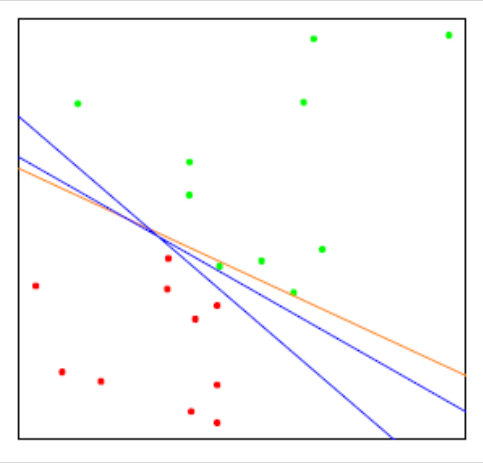


# Big Data Analytics

Basic concepts of analyzing very large amounts of data

**Dr. – Ing. Morris Riedel**
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE BDA6

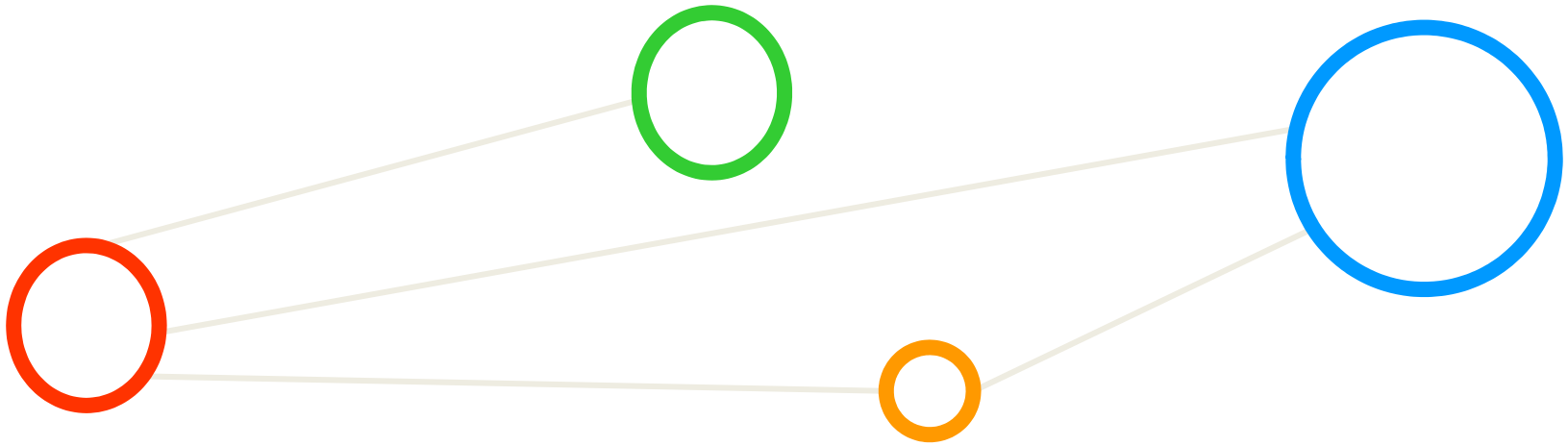
# Outlier Detection

2014-03-08 (v3)
Online Material

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

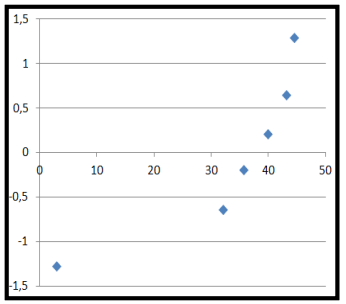
JÜLICH
FORSCHUNGSZENTRUM

# Outline

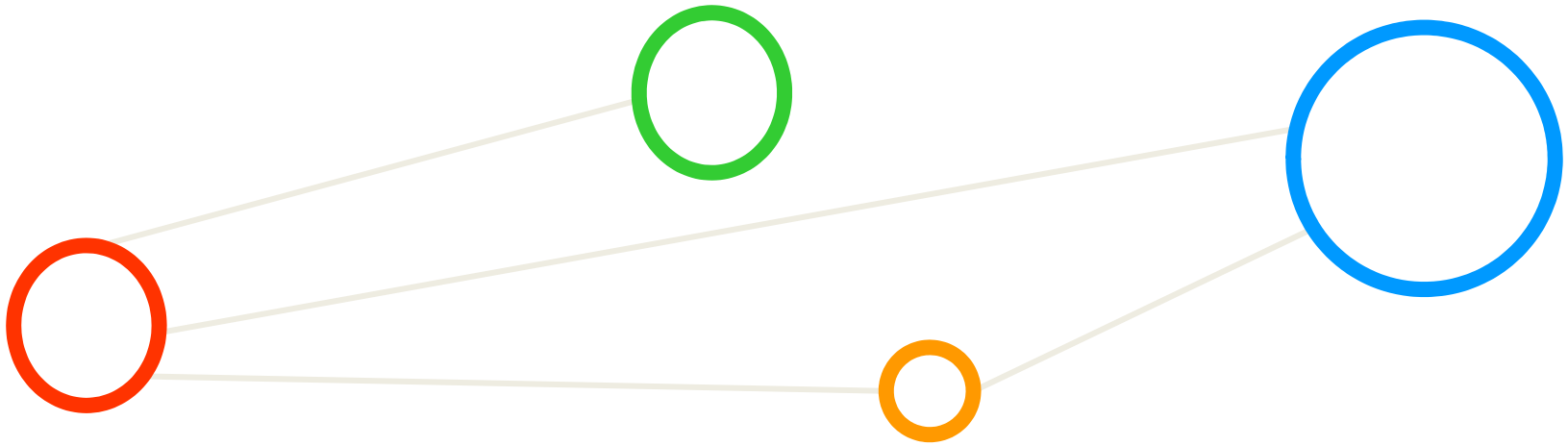


# Outline

- Outlier Detection
  - Understanding Outliers, Importance of Identification
  - Labeling, Accomodation, Identification
  - Normality Assumption, Normal Probability Plots
  - Graphical Tools & Step-wise Summary
- Algorithms & Techniques
  - Distance-based Approaches (e.g. K-Means)
  - Density-based Approaches (e.g. LOF)




# Outlier Detection



# Understanding Outliers

- **An Outlier is an observation that appears to deviate markedly from other observations in the sample**

***[1] Engineering Statistics Handbook***

- Outlier Detection
  - Part of exploratory analysis
  - E.g. used for quality control of data sets

# Importance of Identification

- Outliers may indicate ‘bad data‘
  - E.g. experiment did not run correctly
  - If truly erroneous: the outlying value should be deleted from the analysis (or corrected if possible)

- Outliers may be due to random variation
  - Indicates possibly something scientifically interesting
  - ‘Event’: Do not simply delete the outlying observation
  - If the data contains significant outliers: consider the use of robust statistical techniques

# Labeling, Accomodation, and Identification

- **Iglewicz and Hoaglin differentiate between three issues with respect to outliers: (1) Outlier Labeling; (2) Outlier Accomodation; (3) Outlier Identification**

***[2] Iglewicz and Hoaglin et al., 1993***

- Outlier Labeling
  - Flag potential outliers for further investigation
  - E.g. are the potential outliers erroneous data
  - E.g. indicative of an inappropriate distributional model
- Outlier Accomodation
  - Use robust statistical techniques that will not be unduly affected by outliers
  - If determining potential outliers are erroneous observations is not possible
    - ➢ Modify statistical analysis to more appropriately account for the observations
- Outlier Identification
    - Test whether observations are outliers

