Data Pre-processing

Tseng-Ching James Shen, Ph.D.

Topics

- Dataset Exploratory
- Data Cleaning
- Data Transformation

Dataset Exploratory

- Understand features’ statistical characteristics
- Understand the correlation between features
- Measuring Data Similarity
- Apply data visualization techniques to facilitate the analysis

Understand Features’ Statistical Characteristics

- Measuring Central Tendency
  - Mean (*) $\mu = \frac{\sum_{i=1}^{N} X_i}{N} = \frac{\sum X}{N}$
  - Median
- Measuring Dispersion of Data
  - Variance (*) $\sigma^2 = \frac{\sum (X-\mu)^2}{N}$
  - Standard Deviation (*)
  - Range
  - Quartiles

Understand Features' Statistical Characteristics (cont.)

- Gaussian distribution

$$p(x \mid \mu, \sigma^2) = \mathcal{N}(x; \mu, \sigma^2) = \frac{1}{Z} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right).$$

where $Z = \sqrt{2\pi\sigma^2}.$

Correlation of Features

- Correlation analysis for numerical data
  - Covariance
  - Correlation coefficient (also known as Pearson’s product moment coefficient)
- Joint distribution of multiple features
  - Multivariate Gaussian (Sec 2.5.4 in textbook)
- Correlation analysis for categorical (nominal or ordinal) data
  - Chi-square Test

Covariance

- Feature X and Y are two random variables
- Covariance between X and Y are defined as follows

$$\mathrm{Cov}(X,Y) = E\big[(X - EX)(Y - EY)\big] = E[XY] - (EX)(EY).$$

- The covariance (between -1 and 1) gives some information about how X and Y are statistically related

Correlation Coefficient

- Definition

$$\rho_{XY} = \rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var(X)}\,\mathrm{Var(Y)}}} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

- Use the following to determine if two random variables are correlated

If $\rho(X,Y) = 0$, we say that X and Y are **uncorrelated**.
If $\rho(X,Y) > 0$, we say that X and Y are **positively** correlated.
If $\rho(X,Y) < 0$, we say that X and Y are **negatively** correlated.

Multivariate Gaussian

- Describes the joint distribution over random variables X_1, X_2, …, X_D

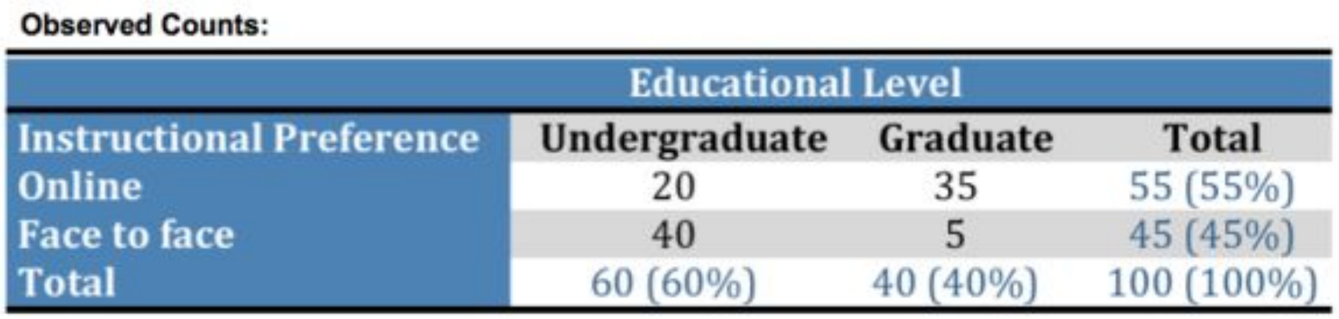
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[(\mathbf{x} - \boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

where Σ is the covariance matrix

- If X_i, and X_j are independent for $i \neq j$, it equals to a product of univariate Gaussian distributions

Chi-square Test

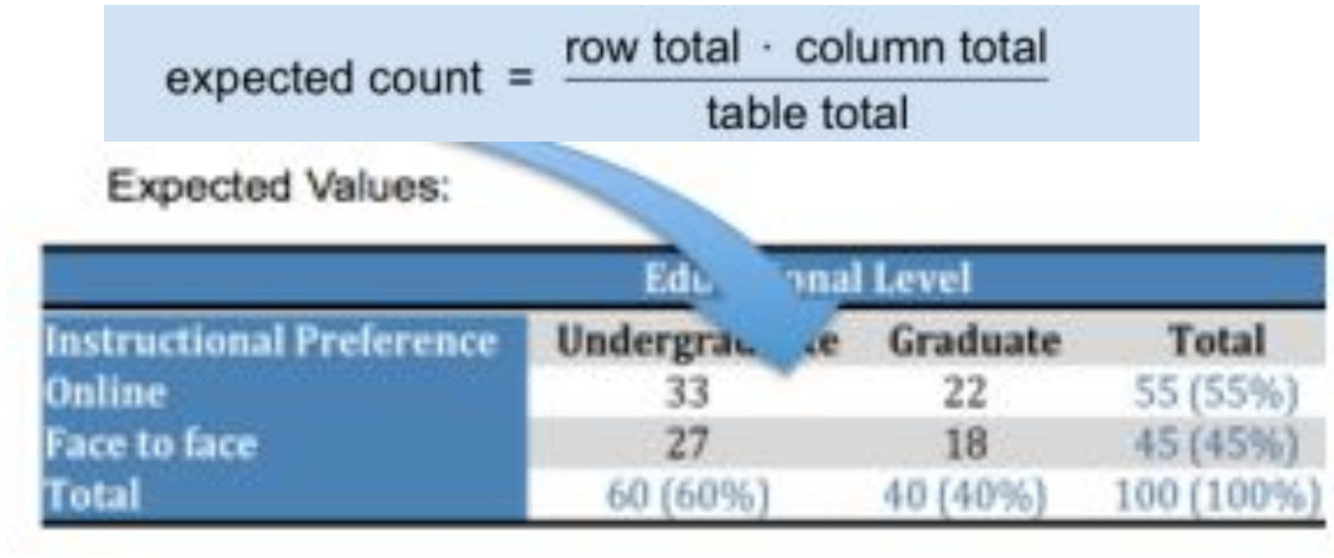
- Create the contingency table



Observed Counts:

	Educational Level		
Instructional Preference	Undergraduate	Graduate	Total
Online	20	35	55 (55%)
Face to face	40	5	45 (45%)
Total	60 (60%)	40 (40%)	100 (100%)

- Calculate the expected counts



$$\text{expected count} = \frac{\text{row total} \cdot \text{column total}}{\text{table total}}$$

Expected Values:

	Educational Level		
Instructional Preference	Undergraduate	Graduate	Total
Online	33	22	55 (55%)
Face to face	27	18	45 (45%)
Total	60 (60%)	40 (40%)	100 (100%)

Chi-square Test (cont.)

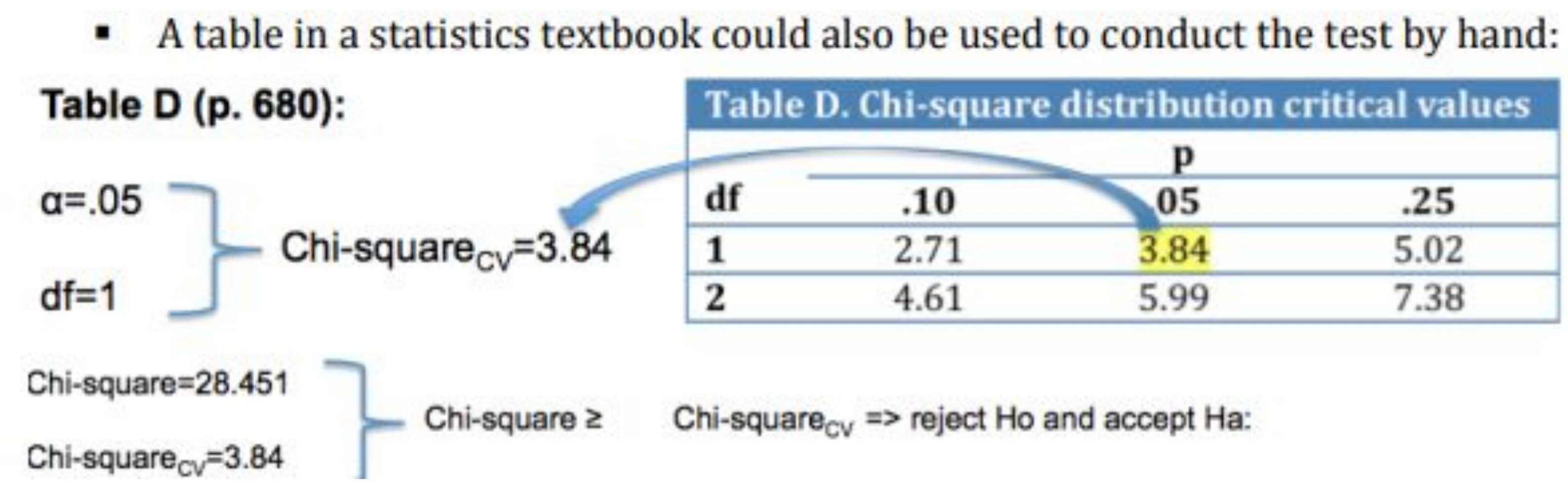
- Calculate the chi-square statistics

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

$$\chi^2 = \frac{(20-33)^2}{33} + \frac{(35-22)^2}{22} + \frac{(40-27)^2}{27} + \frac{(5-18)^2}{18} = 28.451$$

Chi-square Test (cont.)

- Determine the result
  - Degree of freedom $df=(\#rows-1)*(\#columns-1)$
  - Lookup of the hypothesis rejection criteria



- A table in a statistics textbook could also be used to conduct the test by hand:

Table D (p. 680):

$$\left.\begin{array}{l}\alpha=.05\\ df=1\end{array}\right\} \text{Chi-square}_{CV}=3.84$$

Table D. Chi-square distribution critical values

df	p .10	 .05	 .25
1	2.71	3.84	5.02
2	4.61	5.99	7.38

$$\left.\begin{array}{l}\text{Chi-square}=28.451\\ \text{Chi-square}_{CV}=3.84\end{array}\right\} \text{Chi-square} \geq \text{Chi-square}_{CV} \Rightarrow \text{reject Ho and accept Ha:}$$

Data Similarity Measures

- Quantify how likely two objects are like each other
- Dissimilarity Matrix [x(i, j)]
  - x(i, j) measures the “difference” between i-th and j-th object
  - 1 >= x(i, j) >= 0
  - x(i, i) = 0
  - x(i, j) = x(j, i)

Data Similarity Measures (cont.)

