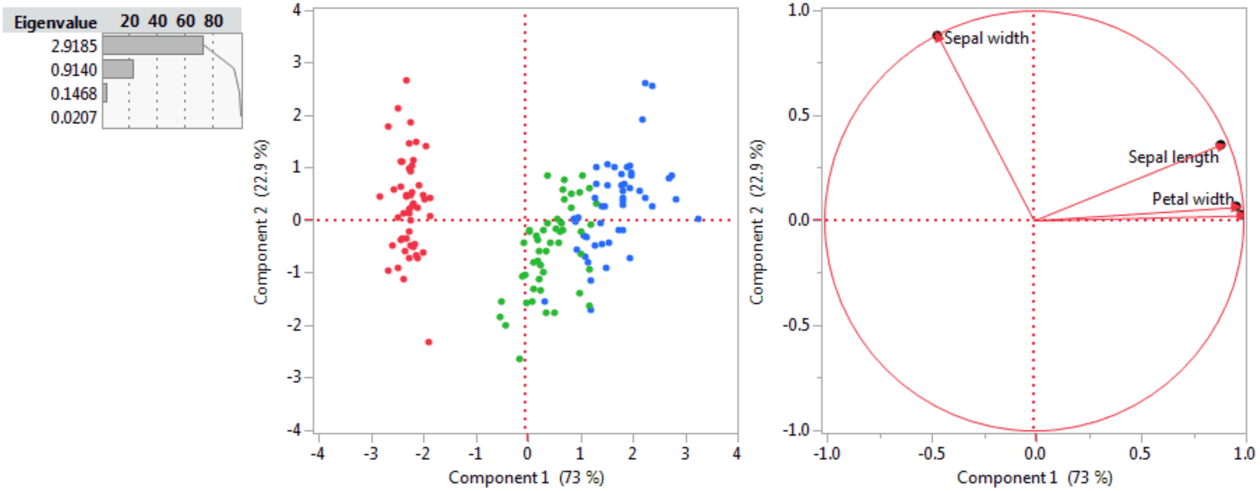
Master of Technology in Intelligent Systems

Problem Solving using Pattern Recognition

Module 3: Dimensionality Reduction



Charles Pang
Institute of Systems Science
National University of Singapore
Email: charlespang@nus.edu.sg

Objectives and Topics

Learning Objectives:

- Gain a practical understanding of dimension reduction in solving complex problems with a large number of features
- Gain skills and knowledge in Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA)

Topics:

- Introduction to Dimensionality Reduction
- Dimensionality and computational complexity
- The curse of dimensionality
- Methods to deal with high dimensionality
- Principal Component Analysis
- Linear Discriminant Analysis
- Programming Workshop

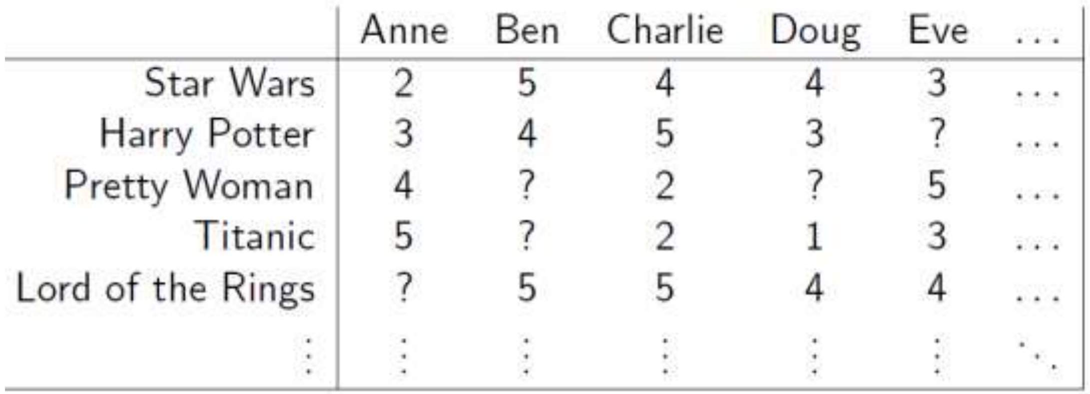
Learning Outcome

At the end of this module, you will have:

- An appreciation of feature reduction as a means to reduce complexities in solving business problems that involves high-dimensionality data
- Working knowledge of Principal Component Analysis
- Working knowledge of Linear Discriminant Analysis
- The skills and knowledge to assess the appropriateness and effectiveness of these techniques through the proper interpretation of the results and the analysis of the output
- The practical ability to use python to code and implement these techniques

What is Dimensionality?

- The **dimensionality** of a data set is the number of attributes or features that the data possesses.
- For example:
  - Healthcare data: Height, Weight, Blood Pressure, Blood Glucose, Cholesterol
  - Image data: 8x8 image = 64 dimensions
  - Netflix rating: Customer ratings (480k) on every film (18k)
- Data with a small number of features are called **low dimensional data**
- Data with a large number of features are called **high dimensional data**



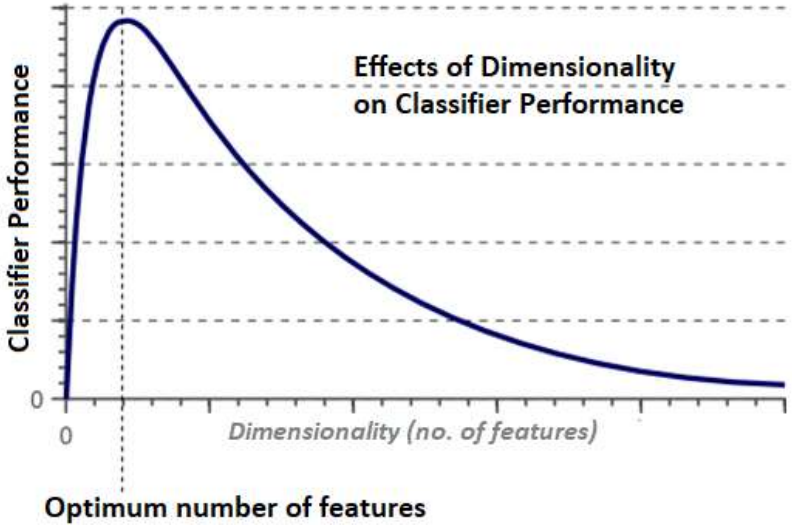
	Anne	Ben	Charlie	Doug	Eve	...
Star Wars	2	5	4	4	3	...
Harry Potter	3	4	5	3	?	...
Pretty Woman	4	?	2	?	5	...
Titanic	5	?	2	1	3	...
Lord of the Rings	?	5	5	4	4	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

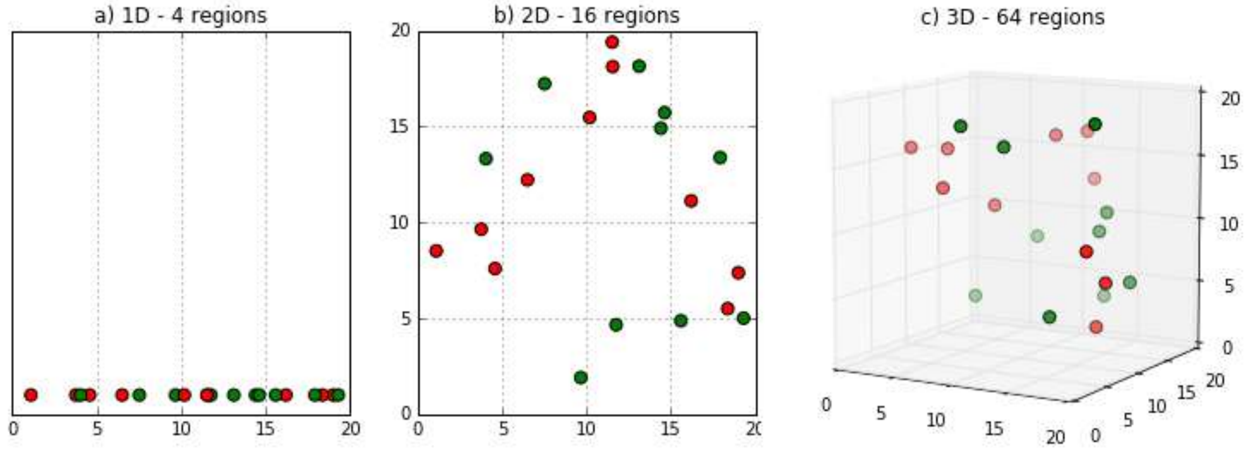
High Dimensionality and computational complexity

- High dimensional data with a staggeringly high number of dimensions will increase the computational **complexity of calculations** for machine learning algorithms.
- As you add more features, the **number of samples increases**. This is because the more features we have, the more samples we will need to so that we can have all combinations of features-values well represented in our dataset.
- Problems start to occur with larger datasets: bigger **storage spaces**, high **computational loads**, complex analytical **model**, and model **overfitting**.
- All these problems will result in **slower learning cycles**, and **poor prediction performance** on real data.

The Curse of Dimensionality

- The curse of dimensionality is a term introduced by Richard E. Bellman (1957) to describe the problem caused by the number of samples needed to estimate an arbitrary function with a given level of accuracy.
- As the number of features or dimensions grows, the amount of data you need to generalize accurately **grows exponentially**.
- This exponential growth in data causes **high sparsity** in the data set and unnecessarily increases storage space and processing time for the particular modelling algorithm.





www.kdnuggets.com

Lower Dimensionality is Desirable

- Many machine learning **algorithms work better** with a smaller set of features versus a larger set of features.
- When irrelevant or redundant features are eliminated, **noise is reduced** to yield more accurate and efficient models
- Fewer features will result in a more **understandable model**
- **Visualisation** is more possible with lower dimensional data
- The overall problem complexity is hopefully reduced, and hence **time and effort** spent by data scientists becomes more productive.

Methods to deal with high dimensionality

- In machine learning problems, the data should possess a certain number of features for the algorithm to learn useful patterns. However, we have also seen that having too many features can cause problems.
- Dimensionality reduction is usually a means to an end, although it can already be useful to understand underlying patterns and trends.
- We will look at two ways to deal with high dimensionality:
  - Feature Subset Selection (feature elimination)
  - Dimensionality Reduction techniques (Principal Component Analysis, Linear Discriminant Analysis)

Feature Subset Selection

In feature subset selection, our objective is to try and **eliminate**:

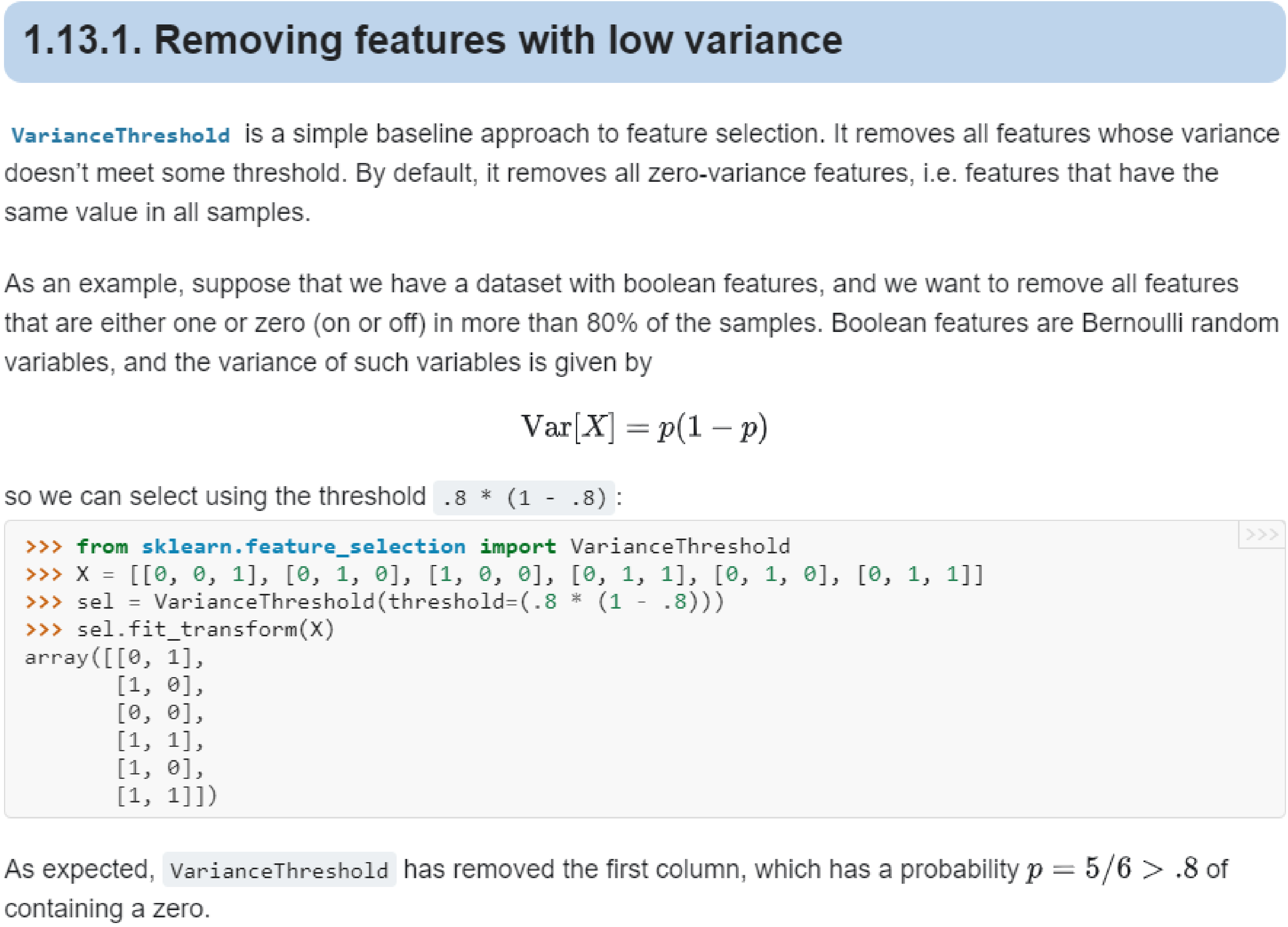
- **Redundant** features- e.g. GST and the purchase price are both are essentially the same piece of information.
- **Irrelevant** features – e.g. CustomerID or some unique identifier. You will not find any insights using these features.
- **Low variance** features – e.g. values are “closely apart” - they behave like constants and will not affect the model.
- **High correlation** features – e.g. consumer spending & GDP, age and income are likely to be correlated.
- Features with too many of **Missing Values** – they are unlikely to carry useful information

What is Variance?

