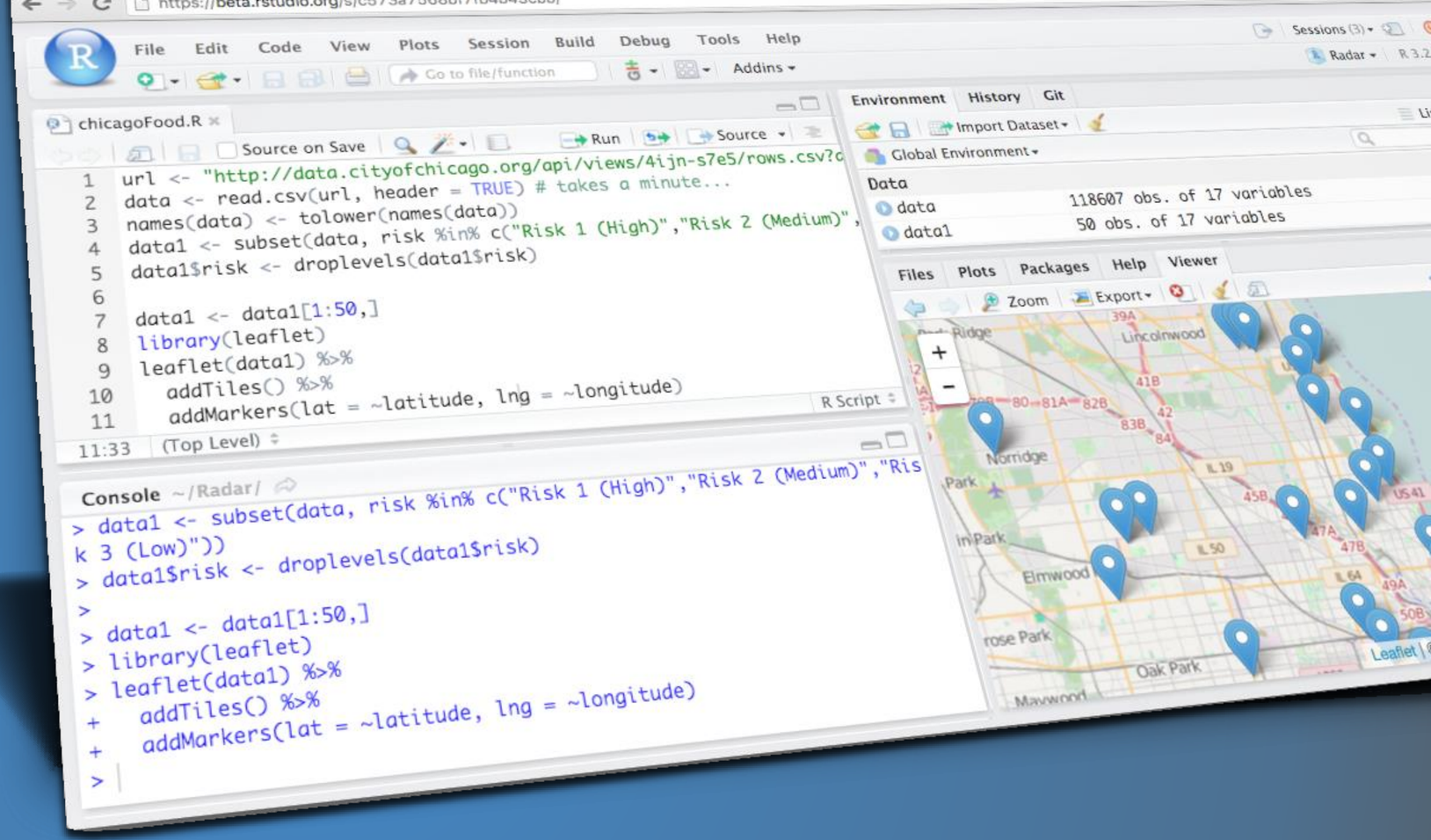
RStudio®



TOOLS FOR DATA SCIENCE IN R

12/29/16

RStudio
250 Northern Ave, Boston, MA 02210
Phone: 844-448-1212
Email: info@rstudio.com
Web: http://www.rstudio.com

All Training materials are provided "as is" and without warranty and RStudio disclaims any and all express and implied warranties including without limitation the implied warranties of title, fitness for a particular purpose, merchantability and noninfringement.

Safe Harbor

This presentation is confidential and intended for existing and prospective customers of RStudio. It may not be distributed to others without permission.

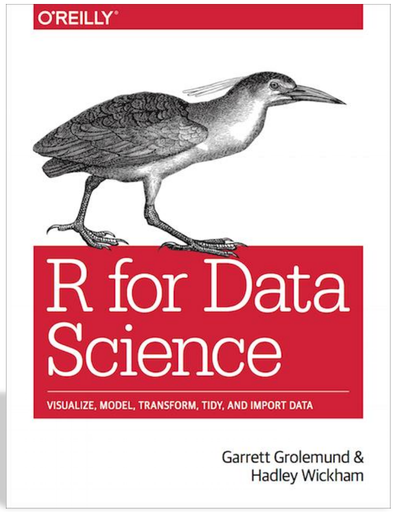
Any future plans are best guesses and may change at any time due to market conditions. Nothing in this presentation should be construed as a commitment.

RStudio®

RStudio®

What is Data Science:

“Data science is an exciting discipline that allows you to turn raw data into understanding, insight, and knowledge. The goal of ‘R for Data Science’ is to introduce you to the most important in R *tools* that you need to do data science.”



http://r4ds.had.co.nz/intro.html

“Data science is a term I use to represent the ownership and management of the *entire modeling process*: discovering the true business need, collecting data, managing data, building models and deploying models into production.”

http://www.win-vector.com/blog/2013/04/data-science-machine-learning-and-statistics-what-is-in-a-name/

How we think about Data Science

Import

Tidy

Transform

Visualise

Model

Communicate

Understand

R for Data Science, Grolemund & Wickham

What is R?

- Open source software environment for statistical computing and graphics
- Created over 20 years ago by two statistics professors in New Zealand
- Based on S, a similar environment originally developed at AT&T Bell Labs

Why R?

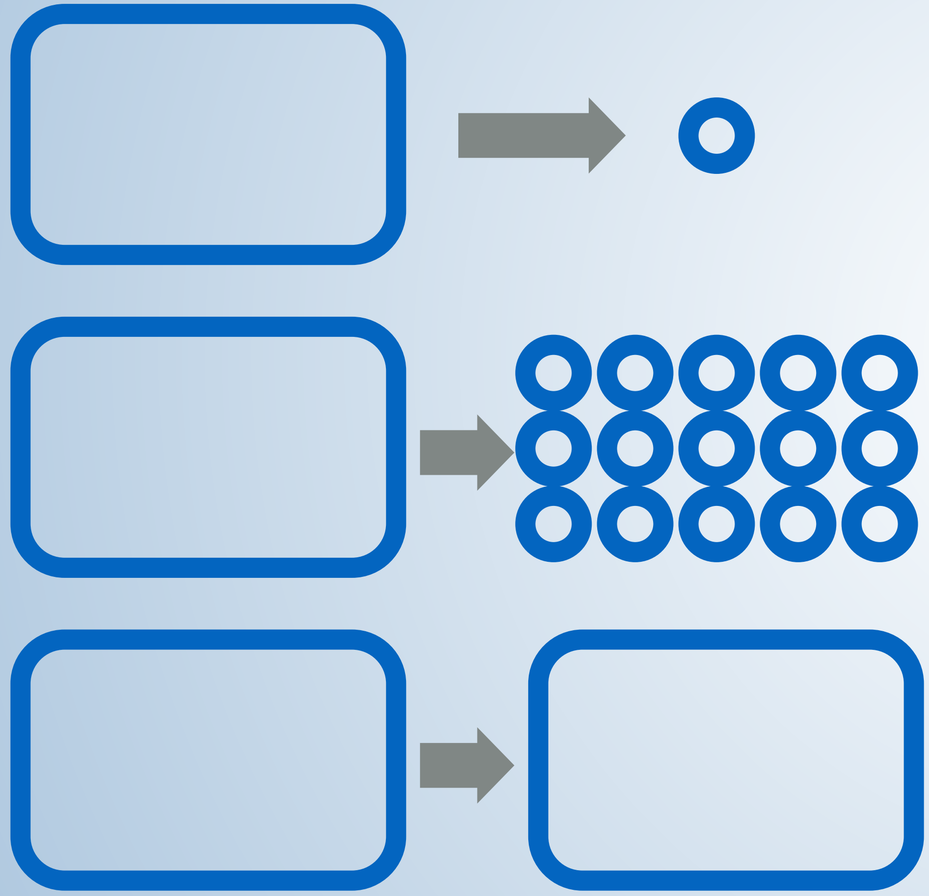
- It's free, open source, and available on every major platform. As a result, if you do your analysis in R, anyone can easily replicate it.
- A massive set of packages for statistical modelling, machine learning, visualization, and importing and manipulating data.
- Cutting edge tools - a powerful and flexible toolkit which allows you to write concise yet descriptive code.
- A fantastic community.
- Powerful tools for communicating your results…*Data Products*

FREE & OPEN SOURCE R PACKAGES

- Massive set of packages – >10K
- RStudio employees contribute extensively to the R community
- This includes the most widely used and newest open-source libraries for data manipulation (plyr and dplyr), data visualization (ggplot2 and ggvis), and publishing (knitr, R Markdown...)
- RStudio packages provide powerful productivity tools for R (packrat, devtools, roxygen2, testthat)
- Next generation of interactive web enabled graphics for R

