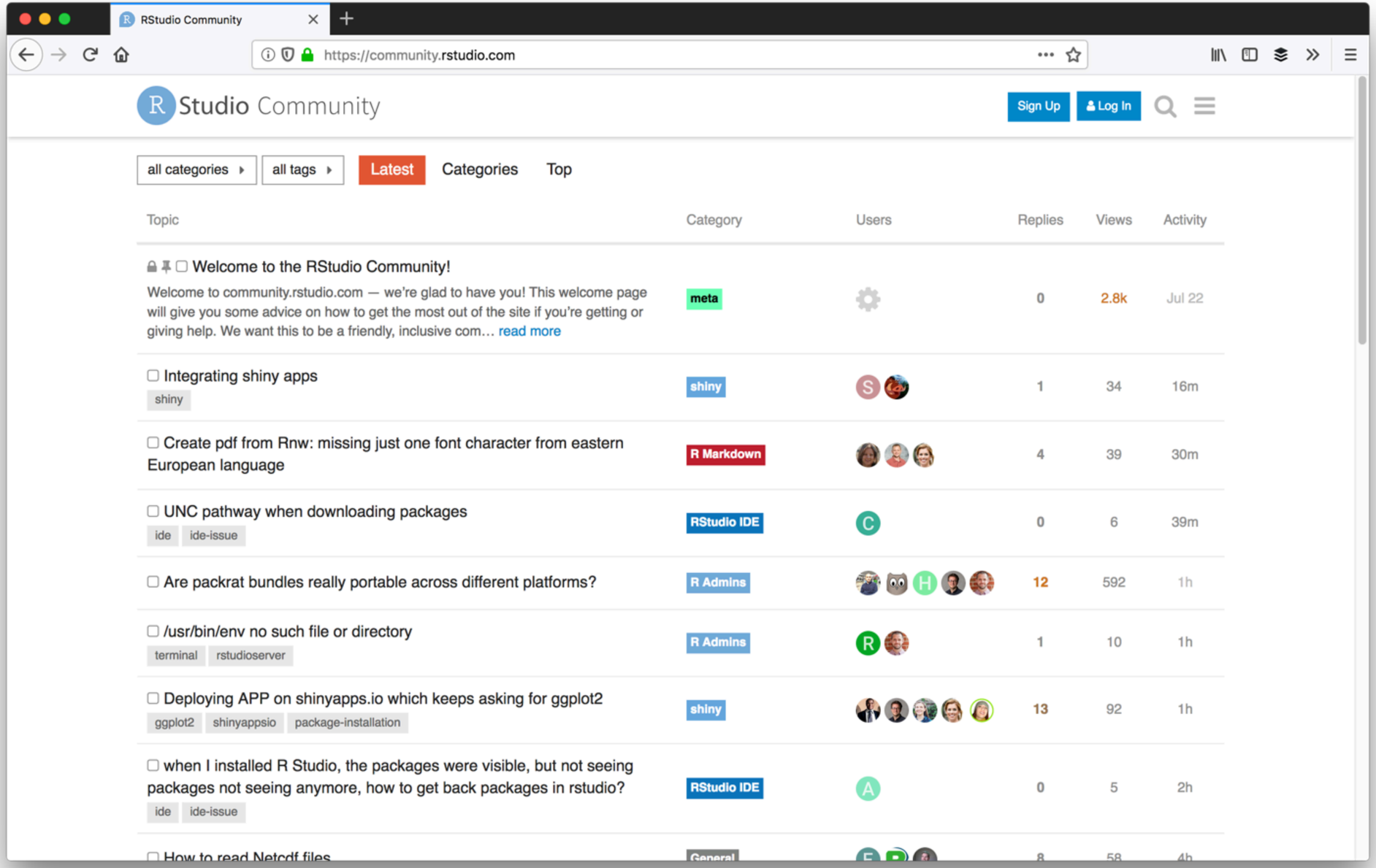
goal = good QA pair



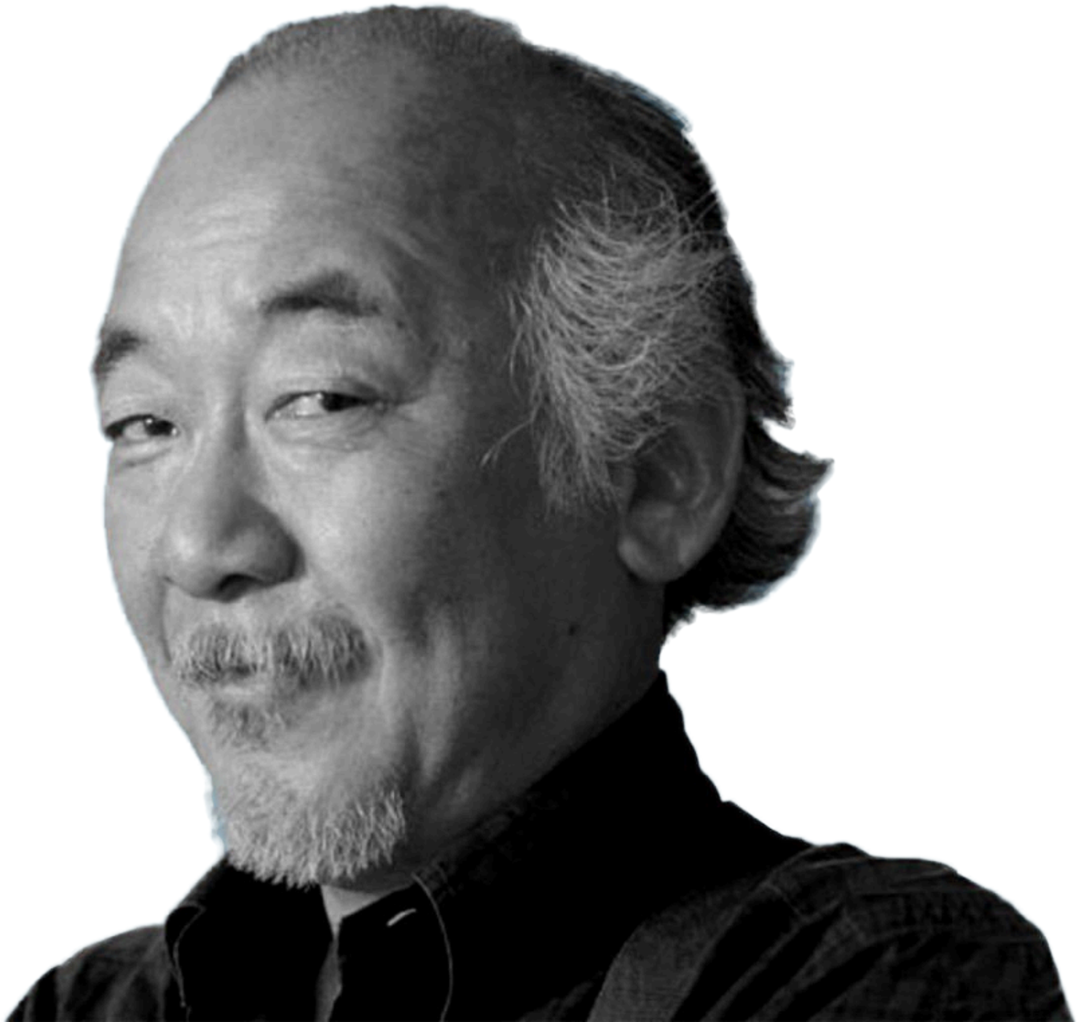
Source: Shawn Chin https://meta.stackexchange.com/a/161346/213022


R Studio Community


Discourse - Civilized Discussio
https://www.discourse.org
Discourse
About
Features
Community
Demo
Pricing
Civilized discussion for your community
Try it FREE
Your Community
Sign Up
Log In
Net Neutrality Day of Action
Sharing our experience changing the Discourse UI
A measure of success
Discourse app: just GREAT
The community platform of champions
Loving Discourse so far
Discourse for discussions: superiority to e.g. GitHub
Discourse running great on forum with a few hundred users
Your Community
Log In
Discourse is Best Forum Solution in the 2017 CMS Critic Awards
Converting, combining, migrating - Yes!
Successful Migration
Thanks to discourse
Post your Discourse T-shirt pics!


RStudio Community
https://community.rstudio.com
RStudio Community
Sign Up
Log In
all categories
all tags
Latest
Categories
Top
Topic
Category
Users
Replies
Views
Activity
Welcome to the RStudio Community!
Welcome to community.rstudio.com — we're glad to have you! This welcome page will give you some advice on how to get the most out of the site if you're getting or giving help. We want this to be a friendly, inclusive com… read more
meta
0
2.8k
Jul 22
Integrating shiny apps
shiny
shiny
1
34
16m
Create pdf from Rnw: missing just one font character from eastern European language
R Markdown
4
39
30m
UNC pathway when downloading packages
ide
ide-issue
RStudio IDE
0
6
39m
Are packrat bundles really portable across different platforms?
R Admins
12
592
1h
/usr/bin/env no such file or directory
terminal
rstudioserver
R Admins
1
10
1h
Deploying APP on shinyapps.io which keeps asking for ggplot2
ggplot2
shinyappsio
package-installation
shiny
13
92
1h
when I installed R Studio, the packages were visible, but not seeing packages not seeing anymore, how to get back packages in rstudio?
ide
ide-issue
RStudio IDE
0
5
2h
How to read Netcdf files
General
8
58
4h

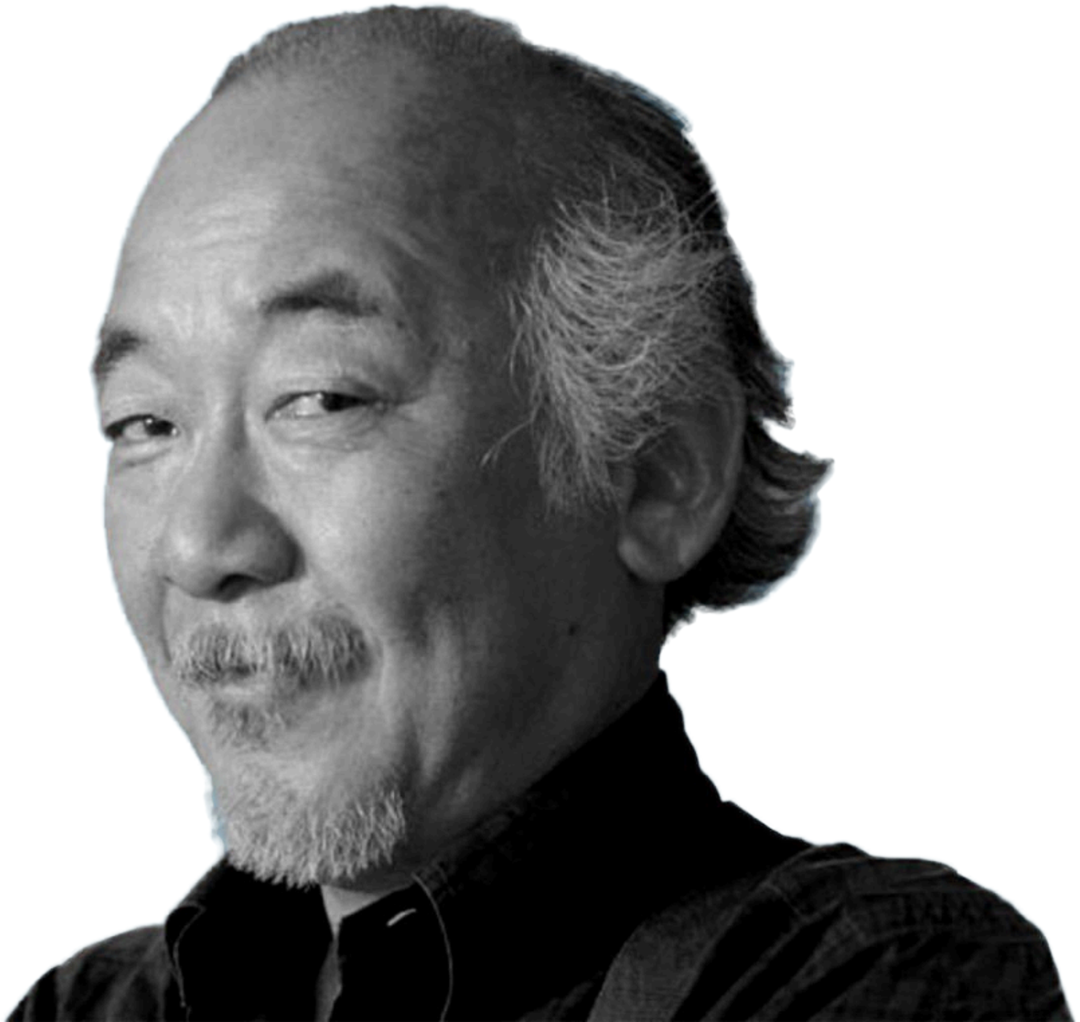
What's in it for me?



The Karate Kid (1984)

What's in it for me?

* mad hot troubleshooting skillz *



The Karate Kid (1984)

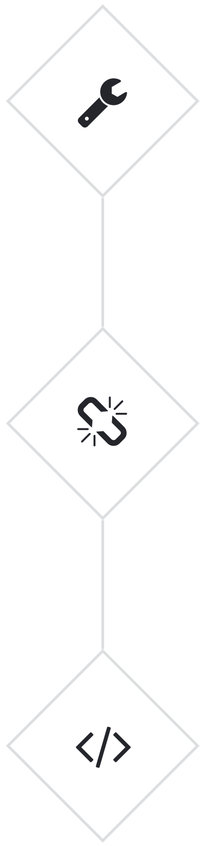
10 simple rules for getting help from online scientific communities

1. Do not be afraid to ask a question
2. State the question clearly
3. New to a mailing list? Learn the established customs before posting
4. Do not ask what has already been answered
5. Always use a good title
6. Do your homework before posting
7. Proofread your post and write in correct English
8. Be courteous to other forum members
9. Remember that the archive of your discussion can be useful to other people
10. Give back to the community

Dall'Olio, Giovanni M., Jacopo Marino, Michael Schubert, Kevin L. Keys, Melanie I. Stefan, Colin S. Gillespie, Pierre Poulain, et al. 2011. "Ten Simple Rules for Getting Help from Online Scientific Communities." PLoS Computational Biology 7 (9): 10–12. doi:10.1371/journal.pcbi.1002202.


