Selected Highlights RStudio Conference 2018

Feb. 2-3, San Diego

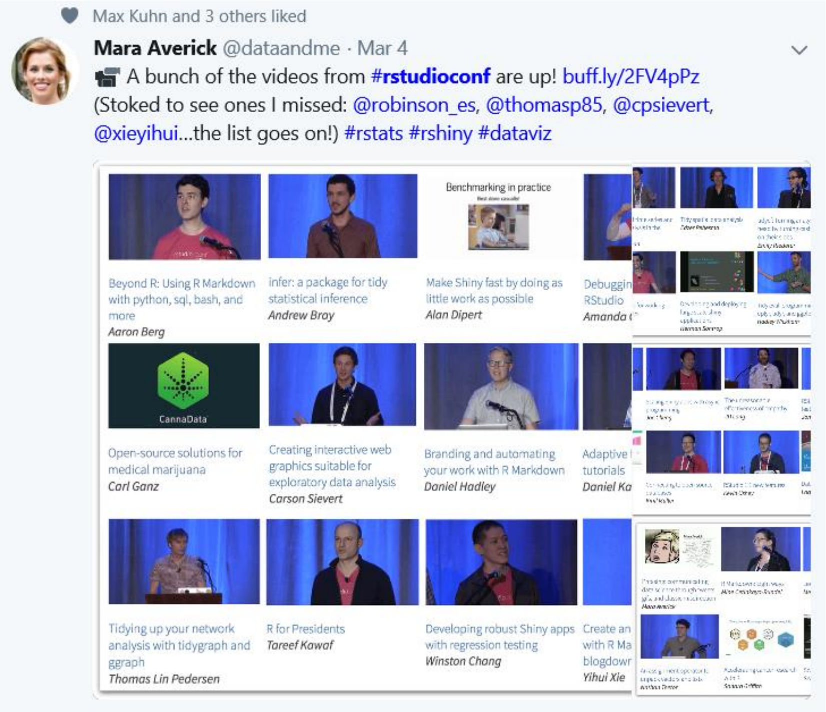
Earl F Glynn

Kansas City R Users Group

March 10, 2018

Selected Highlights of RStudio Conference 2018

- Selected videos: https://www.rstudio.com/resources/webinars/ (all talks should be online eventually; most are 20 minutes long)
- Topics for future KC R Users Group Discussions?



Max Kuhn and 3 others liked

Mara Averick @dataandme · Mar 4

A bunch of the videos from #rstudioconf are up! buff.ly/2FV4pPz (Stoked to see ones I missed: @robinson_es, @thomasp85, @cpsievert, @xieyihui...the list goes on!) #rstats #rshiny #dataviz

Beyond R: Using R Markdown with python, sql, bash, and more
Aaron Berg

infer: a package for tidy statistical inference
Andrew Bray

Make Shiny fast by doing as little work as possible
Alan Dipert

Debuggin RStudio
Amanda

Open-source solutions for medical marijuana
Carl Ganz

Creating interactive web graphics suitable for exploratory data analysis
Carson Sievert

Branding and automating your work with R Markdown
Daniel Hadley

Adaptive tutorials
Daniel Ka

Tidying up your network analysis with tidygraph and ggraph
Thomas Lin Pedersen

R for Presidents
Tareef Kawaf

Developing robust Shiny apps with regression testing
Winston Chang

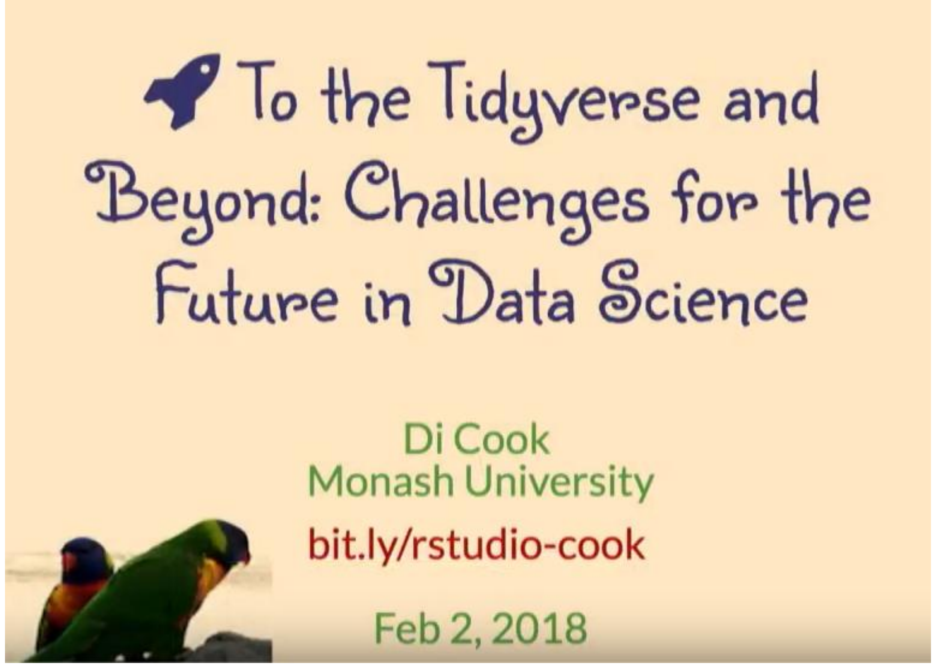
Create an with R Ma blogdown
Yihui Xie

RStudio Conference 2018

- Full conference schedule: rstudio::conf 2018 talks https://beta.rstudioconnect.com/content/3105/
- Day 2 Live Stream: https://www.youtube.com/watch?v=Ol1FjFR2IMU
- Links to slides from rstudio::conf 2018 https://github.com/simecek/RStudioConf2018Slides
- All materials for rstudio::conf2018 https://github.com/rstudio/rstudio-conf/tree/master/2018

Slides: http://www.dicook.org/files/rstudio/

Keynote Day 1



To the Tidyverse and Beyond: Challenges for the Future in Data Science

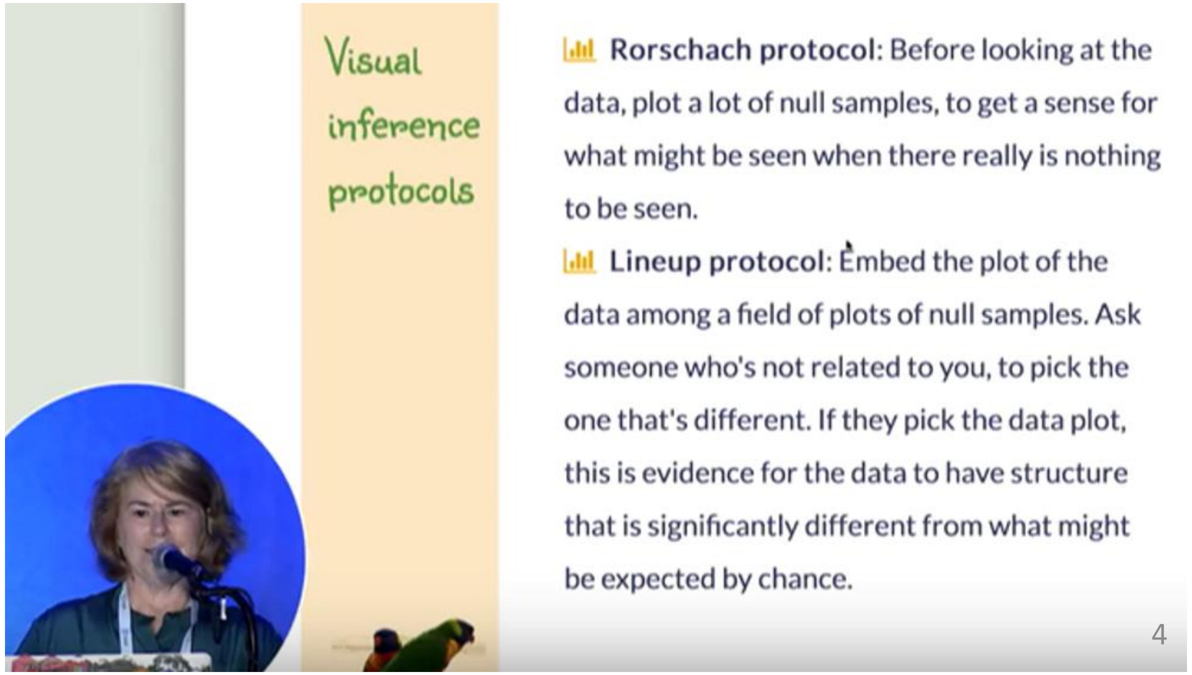
Di Cook
Monash University
bit.ly/rstudio-cook

Feb 2, 2018

Because we make inference with plots anyway, without a firm foundation.

https://github.com/dicook/nullabor

For plots, we can compare the data plot with null plots of samples where, by construction, there really is nothing going on.

