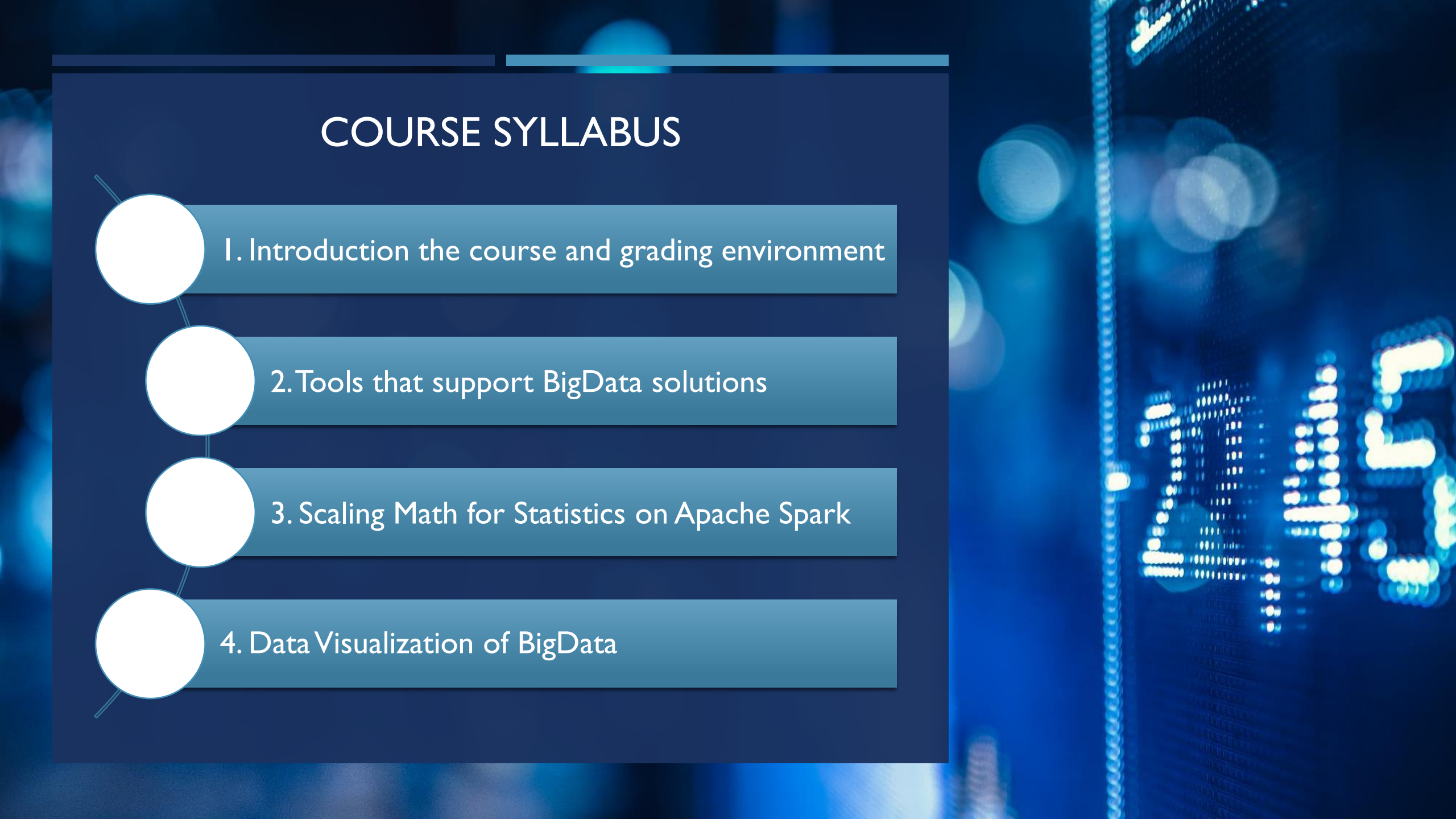
FUNDAMENTALS OF SCALABLE DATA SCIENCE

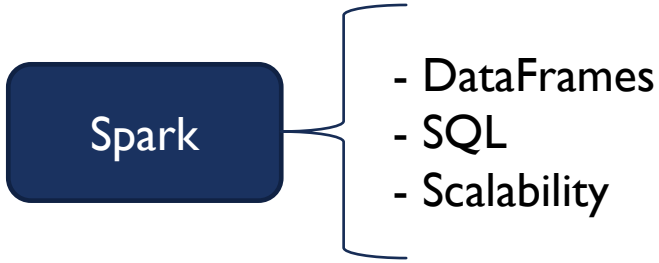
COURSE SYLLABUS

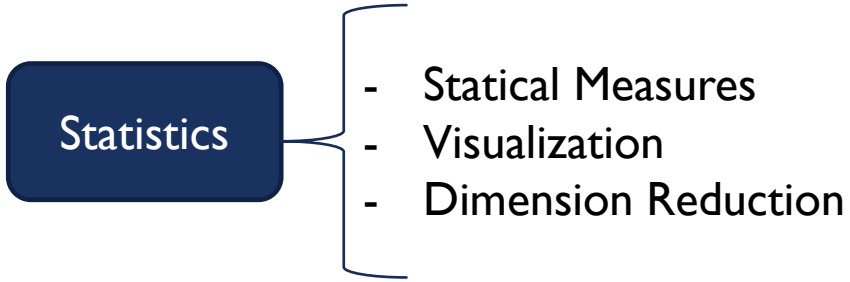
1. Introduction the course and grading environment
2. Tools that support BigData solutions
3. Scaling Math for Statistics on Apache Spark
4. Data Visualization of BigData



1. INTRODUCTION THE COURSE AND GRADING ENVIRONMENT

- First course of the IBM Advanced Data Science Certification
- Spark is a scalable architecture, we won't get any memory barriers
- Spark DataFrames are SQL compatible, so we can reuse our already existing SQL skills for scalable Data Science
- Basics of Statistics
- A view on Multidimensional data (Vector Spaces)