THE ROLLING STONES
EMPATHY FOR THE USER

the anatomy of an issue

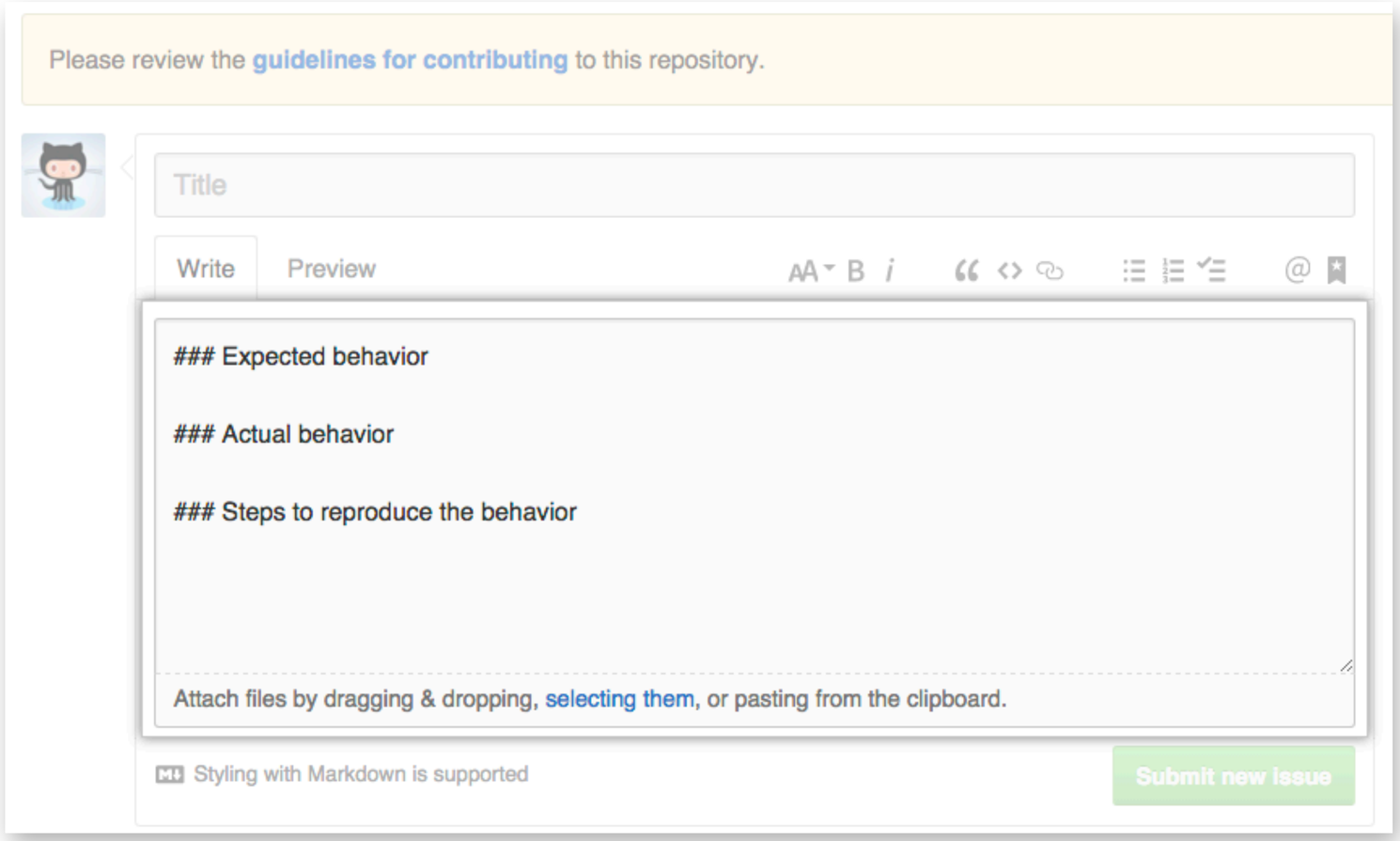


PROBLEM DESCRIPTION

EXPECTED BEHAVIOUR

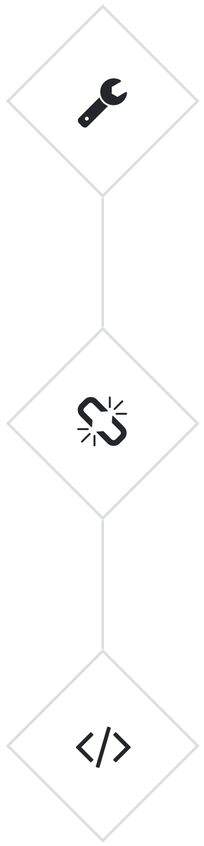
MINIMAL REPRODUCIBLE EXAMPLE

the anatomy of an issue



"Creating an issue template for your repository" - GitHub Help < https://help.github.com/articles/creating-an-issue-template-for-your-repository/>

the anatomy of ~~an issue~~ figuring sh*t out



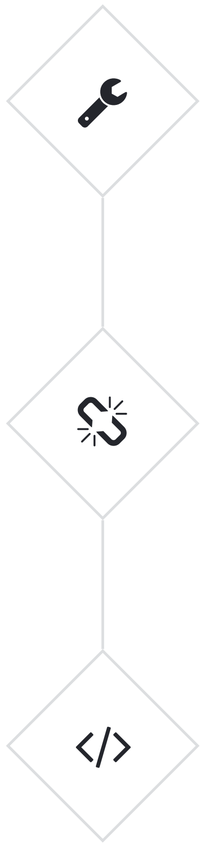
PROBLEM DESCRIPTION

EXPECTED BEHAVIOUR

MINIMAL REPRODUCIBLE EXAMPLE

the anatomy of ~~an issue~~ figuring sh*t out

the anatomy of ~~an issue~~ figuring sh*t out

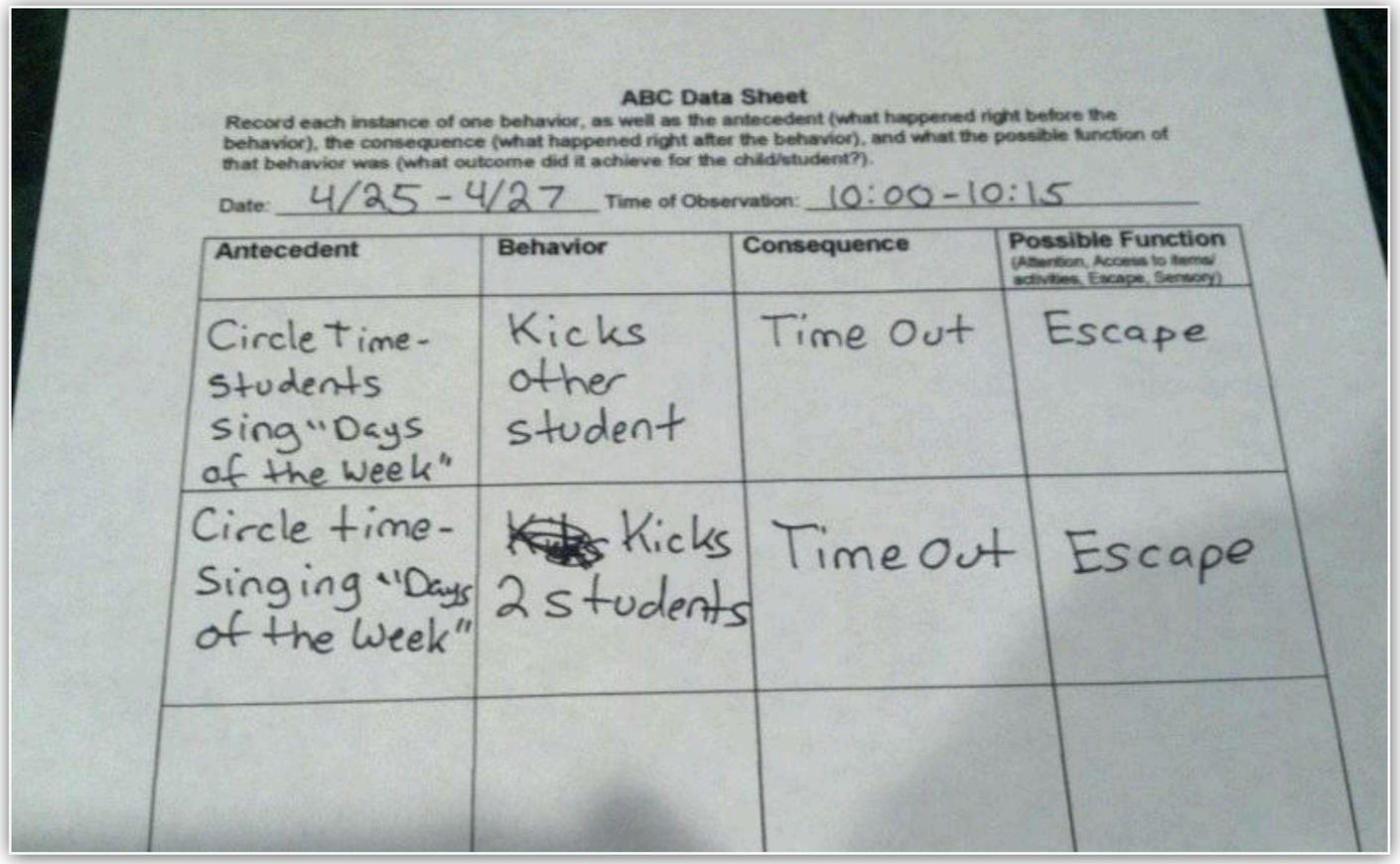


ANTECEDENTS

BEHAVIOURS

CONSEQUENCES

the anatomy of ~~an issue~~ figuring sh*t out



ABC Data Sheet

Record each instance of one behavior, as well as the antecedent (what happened right before the behavior), the consequence (what happened right after the behavior), and what the possible function of that behavior was (what outcome did it achieve for the child/student?).

Date: 4/25 - 4/27 Time of Observation: 10:00 - 10:15

Antecedent	Behavior	Consequence	Possible Function (Attention, Access to items/ activities, Escape, Sensory)
Circle time - students sing "Days of the week"	Kicks other student	Time Out	Escape
Circle time - Singing "Days of the week"	~~Kicks~~ Kicks 2 students	Time Out	Escape

Happy Face Behavior Chart Printable Beauty Within Clinic Template Gallery Design Ideas Consequence

Error Information: Packages missing

Description of issue -

Steps taken so far - i tried and tried..

System Information:

- RStudio Edition: (Desktop or Server)
- RStudio Version:
- OS Version:
- R Version:

Also:

- RStudio diagnostics report:
- Your `sessionInfo()`:
- RStudio crash report:
- RStudio application log files:

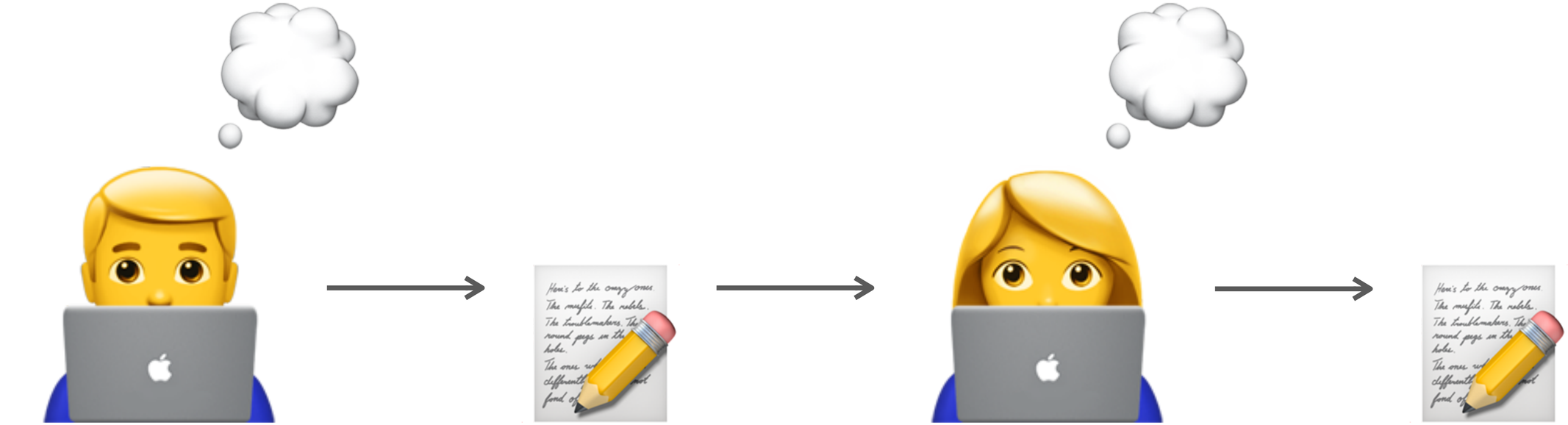
From Troubleshooting Guide: Using RStudio

They're not *wrong*

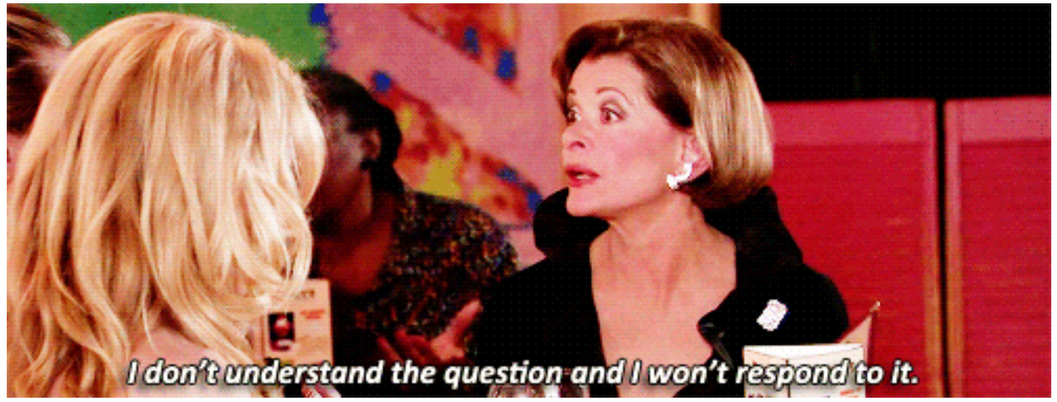
“It is impossible to speak in such a way that you cannot be misunderstood.”

