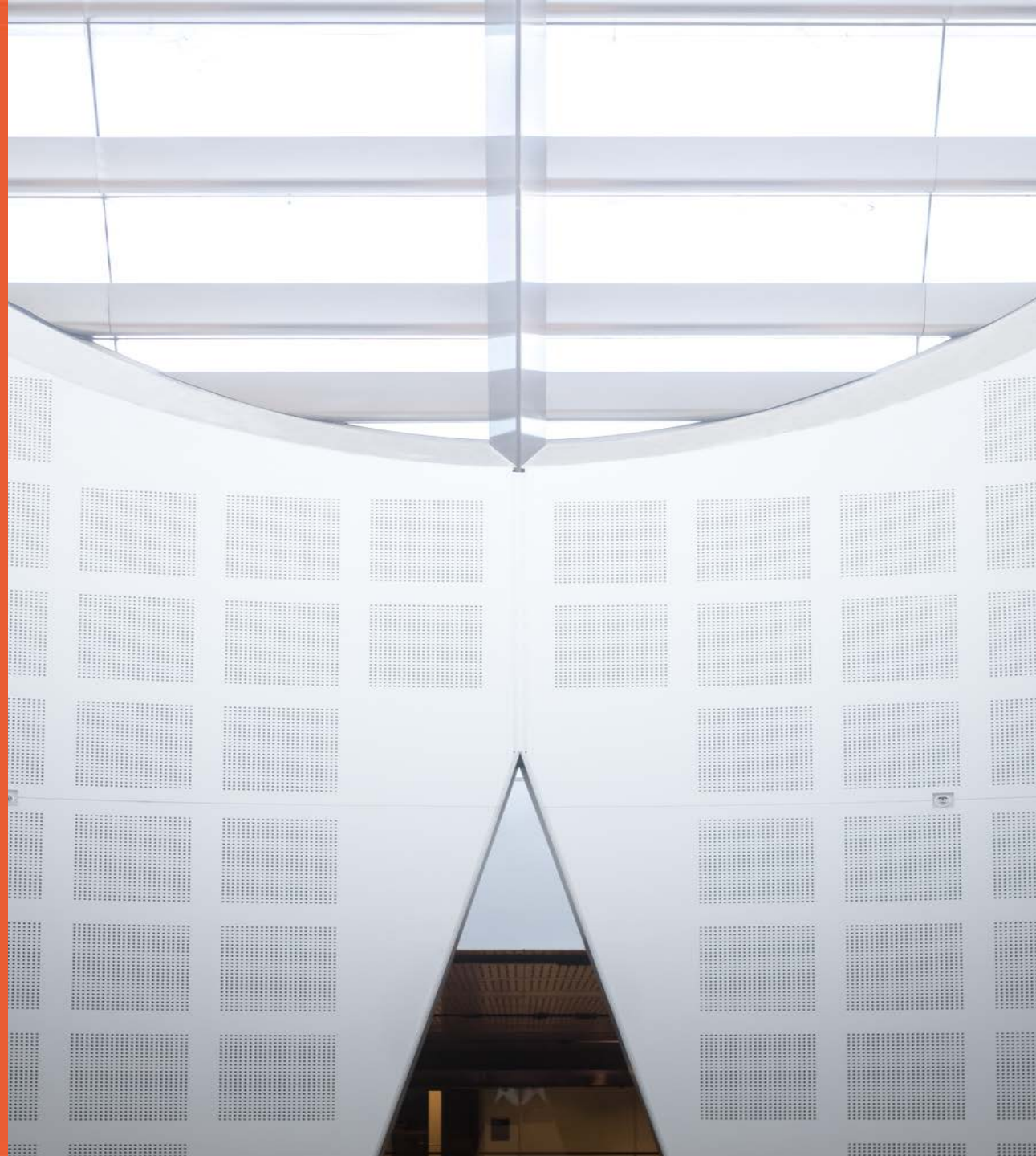
Big Data in Business

BUSS6002 week 10

Presented by
Dr Fabian Held
Fabian.Held@Sydney.edu.au

THE UNIVERSITY OF SYDNEY



Introduction: Dr Fabian Held

- **Senior Advisor** Strategic Ventures
  Strategy Portfolio, University of Sydney
- **Senior Research Fellow** School of Pharmacy, Faculty of Medicine and Health
  - **Research Associate** in Computational Modelling, Charles Perkins Centre and School of Mathematics and Statistics
  - PhD in Marketing
    (U. Sydney, Australia)
  - MSc in Statistics
    (LMU Munich, Germany)
  - BA in Philosophy & Economics
    (U. Bayreuth, Germany)
- **Founder of Data Science Sydney meetup**
- **Software Carpentry Instructor**

fabian.held@sydney.edu.au

Consultation: by appointment

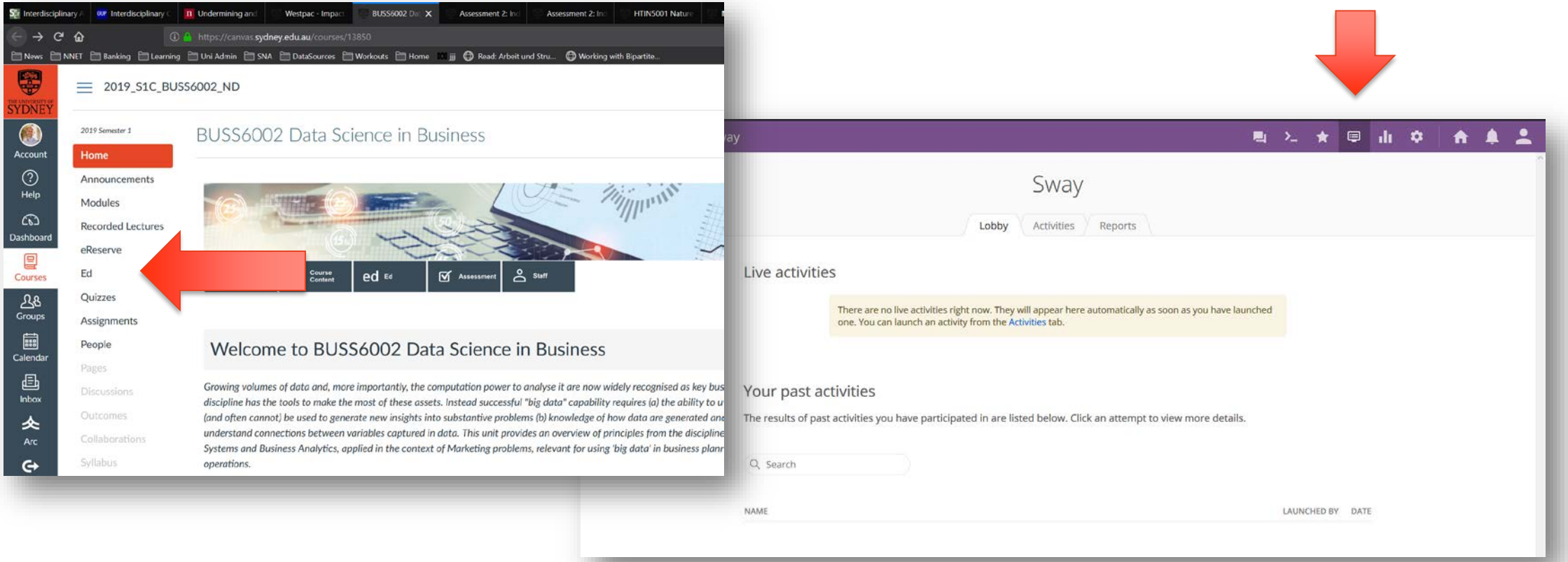
Full Disclosure

Big Data in Marketing

"Marketing is the activity, set of institutions, and processes for creating, communicating, delivering, and exchanging offerings that have value for customers, clients, partners, and society at large."

American Marketing Association

Big Data in Marketing



https://edstem.org/courses/3287/sway/

Overview

Lecture 10

1. Introductions
2. **Data in Marketing, then and now**
3. Unsupervised Learning
    - Clustering
        - Theory
        - Application

Original material by Dr Mike Beweley

Marketing History (1994)

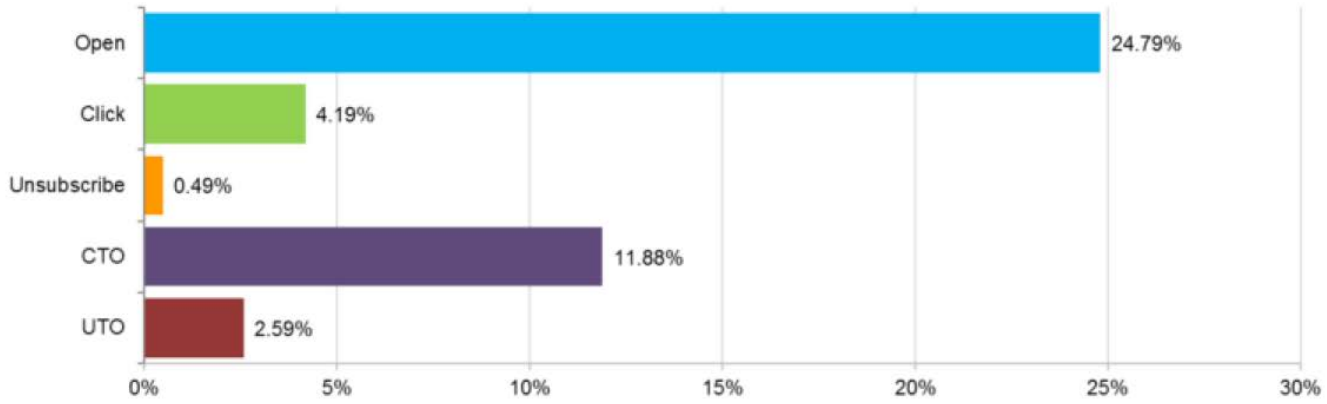
- Email click-through rates of approx. 40%
- The world's first banner ad (AT&T):
  click-through rate of 44%

https://mashable.com/2013/08/09/first-banner-ad/#M7J45e5TIiq3

Where are we now? Email

Cross-industry benchmarks for how many **emails** are:

1. Opened: The email was opened
2. Clicked: A link in the email is clicked

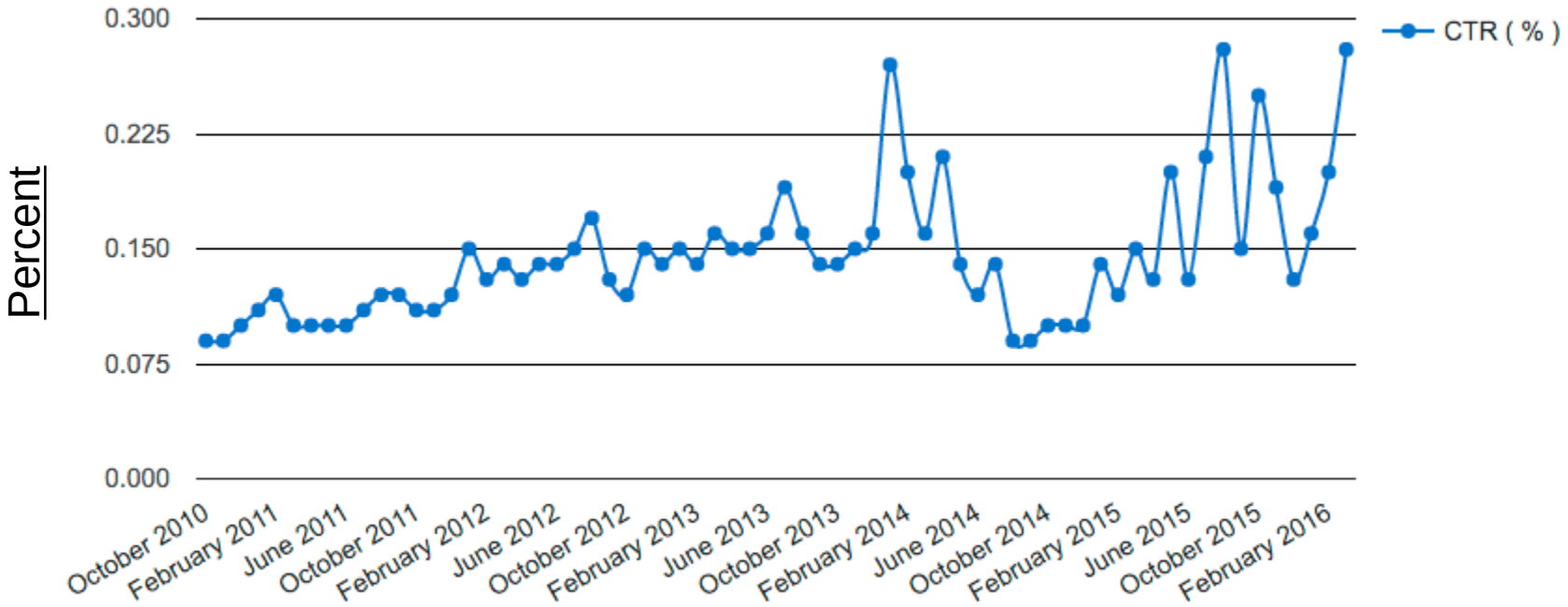


https://www.signupto.com/email-marketing-benchmarks/

Rates by industry

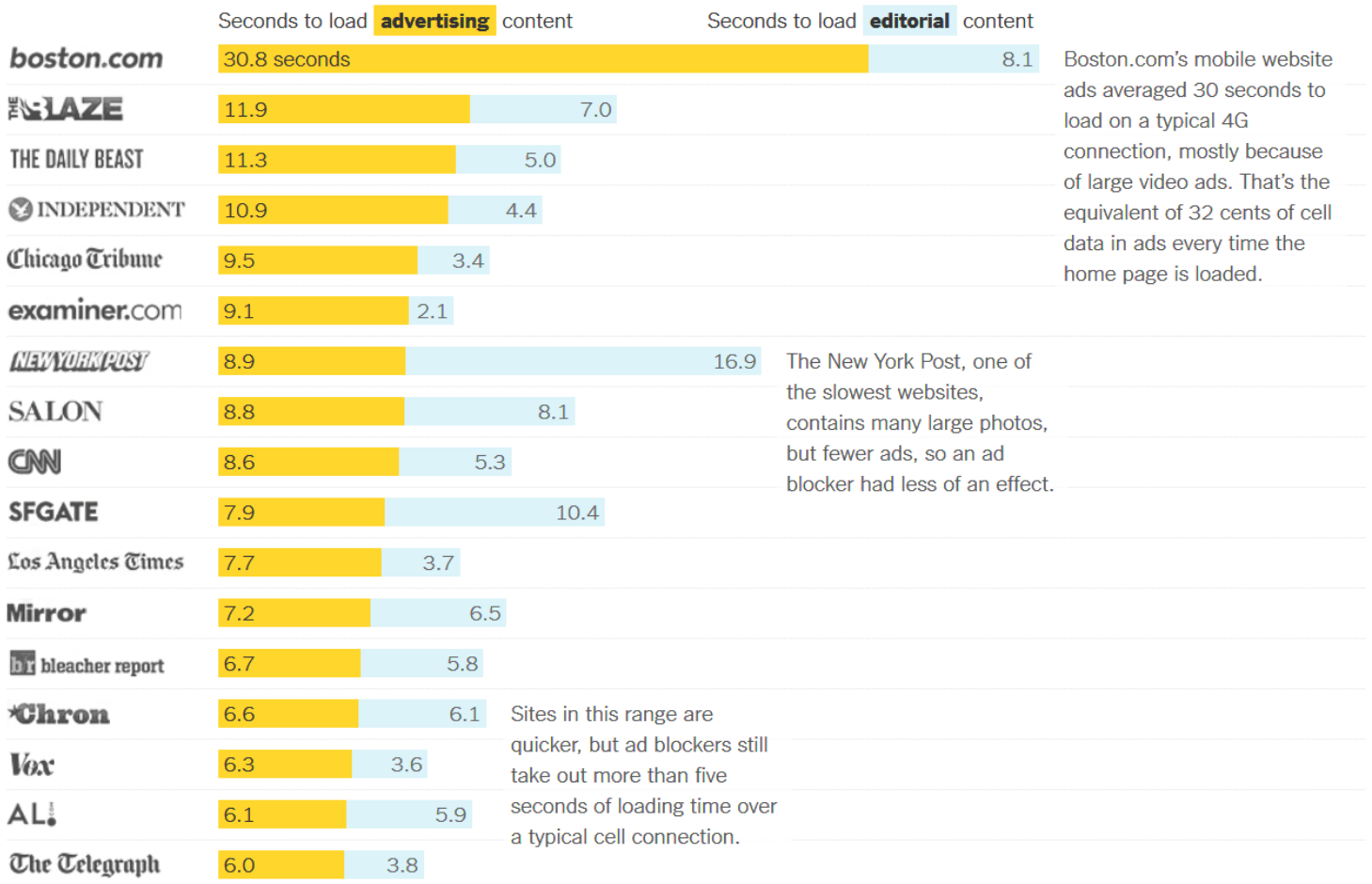
Open	Open rate (%)	Clickthrough rate (%)
Agriculture and Food Services	24.71%	2.98%
Architecture and Construction	24.78%	2.90%
Arts and Artists	27.23%	2.85%
Beauty and Personal Care	18.48%	1.96%
Business and Finance	20.97%	2.73%
Software and Web App	20.95%	2.29%
Sports	25.41%	3.19%
Telecommunications	21.57%	2.43%
Travel and Transportation	20.69%	2.17%
Vitamin Supplements	17.26%	1.80%

Advertisements: Trends in clickthrough rate (CTR)



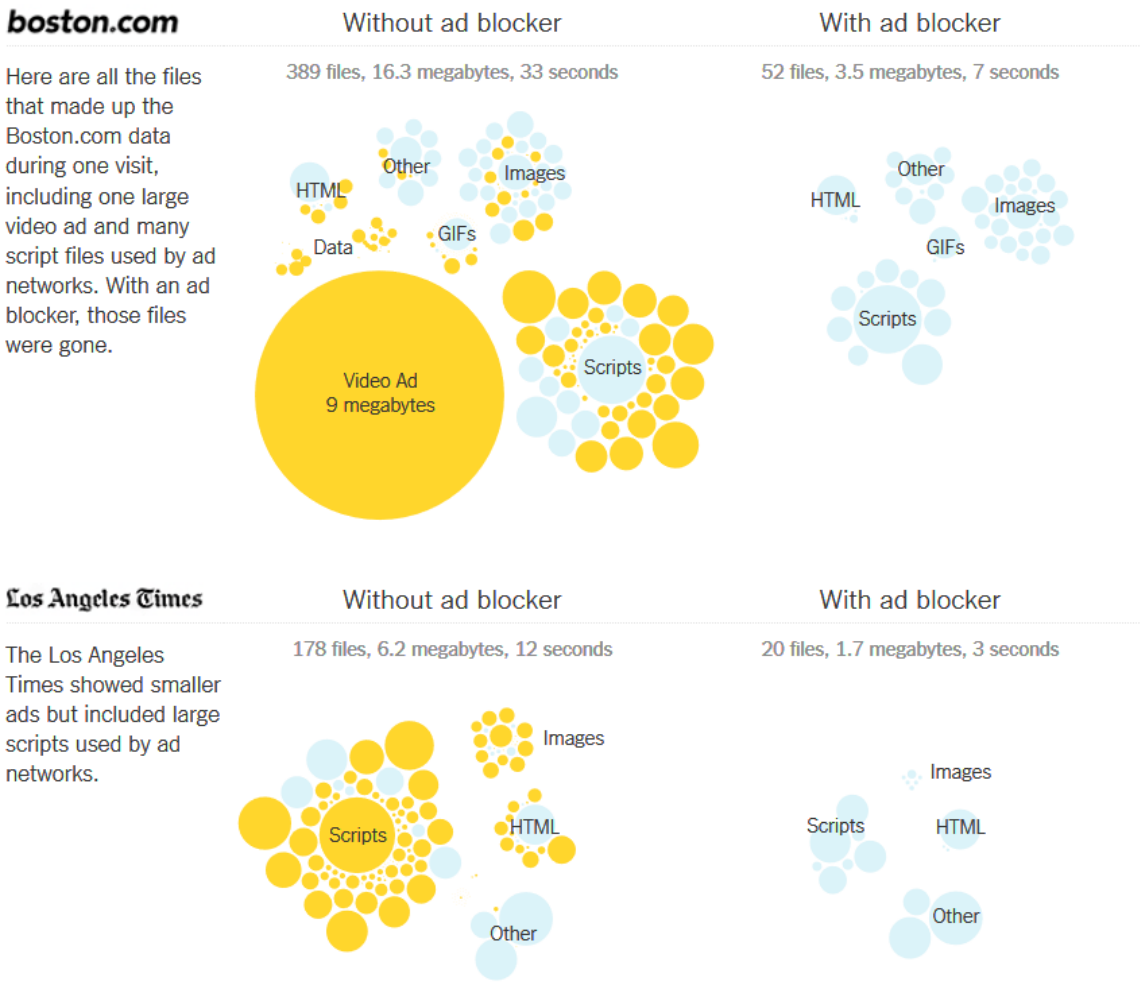
https://www.smartinsights.com/internet-advertising/internet-advertising-analytics/display-advertising-clickthrough-rates/attachment/trends-in-ad-clickthrough-rate/

(Unintended) Consequences of banner ads



https://www.nytimes.com/interactive/2015/10/01/business/cost-of-mobile-ads.html

Many new forms of advertising



https://www.nytimes.com/interactive/2015/10/01/business/cost-of-mobile-ads.html

Customers' Response

- Do you use AdBlockers?

https://edstem.org/courses/2671/sway/

New question for digital marketing

What fraction of these email click throughs (4% CTR) and banner ad click throughs (0.1% CTR) represent genuine interest in a product… that leads to purchase?

Where are we going?

- We need something radically different...
- Contact is a precious resource
  - consider "marketing fatigue".
- Current marketing makes some money, at the cost of good-will of customers/users.
- **Good marketing only contacts the right people, at the right time, when they are ready to take the desired action.**

  Everyone wins!

Why are response rates to traditional marketing so poor?

Models are very simple

- What is the gender and age of the customer?
- I have segmented my customer base into 4 groups, how should I market to each one?

SPEND THE BUDGET mentality

- We might build a great model to predict customers that are interested in a product… then market to the "most promising 40% of the customers", because we have the budget to do so.
- Maybe the model says that only the top 5% are worth talking to?

Advancing segmentation

- A segmentation is a *simplified* model of the world. Market segmentations used by some successful companies include:

	Basic Tech Ability	High Tech Ability
Love Technology	Group 1	Group 2
Hate Technology	Group 3	Group 4

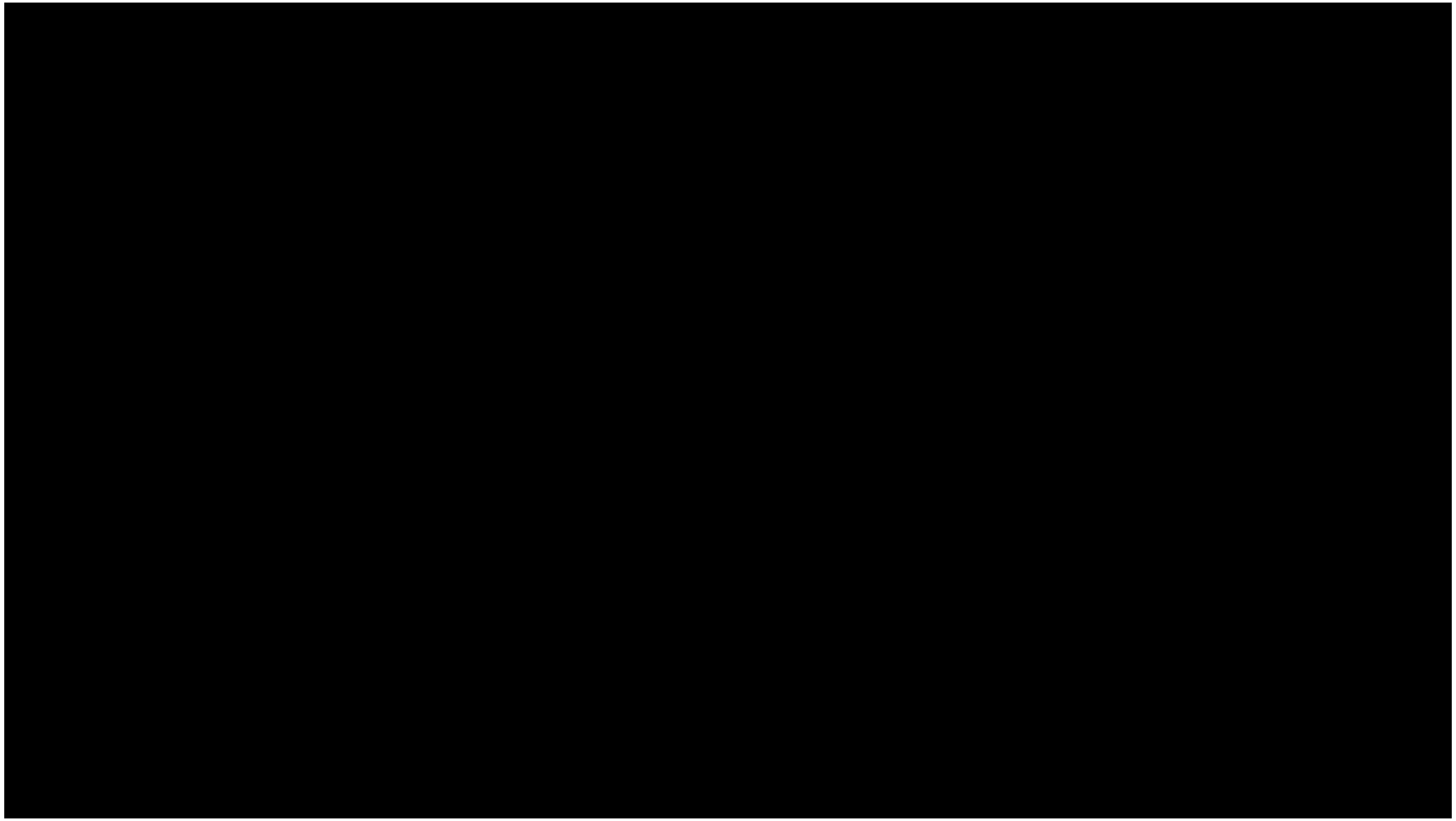
	Has 1 product	Has 2 products	...
18-21 years old	Group 1	Group 2	Group 3
22-25 years old	Group ...	Group ...	Group ...
...	Group ...	Group ...	Group ...

Think – Pair – Share (5mins)

How can we do better in the “Age of Big Data”?