2. TOOLS THAT SUPPORT BIGDATA SOLUTIONS

- SQL
- NoSQL
- ObjectStorage

Summary

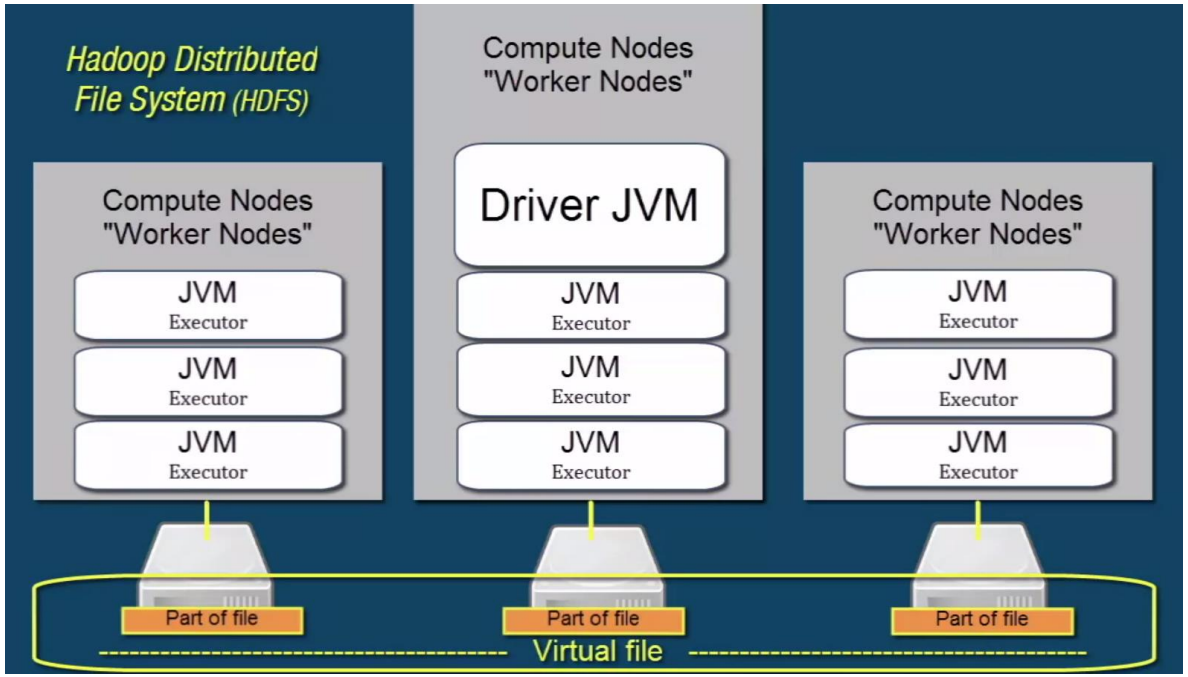
- Three main storage options
- Storage choice depends on:
  - Amount of data
  - Variety of schema
  - Query performance requirements
  - Additional data types like images, audio, or video

Data storage decision matrix

	SQL	NoSQL	ObjectStorage
Storage cost	high	low	very low
Scalability	low	high	very high
Query Speed	high	low	very low
Flexibility	low	high	**very high**

Limited to resources of a single node

JVM
Process
(running data analysis app)



RDDS- LANGUAGES – FUNCTIONAL PROGRAMMING – COUCH DB

Resilient Distributed Dataset (RDD)

- Distributed, immutable collection of data
- Created from HDFS, ObjectStore, Cloudant NoSQL, dashDB SQL, simple files, …
- In-memory, but spillable to disk
- Lazy

Summary

	Scala	Java	R	Python
Spark API support	complete	complete	very limited	limitted
Ease of use	low	very low	high	very high
Speed	very high	high	very low	low
3rd party libs	few	few	many	many

Summary

- Thousands of clients worldwide
- Ideal for storage of IOT sensor data
- Extraordinary high ingestion rates possible
- Storage cost is relatively low
- Dynamic scaling to cope with a changing resource demand

- CouchDB supports update installs with no downtime.

Functional Programming (FP)

- Mathematical concept of “Lambda Calculus"
- 1st implementation was LISP in the 50s
- Haskell is part of every computer science curriculum since the 90s
- Scala most recent representative
- Scala, Python, R and Java also rudimentary support FP

Cloudant is based on Open Source

- Founded in 2008 by MIT researchers
- Used to solve the "big data" problem of the "large hadron collider"
- "born in the cloud"
- Uses ApacheCouchDB NoSQL as it's core

- ApacheCouchDB was not scalable
- Cloudant implemented and donated BigCouch
- Multi-Version Concurrency Control (MVCC)
- ACID semantics
- Replicated, distributed architecture, offline support
- Eventual consistency guarantee

APACHE SPARK SQL

Schemas

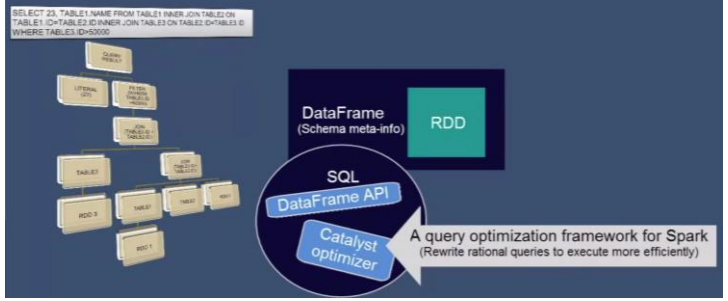
- RDDs are schema less (schema on read)
- DataFrames have a schema
  - --- implicit, lazy, inferred *(from a data read)*
  - --- explicitly defined

First Name	Surname	Address	Phone	Account
Mark	Schmit	53 Mountain view	123-456-7890	abc123
Aiden	White	66 Laurel drive	987-312-4567	abc321
zzz	aaaa	20 Woodmont	567-890-1234	efg444

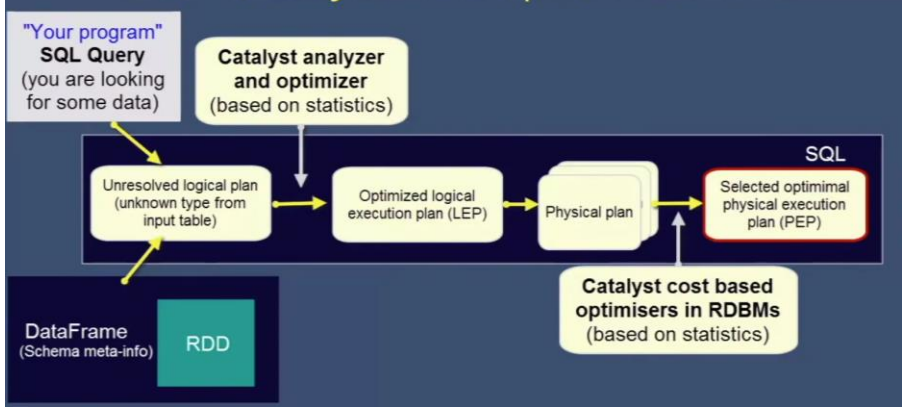
Summary

- ApacheSpark supports SQL via data frame API
- Internally still RDDs are used
- Makes writing ApacheSpark jobs easier
- Performance benefits through Catalyst & Tungsten

The "Catalyst" on top of the RDD



The "Catalyst" on top of the RDD



3. SCALING MATH FOR STATISTICS ON APACHE SPARK

1. Averages (mean/median)
2. Standard Deviation / variance
3. Skewness
4. Kurtosis
5. Covariance, covariance metrices & Correlation