— Karl Popper

Writing prose about code...

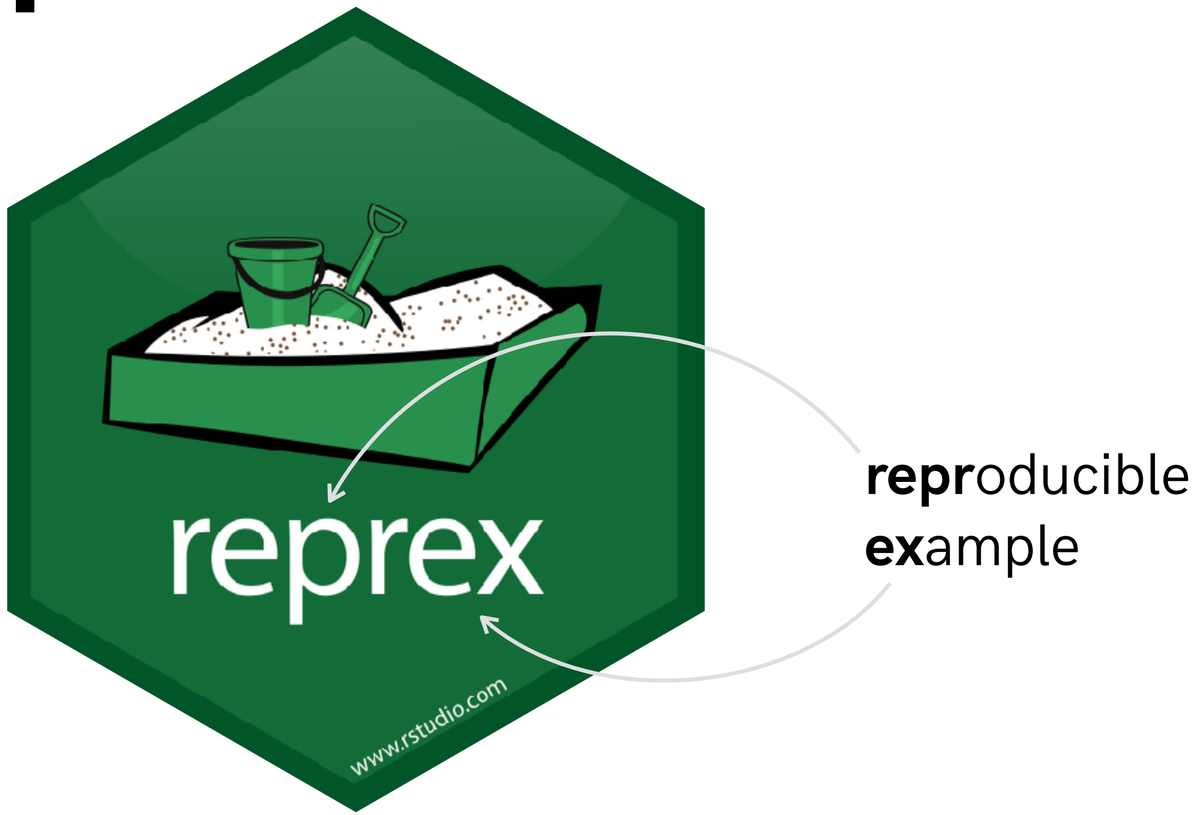


Writing prose about code...



Writing prose about code...

the magic of reprex



reproducible
example

reprex raison d’être

The reprex request trifecta

WHAT I'M ASKING YOU TO DO

Make a reproducible example

WHY I'M ASKING YOU TO DO IT

Help me help you — I need your data to do so

HOW YOU CAN DO THE THING

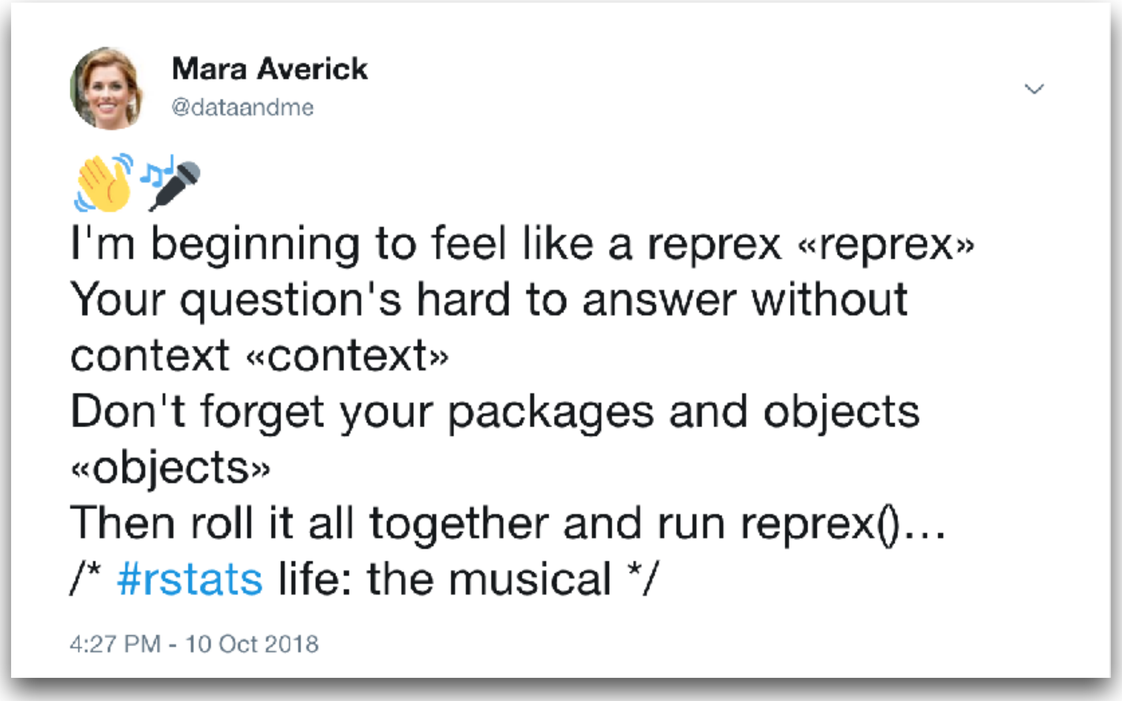
Resources, videos, we've got it all...

Keys to reprex-cellence

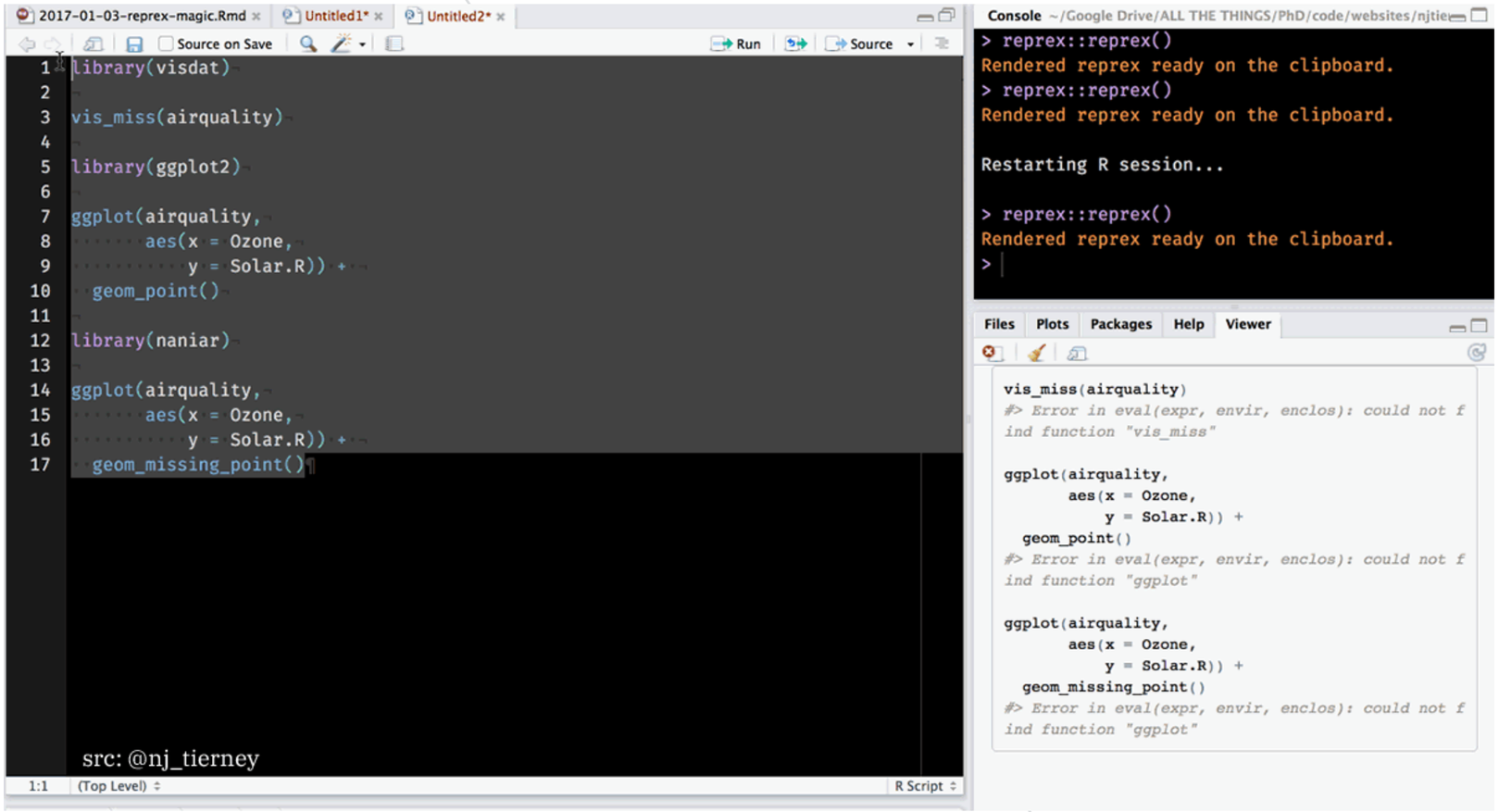
- ✓ Code that **actually runs**
- ✓ Code that **doesn't have to be run**
- ✓ Code that **can be easily run**

Source: Jenny Bryan, 2017. "reprex: the package, the point." https://speakerdeck.com/jennybc/reprex-help-me-help-you

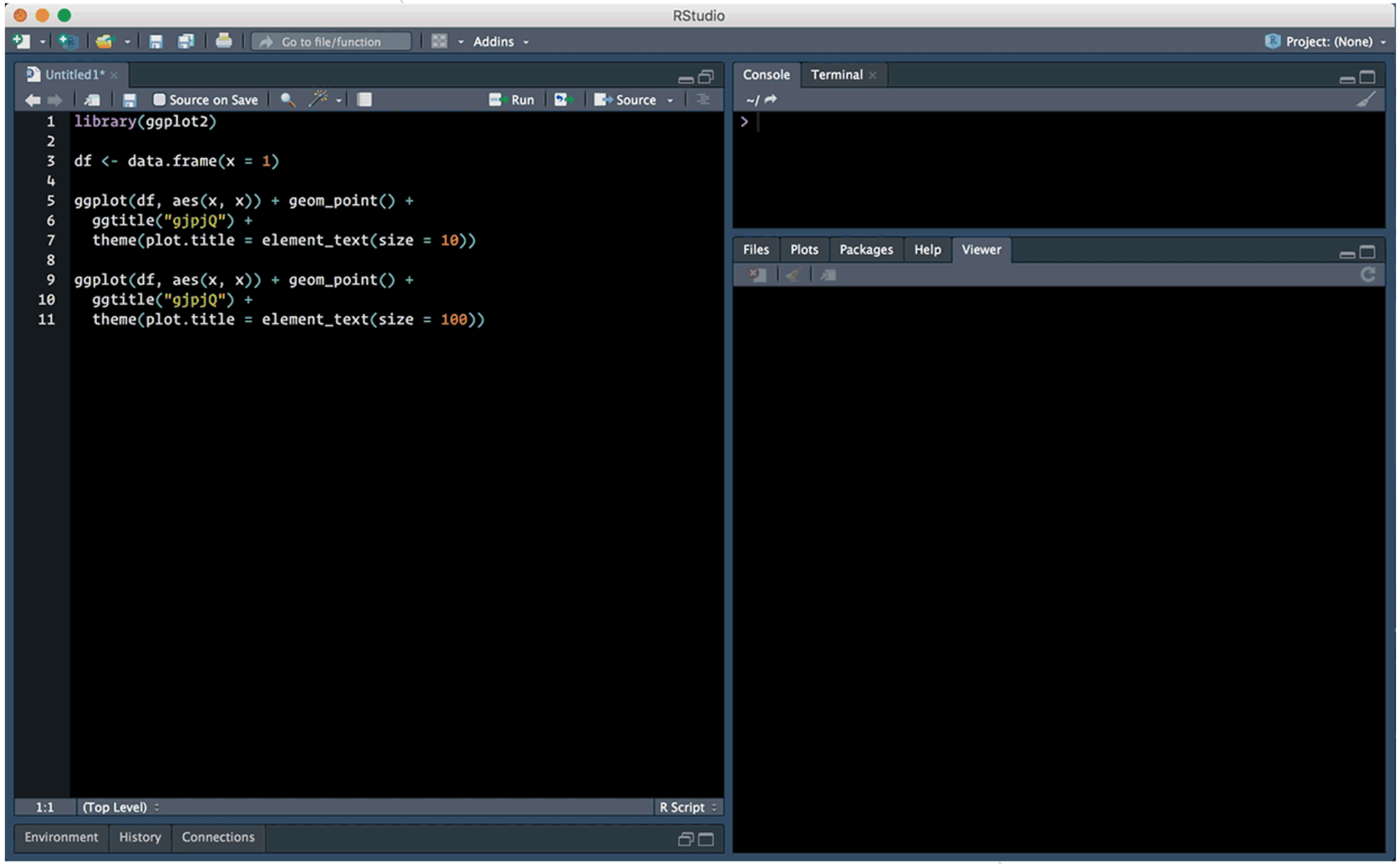
Keys to reprex-cellence



Source: Jenny Bryan, 2017. "reprex: the package, the point." https://speakerdeck.com/jennybc/reprex-help-me-help-you