# Normality Assumption

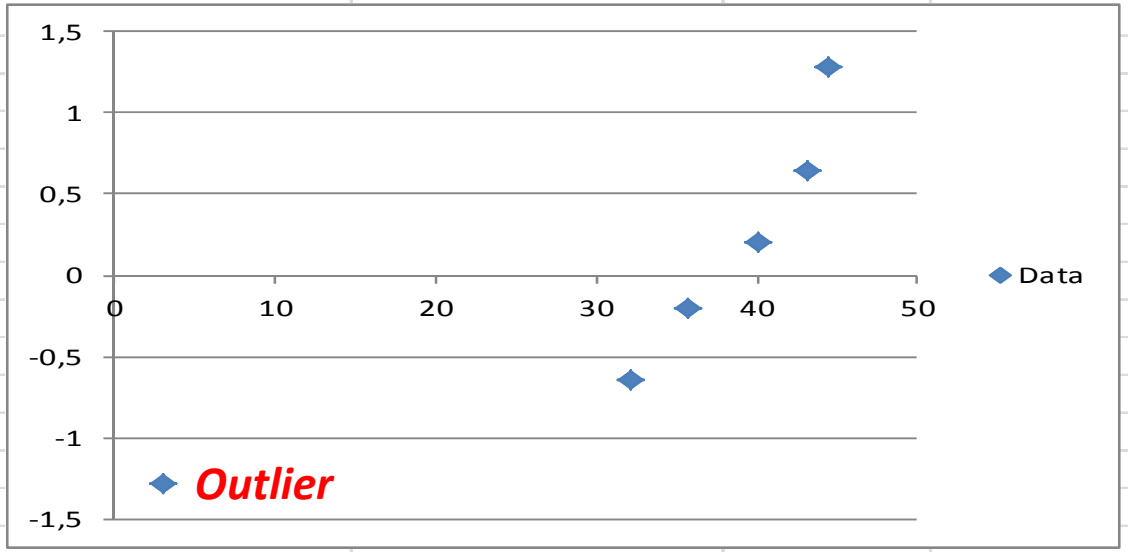
- **Identifying an observation as an outlier depends on the underlying distribution of the data**

***[1] Engineers Statistics Handbook***

- Univariate data sets
  - Assumed to follow an approximately normal distribution
- If 'normality assumption' for the data being tested is not valid
  - A determination that there is an outlier may be due to the non-normality of the data rather than the presence of an outlier
  - Generate a 'normal probability plot' of the data (before applying an outlier test)
  - Perform 'formal tests' for normality is possible: but the prescence of one or more outliers may cause the tests to reject normality (when it is a reasonable assumption for applying the outlier test)

# Normal Probability Plot

| X (Ascending Order) | Expected Z-Score | i (position) | fi |
|---|---|---|---|
| Outlier 3 | -1,281551566 | 1 | 0,1 |
| 32,1 | -0,643345405 | 2 | 0,26 |
| 35,7 | -0,201893479 | 3 | 0,42 |
| 40 | 0,201893479 | 4 | 0,58 |
| 43,2 | 0,643345405 | 5 | 0,74 |
| 44,5 | 1,281551566 | 6 | 0,9 |




***Modified from [3] Normal Probability Plot Example***

# Normal Probability Plot – Creation Steps

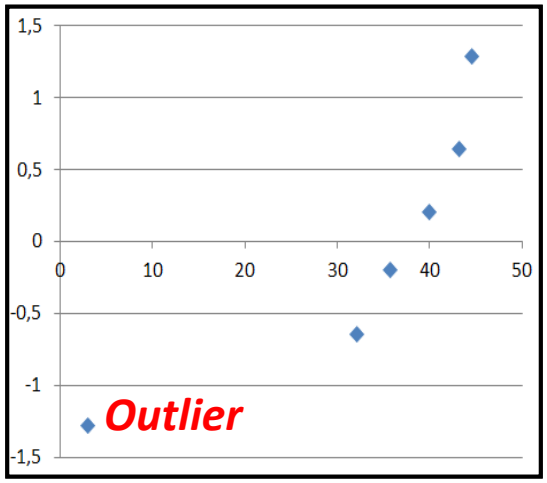
1. Sorting data (list) in ascending order
   - Mark each list item with position **i** in the data point list
2. Compute the formula
   - Where **i** is the position in the list
   - Where **n** is the number of observations

$$f_i = \frac{i - 0.375}{n + 0.25}$$

3. Find the z-score corresponding to $\mathbf{f_i}$ from the Standard Normal Distribution table
4. Plot the observed values on the horizontal axis
5. Plot the corresponding expected z-scores on the vertical axis