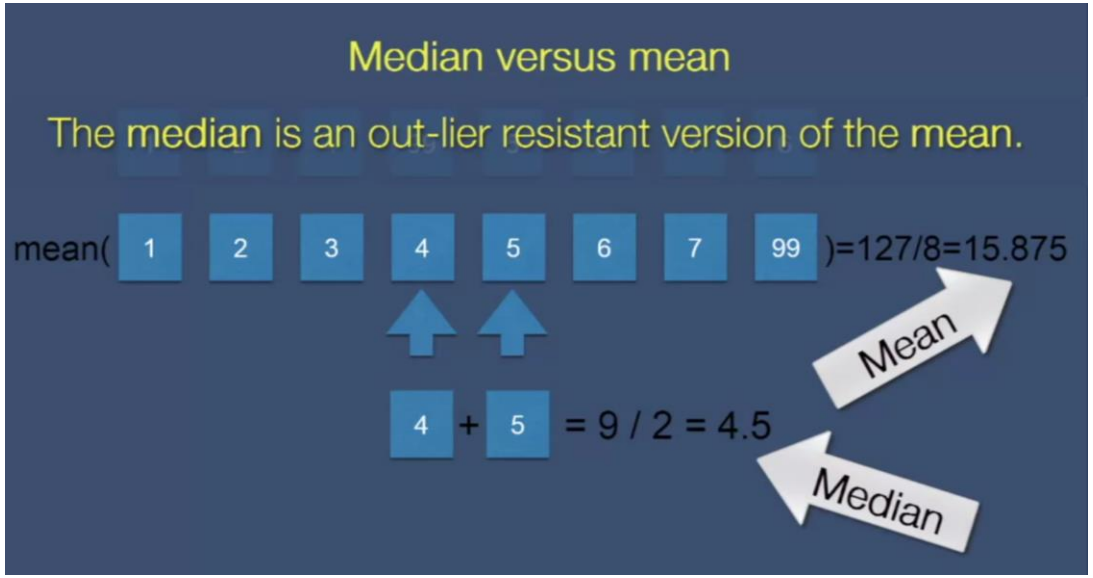
6. Multidimensional vector spaces

AVERAGES

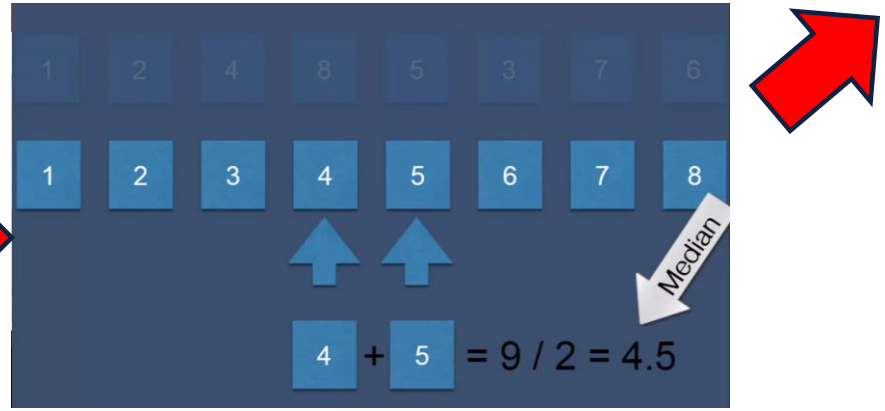
- The **Mean** is the average of all values in the data set

$$AM = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{1}{n}(a_1 + a_2 + \cdots + a_n)$$

Sum up all values and divide by the number of values.



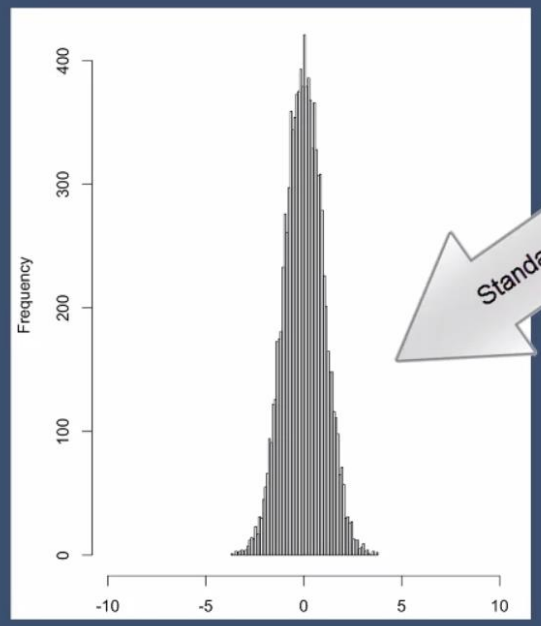
- The **Median** is the middle number in a sequence of numbers
  1. Sort the list
  2. Pick the middle element.



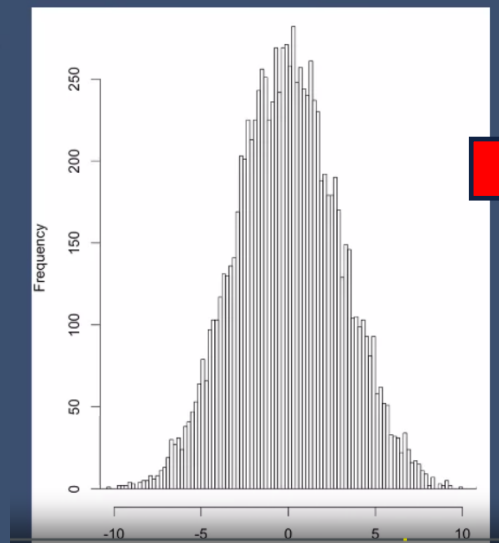
STANDARD DEVIATION

Histogram
Graph that represents data frequency distribution.

- How wide the data is spread around the mean



Standard deviation is low



$$s_N = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

VARIANCE

- Variance = standard deviation (σ) to the power of two

$$\left(\begin{matrix}\text{Standard}\\\text{Deviation}\end{matrix}\right) = \sqrt{\text{Variance}}$$

$$\sigma = \sqrt{\sigma^2}$$

NOTEBOOK - EXAMPLE

- 1° The mean of RDD
- 2° To subtract it from each value in this RDD:

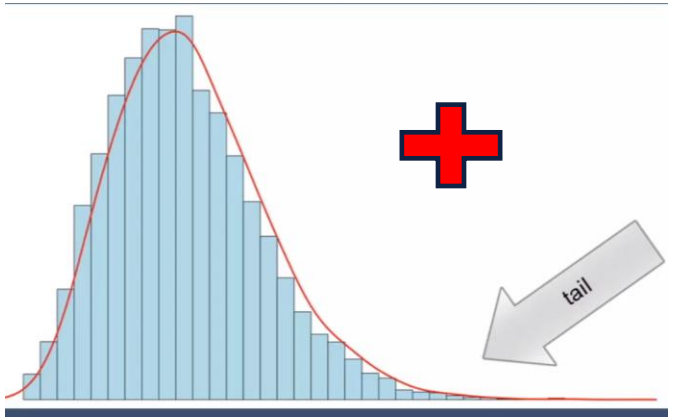
map(): to apply a function to each entry in an RDD, the math function is the one to choose.

```
In [18]: rdd = sc.parallelize(range(100))

In [19]: sum = rdd.sum()
         n = rdd.count()
         mean = sum/n
         print mean

         49

In [22]: from math import sqrt
         sqrt(rdd.map(lambda x : pow(x-mean,2)).sum()/n)

Out[22]: 28.861739379323623
```

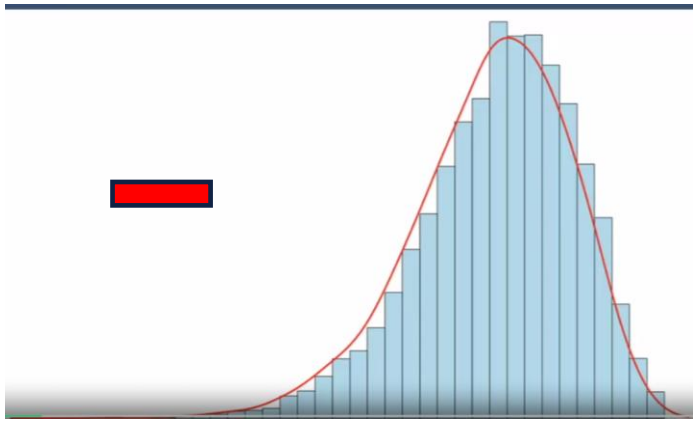
- Median is telling you an outlier resistant average of distances and
- Standard deviation tells you about the range where values are spread around.