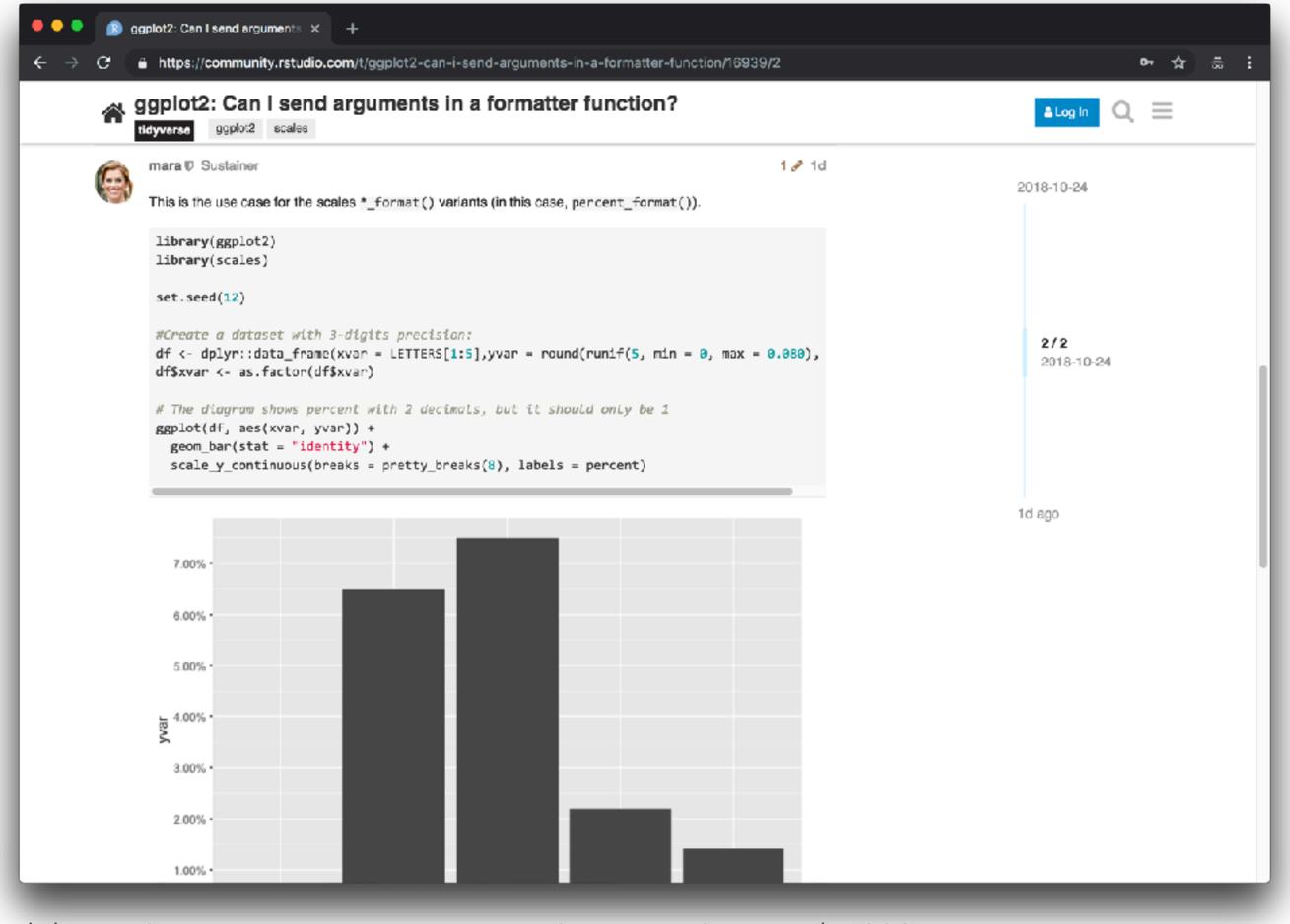


Source: Nick Tierney. "Magic reprex." 2017-01-11 <http://www.njtierney.com/post/2017/01/11/magic-reprex/>



Source: Nick Tierney. "Magic reprex." 2017-01-11 <http://www.njtierney.com/post/2017/01/11/magic-reprex/>

But wait, there's more...



https://community.rstudio.com/t/ggplot2-can-i-send-arguments-in-a-formatter-function/16939

Meta-help

RStudio Community

Why does head() show 6 rows by default?

hadley RStudio Employee Nov '17

To answer a question like this, I first starting looking through the S books I have on hand (e.g. The New S Language 11). I don't see `head()` mentioned in the index, so that suggests it's a function introduced by R.

Since it's an R function, I can next search **@winston**'s GitHub mirror of the R sources: https://github.com/wch/r-source 16, finding the source at https://github.com/wch/r-source/blob/af7f52f70101960861e5d995d3a4bec010bc89e6/src/library/utils/R/head.R 20

This includes a comment which suggest we should ask Patrick Burns:

```
### placed in the public domain 2002
### Patrick Burns patrick@burns-stat.com
###
### Adapted for negative arguments by Vincent Goulet
### <vincent.goulet@act.ulaval.ca>, 2006
```

But it's worth checking just to make sure it's always used 6. I click on history, and then find the first version: https://github.com/wch/r-source/commit/37271cdbdcd7e5d82c79bdb536ef305d93b644ad#diff-941bf47bf09f67538338535bd512d521 28 - so it has been six from the very beginning.

So next step, I'll email Patrick and see if he recollects…

Look in S books

Go to R-source mirror

Find relevant bit

See the history

Send an email

Reading error messages

```
> install.packages('openssl')
* installing *source* package ‘openssl’ ...
** package ‘openssl’ successfully unpacked and MD5 sums checked
Using PKG_CFLAGS=
------------------------- ANTICONF ERROR ---------------------------
Configuration failed because openssl was not found. Try installing:
 * deb: libssl-dev (Debian, Ubuntu, etc)
 * rpm: openssl-devel (Fedora, CentOS, RHEL)
 * csw: libssl_dev (Solaris)
 * brew: openssl@1.1 (Mac OSX)
If openssl is already installed, check that 'pkg-config' is in your
PATH and PKG_CONFIG_PATH contains a openssl.pc file. If pkg-config
is unavailable you can set INCLUDE_DIR and LIB_DIR manually via:
R CMD INSTALL --configure-vars='INCLUDE_DIR=... LIB_DIR=...'
--------------------------------------------------------------------
ERROR: configuration failed for package ‘openssl’
* removing ‘/R/R-3.5.0_SL/lib64/R/library/openssl’

The downloaded source packages are in
        ‘/tmp/RtmpCM5CD4/downloaded_packages’
Warning message:
In install.packages("openssl") :
  installation of package ‘openssl’ had non-zero exit status
```

Super Useful Information

Reading error messages

```
> install.packages('openssl')
* installing *source* package ‘openssl’ ...
** package ‘openssl’ successfully unpacked and MD5 sums checked
Using PKG_CFLAGS=
------------------------- ANTICONF ERROR ---------------------------
Configuration failed because openssl was not found. Try installing:
 * deb: libssl-dev (Debian, Ubuntu, etc)
 * rpm: openssl-devel (Fedora, CentOS, RHEL)
 * csw: libssl_dev (Solaris)
 * brew: openssl@1.1 (Mac OSX)
If openssl is already installed, check that 'pkg-config' is in your
PATH and PKG_CONFIG_PATH contains a openssl.pc file. If pkg-config
is unavailable you can set INCLUDE_DIR and LIB_DIR manually via:
R CMD INSTALL --configure-vars='INCLUDE_DIR=... LIB_DIR=...'
--------------------------------------------------------------------
ERROR: configuration failed for package ‘openssl’
* removing ‘/R/R-3.5.0_SL/lib64/R/library/openssl’

The downloaded source packages are in
        ‘/tmp/RtmpCM5CD4/downloaded_packages’
Warning message:
In install.packages("openssl") :
  installation of package ‘openssl’ had non-zero exit status
```

Super Useful Information

Per the error message…

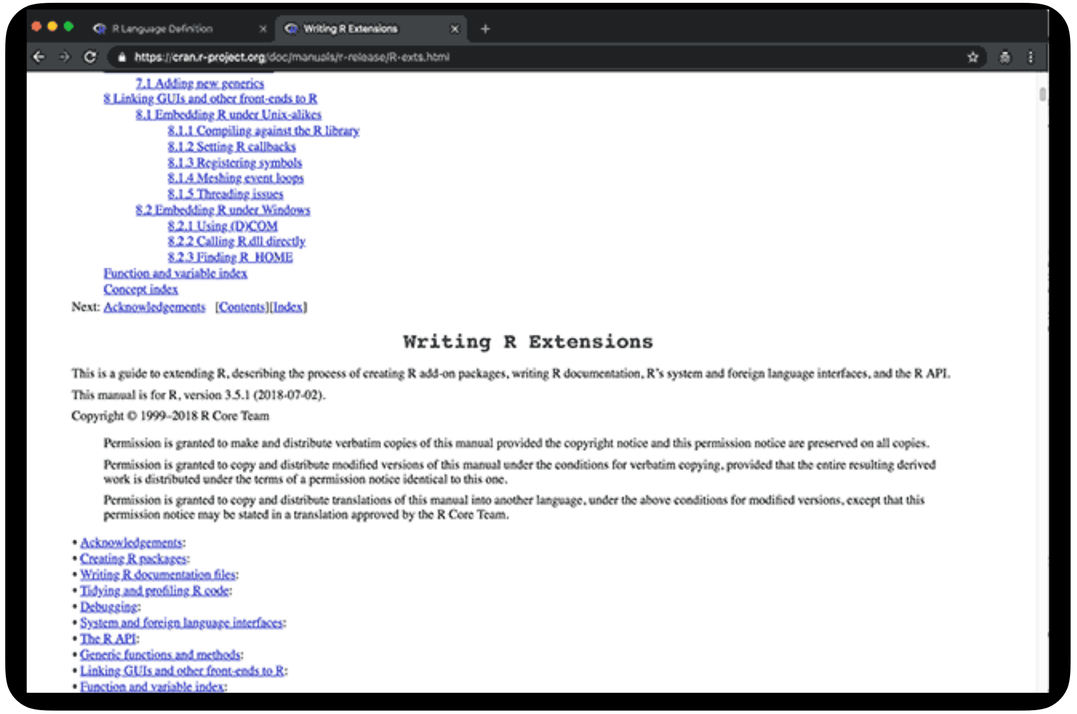
This may literally be the first time they've seen a useful error message

RTFM

The R Core Team "Writing R Extensions." R, version 3.5.1 (2018-07-02). https://cran.r-project.org/doc/manuals/r-release/R-exts.html Copyright © 1999-2018 R Core Team

