GENERAL ASSEMBLY

# DATA SCIENCE

## 10 WEEK PART TIME COURSE

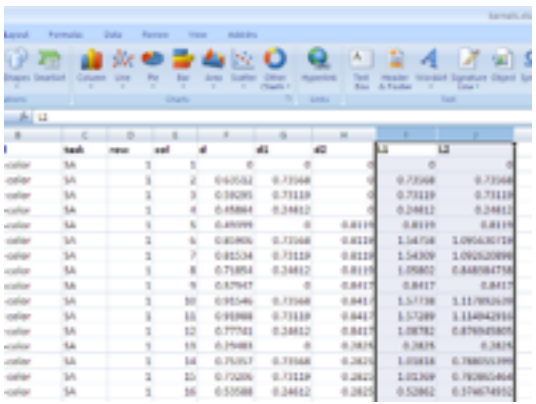
## Week 2 Lesson 1 - Data Visualisation
## Tuesday 30th May

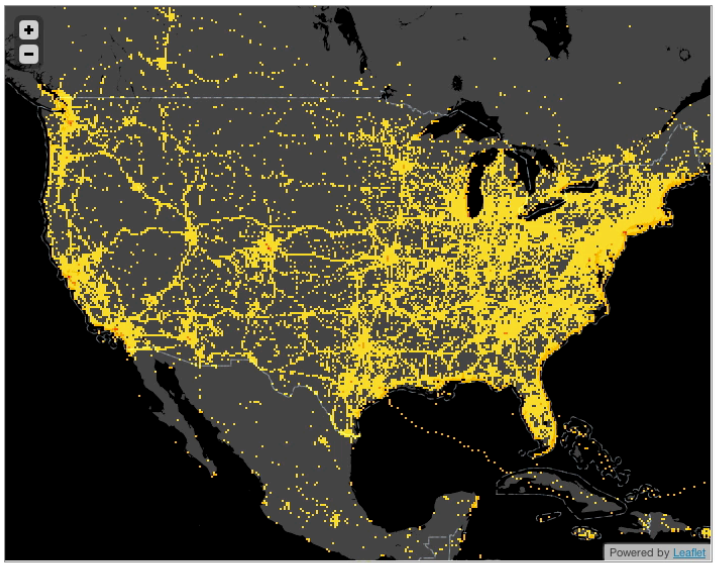
1. What is Data Visualisation?
2. Why do we visualise data?
3. How do we visualise data?
4. Different types of Charts
5. Basic rules for creating a graph
6. Lab
7. Discussion

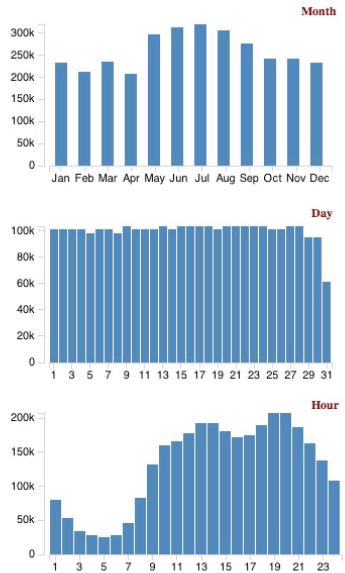
# WHAT IS DATA VISUALISATION?

# WHAT IS DATA VISUALISATION?







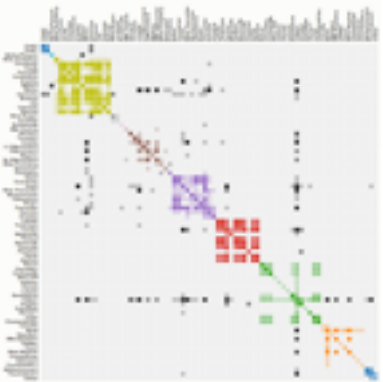

- Present information that is intuitive and clear for the viewer
- Turn numbers in a spreadsheet into something people can interpret and extract insights

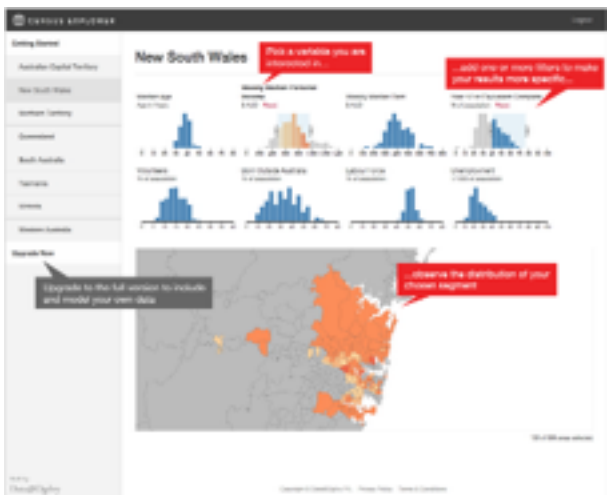
Reporting

- Dashboards and Business Intelligence
- Know the questions you want answers to
- Can detect changes from the norm
- Good for taking a 30,000 foot view of the problem