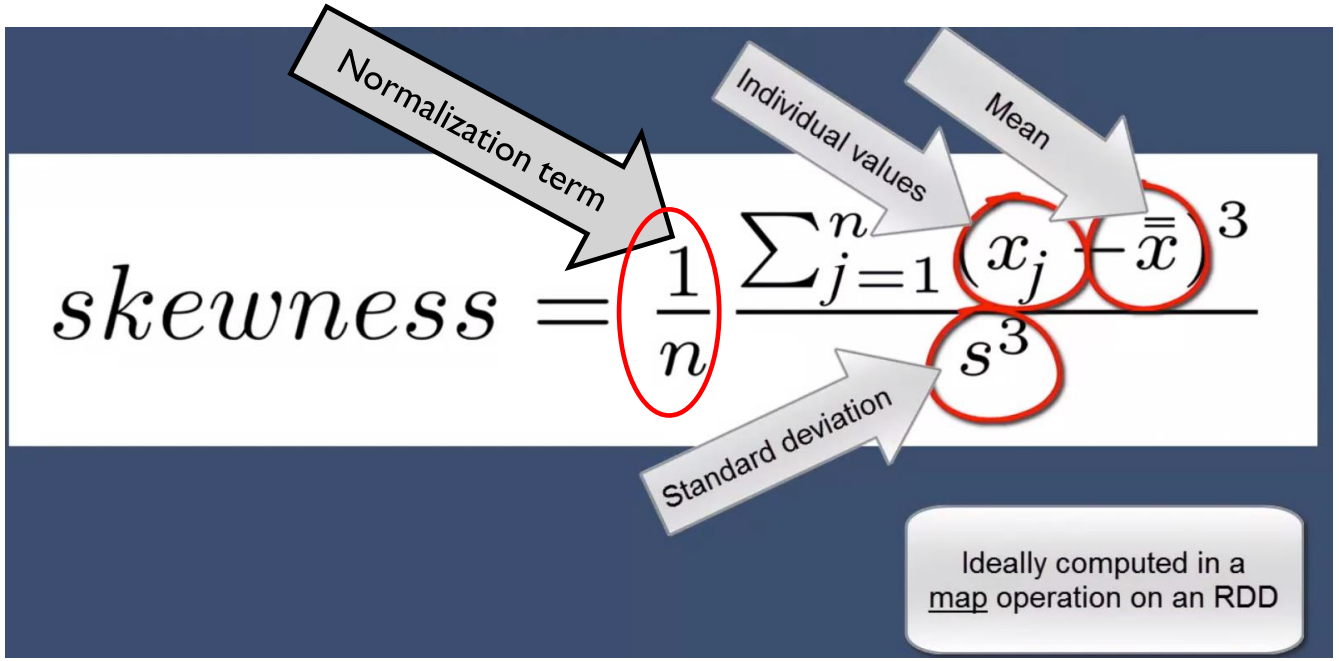
SKEWNESS – HOW ASYMMETRIC DATA IS SPREAD AROUND THE MEAN

- The shape and distribution of data around the mean.



Tail = Skew





NOTEBOOK EXAMPLE

In [109]:
```
sum = rdd.sum()
n = rdd.count()
mean = sum/n
print mean
```

```
49
```

In [110]:
```
from math import sqrt
sd = sqrt(rdd.map(lambda x : pow(x-mean,2)).sum()/n)
sd
```

Out[110]:
```
28.861739379323623
```

In [112]:
```
n = float(n)
skewness = n/((n-1)*(n-2))*rdd.map(lambda x : pow(x-mean,3)/pow(sd,3)).sum()
skewness
```

Out[112]:
```
0.05358968948395961
```

EXAMPLE

Consider a sensor measuring the amplitude of perpendicular vibrations on a car tire by sampling the distance between the wheel and suspension.

How can skewness be used in order to tell us about the healthiness of

the bearing where the wheel is attached to the suspension?

Note: If the bearing is intact vibrations tend to be symmetric. Therefore the amplitude of wheel vibrations should roughly be the same for each direction.

A high or low skew indicates asymmetric measurements (asymmetric around the mean)

Correct

An asymmetric distribution of values around the mean indicates also an asymmetric vibration pattern of the wheel. Therefore the bearing might be broken.

A skew around zero indicates asymmetric measurements (asymmetric around the mean)

Incorrect

An asymmetric distribution of values around the mean indicates also an asymmetric vibration pattern of the wheel. Therefore the bearing might be broken. But skew around zero indicates for a symmetric distribution of values.

KURTOSIS – THE LENGTH OF THE TAILS OF THE HISTOGRAM

- Reports on the shape of the data.
- Indicates outlier content within the data.
- The higher the kurtosis measure is, the more outliers are present

and the longer the tails of the distribution in the histogram are.

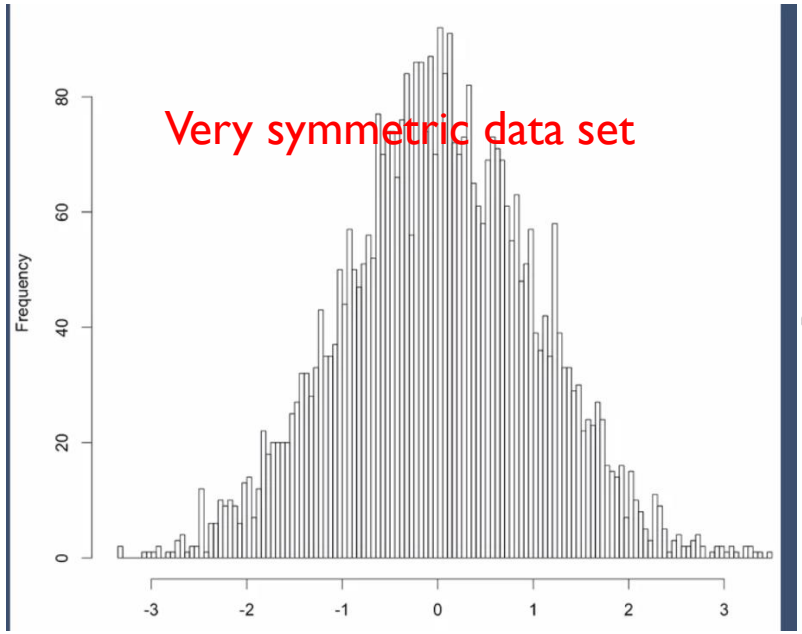
$$kurtosis = \frac{1}{n}\frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{s^4}$$

