Data Analysis and Interpretation

Recap: Data storage with pandas

- Pandas is an open source library
  - Process relational data in memory
  - Support SQL-like query
- Rich relation data tool built on top of NumPy
  - Excellent performance
  - Easy-to-use, highly consistent API
- A foundation for data analysis in Python
  - It also has built-in visualization features
  - It can work with data from a wide variety of sources
  - Takes data preparation and preprocessing to the next level

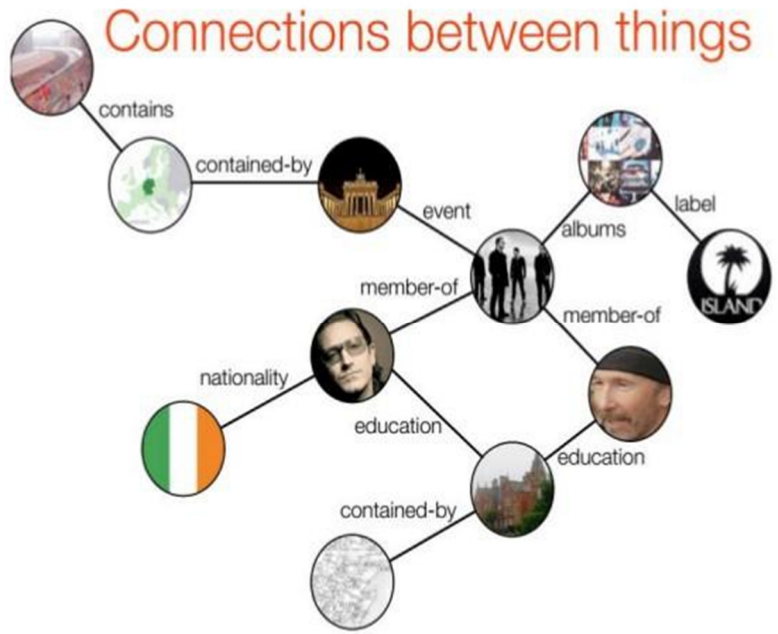
https://pandas.pydata.org/

Recap: Other data models

- Document model (e.g. XML)

```
<contact>
   <id>656</id>
   <firstname>Chuck</firstname>
   <lastname>Smith</lastname>
   <phone>(123) 555-0178</phone>
</contact>
```

- Network model (e.g. graph DB)



Recap: Data Normalisation

- Why data normalisation:
  - Non-normalized data might affect analysis results.
  - Data from different sources may be in different scales, ranges, and units → Meters and Feet.
  - Data may be biased/skewed.
- When data normalization:
  - Data should be in the same scale
  - Bias data into a specific range/size for specific purposes

Recap: Data Cleaning

- Types of dirty data:
  - **Formatting**: the same entity can be inconsistently formatted
  - **Missing data**: some of the data are not there
  - **Erroneous data**: recurring error data in a particular case
  - **Irrelevant data**: data whose non-existence does not affect your results
  - **Inconsistent data**: the same data can be represented in different ways
  - **Malicious data**: data is intended to cause undesired effects
  - **Outliers**: observation points that is distant from other observations (a.k.a. noises, anomalies)

3803ICT course structure

W1. Introduction to Data Analytics

Data Preparation and Preprocessing

W2. Data Preparation and Preprocessing

Data Analysis and Interpretation

W3. Exploratory Data Analytics

W4. Statistical Data Analytics

W5. Predictive Data Analytics

Visualization

W6. Data Visualization

Analysis of special types of data

W7. Time Series

W8. Textual Data

W9. Graph Data

Analysis with big data infrastructure

W10. Distributed Data Analysis

W11. Cloud-based Data Analysis

W12. Revision

Learning Outcomes

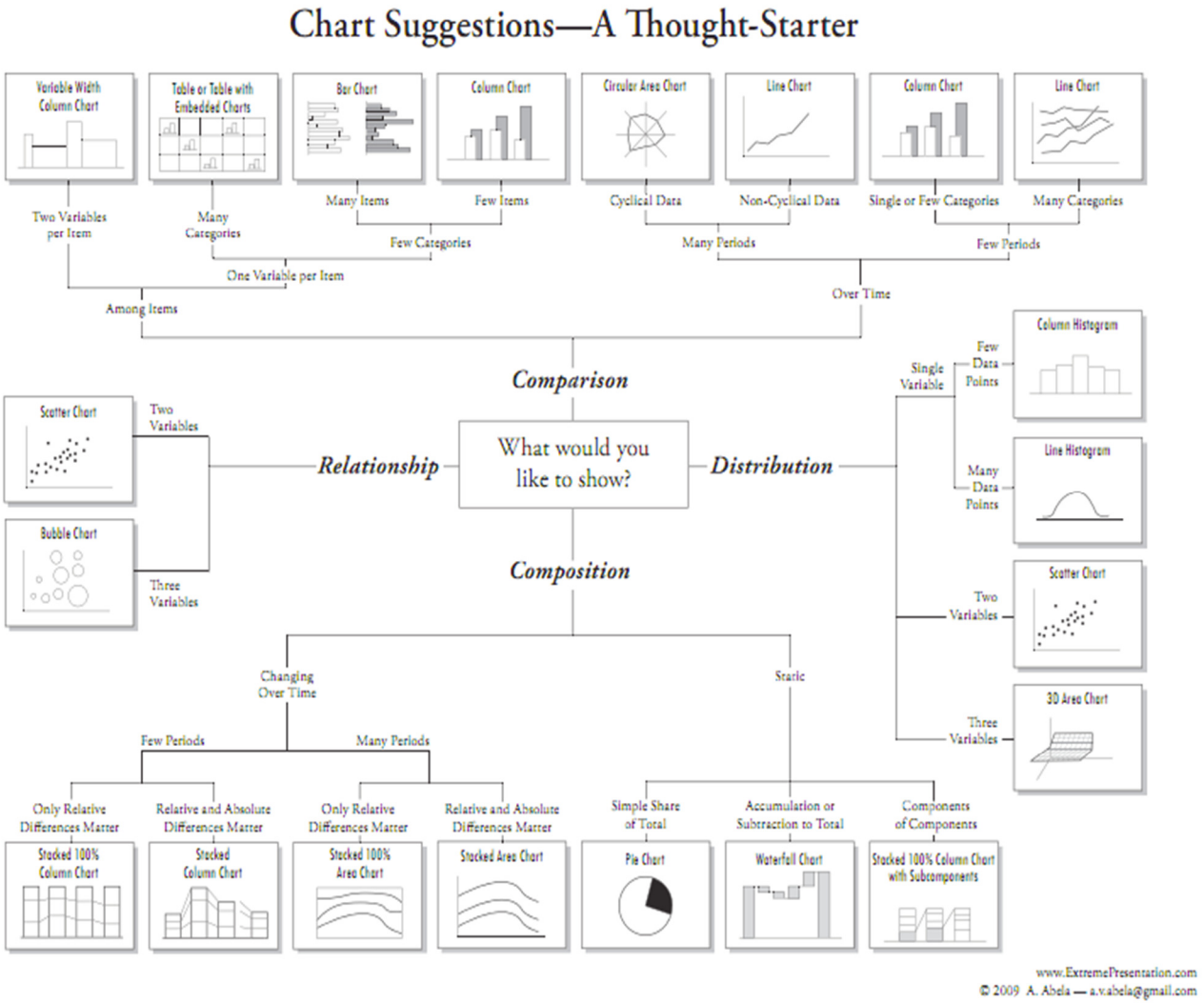
- At the end of this lecture students will be able to know:
  - Different charts and plots
  - Different types of exploratory data analysis
  - Dimensionality Reduction
  - Exploratory data analysis with Matplotlib and Seaborn

1. Charts and Plots

- **Basic charts:**
  - Line chart: trends over time
  - Bar chart: comparing categories
  - Histogram chart: advanced bar chart
  - Pie chart: proportions
  - Scatter plot: correlations
  - Heat map: advanced color usage
  - Box-and-whisker Plot: distribution
- Advanced charts

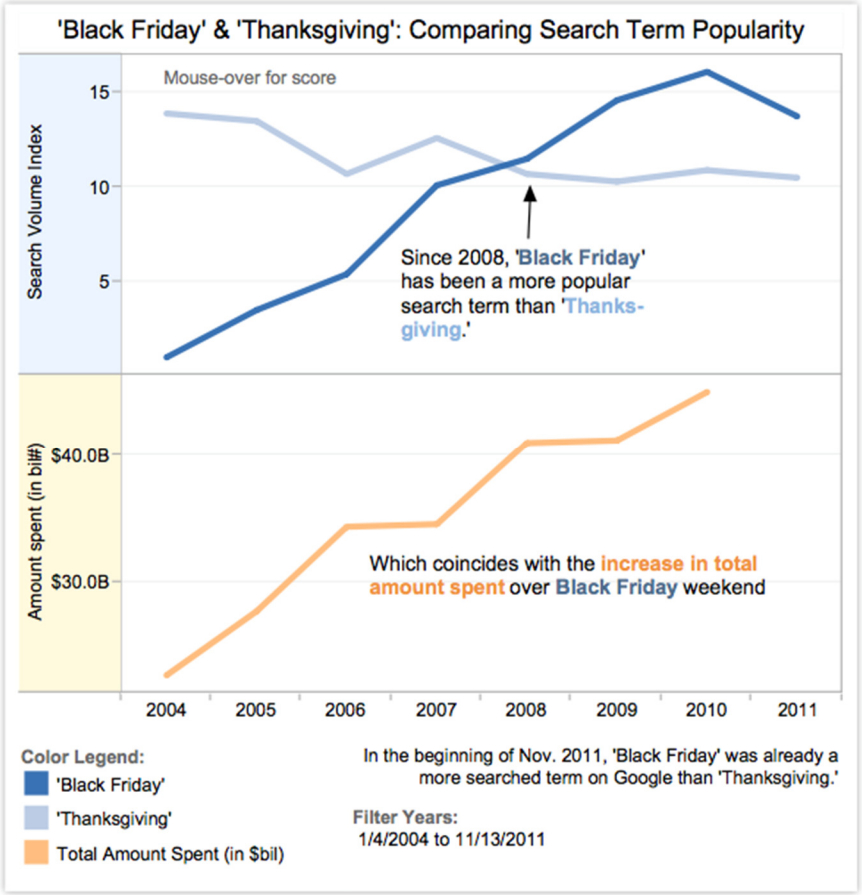
Chart principles

- https://www.tableau.com/solutions/gallery



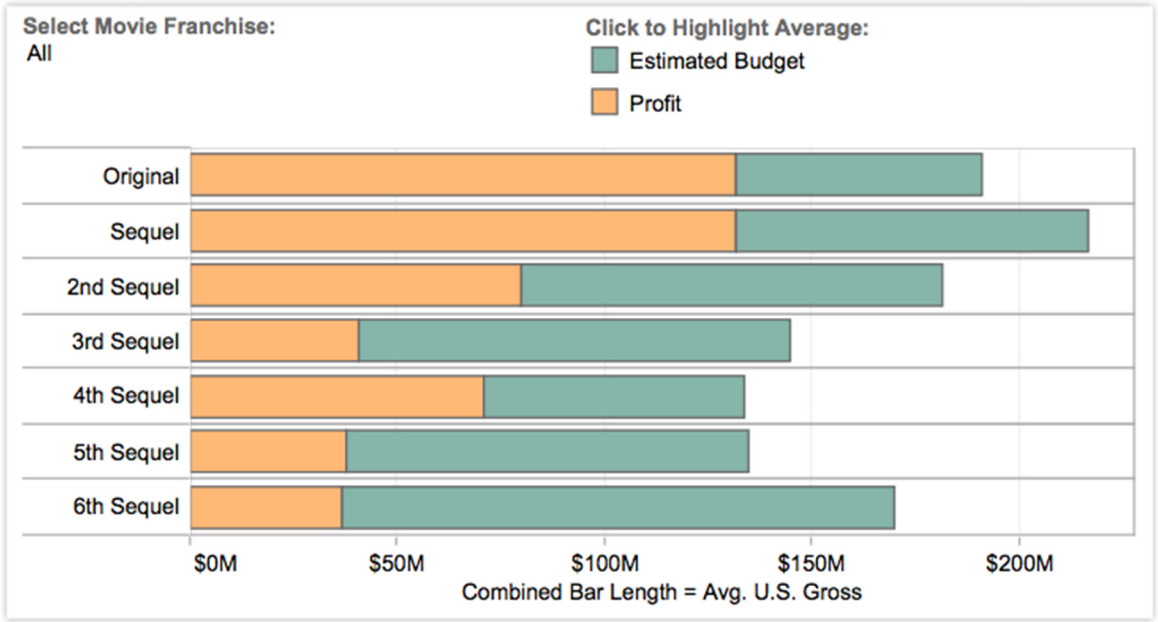
Line chart

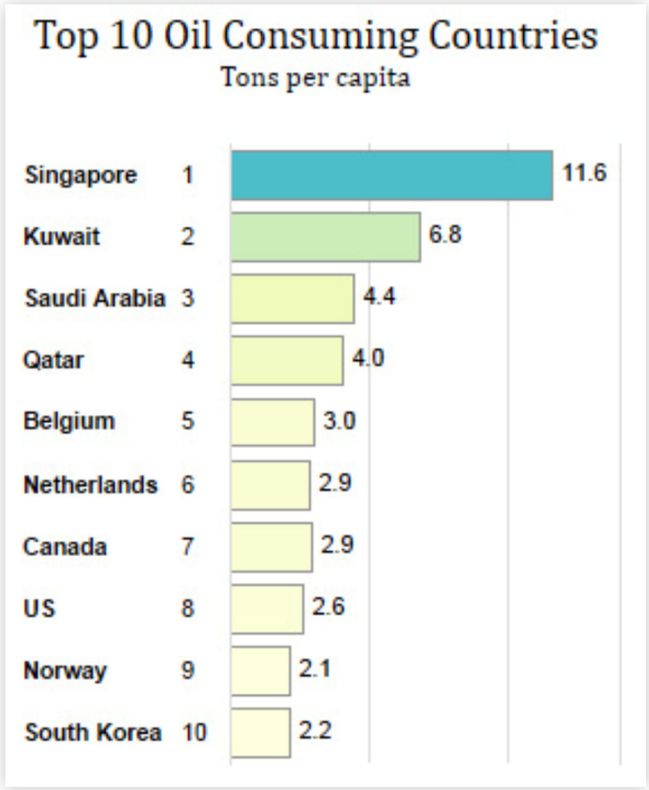
- Connect individual ordered data points
- When to use: viewing trends in data over time
  - Examples: stock price change over a five-year period, website page views during a month, revenue growth by quarter



Bar chart

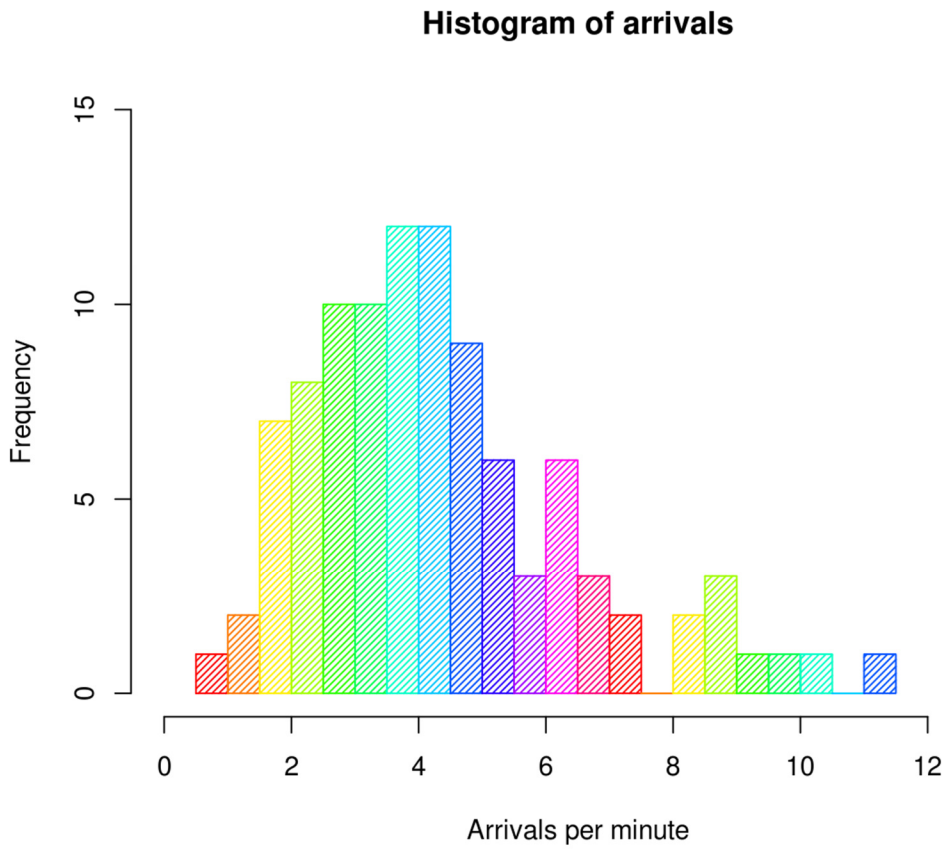
- When to use: comparing/ranking data across categories
- Effective for ordered data in different categories
  - Examples: Volume of shirts in different sizes, website traffic by origin site, percent of spending by department





Histogram chart

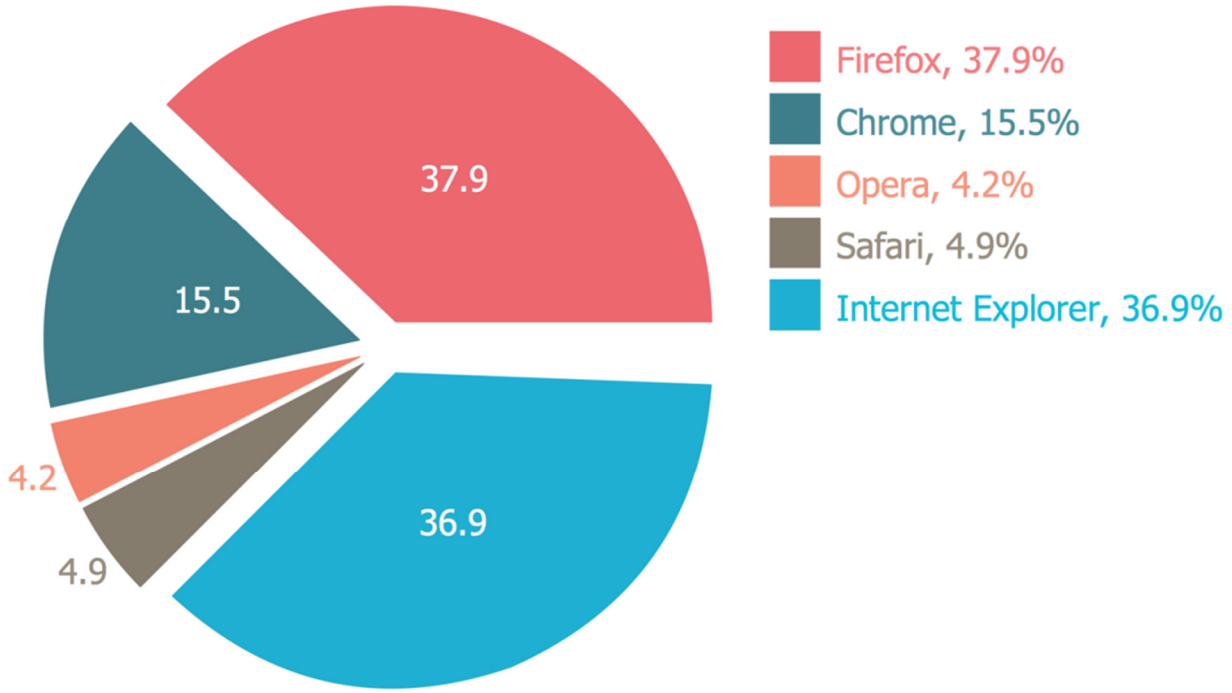
- When to use: understanding the distribution of your data



https://en.wikipedia.org/wiki/Histogram

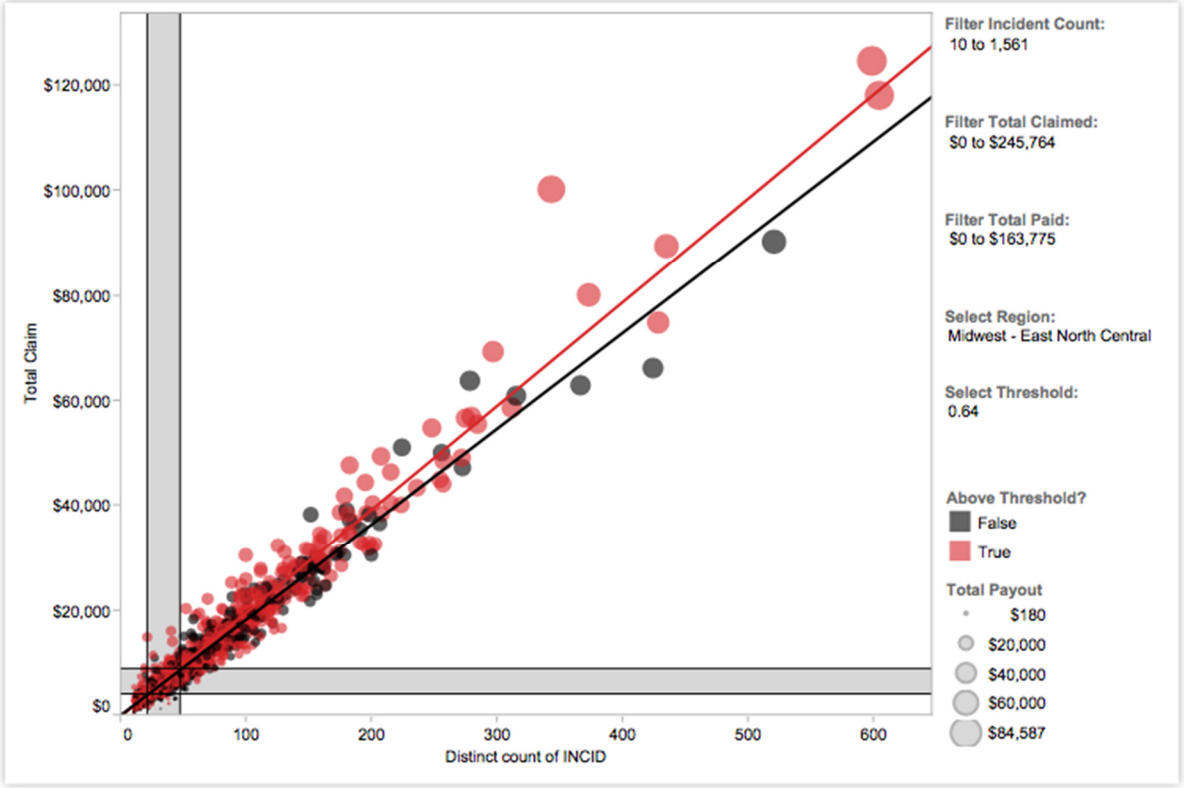
Pie chart

- Show relative proportions or percentages of information
- Best practices:
  - Limit pie wedges to six



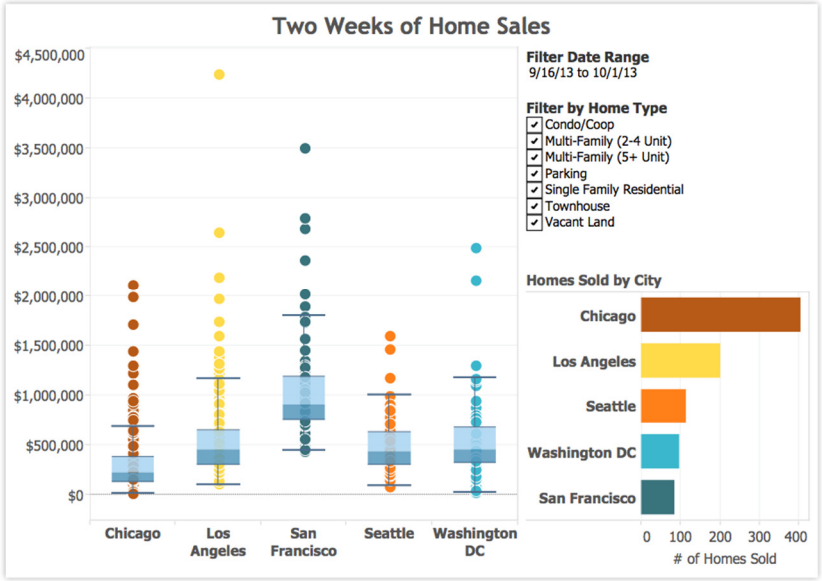
Scatter plot

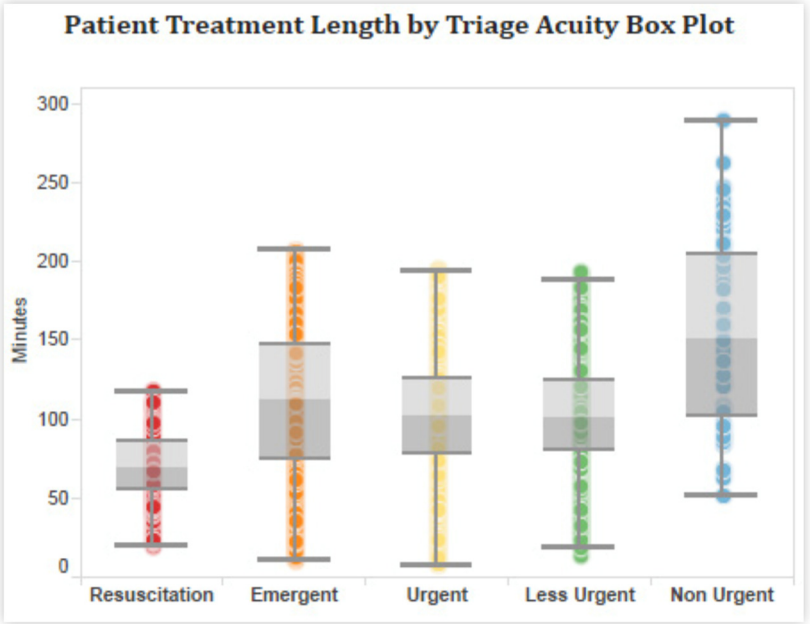
- Deeper insights about into relationships, trends, concentrations, outliers
- Examples: technology early adopters' and laggards' purchase patterns of smart phones, shipping costs of different product categories to different regions.