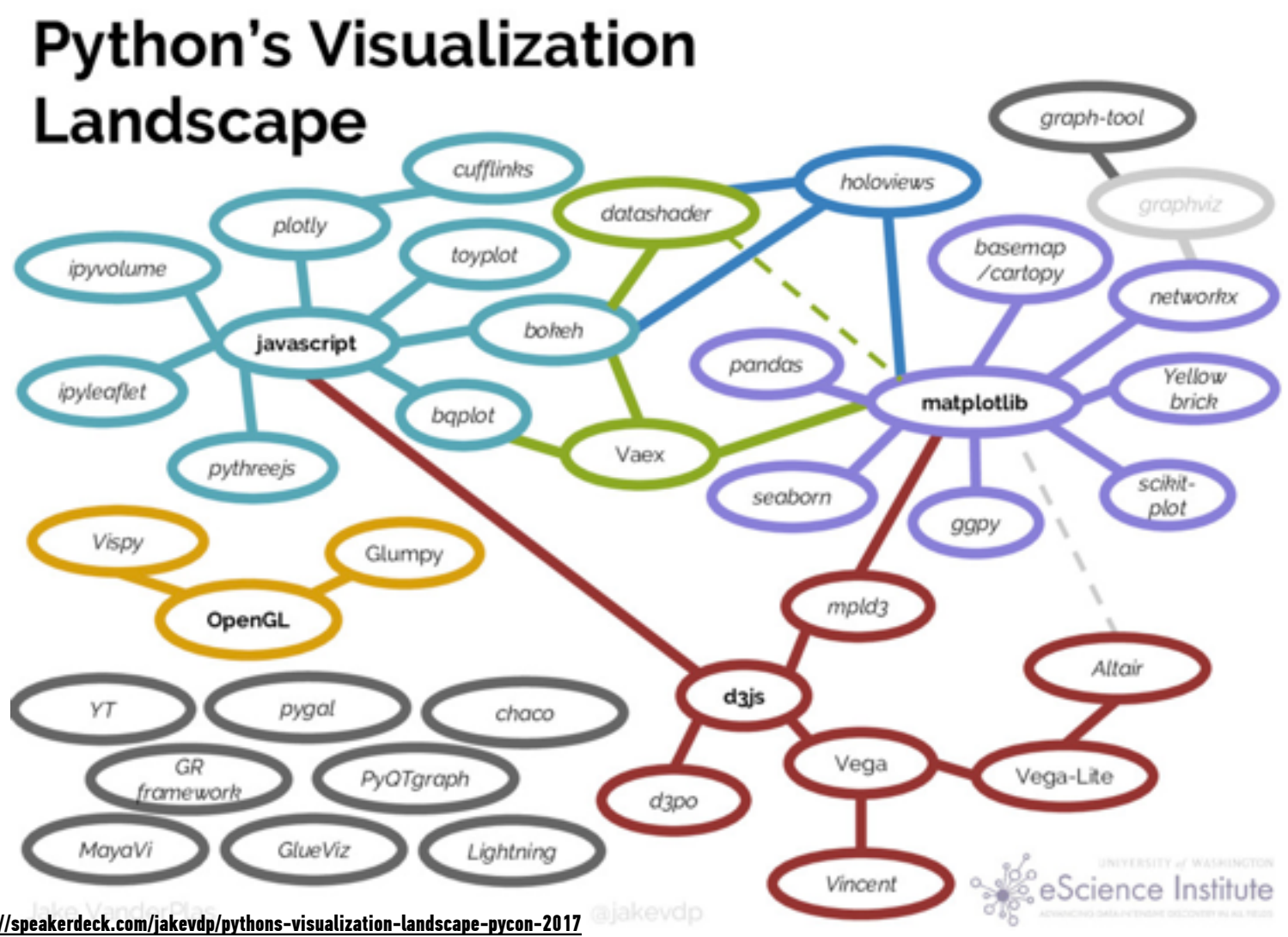

Exploring

- Exploratory Data Analysis
- Combines multiple data sources for single view of a problem
- Technical analysis of data
- Combined with modelling allows for the discovery of new problems and solutions