- Variance is a measurement of the spread between numbers in a data set. It measures how far each number in the set is from the mean
- Technically, it measures the *average of the squared deviation from the mean.*
- A variance value of zero indicates that all values within a set of numbers are identical; all variances that are non-zero will be positive numbers.
- A large variance indicates that numbers in the set are far from the mean and each other, while a small variance indicates the opposite.
- We can analyse the variance by taking using either Variance (σ^2) or the Standard Deviation (σ)

$$\text{variance } \sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$$

How to use the Variance?



1.13.1. Removing features with low variance

`VarianceThreshold` is a simple baseline approach to feature selection. It removes all features whose variance doesn’t meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.

As an example, suppose that we have a dataset with boolean features, and we want to remove all features that are either one or zero (on or off) in more than 80% of the samples. Boolean features are Bernoulli random variables, and the variance of such variables is given by

$$\mathrm{Var}[X] = p(1 - p)$$

so we can select using the threshold `.8 * (1 - .8)`:

```
>>> from sklearn.feature_selection import VarianceThreshold
>>> X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
>>> sel = VarianceThreshold(threshold=(.8 * (1 - .8)))
>>> sel.fit_transform(X)
array([[0, 1],
       [1, 0],
       [0, 0],
       [1, 1],
       [1, 0],
       [1, 1]])
```

As expected, `VarianceThreshold` has removed the first column, which has a probability $p = 5/6 > .8$ of containing a zero.

What is Correlation?

- Correlation is a measure of the extent to which **two variables are related** or associated.
- There are **three possible results** of a correlation analysis:
  - A **positive correlation** is a relationship between two variables in which both variables either increase or decease at the same time. An example would be height and weight. Taller people tend to be heavier.
  - A **negative correlation** is a relationship between two variables in which an increase in one variable is associated with a decrease in the other. An example would be the time you spent playing games versus your test scores (some assumptions here!).
  - A **zero correlation** exists when there is no relationship between two variables. For example their is no relationship between number of cockroaches in your home and your salary amount

Using Correlation to help reduce dimensions

- When a pair of variables are highly correlated it maybe possible to remove one to reduce dimensionality without much loss of information.
- Which one should we keep? Usually the one with a higher correlation to the target.
- It may also be possible to keep a combination of both variables.
- We can analyse correlation graphically, or use the Pearson Correlation Coefficient

Pearson Correlation Coefficient

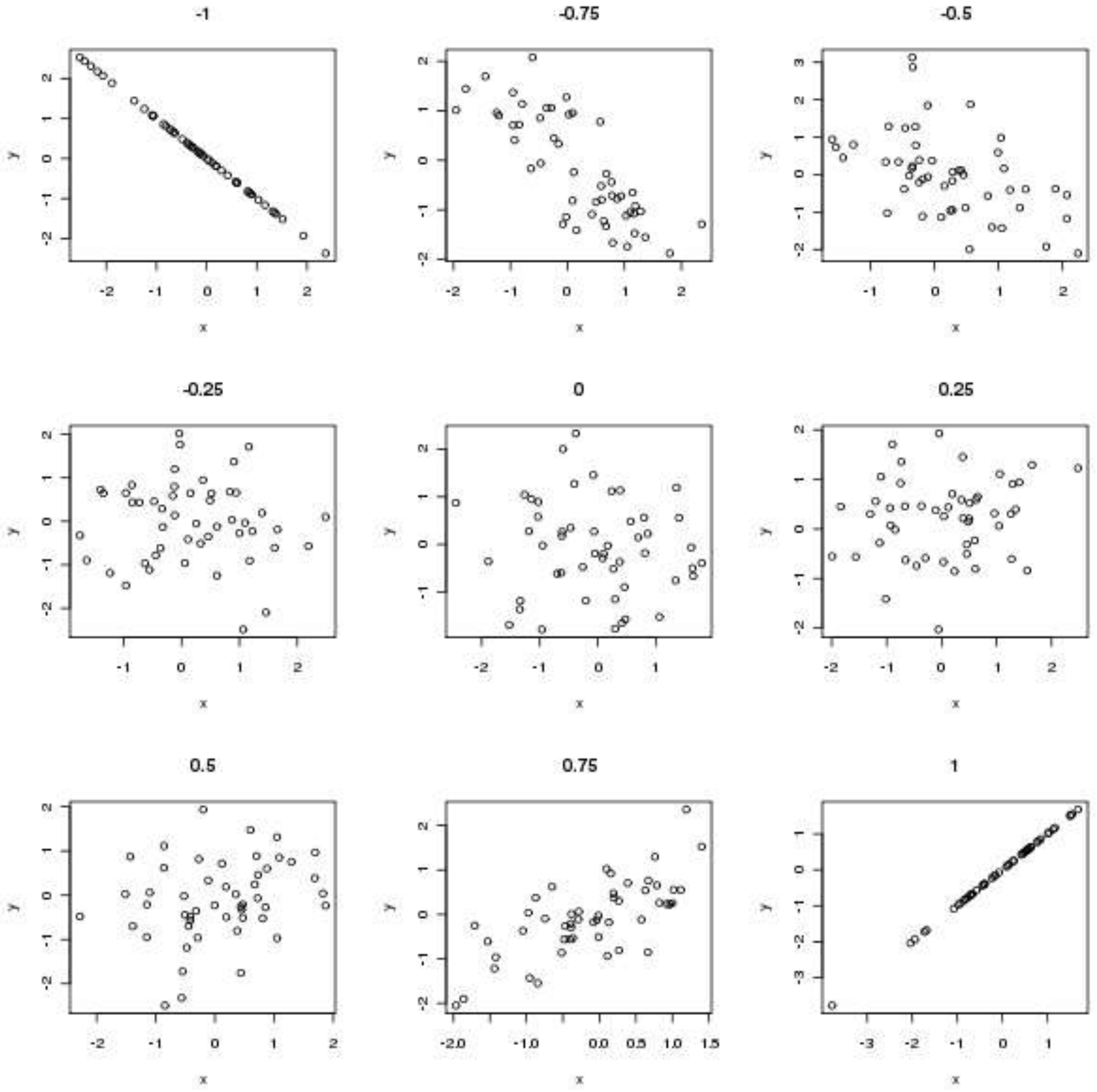
Given two data sets [y_1 , y_2 ,...y_n] and [x_1 , x_2 ,.....x_n]

The *Pearson correlation coefficient* is defined by:

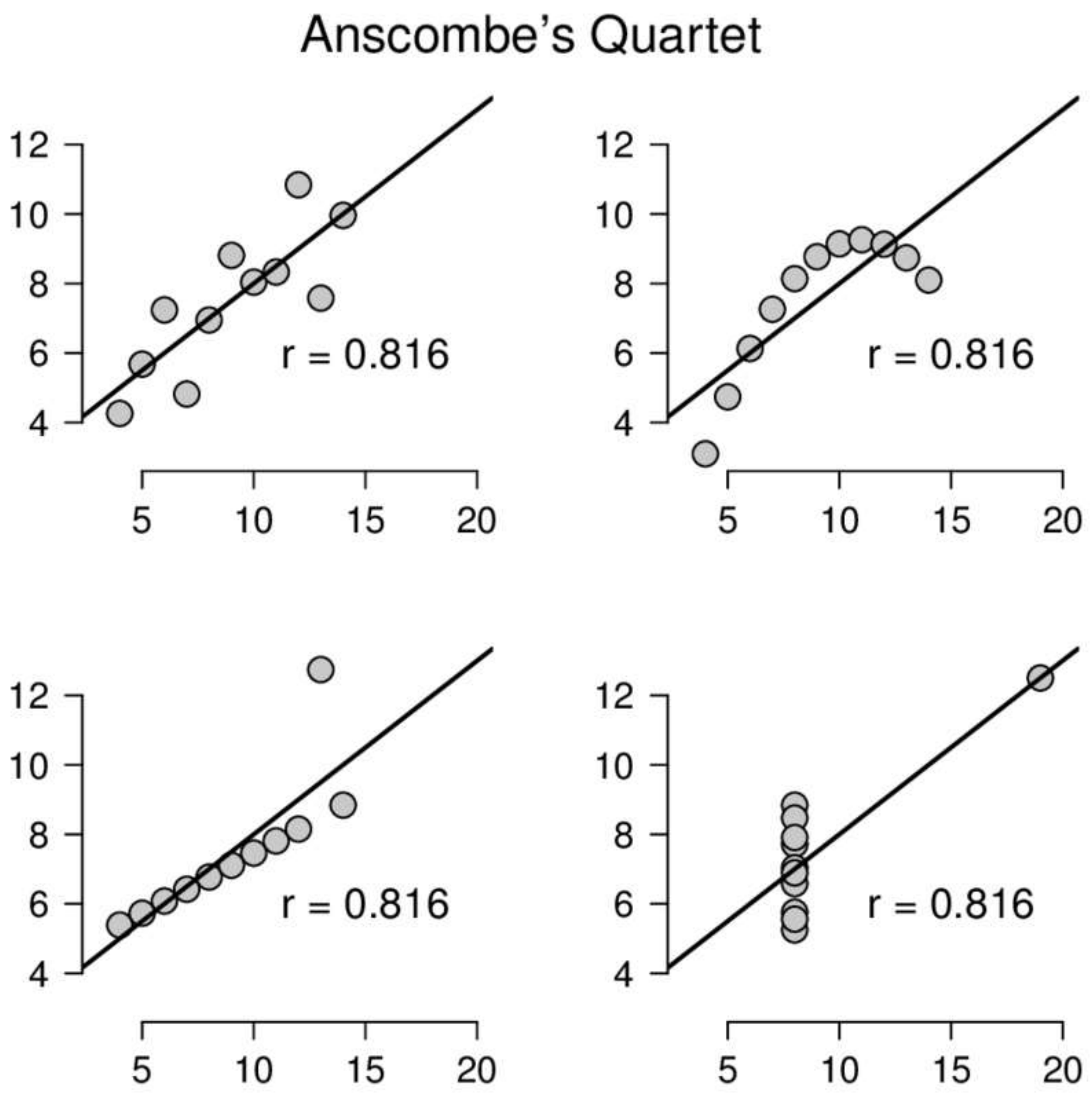
$$r = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right) . \left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}}$$

- If ρ_{xy} = 0 then no linear correlation between x & y
- If ρ_{xy} = +1 or -1 then perfect correlation
- If ρ_{xy} > 0.8 or < - 0.8 then significant correlation

