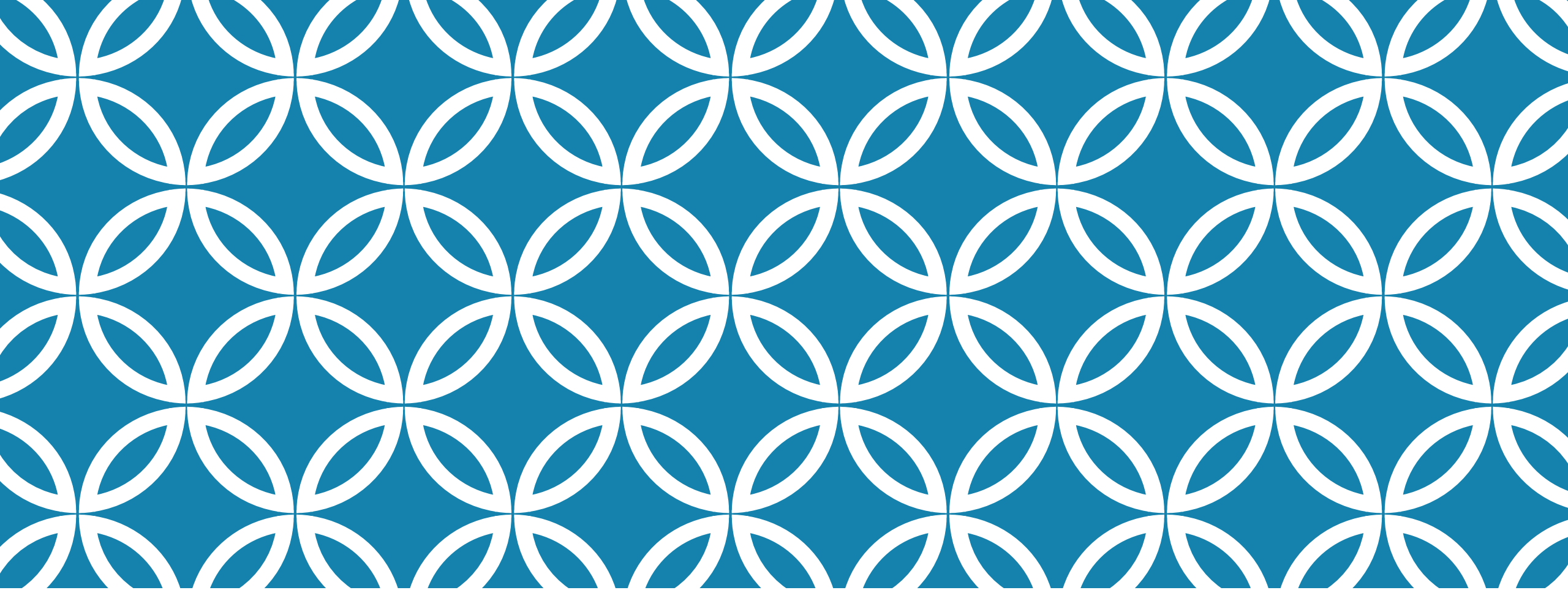


HCDE 530 Vis Lecture: Building Interactive Web Visualization in Python via Plotly/Dash

Nan-Chen Chen

Feb 8, 2018

What we have learned in this class

Basic Python Syntax & Usages

Ways to collect & store tweets

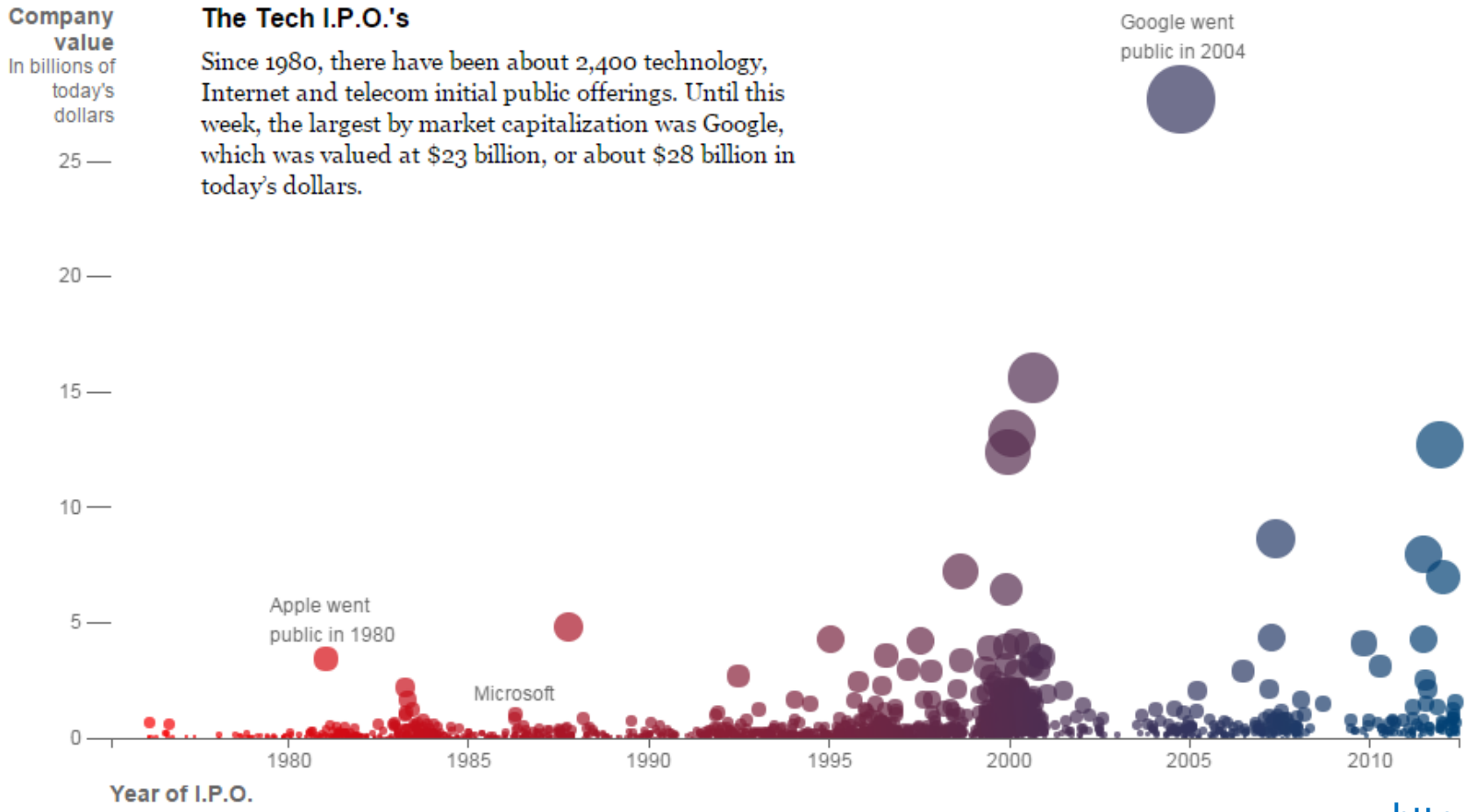
What we will learn in the two VIS lectures

