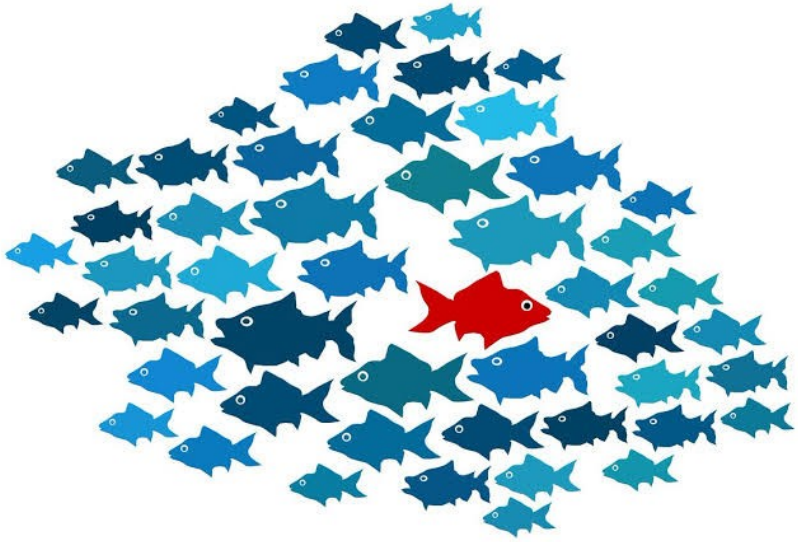
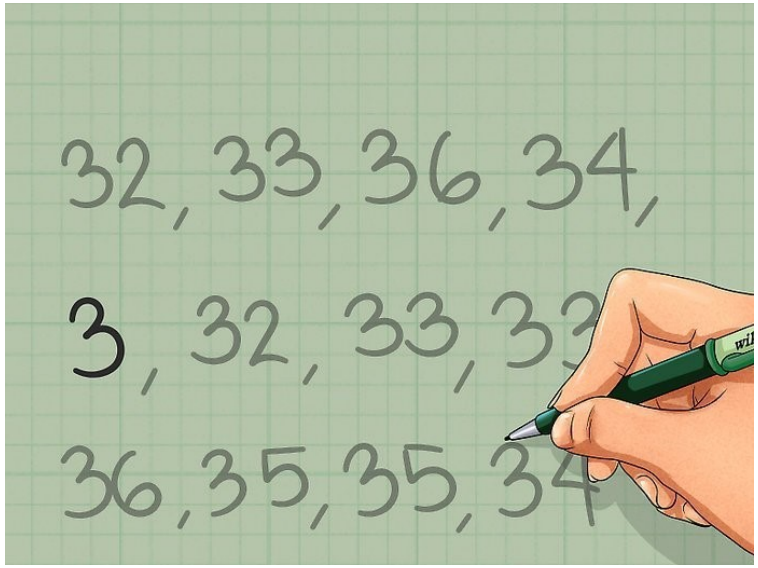
Outliers or Anomaly Detection using Python

