Data Visualization Tools, Gapminder Demo: Tableau vs R

Anamaria Crisan
What we’ll talk about
Data Visualization Tools

Tool #1 : Tableau

Tool #2 : Shiny (R)
A comment on “How should we visualize data”

There are two aspects of visualizations to think about:

How do you make a visualization?  
Is it the right visualization?
Data Visualization Tools
Tools & Libraries: An Overview
See this excellent post by Lisa Charlotte Rost: [http://bit.ly/2gRGx1J](http://bit.ly/2gRGx1J)
(figures taken from her post)
Tools & Libraries: An Overview
See this excellent post by Lisa Charlotte Rost: [http://bit.ly/2gRGx1J](http://bit.ly/2gRGx1J)
(figures taken from her post)

Analysis vs Presentation

![Diagram showing tools and libraries for analysis and presentation](image-url)
Tools & Libraries: An Overview

See this excellent post by Lisa Charlotte Rost: http://bit.ly/2gRGx1J
(figures taken from her post)

Extent of Flexibility

How easy/hard it is to make data visualizations (including custom/novel visualizations)
Tools & Libraries: An Overview

See this excellent post by Lisa Charlotte Rost: http://bit.ly/2gRGx1J
(figures taken from her post)

**Static vs Interactive**

<table>
<thead>
<tr>
<th>Apps</th>
<th>Static</th>
<th>Web - Interactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illustrator,</td>
<td>Illustrator, NodeBox, Excel,</td>
<td>HighCharts Cloud, Quadrigram,</td>
</tr>
<tr>
<td>Polestar, Raw</td>
<td></td>
<td>EasyChrt, Datawrapper, Tableau,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plotly, Google Sheets</td>
</tr>
<tr>
<td>Charting</td>
<td>Ggplot2, Matplotlib, R,</td>
<td>D3, D4, C3, NVD3, GGvis,</td>
</tr>
<tr>
<td>Libraries</td>
<td>Seaborn, Bokeh, Processing</td>
<td>HighCharts, Shiny, Vega, Vega-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lite</td>
</tr>
</tbody>
</table>
Tools & Libraries : An Overview
See this excellent post by Lisa Charlotte Rost : http://bit.ly/2gRGx1J
(figures taken from her post)

“There are no perfect tools, just good tools for people with certain goals”
See a detailed table here: http://bit.ly/2DeWPwV
Tools & Libraries: An Overview

See this excellent post by Lisa Charlotte Rost: http://bit.ly/2gRGx1J
(figures taken from her post)
Important Details on Tableau Public

- I am not a Tableau Expert
  - There are plenty of great training videos & tutorials online
  - I mainly use R, and rarely need to use Tableau
  - I do, however, find Tableau pretty easy to use

- I’ll be demonstrating using Tableau Public
  - Some features are different than the version of Tableau you pay for

- YOU CANNOT PRIVATELY SAVE WORKBOOKS USING TABLEAU PUBLIC
  - Please don’t use Tableau Public for sensitive data, because anything you save will be publicly posted!
  - Tableau Desktop that you pay for *does* allow you to privately save workbooks
Today I’ll demonstrate the following

- Loading data
- Creating two types of maps
- Creating a scatter plot
- Exploring alternative visual designs with “Show Me”
- Putting it all together into a dashboard

I’ve put together some screen shots to help you reconstruct the Tableau analysis
Step 1: Load Gapminder CSV into Tableau
Step 2: Create a map with median gdp per capita as dots
Step 2: Create a map with median gdp per capita as dots

A: Drag Dimensions & measures to marks

B: Set gdpPerCaptive to median (sum is default)
Step 3: Create a small multiples choropleth map
Step 3: Create a small multiples choropleth map

A: Drag Dimensions & measures to marks

B: Create a new group

Same as before

New!
Step 3: Create a small multiples choropleth map

B: Create a new group
Step 4: Create a scatter plot

Try it on your own
Step 5: Explore the ShowMe panel

Try it on your own
Step 6: Create a dashboard

- Drag Map Points and Map Choropleth to Dashboard
- Drag Gapminder to Dashboard
- Create a filter

Dashboard Menu:
- Device Preview
- Size: min 420x560 - max 650x800

Sheets:
- Map Points
- Map Choropleth
- Gapminder
- Play with Showms
- Sheet 5

Objects:
- Horizontal
- Vertical
- Web Page
- Text
- Blank

Tiled vs Floating
- Show dashboard title
See more online at the Tableau Public gallery!

Important Details on R and Shiny

- To introduce shiny, I am using selected slide content from several presentations developed for the “Shiny quick start guide” available here:

- These quick start slides are also presented as a video on the Shiny website
  - https://shiny.rstudio.com/tutorial/
The Challenges of Briefly Introducing Shiny

How to draw an owl

1. Draw some circles
2. Draw the rest of the owl

- Shiny is powerful and somewhat simpler than web languages - but it still takes some time to learn it
- I want to show you some basic info, and give you a sense of what Shiny can do - but there is a lot of in between I won’t cover
What is Shiny?