RStudio

Scaling with R



1. Extract data

Sample into R Memory

2. Compute on the parts

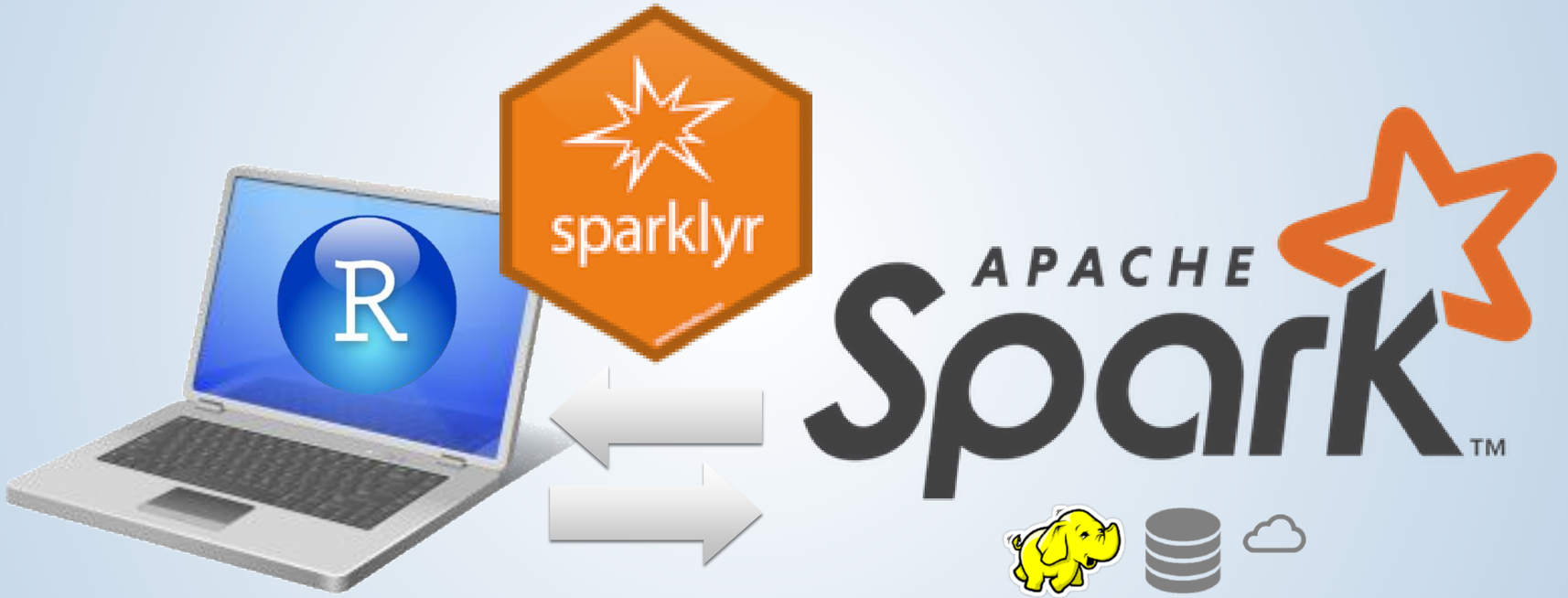
HPC, R parallel, etc.

3. Compute on the whole

sparklyr, MR, etc.

Sparklyr

Makes it easy to use R with Spark



[Spark with AWS EMR]

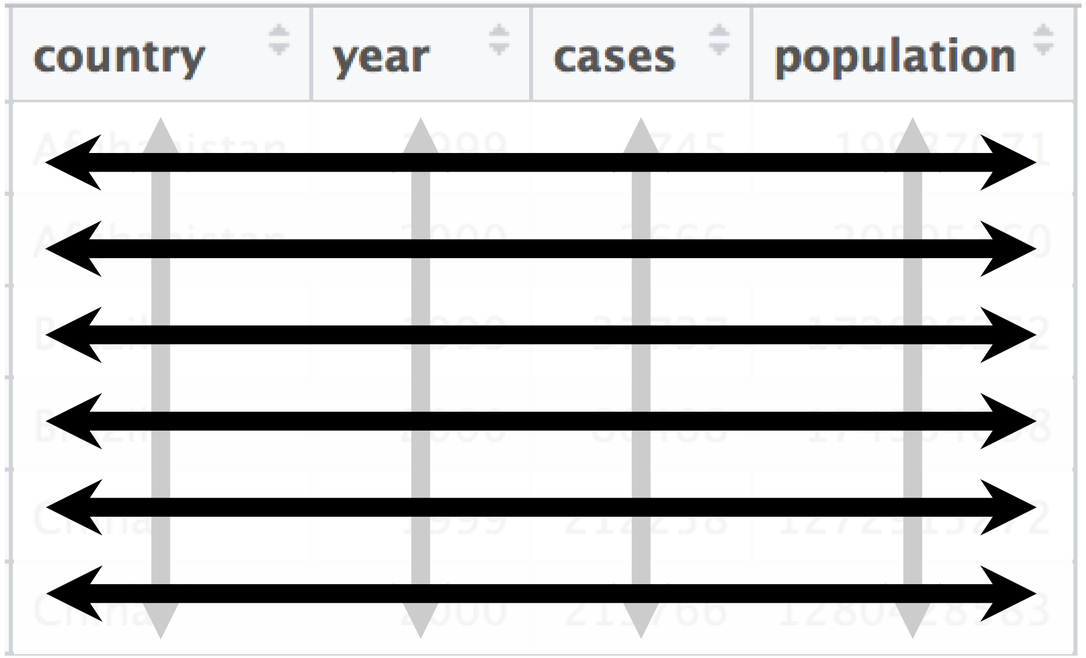
[Spark and the data lake]

Data Types

- R has a wide variety of data types…
- Vectors
- Lists
- **Matrix**
- **Factors**
- **Data frame**
- **Tibble**

tibbles

Tidy data



A data set is tidy iff:

1. It is a data frame (e.g. tibble)
2. Each variable is in its own column
3. Each case is in its own row

Standardized framework allows your data to fit nicely into the APIs provided by base R and other key packages

Examining data frames in base R can be a challenge…

```
dim(iris)
str(iris)
iris
```

Coerce a data frame to a tibble, now it's pretty!...


```
tibble_iris <- tbl_df(iris)
tibble_iris
```

Or

```
tibble(treatment = sample(letters, 10),
       expression_value = rnorm(10, 100, 10))
```

tibbles

A type of data frame common throughout tidyverse packages.
Tibbles enhance data frames in three ways:

1. Subsetting - [always returns a new tibble, [[and $ always return a new vector
2. No partial matching - You must use full column names when subsetting
3. Display - When you print a tibble, R provides a concise view of the data that fits on one screen

tibble

A package with several helper functions for tibbles:

- as_tibble() - convert a data frame to a tibble
- as.data.frame() - convert a tibble to a data frame
- tribble() - make a tibble (transversed)

```
tribble(
   ~x,  ~y,
    1, "a",
    2, "b",
    3, "c")
```

x	y
1	a
2	b
3	c

babynames

Names of male and female babies born in the US from 1880 to 2008. 1.8M rows.

```
# install.packages("babynames")
library(babynames)
```

View(babynames)

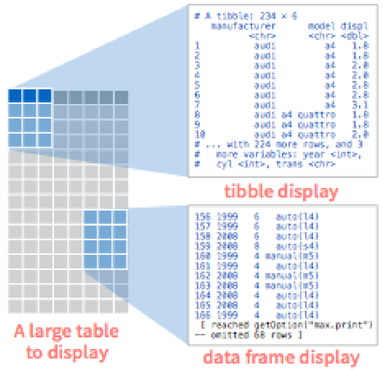
	year	sex	name	n	prop
1	1880	F	Mary	7065	0.0723835869
2	1880	F	Anna	2604	0.0266789611
3	1880	F	Emma	2003	0.0205214897
4	1880	F	Elizabeth	1939	0.0198657856
5	1880	F	Minnie	1746	0.0178884278
6	1880	F	Margaret	1578	0.0161672045
7	1880	F	Ida	1472	0.0150811946
8	1880	F	Alice	1414	0.0144869628
9	1880	F	Bertha	1320	0.0135238973
10	1880	F	Sarah	1288	0.0131960453
11	1880	F	Annie	1258	0.0128886840

tibbles

Tibbles - an enhanced data frame

The **tibble** package provides a new S3 class for storing tabular data, the tibble. Tibbles inherit the data frame class, but improve two behaviors:

- **Display** - When you print a tibble, R provides a concise view of the data that fits on one screen.
- **Subsetting** - [always returns a new tibble, [[and $ always return a vector.
- **No partial matching** - You must use full column names when subsetting



- Control the default appearance with options:
  options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf)
- View entire data set with **View**(x, title) or **glimpse**(x, width = NULL, …)
- Revert to data frame with **as.data.frame()** (required for some older packages)

Construct a tibble in two ways

Both make this tibble

tibble(…)
Construct by columns.

```
tibble(x = 1:3,
       y = c("a", "b", "c"))
```

tribble(…)
Construct by rows.

```
tribble(
  ~x, ~y,
   1, "a",
   2, "b",
   3, "c")
```

```
A tibble: 3 × 2
      x     y
  <int> <dbl>
1     1     a
2     2     b
3     3     c
```

as_tibble(x, …) Convert data frame to tibble.

enframe(x, name = "name", value = "value")
Converts named vector to a tibble with a names column and a values column.

is_tibble(x) Test whether x is a tibble.

Tidy Data with tidyr

Tidy data is a way to organize tabular data. It provides a consistent data structure across packages.

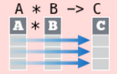
A table is tidy if:

Each **variable** is in its own **column**

Each **observation**, or **case**, is in its own **row**

Tidy data:

Makes variables easy to access as vectors



Preserves cases during vectorized operations

Reshape Data - change the layout of values in a table

Use **gather()** and **spread()** to reorganize the values of a table into a new layout. Each uses the idea of a key column: value column pair.

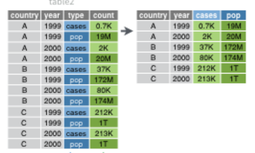
gather(data, key, value, …, na.rm = FALSE, convert = FALSE, factor_key = FALSE)

Gather moves column names into a key column, gathering the column values into a single value column.

gather(table4a, `1999`, `2000`, key = "year", value = "cases")

spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE, sep = NULL)

Spread moves the unique values of a key column into the column names, spreading the values of a value column across the new columns that result.



spread(table2, type, count)

Handle Missing Values

drop_na(data, …)
Drop rows containing NA's in … columns.

drop_na(x, x2)

fill(data, …, direction = c("down", "up"))
Fill in NA's in … columns with most recent non-NA values.

fill(x, x2)

replace_na(data, replace = list(), …)
Replace NA's by column.

replace_na(x,list(x2 = 2), x2)

Expand Tables - quickly create tables with combinations of values

complete(data, …, fill = list())
Adds to the data missing combinations of the values of the variables listed in …
complete(mtcars, cyl, gear, carb)

expand(data, …)
Create new tibble with all possible combinations of the values of the variables listed in …
expand(mtcars, cyl, gear, carb)

Split and Combine Cells

Use these functions to split or combine cells into individual, isolated values.

