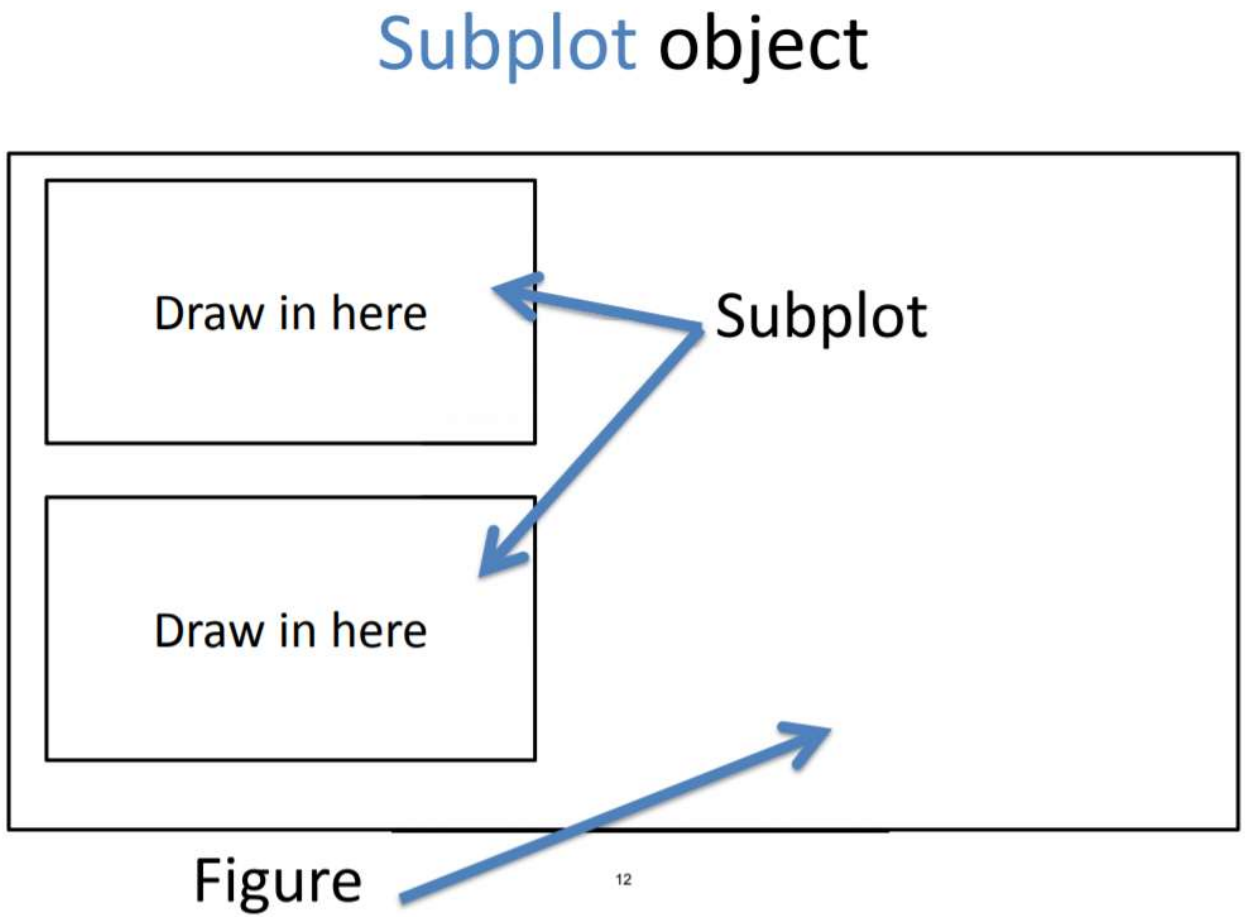
Lab 4: Visualization

Three principles for visualization

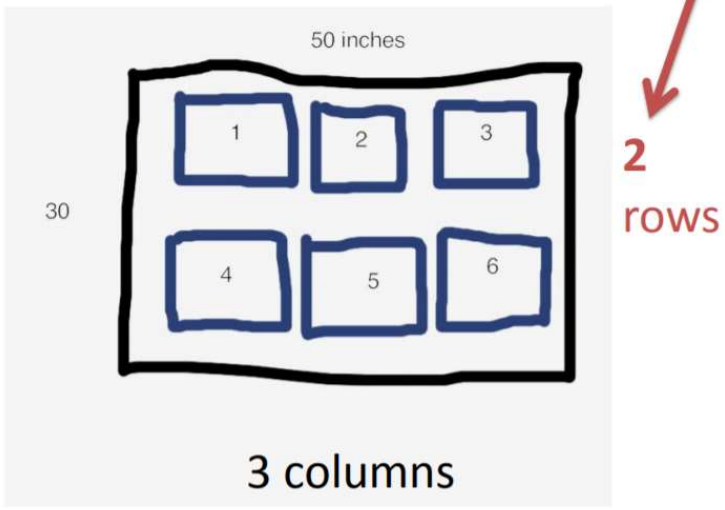
1. be true to your research – design your display to illustrate a particular point
2. maximize information, minimize ink –use the simplest possible representation for the bits you want to convey
3. organize hierarchically – what should a viewer see first? what if they look deeper?

