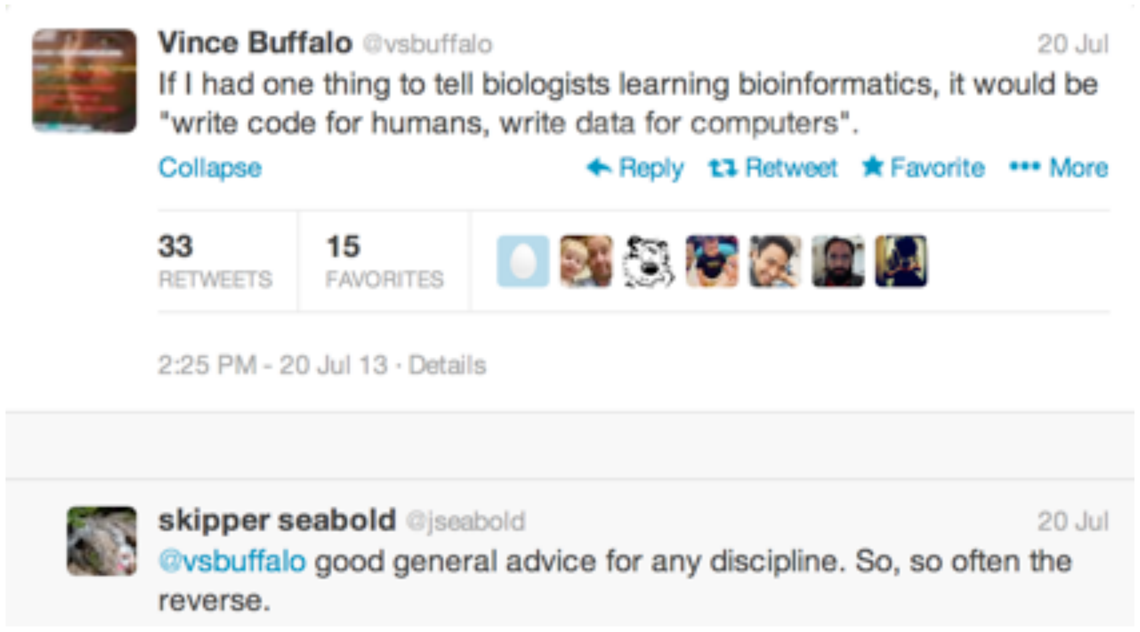
hello SWC bootcamp

Dr. Jennifer (Jenny) Bryan
Department of Statistics and Michael Smith Laboratories
University of British Columbia

write code for humans, write data for computers



Vince Buffalo @vsbuffalo 20 Jul
If I had one thing to tell biologists learning bioinformatics, it would be "write code for humans, write data for computers".

33 RETWEETS 15 FAVORITES

2:25 PM - 20 Jul 13 · Details

skipper seabold @jseabold 20 Jul
@vsbuffalo good general advice for any discipline. So, so often the reverse.

Let us change our traditional attitude to the construction of programs. Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

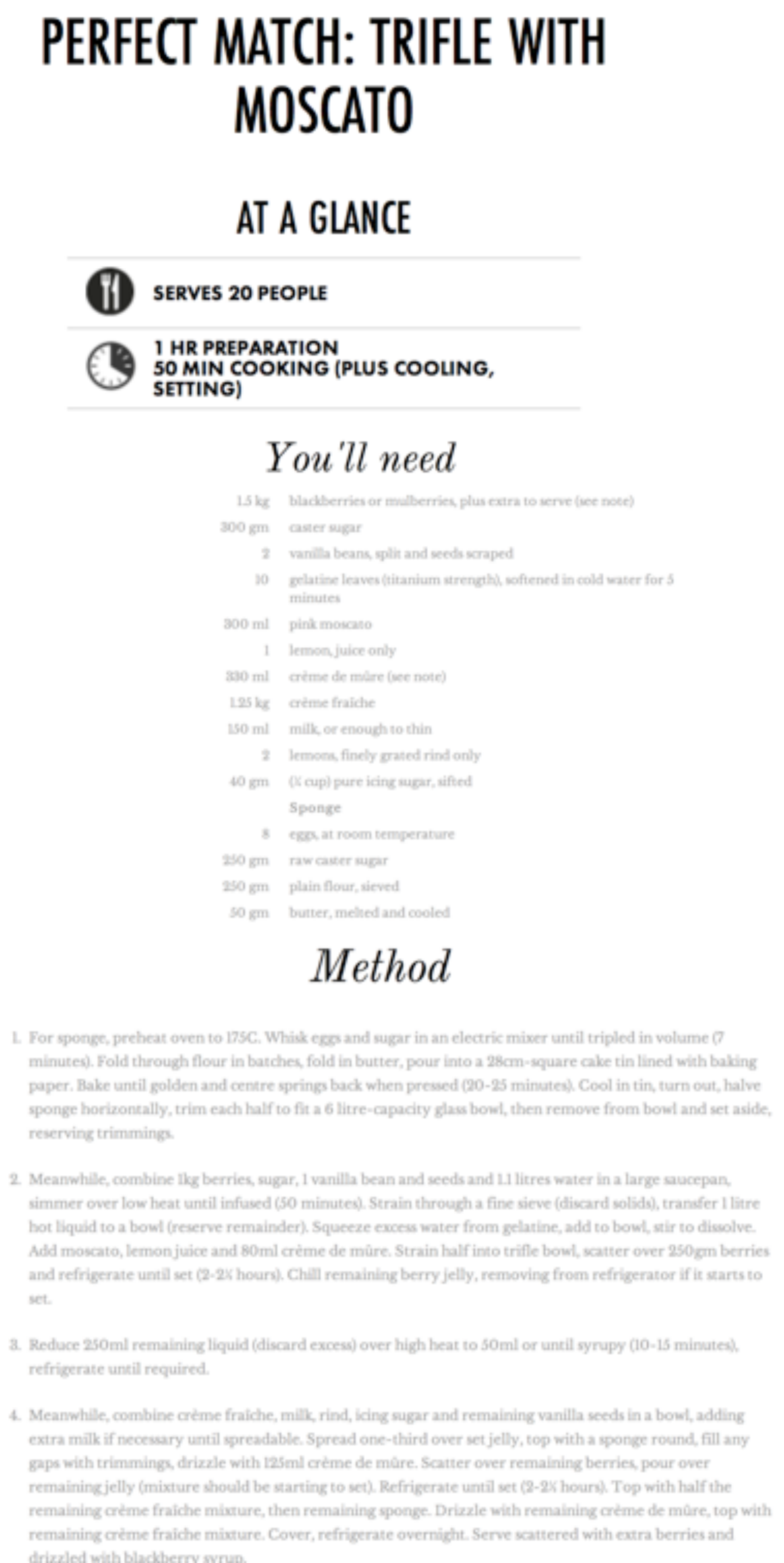
Donald Knuth

A place for everything and everything in its place

≪

source is real

≪



PERFECT MATCH: TRIFLE WITH MOSCATO

AT A GLANCE

SERVES 20 PEOPLE

1 HR PREPARATION
50 MIN COOKING (PLUS COOLING, SETTING)

You'll need

1.5 kg	blackberries or mulberries, plus extra to serve (see note)
300 gm	caster sugar
2	vanilla beans, split and seeds scraped
10	gelatine leaves (titanium strength), softened in cold water for 5 minutes
300 ml	pink moscato
1	lemon, juice only
330 ml	crème de mûre (see note)
1.25 kg	crème fraîche
150 ml	milk, or enough to thin
2	lemons, finely grated rind only
40 gm	(¼ cup) pure icing sugar, sifted
	Sponge
8	eggs, at room temperature
250 gm	raw caster sugar
250 gm	plain flour, sieved
50 gm	butter, melted and cooled