- **Note that values in tools can have different representations in different languages**
- **E.g. German: 44,5; English: 44.5; etc.**

# Normal Probability Plot – Interpretation



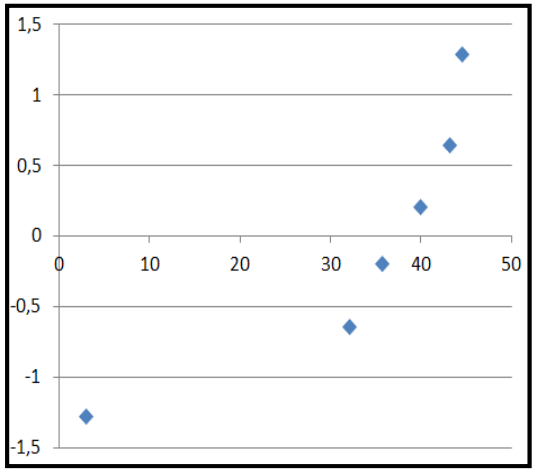

- Checking in general the normality assumption
- The lower and upper tails of the normal probability plot can be a useful graphical technique for identifying potential outliers
- The plot can help determine further actions
  - Whether a check for a single outlier is needed (cf. Figure)
  - Whether a check for multiple outliers is needed

# Graphical Tools Summary

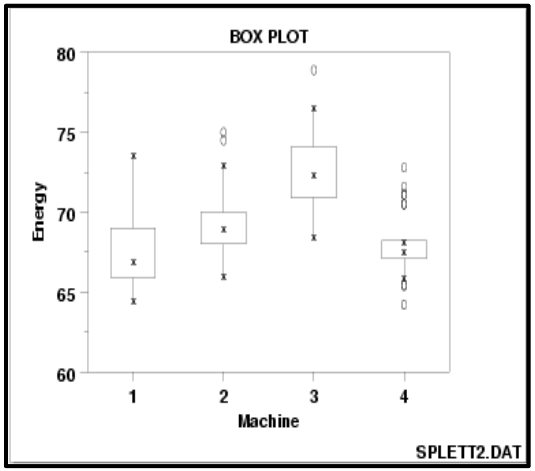
- **Graphical Tools can be useful tools in checking the normality assumption and identifying potential outliers in datasets**

***[1] Engineers Statistics Handbook***

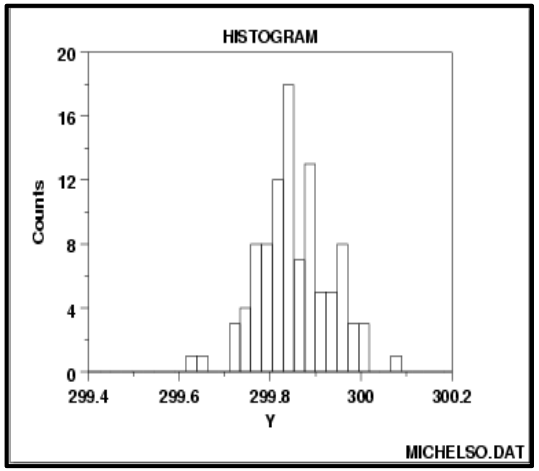
- Boxplots and Histograms can also be useful graphical tools in checking the normality assumption and identify potential outliers