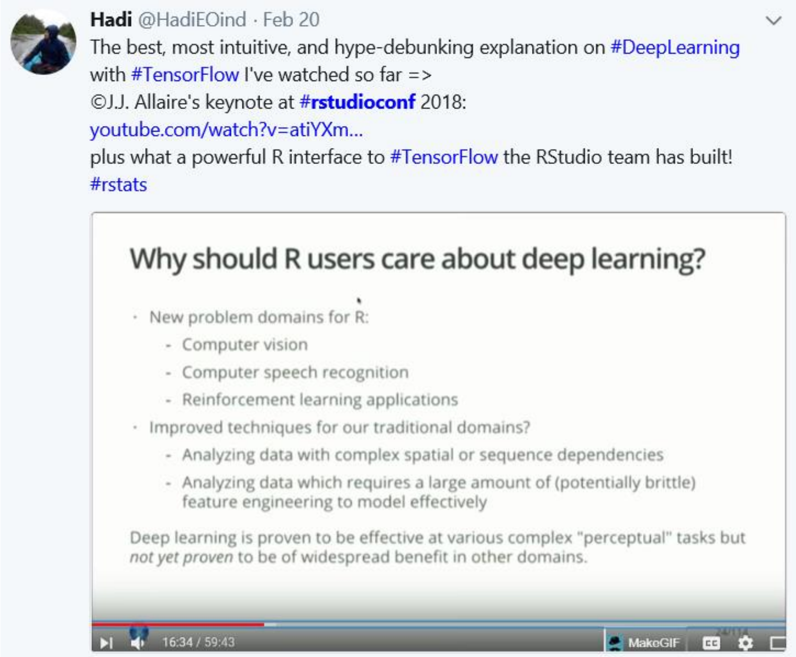


Visual inference protocols

- Rorschach protocol: Before looking at the data, plot a lot of null samples, to get a sense for what might be seen when there really is nothing to be seen.
- Lineup protocol: Embed the plot of the data among a field of plots of null samples. Ask someone who's not related to you, to pick the one that's different. If they pick the data plot, this is evidence for the data to have structure that is significantly different from what might be expected by chance.

https://www.rstudio.com/resources/videos/machine-learning-with-tensorflow-and-r/
https://beta.rstudioconnect.com/ml-with-tensorflow-and-r/ (113 slides)

Keynote Day 2



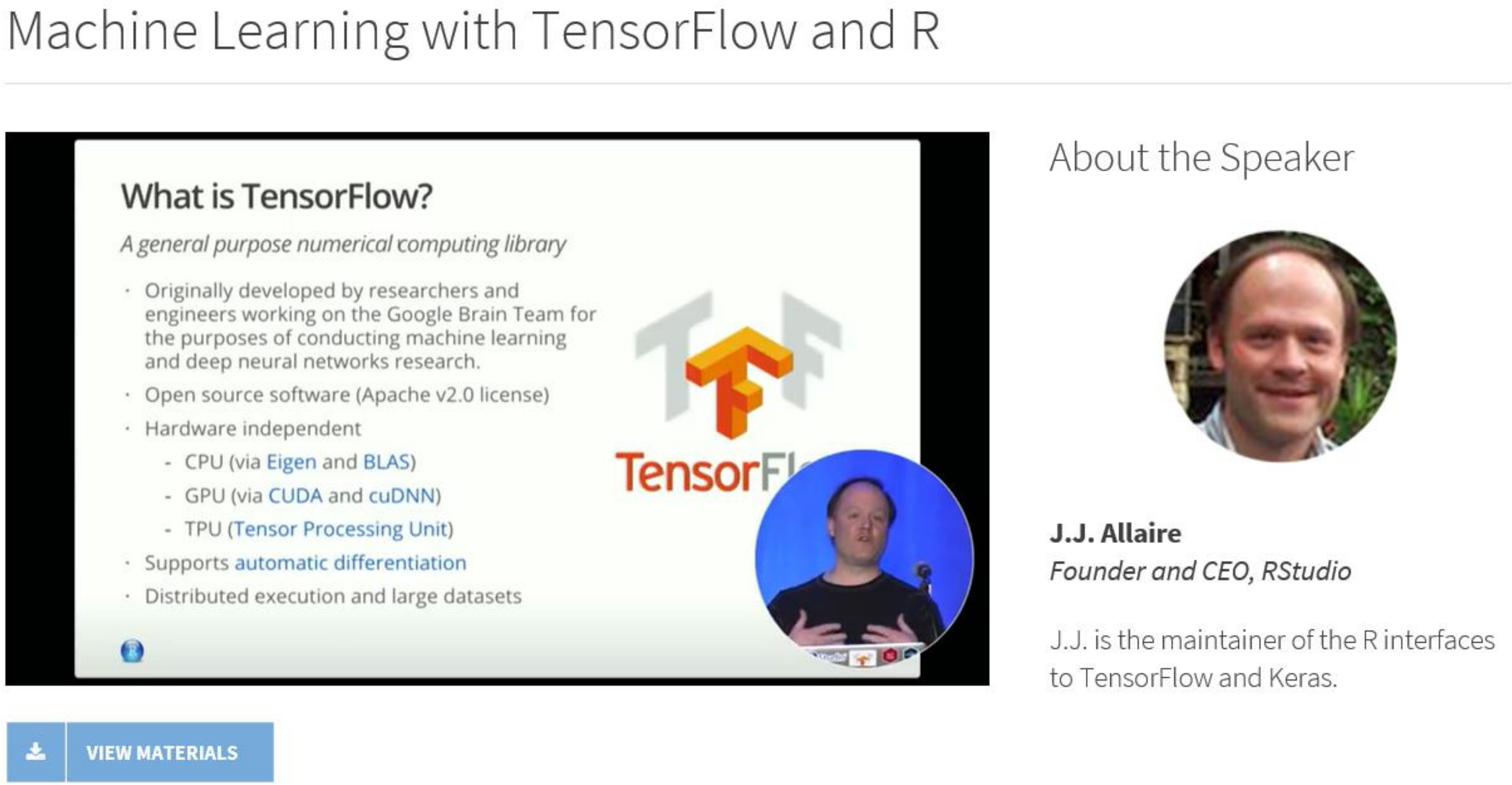
Hadi @HadiEOind · Feb 20

The best, most intuitive, and hype-debunking explanation on #DeepLearning with #TensorFlow I've watched so far =>
©J.J. Allaire's keynote at **#rstudioconf** 2018:
youtube.com/watch?v=atiYXm...
plus what a powerful R interface to #TensorFlow the RStudio team has built!
#rstats

Why should R users care about deep learning?

- New problem domains for R:
  - Computer vision
  - Computer speech recognition
  - Reinforcement learning applications
- Improved techniques for our traditional domains?
  - Analyzing data with complex spatial or sequence dependencies
  - Analyzing data which requires a large amount of (potentially brittle) feature engineering to model effectively

Deep learning is proven to be effective at various complex "perceptual" tasks but *not yet proven* to be of widespread benefit in other domains.



Machine Learning with TensorFlow and R

What is TensorFlow?

A general purpose numerical computing library

- Originally developed by researchers and engineers working on the Google Brain Team for the purposes of conducting machine learning and deep neural networks research.
- Open source software (Apache v2.0 license)
- Hardware independent
  - CPU (via Eigen and BLAS)
  - GPU (via CUDA and cuDNN)
  - TPU (Tensor Processing Unit)
- Supports automatic differentiation
- Distributed execution and large datasets

VIEW MATERIALS

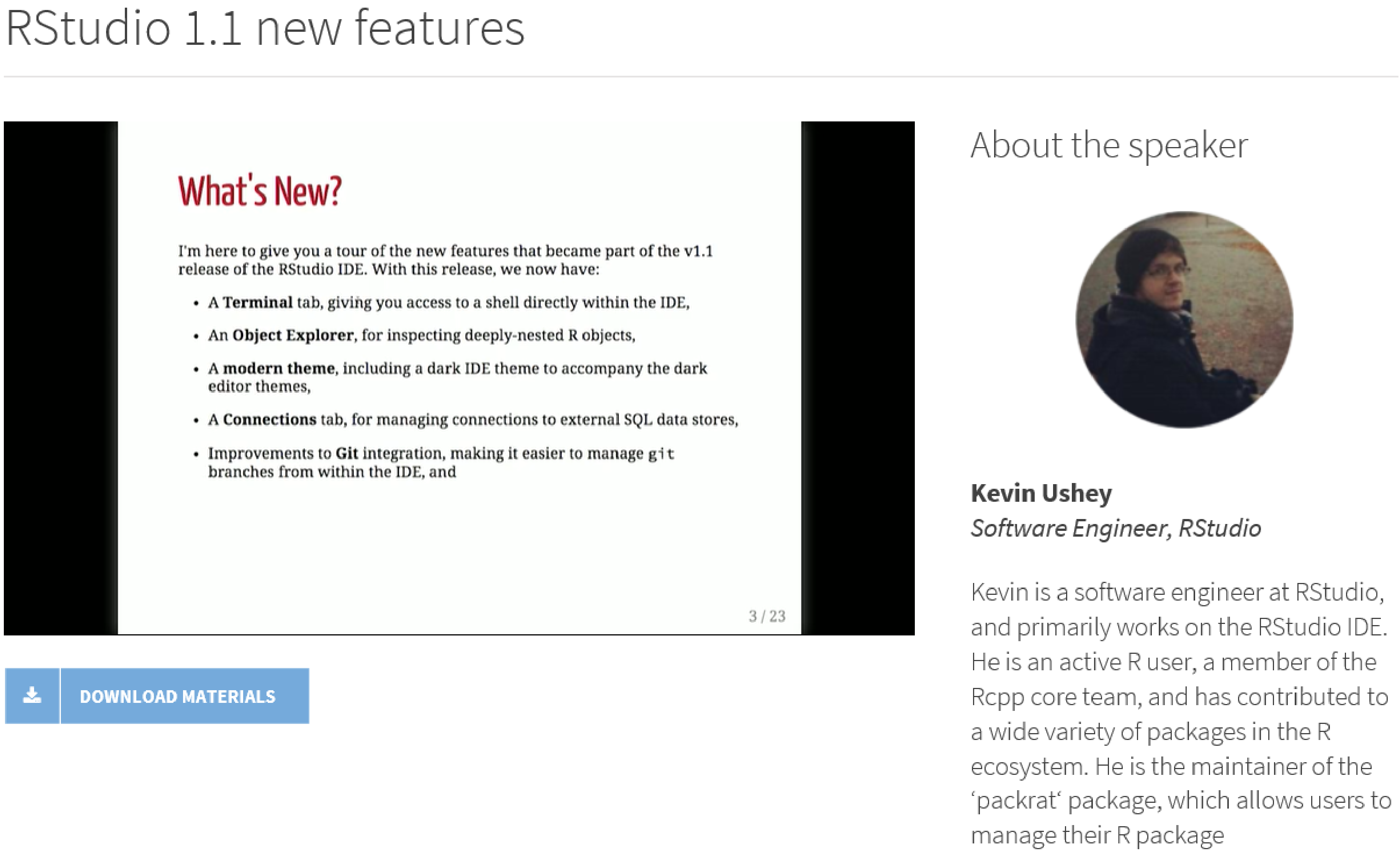
About the Speaker

J.J. Allaire
Founder and CEO, RStudio

J.J. is the maintainer of the R interfaces to TensorFlow and Keras.

https://tensorflow.rstudio.com/
https://github.com/rstudio/tensorflow
Call python modules in R via package reticulate

https://www.rstudio.com/resources/videos/rstudio-1-1-new-features/



RStudio 1.1 new features

What's New?