Box-and-whisker Plot

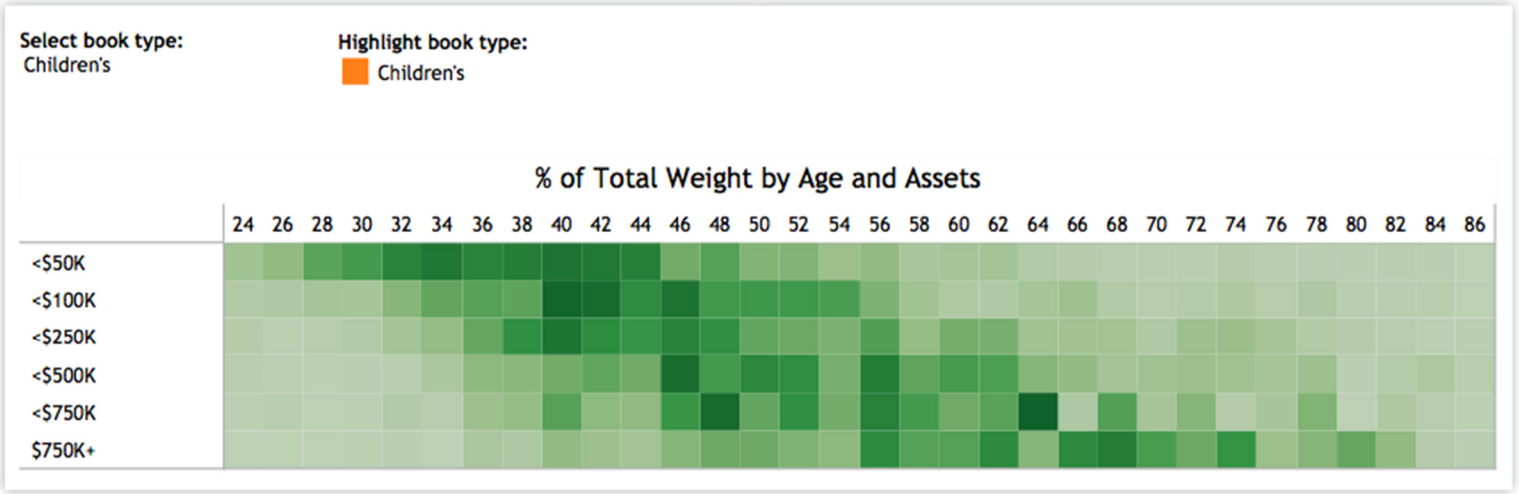
- Showing the distribution of a set of a data
  - Box: contains median, 2nd and 3rd quartiles (25% great and less than the median)
  - Whiskers: within 1.5 times of the difference between 2nd and 3rd quartiles, maximum and minimum points
- Best practices:
  - Hiding the points within the box
  - Comparing boxplots across categorical dimensions





Heat maps

- Compare data across two categories using color, e.g. where the intersection of the categories is strongest and weakest
  - Examples: segmentation analysis of target market, product adoption across regions, sales leads by individual rep
- Best practices:
  - Vary the size of squares
  - Using something other than squares



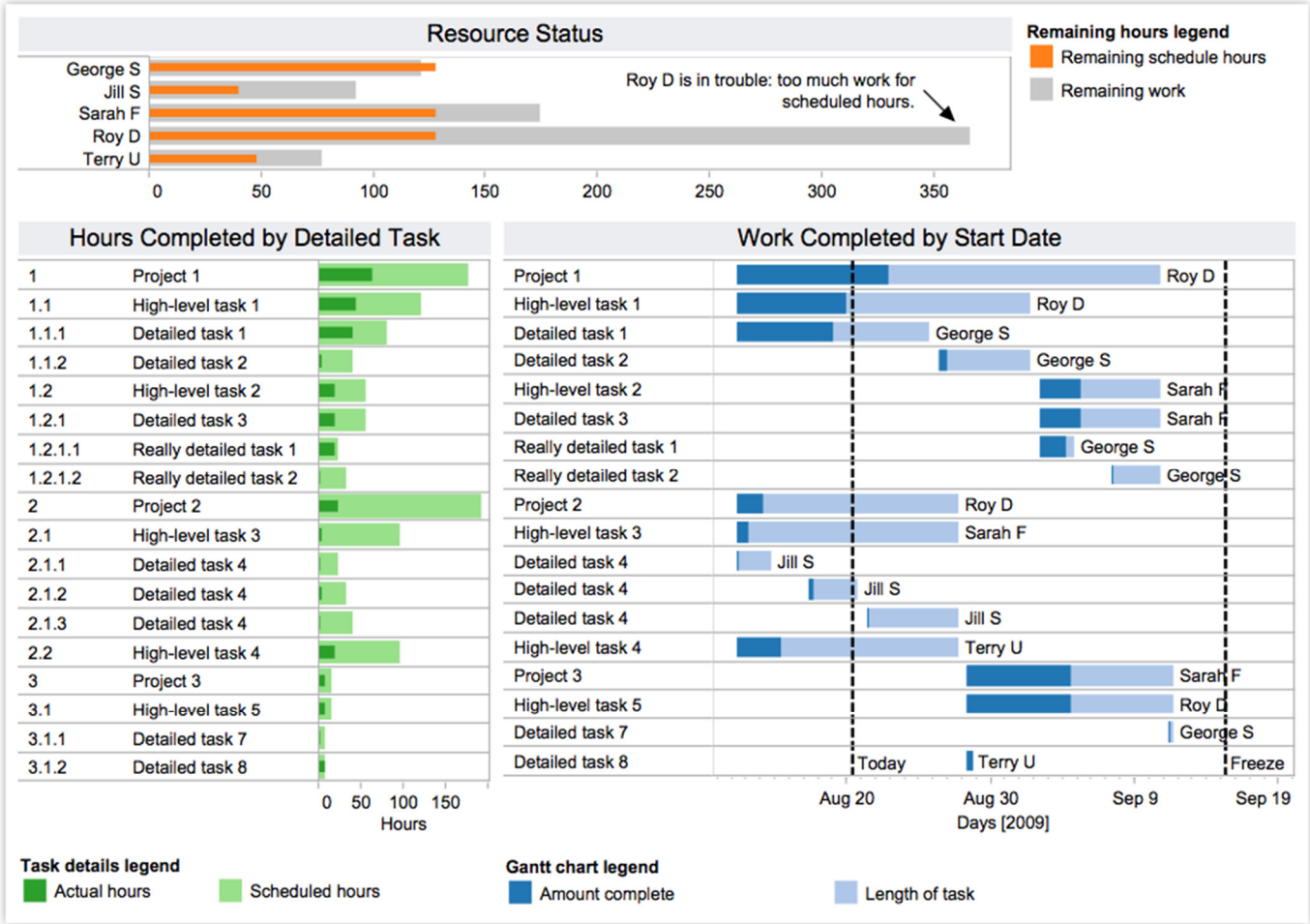
Advanced charts (Optional)

- Gantt chart: schedule, resources over time
- Bubble chart: advanced scatter plot
- Bullet chart: multi-level bar chart
- Treemap: hierarchical structure + highlight table

Gantt chart

- When to use:
  - Displaying a project schedule (project management)
  - Showing other things in use over time (resource planning)

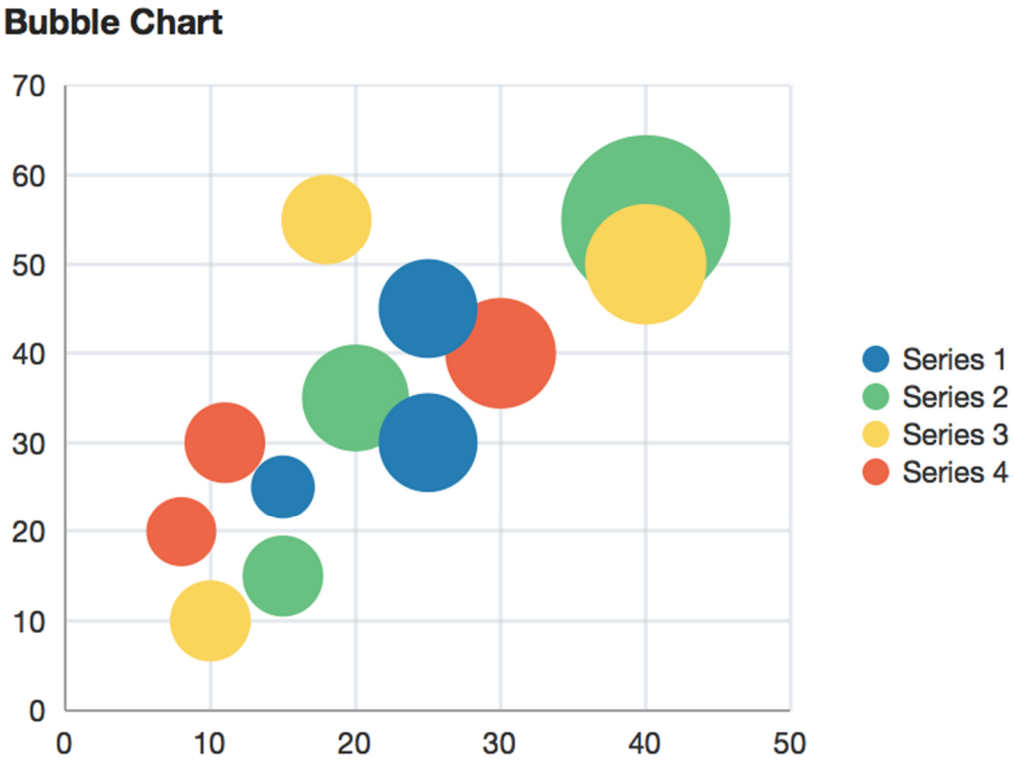
- Best practices:
  - Adding color
  - Combine maps and other chart types with Gantt charts



Bubble chart

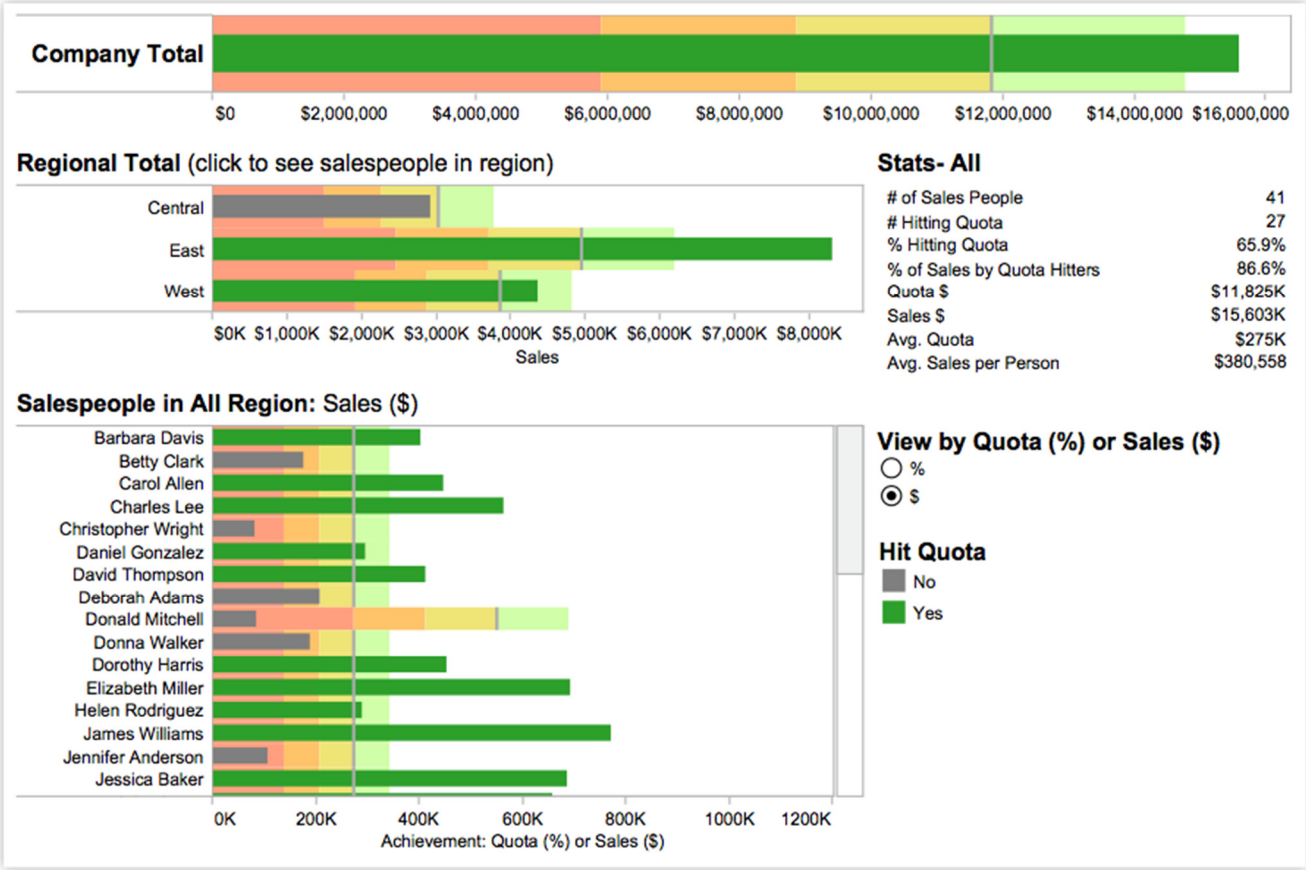
- A technique to accentuate data on scatter plots or maps
- When to use: showing the concentration of data along two axes

  - Examples: sales concentration by product and geography,
  - class attendance by department and time of day



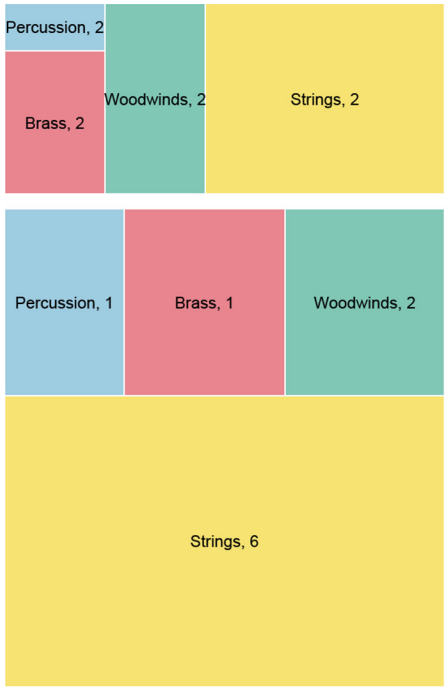
Bullet chart

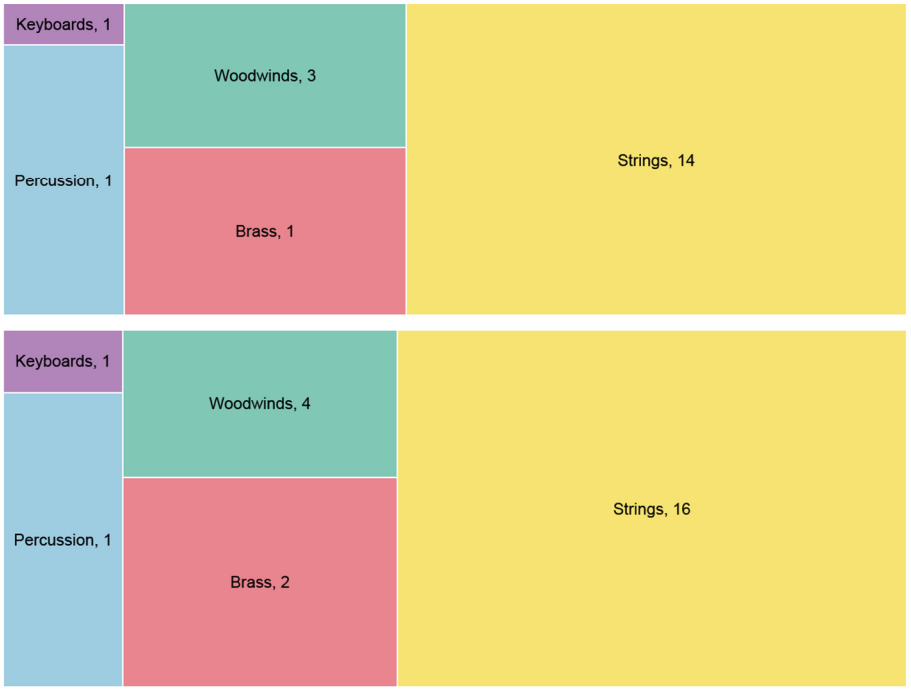
- Evaluating performance of a metric against a goal
  - Examples: sales quota assessment, actual spending vs. budget, performance spectrum (great/good/poor)
- Compare goals vs. achievements
- Best practices:
  - Use color to illustrate achievement thresholds
  - Add bullets to dashboards for summary insights

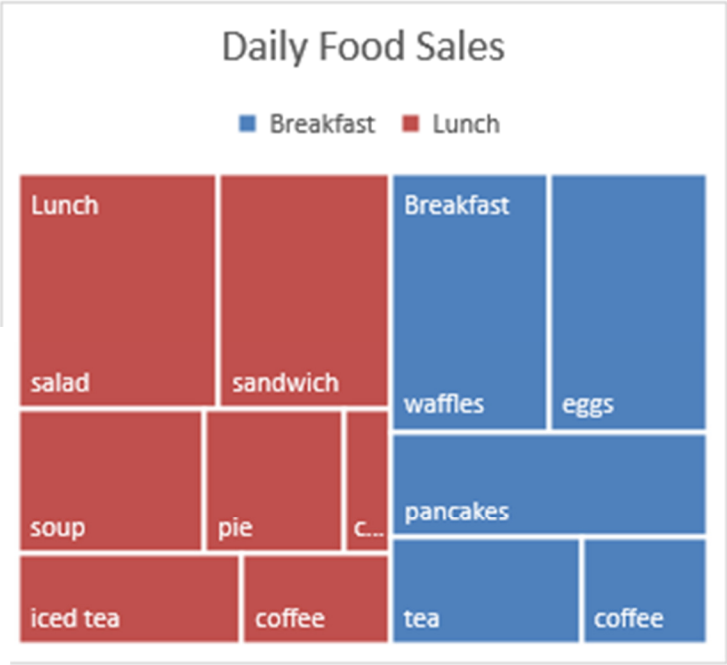


Treemap

- Showing hierarchical data as a proportion of a whole
- Best practices:
  - Coloring the rectangles by a category
  - Combining treemaps with bar charts







2. Types of Exploratory Data Analysis

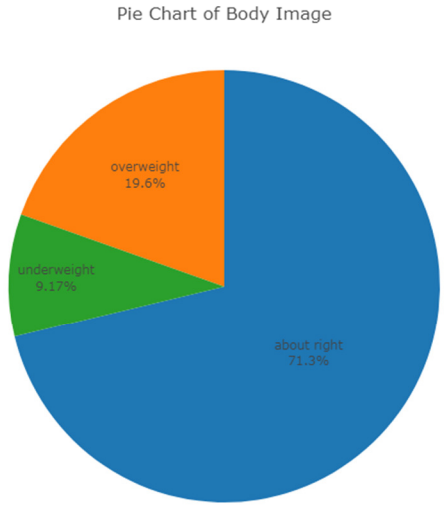
- Exploring Data Distribution
- Exploring Data Similarity
- Exploring Data Relationship
- Exploring Sequential Data
- Exploring Spatial Data
- Exploring Temporal Data

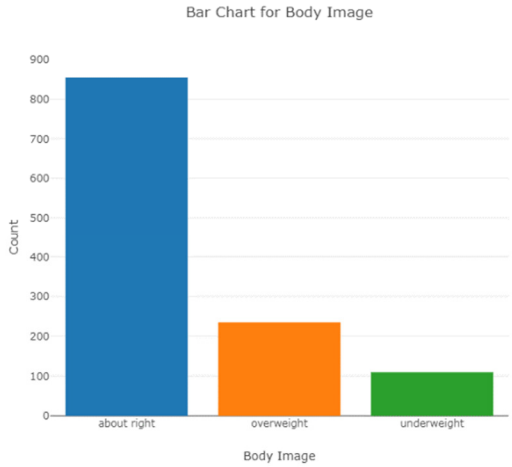
Exploring Data Distribution

- One Categorical Variable:
  - Calculate the percentage of each category
- There are two simple graphical displays for visualizing the distribution of categorical data:

Body Image Distribution

Category	Count	Percent
About right	855	$(\frac{855}{1200}) * 100 = 71.3\%$
Overweight	235	$(\frac{235}{1200}) * 100 = 19.6\%$
Underweight	110	$(\frac{110}{1200}) * 100 = 9.2\%$
Total	n=1200	100%