Method

1. For sponge, preheat oven to 175C. Whisk eggs and sugar in an electric mixer until tripled in volume (7 minutes). Fold through flour in batches, fold in butter, pour into a 28cm-square cake tin lined with baking paper. Bake until golden and centre springs back when pressed (20-25 minutes). Cool in tin, turn out, halve sponge horizontally, trim each half to fit a 6 litre-capacity glass bowl, then remove from bowl and set aside, reserving trimmings.

2. Meanwhile, combine 1kg berries, sugar, 1 vanilla bean and seeds and 1.1 litres water in a large saucepan, simmer over low heat until infused (50 minutes). Strain through a fine sieve (discard solids), transfer 1 litre hot liquid to a bowl (reserve remainder). Squeeze excess water from gelatine, add to bowl, stir to dissolve. Add moscato, lemon juice and 80ml crème de mûre. Strain half into trifle bowl, scatter over 250gm berries and refrigerate until set (2-2½ hours). Chill remaining berry jelly, removing from refrigerator if it starts to set.

3. Reduce 250ml remaining liquid (discard excess) over high heat to 50ml or until syrupy (10-15 minutes), refrigerate until required.

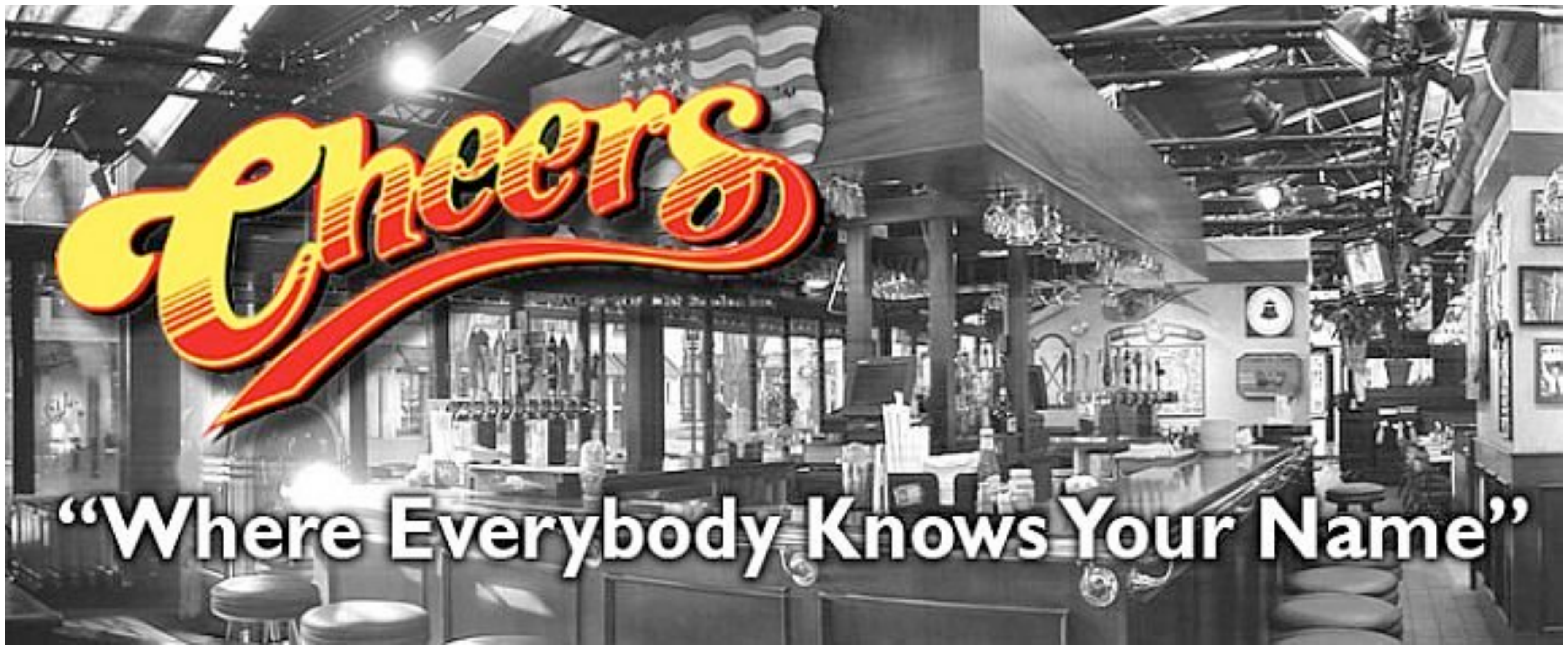
4. Meanwhile, combine crème fraîche, milk, rind, icing sugar and remaining vanilla seeds in a bowl, adding extra milk if necessary until spreadable. Spread one-third over set jelly, top with a sponge round, fill any gaps with trimmings, drizzle with 125ml crème de mûre. Scatter over remaining berries, pour over remaining jelly (mixture should be starting to set). Refrigerate until set (2-2½ hours). Top with half the remaining crème fraîche mixture, then remaining sponge. Drizzle with remaining crème de mûre, top with remaining crème fraîche mixture. Cover, refrigerate overnight. Serve scattered with extra berries and drizzled with blackberry syrup.

source is real

“**<u>The source code is real</u>**. The objects are realizations of the source code. Source for EVERY user modified object is placed in a particular directory or directories, for later editing and retrieval.”

