Week 9 Research: R, Redis, Plotly

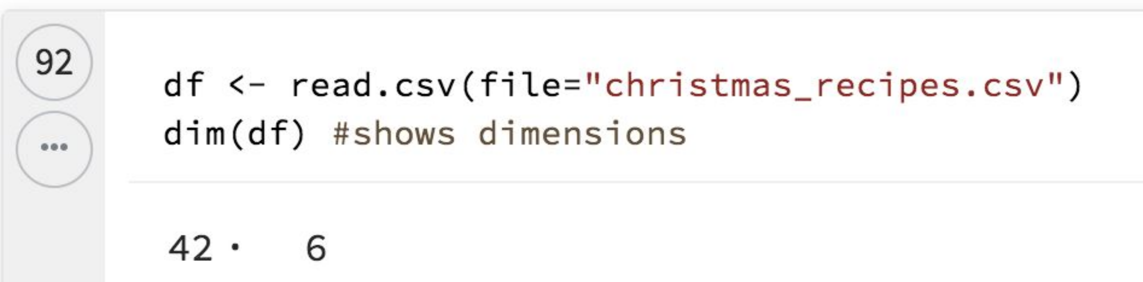
Deana Stuart
Zip Code Wilmington, Data 1.2

R

- Developed from the S programming language and published in 1993
- Widely used by data scientists, statisticians, and data analysts
- Over 10,000 packages available
- R has 6 basic data types:
  - character: "a", "swc"
  - numeric (real or decimal): 2, 15.5
  - integer: 2L (the L tells R to store this as an integer)
  - logical: TRUE, FALSE
  - complex: 1+4i (complex numbers with real and imaginary parts
- Main data structures include vectors, lists and data frames

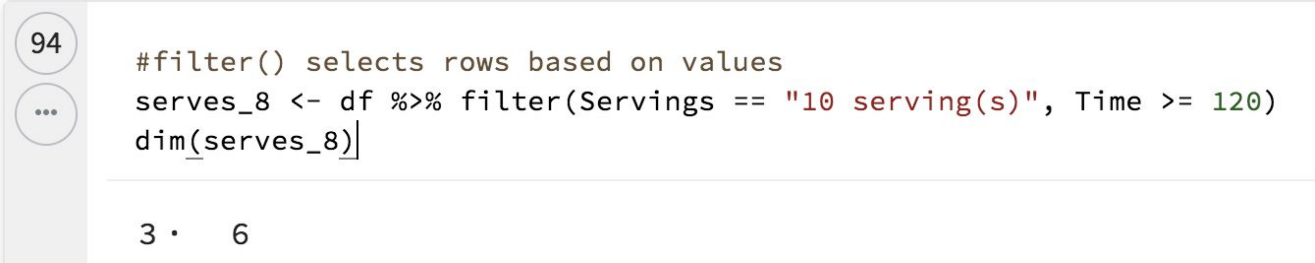
R

- Data frames are the most common structures for manipulating data. The many libraries in R make it easy to work with data. In a few lines of code, I can load a csv into a dataframe and filter it.



```
df <- read.csv(file="christmas_recipes.csv")
dim(df) #shows dimensions
```

```
42 · 6
```



```
#filter() selects rows based on values
serves_8 <- df %>% filter(Servings == "10 serving(s)", Time >= 120)
dim(serves_8)
```