Normal Probability Plot




Boxplot

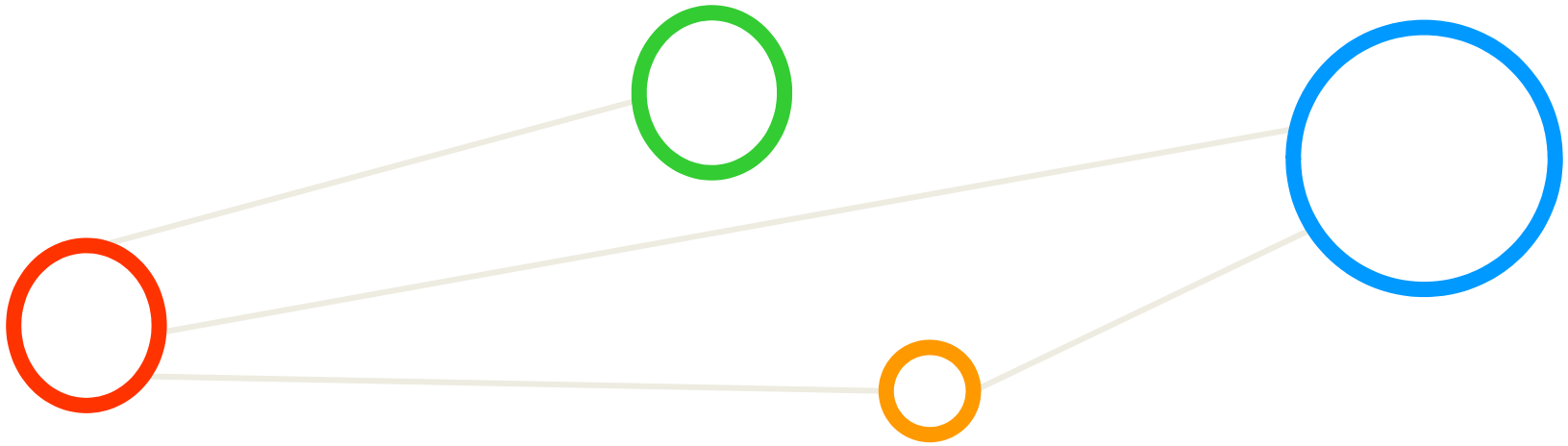



Histogram

# Step-Wise Summary

1. Generate a ‘normal probability plot’ of the data
   (before applying an outlier test)
2. Apply ‘parallel/distributed outlier detection algorithm‘ in case of ‘big data‘
   (but several algorithms are not suitable for large quanties of data)

# Algorithms & Techniques



# Distance-based – Naive Approach

- Takes quadratic time – $O(n^2)$
  - With respect to the number (n) of data points
  - Approach: Comparing each point (n) with the rest of data points (n-1)
- Selected existing algorithms
  - Use 'block nested loop algorithm' *[4] Knorr & Ng, 1998*

**$O(n^2 d)$**
***n : size of data set; d : dimensionality of data set***

  - Partition the dataset into 'cells that are hyper-rectangles' *[4] Knorr & Ng, 1998*

**$O(n d^2)$**
***by pruning rectangles early***

- **Naive outlier algorithms are unsuitable for extremely large datasets, because of the costly quadratic run times**

# Distance-based – Parallel & Distributed Approaches

- Speed-up distance-based outlier detection methods
  - Using parallel and/or distributed computing techniques

***[5] Lozano & Acuna, 2005***

- PBay algorithm
  - Split data in chunks to all processors, compute nearest neighbours
  - Master aggregates global score & updates cutoff,
    cutoff & next data chunks then redistributed

***[6] Hung & Cheung, 2002***

- Parallel basic nested loop algorithm
  - Not suitable for distributed computation, because
    all the dataset is exchanged among all the nodes

# Distance-based – Parallel/Distributed PBay Algorithm

- Major algorithm steps *[5] Lozano & Acuna, 2005*
  - Master node first splits the data into separate chunks for each processor
  - Then the master node loads each block of test data
  - Master node broadcasts test data blocks to each of the worker nodes
  - Each worker node then executes Orca (based on nearest neighbor)
  - Each worker is only using its local database and the test data block
  - The nearest neighbors of the test points from all the workers are aggregated at the master node
  - Master node finds global set of nearest neighbors for those test points
  - All test points whose score is less than the cutoff are dropped and the cutoff is updated accordingly
  - The cutoff is broadcast back to all the worker nodes along with the next block of test data

