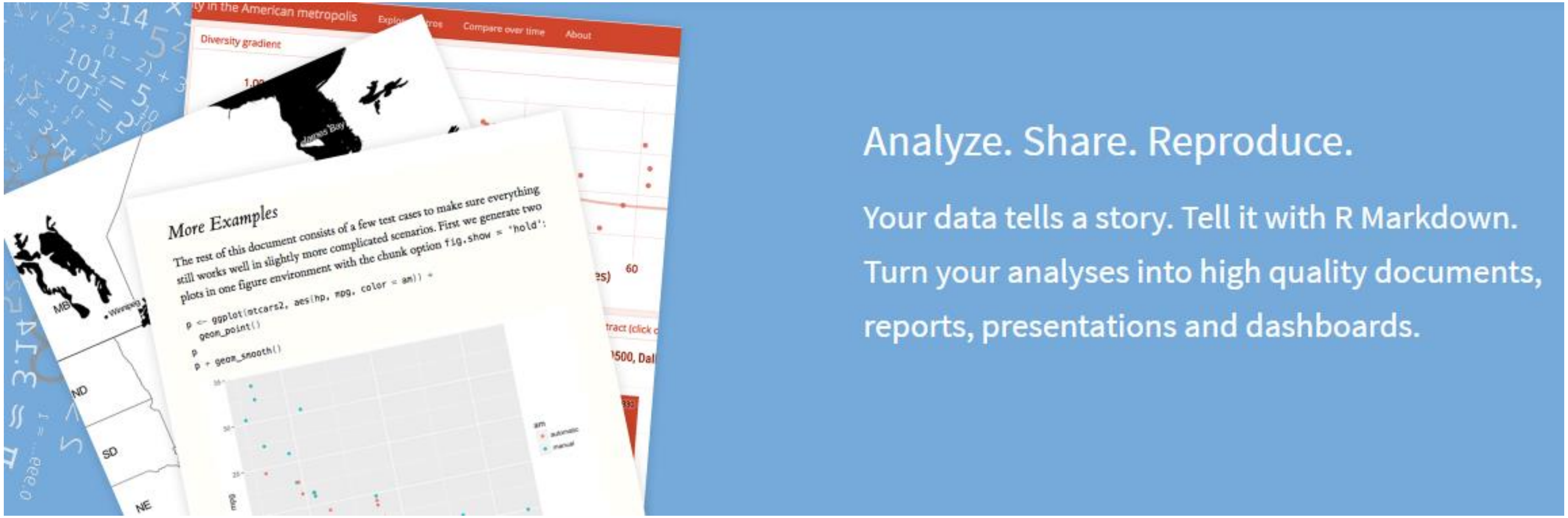
```r
library(dplyr)

rladies_global %>%
  filter(city == 'Lisbon')
```

Present your work with Markdown and Shiny

Hello!

My name is Adrianna Napiórkowska. I come from Poland. I am studying Advanced Analytics at Nova Information Management School and working at REBIS Consulting. I am also a member of Nova Analytics Group.

Markdown



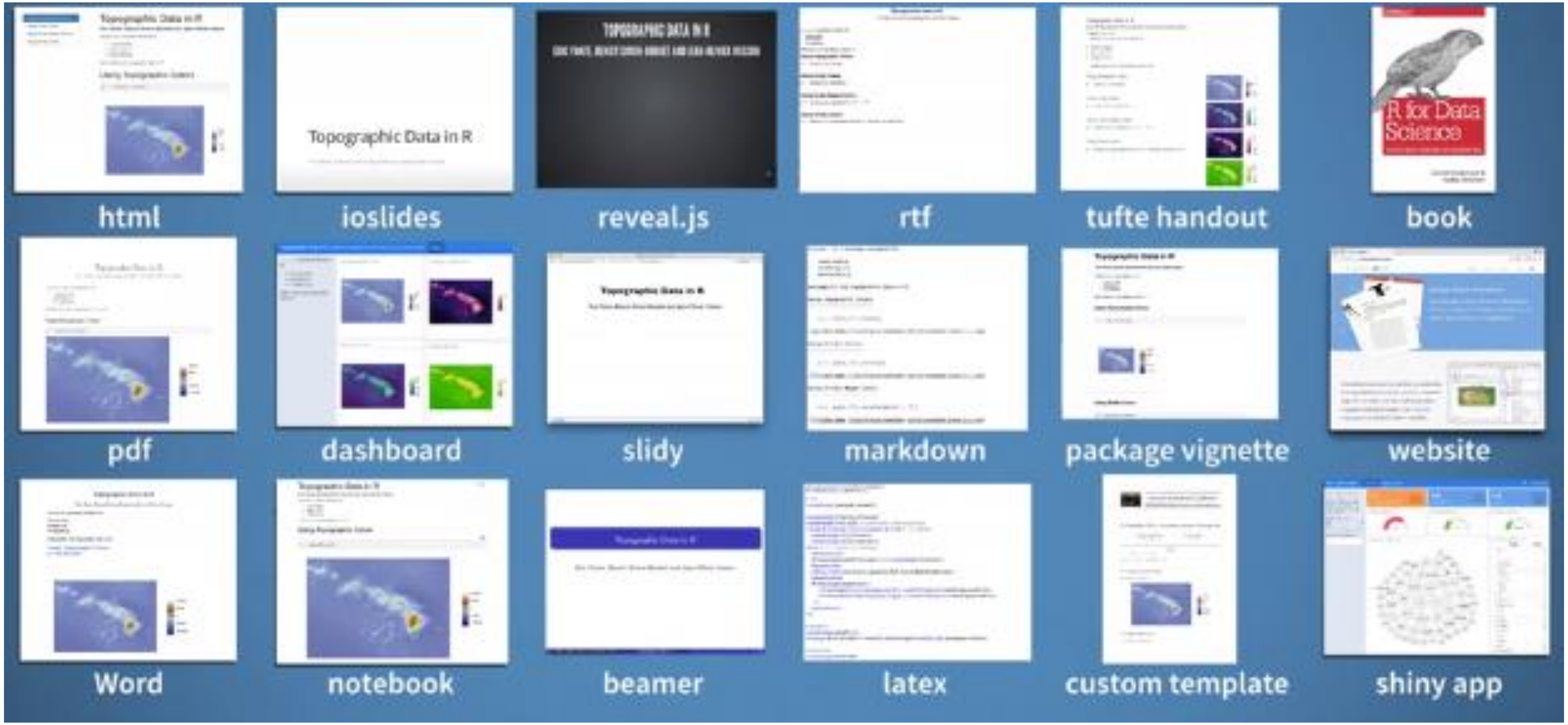
Analyze. Share. Reproduce.

Your data tells a story. Tell it with R Markdown. Turn your analyses into high quality documents, reports, presentations and dashboards.

R Markdown
Why do we need it?