- Calculation of Dissimilarity Matrix [d(i, j)]
  - For Nominal Data Type

$$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$$

  - For Ordinal Data Type

$$d = \frac{||p - q||}{n - 1}$$

  - For Numerical Data Type
    - o Euclidean distance

$$d_E(i,j) = \left(\sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^{\frac{1}{2}}$$

    - o Mahalanobis distance

$$d_{MH}(i,j) = \left((x_i - x_j)^T \Sigma^{-1} (x_i - x_j) \right)^{\frac{1}{2}}$$

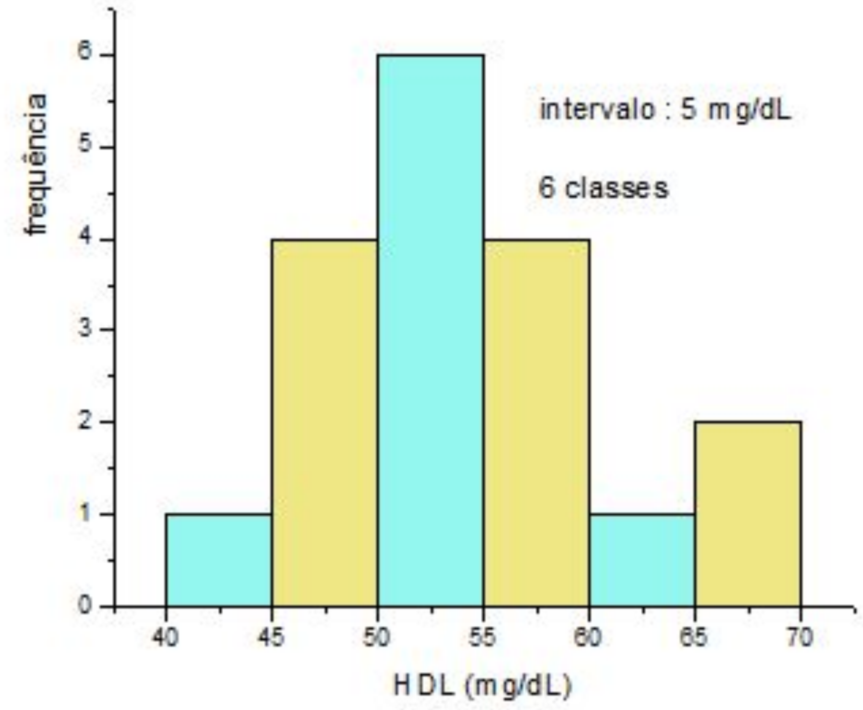
Data Similarity Measures (cont.)

- Numerical data might need to be normalized first
- The measures are used by clustering algorithms

