

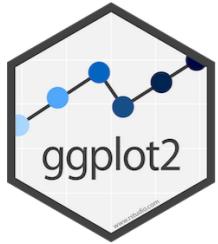


VISUALIZATION & SHINY

CEPE GENES
Frédéric Logé

Program

Visualization : building graphs with ggplot2 [~Morning]



- Why ggplot2 (and not another package) ?
- Grammar of Graphics
- Practice 

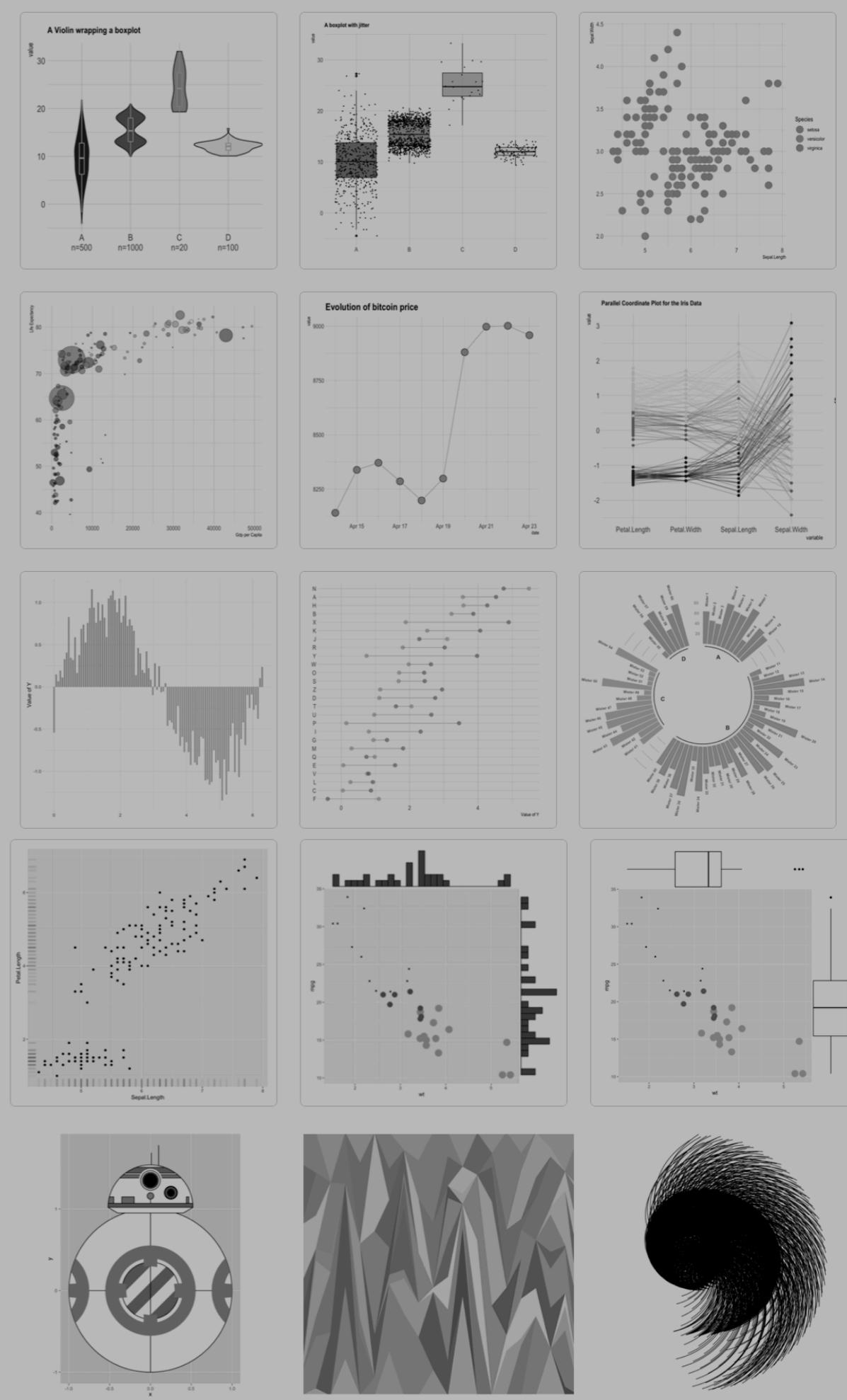
Building dashboard and web apps with Shiny [~Afternoon]



- Basic app with input and output
- Going deeper into reactivity
- Practice 

VISUALIZATION

BUILDING GRAPHS WITH

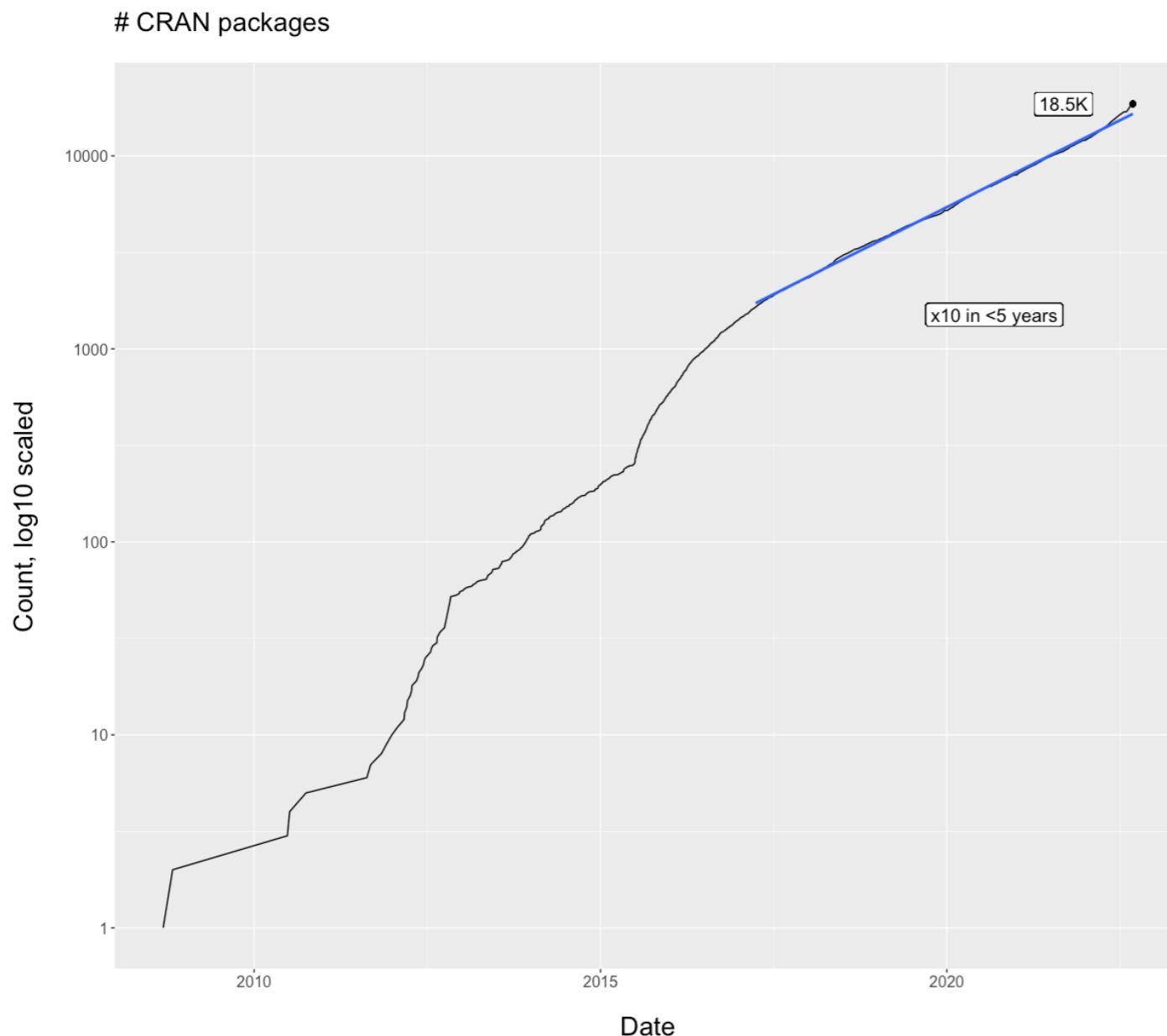


Why ggplot2 ?