HTML 5
Basic Structure

CSS
Presentation, Layouts, formatting

JS
Webpage / Application Behavior

+ R
Most glorious of programming languages
How Does Shiny Work?

Every Shiny app is maintained by a computer running R.

Server runs R code

User Interface (UI)

bit.ly/shiny-quickstart-1
Basic Structure of a Shiny Application

library(shiny)

ui <- fluidPage()

server <- function(input, output) {}

shinyApp(ui = ui, server = server)
Basic Structure of a Shiny Application

An entire application can be a single file, or broken down into a `ui.R` & `server.R` files

```r
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
              label = "Choose a number",
              value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}

shinyApp(ui = ui, server = server)
```

You **must** use `app.r` (single file); or `ui.r` and `server.r` names, or your application won’t run
The user interface (ui.r) houses inputs & outputs

Example of inputs

<table>
<thead>
<tr>
<th>Buttons</th>
<th>Single checkbox</th>
<th>Checkbox group</th>
<th>Date input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td><img src="#" alt="Checkbox Input" /></td>
<td><img src="#" alt="Checkbox Group Input" /></td>
<td><img src="#" alt="Date Input" /></td>
</tr>
<tr>
<td>Submit</td>
<td><img src="#" alt="Selection" /></td>
<td><img src="#" alt="Selection" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td>actionButton()</td>
<td>checkboxGroupInput()</td>
<td>dateInput()</td>
</tr>
<tr>
<td></td>
<td>submitButton()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Date range

- [2014-01-24](#) to [2014-01-24](#)

- dateRangeInput()

File input

- File: No file chosen

- fileInput()

Numeric input

- Value: 1

- numericInput()

Password Input

- Password: 

- passwordInput()

Radio buttons

- Choice 1
- Choice 2
- Choice 3

- radioButtons()

Select box

- Choice 1

- selectInput()

Sliders

- Slider: 75

- sliderInput()

Text input

- Text: Enter text...

- textInput()
The user interface (ui.r) houses **inputs & outputs**

Example of inputs

```
sliderInput(inputId = "num", label = "Choose a number", ...)
```

Changing the slider number (i.e. dragging the circle left to right) causes R to automatically initiate an action to update all outputs that use the slider number.
The user interface (ui.r) houses inputs & outputs

Example of outputs

<table>
<thead>
<tr>
<th>Function</th>
<th>Inserts</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataTableOutput()</td>
<td>an interactive table</td>
</tr>
<tr>
<td>html1Output()</td>
<td>raw HTML</td>
</tr>
<tr>
<td>imageOutput()</td>
<td>image</td>
</tr>
<tr>
<td>plotOutput()</td>
<td>plot</td>
</tr>
<tr>
<td>tableOutput()</td>
<td>table</td>
</tr>
<tr>
<td>textOutput()</td>
<td>text</td>
</tr>
<tr>
<td>uiOutput()</td>
<td>a Shiny UI element</td>
</tr>
<tr>
<td>verbatimTextOutput()</td>
<td>text</td>
</tr>
</tbody>
</table>

Function = command telling R & shiny to do some particular task
server.r stitches inputs & outputs together

**ui.r**
- Shiny widgets
- Interface elements

**server.r**
- ggplot code
- Dataset manipulation

Output & render functions are complimentary

- `output function`
- `render function`
- `plotOutput()` → `renderPlot()`
- `tableOutput()` → `renderTable()`
A simple worked example

```r
# 01-two-inputs
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
              label = "Choose a number",
              value = 25, min = 1, max = 100),
  textInput(inputId = "title",
            label = "Write a title",
            value = "Histogram of Random Normal Values"),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num), main = input$title)
  })
}

shinyApp(ui = ui, server = server)
```

bit.ly/shiny-quickstart-1
Reactivity in Shiny – a Very Brief Introduction

```r
# 01-two-inputs
library(shiny)

ui <- fluidPage(
    sliderInput(inputId = "num",
                label = "Choose a number",
                value = 25, min = 1, max = 100),
    textInput(inputId = "title",
              label = "Write a title",
              value = "Histogram of Random Normal Values"),
    plotOutput("hist")
)

server <- function(input, output) {
    output$hist <- renderPlot({
        hist(rnorm(input$num), main = input$title)
    })
}

shinyApp(ui = ui, server = server)
```

bit.ly/shiny-quickstart-1
Reactivity in Shiny – a **Very** Brief Introduction

```r
input$num

input$title

output$hist <- renderPlot({
  hist(rnorm(input$num),
       main = input$title)
})
```
Let’s go through a more complex example
Shiny Apps for the Enterprise

See more online at the Shiny Gallery!

https://www.rstudio.com/products/shiny/shiny-user-showcase/

bit.ly/shiny-quickstart-1