Data Visualization

2017-01-26

Agenda

- Short presentations
- Interactive applications
  - Event loops
  - Examples of increasing complexity in Python
  - Model-View-Controller framework

Presentations

Interactive Applications

Senate Data from 2008

Available online

Name	*Motion 1*	*Motion 2*	*Motion 3*	*Motion 4*	...
Alexander	Yea	Yea	Yea	Nay	
Biden	Yea	Nay	Nay	Yea	
Bond	Yea	Abstain	Nay	Nay	
Boxer	Yea	Yea	Yea	Abstain	
...					...

Senate Data from 2008

Available online

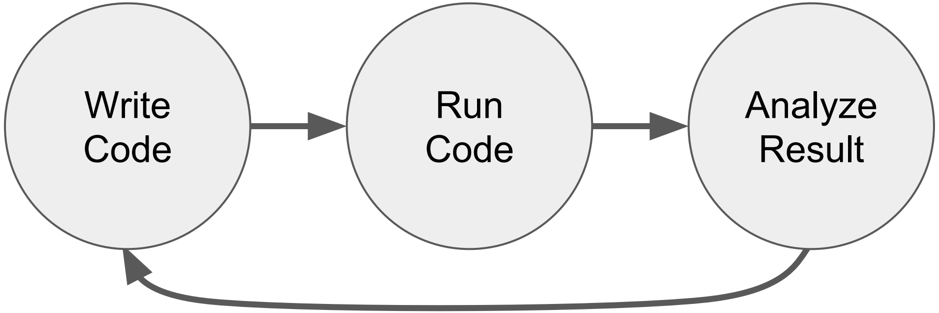
Name	*Motion 1*	*Motion 2*	*Motion 3*	*Motion 4*	...
Alexander	1	1	1	-1	
Biden	1	-1	-1	1	
Bond	1	0	-1	-1	
Boxer	1	1	1	0	
...					...

Back to the Notebook for a Moment

So, who was that outlier?

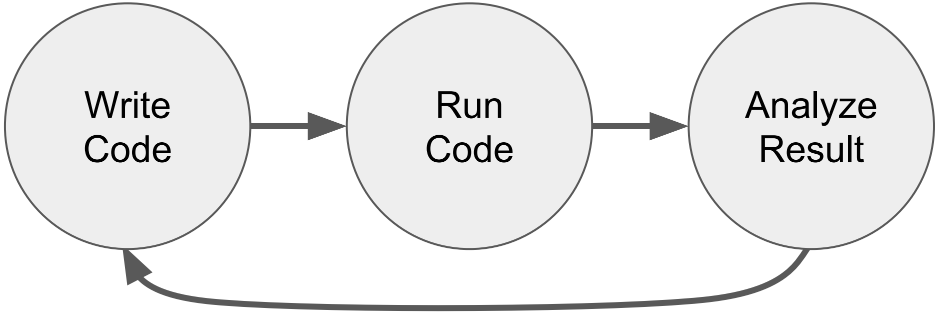
Interactive Applications

Data Scientist's Workflow:

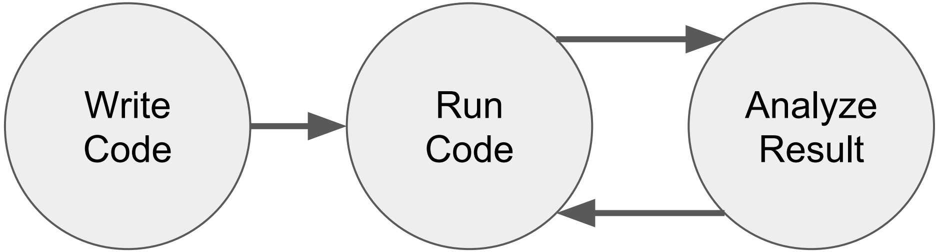


Interactive Applications

Data Scientist’s Workflow:



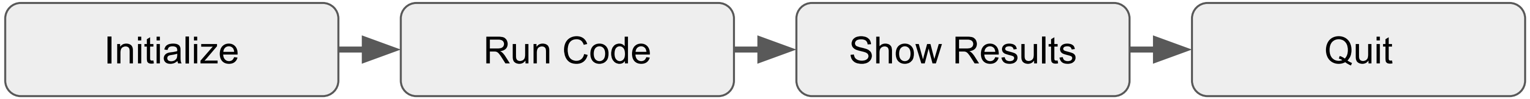
More Efficient Data Scientist’s Workflow:



A non-Notebook example

Event Loops

Non-interactive programs



Initialize

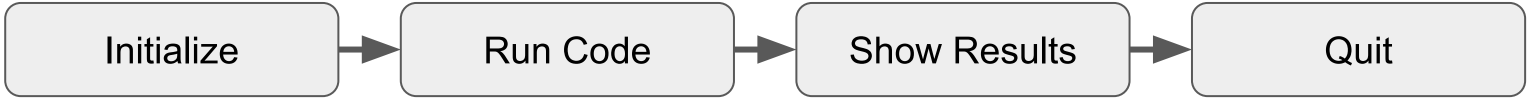
Run Code

Show Results

Quit

Event Loops

Non-interactive programs



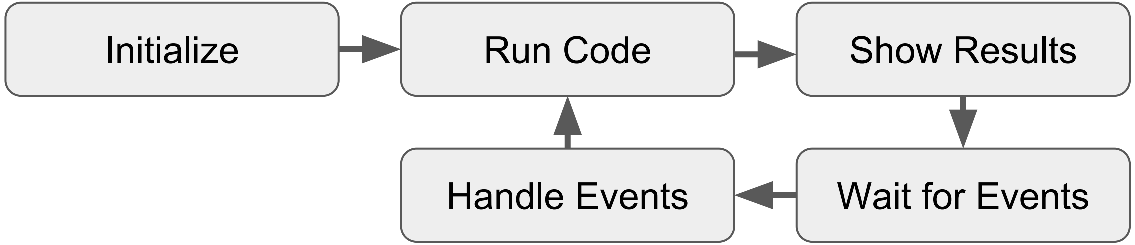
Initialize

Run Code

Show Results

Quit

Interactive programs:



Initialize

Run Code

Show Results

Wait for Events

Handle Events

So … who WAS that outlier?

Somewhat Generalizing [model-view-controller]

http://www.essenceandartifact.com/2012/12/the-essence-of-mvc.html

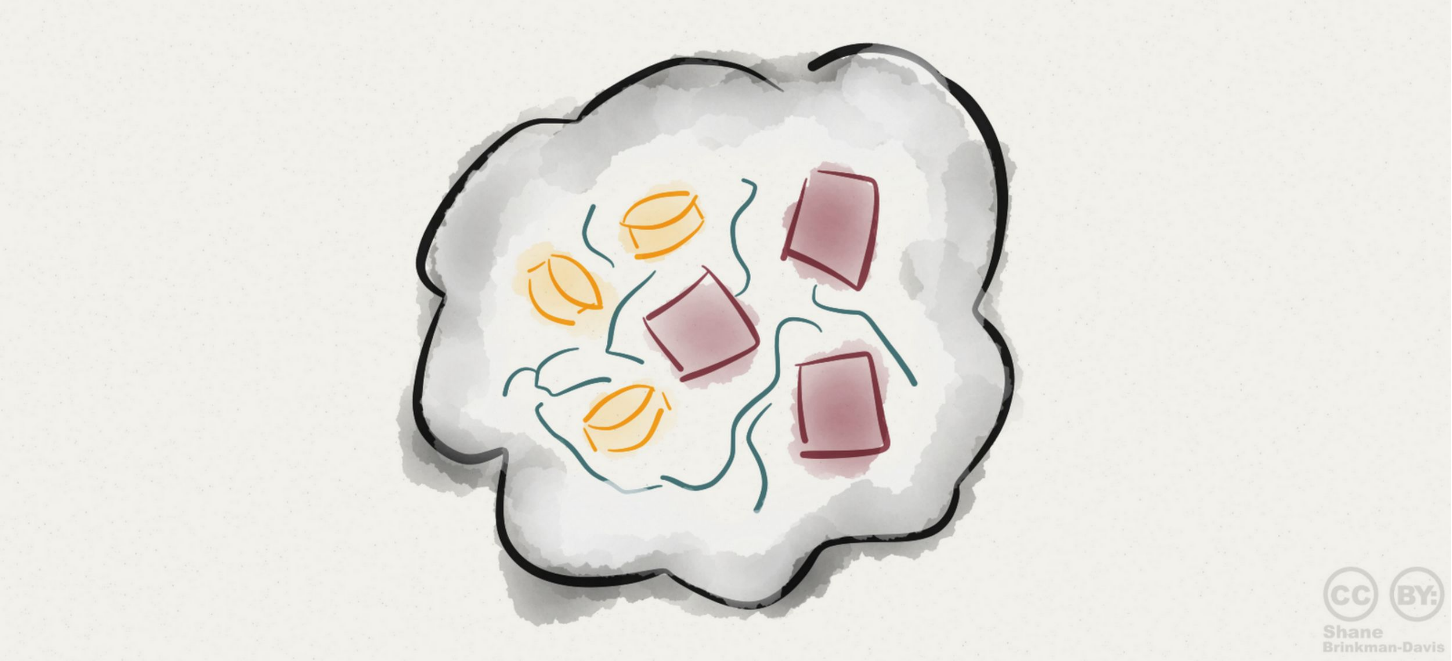
That can get messy...