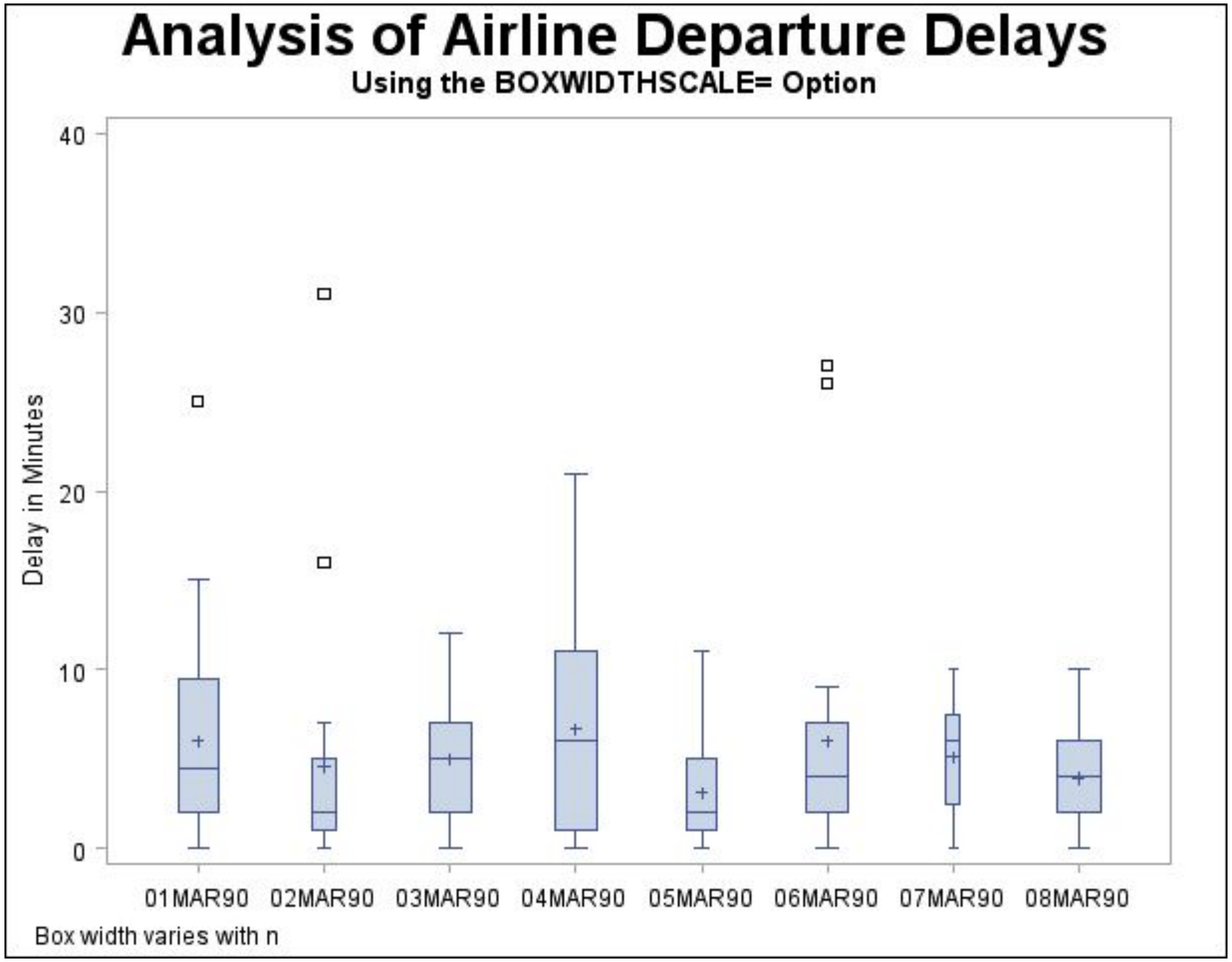
Data Visualization

- Histogram
- Box Plot
- Density Plot
- Scatter Plot (*)
- Heatmap (*)
- 3D Plot (*)
- Contour Plot (*)
- Python matplotlib and seaborn libraries

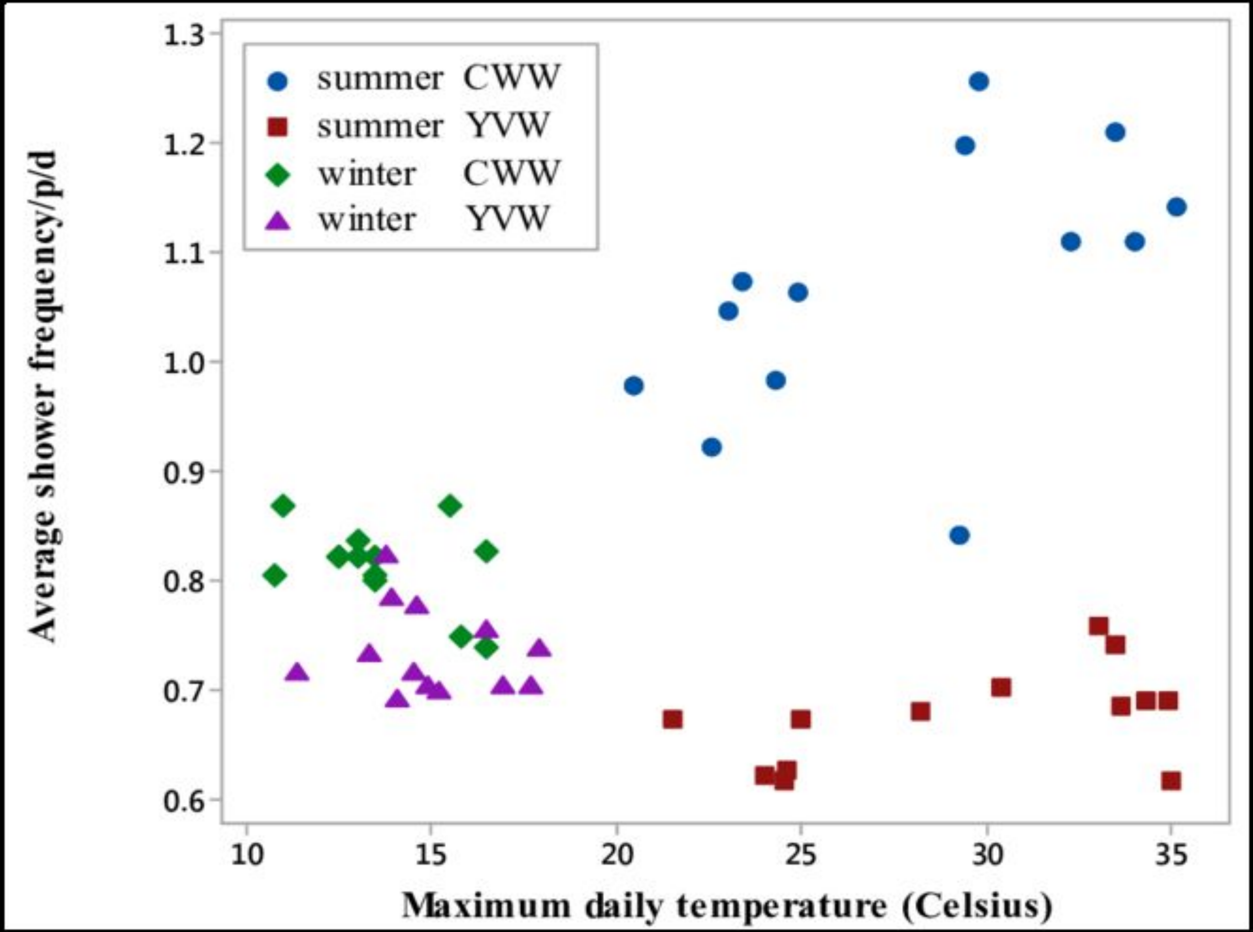
Histogram


intervalo : 5 mg/dL
6 classes
frequência
HDL (mg/dL)
0
1
2
3
4
5
6
40
45
50
55
60
65
70