Matplotlib: architecture



Columns and rows of hierarchical figures

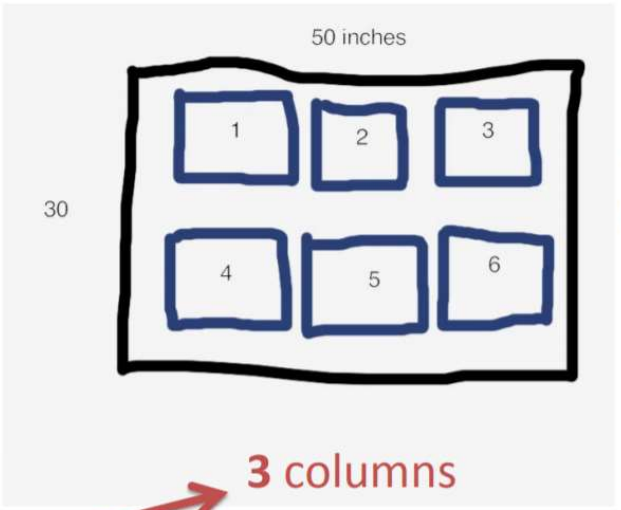
figure.add_subplot(**2**, 3, 1)



2 rows

14

figure.add_subplot(2, **3**, 1)



2 rows

15

Code

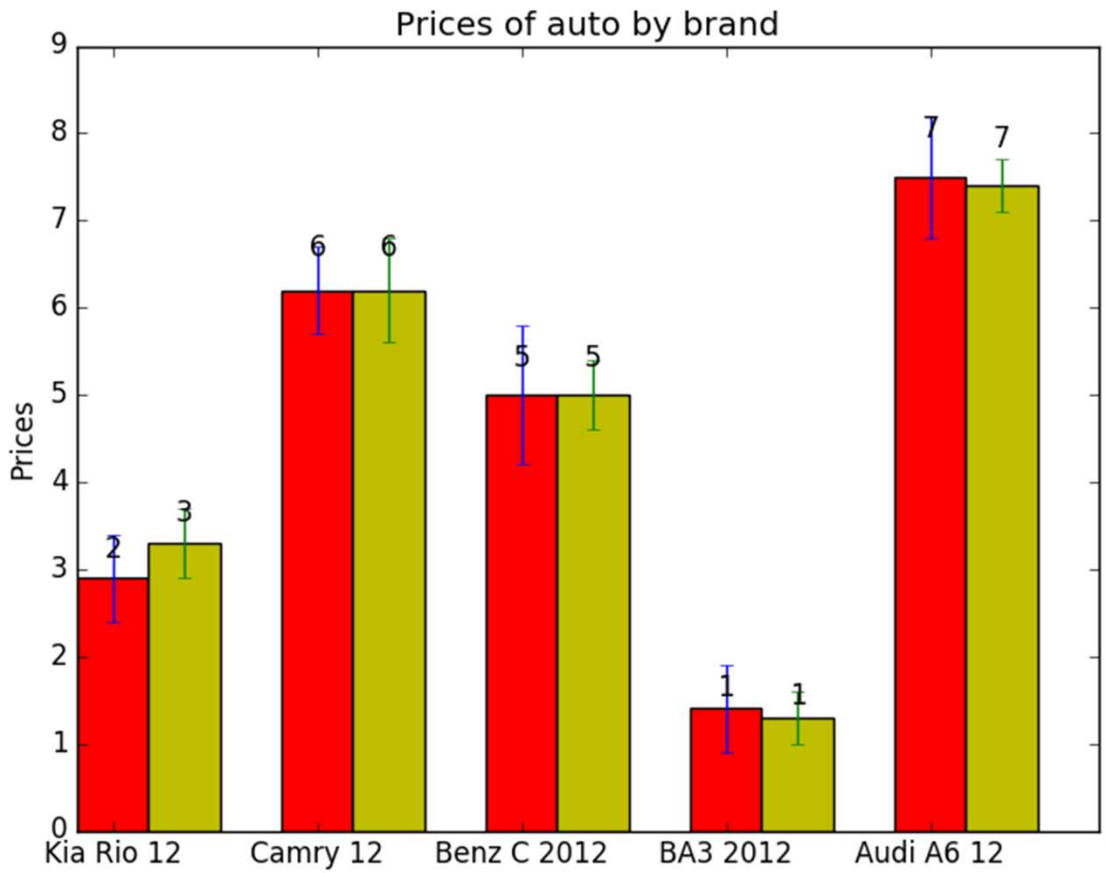
```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(50, 30))
subplot1 = fig.add_subplot(2, 3, 1)
subplot2 = fig.add_subplot(2, 3, 2)
```