Basic Python Syntax & Usages

Ways to collect & store tweets

Visualize & share analysis results

An example:
The Tech I.P.O. Vis by New York Times



http://tinyurl.com/nytimes-tech-ipo

Some common tools for building visualization

Some common tools for building visualization

Plotly & Dash

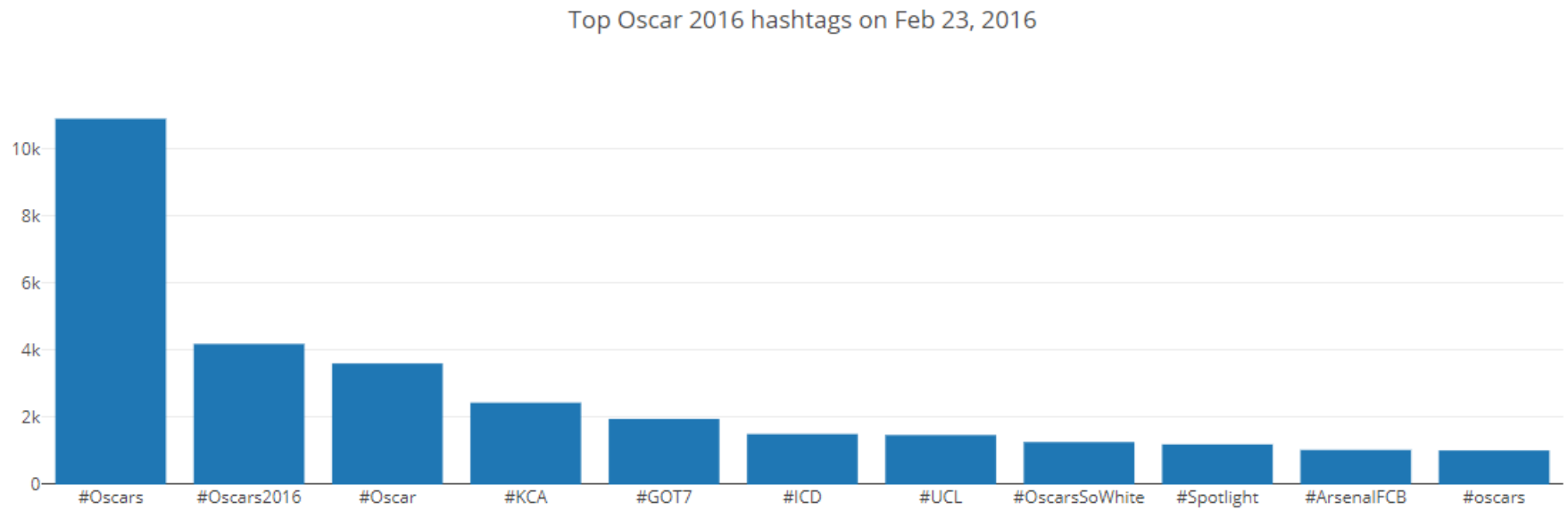
https://plot.ly/python/getting-started/

https://plot.ly/dash/

A simple demo

Hello Dash for HCDE 530

This is a simple Dash application for HCDE 530



Outline of today's lecture

Basics (40 mins)

- Installing packages & running simple barchart codes (15 mins)
- Barchart code explanations (10 mins)
- Exercises with other datasets (or your own data) (15 mins)

Break (5 mins)

Basic Visualization Concepts with Real Examples (25 mins)

More Examples about Other Chart Types (20 mins)

Wrap Up

Package Installation

```
pip install dash==0.20.0
pip install dash-renderer==0.11.3
pip install dash-html-components==0.8.0
pip install dash-core-components==0.18.1
pip install plotly --upgrade
```

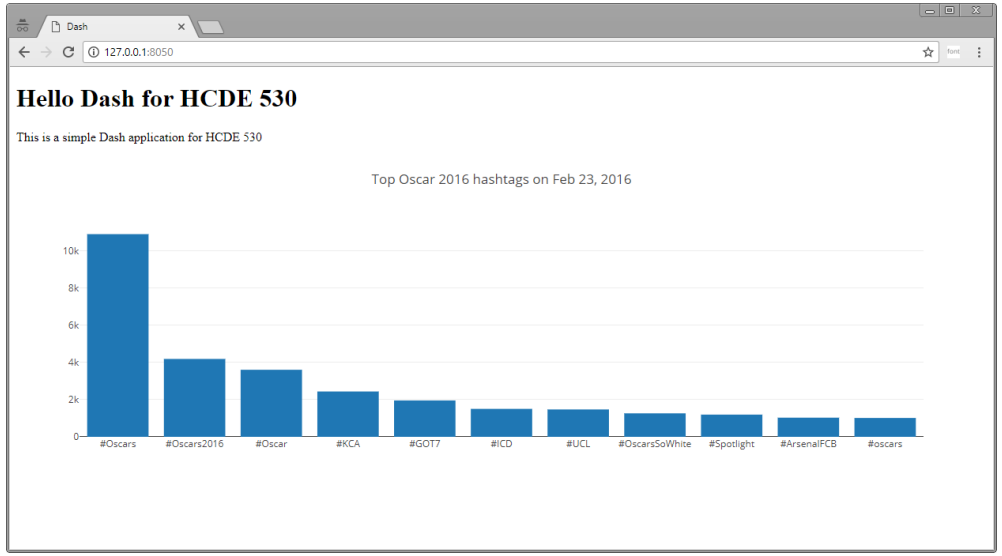
Run the barchart example

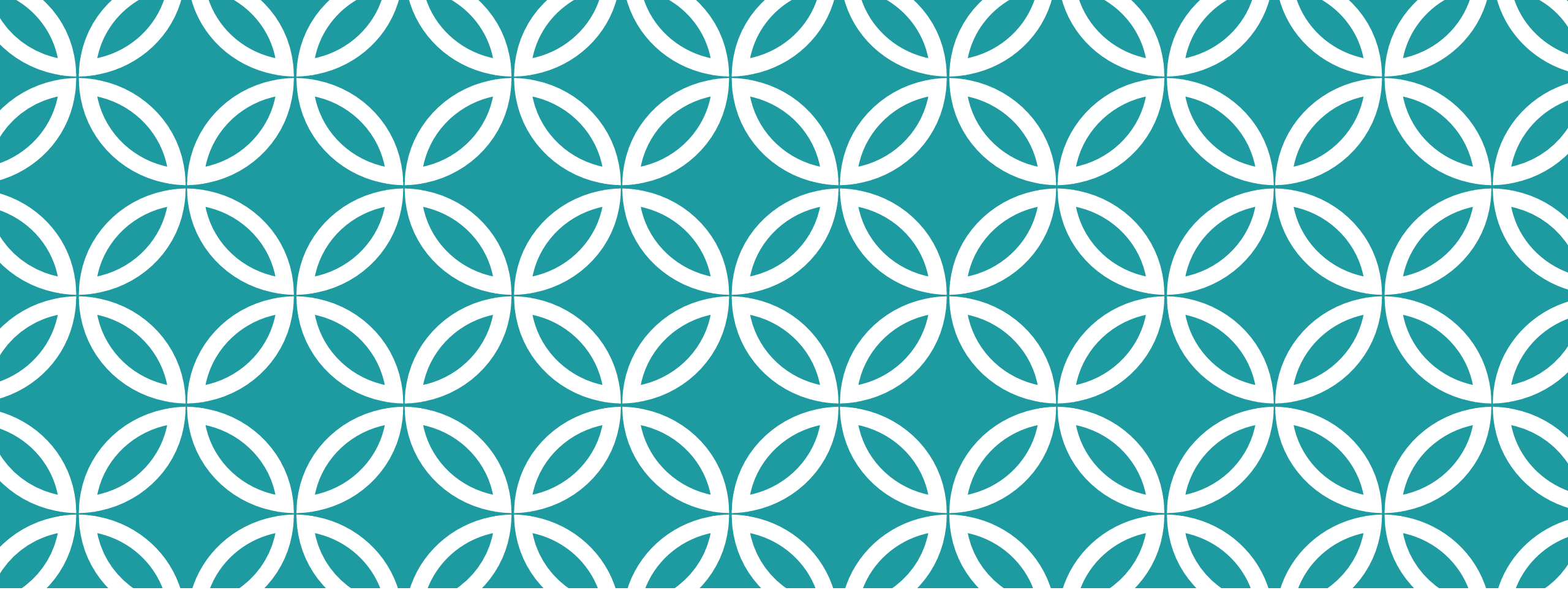
Download the files from Canvas, extract them into a folder.