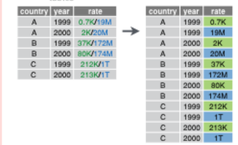
separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE, convert = FALSE, extra = "warn", fill = "warn", …)

Separate each cell in a column to make several columns.

separate_rows(table3, rate, into = c("cases", "pop"))

separate_rows(data, …, sep = "[^[:alnum:].]+", convert = FALSE)

Separate each cell in a column to make several rows. Also **separate_rows_()**.



separate_rows(table3, rate)

unite(data, col, …, sep = "_", remove = TRUE)

Collapse cells across several columns to make a single column.

unite(table5, century, year, col = "year", sep = "")

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio info@rstudio.com • 844-448-1212 • rstudio.com

Learn more at **browseVignettes(package = c("readr", "tibble", "tidyr"))** • readr 1.1.0 • tibble 1.2.12 • tidyr 0.6.0 • Updated: 2016-12

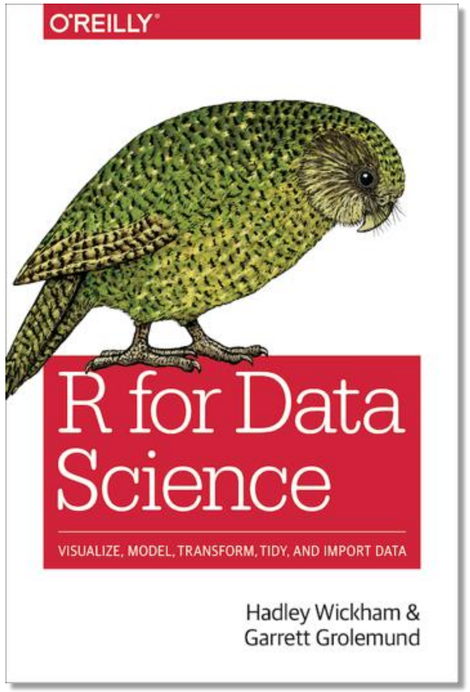
Import Data with

Tidyverse

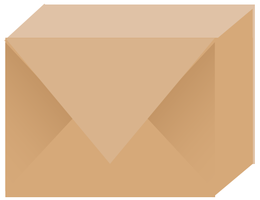
A collection of R packages that share common philosophies and are designed to work together.

tidyverse.org

install.packages("tidyverse")
library(tidyverse)



tidyverse



An R package that serves as a short cut for installing and loading the components of the tidyverse.

```
install.packages("tidyverse")
```

does the equivalent of

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("hms")
install.packages("stringr")
install.packages("lubridate")
install.packages("forcats")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

```
install.packages("tidyverse")
```

does the equivalent of

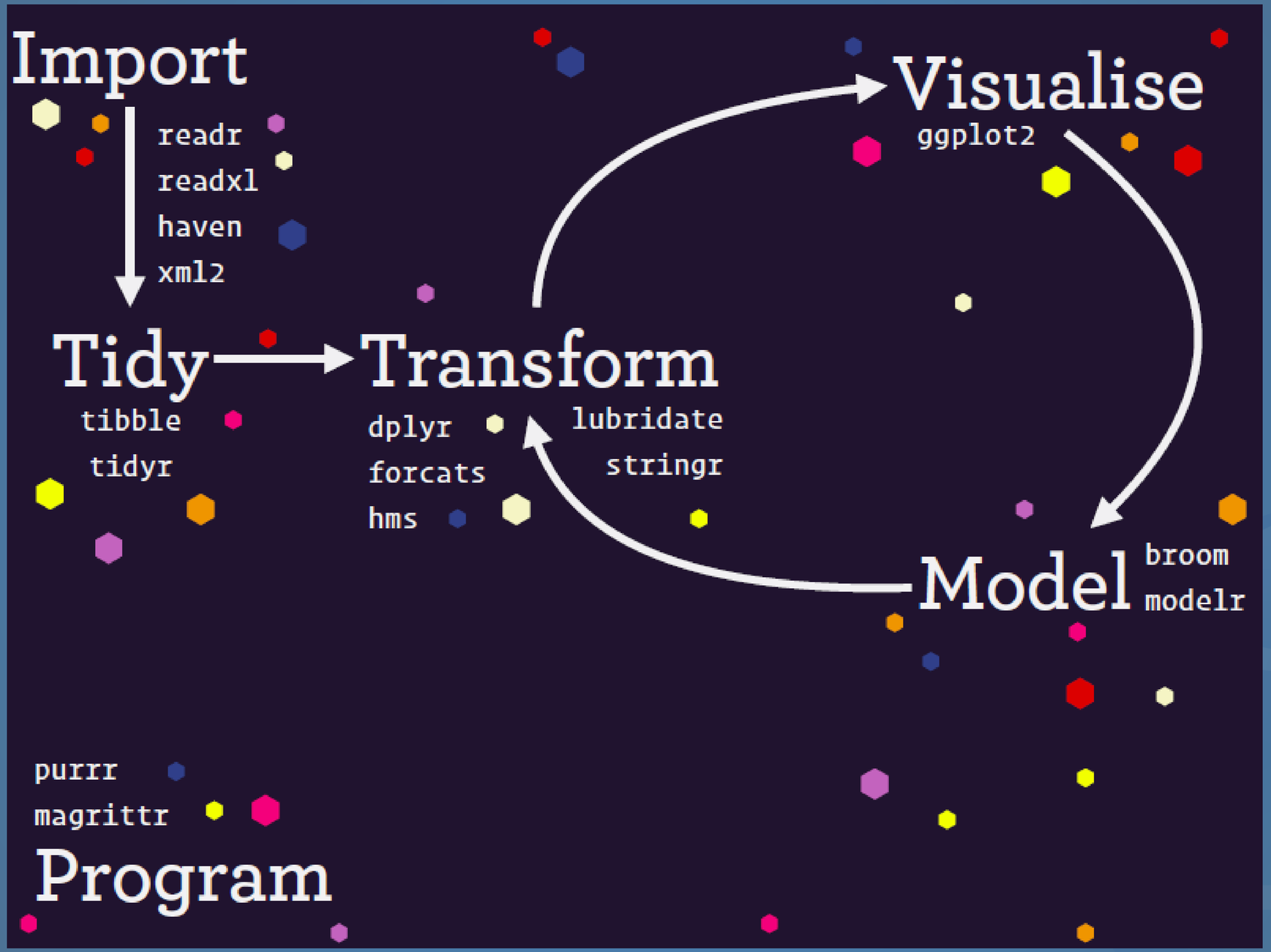
```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("hms")
install.packages("stringr")
install.packages("lubridate")
install.packages("forcats")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

```
library("tidyverse")
```

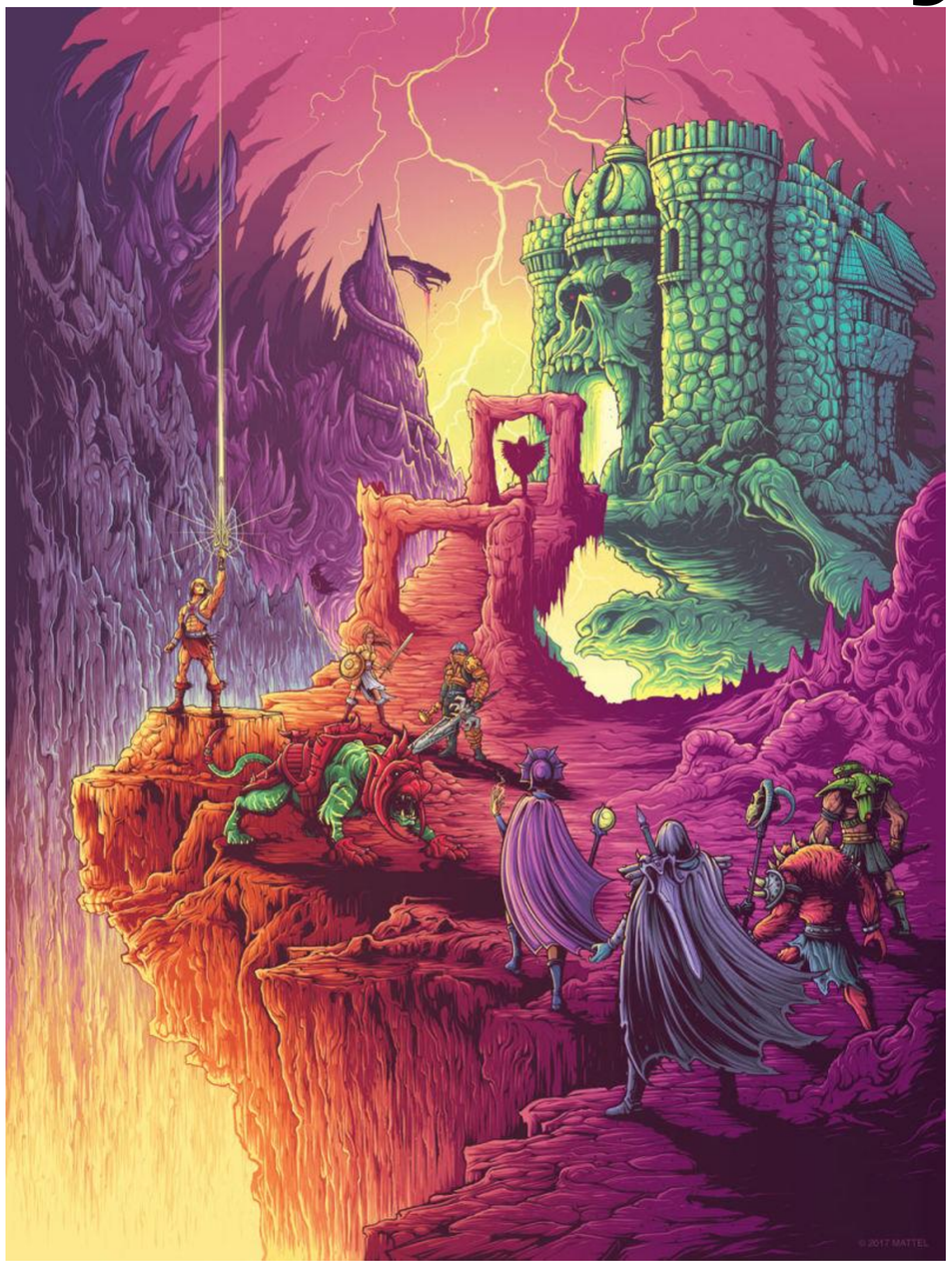
does the equivalent of