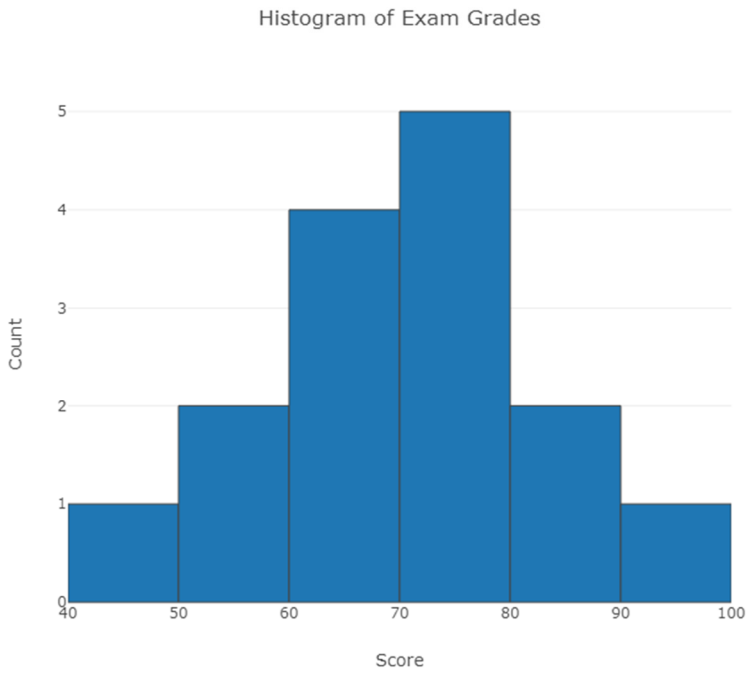
Data Distribution

- One Numerical Variable:

Exam Grades

Score	Count
[40-50)	1
[50-60)	2
[60-70)	4
[70-80)	5
[80-90)	2
[90-100]	1

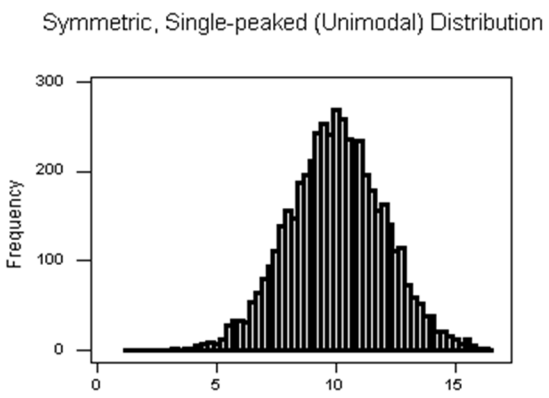
- How to present: histogram (bar chart with percentage)

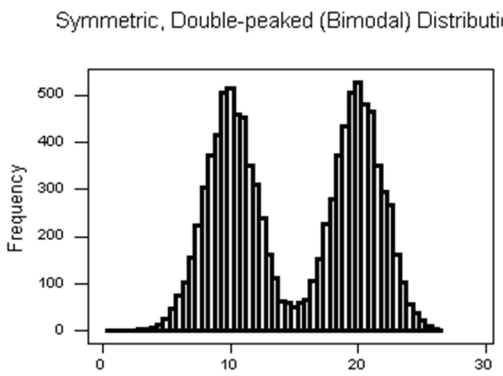


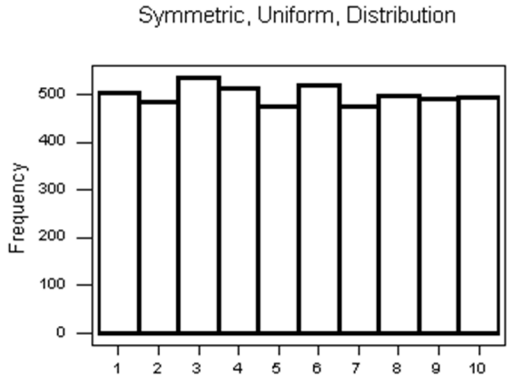
→ Grades has a Symmetric Distribution

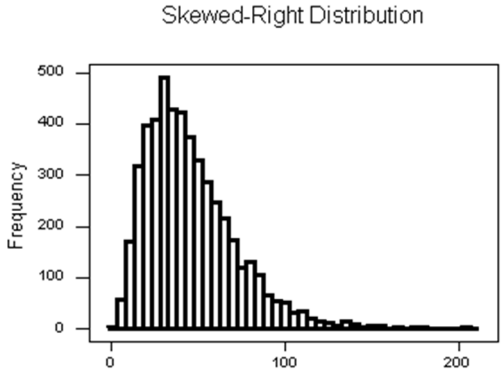
Data Distribution

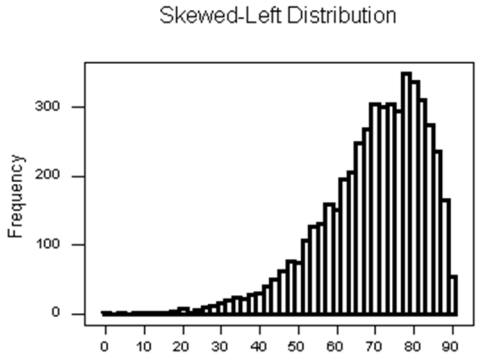
- Interpret the histogram
  - Shape
    - Symmetry/Skewness of the distribution





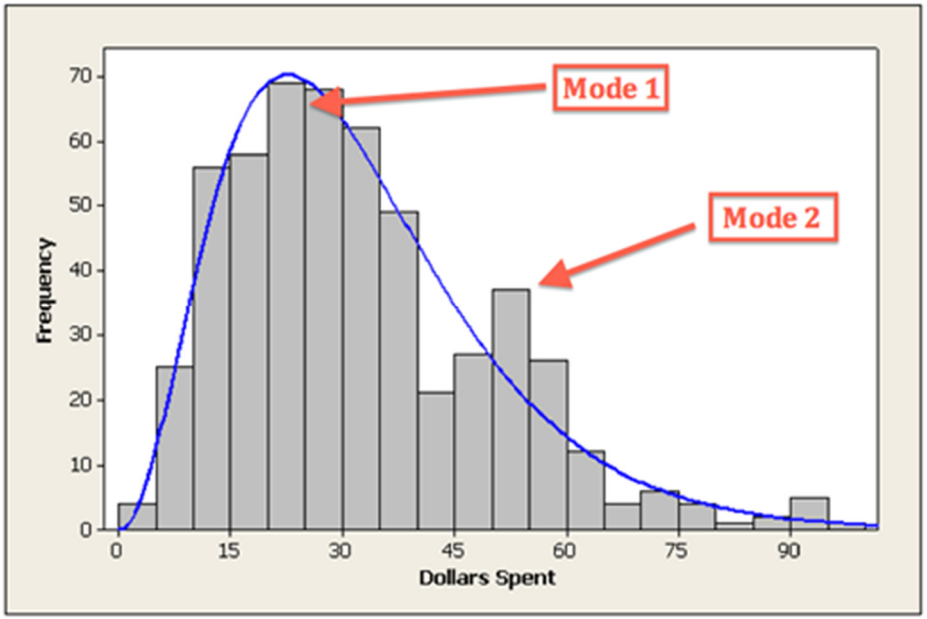






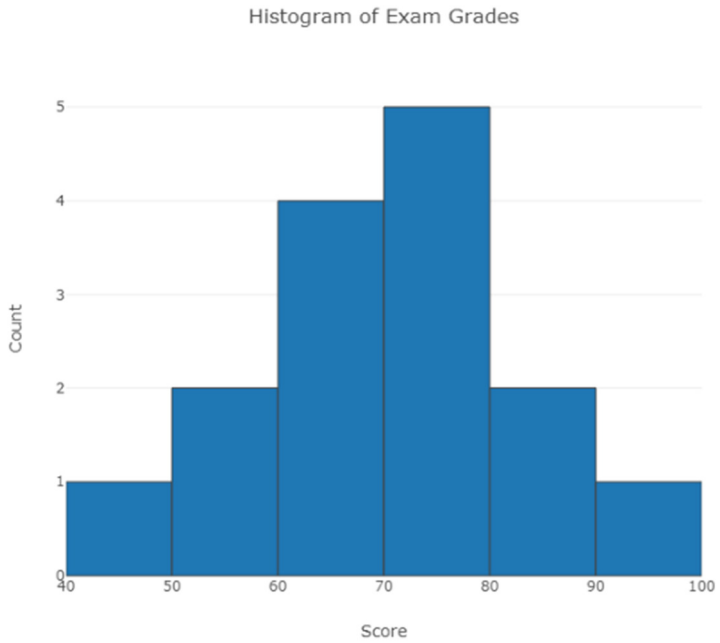
Data Distribution

- Interpret the histogram
  - Shape
    - Symmetry/Skewness of the distribution
    - Peakedness (modality): the number of peaks (modes) the distribution has
      - If a distribution has more than two modes, we say that the distribution is **multimodal**.



Data Distribution

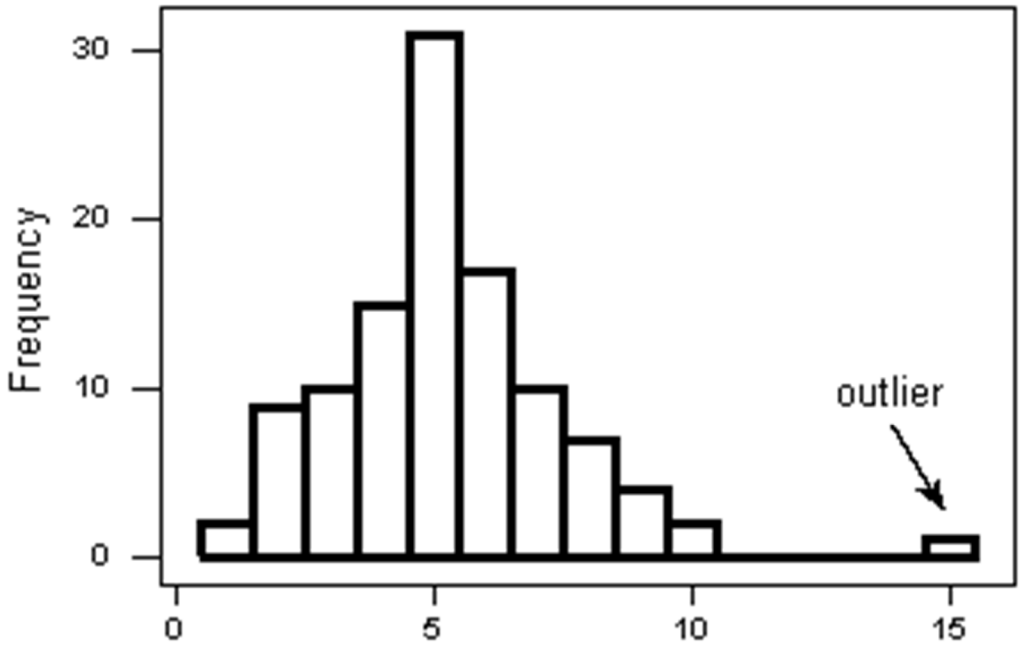
- Interpret the histogram
  - Center/Median: the value that divides the distribution so that approximately half the observations take smaller values, and approximately half the observations take larger values
  - Spread: variability
    - The **spread** (also called **variability**) of the distribution can be described by the approximate range covered by the data



mean is roughly 70
approximate min: 45 (the middle of the lowest interval of scores)
approximate max: 95 (the middle of the highest interval of scores)
approximate range: 95-45=50

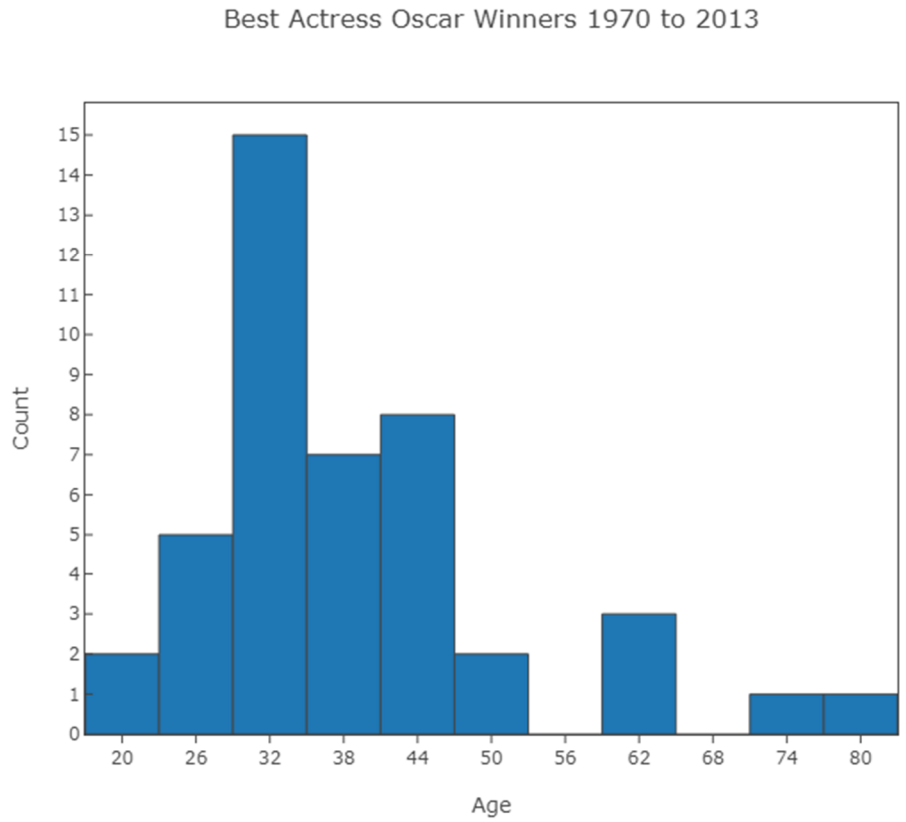
Data Distribution

- Interpret the histogram
  - Outliers: are observations that fall outside the overall pattern.



Data Distribution: Exercise

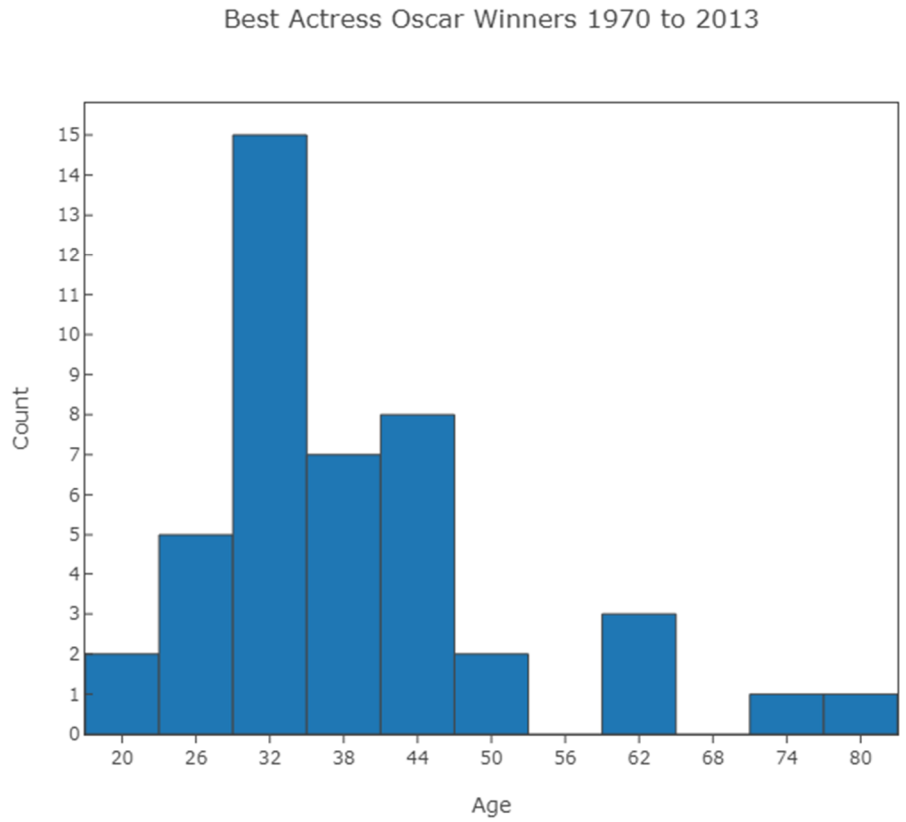
- Shape: ?
- Mean: ?
- Median : ?
- Mode: ?
- Spread: ?
- Outliers: ?



What can be concluded from data distribution?

Data Distribution: Exercise

- Shape: skew-right
- Mean: 50
- Median: 34
- Mode: 32
- Spread: 60
- Outliers: 74, 80

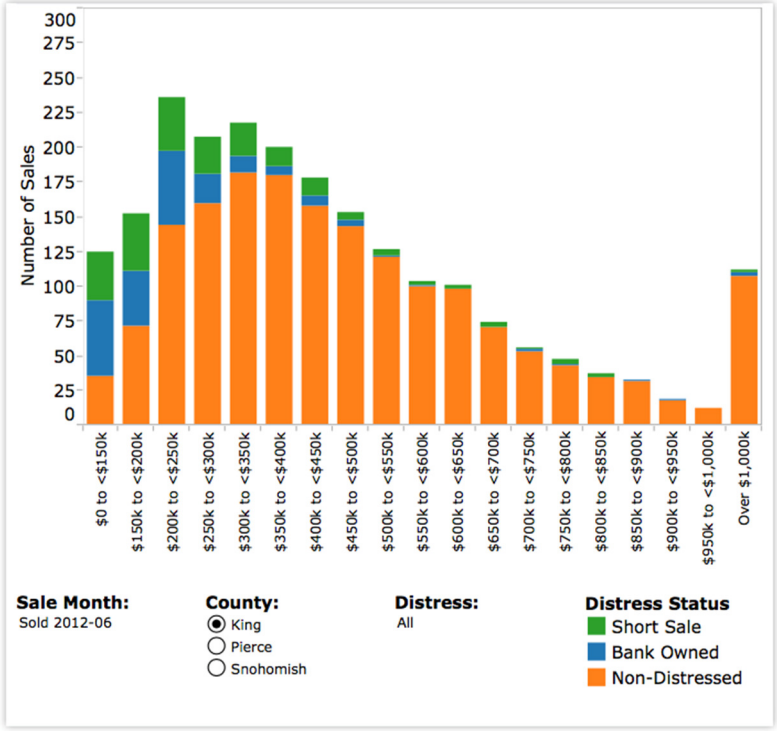


Sample conclusion: majority of Oscar winners is middle-aged

Data Distribution

- Multiple variables: stacked histogram
- Best practices:
  - Add a filter to flexibly choose which dimension to visualize
- Example: number of customers by company size, student performance on an exam, frequency of a product defect

  - Useful for more than one variable.
  - Problems: green and blue variables distributions are not clear.

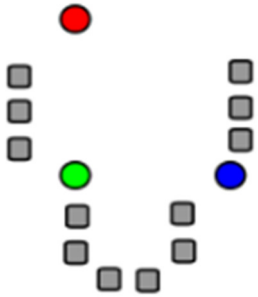


Exploring Data Similarity

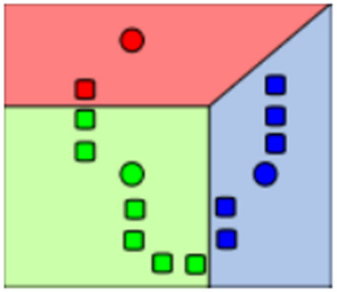
- **Clustering**: group together "similar" instances in the data sample
  - Distribute data into k different groups such that data points similar to each other are in the same group
  - Similarity between data entries is defined in terms of some distance metric (can be chosen)
- Common techniques:
  - K-means clustering
  - Density-base clustering → DBScan
  - Co-clustering

Exploring Data Similarity

- K-means clustering: recap

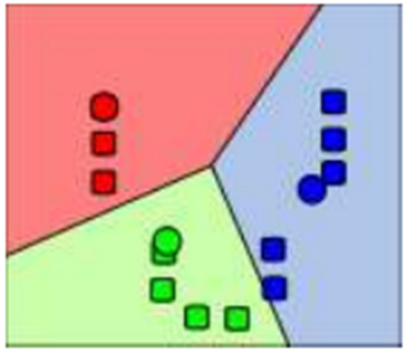


1) k initial "means" are randomly selected from the data set



2) k clusters are created by associating every observation with the nearest mean

3) The centroid of each of the k clusters becomes the new means



4) Steps 2 and 3 are repeated until convergence has been reached

Exploring Data Relationship

- Investigating the relationship between different variables
  - Note: association does not imply causation
- Exploring two variables:
  - the explanatory variable (independent variable) - the variable that claims to explain, predict or affect the response; and
  - the response variable (dependent variable) - the outcome of the study.
- Examples:
  - Can you predict a person's favorite type of music (classical, rock, jazz) based on his/her IQ level?
  - How is the number of calories in a hot dog related to (or affected by) the type of hot dog (beef, pork, or poultry)? In other words, are there differences in the number of calories among the three types of hot dogs?

Data Relationship: Two Variables

- C→Q: categorical to quantitative
- C→C: categorical to categorical
- Q→Q: quantitative to quantitative
- Q→C: not studied

		Response	
		Categorical	**Quantitative**
Explanatory	**Categorical**	C→C	C→Q
	Quantitative	Q→C	Q→Q

Data Relationship: Two Variables

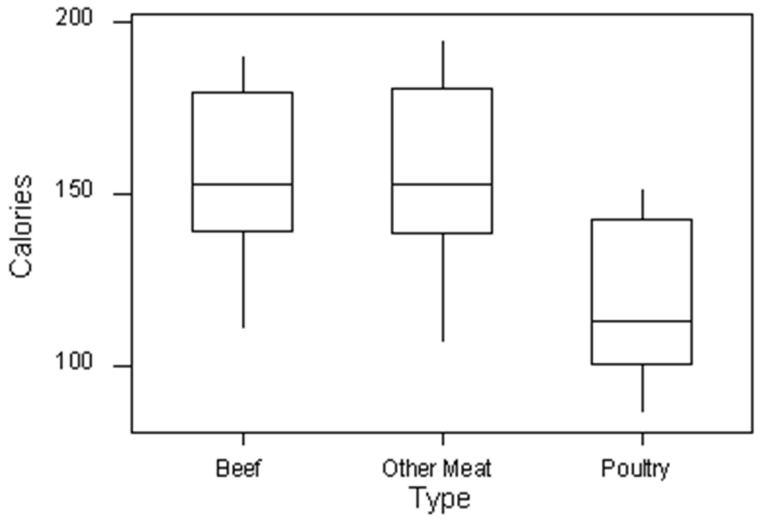
❖ **C→Q**: categorical to quantitative

- Examining using side-by-side boxplots, and the numerical summaries.
- We essentially **compare the distributions of the quantitative response for each category of the explanatory variable** using side-by-side boxplots, supplemented by descriptive statistics.

Explanatory ↖ Response ↗

	Type	Calories
Brand 1	Beef	186
Brand 2	Poultry	129
Brand 3	Beef	181
Brand 4	Meat	173
.	.	.
.	.	.
.	.	.
Brand 54	Poultry	144

Statistic	Beef	Meat	Poultry
min	111	107	86
Q1	139.5	138.5	100.5
Median	152.5	153	113
Q3	179.75	180.5	142.5
Max	190	195	152



Data Relationship: Two Variables

- **C→C**: categorical to categorical

- How to present:
  - In order to summarize the relationship between two categorical variables, we create a display called a **two-way table**.
  - The **Total row** gives the summary of the categorical variable body image.
  - The **Total column** gives the summary of the categorical variable gender
  - Compute conditional percents

Explanatory → Gender

Response → Body Image

Student	Gender	Body Image
.	.	.
.	.	.
student 25	M	overweight
student 26	M	about right
student 27	F	underweight
student 28	F	about right
student 29	M	about right
.	.	.
.	.	.