Examples of Correlations



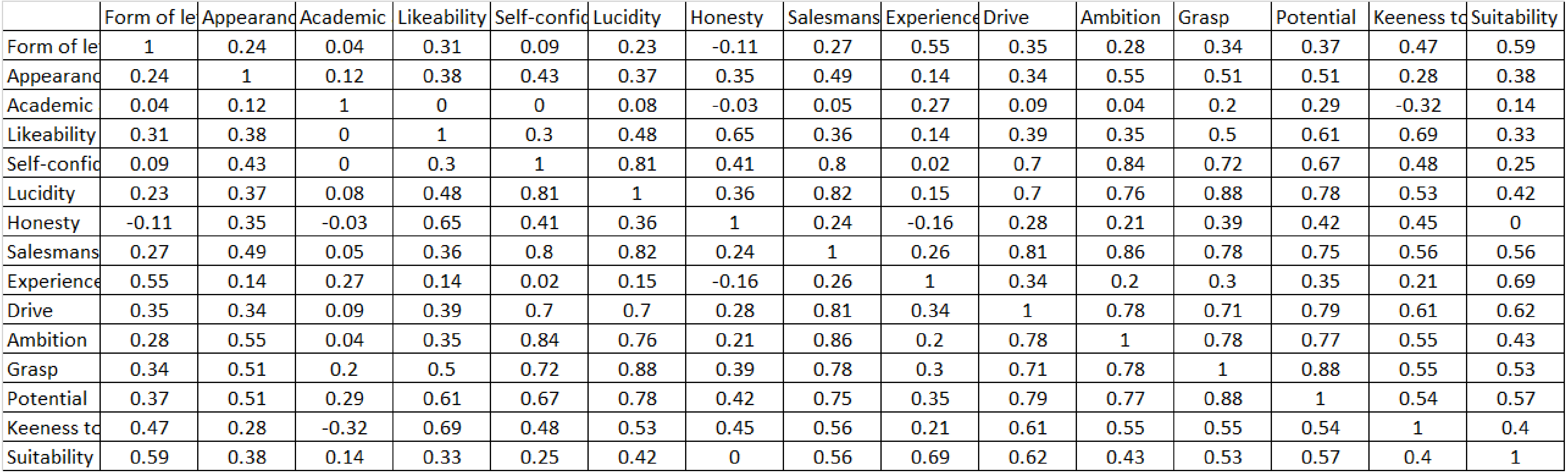
Anscombe's Quartet Correlation



High correlation doesn't necessarily mean that underlying relationship is linear

How to use Correlation?

```
corm = data.corr()
print('Corelation Matrix:')
print(pd.DataFrame(np.round(corm,decimals=2),index=colnames,columns=colnames),'\n')
```



	Form of le	Appearanc	Academic	Likeability	Self-confic	Lucidity	Honesty	Salesmans	Experience	Drive	Ambition	Grasp	Potential	Keeness to	Suitability
Form of le	1	0.24	0.04	0.31	0.09	0.23	-0.11	0.27	0.55	0.35	0.28	0.34	0.37	0.47	0.59
Appearanc	0.24	1	0.12	0.38	0.43	0.37	0.35	0.49	0.14	0.34	0.55	0.51	0.51	0.28	0.38
Academic	0.04	0.12	1	0	0	0.08	-0.03	0.05	0.27	0.09	0.04	0.2	0.29	-0.32	0.14
Likeability	0.31	0.38	0	1	0.3	0.48	0.65	0.36	0.14	0.39	0.35	0.5	0.61	0.69	0.33
Self-confic	0.09	0.43	0	0.3	1	0.81	0.41	0.8	0.02	0.7	0.84	0.72	0.67	0.48	0.25
Lucidity	0.23	0.37	0.08	0.48	0.81	1	0.36	0.82	0.15	0.7	0.76	0.88	0.78	0.53	0.42
Honesty	-0.11	0.35	-0.03	0.65	0.41	0.36	1	0.24	-0.16	0.28	0.21	0.39	0.42	0.45	0
Salesmans	0.27	0.49	0.05	0.36	0.8	0.82	0.24	1	0.26	0.81	0.86	0.78	0.75	0.56	0.56
Experience	0.55	0.14	0.27	0.14	0.02	0.15	-0.16	0.26	1	0.34	0.2	0.3	0.35	0.21	0.69
Drive	0.35	0.34	0.09	0.39	0.7	0.7	0.28	0.81	0.34	1	0.78	0.71	0.79	0.61	0.62
Ambition	0.28	0.55	0.04	0.35	0.84	0.76	0.21	0.86	0.2	0.78	1	0.78	0.77	0.55	0.43
Grasp	0.34	0.51	0.2	0.5	0.72	0.88	0.39	0.78	0.3	0.71	0.78	1	0.88	0.55	0.53
Potential	0.37	0.51	0.29	0.61	0.67	0.78	0.42	0.75	0.35	0.79	0.77	0.88	1	0.54	0.57
Keeness to	0.47	0.28	-0.32	0.69	0.48	0.53	0.45	0.56	0.21	0.61	0.55	0.55	0.54	1	0.4
Suitability	0.59	0.38	0.14	0.33	0.25	0.42	0	0.56	0.69	0.62	0.43	0.53	0.57	0.4	1

Things to note about Correlation

- The Pearson correlation coefficient indicates the strength of a linear relationship between two variables, but its value generally does not completely characterize their relationship
- Correlations are useful because they **can indicate a predictive relationship** that can be exploited in practice
- Correlation analysis can be used to ***help statistically confirm*** our intuitions regarding associations between features
- Some relationships can be strong but not significant; Conversely, relationship may be weak but is significant.
- Correlation does not imply causation

Feature Subset Selection (cont'd)

- The advantages of feature elimination methods include **simplicity and maintaining interpretability** of your variables.
- The use of **domain knowledge** means that you are keeping only the features that are **intuitively relevant** to solving your problem.
- Algorithms such as **Decision Tree** or **Cluster Analysis** can help detect unimportant features that can be ignored.
- On the flipped side, you may **loose the opportunity** of gaining some beneficial information from those features that you have eliminated.
- You may have to experiment.

Dimensionality Reduction Techniques

- Dimensionality reduction is a **feature extraction** method as opposed to feature elimination that we have seen earlier.
- In feature extraction, all original variables are extracted to form **new independent variables**.
- Suppose you have 50 features in your dataset. Feature extraction will create (as many as 50) new independent variables where each new variable is a **combination of each of the original features**.
- You can choose to **keep only the more important** (new) variables and ignore the least important ones.
- Dimensionality Reduction techniques include Principal Component Analysis, Linear Discriminant Analysis, Factor Analysis, Singular Value Decomposition.

PRINCIPAL COMPONENT ANALYSIS

Background of Principal Component Analysis

- Principal Component Analysis was originally proposed by Karl Pearson, the British Statistician in 1901. It is a **mathematical technique** –a vector space transform- that has its roots in linear algebra and in statistics.
- Its main purpose is to reduce a **correlated multidimensional** data set to an **uncorrelated lower dimensional** space with maximum variance.
- PCA concepts can be challenging for non-mathematical oriented students, since the statistical definitions (i.e., variance-covariance, correlation) need to be connected to matrix algebra (eigenvectors of a variance-covariance matrix) and to graphical vector representation (including matrix rotation).

What is Principal Component Analysis?

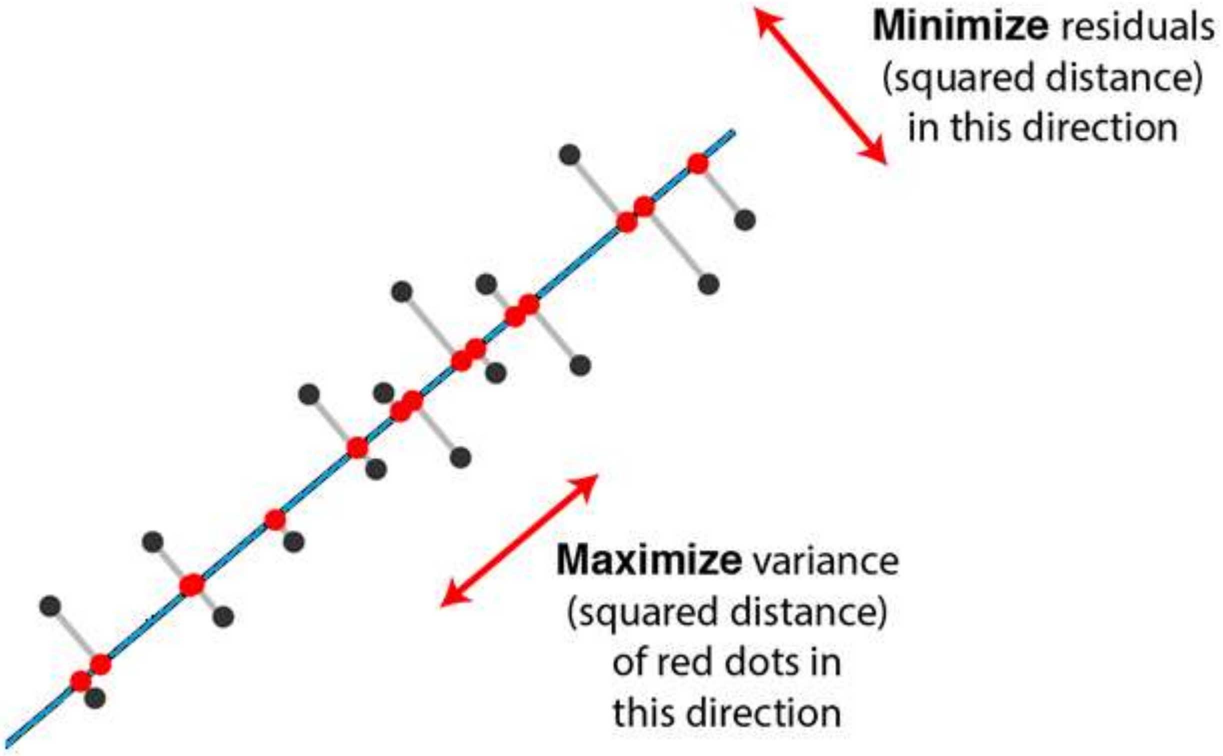
- It is an **unsupervised learning** method that seeks to summarize the content hidden in high dimensional datasets, reducing the information to a smaller set of “summary indices” (i.e. Principal Components) that can be more **easily visualised and analysed**.
- PCA is based on linear correlations. It examines the **correlations among the features** and uses this information to construct the appropriate principal components.
- The goal of PCA is to extract the smallest number of components which **account for as much as possible of the information** of the original features.

What is Principal Component Analysis? (Cont’d)

- The components are all **independent** of one another. This is advantageous if they are used for down stream analysis such as Linear Regression, Clustering, etc. where variable independence is assumed.
- In summary, the properties of Principal Components are:
  - are **linear combinations** of the original features,
  - are **orthogonal** (perpendicular) to each other, and
  - capture the maximum amount of **variance** in the data.

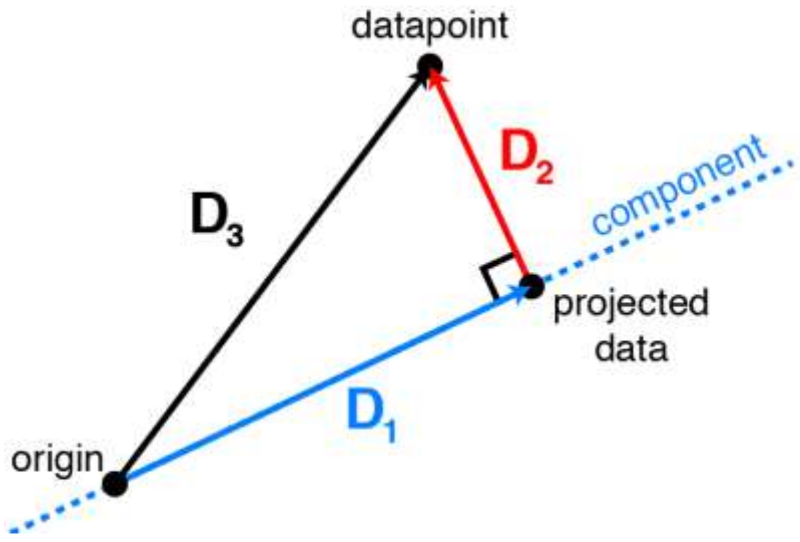
Principal Component Analysis Method

- PCA tries to find a new axis that captures the maximum variance within the data once it is projected onto the new axis:



Minimize residuals (squared distance) in this direction

Maximize variance (squared distance) of red dots in this direction



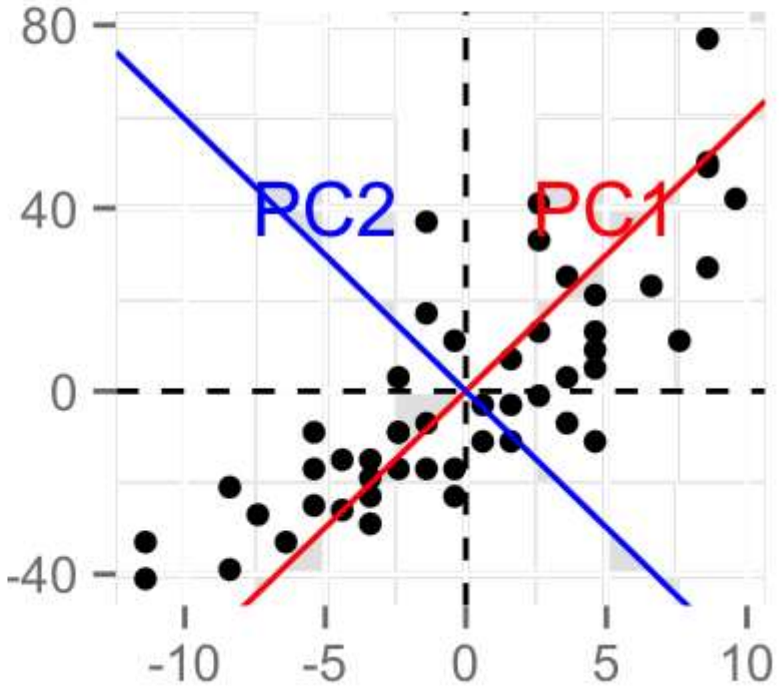
$$D_3^2 = D_1^2 + D_2^2$$

D_3^2 this is constant

D_1^2 maximize this **or** D_2^2 minimize this

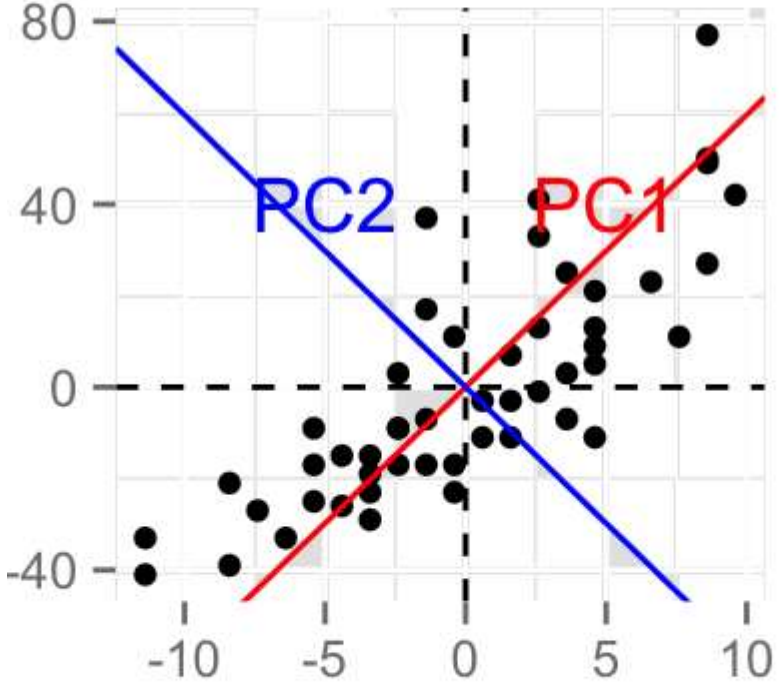
Principal Component Analysis Method

- The new axes are called **principal components** and the values of the new variables are called principal components **score**
- Each principal component is a **linear combination** of the original variables
- The 1st PC accounts for the maximum variance in the data
- The 2nd PC accounts for the maximum variance that has not been accounted by the 1st variable
- The pth PC accounts for the maximum variance that has not been accounted by the p-1 variables



Principal Component Analysis Method

- The Principal Components themselves are considered as **new variables**, and they are all **uncorrelated**
- If a substantial amount of the total variance in the data is accounted for by a few principal components then one can use these fewer no. of variables (subset) for further investigation instead of original features.
- This subset of variables can be identified by looking at the **eigen** values



PCA Example 1 – Construct a metric

Objective : Construct a measure of CPI (Consumer Price Index) for food items

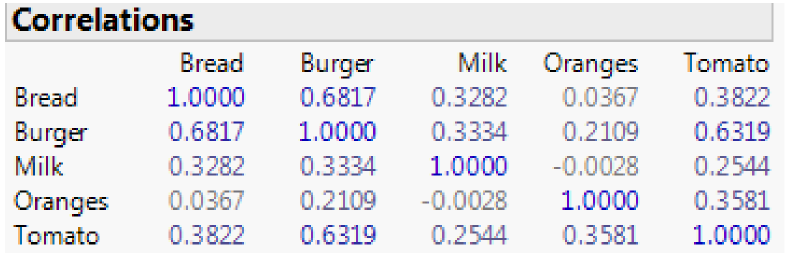
Food Price Data

City	Bread	Burger	Milk	Oranges	Tomato
Atlanta	24.5	94.5	73.9	80.1	41.6
Baltimore	26.5	91	67.5	74.6	53.3
Boston	29.7	100.8	61.4	104	59.6
Buffalo	22.8	86.6	65.3	118.4	51.2
Chicago	26.7	86.7	62.7	105.9	51.2
Cincinnati	25.3	102.5	63.3	99.3	45.6
Cleveland	22.8	88.8	52.4	110.9	46.8
Dallas	23.3	85.5	62.5	117.9	41.8
Detroit	24.1	93.7	51.5	109.7	52.4
Honolulu	29.3	105.9	80.2	133.2	61.7
Houston	22.3	83.6	67.8	108.6	42.4
Kansas City	26.1	88.9	65.4	100.9	43.2
Los Angeles	26.9	89.3	56.2	82.7	38.4
Milwakee	20.3	89.6	53.8	111.8	53.9
Minneapolis	24.6	92.2	51.9	106	50.7
New York	30.8	110.7	66	107.3	62.6
Philadelphia	24.5	92.3	66.7	98	61.7
Pittsburgh	26.2	95.4	60.2	117.1	49.3
St. Louis	26.5	92.4	60.8	115.1	46.2
San Diego	25.5	83.7	57	92.8	35.4
San Franscisco	26.3	87.1	58.3	101.8	41.5
Seattle	22.5	77.7	62	91.1	44.9
Washington, DC	24.2	93.8	66	81.6	46.2

**Source : Estimated Retail Food Prices by Cities, March 1973,
U.S. Department of Labor, Bureau of Labor Statistics,pp. 1-8.**

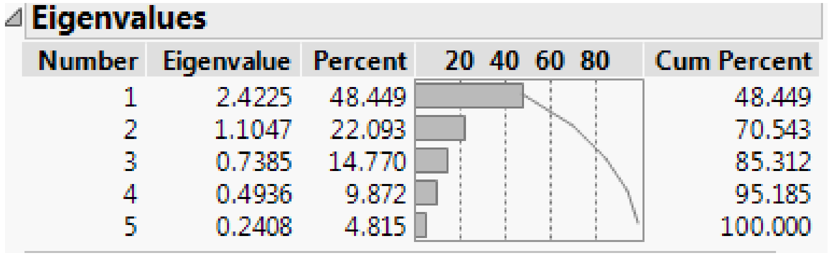
Run the PCA algorithm

Principal components analysis can be done either on mean corrected (X -mean) data or standardized ({X- mean}/sd) data – JMP gives standardized output



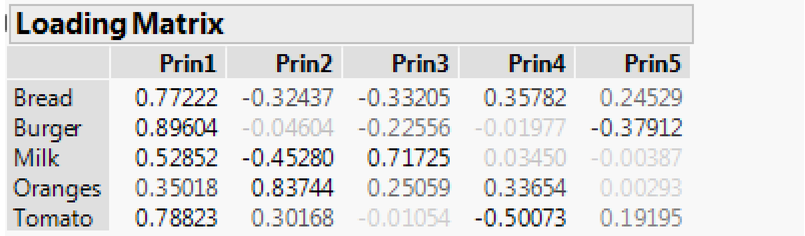
Correlations

	Bread	Burger	Milk	Oranges	Tomato
Bread	1.0000	0.6817	0.3282	0.0367	0.3822
Burger	0.6817	1.0000	0.3334	0.2109	0.6319
Milk	0.3282	0.3334	1.0000	-0.0028	0.2544
Oranges	0.0367	0.2109	-0.0028	1.0000	0.3581
Tomato	0.3822	0.6319	0.2544	0.3581	1.0000



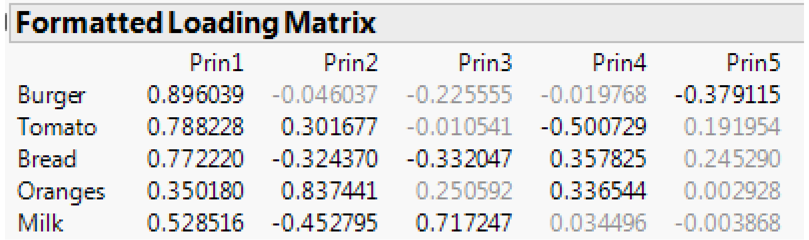
Eigenvalues

Number	Eigenvalue	Percent	20 40 60 80	Cum Percent
1	2.4225	48.449		48.449
2	1.1047	22.093		70.543
3	0.7385	14.770		85.312
4	0.4936	9.872		95.185
5	0.2408	4.815		100.000



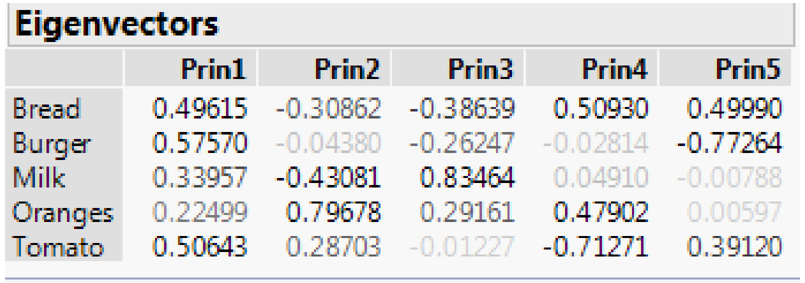
Loading Matrix

	Prin1	Prin2	Prin3	Prin4	Prin5
Bread	0.77222	-0.32437	-0.33205	0.35782	0.24529
Burger	0.89604	-0.04604	-0.22556	-0.01977	-0.37912
Milk	0.52852	-0.45280	0.71725	0.03450	-0.00387
Oranges	0.35018	0.83744	0.25059	0.33654	0.00293
Tomato	0.78823	0.30168	-0.01054	-0.50073	0.19195



Formatted Loading Matrix

	Prin1	Prin2	Prin3	Prin4	Prin5
Burger	0.896039	-0.046037	-0.225555	-0.019768	-0.379115
Tomato	0.788228	0.301677	-0.010541	-0.500729	0.191954
Bread	0.772220	-0.324370	-0.332047	0.357825	0.245290
Oranges	0.350180	0.837441	0.250592	0.336544	0.002928
Milk	0.528516	-0.452795	0.717247	0.034496	-0.003868



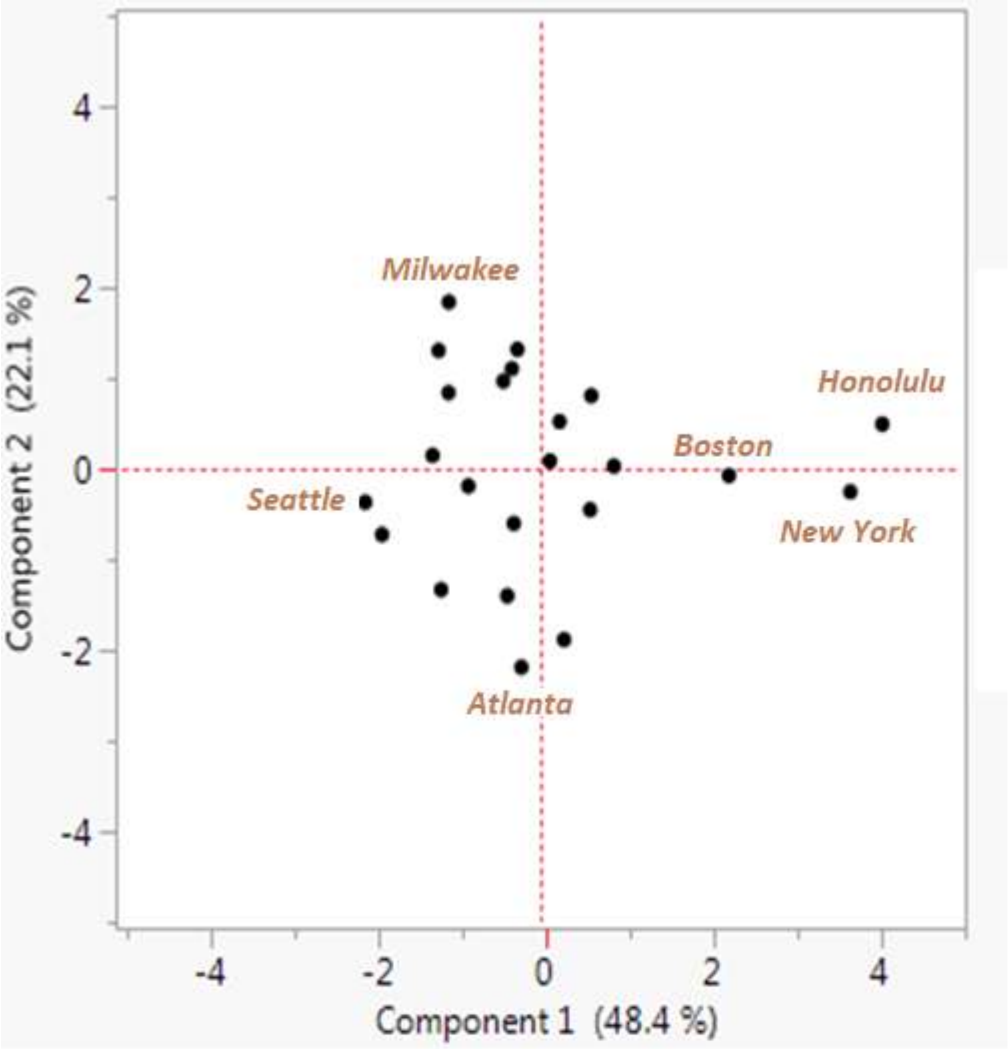
Eigenvectors

	Prin1	Prin2	Prin3	Prin4	Prin5
Bread	0.49615	-0.30862	-0.38639	0.50930	0.49990
Burger	0.57570	-0.04380	-0.26247	-0.02814	-0.77264
Milk	0.33957	-0.43081	0.83464	0.04910	-0.00788
Oranges	0.22499	0.79678	0.29161	0.47902	0.00597
Tomato	0.50643	0.28703	-0.01227	-0.71271	0.39120

```
PC1 = 0.49615*Bread + 0.5757*Burger +
      0.33957*Milk + 0.22499*Oranges +
      0.50643*Tomato
```

