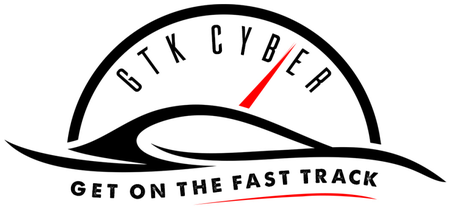

GTK CYBER
GET ON THE FAST TRACK

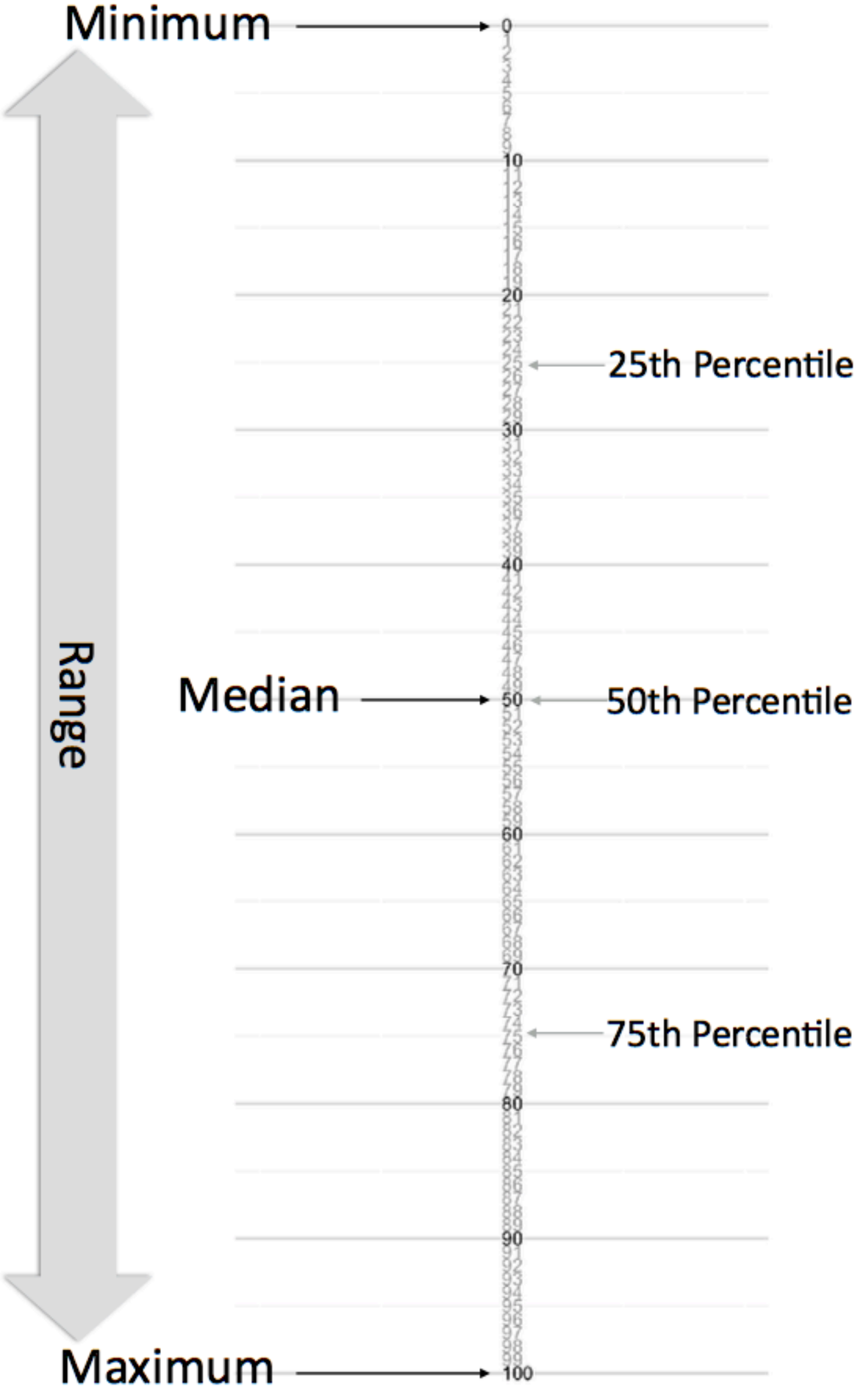
Data Science for Security Professionals - Day 2

Statistical Summaries and Data Visualization

Explore Your Data



- **Mean**: The mean is the average value of a set of numbers
- **Median**: The median is the middle value of an ordered set of numbers
- **Mode**: The mode is the value from a set which is repeated the most frequently.
- **Range**: Difference between minimum and maximum
- **Standard Deviation**: Measures dispersion in a data set. Close to zero indicates little dispersion.



`Series.abs()`	Absolute Value of the Series
`Series.count()`	Returns number of non-empty values in the series
`Series.max()`	Returns maximum value in the Series
`Series.mean()`	Returns the mean of a Series
`Series.median()`	Returns the median of a Series
`Series.min()`	Returns the minimum value in a Series
`Series.mode()`	Take a guess..
`Series.quantile([q])`	Returns the quantiles of a Series
`Series.sum`	Returns the sum of a series
`Series.std`	Returns the standard deviation of a Series

Tukey 5 Number Summary

- **Minimum:** The smallest value in the dataset
- **Lower Quartile:** Smallest 25% of the dataset
- **The Median:** The middle value of the dataset
- **Upper Quartile:** The largest 25% of the dataset
- **Maximum:** The largest value in the dataset


```
Series.describe()
```

Series.describe()

```
>>> random_numbers.describe()
count    50.000000
mean     50.620000
std      30.102471
min       1.000000
25%      25.500000
50%      54.000000
75%      73.000000
max      99.000000
dtype: float64
```


```
names = pd.Series(
 ['Jim','Bob',
 'Bob','Steve','Jim,'Jane','Steph','Emma','Rachel
 '])

names.describe()
```


```
names = pd.Series(
 ['Jim','Bob',
 'Bob','Steve','Jim,'Jane','Steph','Emma','Rachel
 '])

names.describe()
```

```
count        9
unique       7
top        Jim
freq         2
dtype: object
```

Finding Unique Values


```
unique_nums = []
for key, value in random_numbers.iteritems():
    if value not in unique_nums:
        unique_nums.append(value)
```

NO!!!


```
series.drop_duplicates()
series.unique()
```

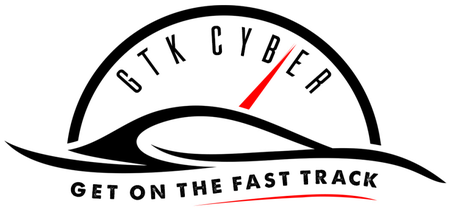
Finding Unique Values


```
series.value_counts()
```

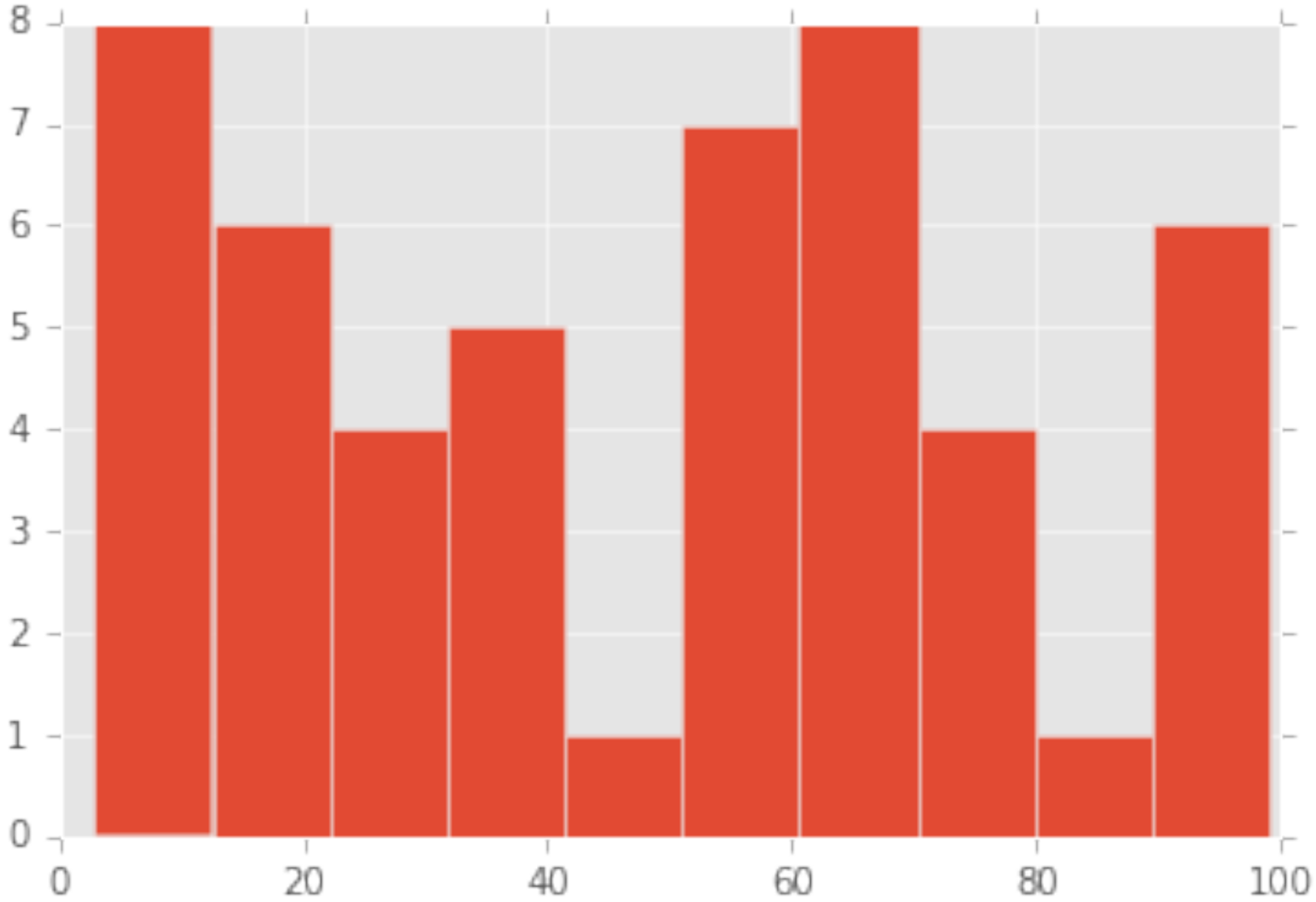

```
series.value_counts(bins=5)
```

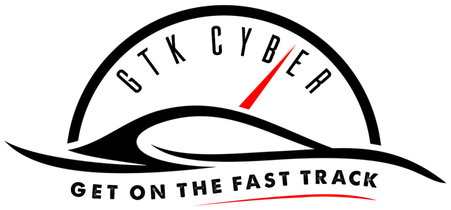

```
series.value_counts(bins=5,
normalize=True)
```


```
series.hist()
```

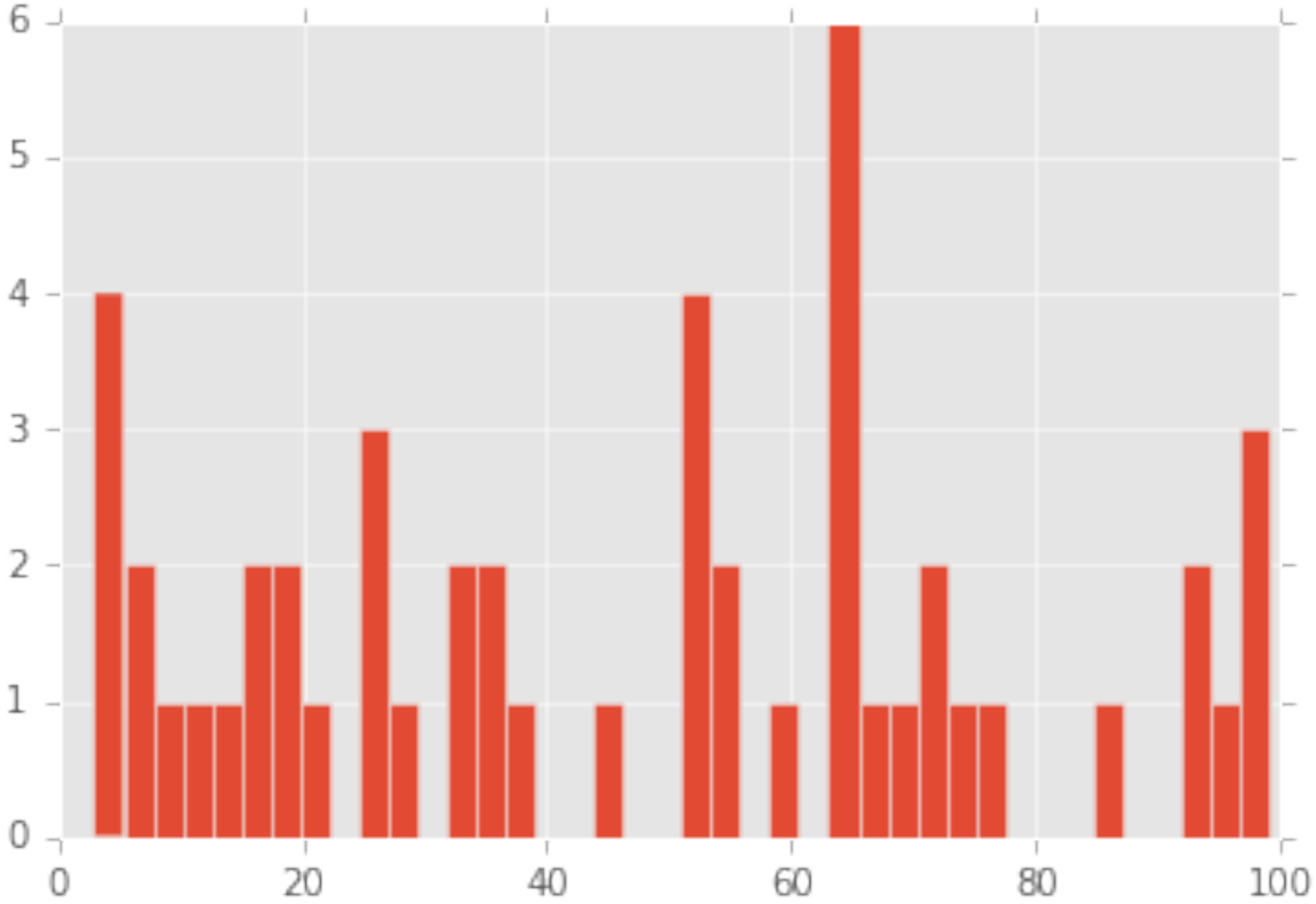


series.hist()



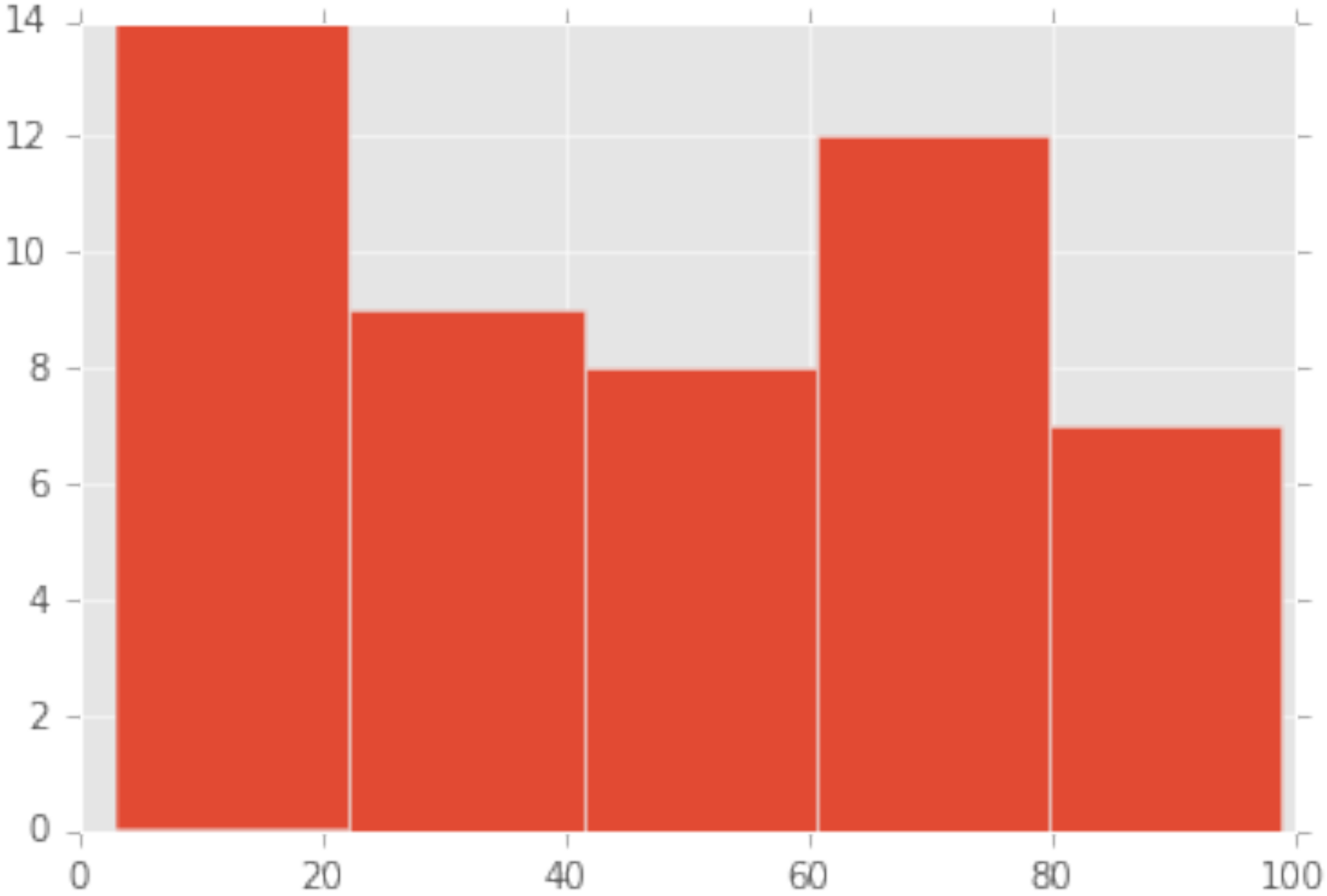


series.hist(bins=40)



GTK CYBER
GET ON THE FAST TRACK

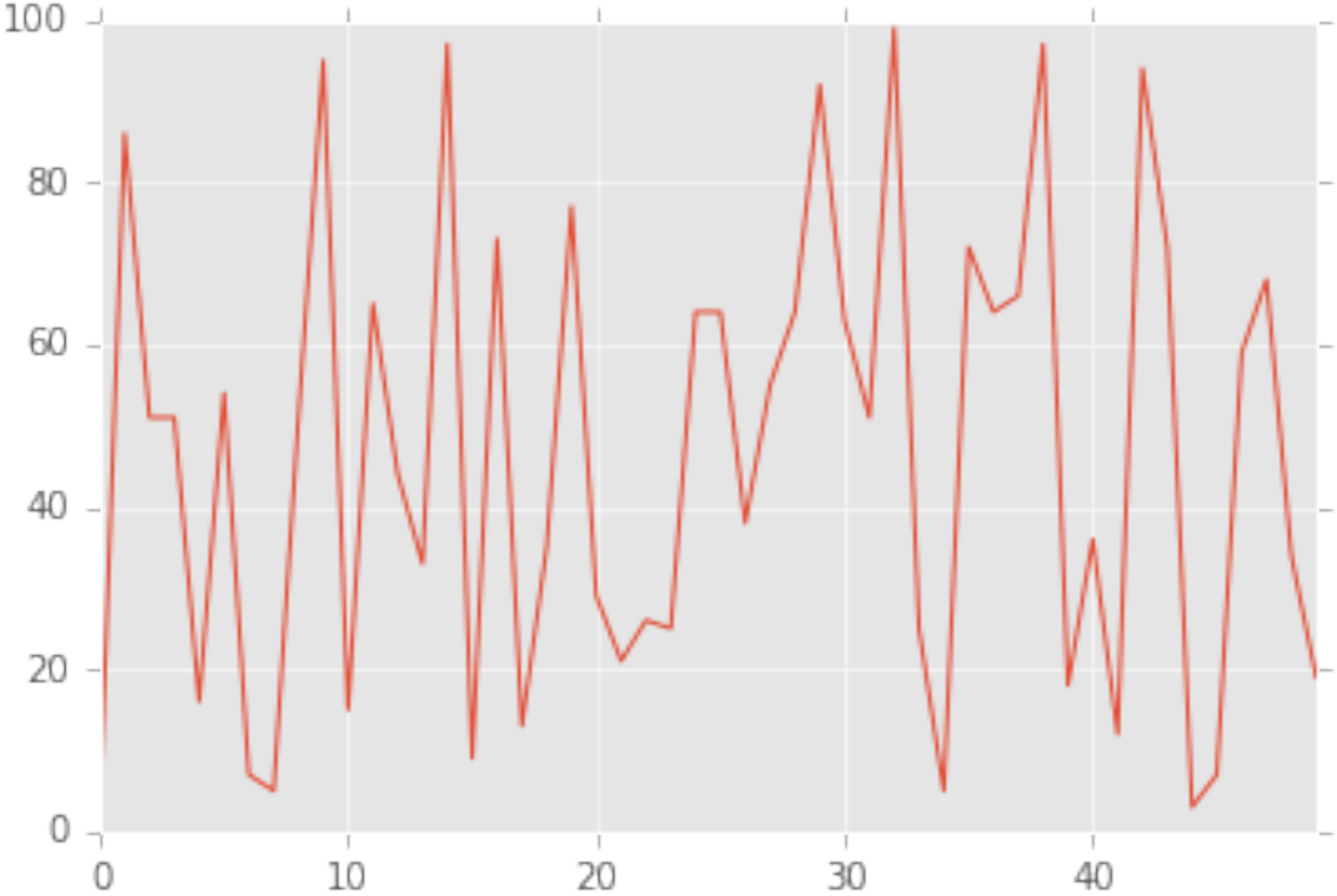
series.hist(bins=5)




```
series.plot()
```

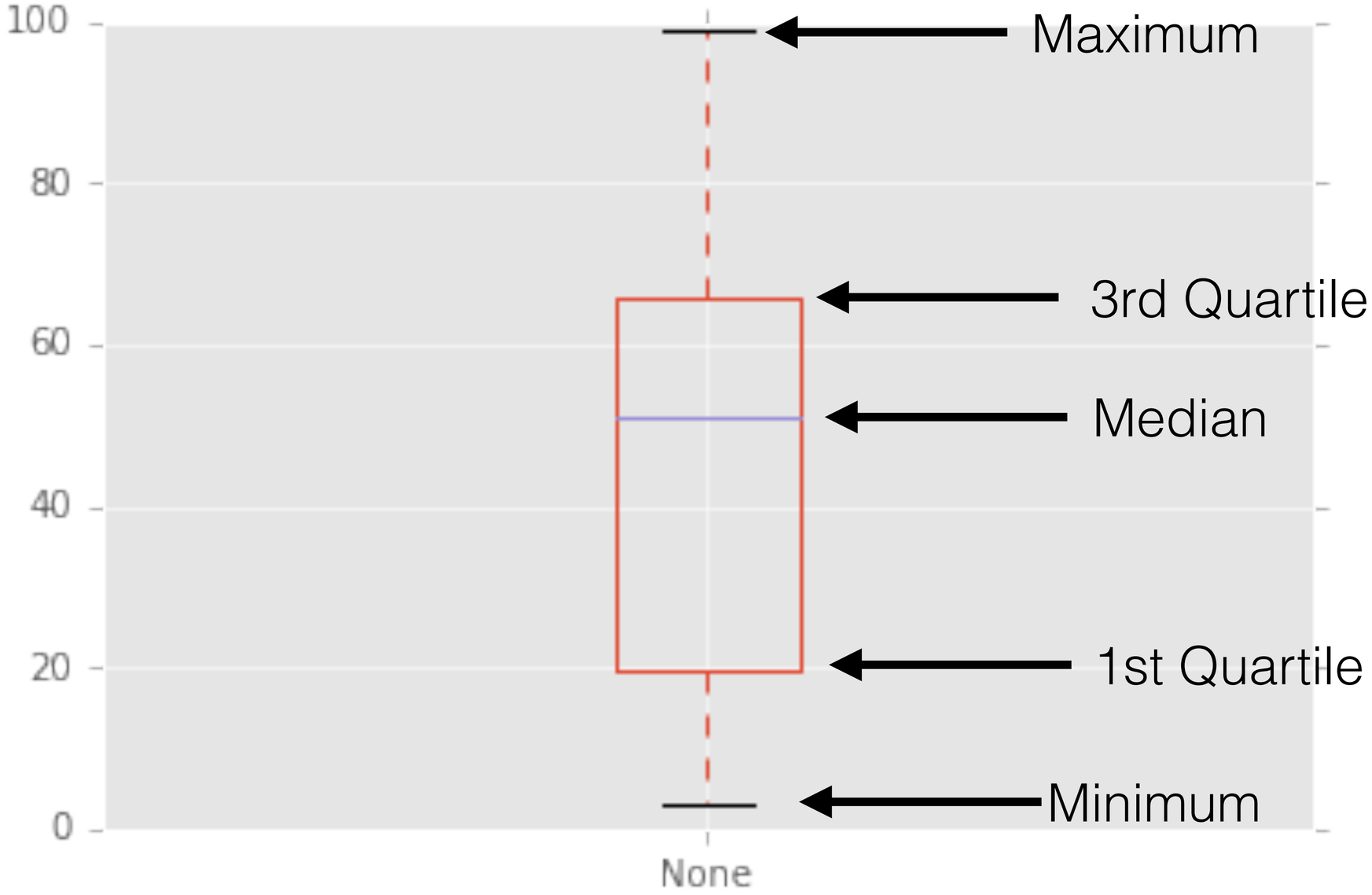
GTK CYBER
GET ON THE FAST TRACK

series.plot()



GET ON THE FAST TRACK

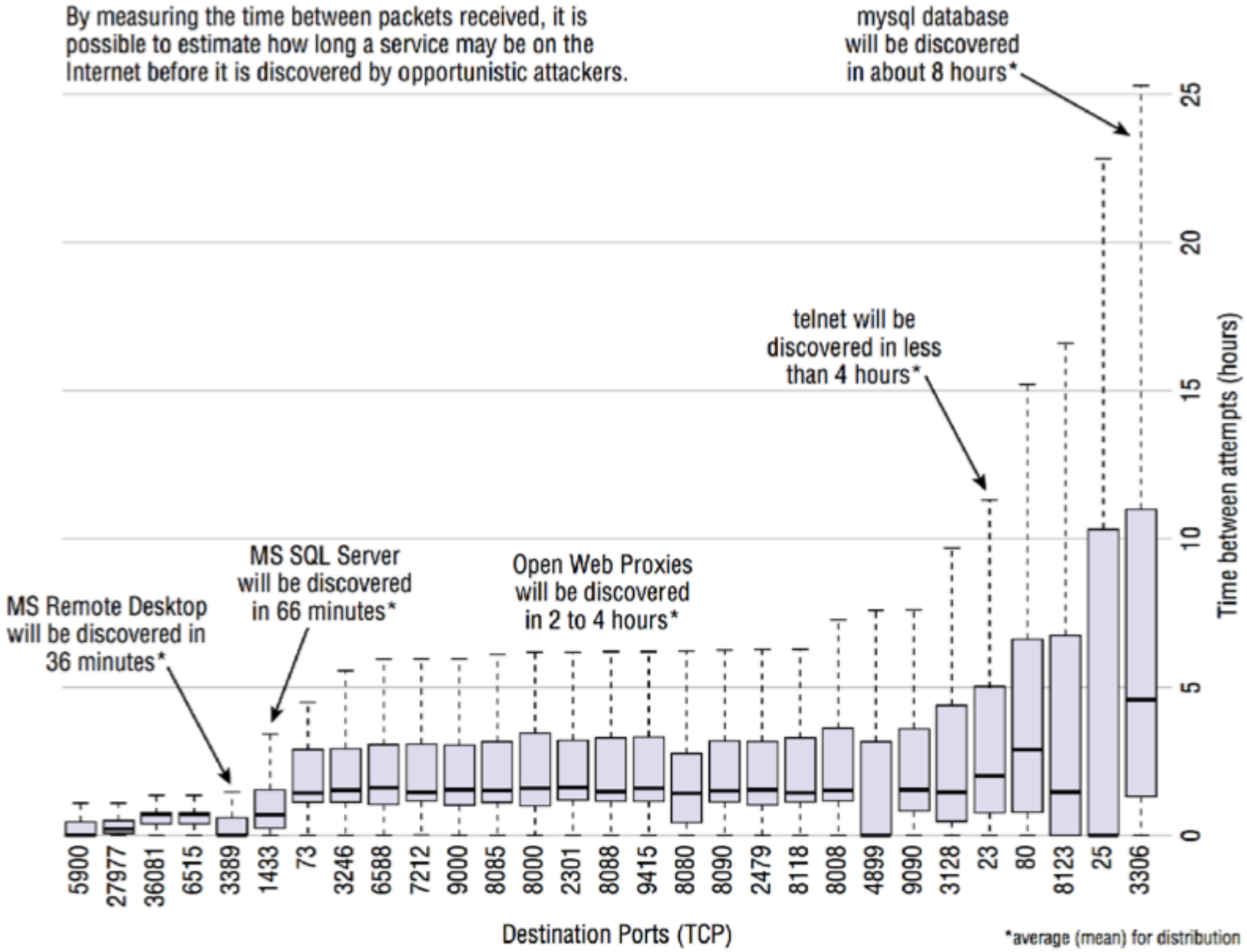
series.plot(kind='box')



GTK CYBER
GET ON THE FAST TRACK

How long will a service go undiscovered by opportunistic attackers?

By measuring the time between packets received, it is possible to estimate how long a service may be on the Internet before it is discovered by opportunistic attackers.

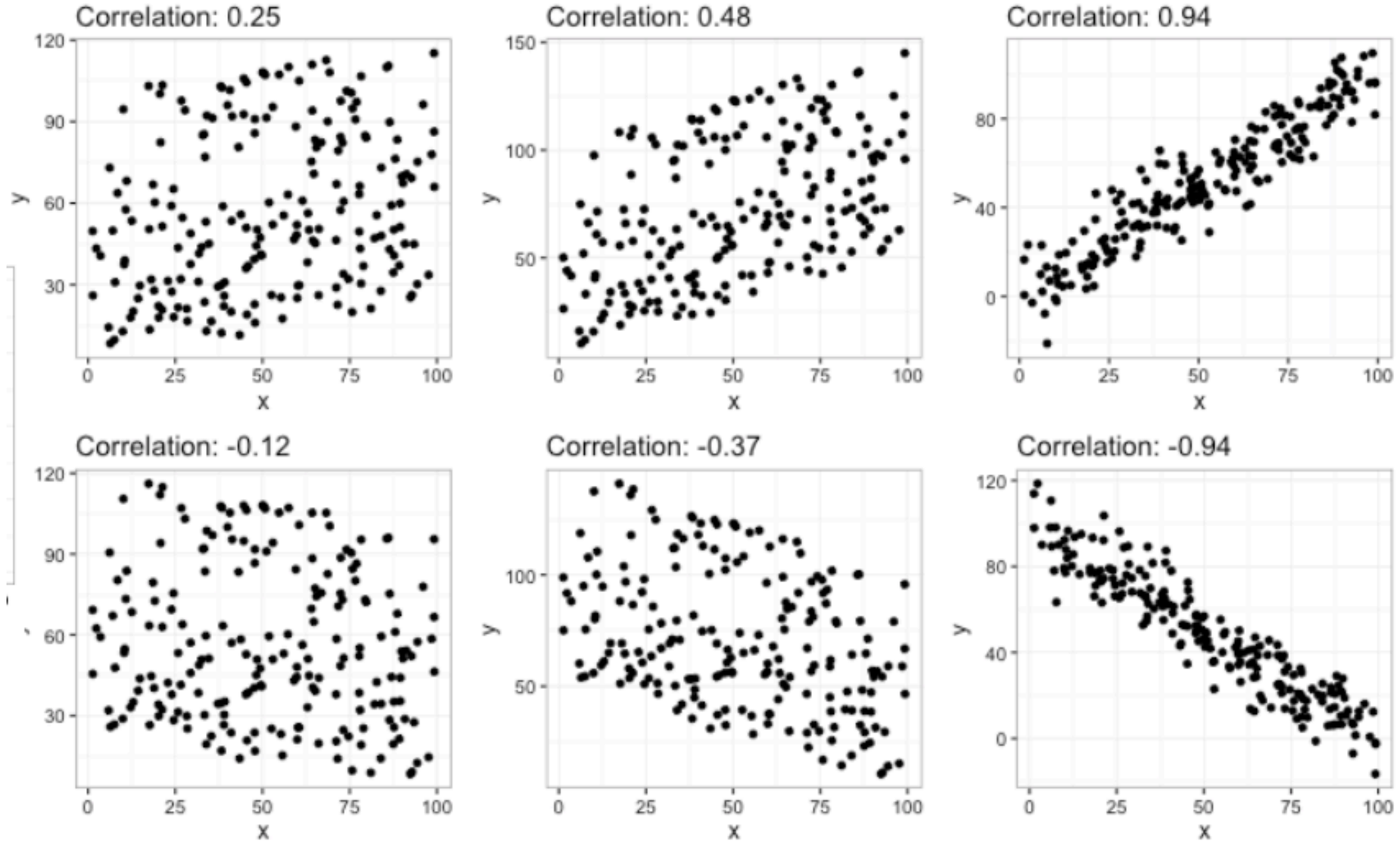


Correlations between Datasets

Correlations

- Measurement of the relationship between **two continuous variables**
- Output is between -1 and 1, with 1 being perfect correlation, -1 is perfect negative correlation, 0 is no correlation.
- Correlation measurement is referred to as r.