I'm here to give you a tour of the new features that became part of the v1.1 release of the RStudio IDE. With this release, we now have:

- A **Terminal** tab, giving you access to a shell directly within the IDE,
- An **Object Explorer**, for inspecting deeply-nested R objects,
- A **modern theme**, including a dark IDE theme to accompany the dark editor themes,
- A **Connections** tab, for managing connections to external SQL data stores,
- Improvements to **Git** integration, making it easier to manage `git` branches from within the IDE, and

3 / 23

DOWNLOAD MATERIALS

About the speaker

Kevin Ushey

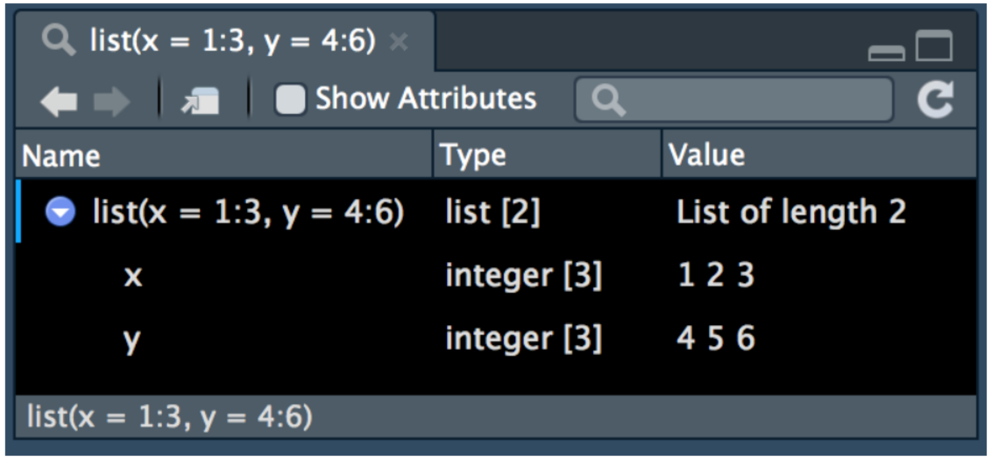
Software Engineer, RStudio

Kevin is a software engineer at RStudio, and primarily works on the RStudio IDE. He is an active R user, a member of the Rcpp core team, and has contributed to a wide variety of packages in the R ecosystem. He is the maintainer of the 'packrat' package, which allows users to manage their R package https://www.rstudio.com/resources/videos/rstudio-1-1-new-features/

RStudio 1.1 New Features

Object Explorer

However, with RStudio v1.1, it's now possible to explore R `lists` (and other hierarchical objects) using the **Object Explorer**.

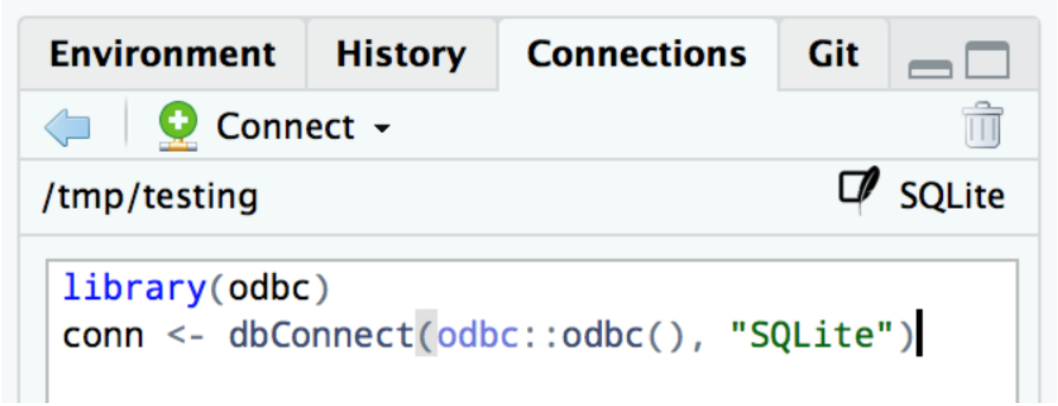


Name	Type	Value
list(x = 1:3, y = 4:6)	list [2]	List of length 2
x	integer [3]	1 2 3
y	integer [3]	4 5 6

list(x = 1:3, y = 4:6)

Connections

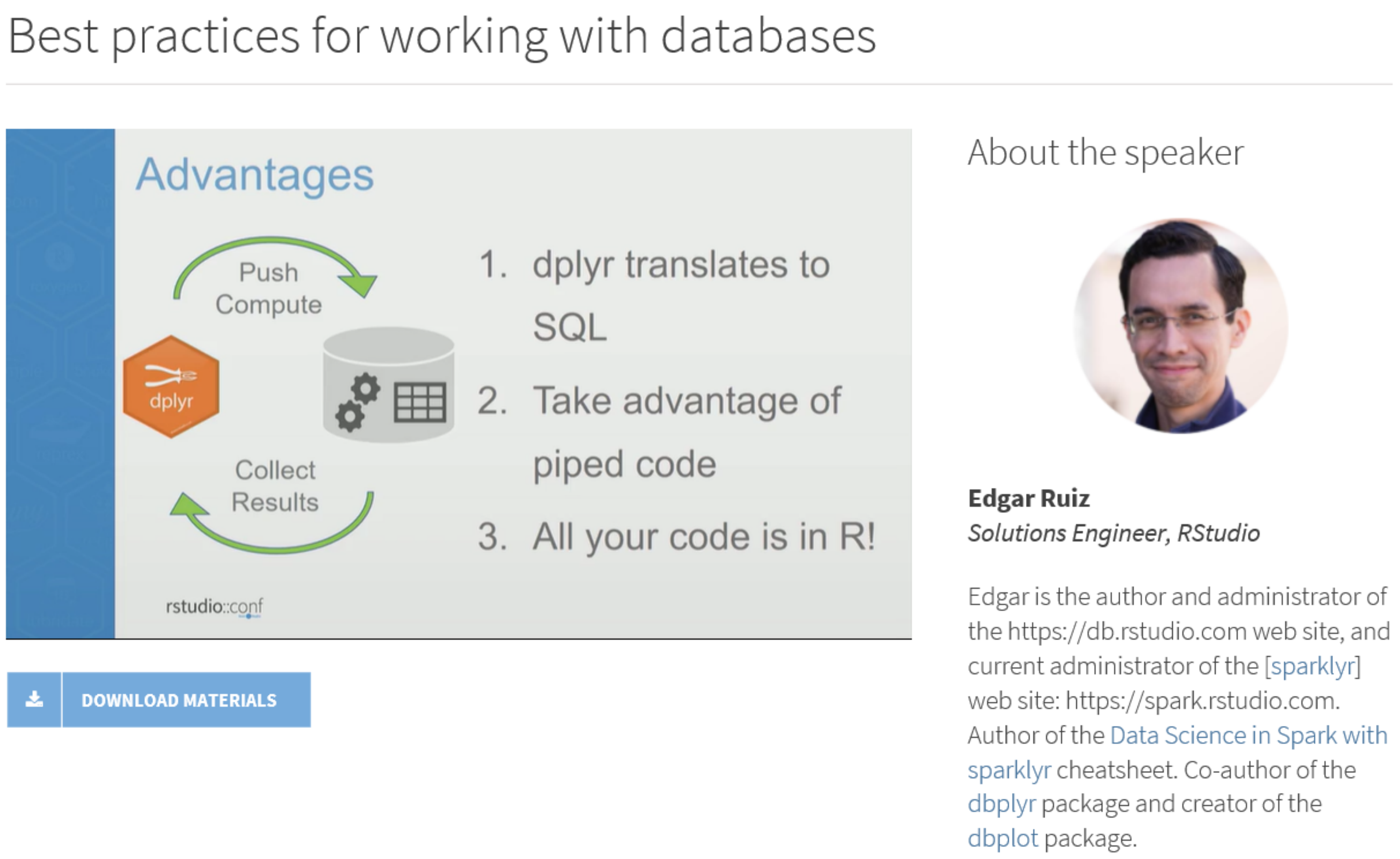
The **Connections** pane helps you manage connections to existing data stores, making it easy to slurp data from an external source into your R session.



```
library(odbc)
conn <- dbConnect(odbc::odbc(), "SQLite")
```

Many features similar to
Microsoft SQL Server Management Studio.
View tables, fields.

https://www.rstudio.com/resources/videos/best-practices-for-working-with-databases/
https://github.com/edgararuiz/db_best_practices_2018/blob/master/demo.Rmd



Best practices for working with databases

Advantages

1. dplyr translates to SQL
2. Take advantage of piped code
3. All your code is in R!

About the speaker

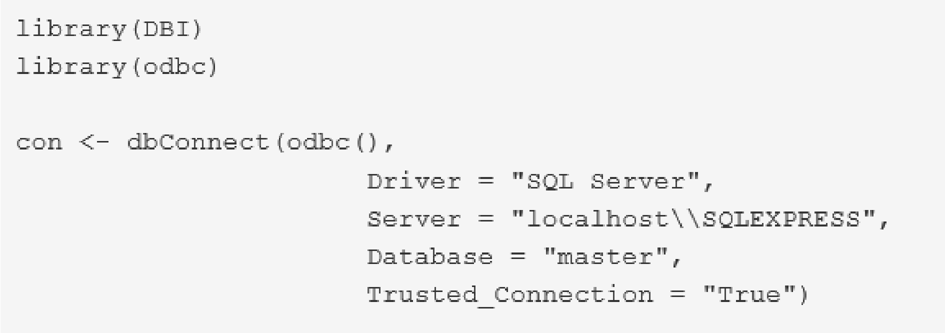
Edgar Ruiz
Solutions Engineer, RStudio

Edgar is the author and administrator of the https://db.rstudio.com web site, and current administrator of the [sparklyr] web site: https://spark.rstudio.com. Author of the Data Science in Spark with sparklyr cheatsheet. Co-author of the dbplyr package and creator of the dbplot package.