Gender	Body Image: About Right	Overweight	Underweight	Total
Female	560	163	37	760
Male	295	72	73	440
Total	855	235	110	1200

Gender	Body Image: About Right	Overweight	Underweight	Total
Female	560/760 = 73.7%	163/760 = 21.5%	37/760 = 4.9%	760/760 = 100%
Male	? %	? %	? %	? %

Data Relationship: Two Variables

- **Q→Q**: quantitative to quantitative
  - Formally, how close two variables have a linear/non-linear relationship between each other

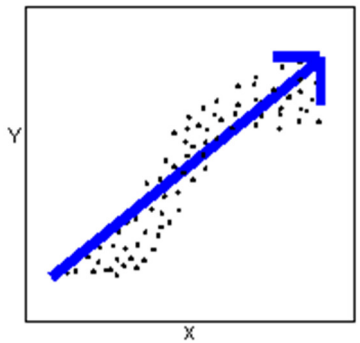
Explanatory ↖ Response ↗

	Age	Distance
Driver 1	**18**	**510**
Driver 2	32	410
Driver 3	55	420
Driver 4	23	510
.	.	.
.	.	.
.	.	.
Driver 30	82	360

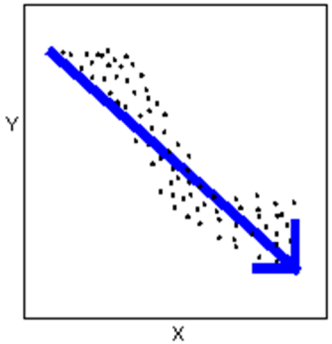
- How to present:
  - Scatter plot

Data Relationship: Two Variables

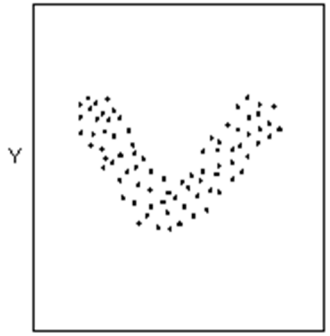
- **Q→Q**: quantitative to quantitative
- How to interpret the scatter plot
  - Direction



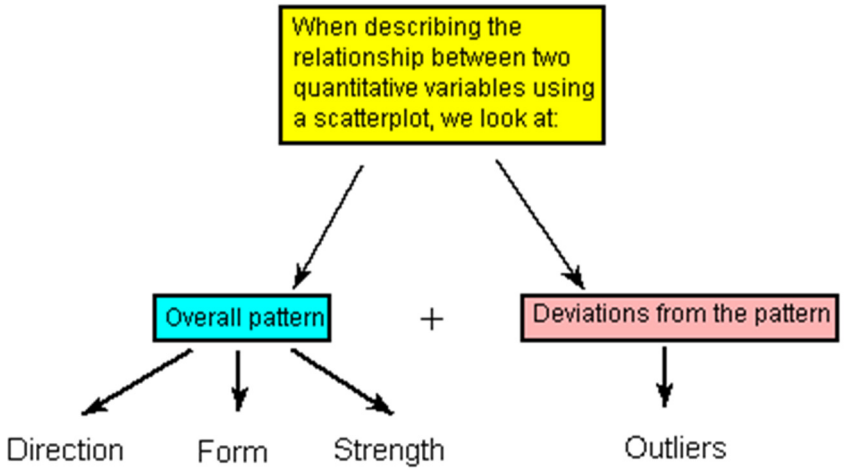
Positive relationship



Negative relationship

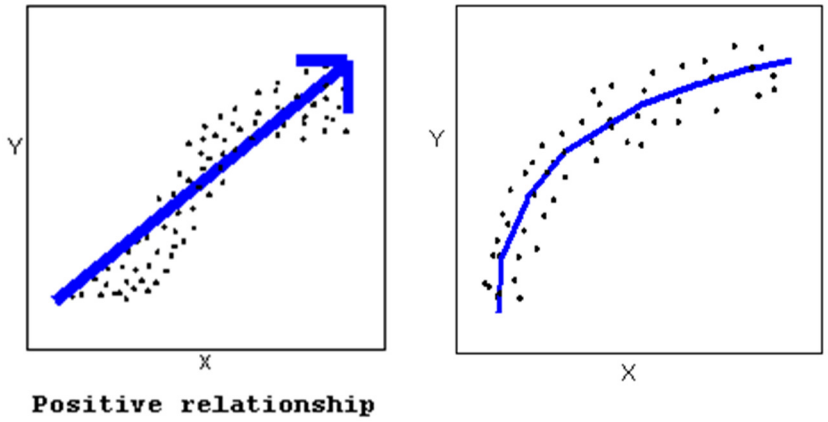


Neither positive nor negative



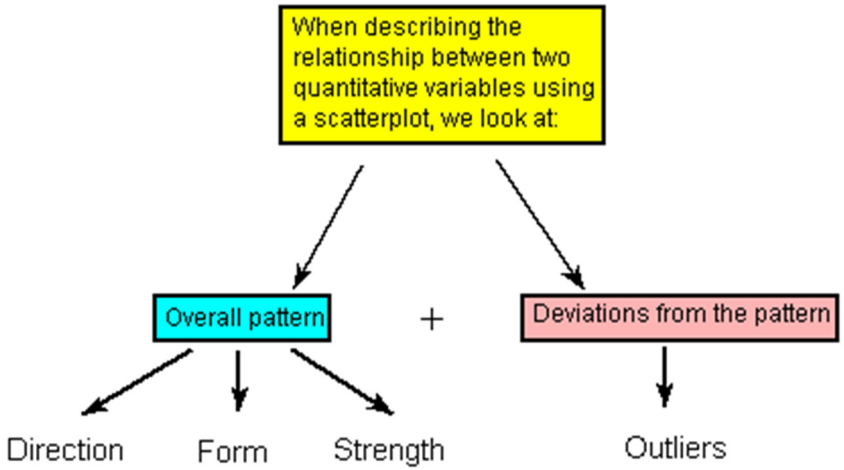
Data Relationship: Two Variables

- **Q→Q**: quantitative to quantitative
- How to interpret the scatter plot
  - Form



Linear

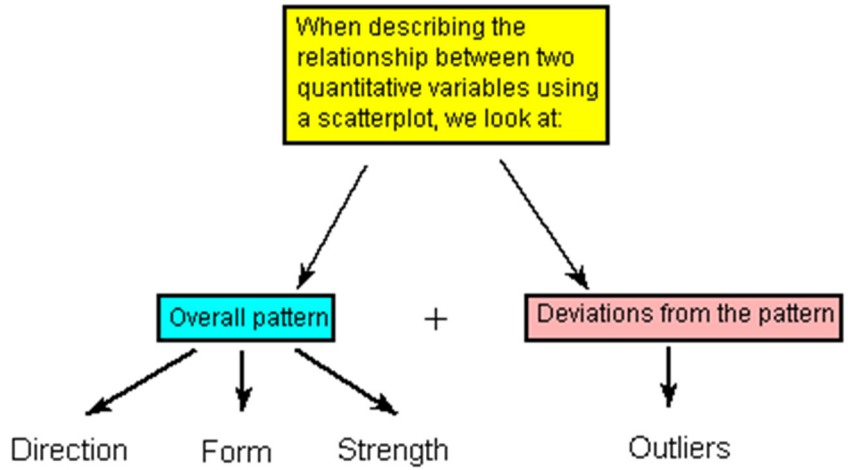
non-linear

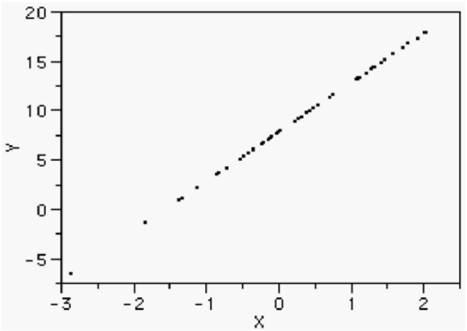


Data Relationship: Two Variables

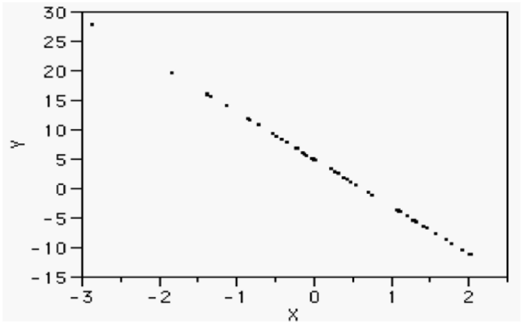
- **Q→Q**: quantitative to quantitative

- If the form is linear
  - Measure correlation coefficient
  - $r = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, -1 \leq r \leq 1$

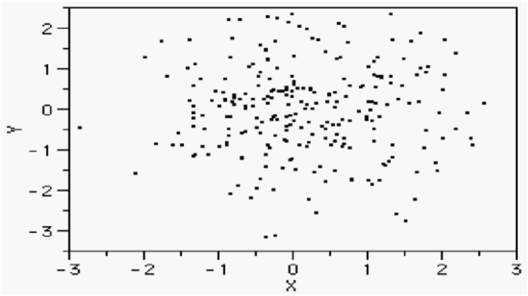




A perfect linear relationship r = 1



A perfect negative linear relationship r = -1

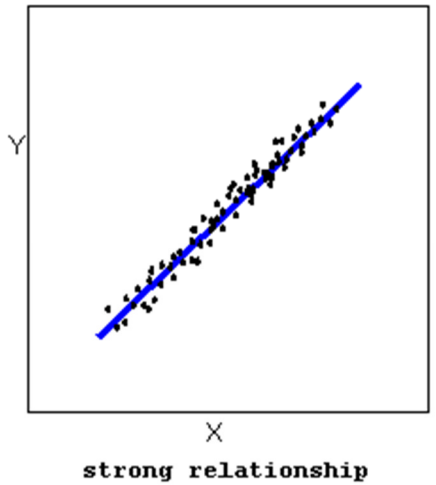


No relationship, r = 0

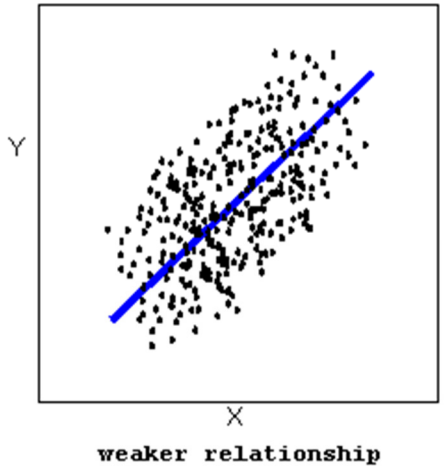
Properties of correlation coefficient:
https://lagunita.stanford.edu/courses/OLI/StatReasoning/Open/courseware/eda_er/_m5_linear/?child=first

Data Relationship: Two Variables

- Q→Q: quantitative to quantitative
- How to interpret the scatter plot
  - Strength

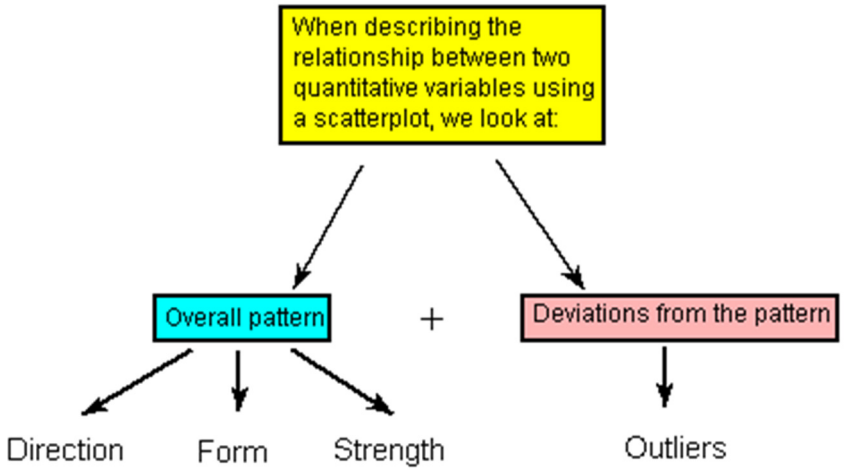


strong relationship



weaker relationship

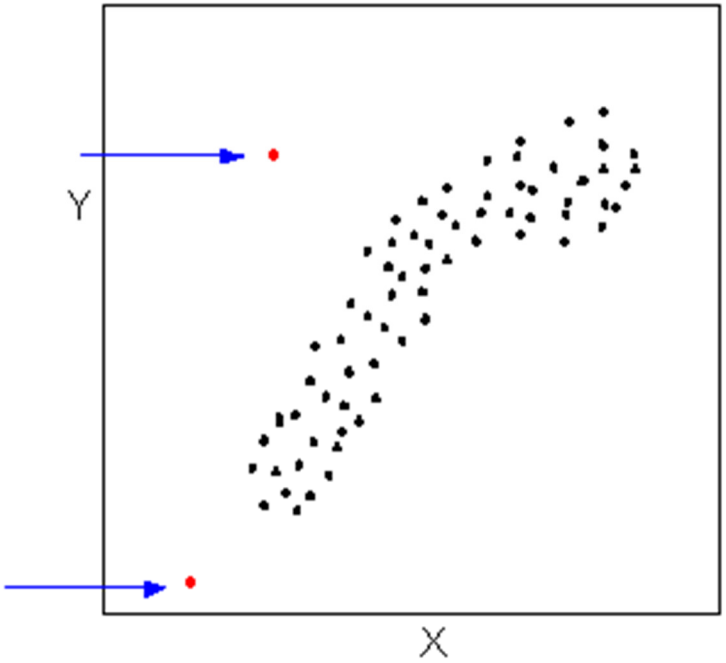
Disperse distribution

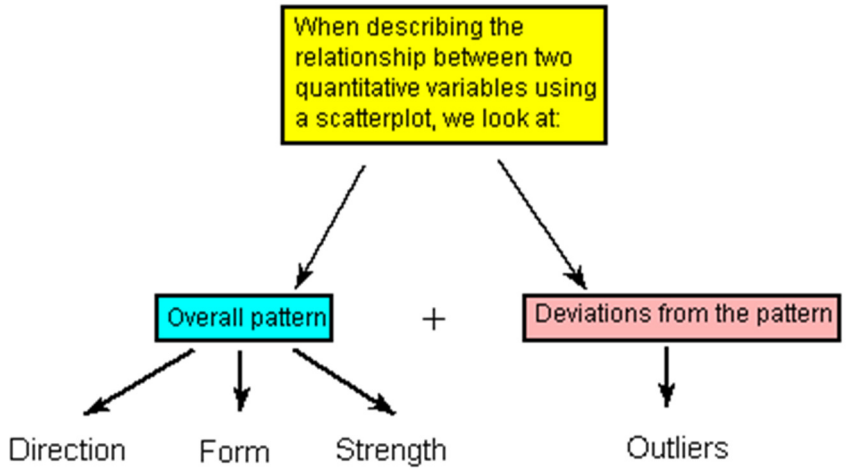


Data Relationship: Two Variables

- **Q→Q**: quantitative to quantitative

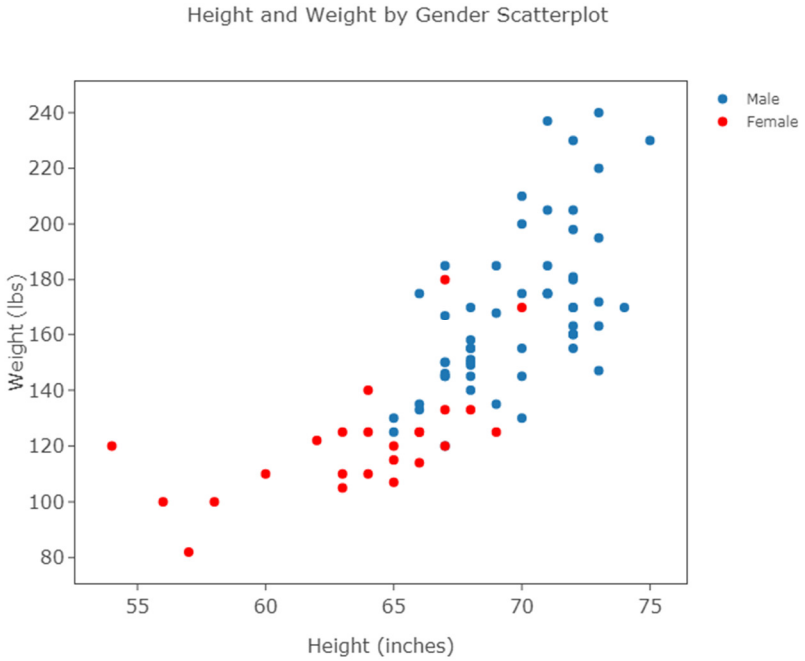
- How to interpret the scatter plot
  - Outliers





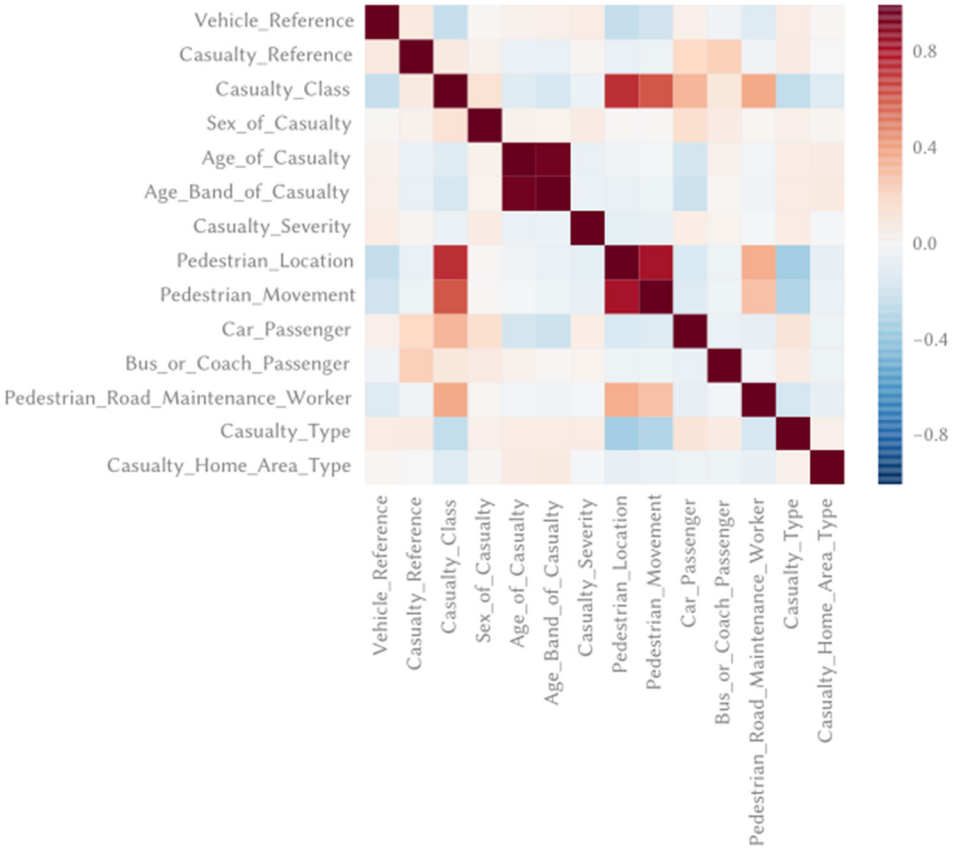
Data Relationship: Multiple Variables

- Investigate the dependence between multiple variables at the same time
- How to present:
  - **Labeled scatter plot:** it may be reasonable to indicate different subgroups or categories within the data on the scatterplot, by labeling each subgroup differently



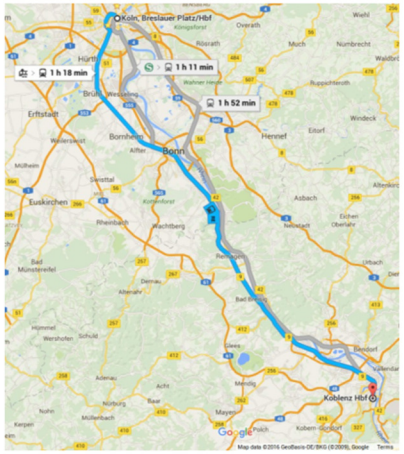
Data Relationship: Multiple Variables

- Investigate the dependence between multiple variables at the same time
- How to present:
  - **Correlation matrix :**
    - Output: a symmetric matrix where element m_{ij} is the correlation coefficient between variables i and j
    - Note: diagonal elements are always 1
    - Can be visualized graphically using a heatmap
    - Allows you to see which variables in your data are informative.