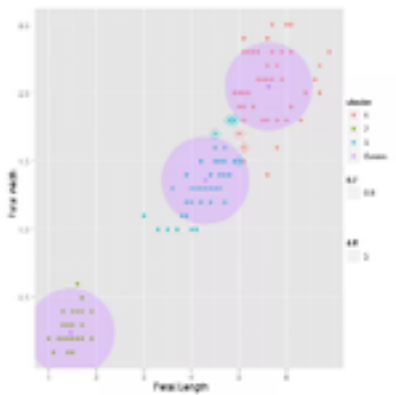

Ease of Use
Power
tableau

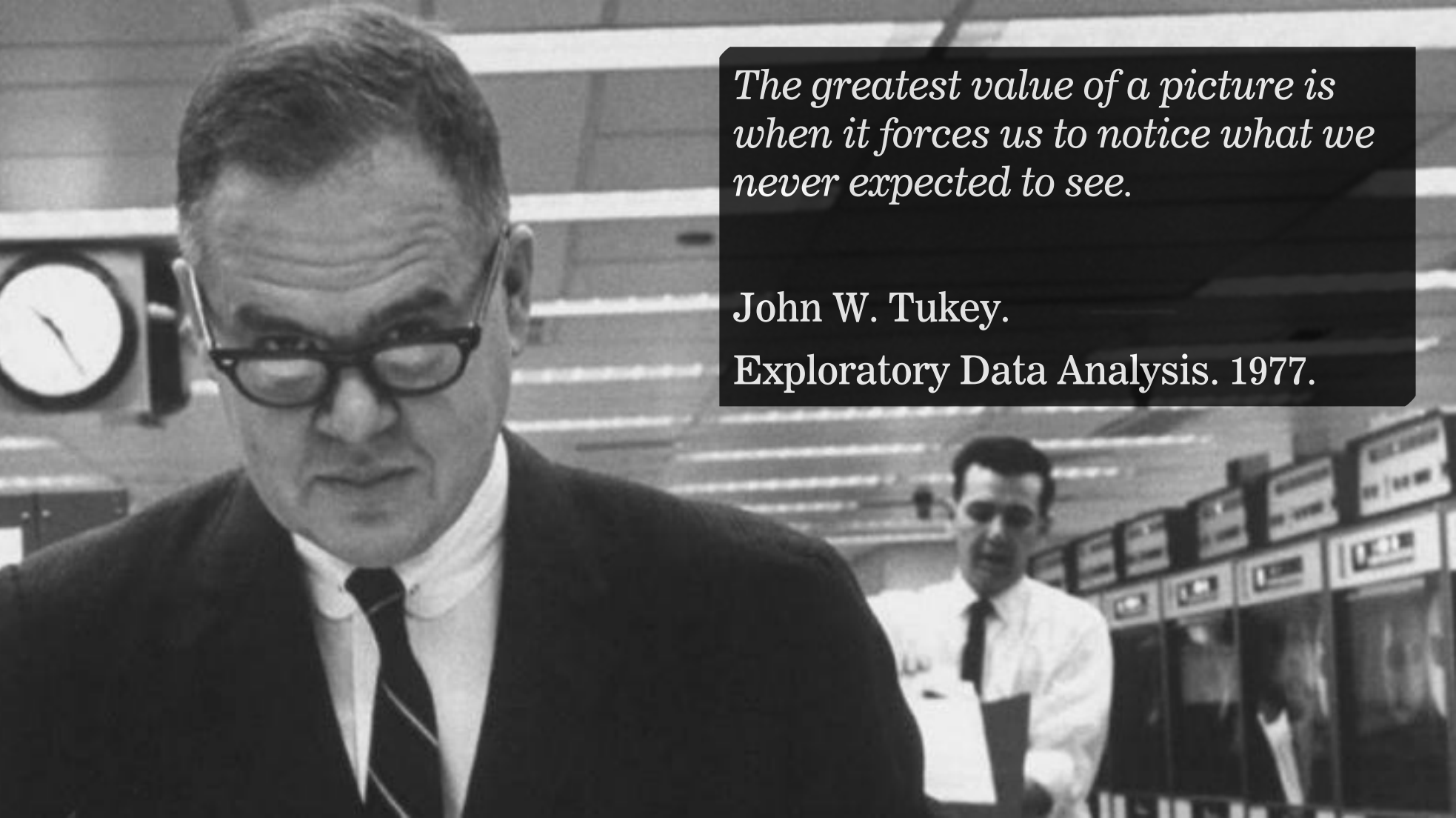



Jake VanderPlas, https://speakerdeck.com/jakevdp/pythons-visualization-landscape-pycon-2017