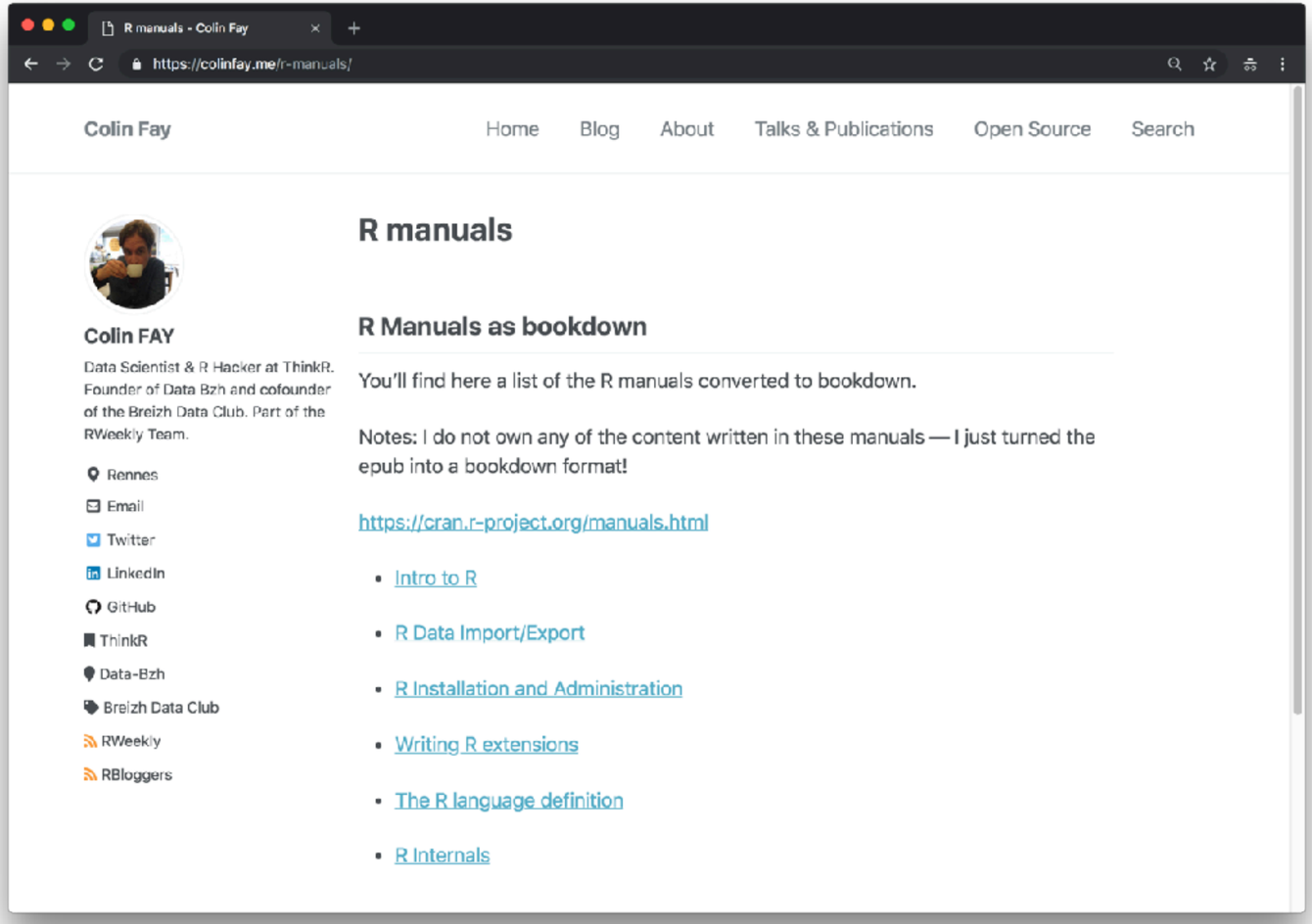
RTFM

TFM



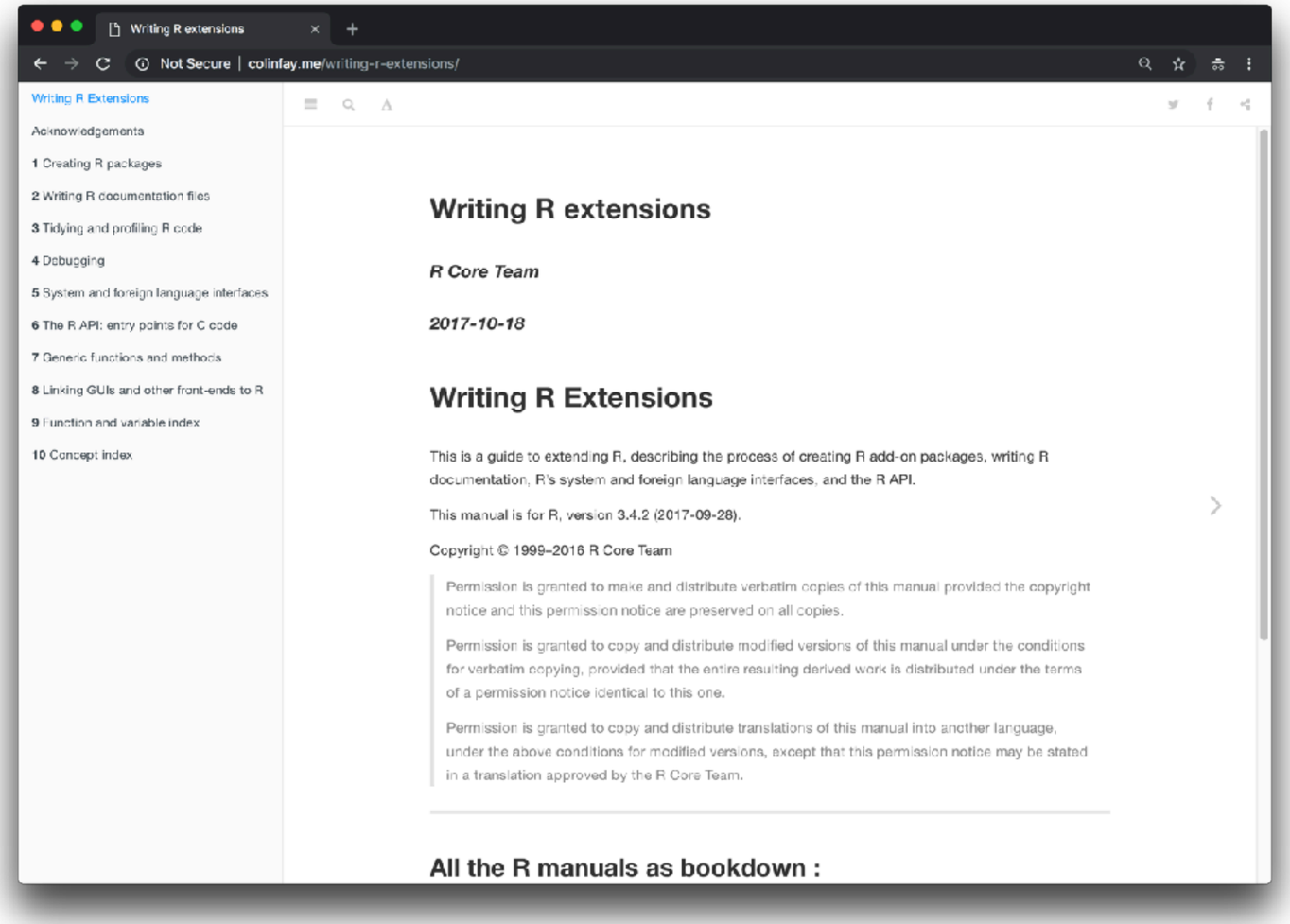
The R Core Team "Writing R Extensions." R, version 3.5.1 (2018-07-02). https://cran.r-project.org/doc/manuals/r-release/R-exts.html Copyright © 1999-2018 R Core Team

The R manuals as bookdown



Colin Fay https://colinfay.me/r-manuals/

The R manuals as bookdown



Colin Fay https://colinfay.me/r-manuals/

Guide them through the docs

Tidyr::separate() at second/last occurence of character

As you guessed, a regular expression is your best bet here. From the tidyr `separate()` function reference 8

> Given either regular expression or a vector of character positions, `separate()` turns a single character column into multiple columns.

workflow all in one place:
https://www.garrickadenbuie.com/project/regexplain/ 30

Here's how I would do it (which is hacky, but it works); you want to split at the underscore that precedes the day, which is numerical. So, I would use a stringr look around 18 to say, in effect, "*split at the underscore that is followed by a digit.*"

```
suppressPackageStartupMessages(library(tidyverse))
example_data <- data.frame(subject_ID = 1:5,
                           daily_measure_1 = 6:10,
                           daily_measure_2 = 11:15,
                           daily_measure_3 = 16:20,
                           daily_measure_4 = 21:25,
                           daily_measure_5 = 26:30)
example_data %>%
  gather(key = "full_name", value = "value", starts_with("daily_")) %>%
  separate(full_name, c("variable", "day"), sep = "_(?=[:digit:])")
#>    subject_ID      variable day value
#> 1           1 daily_measure   1     6
#> 2           2 daily_measure   1     7
#> 3           3 daily_measure   1     8
#> 4           4 daily_measure   1     9
#> 5           5 daily_measure   1    10
```

Tidyr::separate() at second/last occurence of character

tidyverse tidyr regex

mara Sustainer May 20

As you guessed, a regular expression is your best bet here. From the tidyr `separate()` function reference 8

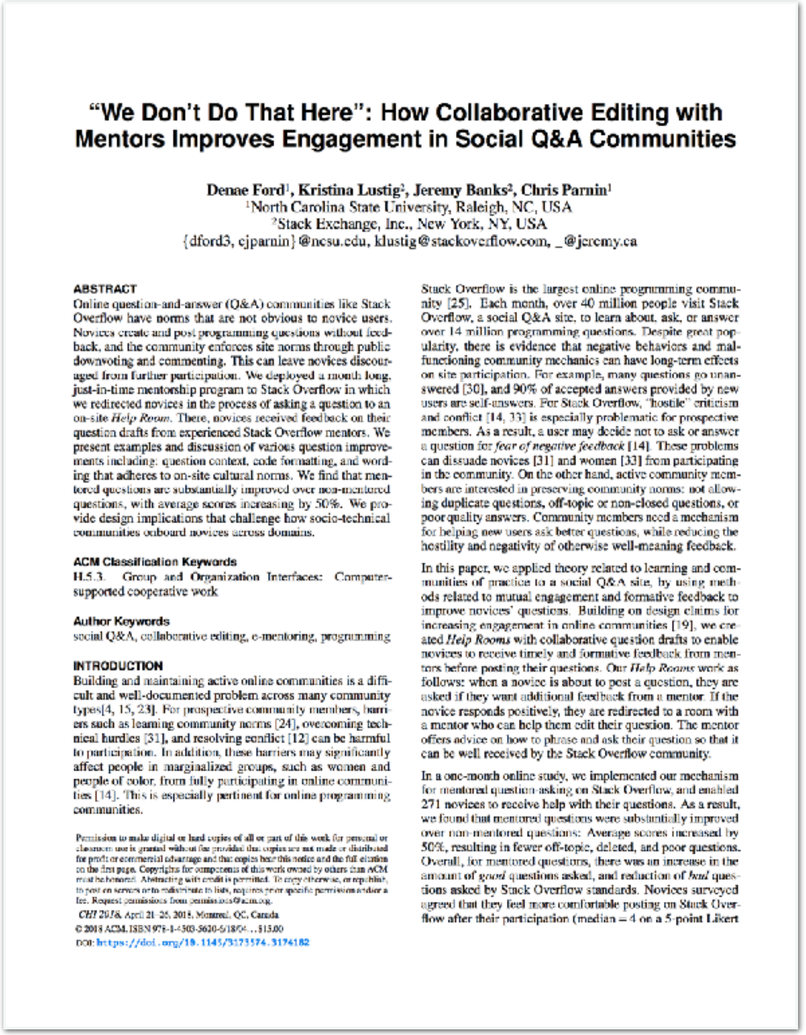
RegExr.com 21 has been my go-to for regular expression testing for a while, but Garrick Aden-Buie recently made an RStudio add-in inspired by the very same, RegExplain, which is a great option for keeping your workflow all in one place:
https://www.garrickadenbuie.com/project/regexplain/ 30

Here's how I would do it (which is hacky, but it works); you want to split at the underscore that precedes the day, which is numerical. So, I would use a stringr look around 18 to say, in effect, "*split at the underscore that is followed by a digit.*"

```
                          daily_measure_4 = 21:25,
                          daily_measure_5 = 26:30)
example_data %>%
  gather(key = "full_name", value = "value", starts_with("daily_")) %>%
  separate(full_name, c("variable", "day"), sep = "_(?=[:digit:])")
#>    subject_ID      variable day value
#> 1           1 daily_measure   1     6
#> 2           2 daily_measure   1     7
#> 3           3 daily_measure   1     8
#> 4           4 daily_measure   1     9
#> 5           5 daily_measure   1    10
```

"We don't do that here"

- Question phrasing
- Formatting posts
- Community triage
- Question framing
- Community culture of asking



"We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities

Denae Ford[1], Kristina Lustig[2], Jeremy Banks[2], Chris Parnin[1]
[1]North Carolina State University, Raleigh, NC, USA
[2]Stack Exchange, Inc., New York, NY, USA
{dford3, cjparnin}@ncsu.edu, klustig@stackoverflow.com, _@jeremy.ca

ABSTRACT
Online question-and-answer (Q&A) communities like Stack Overflow have norms that are not obvious to novice users. Novices create and post programming questions without feedback, and the community enforces site norms through public downvoting and commenting. This can leave novices discouraged from further participation. We deployed a month long, just-in-time mentorship program to Stack Overflow in which we redirected novices in the process of asking a question to an on-site *Help Room*. There, novices received feedback on their question drafts from experienced Stack Overflow mentors. We present examples and discussion of various question improvements including: question context, code formatting, and wording that adheres to on-site cultural norms. We find that mentored questions are substantially improved over non-mentored questions, with average scores increasing by 50%. We provide design implications that challenge how socio-technical communities onboard novices across domains.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported cooperative work

Author Keywords
social Q&A, collaborative editing, e-mentoring, programming

INTRODUCTION
Building and maintaining active online communities is a difficult and well-documented problem across many community types[4, 15, 23]. For prospective community members, barriers such as learning community norms [24], overcoming technical hurdles [31], and resolving conflict [12] can be harmful to participation. In addition, these barriers may significantly affect people in marginalized groups, such as women and people of color, from fully participating in online communities [14]. This is especially pertinent for online programming communities.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

DOI: https://doi.org/10.1145/3173574.3174182

Stack Overflow is the largest online programming community [25]. Each month, over 40 million people visit Stack Overflow, a social Q&A site, to learn about, ask, or answer over 14 million programming questions. Despite great popularity, there is evidence that negative behaviors and malfunctioning community mechanics can have long-term effects on site participation. For example, many questions go unanswered [30], and 90% of accepted answers provided by new users are self-answers. For Stack Overflow, "hostile" criticism and conflict [14, 33] is especially problematic for prospective members. As a result, a user may decide not to ask or answer a question for *fear of negative feedback* [14]. These problems can dissuade novices [31] and women [33] from participating in the community. On the other hand, active community members are interested in preserving community norms: not allowing duplicate questions, off-topic or non-closed questions, or poor quality answers. Community members need a mechanism for helping new users ask better questions, while reducing the hostility and negativity of otherwise well-meaning feedback.