What is out there for data visualization ?

.....

Close to 20K package on CRAN ...



... but there is one to rule them all !



Note: this only on CRAN. There are many packages simply hosted on GitHub.

Why ggplot2 ?

R packages - the data visualization landscape

Package	Monthly Downloads*	Code readability	Aesthetics	Scope	Interactive	Support
Base	-	💩	😱	✓	✗	💪💪
	2,200 K	🤔	😅	✓	✓	💪💪💪💪
Lattice	54 K	🤔	😱	✓	✗	💪💪
rgl	1,000 K	💩	😔	✗	✓	💪💪
Dygraphs	35 K	🤔	😅	✗	✓	💪
	267 K	🤔	😅	✓	✓	💪

* the numbers were collected on January 4th 2023, from <https://cranlogs.r-pkg.org/>

As bonus:

- * if you wish to use ggplot2 in Python, you can !! With plotnine package with the exact same syntax !!
- * >100 packages extensions of ggplot2, see them [here](#)

Why ggplot2 ?

RStudio

Some best practices

- When working on something new, create a new Project in RStudio (or any IDE you work with)
- Always read error messages, they will help you find your solution (Google and StackOverflow)
- For today's session, if you are stuck, tell me 😊

Practice 

Grammar of Graphics

In summary

INIT GG PLOT
Specifying a Tibble as data
ggplot()

AESTHETICS
What to put in x, y,
color, fill, shape, size ...
aes() and aes()_

GEOMETRY
The geometry to display: a
scatterplot, histogram,
boxplot, ...
geom_*

DESIGN
labs() and theme()

FACETTING
split graphic by categories
facet_wrap() or
facet_grid()

SCALE
Managing scales
scale_*_*()

+
Operator to combine
ggproto objects

```
1 library(tidyverse)
2 library(ggrepel)
3
4 data("starwars")
5
6 starwars_ep3 = starwars %>%
7   filter(map_lgl(films, function(x) "Revenge of the Sith" %in% x))
8
9 ggplot(data = starwars_ep3) +
10  aes(x = height, y = mass, label = name, col = gender) +
11  geom_point() +
12  geom_label_repel() +
13  labs(title = 'Physical attributes of Star Wars ep.3',
14    x = 'Height (cm)', y = 'Mass (kg)') +
15  theme(text = element_text(size = 20), legend.position = 'bottom')
```

DASHBOARDS & WEBAPPS

WITH



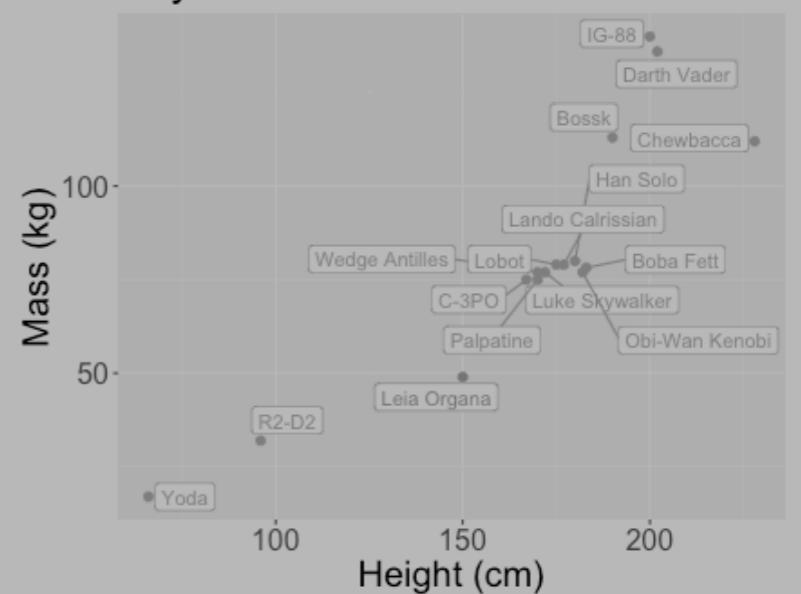
Star Wars characters

Film

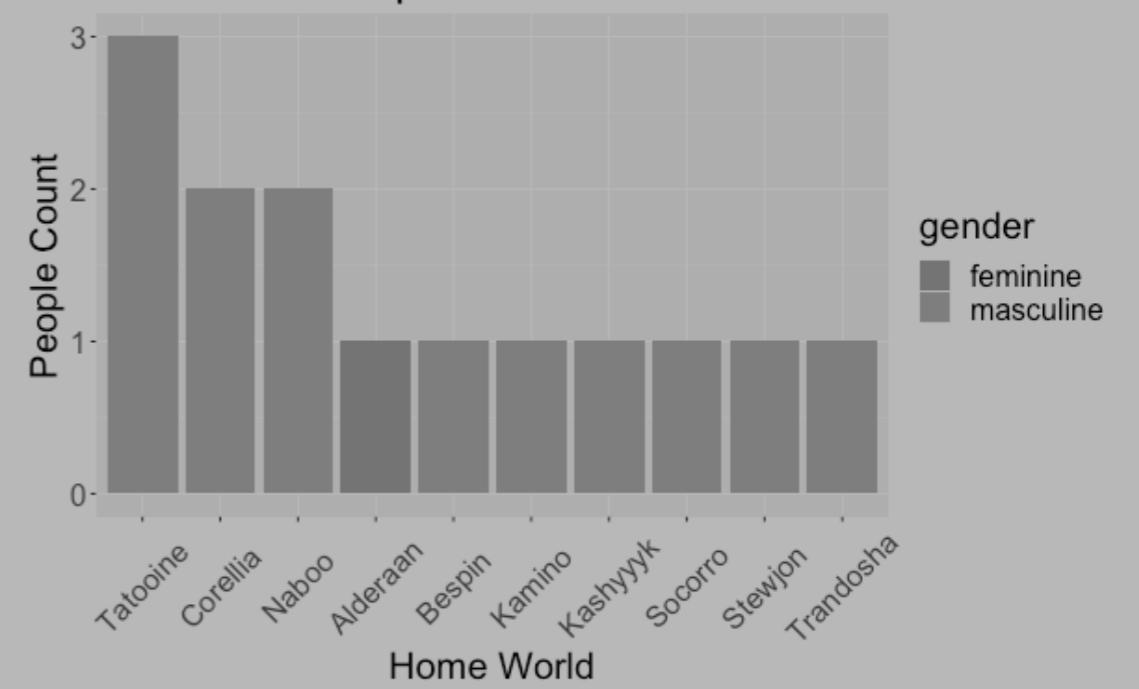
The Empire Strikes Back

	name	birth_year	species
1	Luke Skywalker	19	Human
2	C-3PO	112	Droid
3	R2-D2	33	Droid
4	Darth Vader	41.9	Human

Physical attributes of Star Wars



Home World representation



Shiny: what is that ?