Open up your terminal from the folder and run:

```
python 01_dash-barchart-demo.py
```

Then open up your browser and go to
http://127.0.0.1:8050
to see if you can see the page like the right screenshot





Barchart code Explanations

1. Import modules

2. Data

3. Initialize Dash environment

4. Set the layout of the page

4.1. Set H1 title and text description

4.2. Set the barchart

5. Start the Dash app

```python
# -*- coding: utf-8 -*-
import dash
import dash_core_components as dcc
import dash_html_components as html

# static data
hashtags_in_order = ['#Oscars', '#Oscars2016', '#Oscar', '#KCA', '#GOT7', '#ICD', '#UCL',
counts_in_order = [10886, 4184, 3602, 2435, 1953, 1501, 1471, 1262, 1194, 1029, 1014]

# initialize Dash environment
app = dash.Dash()

# set up an layout
app.layout = html.Div(children=[
    # H1 title on the page
    html.H1(children='Hello Dash for HCDE 530'),

    # a div to put a short description
    html.Div(children='''
        This is a simple Dash application for HCDE 530
    '''),

    # append the visualization to the page
    dcc.Graph(
        id='example-graph',
        figure={
            # configure the data
            'data': [
                # set x to be hashtags, and y to be the counts. We use bars to represent
                {'x': hashtags_in_order, 'y': counts_in_order, 'type': 'bar'},
            ],
            # configure the layout of the visualization -- set the title to be Oscar 2016
            'layout': {
                'title': 'Top Oscar 2016 hashtags on Feb 23, 2016'
            }
        }
    )
])

if __name__ == '__main__':
    # start the Dash app
    app.run_server(debug=True)
```

1. Import modules

2. Data

3. Initialize Dash environment

4. Set the layout of the page

4.1. Set H1 t

4.2. Set the barchart

5. Start the Dash app

```
# -*- coding: utf-8 -*-
import dash
import dash_core_components as dcc
import dash_html_components as html
```

These are modules from Dash that we will use in our demo example

1. Import modules

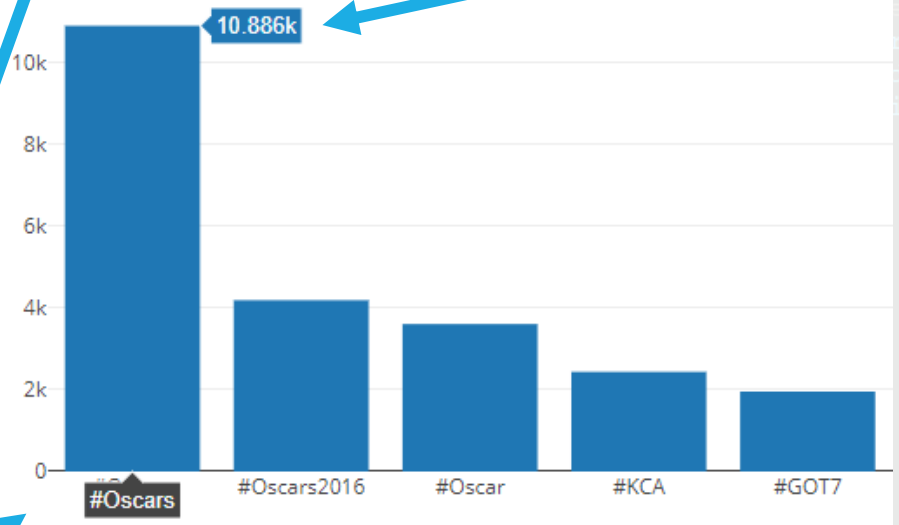
2. Data

3. Initialize Dash environment

4. Set the layout of the page

5. Start the Dash app

```
6  # static data
7  hashtags_in_order = ['#Oscars', '#Oscars2016', '#Oscar', '#KCA'
8  counts_in_order = [10886, 4184, 3602, 2435, 1953, 1501, 1471, 1
```



Data can come from anywhere –
computation, files, user inputs, streaming APIs.
In this case, we use hard-coded strings and numbers.
But as long as we have data, we can use them to create visualization.

1. Import modules

2. Data

3. Initialize Dash environment

4. Set the layout of the page

4.1. Set H1 title and text description

4.2. Set the barchart

5. Start the Dash app

```
10  # initialize Dash environment
11  app = dash.Dash()
```

By Line 11, app is now an instance of Dash's environment that we can manipulate.

1. Import modules

2. Data

3. Initialize Dash environment

4. Set the layout of the page

4.1. Set H1 title and text

4.2. Se

5. Start the Dash app

```
# set up an layout
app.layout = html.Div(children=[
    # H1 title on the page
    html.H1(children='Hello Dash for HCDE 530'),

    # a div to put a short description
    html.Div(children='''
        This is a simple Dash application for HCDE 530
    '''),

    # append the visualization to the page
    dcc.Graph(
        id='example-graph',
        figure={
            # configure the data
            'data': [
                # set x to be hashtags, and y to be the counts. We use bars to represent our data.
                {'x': hashtags_in_order, 'y': counts_in_order, 'type': 'bar'},
            ],
            # configure the layout of the visualization -- set the title to be Oscar 2016 hashtags
            'layout': {
                'title': 'Top Oscar 2016 hashtags on Feb 23, 2016'
            }
        }
    )
])
```

From Line 14 to Line 38, we are setting the layout of the page.

1. Import modules

2. Data

3. Initialize Dash environment

4. Set the layout of the page

4.1. Set H1 title and text description

4.2. Set the barchart

5. Start the Dash app


```
    html.H1(children='Hello Dash for HCDE 530'),

    # a div to put a short description
    html.Div(children='''
        This is a simple Dash application for HCDE 530
    '''),
```

1. Import modules

3. Initialize Dash environment

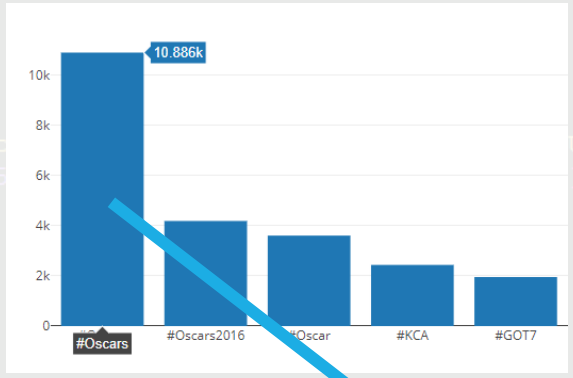
4. Set the layout of the page

description

4.2. Set the barchart

5. Start the Dash app

```
6  # static data
7  hashtags_in_order = ['#Oscars', '#Oscars2016', '#Oscar', '#KCA', '#GO
8  counts_in_order = [10886, 4184, 3602, 2435, 1953, 1501, 1471, 1262, 1
```




```
30                  {'x': hashtags_in_order, 'y': counts_in_order, 'type': 'bar'},
```

```
    # append the visualization to the page
    dcc.Graph(
        id='example-graph',
        figure={
            # configure the data
            'data': [
                # set x to be hashtags, and y to be the counts. We use bars to represent
                {'x': hashtags_in_order, 'y': counts_in_order, 'type': 'bar'},
            ],
            # configure the layout of the visualization -- set the title to be Oscar 2016
            'layout': {
                'title': 'Top Oscar 2016 hashtags on Feb 23, 2016'
            }
        }
    )
])
```