-- from the <u>Emacs Speaks Statistics (ESS) manual</u>

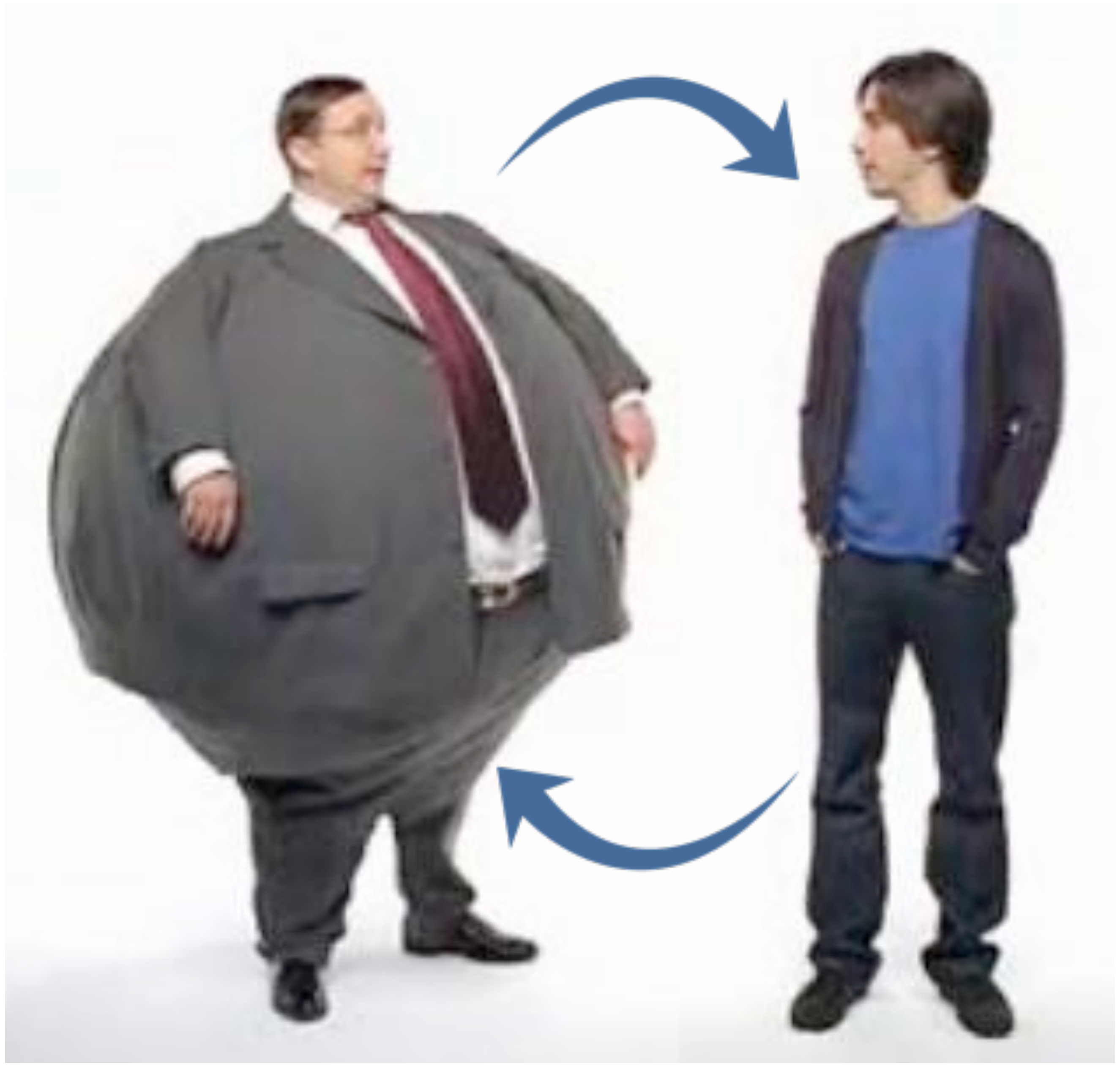
Names matter



ain’t nothing like the real thing

minimize the creation of excerpts and copies of your data ... it will just confuse you later

reshape your data



as in real life, it has a tendency to get short and fat, when you'd really prefer tall and skinny you won’t believe how important this is:

what is your working directory?

where is the file or executable you need?

you need to be fluent with file paths

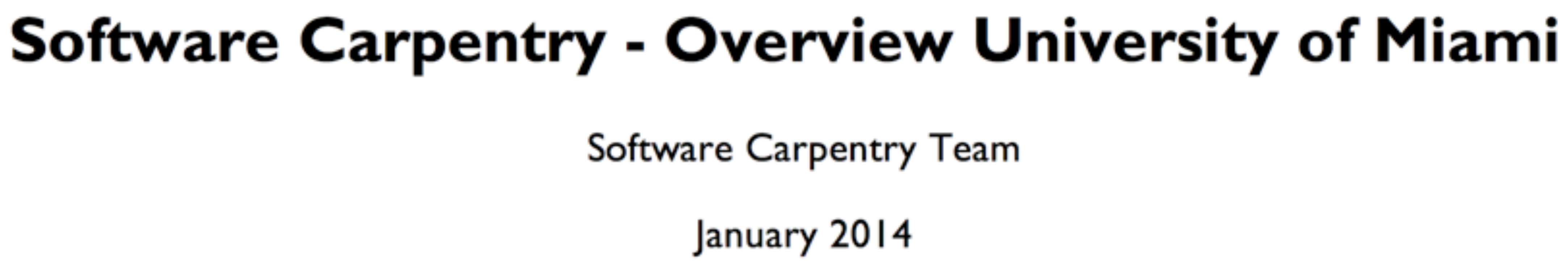
tough love:

get better at typing

typos matter
case matters
th_is is different from th-is
spaces in filenames are EVIL

you want computer to tedious work for you, right?
then you must give precise instructions

a few remarks borrowed from Jonah Duckles' intro (zero-entry R room)



Software Carpentry - Overview University of Miami

Software Carpentry Team

January 2014

What We *Actually* Teach

- A program is just another piece of lab equipment
- Programming is a human activity
- Little pieces loosely joined
- Let the computer repeat it
- Paranoia makes us productive
- Better algorithms beat better hardware

How to THINK like a programmer

Make it work right first, make it fast later.

- "Premature optimization is the root of all evil." -- Donald Knuth
- Directing your attention to making it use less disk / less memory / less time from the start is wrongly directed attention.