# Distance-based – Using K-means Clustering Algorithm

- Outlier detection by clustering approach
  - Grouping data into clusters
  - Data that is not assigned to any clusters are taken as outliers
  - Several clustering algorithms can be used
  - Example: density clustering with DBSCAN

***[10] Ester, 'A density-based algorithm for discovering clusters in large spatial databases with noise', 1996***

- K-means distance-based clustering method & outlier
  - Data is partitioned into k groups by assigning them to the closest cluster centers
  - Then calculate the distance (or dissimilarity) between each object and its cluster center
  - Finally pick those with largest distances as outliers

***[9] Zhao,'R and Data Mining Examples and Case Studies', 2013***

# Distance-based – Using K-means Example in R (1)

```
> kmeans
function (x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong",
    "Lloyd", "Forgy", "MacQueen"), trace = FALSE)
```

- K-Means clustering algorithm is available in R by default
  - *x* – data (e.g. numeric matrix of data)
  - *centers* – preset number of clusters k
  - *iter.max* – the maximum number of iterations allowed
  - *nstart* – how many random sets should be chosen?
  - *algorithm – c("Names of the different algorithms")*
  - *trace* – if positive, tracing information on the progress of the algorithm is produced
- Goal: Data given by x is clustered by the *k*-means algorithm
  - Partition the points into *k* groups such that the sum of squares from points to the assigned cluster centres is minimized
  - At the minimum, all cluster centres are at the mean of their Voronoi sets (the set of data points which are nearest to the cluster centre)

# Distance-based – Using K-means Example in R (2)

- Required R commands for the outlier detection & visualization
  - *head(data)* – lists the first entries of a certain data set

# Distance-based – Using K-means Example in R (3)

- Example data set: iris standard R data set

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

- Remove species information (column 5) from the iris data set
  - Create a subset of the iris data in a new data structure: irisdata

```
> irisdata <- iris[,1:4]
> head(irisdata)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
```

***[9] Zhao,'R and Data Mining Examples and Case Studies', 2013***

# Distance-based – Using K-means Example in R (4)

- Run K-means algorithm on the data
  - Use a predefined number of 3 clusters as input
  - Show attributes of the result

```
> kmeans.result <- kmeans(irisdata, centers=3)
> attributes(kmeans.result)
$names
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"

$class
[1] "kmeans"
```

- Show cluster centers

```
> kmeans.result$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.006000    3.428000     1.462000    0.246000
3     5.901613    2.748387     4.393548    1.433871
```

- Show cluster IDs

```
> kmeans.result$cluster
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [69] 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 3 1 1
[137] 1 1 3 1 1 1 3 1 1 1 3 1 1 3
```

*[9] Zhao,'R and Data Mining Examples and Case Studies', 2013*

# Distance-based – Using K-means Example in R (5)

- Create new data set with centers
  - Used for conveniently compute the distances later

```
> centers <- kmeans.result$centers[kmeans.result$cluster, ]
> head (centers)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
2        5.006       3.428        1.462       0.246
2        5.006       3.428        1.462       0.246
2        5.006       3.428        1.462       0.246
2        5.006       3.428        1.462       0.246
2        5.006       3.428        1.462       0.246
2        5.006       3.428        1.462       0.246
```