| | Easy to Use | Powerful |
|---|---|---|
| Advantages | • Provides a useful starting point<br>• Familiar to a large audience<br>• Prototyping and design time is reduced<br>• Default settings reduce the options and thinking that goes into producing a graph | • Scales to larger datasets<br>• Customised visualisations can create engaging visualisations<br>• Open-source (so free to run and extend)<br>• Non-obvious insights can be discovered with modelling tools<br>• Re-use code to produce similar charts for different data |
| Disadvantages | • Reproducing analysis requires lots of manual effort<br>• Limited to relative small data sets<br>• Solves known problems and cannot answer complex questions<br>• Licensing can be expensive | • Requires specialist skills to produce a graph<br>• Training and education for some of the output might be necessary |

- Communicate what's happening within the business
- Support decisions with information
- Measure and report the impact of decisions
- Discover ways to improve the business

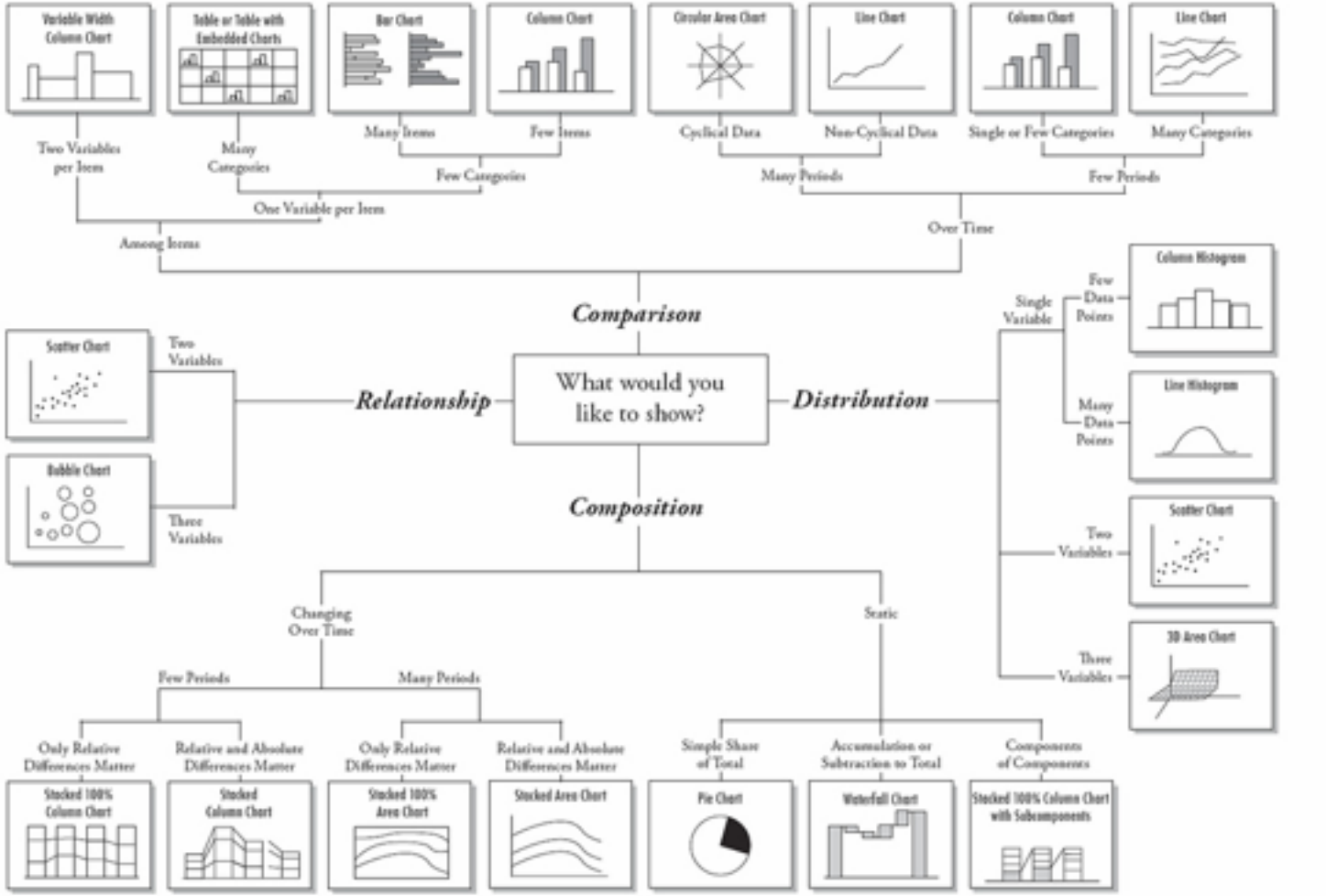



*The greatest value of a picture is when it forces us to notice what we never expected to see.*
John W. Tukey.
Exploratory Data Analysis. 1977.