Don't Repeat Yourself (or Others)

Automate common actions by saving simple blocks of code into scripts

- A script is a set of commands organized into a single file
- The script is the basest unit of scientific programming, you should be comfortable writing these whenever you want to save or otherwise document or repeat your actions
- Use scripts to explore new ideas, they are easy to write and throw away

Use version control for checkpointing and collaboration

- use local version control software to checkpoint personal code development
- checkpointing your work encourages wild ideas and late-night coding sessions
- you can easily restore back in the morning if it was a bad idea
- use **distributed version control** to collaborate with others
- We advocate *Git*, but you may be stuck with whatever your group uses

Document your computational work

- Save every bit of code you use for generating publishable results
- Document and comment your code for yourself as if you will need to understand it in 6 months
- use README files liberally
- as well as file-level, function-level, and inline documentation
- If any piece of code is too complex to easily describe, consider refactoring it

end sermon

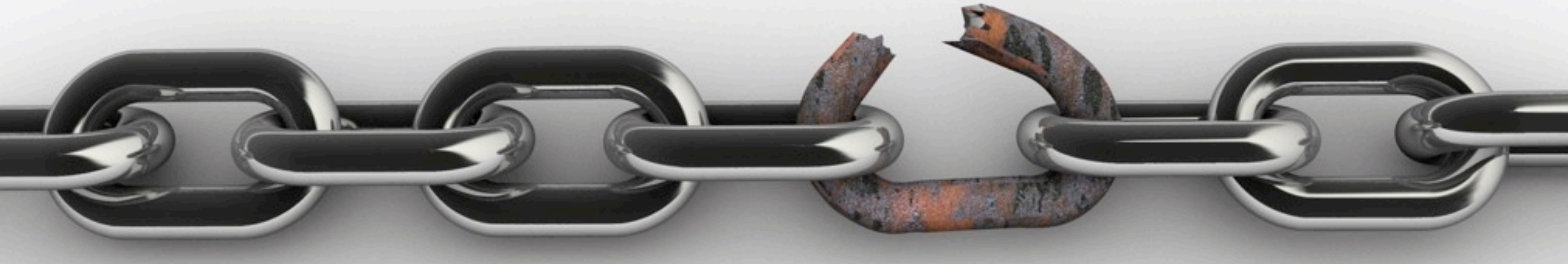
any questions from first tutorial re: basic use of R via RStudio, RStudio projects?

you do know that R ≠ RStudio, right?

we use RStudio because it makes us happier in our work, but notice that **nothing we produce -- no code, no figures, nothing -- requires RStudio to be created, appreciated or reused**

you should master various ways of sending code from editor to Console so that saving scripts becomes your R Way of Life

weak links in the chain: process, packaging and presentation



I assume your toyline.R script looks similar to this.

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))
(avgX <- mean(x))

write(avgX, "results/avgX.txt")

pdf("figs/niftyPlot.pdf")
plot(x, y)
abline(a, b, col = "blue", lwd = 2)
dev.off()
```

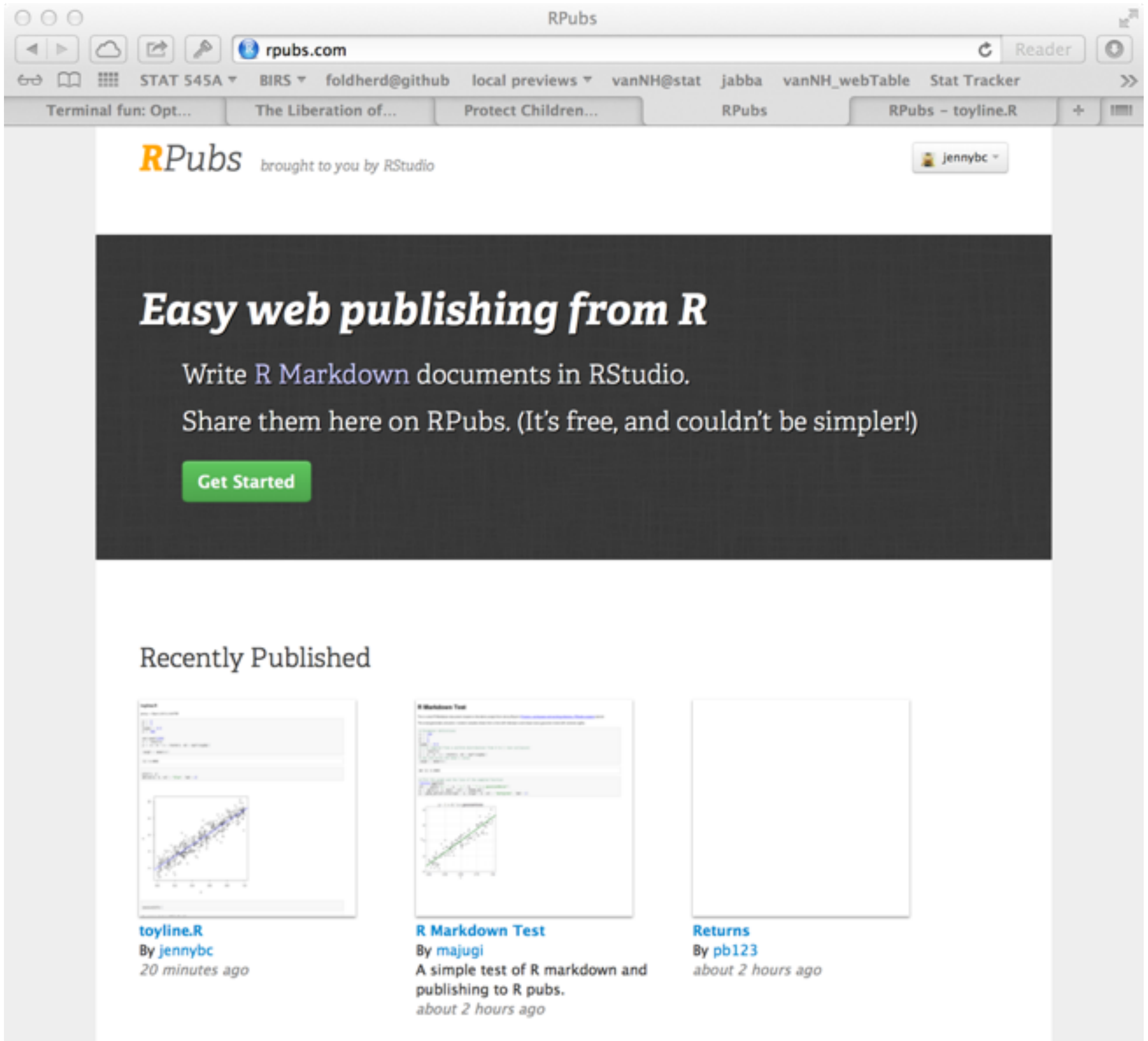
It’s great we are saving important things to file with code -- versus letting them cruise by in the Console and/or saving via mouse clicks -- but we can do better.

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))
(avgX <- mean(x))

write(avgX, "results/avgX.txt")

pdf("figs/niftyPlot.pdf")
plot(x, y)
abline(a, b, col = "blue", lwd = 2)
dev.off()
```

You did install `knitr` and its dependencies, as instructed in the set-up tutorial, right?

```
install.packages("knitr", dependencies = TRUE)
```

You did get an RPubs account, as requested, right?



RPubs *brought to you by RStudio*

Easy web publishing from R

Write R Markdown documents in RStudio.

Share them here on RPubs. (It's free, and couldn't be simpler!)