Code that handles

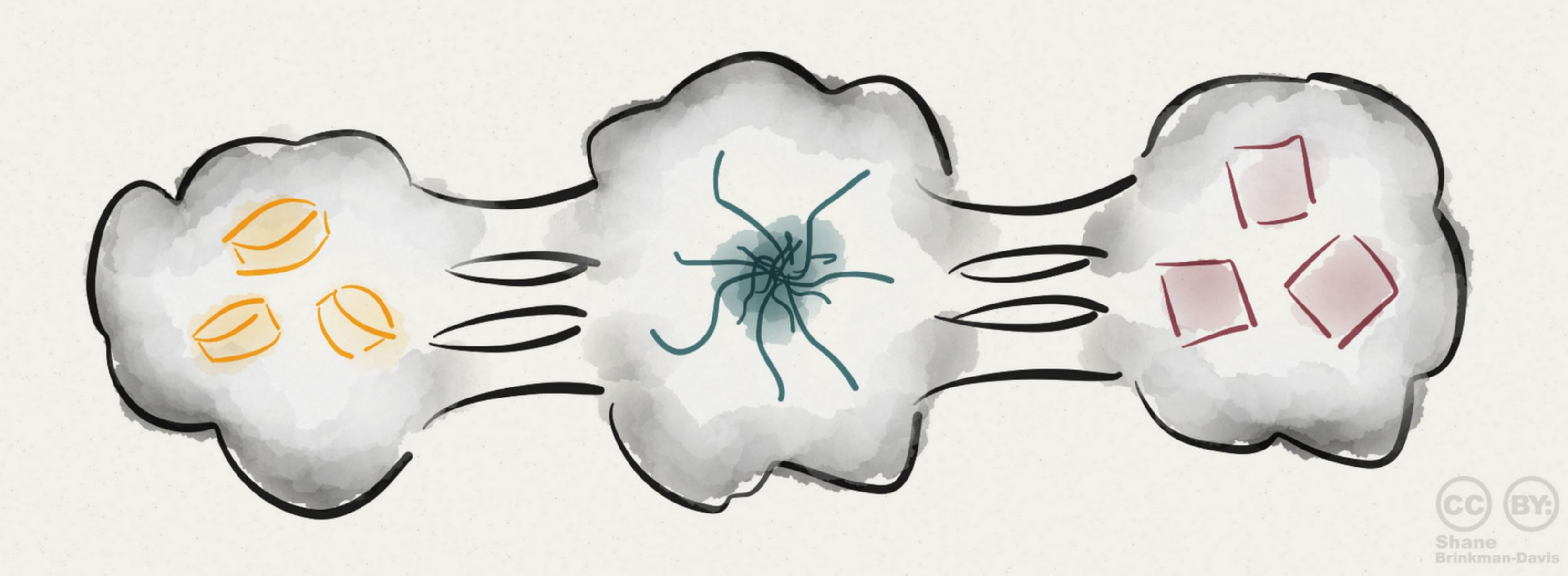
- Data modification
- Handling input events
- Displaying to the user, drawing
- Animating

All living together! Not so happily!