Correlations



Correlations

```
series1.corr( series2 )
```

In Class Exercise:

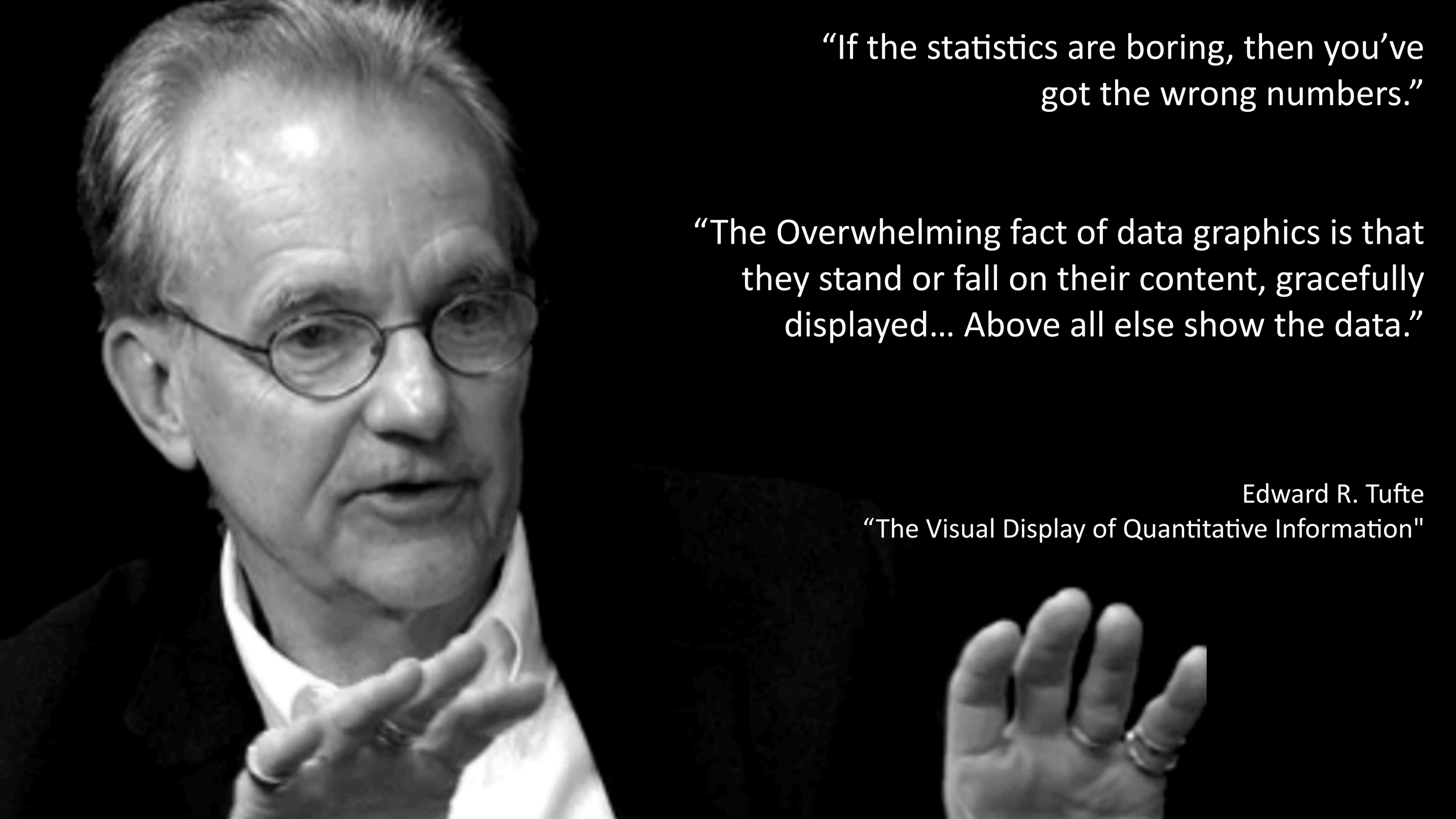
Please complete Worksheet 3: EDA Worksheet

Questions?

Exploring Data Through Visualization

Visualization Goals

- Analyze
- Explore
- Assess
- Determine
- Decide
- Communicate
- Explain
- Present
- Prove
- Persuade



“If the statistics are boring, then you’ve got the wrong numbers.”

“The Overwhelming fact of data graphics is that they stand or fall on their content, gracefully displayed… Above all else show the data.”

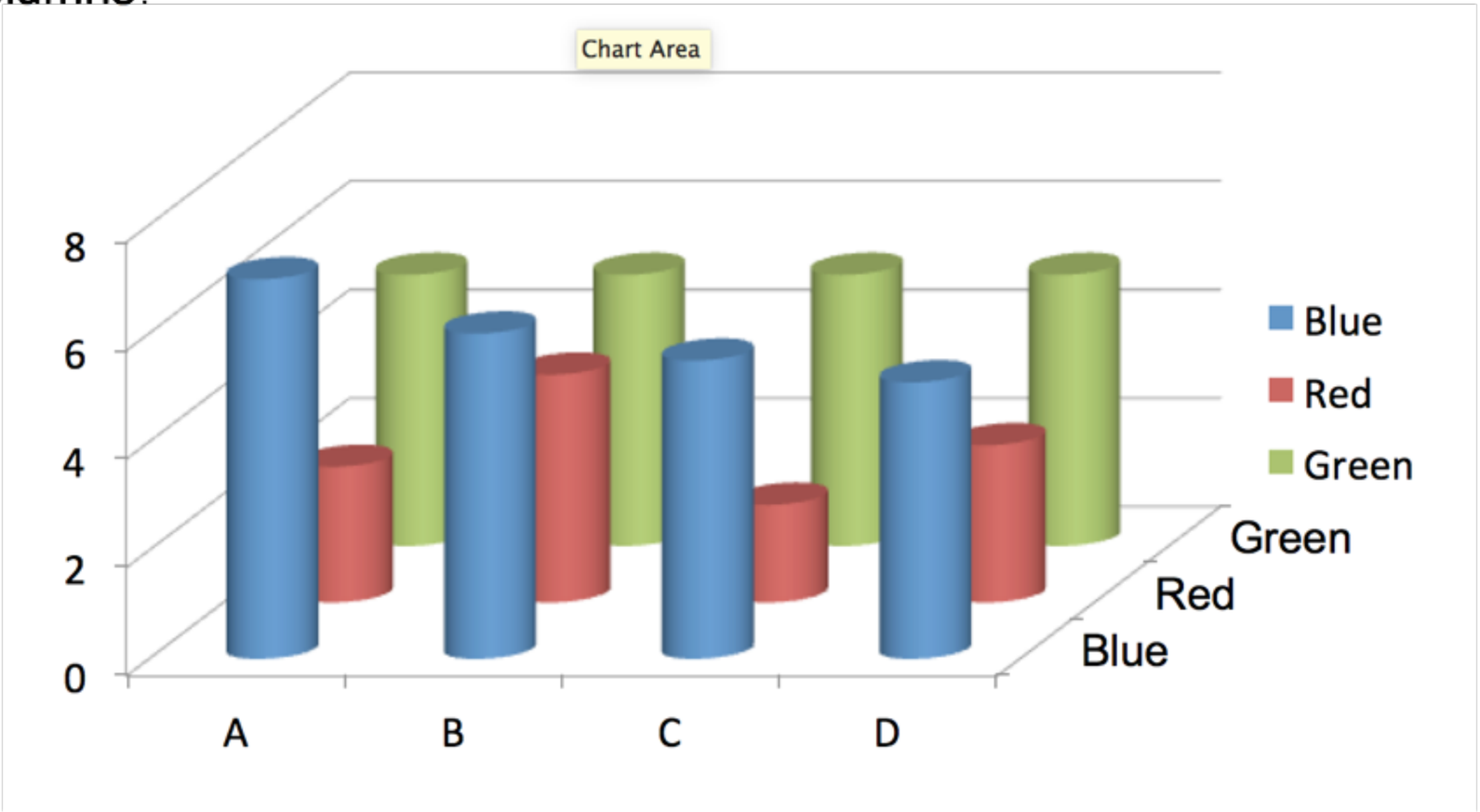
Edward R. Tufte
“The Visual Display of Quantitative Information"

Elements of Good Visualizations

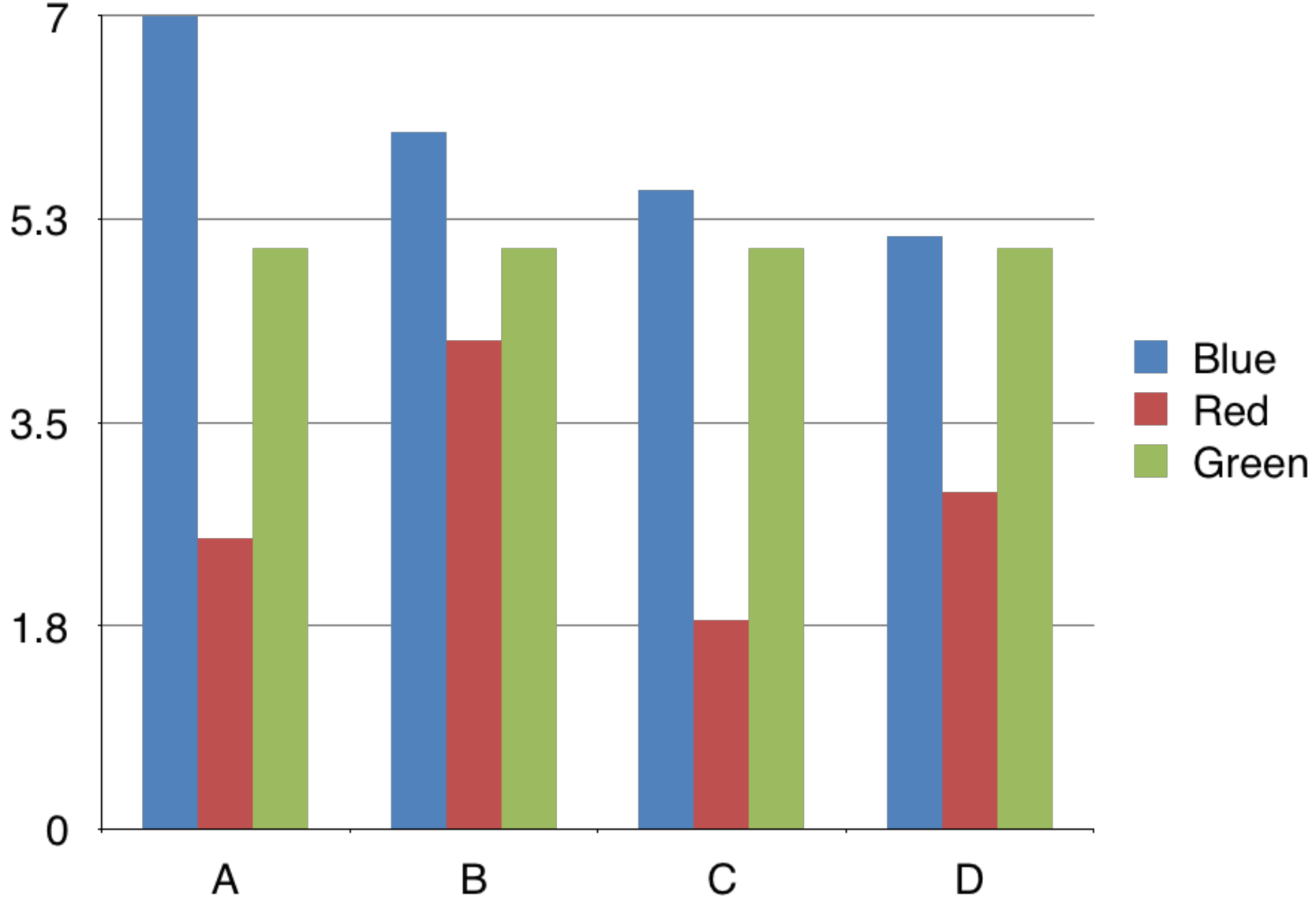
1. Graphical Integrity
2. Simple
3. Proper Display
4. Proper Color
5. Tells a story

Proper Display

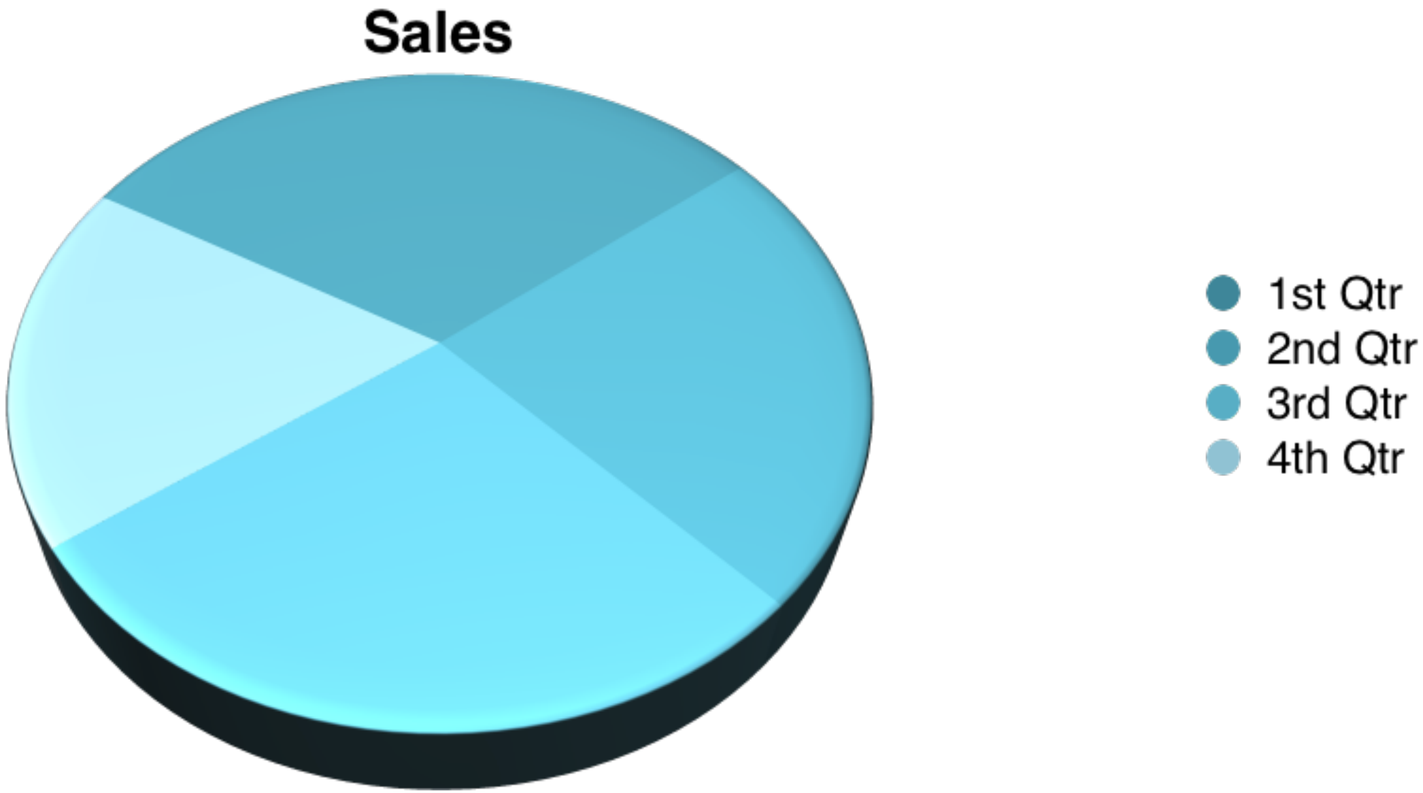
Questions: What is the height of the green columns? For which categories (A,B,C,D) are the blue columns taller than the green columns?



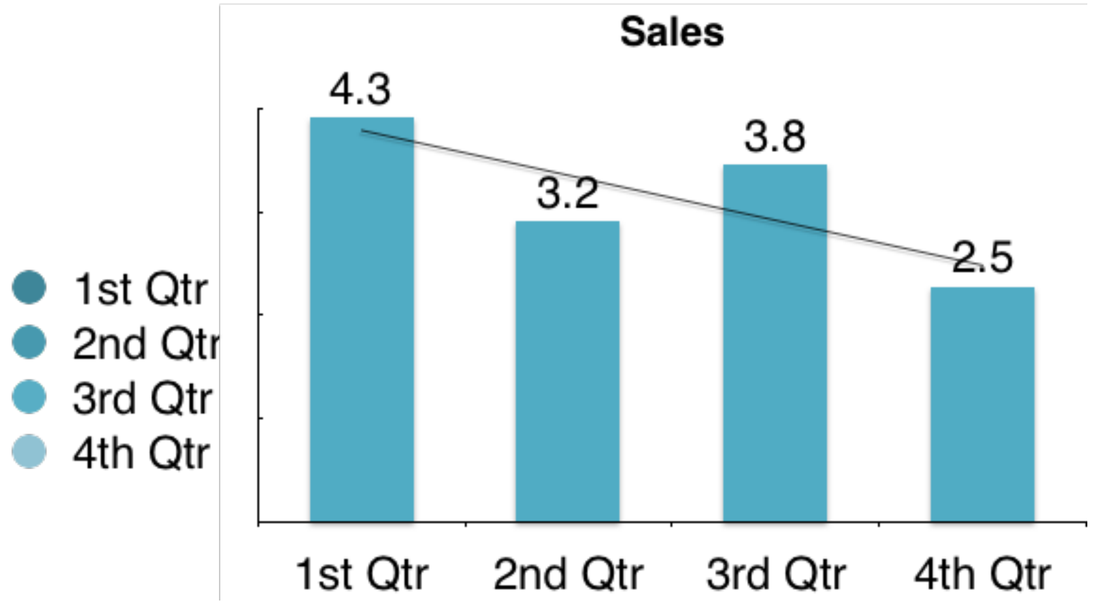
Proper Display



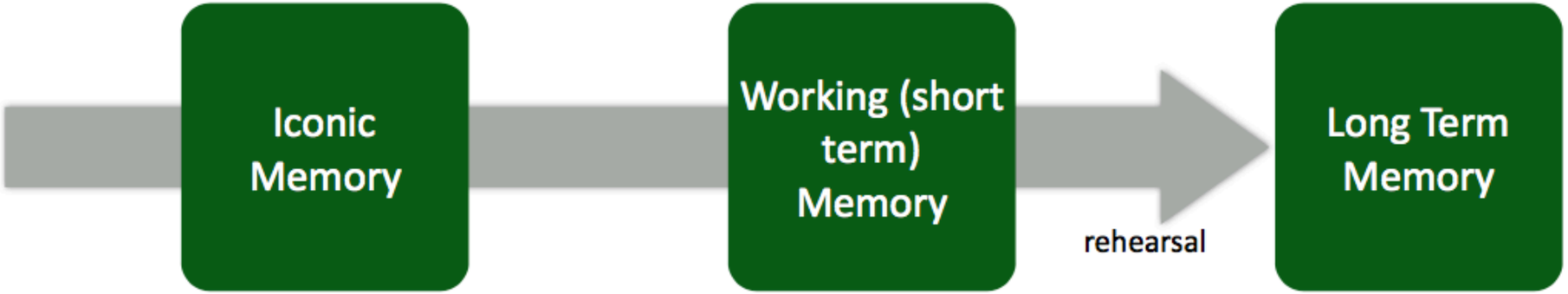
Proper Display



Proper Display



Visual Processing System



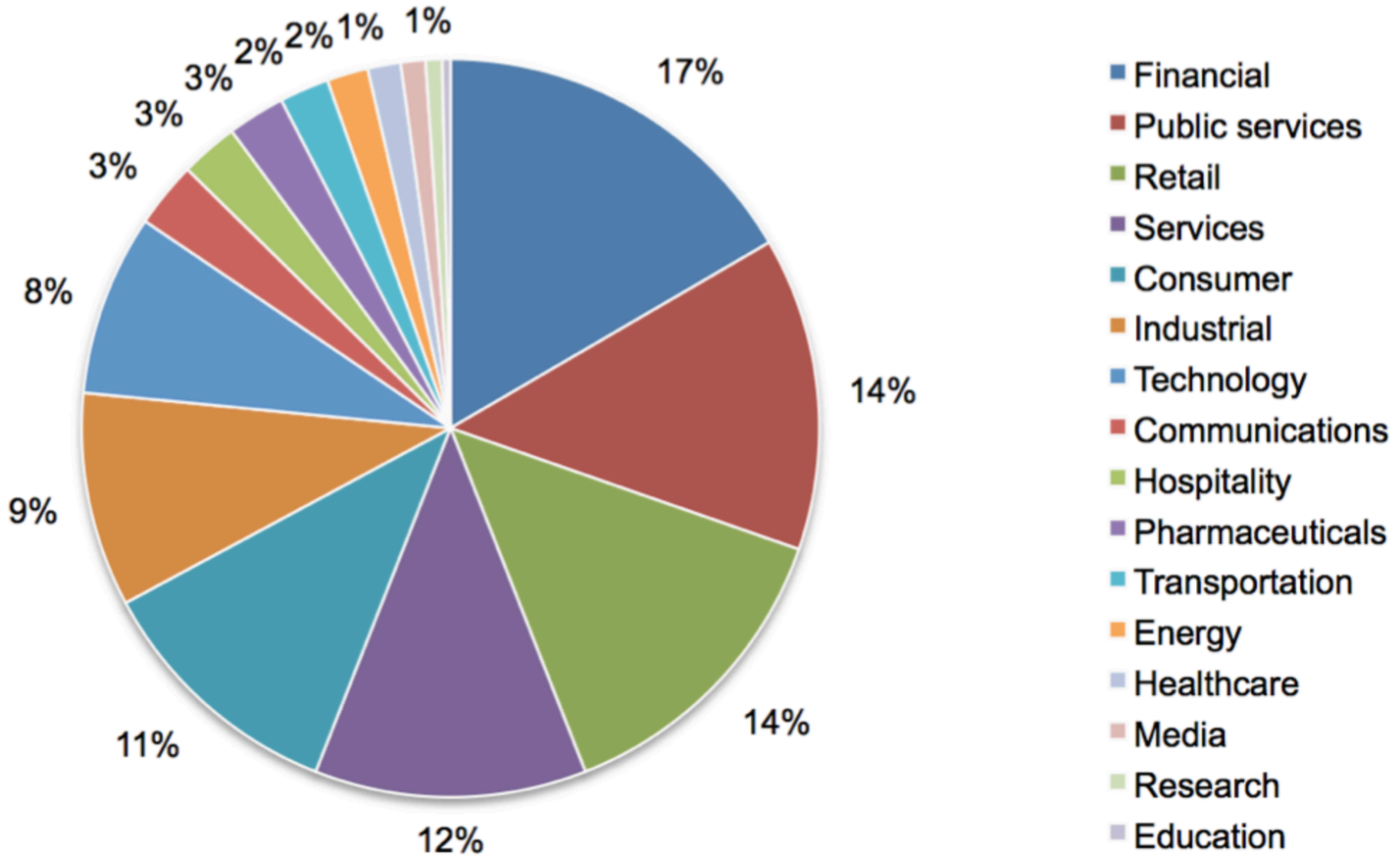
- < 1 second (or until replaced)
- Preattentive processing
- Saccadic movement
- ~20 objects

- Visual groupings
- Flexible, conscious
- 3-5 objects

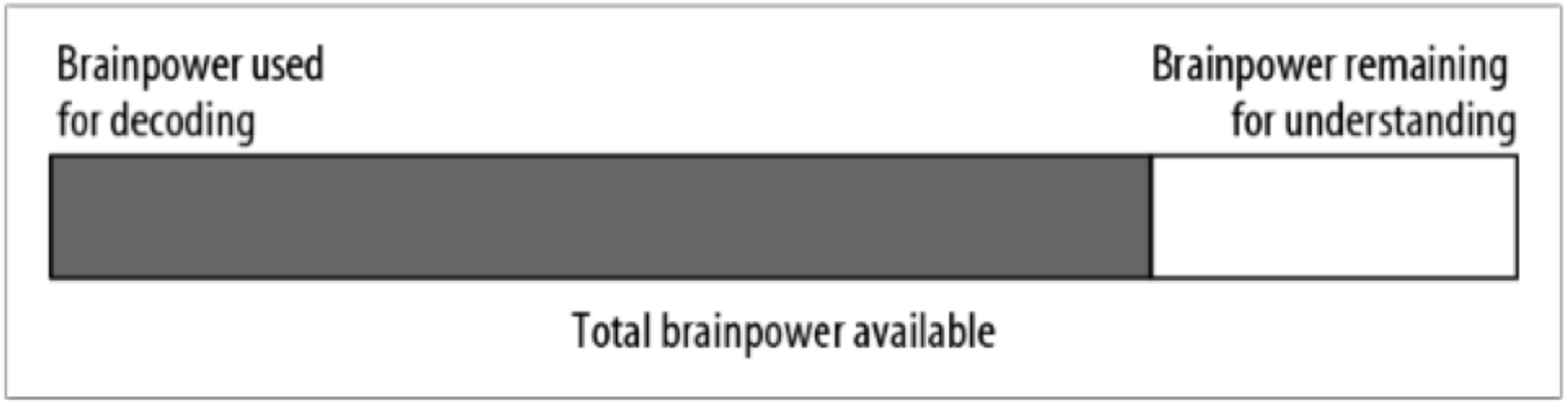
- Affects dataviz through expectations

Overworking Visual Memory

Figure 20. Distribution of the benchmark sample by industry segment
Consolidated (n = 277 organizations)



https://www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf


Brainpower used
for decoding
Brainpower remaining
for understanding
Total brainpower available

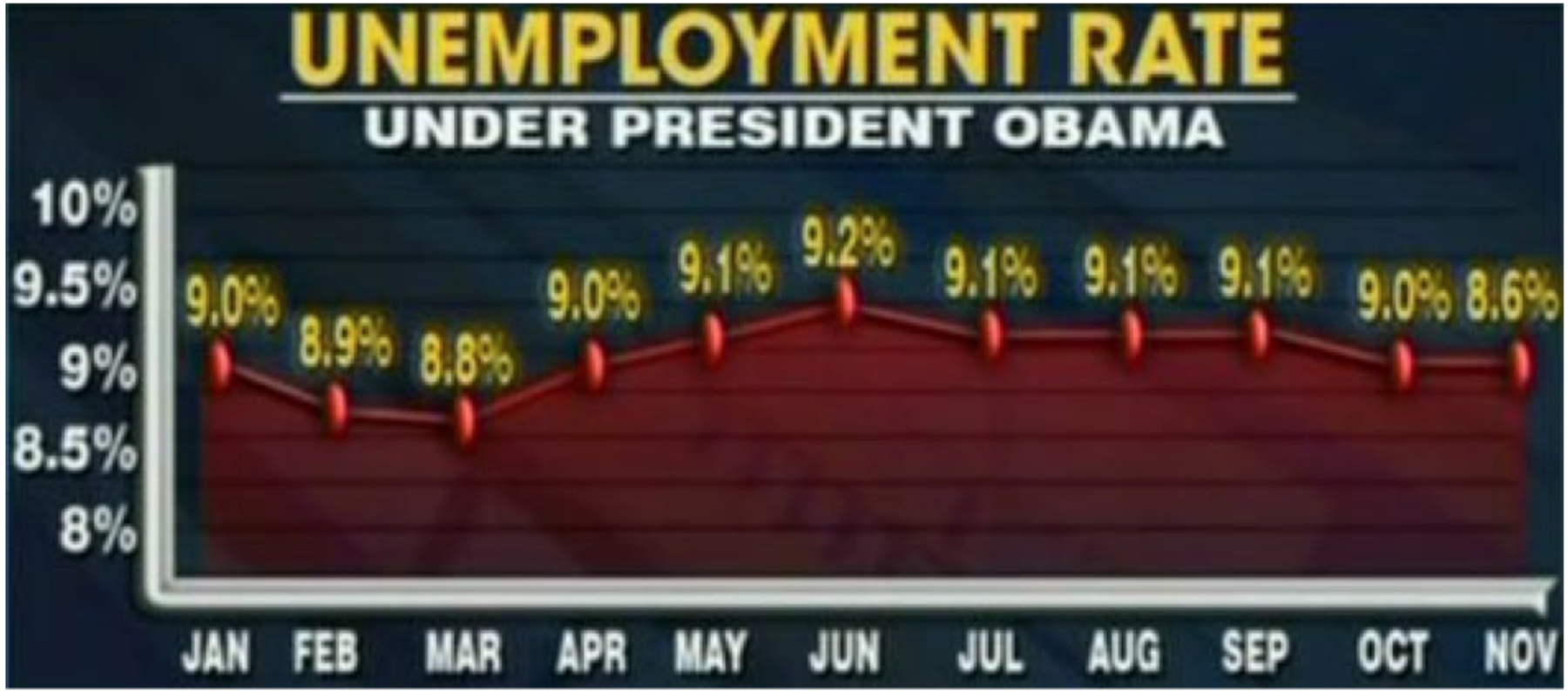
Encode

Noise

Decode

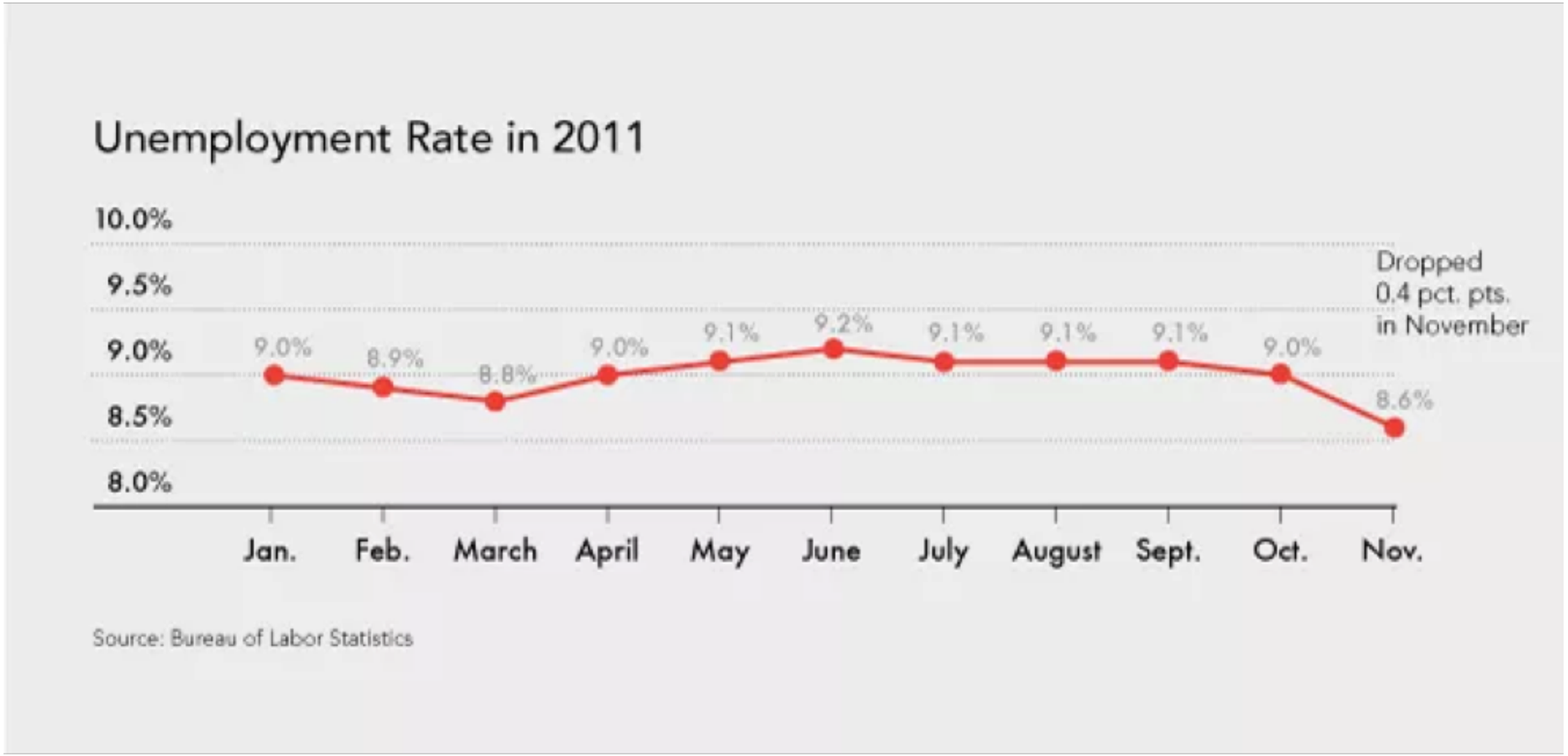
gtkcyber.com

Graphical Integrity



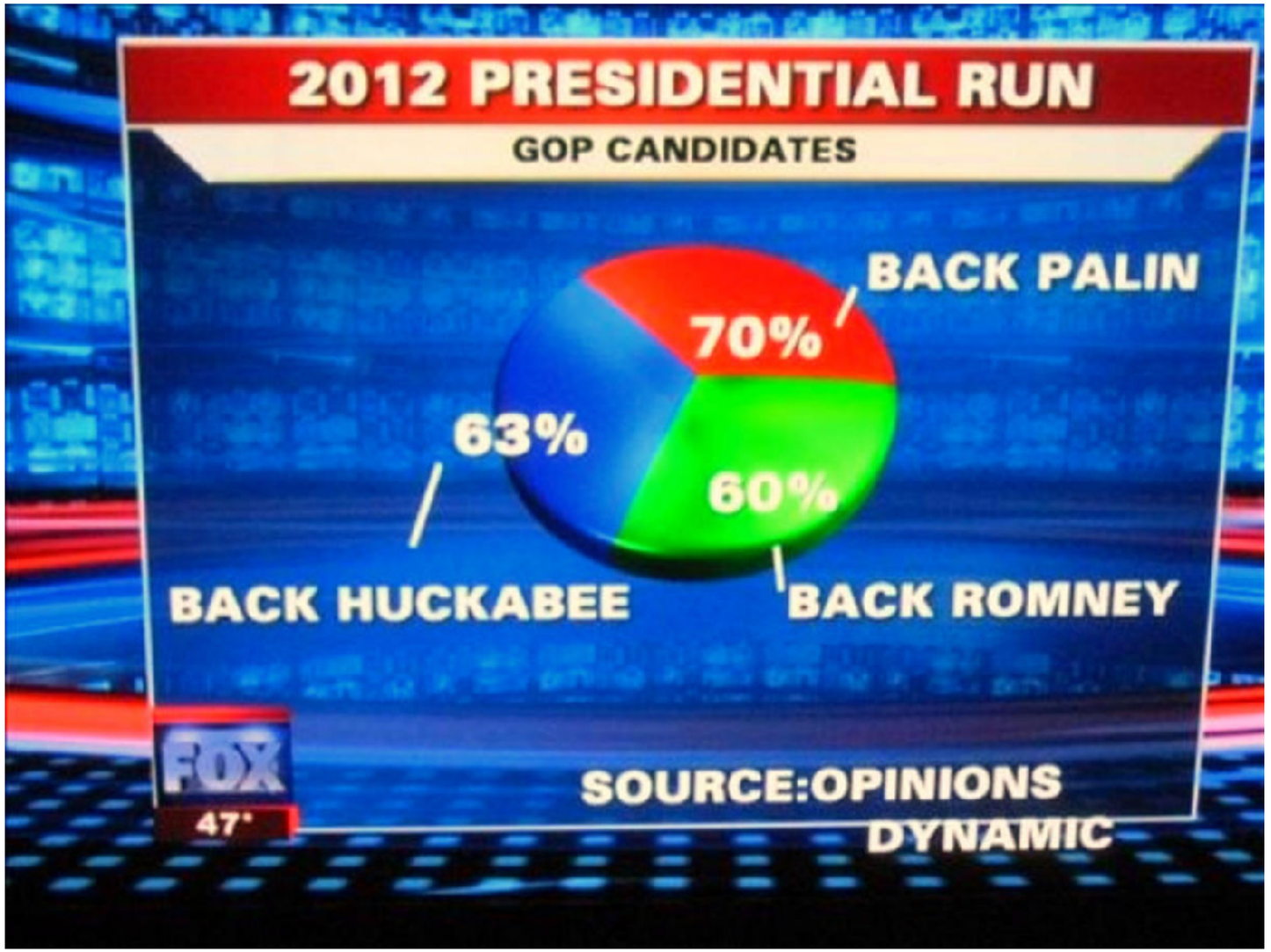
flowingdata.com

Graphical Integrity



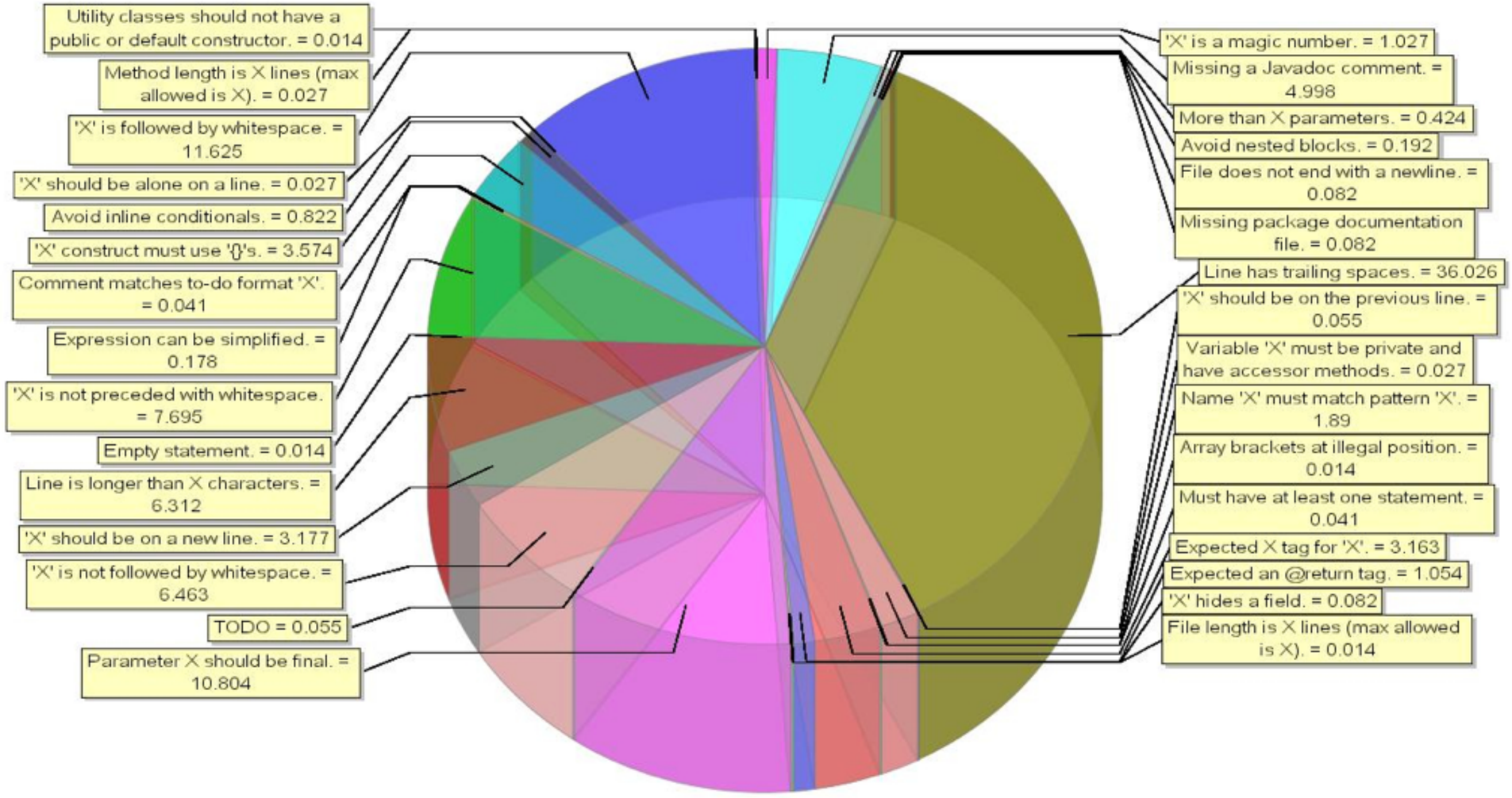
flowingdata.com

Graphical Integrity?

