Grammar of Graphics

Jeff Hemsley, PhD
Assistant Professor
School of Information Studies

School of Information Studies
Syracuse University

ggplot2

Hadley Wickham

- Chief scientist at Rstudio, professorial appointments at University of Auckalnd, Rice, and Stanford

Grammar of graphics: "gg" in ggplot

Grammar: a system and structure of a language. Rules that govern composition, clauses, and phrases

Grammar of graphics: systematic and consistent rules

Graphic built up from blocks

- A function call for each block
- Defaults

Ggplot2: Building Blocks

Data + Aesthetic Mapping + Geometric Object + Statistical Transformations + Faceting + Scales + Coordinate System + Position Adjustments

Ggplot2: Building Blocks

Data

- Must be a dataframe

Aesthetic mapping

- The variables you will **see**; a way of seeing the data
- Position, size, color, shape, etc.

Geometric object

- The way we represent the data; visual encoding
- Boxplot, hist, lines, points

Ggplot2: Building Blocks

Statistical transformation (stats)

- Every geom has a default stat—"hidden"
  - Boxplot: find quartiles and medians
  - Barplot: ...count...
- Identity, binning, aggregating, log10, log2
- Transform data before plotting

Faceting

- Conditioning plot (trellising, latticing, and small multiples)
- Data subsetting

Ggplot2: Building Blocks

Scales

- Maps the data to the actual aesthetic: 1-to-1 or scale-aesthetic
- Provides consistency across the plot
- Scales x, y, color, shape <- anything you can see has scale
- Control axis and legends

Coordinate system

- Cartesian and polar
- Not used much

Position adjustments

- Jitter, identity, dodge, stack, fill

Base R Graphics vs. ggplot

ggplot +

- Defaults look good (?)
- Grammar allows for building up complex plots
- Less code for very complicated graphs
- Rich environment with lots of developers
- Packages that extend ggplot

ggplot -

- More code for simple graphics
- Always requires a df
- Inhibits $ and auto/tab var selection
- Requires learning a new language on top of learning R
- **Generally slower**
- **Copies objects so uses more memory**

Religious argument (love or hate)

Some Comparisons

Base R graphics vs. ggplot

The R-memory Model and Graphics

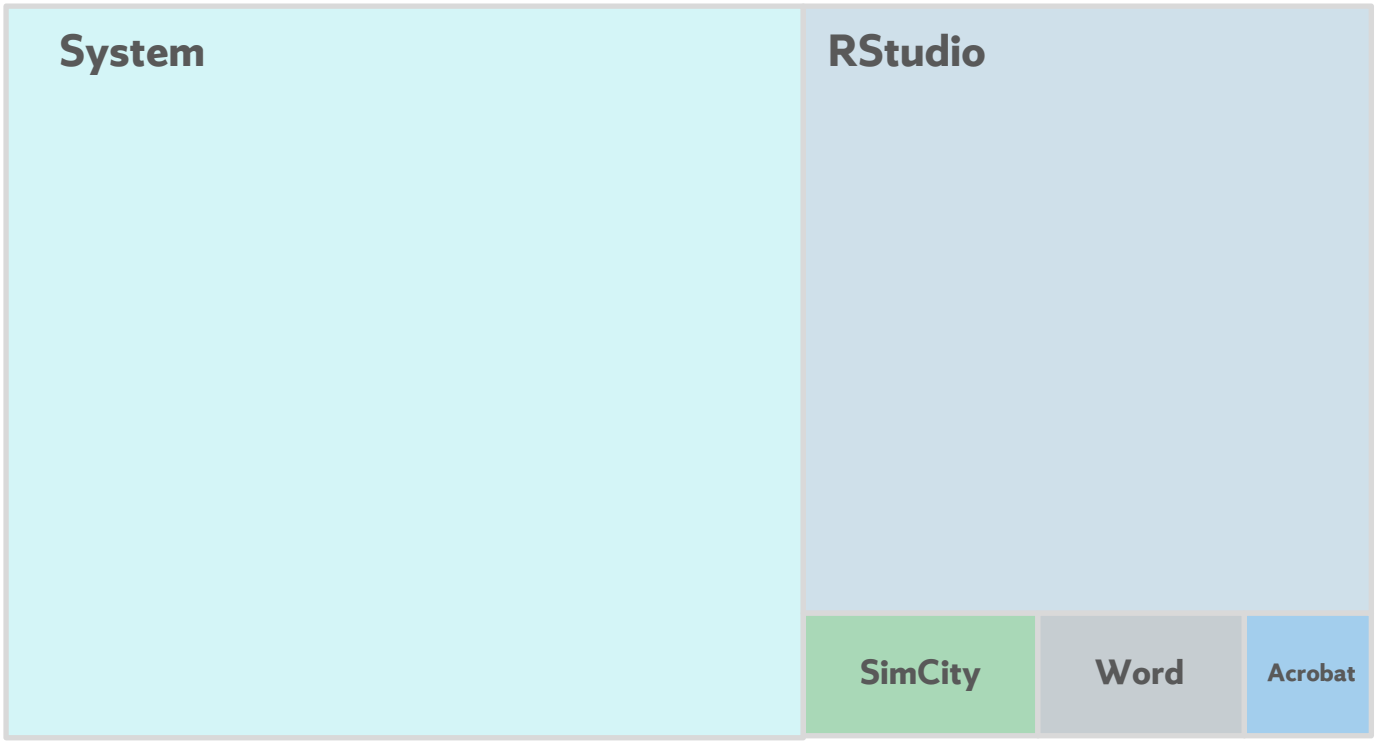
School of Information Studies
Syracuse University

