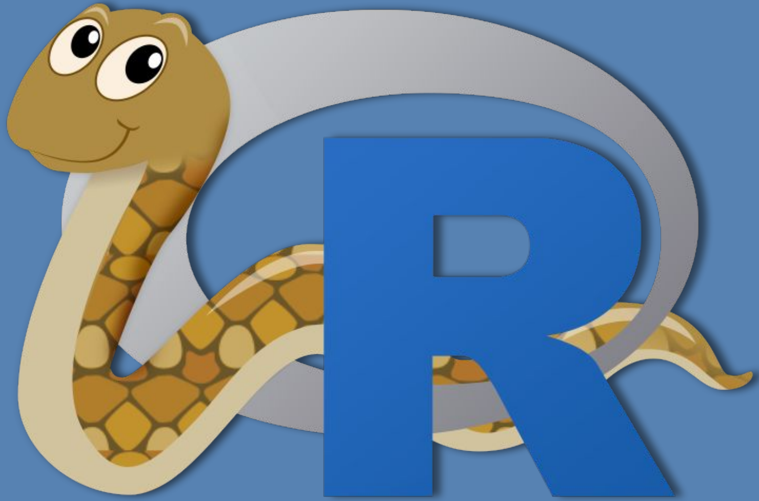
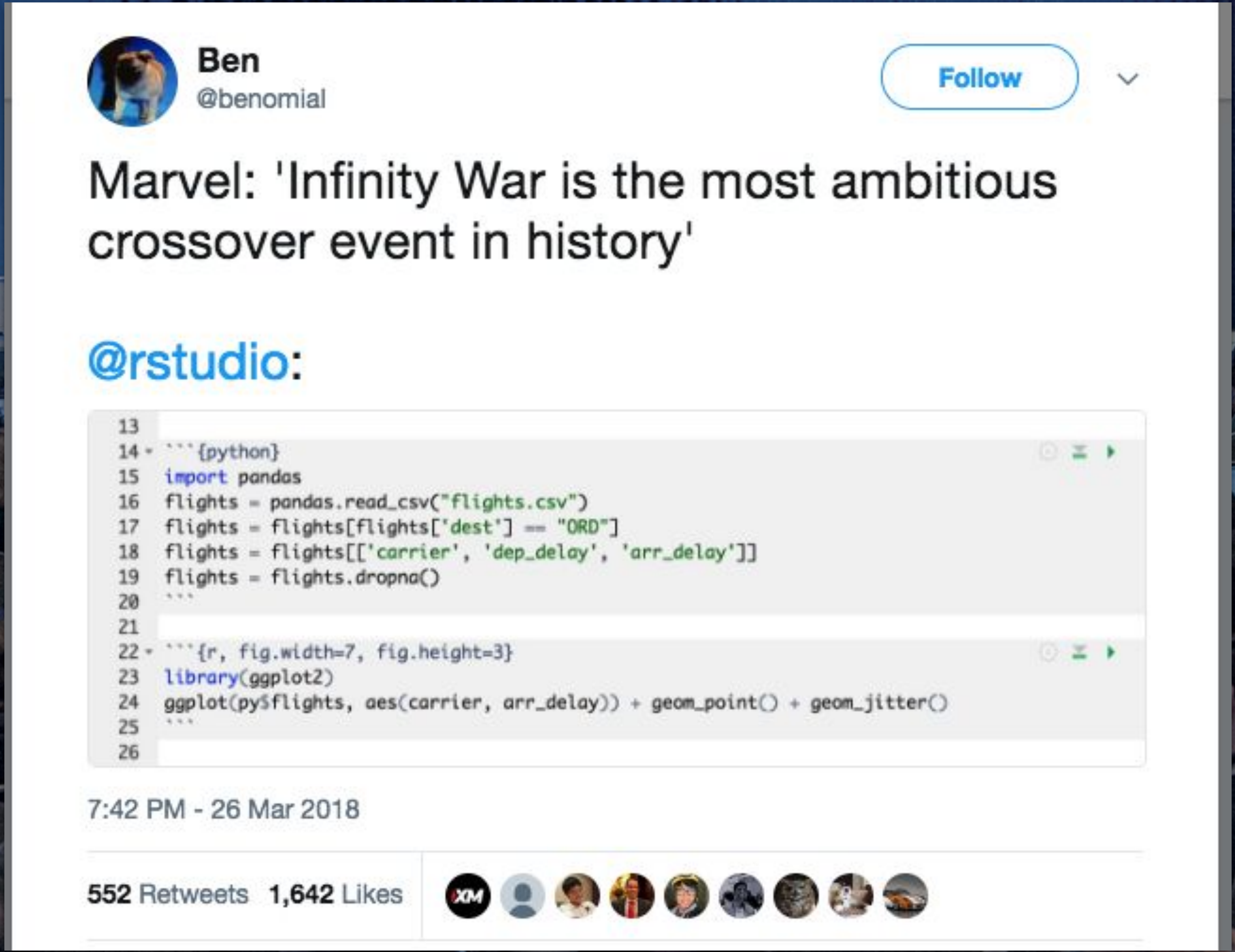
Reticulate