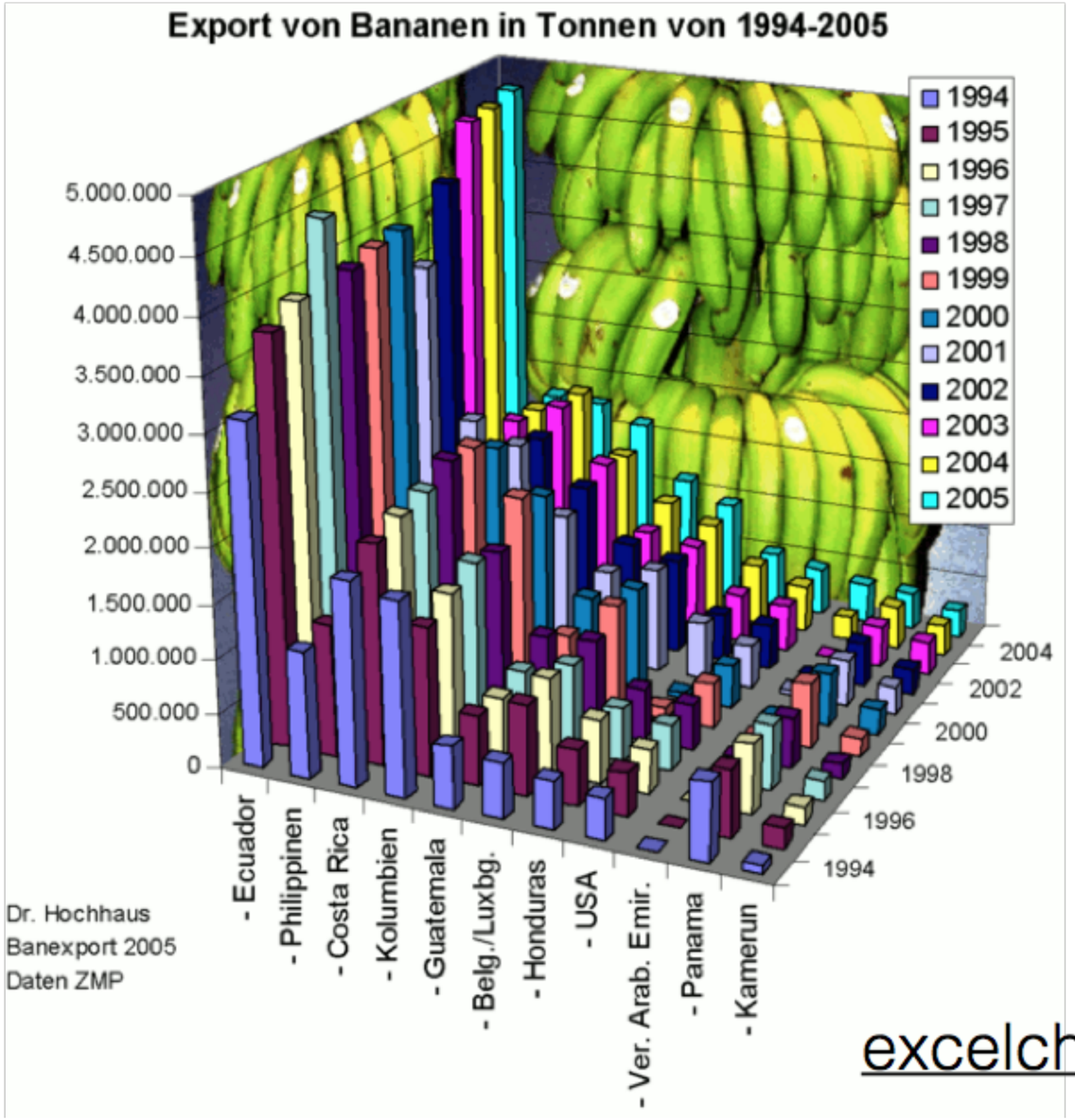


Graphical Integrity

Simple

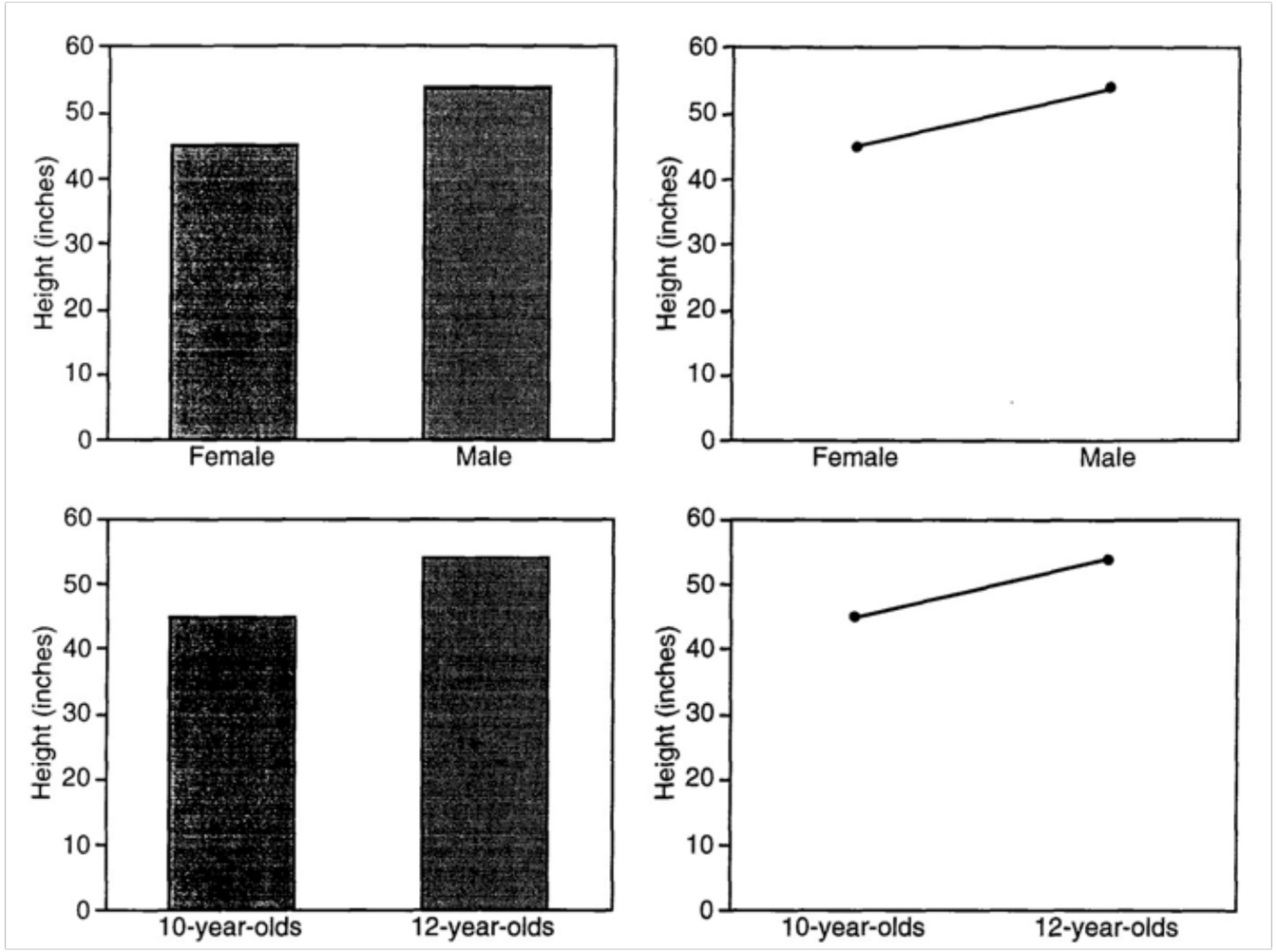

Utility classes should not have a public or default constructor. = 0.014
Method length is X lines (max allowed is X). = 0.027
'X' is followed by whitespace. = 11.625
'X' should be alone on a line. = 0.027
Avoid inline conditionals. = 0.822
'X' construct must use '{}'s. = 3.574
Comment matches to-do format 'X'. = 0.041
Expression can be simplified. = 0.178
'X' is not preceded with whitespace. = 7.695
Empty statement. = 0.014
Line is longer than X characters. = 6.312
'X' should be on a new line. = 3.177
'X' is not followed by whitespace. = 6.463
TODO = 0.055
Parameter X should be final. = 10.804
'X' is a magic number. = 1.027
Missing a Javadoc comment. = 4.998
More than X parameters. = 0.424
Avoid nested blocks. = 0.192
File does not end with a newline. = 0.082
Missing package documentation file. = 0.082
Line has trailing spaces. = 36.026
'X' should be on the previous line. = 0.055
Variable 'X' must be private and have accessor methods. = 0.027
Name 'X' must match pattern 'X'. = 1.89
Array brackets at illegal position. = 0.014
Must have at least one statement. = 0.041
Expected X tag for 'X'. = 3.163
Expected an @return tag. = 1.054
'X' hides a field. = 0.082
File length is X lines (max allowed is X). = 0.014

Simple



excelcharts.com

Proper Display

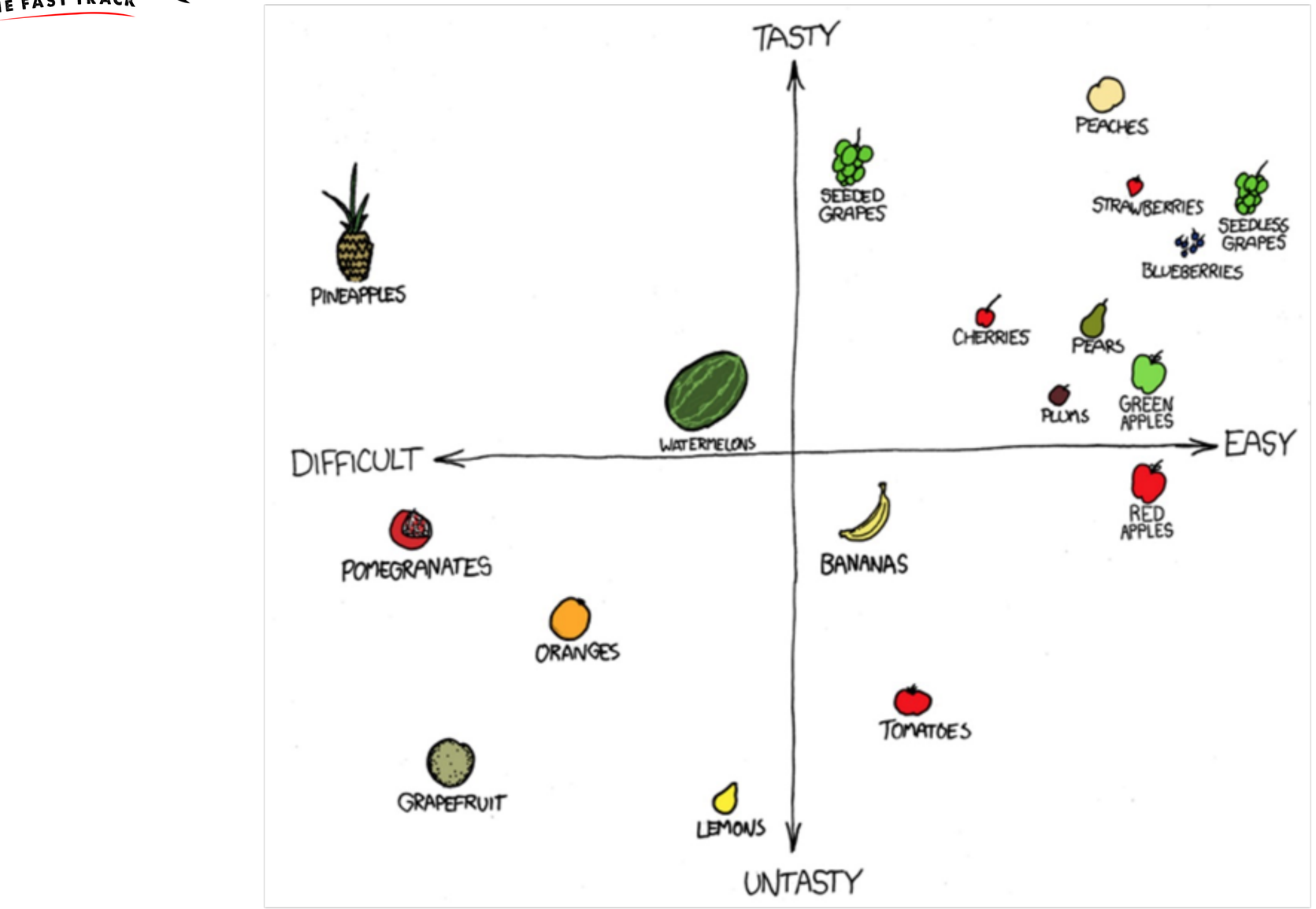


http://lenagroeger.s3.amazonaws.com/papers/BarsLines.pdf

Proper Display

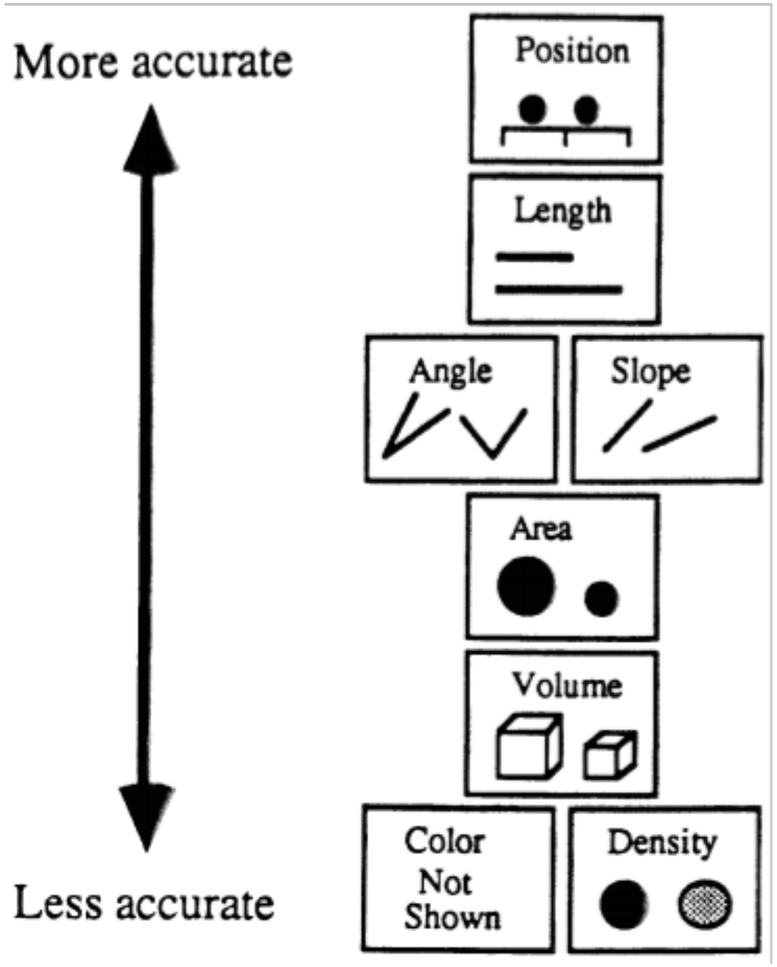
Google Finance

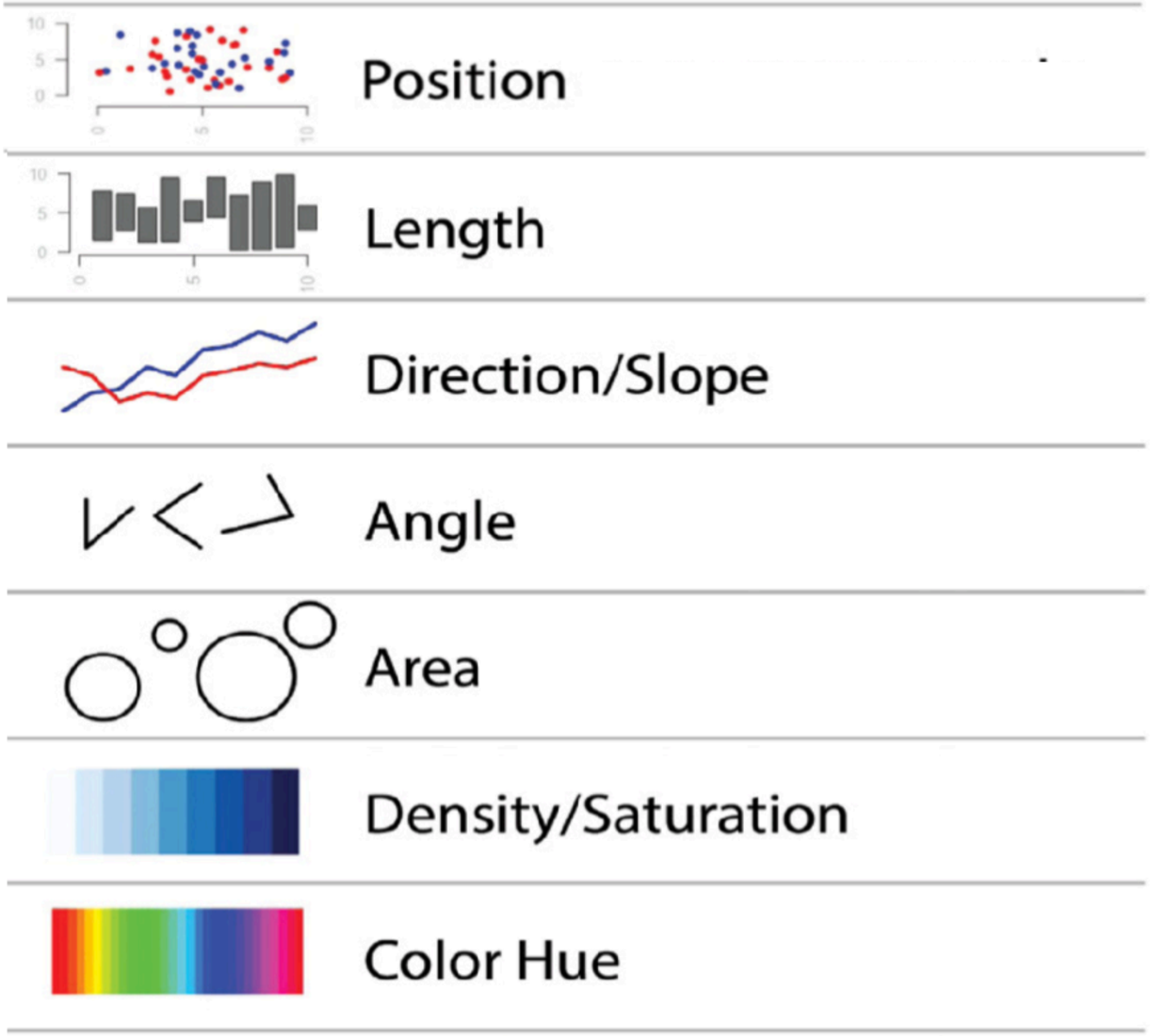
Proper Display



xkcd.com/388

Visual Encoding


More accurate
Position
Length
Angle
Slope
Area
Volume
Color
Not
Shown
Density
Less accurate


Position
Length
Direction/Slope
Angle
Area
Density/Saturation
Color Hue

Gestalt Laws

Law of Similarity:

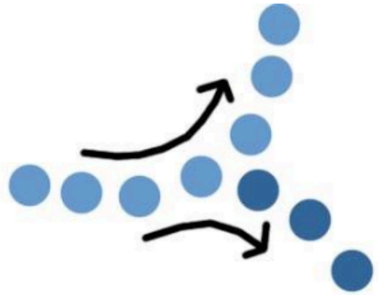
Items that are similar tend to be grouped together.

In the image above, most people see vertical columns of circles and squares.

Law of Pragnanz:

Reality is organized or reduced to the simplest form possible.

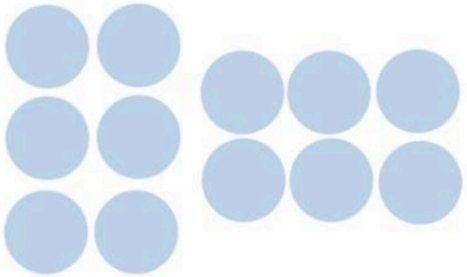
For example, we see the image above as a series of circles rather than as many much more complicated shapes.



Law of Continuity:

Lines are seen as following the smoothest path.

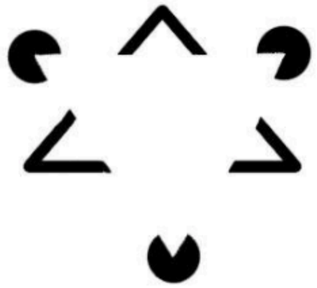
In the image above, the top branch is seen as continuing the first segment of the line. This allows us to see things as flowing smoothly without breaking lines up into multiple parts.



Law of Proximity:

Objects near each other tend to be grouped together.

The circles on the left appear to be grouped in vertical columns, while those on the right appear to be grouped in horizontal rows.



Law of Closure:

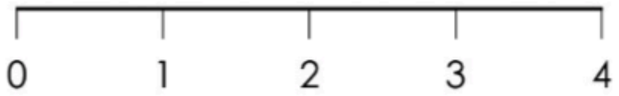
Objects grouped together are seen as a whole.

We tend to ignore gaps and complete contour lines. In the image above, there are no triangles or circles, but our minds fill in the missing information to create familiar shapes and images.

http://psychology.about.com/od/sensationandperception/ss/gestaltlaws.htm

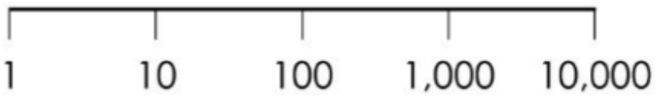
Linear

Values are evenly spaced



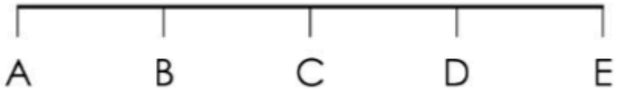
Logarithmic

Focus on percent change



Categorical

Discrete placement in bins

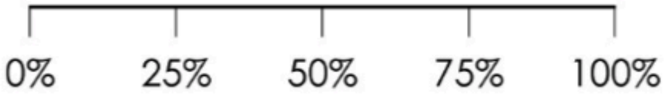


Ordinal

Categories where order matters

Percent

Representing parts of a whole

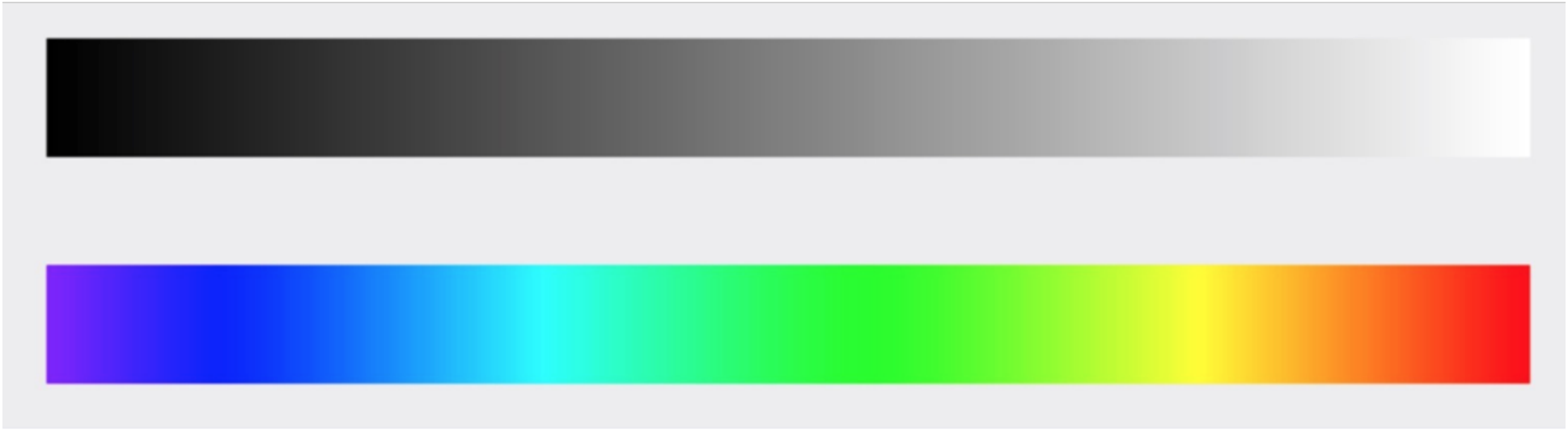


Time

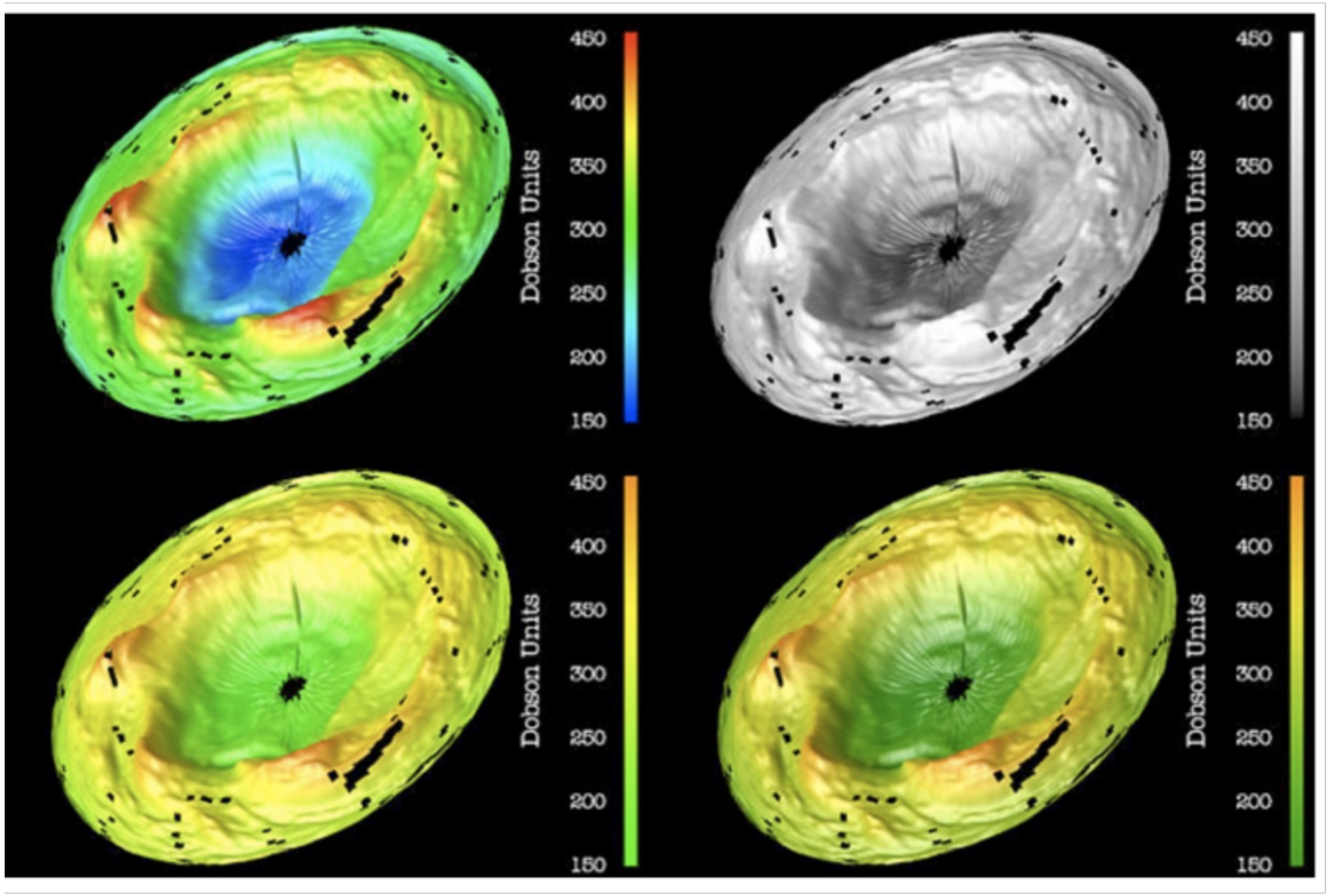
Units of months, days, or hours

Source: Nathan Yau, Data Points

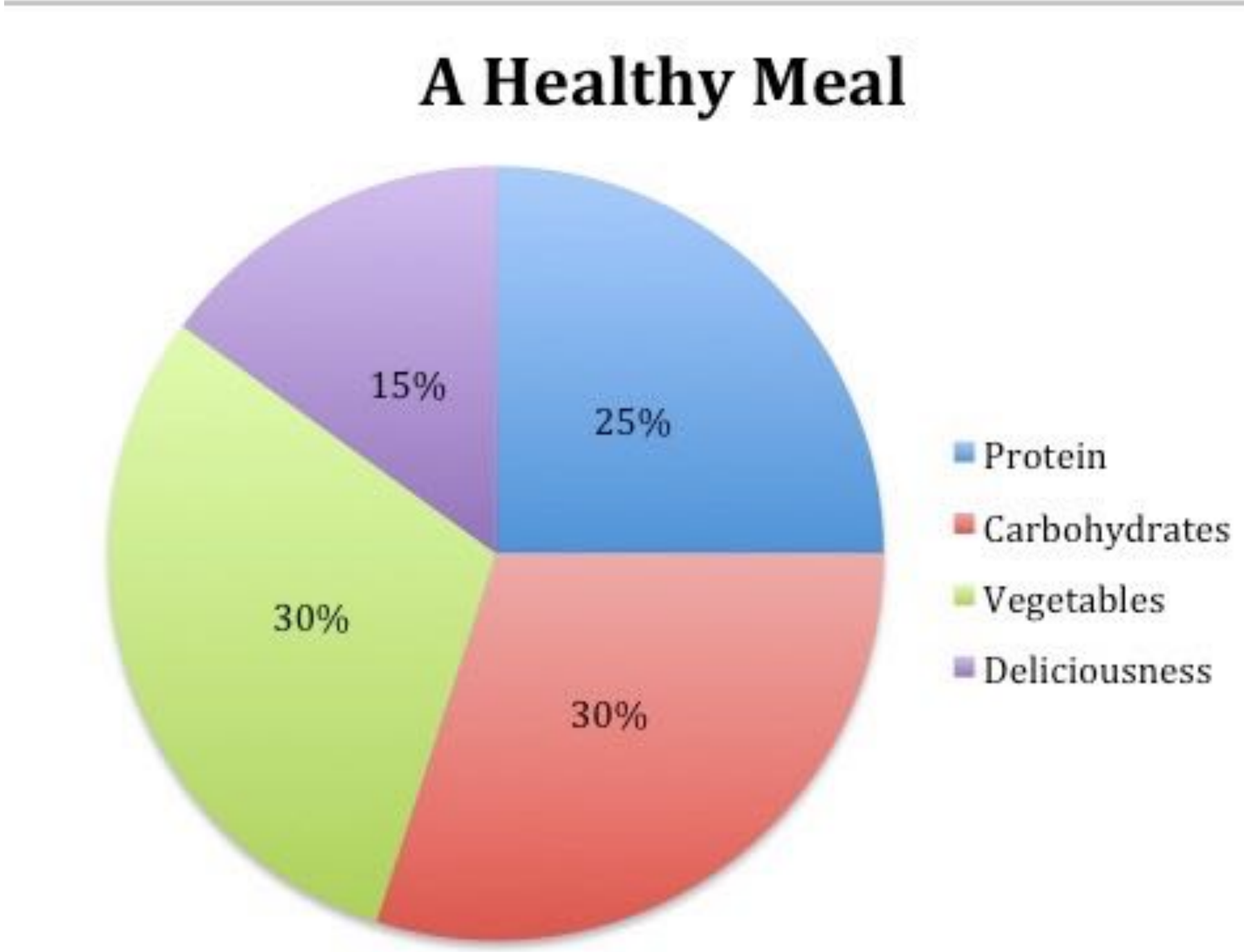
Color

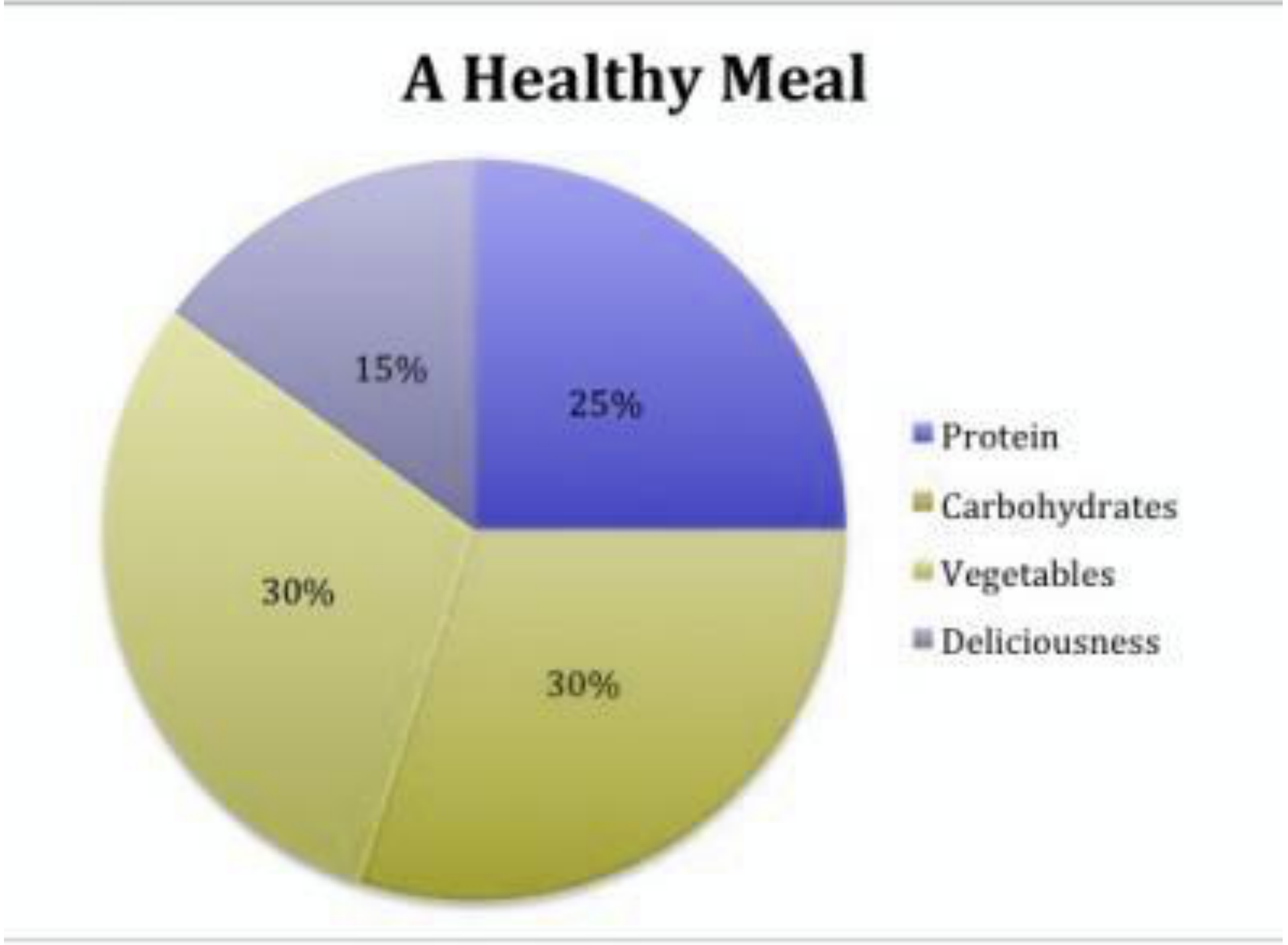


Color



Rogowitz and Treinish


A Healthy Meal
25%
30%
30%
15%
Protein
Carbohydrates
Vegetables
Deliciousness


A Healthy Meal
15%
25%
30%
30%
Protein
Carbohydrates
Vegetables
Deliciousness


elastica
Dashboard
Users
Total (192)
28
High Risk
0
Med Risk
4
Blocked
Policies
Total (231)
3
Blocking
142
Alerting
Policy Alerts
Alerting
443
Blocked
13
Rest
0
Threat Alerts
High Risk
2
Med Risk
2
Low Risk
17
Audited Services
by Users
High Risk (736)
Medium Risk (3k)
Low Risk (3k)
3k
494.3 GB
963k
243