Recently Published

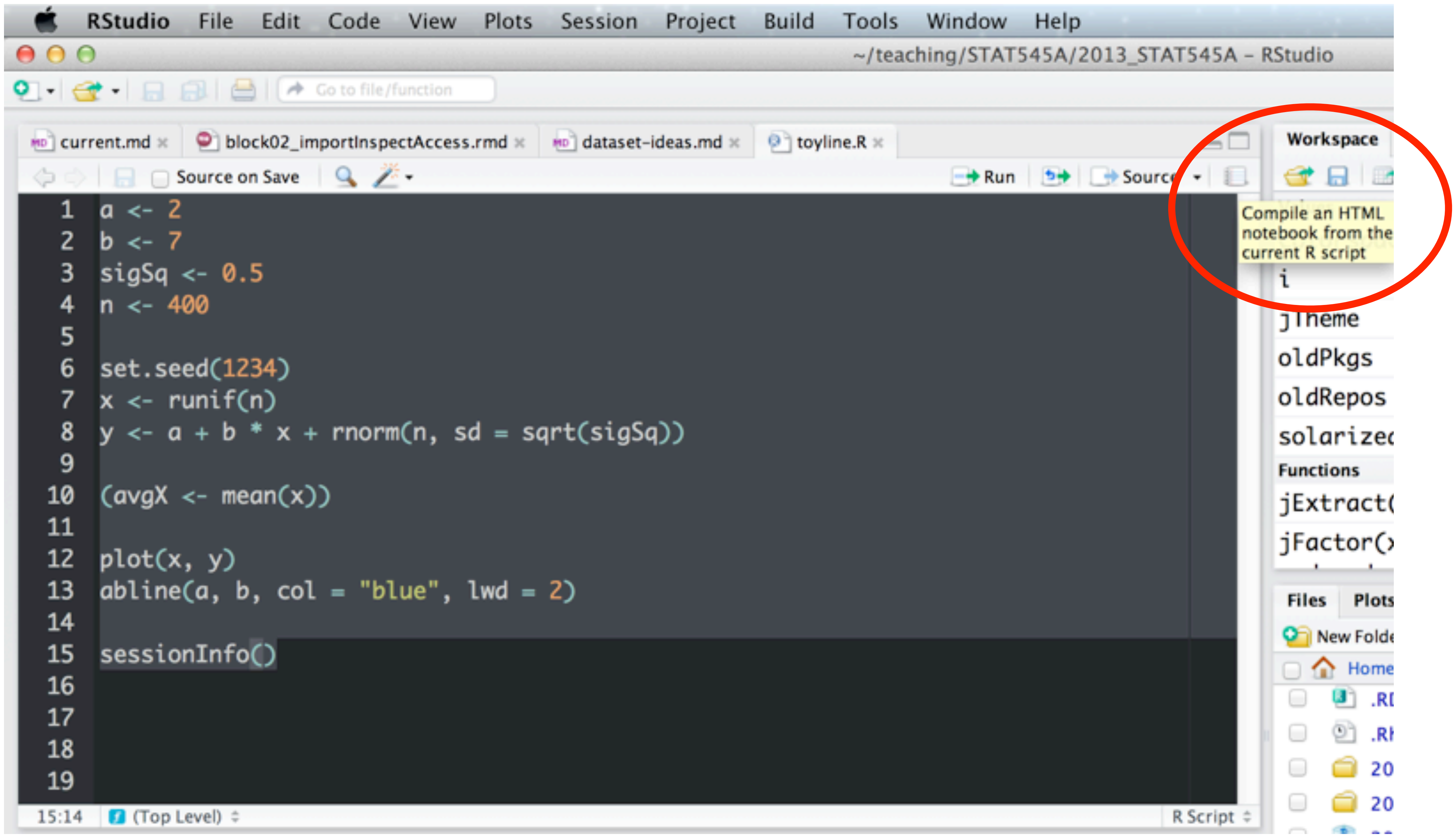
toyline.R
By jennybc
20 minutes ago

R Markdown Test
By majugi
A simple test of R markdown and publishing to R pubs.
about 2 hours ago

Returns
By pb123
about 2 hours ago

Edit the script -- more like it was during development, when we were watching results and figures appear on the screen.

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))

plot(x, y)
abline(a, b, col = "blue", lwd = 2)

sessionInfo()
```

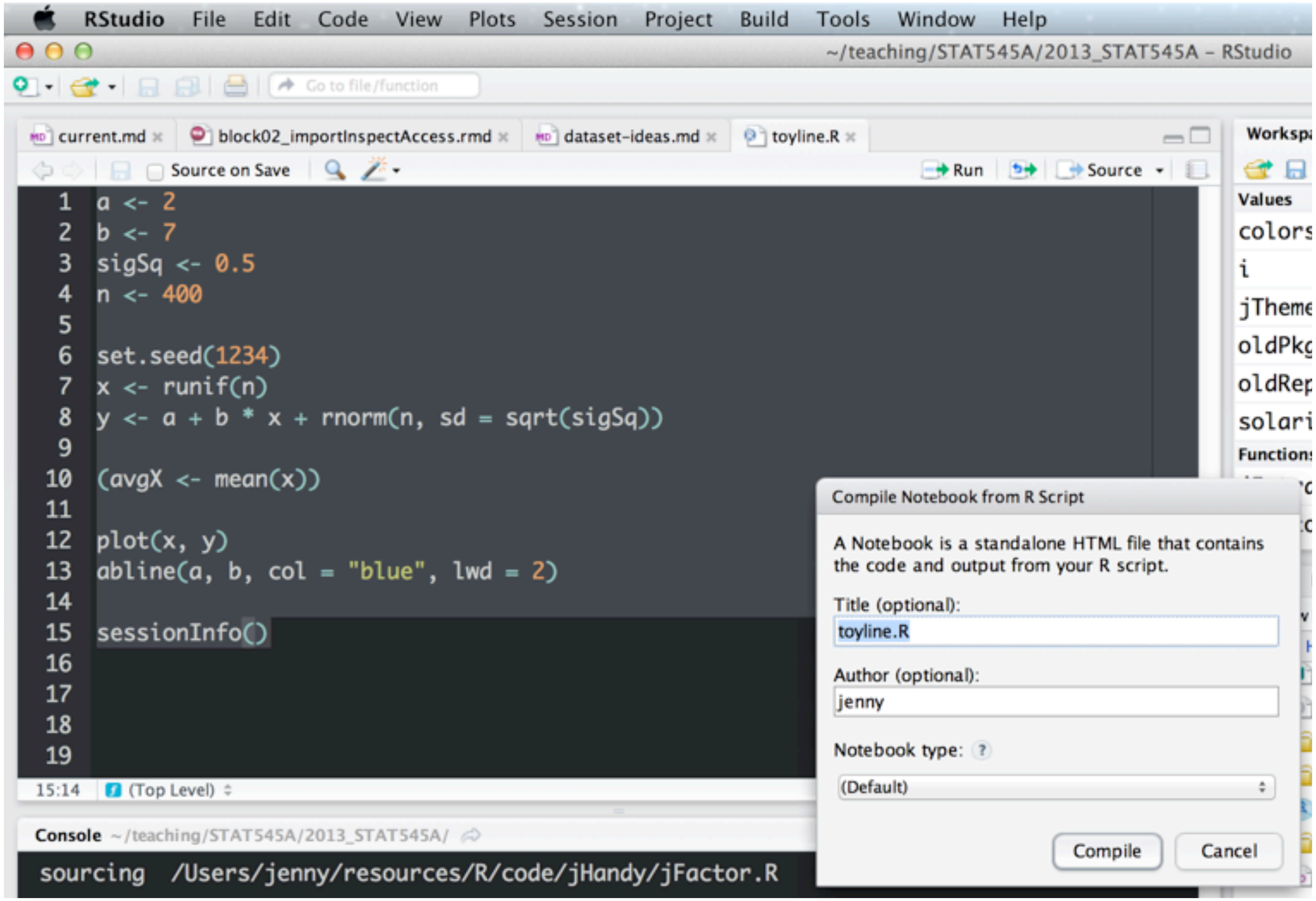

RStudio File Edit Code View Plots Session Project Build Tools Window Help
~/teaching/STAT545A/2013_STAT545A – RStudio
Go to file/function
current.md
block02_importInspectAccess.rmd
dataset-ideas.md
toyline.R
Source on Save
Run
Source
Workspace
Compile an HTML notebook from the current R script
Functions
Files
Plots
New Folder
Home
15:14
(Top Level)
R Script