```
library("ggplot2")
library("dplyr")
library("tidyr")
library("readr")
library("purrr")
library("tibble")
```


tidyverse
www.rstudio.com


Import
readr
readxl
haven
xml2
Visualise
ggplot2
Tidy
tibble
tidyr
Transform
dplyr
forcats
hms
lubridate
stringr
Model
broom
modelr
purrr
magrittr
Program

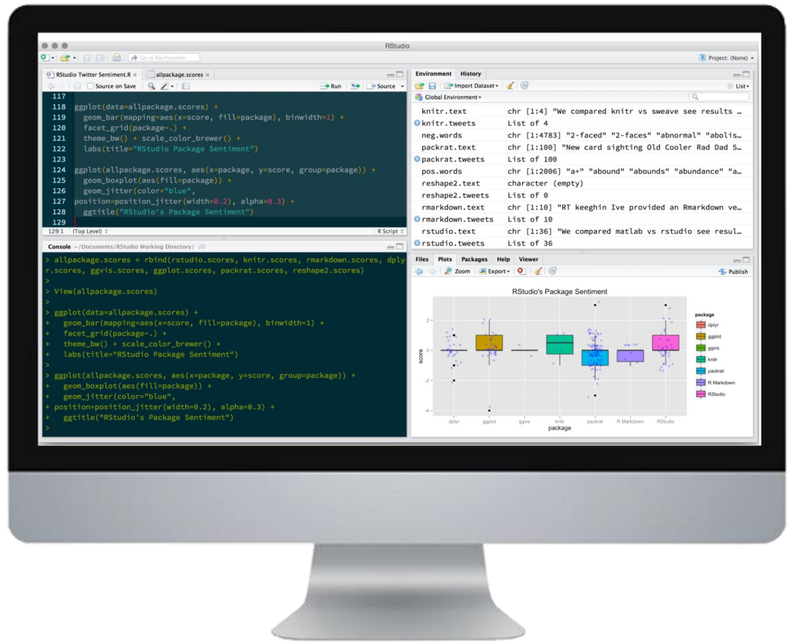
Masters of the Tidyverse



Art by Dan Mumford

RSTUDIO IDE

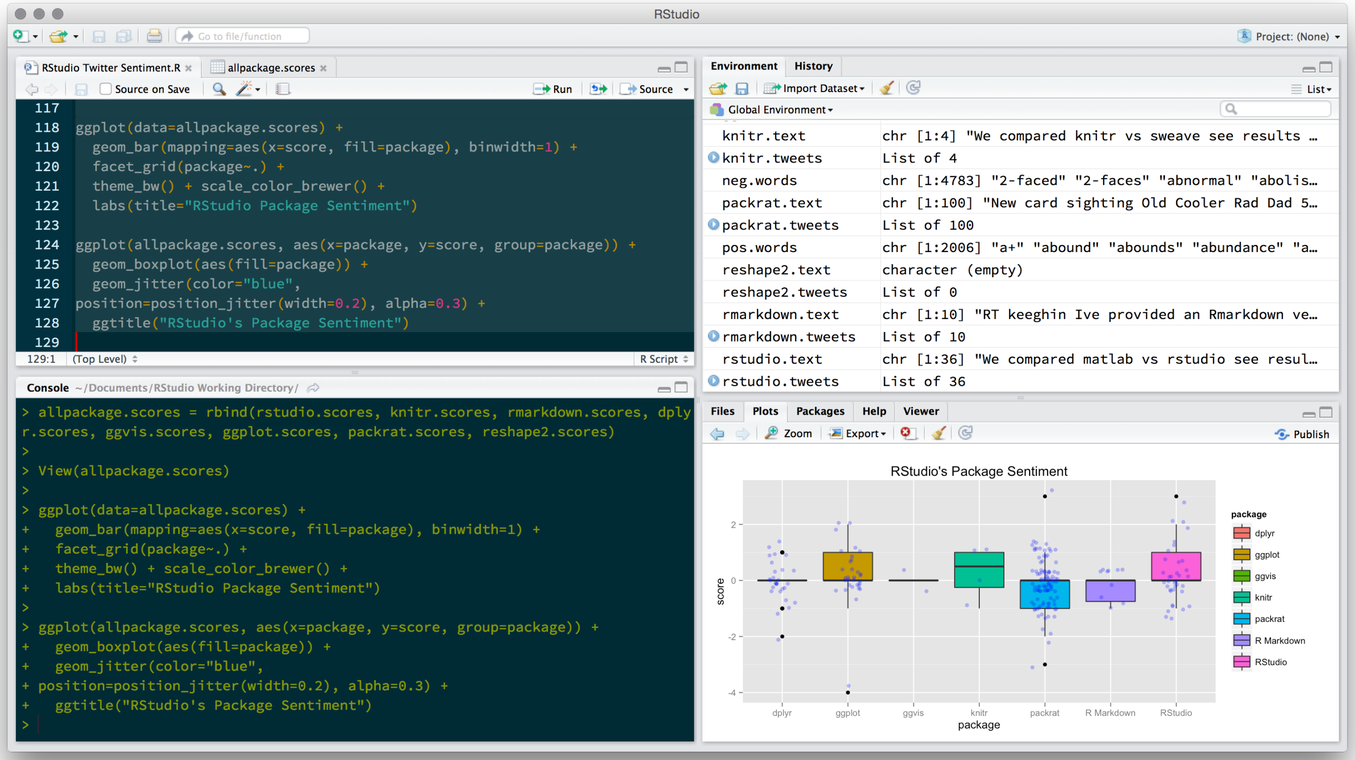
- Integrated Development Environment for R (i.e. like Eclipse or Visual Studio but for R)
- Runs on the Desktop (Windows, OSX, Linux) or over the Web as a server to enable shared resources and collaboration
- Released in 2012 and now the most popular front-end for R users with thousands of downloads per day
- Download the desktop IDE here: http://www.rstudio.com/products/RStudio/#Desk



RStudio®

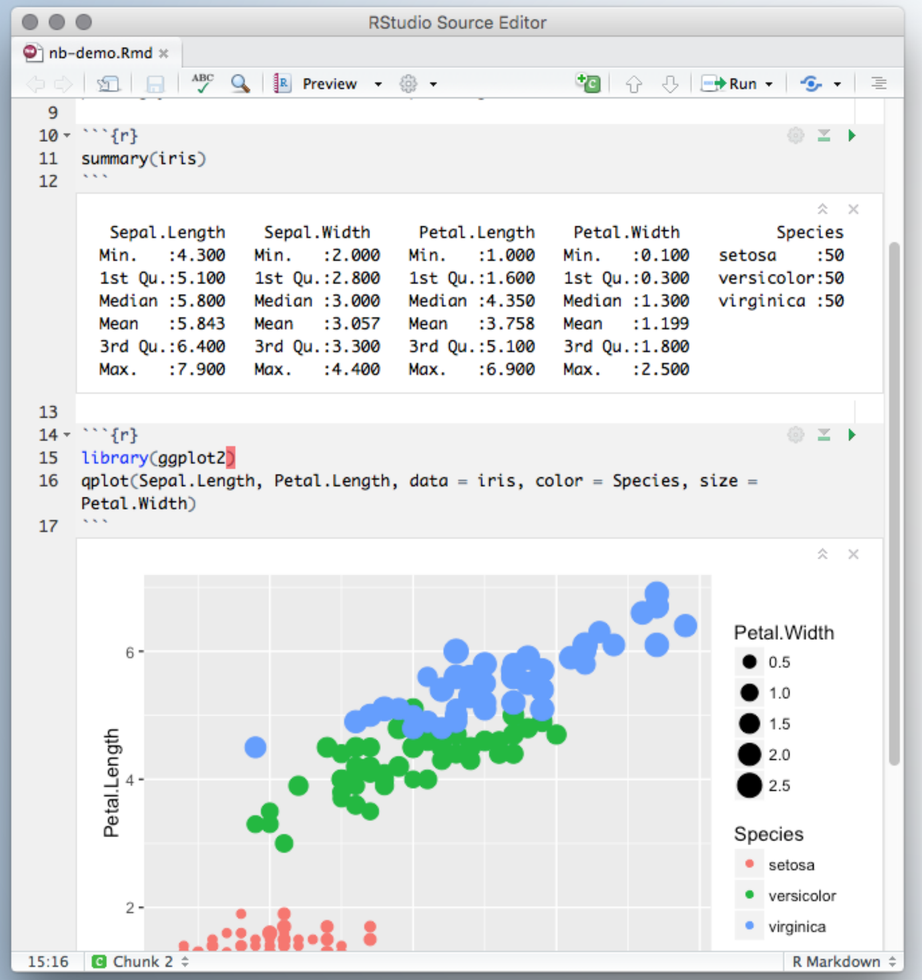
RSTUDIO IDE

- Data viewer -- support large datasets, filtering, searching, sorting
- Code completion
- Code diagnostics
- Customizable code snippets
- Improved tools for Rcpp
- Multiple cursors
- Tab re-ordering
- New themes
- Enhanced Vim mode
- Notebooks
- Add-ins
- Session Management
- Project Sharing
- Collaborative Editing



And much more!

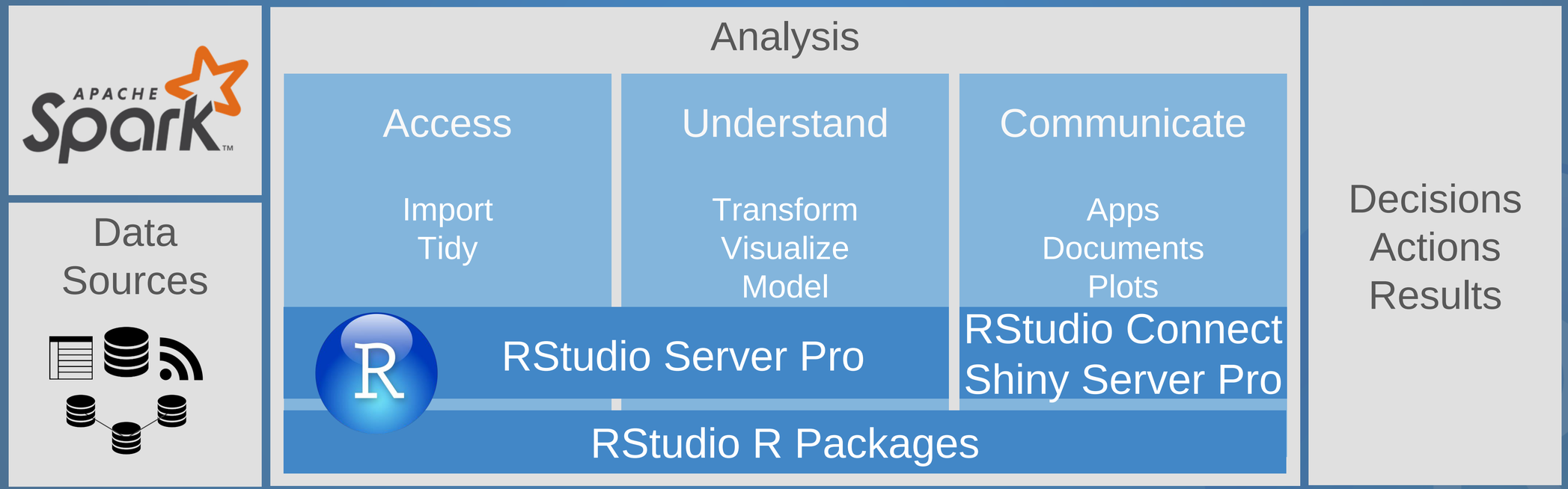
RStudio Server Pro



- Notebooks
- Bookdown
- R Markdown websites
- Data import
- Flex dashboards
- RStudio Add-ins
- Profvis
- Crosstalk

Our Toolchain

FAST, EASY, REPRODUCIBLE…



RSTUDIO PRODUCTS

RStudio and RStudio Server