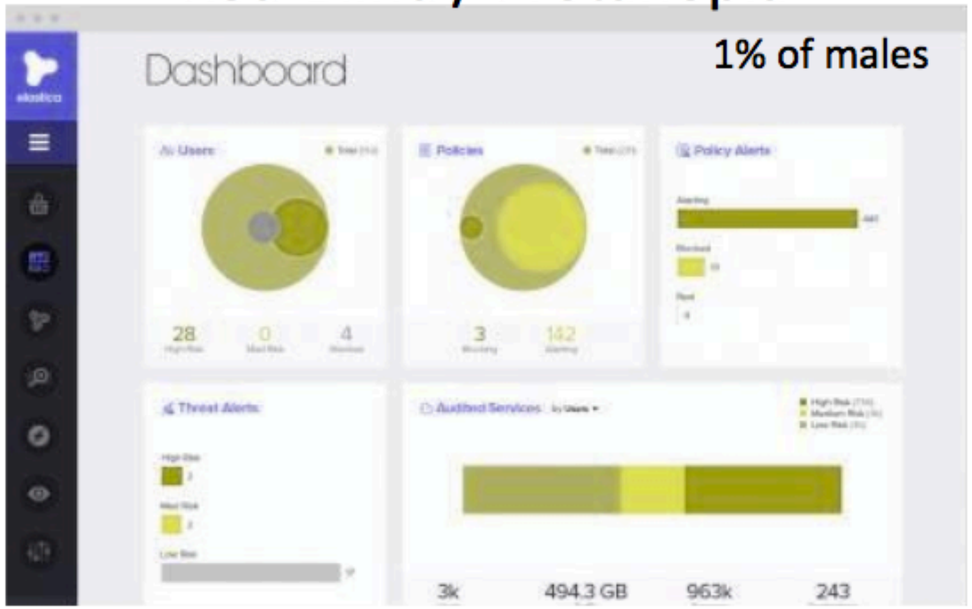
Original

Green-Blind / Deuteranopia

8% of male
0.5% female

Red-Blind / Protanopia

1% of males

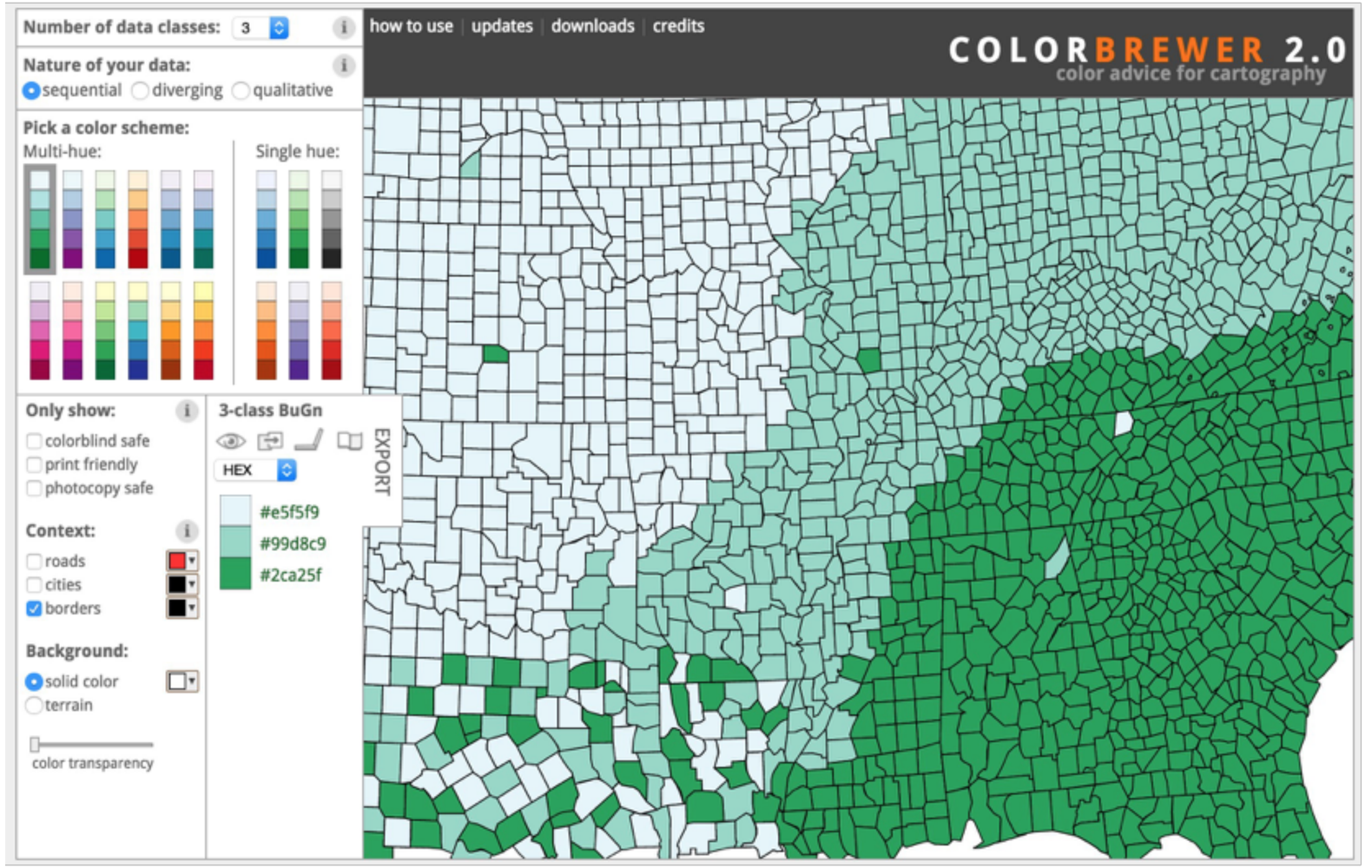


Blue-Blind / Tritanopia

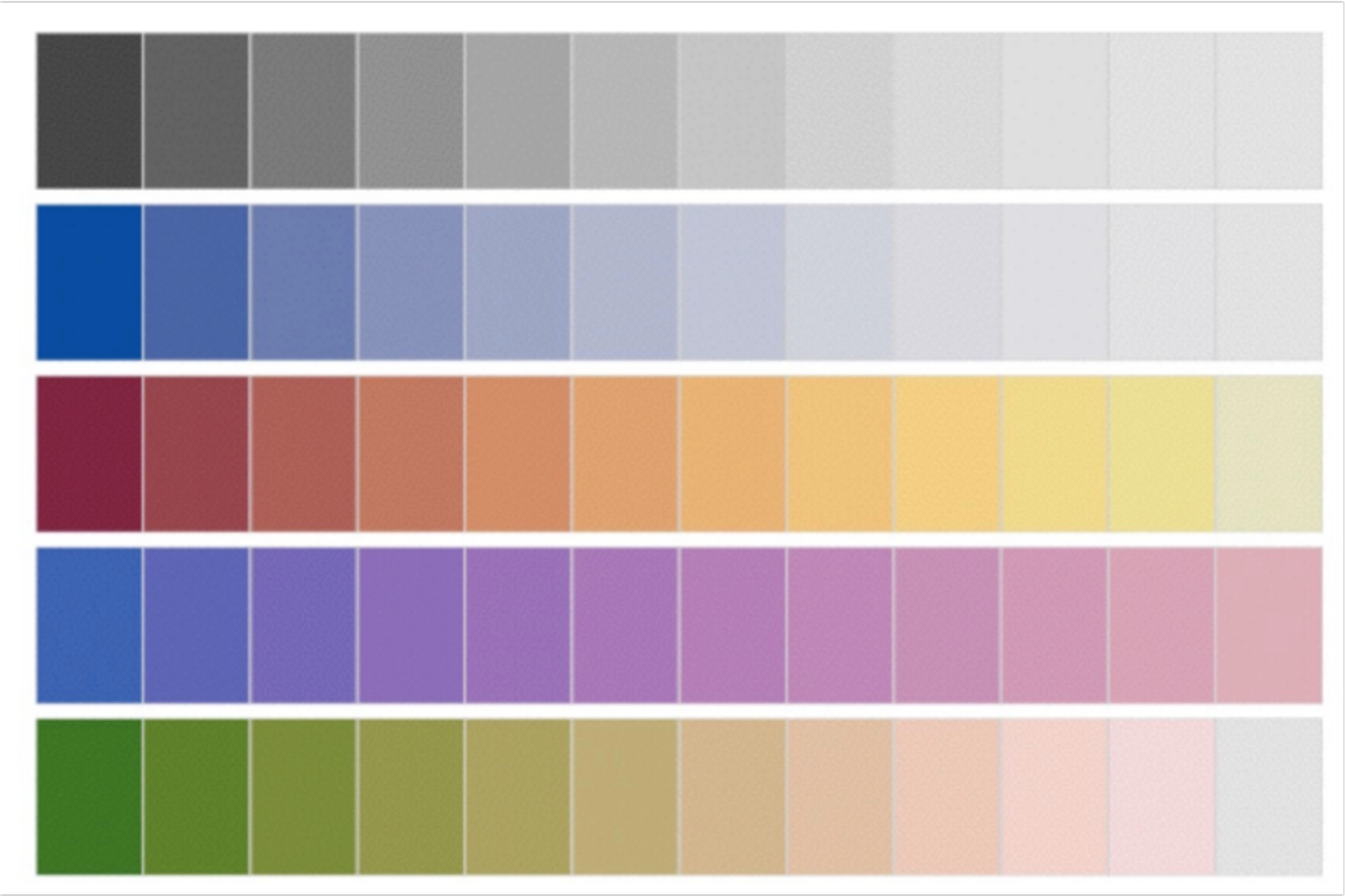
rare

http://www.color-blindness.com/coblis-color-blindness-simulator/

Color

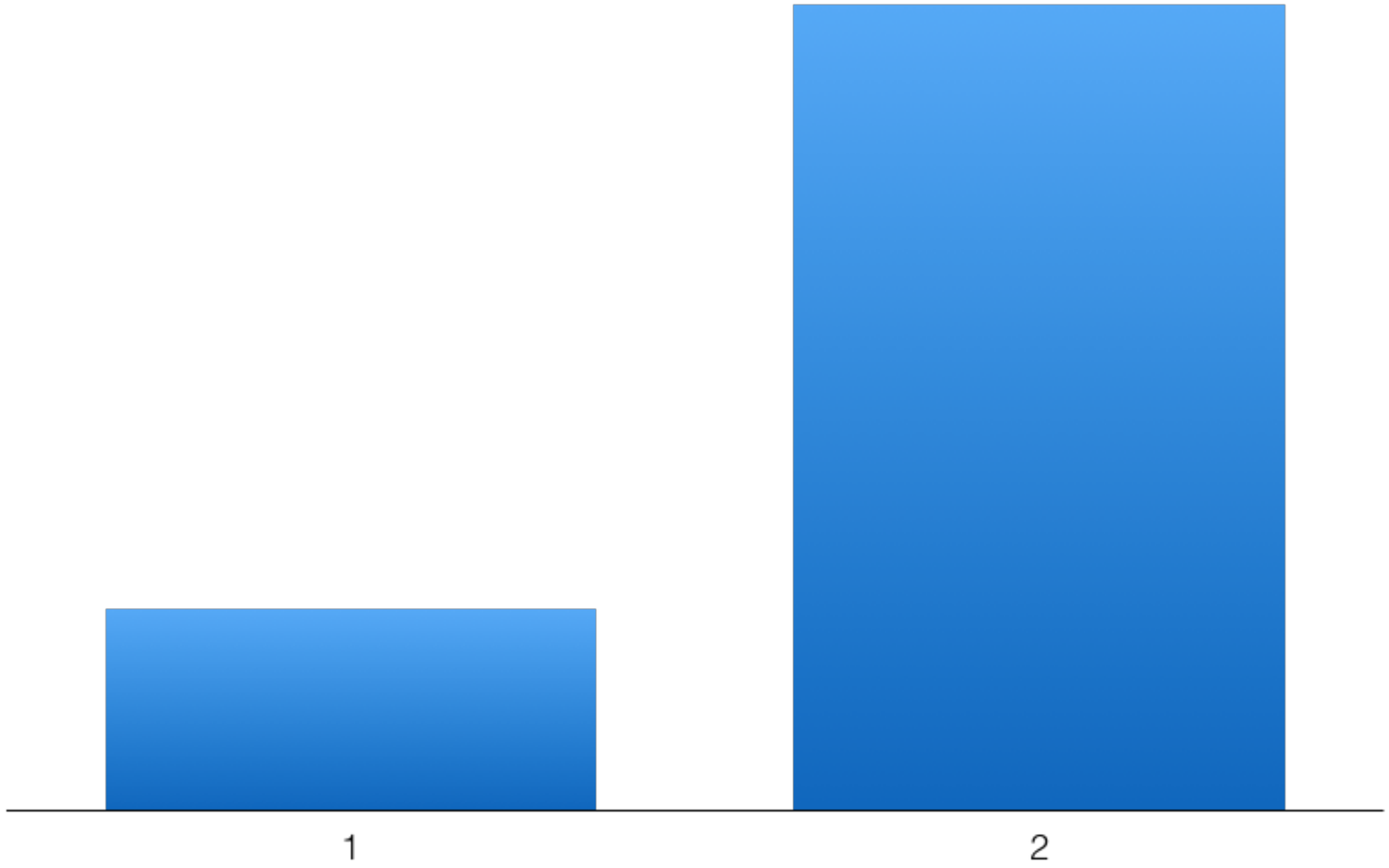

Number of data classes: 3
Nature of your data:
sequential diverging qualitative
Pick a color scheme:
Multi-hue:
Single hue:
Only show:
colorblind safe
print friendly
photocopy safe
Context:
roads
cities
borders
Background:
solid color
terrain
color transparency
3-class BuGn
HEX
EXPORT
#e5f5f9
#99d8c9
#2ca25f
how to use | updates | downloads | credits
COLORBREWER 2.0
color advice for cartography

Color



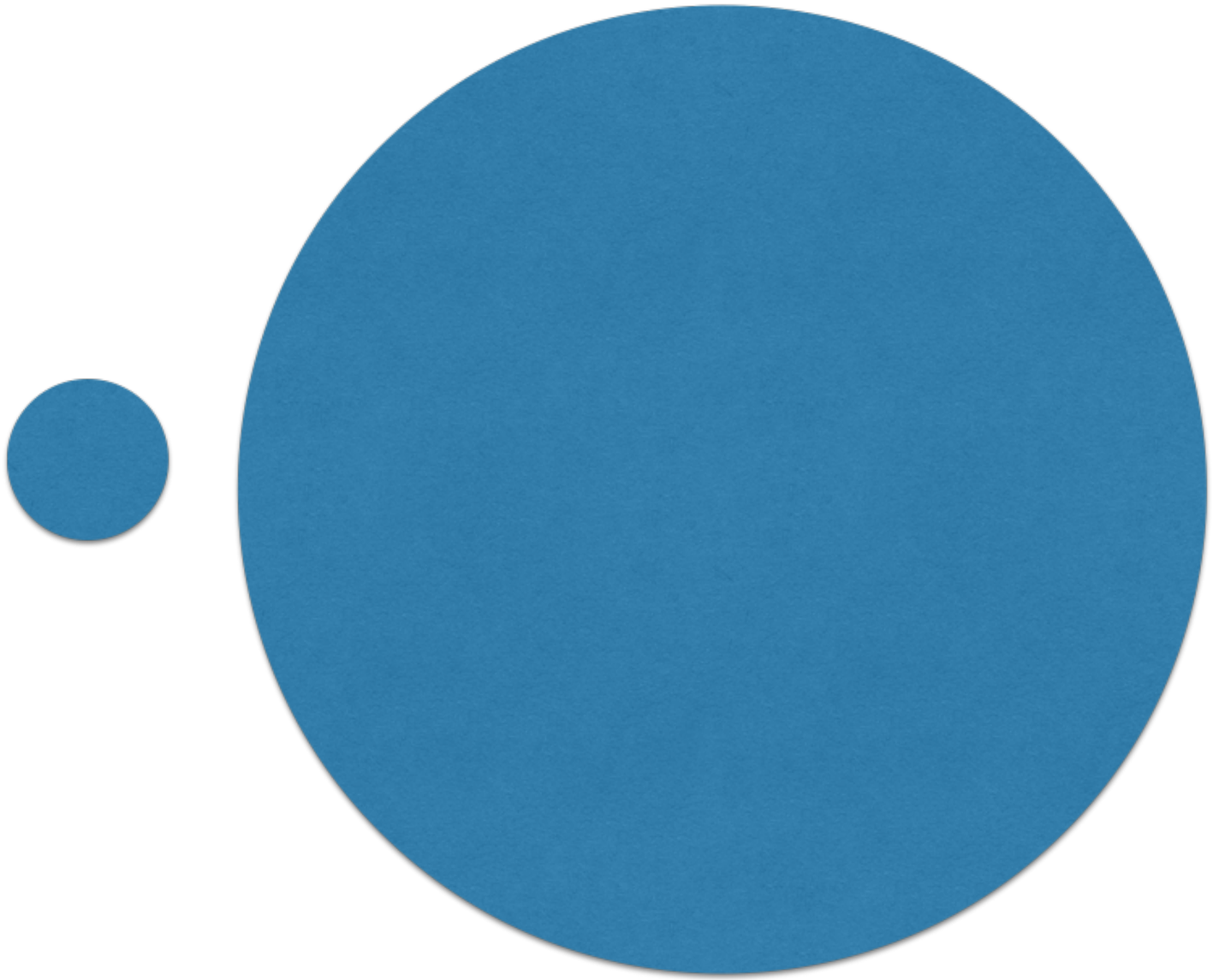
colorbrewer2.org

Height Difference?

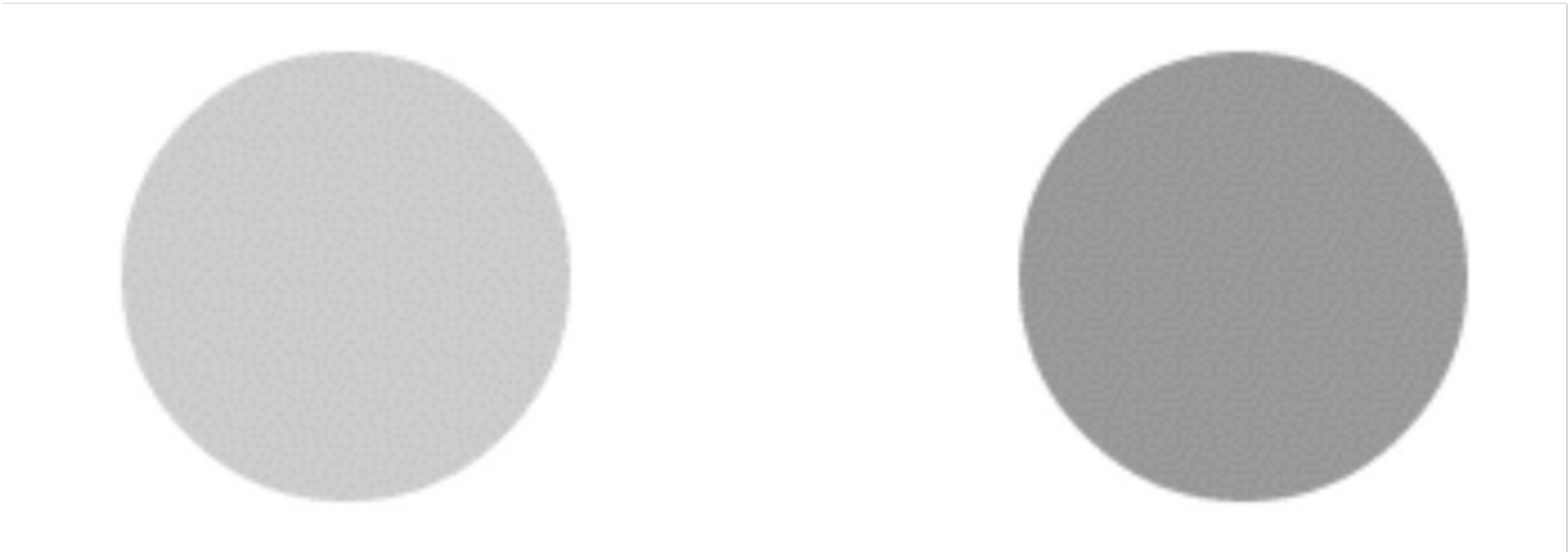


Slope Difference?

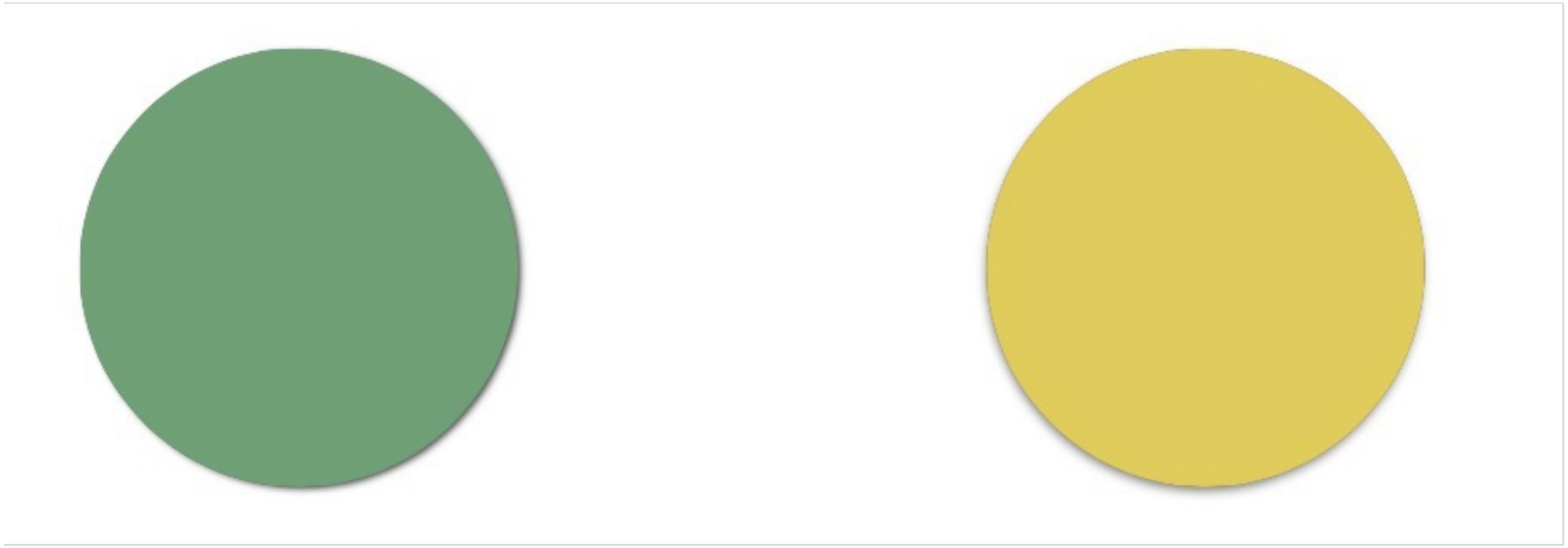
Area Difference?

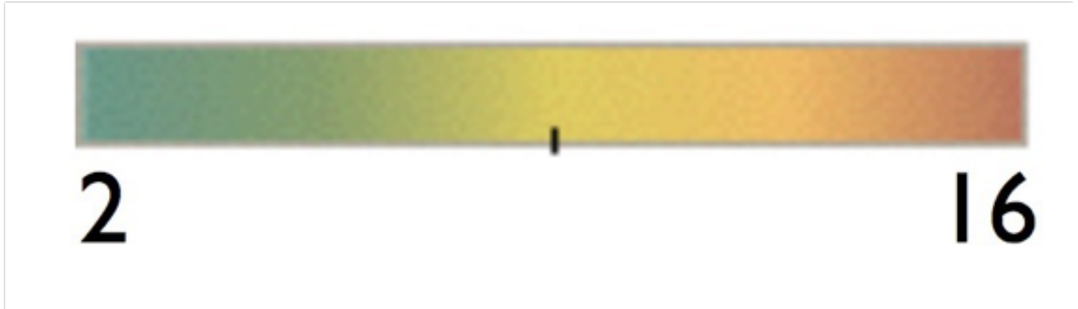


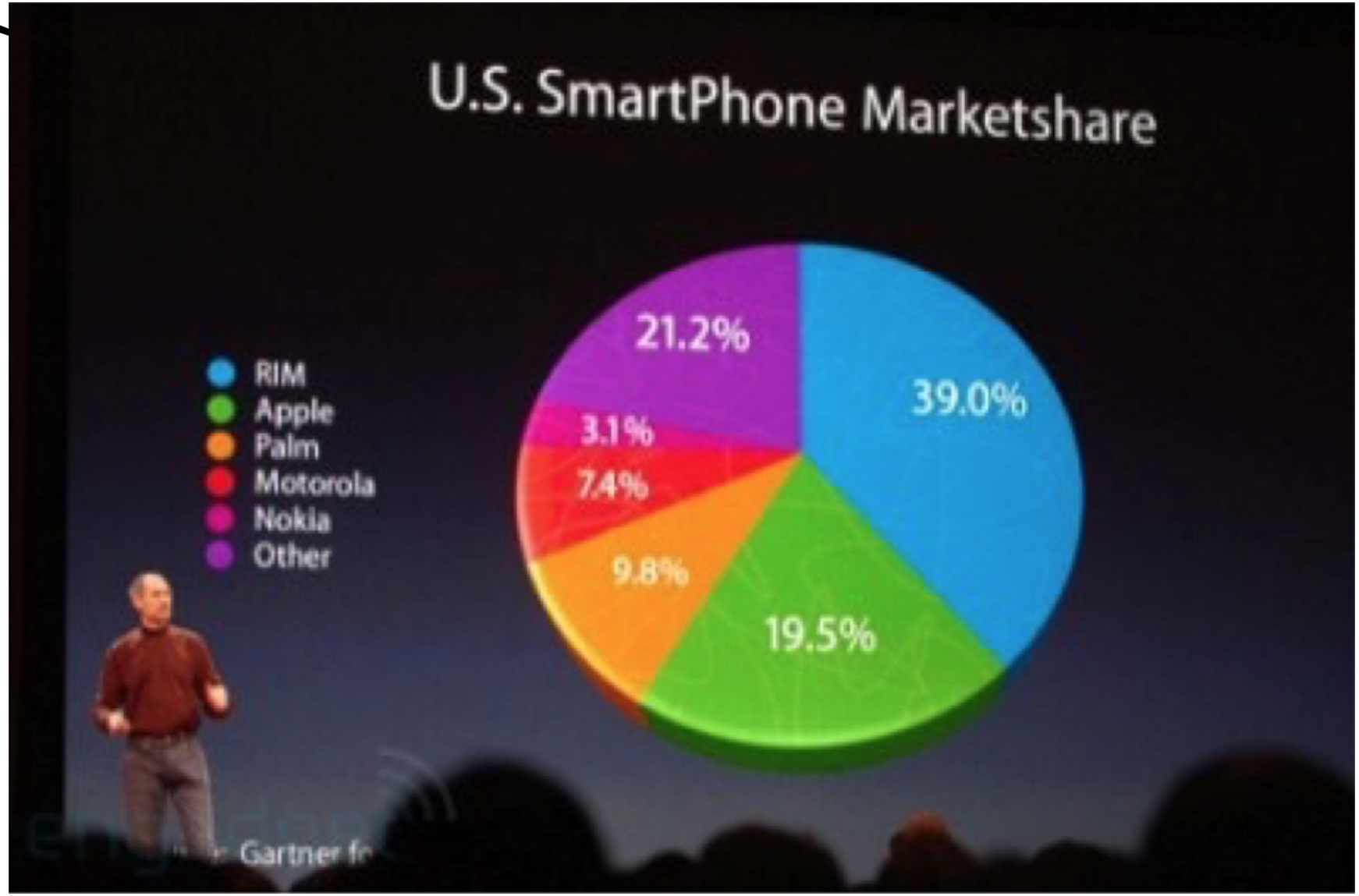
Saturation Difference?



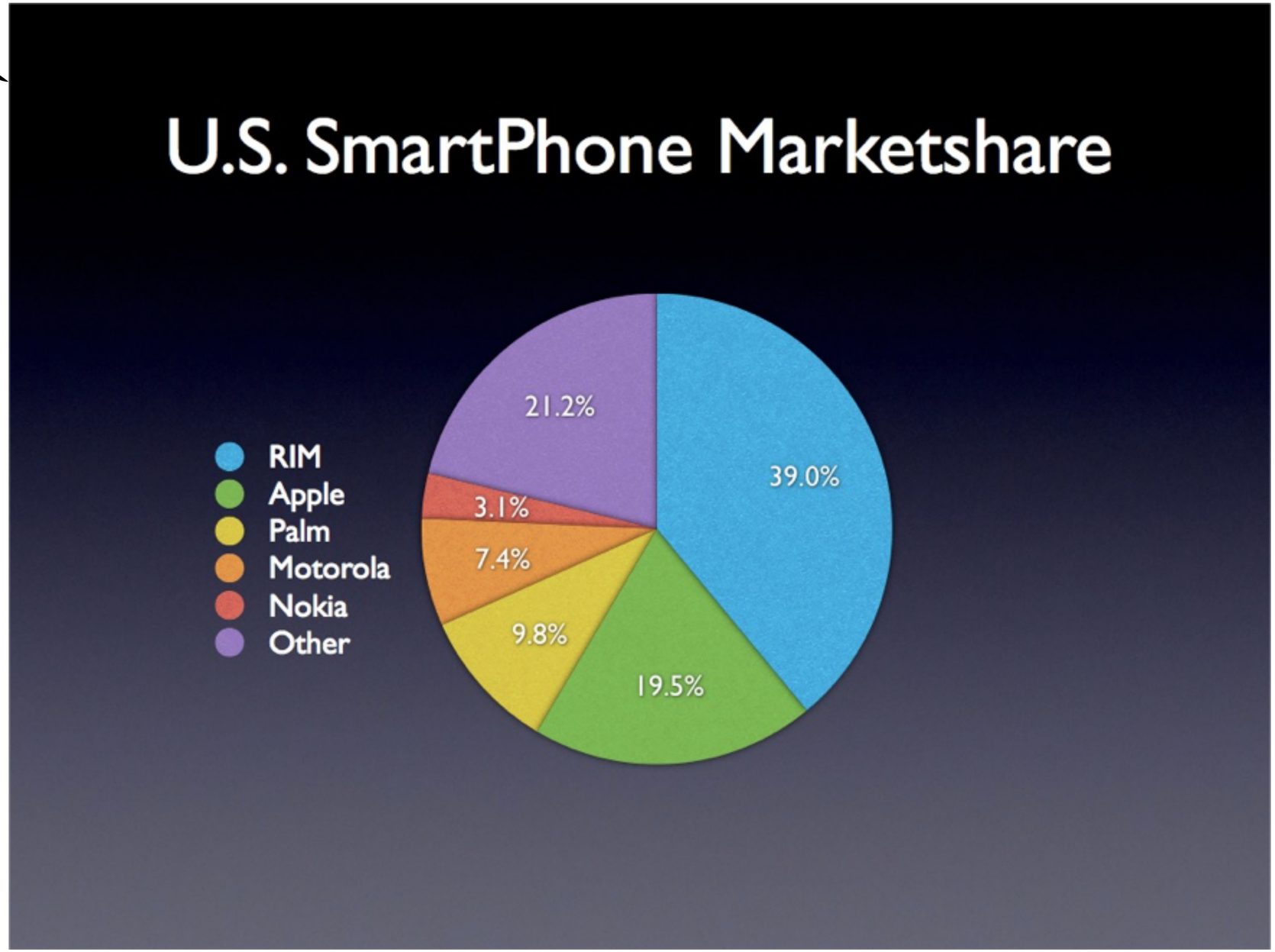
Mapping Difference?