Individual values

Mean

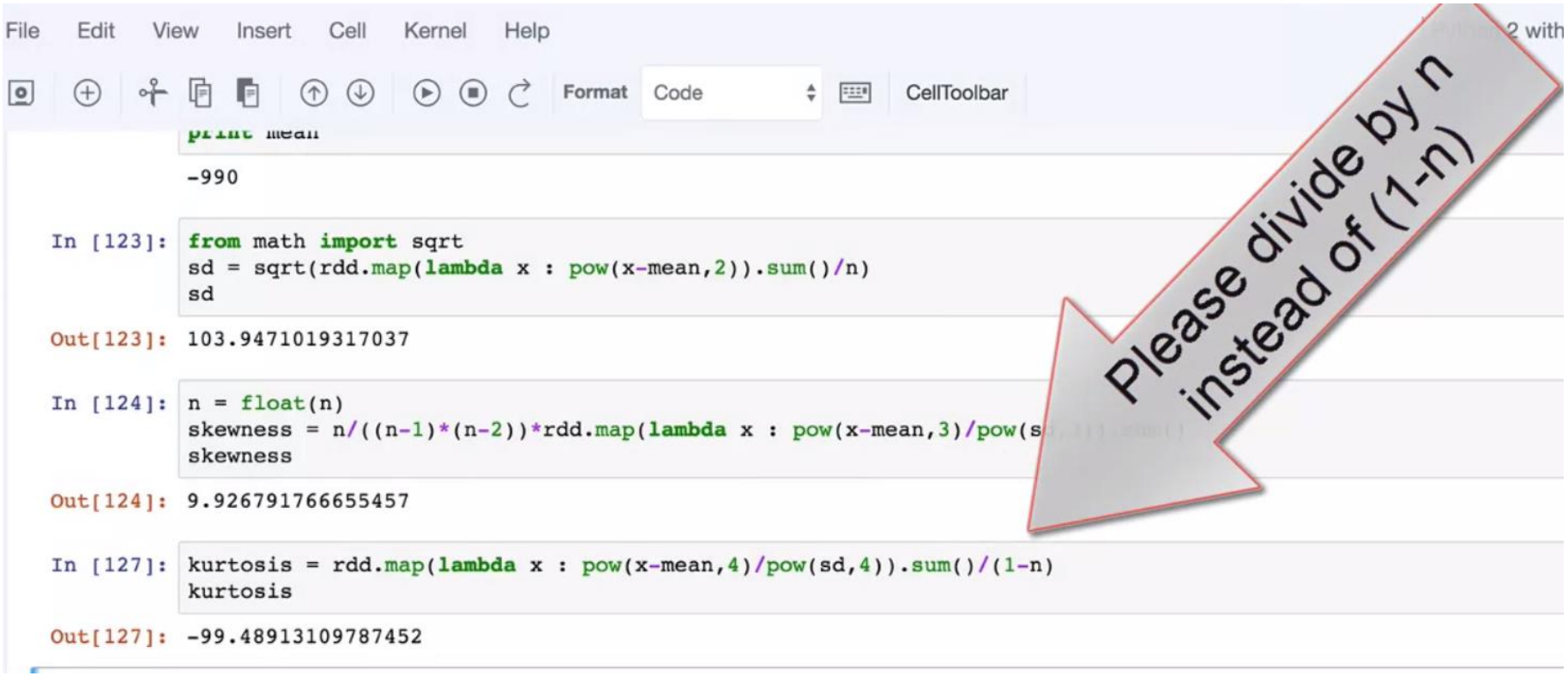
Normalization term

Standard Deviation



NOTEBOOK EXAMPLE

The mean and standard deviation we've already calculated previously, and start a standard deviation to a variable.



EXAMPLE

Consider an IoT system measuring fuel consumption of cars. The idea is to detect engine anomalies or potential failures using the collected fuel consumption of all cars of a specific type. If fuel consumption exceeds a certain threshold an alert should be generated. How can kurtosis be used to find a good threshold value?

Use Kurtosis to assess how far a threshold value should be away from the mean

Correct

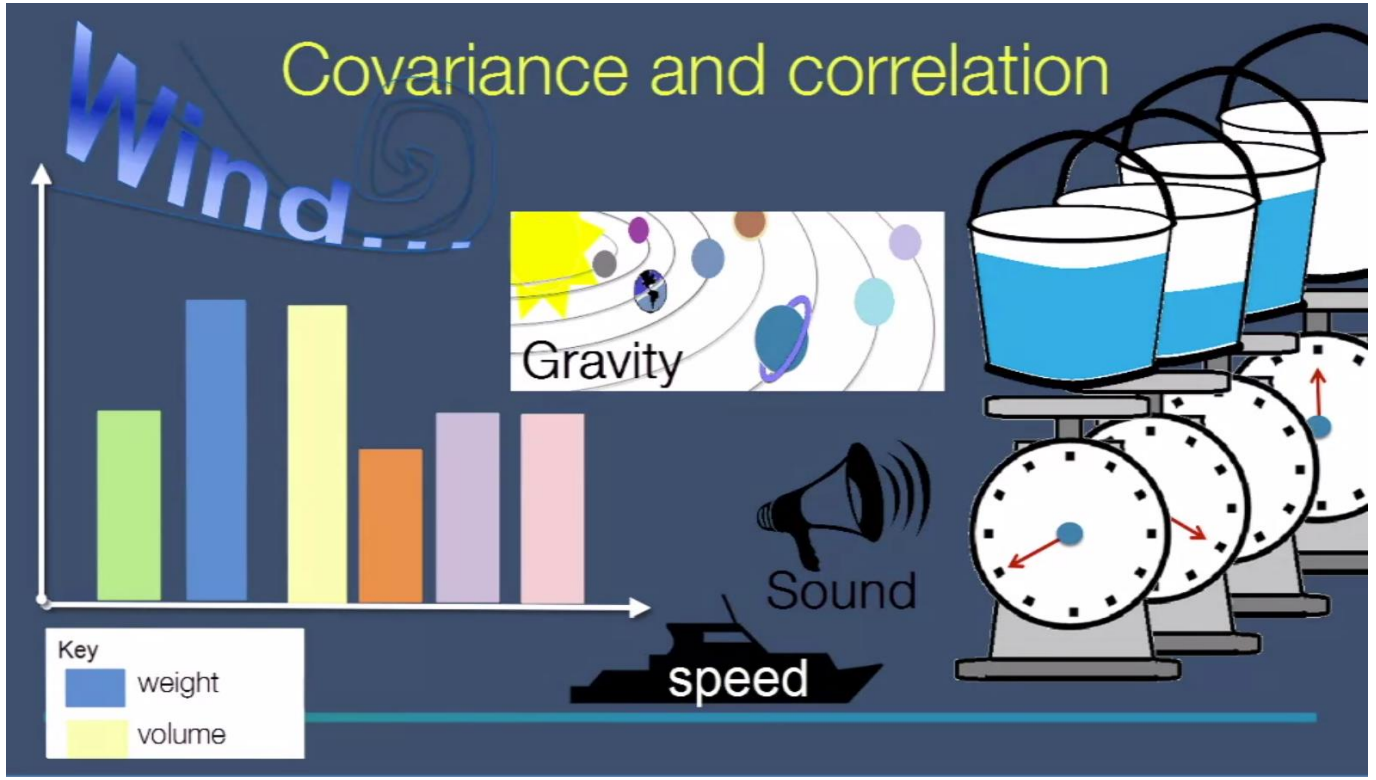
Kurtosis is a measure of outlier content. The more outliers are present in examples of fuel consumption where engines didn't fail, the higher the false positive rate gets when choosing a threshold value which is too low