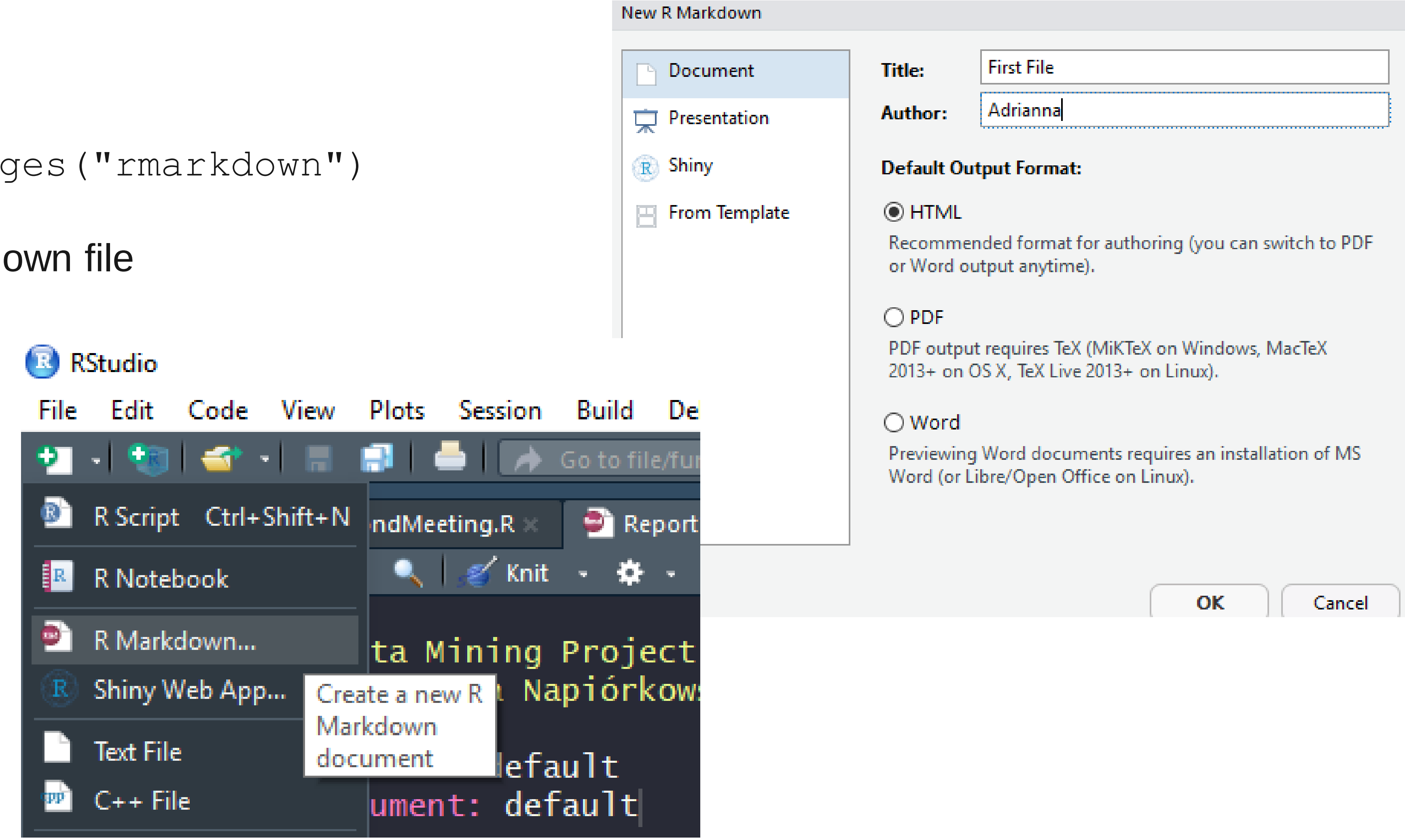
- Standard way of delivering reports – *copy+paste*
- Imagine a situation when one day before deadline you get the information that you were using wrong version of the data
- Now you have to repeat everything and once again generate all summaries and plots and *copy+paste* it to your report.
- If you were using Markdown, you just need to replace data source and knit new report.

What you can do with
Markdown



How to start

- `install.packages("rmarkdown")`
- Create a R Markdown file



Title: First File

Author: Adrianna

Default Output Format:

● HTML

Recommended format for authoring (you can switch to PDF or Word output anytime).