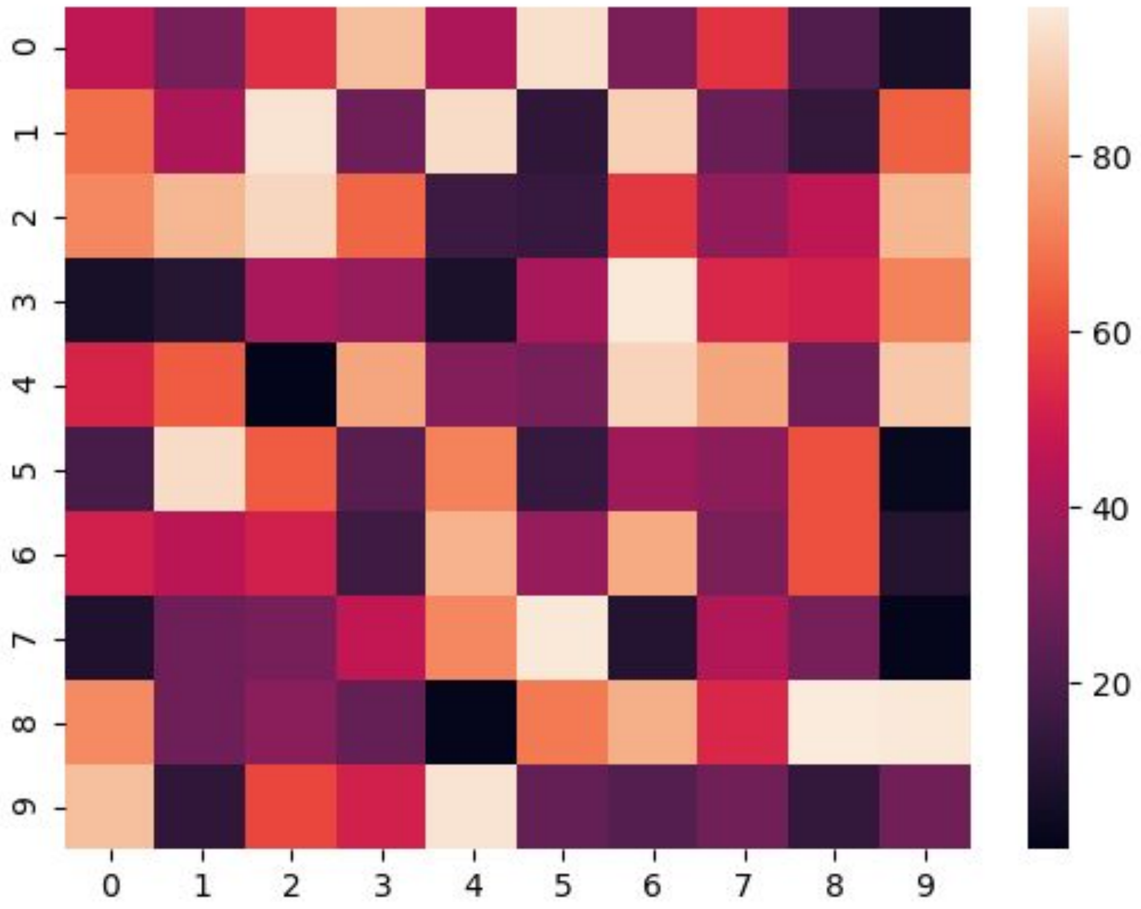
Box Plot


Analysis of Airline Departure Delays
Using the BOXWIDTHSCALE= Option
Delay in Minutes
0
10
20
30
40
01MAR90
02MAR90
03MAR90
04MAR90
05MAR90
06MAR90
07MAR90
08MAR90
Box width varies with n

Scatter Plot

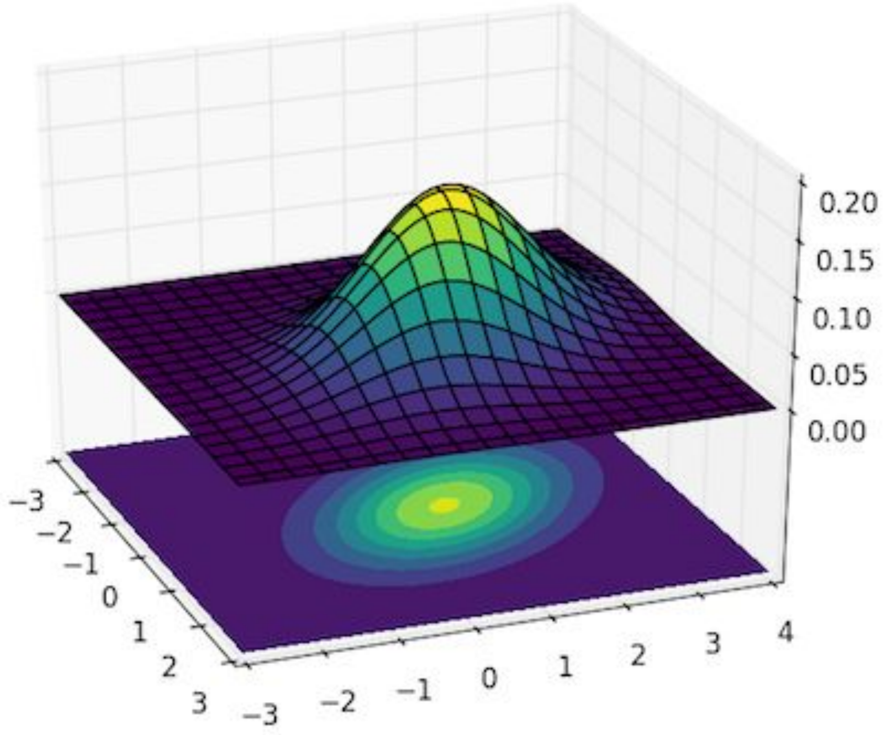


Heatmap



3D and Contour Plot

Bivariate Gaussian Distribution



Data Cleaning

- Missing Values Treatment
- Noisy Data Treatment

Missing Value Treatment

- Deletion
- Mean/Median/Mode Imputation
- Most Probable Value Imputation
- K-nearest neighbors (KNN) Imputation