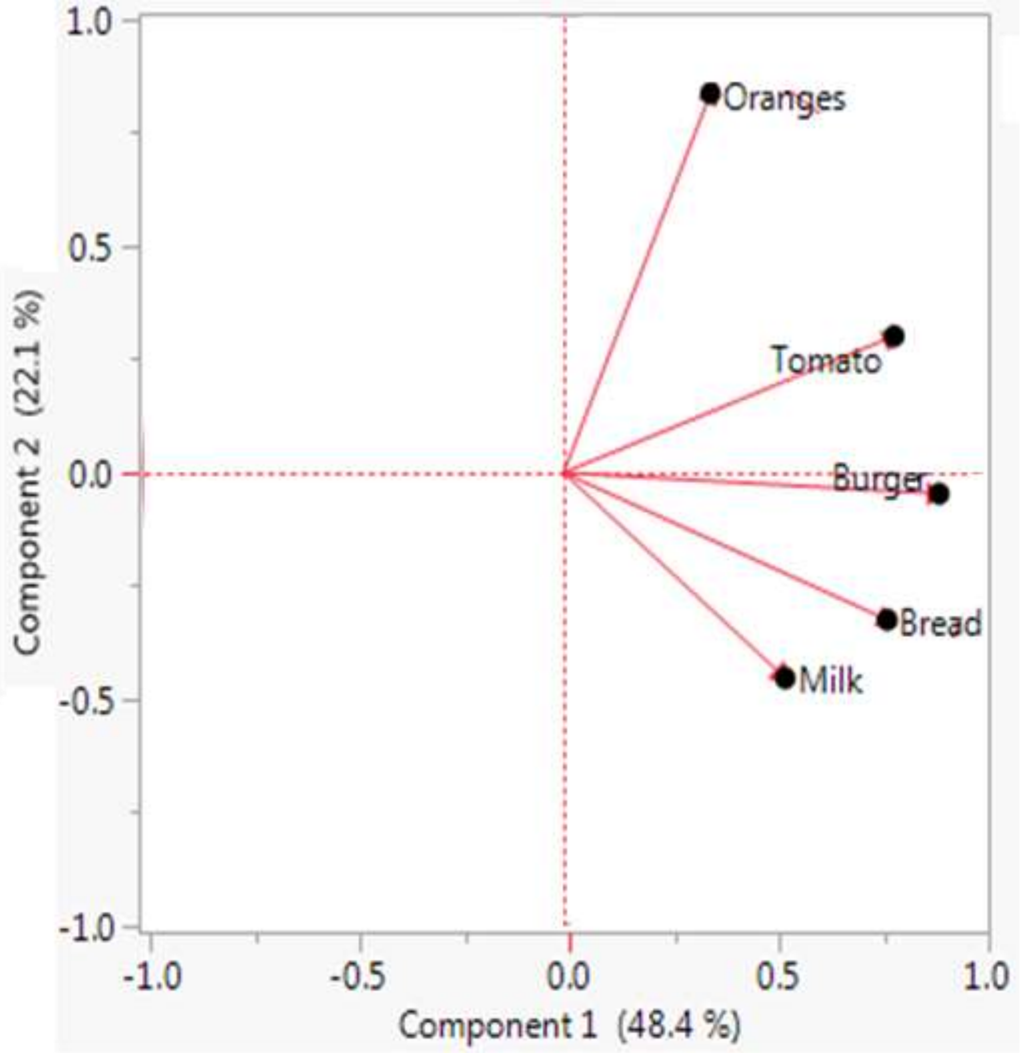
Score Plot

- The Score Plot graphs the **PC Scores** of the First Principal Component (*X*-axis) versus the scores of the Second Principal Component (*Y*-axis).
- In this case, the countries are being plotted.
- Score Plots allow you to visually **detect clusters, outliers, and trends**



Loading Plot

- The Loading Plot graphs the **loadings** of PC1 and PC2.
- Loading Plots allow you to identify which variables have the **largest effect** on each component.
- Loadings close to -1 or 1 indicate that the variable **strongly influences** the component.
- Loadings close to 0 indicate that the variable has a **weak influence** on the component.
- Evaluating the loadings can also help you **characterize each component** in terms of the variables.



PCA Example 2 - Phone Data

We want to analyse customer behaviours and reveal the true data dimensions which underlie the usages

Field Name	Description
Voice Out Calls	Monthly average of outgoing voice calls
Voice Out Mins	Monthly average no. of mins of out going voice calls
SMS Out Calls	Monthly average of outgoing SMS calls
MMS Out Calls	Monthly average of outgoing MMS calls
Out Calls Roaming	Monthly average of outgoing roaming calls
GPRS Traffic	Monthly average of GPRS traffic
PRC Voice Out Calls	Percentage of outgoing voice calls (outgoing voice calls as a percentage of total outgoing calls)
PRC SMS Out Calls	Percentage of outgoing SMS calls
PRC MMS Out Calls	Percentage of outgoing MMS calls
PRC Internet Calls	Percentage of outgoing internet calls
PRC Out Calls Roaming	Percentage of outgoing roaming calls (roaming calls as a percentage of total outgoing calls)

Customer ID	Monthly Avg No. of SMS Calls	Monthly Avg No. of MMS Calls	Monthly Avg No. of Voice Calls	Monthly Avg No. of Voice Call Minutes
1	19	4	90	150
2	43	12	30	35
3	13	3	10	20
4	60	14	100	80
5	5	1	30	55
6	56	11	25	35
7	25	7	30	28
8	3	1	65	82
9	40	9	15	30
10	65	15	20	40

Data Mining Techniques in CRM: Inside Customer Segmentation, Tsiptsis & Chorianopoulos (2009)

Approach for this Example

In this example, we will learn how **PCA works** and examine the PCA output to **judge the effectiveness** of dimensionality reduction. We will walk through the example using the following:

1. Correlation Matrix
2. eigenvalues and total variance explained
3. Scree Plot
4. Component Loading Matrix
5. Rotated Component Loading Matrix
6. Communalities Matrix
7. Component Scores Table

1. Correlation Matrix

	Voice Out Calls	Voice Out Mins	SMS Out Calls	MMS Out Calls	Out Calls Roaming	GPRS Traffic	PRC Voice Out Calls	PRC SMS Out Calls	PRC MMS Out Calls	PRC Internet Calls	PRC Out Calls Roaming
Voice Out Calls	1	0.84	0.2	0.16	0.04	0.14	0.07	-0.07	-0.02	-0.02	-0.02
Voice Out Mins	0.84	1	0.22	0.14	0.05	0.17	0	-0.01	-0.01	-0.01	0
SMS Out Calls	0.2	0.22	1	0.18	0.05	0.28	-0.72	0.72	0.03	0	0.05
MMS Out Calls	0.14	0.17	0.28	0.03	0.17	1	-0.2	0.19	-0.01	0.06	0.59
Out Calls Roaming	0.16	0.14	0.18	1	0	0.03	-0.04	0.13	0.66	0	-0.01
GPRS Traffic	0.04	0.05	0.05	0	1	0.17	-0.05	0.05	-0.01	0.53	0.15
PRC Voice Out Calls	0.07	0	-0.72	-0.04	-0.05	-0.2	1	-0.98	0.02	-0.03	-0.09
PRC SMS Out Calls	-0.07	-0.01	0.72	0.13	0.05	0.19	-0.98	1	0.09	0.03	0.09
PRC MMS Out Calls	-0.02	0	0.05	-0.01	0.15	0.59	-0.09	0.09	-0.02	0.17	1
PRC Internet Calls	-0.02	-0.01	0	0	0.53	0.06	-0.03	0.03	0	1	0.17
PRC Out Calls Roaming	-0.02	-0.01	0.03	0.66	-0.01	-0.01	0.02	0.09	1	0	1

2. eigenvalues & Total Variance Explained

Components	Eigenvalue	Variance	Cumulative %
1	**2.84**	**25.84**	**25.84**
2	**1.96**	**17.78**	**43.62**
3	**1.76**	**16.01**	**59.63**
4	**1.56**	**14.21**	**73.84**
5	**1.25**	**11.33**	**85.16**
6	0.49	4.45	89.62
7	0.38	3.41	93.03
8	0.34	3.06	96.09
9	0.26	2.38	98.47
10	0.16	1.44	99.92
11	0.01	0.08	100.0

Eigenvalues
A measure of the variance that each component accounts for.

Eigenvalue Criterion
The Kaiser rule calls for 1.0 cut-off. Therefore, 5 components will be shortlisted for consideration.

Percentage of Total Variance Explained
Usually should not fall below 65% depending on the application domain.

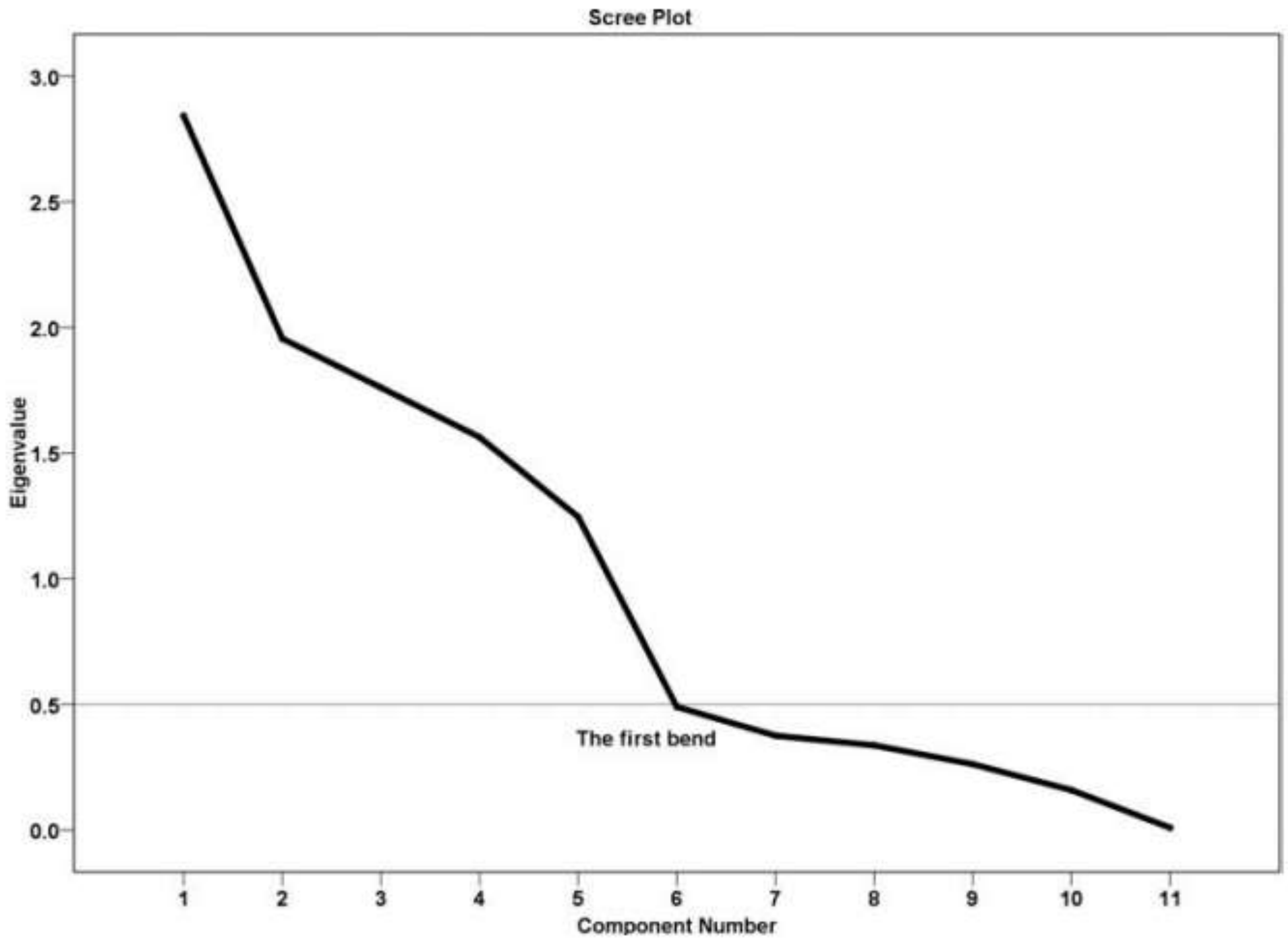
The first 5 components explains 85% of the variability with 15% loss of information.