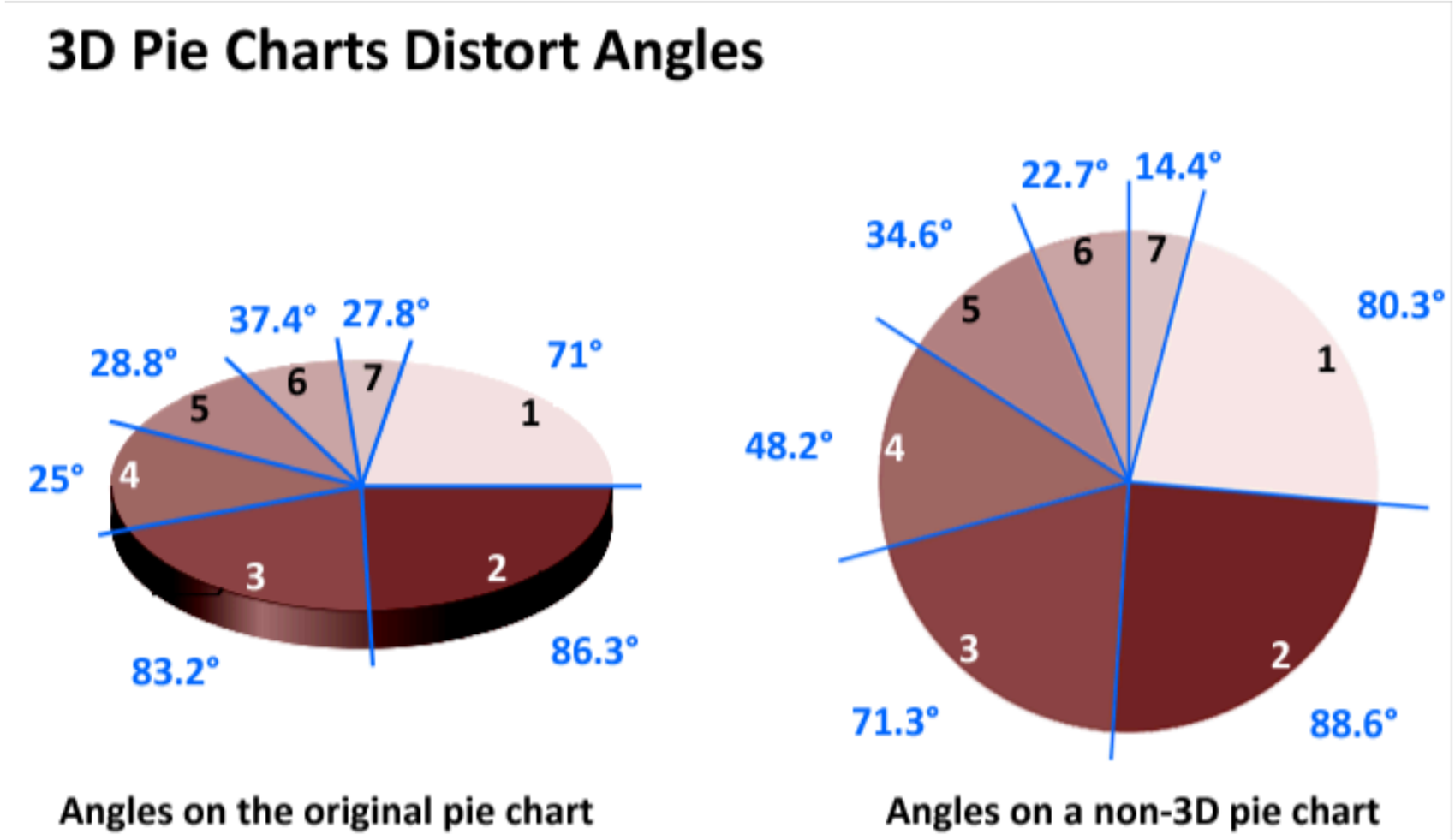



U.S. SmartPhone Marketshare
RIM
Apple
Palm
Motorola
Nokia
Other
39.0%
19.5%
9.8%
7.4%
3.1%
21.2%
Gartner fe

GTK CYBER
GET ON THE FAST TRACK



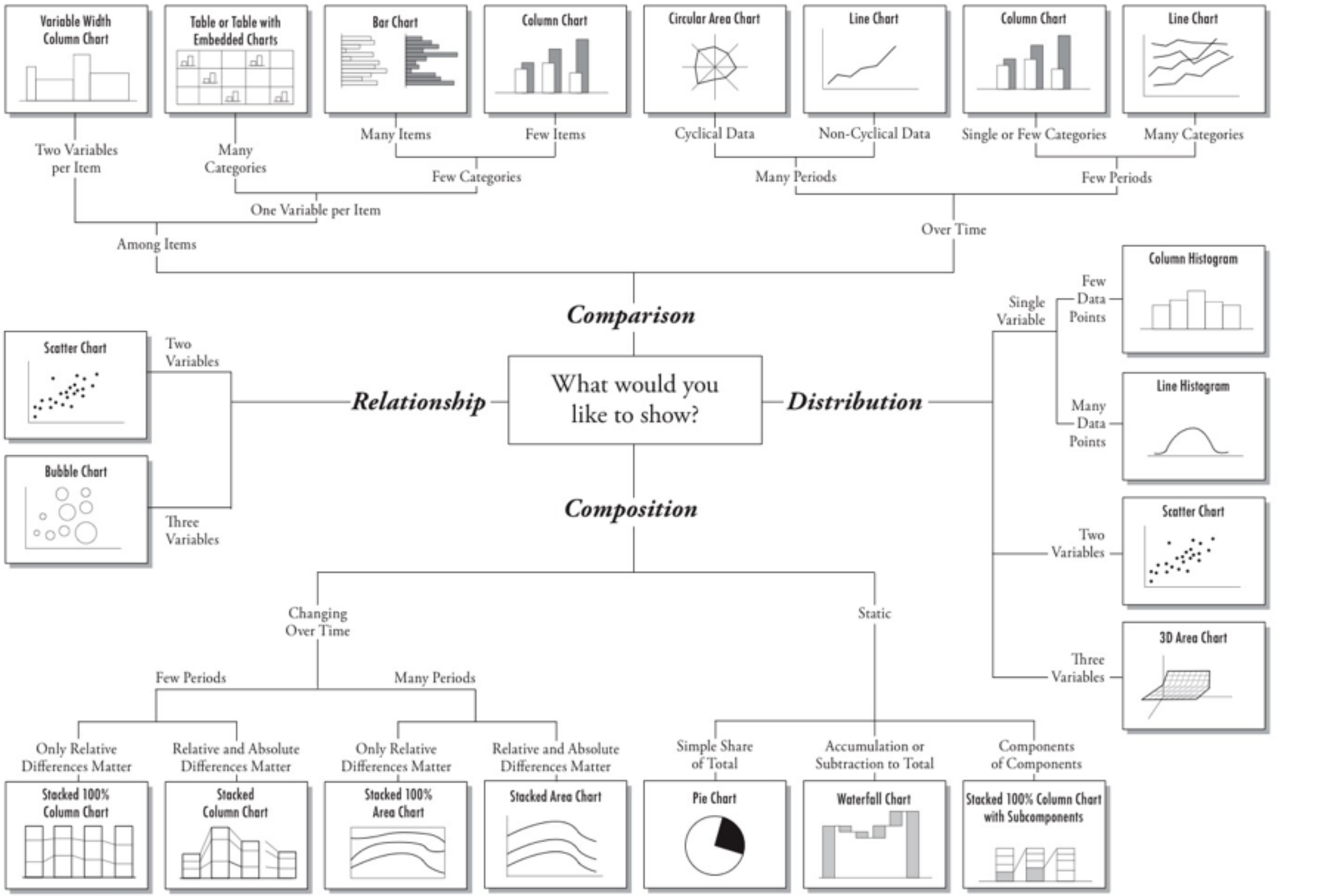
3D Pie Charts are BAD!

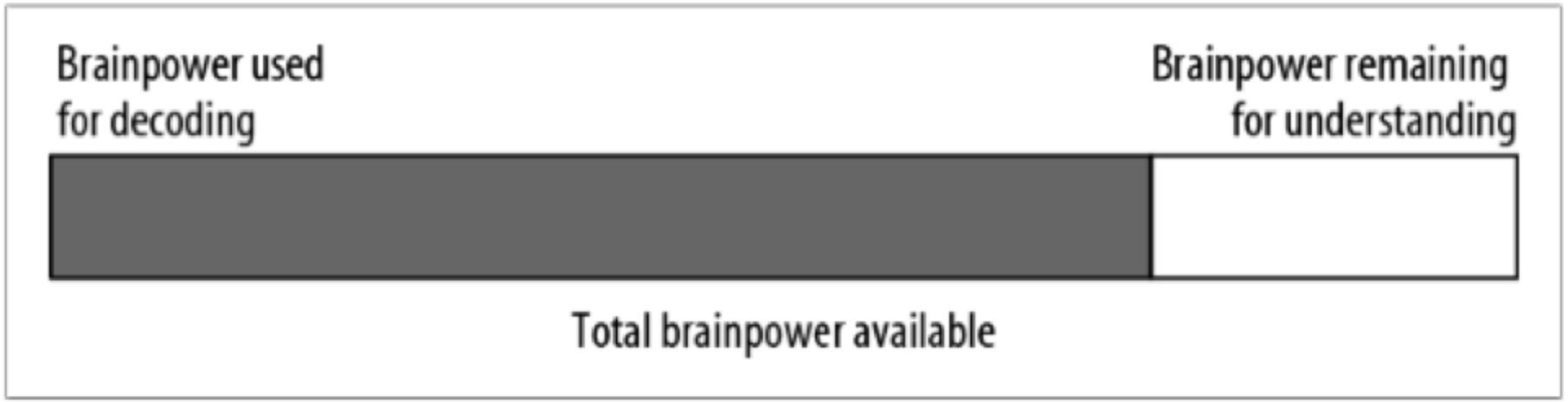


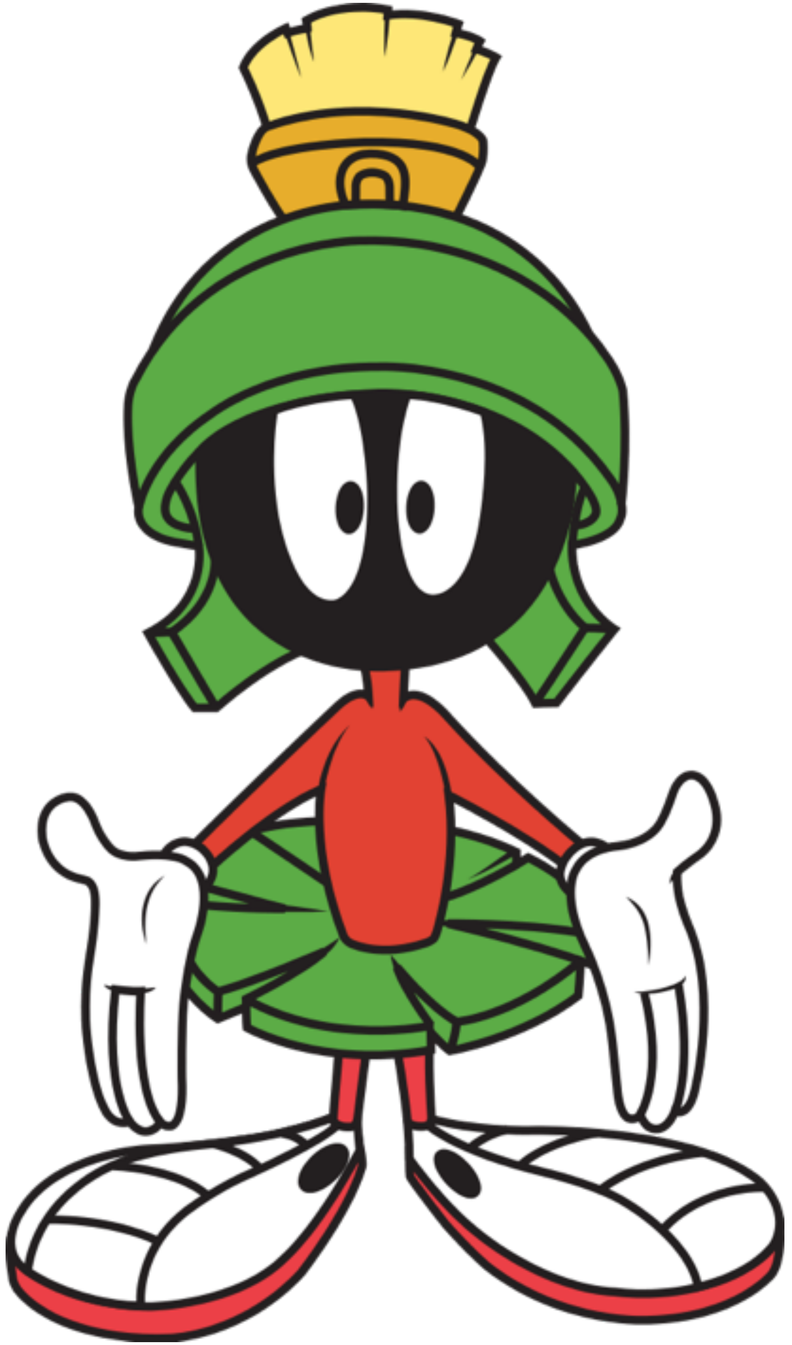
h8p://themendozaline.org/post/113828947026/3d-pie-charts-are-lie-charts

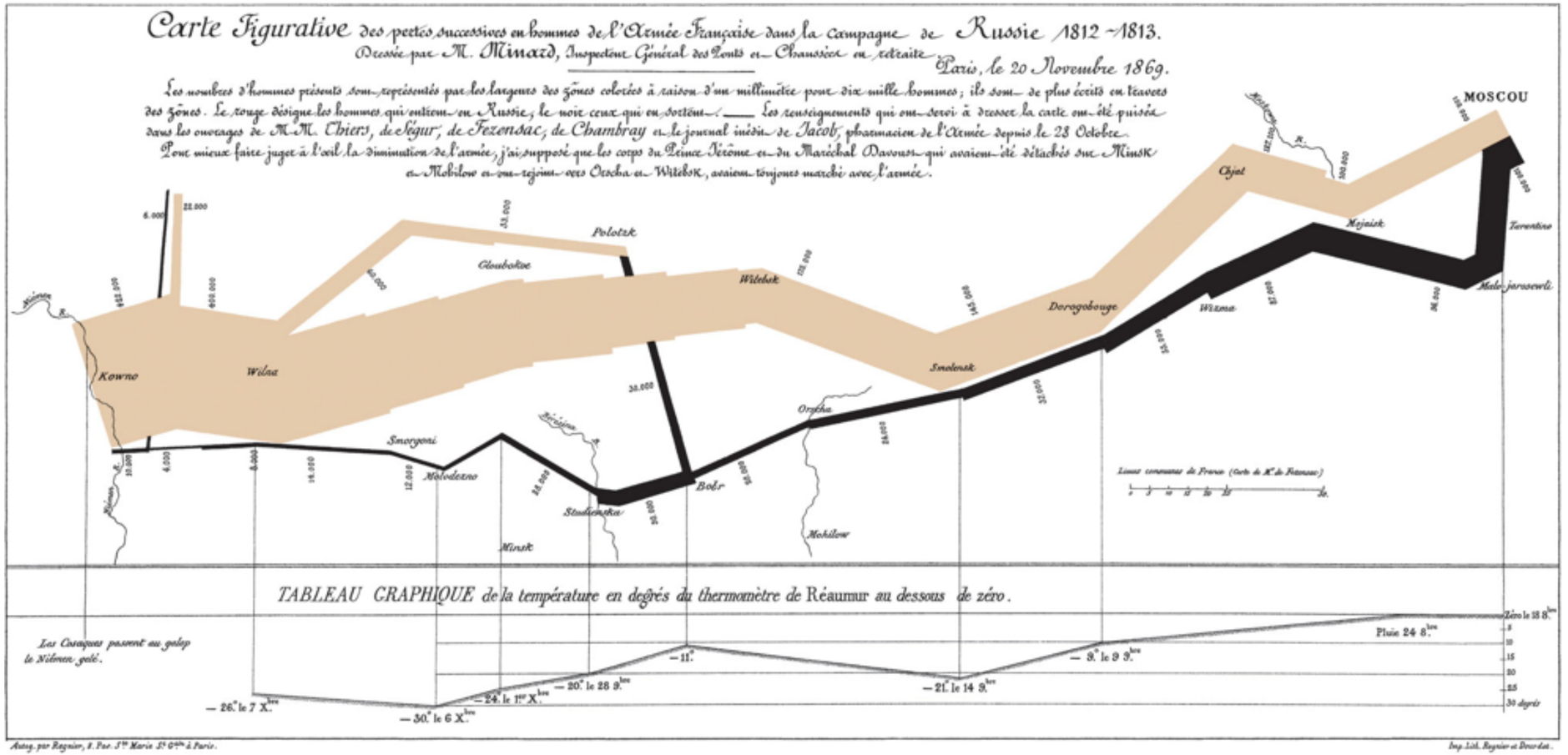
GET ON THE FAST TRACK

Chart Suggestions—A Thought-Starter

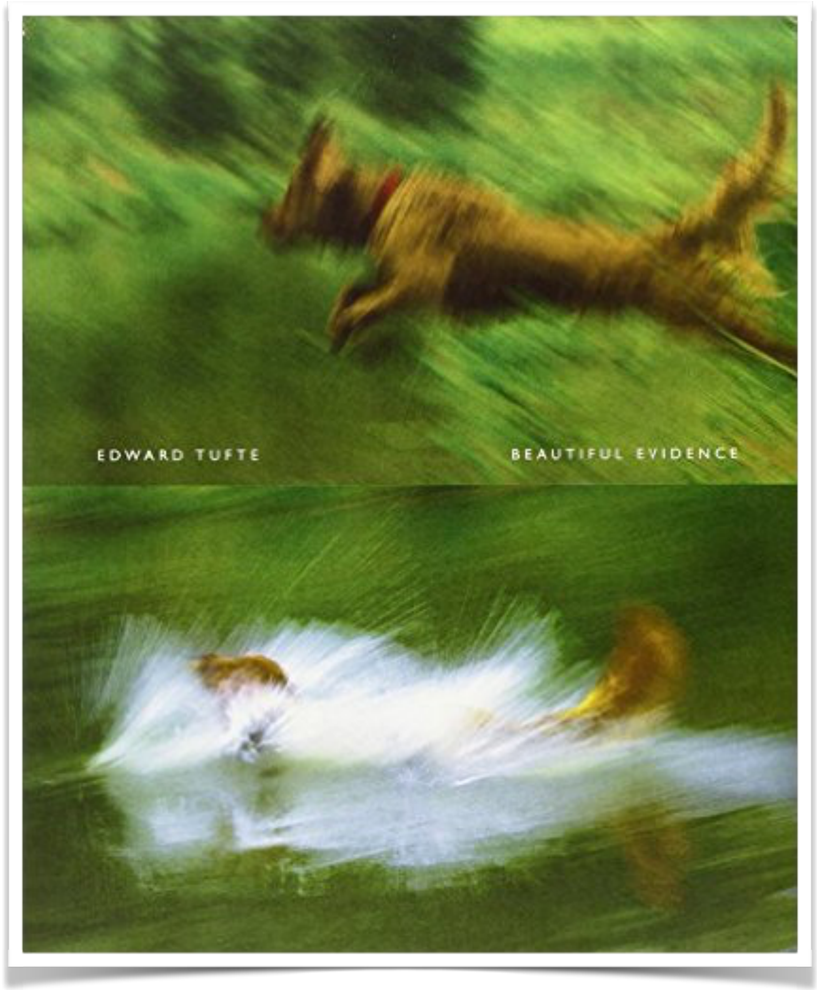



Brainpower used
for decoding
Brainpower remaining
for understanding
Total brainpower available





Recommended Reading



Storytelling

Storytelling

What is the Question?

Who Cares?

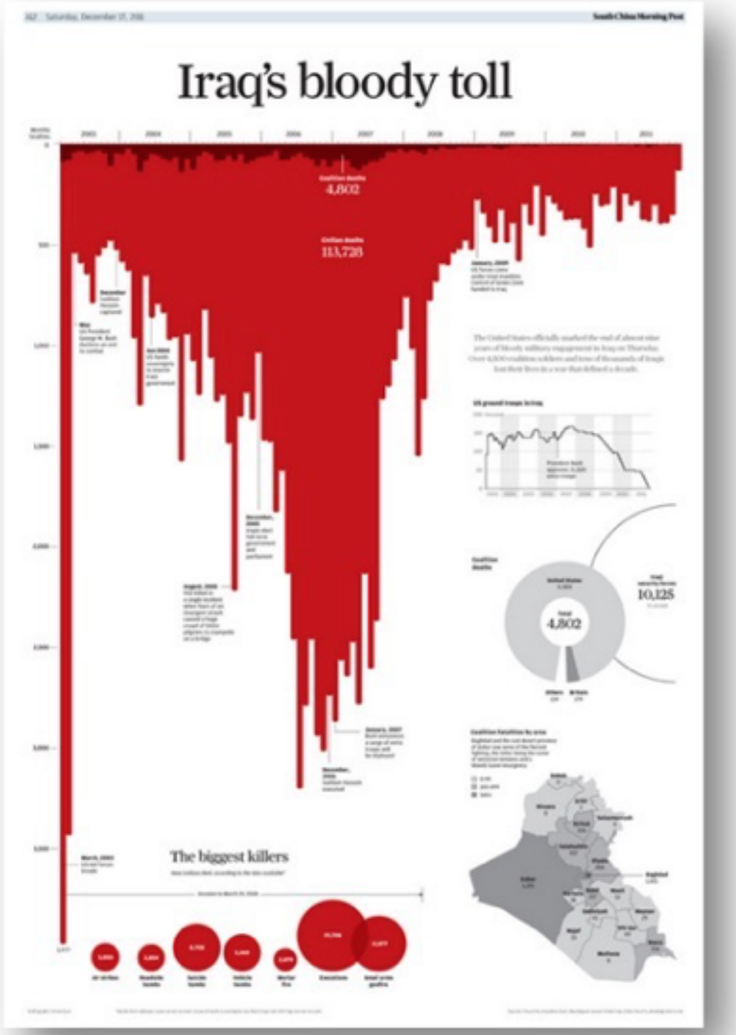
Storytelling

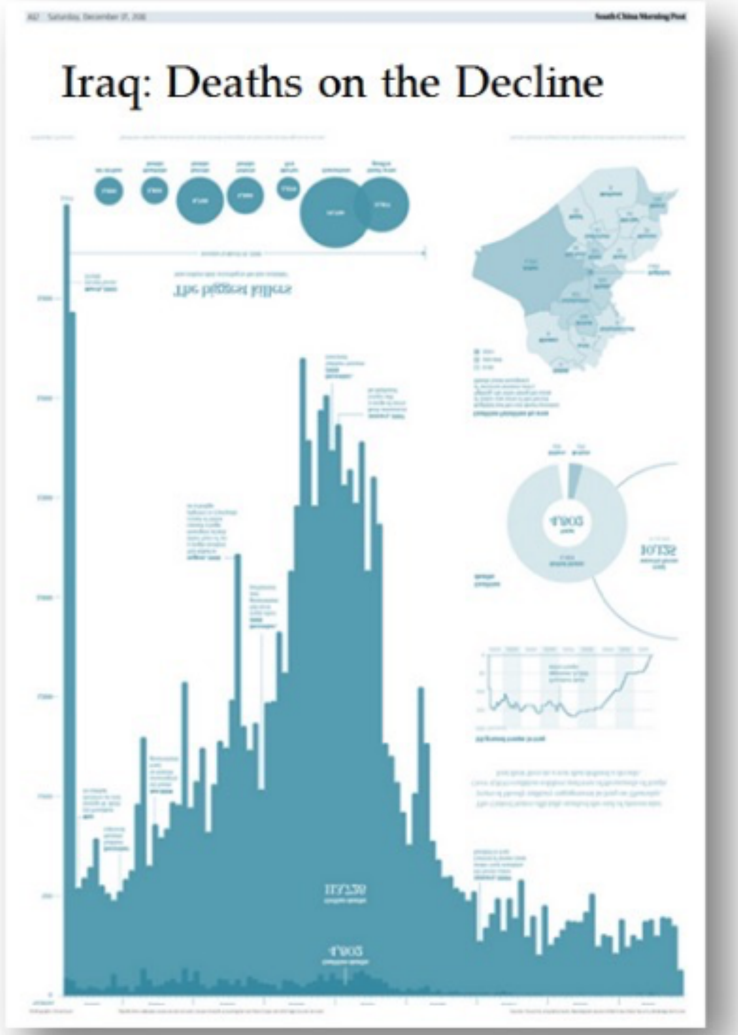
1. Know Your Audience
2. Tell a Story
3. Don’t Confuse
4. Go with the Flow