Incorrect

he Kurtosis value itself is an optimal value to be used as threshold

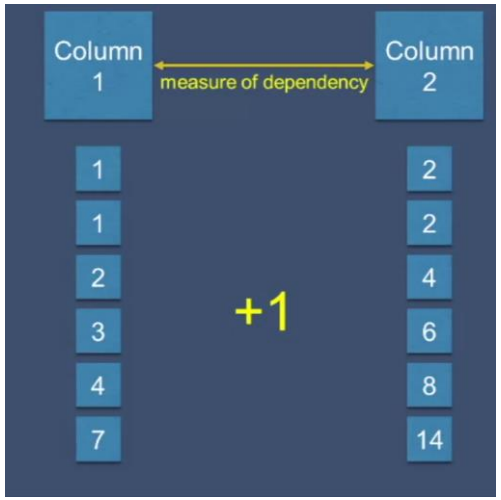
COVARIANCE AND CORRELATION

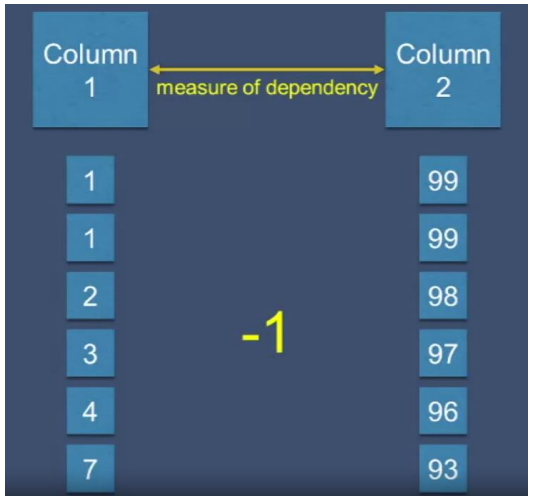
- How two columns interact with each other.
- How all columns interact with each other.

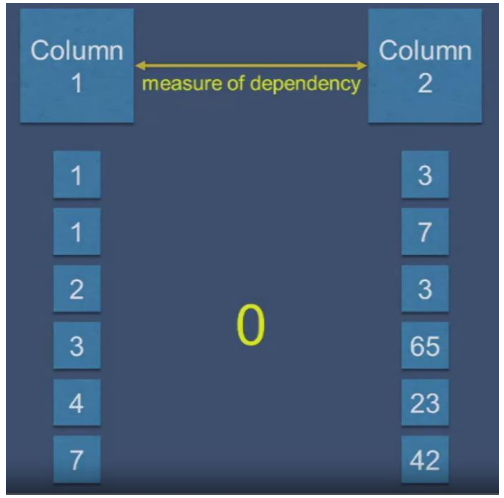


COVARIANCE AND CORRELATION

- Covariance explains interactions between columns
- Covariance and correlation basically is the same covariance matrix shows all column interactions







$$cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Covariance

$$corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Measure of dependency - Correlation

+1 Columns totally correlate
0 Columns show no interaction at all
-1 Inverse dependency

Standard Deviation

NOTEBOOK EXAMPLE

```
In [11]: rddX = sc.parallelize(range(100))
         rddY = sc.parallelize(range(100))

In [12]: meanX = rddX.sum()/rddX.count()
         meanY = rddY.sum()/rddY.count()
         print meanX
         print meanY

         49
         49

In [13]: rddXY = rddX.zip(rddY)
         rddXY.take(10)

Out[13]: [(0, 0),
          (1, 1),
          (2, 2),
          (3, 3),
          (4, 4),
          (5, 5),
          (6, 6),
          (7, 7),
```


```
In [11]: rddX = sc.parallelize(range(100))
         rddY = sc.parallelize(range(100))

In [12]: meanX = rddX.sum()/rddX.count()
         meanY = rddY.sum()/rddY.count()
         print meanX
         print meanY

         49
         49
```


```
In [16]: rddXY = rddX.zip(rddY)
         covXY = rddXY.map(lambda (x,y) : (x-meanX)*(y-meanY)).sum()/rddXY.count()
         covXY

Out[16]: 833
```

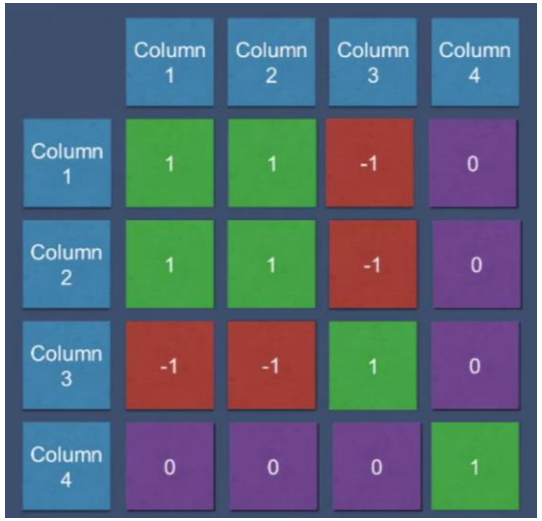
Measure of dependency - Correlation

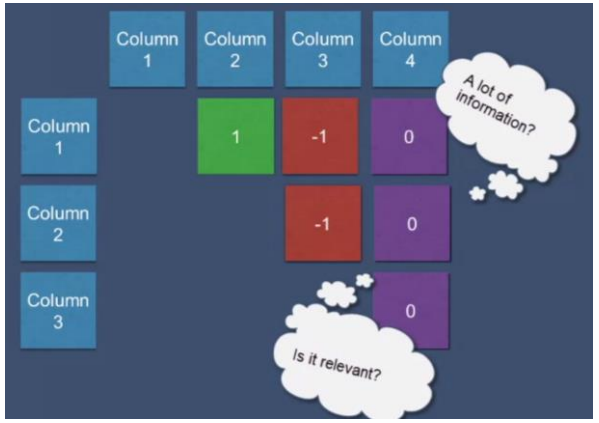
+1 Columns totally correlate
0 Columns show no interaction at all
-1 Inverse dependency

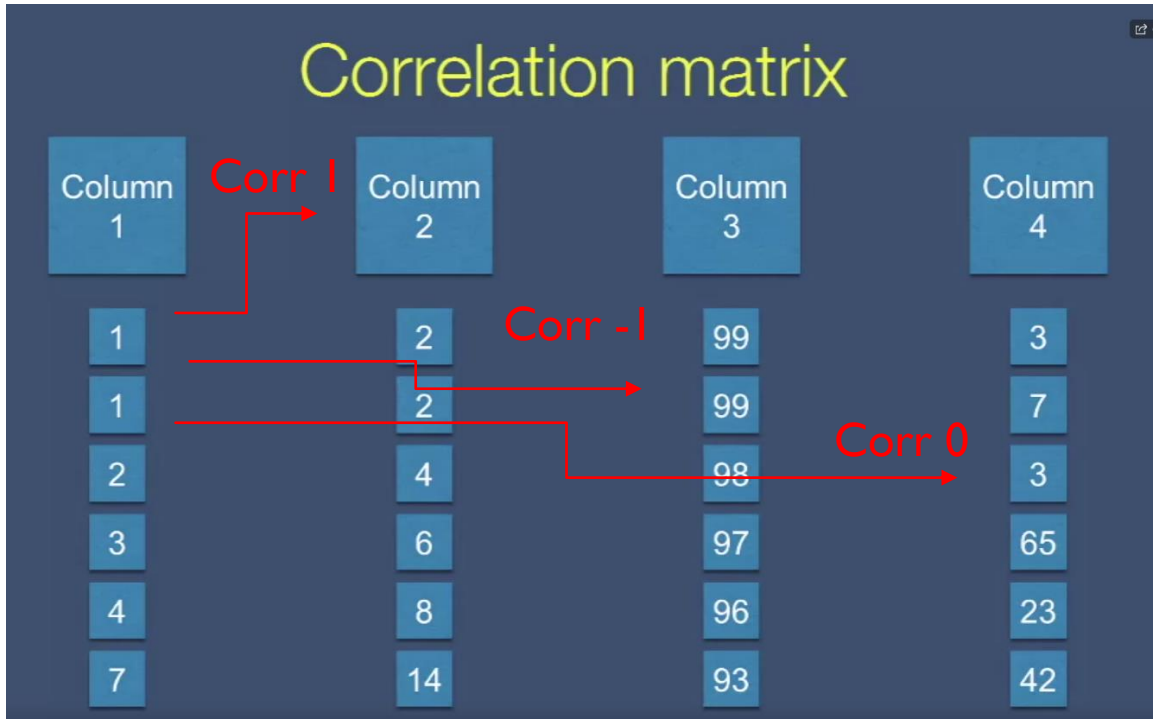
```
In [27]: from math import sqrt
         n = rddXY.count()
         sdX = sqrt(rddX.map(lambda x : pow(x-meanX,2)).sum()/n)
         sdY = sqrt(rddY.map(lambda x : pow(x-meanY,2)).sum()/n)
         print sdX
         print sdY

         28.8617393793
         28.8617393793

In [28]: corrXY = covXY / (sdX * sdY)
         corrXY

Out[28]: 1.0
```