Worked example (simple)

- Bar graph with standard errors:
- Box plots
- Hist pllots
- Scatter plots
- Choropleth plots

- Summary statistics
- Distribution is important
- Fancier is not always better

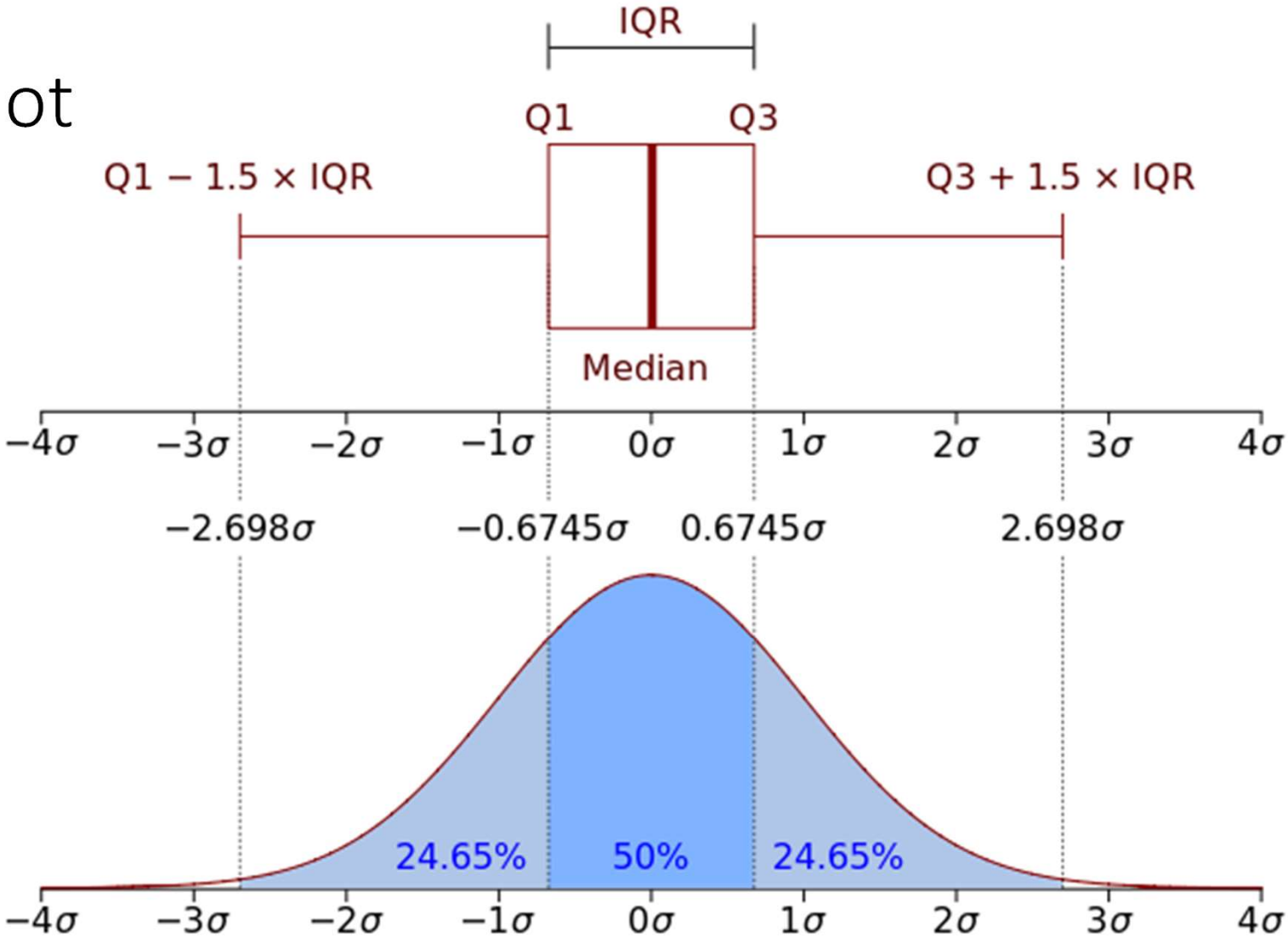
Bar graph with standard errors



Bar graph with standard errors: matplotlib

```
ast_means = (3.3, 6.2, 5, 1.3, 7.4)
ast_std = (0.4, 0.6, 0.4, 0.3, 0.3)
rects2 = ax.bar(ind + width, ast_means, width, color='y', yerr=ast_std)

# add some text for labels, title and axes ticks
ax.set_ylabel('Prices')
ax.set_title('Prices of auto by brand')
ax.set_xticks(ind + width / 2)
ax.set_xticklabels(('Kia Rio 12', 'Camry 12', 'Benz C 2012', 'BA3 2012', 'Audi A6 12'))
```