Shiny is an R package to create web application based solely on R code. It is structured as such:

```
1 # Define UI for application that checks pwd
2 ui <- fluidPage(
3    textInput("pwd", "Your pwd:"), 
4     textOutput("pwd_is_ok")
5 )
6
7 # Define server logic required to check pwd
8 server <- function(input, output) {
9     output$pwd_is_ok <- renderText({
10         n = nchar(input$pwd)
11         if(n<=8){
12             return("NOT OK: >= 8 chars!")
13         }else{
14             return("ALL GOOD !")
15         }
16     })
17 }
18
19 # Run the application
20 shinyApp(ui = ui, server = server)
```

Shiny functions generating the HTML and CSS, specifying:
➤ input element HTML tags
➤ Output element HTML tags
➤ Overall layout and design

R function with args (input, output) contains calls to specific Shiny functions to handle reactivity logic of the

Shiny: what is that ?

Shiny is an R package to create web application based solely on R code. It is structured as such:

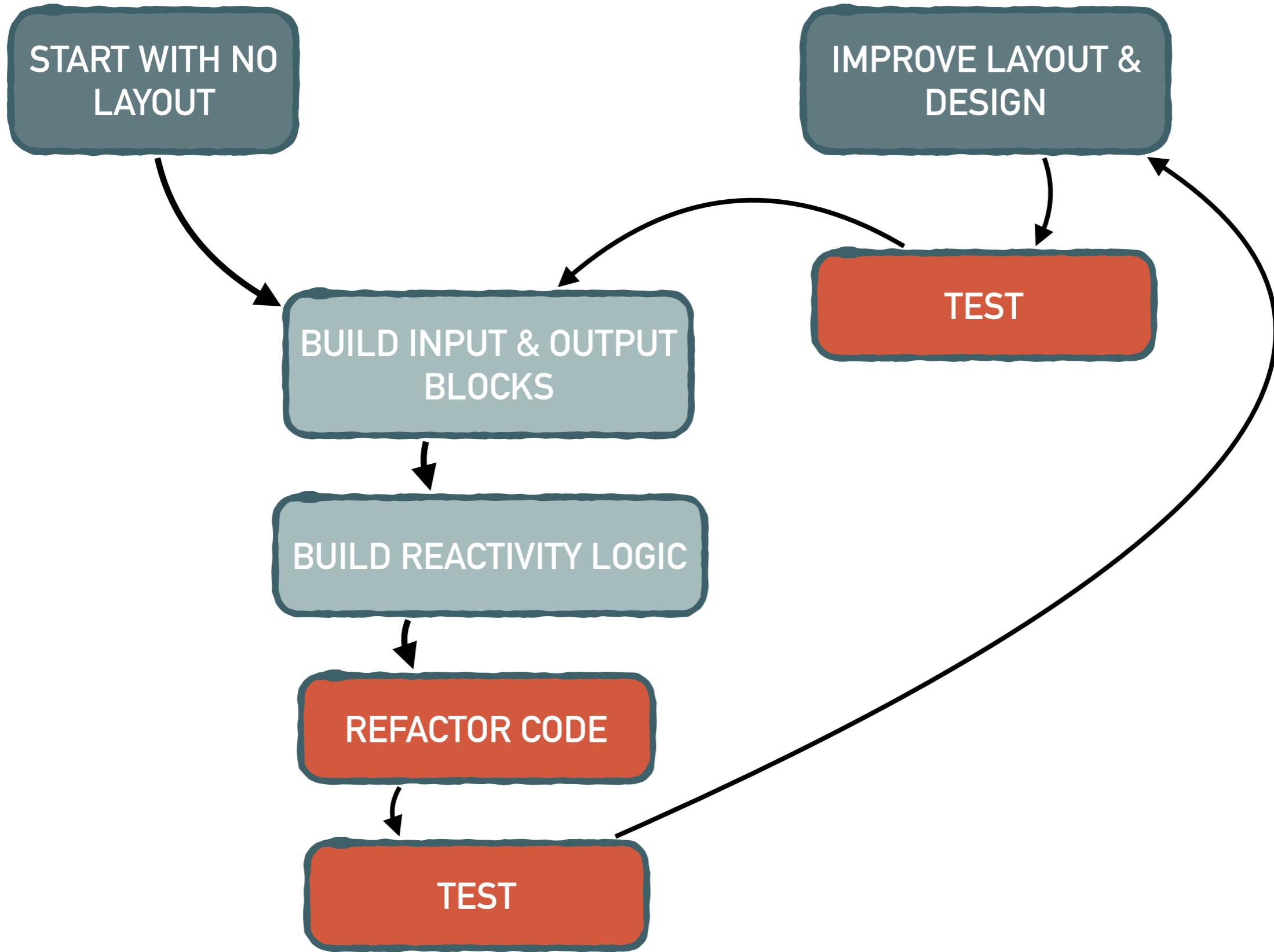
```
1 # Define UI for application that checks pwd
2 ui <- fluidPage(
3    textInput("pwd", "Your pwd:"), 
4     textOutput("pwd_is_ok")
5 )
6
7 # Define server logic required to check pwd
8 server <- function(input, output) {
9     output$pwd_is_ok <- renderText({
10         n = nchar(input$pwd)
11         if(n<=8){
12             return("NOT OK: >= 8 chars!")
13         }else{
14             return("ALL GOOD !")
15         }
16     })
17 }
18
19 # Run the application
20 shinyApp(ui = ui, server = server)
```

Shiny functions generating the HTML and CSS, specifying:
➤ input element HTML tags
➤ Output element HTML tags
➤ Overall layout and design

R function with args (input, output) contains calls to specific Shiny functions to handle reactivity logic of the



Incremental process for building the app



Handling outputs

Outputs are specified in UI (e.g. `textOutput`) and the rendering is specified in server
(e.g. `output$some_output_id <- renderOutput({ ... })`)

Creating :	In UI	In server
Raw HTML	<code>htmlOutput('id')</code> <code>uiOutput('id')</code>	<code>output\$id = renderUI({ ... })</code>
Text	<code>textOutput('id')</code> <code>verbatimTextOutput('id')</code>	<code>output\$id = renderText({ ... })</code> <code>output\$id = renderPrint({ ... })</code>
Table	<code>tableOutput('id')</code>	<code>output\$id = renderTable({ ... })</code>
DataTable	<code>dataTableOutput('id')</code>	<code>output\$id = renderDataTable({ ... })</code>
Image	<code>imageOutput('id')</code>	<code>output\$id = renderImage({ ... })</code>
Plot	<code>plotOutput('id')</code>	<code>output\$id = renderPlot({ ... })</code>
Plotly	<code>plotlyOutput('id')</code>	<code>output\$id = renderPlotly({ ... })</code>