Jeff Hemsley, PhD
Assistant Professor
School of Information Studies

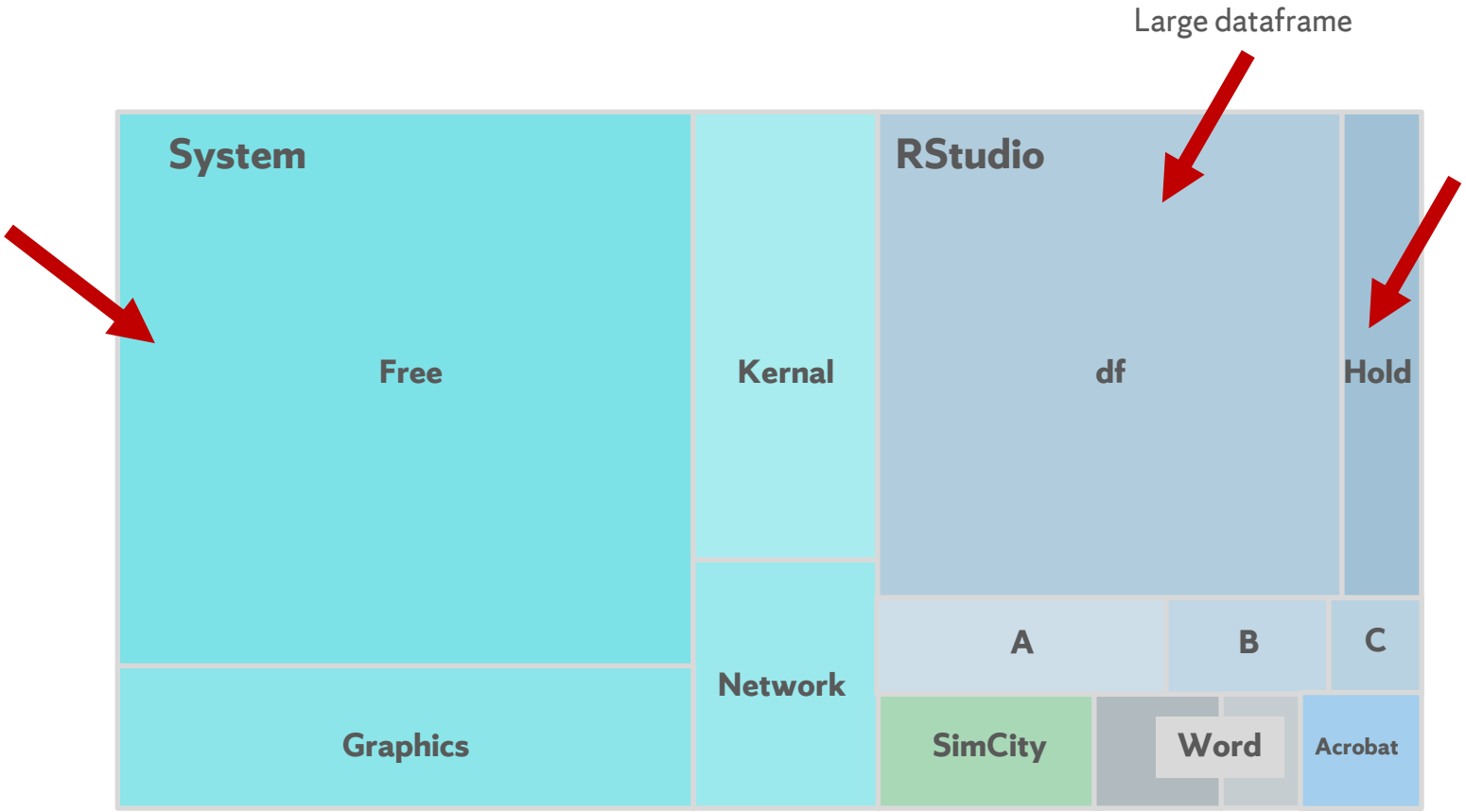
Operating System Memory

```
> memory.size()
41.89
> rts <- read.csv(fname)
> dim(rts)
6815712, 15 # almost 7 million rows
> object.size(rts)
1,826,819,880 bytes # 2GB!
> p <- ggplot(rts) + aes(x, y)
> object.size(p)
1,826,822,576 bytes # 2GB!
> memory.size()
[1] 2890.42 # almost 3GB
> p + geom_point() # wait 20 minutes for plot
> memory.size()
[1] 4107.15 # R has 4GB of memory
```