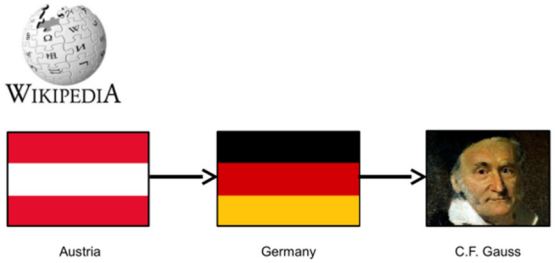


Exploring Sequential Data

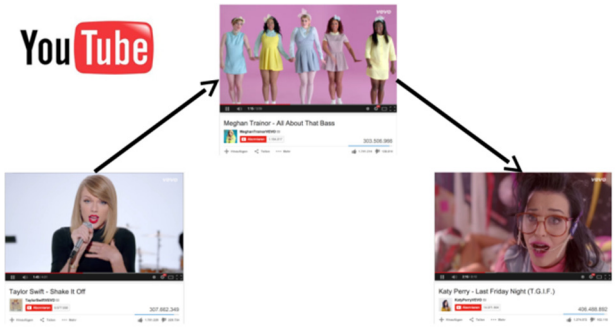
- Example:
  - Human mobility



  - Web navigation



  - Song listening sequences

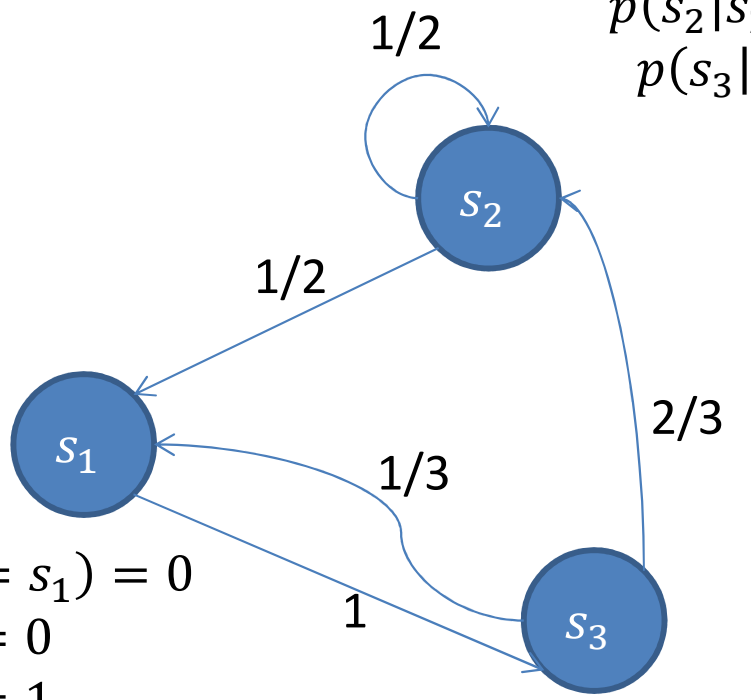


Exploring Sequential Data

- How to model:
  - Input: a list of sequences $\mathrm{D} = \{P_1, P_2, \dots, P_n\}$
  - Each sequence is indexed by time-steps: $P = q_0 \to q_1 \to \cdots \to q_t$
  - Has N states, called $s_1, s_2, \dots, s_n$
  - On the t-th time-step, the sequence is in exactly one of the available states. That is, a random variable $q_t \in \{s_1, s_2, \dots, s_n\}$
  - Between each time-step, the next state is chosen randomly.
  - The current state determines the probability for the next state

Exploring Sequential Data

- How to present:



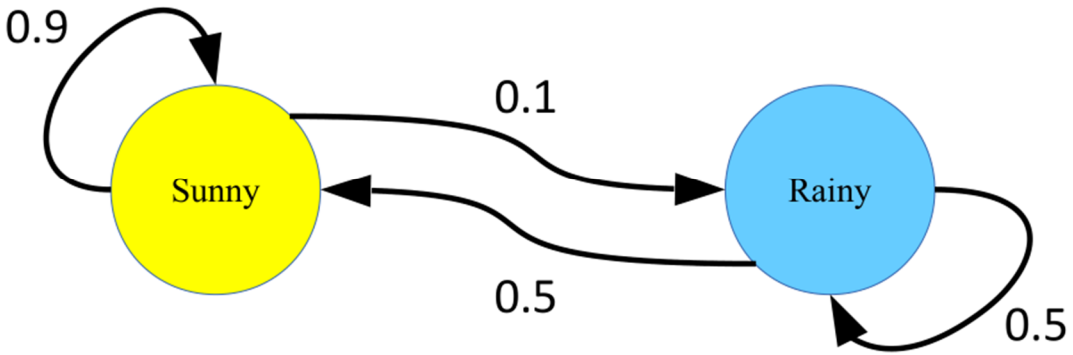
$p(s_1|s_2) = 1/2$
$p(s_2|s_2) = 1/2$
$p(s_3|s_2) = 0$

$p(q_{t+1} = s_1|q_t = s_1) = 0$
$p(s_2|s_1) = 0$
$p(s_3|s_1) = 1$

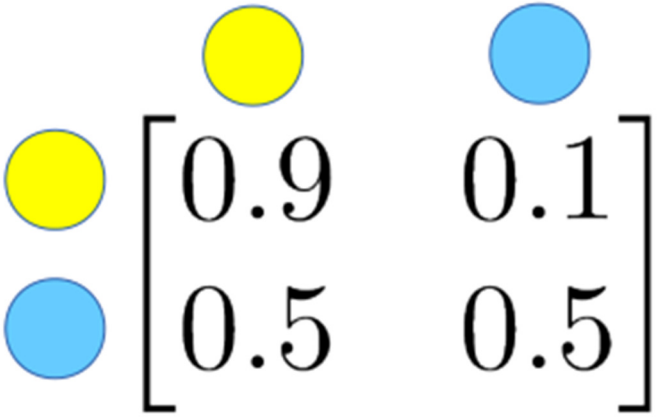
$p(s_1|s_3) = 1/3$
$p(s_2|s_3) = 2/3$
$p(s_3|s_3) = 0$

Exploring Sequential Data

- How to present: more example



Transition graph



$$\begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$

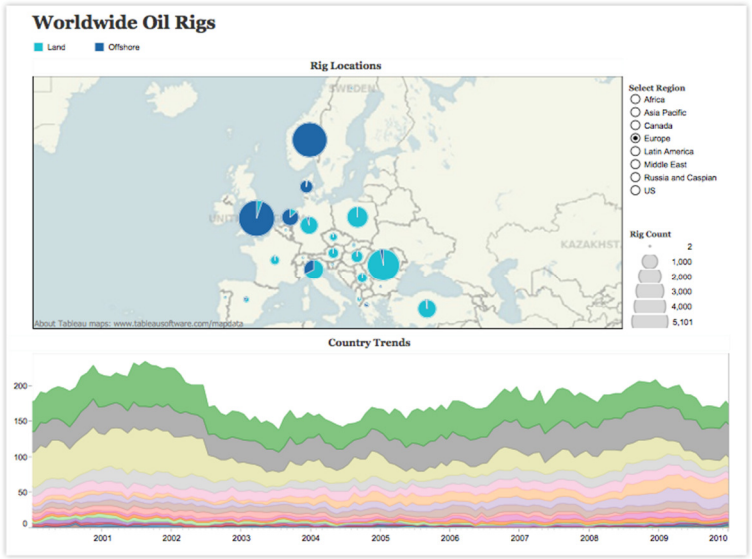
Transition matrix

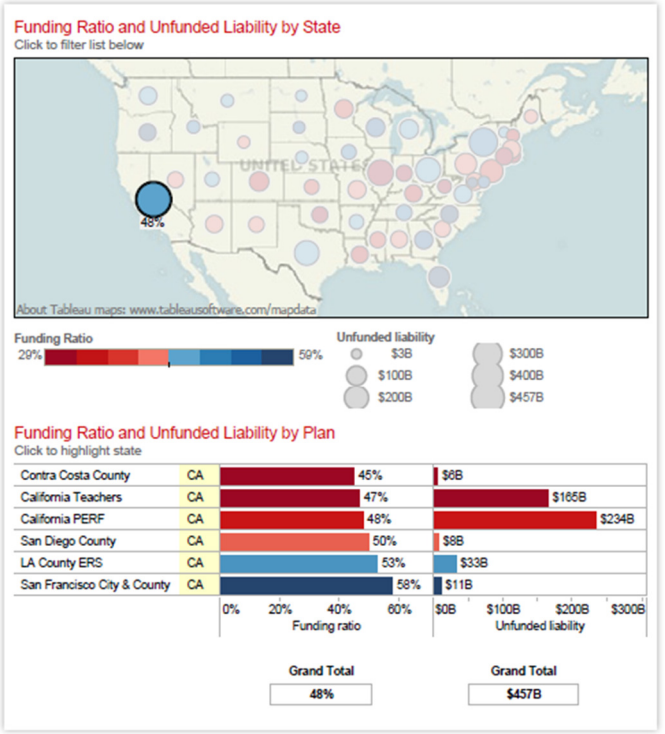
Exploring Sequential Data

- How to compute transition matrix:
  - Input: a list of sequences $\mathrm{D} = \{P_1, P_2, \dots, P_n\}$
  - For each state s_i:
    - $C_{s_i} = 0$
  - For each pair of states s_i, s_j:
    - $N_{s_i,s_j} = 0$
  - For each sequence $P \in D$:
    - For $x_t \in P$:
      - $C_{x_t} \mathrel{+}= 1$
      - $N_{x_t,x_{t+1}} \mathrel{+}= 1$
  - For each pair of states s_i, s_j:
    - $N_{s_i,s_j} = N_{s_i,s_j}/C_{s_i}$

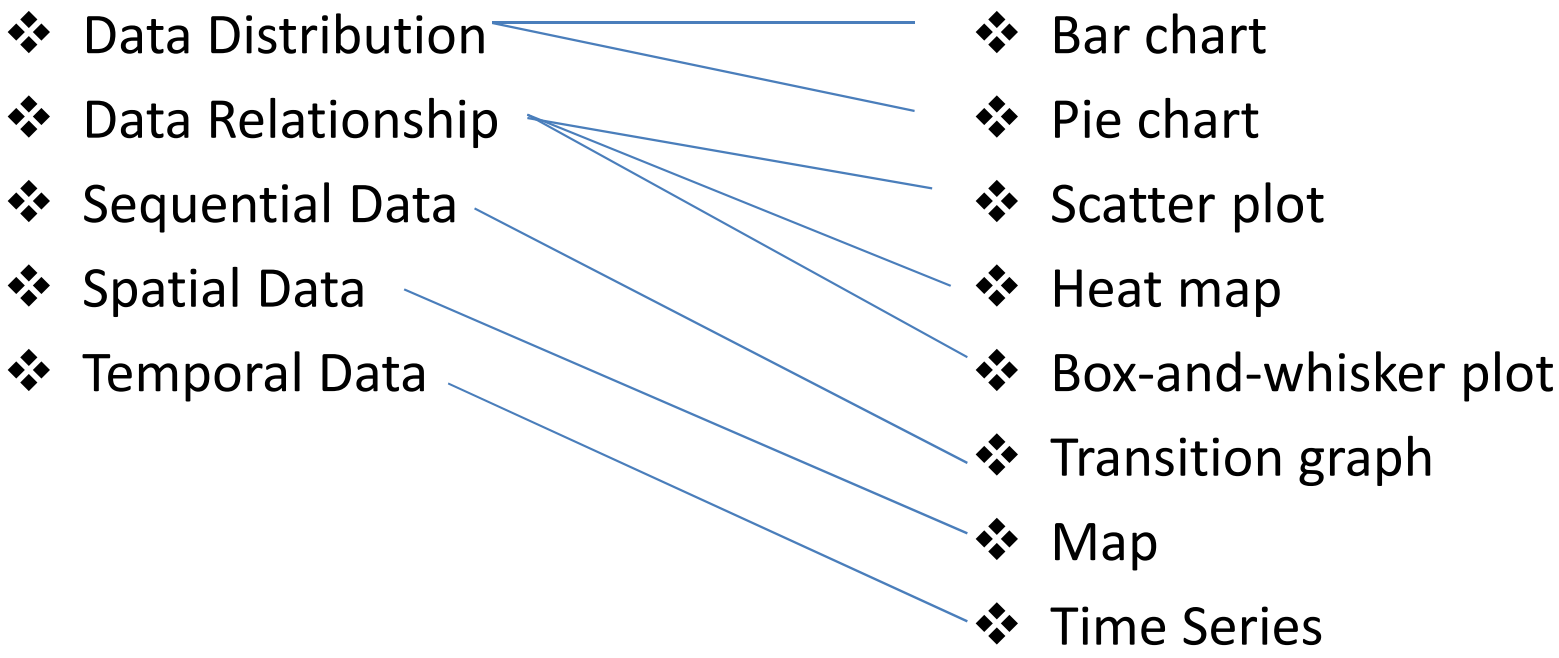
Exploring spatial data

- How to present: charts within map
- Best practices:
  - Include multiple bar charts on a dashboard.
  - Add color to bars for more impact
  - Use stacked bars or side-by-side bars.
  - Combine bar charts with maps.
  - Put bars on both sides of an axis





Summary of Exploratory Data Analyses and Charts



3. Dimensionality Reduction

- Projection of high-dimensional data onto smaller dimensions
- Why?: The curse of dimensionality
  - Hard to visualize
  - Hard to analyze since high-dimensional data points are far from each other
  - Computational expensive
  - → Dimensionality reduction: distill higher-dimensional data down to a smaller number of dimensions, while preserving as much of the variance in the data as possible.
    - Good for visualization/compression
    - Good for feature extraction
- Techniques:
  - Feature selection: $\{d_1, d_2, d_3, d_4, d_5\} \rightarrow \{\mathrm{d}_1, \mathrm{d}_3, \mathrm{d}_4\}$
  - Feature reduction: $\{d_1, d_2, d_3, d_4, d_5\} \rightarrow \{\mathrm{d'}_1, \mathrm{d'}_2, \mathrm{d'}_3\}$

Techniques of dimensionality reduction

- **Feature Selection**: chooses an optimal subset of dimensions to represent data
  - Only a subset of the original dimensions are selected
  - Good for discrete data
- **Feature Reduction**: refers to the mapping of the original high-dimensional data onto a lower-dimensional space
  - All original dimensions are used
  - The transformed dimensions are linear combinations of the original dimensions
  - Good for continuous data

Feature Selection – Subset Search

- Treat as a subset search problem

d_1	d_2	d_3	d_4	d_5	C
0	0	1	0	1	0
0	1	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1

Example: data has 5 dimensions

- Class of data is C
- What is the optimal subset of dimensions sufficient enough for representing data?

Answer: $\{d_1, d_2\}$ or $\{d_1, d_3\}$

- Because $C = d_1 \vee d_2$
- Or $C = d_1 \vee \neg d_3$

Feature Selection

- Reducing the number of N features to an optimal subset
- There are $\binom{N}{2}$ possible subsets!
- Efficient Approach:
  - Filtering (Ranking): consider features as independent

Feature Selection: Filtering

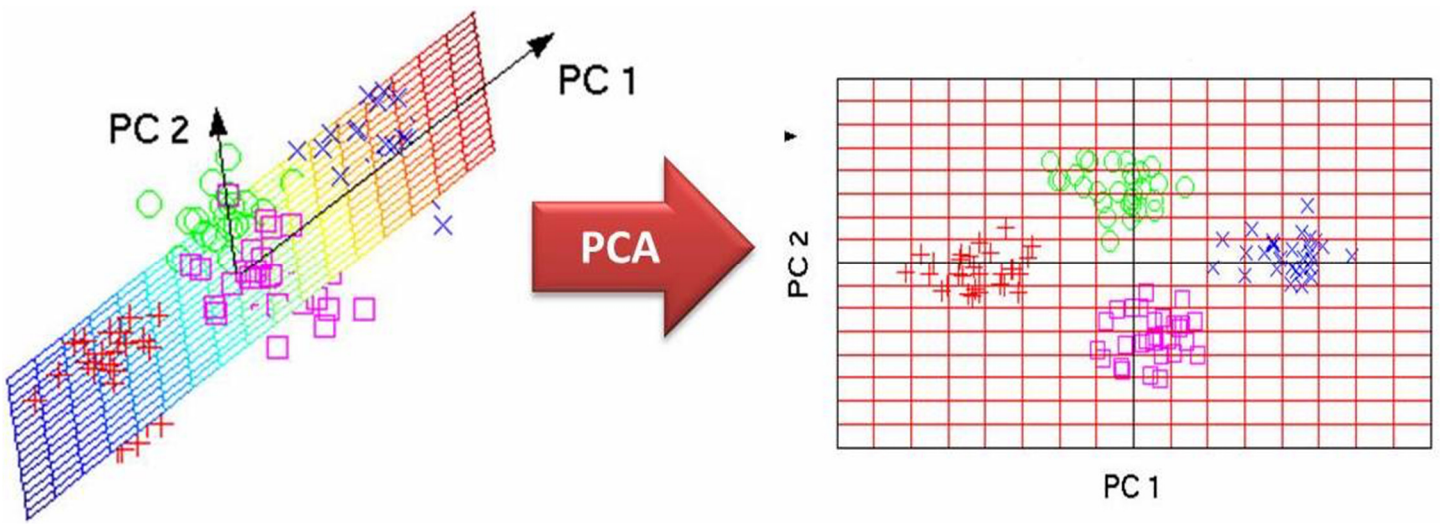
- Treat as a ranking problem:
  - Rank features according to their predictive power
  - Select top-ranked ones
- Advantages:
  - Efficient: linear with the number of dimensions
  - Easy to implement
- Disadvantages:
  - Unable to consider correlation between dimensions
- Ranking measures:
  - Information measure: e.g. mutual information

Feature Reduction

- Refers to the mapping of the original high-dimensional data onto a lower-dimensional space
- Common techniques:
  - Principal Component Analysis (PCA)

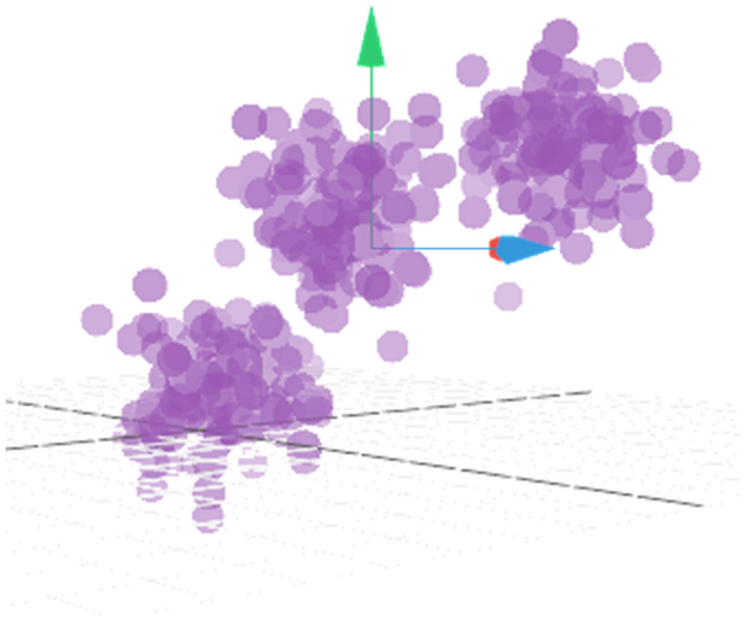
Principal Component Analysis (PCA)

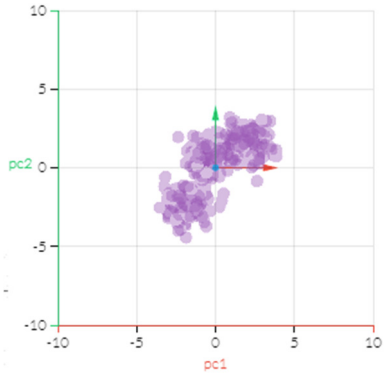
- What:
  - Allows visualization of high-dimensional continuous data in 2-3D
  - The principal components are the strongest (highest variation) dimensions in the dataset, and are orthogonal
  - Really useful for things like image compression and facial recognition



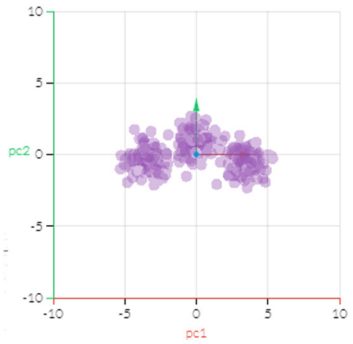
PCA: Another Example

- http://setosa.io/ev/principal-component-analysis/





Bad feature reduction

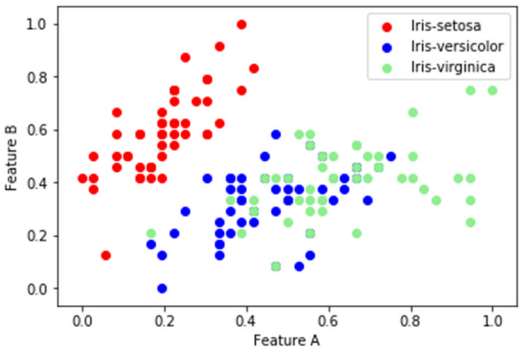


Good feature reduction

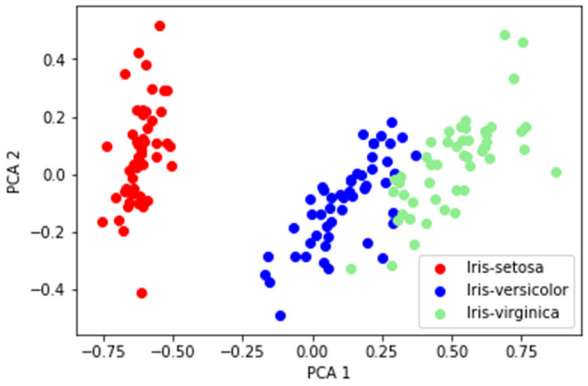
PCA using Scikit-learn

Visualize 4-D Iris Flower Data in 2-D

- Naïve approach: using 2 random dimensions
- PCA: use 2 dimensions, while still preserving variance.



Cannot see a clear separation between different flower classes



A better separation between different flower classes

```
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.decomposition import PCA as sklearnPCA
```

```
pca = sklearnPCA(n_components=2) #2-dimensional PCA
transformed = pd.DataFrame(pca.fit_transform(X_norm))
```

PCA: Underlying theory (Optional)

- Finds “eigenvectors” in the higher dimensional data
  - These define hyperplanes that split the data while preserving the most variance in it
- The data gets projected onto these hyperplanes, which represent the lower dimensions you want to represent
- A popular implementation of this is called Singular Value Decomposition (SVD)

PCA: Underlying Theory (Optional)

- Steps to do:
  - Form the covariance matrix S of d original dimensions $\{a_i\}_{i=1}^{d}$
  - Compute the eigenvectors:

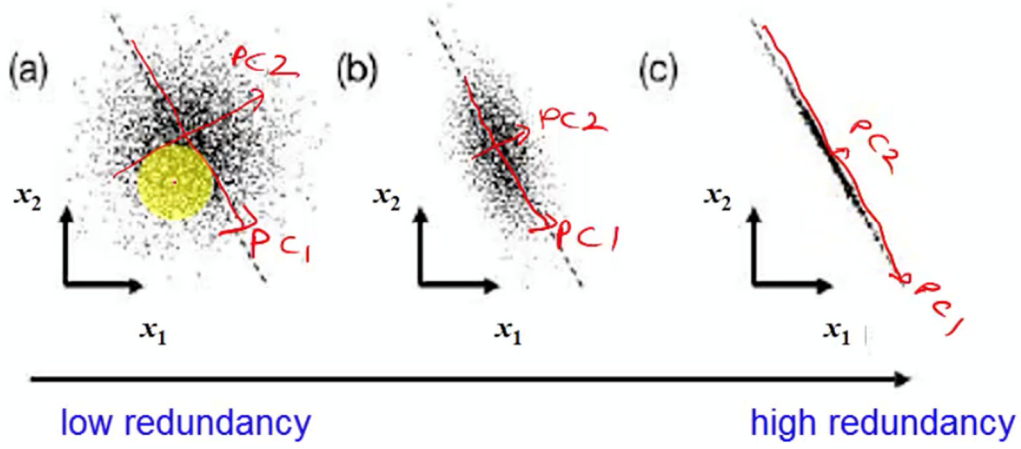
$$G \leftarrow [a_1, a_2, ..., a_p]$$

  - The first K eigenvectors form the K PCs (ranked by maximum variability)
  - Visualize data in K dimensions using the transformation:

$$\text{A testpoint } x \in \Re^d \rightarrow G^T x \in \Re^p.$$

PCA: Pros and Cons (Optional)

- **Pros:** good with
  - High-redundant data
  - Large sample size
  - Normal distribution
  - Linear correlation
- **Cons:** bad for
  - Low-redundant data
  - Small sample size
  - Abnormal distribution
  - Non-linear correlation



4. Exploratory data analysis with Matplotlib

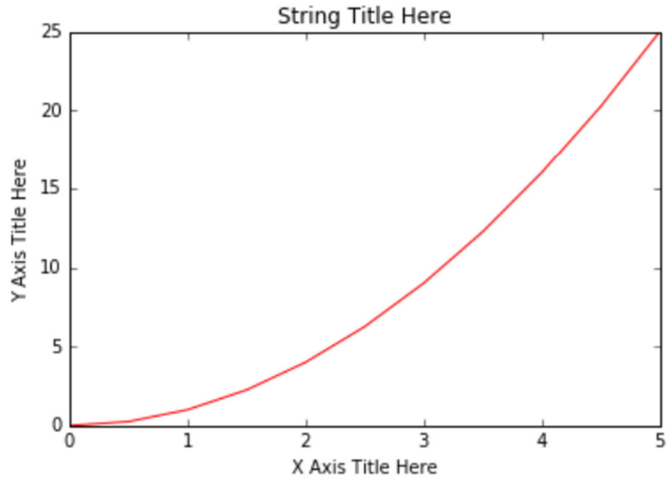
- Office website: http://matplotlib.org/
- Matplotlib allows you to explore data and create reproducible visual results (2D and 3D figures) programmatically
- Features:
  - Generally easy to get started for simple plots
  - Support for custom labels and texts
  - Great control of every element in a figure
  - High-quality output in many formats
  - Very customizable in general

Matplotlib: Exploratory Data Analysis

- Line chart:

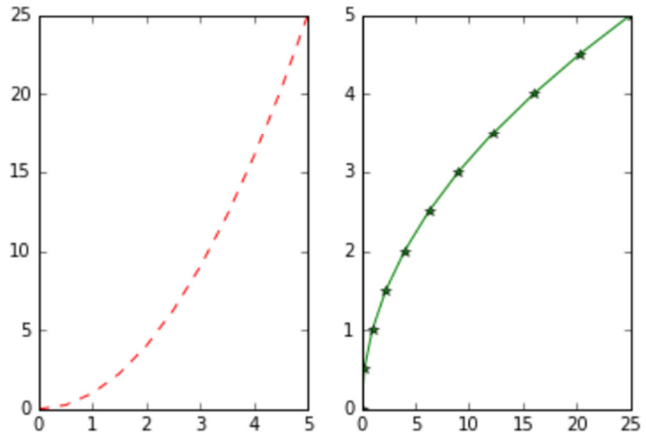
```
fig.savefig("filename.png", dpi=300, bbox_inches='tight')
```

```
plt.plot(x, y, 'r') # 'r' is the color red
plt.xlabel('X Axis Title Here')
plt.ylabel('Y Axis Title Here')
plt.title('String Title Here')
plt.show()
```