Render your plots interactive, with hovering and clicking options

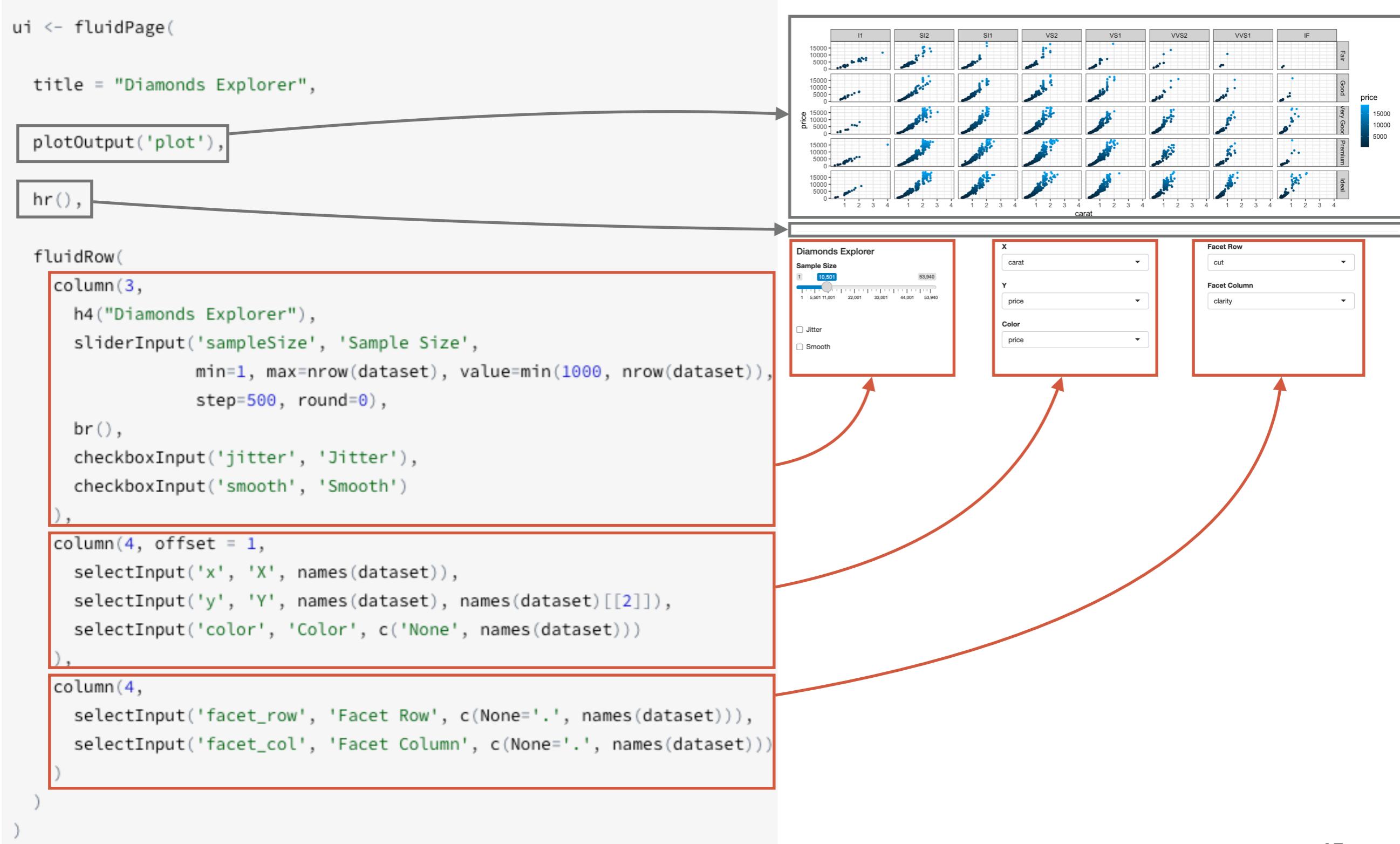
<https://shiny.rstudio.com/articles/plot-interaction.html>

Introducing reactivity

	Reactivity functions	Objective	Output
Trigger	observe	Do some operations whenever a called input changes	None
	observeEvent	Do some operations whenever a trigger input is modified	None
	isolate	Isolate an input such that reactivity is not affected by change of its value	None
Data handling	reactive	Compute some data whenever a called input changes	Function
	eventReactive	Compute some data whenever a trigger input is modified	Function
	reactiveValues	Store data that you wish to be accessible and modifiable from any part of the server -> best way to handle loaded data on app	None

Check out <https://rstudio.github.io/reactlog/> to test your reactivity processes

Grid Layout



Layout & design

Improving layout

For custom layout

<https://shiny.rstudio.com/articles/layout-guide.html>

Rely on shiny extensions

<https://rstudio.github.io/flexdashboard/articles/examples.html> [**flexdashboard**]

<https://rstudio.github.io/shinydashboard/examples.html> [**shinydashboard**]

<https://rinterface.github.io/shinydashboardPlus/index.html> [**shinydashboardPlus**]

Improve rendering

Customize theme

<https://shiny.rstudio.com/articles/themes.html>

Customizing UI with html and css

Deployment

Where to deploy the app, and how ?

Local machine with R env running

Server with R env running - access and setup requires IT expertise if internal, if external quite easy to do (see this auto on using AWS's servers to deploy an app <https://www.charlesbordet.com/en/guide-shiny-aws/#>)

For open access : shinyapp.io

Authentication management in the app

Packages: [shinymanager](#), [Shinyauthr](#)

Studio resource: <https://shiny.rstudio.com/gallery/authentication-and-database.html>

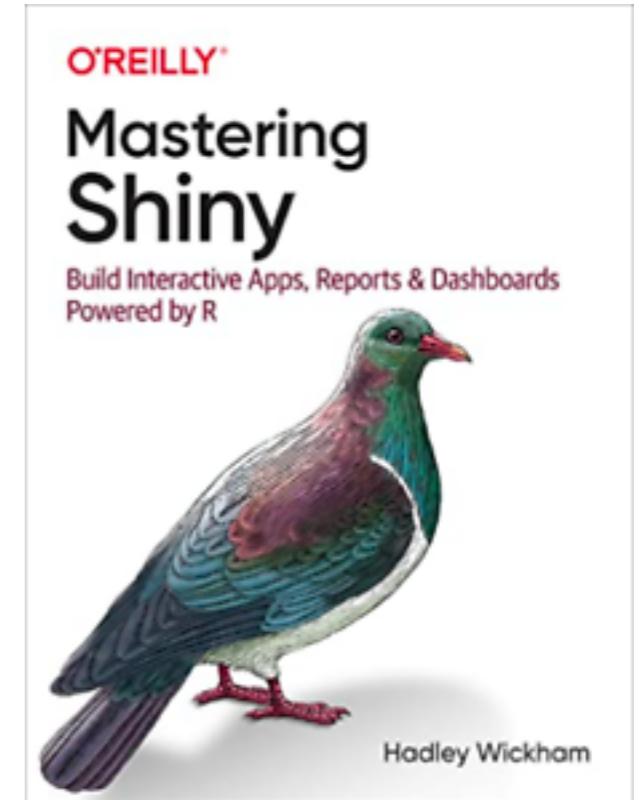
Response time

R profiling is a great tool to look at the amount of time each piece of your code takes, in order to prioritize dev efforts

Last words

For a deeper understanding of Shiny, check Mastering Shiny, by Hadley Wickham at O'Reilly.

Available for free at <https://mastering-shiny.org> or get the hard copy for 50€.