August 22nd, 2018 rstd.io/reticulate



R Studio



Ben
@benomial

Marvel: 'Infinity War is the most ambitious crossover event in history'

@rstudio:

~~~
```{python}
import pandas
flights = pandas.read_csv("flights.csv")
flights = flights[flights['dest'] == "ORD"]
flights = flights[['carrier', 'dep_delay', 'arr_delay']]
flights = flights.dropna()
```

```{r, fig.width=7, fig.height=3}
library(ggplot2)
ggplot(py$flights, aes(carrier, arr_delay)) + geom_point() + geom_jitter()
```
~~~

7:42 PM - 26 Mar 2018

552 Retweets 1,642 Likes

Game Plan

Philosophy

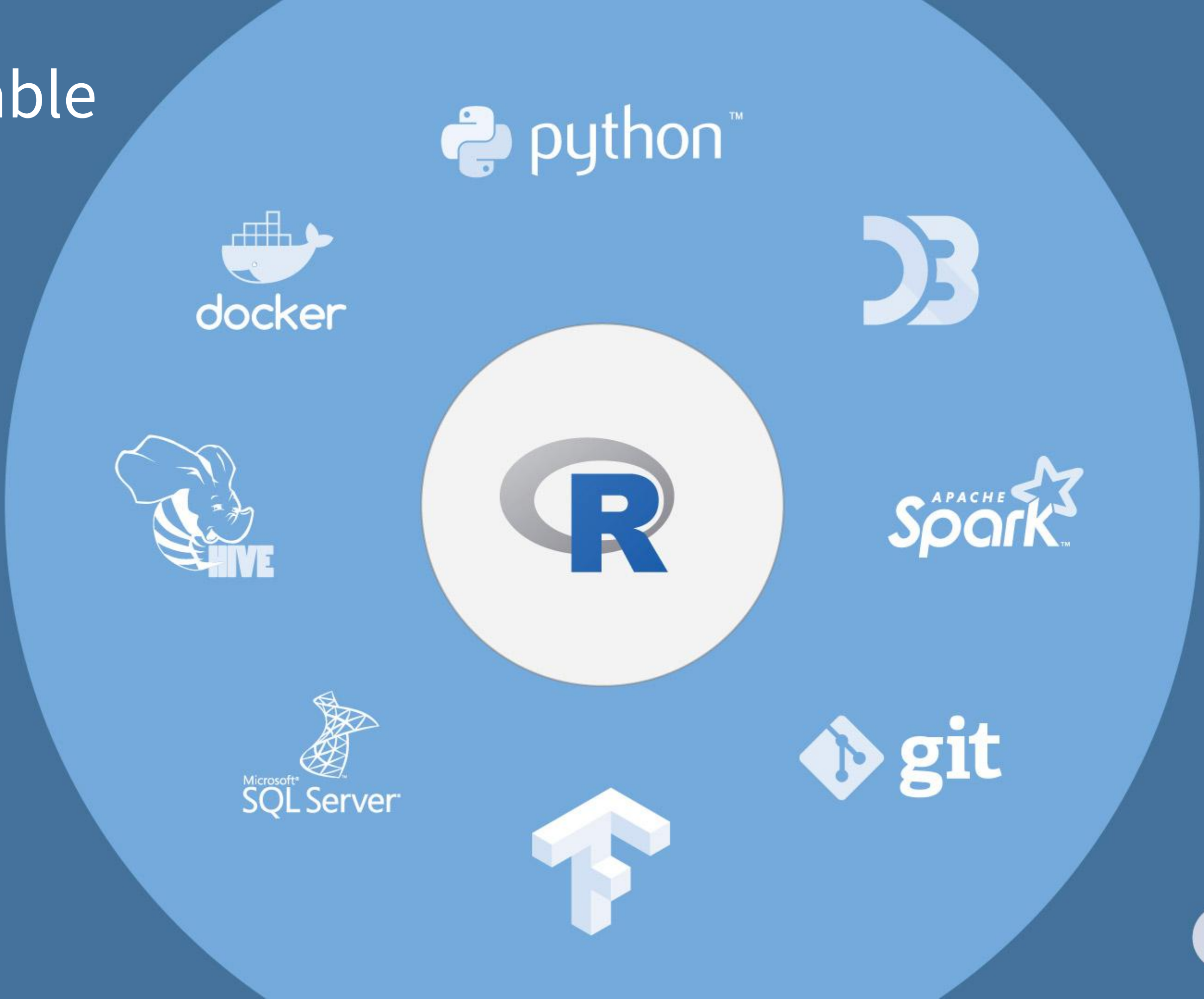
Basics of Reticulate, RStudio, and Python