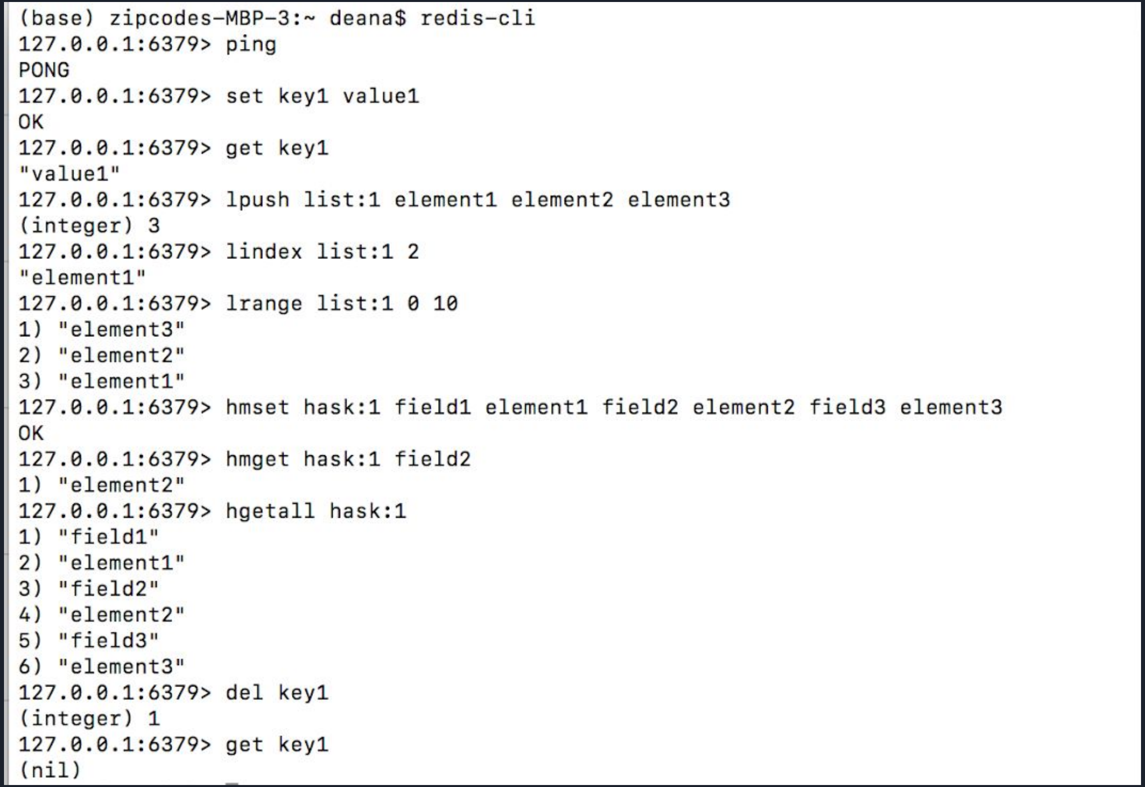
```
3 · 6
```

Redis

- NoSQL Database that used key/value storage format
- Stores data in memory for fast retrieval
  - Can perform 110,000 SETs and 81,000 GETs per second
- Supports many data structures:
  - Strings, Lists, Sets, Sorted sets, Hashes, Bit Arrays, Streams, HyperLogLogs
- Over 200 program languages are supported
- Common commands include SET, GET, DEL, APPEND, BITCOUNT, HMSET, HMGET, LPOP, LSET, LLEN