DATA SCIENCE PART TIME COURSE

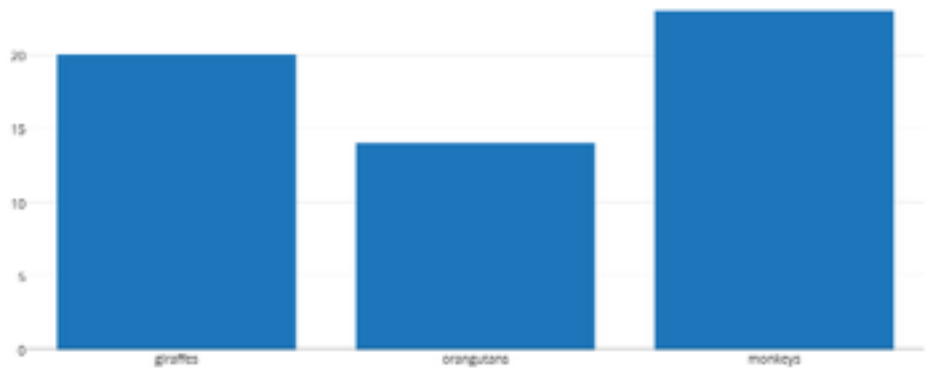
# DIFFERENT TYPES OF CHARTS

# Chart Suggestions—A Thought-Starter

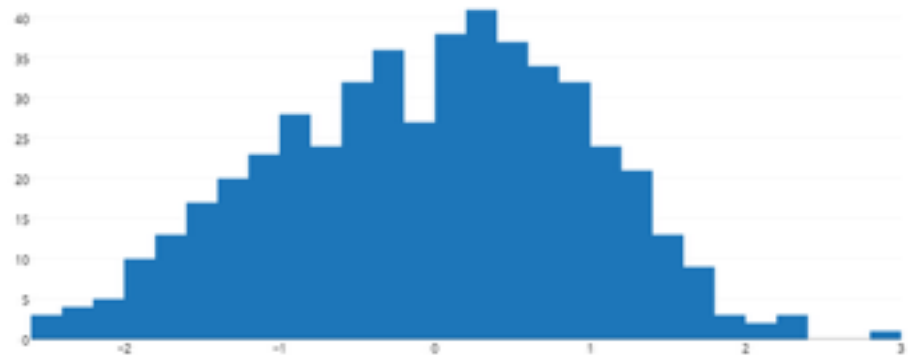


- Shows numeric summaries across different categories (either horizontally or vertically)
- Each bar represents a different category in the data

```
import plotly.plotly as py
import plotly.graph_objs as go

data = [go.Bar(
            x=['giraffes', 'orangutans', 'monkeys'],
            y=[20, 14, 23]
    )]
py.iplot(data, filename='basic-bar')
```

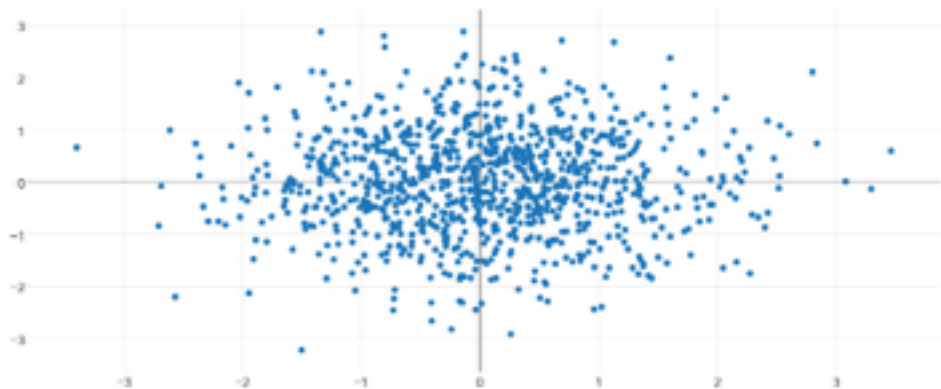


- Shows the distribution of data over a continuous interval
- Allows us to see the shape of our data




```
import plotly.plotly as py
import plotly.graph_objs as go

import numpy as np

x = np.random.randn(500)
data = [go.Histogram(x=x)]

py.iplot(data, filename='basic histogram')
```