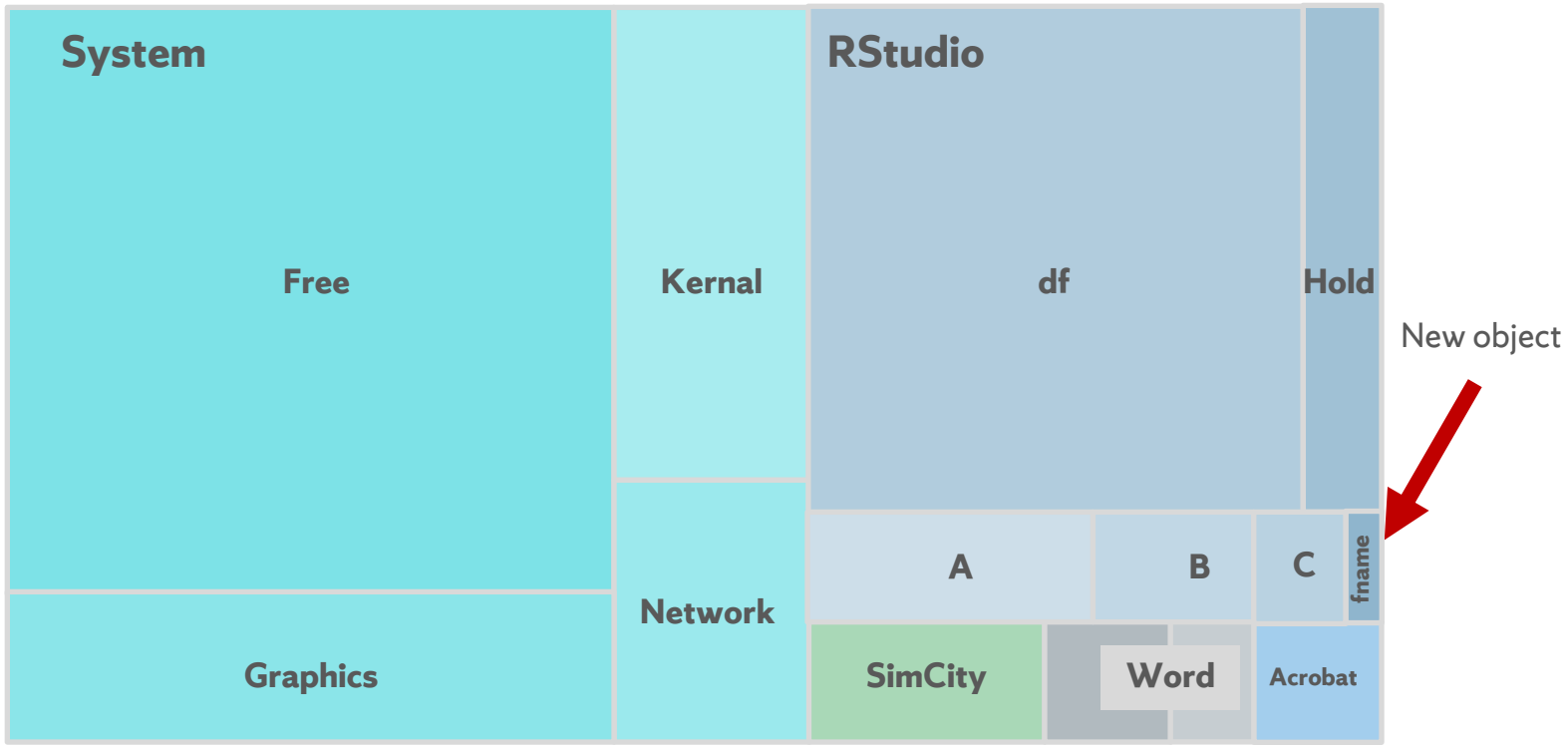
Operating System Memory



Operating System Memory

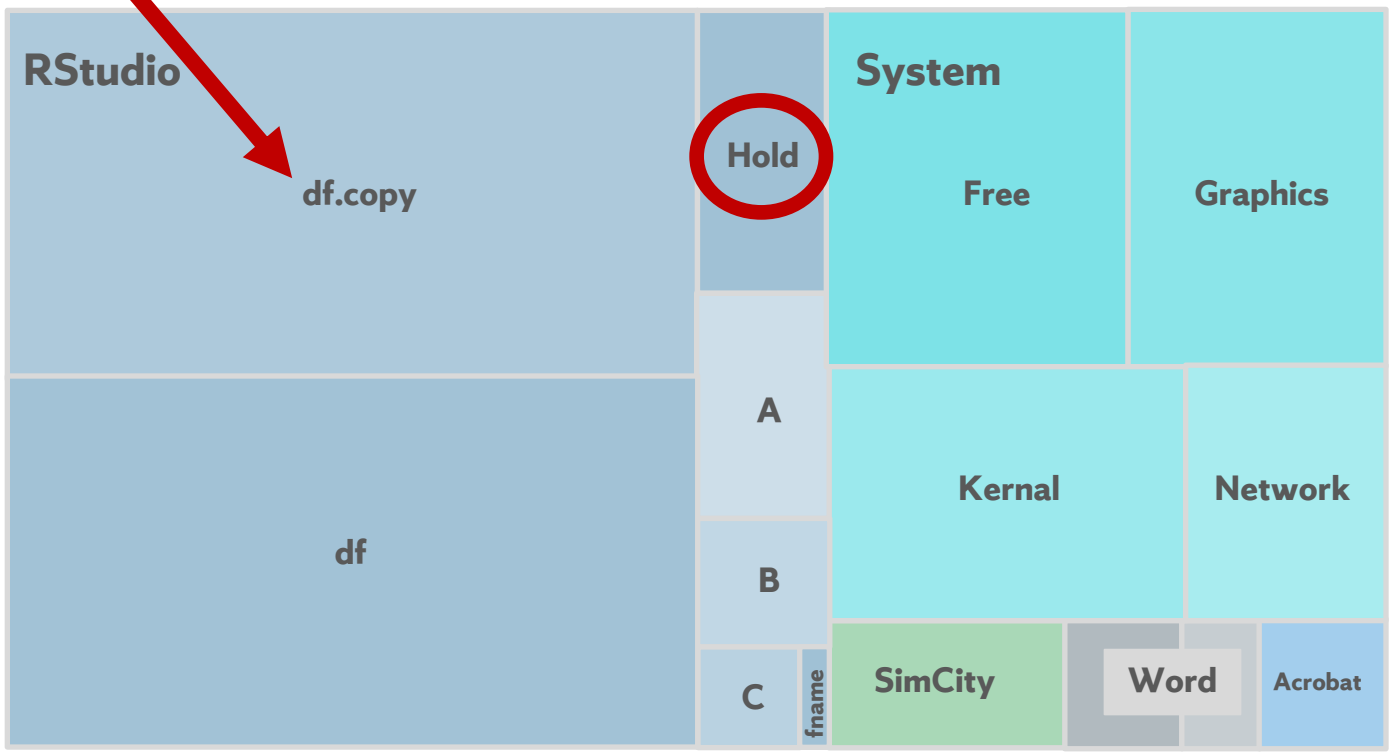