Redis

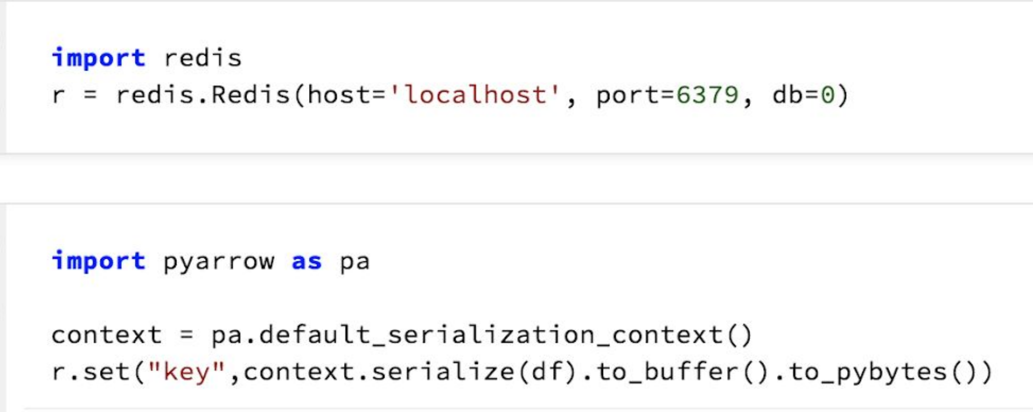
Basic CRUD commands with Redis CLI



```
(base) zipcodes-MBP-3:~ deana$ redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> set key1 value1
OK
127.0.0.1:6379> get key1
"value1"
127.0.0.1:6379> lpush list:1 element1 element2 element3
(integer) 3
127.0.0.1:6379> lindex list:1 2
"element1"
127.0.0.1:6379> lrange list:1 0 10
1) "element3"
2) "element2"
3) "element1"
127.0.0.1:6379> hmset hask:1 field1 element1 field2 element2 field3 element3
OK
127.0.0.1:6379> hmget hask:1 field2
1) "element2"
127.0.0.1:6379> hgetall hask:1
1) "field1"
2) "element1"
3) "field2"
4) "element2"
5) "field3"
6) "element3"
127.0.0.1:6379> del key1
(integer) 1
127.0.0.1:6379> get key1
(nil)
```

Redis

Adding Data Frames to Redis, serialized as a whole piece:

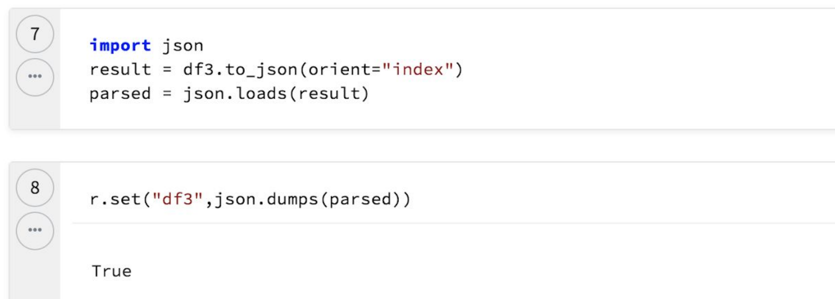


```
import redis
r = redis.Redis(host='localhost', port=6379, db=0)
```

```
import pyarrow as pa

context = pa.default_serialization_context()
r.set("key",context.serialize(df).to_buffer().to_pybytes())
```

As Json data:



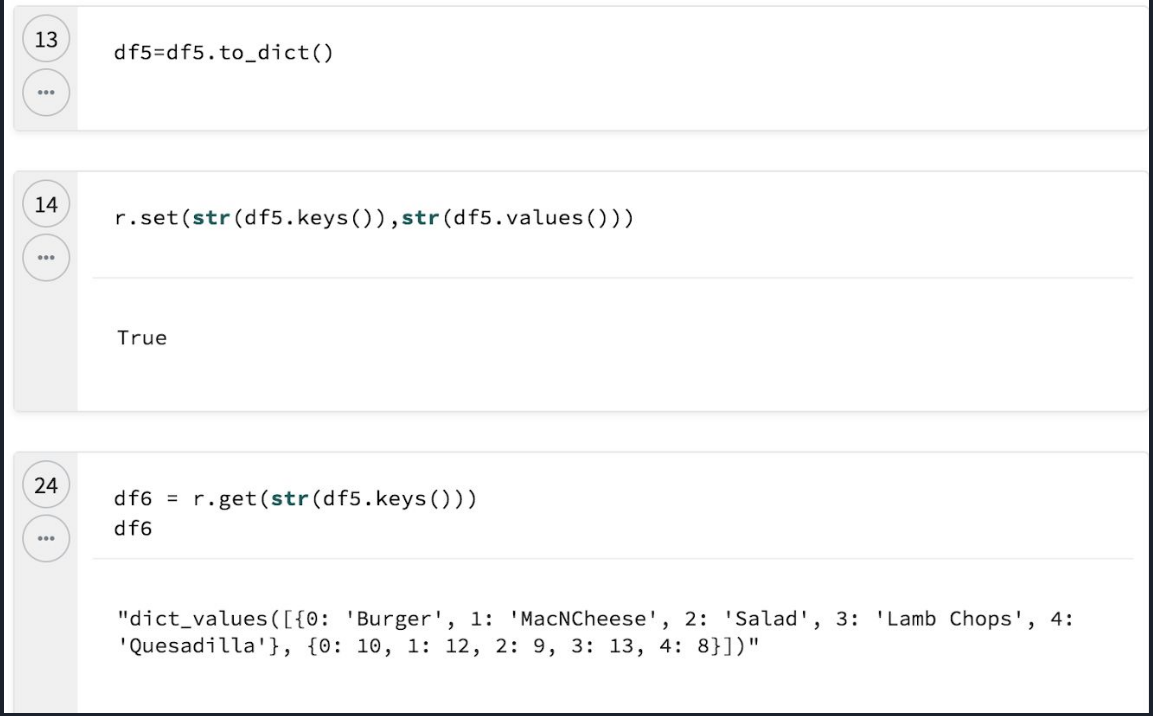
```
import json
result = df3.to_json(orient="index")
parsed = json.loads(result)
```

```
r.set("df3",json.dumps(parsed))
```

```
True
```