Tushar B. Kute,
http://tusharkute.com

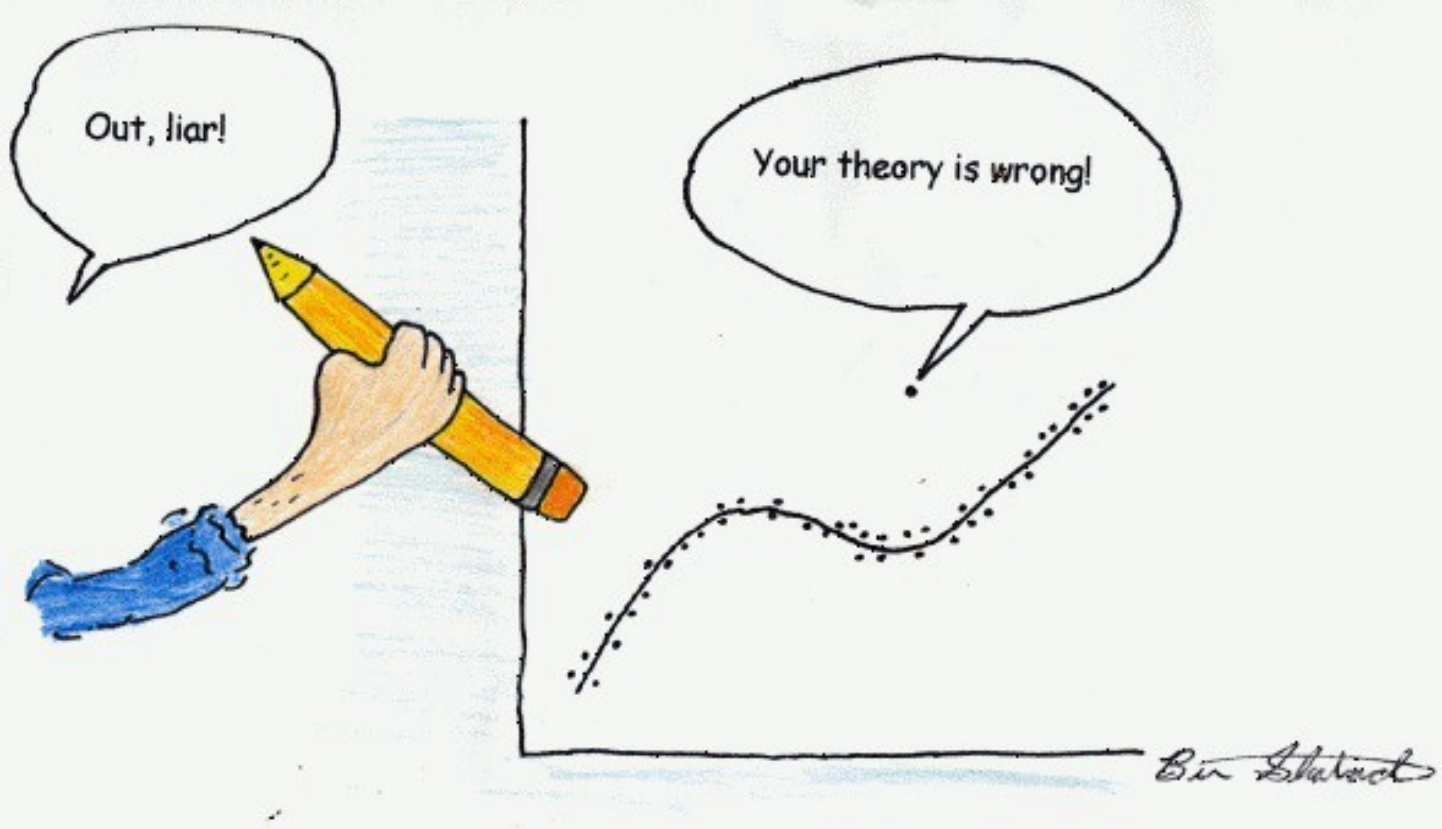


Outlier ?

- In statistics, an outlier is an observation point that is distant from other observations.
- The above definition suggests that outlier is something which is separate/different from the crowd.