```r
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))

plot(x, y)
abline(a, b, col = "blue", lwd = 2)

sessionInfo()
```

```
i
jTheme
oldPkgs
oldRepos
solarized
Functions
jExtract(
jFactor(
```

Compile an HTML notebook.

Yes this can be accomplished outside of RStudio, using knitr functions at the command line, so we are not creating unhealthy dependency on RStudio.



RStudio File Edit Code View Plots Session Project Build Tools Window Help
~/teaching/STAT545A/2013_STAT545A – RStudio
Go to file/function
current.md
block02_importInspectAccess.rmd
dataset-ideas.md
toyline.R
Source on Save
Run
Source
Workspace
Values
Functions
15:14
(Top Level)

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))

plot(x, y)
abline(a, b, col = "blue", lwd = 2)

sessionInfo()
```

Console ~/teaching/STAT545A/2013_STAT545A/

```
sourcing  /Users/jenny/resources/R/code/jHandy/jFactor.R
```

Compile Notebook from R Script

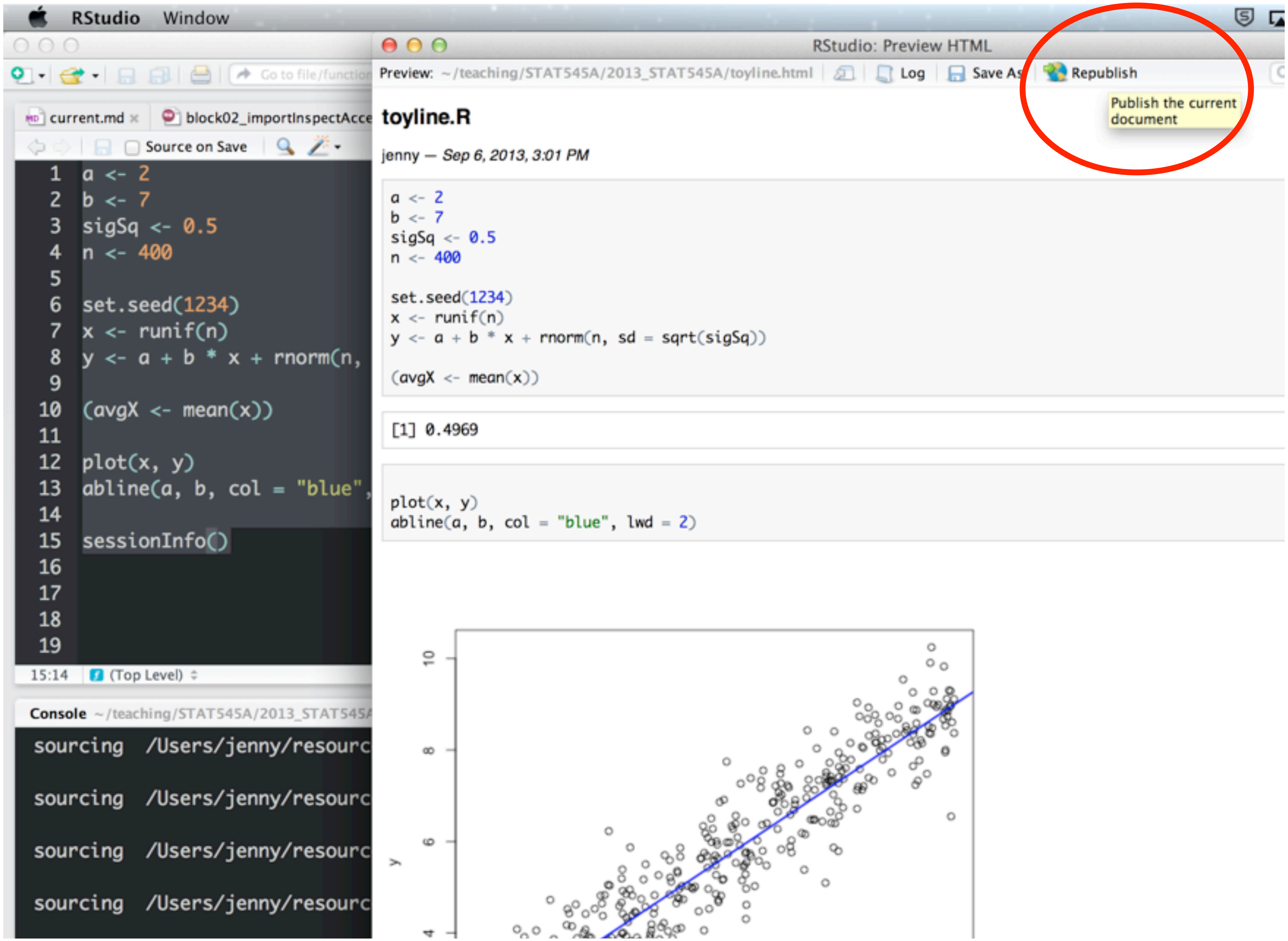
A Notebook is a standalone HTML file that contains the code and output from your R script.

Title (optional): toyline.R

Author (optional): jenny

Notebook type: (Default)

I just accept all these defaults.