Redis

Or as string of a dictionary:



```
df5=df5.to_dict()
```

```
r.set(str(df5.keys()),str(df5.values()))
```

```
True
```

```
df6 = r.get(str(df5.keys()))
df6
```

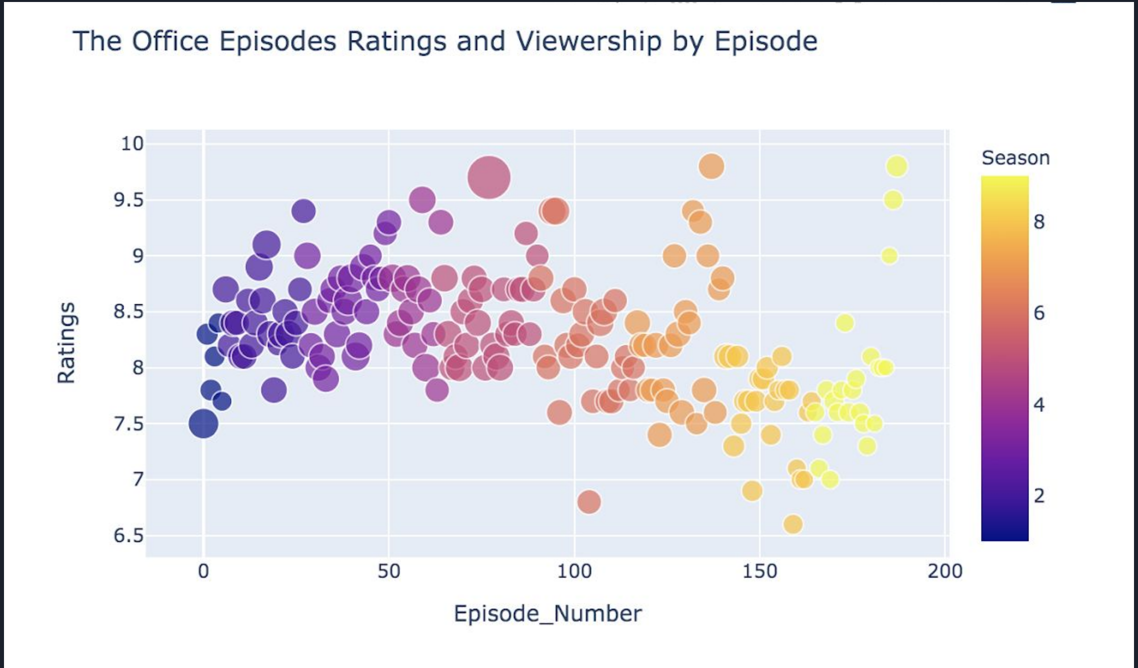
```
"dict_values([{0: 'Burger', 1: 'MacNCheese', 2: 'Salad', 3: 'Lamb Chops', 4:
'Quesadilla'}, {0: 10, 1: 12, 2: 9, 3: 13, 4: 8}])"
```