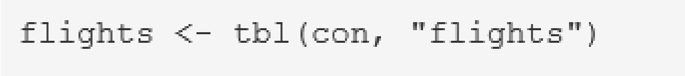
Packages: ODBC, DBI, dplyr, dbplyr

https://www.rstudio.com/resources/videos/best-practices-for-working-with-databases/
https://github.com/edgararuiz/db_best_practices_2018/blob/master/demo.Rmd

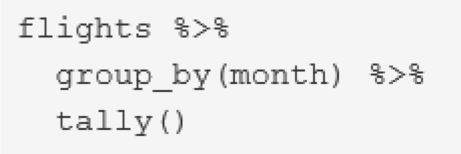
demo.nb.html



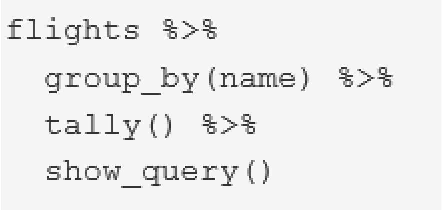
```
library(DBI)
library(odbc)

con <- dbConnect(odbc(),
                      Driver = "SQL Server",
                      Server = "localhost\\SQLEXPRESS",
                      Database = "master",
                      Trusted_Connection = "True")
```



```
flights <- tbl(con, "flights")
```



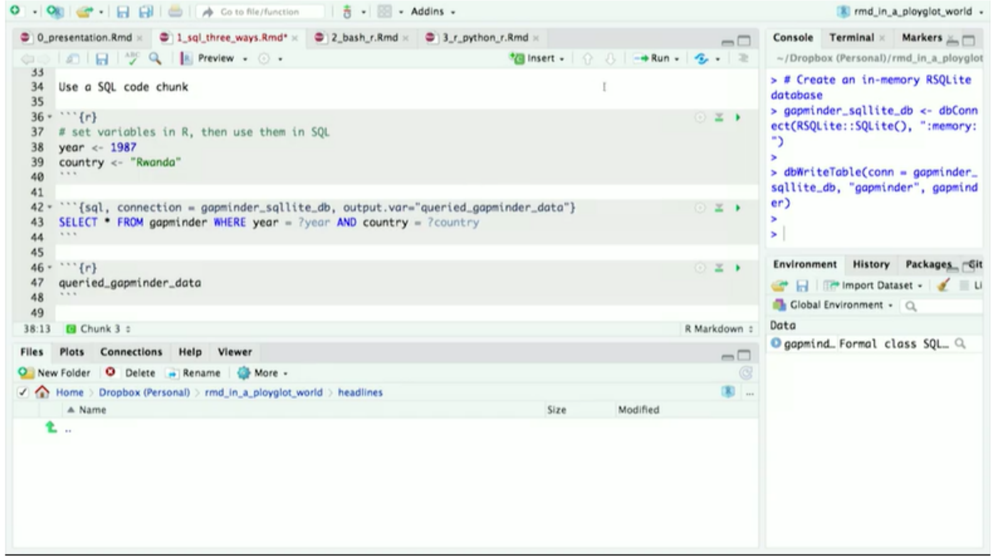
```
flights %>%
  group_by(month) %>%
  tally()
```



```
flights %>%
  group_by(name) %>%
  tally() %>%
  show_query()
```

https://www.rstudio.com/resources/videos/beyond-r-using-r-markdown-with-python-sql-bash-and-more/

Beyond R: Using R Markdown with python, sql, bash, and more



~~~
Use a SQL code chunk

```{r}
# set variables in R, then use them in SQL
year <- 1987
country <- "Rwanda"
```

```{sql, connection = gapminder_sqllite_db, output.var="queried_gapminder_data"}
SELECT * FROM gapminder WHERE year = ?year AND country = ?country
```

```{r}
queried_gapminder_data
```
~~~

```
> # Create an in-memory RSQLite
database
> gapminder_sqllite_db <- dbConn
ect(RSQLite::SQLite(), ":memory:
")
>
> dbWriteTable(conn = gapminder_
sqllite_db, "gapminder", gapmind
er)
>
> |
```

Abstract

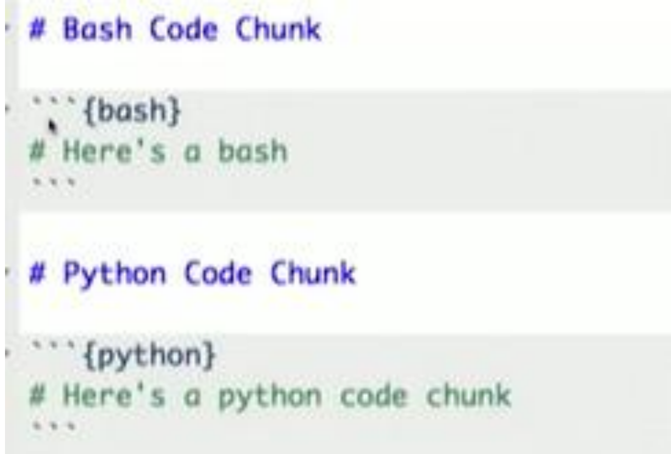
This talk gives an overview of three major use cases for multilingual RMarkdown: building self-documenting data pipelines, rapidly prototyping data science assets, and building ad hoc reports. Our focus is on why multilingual Rmd is valuable *in addition to* the reasons Rmdis already a valuable format (a good general case for Rmd exists [here](https://rviews.rstudio.com/2017/03/15/why-i-love-r-notebooks/). The case for multilingual Rmd focuses on flexibility, collaboration, time-to-value, and indecisiveness (in a good way!). Three examples demonstrate why multi-lingual Rmd should be a part of a data scientist's toolkit.

About the speaker

Aaron Berg
Customer Success, RStudio

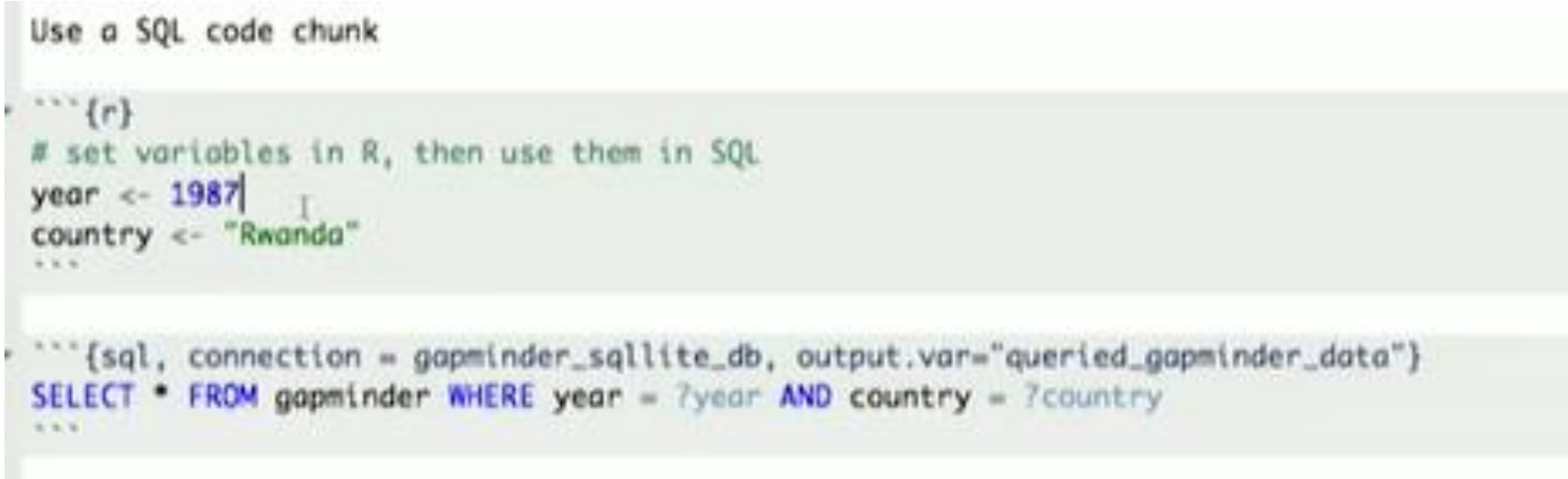
Aaron's background is in building business processes and data systems for commodity companies. Most recently he used R to automate finance, risk management, and reporting activities for a coffee trading business.

https://www.rstudio.com/resources/videos/beyond-r-using-r-markdown-with-python-sql-bash-and-more/



~~~
# Bash Code Chunk

```{bash}
# Here's a bash
~~~

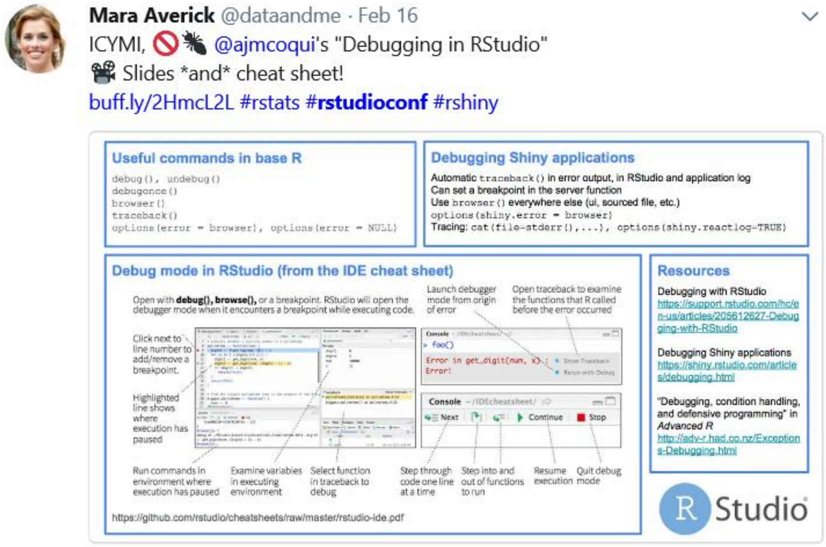
Python Code Chunk

```{python}
# Here's a python code chunk
```
```

Package feather to exchange data



```
Use a SQL code chunk

```{r}
# set variables in R, then use them in SQL
year <- 1987
country <- "Rwanda"
```