THE UNIVERSITY OF SYDNEY

Local Example



https://www.quantium.com/retail/

Tracking Consumers' Real World Behaviour

Current Tracking Technologies

- Vision Analytics
- Thermal Imaging
- Infrared Beams
- WiFi (Wide Area Network) Tracking
- BLE (Bluetooth Low Energy) Beacons
- GPS (Global Positioning System) Personal Tracker
- RFID (Radio Frequency Identification) Tags & Tracking
- Bio-Metrics (Facial Recognition & Anonymous Demographics)

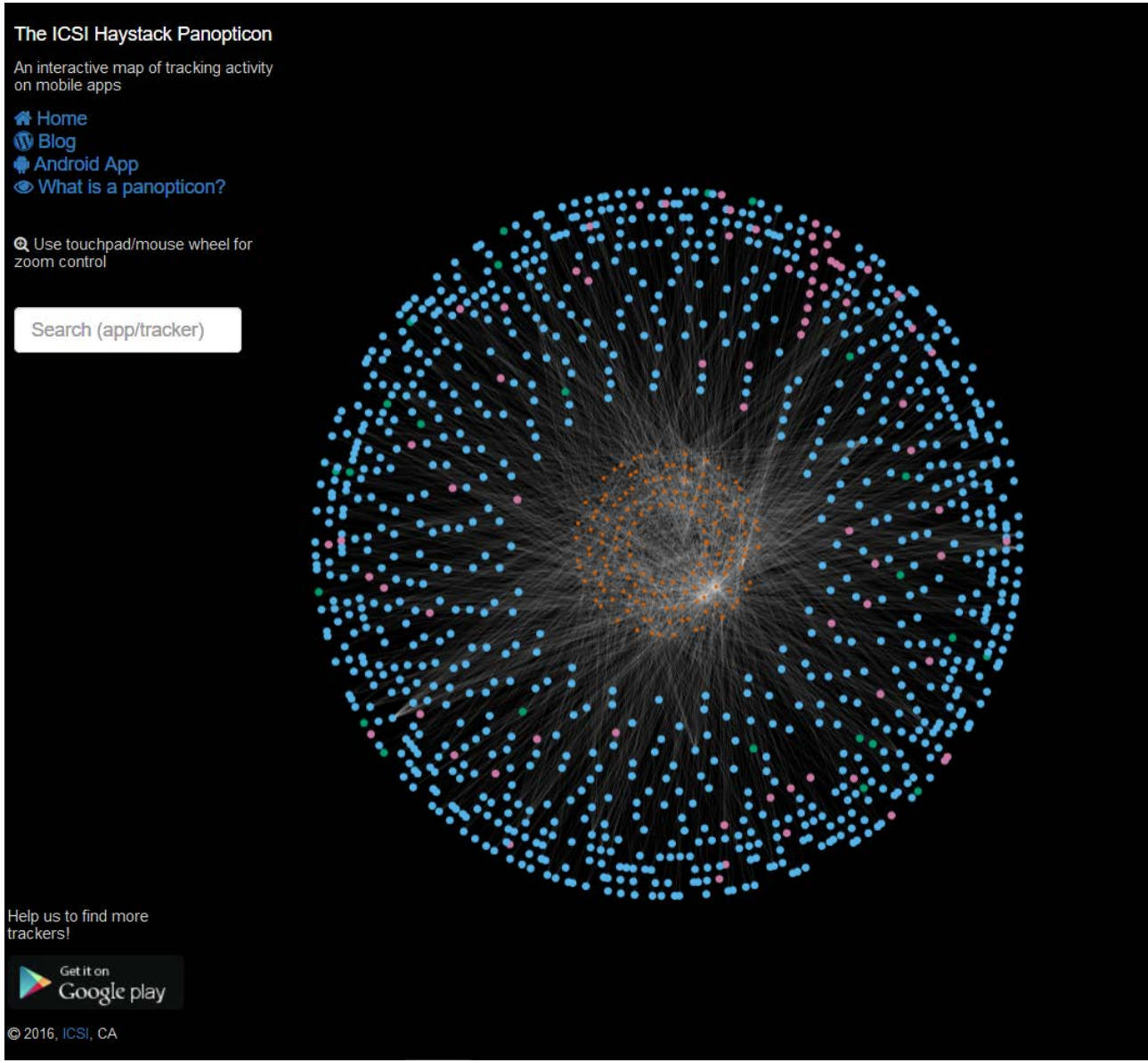
Potential Applications

- Shopper Traffic to store (#Visitors)
- Proximity Traffic (% Capture Rate)
- Choosing Physical Store Site (Trade Area Analytics)
- InStore Product Positioning (Path to Purchase)
- InStore Employee Locations (Service Productivity KPIs)
- InStore Live Map & Product Information (Path Analysis)

Your Mobile Apps track you, too

Lumen App (Android only)

- https://haystack.mobi/
- Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem, Razaghpanah et. al. (2018) *Network and Distributed System Security Symposium* (NDSS)
- https://theconversation.com/7-in-10-smartphone-apps-share-your-data-with-third-party-services-72404

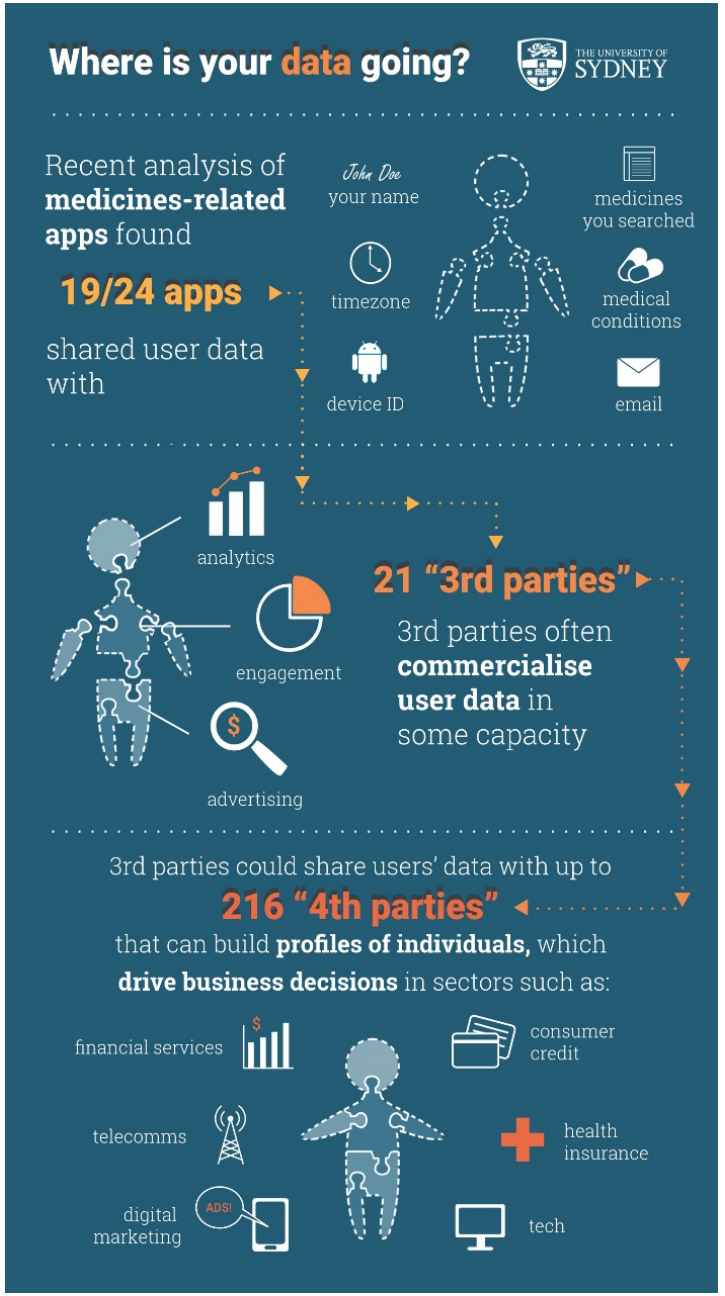


Recent research

Data sharing practices of medicines-related apps and the mobile ecosystem

Quinn Grundy
Fabian Held
Kellia Chiu
Andrea Continella
Lisa Bero
Ralph Holz

BMJ, published March 20th 2019



Ask me 3 questions!

THE UNIVERSITY OF SYDNEY

Overview

Lecture 10

1. Introductions
2. Data in Marketing, then and now
3. **Unsupervised Learning**
    - Clustering
        - Theory
        - Application

Original material by Dr Mike Beweley

What is Unsupervised Learning?

Make a **simpler representation** of the data

- *Without* a pre-conceived idea of what this should look like.
- Contrast to *supervised learning* (regression, etc.), where we have some answer that we're trying to predict.

Dimensionality Reduction

- Dimensionality reduction takes lots of variables (e.g. 20,000 facts about a customer), and combines them mathematically to only a few (the most meaningful ones, hopefully!).

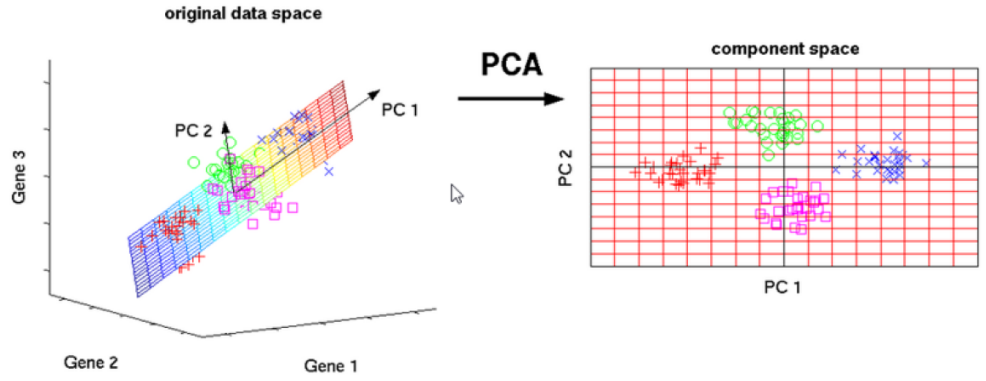
Clustering

- Split data set (e.g. customers, measured by different attributes such as demographics, behaviours, etc.), into a **groups** of observations that look "similar".

From Week 4: Principal Component Analysis (PCA)

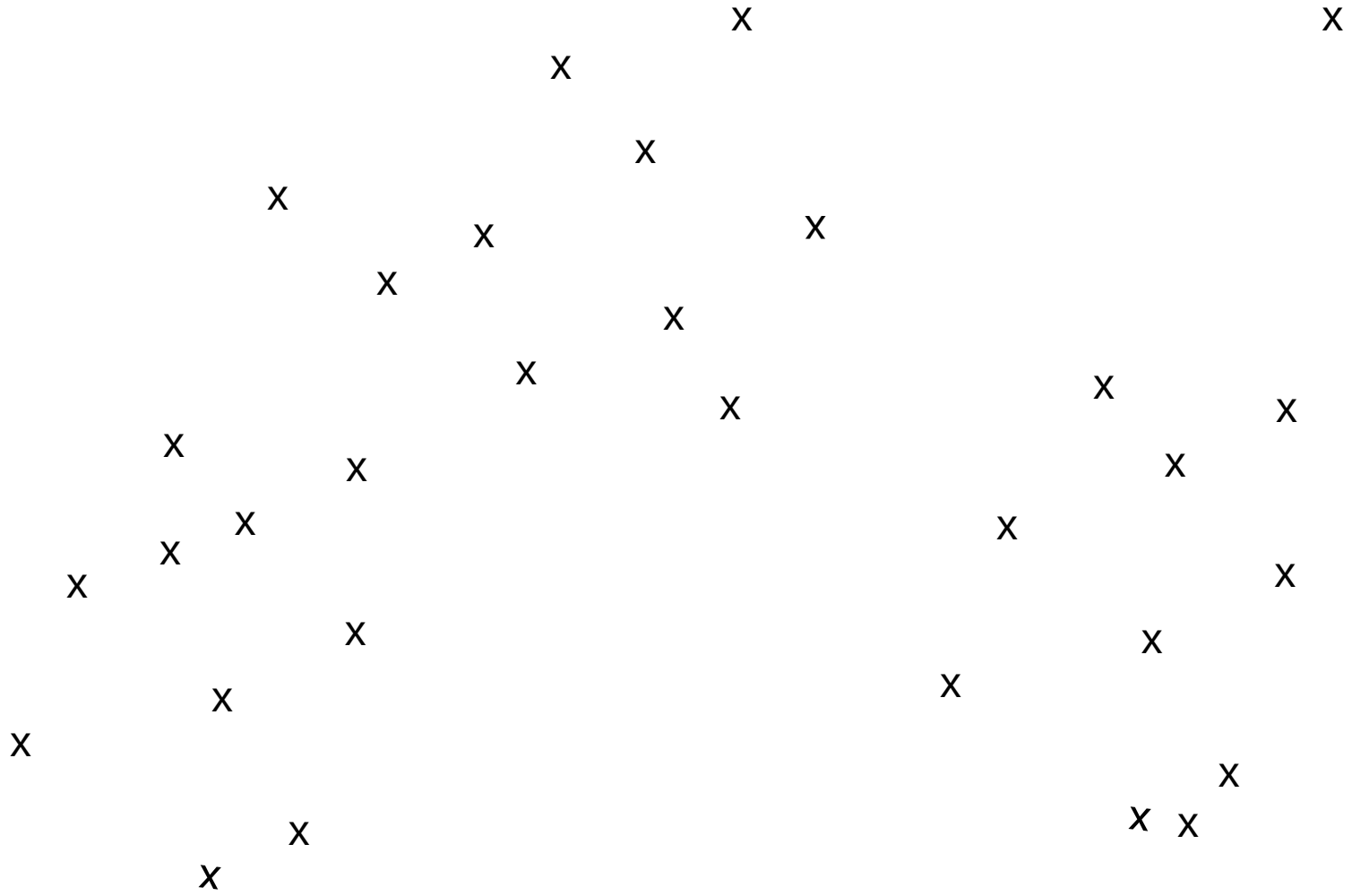
PCA intuition

- PCA projects the original data into a lower-dimensional space with the axes pointing to the directions of highest variance of the data. The axes are termed **principal components** (PCs).
- Find the first PC1, and then the second PC2 orthogonal to PC1. Repeat this process till it is impossible to find another PC that is orthogonal to the previous PCs.
- Keep the top few components and ignore the rest.