RStudio
Window
Go to file/function
current.md
block02_importInspectAcce
Source on Save
15:14
(Top Level)
Console ~/teaching/STAT545A/2013_STAT545A
RStudio: Preview HTML
Preview: ~/teaching/STAT545A/2013_STAT545A/toyline.html
Log
Save As
Republish
Publish the current document
10
8
6
4
y

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n,

(avgX <- mean(x))

plot(x, y)
abline(a, b, col = "blue",

sessionInfo()
```

```
sourcing  /Users/jenny/resourc
sourcing  /Users/jenny/resourc
sourcing  /Users/jenny/resourc
sourcing  /Users/jenny/resourc
```

toyline.R

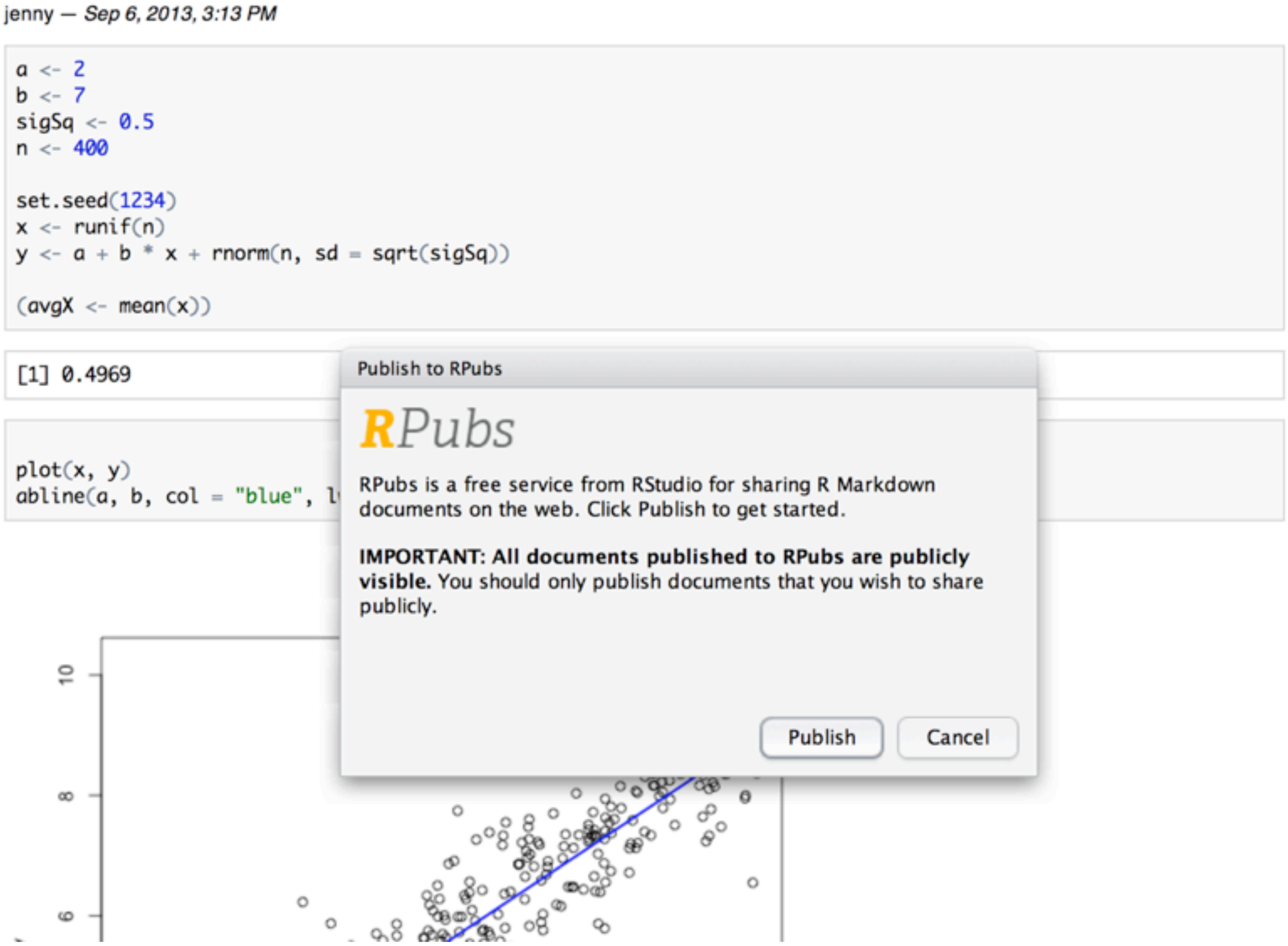
jenny — Sep 6, 2013, 3:01 PM

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))
```

```
[1] 0.4969
```

```
plot(x, y)
abline(a, b, col = "blue", lwd = 2)
```

This is where you'll need that RPubs account.