Outlier when plotted



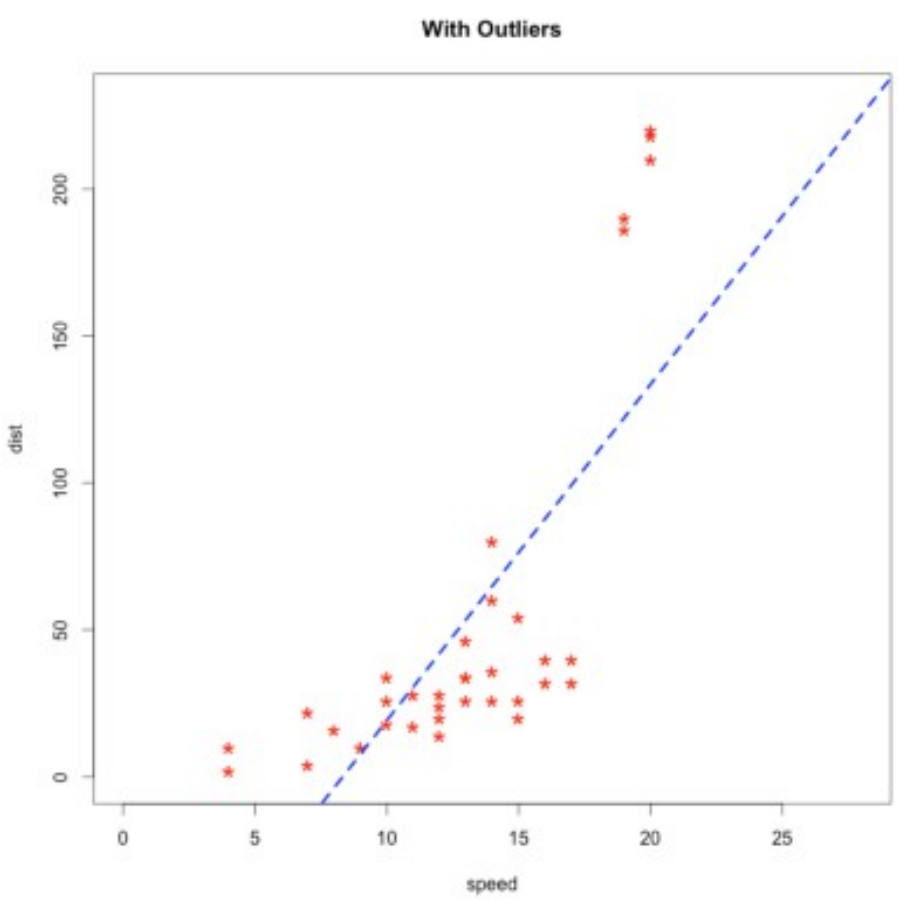
See through examples

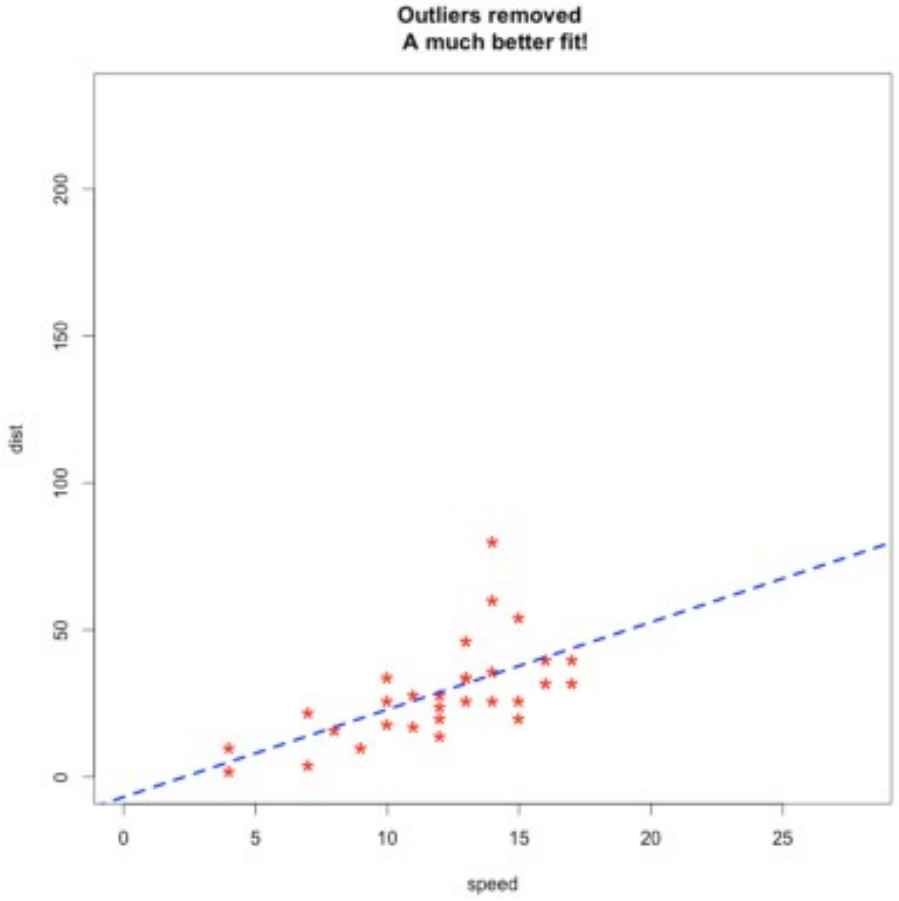
- An outlier is any data point which differs greatly from the rest of the observations in a dataset. Let's see some real life examples to understand outlier detection:
  - When one student averages over 90% while the rest of the class is at 70% – a clear outlier
  - While analyzing a certain customer's purchase patterns, it turns out there's suddenly an entry for a very high value. While most of his/her transactions fall below Rs. 10,000, this entry is for Rs. 1,00,000. It could be an electronic item purchase – whatever the reason, it's an outlier in the overall data
  - How about Usain Bolt? Those record breaking sprints are definitely outliers when you factor in the majority of athletes.