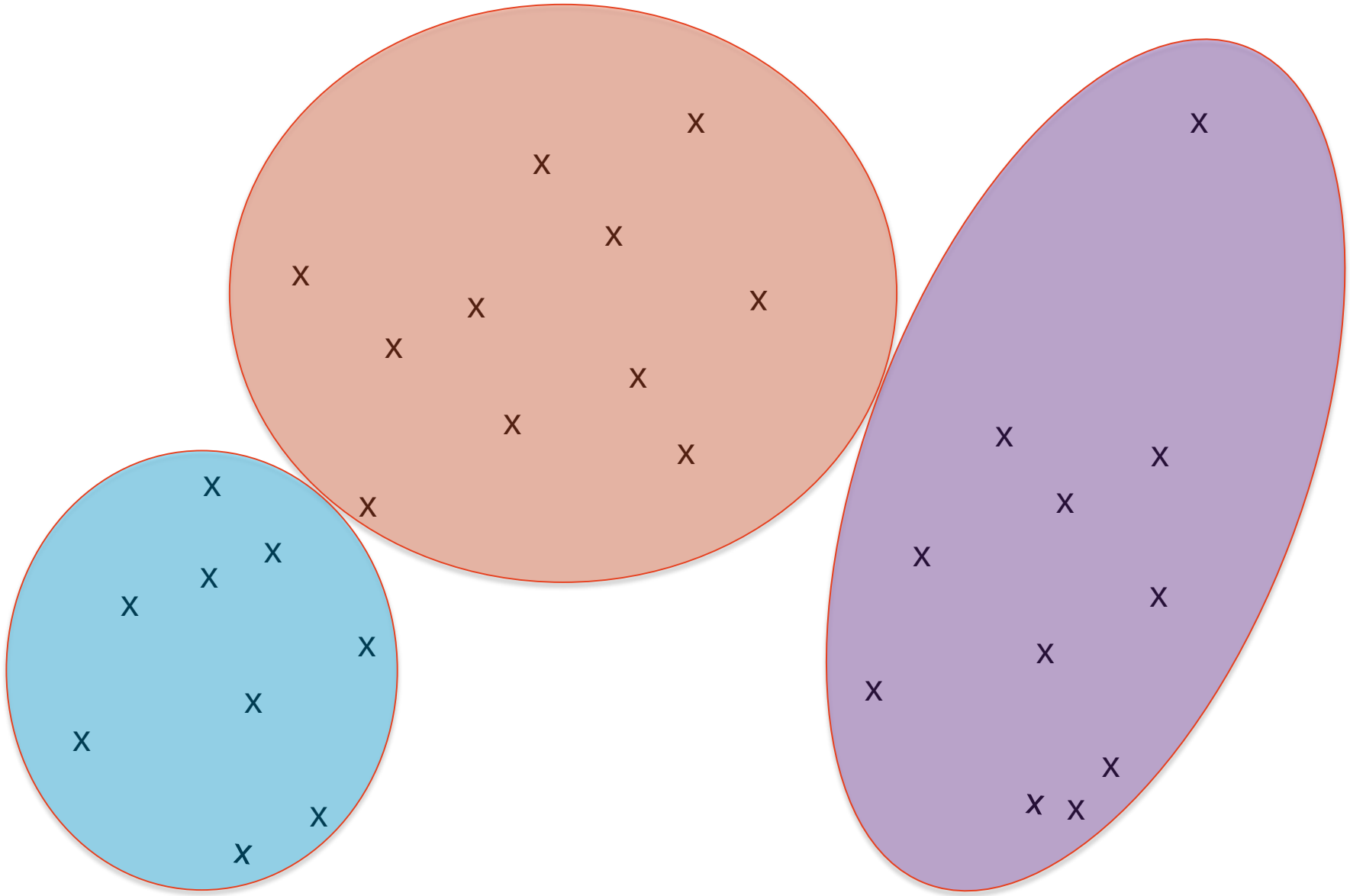


source: http://www.nlpca.org/pca_principal_component_analysis.html

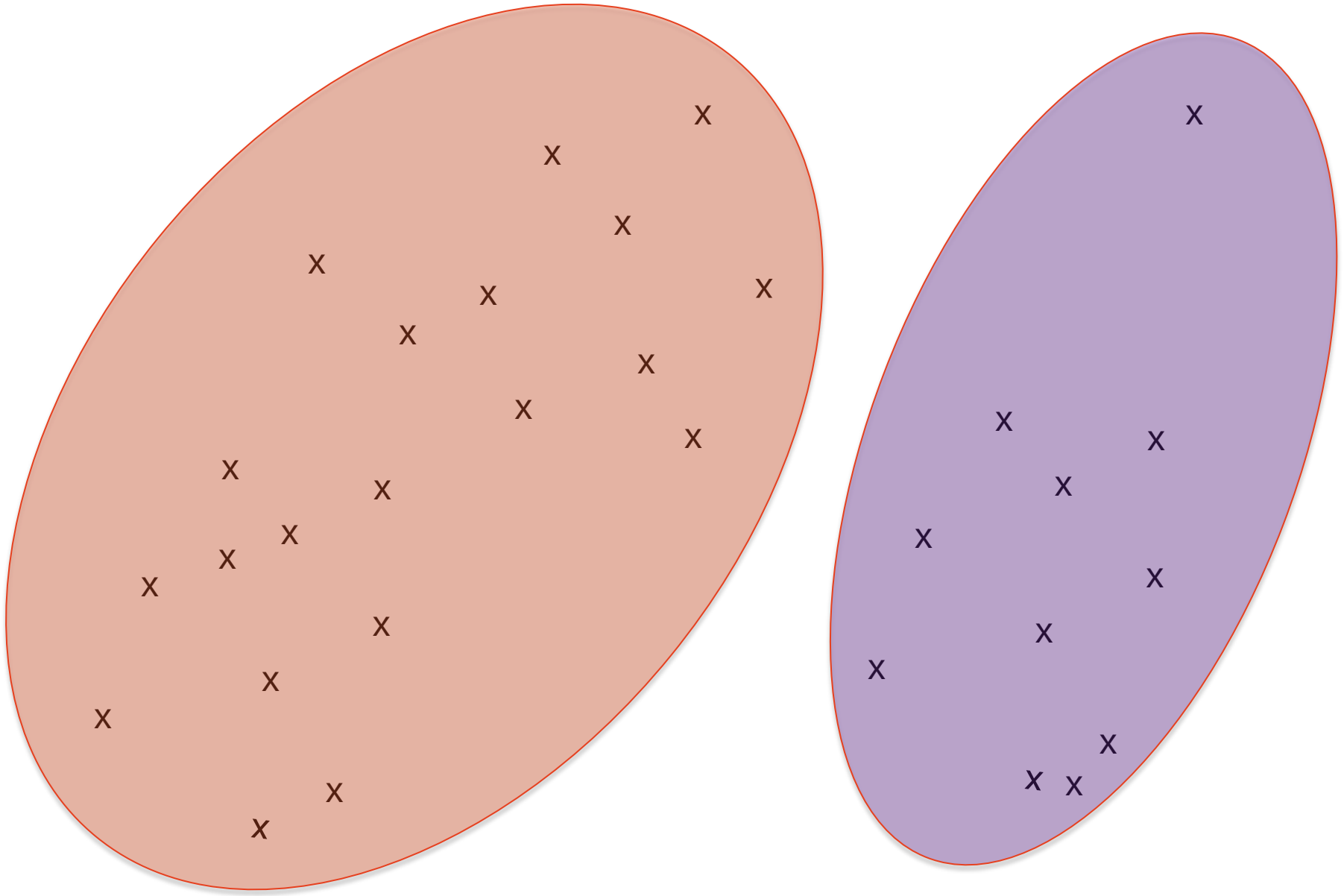
Clustering, conceptually



Clustering, conceptually



Clustering, conceptually



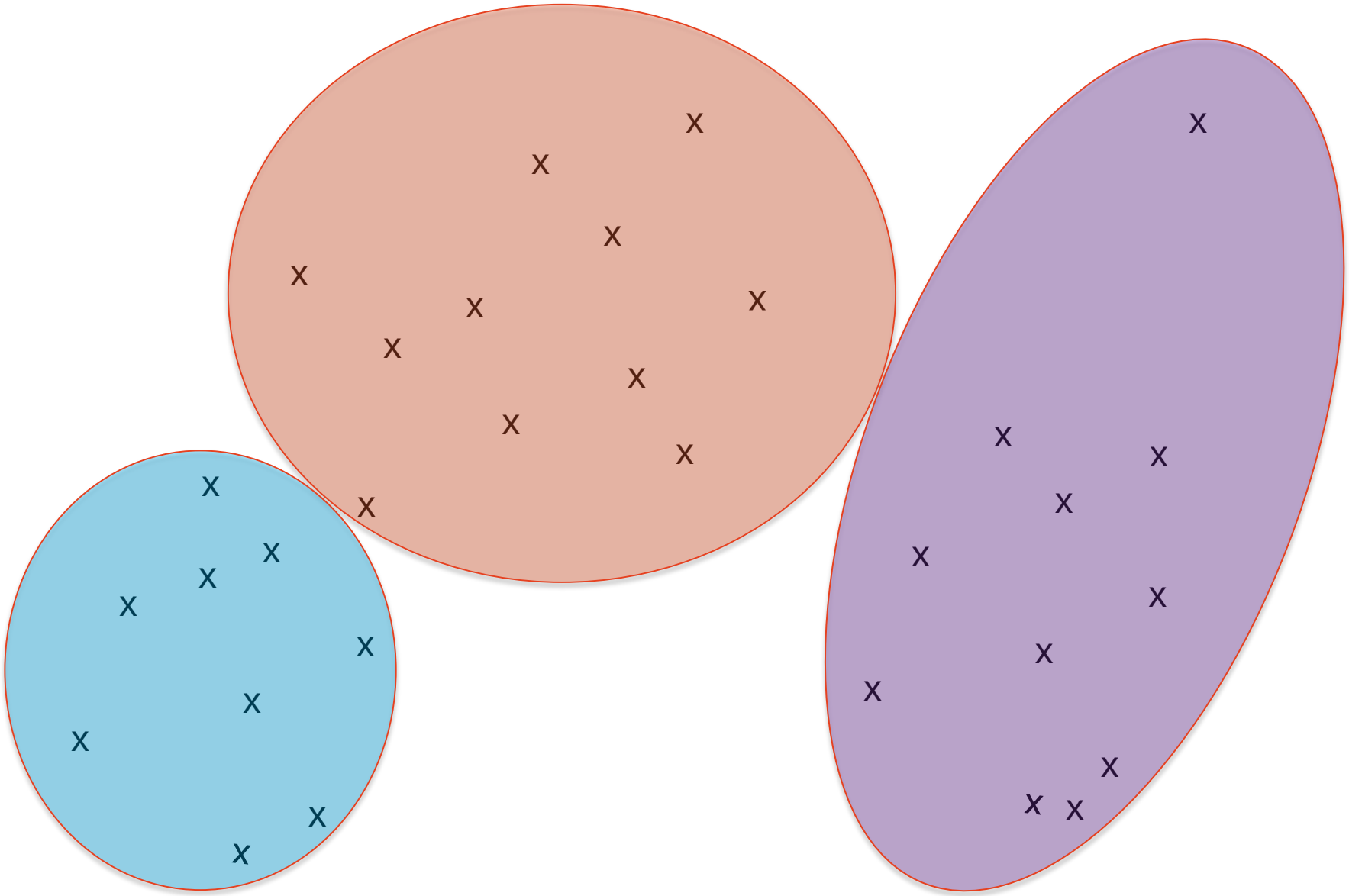
Clustering with K-Means Algorithm

1. Choose a number of clusters, *k*
   - this is the number of groups we are looking for.
2. Create *k* "pretend" data points as initial cluster centers
   - e.g. customers with random attributes
3. Iteratively move each cluster centre closer to the *centroid* of points it is closest to.
4. Stop when the points no longer need to move (they have reached a stable centroid). These are the centres of your clusters.
5. Assign all points (customers) to the group defined by their nearest cluster centre.