- Shows values between two variables, one on each axis.
- Used to see a relationship between variables

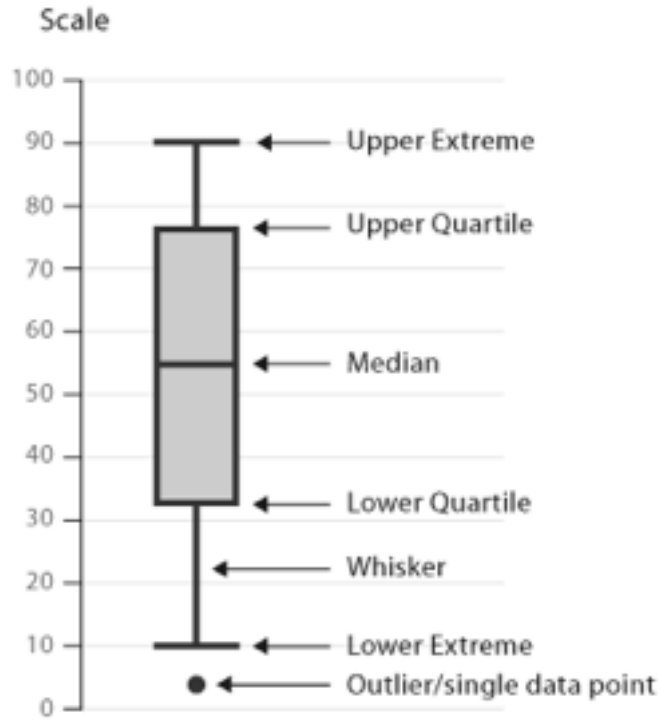


```
import numpy as np
N = 1000
random_x = np.random.randn(N)
random_y = np.random.randn(N)
trace = go.Scatter(
    x = random_x,
    y = random_y,
    mode = 'markers'
)
data = [trace]
py.iplot(data, filename='basic-scatter')
```

# BOX AND WHISKER PLOT

- Displays numerical distribution summaries by groups through quartiles
- Can compare different distributions

```
import numpy as np
y0 = np.random.randn(50)-1
y1 = np.random.randn(50)+1
trace0 = go.Box(
    y=y0
)
trace1 = go.Box(
    y=y1
)
data = [trace0, trace1]
py.iplot(data)
```

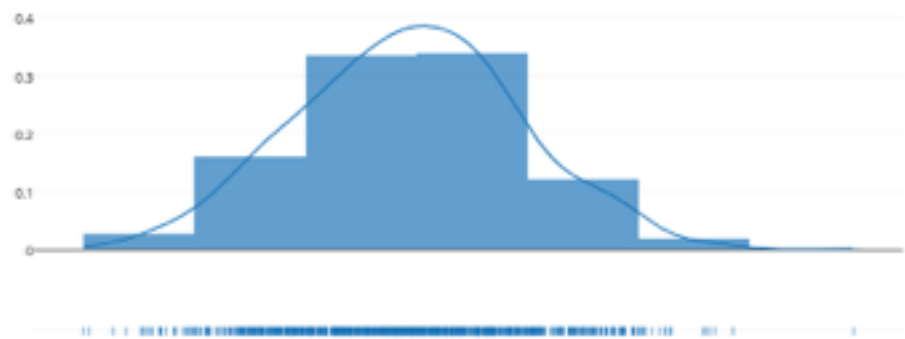



http://www.datavizcatalogue.com/methods/box_plot.html

# DENSITY PLOT



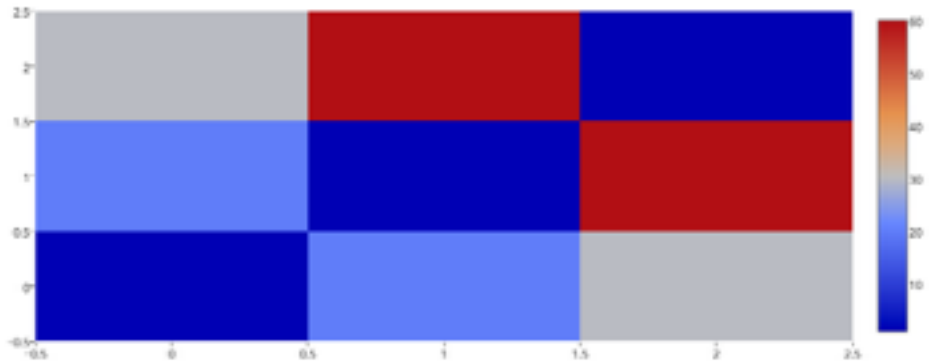
- Similar to a histogram but smooths out the distribution with a continuous line
- Not affected by bin choices




```
import numpy as np

x = np.random.randn(1000)
hist_data = [x]
group_labels = ['distplot']

fig = ff.create_distplot(hist_data, group_labels)
py.iplot(fig, filename='Basic Distplot')
```