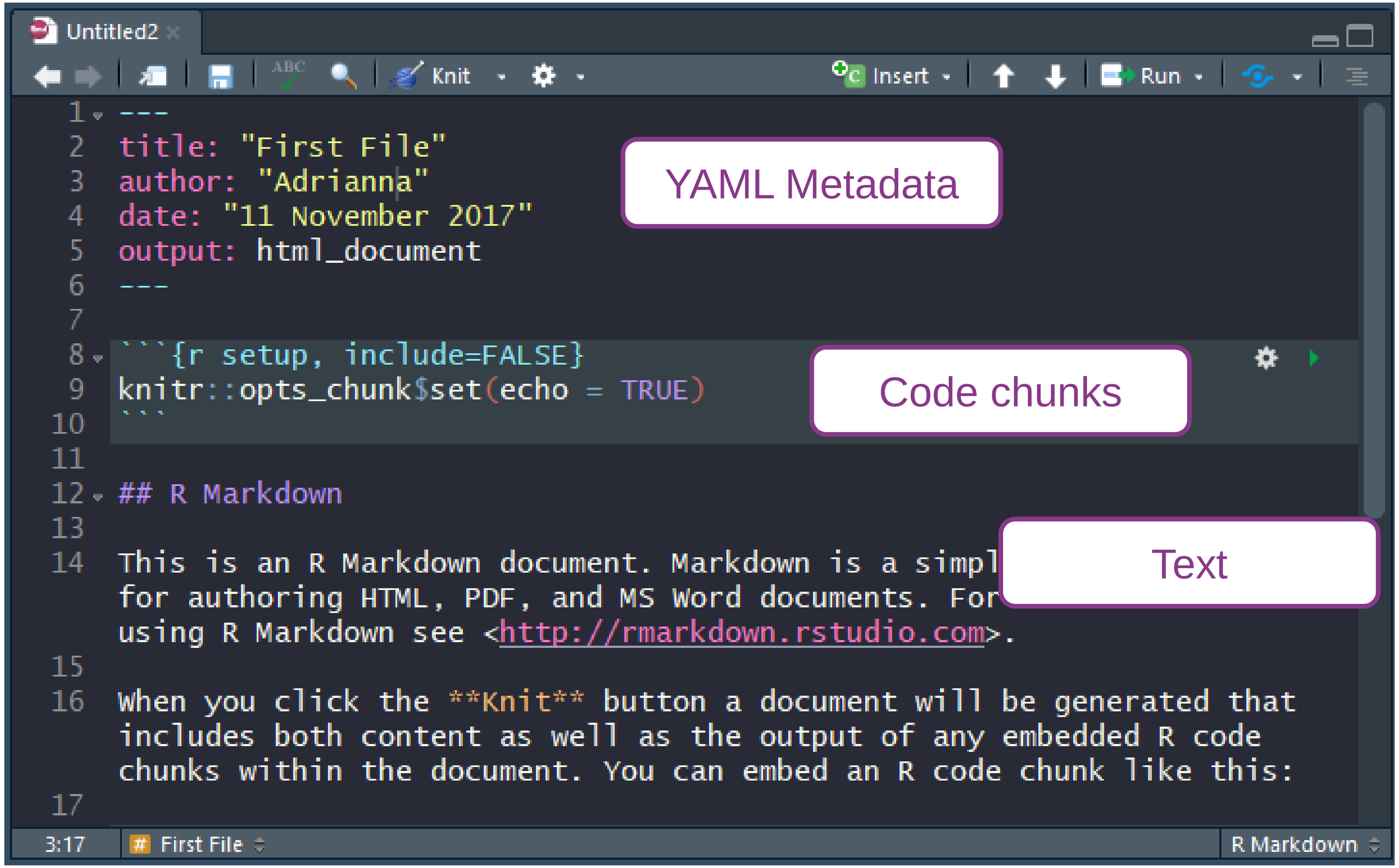
○ PDF

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

File structure



~~~
---
title: "First File"
author: "Adrianna"
date: "11 November 2017"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simpl
for authoring HTML, PDF, and MS Word documents. For
using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

~~~

YAML Metadata

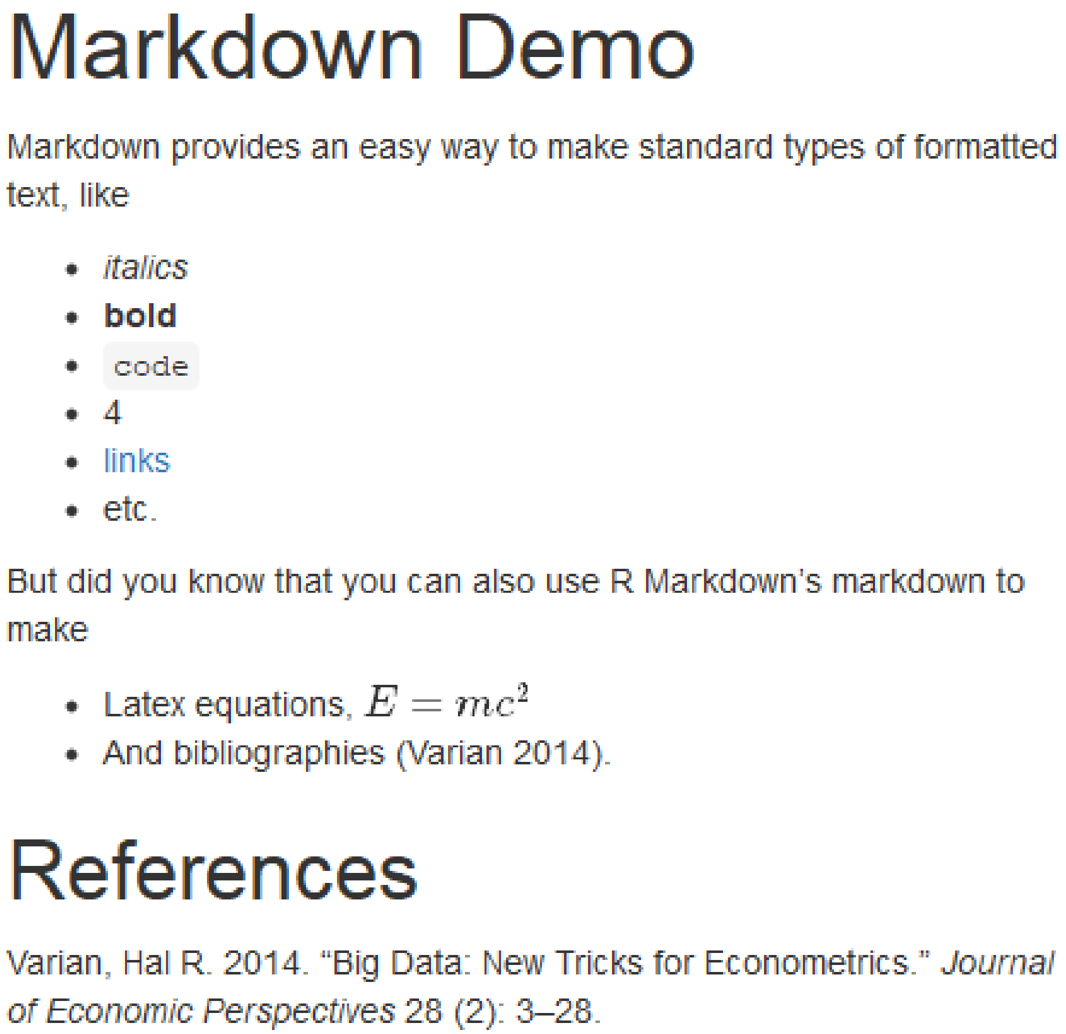
Code chunks

Text

To include R code in a text use: `r `

Markdown demo

```
---
title: "Markdown Demo"
output: html_document
bibliography: Biblio.bib
---

Markdown provides an easy way to make standard
types of formatted text, like

- *italics*
- **bold**
-  `code`
-  `r 2+2`
- [links](rmarkdown.rstudio.com)
- etc.

But did you know that you can also use R Markdown's
markdown to make

- Latex equations, $E = mc^{2}$
- And bibliographies [@Varian2014].

# References
```



Markdown Demo

Markdown provides an easy way to make standard types of formatted text, like

- *italics*
- **bold**
- `code`
- 4
- links
- etc.

But did you know that you can also use R Markdown's markdown to make

- Latex equations, $E = mc^2$
- And bibliographies (Varian 2014).

References

Varian, Hal R. 2014. "Big Data: New Tricks for Econometrics." *Journal of Economic Perspectives* 28 (2): 3–28.

Not only R

- In R Markdown documents you can use other programming languages:

Python

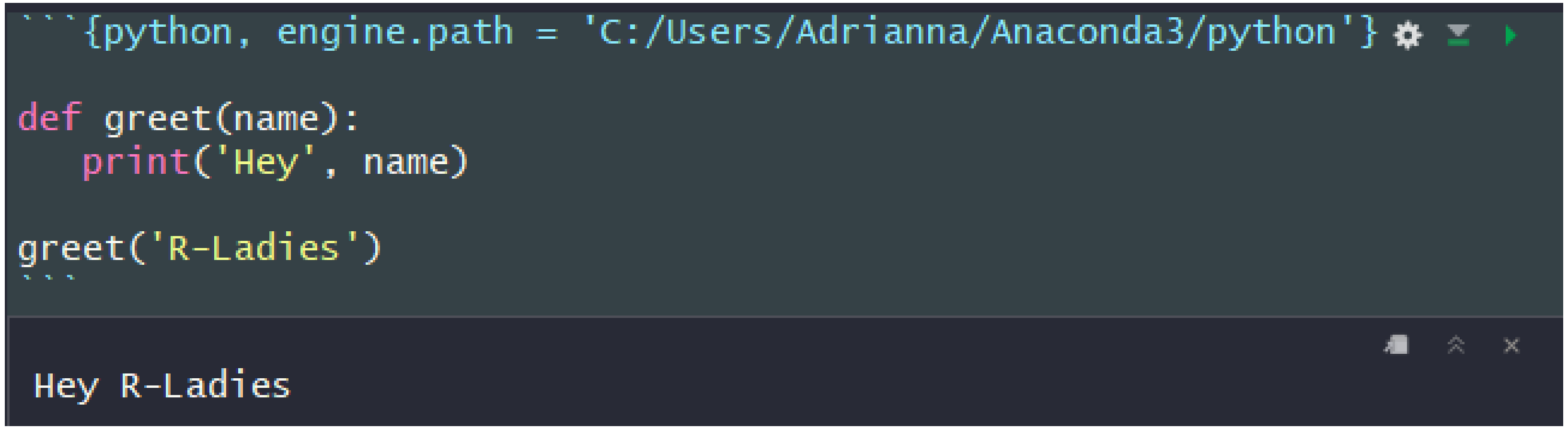
SQL

Bash

C++

JavaScript

CSS

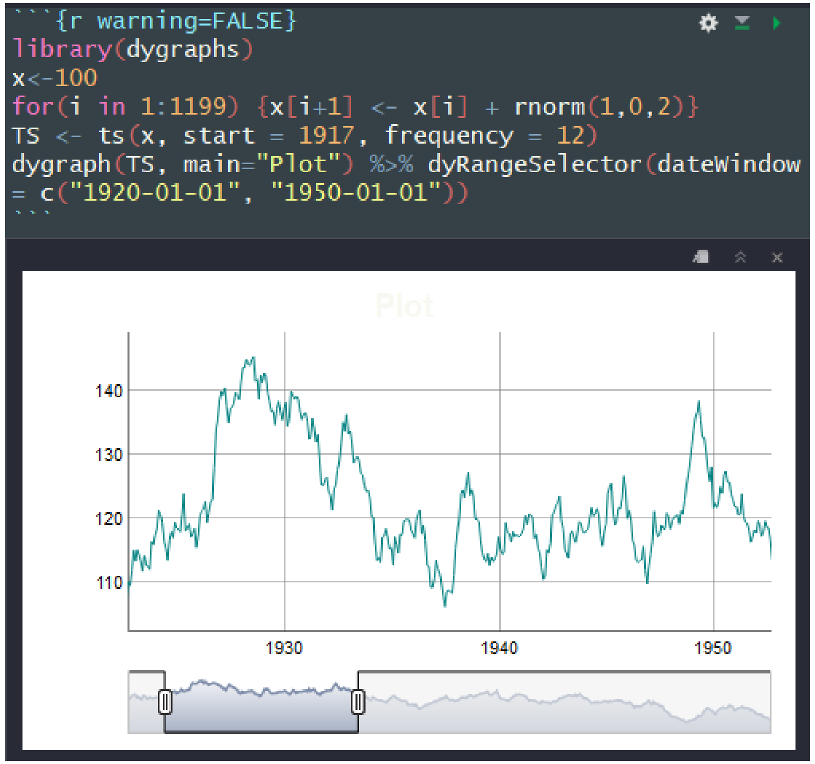


~~~
```{python, engine.path = 'C:/Users/Adrianna/Anaconda3/python'}

def greet(name):
   print('Hey', name)

greet('R-Ladies')
```
~~~

```
Hey R-Ladies
```

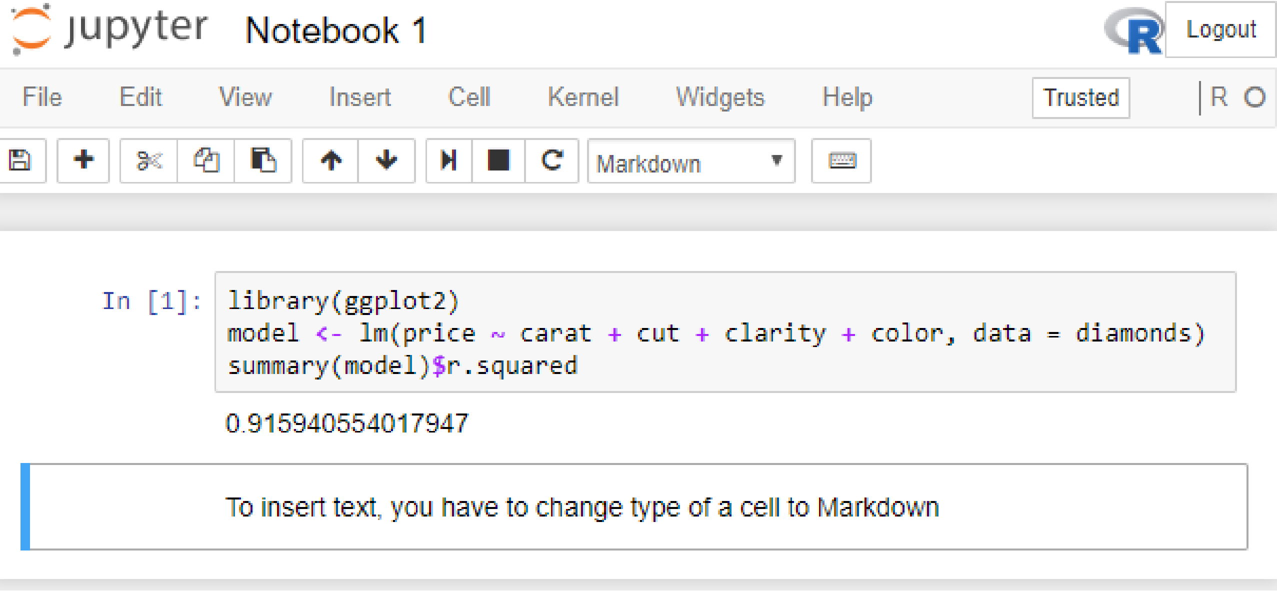
Work in a notebook

- You can use R Markdown files as a notebook:



~~~
```{r warning=FALSE}
library(dygraphs)
x<-100
for(i in 1:1199) {x[i+1] <- x[i] + rnorm(1,0,2)}
TS <- ts(x, start = 1917, frequency = 12)
dygraph(TS, main="Plot") %>% dyRangeSelector(dateWindow
= c("1920-01-01", "1950-01-01"))
```
~~~