Outlier Types

- Outliers are of two types: Univariate and Multivariate.
- A univariate outlier is a data point that consists of extreme values in one variable only, whereas a multivariate outlier is a combined unusual score on at least two variables.
- Suppose you have three different variables – X, Y, Z. If you plot a graph of these in a 3-D space, they should form a sort of cloud.
- All the data points that lie outside this cloud will be the multivariate outliers.

Why to detect outliers?





What did they say ?

- *"Outliers are not necessarily a bad thing. These are just observations that are not following the same pattern as the other ones. But it can be the case that an outlier is very interesting. For example, if in a biological experiment, a rat is not dead whereas all others are, then it would be very interesting to understand why. This could lead to new scientific discoveries. So, it is important to detect outliers."*

  – Pierre Lafaye de Micheaux, Author and Statistician

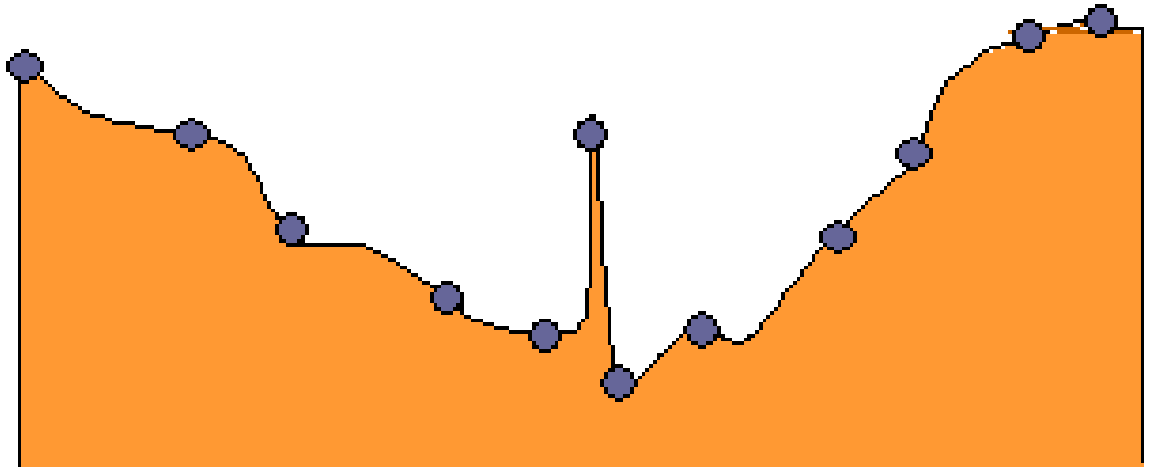
Prime Applications

- Fault diagnosis,
- Intrusion detection
- Fraud Detection
- Web analytics
- Medical diagnosis
- Financial industry
- Quality control