Top Oscar 2016 hashtags on Feb 23, 2016

1. Import modules

2. Data

3. Initialize Dash environment

4. Set the layout of the page

4.1. Set H1 title and text description

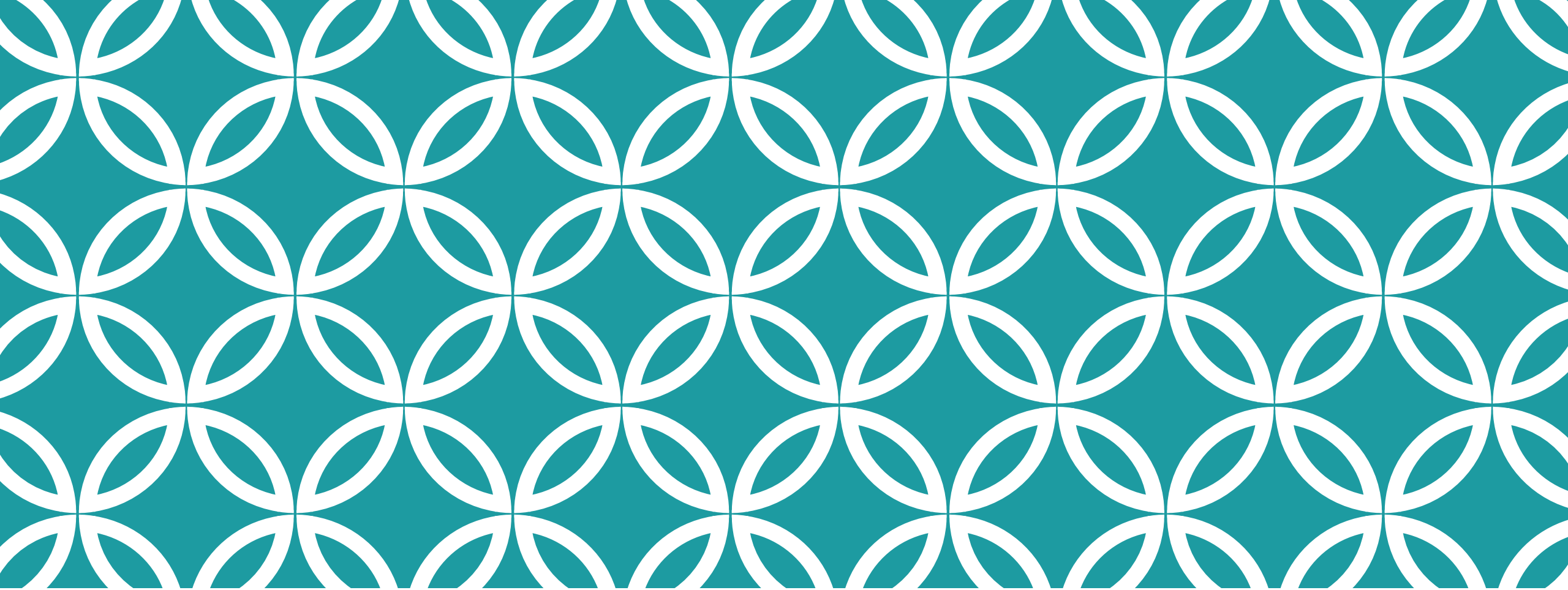
4.2. Set the barchart

5. Start the Dash app

```
# -*- coding: utf-8 -*-
import dash
import dash_core_components as dcc
import dash_html_components as html

# static data
hashtags_in_order = ['#Oscars', '#Oscars2016', '#Oscar', '#KCA', '#GOT7', '#ICD', '#UCL',
counts_in_order = [10886, 4184, 3602, 2435, 1953, 1501, 1471, 1262, 1194, 1029, 1014]

# initialize Dash environment
app = dash.Dash()

# set up an layout
app.layout = html.Div(children=[
    # H1 title on the page
    html.H1(children='Hello Dash for HCDE 530'),

    # a div to put a short description
    html.Div(children='''
        This is a simple Dash application for HCDE 530
    '''),

    # append the visualization to the page
    dcc.Graph(
        id='example-graph',
        figure={
            # configure the data
            'data': [
                # set x to be hashtags, and y to be the counts. We use bars to represent
                {'x': hashtags_in_order, 'y': counts_in_order, 'type': 'bar'},
            ],
            # configure the layout of the visualization -- set the title to be Oscar 2016
            'layout': {
                'title': 'Top Oscar 2016 hashtags on Feb 23, 2016'
            }
        }
    )
])

if __name__ == '__main__':
    # start the Dash app
    app.run_server(debug=True)
```

Exercise:
Load data from a json file and create a barchart

See instructions in 02_dash-barchart-exercise.py

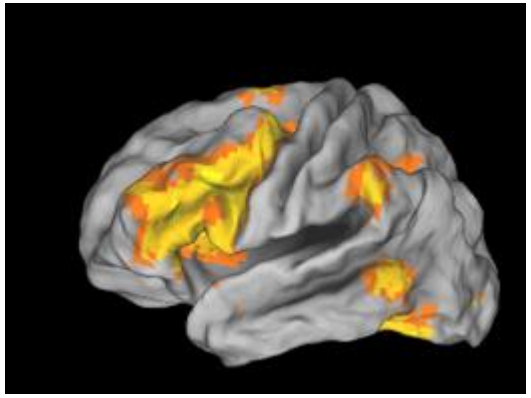


Basic Visualization Concepts with Real Examples

Essence of Visualization: Using visual attributes to represent data

Scientific visualization

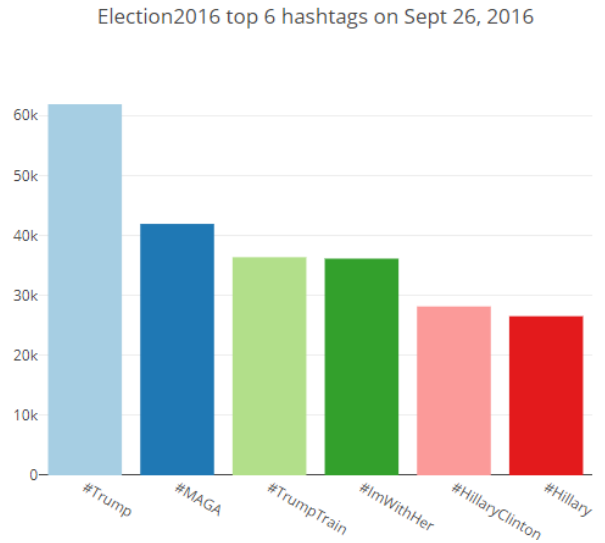
Visualize things that have natural shapes