- Use Markdown to describe your work in jupyter Notebook

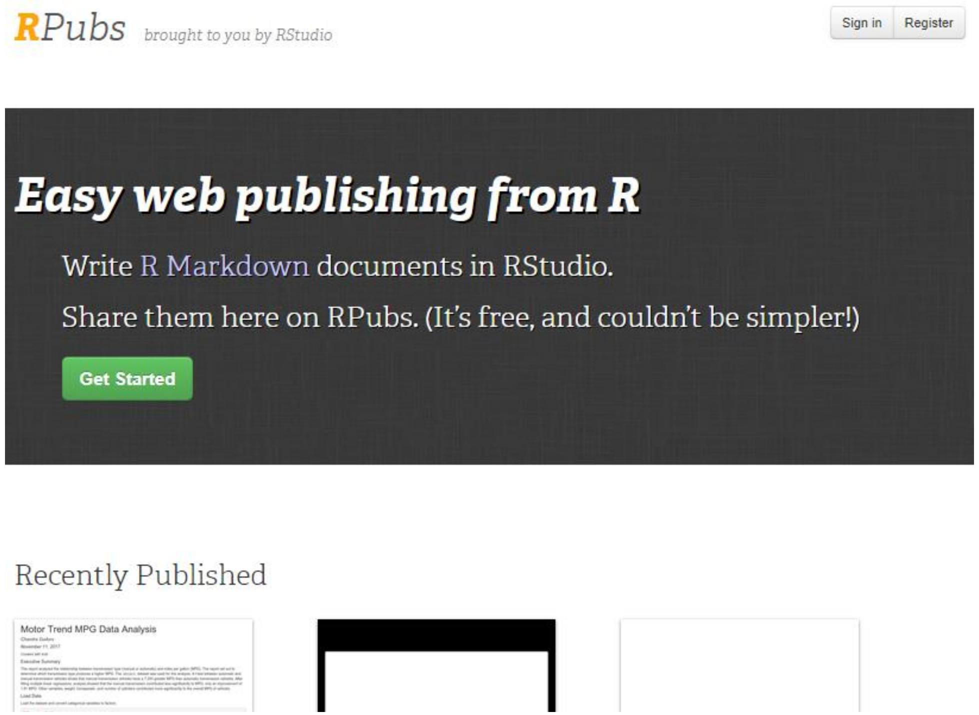


```
In [1]: library(ggplot2)
        model <- lm(price ~ carat + cut + clarity + color, data = diamonds)
        summary(model)$r.squared

        0.915940554017947
```

To insert text, you have to change type of a cell to Markdown

How to share your work

Rpubs.com
For non-interactive documents



RPubs brought to you by RStudio

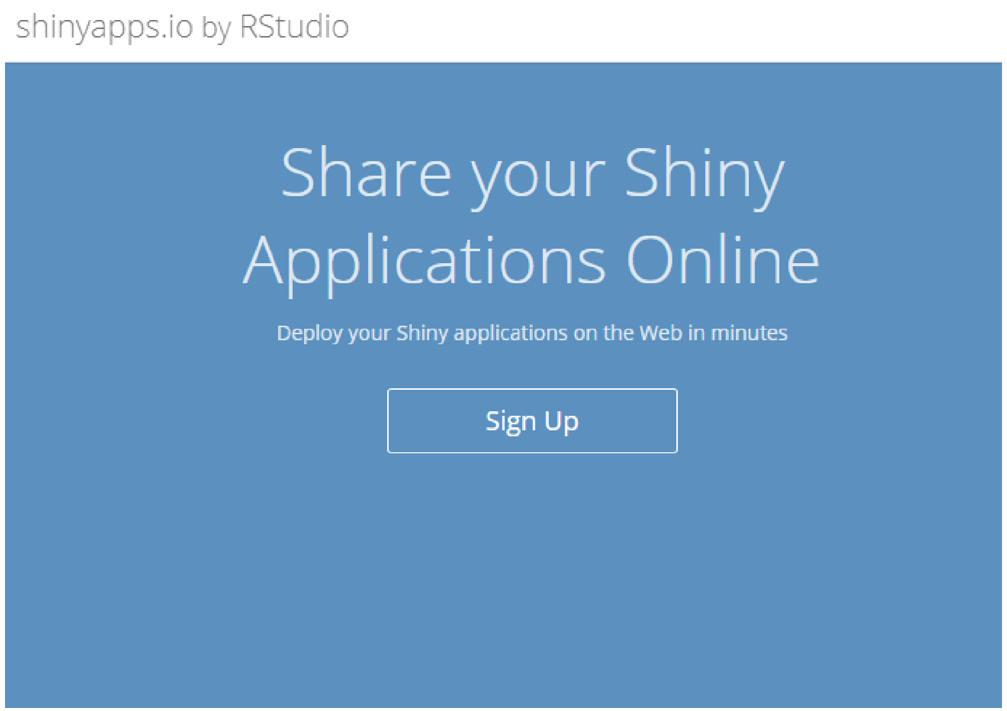
Easy web publishing from R

Write R Markdown documents in RStudio.

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

Recently Published

ShinyApps.io
For interactive documents

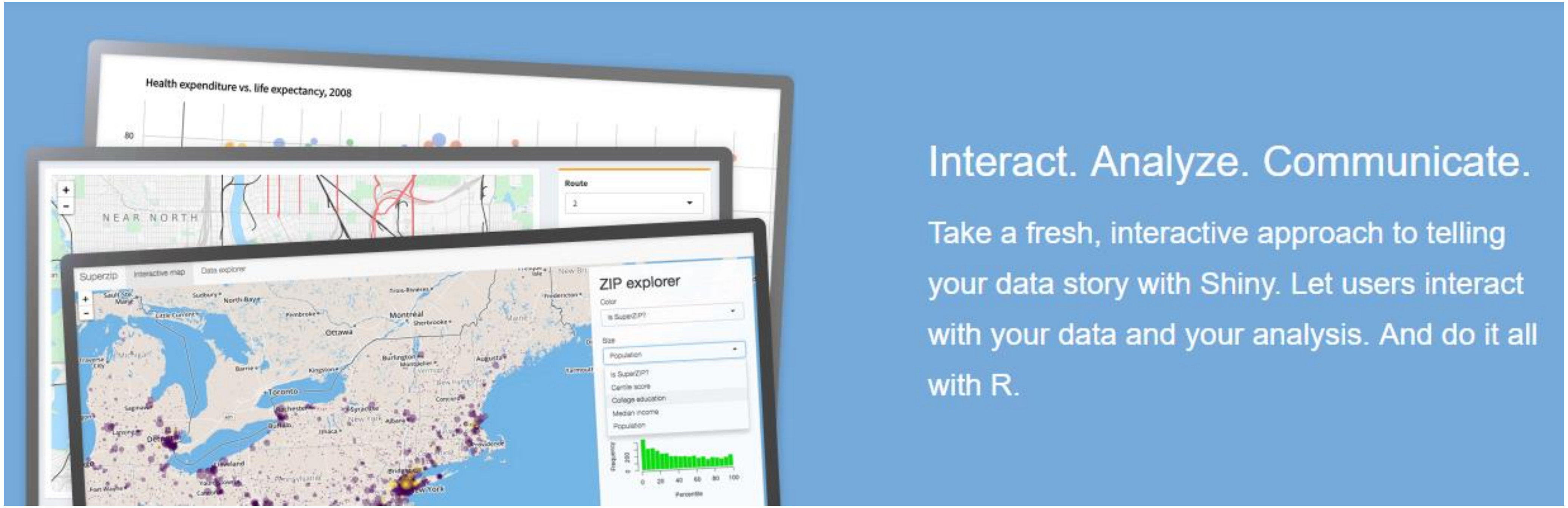


shinyapps.io by RStudio

Share your Shiny Applications Online

Deploy your Shiny applications on the Web in minutes

Shiny



Interact. Analyze. Communicate.