Case Study: Reticulated Shiny App

Philosophy

~~Python vs R~~

Interoperable



What about **R**Studio?

> A note about the philosophy behind Python tools within RStudio: these tools are not intended for standalone Python work but rather explicitly aimed at the *integration of Python into R projects* (and as such are closely tied to the reticulate package).
>
> There are many IDEs available for doing data science with Python including JupyterLab, Rodeo, Spyder, and Visual Studio Code, and we strongly recommend using one of them for Python-only projects. However, if you are using reticulated Python within an R project then RStudio provides a set of tools that we think you will find extremely helpful.

https://rstudio.github.io/reticulate

RStudio IDE Tools

Recap

Setting the Environment

`use_python`	Specify which python version to use, prefers numpy	Run *before* any other reticulate commands.
`Use_virtualenv`	Activate a virtualenv, try `required = TRUE`	
`use_conda`	Use a conda environment, try `required = TRUE`	
`py_config`	See what you’re actually using	

Inside of R Markdown

`py$`	Access Python objects from a R code chunk.
`r.`	Access R objects from a python code chunk.

Inside of RStudio

`repl_python`	Get a python console!

Conversions (https://rstudio.github.io/reticulate)

Case Study

Reticulated Shiny

Part 1: Drop-In Python Functions

Original Python Simulation Functions

```
def run_simulation(vehicle, map):
  # crazy physics based PDE solver
  return output
```

R function inside app.R

```
source_python( ‘./python_simulation_functions.py’ )

results <- run_simulation(vehicle, map)
```

Reticulated Shiny

Part 1: Drop-In Python Functions

Original Python Simulation Functions

Rectangular Python Dictionary

```
def run_simulation(vehicle, map):
  # crazy physics based PDE solver
  return output
```

Dictionary

R function inside app.R

Data Frame

```
source_python( ‘./python_simulation_functions.py’ )

results <- run_simulation(vehicle, map)
```

List

Reticulated Shiny

Part 2: Writing Reticulated R Wrappers

Google’s Example Python Script

```
import googlemaps
import os
from datetime import datetime

gmaps = googlemaps.Client(
   key = os.environ[“API_KEY”]
)

def get_route(start, end):
   directions = gmaps.directions(
     start,
     end,
     mode = “driving”,
     departure_time = datetime.now()
   )
   return directions
```

R “Wrapper” Function

```
googlemaps <- import(‘googlemaps’)


gmaps <- googlemaps$Client(
  key = Sys.getenv(“API_KEY”)
)

get_route <- function(start, end){
  route <- gmaps$directions(
     start,
     end,
     mode = “driving”,
     departure_time = Sys.time()

   return(route)
}
```

Questions