```{sql, connection = gapminder_sqllite_db, output.var="queried_gapminder_data"}
SELECT * FROM gapminder WHERE year = ?year AND country = ?country
```
```
```

https://www.rstudio.com/resources/videos/debugging-techniques-in-rstudio/



Mara Averick @dataandme · Feb 16

ICYMI, 🚫🐛 @ajmcoqui's "Debugging in RStudio"
📽 Slides *and* cheat sheet!
buff.ly/2HmcL2L #rstats #**rstudioconf** #rshiny

Useful commands in base R

debug(), undebug()
debugonce()
browser()
traceback()
options(error = browser), options(error = NULL)

Debugging Shiny applications

Automatic traceback() in error output, in RStudio and application log
Can set a breakpoint in the server function
Use browser() everywhere else (ui, sourced file, etc.)
options(shiny.error = browser)
Tracing: cat(file=stderr(),...), options(shiny.reactlog=TRUE)

Debug mode in RStudio (from the IDE cheat sheet)

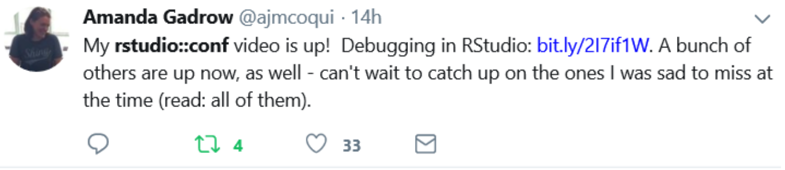
Resources

Debugging with RStudio
https://support.rstudio.com/hc/en-us/articles/205612627-Debugging-with-RStudio

Debugging Shiny applications
https://shiny.rstudio.com/articles/debugging.html

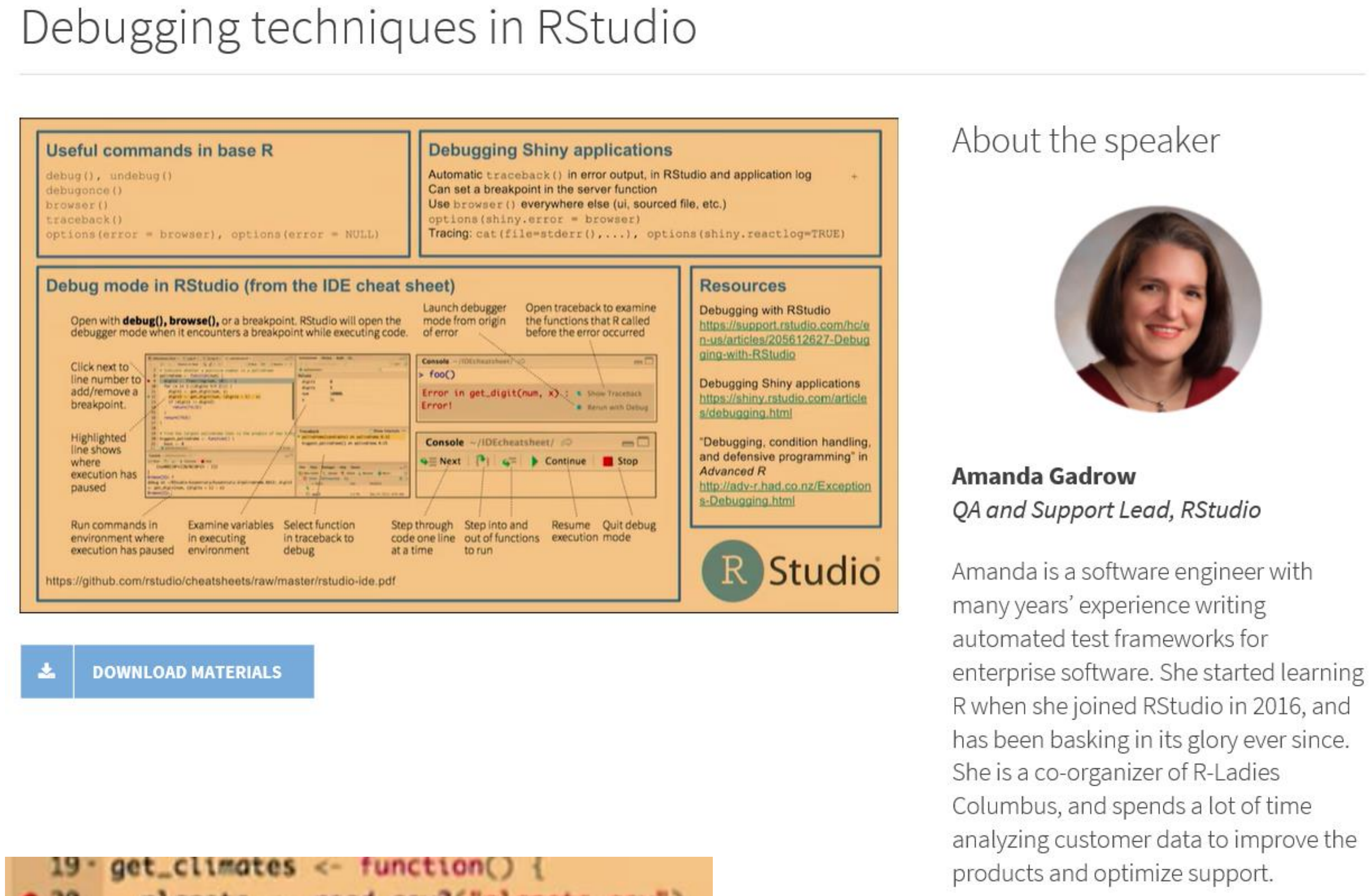
"Debugging, condition handling, and defensive programming" in *Advanced R*
http://adv-r.had.co.nz/Exceptions-Debugging.html

https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf



Amanda Gadrow @ajmcoqui · 14h

My **rstudio::conf** video is up! Debugging in RStudio: bit.ly/2I7if1W. A bunch of others are up now, as well - can't wait to catch up on the ones I was sad to miss at the time (read: all of them).



Debugging techniques in RStudio

Useful commands in base R

```
debug(), undebug()
debugonce()
browser()
traceback()
options(error = browser), options(error = NULL)
```

Debugging Shiny applications

Automatic traceback() in error output, in RStudio and application log
Can set a breakpoint in the server function
Use browser() everywhere else (ui, sourced file, etc.)
options(shiny.error = browser)
Tracing: cat(file=stderr(),...), options(shiny.reactlog=TRUE)

Debug mode in RStudio (from the IDE cheat sheet)

Open with **debug(), browse(),** or a breakpoint. RStudio will open the debugger mode when it encounters a breakpoint while executing code.

```
> foo()
Error in get_digit(num, x) :
Error!
```

Launch debugger mode from origin of error
Open traceback to examine the functions that R called before the error occurred
Click next to line number to add/remove a breakpoint.
Highlighted line shows where execution has paused
Show Traceback
Rerun with Debug
Console ~/IDEcheatsheet/
Next
Continue
Stop
Run commands in environment where execution has paused
Examine variables in executing environment
Select function in traceback to debug
Step through code one line at a time
Step into and out of functions to run
Resume execution
Quit debug mode

https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf

Resources

Debugging with RStudio
https://support.rstudio.com/hc/en-us/articles/205612627-Debugging-with-RStudio

Debugging Shiny applications
https://shiny.rstudio.com/articles/debugging.html

"Debugging, condition handling, and defensive programming" in *Advanced R*
http://adv-r.had.co.nz/Exceptions-Debugging.html

RStudio

DOWNLOAD MATERIALS

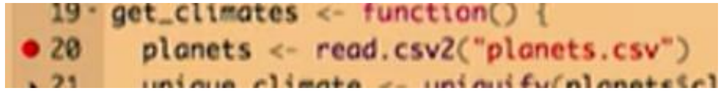
About the speaker

Amanda Gadrow

QA and Support Lead, RStudio

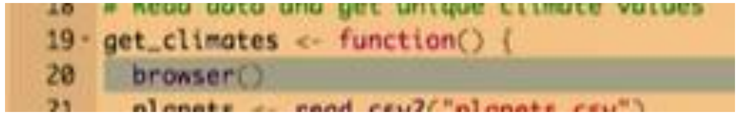
Amanda is a software engineer with many years' experience writing automated test frameworks for enterprise software. She started learning R when she joined RStudio in 2016, and has been basking in its glory ever since. She is a co-organizer of R-Ladies Columbus, and spends a lot of time analyzing customer data to improve the products and optimize support.

Editor breakpoint



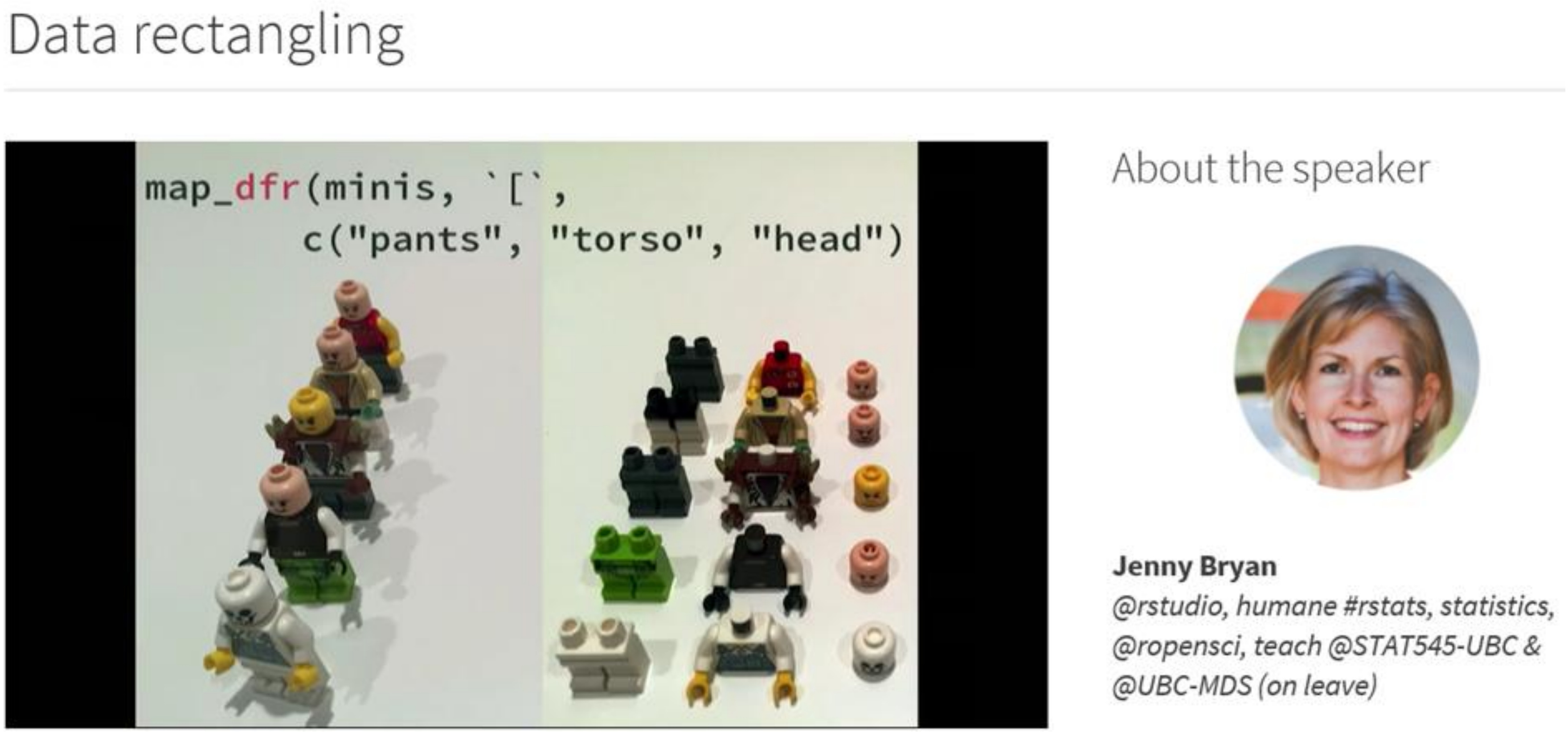
```
get_climates <- function() {
  planets <- read.csv2("planets.csv")
  unique_climate <- uniquify(planets$cl
```

Browser breakpoint