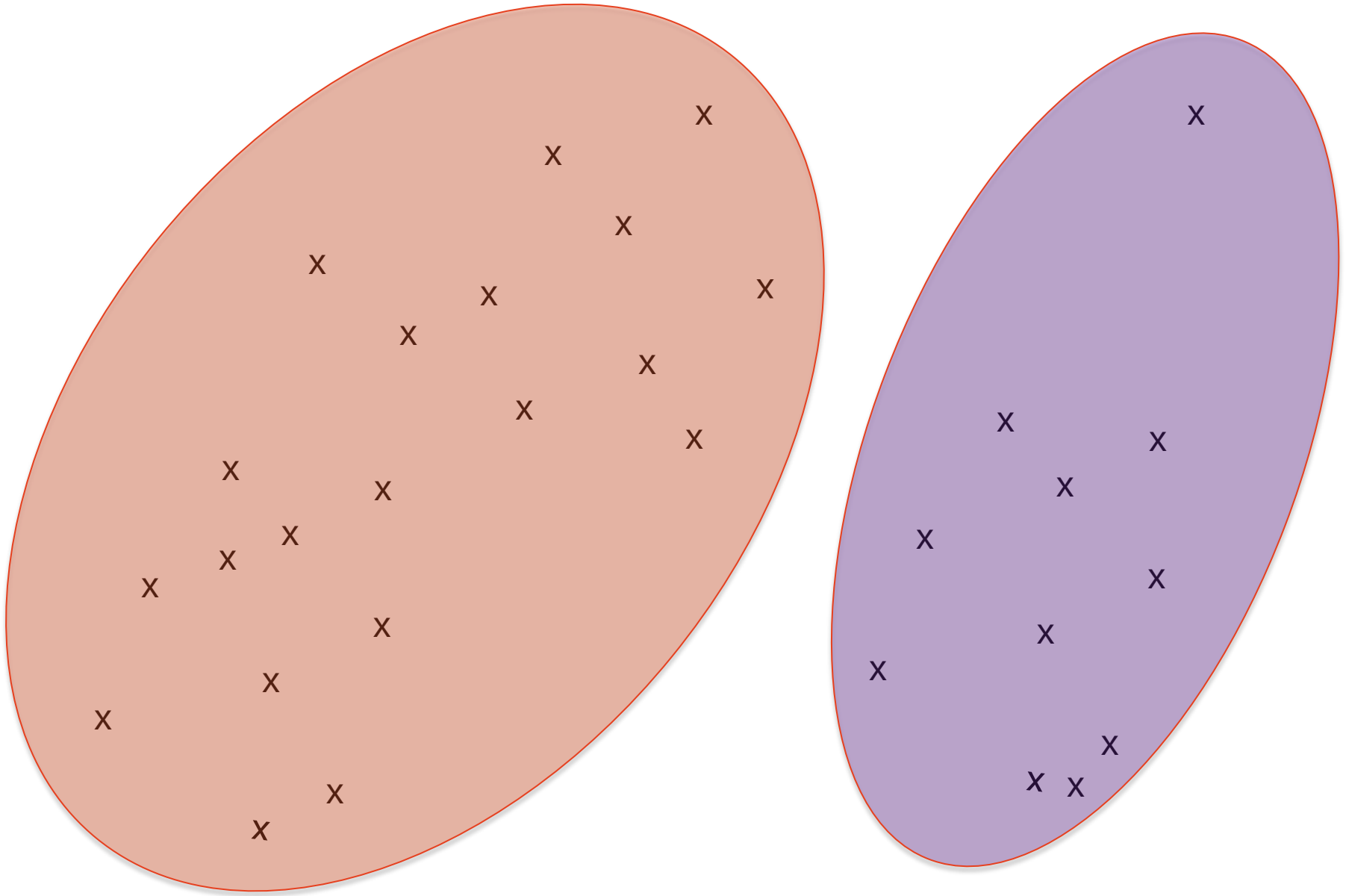
This is what k-means looks like

http://shabal.in/visuals/kmeans/2.html

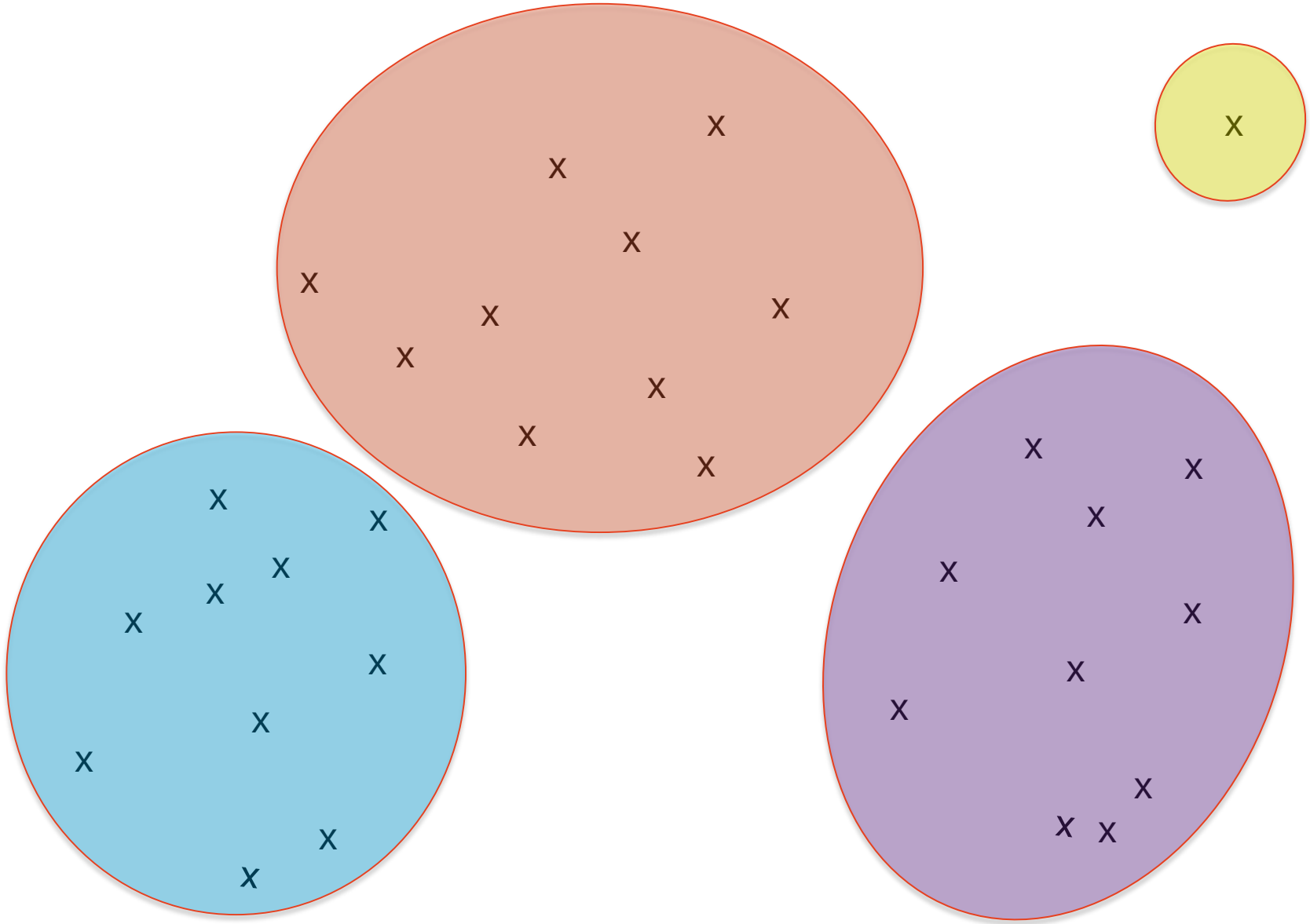
Choice of K matters



Choice of K matters



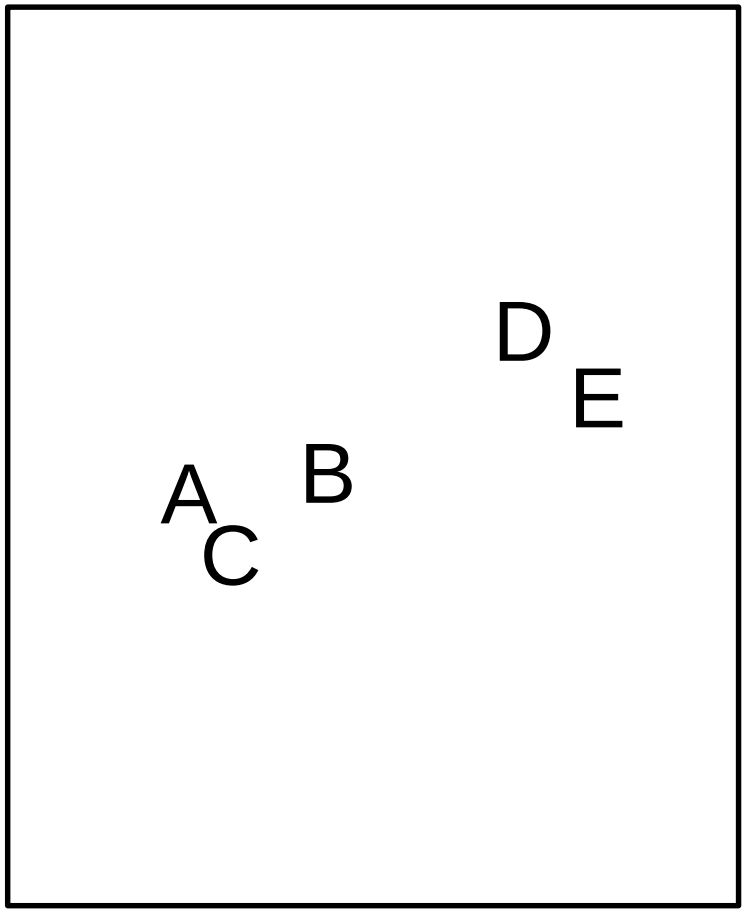
Choice of K matters



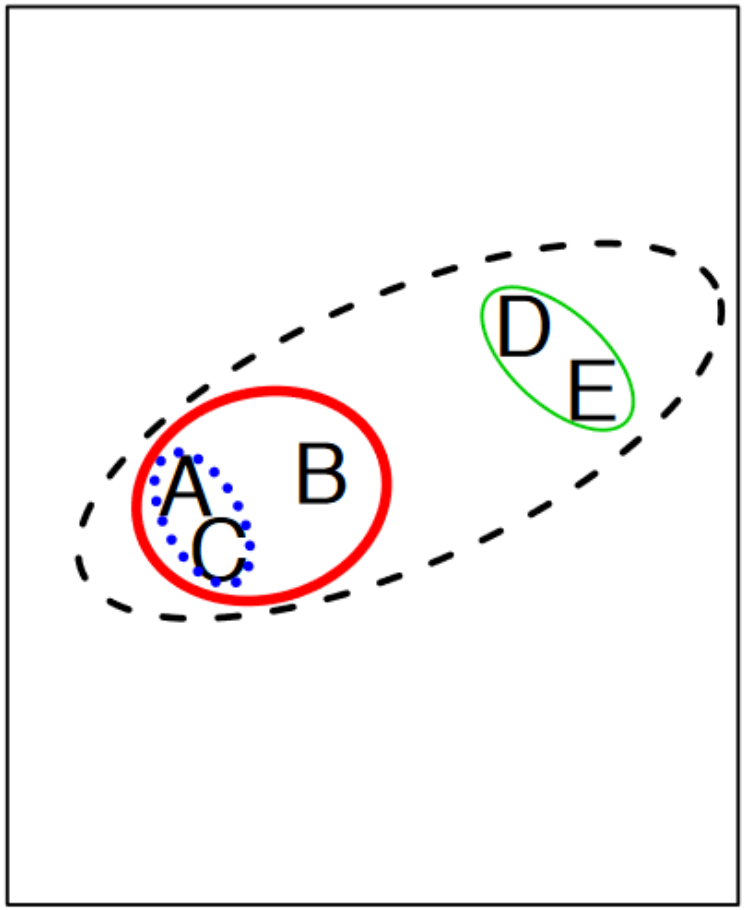
Determining the optimal number of Clusters

- Elbow method
  - Minimise total intra-cluster variation
- Average silhouette method
  - Closeness of points in one cluster to points in the neighbouring clusters
- Gap statistic
  - Compare intra-cluster variation with their expected values under a random uniform distribution