Elements of a Story

1. Characters
2. Conflict
3. Resolution
4. Sequel?

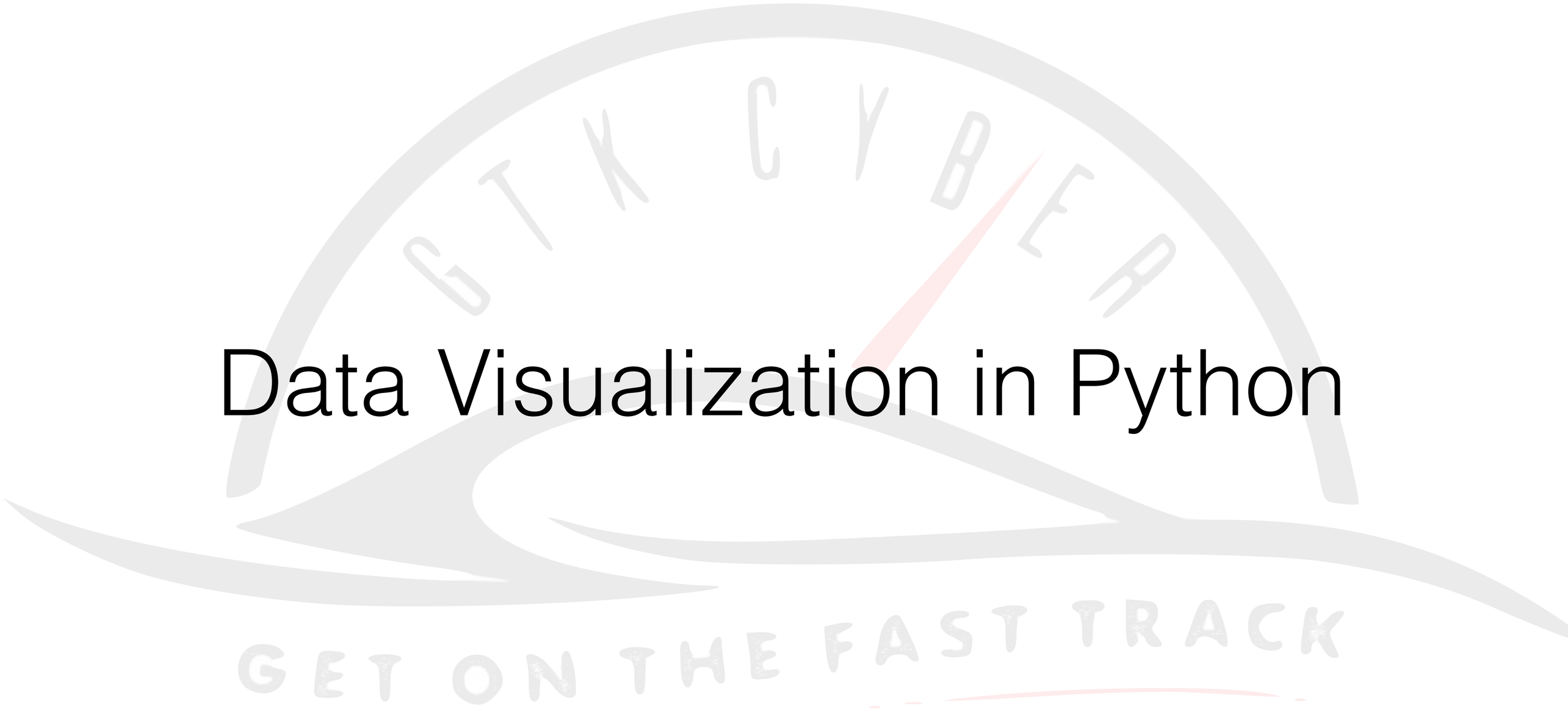
Influential Storytelling



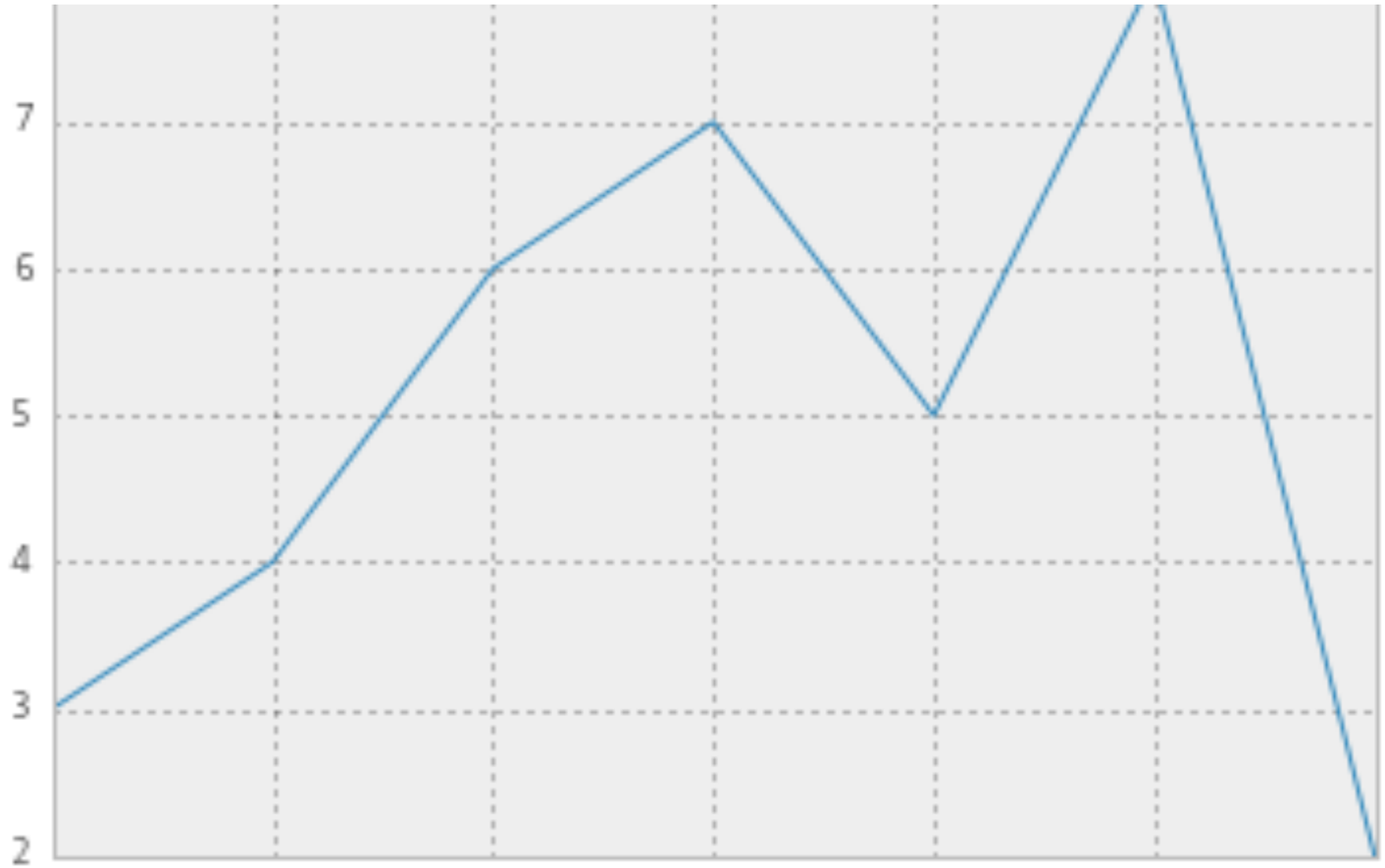


Andy Cotgreave

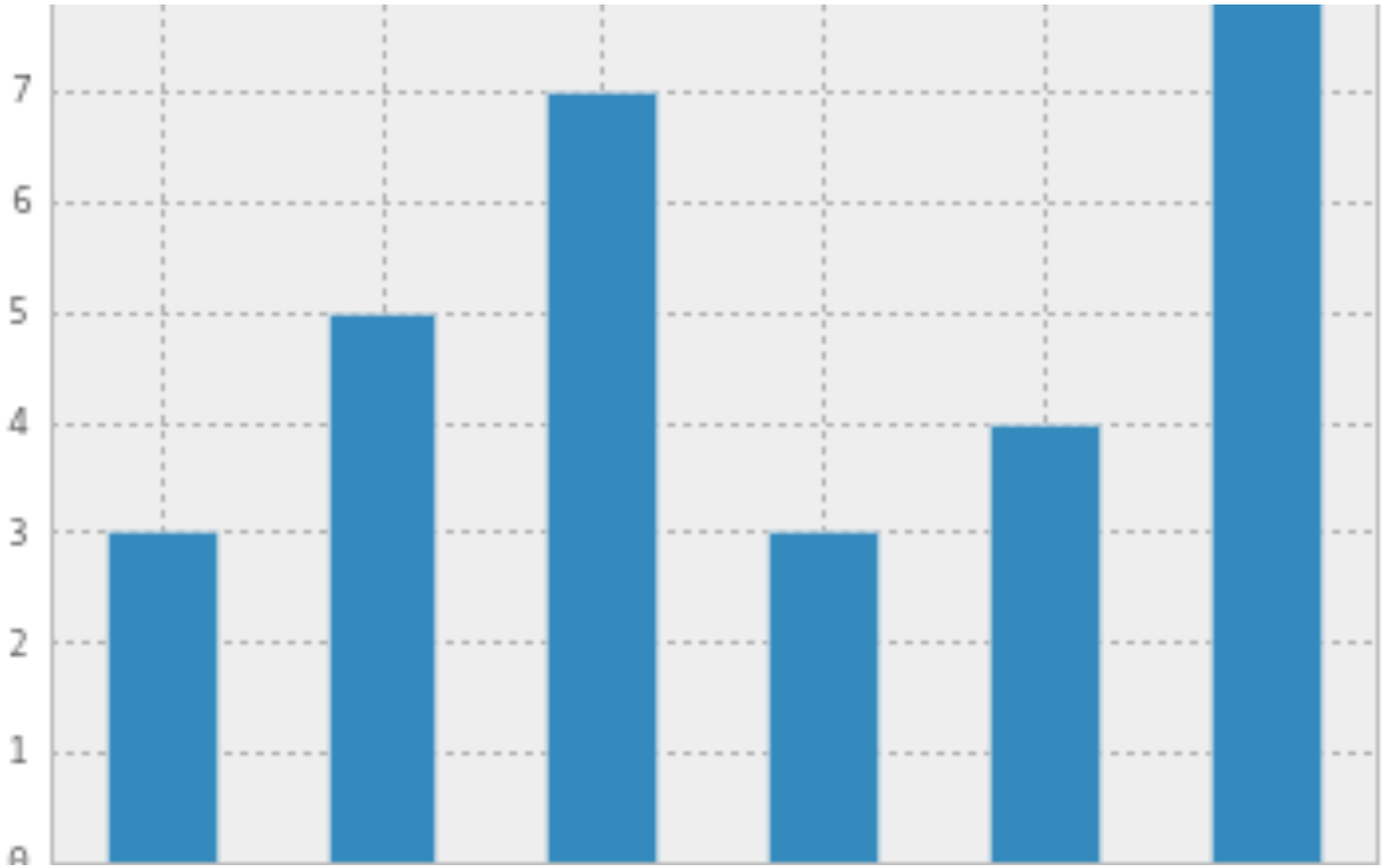
Data Visualization in Python



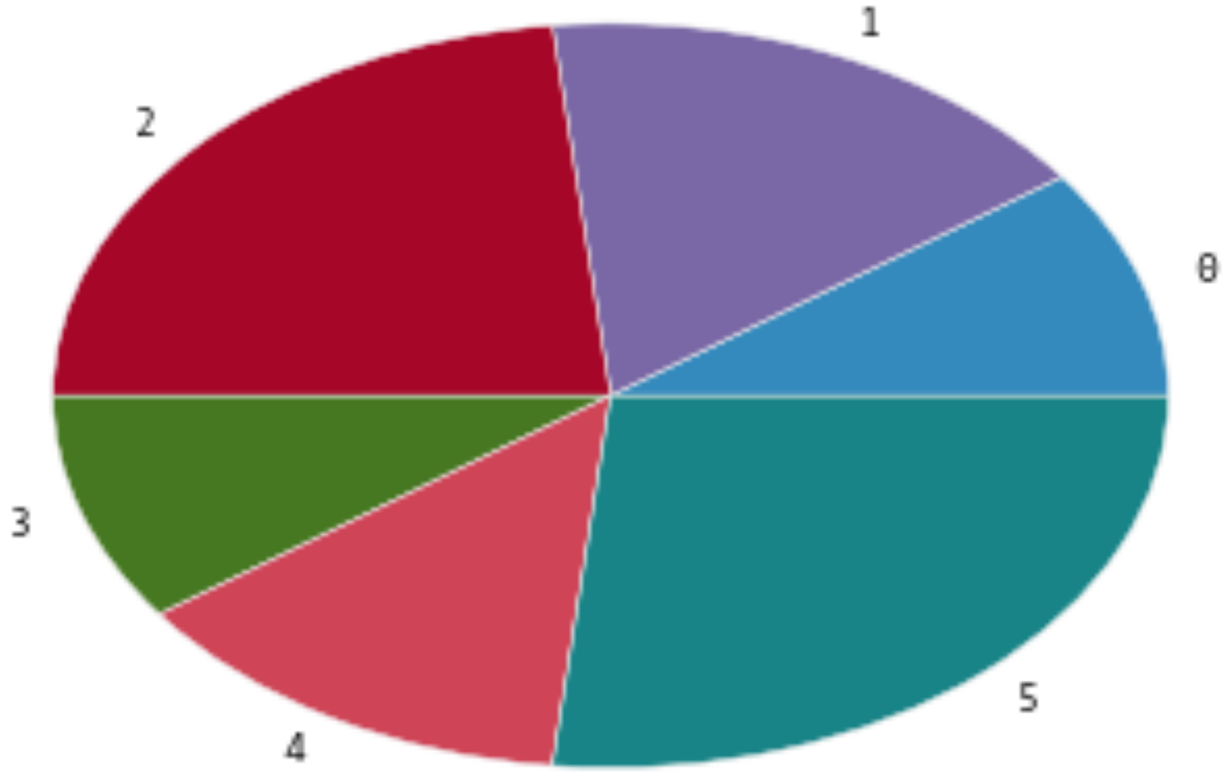
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
pd.options.display.mpl_style = 'default'
```



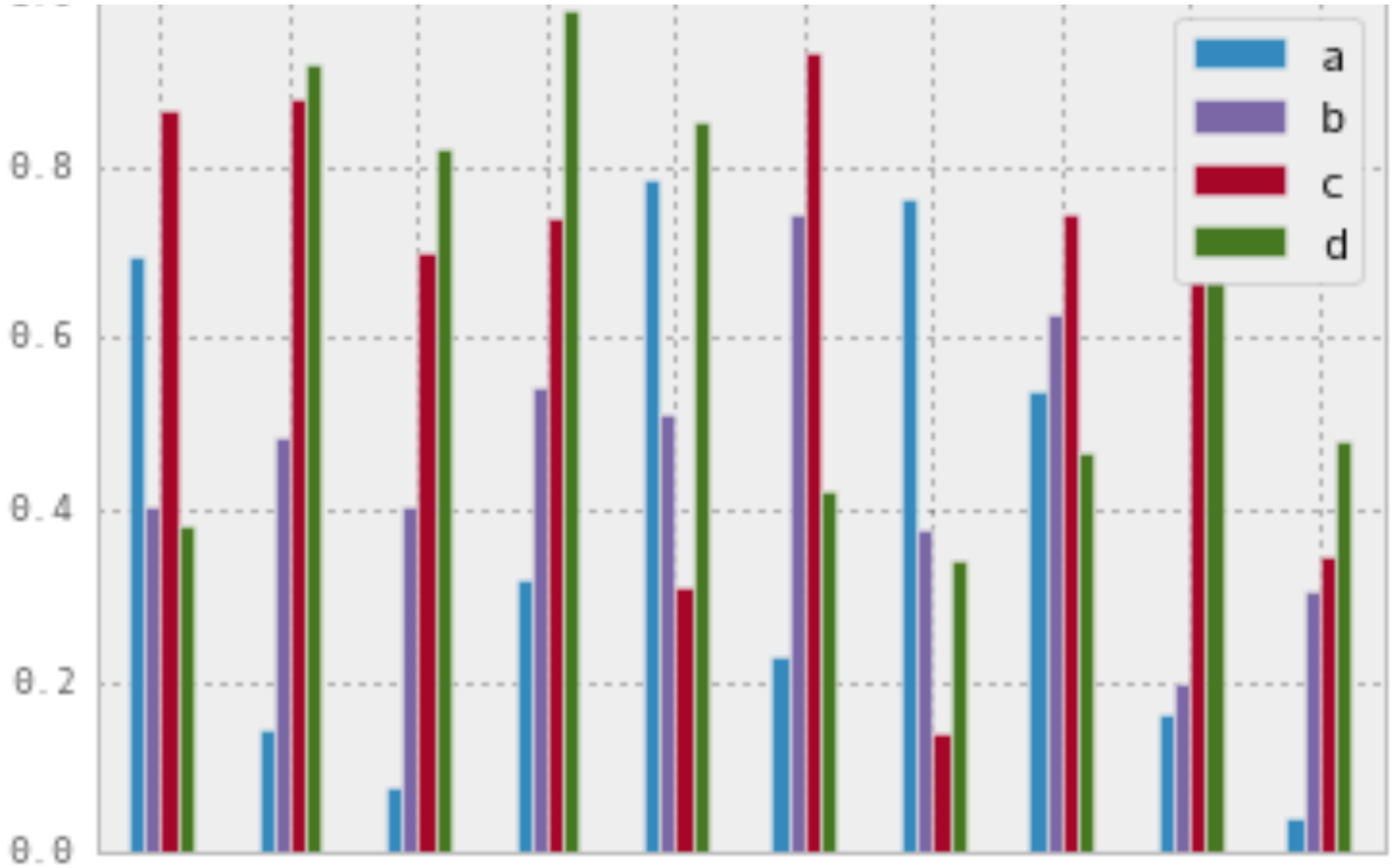

```
data = pd.Series( [3,4,6,7,5,8,2 ])
graph = data.plot()
```



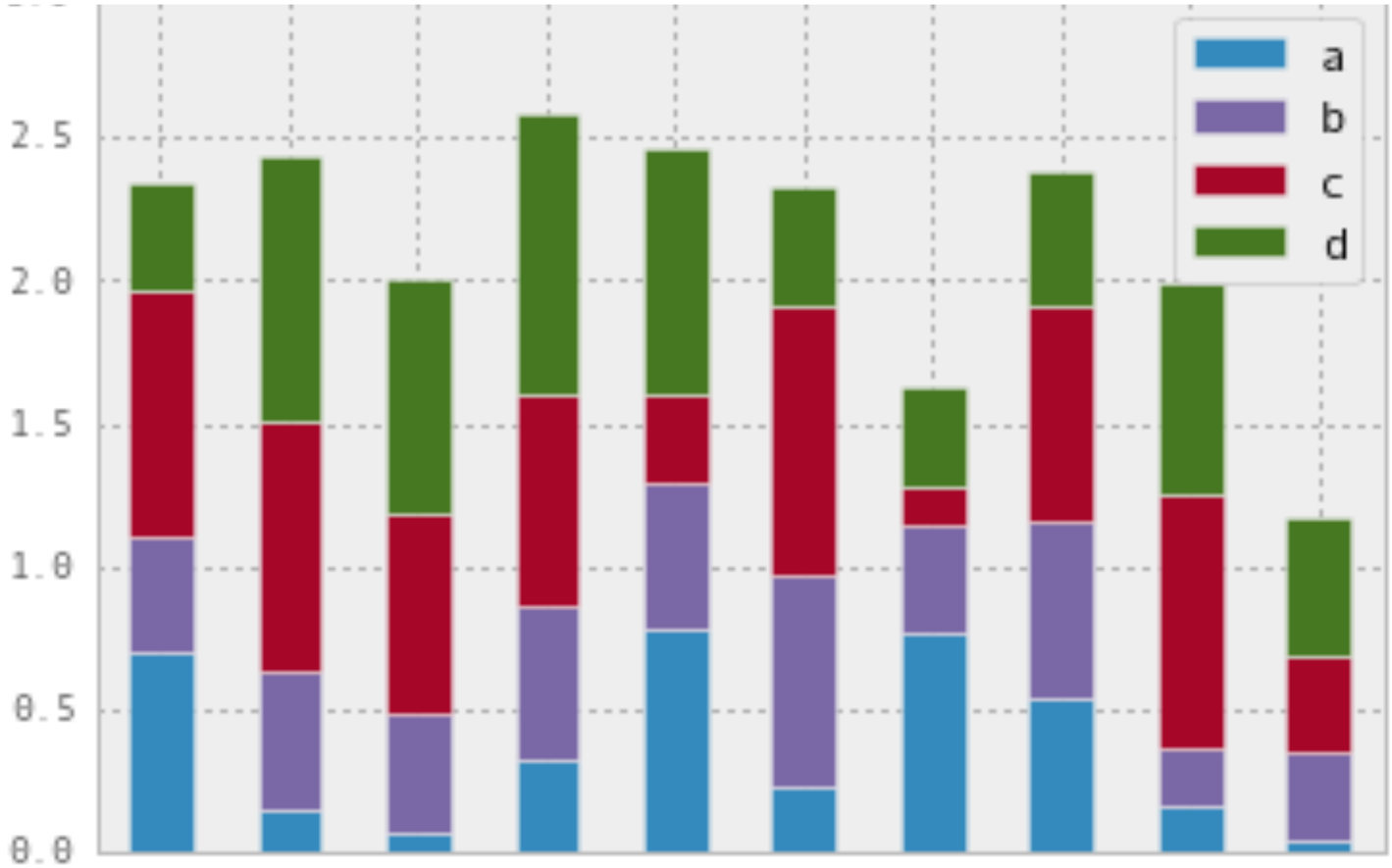

```
barchart = data.plot( kind="bar" )
```



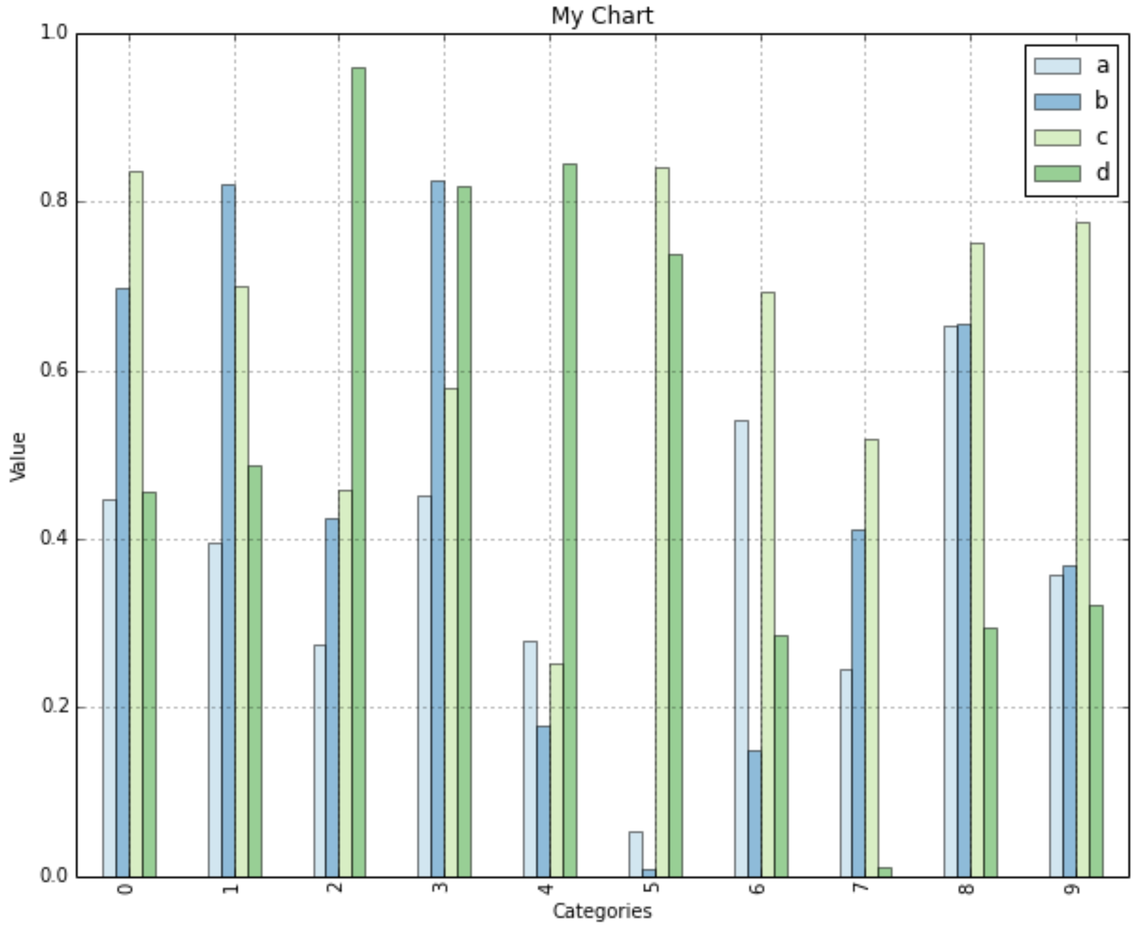

```
piechart = data.plot( kind="pie" )
```



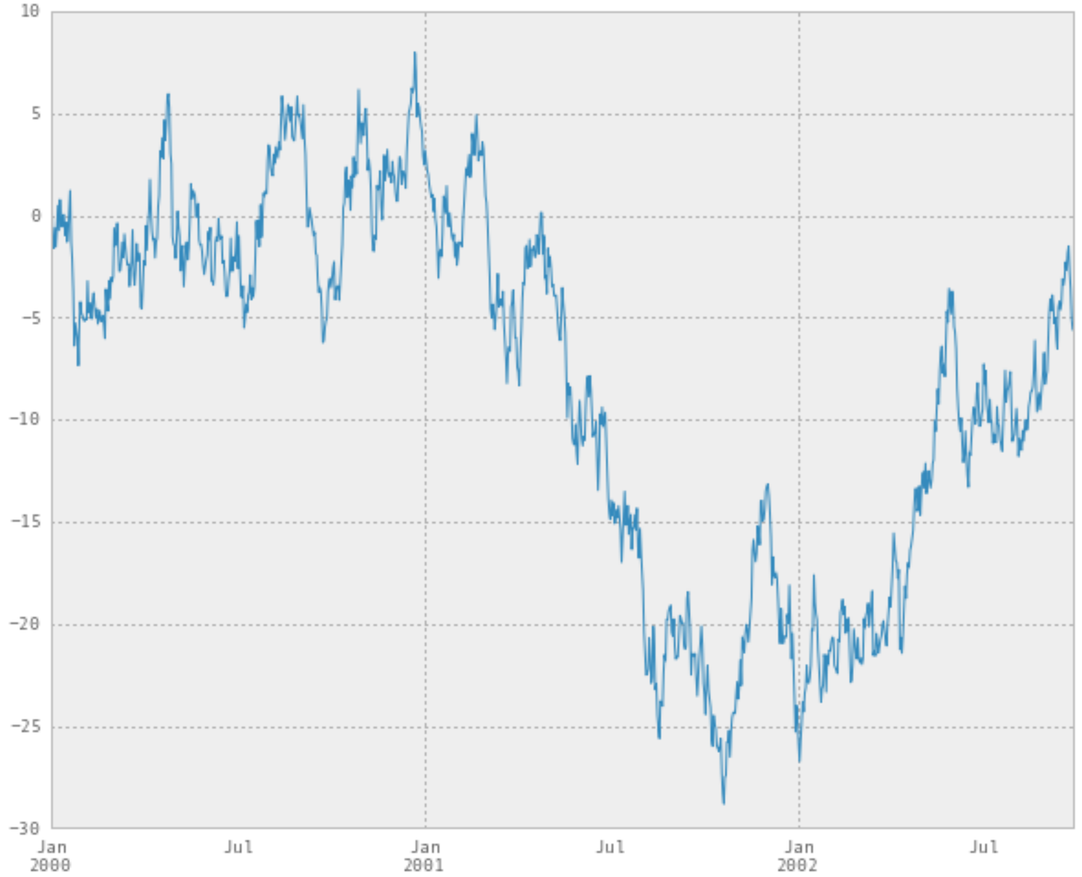

```
df2 = pd.DataFrame(np.random.rand(10, 4),
columns=['a', 'b', 'c', 'd'])
df2.plot( kind='bar' )
```



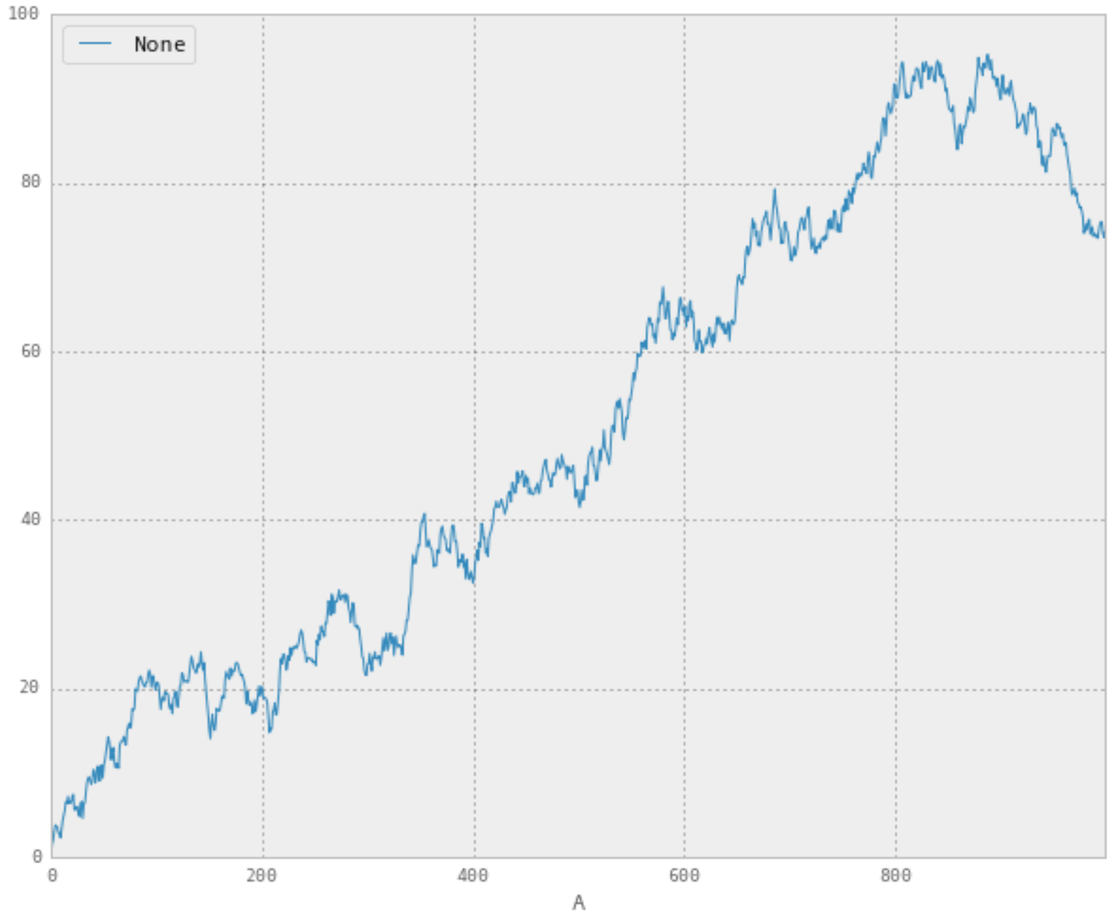

```
df2.plot( kind='bar', stacked=True )
```



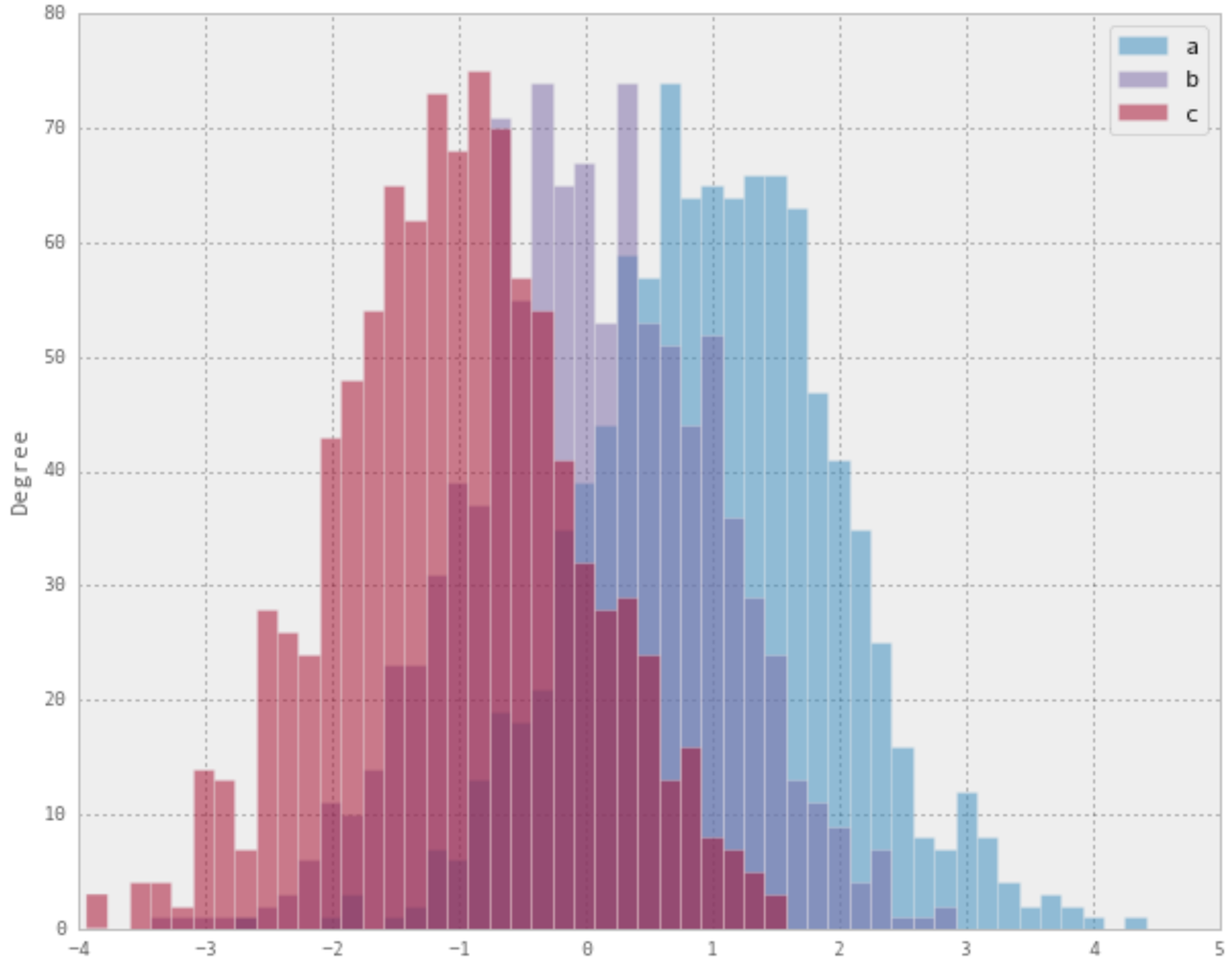

```
df2.plot( kind='bar',
          color=('#a6cee3','#1f78b4','#b2df8a','#33a02c' ),
           alpha=0.5,
             width=0.5,
             figsize=(10, 8))
plt.title( "My Chart" )
plt.xlabel( "Categories" )
plt.ylabel( "Value" )
```



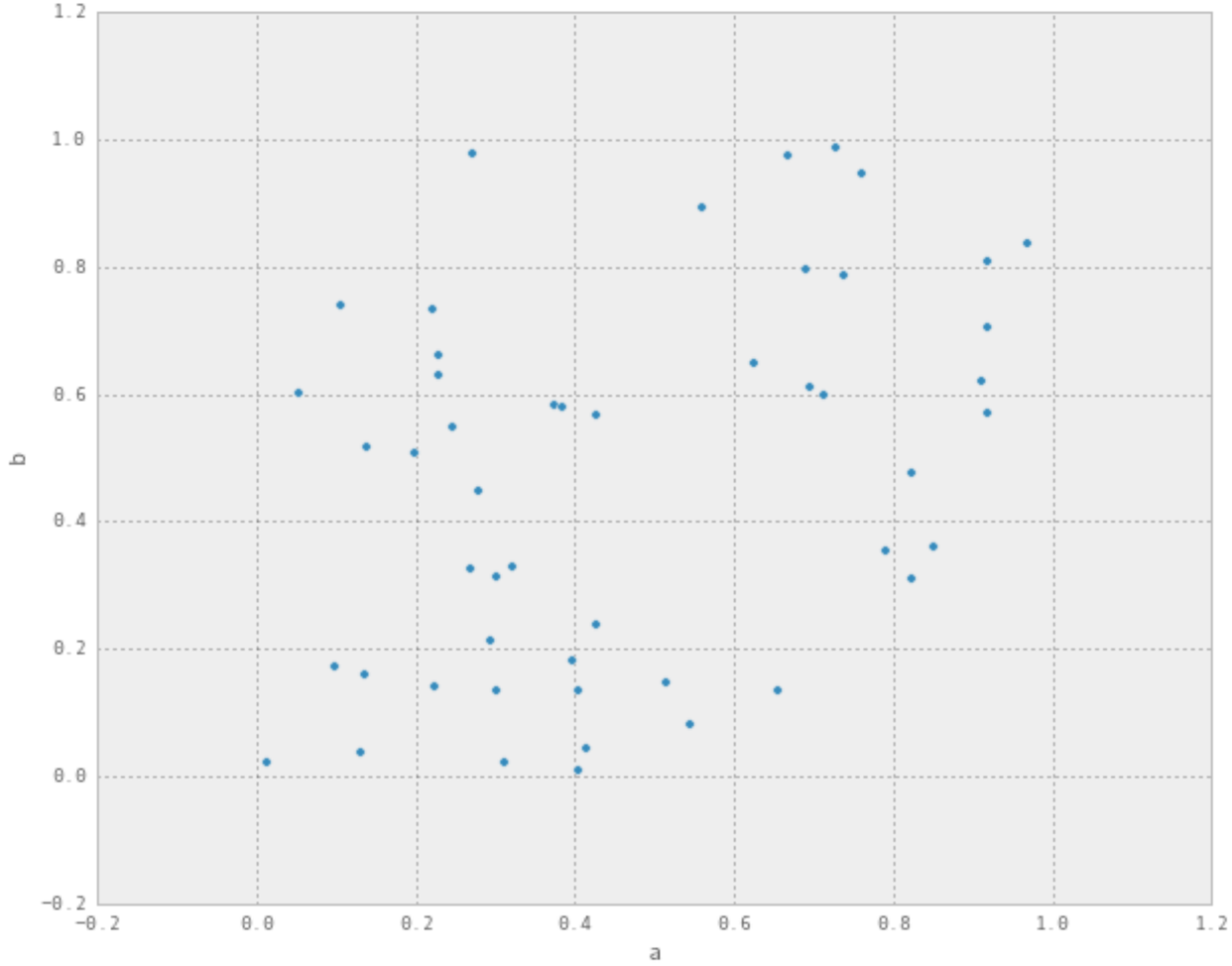

```
ts = pd.Series(np.random.randn( 1000),
index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()
timeseriesChart = ts.plot( figsize=(10, 8) )
```



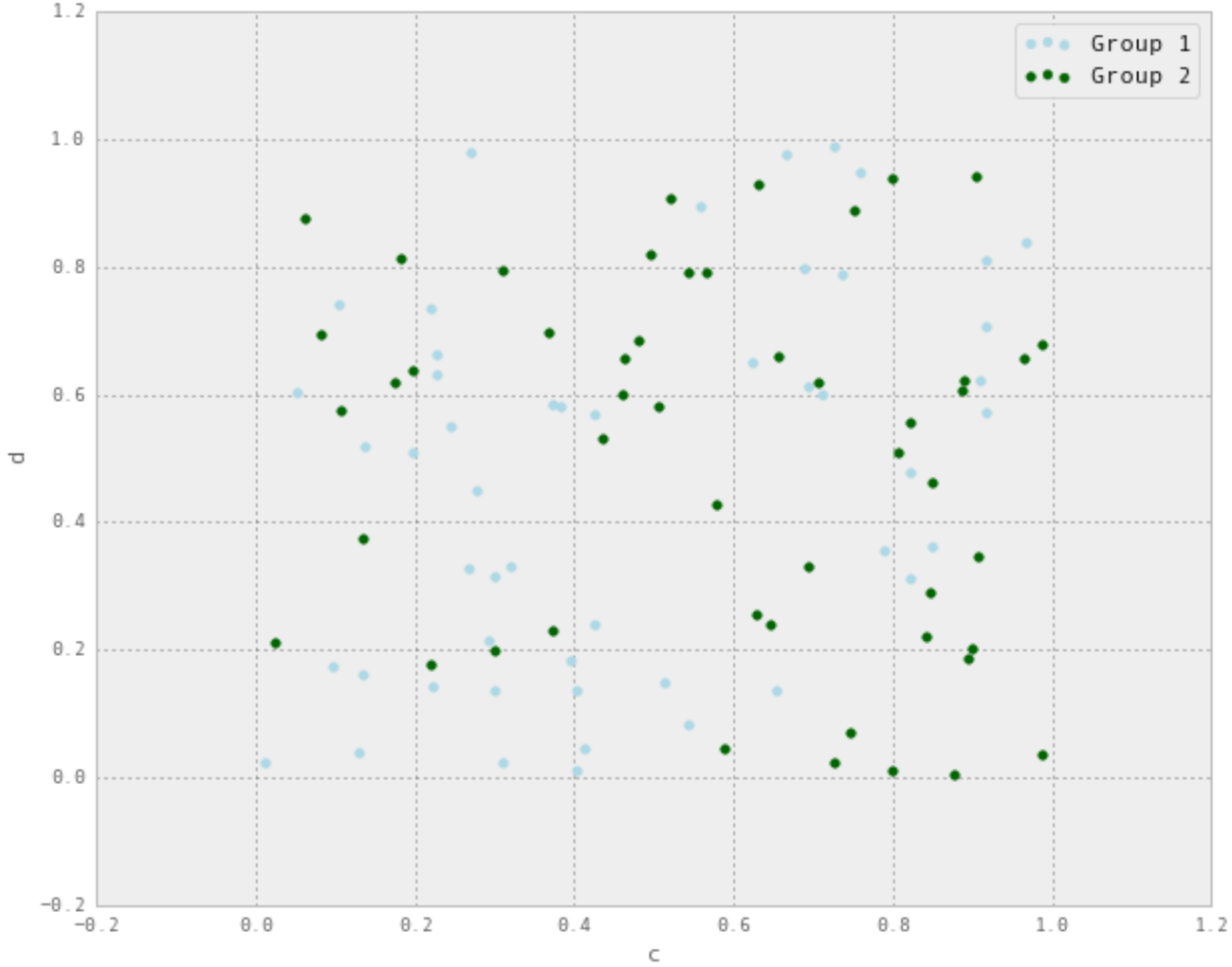