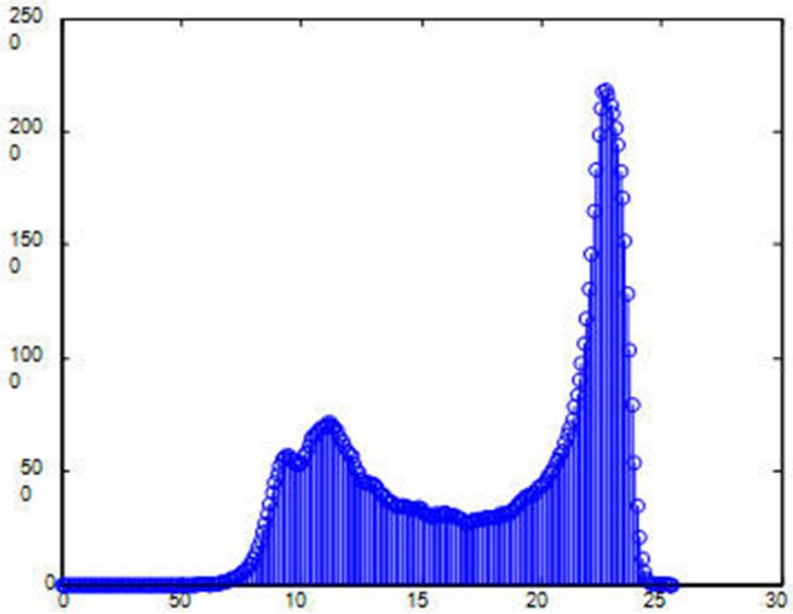
Box plot

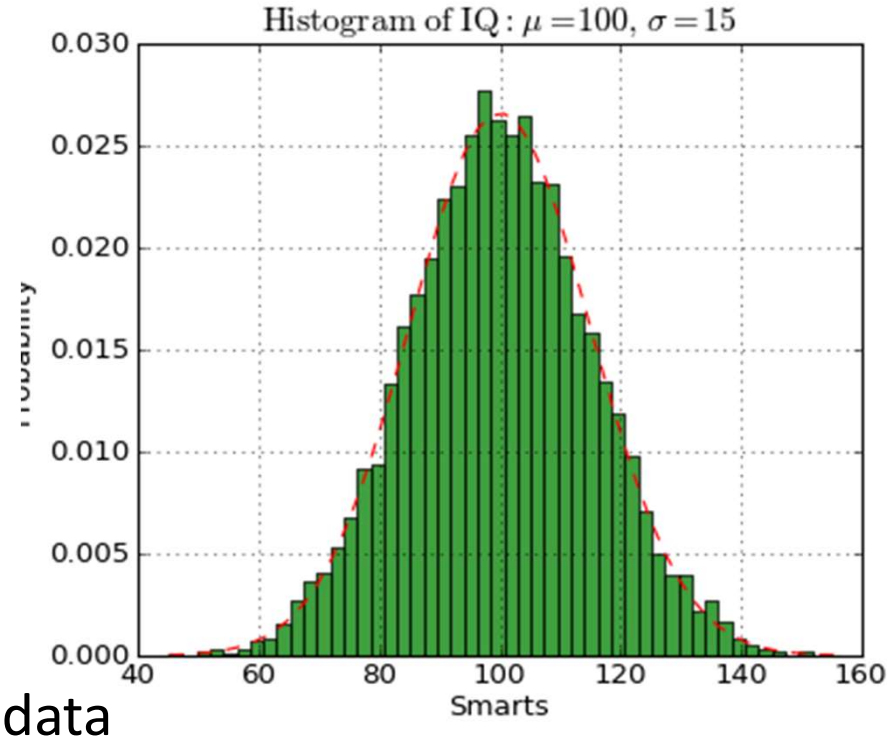

IQR
Q1
Q3
Q1 − 1.5 × IQR
Q3 + 1.5 × IQR
Median
−4σ
−3σ
−2σ
−1σ
0σ
1σ
2σ
3σ
4σ
−2.698σ
−0.6745σ
0.6745σ
2.698σ
24.65%
50%
24.65%

Box plot

```
import matplotlib.pyplot as plt
import numpy as np

# fake up some data
spread = np.random.rand(50) * 100
center = np.ones(25) * 50
flier_high = np.random.rand(10) * 100 + 100
flier_low = np.random.rand(10) * -100
data = np.concatenate((spread, center, flier_high, flier_low), 0)

# basic plot
plt.boxplot(data)
```

Histogram





- Important first way of looking at your data
- One dimensional
- Shows shape by binning a continuous distribution