Single plot

```
# plt.subplot(nrows, ncols, plot_number)
plt.subplot(1,2,1)
plt.plot(x, y, 'r--') # More on color options later
plt.subplot(1,2,2)
plt.plot(y, x, 'g*-');
```



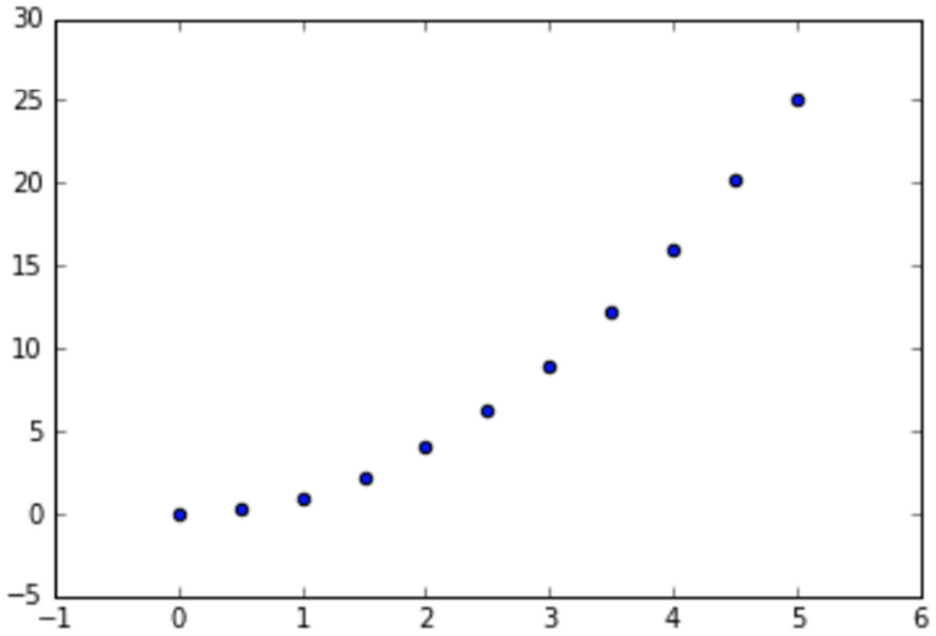
Multiple plots

Matplotlib: Exploratory Data Analysis

- Scatter plot:

```
plt.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x1122be438>
```

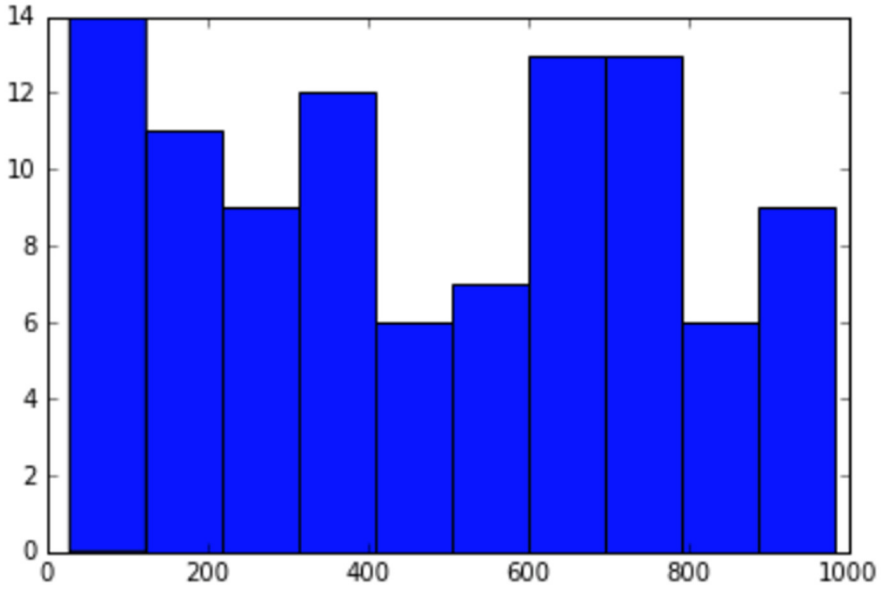


Matplotlib: Exploratory Data Analysis

- Histogram:

```
from random import sample
data = sample(range(1, 1000), 100)
plt.hist(data)
```

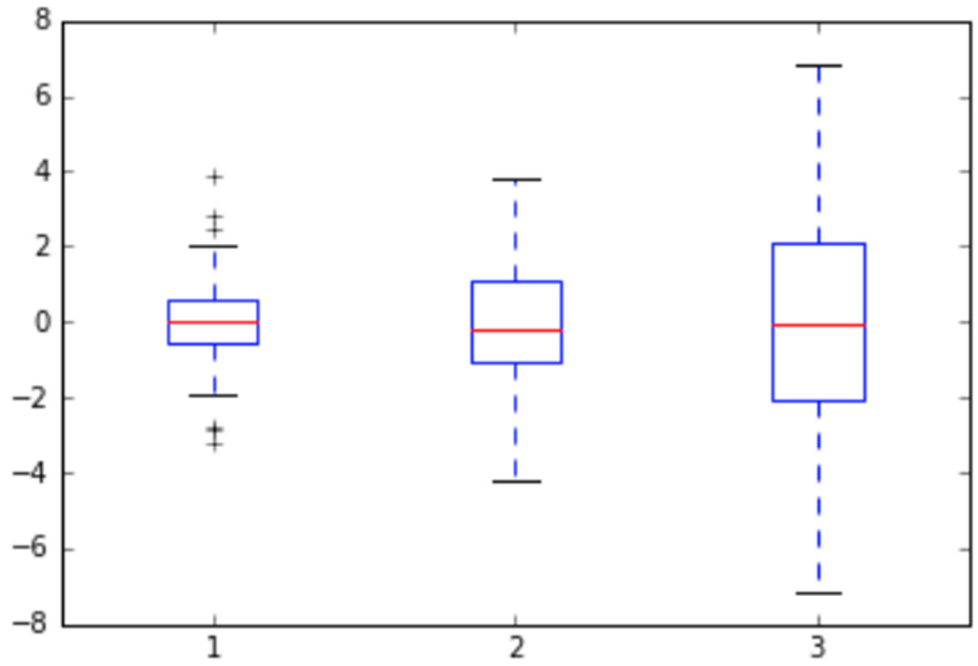
```
(array([ 14.,  11.,   9.,  12.,   6.,   7.,  13.,  13.,   6.,   9.]),
 array([  28. ,  123.5,  219. ,  314.5,  410. ,  505.5,  601. ,  696.5,
         792. ,  887.5,  983. ]),
 <a list of 10 Patch objects>)
```



Matplotlib: Exploratory Data Analysis

- Box-and-whisker plot

```
data = [np.random.normal(0, std, 100) for std in range(1, 4)]

# rectangular box plot
plt.boxplot(data,vert=True,patch_artist=True);
```

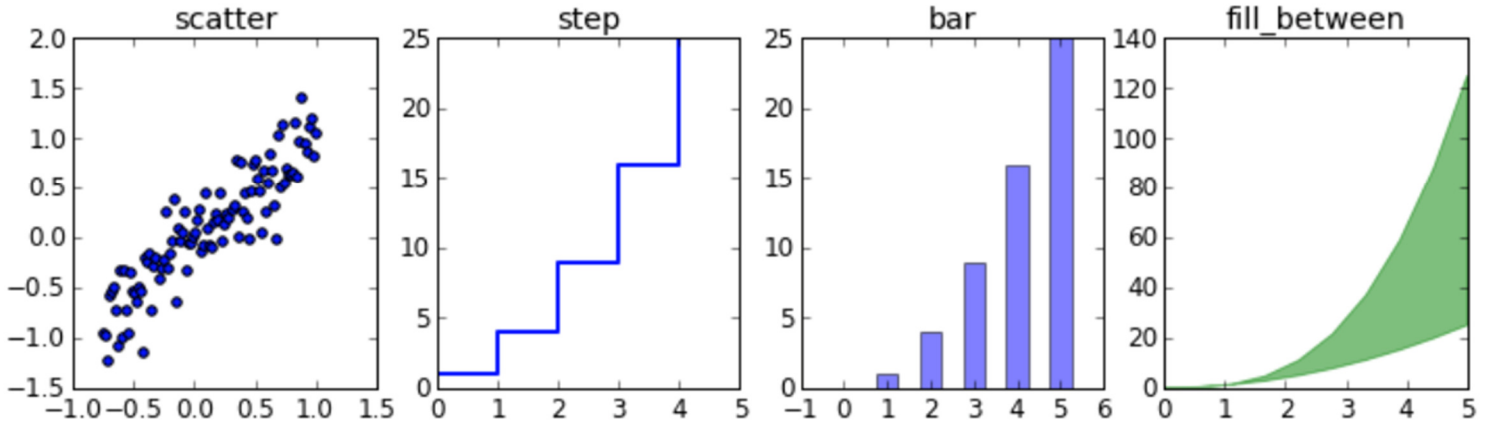


Matplotlib: Exploratory Data Analysis

- Other 2D plots:

```
n = np.array([0,1,2,3,4,5])
```

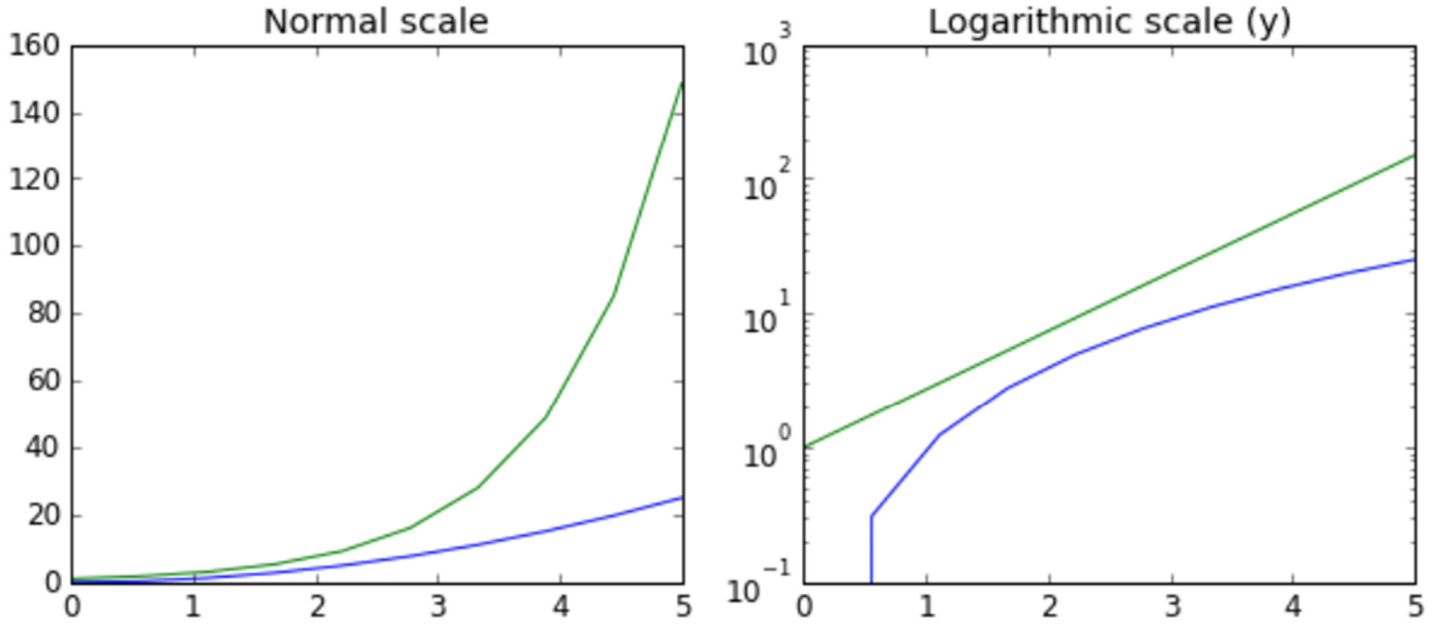
```
fig, axes = plt.subplots(1, 4, figsize=(12,3))

axes[0].scatter(xx, xx + 0.25*np.random.randn(len(xx)))
axes[0].set_title("scatter")

axes[1].step(n, n**2, lw=2)
axes[1].set_title("step")

axes[2].bar(n, n**2, align="center", width=0.5, alpha=0.5)
axes[2].set_title("bar")

axes[3].fill_between(x, x**2, x**3, color="green", alpha=0.5);
axes[3].set_title("fill_between");
```



Matplotlib: Exploratory data analysis

- Scale: use the right scale to understand your data

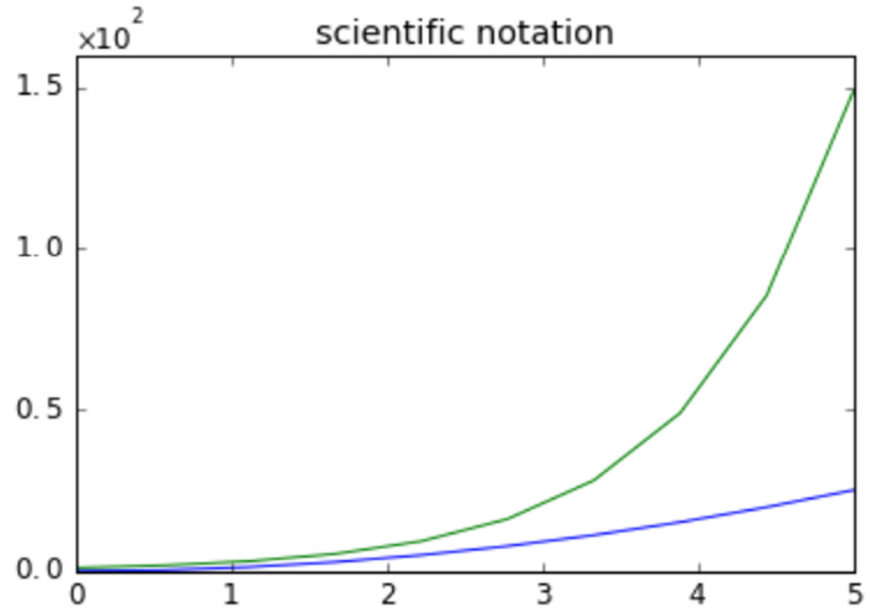
```
fig, axes = plt.subplots(1, 2, figsize=(10,4))

axes[0].plot(x, x**2, x, np.exp(x))
axes[0].set_title("Normal scale")

axes[1].plot(x, x**2, x, np.exp(x))
axes[1].set_yscale("log")
axes[1].set_title("Logarithmic scale (y)");
```



Matplotlib: Exploratory data analysis

- Scientific notation:

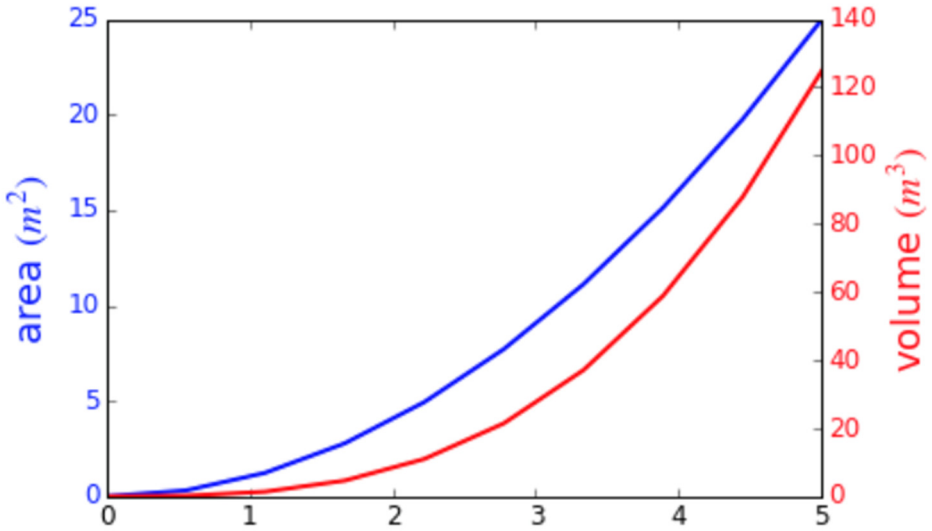
```
fig, ax = plt.subplots(1, 1)

ax.plot(x, x**2, x, np.exp(x))
ax.set_title("scientific notation")

ax.set_yticks([0, 50, 100, 150])

from matplotlib import ticker
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
```



Matplotlib: Exploratory data analysis

- Twin axes: show two dimensions in the same plot

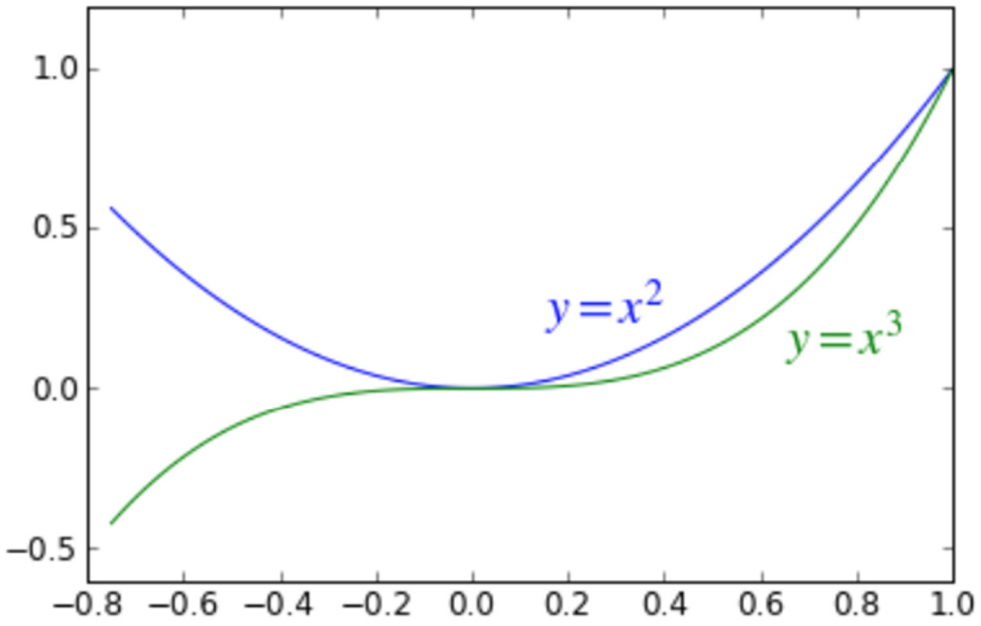
```
fig, ax1 = plt.subplots()

ax1.plot(x, x**2, lw=2, color="blue")
ax1.set_ylabel(r"area $(m^2)$", fontsize=18, color="blue")
for label in ax1.get_yticklabels():
    label.set_color("blue")

ax2 = ax1.twinx()
ax2.plot(x, x**3, lw=2, color="red")
ax2.set_ylabel(r"volume $(m^3)$", fontsize=18, color="red")
for label in ax2.get_yticklabels():
    label.set_color("red")
```



Matplotlib: Exploratory Data Analysis

- Text annotation: explain your analytical result

```
fig, ax = plt.subplots()

ax.plot(xx, xx**2, xx, xx**3)

ax.text(0.15, 0.2, r"$y=x^2$", fontsize=20, color="blue")
ax.text(0.65, 0.1, r"$y=x^3$", fontsize=20, color="green");
```



5. Exploratory data analysis with Seaborn

- Office website: https://seaborn.pydata.org/
- Built on top of matplotlib
- Features:
  - Beautiful default styles → less customization effort than Matplotlib
  - Exploratory data analysis
  - Statistical data analysis (next lecture)
  - Designed to work well with Pandas data frame
  - A rich gallery: https://seaborn.pydata.org/

Seaborn: Data Distribution

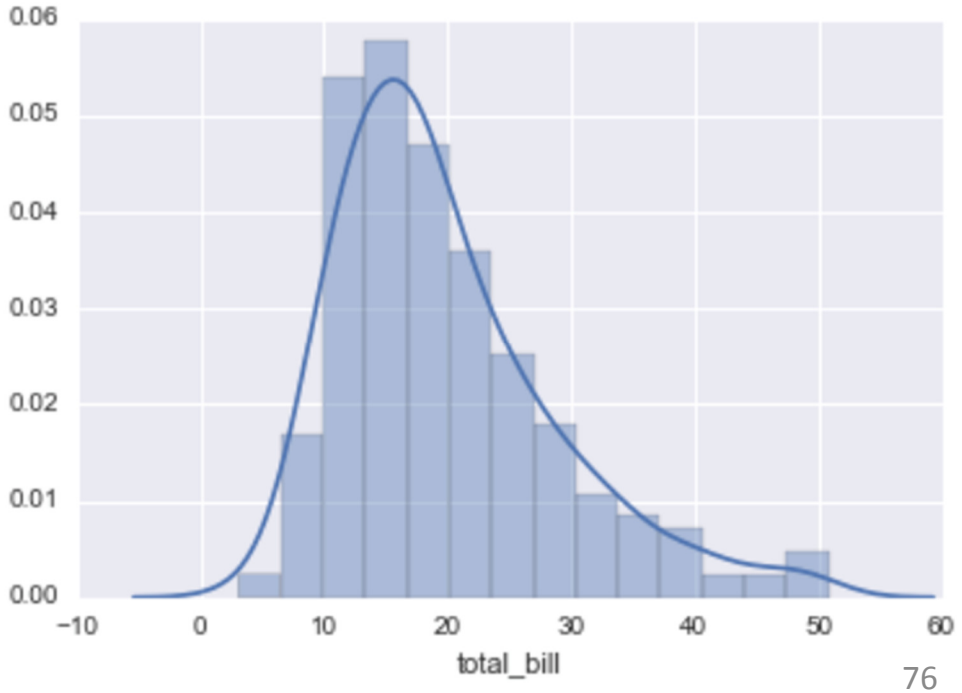
- Histogram for a single variable

```
tips = sns.load_dataset('tips')
```

```
tips.head()
```