Noisy Data Treatment

- Binning Methods
  - By bin means
  - By bin medians
  - By bin boundaries
- Regression Analysis
- Outlier Analysis/Removal

Data Transformation

- Feature Scaling
- Conversion of Categorical Data
- Image Feature Extraction
- Dimensionality Reduction
    - Feature Selection
    - Feature Extraction

Feature Scaling

- Min-max Normalization

$$\frac{value - min}{max - min}$$

- Z-score Normalization

$$\frac{value - \mu}{\sigma}$$

Conversion of Categorical Data

- Most Machine Learning Algorithms Only Handle Numeric Data
- Conversion of Ordinal Data into Numeric Data
- Conversion of Nominal Data into Numeric Data
    - One-hot Encoding Scheme
    - Dummy Coding Scheme
    - Effect Coding Scheme etc.

Image Feature Extraction

- It is not mandatory for image classification
- Main usage for image clustering
- Methods
    - Convolutional Neural Network (CNN)
    - Autoencoders
    - Edge features
    - Grayscale features etc.

Dimensionality Reduction

- Reduce the number of features to be considered for analysis
- Purpose
  - Lower computational complexity
  - Decrease required storage
  - Improve learning performance
  - Build better generalizable model (Reduce Over-fitting)
- Approaches
  - Feature Selection
  - Feature Extraction

Feature Selection

- Domain expertise
- Process to find a subset of Features by removing the following from the original features
  - Redundant features
  - Irrelevant features
- Use correlation between features to determine redundant and irrelevant features

Identify Redundant Features

- High correlation between two independent features => high redundancy
- Three types of correlations:
    - The correlation between two continuous features
    - The correlation between one continuous feature and one categorical feature
    - The correlation between two categorical features

Identify Irrelevant Features

- Missing values
- Zero-variance check
- Low correlation between an features and the response => high irrelevancy

Feature Extraction

- Dimension Reduction -

- Original Features are projected into a new space of a Lower Dimensionality
- Approaches
  - Principle Component Analysis (PCA)
  - Kernel PCA
  - Linear Discriminant Analysis (LDA)
  - etc.

Principal Component Analysis (PCA)

- Motivation
- Mathematics
- How does it work

Motivation

- The original features are projected into new features with a set of linear functions
- The linear functions are defined such that
  - The variance of the new features are ‘maximized’
  - Fewer number of new features than the original one can represent the total variance of all original features
  - The new features selected to represent the total variance are *Principal Components*

Mathematics

- Transformation vectors $\boldsymbol{w}_{\mathrm{d}}$
- The d-th feature is calculated with the linear equation $\boldsymbol{X}\,\boldsymbol{w}_{\mathrm{d}}$
- Find $\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_n}$ such that the variance of every new feature is maximized
- This can be formulated as an optimization problem and *eigenvalues/eigenvectors of* a covariance matrix are the solution given the following two assumptions:
  - Original features have zero mean
  - Transformation vectors are orthogonal

How PCA Works

- Apply feature scaling to all original features so that the means are zero
- Calculate the covariance matrix of the original features
- Find the eigenvalues and eigenvectors for the matrix
- Sort the eigenvalues
- The first D eigenvalues as the transformation vectors where D is the dimension of new feature space

How PCA Works (Cont.)

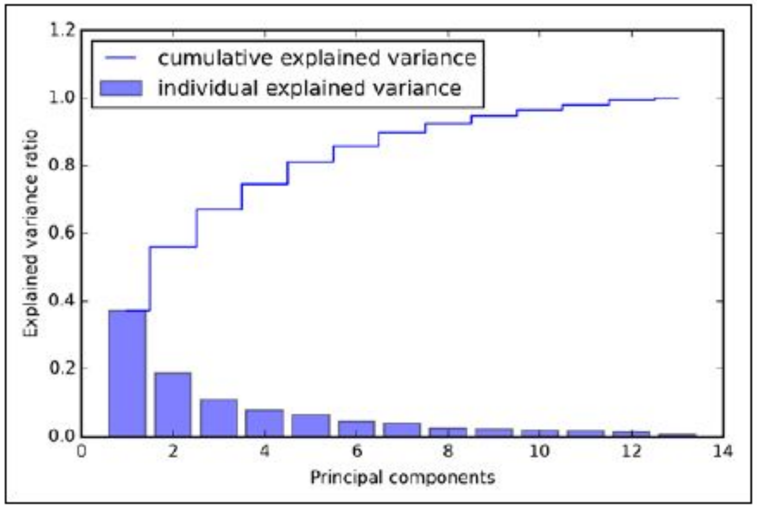
```
>>> import pandas as pd
>>> df_wine = pd.read_csv('https://archive.ics.uci.edu/ml/machine-
learning-databases/wine/wine.data', header=None)
```

```
>>> from sklearn.cross_validation import train_test_split
>>> from sklearn.preprocessing import StandardScaler
>>> X, y = df_wine.iloc[:, 1:].values, df_wine.iloc[:, 0].values
>>> X_train, X_test, y_train, y_test = \
...              train_test_split(X, y,
...              test_size=0.3, random_state=0)
>>> sc = StandardScaler()
>>> X_train_std = sc.fit_transform(X_train)
>>> X_test_std = sc.fit_transform(X_test)
>>> import numpy as np
>>> cov_mat = np.cov(X_train_std.T)
>>> eigen_vals, eigen_vecs = np.linalg.eig(cov_mat)
>>> print('\nEigenvalues \n%s' % eigen_vals)
```

```
Eigenvalues
[ 4.8923083   2.46635032  1.42809973  1.01233462  0.84906459
0.60181514
0.52251546  0.08414846  0.33051429  0.29595018  0.16831254  0.21432212
0.2399553 ]
```

```
>>> tot = sum(eigen_vals)
>>> var_exp = [(i / tot) for i in
...             sorted(eigen_vals, reverse=True)]
>>> cum_var_exp = np.cumsum(var_exp)

>>> import matplotlib.pyplot as plt
>>> plt.bar(range(1,14), var_exp, alpha=0.5, align='center',
...         label='individual explained variance')
>>> plt.step(range(1,14), cum_var_exp, where='mid',
...         label='cumulative explained variance')
>>> plt.ylabel('Explained variance ratio')
>>> plt.xlabel('Principal components')
>>> plt.legend(loc='best')
>>> plt.show()
```