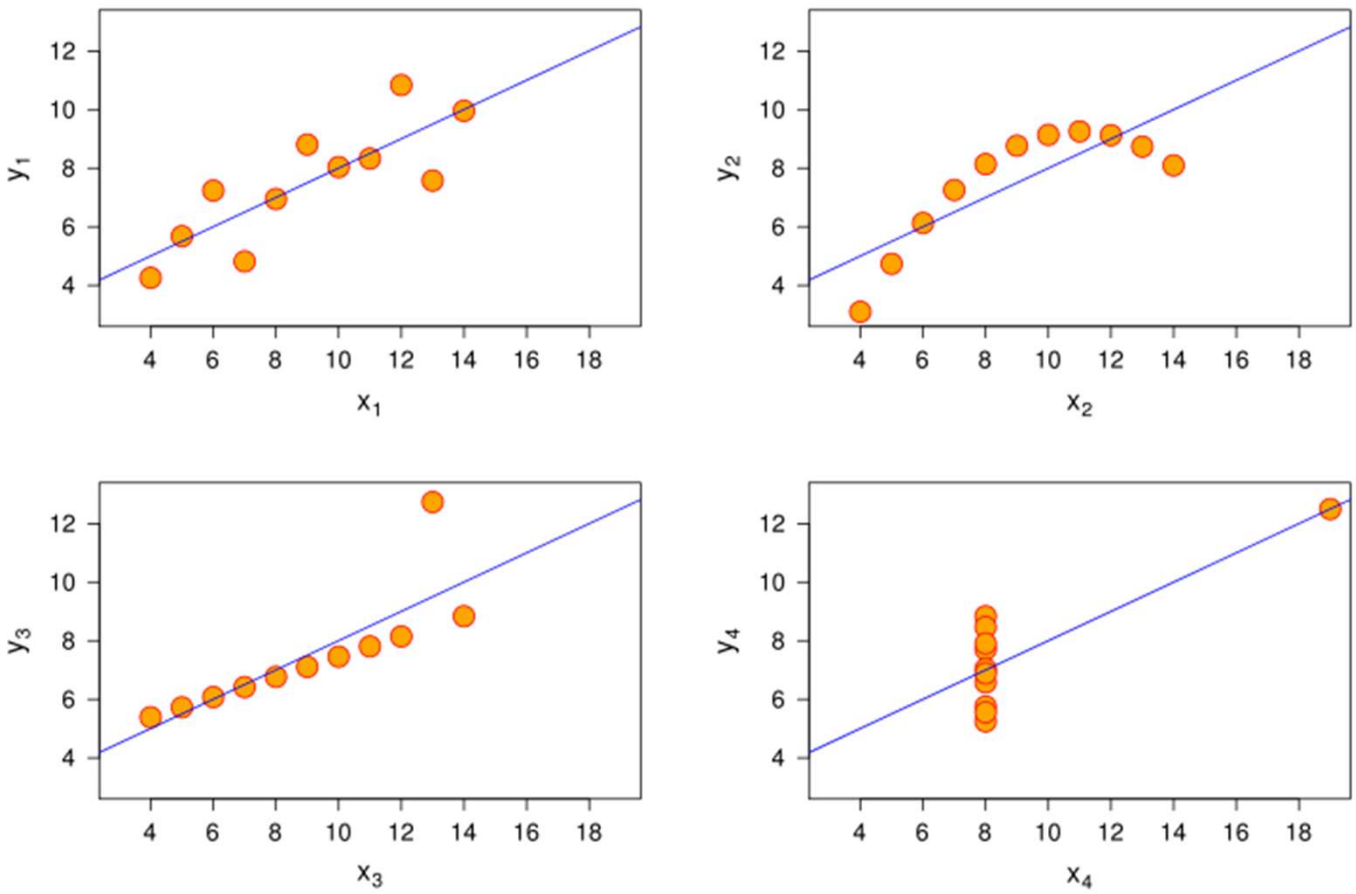
Histogram

```
mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 50, normed=1, facecolor='green', alpha=0.75)

# add a 'best fit' line
y = mlab.normpdf( bins, mu, sigma)
l = plt.plot(bins, y, 'r--', linewidth=1)

plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title(r'$\mathrm{Histogram\ of\ IQ:}\ \mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
```

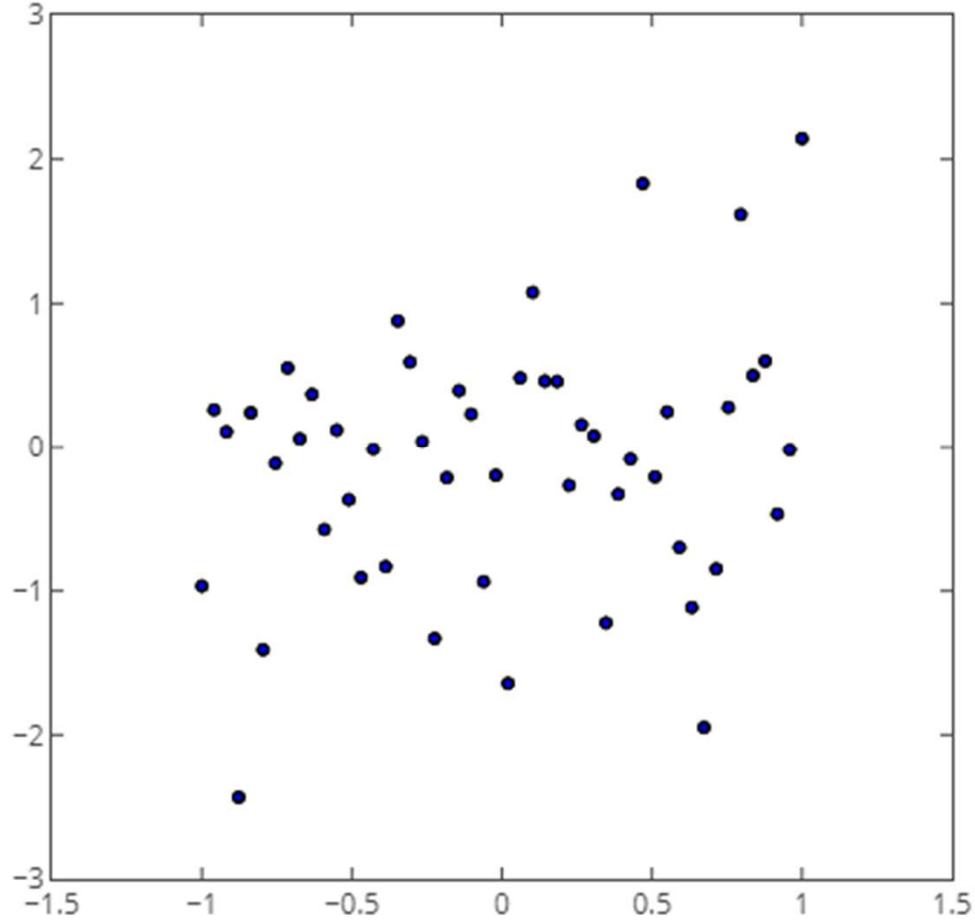
Anscombe’s Quartet (Квартет Энскомба)