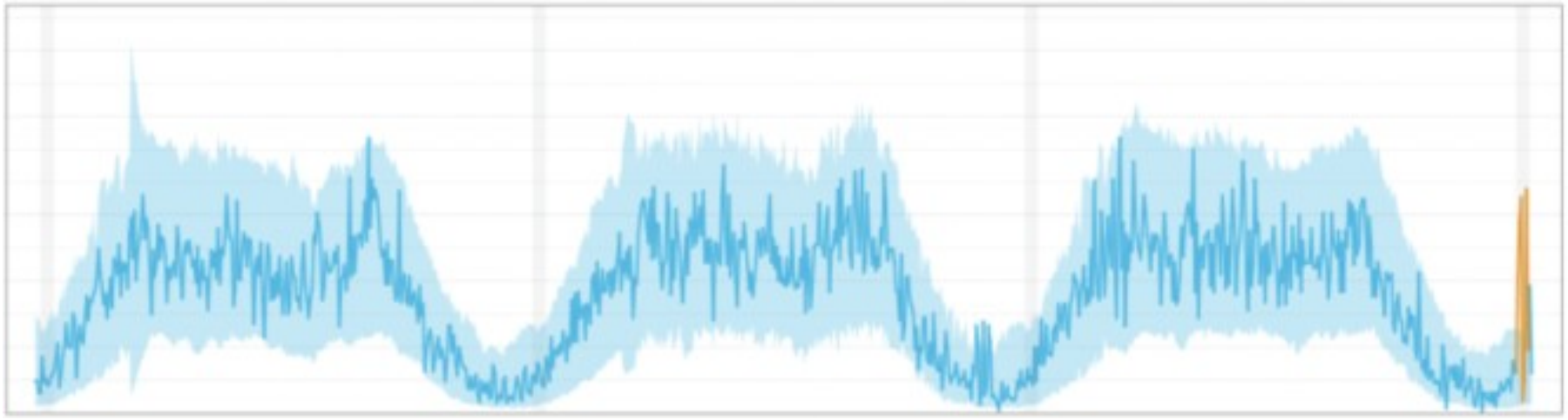
Types of Anomalies

- Global Anomalies
- Contextual Anomalies
- Collective Anomalies

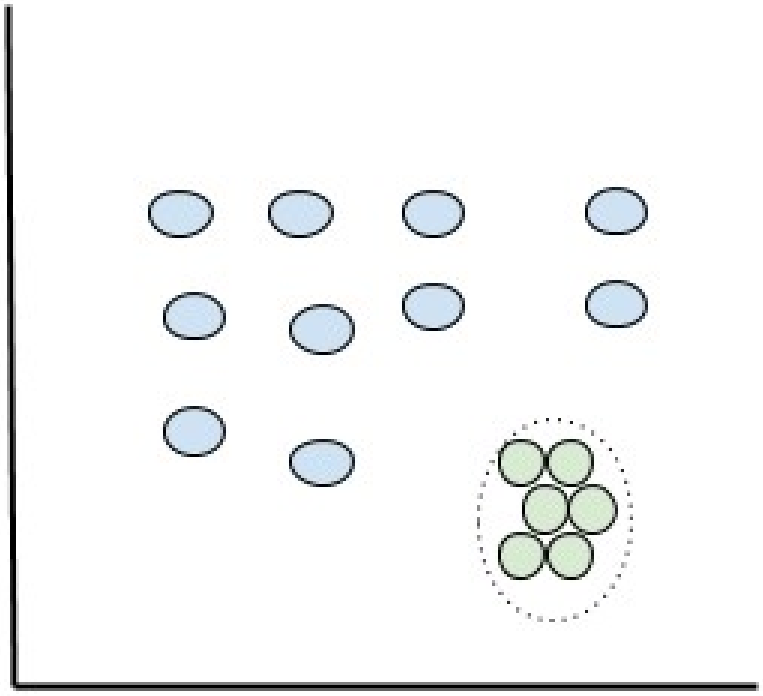
Global Anomalies



Contextual Anomalies



Collective Anomalies



General Methods for Detection

- Box Plot
- Histogram
- Clustering
- Isolation Forest

Packags needed

- Data Analytics:
  - pandas
- Numerical Python:
  - Numpy
  - scipy
- Random Number
  - faker
- Visualization
  - Matplotlib

Let's Start

```
# Import the necessary packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Use a predefined style set
plt.style.use('ggplot')

# Import Faker
from faker import Faker
fake = Faker()

# To ensure the results are reproducible
fake.seed(4321)
names_list = []
```

Create a random list

```
for _ in range(100):
  names_list.append(fake.name())

# To ensure the results are reproducible
np.random.seed(7)

salaries = []
for _ in range(100):
    salary = np.random.randint(1000,2500)
    salaries.append(salary)

# Create pandas DataFrame
salary_df = pd.DataFrame({'Person': names_list,
     'Salary': salaries  })
```