```
df3 = pd.DataFrame(np.random.randn(1000, 2),
columns=['B', 'C']).cumsum()
df3['A'] = pd.Series(list(range(len(df3))))

df3.plot( x='A', y='B' )
```



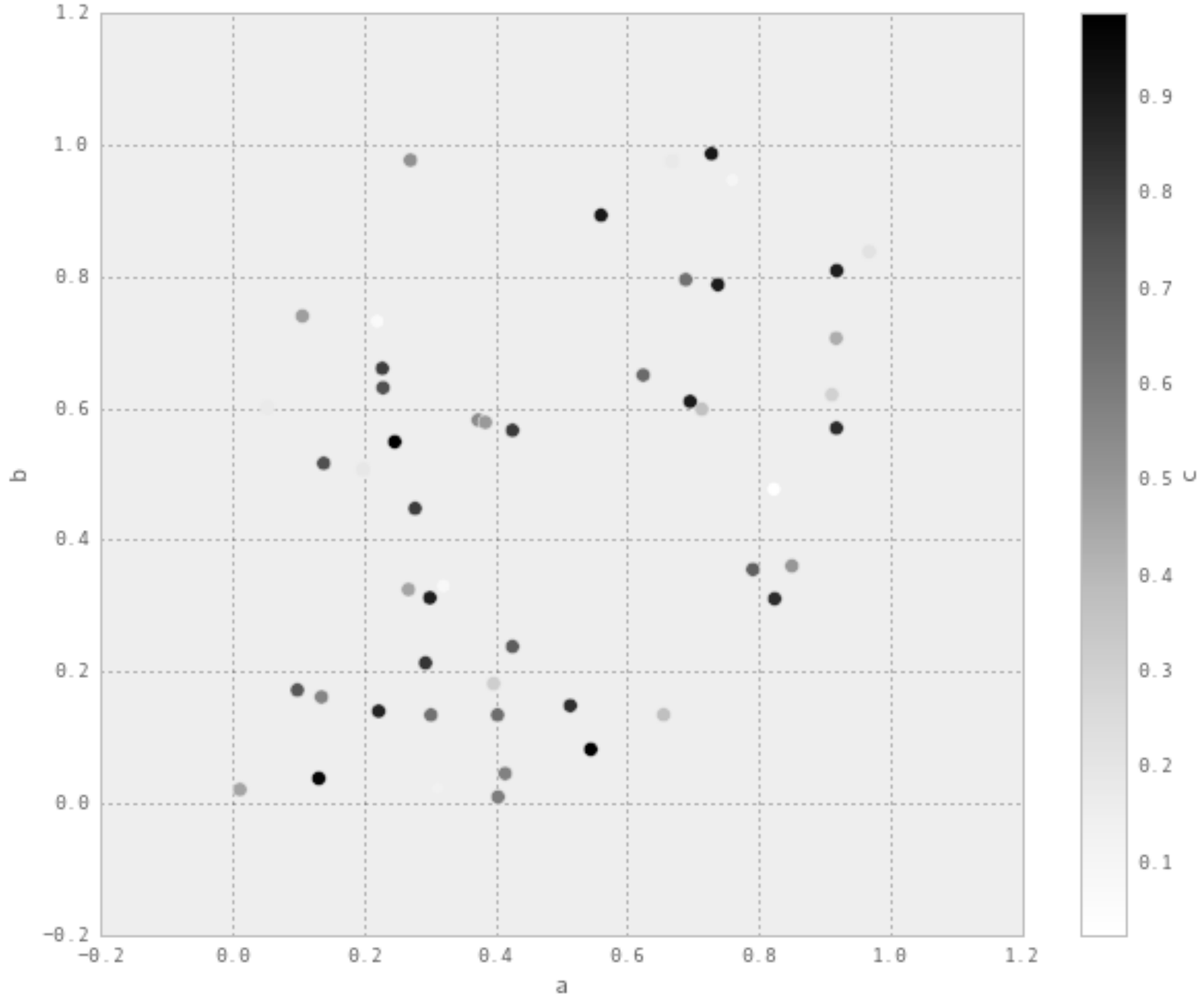

```
df4.plot(kind='hist',
         alpha=0.5,
         bins=50 )
```



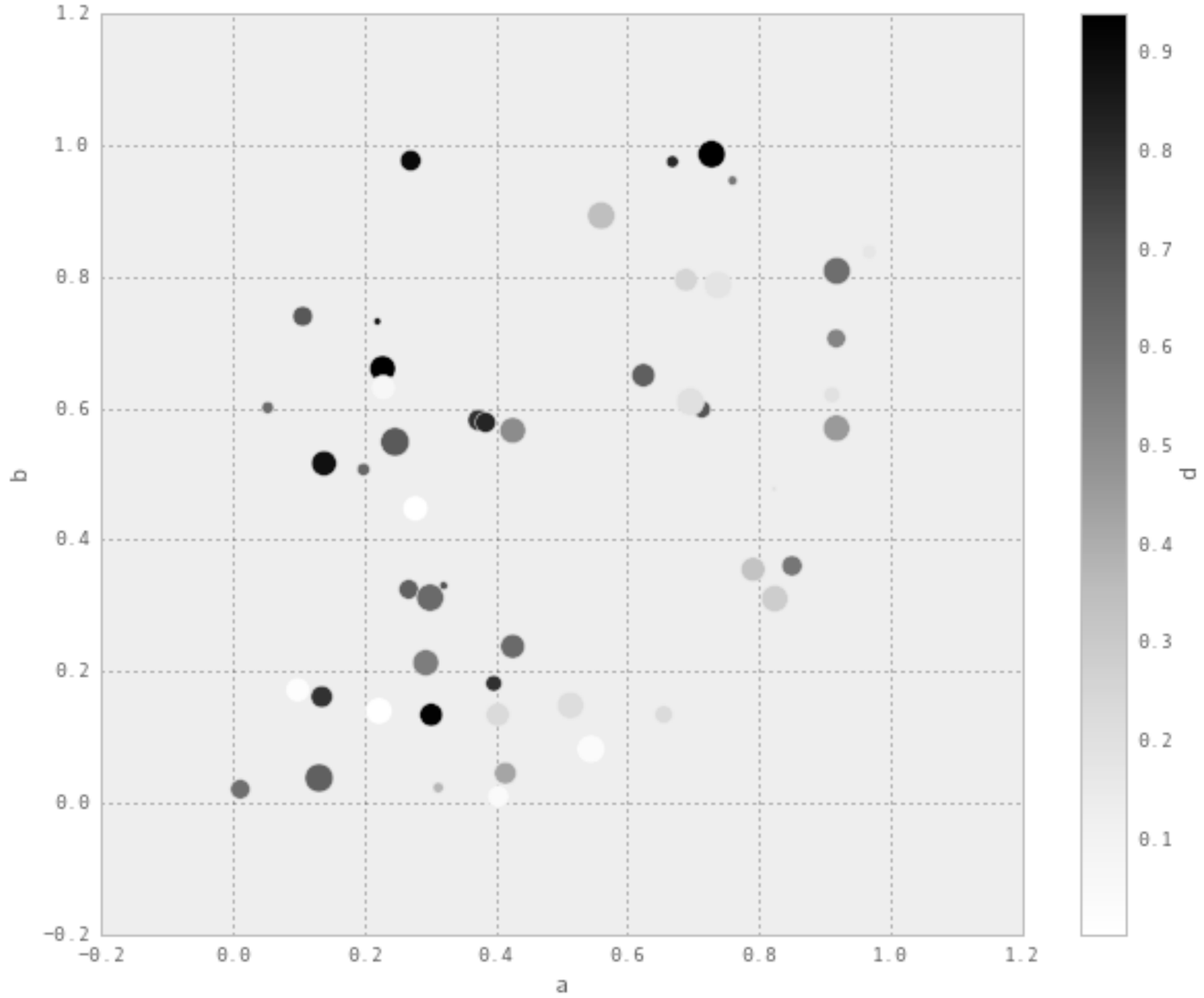

```
df5 = pd.DataFrame(np.random.rand(50, 4),
columns=['a', 'b', 'c', 'd'] )
df5.plot(kind='scatter', x='a', y='b' )
```



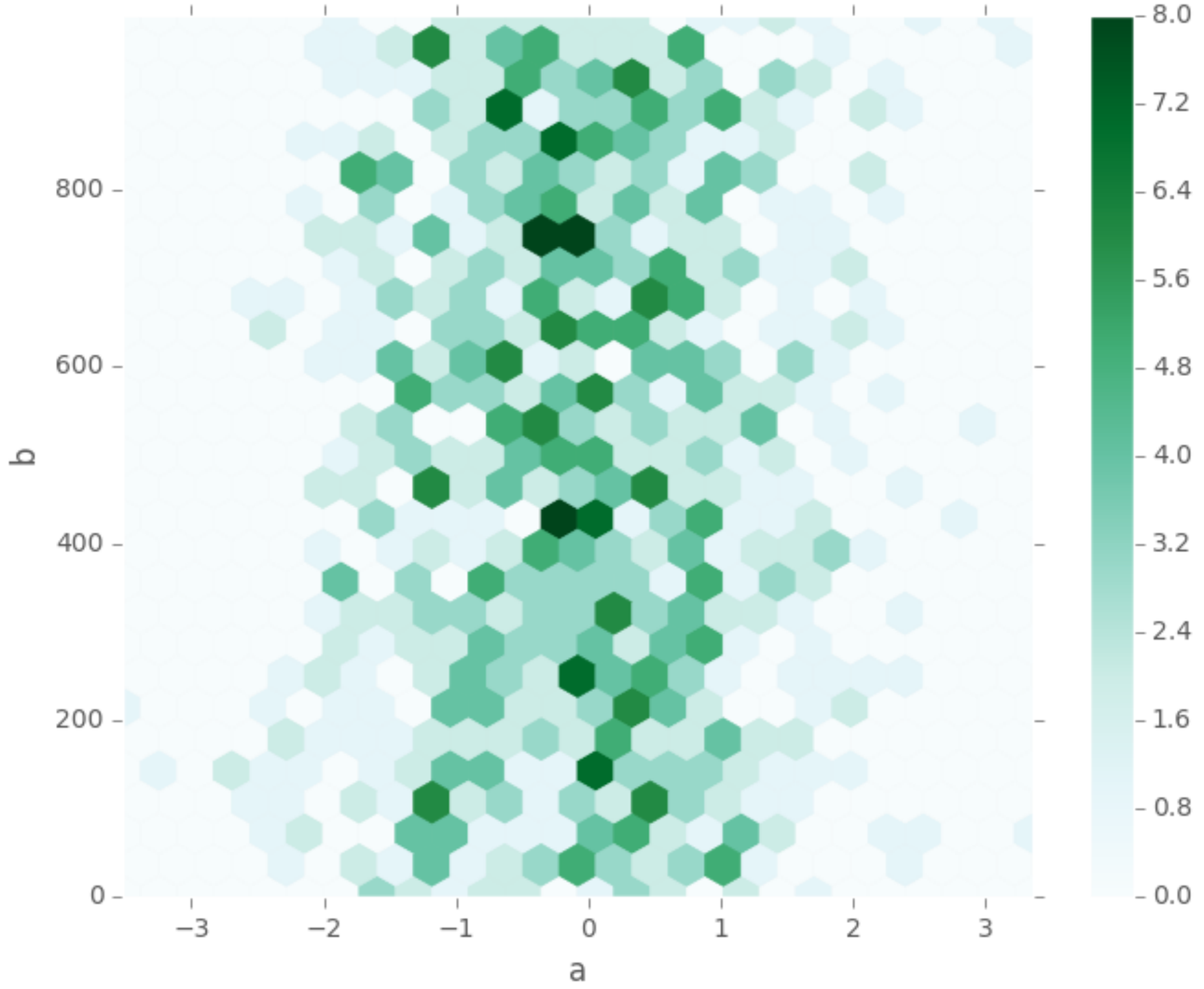

```
ax = df5.plot(kind='scatter', x='a', y='b',
              color='LightBlue',
              label='Group 1',
              figsize=(10, 8) )

df5.plot(kind='scatter', x='c', y='d',
             color='DarkGreen',
            label='Group 2',
            ax=ax)
```




```
df5.plot(kind='scatter', x='a', y='b',
c='c', s=50 )
```




```
scatter = df5.plot(kind='scatter',
                   x='a',
                   y='b',
                   s=df5['c']*200,
                   c='d')
```

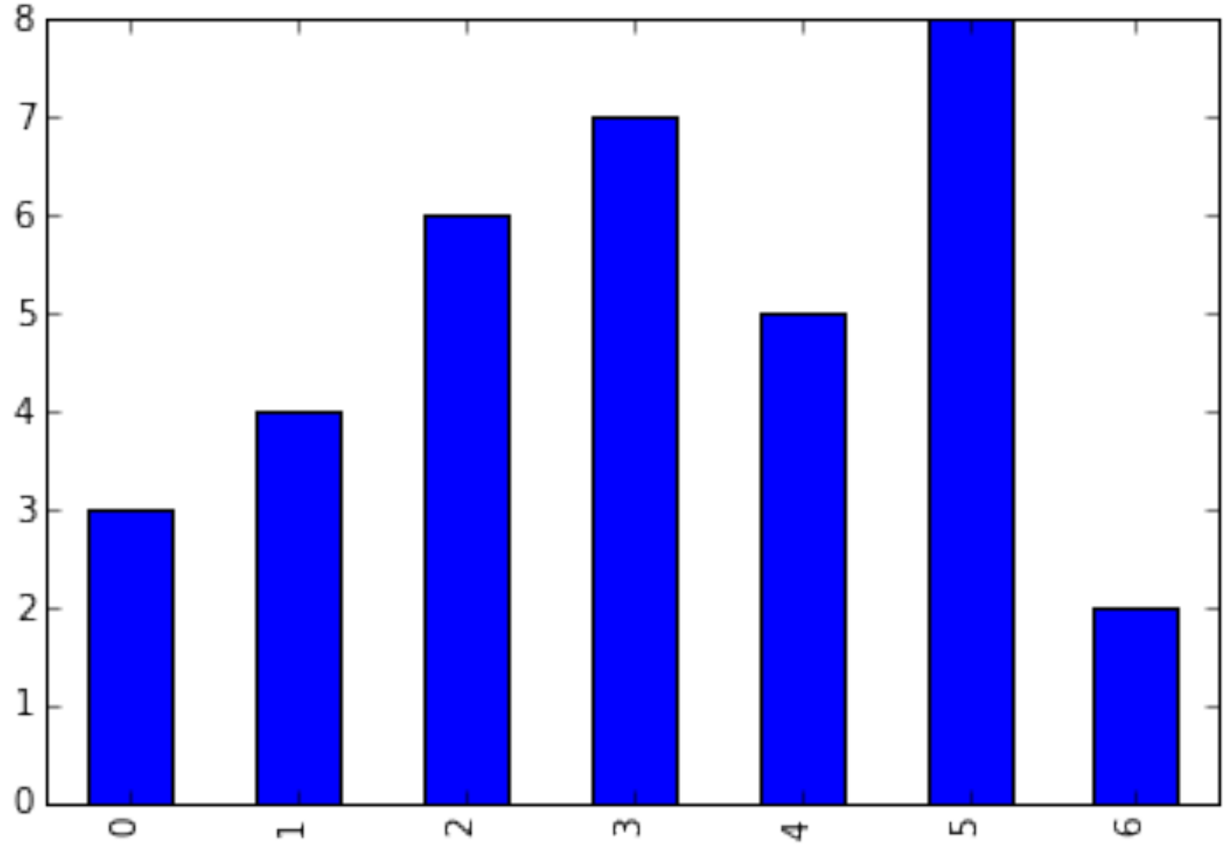



```
df.plot(kind='hexbin', x='a', y='b',
gridsize=25)
```

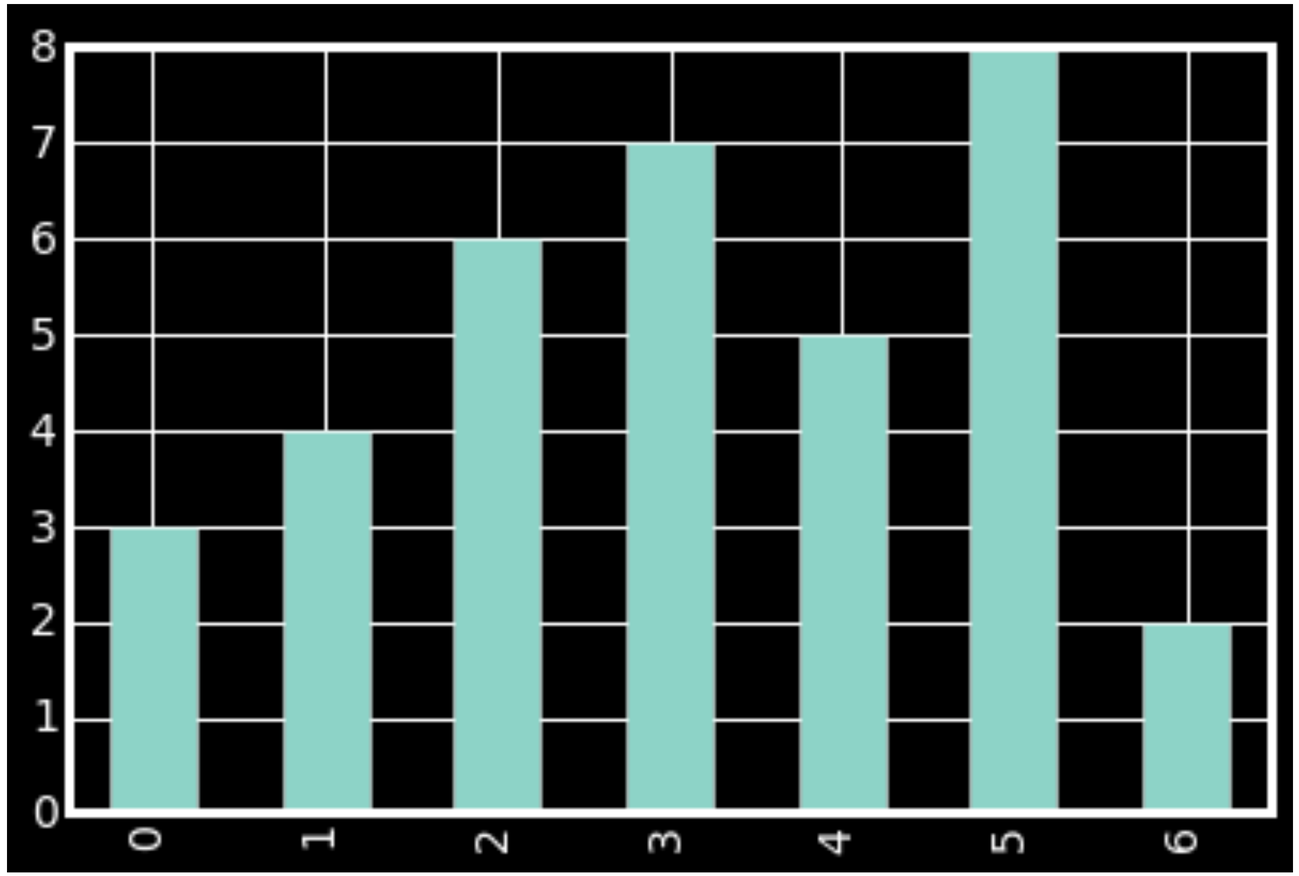

```
scatterPlot.get_figure().savefig( "scatterPlot.png" )
```


```
print(plt.style.available)

['dark_background', 'grayscale', 'ggplot',
'bmh', 'fivethirtyeight']
```

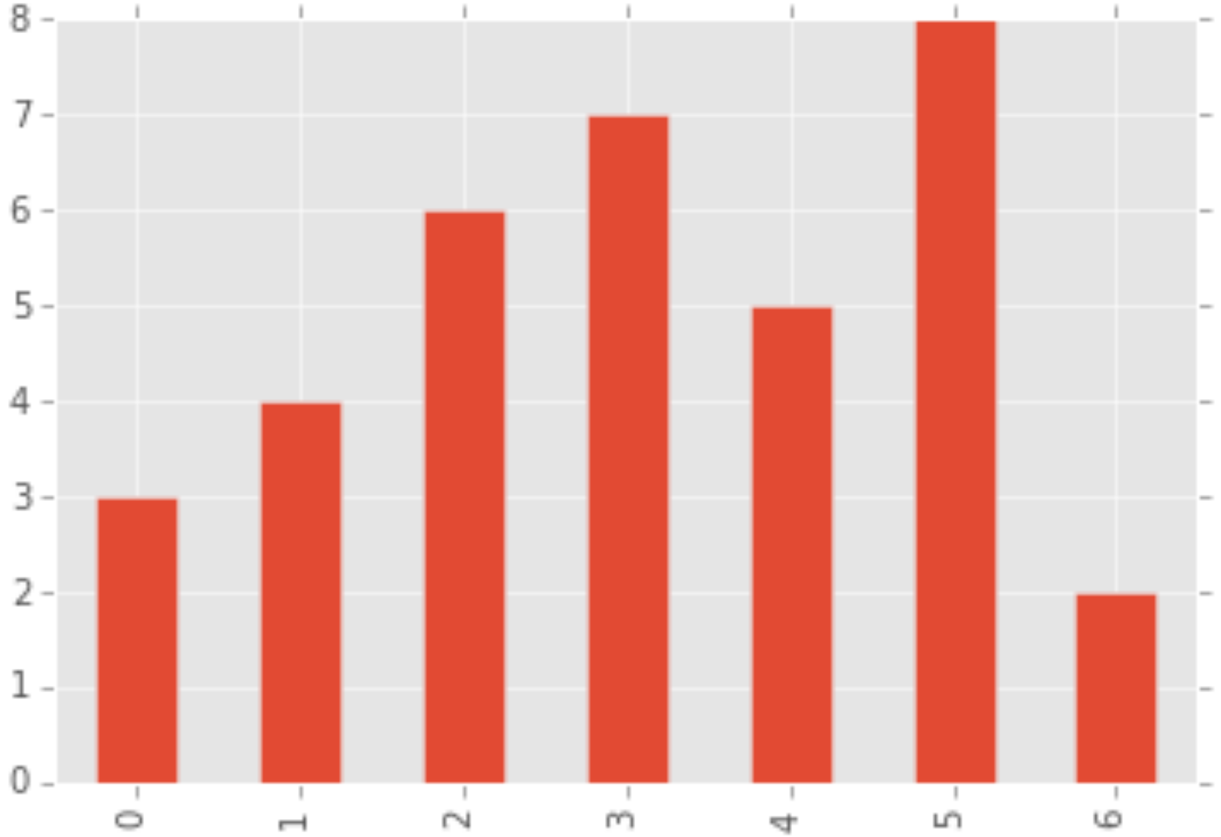
UGLY



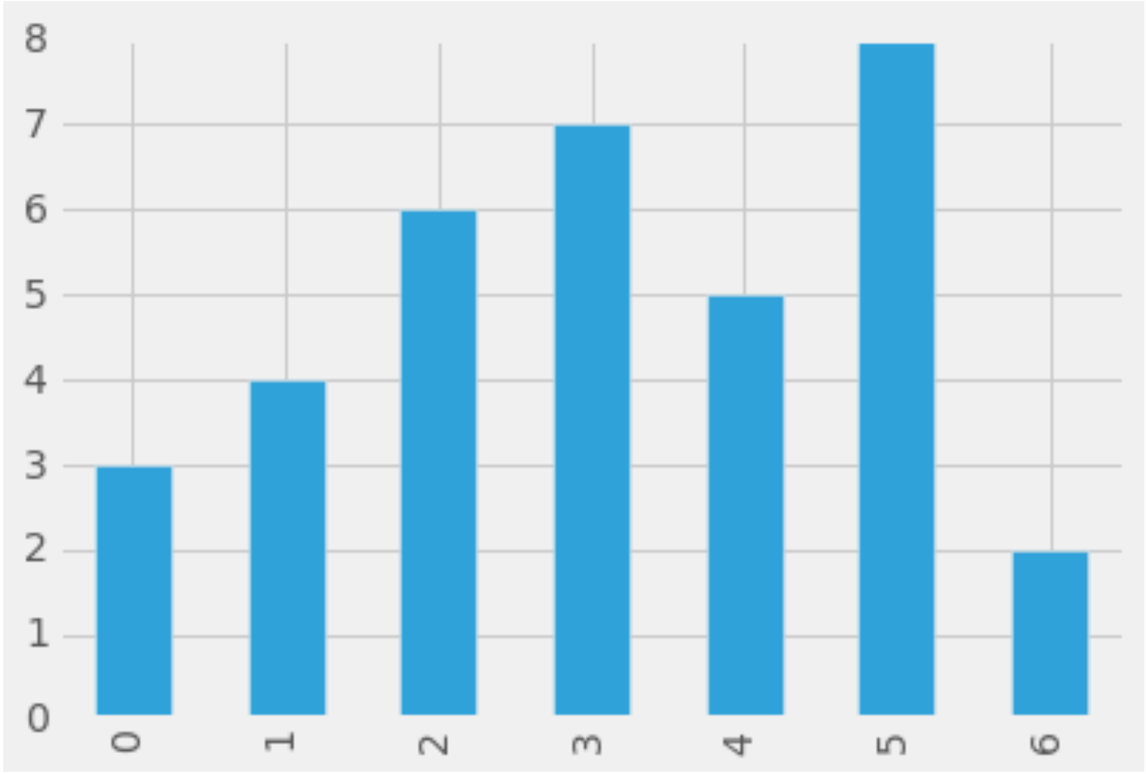

```
data = pd.Series( [3,4,6,7,5,8,2 ])
barchart = data.plot( kind="bar" )
```



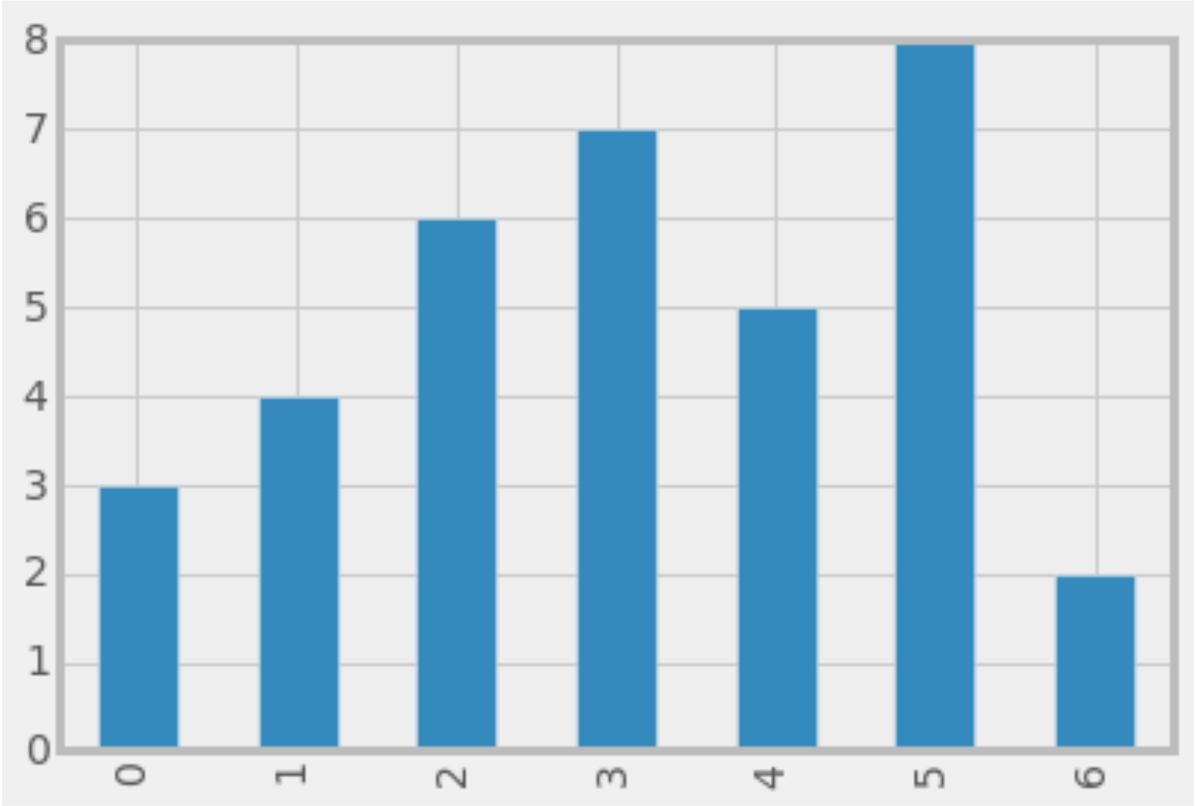

```
plt.style.use('dark_background')
barchart = data.plot( kind="bar" )
```




```
plt.style.use('ggplot')
barchart = data.plot( kind="bar" )
```




```
plt.style.use('fivethirtyeight')
barchart = data.plot( kind="bar" )
```

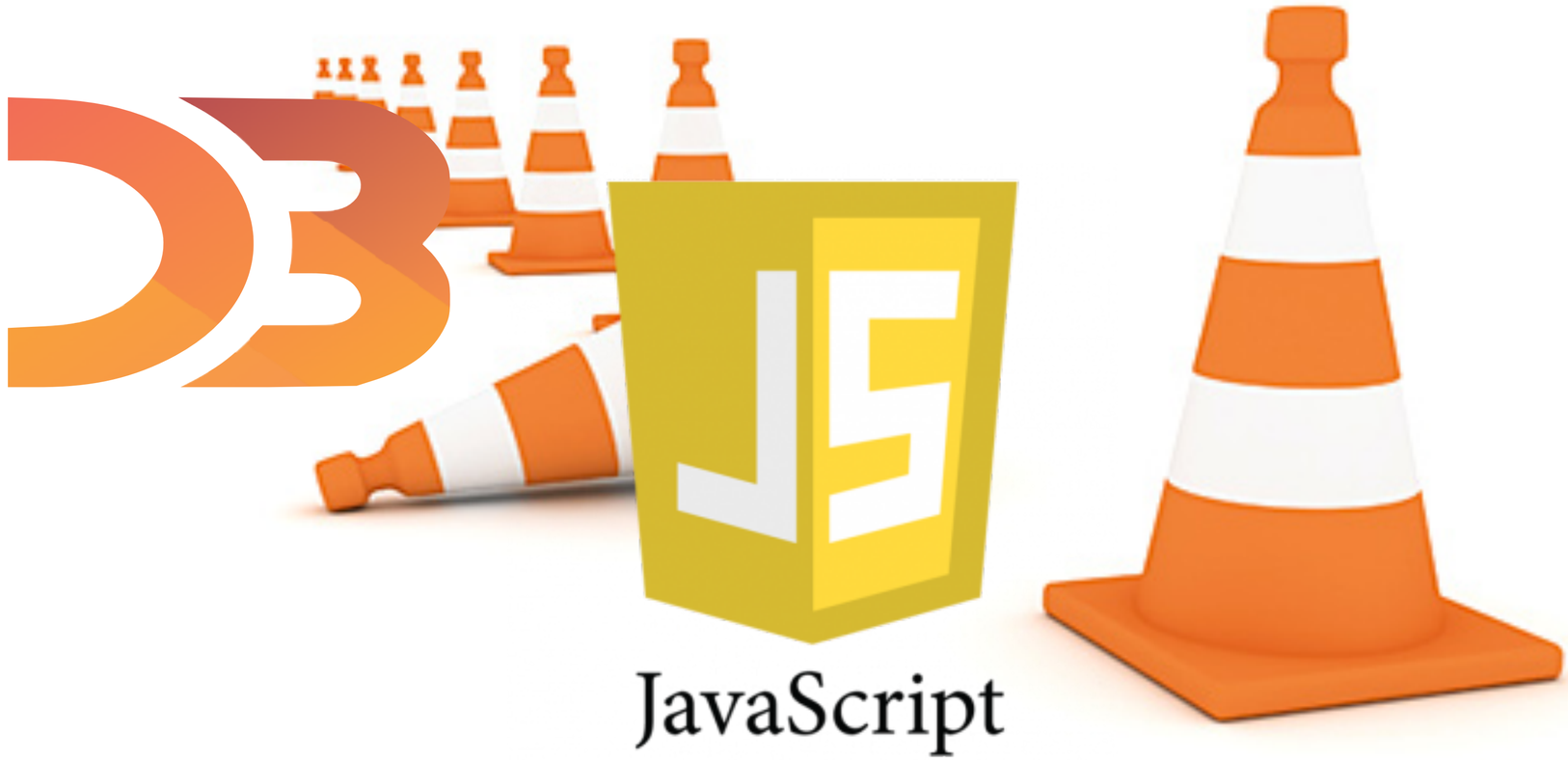