In this paper, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback to improve novices' questions. Building on design claims for increasing engagement in online communities [19], we created *Help Rooms* with collaborative question drafts to enable novices to receive timely and formative feedback from mentors before posting their questions. Our *Help Rooms* work as follows: when a novice is about to post a question, they are asked if they want additional feedback from a mentor. If the novice responds positively, they are redirected to a room with a mentor who can help them edit their question. The mentor offers advice on how to phrase and ask their question so that it can be well received by the Stack Overflow community.

In a one-month online study, we implemented our mechanism for mentored question-asking on Stack Overflow, and enabled 271 novices to receive help with their questions. As a result, we found that mentored questions were substantially improved over non-mentored questions: Average scores increased by 50%, resulting in fewer off-topic, deleted, and poor questions. Overall, for mentored questions, there was an increase in the amount of *good* questions asked, and reduction of *bad* questions asked by Stack Overflow standards. Novices surveyed agreed that they feel more comfortable posting on Stack Overflow after their participation (median = 4 on a 5-point Likert

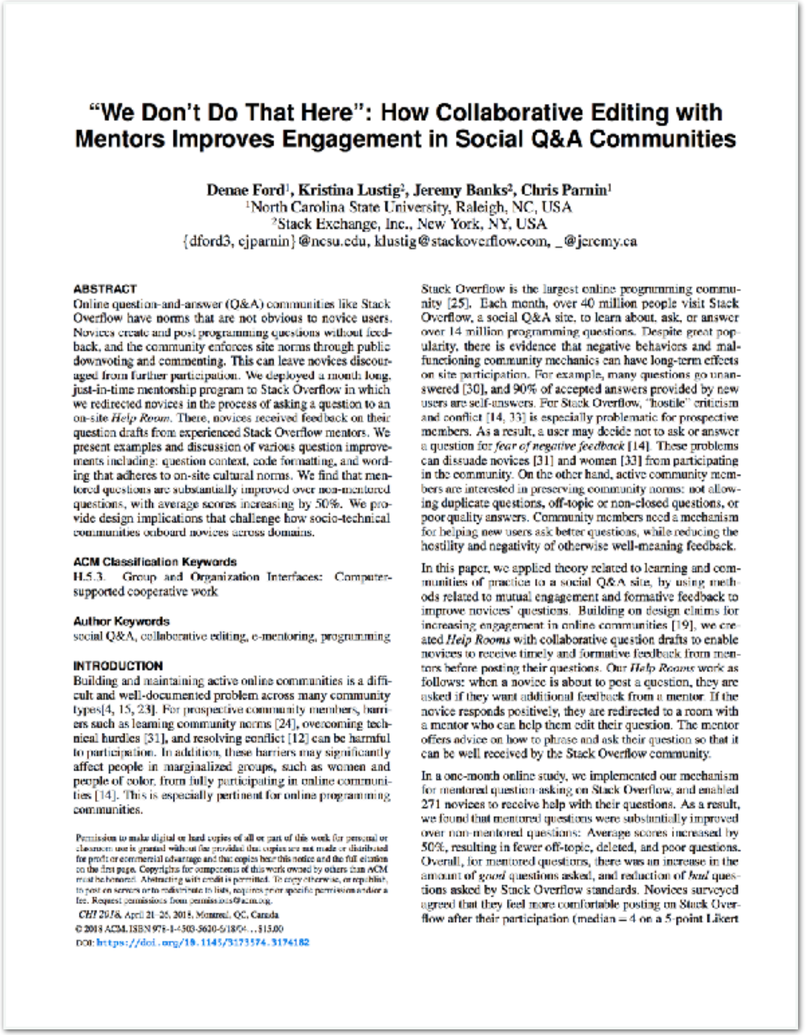
Ford, Denae, Kristina Lustig, Jeremy Banks, Chris Parnin, and North Carolina. 2018. "'We Don't Do That Here': How Collaborative Editing with Mentors Improves Engagement in Social Q & A Communities." In CHI 2018. Montreal, QC, Canada: ACM. doi:10.1145/3173574.3174182.

"We don't do that here"

- Question phrasing

title = your chance to reel 'em in

- Formatting posts
- Community triage
- Question framing
- Community culture of asking



"We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities

Denae Ford[1], **Kristina Lustig**[2], **Jeremy Banks**[2], **Chris Parnin**[1]
[1]North Carolina State University, Raleigh, NC, USA
[2]Stack Exchange, Inc., New York, NY, USA
{dford3, cjparnin}@ncsu.edu, klustig@stackoverflow.com, _@jeremy.ca

ABSTRACT
Online question-and-answer (Q&A) communities like Stack Overflow have norms that are not obvious to novice users. Novices create and post programming questions without feedback, and the community enforces site norms through public downvoting and commenting. This can leave novices discouraged from further participation. We deployed a month long, just-in-time mentorship program to Stack Overflow in which we redirected novices in the process of asking a question to an on-site *Help Room*. There, novices received feedback on their question drafts from experienced Stack Overflow mentors. We present examples and discussion of various question improvements including: question context, code formatting, and wording that adheres to on-site cultural norms. We find that mentored questions are substantially improved over non-mentored questions, with average scores increasing by 50%. We provide design implications that challenge how socio-technical communities onboard novices across domains.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported cooperative work

Author Keywords
social Q&A, collaborative editing, e-mentoring, programming

INTRODUCTION
Building and maintaining active online communities is a difficult and well-documented problem across many community types[4, 15, 23]. For prospective community members, barriers such as learning community norms [24], overcoming technical hurdles [31], and resolving conflict [12] can be harmful to participation. In addition, these barriers may significantly affect people in marginalized groups, such as women and people of color, from fully participating in online communities [14]. This is especially pertinent for online programming communities.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

DOI: https://doi.org/10.1145/3173574.3174182

Stack Overflow is the largest online programming community [25]. Each month, over 40 million people visit Stack Overflow, a social Q&A site, to learn about, ask, or answer over 14 million programming questions. Despite great popularity, there is evidence that negative behaviors and malfunctioning community mechanics can have long-term effects on site participation. For example, many questions go unanswered [30], and 90% of accepted answers provided by new users are self-answers. For Stack Overflow, "hostile" criticism and conflict [14, 33] is especially problematic for prospective members. As a result, a user may decide not to ask or answer a question for *fear of negative feedback* [14]. These problems can dissuade novices [31] and women [33] from participating in the community. On the other hand, active community members are interested in preserving community norms: not allowing duplicate questions, off-topic or non-closed questions, or poor quality answers. Community members need a mechanism for helping new users ask better questions, while reducing the hostility and negativity of otherwise well-meaning feedback.

In this paper, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback to improve novices' questions. Building on design claims for increasing engagement in online communities [19], we created *Help Rooms* with collaborative question drafts to enable novices to receive timely and formative feedback from mentors before posting their questions. Our *Help Rooms* work as follows: when a novice is about to post a question, they are asked if they want additional feedback from a mentor. If the novice responds positively, they are redirected to a room with a mentor who can help them edit their question. The mentor offers advice on how to phrase and ask their question so that it can be well received by the Stack Overflow community.

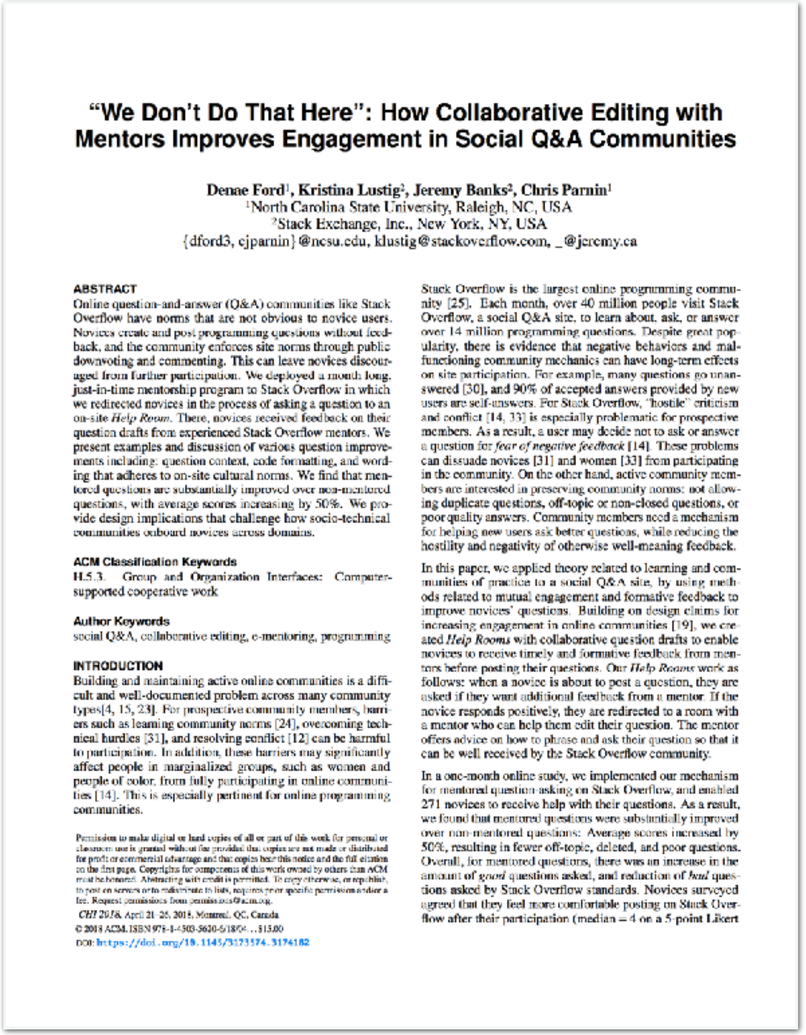
In a one-month online study, we implemented our mechanism for mentored question-asking on Stack Overflow, and enabled 271 novices to receive help with their questions. As a result, we found that mentored questions were substantially improved over non-mentored questions: Average scores increased by 50%, resulting in fewer off-topic, deleted, and poor questions. Overall, for mentored questions, there was an increase in the amount of *good* questions asked, and reduction of *bad* questions asked by Stack Overflow standards. Novices surveyed agreed that they feel more comfortable posting on Stack Overflow after their participation (median = 4 on a 5-point Likert

Ford, Denae, Kristina Lustig, Jeremy Banks, Chris Parnin, and North Carolina. 2018. "'We Don't Do That Here': How Collaborative Editing with Mentors Improves Engagement in Social Q & A Communities." In CHI 2018. Montreal, QC, Canada: ACM. doi:10.1145/3173574.3174182.

"We don't do that here"

- Question phrasing
- Formatting posts
- Community triage
- Question framing
- Community culture of asking

format code as code



"We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities

Denae Ford[1], Kristina Lustig[2], Jeremy Banks[2], Chris Parnin[1]
[1]North Carolina State University, Raleigh, NC, USA
[2]Stack Exchange, Inc., New York, NY, USA
{dford3, cjparnin}@ncsu.edu, klustig@stackoverflow.com, _@jeremy.ca

ABSTRACT
Online question-and-answer (Q&A) communities like Stack Overflow have norms that are not obvious to novice users. Novices create and post programming questions without feedback, and the community enforces site norms through public downvoting and commenting. This can leave novices discouraged from further participation. We deployed a month long, just-in-time mentorship program to Stack Overflow in which we redirected novices in the process of asking a question to an on-site *Help Room*. There, novices received feedback on their question drafts from experienced Stack Overflow mentors. We present examples and discussion of various question improvements including: question context, code formatting, and wording that adheres to on-site cultural norms. We find that mentored questions are substantially improved over non-mentored questions, with average scores increasing by 50%. We provide design implications that challenge how socio-technical communities onboard novices across domains.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported cooperative work

Author Keywords
social Q&A, collaborative editing, e-mentoring, programming

INTRODUCTION
Building and maintaining active online communities is a difficult and well-documented problem across many community types[4, 15, 23]. For prospective community members, barriers such as learning community norms [24], overcoming technical hurdles [31], and resolving conflict [12] can be harmful to participation. In addition, these barriers may significantly affect people in marginalized groups, such as women and people of color, from fully participating in online communities [14]. This is especially pertinent for online programming communities.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

DOI: https://doi.org/10.1145/3173574.3174182

Stack Overflow is the largest online programming community [25]. Each month, over 40 million people visit Stack Overflow, a social Q&A site, to learn about, ask, or answer over 14 million programming questions. Despite great popularity, there is evidence that negative behaviors and malfunctioning community mechanics can have long-term effects on site participation. For example, many questions go unanswered [30], and 90% of accepted answers provided by new users are self-answers. For Stack Overflow, "hostile" criticism and conflict [14, 33] is especially problematic for prospective members. As a result, a user may decide not to ask or answer a question for *fear of negative feedback* [14]. These problems can dissuade novices [31] and women [33] from participating in the community. On the other hand, active community members are interested in preserving community norms: not allowing duplicate questions, off-topic or non-closed questions, or poor quality answers. Community members need a mechanism for helping new users ask better questions, while reducing the hostility and negativity of otherwise well-meaning feedback.

In this paper, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback to improve novices' questions. Building on design claims for increasing engagement in online communities [19], we created *Help Rooms* with collaborative question drafts to enable novices to receive timely and formative feedback from mentors before posting their questions. Our *Help Rooms* work as follows: when a novice is about to post a question, they are asked if they want additional feedback from a mentor. If the novice responds positively, they are redirected to a room with a mentor who can help them edit their question. The mentor offers advice on how to phrase and ask their question so that it can be well received by the Stack Overflow community.