- Colour coding applied to tabular data.
- Provides a generalised view of the data by each cell

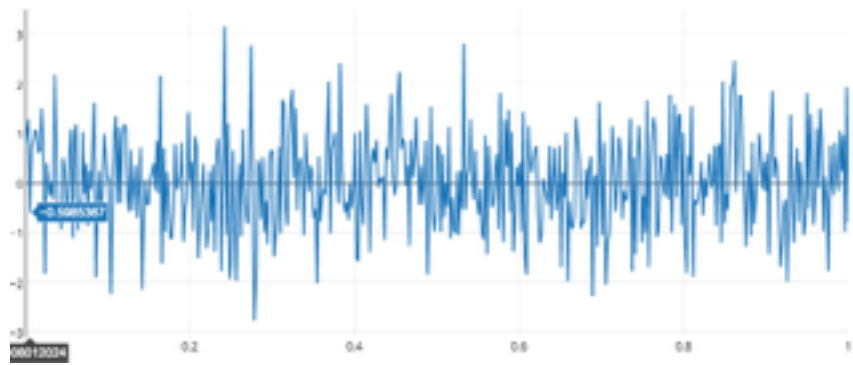
```
trace = go.Heatmap(z=[[1, 20, 30],
                      [20, 1, 60],
                      [30, 60, 1]])
data=[trace]
py.iplot(data, filename='basic-heatmap')
```



# LINE GRAPH

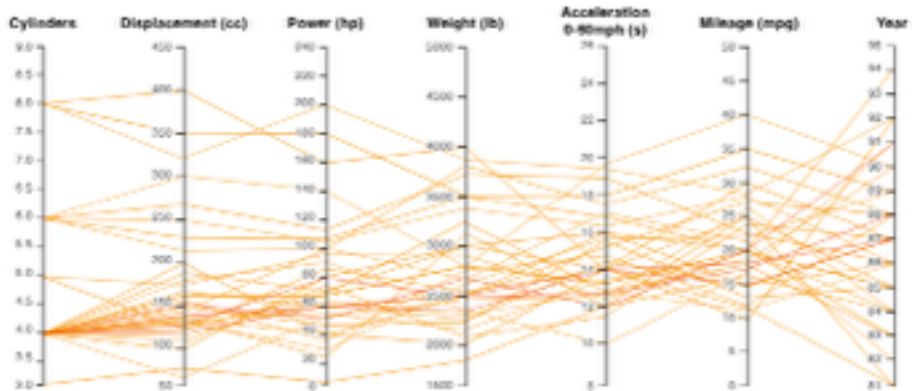


- Used to display a numeric value over a continuous value or time
- Used to observe trends and changes over time



```
import numpy as np
N = 500
random_x = np.linspace(0, 1, N)
random_y = np.random.randn(N)
trace = go.Scatter(
    x = random_x,
    y = random_y
)
data = [trace]
py.iplot(data, filename='basic-line')
```

- Plot multiple numeric variables across each observation
- Each axis is scaled and each line through the graph is an observation




```
import plotly.plotly as py
import plotly.graph_objs as go

data = [go.Bar(
            x=['giraffes', 'orangutans', 'monkeys'],
            y=[20, 14, 23]
    )]
py.iplot(data, filename='basic-bar')
```

- Allows us to plot points geographically
- We can overlay information on a map, usually loaded as a collection of ‘tiles’.