For any questions, reach me at frederic.logemunerel@gmail.com

EXTRAS

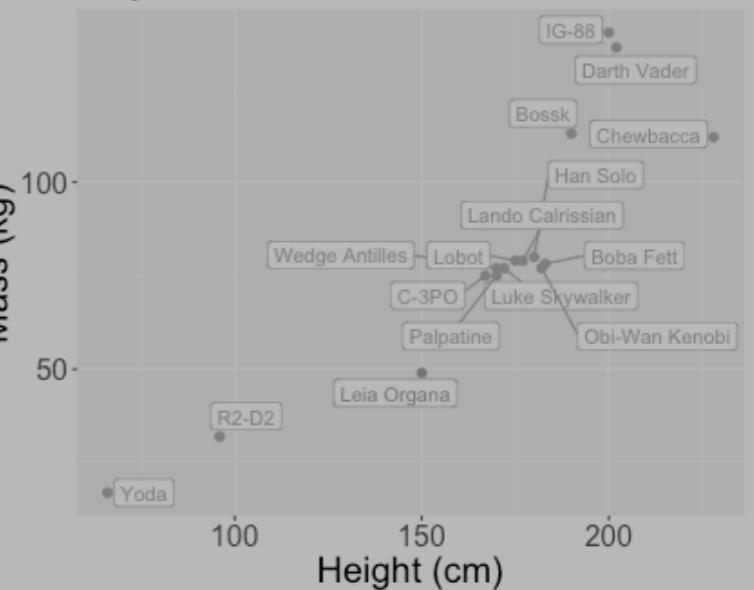
Star Wars characters

Film

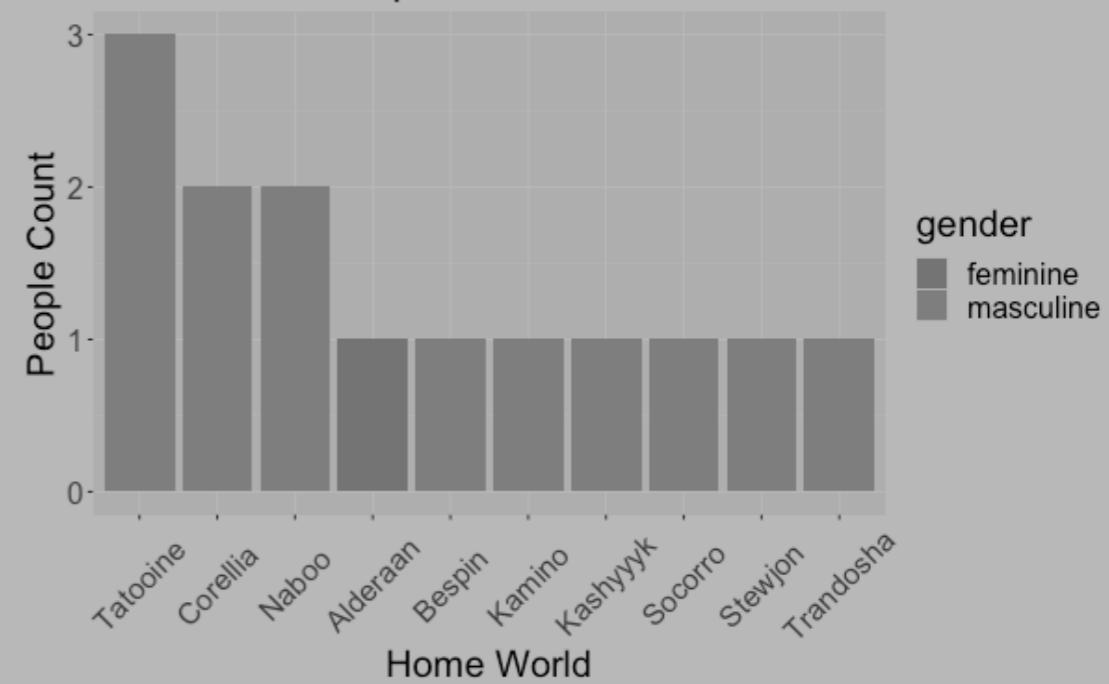
The Empire Strikes Back

	name	birth_year	species
1	Luke Skywalker	19	Human
2	C-3PO	112	Droid
3	R2-D2	33	Droid
4	Darth Vader	41.9	Human

Physical attributes of Star Wars



Home World representation



Reactivity - Trigger

observe({}) - do some operations whenever a called input changes, no output

```
4 ui <- fluidPage(  
5  textInput(inputId = "free_input", label = "Free Input"),  
6   selectInput(inputId = "film", label = "Film",  
7               choices = unique(unlist(starwars$films)))  
8 )  
9  
10 random_code <- function(){  
11   paste0(sample(letters, 10), collapse = '')  
12 }  
13  
14 server <- function(input, output, session) {  
15  
16   observe({  
17     print(input$film)  
18     updateSliderInput(inputId = "free_input",  
19                         session = session,  
20                         value = random_code())  
21   })  
22  
23 }  
24  
25 shinyApp(ui, server)
```

LAYOUT

INPUT

OUTPUT

REACTIVITY

Reactivity - Trigger

observeEvent({}) : do some operations whenever a specific input is modified, no output

```
3 ui <- fluidPage(  
4   actionButton("doMagicTrick", "Show something"),  
5   textOutput("myText")  
6 )  
7  
8 server <- function(input, output) {  
9   observeEvent(input$doMagicTrick, {  
10     output$myText <- renderText({  
11       paste0('You are ...', sample(starwars$name, 1), '!')  
12     })  
13   })  
14 }  
15  
16 shinyApp(ui, server)
```

Reactivity - Trigger / or not

isolate({}) - stop input modification from causing changes

.....

```
4 ui <- fluidPage(  
5  textInput(inputId = "free_input", label = "Free Input"),  
6   selectInput(inputId = "film", label = "Film", choices = unique(unlist(starwars$films))),  
7   textOutput('output_text')  
8 )  
9  
10 server <- function(input, output) {  
11  
12   output$output_text <- renderText({  
13     paste0(isolate(input$film), " - ", input$free_input)  
14   })  
15  
16 }  
17  
18 shinyApp(ui, server)
```

LAYOUT

INPUT

OUTPUT

REACTIVITY

Reactivity - Data handling

reactive({}) - update dataset required for several outputs when input changes

```
5 ui <- fluidPage(  
6   selectInput(inputId = "film", label = "Film", choices = unique(unlist(starwars$films))),  
7   textOutput('genericInfo'),  
8   br(),  
9   h5('First 4 characters : '),  
10  dataTableOutput("myTable")  
11 )  
12  
13 server <- function(input, output, session) {  
14  
15   df <- reactive({  
16     starwars %>%  
17     filter(map_lgl(films, function(x) input$film %in% x))  
18   })  
19  
20   output$genericInfo <- renderText({  
21     paste0('Nb of characters : ', nrow(df()))  
22   })  
23  
24   output$myTable <- renderDataTable({  
25     datatable(head(df(),c('name', 'birth_year', 'species')), 4),  
26     options = list(dom = 't'))  
27   })  
28  
29  
30 }  
31  
32 shinyApp(ui, server)
```

Note that reactive outputs a function
callable anywhere in server

LAYOUT

INPUT

OUTPUT

REACTIVITY

Reactivity - Data handling

eventReactive({}) : when a single input is modified, output new data

```
3 ui <- fluidPage(  
4   headerPanel("Example eventReactive"),  
5  
6   mainPanel(  
7  
8     # input field  
9    textInput("user_text", label = "Enter some text:", placeholder = "Please enter some text."),  
10  
11    # submit button  
12    actionButton("submit", label = "Submit"),  
13  
14    # display text output  
15    textOutput("text"))  
16 )  
17  
18 server <- function(input, output) {  
19  
20   # reactive expression  
21   text_reactive <- eventReactive(input$submit, {  
22     input$user_text  
23   })  
24  
25   # text output  
26   output$text <- renderText({  
27     text_reactive()  
28   })  
29 }
```