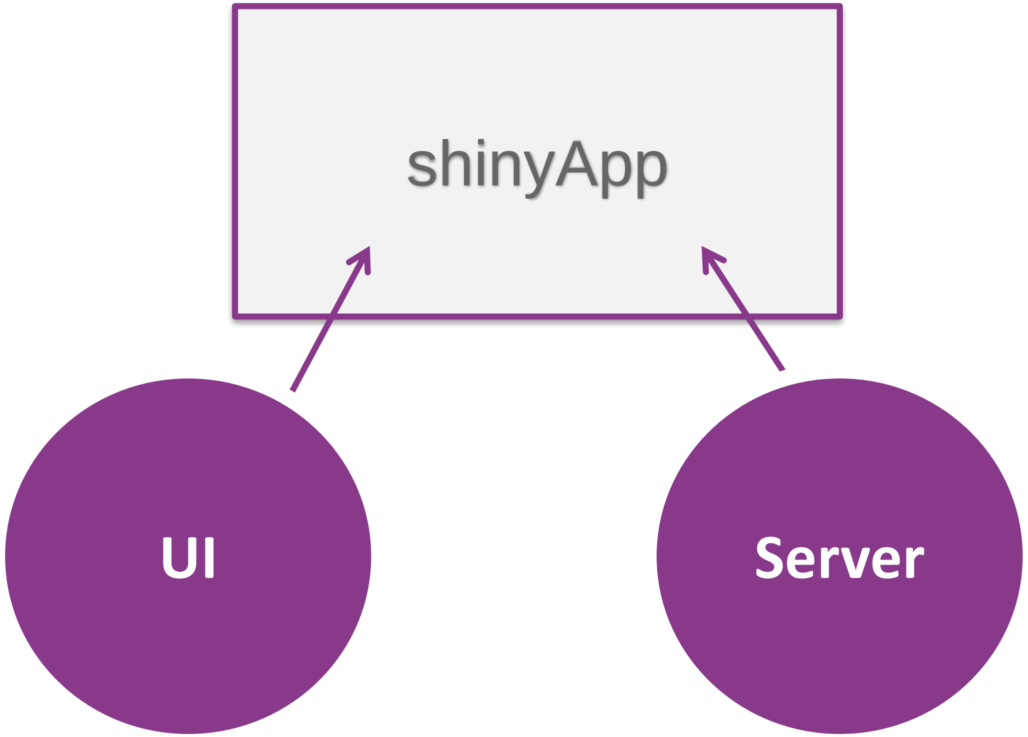
Take a fresh, interactive approach to telling your data story with Shiny. Let users interact with your data and your analysis. And do it all with R.

How to build your first Shiny App

```
install.packages("shiny")
```

```
library(shiny)

ui <- ...
server <- ...
shinyApp(ui = ui, server = server)
```

```
runApp("my_app")
```



How to work with your application

Run it

- `runApp` - run from local files
- `runGitHub` - run from files hosted on www.GitHub.com
- `runGist` - run from files saved as a gist (gist.github.com)
- `runURL` - run from files saved at any URL

Share it

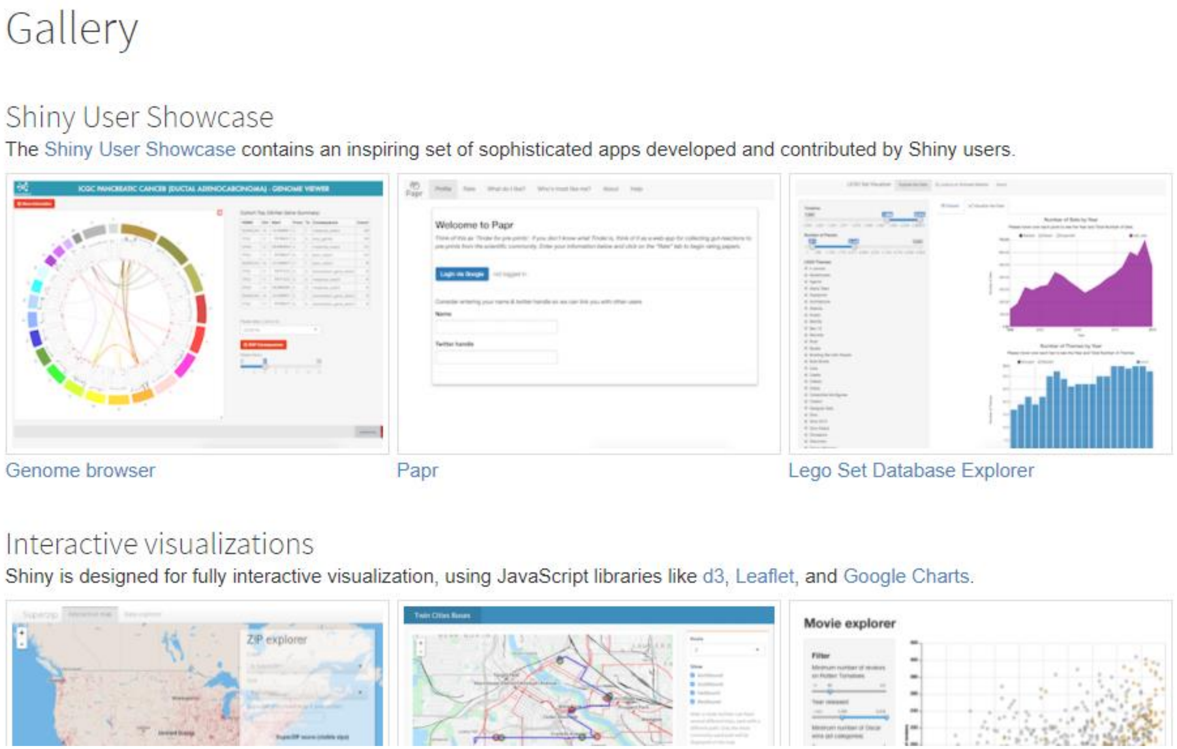
- ShinyApps.io
- Shiny Server
- Shiny Server Pro

Shiny – first steps

- Start with existing examples:

```
runExample("01_hello")
runExample("02_text")
runExample("03_reactivity")
……
```

- Visit Shiny Gallery for inspiration:



Gallery

Shiny User Showcase

The Shiny User Showcase contains an inspiring set of sophisticated apps developed and contributed by Shiny users.