jenny — *Sep 6, 2013, 3:13 PM*

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))
```

```
[1] 0.4969
```

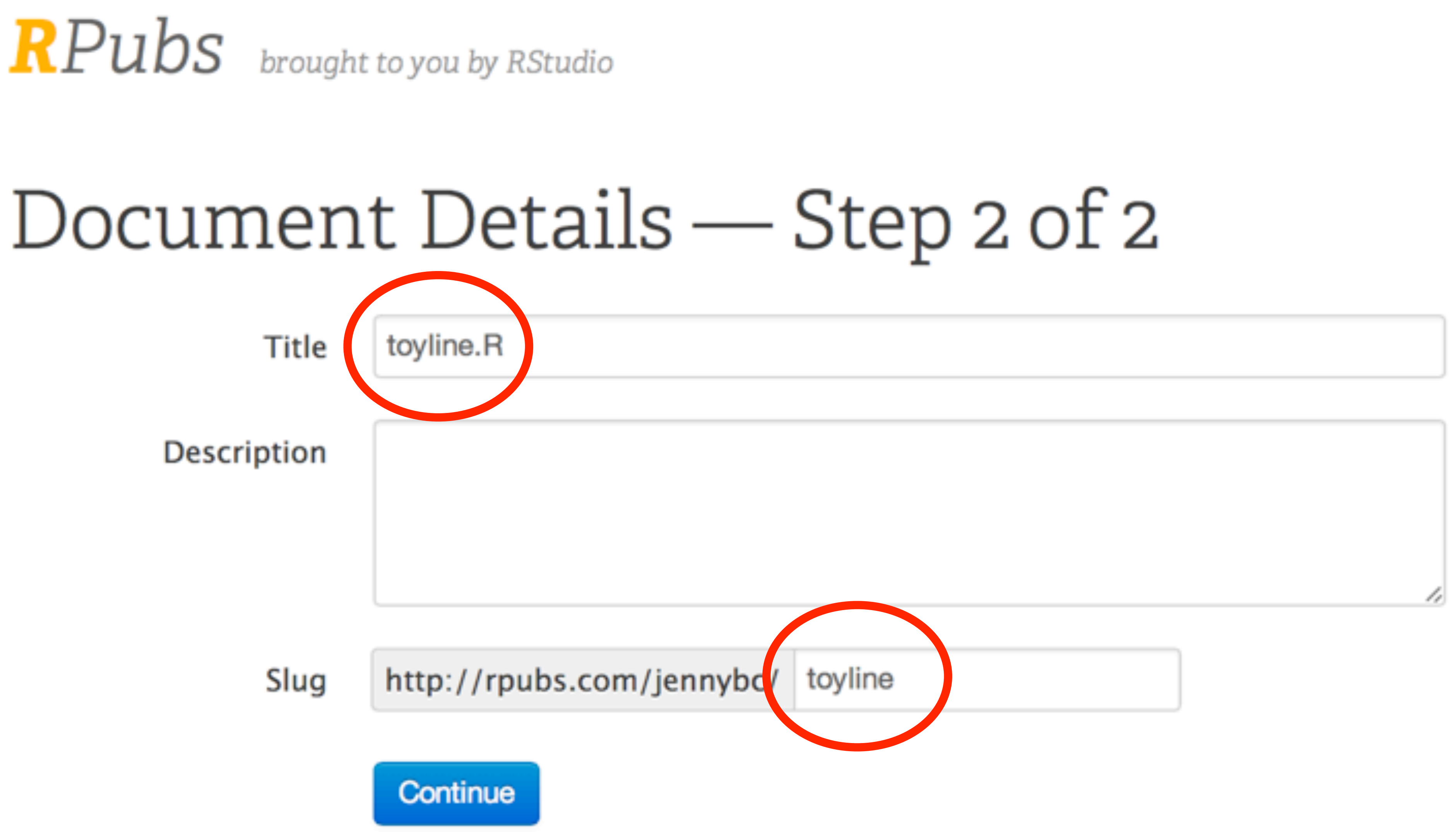
```
plot(x, y)
abline(a, b, col = "blue", l
```

Publish to RPubs

RPubs

RPubs is a free service from RStudio for sharing R Markdown documents on the web. Click Publish to get started.

IMPORTANT: All documents published to RPubs are publicly visible. You should only publish documents that you wish to share publicly.


RPubs brought to you by RStudio

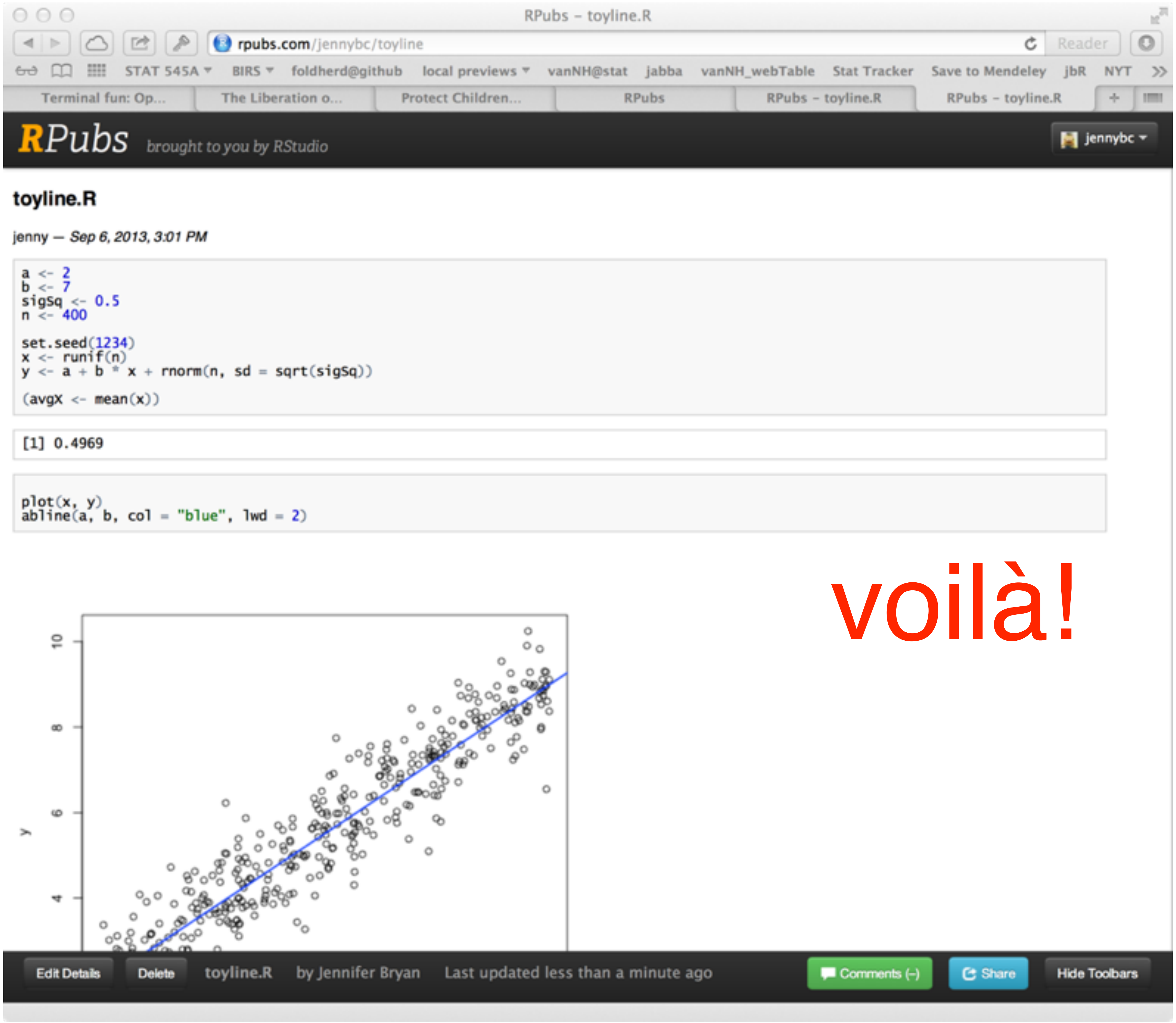
Document Details — Step 2 of 2

Title: toyline.R

Description:

Slug: http://rpubs.com/jennybc/ toyline

Seems like a good idea to keep script name and slug same, at least as default.
Expect me to give you a naming convention for future STAT 545A coursework.



toyline.R

jenny — *Sep 6, 2013, 3:01 PM*

```
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))
```

```
[1] 0.4969
```

```
plot(x, y)
abline(a, b, col = "blue", lwd = 2)
```

voilà!

http://rpubs.com/jennybc/toyline