How many components are to be extracted?

- As a practical guide, we should examine the following PCA output :
  - Eigenvalues (Kaiser rule)
  - Total Variance explained
  - Scree Plot
- Other criteria (non-quantitative):
  - The final solution should take into account criteria such as the interpretability and business meaning of the components.
- Your choice of the final solution should balance simplicity with effectiveness, consisting of a **reduced and interpretable set of components** that can adequately represent the original fields.

3. Scree Plot

The Scree Plot shows the eigenvalues vs the components in decreasing order:



Scree Plot test
Look for a large drop, followed by a "plateau" in the eigenvalues – i.e. a bend or elbow. This indicates a transition from the large to smaller values. Beyond the elbow point, the variances explained tapers off.

If the scree is "not ideal" – i.e. you don't see any clear bends, then fall back on the previous 2 methods.

4. Component Loading Matrix

Field Name	Prin 1	Prin 2	Prin 3	Prin 4	Prin 5
PRC SMS Out Calls	0.89	-0.34	-0.17	-0.06	-0.10
PRC Voice Out Calls	-0.88	0.36	0.11	0.15	0.11
SMS Out Calls	0.86	-0.01	-0.16	-0.16	-0.09
Voice Out Calls	0.20	0.88	-0.04	-0.28	-0.12
Voice Out Mins	0.26	0.86	-0.02	-0.29	-0.11
GPRS Traffic	0.19	0.09	0.60	0.35	-0.48
PRC Internet Calls	0.12	0.02	0.58	0.40	-0.51
PRC Out Calls Roaming	0.14	0.18	-0.44	0.77	0.11
Out Calls Roaming	0.26	0.34	-0.46	0.66	0.08
PRC MMS Out Calls	0.28	0.04	0.59	0.19	0.60
MMS Out Calls	0.47	0.19	0.49	0.04	0.56

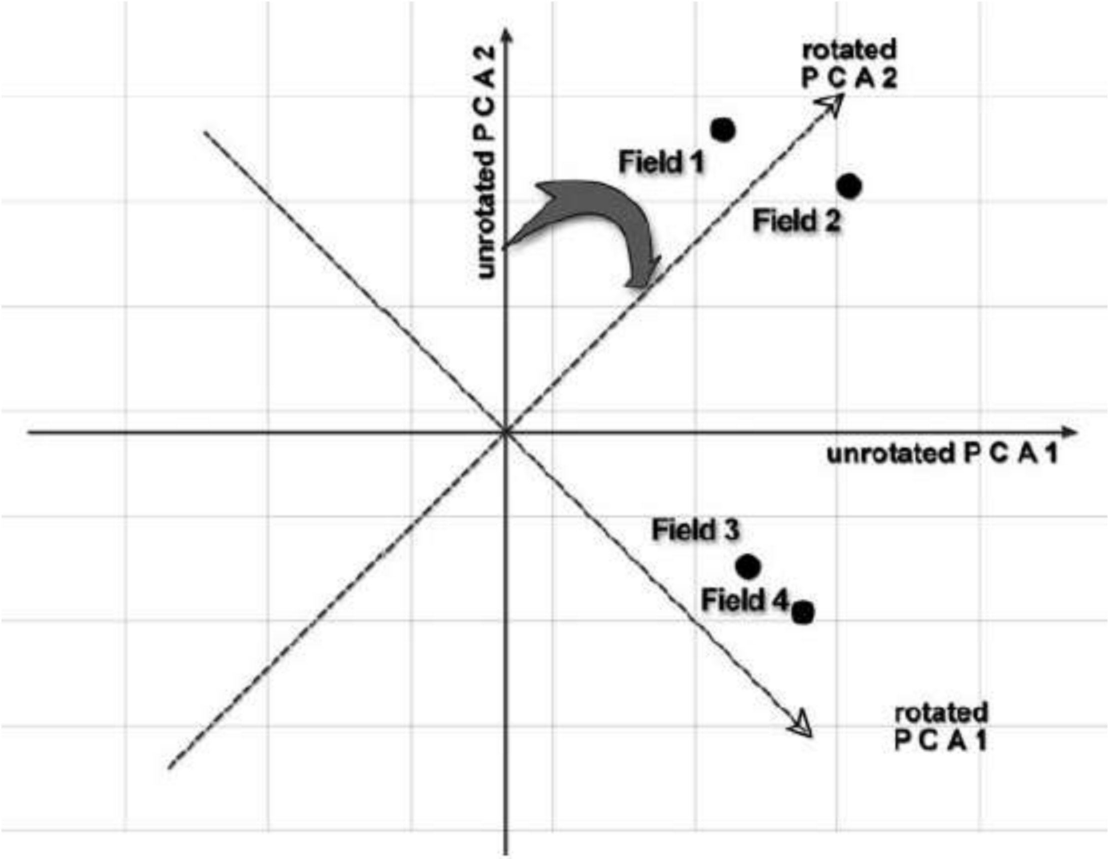
Interpretation and labeling

- The interpretation process involves the examination of the loading values and their signs and identification of significant correlations
- The goal is to understand the **information that they convey** and **name** them accordingly
- Typically correlations above 0.4 (sometimes subjective**)** in absolute value are considered to be a practical significance
- The interpretation process ends with the labelling of the derived components with names that appropriately summarize their meaning
- However, it is easier to interpret the loading matrix if we applied a **Factor Rotation** to it.

Factor Loading Rotation

- Rotation is a common practice used to simplify the interpretation of components.
- There are numerous rotation techniques, with the **Varimax** being the most popular for data reduction purposes and since it yields transparent components which are also uncorrelated.
- Varimax stands for Variance Maximization
- The Varimax **rotation minimizes the complexity of the factors** by making the **large loadings larger** and the **small loadings smaller** within each factor – thus maximizing the variance.

Varimax Rotation



Orthogonal Rotation of the derived components

5. Rotated Component Matrix

This is a **Rotated Component Matrix** showing the factor loadings (>0.4) for each component. This provides a clearer explanation.

Field Name	PC-1	PC-2	PC-3	PC-4	PC-5
PRC Voice Out Calls	-0.97				
PRC SMS Out Calls	0.968				
SMS Out Calls	0.85				
Voice Out Calls		0.954			
Voice Out Mins		0.946			
PRC Out Calls Roaming			0.918		
Out Calls Roaming			0.900		
PRC MMS Out Calls				0.897	
MMS Out Calls				0.866	
PRC Internet Calls					0.880
GPRS Traffic					0.865

- PC-1 is strongly associated with SMS usage which loads heavily on this component.
- PC-2 Is strongly associated with voice calls. High correlation between Calls and Mins.
- PC-3 is strongly associated with roaming usage.
- PC-4 is strongly associated with MMS usage.
- PC-5 is strongly associated with internet usage.

6. Communalities Matrix

Field Name	Communalities
PRC SMS Out Calls	0.954
PRC Voice Out Calls	0.953
Voice Out Calls	0.913
Voice Out Mins	0.901
PRC Out Calls Roaming	0.847
Out Calls Roaming	0.838
PRC MMS Out Calls	0.823
MMS Out Calls	0.808
SMS Out Calls	0.791
PRC Internet Calls	0.778
GPRS Traffic	0.762

Does the solution account for all the original variables?

- Communality represents the total amount of variance of a specific field that is jointly accounted for by all the components. It is calculated as the sum of squared loadings of the field across all components.
- **High communality** values indicate that the original field is **sufficiently explained** by the reduced PCA solution.
- **Low communality** implies an **insignificant** contribution to the formation of the PCA Solution
- In the example presented here, all communalities are > 0.5 (quite high).

7. Component Scores for 10 customers

Customer ID	Prin 1	Prin 2	Prin 3	Prin 4	Prin 5
1	0.633	-0.182	-0.263	1.346	-0.209
2	-0.964	-0.500	**8.805**	-0.090	-0.036
3	-0.501	-0.381	-0.196	-0.197	-0.063
4	-0.501	**1.677**	-0.272	-0.305	-0.055
5	**3.66**	-1.041	-0.385	-0.596	-0.084
6	-0.450	0.720	0.433	-0.251	-0.056
7	1.249	-0.276	1.043	-0.384	-0.028
8	-0.695	0.192	-0.204	0.461	-0.117
9	-0.902	-0.959	0.247	2.265	-0.164
10	0.028	0.212	2.715	1.186	-0.165

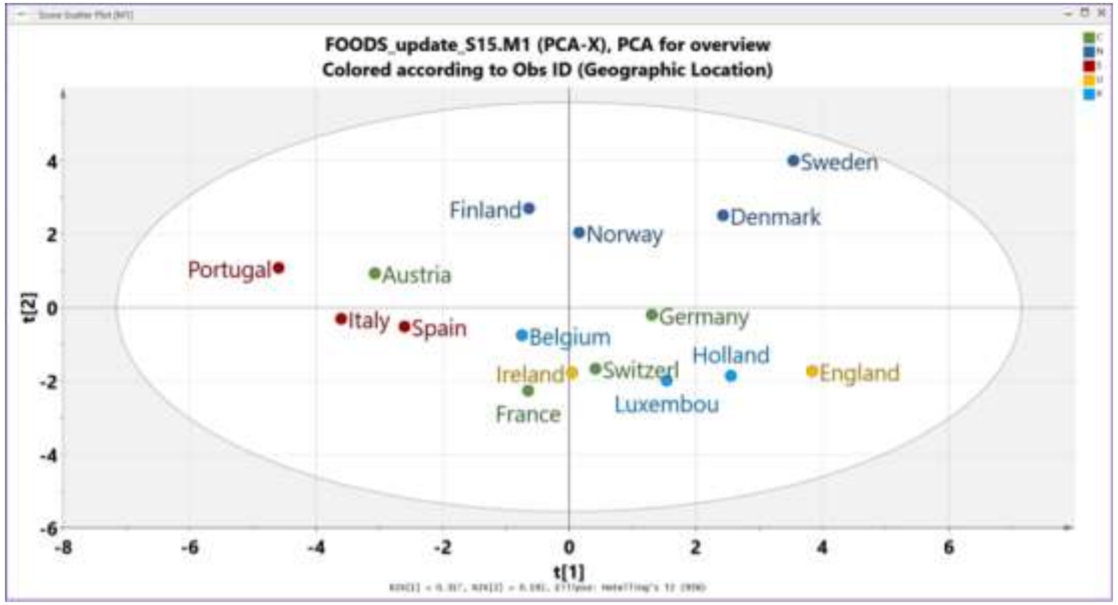
- Components scores can be used as input to downstream analytical modelling.
- Scores are standardized values (mean=0; SD=1)
- Represent the number of standard deviations above or below the overall mean where each customer lies.