http://prefrontal.org/blog/wp-content/uploads/2009/04/caret.png

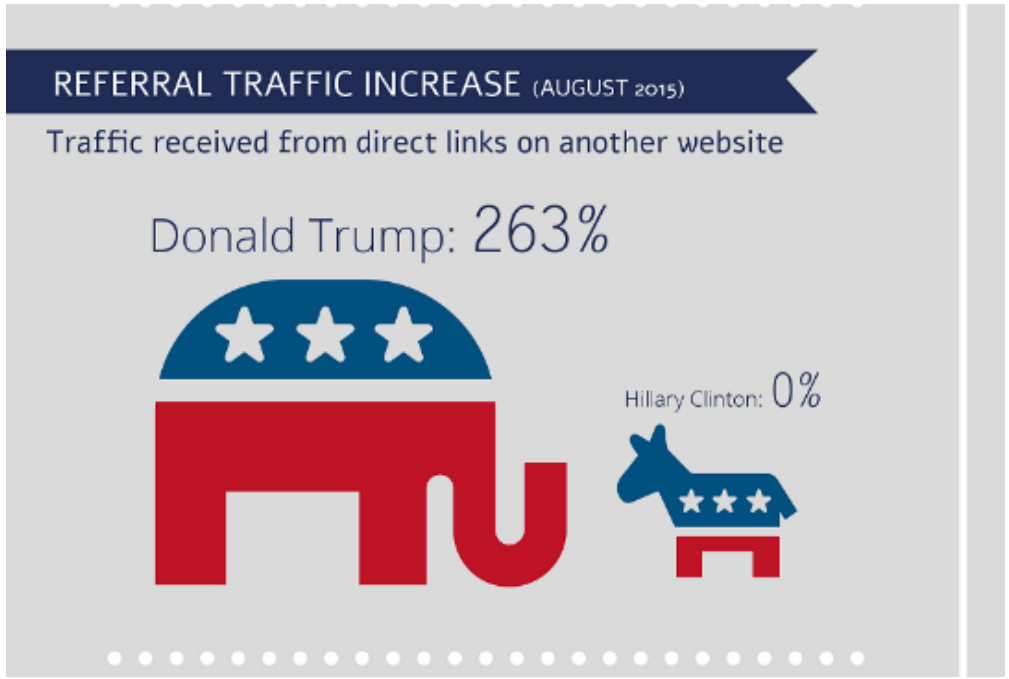
Information visualization

Map visual attributes to various abstract data dimensions/information (i.e., design visual encoding of data)

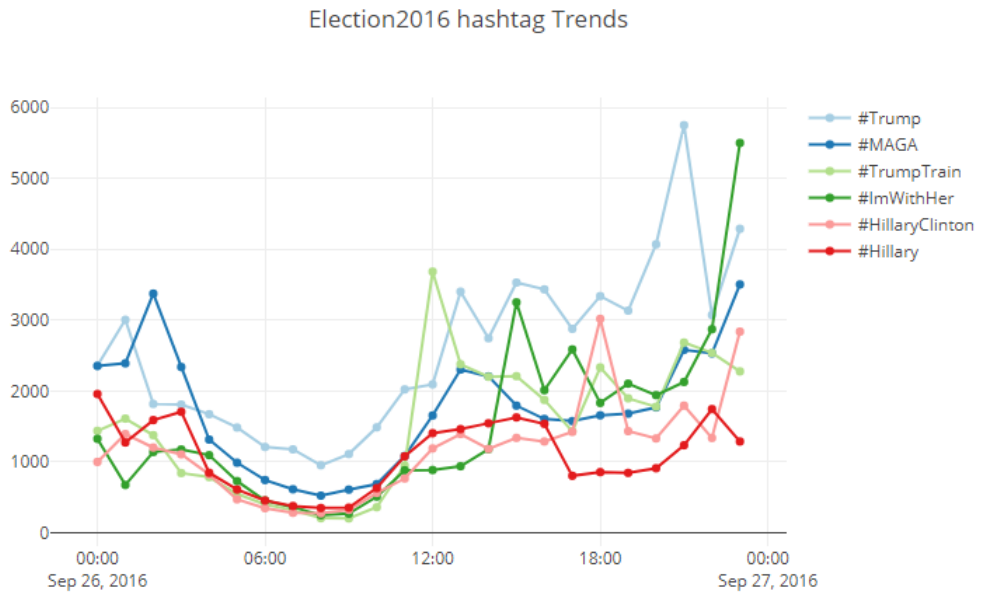


Purposes of Visualization

Presentation / Communication



Analysis / Exploration



https://www.similarweb.com/blog/infographic-us-presidential-election-campaign-2016-countdown

Variable types vs. Visual Attributes

Variable types

Nominal

- Categories that cannot be compared (e.g., names and labels)

Ordinal

- Categories that have orders (e.g., meat quality levels) but the diff. values are not meaningful

Quantitative – Interval

- Continuous variables where the differences are comparable, but there is no absolute zero point (e.g., date) so the ratio of two values is not meaningful

Quantitative – Ratio

- Interval variable + absolute zero points (e.g., length)

Visual attributes

Position

Size (length/area/volume)

Orientation (angle/slope)

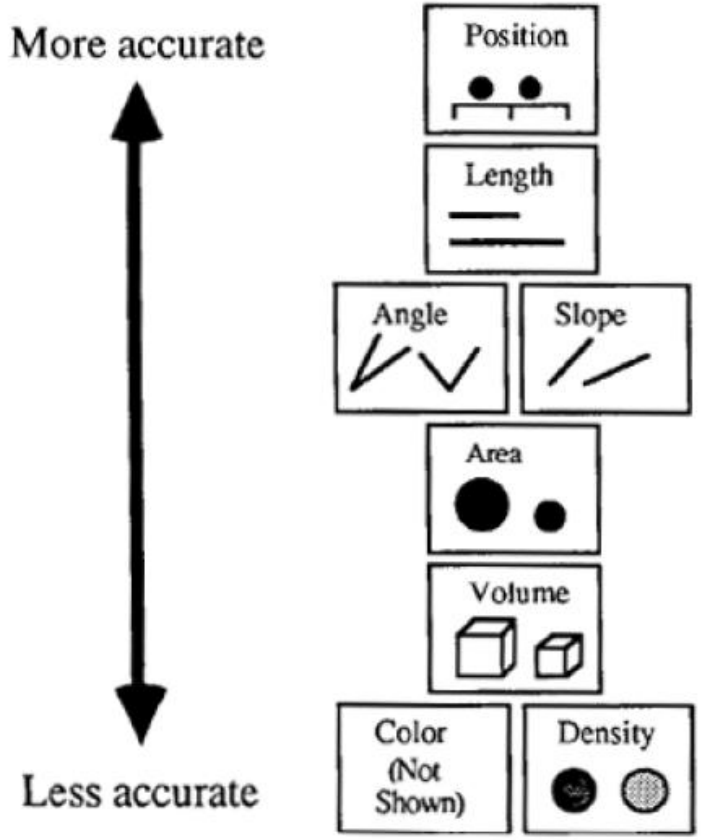
Density (lightness/opacity)

Shape

Texture

Color

Not all visual attributes are the same



Mackinlay, APT (A Presentation Tool), 1986

Visual attribute rankings

https://richardbrath.wordpress.com/2015/10/05/meta-ranking-of-visual-attributes-in-data-visualization/

Visual Attribute Rankings by Encoding showing each researchers' rank and an overall average score (lower is better match)