Note that eventReactive outputs a function callable anywhere in server

LAYOUT INPUT OUTPUT REACTIVITY

Reactivity - Data handling

reactiveValues({}) : handling data

```
3 ui <- fluidPage(  
4   # input field  
5  textInput("user_text", label = "Enter some text:", placeholder = "Please enter some text."),  
6   actionButton("submit", label = "Submit"),  
7  
8   # display text output  
9   textOutput("text")  
10 )  
11  
12 server <- function(input, output) {  
13  
14   # observe event for updating the reactiveValues  
15   observeEvent(input$submit,  
16     {  
17       text_reactive$text <- input$user_text  
18     })  
19  
20   # reactiveValues  
21   text_reactive <- reactiveValues(  
22     text = "No text has been submitted yet."  
23   )  
24  
25   # text output  
26   output$text <- renderText({  
27     text_reactive$text  
28   })  
29 }
```

Note that reactiveValues outputs a list callable anywhere in server

LAYOUT INPUT OUTPUT REACTIVITY

The need for refactoring

Before

```

18- server = shinyServer(function(input, output) {
19
20   df <- reactive({ starwars %>% filter(map_lgl(films, function(x) input$film %in% x)) })
21
22- output$table1 <- renderDataTable({
23   datatable(head(df(),c('name', 'birth_year', 'species')), 4),
24     options = list(dom = 't'))
25 })
26
27- output$physic1 <- renderPlot({
28   h = df()
29   print(h)
30   ggplot(data = h) +
31     aes(x = height, y = mass, label = name, col = gender) +
32     geom_point(cex = 2) +
33     geom_label_repel() +
34     coord_fixed() +
35     labs(title = 'Physical attributes of Star Wars',
36          x = 'Height (cm)', y = 'Mass (kg)') +
37     theme(text = element_text(size = 20), legend.position = 'bottom')
38 })
39
40- output$physic2 <- renderPlot({
41   hh = df() %>%
42     filter(is.na(homeworld) == FALSE) %>%
43     count(homeworld, gender) %>%
44     group_by(homeworld) %>%
45     mutate(nn = sum(n)) %>%
46     slice_max(order_by = nn, n = 10)
47   ggplot(data = hh) +
48     aes(x = reorder(homeworld, -n), y = n, fill = gender) +
49     geom_bar(stat = 'identity') +
50     labs(x = 'Home World', y = 'People Count', title = 'Home World representation') +
51     theme(text = element_text(size = 20),
52           axis.text.x = element_text(vjust = 0.5, angle = 45))
53 })
54
55 })

```

After

```
1 height_vs_mass_graph <- function(filtered_df){  
11  
12 homeworld_count <- function(filtered_df){  
29  
  
48 server = shinyServer(function(input, output) {  
49  
50   df <- reactive({ starwars %>% filter(map_lgl(films, function(x) input$film %in% x)) })  
51  
52   output$table1 <- renderDataTable({  
53     datatable(head(df(),c('name', 'birth_year', 'species')), 4),  
54       options = list(dom = 't'))  
55   })  
56  
57   output$physic1 <- renderPlot({ height_vs_mass_graph(df()) })  
58  
59   output$physic2 <- renderPlot({ homeworld_count(df()) })  
60  
61 })
```

Graph code and data process can render overall code readability a nightmare
=> create outside app functions for increased readability + testing

The app can be organized in different ways :

- functions called can be in separate files, sourced at the beginning
- ui and server can be defined in separate files

You can even integrate it in markdown document !!

Shiny app content

An app can be decomposed in five essentials:

- Inputs : where the user can specify information
- Outputs : where information is displayed and visualized
- Reactivity links : defining how some element is affected by another (e.g. an output by an input)
- Layout : how inputs and outputs and other fixed material (e.g. logo) is displayed on the app
- Custom design : font and color choices for the app

Radiant Data Design ▾ Basics ▾ Model ▾ Multivariate ▾ Report ▾ ⚒ ⚑ ? ▾

Datasets:

diamonds

Add/edit data description
 Rename data

Display:

preview str summary

Load data of type:

rds | rda | rdata

Browse... No file selected

Save data to type:

rds

Save

Show R-code

Manage View Visualize Pivot Explore Transform Combine

Data preview

price	carat	clarity	cut	color	depth	table	x	y	z	date
580	0.32	VS1	Ideal	H	61.00	56.00	4.43	4.45	2.71	2012-02-26
650	0.34	SI1	Very Good	G	63.40	57.00	4.45	4.42	2.81	2012-02-26
630	0.30	VS2	Very Good	G	63.10	58.00	4.27	4.23	2.68	2012-02-26
706	0.35	VVS2	Ideal	H	59.20	56.00	4.60	4.65	2.74	2012-02-26
1080	0.40	VS2	Premium	F	62.60	58.00	4.72	4.68	2.94	2012-02-26
3082	0.60	VVS1	Ideal	E	62.50	53.70	5.35	5.43	3.38	2012-02-26
3328	0.88	SI1	Ideal	I	61.70	56.00	6.14	6.18	3.80	2012-02-26
4229	0.93	SI1	Premium	E	61.40	57.00	6.34	6.23	3.86	2012-02-26
1895	0.51	VVS2	Very Good	G	63.40	57.00	5.09	5.06	3.22	2012-02-26
3546	1.01	SI2	Good	E	63.90	58.00	6.31	6.37	4.05	2012-02-26

10 of 3,000 rows shown. See View-tab for details.

Inputs

Output

Prices of 3,000 round cut diamonds

Geyser example

Follow these steps :

1. File > New File > Shiny Web App
2. A popup appears : put a random name (modifiable after), and click on `Single file`
3. A folder is created with an `app` R file. Open it. `<!> install.packages('shiny')`
4. Click on the RunApp button to see the result !

User Interface

```
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
15   # Application title
16   titlePanel("Old Faithful Geyser Data"),
17
18   # Sidebar with a slider input for number of bins
19   sidebarLayout(
20     sidebarPanel(
21       sliderInput("bins",
22         "Number of bins:",
23         min = 1,
24         max = 50,
25         value = 30)
26     ),
27
28   # Show a plot of the generated distribution
29   mainPanel(
30     plotOutput("distPlot")
31   )
32 )
33 )
```

Check value of ui to see html code

Server

```
35 # Define server logic required to draw a histogram
36 server <- function(input, output) {
37
38   output$distPlot <- renderPlot({
39     # generate bins based on input$bins from ui.R
40     x <- faithful[, 2]
41     bins <- seq(min(x), max(x), length.out = input$bins + 1)
42
43     # draw the histogram with the specified number of bins
44     hist(x, breaks = bins, col = 'darkgray', border = 'white')
45   })
46 }
47
48 # Run the application
49 shinyApp(ui = ui, server = server)
```

LAYOUT INPUT OUTPUT

Three basic components with
which you'll be working

Why ggplot2 ?

R packages - the dataviz landscape

ggplot2



The Standard package for data visualization

3 300K Monthly Downloads

Extremely readable syntax (grammar of graphics)

Cost of entry : medium

Maintained by RStudio and co, part of the tidyverse

#1 choice of R community and the standard of graphics
can be re-used in Python via nineplot.

So many packages are built as add-ons of ggplot2,
see all of them [here](#) and some of theme below:



Base plots

Installed by default

Hard/customized plots : nightmare

Syntax quickly becomes unreadable

Not the nicest designs

Lattice, DiagrammeR, igraph, ...

so many packages, often doing single-type plot

far less used than ggplot2 and extensions

most of what they do can be done via ggplot2 and extensions

often require less ‘work’ than ggplot2 for quick tests

Package	Nb monthly downloads
Cowplot	280K
Lattice	155K
LatticeExtra	240K
ggrepel*	500K

*ggplot2 add-on, adding nice annotation to graph

Plotly, Echarts4, ...