- Open source IDE to improve the usability of R and the productivity of R users (like Eclipse or Visual Studio but for R). Runs on Windows, OSX, and Linux or over the Web as a server

RStudio Commercial Desktop and RStudio Server Pro

- Commercial IDEs to enable wider adoption of R in the enterprise

Shiny Server

- Open source web application server to share interactive data visualizations created with R

Shiny Server Pro

- Commercial web application server to enable enterprise scale deployment of interactive data visualizations, web applications and reproducible reports created with R

shinyapps.io

- Hosted service for deploying interactive reports and web applications created with R

RStudio Connect

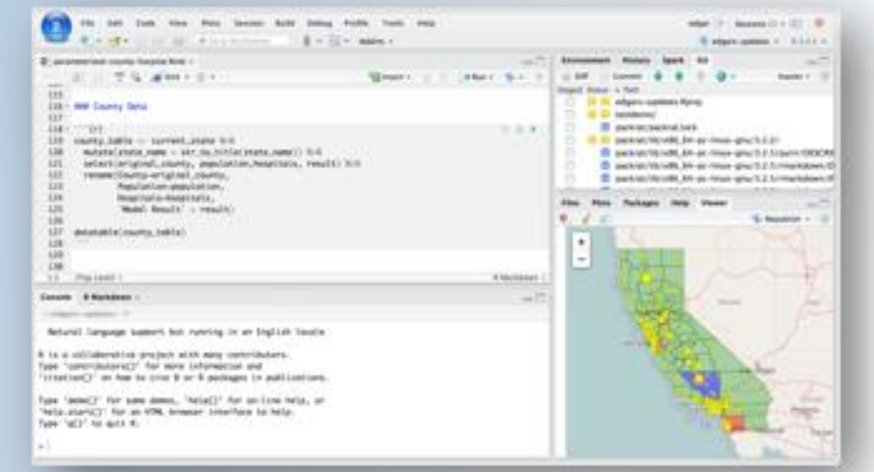
- A new publishing platform for the work your teams create in R. Share Shiny applications, R Markdown reports, dashboards, plots, and more in one convenient place. Push-button publishing from the RStudio IDE, scheduled execution of reports, and flexible security policies to bring the power of data science to your entire enterprise.

R Packages

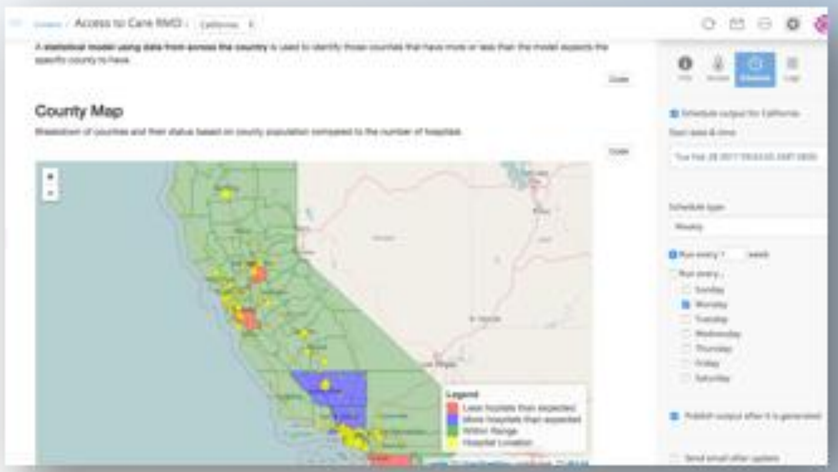
- Open source libraries for data manipulation (plyr and dplyr), data visualization (shiny, ggplot2, ggvis, htmlwidgets), and publishing (e.g., knitr, R Markdown...) and powerful productivity tools for R (packrat, devtools, roxygen2, testthat, tidyr, fastread)

RStudio®

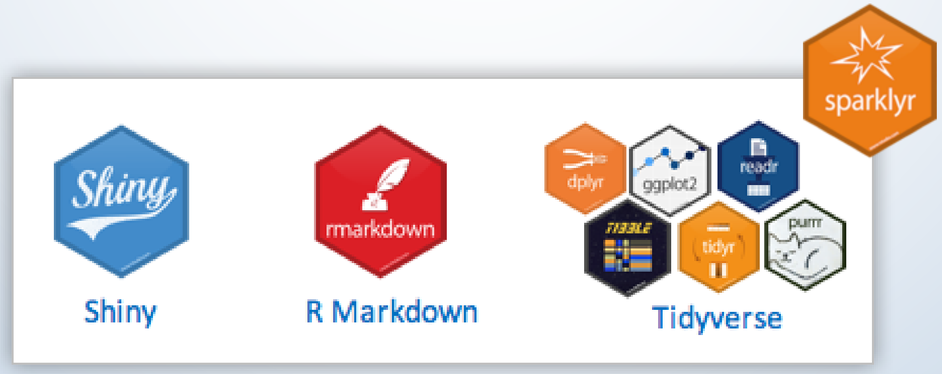
RStudio Products and Packages