	QUANTITATIVE						ORDERED						CATEGORICAL					
Visual Attribute: Researcher:	Ber67	Mac86	Mac06	Mac96	Maz09	Avg Score*	Ber67	Mac86	Mac06	Mac96	Maz09	Avg Score*	Ber67	Mac86	Mac06	Mac96	Maz09	Avg Score*
Position	Yes	1	1	good	suitable	1.2	Yes	1	1	good	suitable	1.1	Yes	1	1	good	limited	1.9
Size (inc length, area, volume)	Yes	2	2	good	suitable	1.8	Yes	7	5	good	limited	4.5	no	8	6	good	not	6.9
Angle (inc slope, orientation)	no	3	3	marginal	limited	5.6	no	8	6	marginal	limited	7.4	Yes	9	7	good	not	5.8
Brightness (value, intensity)	no	4	4	marginal	limited	6.2	Yes	2	2	good	suitable	1.7	no	5	4	poor	not	7.4
Color Hue	no	6	6	marginal	not	8.2	no	4	4	marginal	not	6.7	Yes	2	3	good	suitable	1.8
Shape	no	no	no	poor	not	9.0	no	no	no	poor	not	9.0	Yes	7	2	good	suitable	2.5
Texture	no	no		marginal		7.0	Yes	5		marginal		3.9	Yes	3		good		1.7
Saturation		5	5	marginal		6.7		3	3	good		3.0		6	5	poor		7.1
Arrangement† (inc connection, cor		no		poor	not	9.0		6		poor	not	8.3		4		marginal	not	6.0
Crispness / Resolution				poor						good						poor		
Transparency				poor						good						marginal		
Curvature					limited						limited						not	
Added marks					not						not						suitable	
Numerosity					suitable						suitable						not	
Concavity/Convexity					limited						limited						not	
Flicker					not						not						limited	
Motion					limited						limited						not	

** A lower score indicates a better match. Avg score only provided if more than one author lists attribute*

† Arrangement includes connection, containment, and spatial grouping

*** Table should be updated with Illinsky and Munzner*

Interaction

Pointing / Selecting
- Mouse over
- click

Filtering / Searching

Navigation (Zoom, pan, scale, rotate)

Sorting

Brushing & Linking
- Brushing: Selecting a subset of points
- Linking: Points or ranges are linked with the selected points

https://plot.ly/dash/interactive-graphing

Dash has a lot of these interaction techniques by default

Ben Shneiderman's Visualization Mantra:
Overview first, Zoom & Filter, Details on Demand

Animations & Transitions

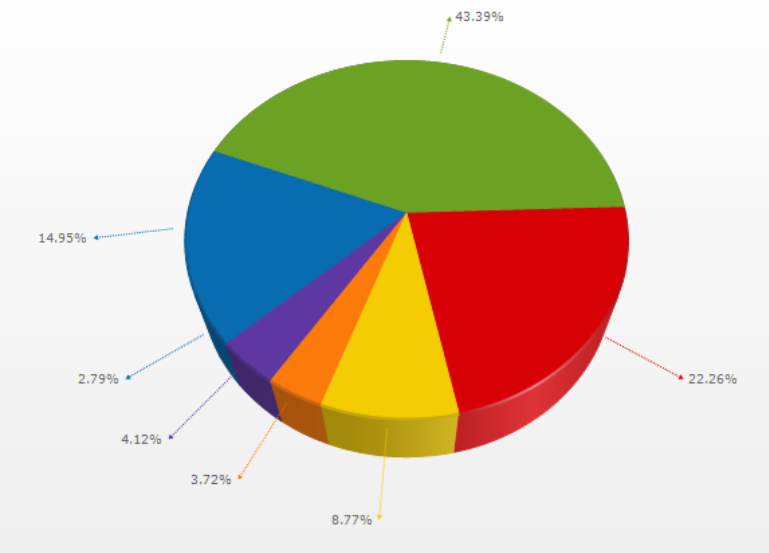
Need to be used carefully, but when use properly, they help keep context

See GapMinder:

- https://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen
- https://plot.ly/python/gapminder-example/

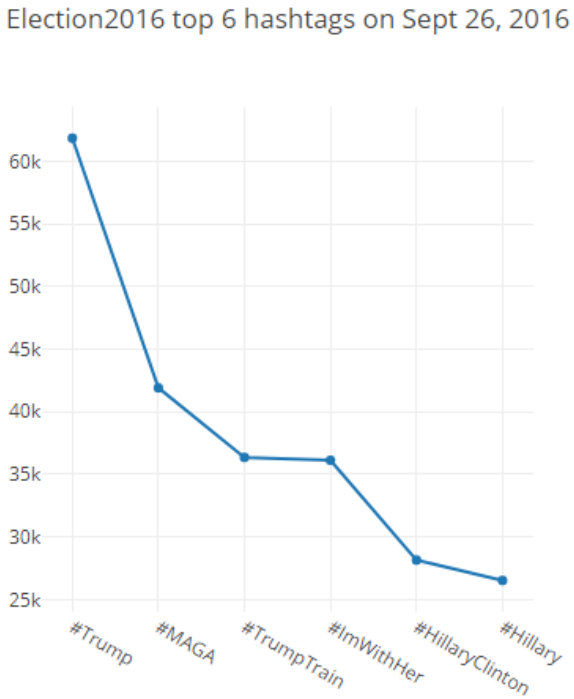
Two DONTs

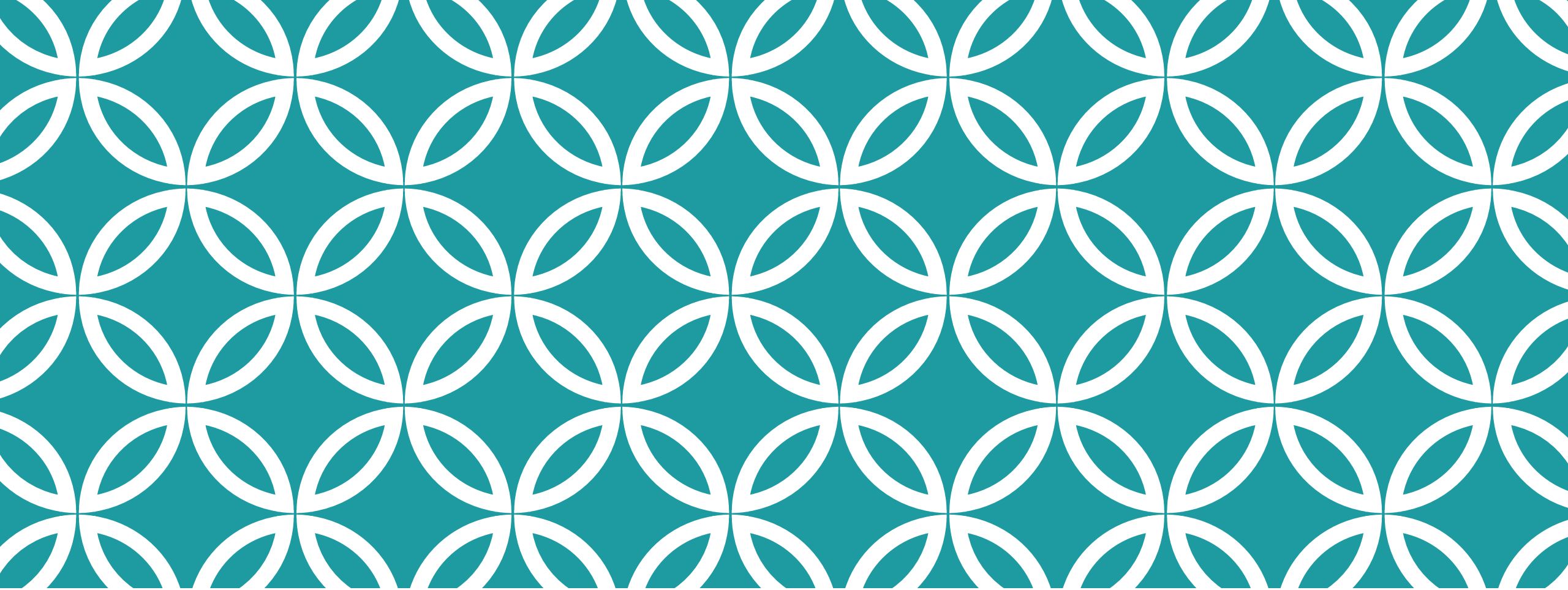
No pie chart, especially 3D pie charts



https://www.infragistics.com/samples/WebImages/windows-forms/Controls/chart/windows-forms-chart-3d-pie-chart-en-us.png