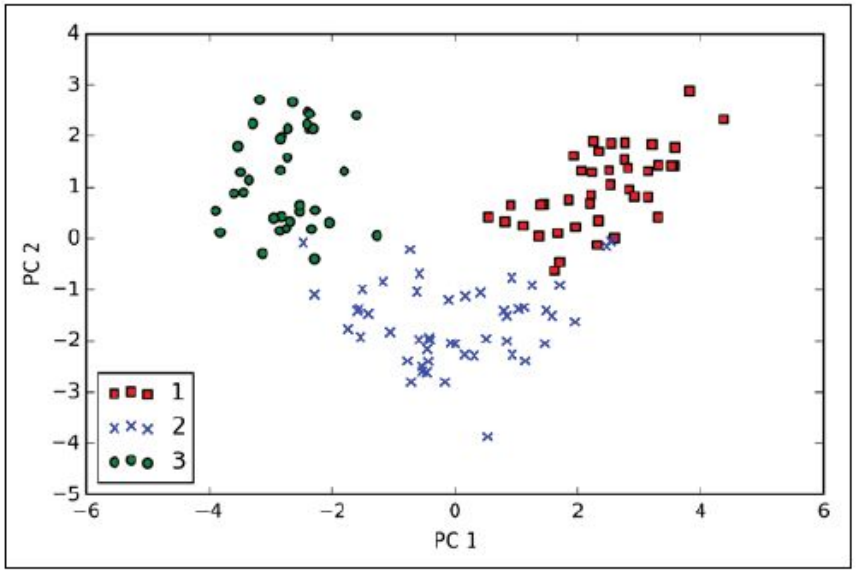
```
>>> eigen_pairs =[(np.abs(eigen_vals[i]),eigen_vecs[:,i])
...               for i inrange(len(eigen_vals))]
>>> eigen_pairs.sort(reverse=True)
```

```
>>> w= np.hstack((eigen_pairs[0][1][:, np.newaxis],
...                eigen_pairs[1][1][:, np.newaxis]))
>>> print('Matrix W:\n',w)
Matrix W:
[[ 0.14669811  0.50417079]
[-0.24224554  0.24216889]
[-0.02993442  0.28698484]
[-0.25519002 -0.06468718]
[ 0.12079772  0.22995385]
[ 0.38934455  0.09363991]
[ 0.42326486  0.01088622]
[-0.30634956  0.01870216]
[ 0.30572219  0.03040352]
[-0.09869191  0.54527081]
```

```
>>> X_train_std[0].dot(w)
array([ 2.59891628,  0.00484089])
```

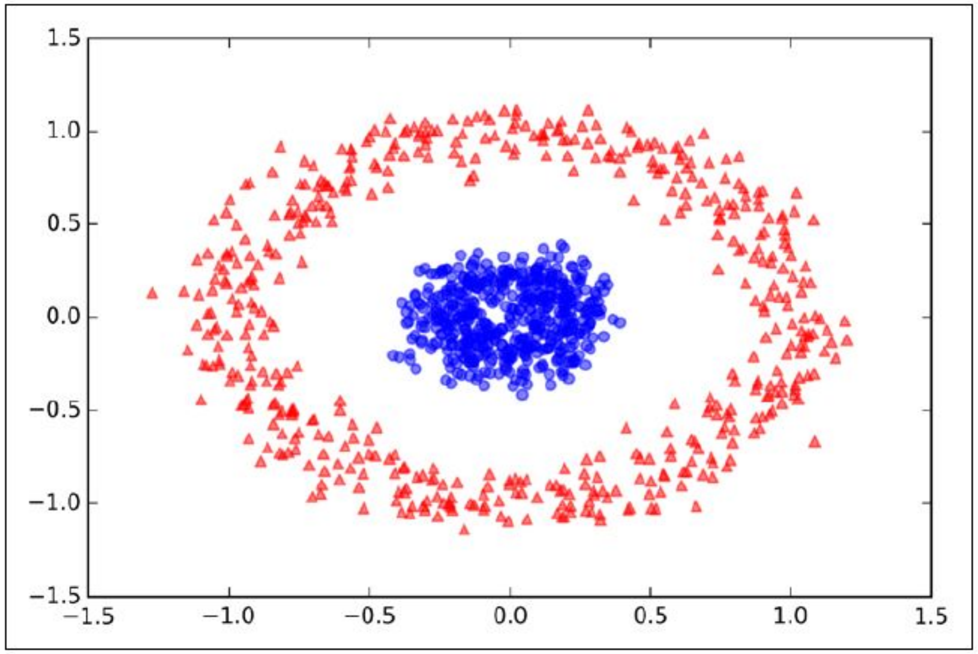
```
>>> X_train_pca = X_train_std.dot(w)
```