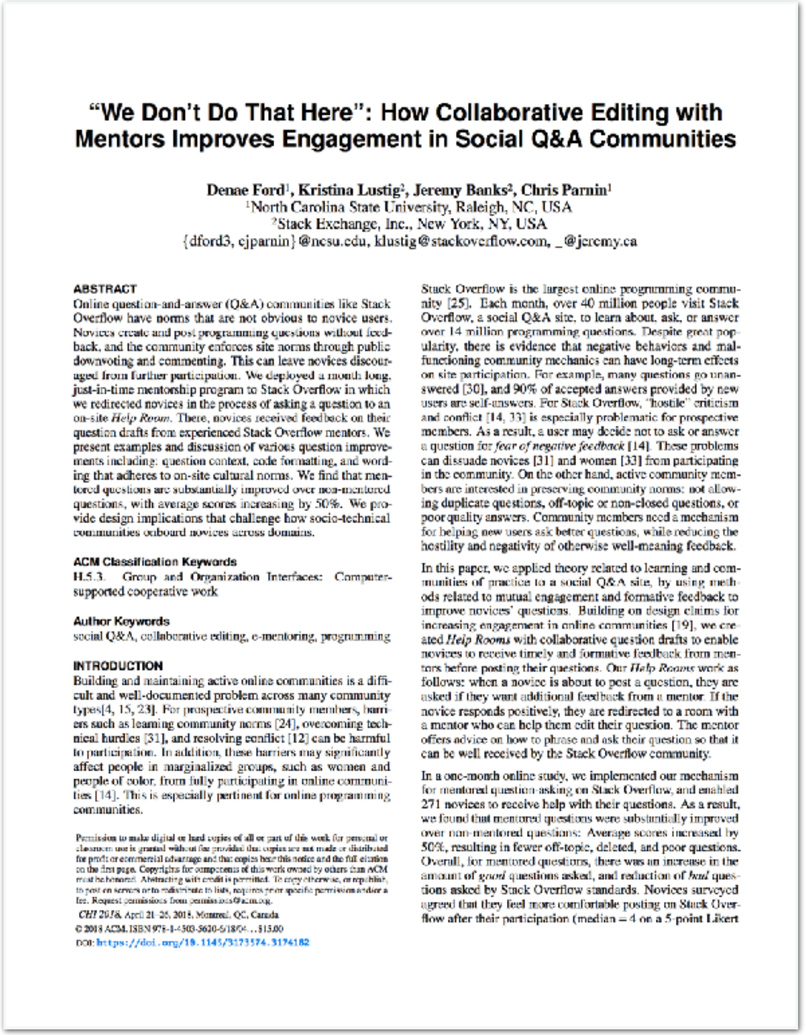
In a one-month online study, we implemented our mechanism for mentored question-asking on Stack Overflow, and enabled 271 novices to receive help with their questions. As a result, we found that mentored questions were substantially improved over non-mentored questions: Average scores increased by 50%, resulting in fewer off-topic, deleted, and poor questions. Overall, for mentored questions, there was an increase in the amount of *good* questions asked, and reduction of *bad* questions asked by Stack Overflow standards. Novices surveyed agreed that they feel more comfortable posting on Stack Overflow after their participation (median = 4 on a 5-point Likert

Ford, Denae, Kristina Lustig, Jeremy Banks, Chris Parnin, and North Carolina. 2018. "'We Don't Do That Here': How Collaborative Editing with Mentors Improves Engagement in Social Q & A Communities." In CHI 2018. Montreal, QC, Canada: ACM. doi:10.1145/3173574.3174182.

"We don't do that here"

- Question phrasing
- Formatting posts
- Community triage
- Question framing
- Community culture of asking

is this an appropriate place for your Q?



"We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities

Denae Ford[1], Kristina Lustig[2], Jeremy Banks[2], Chris Parnin[1]
[1]North Carolina State University, Raleigh, NC, USA
[2]Stack Exchange, Inc., New York, NY, USA
{dford3, cjparnin}@ncsu.edu, klustig@stackoverflow.com, _@jeremy.ca

ABSTRACT
Online question-and-answer (Q&A) communities like Stack Overflow have norms that are not obvious to novice users. Novices create and post programming questions without feedback, and the community enforces site norms through public downvoting and commenting. This can leave novices discouraged from further participation. We deployed a month long, just-in-time mentorship program to Stack Overflow in which we redirected novices in the process of asking a question to an on-site *Help Room*. There, novices received feedback on their question drafts from experienced Stack Overflow mentors. We present examples and discussion of various question improvements including: question context, code formatting, and wording that adheres to on-site cultural norms. We find that mentored questions are substantially improved over non-mentored questions, with average scores increasing by 50%. We provide design implications that challenge how socio-technical communities onboard novices across domains.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported cooperative work

Author Keywords
social Q&A, collaborative editing, e-mentoring, programming

INTRODUCTION
Building and maintaining active online communities is a difficult and well-documented problem across many community types[4, 15, 23]. For prospective community members, barriers such as learning community norms [24], overcoming technical hurdles [31], and resolving conflict [12] can be harmful to participation. In addition, these barriers may significantly affect people in marginalized groups, such as women and people of color, from fully participating in online communities [14]. This is especially pertinent for online programming communities.

Stack Overflow is the largest online programming community [25]. Each month, over 40 million people visit Stack Overflow, a social Q&A site, to learn about, ask, or answer over 14 million programming questions. Despite great popularity, there is evidence that negative behaviors and malfunctioning community mechanics can have long-term effects on site participation. For example, many questions go unanswered [30], and 90% of accepted answers provided by new users are self-answers. For Stack Overflow, "hostile" criticism and conflict [14, 33] is especially problematic for prospective members. As a result, a user may decide not to ask or answer a question for *fear of negative feedback* [14]. These problems can dissuade novices [31] and women [33] from participating in the community. On the other hand, active community members are interested in preserving community norms: not allowing duplicate questions, off-topic or non-closed questions, or poor quality answers. Community members need a mechanism for helping new users ask better questions, while reducing the hostility and negativity of otherwise well-meaning feedback.

In this paper, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback to improve novices' questions. Building on design claims for increasing engagement in online communities [19], we created *Help Rooms* with collaborative question drafts to enable novices to receive timely and formative feedback from mentors before posting their questions. Our *Help Rooms* work as follows: when a novice is about to post a question, they are asked if they want additional feedback from a mentor. If the novice responds positively, they are redirected to a room with a mentor who can help them edit their question. The mentor offers advice on how to phrase and ask their question so that it can be well received by the Stack Overflow community.

In a one-month online study, we implemented our mechanism for mentored question-asking on Stack Overflow, and enabled 271 novices to receive help with their questions. As a result, we found that mentored questions were substantially improved over non-mentored questions: Average scores increased by 50%, resulting in fewer off-topic, deleted, and poor questions. Overall, for mentored questions, there was an increase in the amount of *good* questions asked, and reduction of *bad* questions asked by Stack Overflow standards. Novices surveyed agreed that they feel more comfortable posting on Stack Overflow after their participation (median = 4 on a 5-point Likert

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

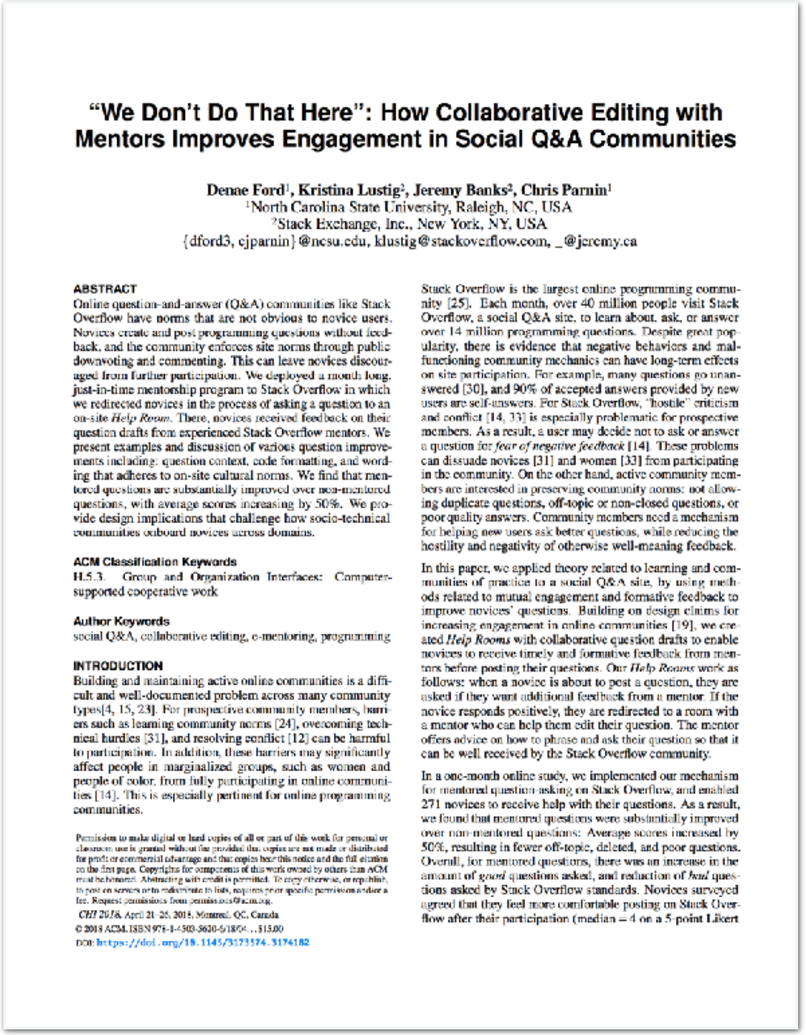
DOI: https://doi.org/10.1145/3173574.3174182

Ford, Denae, Kristina Lustig, Jeremy Banks, Chris Parnin, and North Carolina. 2018. "'We Don't Do That Here': How Collaborative Editing with Mentors Improves Engagement in Social Q & A Communities." In CHI 2018. Montreal, QC, Canada: ACM. doi:10.1145/3173574.3174182.

"We don't do that here"

- Question phrasing
- Formatting posts
- Community triage
- Question framing
- Community culture of asking

clarity, research of problem, context



"We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities

Denae Ford[1], Kristina Lustig[2], Jeremy Banks[2], Chris Parnin[1]
[1]North Carolina State University, Raleigh, NC, USA
[2]Stack Exchange, Inc., New York, NY, USA
{dford3, cjparnin}@ncsu.edu, klustig@stackoverflow.com, _@jeremy.ca

ABSTRACT
Online question-and-answer (Q&A) communities like Stack Overflow have norms that are not obvious to novice users. Novices create and post programming questions without feedback, and the community enforces site norms through public downvoting and commenting. This can leave novices discouraged from further participation. We deployed a month long, just-in-time mentorship program to Stack Overflow in which we redirected novices in the process of asking a question to an on-site *Help Room*. There, novices received feedback on their question drafts from experienced Stack Overflow mentors. We present examples and discussion of various question improvements including: question context, code formatting, and wording that adheres to on-site cultural norms. We find that mentored questions are substantially improved over non-mentored questions, with average scores increasing by 50%. We provide design implications that challenge how socio-technical communities onboard novices across domains.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported cooperative work

Author Keywords
social Q&A, collaborative editing, e-mentoring, programming

INTRODUCTION
Building and maintaining active online communities is a difficult and well-documented problem across many community types[4, 15, 23]. For prospective community members, barriers such as learning community norms [24], overcoming technical hurdles [31], and resolving conflict [12] can be harmful to participation. In addition, these barriers may significantly affect people in marginalized groups, such as women and people of color, from fully participating in online communities [14]. This is especially pertinent for online programming communities.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

DOI: https://doi.org/10.1145/3173574.3174182

Stack Overflow is the largest online programming community [25]. Each month, over 40 million people visit Stack Overflow, a social Q&A site, to learn about, ask, or answer over 14 million programming questions. Despite great popularity, there is evidence that negative behaviors and malfunctioning community mechanics can have long-term effects on site participation. For example, many questions go unanswered [30], and 90% of accepted answers provided by new users are self-answers. For Stack Overflow, "hostile" criticism and conflict [14, 33] is especially problematic for prospective members. As a result, a user may decide not to ask or answer a question for *fear of negative feedback* [14]. These problems can dissuade novices [31] and women [33] from participating in the community. On the other hand, active community members are interested in preserving community norms: not allowing duplicate questions, off-topic or non-closed questions, or poor quality answers. Community members need a mechanism for helping new users ask better questions, while reducing the hostility and negativity of otherwise well-meaning feedback.

In this paper, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback to improve novices' questions. Building on design claims for increasing engagement in online communities [19], we created *Help Rooms* with collaborative question drafts to enable novices to receive timely and formative feedback from mentors before posting their questions. Our *Help Rooms* work as follows: when a novice is about to post a question, they are asked if they want additional feedback from a mentor. If the novice responds positively, they are redirected to a room with a mentor who can help them edit their question. The mentor offers advice on how to phrase and ask their question so that it can be well received by the Stack Overflow community.