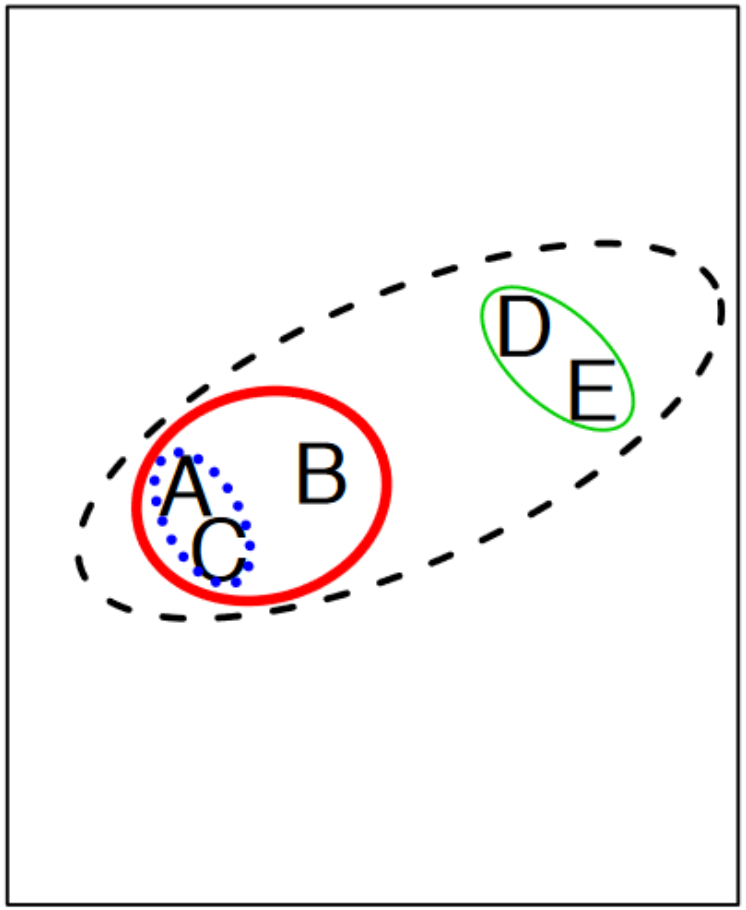
Hierarchical Clustering

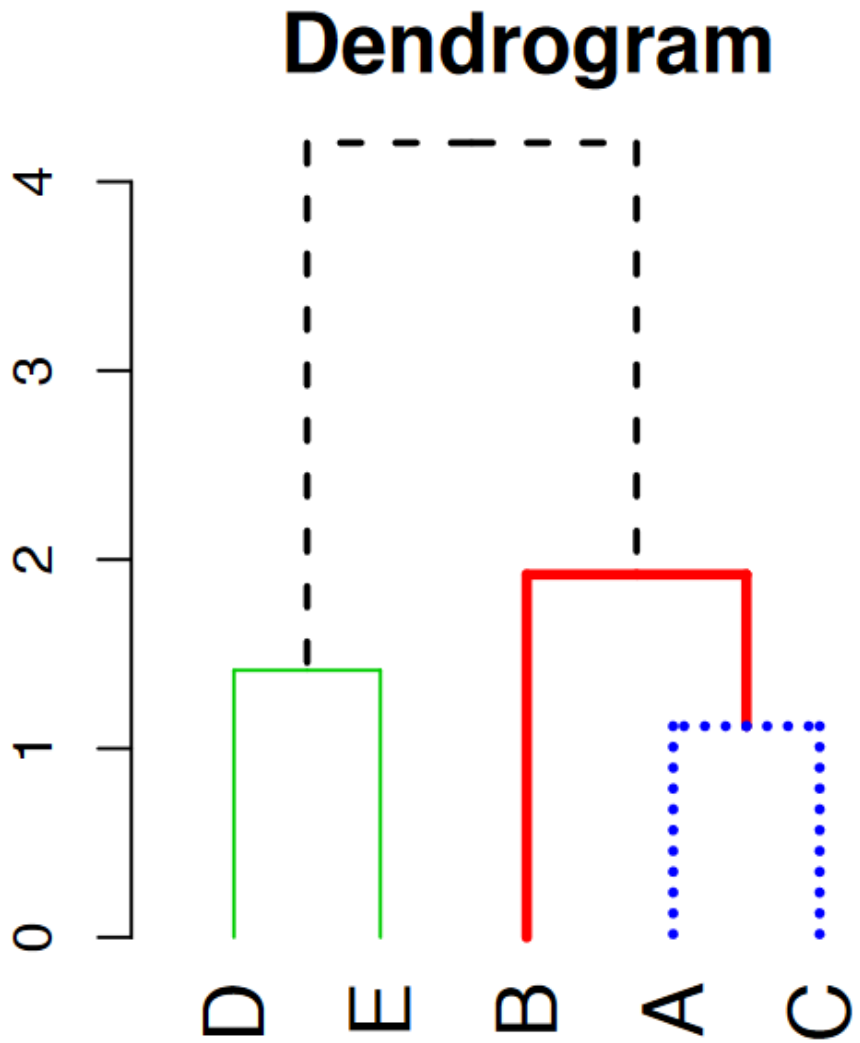


Hierarchical Clustering



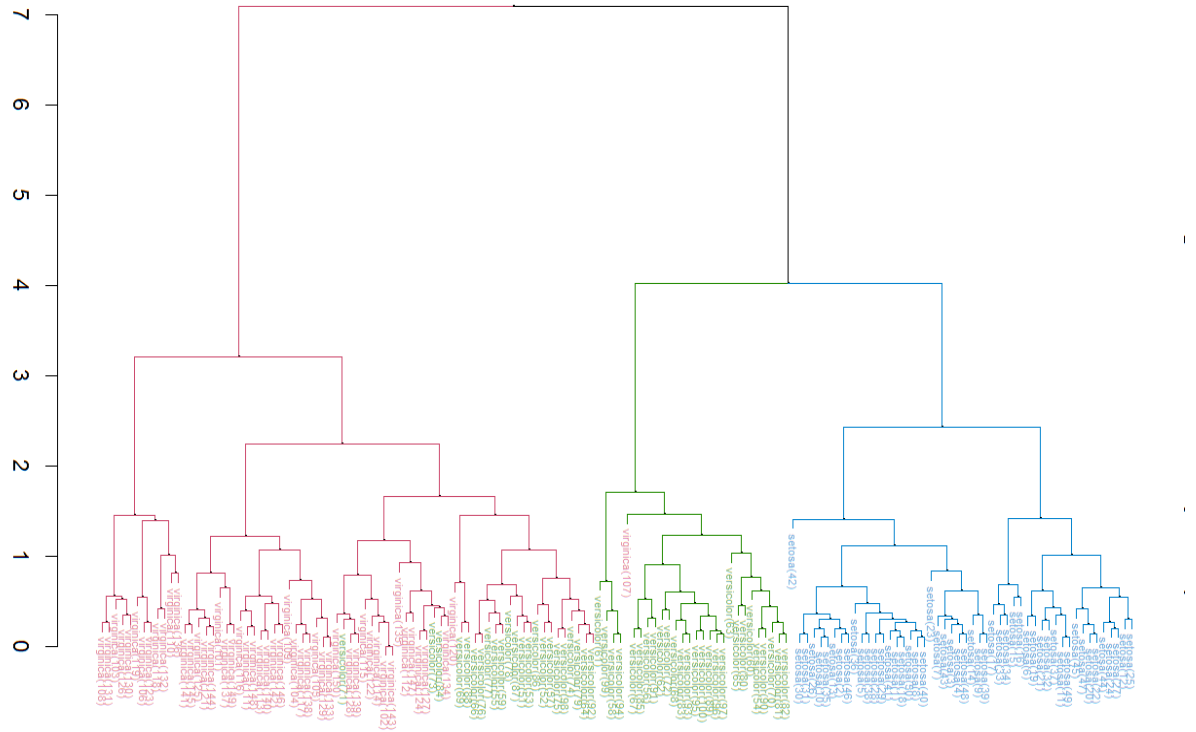
Hierarchical Clustering





Hierarchical clustering

- Calculate distances between observations.
- Successively merge or split observations depending on their similarity.



Clustering - technicalities

Distance Measures

- **Euclidean distance:** Root of the sum of squared distances in each dimension

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ir} - x_{jr})^2}$$

- **Manhattan (city block) distance:** Sum of absolute distances

$$dist(\mathbf{x}_i, \mathbf{x}_j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + ... + | x_{ir} - x_{jr} |$$

- **Squared Euclidean distance:** Progressively greater weight on data points that are further apart

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ir} - x_{jr})^2$$

- **Chebychev distance:** Define two points as "different" if they are different on any one of the attributes

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(| x_{i1} - x_{j1} |, | x_{i2} - x_{j2} |, ..., | x_{ir} - x_{jr} |)$$

Distance Measures

- **Euclidean distance:** Root of the sum of squared distances in eac

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} -$$

- **Manhattan (city block) distance:** Sum of absolute distances

$$dist(\mathbf{x}_i, \mathbf{x}_j) = | x_{i1} - x_{j1} | + | x_{i2} - x_j$$

- **Squared Euclidean distance:** Progressively greater weight o rt

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- **Chebychev distance:** Define two points as "different" if they are different on any one of the attributes

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(| x_{i1} - x_{j1} |, | x_{i2} - x_{j2} |, \ldots, | x_{ir} - x_{jr} |)$$

Special distance measures for binary and categorical measures

Common Linkage Methods (to calculate distance between clusters)

Linkage	Description
Complete	Maximal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities.
Single	Minimal inter-cluster dissimilarity. Compute all pair wise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities.
Average	Mean inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the average of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable inversions.

K-means vs Hierarchical clustering

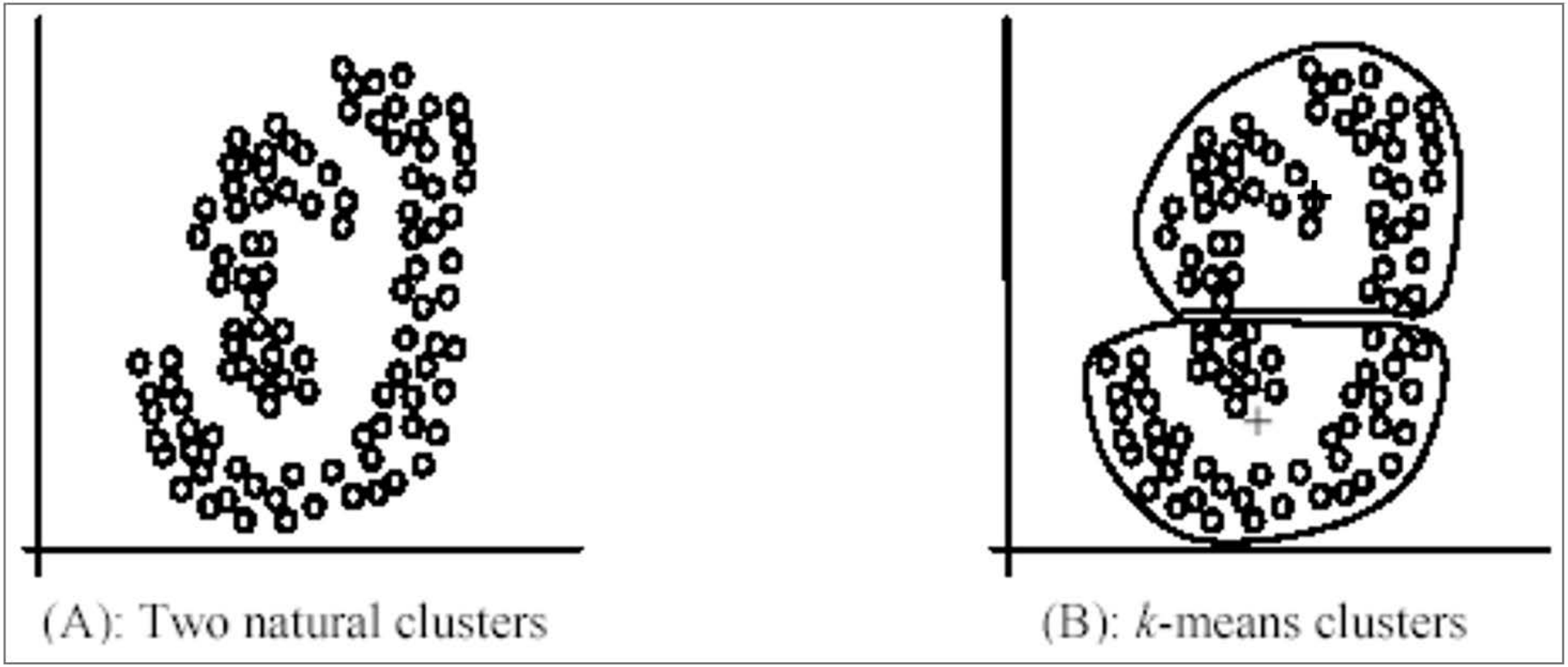
K-means

- Low memory usage
- Results are sensitive to random initialization and outliers
- Number of clusters needs to be pre-defined
- Awkward with categorical variables

Hierarchical clustering

- Deterministic algorithm
- Dendrogram shows clusterings for various choices of “K”
- Requires only a distance matrix, quantifying how dissimilar observations are from one another
- We can use a dissimilarity measures that gracefully handle categorical variables, missing values, etc.
- Memory-heavy, more computationally intensive than K-means

K-means has problems with clusters that are not spherical



source: CS583, Bing Liu, UIC

Cluster evaluation is a hard problem

- The quality of a clustering is very hard to evaluate because
  - We do not know the correct clusters

- Some methods are used:
  - **Internal cluster validation**, which uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information.
  - **External cluster validation**, compare the results of a cluster analysis to an externally known result, eg externally provided class labels. As we need know the "true" cluster membership in advance, this approach is mainly used for theoretical performance evaluation.
  - **Relative cluster validation**, which evaluates the clustering structure by varying different parameter values for the same algorithm. Useful eg for determining the optimal number of clusters.
  - **Indirect evaluation**, if clustering results are input to another method, evaluate and compare clustering options by performance of primary method.

Evaluation based on internal information

- **Cluster cohesion (compactness):**
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure to summarise and compare clusters.
- **Inter-cluster separation (isolation):**
  - Separation means that different clusters should be far away from one another.
    - distances between cluster centers
    - pairwise minimum distances between objects in different clusters
- **Connectivity:**
  - To what extent items are placed in the same cluster as their nearest neighbours in the data space? Connectivity has a value between 0 and infinity. It should be minimized

Common:

- **Silhouette coefficient** (average distance between clusters)
- **Dunn Index** (distance vs compactness)

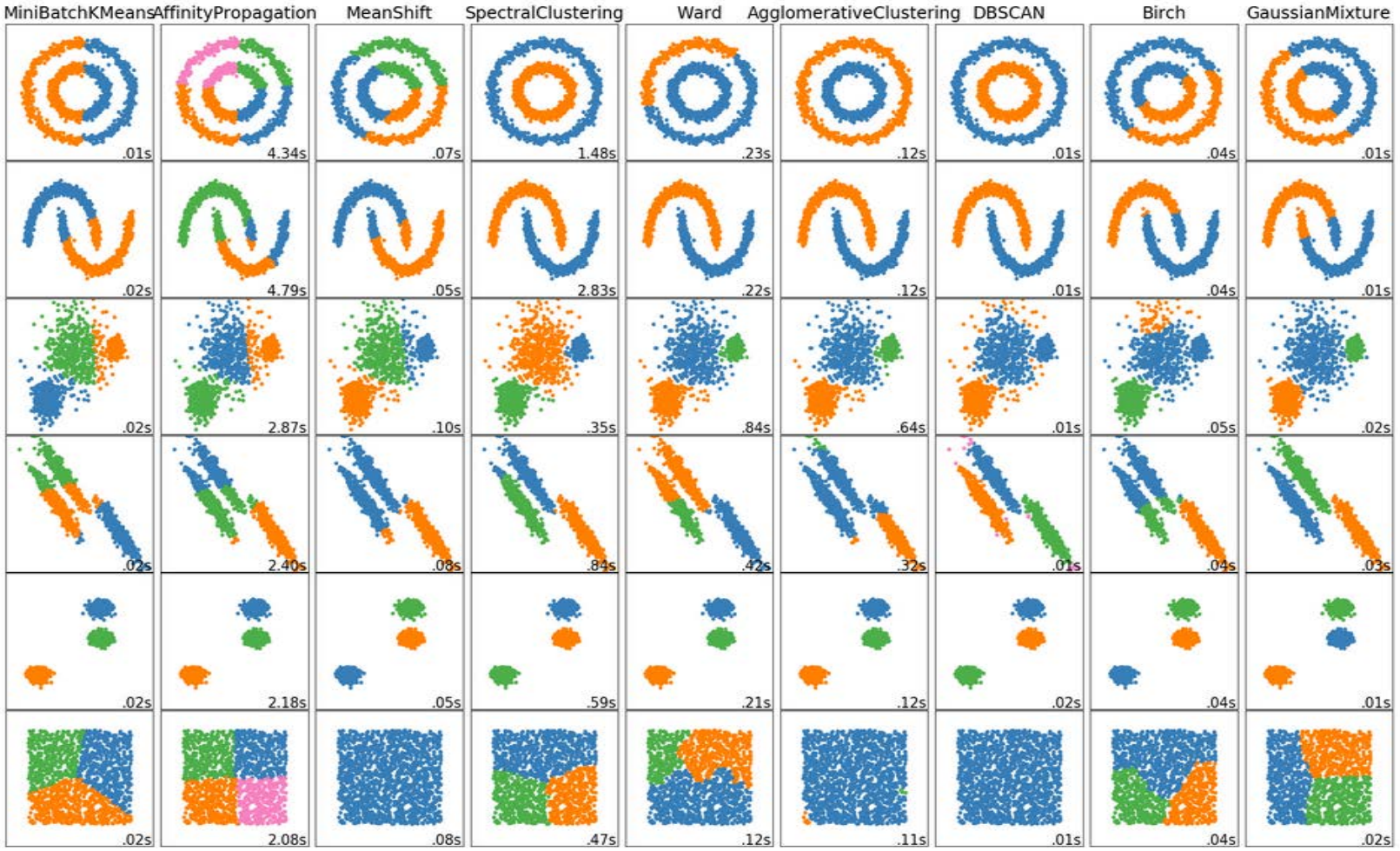
In most applications, expert judgments are still the key.