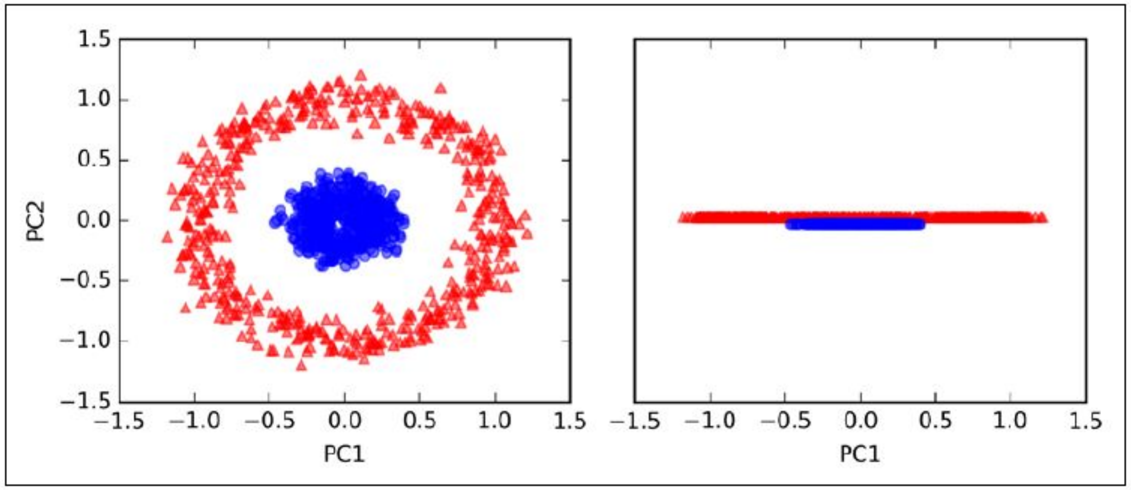
```
>>> colors = ['r', 'b', 'g']
>>> markers = ['s', 'x', 'o']
>>> for l, c, m in zip(np.unique(y_train), colors, markers):
...     plt.scatter(X_train_pca[y_train==l, 0],
...                 X_train_pca[y_train==l, 1],
...                 c=c, label=l, marker=m)
>>> plt.xlabel('PC 1')
>>> plt.ylabel('PC 2')
>>> plt.legend(loc='lower left')
>>> plt.show()
```


PC 1
PC 2
1
2
3
−6
−4
−2
0
2
4
6
4
3
2
1
0
−1
−2
−3
−4
−5

Problem of PCA

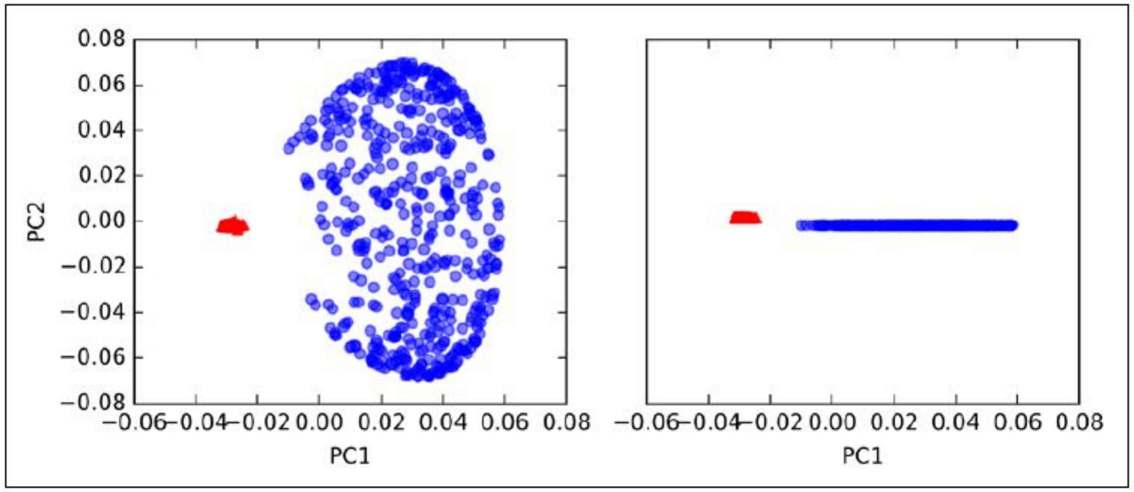
- Dataset is not linearly separable




PC2
PC1
PC1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5

Kernel PCA

- Motivation: Transform features with nonlinear transformation (kernel) functions
- Approach:
  - Apply the *kernel trick* to the PCA linear transformation function
  - Use the revised function to
    - Formulate the revised PCA optimization problem
    - Find the solution


0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
PC2
−0.06 −0.04 −0.02 0.00 0.02 0.04 0.06 0.08
PC1
−0.06 −0.04 −0.02 0.00 0.02 0.04 0.06 0.08
PC1

Limitations of PCA

- No missing values
- Only continuous features