RStudio Server Pro



RStudio Connect
Shiny Server Pro



R Packages

htmlwidgets & R Graphics

htmlwidgets

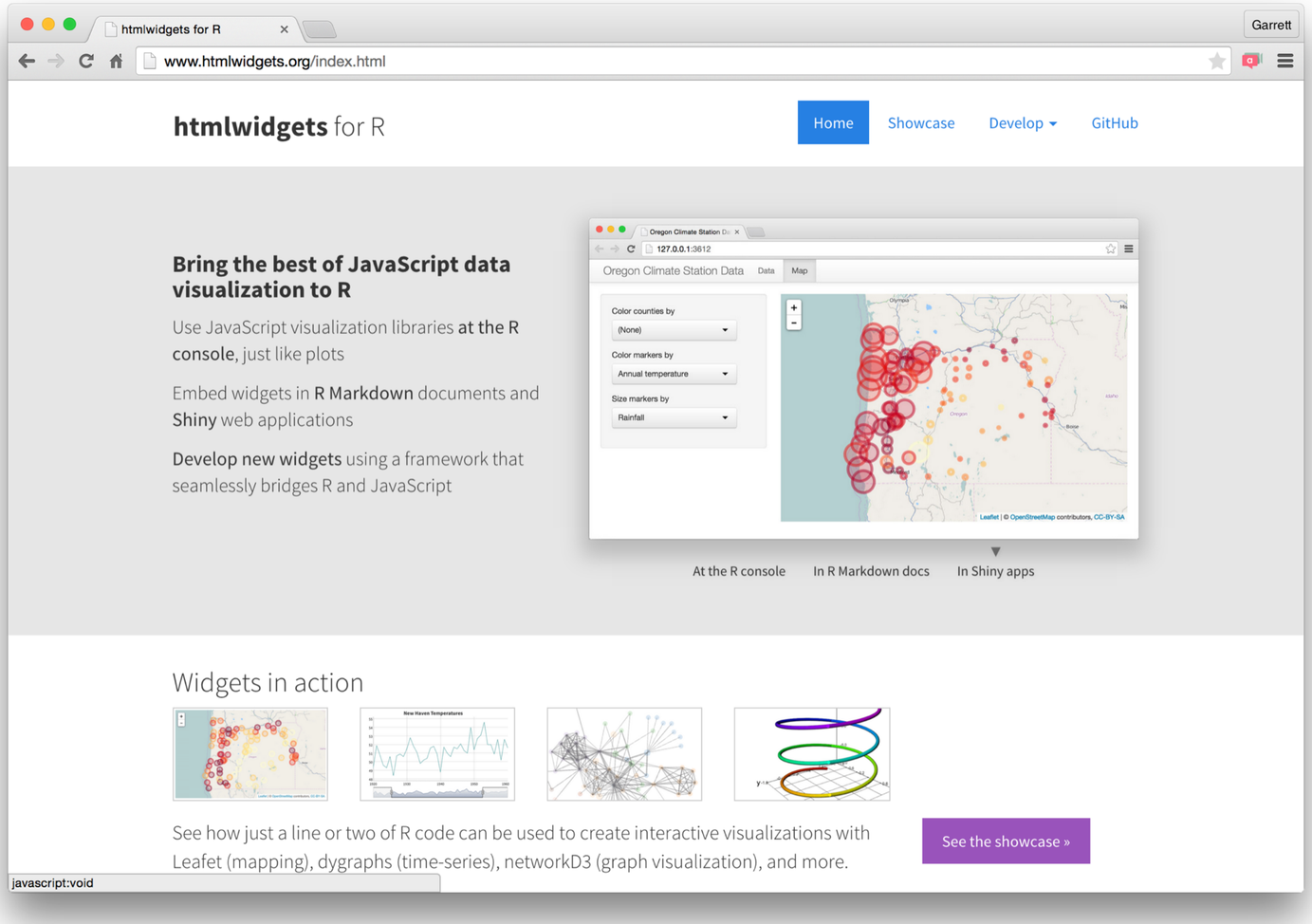
htmlwidgets for R:

- R bindings to JavaScript libraries
- Used to create interactive visualizations
- A line or two of R code is all it takes to produce an example

Use htmlwidgets in:

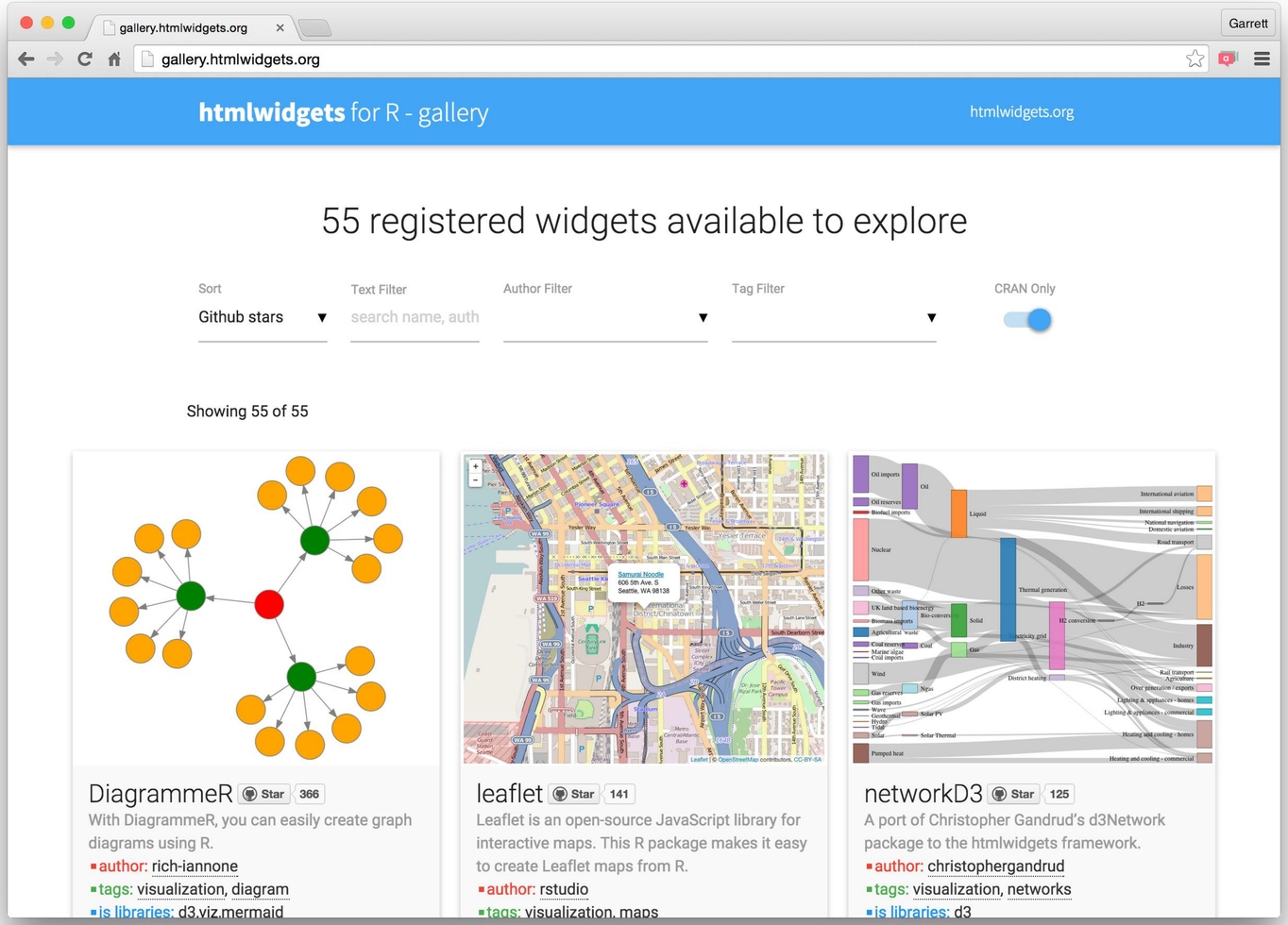
- RStudio viewer pane
- R Markdown files
- Shiny Apps

www.htmlwidgets.org



htmlwidgets gallery

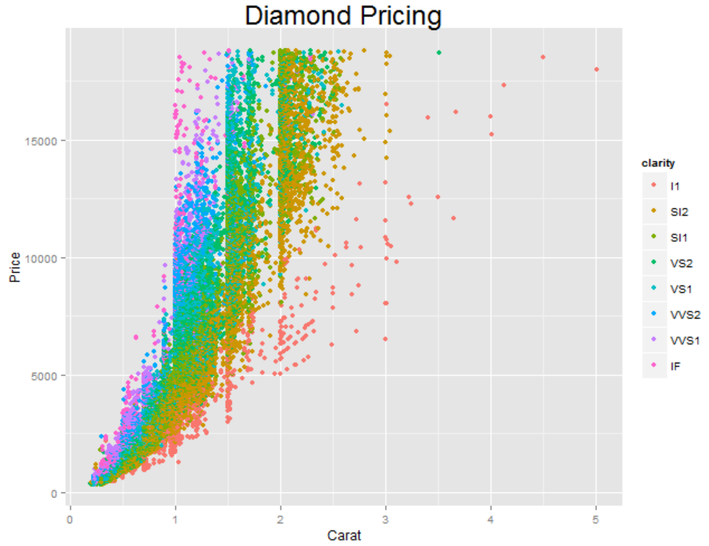
http://gallery.htmlwidgets.org/



R Graphics:

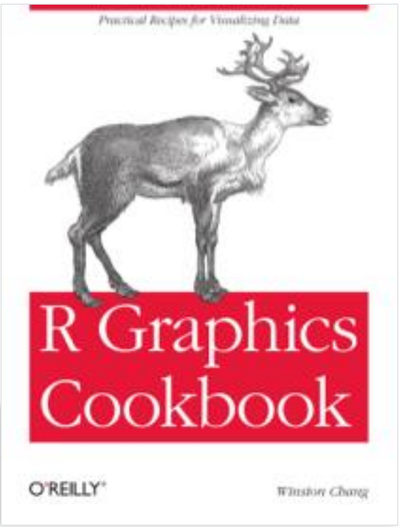
Four Main Graphical Systems in R:

- R's Base Graphics
- Grid Graphics System
- The lattice Package
- The ggplot2 Package – Created by Hadley Wickham

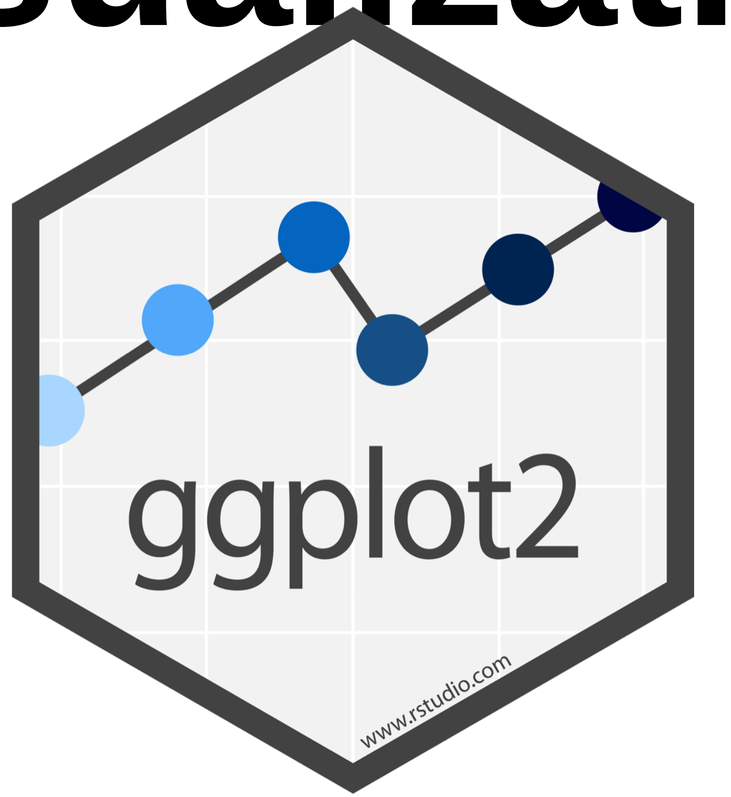


Why ggplot2?

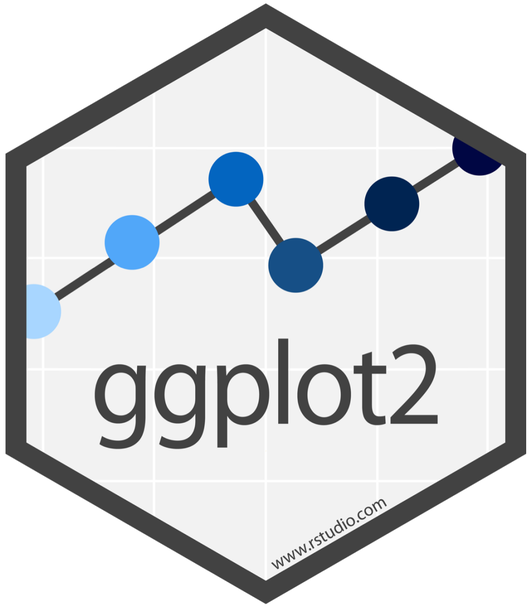
- Consistent underlying - Grammar of Graphics (Wilkinson, 2005)
- Very flexible
- Mature and complete graphics system
- Many users, active mailing list
- http://www.cookbook-r.com & http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html



Data Visualization with



ggplot2



A package that visualizes data.

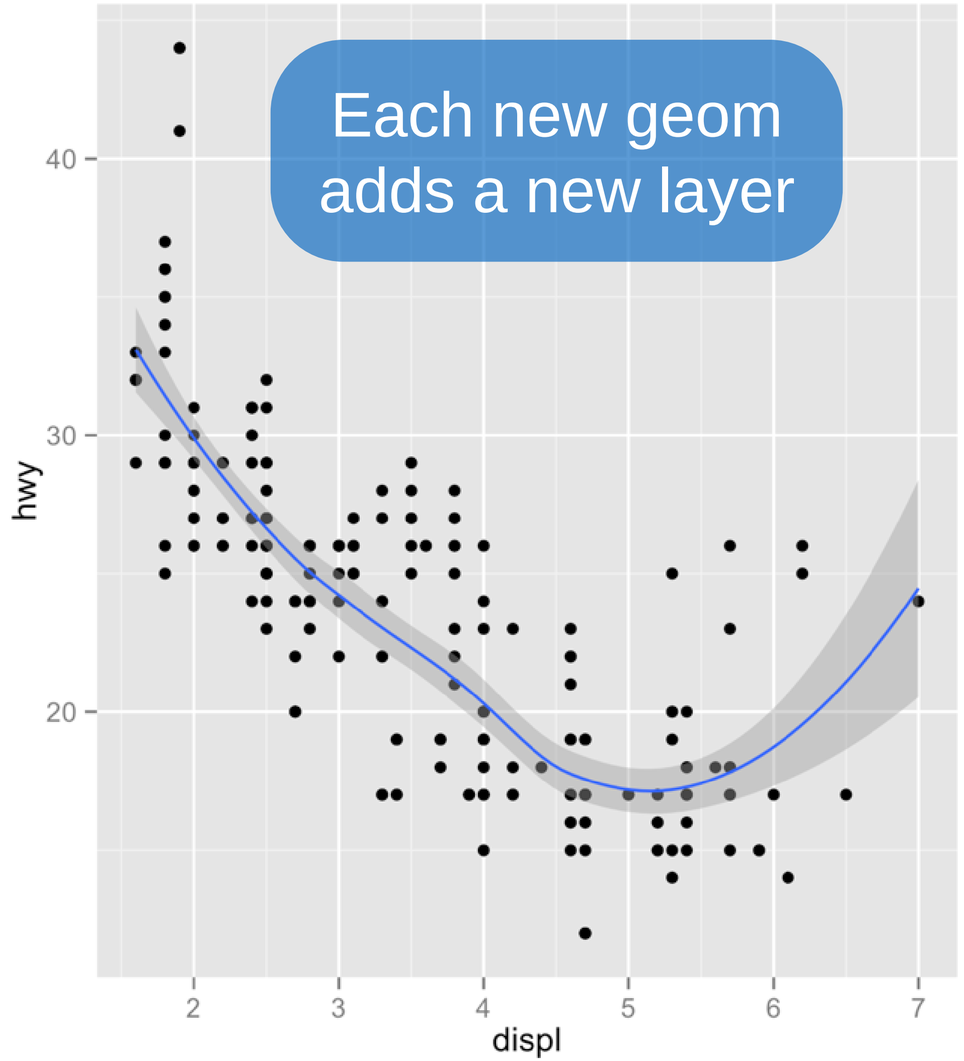
ggplot2 implements the *grammar of graphics*, a system for building visualizations that is built around cases and variables.

A ggplot2 template

Make any plot by filling in the parameters of this template

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```