- Compute the distance
  - Using the irisdata dataset

```
> distances <- sqrt(rowSums((irisdata - centers)^2))
> head(distances)
[1] 0.1413506 0.4476382 0.4171091 0.5253380 0.1886266 0.6770377
> distances
  [1] 0.14135063 0.44763825 0.41710910 0.52533799 0.18862662 0.67703767 0.41518670 0.06618157 0.80745278 0.37627118 0.48247280 0.25373214
 [13] 0.50077939 0.91322505 1.01409073 1.20481534 0.65420180 0.14415270 0.82436642 0.38933276 0.46344363 0.32860310 0.64029681 0.38259639
 [25] 0.48701129 0.45208406 0.20875823 0.21536016 0.21066561 0.40838707 0.41373905 0.42565244 0.71552778 0.91977171 0.34982853 0.35039977
 [37] 0.52685861 0.25686572 0.76077592 0.11480418 0.18541845 1.24803045 0.66901420 0.38675574 0.60231221 0.48205809 0.41034132 0.47199576
 [49] 0.40494444 0.14959947 1.22697525 0.68414100 1.01903626 0.73153652 0.63853451 0.26937898 0.76452634 1.58388575 0.75582717 0.85984838
 [61] 1.53611907 0.32426175 0.80841374 0.39674141 0.87269542 0.87306498 0.41229163 0.53579956 0.63676390 0.71254917 0.70937310 0.46349013
 [73] 0.69373966 0.43661144 0.54593856 0.74313017 0.98798453 0.84636259 0.21993519 1.02437260 0.86396528 0.97566381 0.55763082 0.73395781
 [85] 0.57500396 0.68790275 0.92700552 0.61459444 0.50830256 0.62911910 0.48790256 0.38266958 0.49185351 1.54856350 0.38560870 0.44284695
 [97] 0.34498790 0.37241653 1.66064034 0.38393196 0.77731871 0.85382472 0.30610139 0.65293923 0.38458885 1.14225684 1.07101875 0.78573677
[109] 0.65454939 0.84355960 0.74552218 0.75289837 0.25958095 0.88917352 1.20227628 0.68288333 0.50991553 1.47791217 1.52971038 0.82617494
[121] 0.26952816 0.81891975 1.31149299 0.74269596 0.27627819 0.52766931 0.62526165 0.70228926 0.54629196 0.59428255 0.73126650 1.43802246
[133] 0.56055720 0.81536685 1.12133058 0.95311851 0.73306362 0.57903109 0.61011676 0.34794609 0.38934920 0.68403844 0.85382472 0.30952112
[145] 0.50939919 0.61173881 0.89747884 0.65334214 0.83572418 0.83452741
```

*[9] Zhao,'R and Data Mining Examples and Case Studies', 2013*

# Distance-based – Using K-means Example in R (6)

- Order the data set and pick top 5 with largest distance
  - Largest distances data points might be outliers
  - Show the top 5 dataset (ids) as candidates for outliers

```
> outliers <- order(distances, decreasing=T)[1:5]
> outliers
[1]  99  58  94  61 119
```

- Show outlier data of the top 5 data points

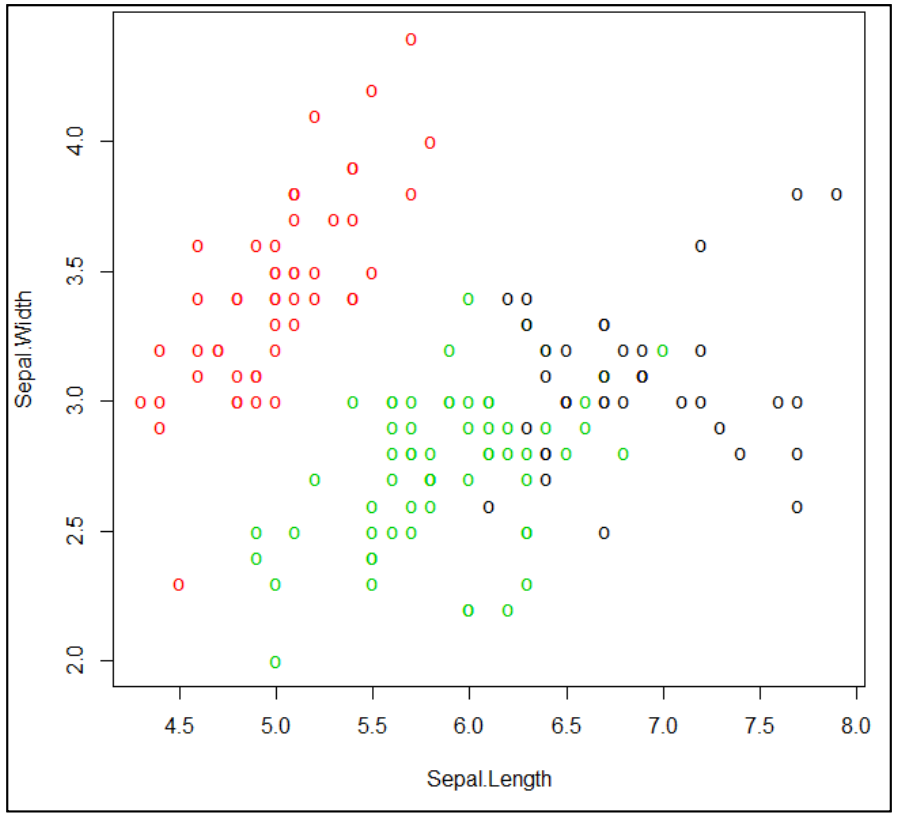
```
> print(irisdata[outliers,])
    Sepal.Length Sepal.Width Petal.Length Petal.Width
99           5.1         2.5          3.0         1.1
58           4.9         2.4          3.3         1.0
94           5.0         2.3          3.3         1.0
61           5.0         2.0          3.5         1.0
119          7.7         2.6          6.9         2.3
```