Cross-platform

Very pretty outputs

Interactivity

Similar grammar to ggplot2

Much less support



300K Monthly
Downloads



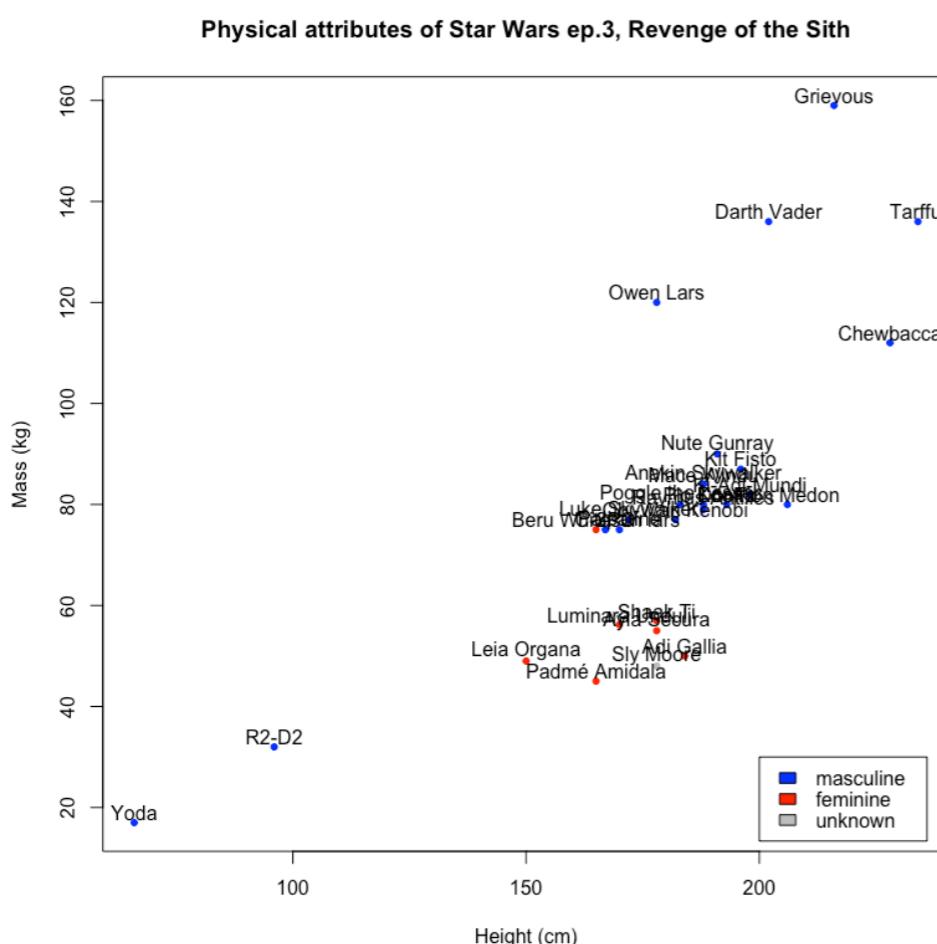
3K Monthly
Downloads

How ggplot2 ?

From base to ggplot2 - radical change of grammar, for the best

base

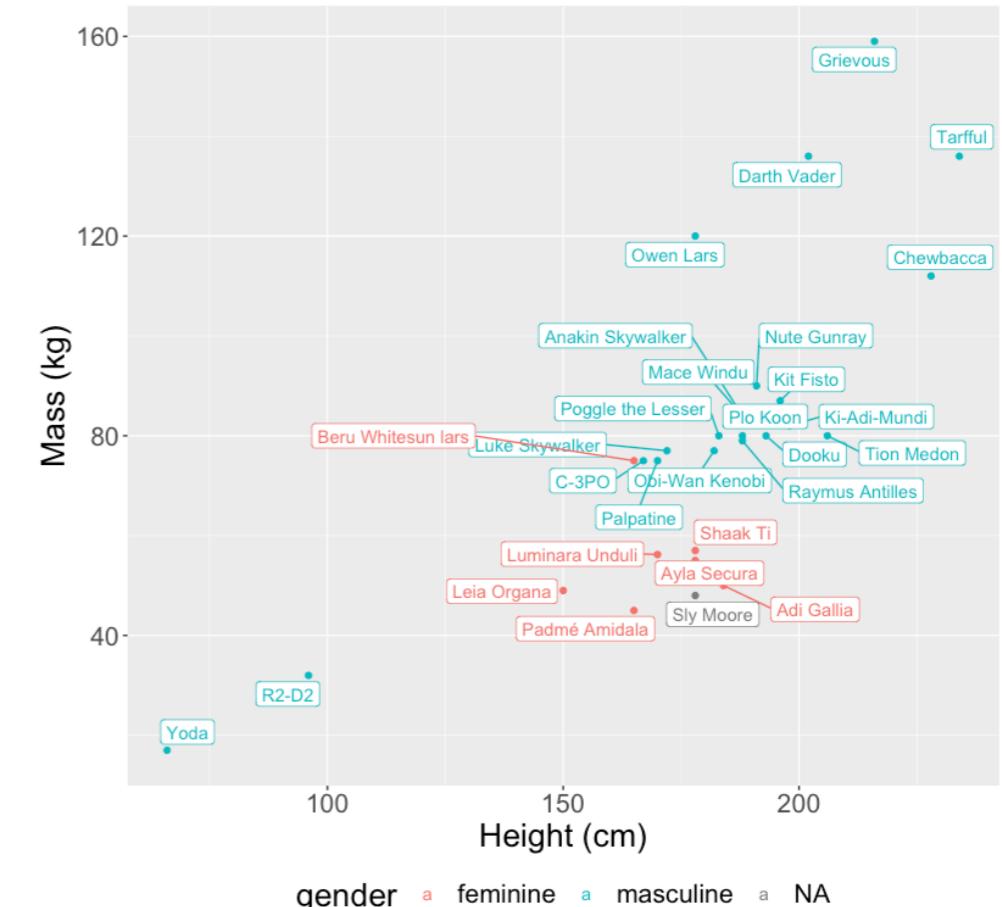
```
1 l = c('masculine' = 'blue', 'feminine' = 'red')
2 plot(x = starwars_ep3$height, y = starwars_ep3$mass,
3       col = map_var(l, starwars_ep3$gender, 'gray'), pch = 20,
4       main = 'Physical attributes of Star Wars ep.3',
5       xlab = 'Height (cm)', ylab = 'Mass (kg)')
6 text(x = starwars_ep3$height, y = starwars_ep3$mass + 2,
7       labels = starwars_ep3$name)
8 legend(x = 200, y = 30, fill = c(as.character(l), 'gray'),
9        legend = c(names(l), 'unknown'))
```



ggplot2

```
9 ggplot(data = starwars_ep3) +
10   aes(x = height, y = mass, label = name, col = gender) +
11   geom_point() +
12   geom_label_repel() +
13   labs(title = 'Physical attributes of Star Wars ep.3',
14        x = 'Height (cm)', y = 'Mass (kg}') +
15   theme(text = element_text(size = 20), legend.position = 'bottom')
```

Physical attributes of Star Wars ep.3



gender a feminine a masculine a NA

How ggplot2 ?