No line charts for categorical data





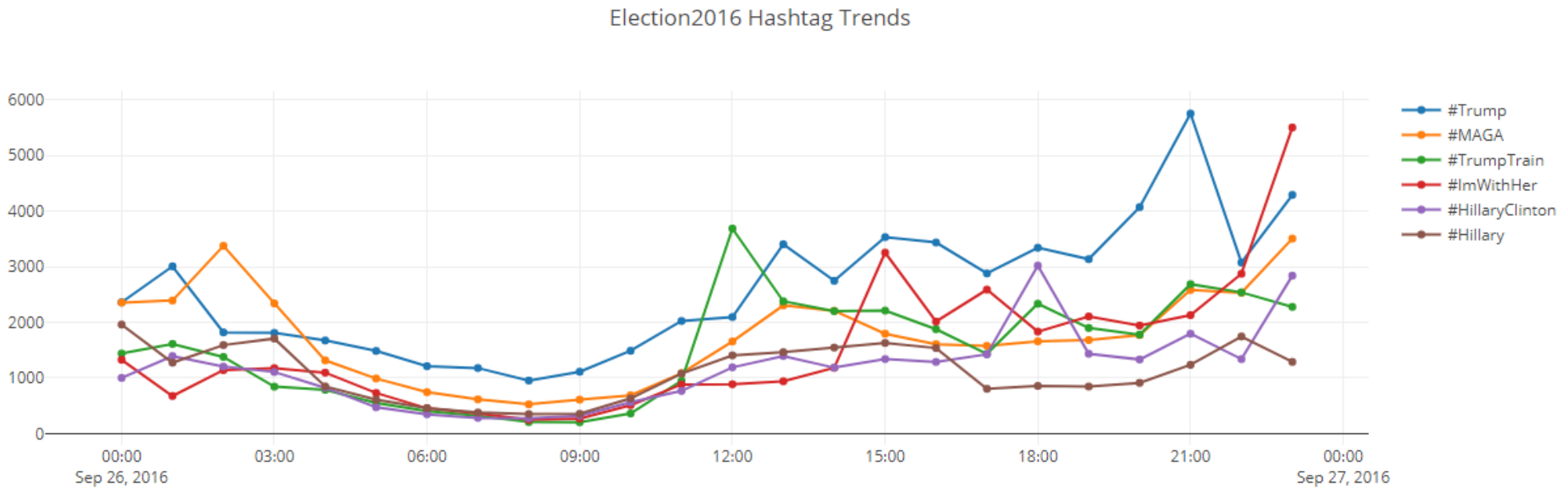
Other Chart Examples

Line Chart

03_dash-election2016-linechart.py

Line Chart

A demo to show a line chart.

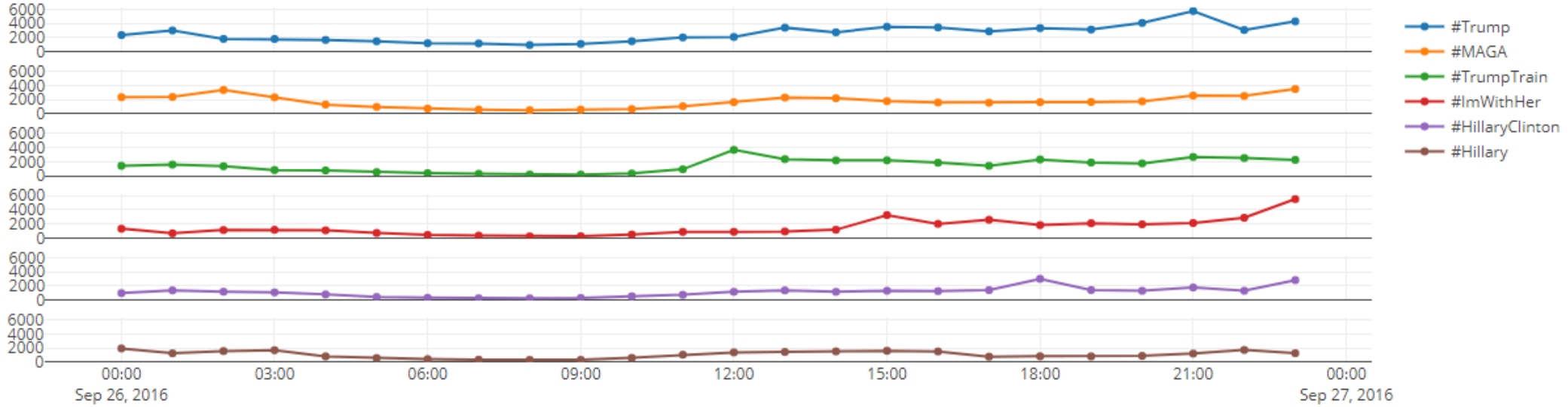


Small multiples

04_dash-election2016-small-multiples.py

Small Multiples

A example of small multiples using top hashtag trends on Sept 26, 2016.

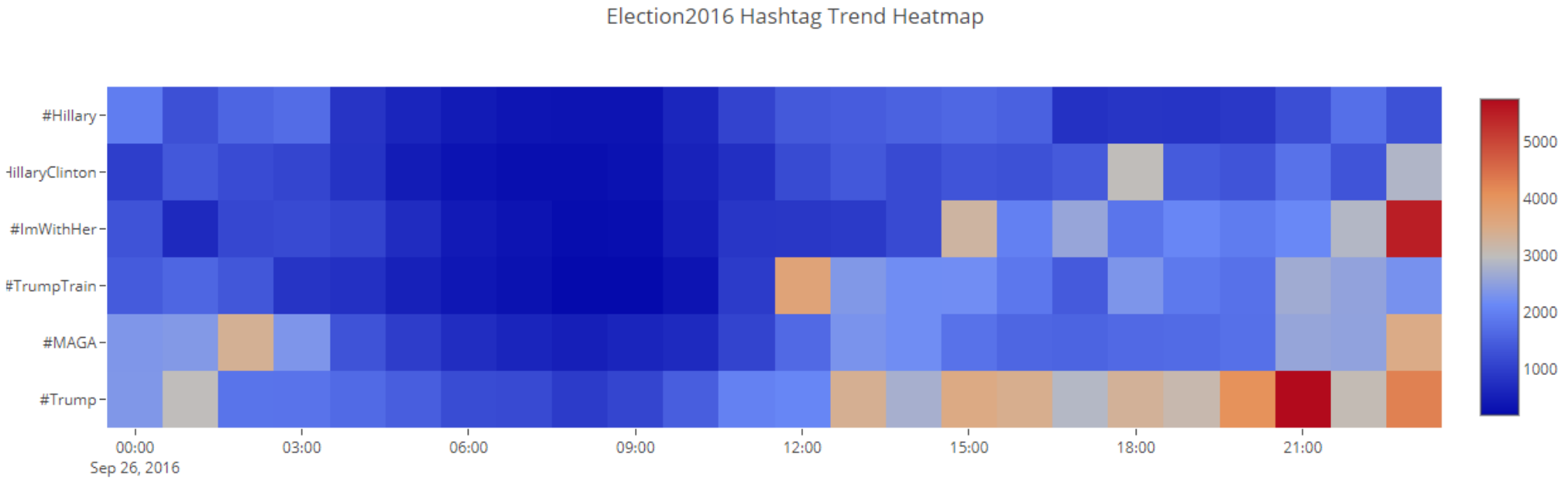


Heatmap

05_dash-election2016-heatmap.py

Heatmap

A demo to show a heatmap.