I		II		III		IV	
x	y	x	y	x	y	x	y
10,0	8,04	10,0	9,14	10,0	7,46	8,0	6,58
8,0	6,95	8,0	8,14	8,0	6,77	8,0	5,76
13,0	7,58	13,0	8,74	13,0	12,74	8,0	7,71
9,0	8,81	9,0	8,77	9,0	7,11	8,0	8,84
11,0	8,33	11,0	9,26	11,0	7,81	8,0	8,47
14,0	9,96	14,0	8,10	14,0	8,84	8,0	7,04
6,0	7,24	6,0	6,13	6,0	6,08	8,0	5,25
4,0	4,26	4,0	3,10	4,0	5,39	19,0	12,50
12,0	10,84	12,0	9,13	12,0	8,15	8,0	5,56
7,0	4,82	7,0	7,26	7,0	6,42	8,0	7,91
5,0	5,68	5,0	4,74	5,0	5,73	8,0	6,89

Anscombe’s Quartet (Квартет Энскомба)



Scatter plot



Scatter plot

```
import numpy as np
import matplotlib.pyplot as plt


N = 50
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2  # 0 to 15 point radii

plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```

Intro ToDO

- Exploring
- Storytelling

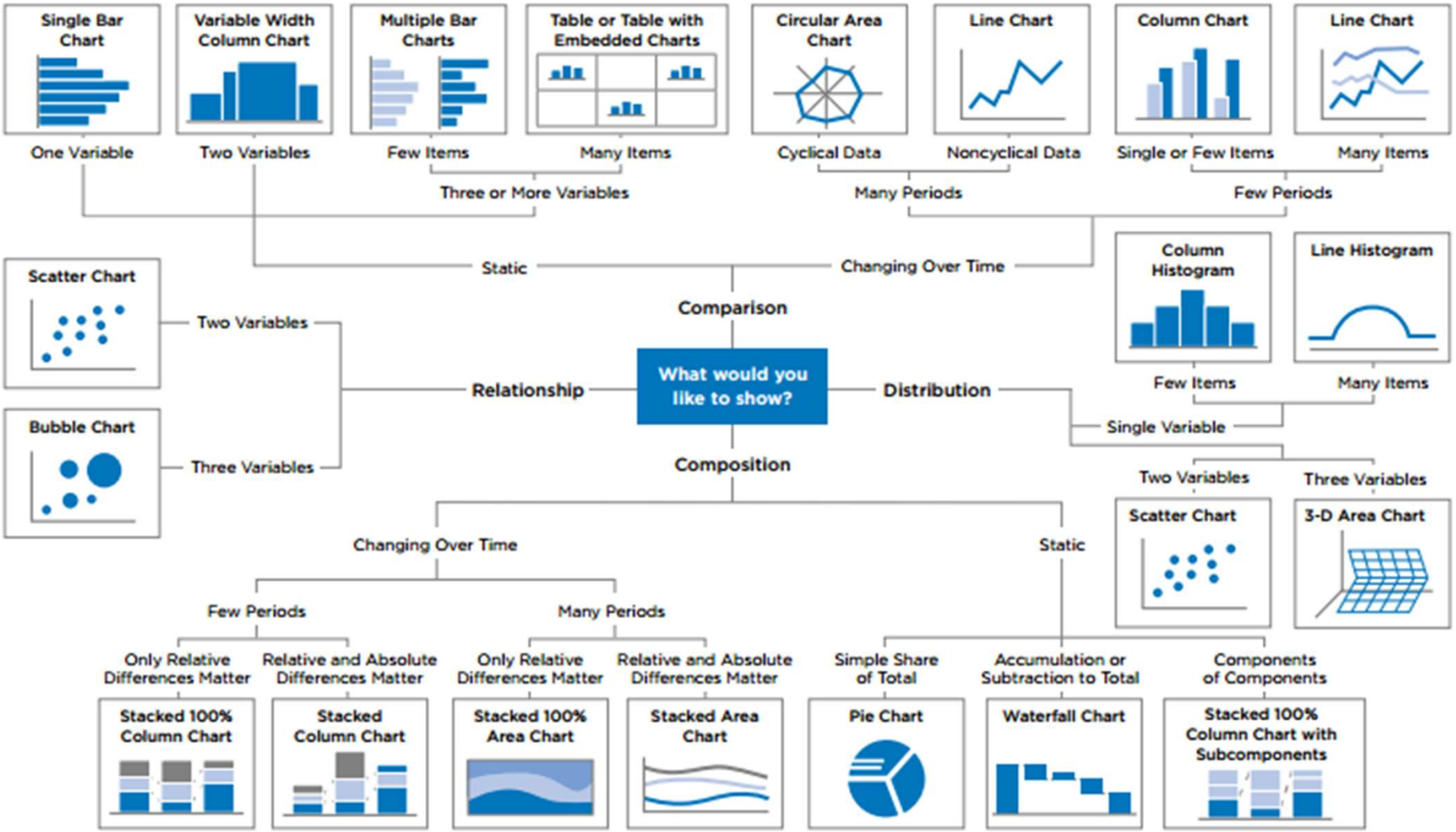
Mapping data to a visual representation

- 1. What dimensions matter
- 2. What vocabulary elements (color, shape, orientation) will map to those dimensions
- Three principles
- 1. be true to your research
- 2. maximize information, minimize ink
- 3. organize hierarchically

Others plots

- Grouped histogram
- Piechart
- Stacked bar graph
- Venn diagram
- Scatter plot
- Line graph
- Heat map
- Bubble plot
- Bar graph with\out standard errors
- Box plot
- Viola plot
- Strip chart with\out means

SELECTING THE APPROPRIATE CHART FOR STRATEGY PRESENTATIONS



```
# this creates an array of grey colors from white to black
colors = ['0','1','2','3','4','5','6','7','8','9','a', 'b', 'c', 'd', 'e', 'f']
colors = map(lambda s: '#%s' % (s*6), colors)
colors.sort(reverse=True)
# assume MAXDONATION was defined
# assume curdonation is the donation to pick a color for
ratio = (curdonation/float(MAXDONATION))
color_idx = int( ratio * ( len(colors) - 1) )
colors[color_idx]
```

Recap

- Always start by looking at your data with the simplest visualizations possible. For most datasets, a scatter plot or line graph is sufficient.
- First view a summary of the whole dataset so that you know which subsets are worth visualizing in more detail, and how significant the details really are.
- Plot your interesting data along a bunch of different dimensions.
- Stare at your data, try to identify trends, outliers and other interesting regions and form hypotheses.
- Use statistics to see if your hypotheses were correct (tomorrow's lecture)
- Repeat

Others frameworks

- http://etetoolkit.org/
- http://www.jfreechart.org/
- http://plot.ly/

Exercise 1: Quantile plot of your project data
Exercise 2: Scatter plots of project data