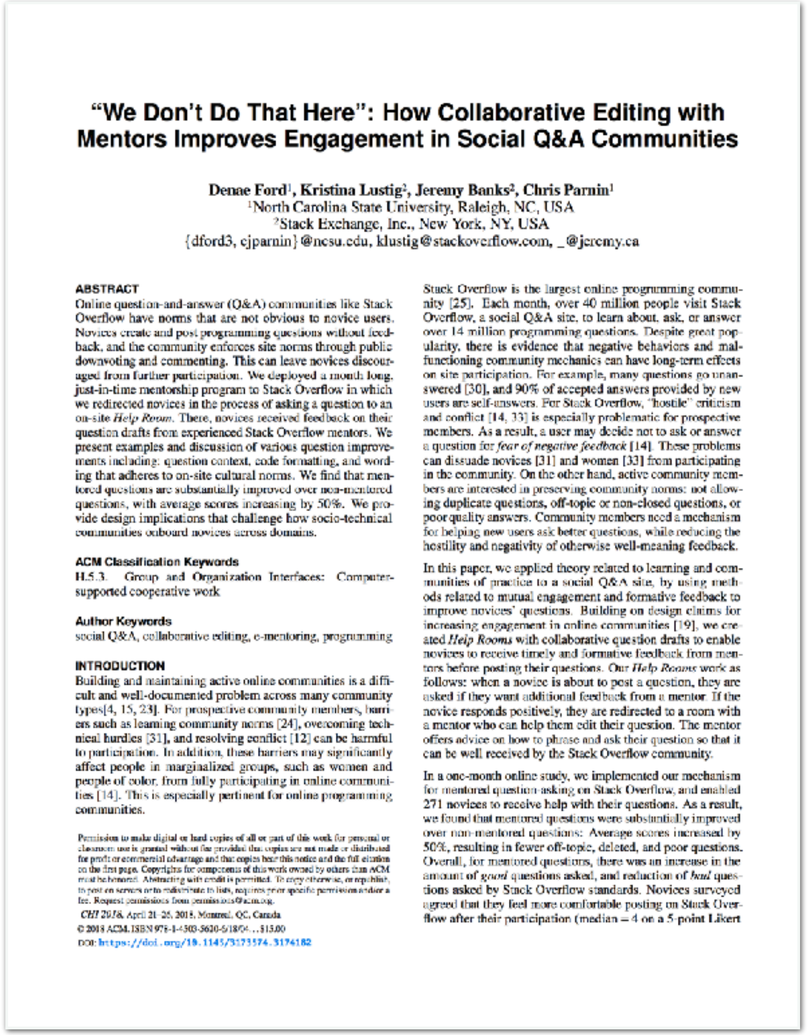
In a one-month online study, we implemented our mechanism for mentored question-asking on Stack Overflow, and enabled 271 novices to receive help with their questions. As a result, we found that mentored questions were substantially improved over non-mentored questions: Average scores increased by 50%, resulting in fewer off-topic, deleted, and poor questions. Overall, for mentored questions, there was an increase in the amount of *good* questions asked, and reduction of *bad* questions asked by Stack Overflow standards. Novices surveyed agreed that they feel more comfortable posting on Stack Overflow after their participation (median = 4 on a 5-point Likert

Ford, Denae, Kristina Lustig, Jeremy Banks, Chris Parnin, and North Carolina. 2018. "'We Don't Do That Here': How Collaborative Editing with Mentors Improves Engagement in Social Q & A Communities." In CHI 2018. Montreal, QC, Canada: ACM. doi:10.1145/3173574.3174182.

"We don't do that here"

- Question phrasing
- Formatting posts
- Community triage
- Question framing
- Community culture of asking

> *You also might want to edit out the "Thank you!" at the end. I know it seems polite, but people object to it on Stack Overflow.*



"We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities

Denae Ford[1], Kristina Lustig[2], Jeremy Banks[2], Chris Parnin[1]
[1]North Carolina State University, Raleigh, NC, USA
[2]Stack Exchange, Inc., New York, NY, USA
{dford3, cjparnin}@ncsu.edu, klustig@stackoverflow.com, _@jeremy.ca

ABSTRACT
Online question-and-answer (Q&A) communities like Stack Overflow have norms that are not obvious to novice users. Novices create and post programming questions without feedback, and the community enforces site norms through public downvoting and commenting. This can leave novices discouraged from further participation. We deployed a month long, just-in-time mentorship program to Stack Overflow in which we redirected novices in the process of asking a question to an on-site *Help Room*. There, novices received feedback on their question drafts from experienced Stack Overflow mentors. We present examples and discussion of various question improvements including: question context, code formatting, and wording that adheres to on-site cultural norms. We find that mentored questions are substantially improved over non-mentored questions, with average scores increasing by 50%. We provide design implications that challenge how socio-technical communities onboard novices across domains.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported cooperative work

Author Keywords
social Q&A, collaborative editing, e-mentoring, programming

INTRODUCTION
Building and maintaining active online communities is a difficult and well-documented problem across many community types[4, 15, 23]. For prospective community members, barriers such as learning community norms [24], overcoming technical hurdles [31], and resolving conflict [12] can be harmful to participation. In addition, these barriers may significantly affect people in marginalized groups, such as women and people of color, from fully participating in online communities [14]. This is especially pertinent for online programming communities.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

DOI: https://doi.org/10.1145/3173574.3174182

Stack Overflow is the largest online programming community [25]. Each month, over 40 million people visit Stack Overflow, a social Q&A site, to learn about, ask, or answer over 14 million programming questions. Despite great popularity, there is evidence that negative behaviors and malfunctioning community mechanics can have long-term effects on site participation. For example, many questions go unanswered [30], and 90% of accepted answers provided by new users are self-answers. For Stack Overflow, "hostile" criticism and conflict [14, 33] is especially problematic for prospective members. As a result, a user may decide not to ask or answer a question for *fear of negative feedback* [14]. These problems can dissuade novices [31] and women [33] from participating in the community. On the other hand, active community members are interested in preserving community norms: not allowing duplicate questions, off-topic or non-closed questions, or poor quality answers. Community members need a mechanism for helping new users ask better questions, while reducing the hostility and negativity of otherwise well-meaning feedback.

In this paper, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback to improve novices' questions. Building on design claims for increasing engagement in online communities [19], we created *Help Rooms* with collaborative question drafts to enable novices to receive timely and formative feedback from mentors before posting their questions. Our *Help Rooms* work as follows: when a novice is about to post a question, they are asked if they want additional feedback from a mentor. If the novice responds positively, they are redirected to a room with a mentor who can help them edit their question. The mentor offers advice on how to phrase and ask their question so that it can be well received by the Stack Overflow community.

In a one-month online study, we implemented our mechanism for mentored question-asking on Stack Overflow, and enabled 271 novices to receive help with their questions. As a result, we found that mentored questions were substantially improved over non-mentored questions: Average scores increased by 50%, resulting in fewer off-topic, deleted, and poor questions. Overall, for mentored questions, there was an increase in the amount of *good* questions asked, and reduction of *bad* questions asked by Stack Overflow standards. Novices surveyed agreed that they feel more comfortable posting on Stack Overflow after their participation (median = 4 on a 5-point Likert

Ford, Denae, Kristina Lustig, Jeremy Banks, Chris Parnin, and North Carolina. 2018. "'We Don't Do That Here': How Collaborative Editing with Mentors Improves Engagement in Social Q & A Communities." In CHI 2018. Montreal, QC, Canada: ACM. doi:10.1145/3173574.3174182.

Avoid the "data dump"

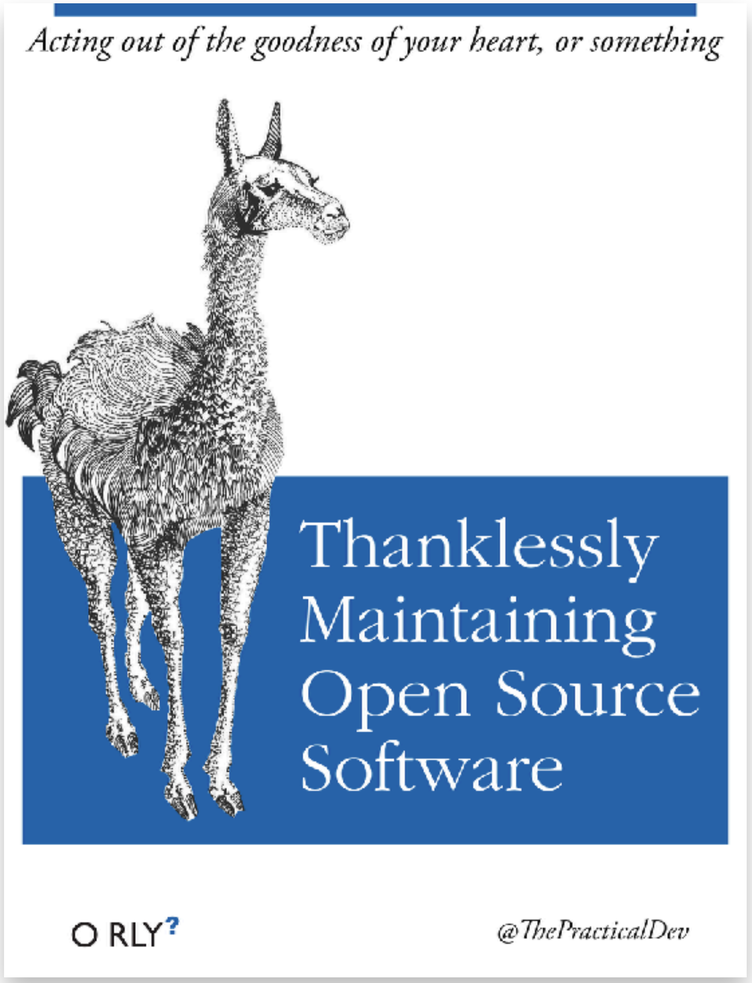
what I think I'm doing

Avoid the "data dump"

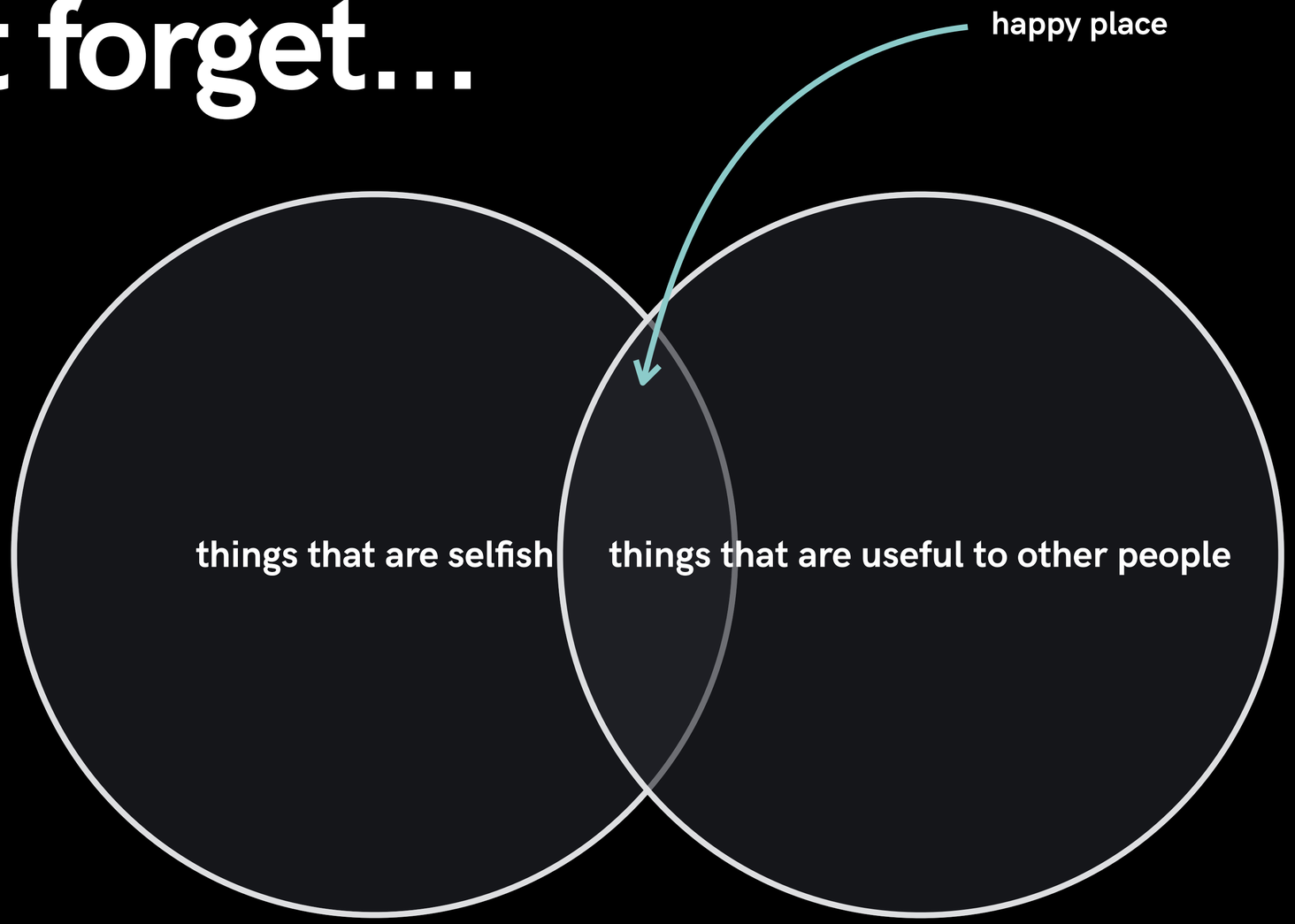
what they think I'm doing…

The source code…

giving back


Acting out of the goodness of your heart, or something
Thanklessly Maintaining Open Source Software
O RLY?
@ThePracticalDev

Don't forget...


happy place
things that are selfish
things that are useful to other people

Thank You

bit.ly/noreastr

GitHub surfing links

- https://help.github.com/articles/about-searching-on-github/
- https://help.github.com/articles/searching-code/
- http://stat545.com/bit006_github-browsability-wins.html
- https://help.github.com/articles/understanding-the-search-syntax/
- https://help.github.com/articles/searching-on-github/

Presentation matters

Thanks, Colin Fay

and R Core, obviously

http://colinfay.me/r-language-definition/

http://colinfay.me/writing-r-extensions/

http://colinfay.me/r-data-import-export/

http://colinfay.me/intro-to-r/