CORRELATION MATRIX

- Covariance and correlation describe dependence between pairs of data
- What if we want to view all columns at once?





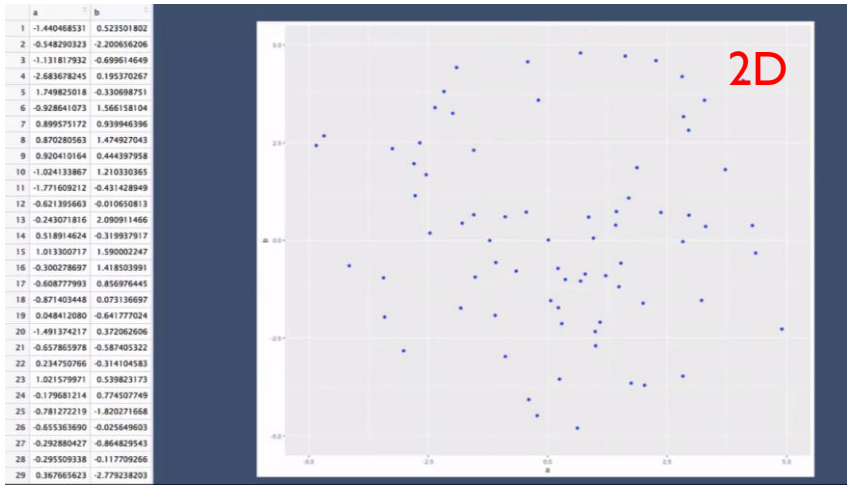


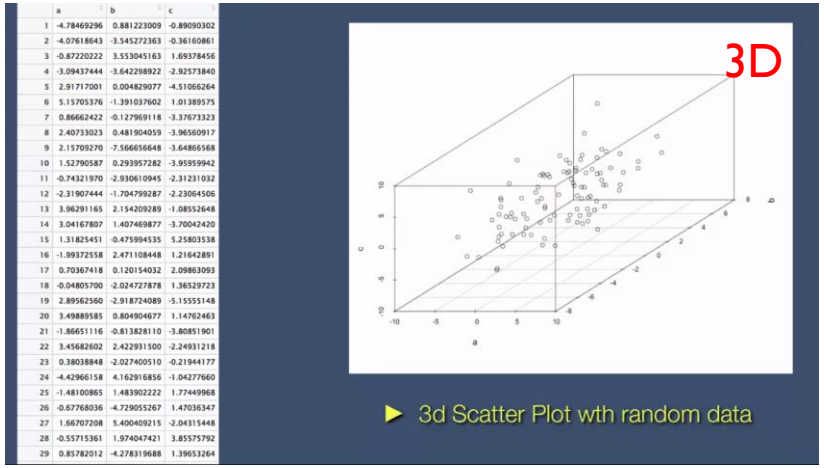
MULTIDIMENSIONAL VECTOR SPACES (2D & 3D)

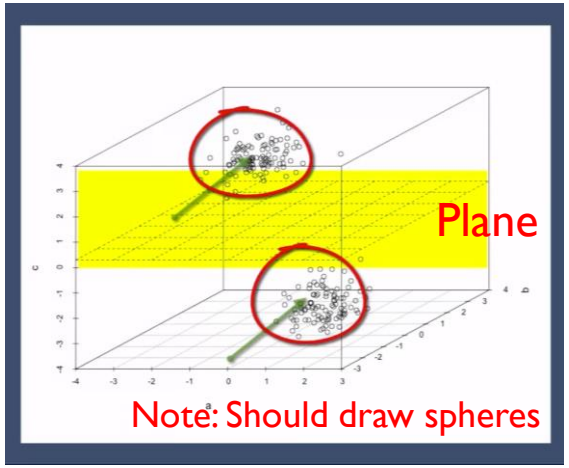
- We can treat any data set as points in a multidimensional vector space.

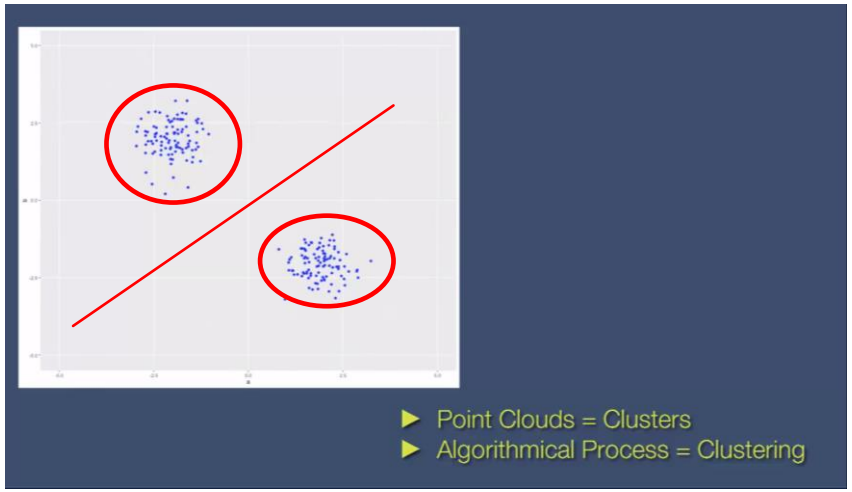
What if you have more than three ...

- ~~columns & tables~~
- dimensions - purely mathematical properties
- features - machine learning









MULTIDIMENSIONAL VECTOR SPACES (4D OR MORE)

What about additional dimensions?