Genome browser

Papr

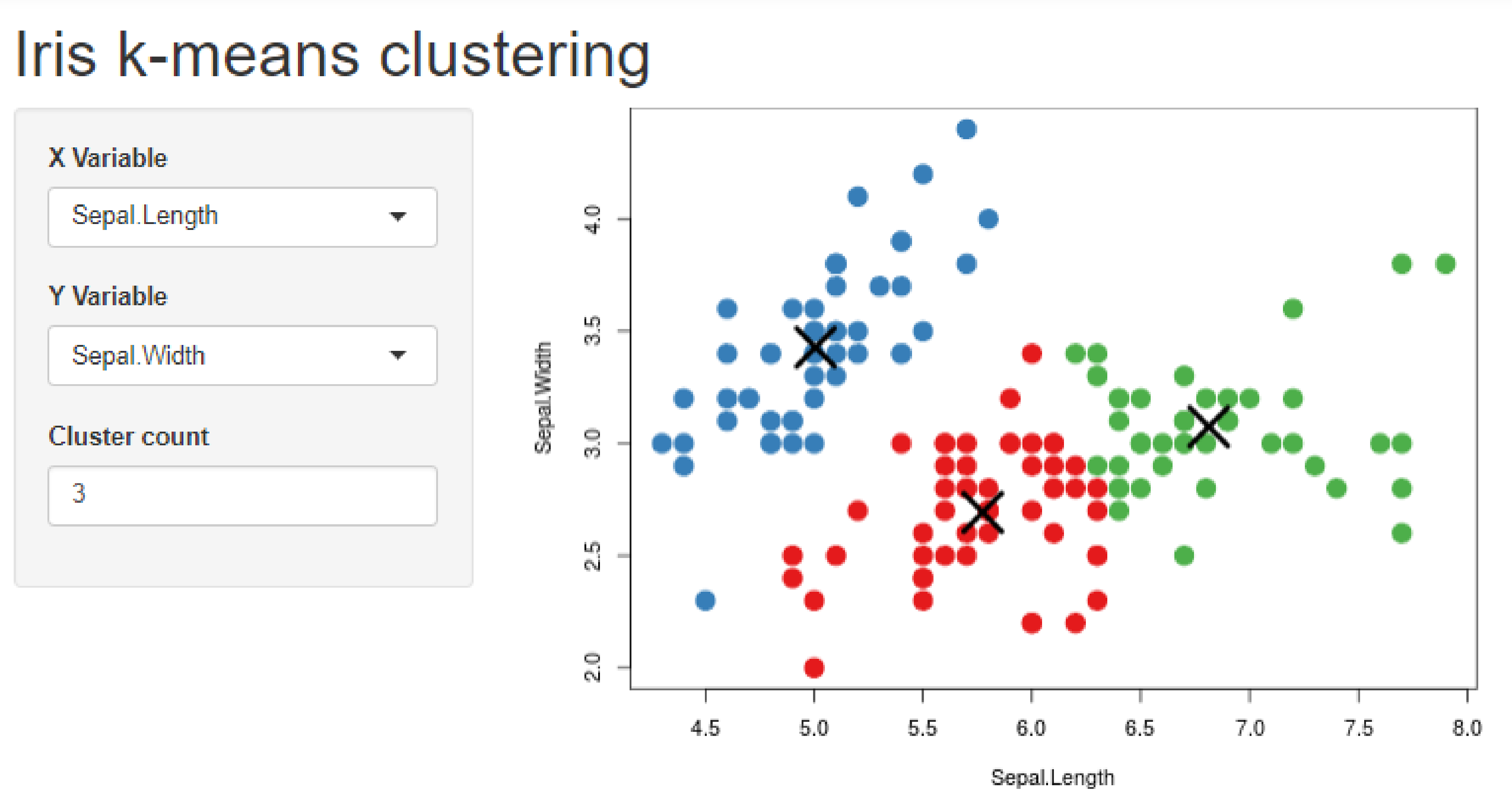
Lego Set Database Explorer

Interactive visualizations

Shiny is designed for fully interactive visualization, using JavaScript libraries like d3, Leaflet, and Google Charts.

Shiny

Why do we need it?