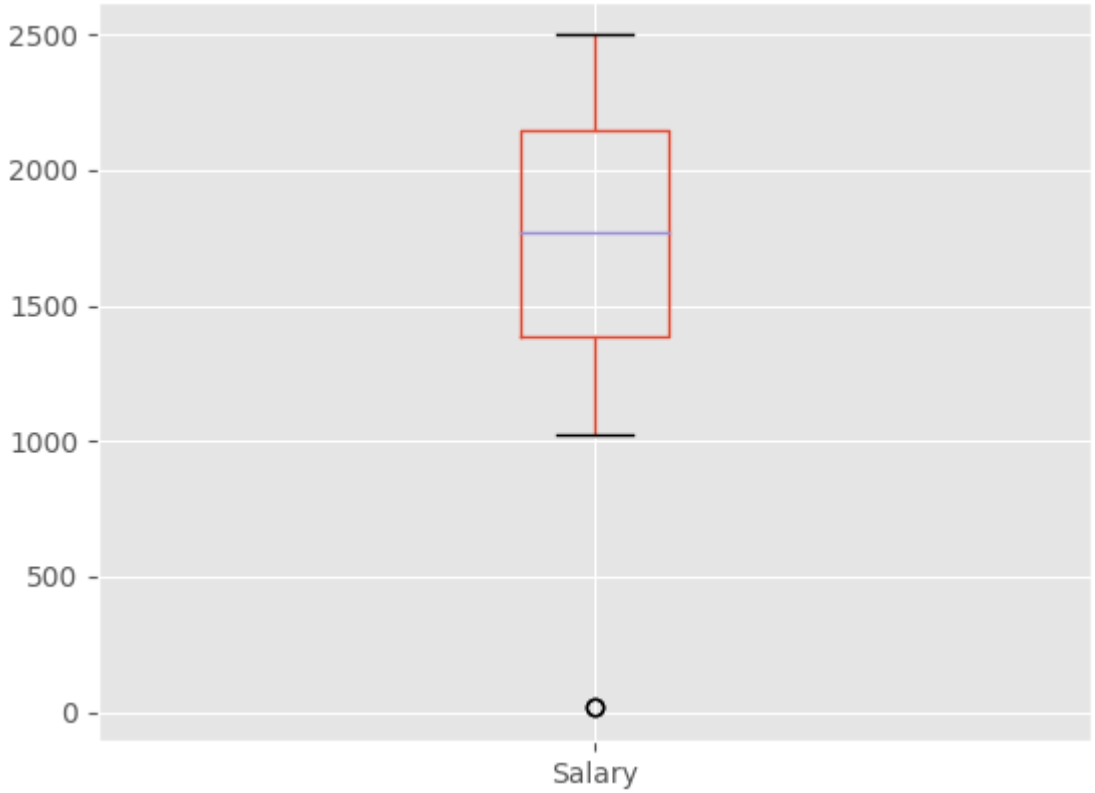
Add outliers and view

```
# Print a subsection of the DataFrame
print(salary_df.head())

salary_df.at[16, 'Salary'] = 23
salary_df.at[65, 'Salary'] = 17

# Verify if the salaries were changed
print(salary_df.loc[16])
print(salary_df.loc[65])

# Generate a Boxplot
salary_df['Salary'].plot(kind='box')
plt.show()
```