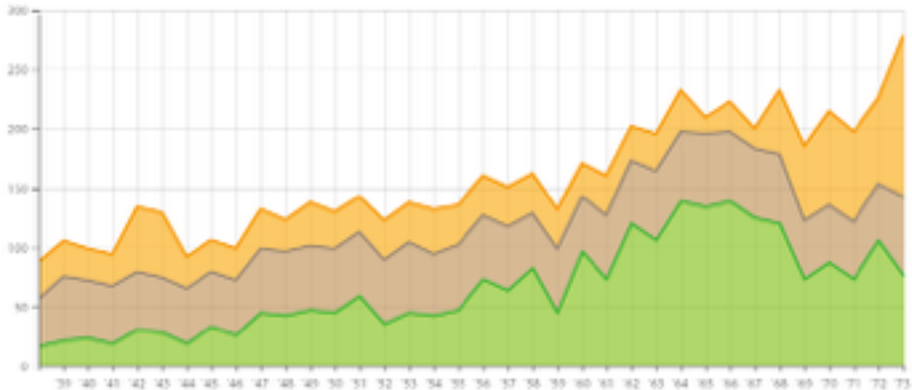
```
import folium


map_object = folium.Map(location=[-33.8, 151.2], zoom_start=6,
tiles="Stamen toner")

marker = folium.features.Marker([-33.869824, 151.206423],
popup=“General Assembly!”)

map_object.add_children(marker)
```

DATA SCIENCE PART TIME COURSE

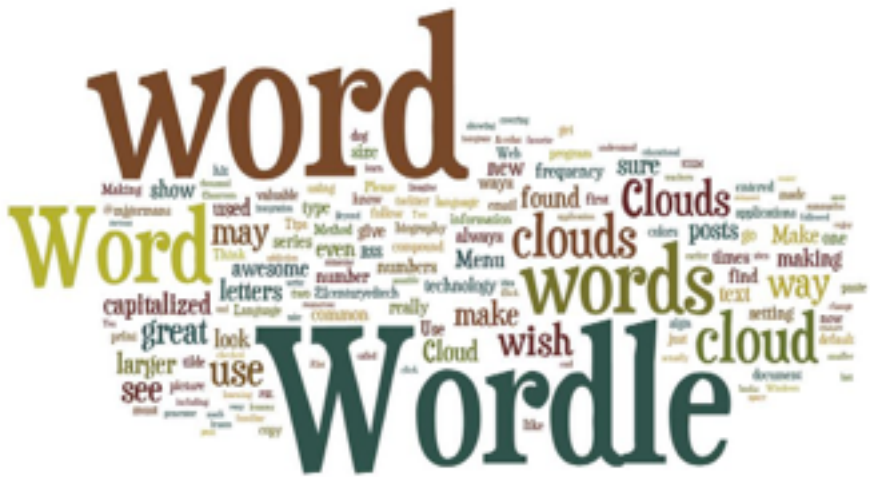
# DIFFERENT TYPES OF CHARTS

To avoid

- Try to track changes in a numeric variable across multiple categories
- The area under each category is represented by different colours
- Better to plot multiple lines rather than stacking the area



- Collection of words in a text or summary of a corpus
- Better to apply some form of categorical detection otherwise similar words will be lots of collection



- Represent a proportion that a category makes up of the whole
- Better to use column charts in most cases

DATA SCIENCE PART TIME COURSE

# DATA VISUALISATION LAB

# SYNCHING YOUR FORK WITH THE COURSE REPO



1. re-name your labs with lab_name.<yourname>.ipynb (to prevent a conflict)
2. cd <path to the root of your SYD_DAT_8 local repo>
3. commit your changes ahead of sync
   - git status
   - git add .
   - git commit -m "descriptive label for the commit"
   - git status
4. download new material from official course repo (upstream) and merge it
   - git checkout master (ensures you are in the master branch)
   - git fetch upstream
   - git merge upstream/master

DATA SCIENCE – Week 2 Day 1

# DISCUSSION TIME

- Review of last week
- Further Reading for Data Visualisation
- Homework due Friday 9th May
- Check in with course project
- Pre-reading for next lesson

## WEEK 1 Review

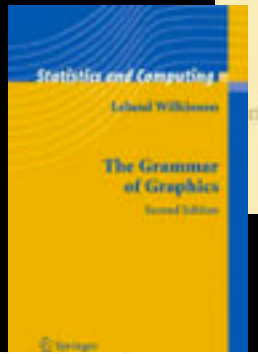
# DISCUSSION TIME

- Course Overview
- Data Science Overview
- Pre-work
- Python basics
- Git Basics

# DISCUSSION TIME

## Further Reading

- Edward Tufte, The Visual Display of Quantitative Information
- Leland Wilkinson, The Grammar of Graphics
- Scott Murray, Interactive Data Visualisation for the Web (free online)
- flowingdata.com
- New York Times (Upshot)



# DISCUSSION TIME

**Homework/Course/Project**

- How’s it going ?

DATA SCIENCE – Week 2 Day 1

# PRE-READING

An Introduction to Statistical Learning

- Chapter 3 – Linear Regression