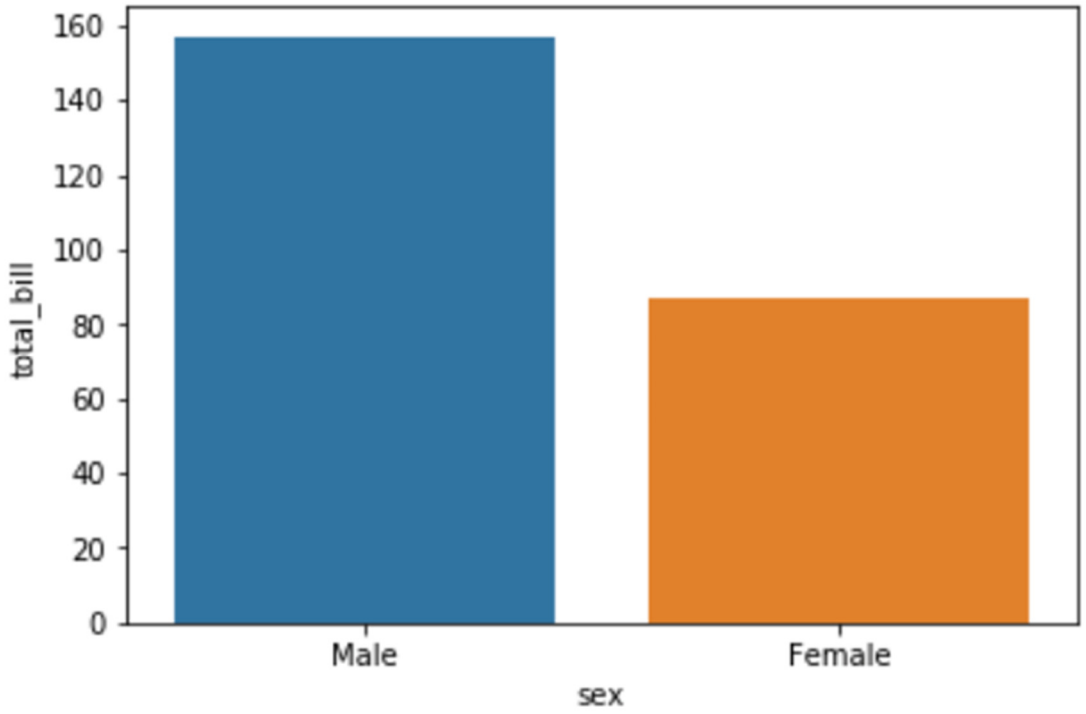
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
sns.distplot(tips['total_bill'])
```



Seaborn: Data Distribution

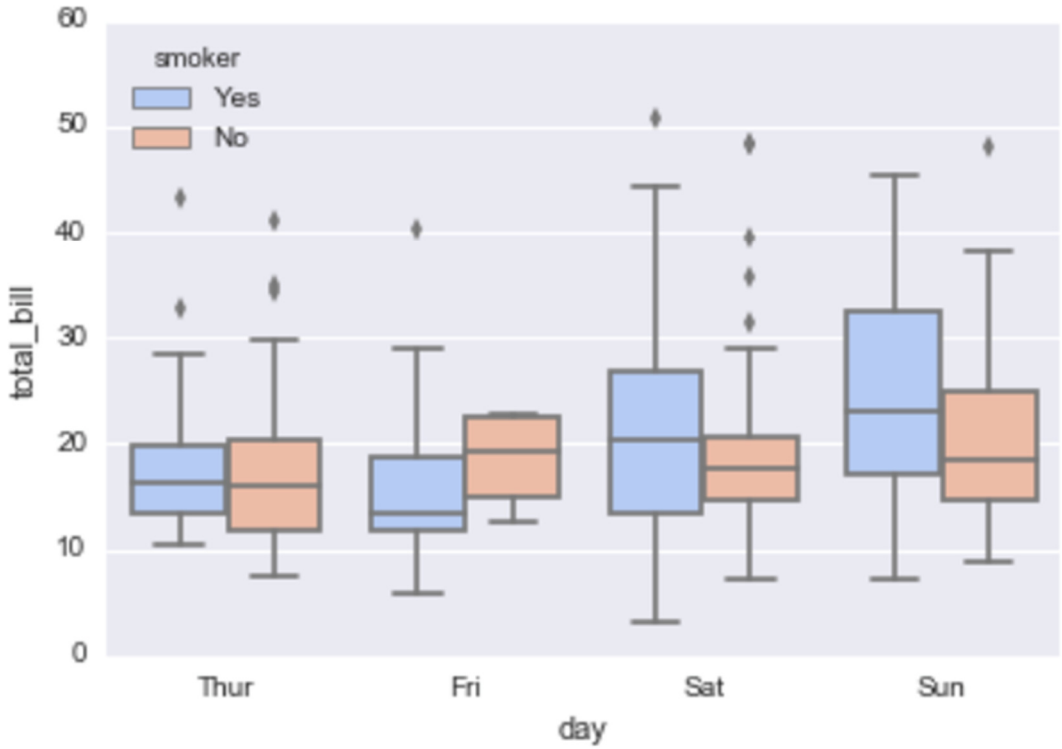
- Aggregate your data with barplot



sns.barplot(x='sex',y='total_bill', data=tips, estimator=len)

Seaborn: Data Distribution

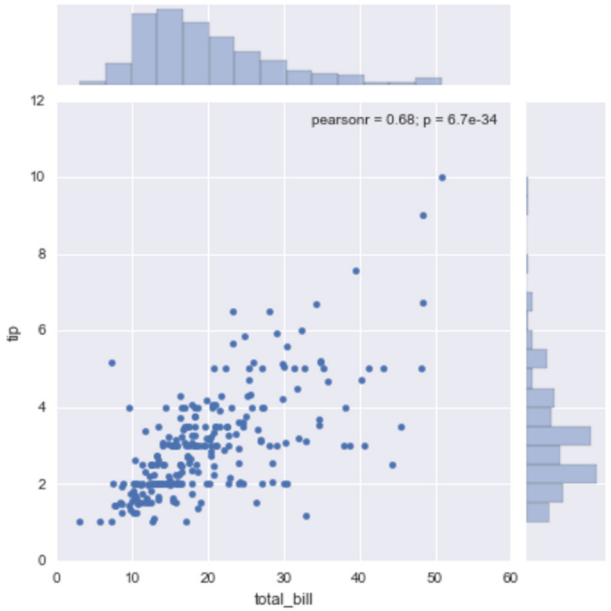
- Box-and-whisker plot



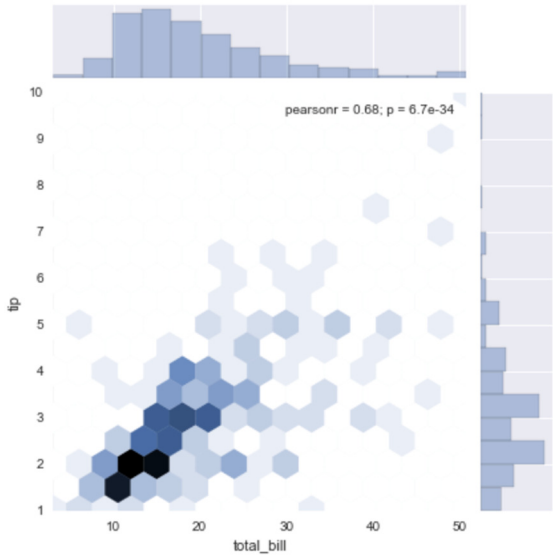

```
sns.boxplot(x="day", y="total_bill", hue="smoker",data=tips, palette="coolwarm")
```

Seaborn: Data Relationship

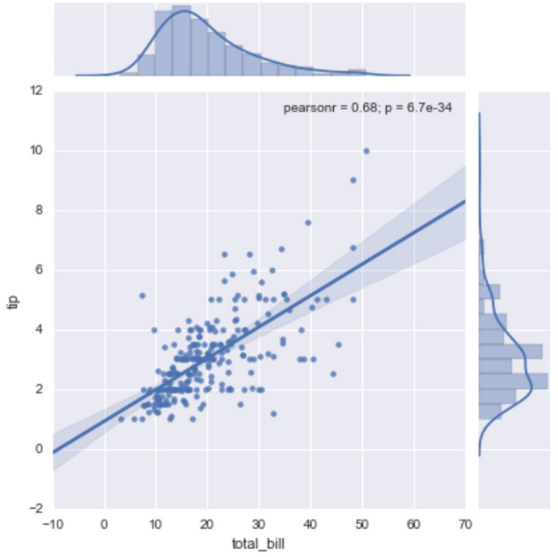
- Jointplot do distribution and correlation at once
  - sns.jointplot(x='total_bill',y='tip',data=tips,kind='scatter')
  - sns.jointplot(x='total_bill',y='tip',data=tips,kind='hex')
  - sns.jointplot(x='total_bill',y='tip',data=tips,kind='reg')



With scatter



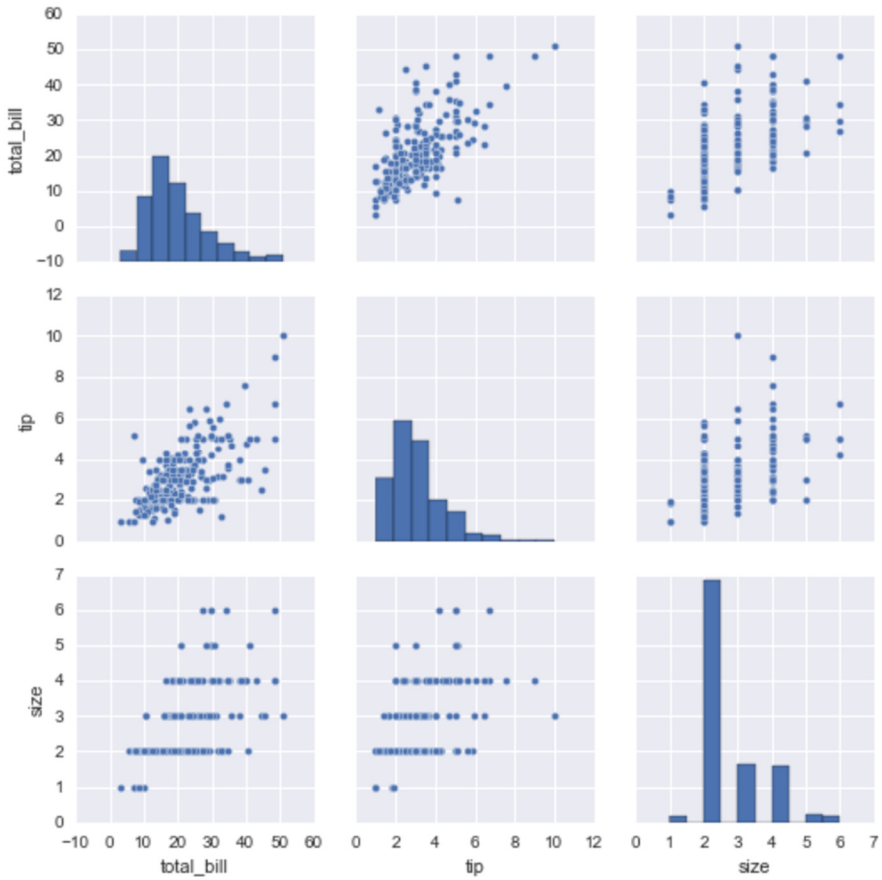
With hexagon shape



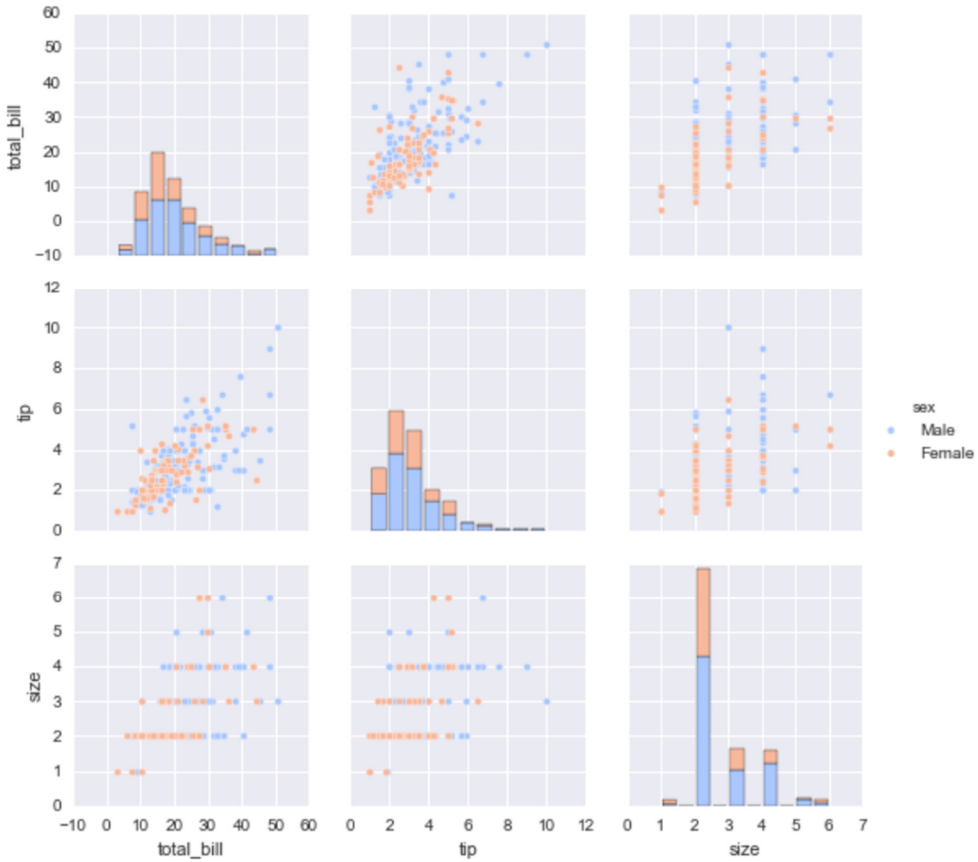
With linear regression

Seaborn: Data Relationship

- Pairplot automatically analyze pairwise relationships



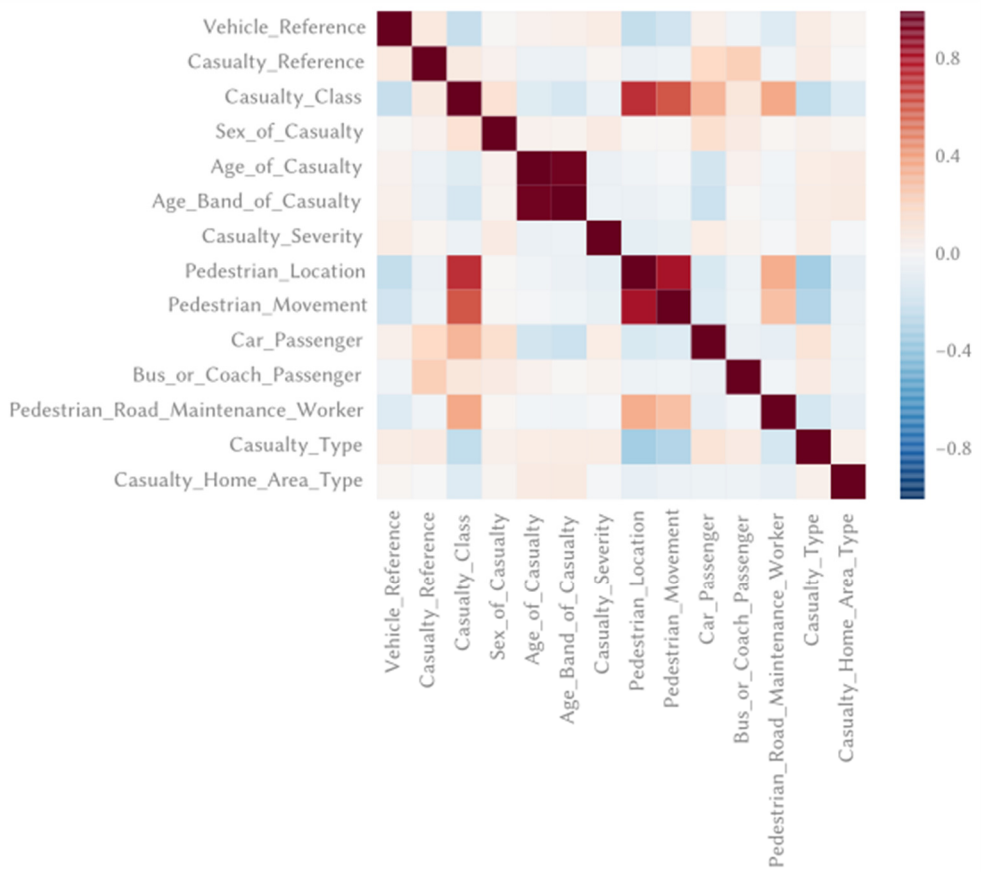
sns.pairplot(tips)



sns.pairplot(tips,hue='sex',palette='coolwarm')

Seaborn: Data Relationship

- Correlation matrix



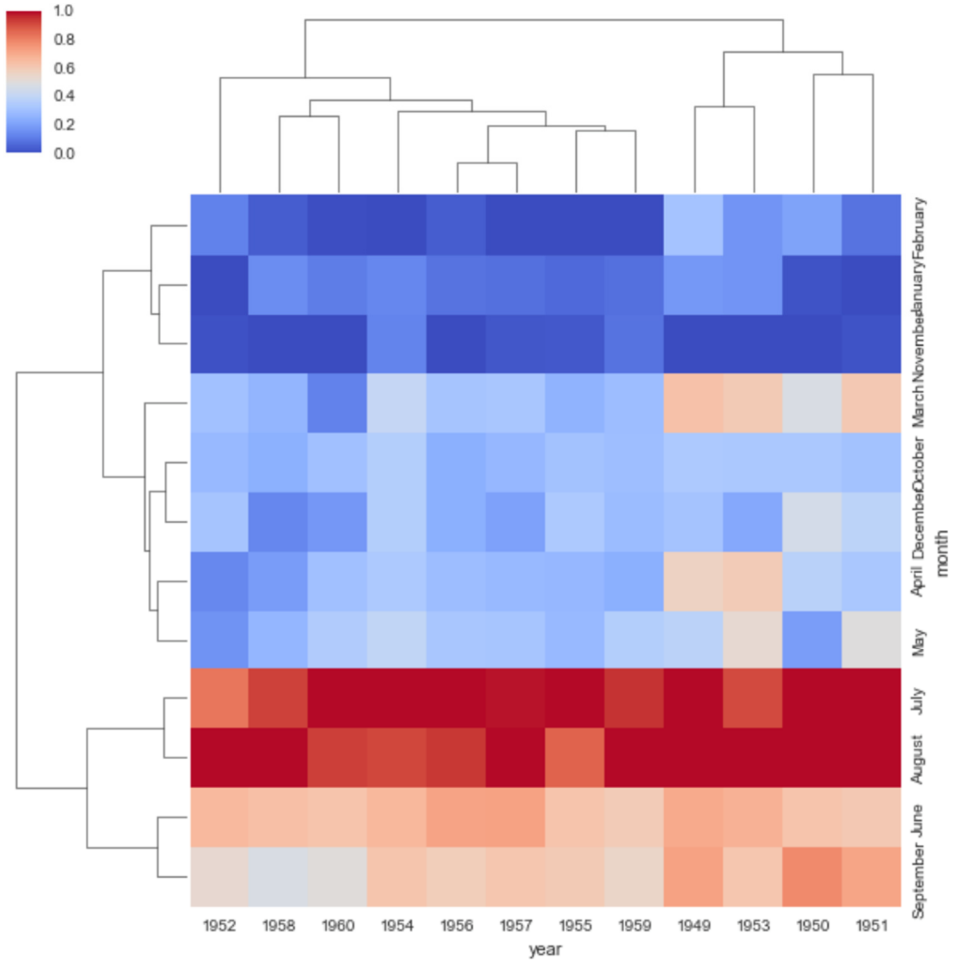
Analysis of the correlations between different variables affecting road casualties

```
import pandas
import matplotlib.pyplot as plt
import seaborn as sns

data = pandas.read_csv("casualties.csv")
cm = data.corr()
sns.heatmap(cm, square=True)
plt.yticks(rotation=0)
plt.xticks(rotation=90)
```

Data source: UK Department for Transport, data.gov.uk/dataset/road-accidents-safety-data

Seaborn: Data Similarity

- ClusterMap: use hierarchical clustering on heatmap



sns.clustermap(pvflights,cmap='coolwarm',standard_scale=1)

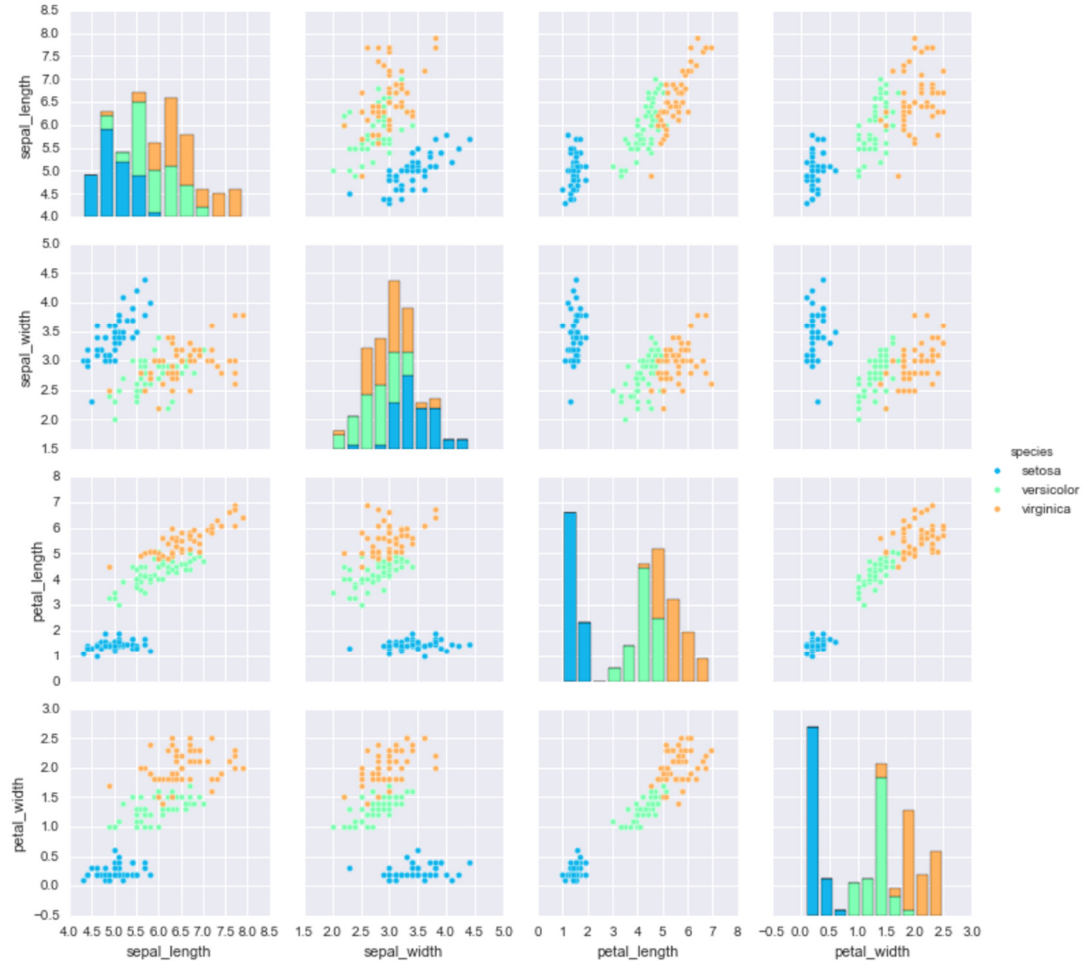
Seaborn: Exploratory Data Analysis

- Grids: combine them together

```
iris = sns.load_dataset('iris')
```

```
iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa



sns.pairplot(iris,hue='species',palette='rainbow')