```
ggplot(mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

A ggplot2 template

Make any plot by filling in the parameters of this template

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
                  stat = <STAT>) +
  <FACET_FUNCTION>
```

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut), stat = "count")
```

Reproducible Research via R Markdown

R Markdown

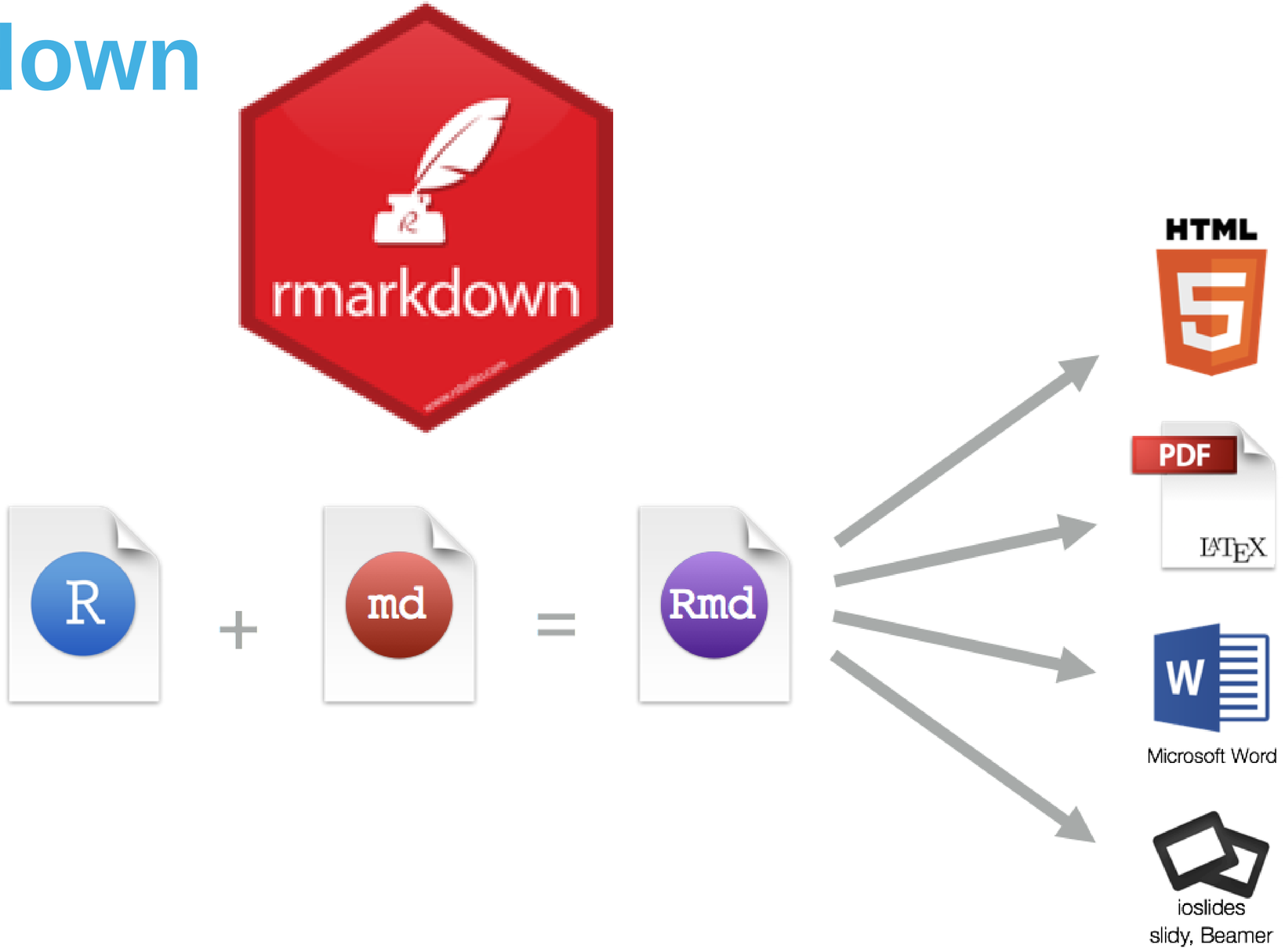
Can’t we do better than?

Ctrl + C (Copy)

Ctrl + V (Paste)

RStudio®

R Markdown



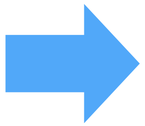
RStudio

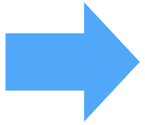
Parameters

A list of values that you can call in R code chunks

params list

elements and values

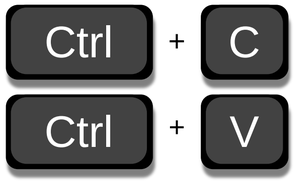




```
---
title: "Untitled"
output: html_document
params:
  filename: "data.csv"
  symbol: "GOOG"
---
```

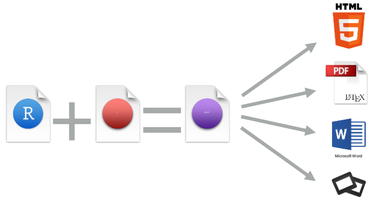
Access as params$filename and params$symbol

Recap: R Markdown

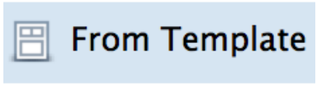


Reproducible

Automatic



Flexible



Reusable

params:

Parameterizable

rmarkdown::**render**

Render at the command line with YAML options

```
> render("doc.Rmd")
```

Render at the command line, override output format.

```
> render("doc.Rmd", "html_document")
```

Render at the command line to multiple formats.

```
> render("doc.Rmd", c("html_document", "pdf_document"))
```

rmarkdown::**render**

Render at the command line with YAML options

```
> render("doc.Rmd")
```

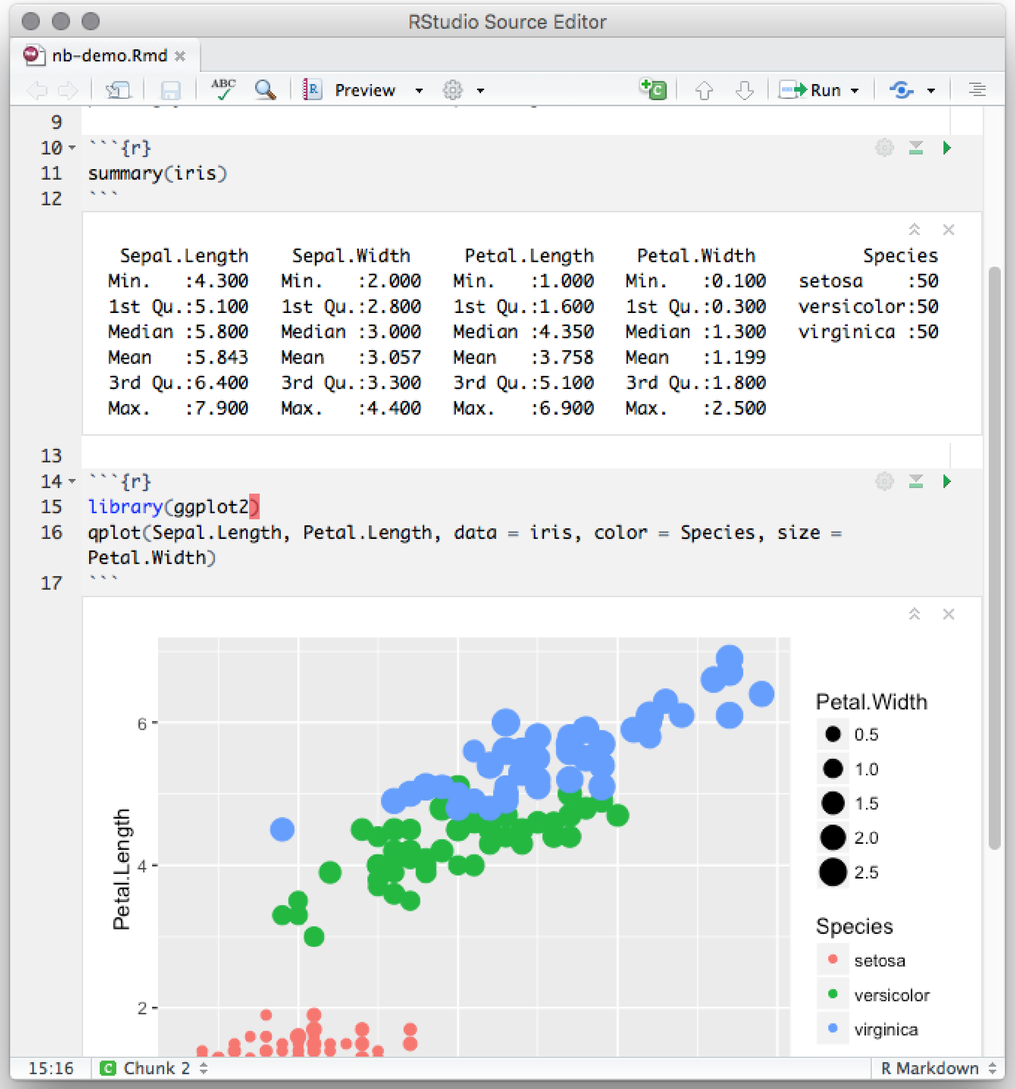
Render at the command line, set parameters.

```
> render("doc.Rmd", params = list(
      filename = "other_data.csv",
      symbol = "AAPL")
```

Other R Markdown Output Types

- Blogdown
- RMD Websites
- Bookdown
- Presentations

Notebooks



Combine in a single document:

- Narrative
- Code
- Output

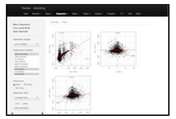
Then Render to HTML

SHINY

- Open source R package that facilitates the rapid creation and deployment of interactive web apps
- Targets data scientists/analysts with R expertise who want to share their analyses
- Uses spreadsheet-like programming semantics to make creating web applications with R very straightforward
- Allows analysts to put interactive analysis tools into the hands of decision makers immediately
- Building applications requires no HTML, CSS or JavaScript knowledge and eliminates the need for proprietary BI clients
- To learn more visit: http://shiny.rstudio.com/

RStudio®

What can a Shiny App do?