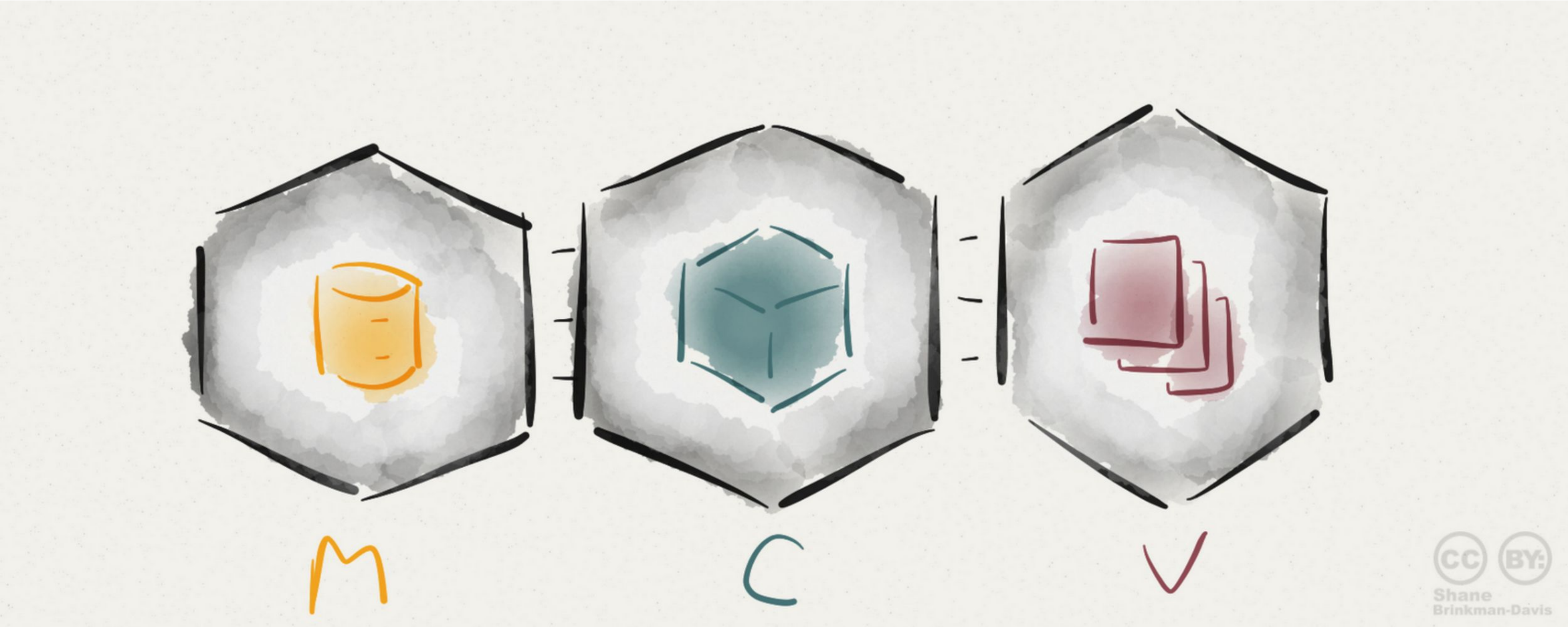
Model-View-Controller



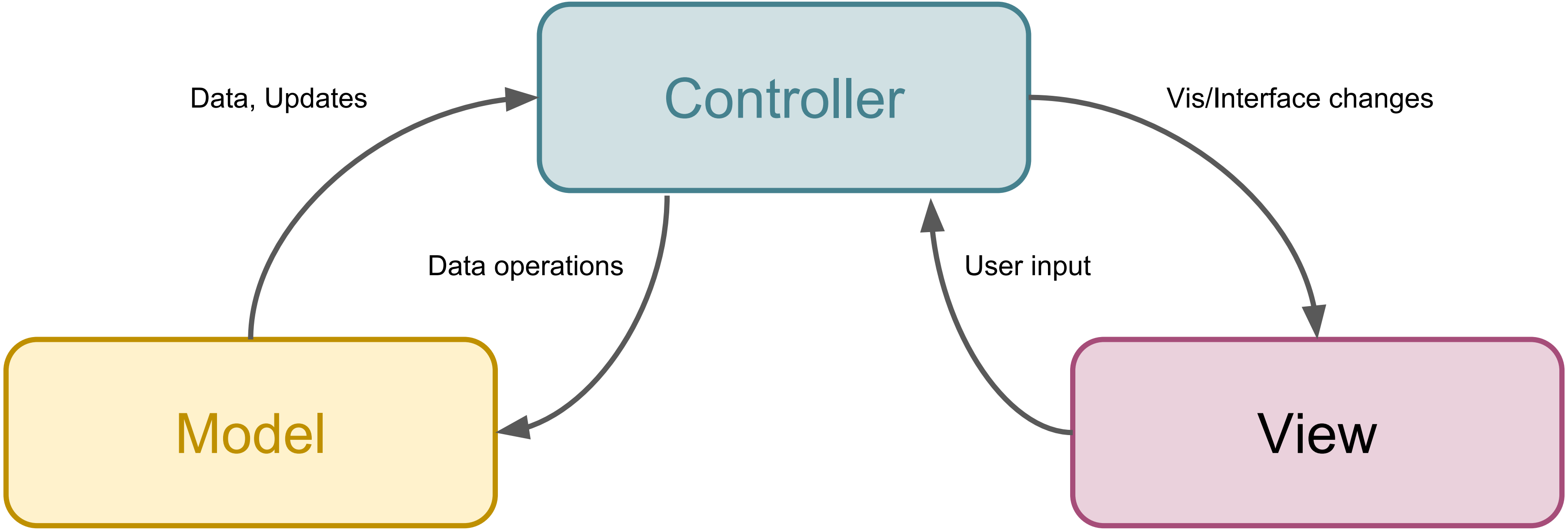
Model-View-Controller



Model-View-Controller



Model-View-Controller



Model-View-Controller

Model:

- store and modify data

View

- present visualization of data to the user, present user interface

Controller

- respond to user inputs and events, update the view and model

Take-Aways

- Interactivity
  - Rapid evaluation can save time
  - Use MVC structure to keep code simple