Overview of advanced clustering methods

Method name	Parameters	Scalability	Usecase
K-Means	number of clusters	Very large n_samples, medium n_clusters with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry
Mean-shift	bandwidth	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry
Spectral clustering	number of clusters	Medium n_samples, small n_clusters	Few clusters, even cluster size, non-flat geometry
Ward hierarchical clustering	number of clusters	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints
Agglomerative clustering	number of clusters, linkage type, distance	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, non Euclidean distances
DBSCAN	neighborhood size	Very large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation
Birch	branching factor, threshold, optional global clusterer.	Large n_clusters and n_samples	Large dataset, outlier removal, data reduction.

http://scikit-learn.org/stable/modules/clustering.html

Advanced clustering methods in

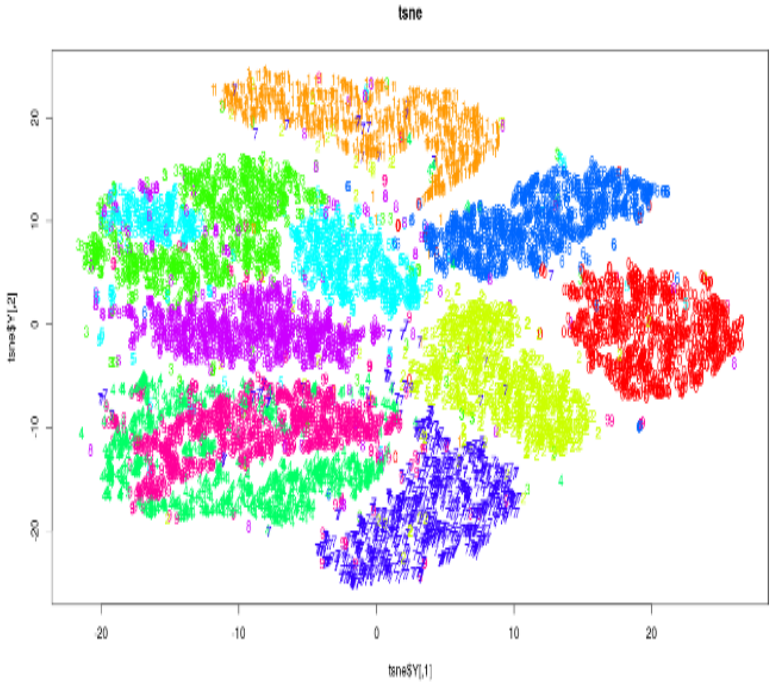


http://scikit-learn.org/stable/modules/clustering.html

Most advanced dimensionality reduction methods: t-distributed stochastic neighbour embedding (t-SNE)

Especially for visualising high-dimensional data in two or three dimensions

1. Measure the distances between points in N-dimensional space
2. Plot the points on (usually) a 2-dimensional graph, where the distances between the points are preserved.



Summary

- The current state of marketing, and why it needs help from data science
- *Unsupervised Learning*, where we use algorithms to explore for structure in the data.
- *Clustering*, a type of unsupervised learning where we create groups.
- The *k-means and hierarchical clustering* clustering as examples, as well as a little look at advanced methods.
- Evaluation is difficult, due to lack of ground truth.

Ask me 3 questions!

THE UNIVERSITY OF SYDNEY

Outlook

Lecture 11

Customer Analytics

- Product vs Customer Centric Marketing
- Measuring Success
- Machine Learning as a Process

Lecture 12

Natural Language Processing

Lecture 13

Case Study

Backup for ipynb

THE UNIVERSITY OF SYDNEY

Doing k-means
Revisit: Read csv data

```
# csv
df_directmarketing = (pd
    .read_csv('data/DirectMarketing.csv')
    .select(lambda c: '_' not in c, axis=1) # Keep all columns unless there is a '_' in the name
    .assign(customer_id=lambda d: range(9000, 9000+len(d))) # Invent a fake customer ID
)
df_directmarketing.head()
```

	Age	Gender	OwnHome	Married	Location	Salary	Children	History	Catalogs	AmountSpent	customer_id
0	Old	Female	Own	Single	Far	47500	0	High	6	755	9000
1	Middle	Male	Rent	Single	Close	63600	0	High	6	1318	9001
2	Young	Female	Rent	Single	Close	13500	0	Low	18	296	9002
3	Middle	Male	Own	Married	Close	85600	1	High	18	2436	9003
4	Middle	Female	Own	Single	Close	68400	0	High	12	1304	9004

Data preparation

```
# Do some more cleaning up of the data frame - tell it more about some of the categorical variables
df_directmarketing = (df_directmarketing
    .assign(
      Age=lambda d: pd.Categorical(d.Age, ordered=True, categories=['Young', 'Middle', 'Old']),
      History=lambda d: pd.Categorical(d.History, ordered=True, categories=['Low', 'Medium', 'High']),
      Gender=lambda d: pd.Categorical(d.Gender),
      OwnHome=lambda d: pd.Categorical(d.OwnHome),
      Married=lambda d: pd.Categorical(d.Married)
    ))
df_directmarketing.head(3)
```

	Age	Gender	OwnHome	Married	Location	Salary	Children	History	Catalogs	AmountSpent	customer_id
0	Old	Female	Own	Single	Far	47500	0	High	6	755	9000
1	Middle	Male	Rent	Single	Close	63600	0	High	6	1318	9001
2	Young	Female	Rent	Single	Close	13500	0	Low	18	296	9002

More Data Preparation

```
# Then turn categorical variables into "dummy" binary variables
df = pd.get_dummies(df_directmarketing).drop(['customer_id', 'AmountSpent'],axis=1)
df.head(3)
```

	Salary	Children	Catalogs	Age_Young	Age_Middle	Age_Old	Gender_Female	Gender_Male	OwnHome_Own	OwnHome_R
0	47500	0	6	0	0	1	1	0	1	0
1	63600	0	6	0	1	0	0	1	0	1
2	13500	0	18	1	0	0	1	0	0	1

```
# ... and normalise, so algorithms don't get dominated by differing scales of variables.
df = (df-df.mean())/df.std()
df.head(3)
```

	Salary	Children	Catalogs	Age_Young	Age_Middle	Age_Old	Gender_Female	Gender_Male	OwnHome_Own	Owr
0	-0.281023	-0.888618	-1.310907	-0.634131	-1.015622	1.968291	0.987577	-0.987577	0.968012	-0.9
1	0.244840	-0.888618	-1.310907	-0.634131	0.983634	-0.507547	-1.011567	1.011567	-1.032012	1.03
2	-1.391542	-0.888618	0.500989	1.575384	-1.015622	-0.507547	0.987577	-0.987577	-1.032012	1.03

Doing k-means
Run algorithm

Run the k-means algorithm to find cluster centres, then assign each data point to a cluster.

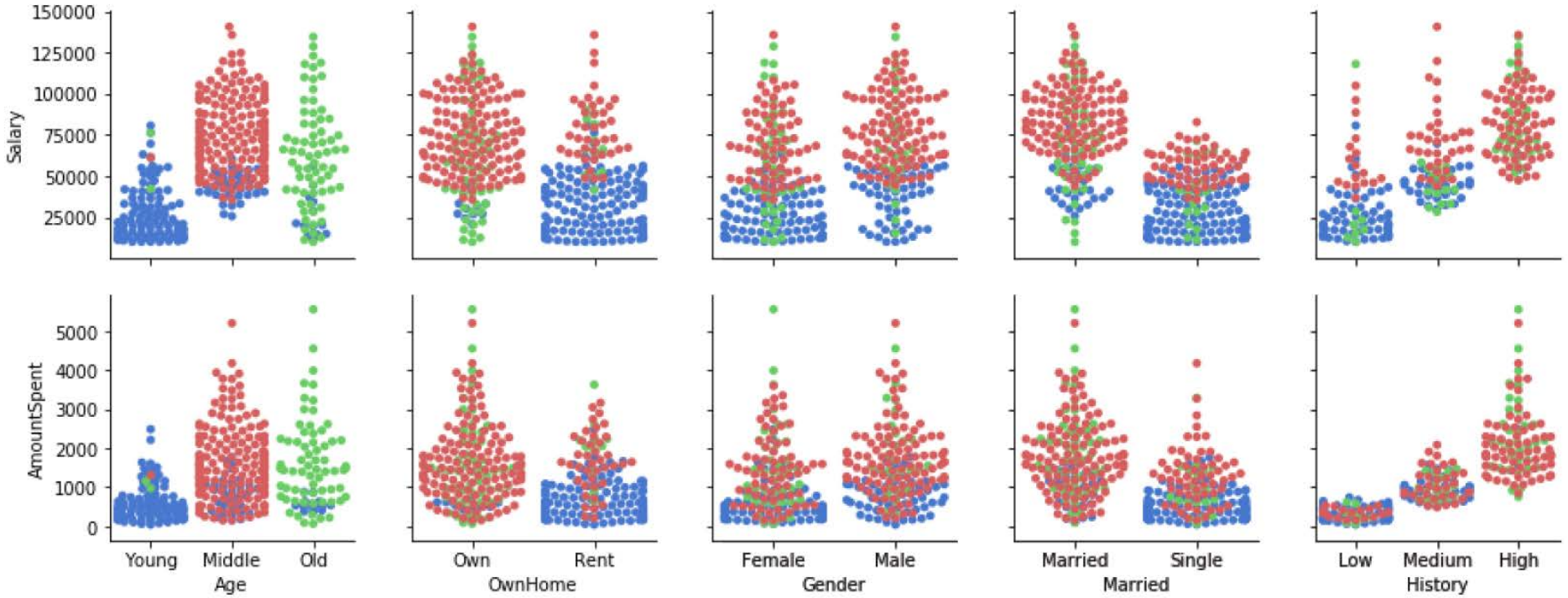
```
import sklearn.cluster

km = sklearn.cluster.KMeans(n_clusters=3, random_state=0)


clusters = km.fit_predict(df)

df2 = df_directmarketing.assign(
    Cluster=pd.Categorical(clusters))

display(df2.head(10))
```

Cluster assignment appended to original data

	Age	Gender	OwnHome	Married	Location	Salary	Children	History	Catalogs	AmountSpent	customer_id	Cluster
0	Old	Female	Own	Single	Far	47500	0	High	6	755	9000	1
1	Middle	Male	Rent	Single	Close	63600	0	High	6	1318	9001	2
2	Young	Female	Rent	Single	Close	13500	0	Low	18	296	9002	0
3	Middle	Male	Own	Married	Close	85600	1	High	18	2436	9003	2
4	Middle	Female	Own	Single	Close	68400	0	High	12	1304	9004	2
5	Young	Male	Own	Married	Close	30400	0	Low	6	495	9005	0
6	Middle	Female	Rent	Single	Close	48100	0	Medium	12	782	9006	0
7	Middle	Male	Own	Single	Close	68400	0	High	18	1155	9007	2
8	Middle	Female	Own	Married	Close	51900	3	Low	6	158	9008	2
9	Old	Male	Own	Married	Far	80700	0	NaN	18	3034	9009	1

Complex relations of cluster results to other variables

```
figure(figsize=(12,20))
g = sns.PairGrid(
    x_vars=['Age', 'OwnHome', 'Gender', 'Married', 'History'], y_vars=['Salary','AmountSpent'],
    hue='Cluster', data=df2.sample(400), palette='muted')
g.map(sns.swarmplot);
```



Hierarchical clusters by correlation

```
correlation_table = df.corr()
correlation_table
```