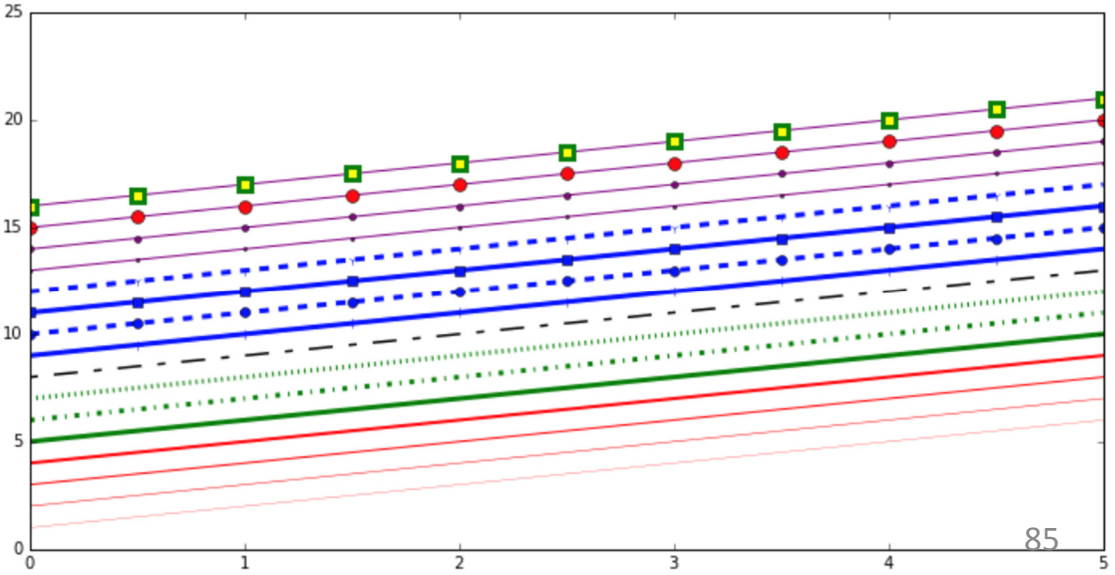
References

[1] http://scikit-learn.org/stable/auto_examples/bicluster/plot_spectral_coclustering.html
[2] https://www.kaggle.com/ekami66/detailed-exploratory-data-analysis-with-python
[3] https://www.slideshare.net/ajandne/pearson-correlation
[4] https://www.slideshare.net/AnishMaman/correlation-36754774
[5] https://www.slideshare.net/EricMarsden1/modelling-correlations-using-python
[6] https://lagunita.stanford.edu/courses/OLI/StatReasoning/Open/about
[7] https://www.slideshare.net/killver/modeling-and-mining-sequential-data
[8] https://www.slideshare.net/hnly228078/spectral-clustering-tutorial
[9] https://www.slideshare.net/AllenWu/information-theoretic-co-clustering
[10] https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/

Matplotlib: Exploratory Data Analysis

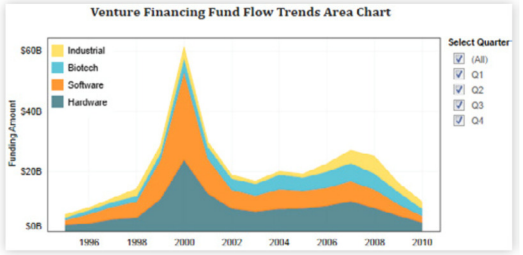
- Markers and Colors: represents different categories

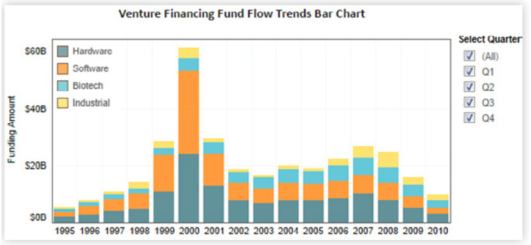
```
fig, ax = plt.subplots(figsize=(12,6))

ax.plot(x, x+1, color="red", linewidth=0.25)
ax.plot(x, x+2, color="red", linewidth=0.50)
ax.plot(x, x+3, color="red", linewidth=1.00)
ax.plot(x, x+4, color="red", linewidth=2.00)

# possible linestype options '-', '--', '-.', ':', 'steps'
ax.plot(x, x+5, color="green", lw=3, linestyle='-')
ax.plot(x, x+6, color="green", lw=3, ls='-.')
ax.plot(x, x+7, color="green", lw=3, ls=':')

# custom dash
line, = ax.plot(x, x+8, color="black", lw=1.50)
line.set_dashes([5, 10, 15, 10]) # format: line length, space length, ...

# possible marker symbols: marker = '+', 'o', '*', 's', ',', '.', '1', '2', '3', '4', ...
ax.plot(x, x+ 9, color="blue", lw=3, ls='-', marker='+')
ax.plot(x, x+10, color="blue", lw=3, ls='--', marker='o')
ax.plot(x, x+11, color="blue", lw=3, ls='-', marker='s')
ax.plot(x, x+12, color="blue", lw=3, ls='--', marker='1')

# marker size and color
ax.plot(x, x+13, color="purple", lw=1, ls='-', marker='o', markersize=2)
ax.plot(x, x+14, color="purple", lw=1, ls='-', marker='o', markersize=4)
ax.plot(x, x+15, color="purple", lw=1, ls='-', marker='o', markersize=8, markerfacecolor="red")
ax.plot(x, x+16, color="purple", lw=1, ls='-', marker='s', markersize=8,
        markerfacecolor="yellow", markeredgewidth=3, markeredgecolor="green");
```

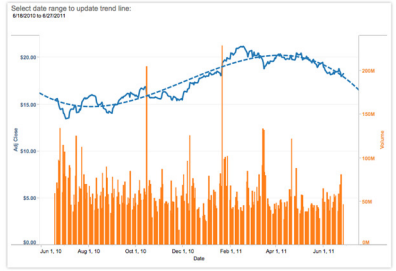


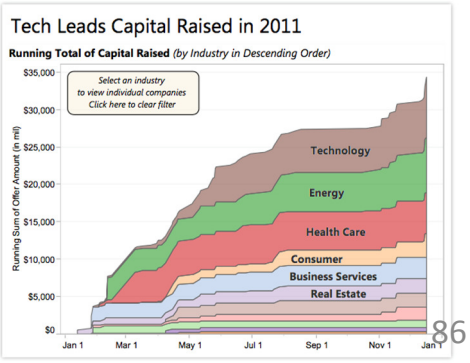
Exploring Temporal Data

- How to present
  - Area charts: drill down into different categories
  - Bar charts: drill down into each year: bar charts
  - Combine a line graph with bar chart
  - Shade the area under lines



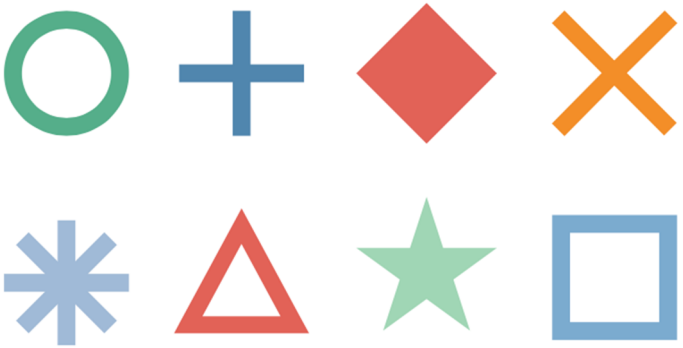


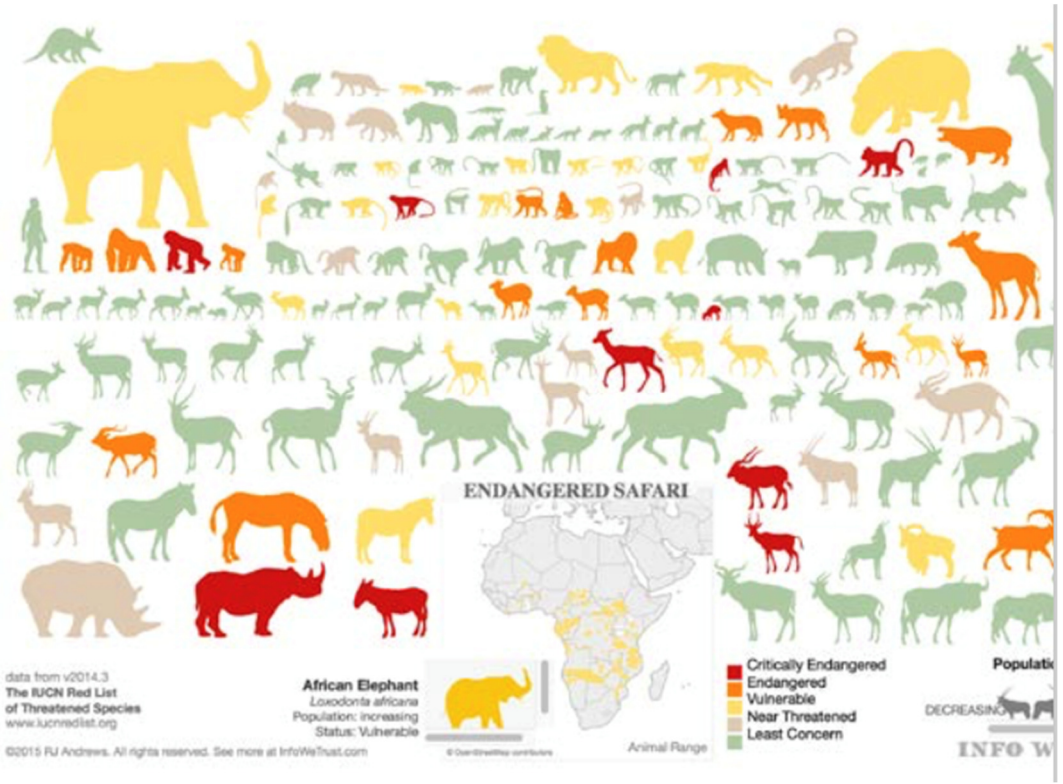




Custom shapes

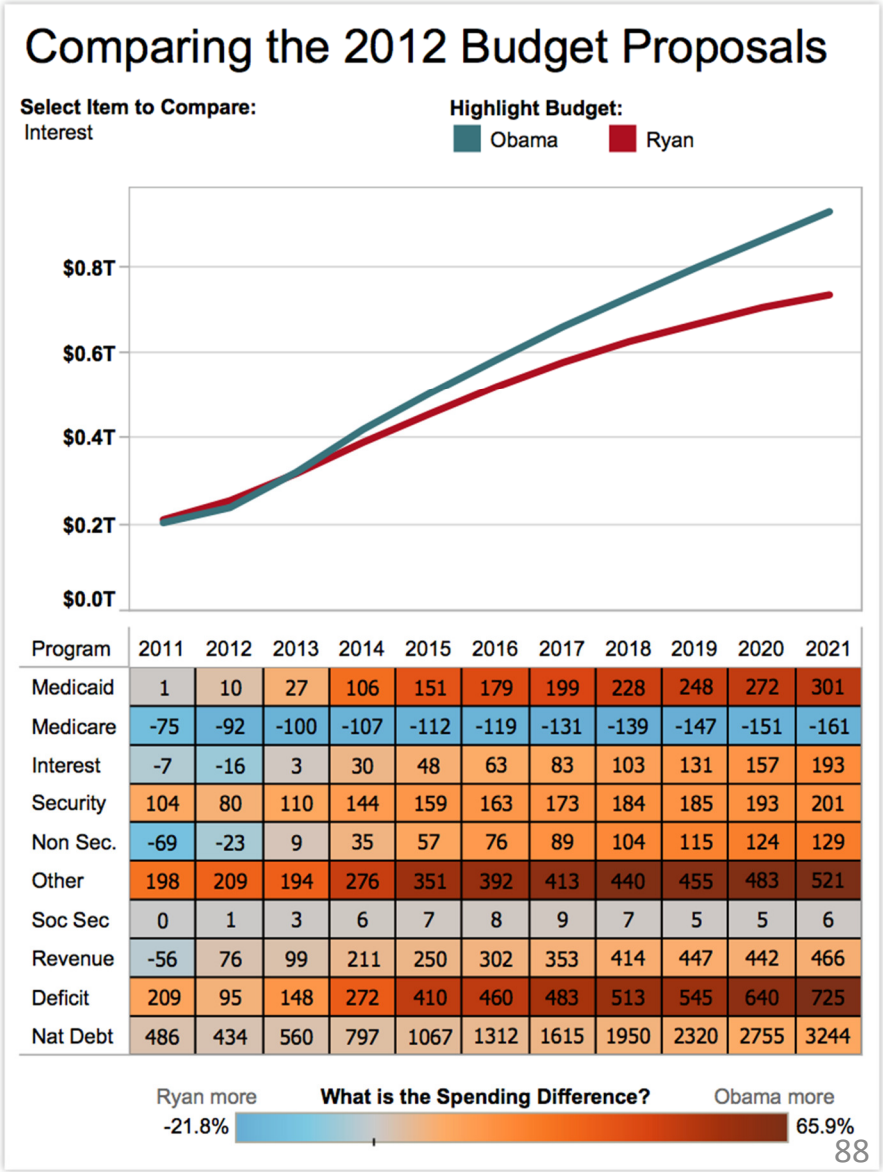
- Use subject matter shapes to tell a more compelling story





Highlight table

- Add a number on top of color
- When to use: providing detailed information on heat maps
  - Examples: the percent of a market for different segments, sales numbers by a reps in a particular region, population of cities in different years.
- Best practices:
  - Combine highlight tables with other chart types



Exploring Data Similarity

- DBScan: density-based clustering
- Problem statement: construct a set of clusters such that
  - Each cluster satisfies
    - Maximality: if $q \in C$ is a core point, and p is density reachable from q, then also $p \in C$
    - Connectivity: any two points in C must be density connected

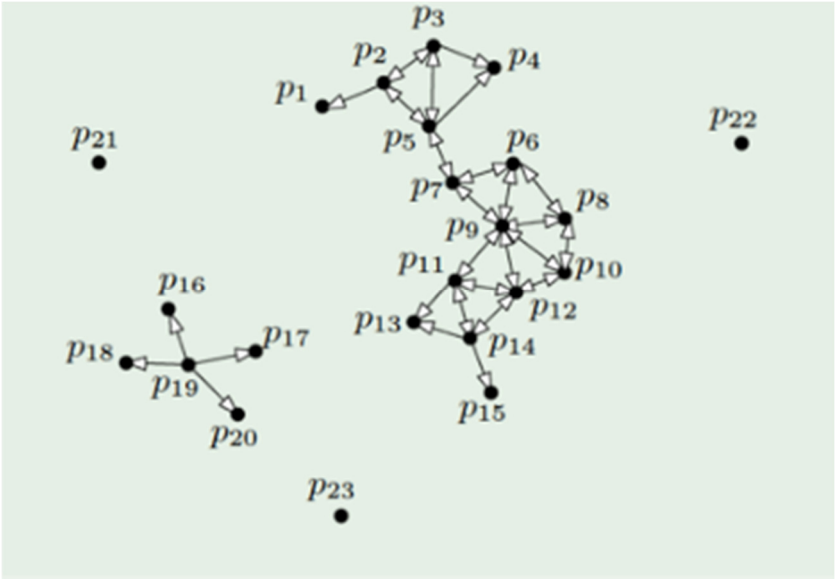
      →A cluster contains at least one core point
  - The set of clusters is unique
  - Clusters are not necessarily disjoint

DBScan Algorithm (Optional)

- Construct a directed graph G using direct density-reachability
- Initialize
  - V_{core}=set of core points
  - P=set of all points
  - set of clusters $C = \{\}$
- While V_{core} not empy
  - Select a point p from V_{core} and construct $S(p)$, the set of all points density-reachable from p: Breadth-first search on G starting from p
  - $C = C \cup \{S(p)\}$
  - $P = P \setminus S(p)$
  - $V_{core} = V_{core} \setminus S_{core}(p)$
    - where $S_{core}(p)$=core points in $S(p)$
- Mark remaining points in P as unclustered

DBScan - Example

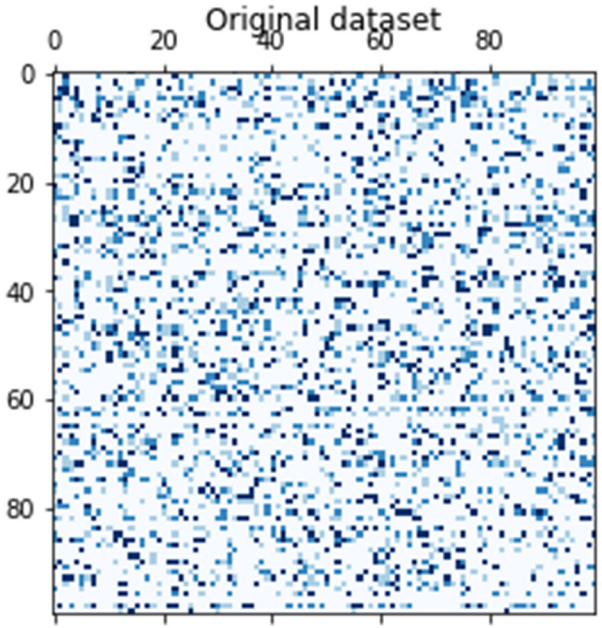
- First cluster: choose p_2 and compute $S(p_2) = \{p_1, \dots, p_{15}\}$
- Then, $V_{core} = \{p_{19}\}$
- Second cluster: $\{p_{16}, \dots, p_{20}\}$
- $p_{21}, \dots, p_{23}$ are outliers

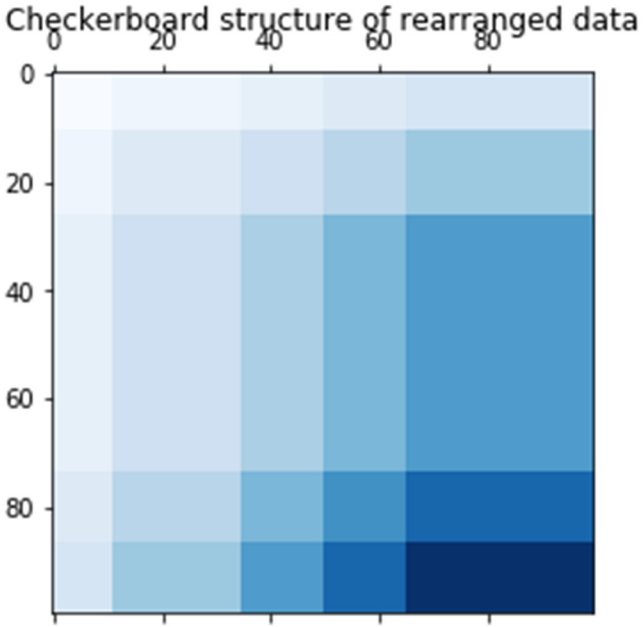


Source: Yufei Tao, Chinese University of Hong Kong

Exploring Data Similarity

- Co-clustering: useful for clustering 2 dimensions at the same time

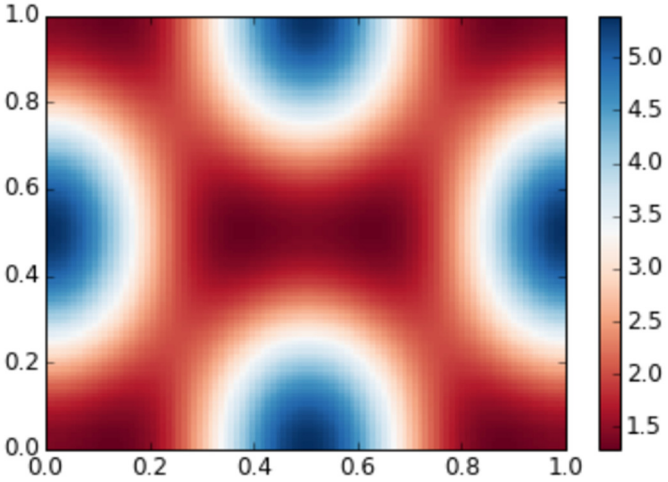




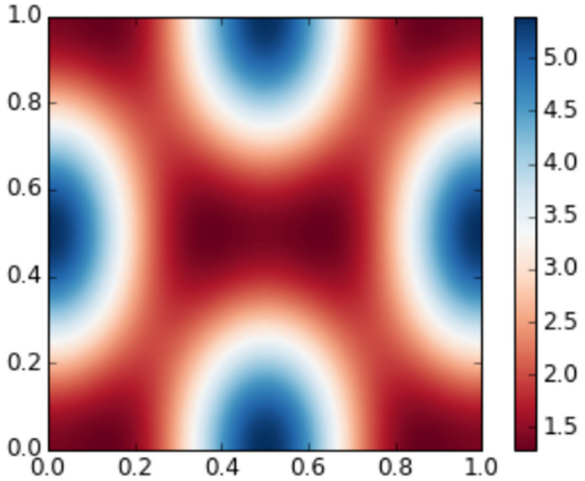
http://scikit-learn.org/stable/auto_examples/bicluster/plot_spectral_coclustering.html

Matplotlib: Exploratory Data Analysis

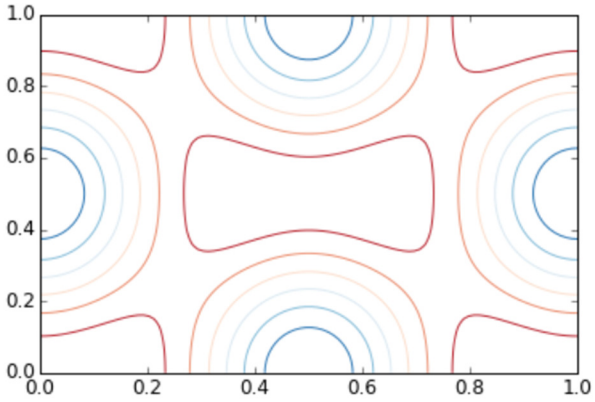
- Colormaps:



pcolor



imshow



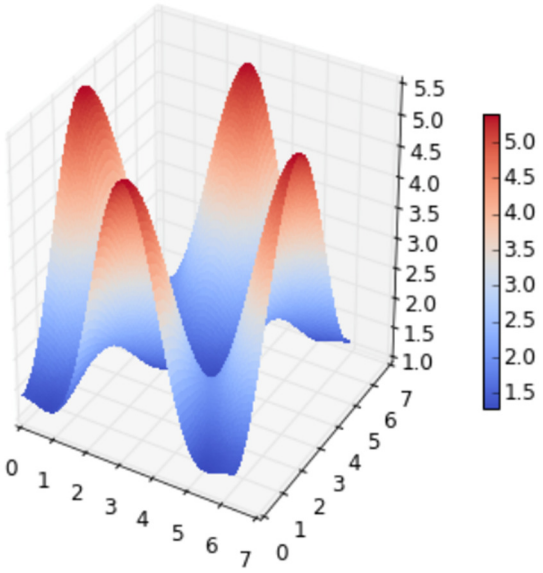
contour

https://stackoverflow.com/questions/21166679/when-to-use-imshow-over-pcolormesh

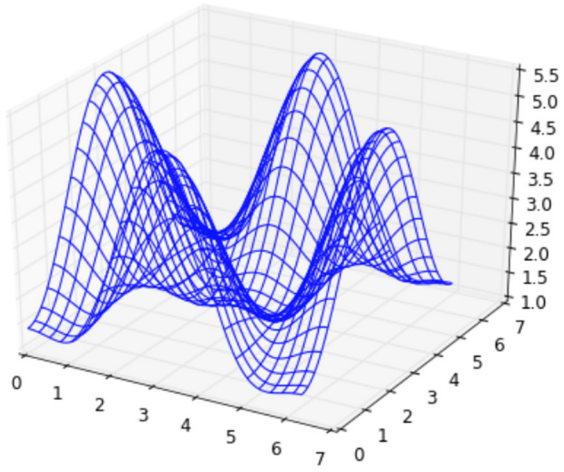
http://thomas-cokelaer.info/blog/2014/05/matplotlib-difference-between-pcolor-pcolormesh-and-imshow/

Matplotlib: Exploratory Data Analysis

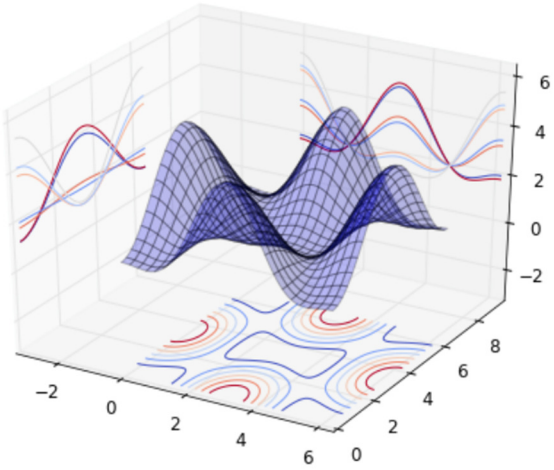
- 3D figures: sometimes we need to analyze data in 3D, but don't overuse it



Surface plot



Wire-frame plot



Contour plot with projections