Plotly

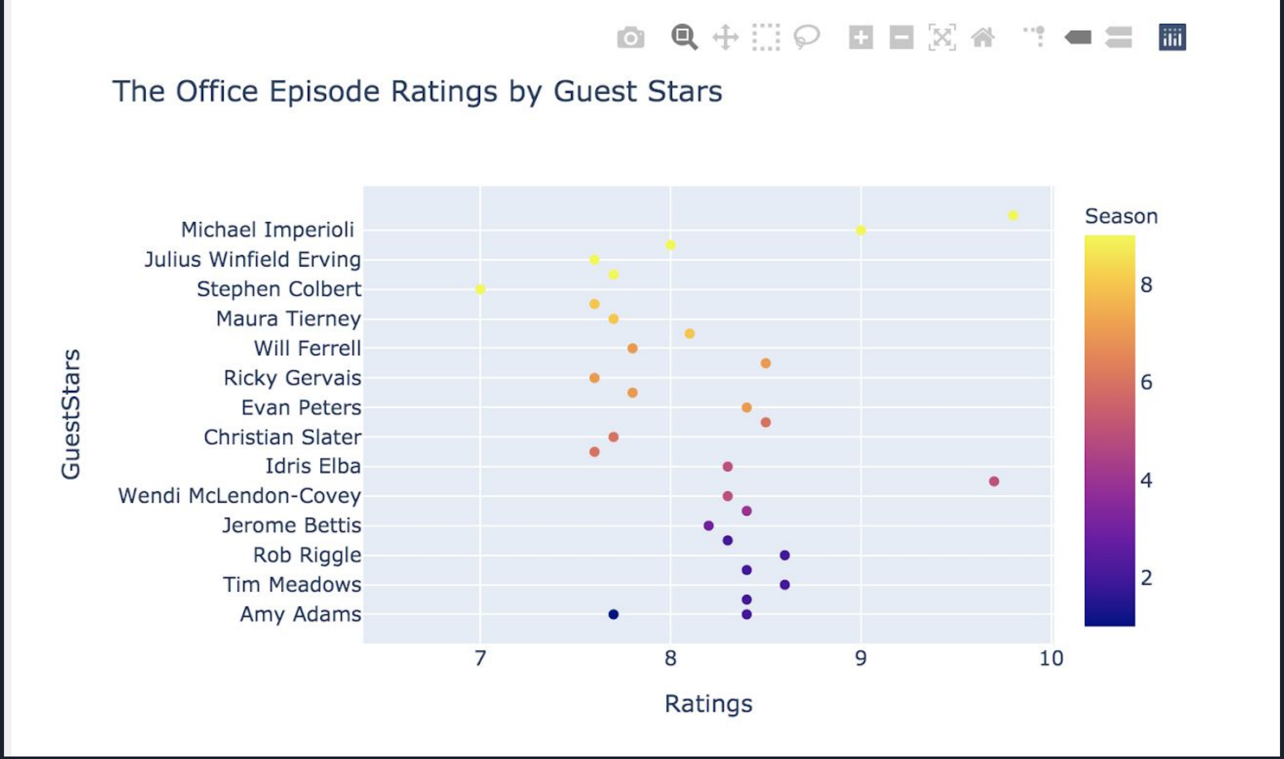
- Canadian company that provides open source graphing libraries in Python, R, and Javascript, as well as Dash, a tool for building web analytic applications which are rendered in web browsers
- Provides over 40 different kinds of publication quality charts:
    - Financial, Statistic, and Scientific
    - Mapping
    - 3 Dimensional
    - Jupyter Interactive Widgets
    - AI and Machine Learning

Plotly

Visualizations of The Office Dataset:

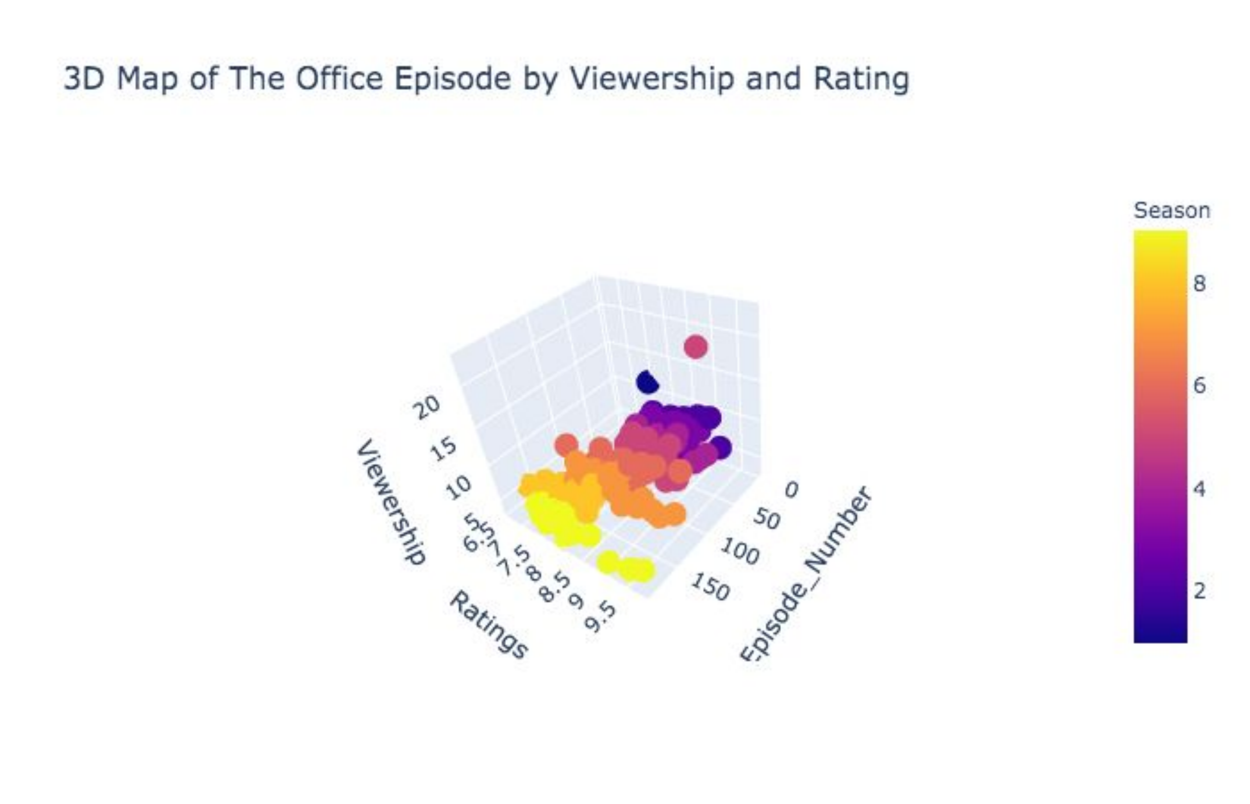


Plotly



The Office Episode Ratings by Guest Stars
GuestStars
Michael Imperioli
Julius Winfield Erving
Stephen Colbert
Maura Tierney
Will Ferrell
Ricky Gervais
Evan Peters
Christian Slater
Idris Elba
Wendi McLendon-Covey
Jerome Bettis
Rob Riggle
Tim Meadows
Amy Adams
7
8
9
10
Ratings
Season
8
6
4
2

Plotly


3D Map of The Office Episode by Viewership and Rating
Viewership
20
15
10
5
Ratings
6.5
7
7.5
8
8.5
9
9.5
Episode_Number
0
50
100
150
Season
8
6
4
2