```
# Read data and get unique climate values
get_climates <- function() {
  browser()
  planets <- read.csv2("planets.csv")
```

https://www.rstudio.com/resources/videos/data-rectangling/
https://speakerdeck.com/jennybc/data-rectangling



Data rectangling

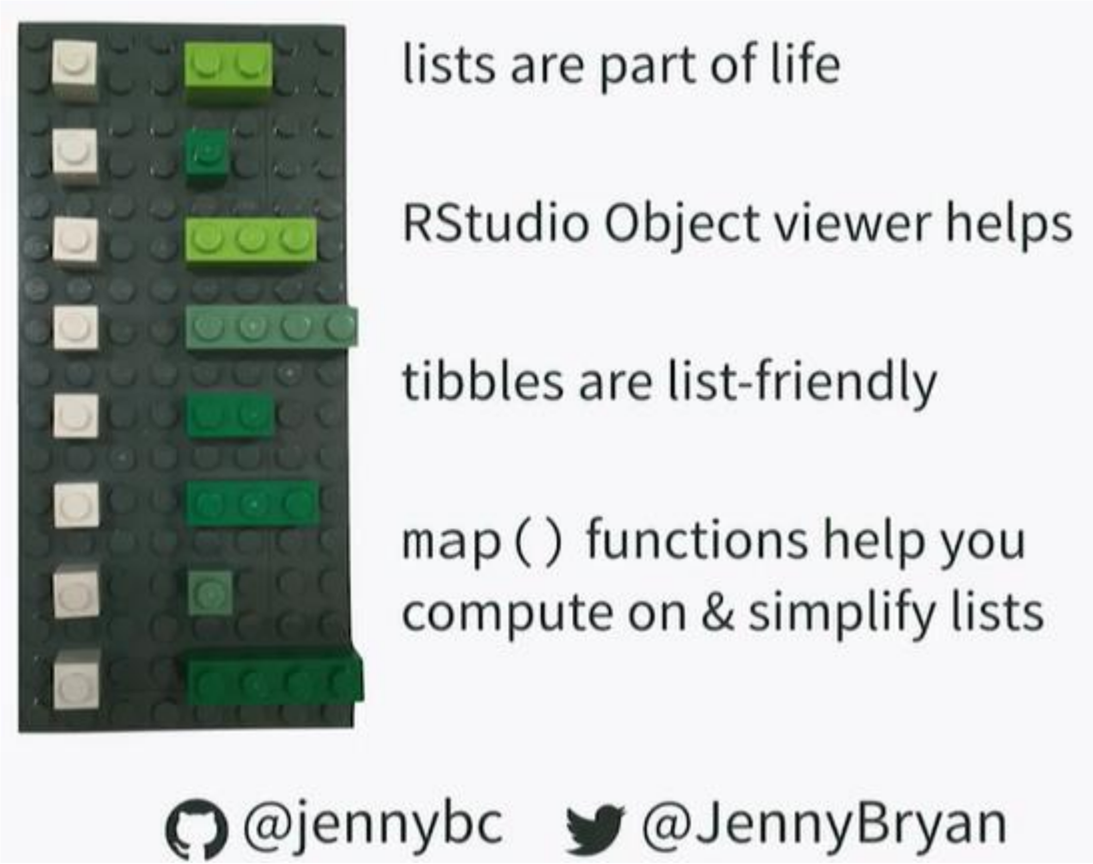
```
map_dfr(minis, `[`,
        c("pants", "torso", "head")
```

About the speaker

Jenny Bryan
@rstudio, humane #rstats, statistics, @ropensci, teach @STAT545-UBC & @UBC-MDS (on leave)

- Lists in data frames
- purrrr::map

https://www.rstudio.com/resources/videos/data-rectangling/
https://speakerdeck.com/jennybc/data-rectangling



lists are part of life

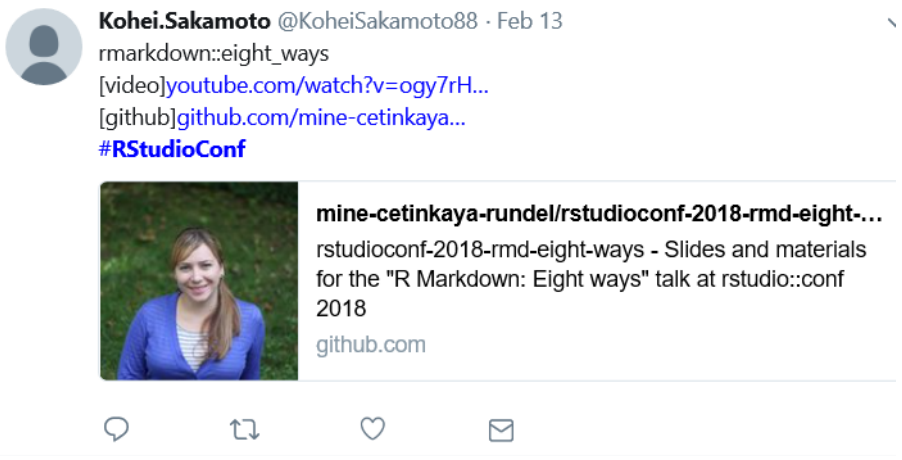
RStudio Object viewer helps

tibbles are list-friendly

map ( ) functions help you compute on & simplify lists

@jennybc @JennyBryan

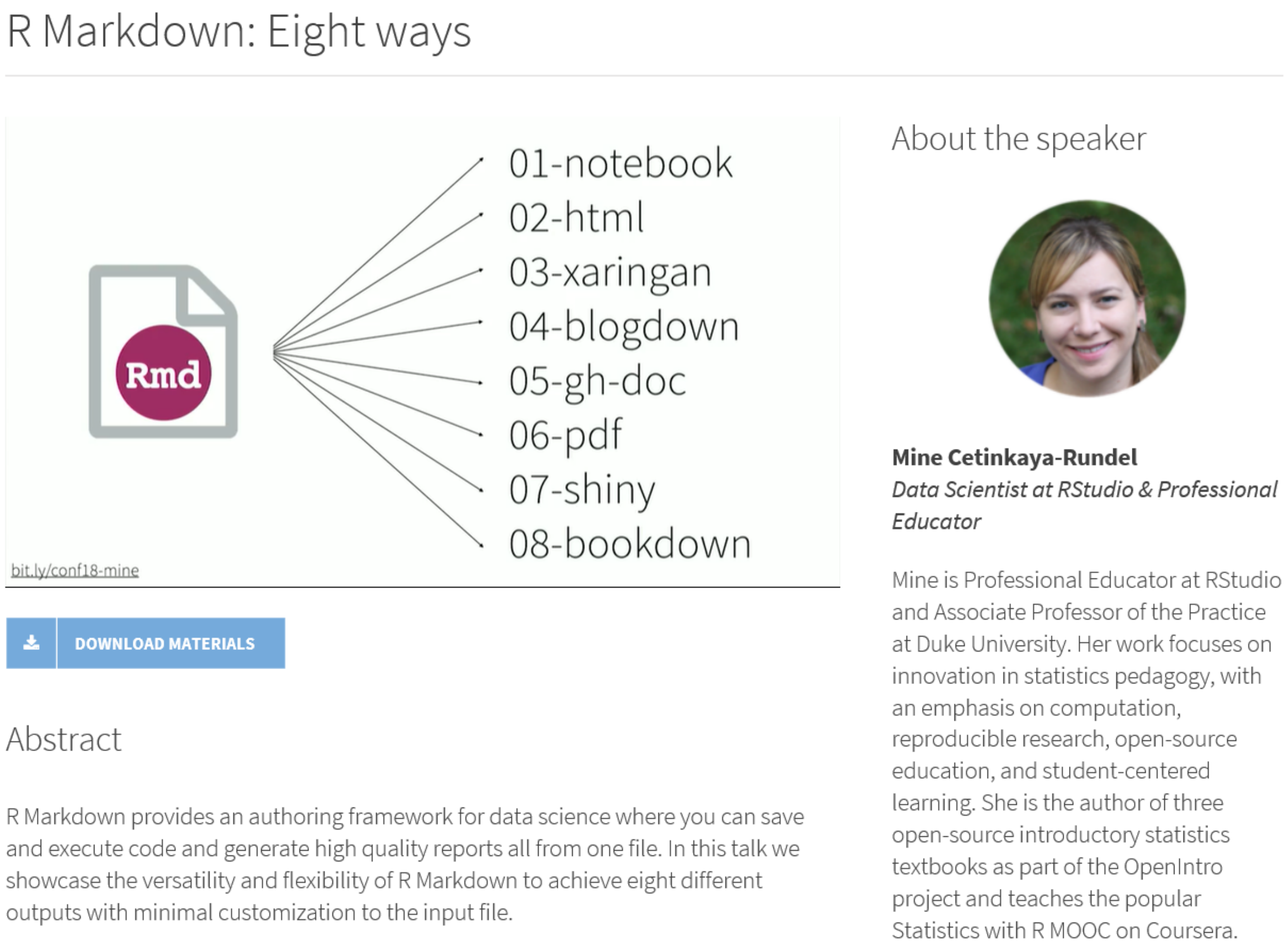
https://www.rstudio.com/resources/videos/r-markdown-eight-ways/
https://github.com/mine-cetinkaya-rundel/rstudioconf-2018-rmd-eight-ways



Kohei.Sakamoto @KoheiSakamoto88 · Feb 13

rmarkdown::eight_ways
[video]youtube.com/watch?v=ogy7rH...
[github]github.com/mine-cetinkaya...
#RStudioConf

mine-cetinkaya-rundel/rstudioconf-2018-rmd-eight-...
rstudioconf-2018-rmd-eight-ways - Slides and materials for the "R Markdown: Eight ways" talk at rstudio::conf 2018
github.com



R Markdown: Eight ways

Abstract

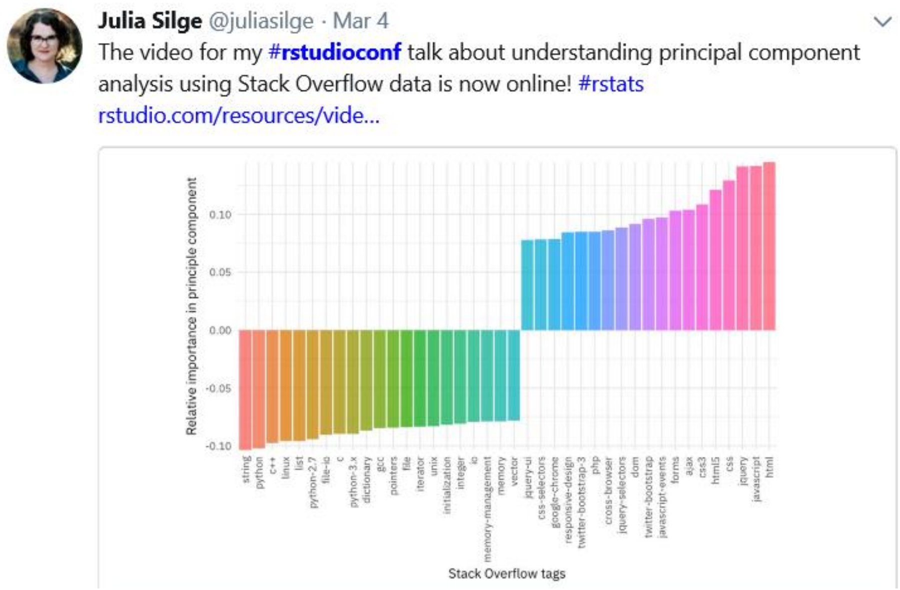
R Markdown provides an authoring framework for data science where you can save and execute code and generate high quality reports all from one file. In this talk we showcase the versatility and flexibility of R Markdown to achieve eight different outputs with minimal customization to the input file.

About the speaker

Mine Cetinkaya-Rundel
Data Scientist at RStudio & Professional Educator

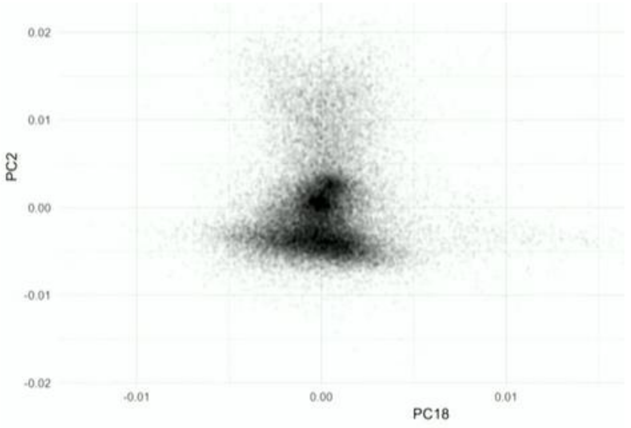
Mine is Professional Educator at RStudio and Associate Professor of the Practice at Duke University. Her work focuses on innovation in statistics pedagogy, with an emphasis on computation, reproducible research, open-source education, and student-centered learning. She is the author of three open-source introductory statistics textbooks as part of the OpenIntro project and teaches the popular Statistics with R MOOC on Coursera.