```
plt.style.use('bmh')
barchart = data.plot( kind="bar" )
```

Interactive Visualizations

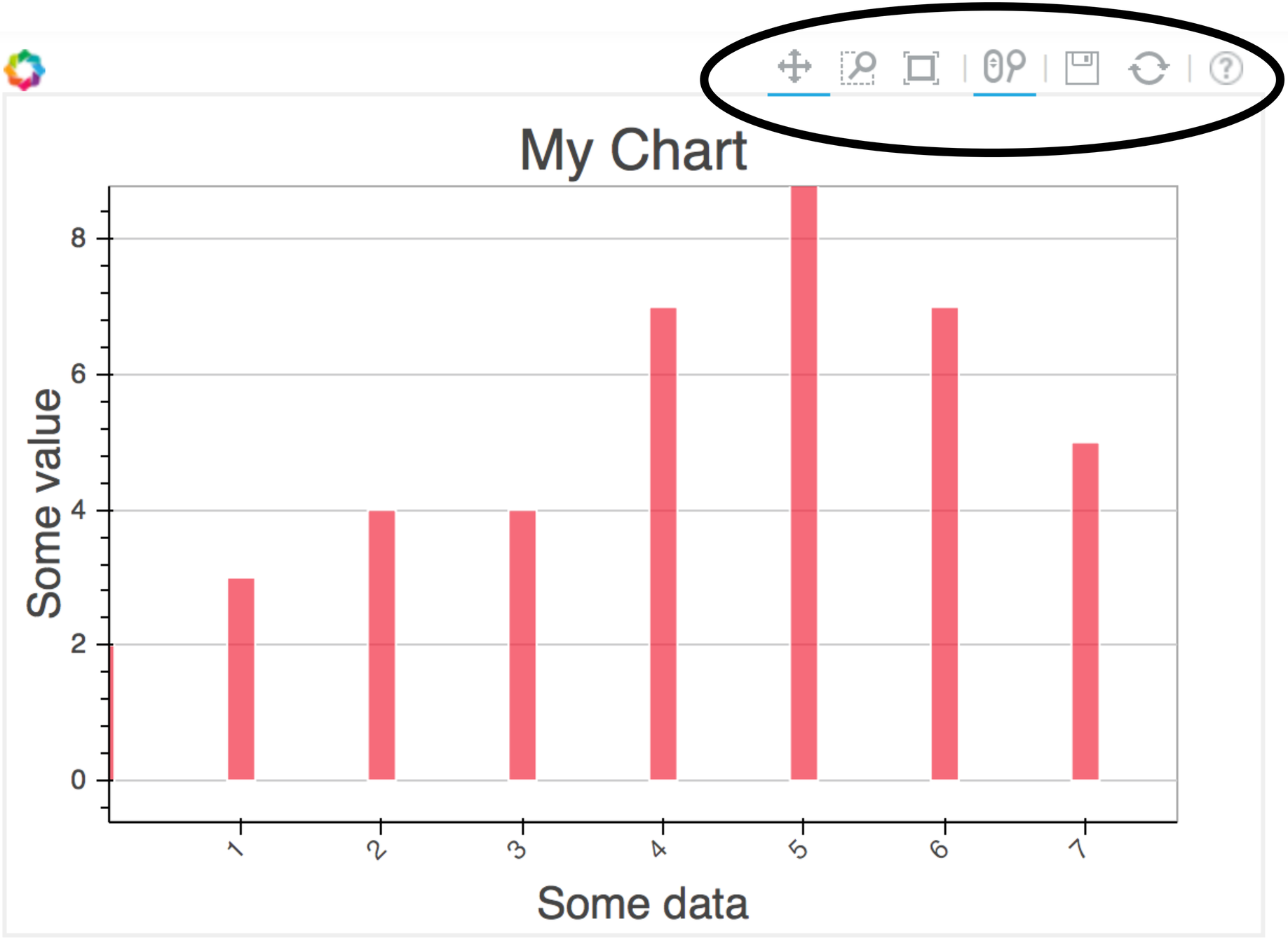
QlikView

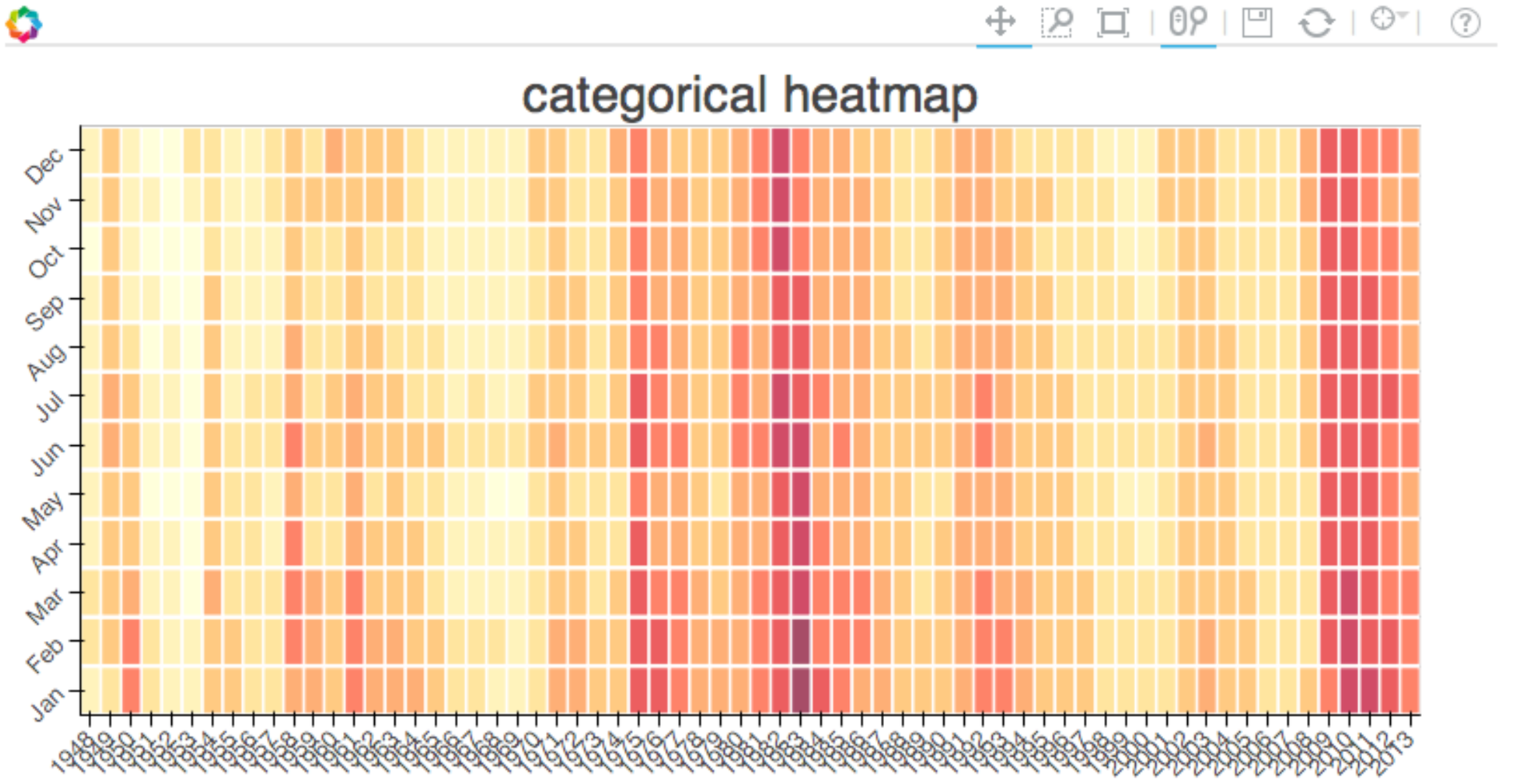

THE UNITED STATES OF AMERICA
FEDERAL RESERVE NOTE
AE 63456225 C
ONE HUNDRED DOLLARS


JavaScript

Easy to use… if you know R

Introducing Bokeh

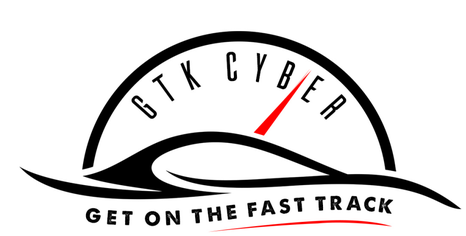

My Chart
Some value
0
2
4
6
8
1
2
3
4
5
6
7
Some data



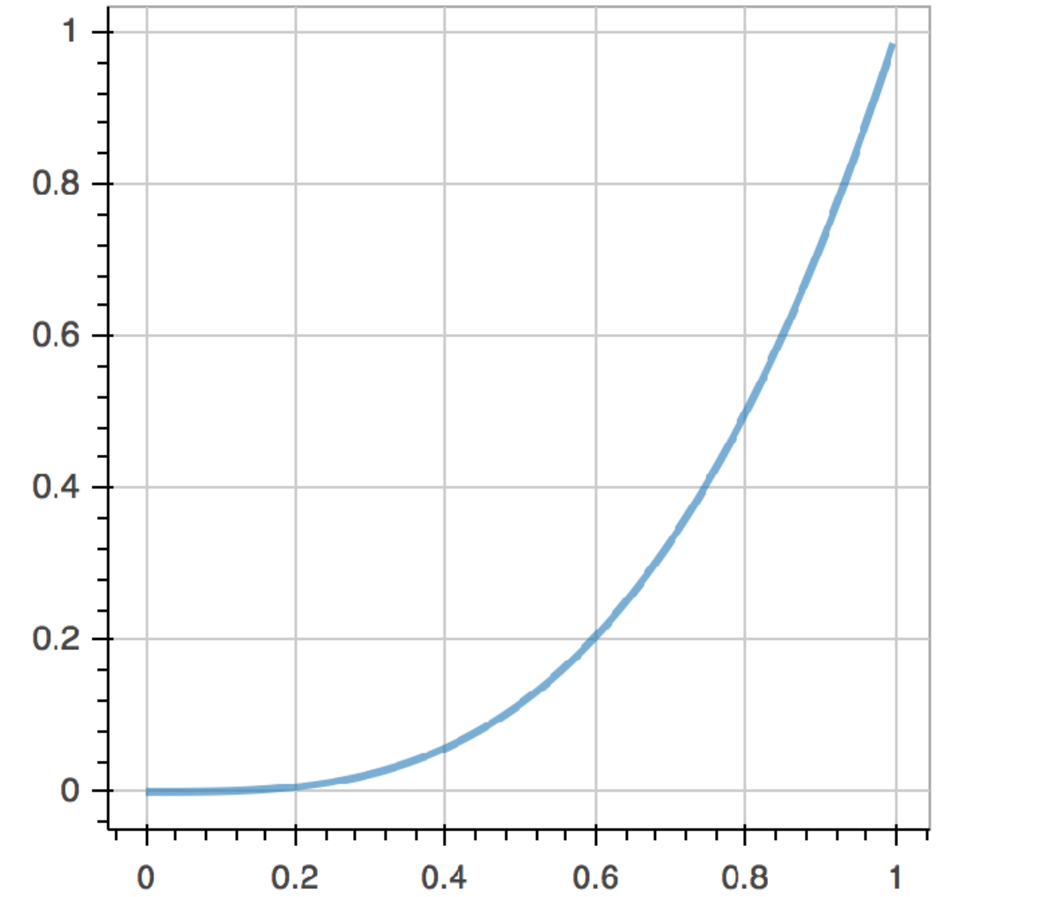
http://bokeh.pydata.org/en/latest/docs/gallery/cat_heatmap_chart.html

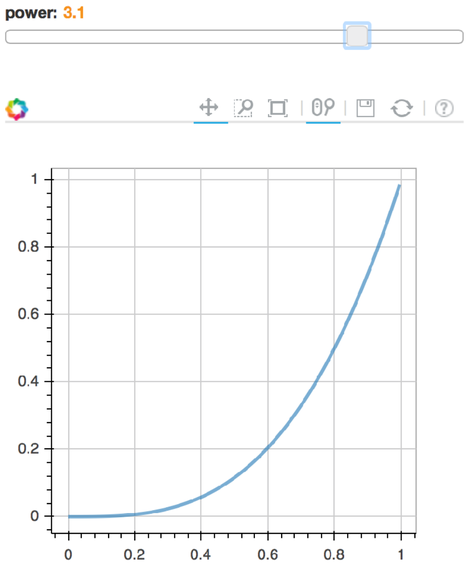

```
from bokeh.charts import HeatMap, output_file, show
from bokeh.palettes import YlOrRd9 as palette
from bokeh.sampledata.unemployment1948 import data

# pandas magic
df = data[data.columns[:-1]]
df2 = df.set_index(df[df.columns[0]].astype(str))
df2.drop(df.columns[0], axis=1, inplace=True)
df3 = df2.transpose()

output_file("cat_heatmap.html")

palette = palette[::-1]  # Reverse the color order so dark red is highest un
hm = HeatMap(df3, title="categorical heatmap", width=800, palette=palette)

show(hm)
```

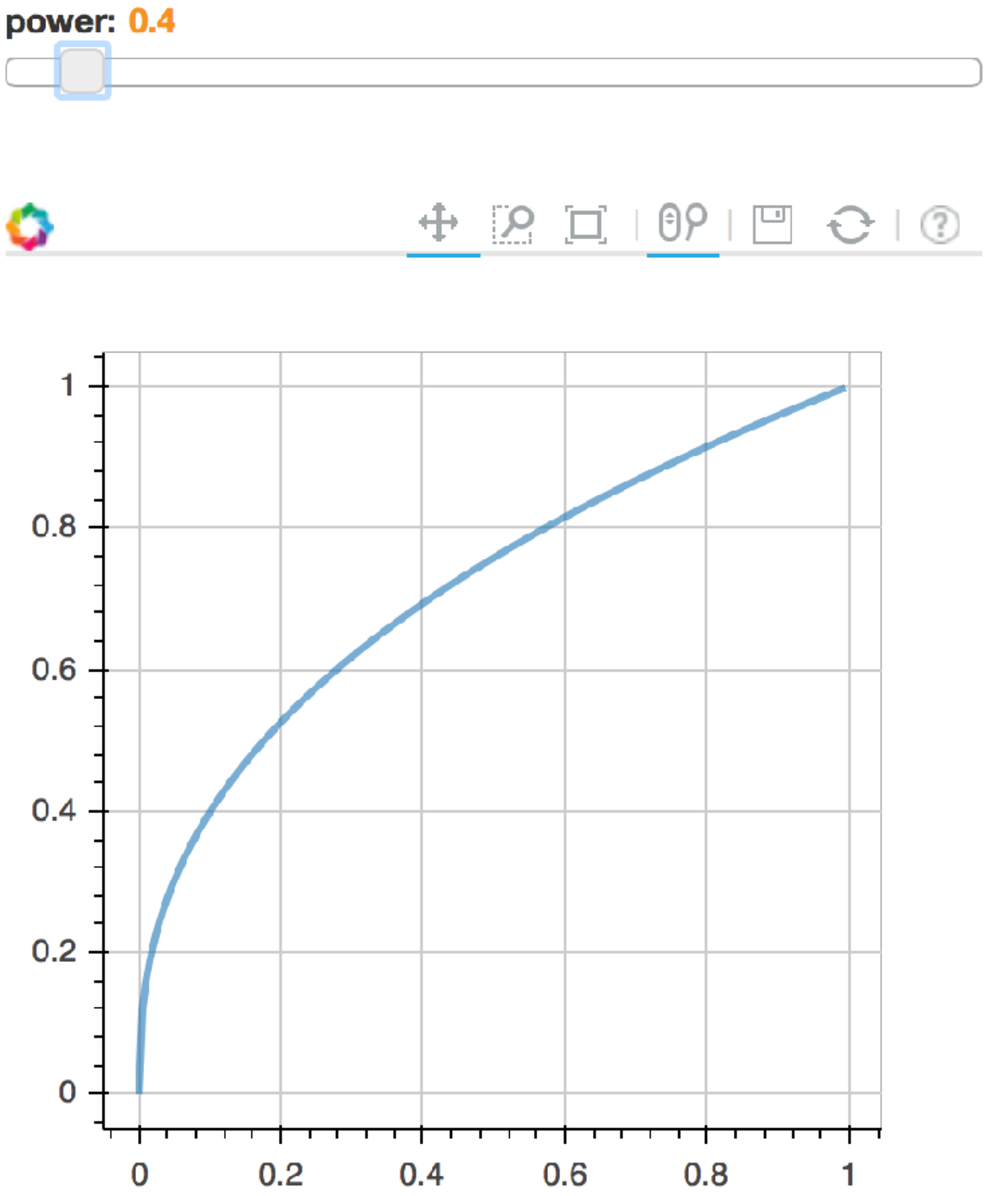
http://bokeh.pydata.org/en/latest/docs/gallery/cat_heatmap_chart.html



power: 3.1




power: 3.1
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


power: 0.4
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


```
from bokeh.io import vform
from bokeh.models import Callback, ColumnDataSource, Slider
from bokeh.plotting import figure, output_file, show

x = [x*0.005 for x in range(0, 200)]
y = x

source = ColumnDataSource(data=dict(x=x, y=y))

plot = figure(plot_width=400, plot_height=400)
plot.line('x', 'y', source=source, line_width=3, line_alpha=0.6)

callback = Callback(args=dict(source=source), code="""
        var data = source.get('data');
        var f = cb_obj.get('value')
        x = data['x']
        y = data['y']
        for (i = 0; i < x.length; i++) {
            y[i] = Math.pow(x[i], f)
        }
        source.trigger('change');
    """)

slider = Slider(start=0.1, end=4, value=1, step=.1, title="power", callback=callback)

layout = vform(slider, plot)

show(layout)
```

Using Bokeh

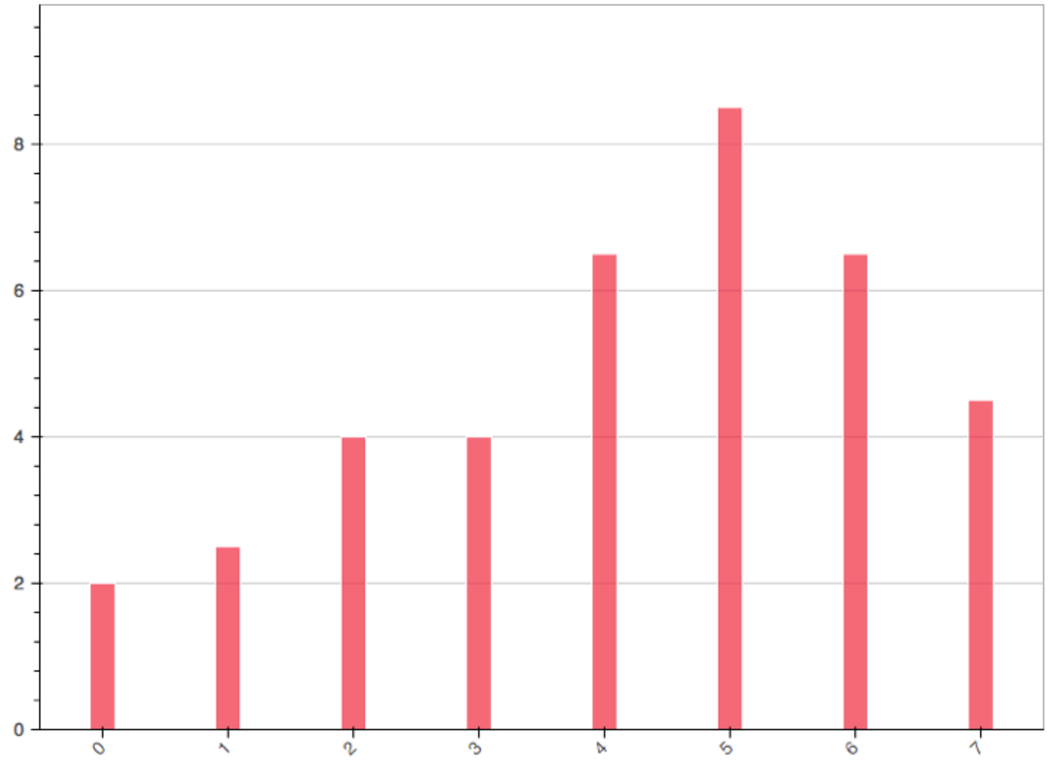

```
from bokeh.plotting import output_notebook
output_notebook()
```

- Area (Overlapped & Stacked)
- Bar (Grouped & Stacked)
- BoxPlot
- Donut (Ick!)
- HeatMap
- Line
- Scatter
- Step
- Timeseries


```
from bokeh.charts import <chartname>
from bokeh.io import show
```


```
from bokeh.charts import Bar, show
data = [2,3,4,4,7,9,7,5]

barchart = Bar( data, notebook=True )
show()
```


```
from bokeh.charts import Bar
from bokeh.io import show
data = [2,3,4,4,7,9,7,5]

barchart = Bar( data, notebook=True )
show( bar chart )
```



Method Chaining


```
barchart =
   Bar( data, title="My Chart", xlabel="Categories", ylabel="Value",
notebook=True )

show( barchart )
```


```
barchart =
   Bar( data, title="My Chart", xlabel="Categories", ylabel="Value",
notebook=True )

barchart.show()
```

```
barchart3 = Bar(data)
   .title( "My Chained Chart" )
   .notebook( True )
   .legend( True )
   .xlabel( "Categories" )
   .ylabel( "Value" )

show( barchart3 )
```

Bokeh Config Options

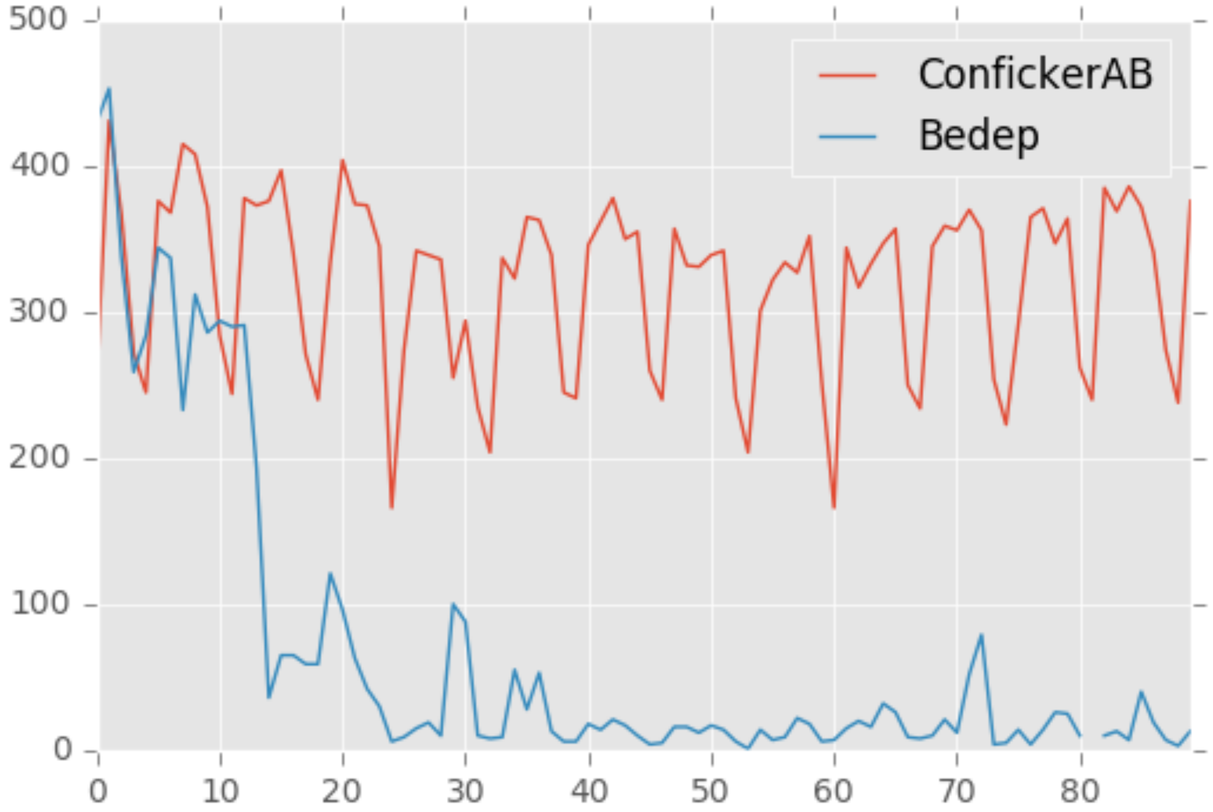
- **title** (str): the title of your plot.
- **xlabel** (str): the x-axis label of your plot.
- **ylabel** (str): the y-axis label of your plot.
- **legend** (str, bool): the legend of your plot.
- **xscale** (str): the x-axis type scale of your plot.
- **yscale** (str): the y-axis type scale of your plot.
- **width** (int): the width of your plot in pixels.
- **height** (int): the height of you plot in pixels.
- **tools** (bool): to enable or disable the tools in your plot.
- **filename** (str or bool): the name of the file where your plot will be written.
- **server** (str or bool): the name of your plot in the server.
- **notebook** (bool):if you want to output (or not) your plot into the IPython notebook.

In Class Exercise

Please complete Worksheet 5: Data Visualization

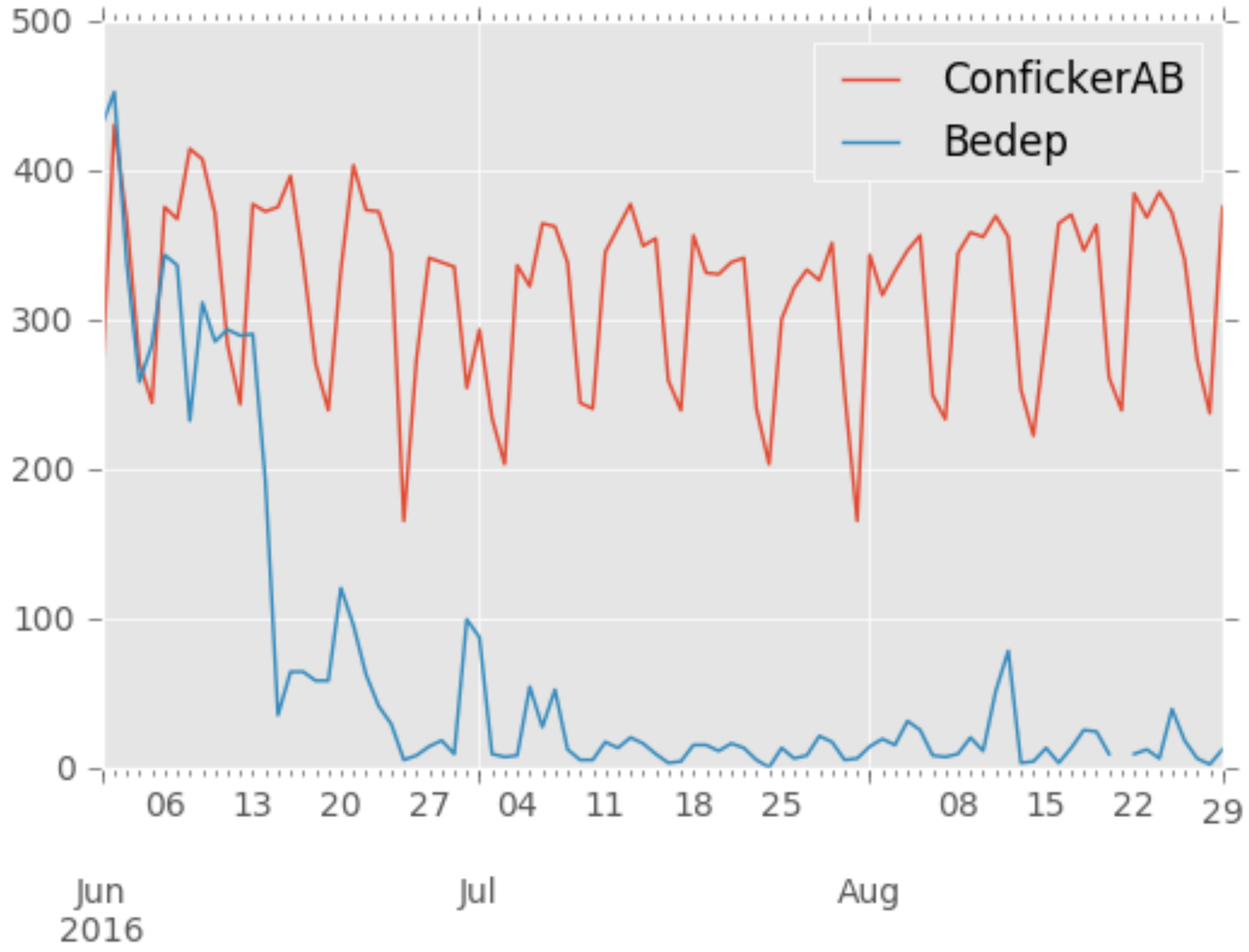
Exercise 1

```
data = pd.read_csv('../data/dailybots.csv')
filteredData = data[data['industry'] == "Government/Politics"]
filteredData2 = filteredData[filteredData['botfam']== 'ConfickerAB' ][['date','hosts']]
filteredData2.columns = ['date', 'ConfickerAB']
filteredData2.date = pd.to_datetime( filteredData2.date )
finalData.plot(kind='line' )
```



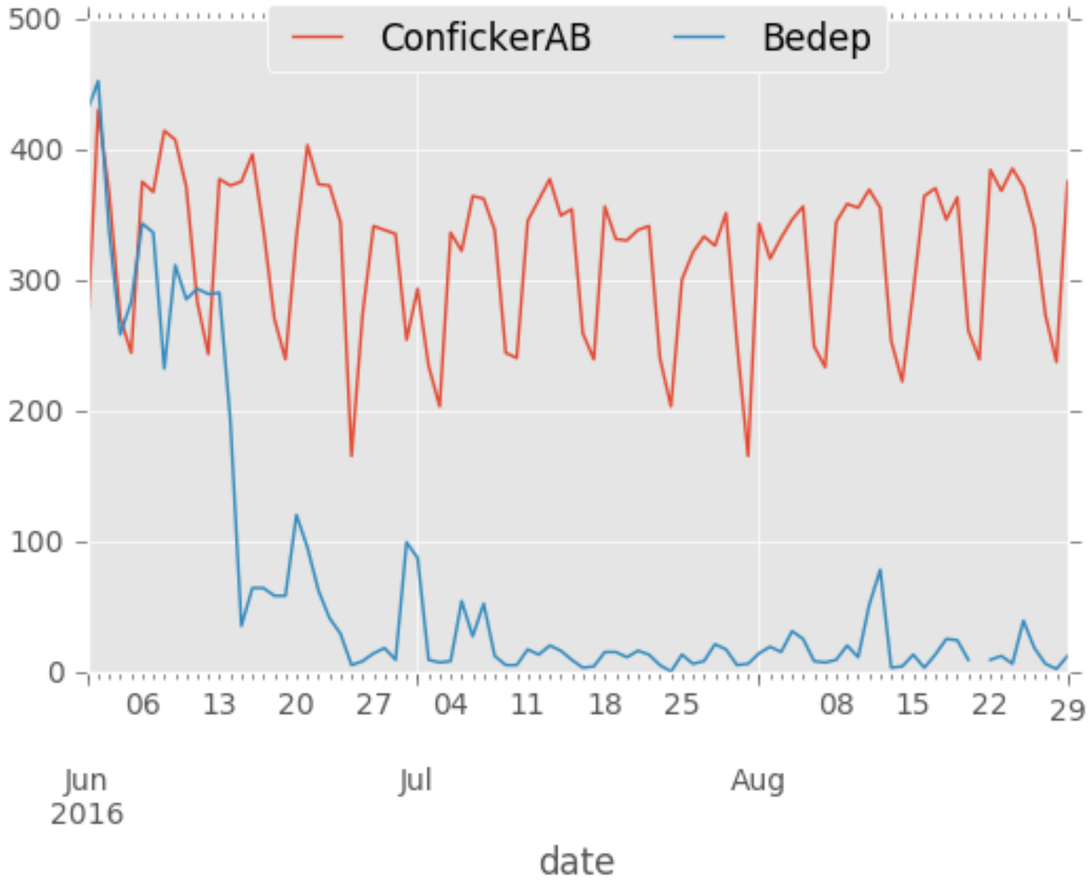
Exercise 1

```
finalData.set_index('date', inplace=True)
finalData.plot(kind='line' )
```



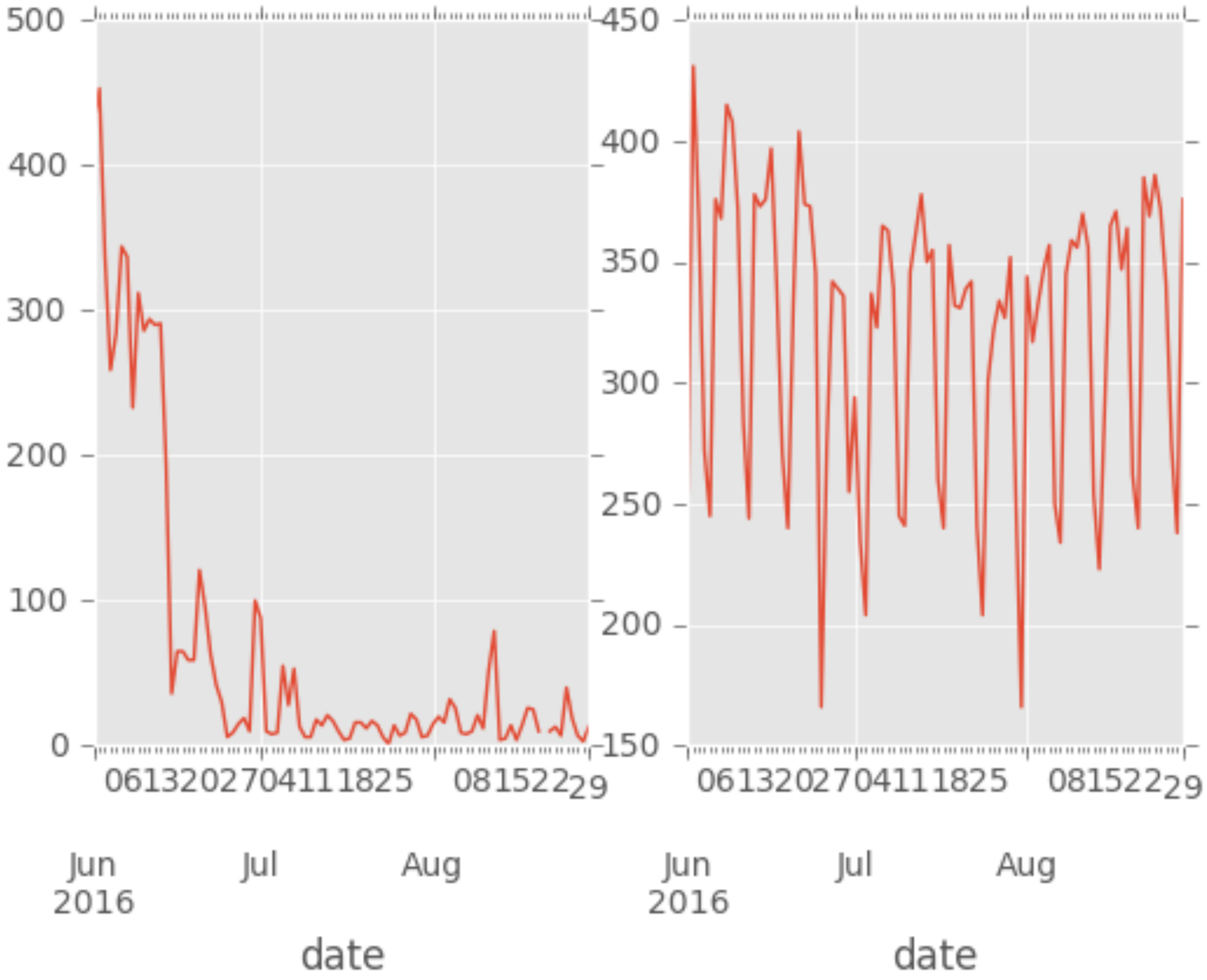
Exercise 1

```
nicePlot = finalData.plot( kind="line")
nicePlot.legend(loc='upper center', bbox_to_anchor=(0.5, 1.05),
          ncol=3, fancybox=True, shadow=False)
```



Exercise 1

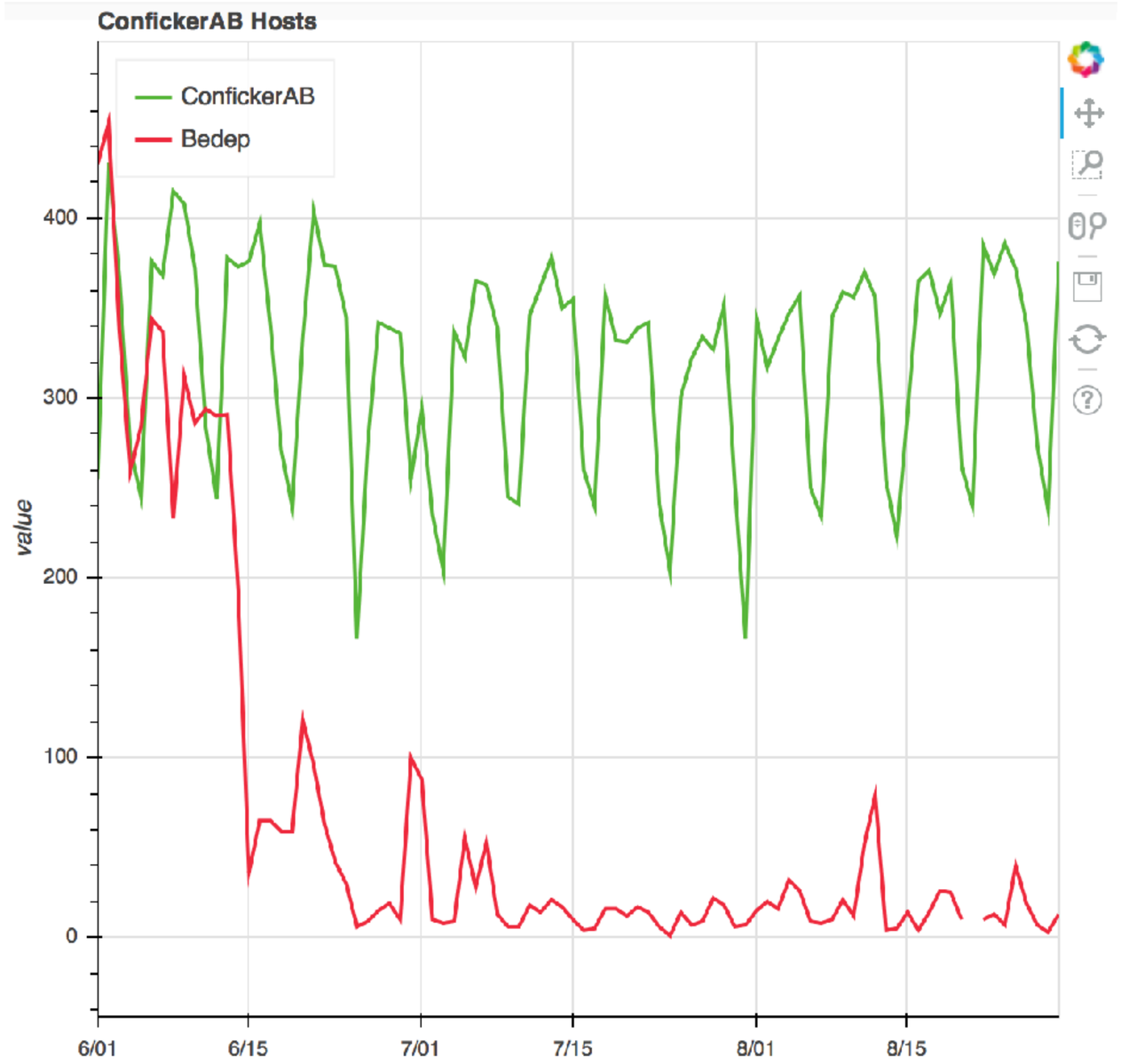
```
fig, axes = plt.subplots(nrows=1, ncols=2)

finalData['ConfickerAB'].plot()
finalData['Bedep'].plot(ax=axes[0])
```



Exercise 2

```
from bokeh.plotting import output_notebook
output_notebook()
from bokeh.charts import TimeSeries
from bokeh.io import show

linechart = TimeSeries( data=finalData,
                        title="ConfickerAB Hosts",
                        legend="top_left" )
show( linechart )
```

Exercise 2


ConfickerAB Hosts
Conficker AB
Bedep
value
400
300
200
100
0
6/01
6/15
7/01
7/15
8/01
8/15