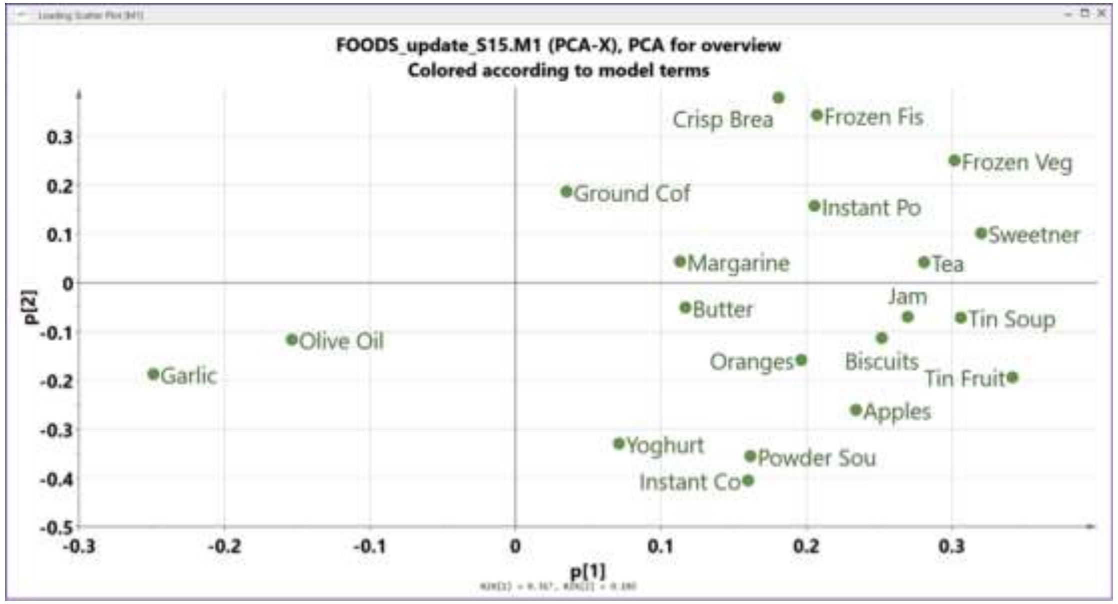
Interpreting the Customer Component Scores

- The high score of customer 5 in component 1 denotes a customer with above average SMS usage.
- The negative score in component 2 indicates low voice usage.
- Similarly, customer2 seems to be a person who frequently uses their phone abroad (roaming usage measured by component 3).
- Customer 4 seems like a typical example of a “voice only” customer

Score Plots & Loading Plots

- Countries close to each other have similar food consumption profiles, whereas those far from each other are dissimilar.
  - Finland, Norway, Denmark and Sweden are located together in the upper right-hand corner, thus representing a group of nations with some similarity in food consumption.
  - Belgium and Germany are close to the center (origin) of the plot, indicates they have average properties.
- Variables contributing similar information are grouped together (correlated)
  - Crisp bread and frozen fish are positively correlated.
  - Garlic and sweetener are negatively correlated
- The further away from the plot origin, the stronger the impact that variable has on the model.
  - Crisp bread, frozen fish, frozen vegetables and Garlic separate the four Nordic countries from the others.
  - The four Nordic countries are characterized as having high values/consumption of the former three provisions, and low consumption of garlic.
  - Moreover, Italy, Portugal, Spain and to some extent, Austria have high consumption of garlic, and low consumption of sweetener, tinned soup and tinned fruit.





You can overlay both plots into a "biplot"

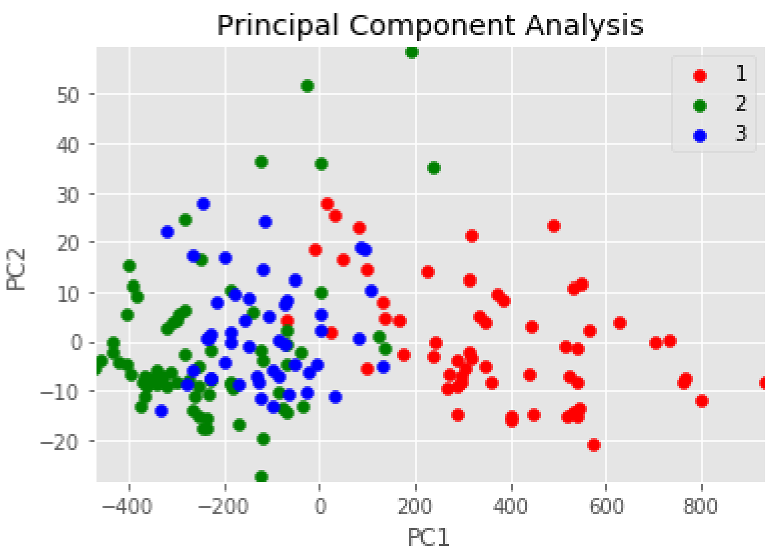
PCA Process and Procedures Using Python

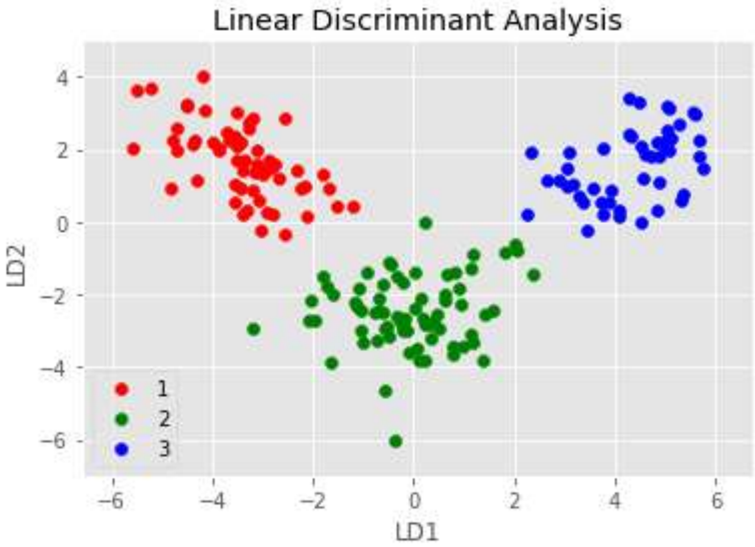
- Since we are not doing PCA from scratch, let us make sure that we know the process of performing it.
- Read in the dataset
    - Preparation: cleaning and transformation
- Standardizing the data
    - Z-Transformation: mean = 0; sd = 1
- Call the PCA algorithm
    - Set the parameters
- Plot the results
- Interpret the results

Demo, Exercises and Workshop

LINEAR DISCRIMINANT ANALYSIS (LDA)

What if you had access to class labels?





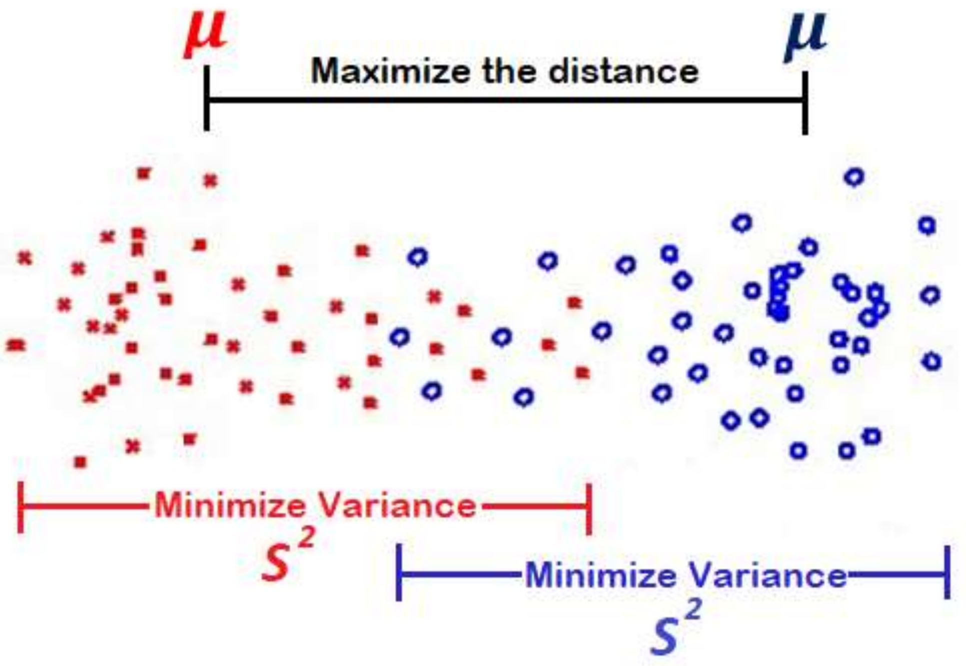
We not only reduce the number of features BUT at the same time create class separability. This is most useful for multi-variate classification-type problems.

Linear Discriminant Analysis

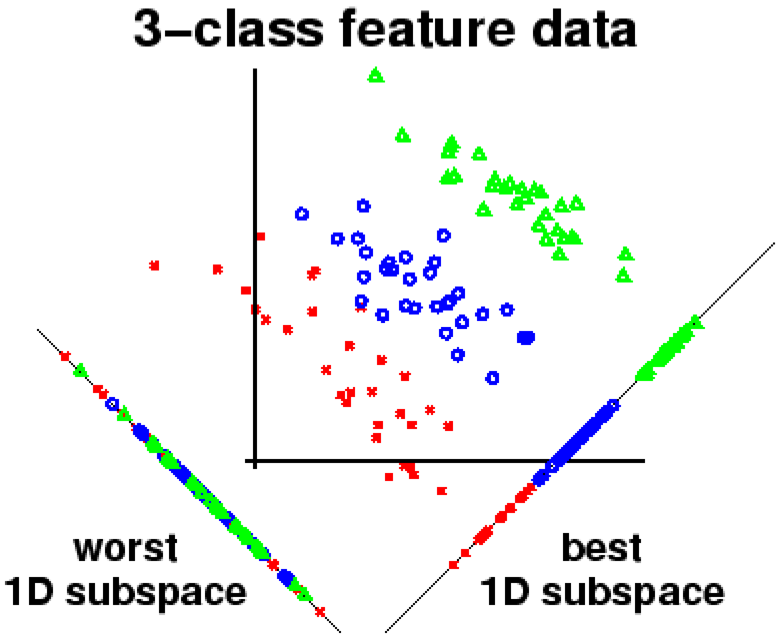
- Originally developed in 1936 by R.A. Fisher, Discriminant Analysis is a classic method of classification that has stood the test of time. Discriminant analysis often produces models whose accuracy approaches (and occasionally exceeds) more complex modern methods.
- There are lots of similarities between LDA and PCA.
- Whereas PCA yields the directions (principal components) that maximize the variance of the data, LDA also aims to find the directions that **maximize the separation** (or discrimination) between different classes, which can be useful in pattern classification problem.
- LDA takes the **class label** into consideration whereas PCA ignores them.
- Since it uses the class information, LDA is a "**supervised learning**" method while PCA is an unsupervised learning method.

The LDA Method

- LDA finds a new dimension that yields
  - Maximum separation between the class means
  - Minimum variance within class
- It uses eigenvectors from between classes, and covariance matrix from within-class



$$\frac{(\mu - \mu)}{s^2 + s^2}$$



Example

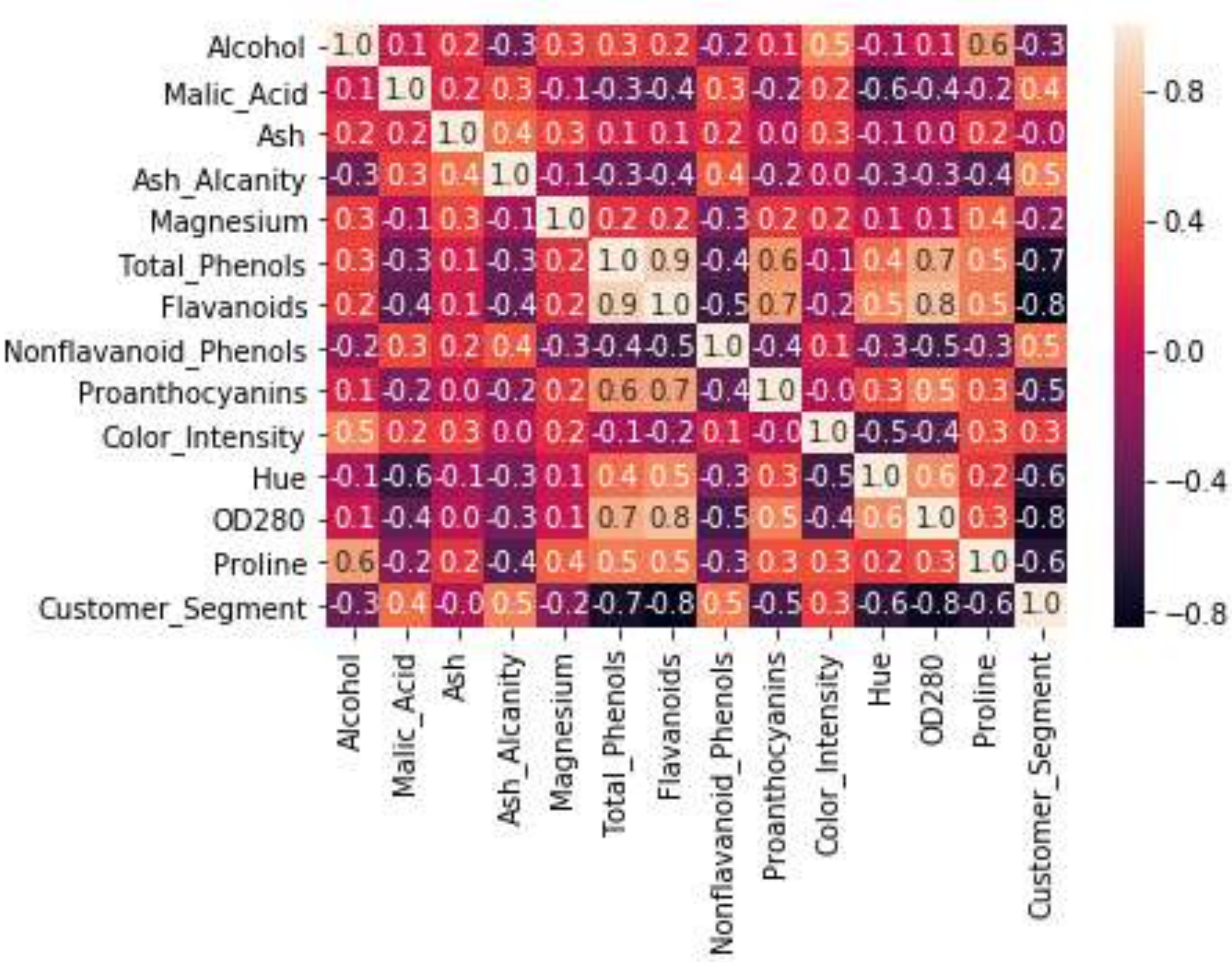
Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proline	class
14.23	1.71	2.43	15.6	127	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1065	1
13.2	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1050	1
13.16	2.36	2.67	18.6	101	2.8	3.24	0.3	2.81	5.68	1.03	3.17	1185	1
14.37	1.95	2.5	16.8	113	3.85	3.49	0.24	2.18	7.8	0.86	3.45	1480	1
13.24	2.59	2.87	21	118	2.8	2.69	0.39	1.82	4.32	1.04	2.93	735	1
14.2	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1450	1
14.39	1.87	2.45	14.6	96	2.5	2.52	0.3	1.98	5.25	1.02	3.58	1290	1
14.06	2.15	2.61	17.6	121	2.6	2.51	0.31	1.25	5.05	1.06	3.58	1295	1
14.83	1.64	2.17	14	97	2.8	2.98	0.29	1.98	5.2	1.08	2.85	1045	1
13.86	1.35	2.27	16	98	2.98	3.15	0.22	1.85	7.22	1.01	3.55	1045	1
14.1	2.16	2.3	18	105	2.95	3.32	0.22	2.38	5.75	1.25	3.17	1510	1