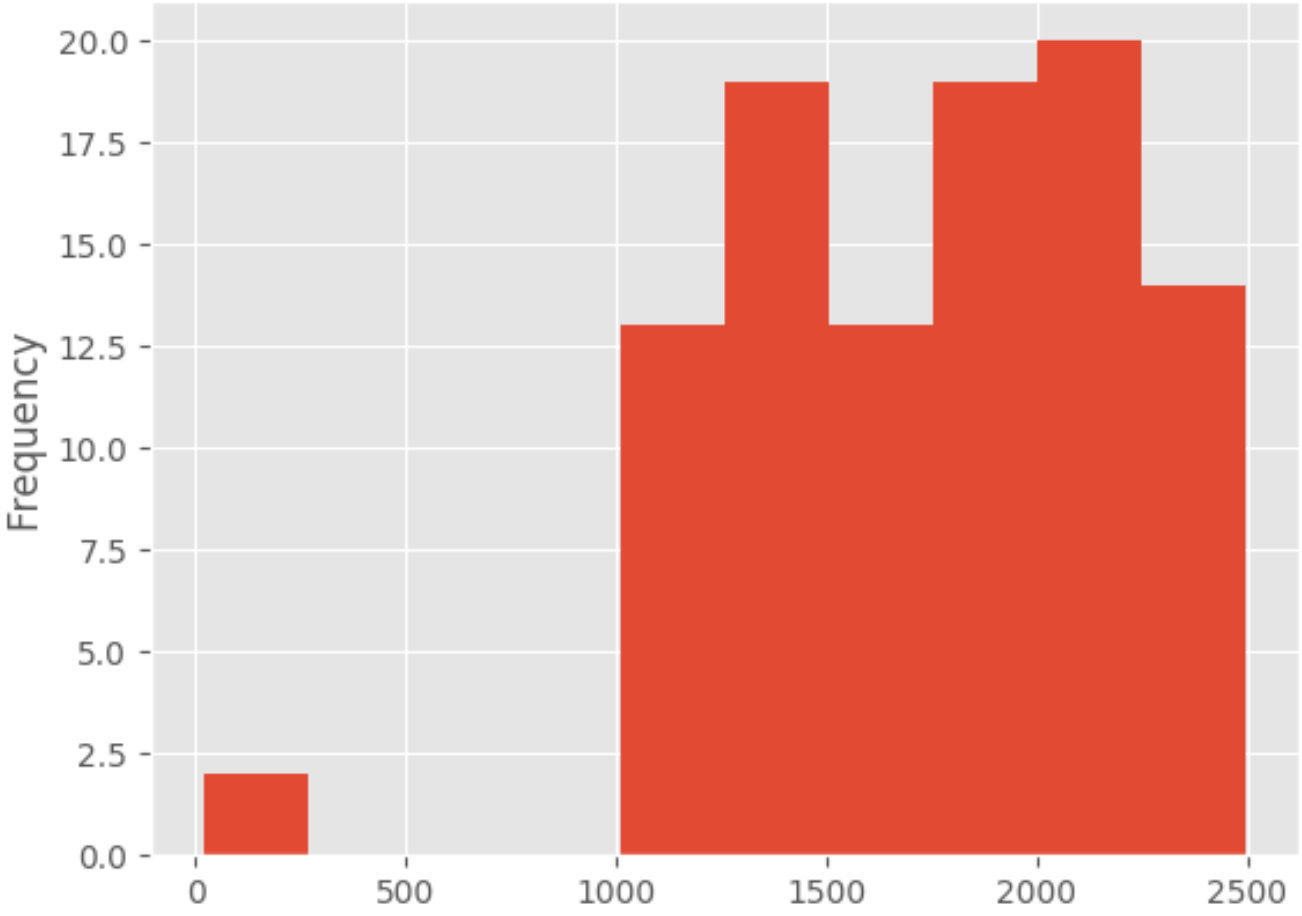
Check the outliers

```
# Generate a Boxplot
salary_df['Salary'].plot(kind='box')
plt.show()

# Generate a Histogram plot
salary_df['Salary'].plot(kind='hist')
plt.show()

# Minimum and maximum salaries
print('Min salary ' + str(salary_df['Salary'].min()))
print('Max salary ' + str(salary_df['Salary'].max()))
```

Boxplot



Histogram


Frequency
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
500
1000
1500
2000
2500

Using Clustering

- We are going to use K-Means clustering which will help us cluster the data points (salary values in our case).
- The implementation that we are going to be using for KMeans uses Euclidean distance internally. Let's get started.