Operating System Memory



Operating System Memory

```
p <- ggplot(rts) + aes(x, y)
```

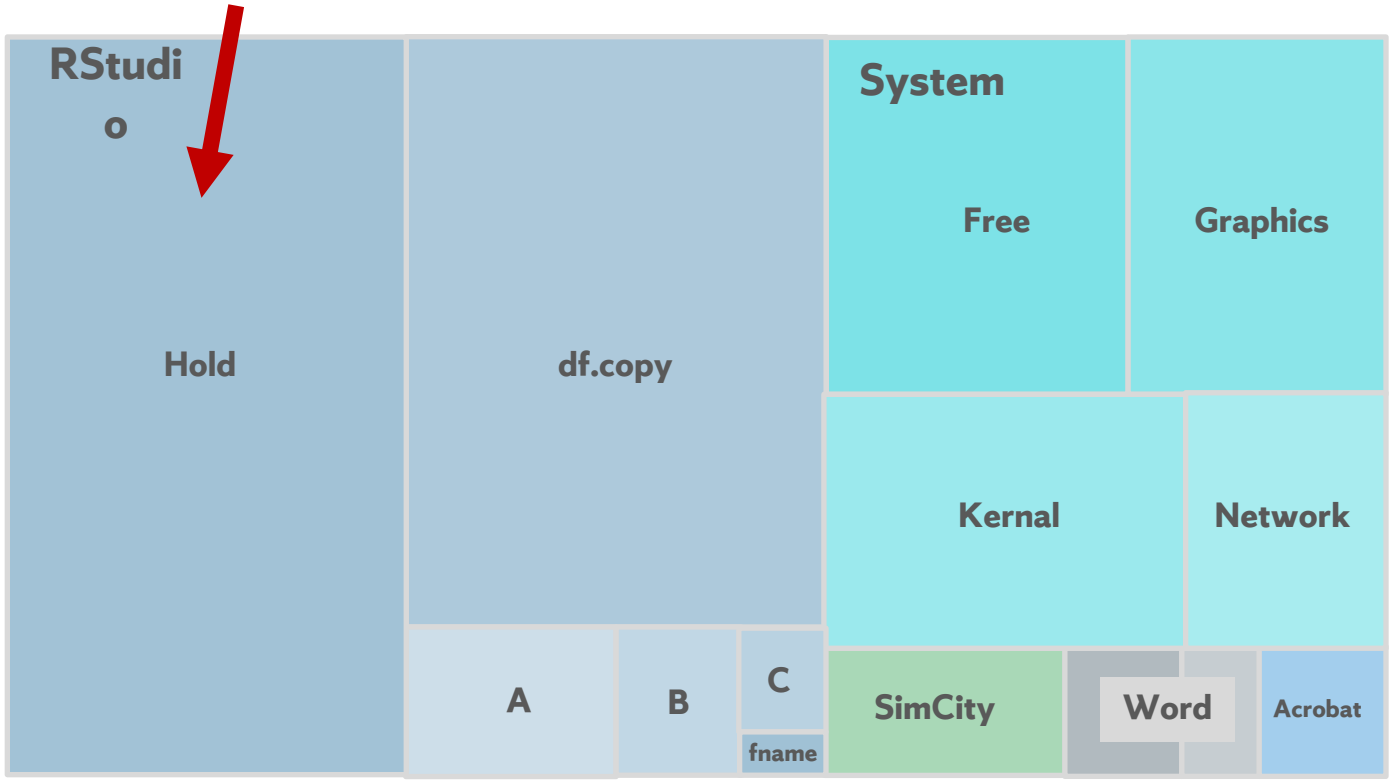


Operating System Memory

```
> memory.size()
[1] 4107.15 # R has 4GB of memory
> rm(rts) # removes the r object rts
> memory.size()
[1] 4107.15 # ??????
```