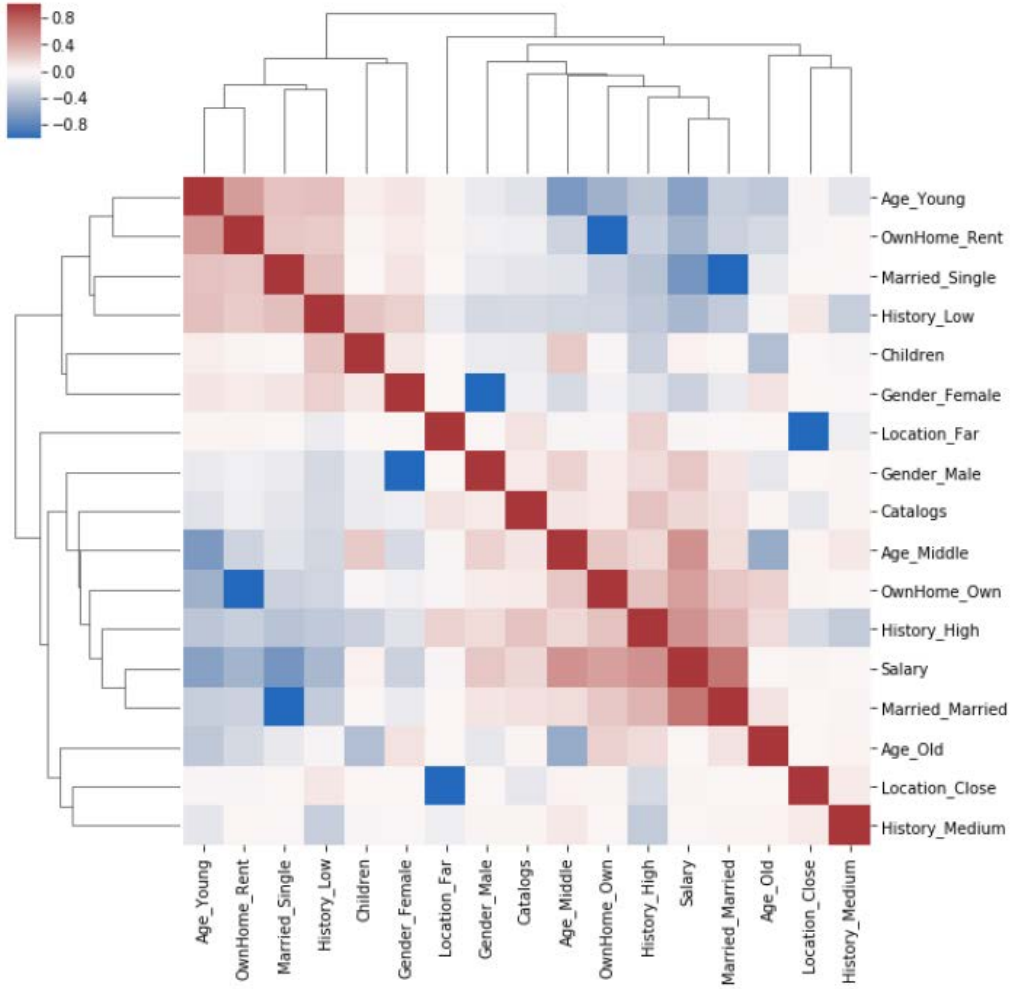
	Salary	Children	Catalogs	Age_Young	Age_Middle	Age_Old	Gender_Female	Gender_Male	OwnHome_Own	OwnHome_Re
Salary	1.000000	0.049663	0.183551	-0.588571	0.529052	0.004347	-0.261492	0.261492	0.460736	-0.460736
Children	0.049663	1.000000	-0.113455	0.073527	0.244720	-0.385446	0.105469	-0.105469	-0.032274	0.032274
Catalogs	0.183551	-0.113455	1.000000	-0.158872	0.114083	0.036743	-0.087351	0.087351	0.093132	-0.093132
Age_Young	-0.588571	0.073527	-0.158872	1.000000	-0.644682	-0.322173	0.113979	-0.113979	-0.469299	0.469299
Age_Middle	0.529052	0.244720	0.114083	-0.644682	1.000000	-0.515992	-0.204233	0.204233	0.251649	-0.251649
Age_Old	0.004347	-0.385446	0.036743	-0.322173	-0.515992	1.000000	0.125200	-0.125200	0.214228	-0.214228
Gender_Female	-0.261492	0.105469	-0.087351	0.113979	-0.204233	0.125200	1.000000	-1.000000	-0.084433	0.084433
Gender_Male	0.261492	-0.105469	0.087351	-0.113979	0.204233	-0.125200	-1.000000	1.000000	0.084433	-0.084433
OwnHome_Own	0.460736	-0.032274	0.093132	-0.469299	0.251649	0.214228	-0.084433	0.084433	1.000000	-1.000000
OwnHome_Rent	-0.460736	0.032274	-0.093132	0.469299	-0.251649	-0.214228	0.084433	-0.084433	-1.000000	1.000000
Married_Married	0.675633	0.009770	0.137060	-0.283289	0.155957	0.124301	-0.116057	0.116057	0.264009	-0.264009
Married_Single	-0.675633	-0.009770	-0.137060	0.283289	-0.155957	-0.124301	0.116057	-0.116057	-0.264009	0.264009
Location_Close	0.037127	-0.002391	-0.128581	-0.032982	0.041084	-0.013920	-0.005554	0.005554	0.033691	-0.033691
Location_Far	-0.037127	0.002391	0.128581	0.032982	-0.041084	0.013920	0.005554	-0.005554	-0.033691	0.033691
History_Low	-0.425830	0.271837	-0.204205	0.299367	-0.227388	-0.053858	0.212071	-0.212071	-0.240979	0.240979
History_Medium	-0.012371	-0.041939	0.016047	-0.139779	0.094472	0.039636	-0.025799	0.025799	-0.001919	0.001919
History_High	0.524690	-0.273362	0.284932	-0.335680	0.171912	0.163249	-0.160751	0.160751	0.277386	-0.277386

Show clusters in correlation matrix

```
sns.clustermap(
    correlation_table,
    cmap='vlag');
```

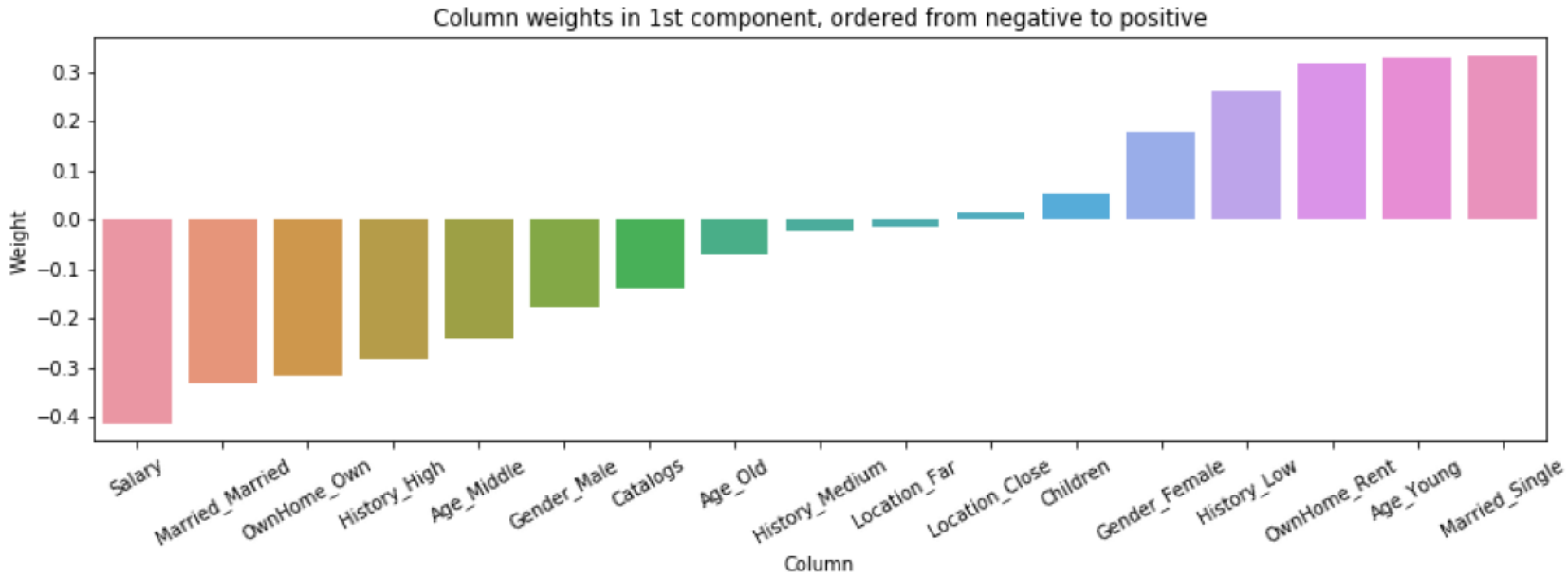
Blocks of red on the diagonal show variables that are correlated with each other.

The trees show the result of the hierarchical clustering.



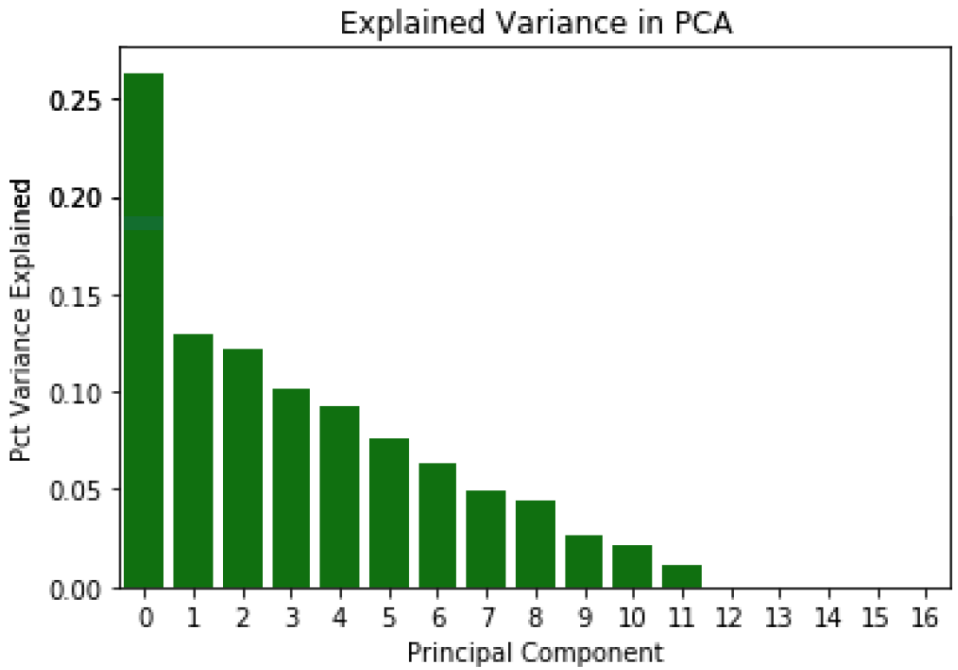
What is in this first principal component?

```
from sklearn.decomposition import PCA
pca = PCA()
df_transformed = pd.DataFrame(pca.fit_transform(df))
first_component_weights = pd.DataFrame({'Weight': pca.components_[0], 'Column': df.columns})
first_component_weights = first_component_weights.sort_values('Weight')
figure(figsize=(14,4))
sns.barplot(x='Column', y='Weight', data=first_component_weights)
xticks(rotation=30)
title('Column weights in 1st component, ordered from negative to positive');
```



How much variance is explained by each component i.e. show normalised eigenvalues

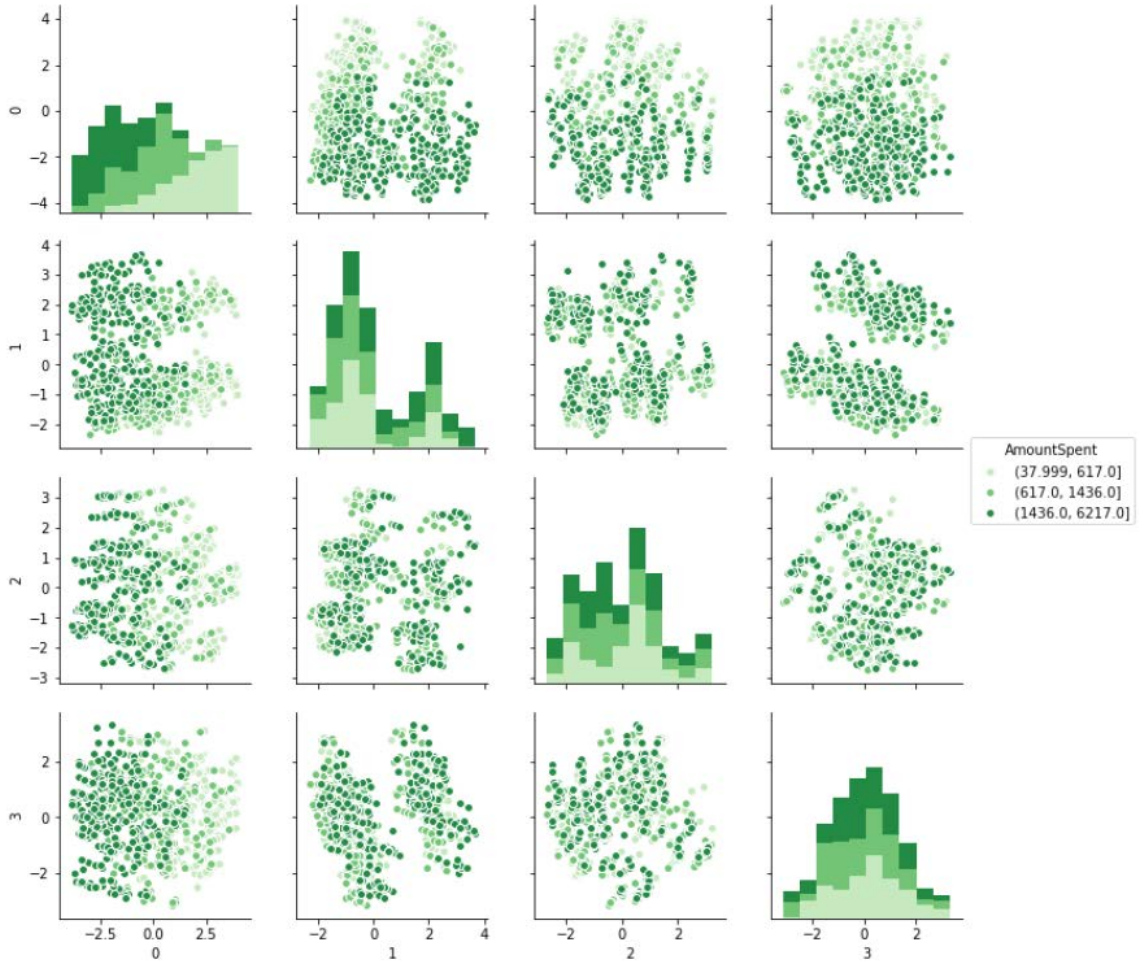
```
x = np.arange(df_transformed.shape[1])
sns.barplot(x=x, y=pca.explained_variance_ratio_, color='Green')
title('Explained Variance in PCA')
ylabel('Pct Variance Explained')
xlabel('Principal Component');
```



Further analysis of the PCA result

Plot each component against another component, and colour by the amount spent to show how they relate.

- *NB: Note that we didn't teach the model anything about "Amount Spent". it just so happens that is the natural grouping.*



Code: Further analysis of PCA result

```
df_transformed['AmountSpent'] =
   pd.qcut(df_directmarketing['AmountSpent'], 3)

sns.pairplot(df_transformed.loc[:,
   (0,1,2,3,'AmountSpent')],
   hue='AmountSpent',
   palette='Greens');
```