Iris k-means clustering

X Variable

Sepal.Length

Y Variable

Sepal.Width

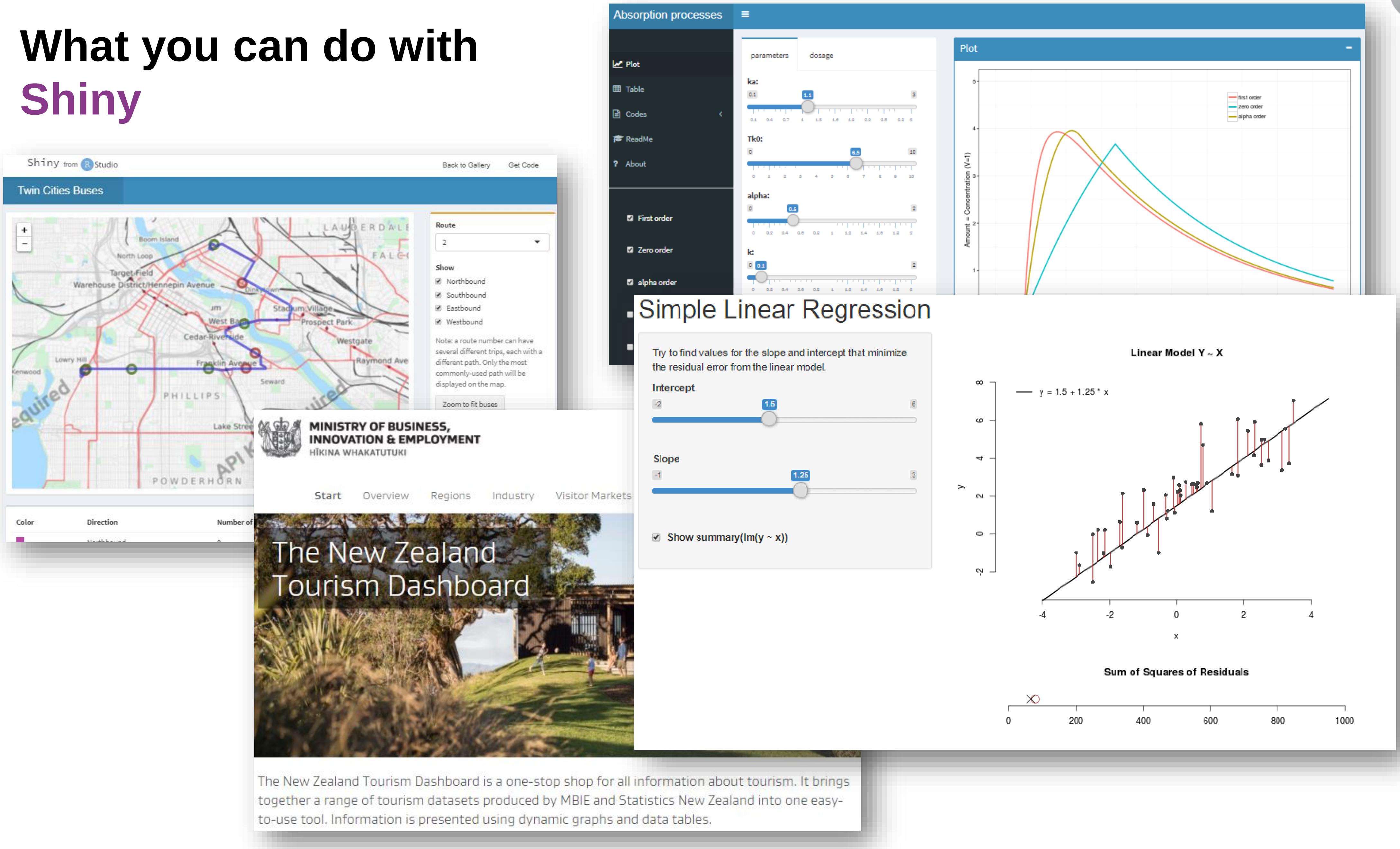
Cluster count

3

https://shiny.rstudio.com/gallery/kmeans-example.html

```
library(shiny)

UI <- pageWithSidebar(
  headerPanel('Iris k-means clustering'),
  sidebarPanel(
    selectInput('xcol', 'X Variable', names(iris)),
    selectInput('ycol', 'Y Variable', names(iris), selected=names(iris)[[2]]),
    numericInput('clusters', 'Cluster count', 3, min = 1, max = 9)
  ),
  mainPanel(
    plotOutput('plot1')
  )
)

SERVER <- function(input, output, session) {

   selectedData <- reactive({iris[, c(input$xcol, input$ycol)]
                              })
  clusters <- reactive({ kmeans(selectedData(), input$clusters)
                          })
  output$plot1 <- renderPlot({
    par(mar = c(5.1, 4.1, 0, 1))
    plot(selectedData(), col = clusters()$cluster, pch = 20, cex = 3)
    points(clusters()$centers, pch = 4, cex = 4, lwd = 4)
  })
}

shinyApp(UI, SERVER)
```

What you can do with **Shiny**



Shiny from RStudio · Back to Gallery · Get Code

Twin Cities Buses

Route: 2

Show
☑ Northbound
☑ Southbound
☑ Eastbound
☑ Westbound

Note: a route number can have several different trips, each with a different path. Only the most commonly-used path will be displayed on the map.

Zoom to fit buses

Color	Direction	Number of

Absorption processes

Plot · Table · Codes · ReadMe · About

☑ First order
☑ Zero order
☑ alpha order

parameters · dosage

ka: 1.1

Tk0: 6.5

alpha: 0.5

k: 0.1

Plot

MINISTRY OF BUSINESS, INNOVATION & EMPLOYMENT
HĪKINA WHAKATUTUKI

Start · Overview · Regions · Industry · Visitor Markets

The New Zealand Tourism Dashboard

The New Zealand Tourism Dashboard is a one-stop shop for all information about tourism. It brings together a range of tourism datasets produced by MBIE and Statistics New Zealand into one easy-to-use tool. Information is presented using dynamic graphs and data tables.

Simple Linear Regression

Try to find values for the slope and intercept that minimize the residual error from the linear model.

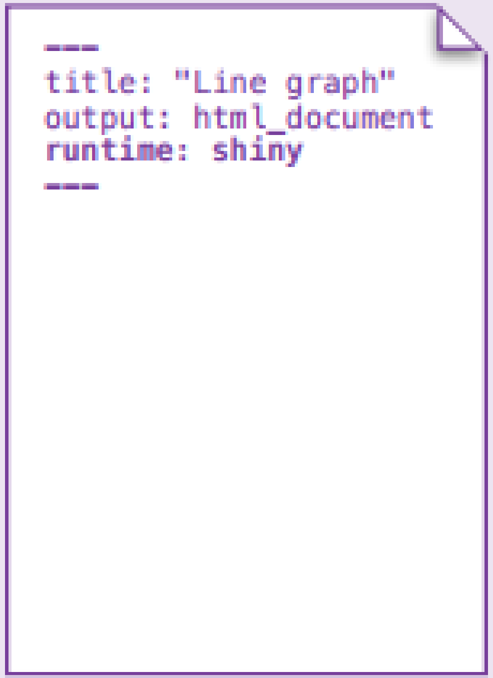
Intercept: 1.5 (-2 to 6)

Slope: 1.25 (-1 to 3)

☑ Show summary(lm(y ~ x))

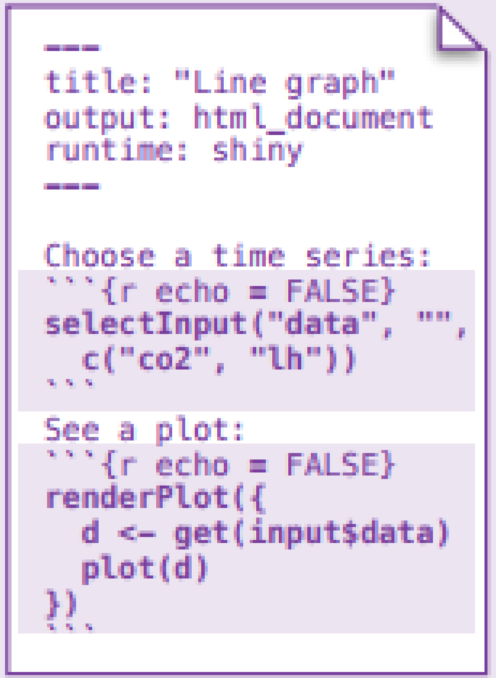
7. Interactive Docs Turn your report into an interactive Shiny document in 3 steps

1 Add **runtime: shiny** to the YAML header



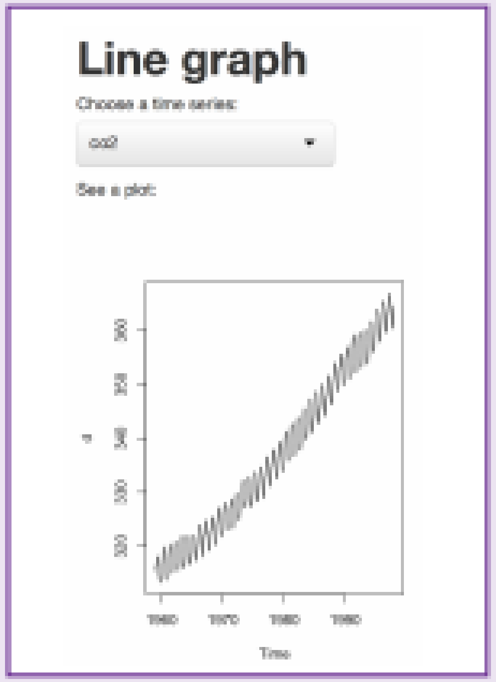
```
---
title: "Line graph"
output: html_document
runtime: shiny
---
```

2 In the code chunks, add Shiny **input** functions to embed widgets. Add Shiny **render** functions to embed reactive output



~~~
---
title: "Line graph"
output: html_document
runtime: shiny
---

Choose a time series:
```{r echo = FALSE}
selectInput("data", "",
  c("co2", "lh"))
```
See a plot:
```{r echo = FALSE}
renderPlot({
  d <- get(input$data)
  plot(d)
})
```
~~~

3 Render with **rmarkdown::run** or click **Run Document** in RStudio



Line graph

Choose a time series:

co2

See a plot:

Thank you ☺

Questions?

napiorkowska.adrianna@gmail.com

References:

- http://rmarkdown.rstudio.com
- https://shiny.rstudio.com/