https://www.rstudio.com/resources/videos/understanding-pca-using-shiny-and-stack-overflow-data/



Julia Silge @juliasilge · Mar 4

The video for my **#rstudioconf** talk about understanding principal component analysis using Stack Overflow data is now online! #rstats rstudio.com/resources/vide...

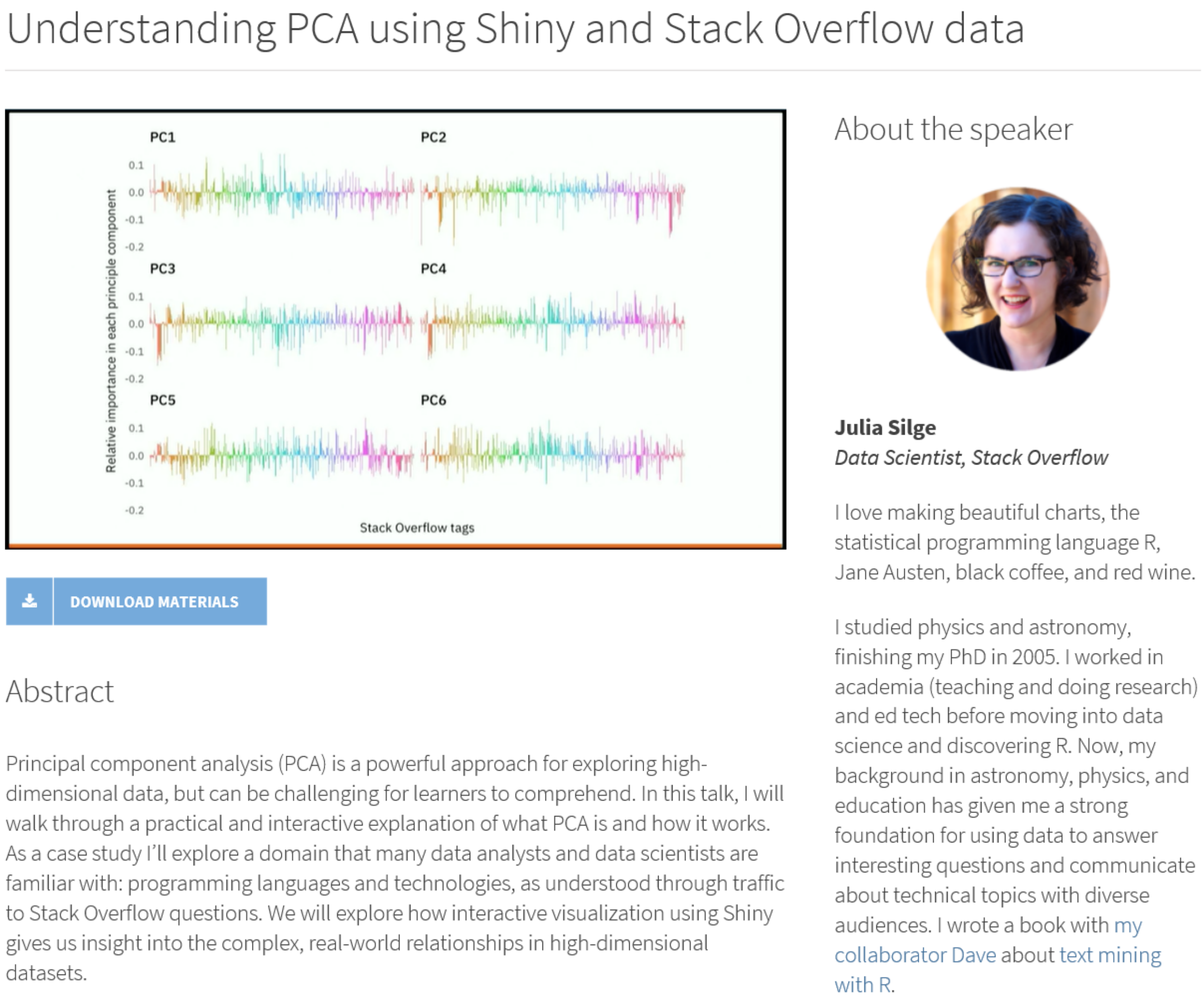


Microsoft

Not Microsoft

Python

R



Understanding PCA using Shiny and Stack Overflow data

Abstract

Principal component analysis (PCA) is a powerful approach for exploring high-dimensional data, but can be challenging for learners to comprehend. In this talk, I will walk through a practical and interactive explanation of what PCA is and how it works. As a case study I'll explore a domain that many data analysts and data scientists are familiar with: programming languages and technologies, as understood through traffic to Stack Overflow questions. We will explore how interactive visualization using Shiny gives us insight into the complex, real-world relationships in high-dimensional datasets.

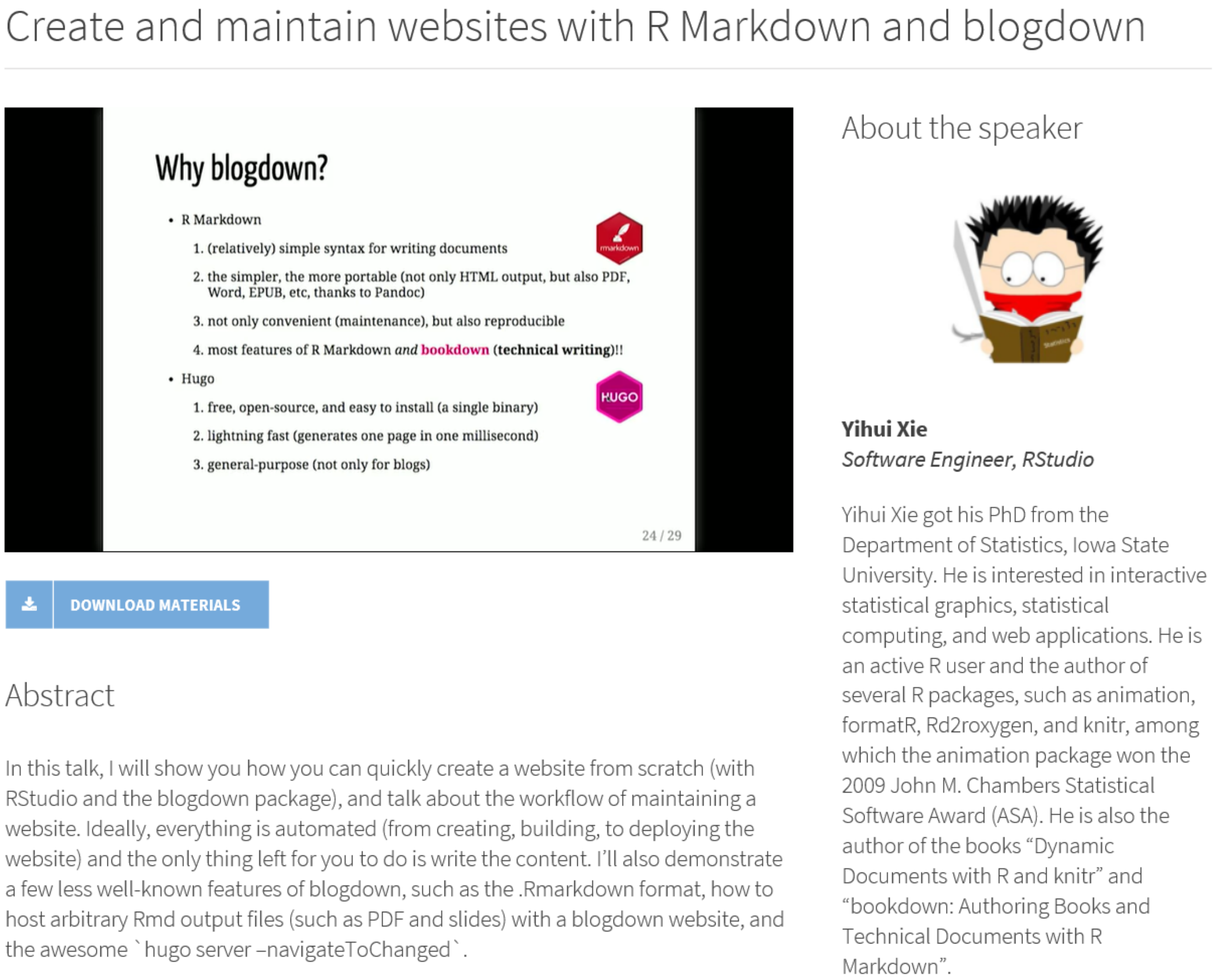
About the speaker

Julia Silge
Data Scientist, Stack Overflow

I love making beautiful charts, the statistical programming language R, Jane Austen, black coffee, and red wine.

I studied physics and astronomy, finishing my PhD in 2005. I worked in academia (teaching and doing research) and ed tech before moving into data science and discovering R. Now, my background in astronomy, physics, and education has given me a strong foundation for using data to answer interesting questions and communicate about technical topics with diverse audiences. I wrote a book with my collaborator Dave about text mining with R.

https://www.rstudio.com/resources/videos/create-and-maintain-websites-with-r-markdown-and-blogdown/
https://slides.yihui.name/2018-blogdown-rstudio-conf-Yihui-Xie.html



Create and maintain websites with R Markdown and blogdown

Why blogdown?

- R Markdown
  1. (relatively) simple syntax for writing documents
  2. the simpler, the more portable (not only HTML output, but also PDF, Word, EPUB, etc, thanks to Pandoc)
  3. not only convenient (maintenance), but also reproducible
  4. most features of R Markdown *and* **bookdown** **(technical writing)**!!
- Hugo