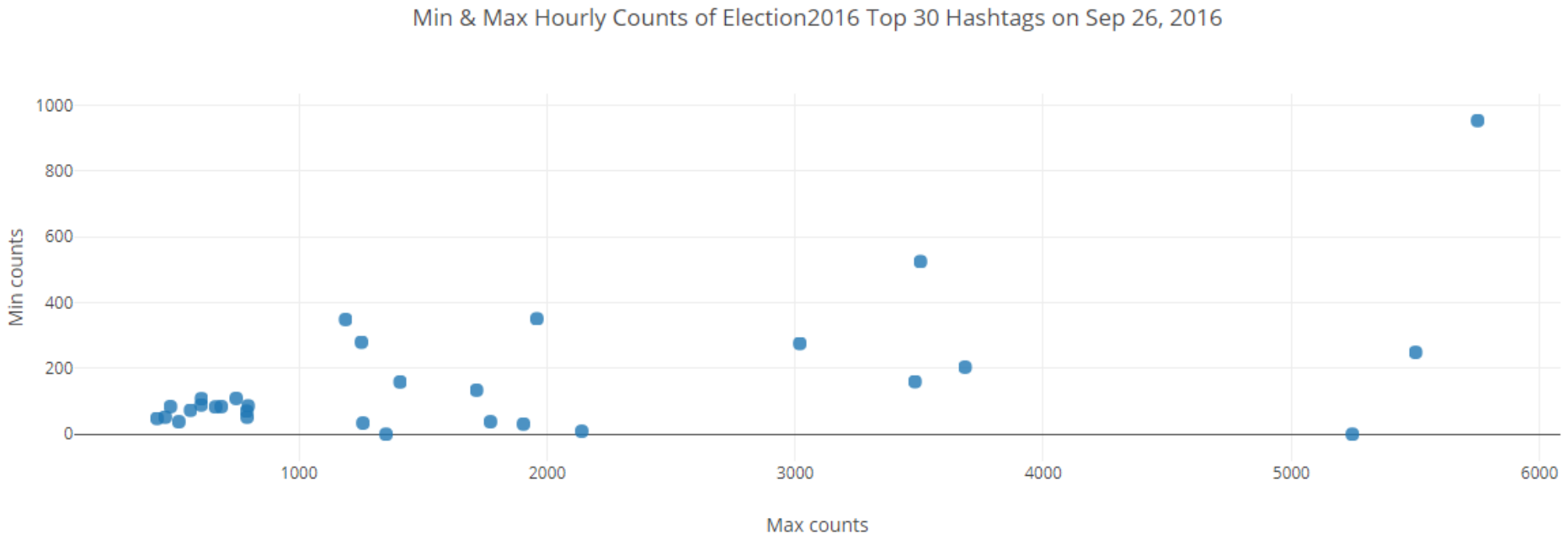


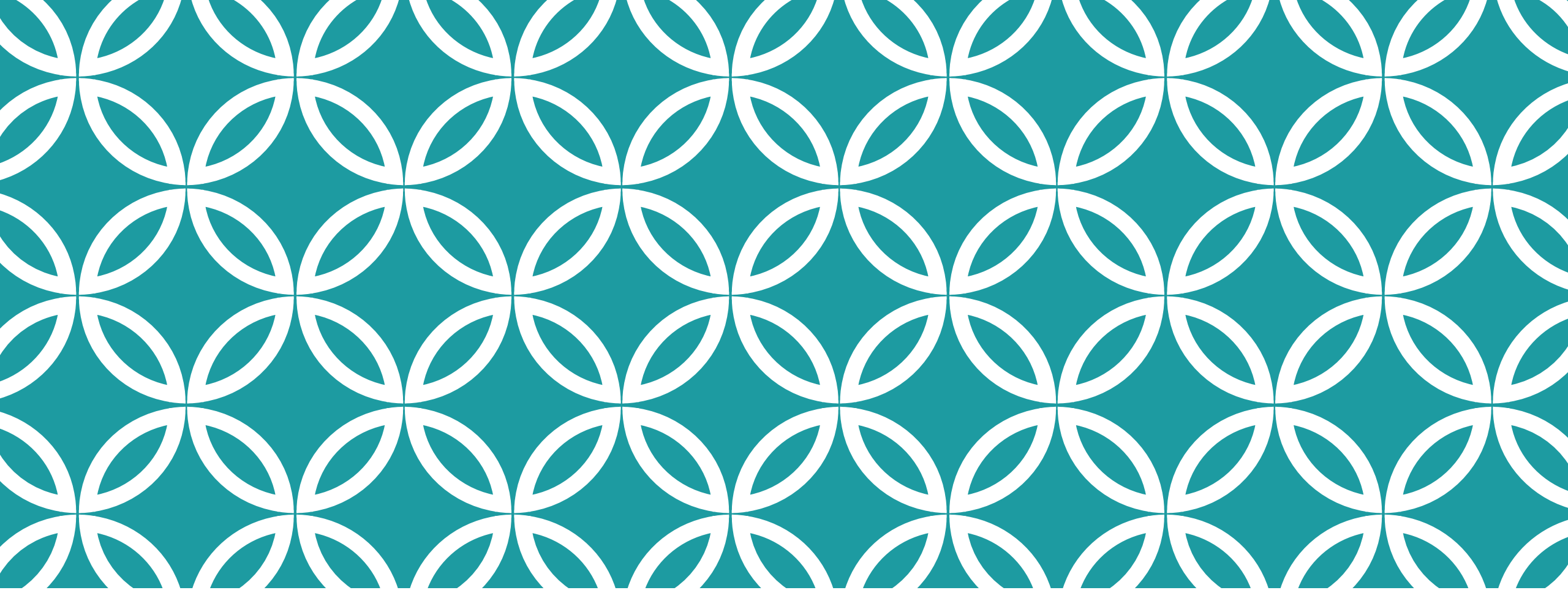
Scatter Plot

06_dash-election2016-scatter-plot.py

Scatter Plot

A demo to show a scatter plot.





Wrap Up...

In today's lecture, we have learned...

Basic Setup and Usage of Dash/Ploty

- Barchart
- Linechart
- Small multiples

Basic Visualization Concepts

Before the next class

Think about how you would want to visualize your data

Bring some small & hand-generated mock data

We will learn some more advanced uses of Dash and let you work on your mock data!

References for Plotly & Dash

Plotly.py

https://plot.ly/python/getting-started/

Dash Website

https://plot.ly/dash/

Dash Official Tutorials

https://plot.ly/dash/getting-started

D3.js Gallery

https://github.com/d3/d3/wiki/Gallery

Data-Driven Documents