# Debugging in RStudio

Amanda Gadrow
June 20, 2018

https://thumbs.dreamstime.com/b/lousing-monkeys-6160793.jpg

RStudio®

# Base R debug functions

```
debugonce()

debug(), undebug()

browser()

traceback()

options(error = browser), options(error = NULL)
```

# RStudio IDE debugging tools

- Editor breakpoint
- Browser breakpoint
- Debug mode
  - Source pane highlighting
  - Context-specific Environment pane, including a Traceback panel
  - Browser prompt in Console pane
- “Show traceback” shortcut
- “Rerun with debug” shortcut
- Debug menu

# Shiny debugging

```
options(shiny.fullstacktrace = TRUE)

options(shiny.error = browser)

browser()
```

# Shiny tracing

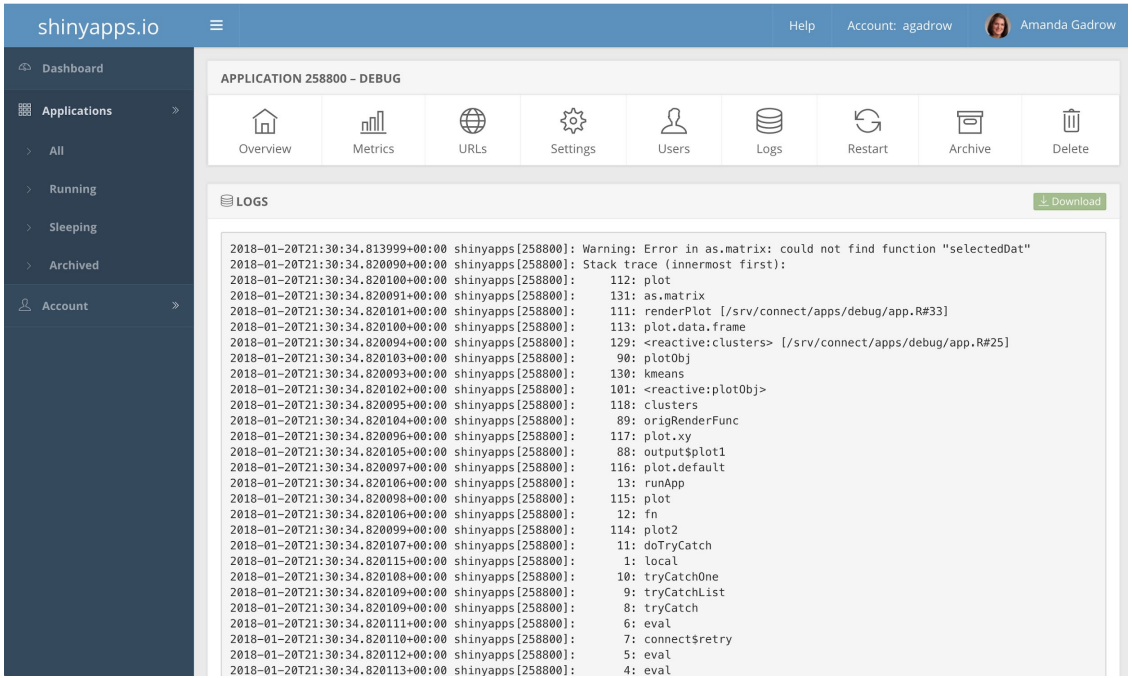
```
runApp(appDir = “myApp”, display.mode = “showcase”)

options(shiny.reactlog=TRUE)
runApp(...)
showReactLog()

cat(file=stderr(), ...)
```

# shinyapps.io debugging

```
rsconnect::showLogs(streaming = TRUE)
```



# RStudio Connect debugging

Content / cars_source_with_source

amanda

## R Markdown Example

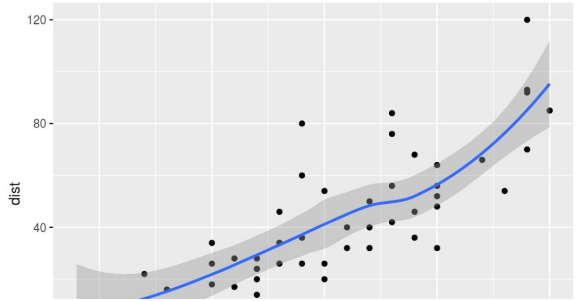
*2015-10-01 19:00:00*

```
library(ggplot2)
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

```
qplot(speed, dist, data=cars) + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```




Info Access Runtime Schedule Tags Logs

Jobs

**7734** g7sFpnnkfdObmpWD
*Document render ran a few seconds ago in a few seconds*

**7557** dx1VGifhHvWvON7X
*Document render ran a minute ago in a few seconds*

2 jobs.

Log Download

```
01/22 20:21:04.718
Using Packrat dir /opt/rstudio-connect/mnt/app/packrat/lib/x86_64-pc-linux-gnu/3.3.1
01/22 20:21:04.721
LANG: en_US.UTF-8
01/22 20:21:04.721
R version: 3.3.1
01/22 20:21:04.721
rmarkdown version: 1.3
```

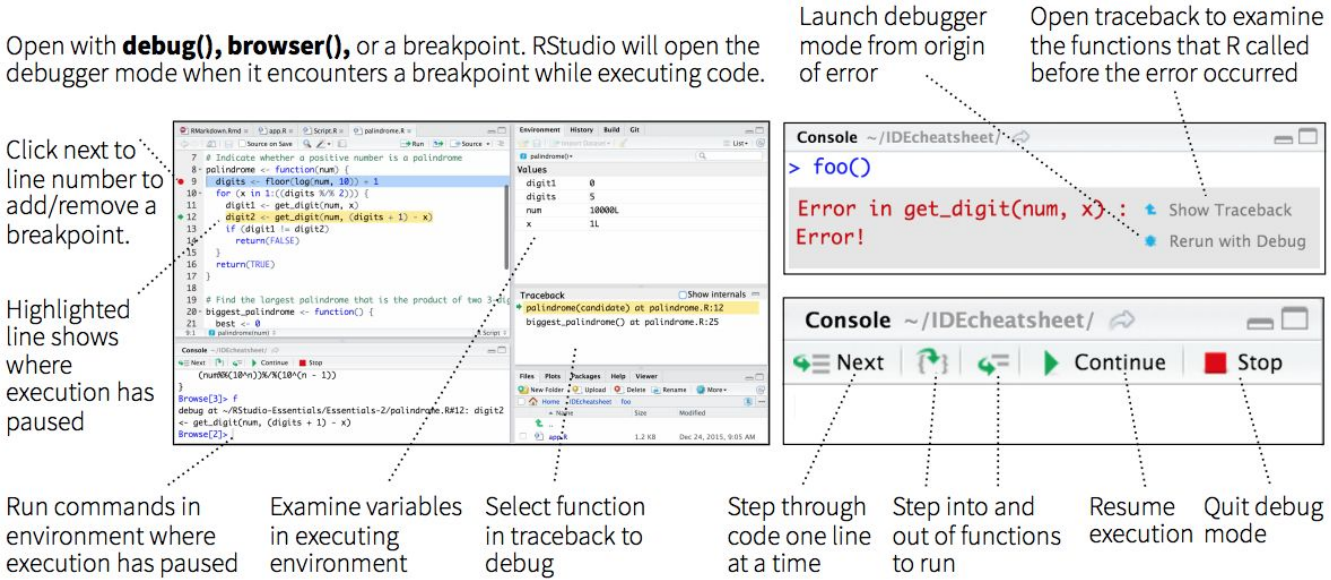
## Useful commands in base R

```
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)




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®