This data set has 178 examples from three classes. Each example consists of 13 real-valued features.

Summary Statistics

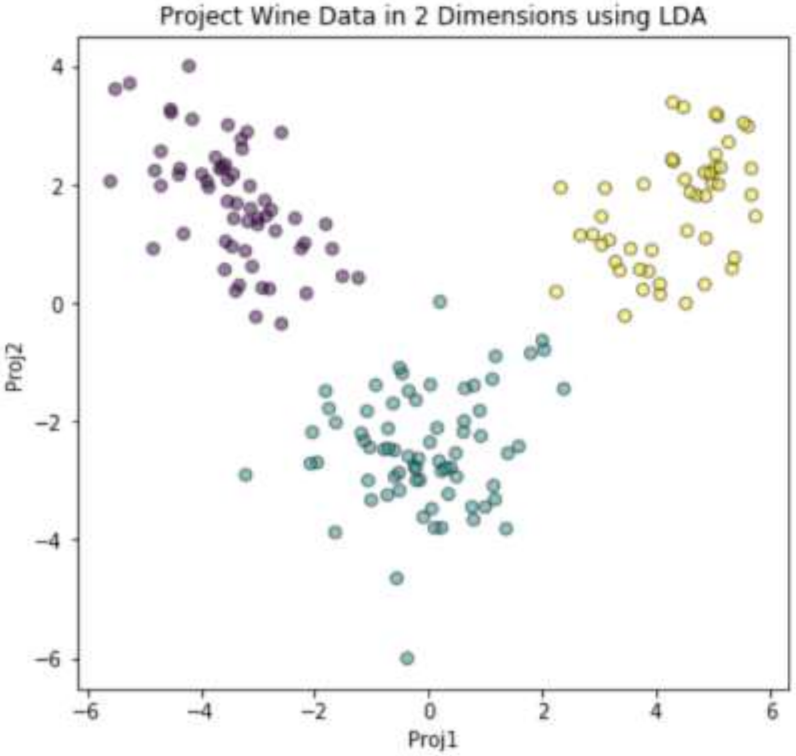
	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proline	class
count	178	178	178	178	178	178	178	178	178	178	178	178	178	178
mean	13	2.34	2.37	19.49	99.74	2.3	2.03	0.36	1.59	5.06	0.96	2.61	746.89	1.94
std	0.81	1.12	0.27	3.34	14.28	0.63	1	0.12	0.57	2.32	0.23	0.71	314.91	0.78
min	11.03	0.74	1.36	10.6	70	0.98	0.34	0.13	0.41	1.28	0.48	1.27	278	1
25%	12.36	1.6	2.21	17.2	88	1.74	1.2	0.27	1.25	3.22	0.78	1.94	500.5	1
50%	13.05	1.87	2.36	19.5	98	2.36	2.13	0.34	1.56	4.69	0.96	2.78	673.5	2
75%	13.68	3.08	2.56	21.5	107	2.8	2.88	0.44	1.95	6.2	1.12	3.17	985	3
max	14.83	5.8	3.23	30	162	3.88	5.08	0.66	3.58	13	1.71	4	1680	3

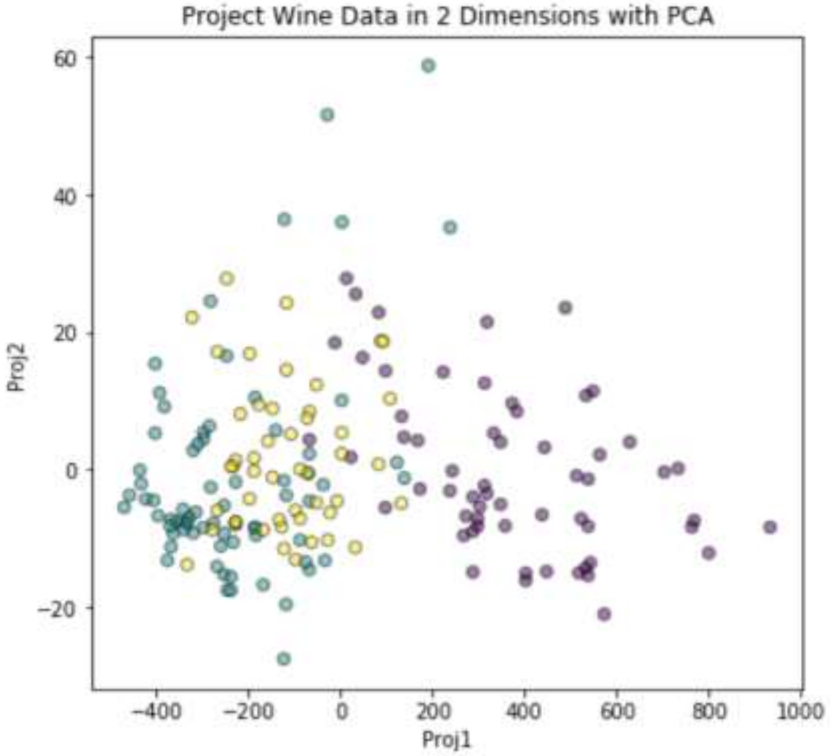
Correlation Matrix

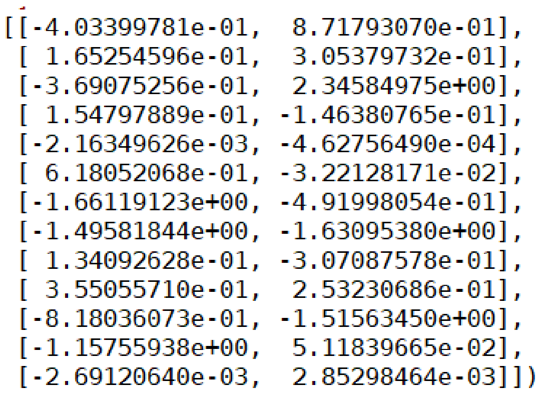


	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proline	Customer_Segment
Alcohol	1.0	0.1	0.2	-0.3	0.3	0.3	0.2	-0.2	0.1	0.5	-0.1	0.1	0.6	-0.3
Malic_Acid	0.1	1.0	0.2	0.3	-0.1	-0.3	-0.4	0.3	-0.2	0.2	-0.6	-0.4	-0.2	0.4
Ash	0.2	0.2	1.0	0.4	0.3	0.1	0.1	0.2	0.0	0.3	-0.1	0.0	0.2	-0.0
Ash_Alcanity	-0.3	0.3	0.4	1.0	-0.1	-0.3	-0.4	0.4	-0.2	0.0	-0.3	-0.3	-0.4	0.5
Magnesium	0.3	-0.1	0.3	-0.1	1.0	0.2	0.2	-0.3	0.2	0.2	0.1	0.1	0.4	-0.2
Total_Phenols	0.3	-0.3	0.1	-0.3	0.2	1.0	0.9	-0.4	0.6	-0.1	0.4	0.7	0.5	-0.7
Flavanoids	0.2	-0.4	0.1	-0.4	0.2	0.9	1.0	-0.5	0.7	-0.2	0.5	0.8	0.5	-0.8
Nonflavanoid_Phenols	-0.2	0.3	0.2	0.4	-0.3	-0.4	-0.5	1.0	-0.4	0.1	-0.3	-0.5	-0.3	0.5
Proanthocyanins	0.1	-0.2	0.0	-0.2	0.2	0.6	0.7	-0.4	1.0	-0.0	0.3	0.5	0.3	-0.5
Color_Intensity	0.5	0.2	0.3	0.0	0.2	-0.1	-0.2	0.1	-0.0	1.0	-0.5	-0.4	0.3	0.3
Hue	-0.1	-0.6	-0.1	-0.3	0.1	0.4	0.5	-0.3	0.3	-0.5	1.0	0.6	0.2	-0.6
OD280	0.1	-0.4	0.0	-0.3	0.1	0.7	0.8	-0.5	0.5	-0.4	0.6	1.0	0.3	-0.8
Proline	0.6	-0.2	0.2	-0.4	0.4	0.5	0.5	-0.3	0.3	0.3	0.2	0.3	1.0	-0.6
Customer_Segment	-0.3	0.4	-0.0	0.5	-0.2	-0.7	-0.8	0.5	-0.5	0.3	-0.6	-0.8	-0.6	1.0

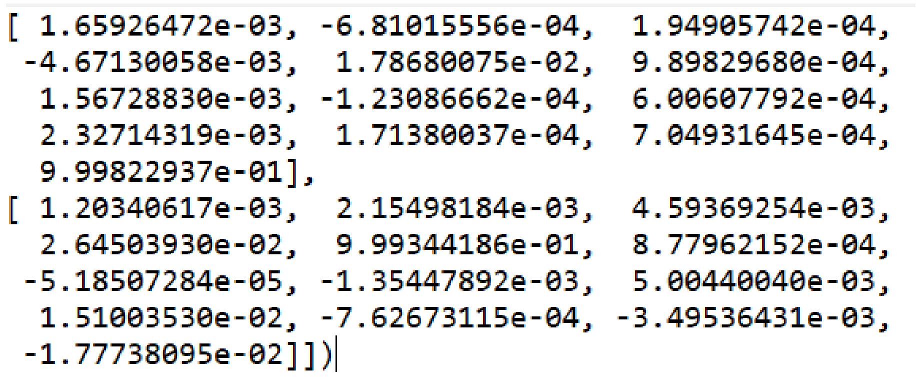
Results of LDA vs PCA







```
[[-4.03399781e-01,  8.71793070e-01],
 [ 1.65254596e-01,  3.05379732e-01],
 [-3.69075256e-01,  2.34584975e+00],
 [ 1.54797889e-01, -1.46380765e-01],
 [-2.16349626e-03, -4.62756490e-04],
 [ 6.18052068e-01, -3.22128171e-02],
 [-1.66119123e+00, -4.91998054e-01],
 [-1.49581844e+00, -1.63095380e+00],
 [ 1.34092628e-01, -3.07087578e-01],
 [ 3.55055710e-01,  2.53230686e-01],
 [-8.18036073e-01, -1.51563450e+00],
 [-1.15755938e+00,  5.11839665e-02],
 [-2.69120640e-03,  2.85298464e-03]])
```



```
[ 1.65926472e-03, -6.81015556e-04,  1.94905742e-04,
 -4.67130058e-03,  1.78680075e-02,  9.89829680e-04,
  1.56728830e-03, -1.23086662e-04,  6.00607792e-04,
  2.32714319e-03,  1.71380037e-04,  7.04931645e-04,
  9.99822937e-01],
[ 1.20340617e-03,  2.15498184e-03,  4.59369254e-03,
  2.64503930e-02,  9.99344186e-01,  8.77962152e-04,
 -5.18507284e-05, -1.35447892e-03,  5.00440040e-03,
  1.51003530e-02, -7.62673115e-04, -3.49536431e-03,
 -1.77738095e-02]])
```

Demo, Exercises and Workshop

Thank You!