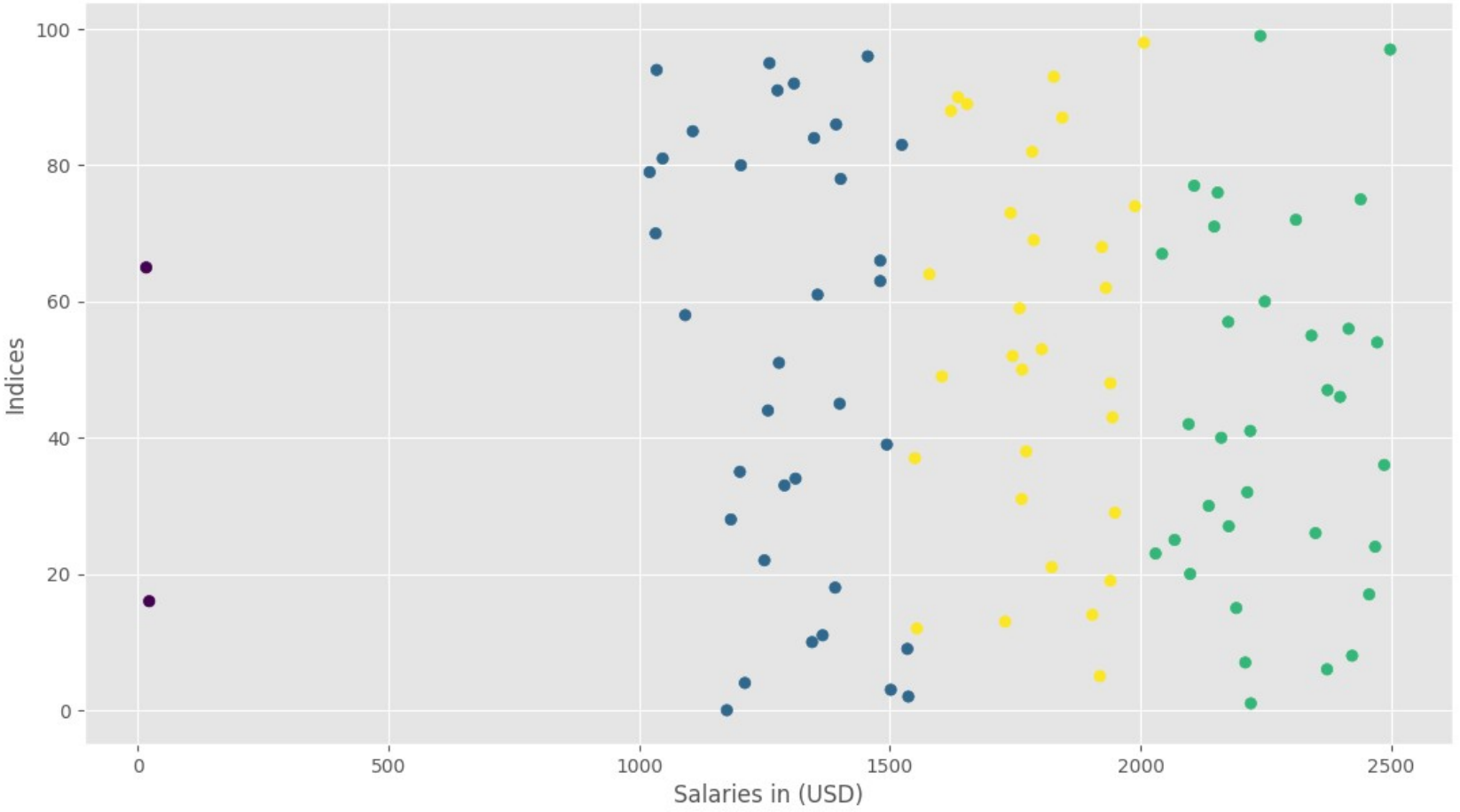
Getting Started

```
# Convert the salary values to a numpy array
salary_raw = salary_df['Salary'].values

# For compatibility with the SciPy
salary_raw = salary_raw.reshape(-1, 1)
salary_raw = salary_raw.astype('float64')

# Import kmeans from SciPy
from scipy.cluster.vq import kmeans
import scipy.cluster as cluster

# Specify the data, the no. of clusters
centroids, avg_distance = kmeans(salary_raw, 4)
```

Create the group and plot

```
# Get the groups (clusters) and distances
groups, cdist = cluster.vq.vq(salary_raw, centroids)

plt.scatter(salary_raw, np.arange(0,100), c=groups)
plt.xlabel('Salaries in (USD)')
plt.ylabel('Indices')
plt.show()
```

Outputs


100
80
60
40
20
0
Indices
0
500
1000
1500
2000
2500
Salaries in (USD)

Isolation Forests

- The Isolation Forest algorithm isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.
- The logic argument goes: isolating anomaly observations is easier because only a few conditions are needed to separate those cases from the normal observations. On the other hand, isolating normal observations require more conditions. Therefore, an anomaly score can be calculated as the number of conditions required to separate a given observation.
- The way that the algorithm constructs the separation is by first creating isolation trees, or random decision trees. Then, the score is calculated as the path length to isolate the observation.

How One Class SVM work ?

- The problem addressed by One Class SVM, as the documentation says, is novelty detection. The original paper describing how to use SVMs for this task is "Support Vector Method for Novelty Detection".
- The idea of novelty detection is to detect rare events, i.e. events that happen rarely, and hence, of which you have very little samples. The problem is then, that the usual way of training a classifier will not work.
- So how do you decide what a novel pattern is?. Many approaches are based on the estimation of the density of probability for the data. Novelty corresponds to those samples where the density of probability is "very low". How low depends on the application.
- Now, SVMs are max-margin methods, i.e. they do not model a probability distribution. Here the idea is to find a function that is positive for regions with high density of points, and negative for small densities.

Useful resources

- www.scikit-learn.org
- www.towardsdatascience.com
- www.medium.com
- www.analyticsvidhya.com
- www.depends-on-the-definition.com
- www.kaggle.com
- www.github.com

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
https://mituresearch.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com