Operating System Memory

```
rm(rts)
```

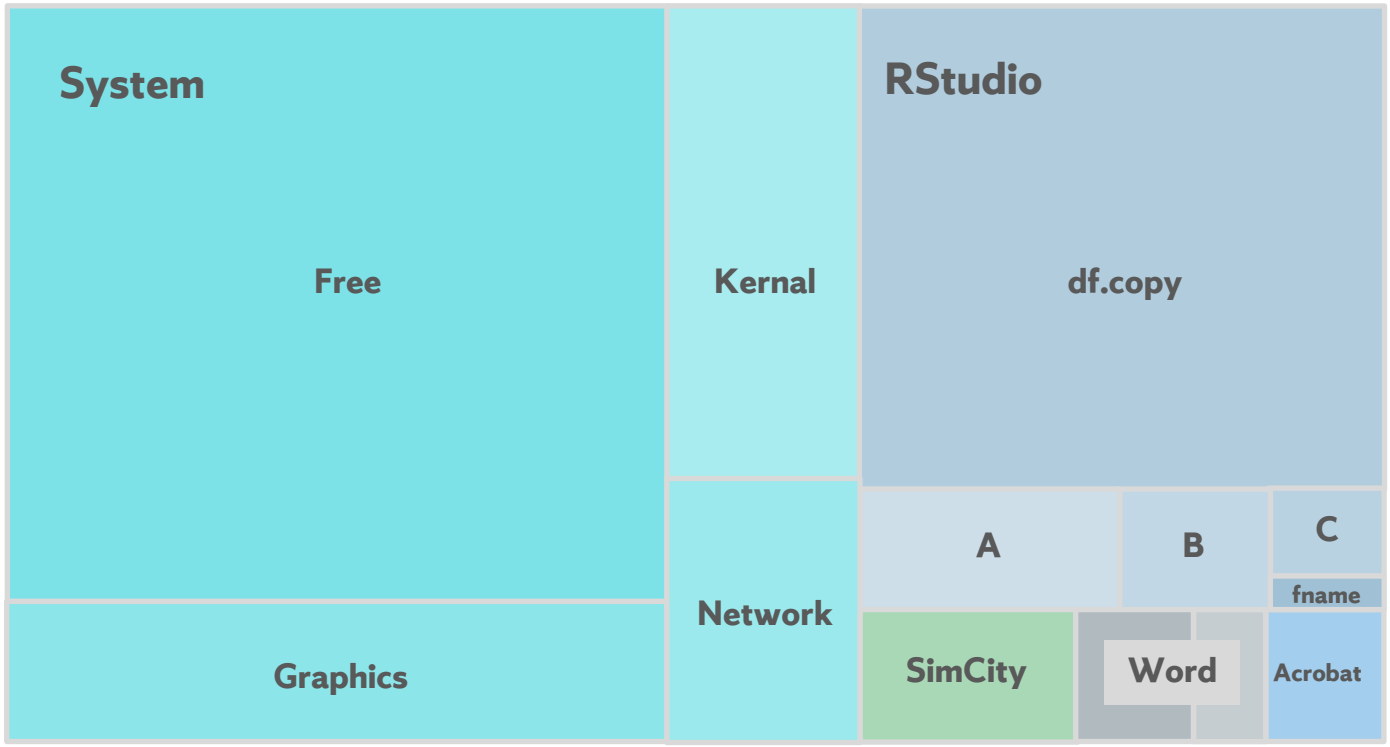


Operating System Memory

```
> memory.size()
[1] 4107.15
> gc()
used (Mb) gc trigger (Mb) max used (Mb)
Ncells 178005 950.7 30163 1610.9 2109 1121.7
Vcells 338131 257.8 63725 4861.9 5293 4038.6
> memory.size()
[1] 3338.7
> dev.off()
> gc()
> memory.size()
[1] 1971.29
```

Operating System Memory

```
gc() # function to free memory
```



Operating System Memory

R functions you should know

- `object.size()`
- `memory.size()`
- `gc()`
- `dev.off()`

Pro tip: If R is opening slowly, clean out the memory, shut it down, then restart.