- If low cardinaity:
  - color code data points or
  - use different shapes
- Plot interesting dimensions (remove irrelevant from plot)
- Create multiple plots with subsets
- Reduce dimensions using"dimensionality reduction"
- Use clustering and classification algorithms

Summary

- Low dimensional data can be easily plotted
- Use alternative methods with four or more dimensions
- Patterns in data are expressed as the shape of point distribution
- Points, circles, lines, spheres, planes, hyperespheres, and hyperplanes define centers and boundaries

4. DATA VISUALIZATION OF BIGDATA

1. Plot Diagrams of low dimensional datasets like Box Plot, Run Chart, Scatter Plot and Histograms
2. Draw conclusions out the diagrams you've plotted
3. Reduce dimensions of your dataset
4. Phase diagrams…

Multi-dimentional data transformation and basic visualization ***over time***

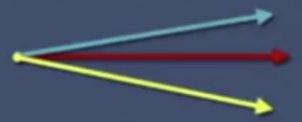
PLOTTING WITH APACHE SPARK AND PYTHON'S MATPLOTLIB

Plotting graphs ...

Depend on data size

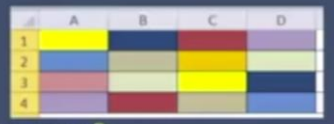
Plotting libraries run on a single machine

- Expects data that fits into memory



Too much data leads to...

Memory problems or Performance problems



- What to do if you have more data than memory?

Sampling

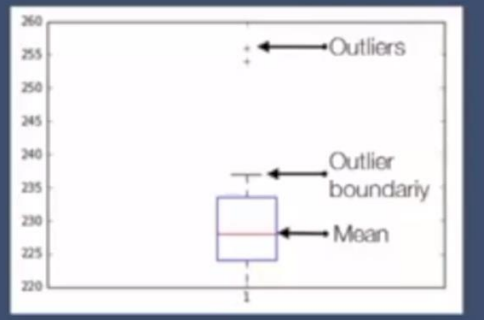
- What to do if you have more data than memory?
- Subset of original data
- Due to inherent randomness, preserves most properties of original data
- Reduces computational cost

Sampling reduces the size of the dataset by preserving most of the statistical properties necessary for plotting.

PLOTTING WITH APACHE SPARK AND PYTHON'S MATPLOTLIB

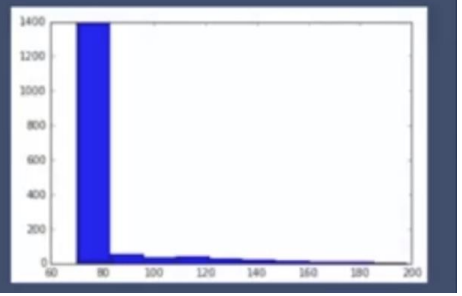
Box Plots

- Show many properties at the same time
  - mean
  - standard deviation
  - skew
  - outliers (kurtosis)
- Distribution of your data (vector)



Histograms

- Get an idea of the distribution of values
- Find regions of high and low value concentration



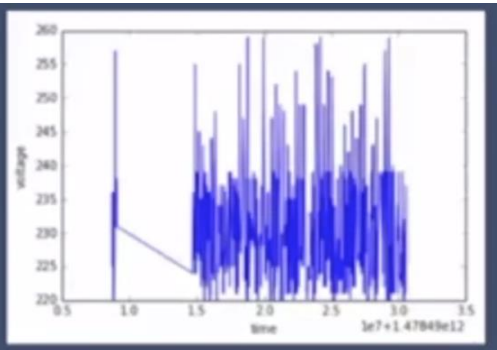
Scatter Plots

- Individual data points addressed by 2 or 3 dimensions
- Classification boundaries
- Clusters
- Anomalies

Run charts

- The ones you know from stock market
- Perfectly suited for time series
- x-axis always is time
- y-axis observed value over time

Serie Temporal



DIMENSIONALITY REDUCTION

- Sometimes becomes hard to choose the correct dimensions for plotting, so we will see how an algorithm can do this for us.

Principal Component Analysis

- Take an n - dimensional, euclidian vector space R^n
- Span a new k - dimensional, euclidian vector space R^k such that

$$\frac{dist(p_a^n,p_b^n)}{dist(p_c^n,p_d^n)} = \frac{dist(p_a^k,p_b^k)}{dist(p_c^k,p_d^k)}, p_a^n, p_b^n, p_c^n, p_d^n \in R^n, p_a^k, p_b^k, p_c^k, p_d^k \in R^k$$

- k dimensions "explain" most of the variation
- If k=3, plottable and distance ratios are preserved

Information Loss

- PCA is loose full compression
- The lower k is, the higher the loss
- Loss is measured by reconstruction error

$$loss = sse(D, PCA^{-1}(PCA(D)))$$

$$sse(X,Y) = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}, X = X_1, X_2, .., X_N, X_n = x_n^1, x_n^2, .., x_n^d$$

PCA removes information of dimensions which have... a high correlation with each other. ...because information content of highly correlated dimension is lower as with weakly correlated ones.

PCA - NOTEBOOK EXAMPLE

```
In [130]: result = spark.sql("""
          SELECT * from (
              SELECT
              min(temperature) over w as min_temperature,
              max(temperature) over w as max_temperature,
              min(voltage) over w as min_voltage,
              max(voltage) over w as max_voltage,
              min(flowrate) over w as min_flowrate,
              max(flowrate) over w as max_flowrate,
              min(frequency) over w as min_frequency,
              max(frequency) over w as max_frequency,
              min(hardness) over w as min_hardness,
              max(hardness) over w as max_hardness,
              min(speed) over w as min_speed,
              max(speed) over w as max_speed
              FROM washingflat
              WINDOW w AS (ORDER BY ts ROWS BETWEEN CURRENT ROW AND 10 FOLLOWING)
          )
          WHERE min_temperature is not null
          AND max_temperature is not null
          AND min_voltage is not null
          AND max_voltage is not null
```

Temperature, Voltage, Flowrate, Frequency, Hardness, and Speed.

More than 3D!

```
)
WHERE min_temperature is not null
AND max_temperature is not null
AND min_voltage is not null
AND max_voltage is not null
AND min_flowrate is not null
AND max_flowrate is not null
AND min_frequency is not null
AND max_frequency is not null
AND min_hardness is not null
AND min_speed is not null
AND max_speed is not null
""")
```

```
In [160]: result.count()

Out[160]: 27723

In [134]: from pyspark.ml.feature import PCA
          from pyspark.ml.linalg import Vectors
          from pyspark.ml.feature import VectorAssembler

In [135]: assembler = VectorAssembler(inputCols=result.columns, outputCol="features")

In [136]: features = assembler.transform(result)

In [ ]: features.rdd.map(lambda r : r.features).take(10)

In [138]: pca = PCA(k=3, inputCol="features", outputCol="pcaFeatures")
          model = pca.fit(features)

In [ ]: result_pca = model.transform(features).select("pcaFeatures")
        result_pca.show(truncate=False)
```

So in the end we have a list of DenseVectors containing our dimensionality reduced dataset.

SUMMARY

1

- Low dimensional data can be plotted using charts
- Depending on the chart types used, these properties can be visually observed
  - mean, variance, skew
  - clustering, outliers and means of separation
- Although each plot has a clear purpose…
- …often randomly playing with plots does the job best

Note: PCA requires denseVectors!!

2

- Dimensions of high dimensional data can be reduced
  - Preserving key properties
  - Making it feasible to plot
- Information does get lost
- The amount of loss can be quantified

**TO DETECT OUTLIERS

1. Median vs Mean.
2. Standard deviation - the range where values are spread around.
3. Skewness – how asymmetric data is spread around the mean.
4. Kurtosis - the length of the tails of the histogram. The higher the kurtosis measure is, the more outliers are present and the longer the tails of the distribution in the histogram are.

THANKS