Make R analysis accessible to **non-programmers**

Highly **customizable**, highly **shareable** HTML front end

Read and write to **databases**

Monitor **streaming data**

Require and use **authentication**

Ideal for Exploratory Data Analysis

Ideal Data Portal / Results Explorer / Simulation API / Dashboard

Share your app

RStudio®

Shiny Server Open Source And Pro

Shiny Server

- A web application framework for R (i.e. the bridge enabling the power of R for the web)
- Reactive - Uses spreadsheet like programming semantics to make creating web applications with R
- Start and Stop a Shiny Process
- Translate non-Websocket traffic into Websockets
- Map URL's to particular applications

Shiny Server Pro

Everything that's available in Shiny Server plus enterprise features:

- Administrative Tools - View and manage users, applications & resources
- Authentication - Secure access to applications
- Scalability - Enable multiple R instances per application
- No Centralized Storage - Only "Stateful" information for active sessions; all authentication information is stored in an encrypted cookie

RStudio®

What is the difference between Shiny and Shiny Server?

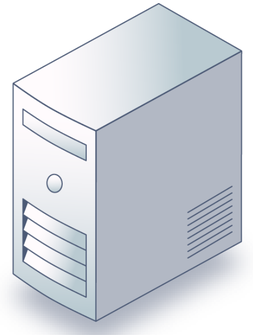
Shiny

Free and open source R Package

Makes it incredibly easy to
build interactive web applications with R

Automatic reactive binding between inputs and outputs and extensive pre-built widgets make it possible to build beautiful, responsive, and powerful applications.

Shiny Server



Software you install on your server

Enable users to
host and managing Shiny applications

Scale a Shiny application to support many users
Protect and secure your applications
Manage the user experience

RSTUDIO CONNECT

A new publishing platform for the work your teams create in R.

Share Shiny applications, R Markdown reports, dashboards, plots, and more in one convenient place.

RStudio®

RStudio Connect

A publishing platform for all the work your team creates in R

Content Creators

Publish

Push button

RStudio Connect

Create variants
Schedule jobs
Distribute reports

Self-service

Content Consumers

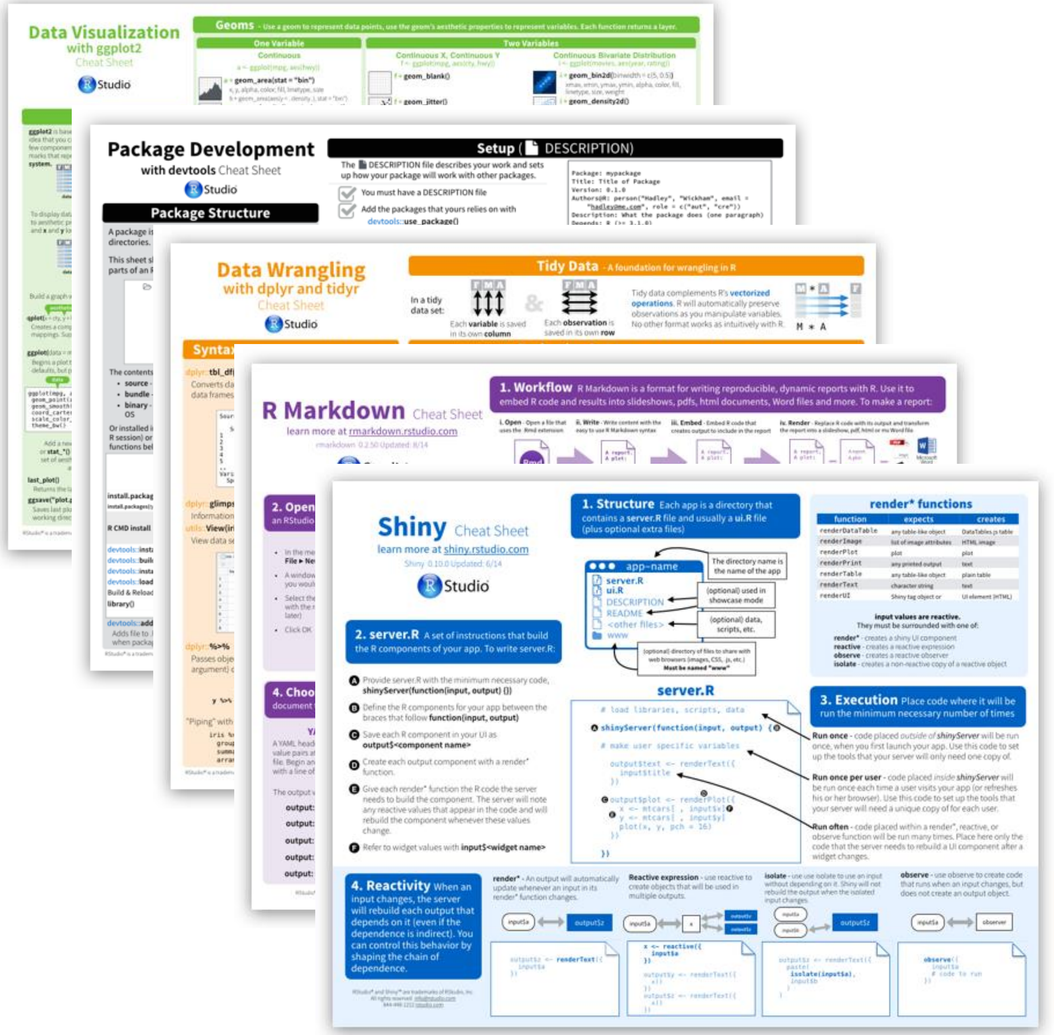
Make decisions
Take actions
See results

Connect is Built for Data Scientists and IT

	Features	Solutions
Analyst	Push Button Deployment	Rapid Iteration
	Web UI for Managing Content	Self Service
	Email Integration	Content Sharing
	Job Scheduler	Automatic Updates
IT	User Roles	Access Control
	Authentication: AD / PAM / LDAP	Secure & Integrated
	Metrics and Logging	Scalable & Reliable

Other Goodies

- Cheatsheets
- Webinars & Recordings



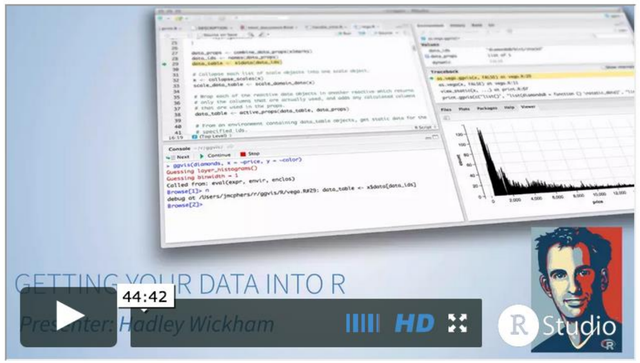
Latest Webinars

Getting your data into R

You can't use R for data analysis unless you can get your data into R. Getting your data into R can be a major hassle, so in the last few months Hadley Wickham has been working hard to make it easier.

In this webinar Hadley will discuss the places you most often find data (databases, excel, text files, other statistical packages, web apis, and web pages) and the packages (DBI, xml2, jsonlite, haven, readr, exel) that make it easy to get your data into R.

GitHub Webinar Repository
Materials for this specific webinar



How to start with Shiny – Part 3

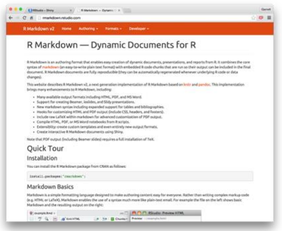
In this talk (Part 3 of 3), Garrett Grolemund will show you how to customize the appearance of your app. You will learn how to arrange the components of your app into an attractive layout, as well as how to change the appearance of text, images, and other HTML elements in your app.

GitHub Webinar Repository
Materials for this specific webinar
Alternate Link: bit.ly/shiny-quickstart-3

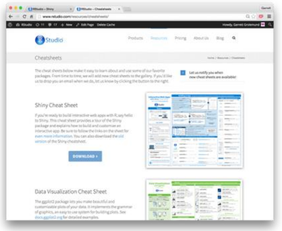
RStudio

Useful Websites

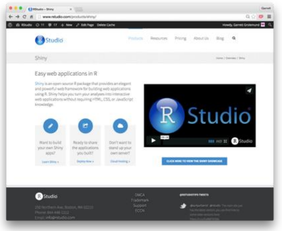
The Shiny development center:
shiny.rstudio.com



The R Markdown development center:
rmarkdown.rstudio.com



Shiny and R Markdown cheat sheets:
www.rstudio.com/resources/cheatsheets



RStudio products:
www.rstudio.com/products/shiny

RStudio®

Open Source & Free

Desktop: http://www.rstudio.com/products/rstudio/download/

RStudio Server: http://www.rstudio.com/products/rstudio/download-server/

Shiny Server: http://www.rstudio.com/products/shiny/download-server/

shinyapps.io beta: https://www.shinyapps.io/admin/#/signup

45 Day Evaluation of Pro Products

RStudio Server Pro: http://www.rstudio.com/products/rstudio-server-pro/evaluation/

Shiny Server Pro: http://www.rstudio.com/products/shiny-server-pro/evaluation/

PLEASE STAY IN TOUCH

Blog - http://blog.rstudio.org/

Twitter - @rstudio #rstats http://twitter.com/rstudio/

GitHub - https://github.com/rstudio/

LinkedIn - https://linkedin.com/company/rstudio-inc

Facebook - https://www.facebook.com/pages/RStudio-inc

Google+ - https://plus.google.com/110704473211154995841/posts

RStudio®

Community Activities

R Consortium

Supports the R foundation

RStudio

Community

rstudio::conf

Annual conference

2016 - Palo Alto
2017 - Orlando Florida
2018 - San Diego

R Views

Community blog