Practice & Tips

Some Ressources



<https://raw.githubusercontent.com/rstudio/cheatsheets/main/pngs/data-visualization.png>

<https://ggplot2.tidyverse.org>

<https://www.r-graph-gallery.com>

<https://www.r-pkg.org/search.html?q=visualization>

Some tips & common issues

- > add interactivity easily : transform ggplot to plotly graph
- > check error message
- > color/fill are often confused
- > R interprets code without regard of line indexes, so make sure you don't have open code before, and that all '+' signs are present

Examples

Learn how to build such beautiful apps, with (mostly) R code.

Radiant Data Design ▾ Basics ▾ Model ▾ Multivariate ▾ Report ▾ ⚙ ▾ ⚡ ▾ ? ▾

Datasets:

diamonds

Add/edit data description
 Rename data

Display:

preview str summary

Load data of type:

rds | rda | rdata

[Browse...](#) No file selected

Save data to type:

rds

[Save](#)

Show R-code

Manage View Visualize Pivot Explore Transform Combine

Data preview

price	carat	clarity	cut	color	depth	table	x	y	z	date
580	0.32	VS1	Ideal	H	61.00	56.00	4.43	4.45	2.71	2012-02-26
650	0.34	SI1	Very Good	G	63.40	57.00	4.45	4.42	2.81	2012-02-26
630	0.30	VS2	Very Good	G	63.10	58.00	4.27	4.23	2.68	2012-02-26
706	0.35	VVS2	Ideal	H	59.20	56.00	4.60	4.65	2.74	2012-02-26
1080	0.40	VS2	Premium	F	62.60	58.00	4.72	4.68	2.94	2012-02-26
3082	0.60	VVS1	Ideal	E	62.50	53.70	5.35	5.43	3.38	2012-02-26
3328	0.88	SI1	Ideal	I	61.70	56.00	6.14	6.18	3.80	2012-02-26
4229	0.93	SI1	Premium	E	61.40	57.00	6.34	6.23	3.86	2012-02-26
1895	0.51	VVS2	Very Good	G	63.40	57.00	5.09	5.06	3.22	2012-02-26
3546	1.01	SI2	Good	E	63.90	58.00	6.31	6.37	4.05	2012-02-26

10 of 3,000 rows shown. See View-tab for details.

Diamond prices

Prices of 3,000 round cut diamonds

The ten most similar players - Pro Evolution Soccer 2019

Graphic About Developers

Apply filters

Choose a player:

C. RONALDO - PM BLACK WHITE

Overall: 50 → 100

Height (cm): 155 → 203

Position:

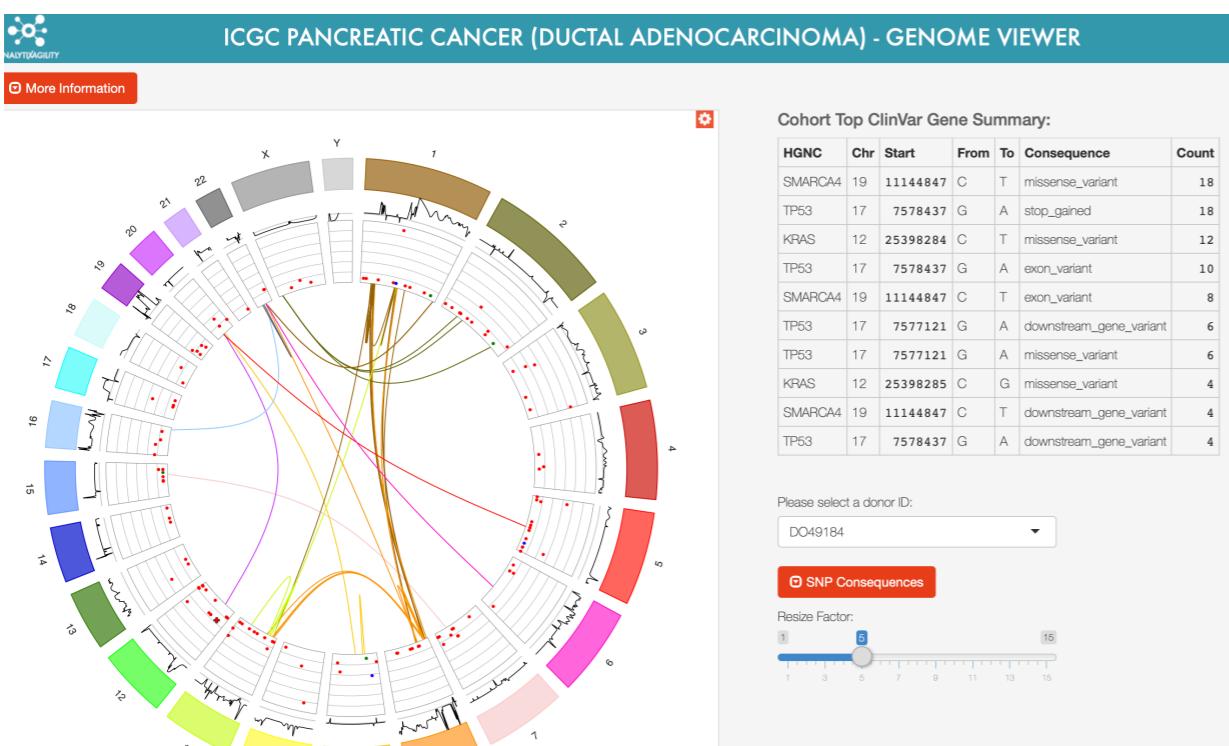
GK CB RB LB DMF CMF AMF RMF LMF RWF LWF SS CF

Foot:

Right foot Left foot

Update filters

A radar chart comparing the similarity of C. Ronaldo to other players across 15 different attributes. The attributes are represented by the axes: DRI, BCO, ATT, EXP, SPE, STM, BAL, COV, REF, HEA, SWE, PKIC, FIN, KPOW, APAS, and LPAS. The chart uses a color gradient from orange (highest similarity) to green (lowest similarity). C. Ronaldo's profile is shown in orange, while other players' profiles are shown in green. The legend lists the top 10 most similar players: C. RONALDO, E. CAVANI, R. LEWANDOWSKI, M. ICARDI, G. HIGUAIN, P. AUBAMEYANG, ROBERTO FIRMINO, R. LUKAKU, A. GRIEZMANN, and L. SUÁREZ.



 Living in the Lego World Demographics Fashion Moods Ecology

Ethnicity and gender Ethnic diversity and gender parity by theme Find sets with a specific ethnicity or gender

Filter to one or more themes:
Star Wars (1258)

Filter to one or more genders:
Nothing selected

Large graphs (e.g., of the full dataset) may take a few seconds to render. The first graph may take up to two minutes if the app is retrieving new data from Rebrickable.

Hover to see the part name.

Each circle represents a unique minifigure or minidoll head.

Area is proportional to the number of pieces across all sets.

"Ethnicity" is the color of the piece. Yes, it's silly.

Gender is inferred from keywords in the part name ("Male", "Female", etc., plus references to facial hair).

Some heads are not labeled male/female but contain the name of a character of known gender (e.g., "Han Solo"). Incorporating this information would require a hand-maintained list of character names and their genders; I haven't done this.



Why ggplot2 ?

R packages - the tidyverse

tidyverse

opinionated collection of R packages designed for data science

```
library(tidyverse)
```

- **ggplot2**: data visualization
- **dplyr**: data wrangling
- **tidyrr**: data tidying
- **readr**: data reading/writing
- **forcats**: working with factors
- **stringr**: working with strings
- **tibble**: modern data frames
- **purrr**: functional programming

```
install.packages(tidyverse)  above + a few more
```