  1. free, open-source, and easy to install (a single binary)
  2. lightning fast (generates one page in one millisecond)
  3. general-purpose (not only for blogs)

24 / 29

Abstract

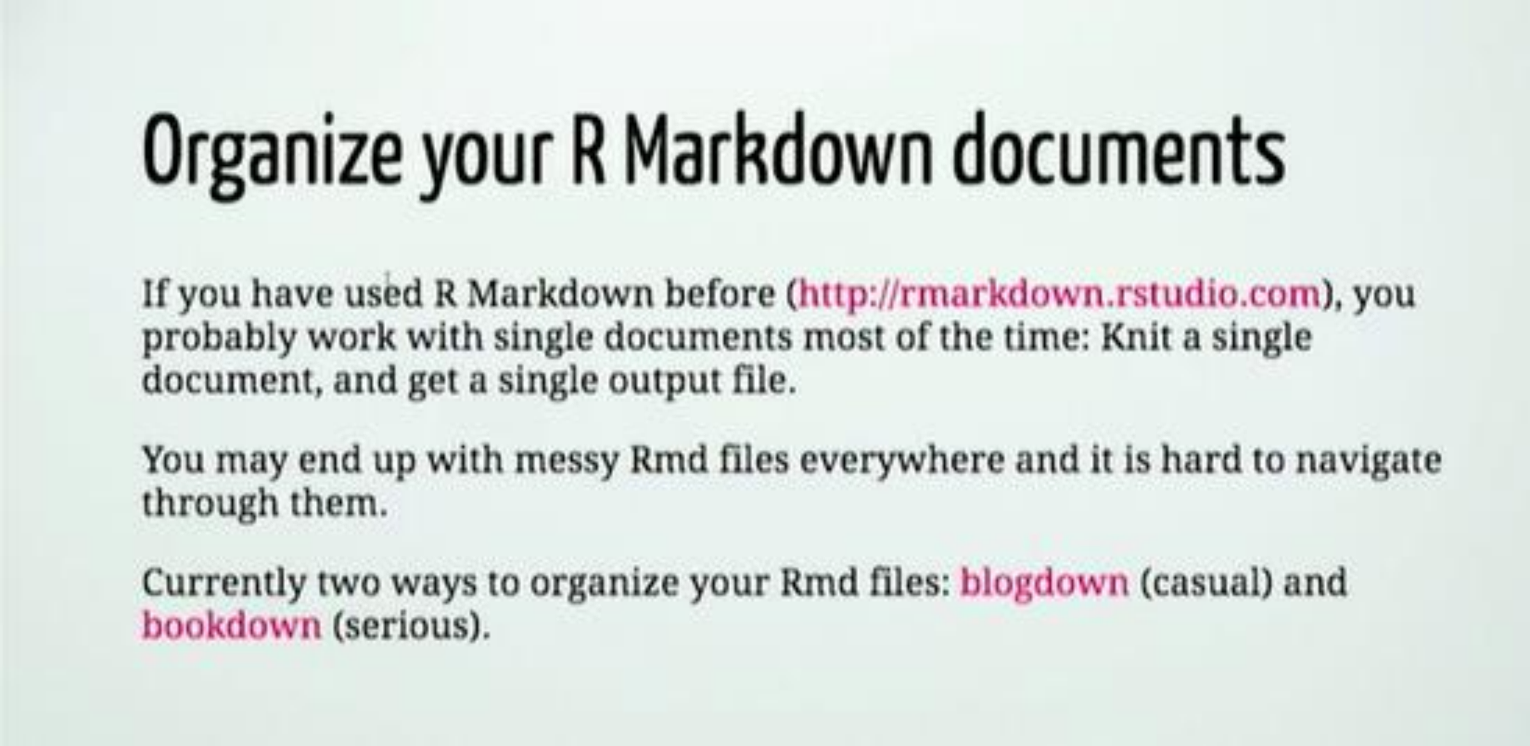
In this talk, I will show you how you can quickly create a website from scratch (with RStudio and the blogdown package), and talk about the workflow of maintaining a website. Ideally, everything is automated (from creating, building, to deploying the website) and the only thing left for you to do is write the content. I'll also demonstrate a few less well-known features of blogdown, such as the .Rmarkdown format, how to host arbitrary Rmd output files (such as PDF and slides) with a blogdown website, and the awesome `hugo server –navigateToChanged`.

About the speaker

Yihui Xie
Software Engineer, RStudio

Yihui Xie got his PhD from the Department of Statistics, Iowa State University. He is interested in interactive statistical graphics, statistical computing, and web applications. He is an active R user and the author of several R packages, such as animation, formatR, Rd2roxygen, and knitr, among which the animation package won the 2009 John M. Chambers Statistical Software Award (ASA). He is also the author of the books "Dynamic Documents with R and knitr" and "bookdown: Authoring Books and Technical Documents with R Markdown".

https://www.rstudio.com/resources/videos/create-and-maintain-websites-with-r-markdown-and-blogdown/
https://slides.yihui.name/2018-blogdown-rstudio-conf-Yihui-Xie.html



Organize your R Markdown documents

If you have used R Markdown before (http://rmarkdown.rstudio.com), you probably work with single documents most of the time: Knit a single document, and get a single output file.

You may end up with messy Rmd files everywhere and it is hard to navigate through them.

Currently two ways to organize your Rmd files: blogdown (casual) and bookdown (serious).

RStudio Conference 2018 Links

- rstudio::conf 2018 Highlights https://jenthompson.me/2018/02/20/rstudioconf-2018/
- Highlights from the rstudio::conf 2018 https://www.analyticsvidhya.com/blog/2018/02/highlights-rstudio-conference-2018
- RStudio Conf 2018: I didn't lose my wallet or my keys https://sharlagelfand.netlify.com/posts/rstudio-conf-2018/