***[9] Zhao,'R and Data Mining Examples and Case Studies', 2013***

# Distance-based – Using K-means Example in R (7)

- Visualize (plot) clusters

```
> plot(irisdata[,c("Sepal.Length", "Sepal.Width")], pch="o", col=kmeans.result$cluster, cex=1)
```

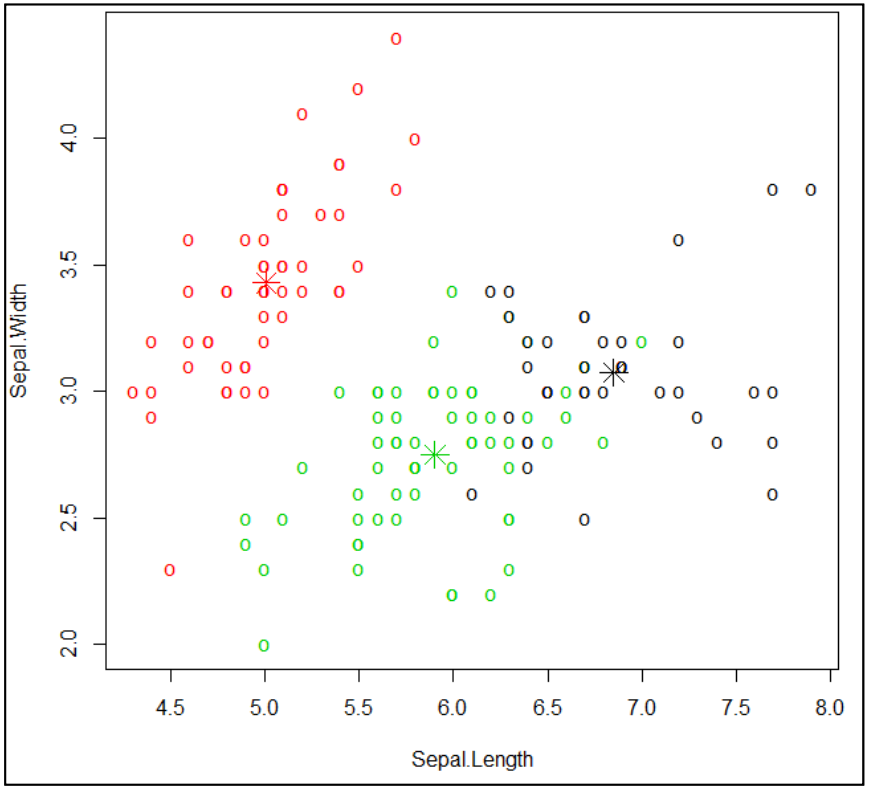



*[9] Zhao,'R and Data Mining Examples and Case Studies', 2013*

# Distance-based – Using K-means Example in R (8)

- Add cluster centers (asterisks) to the visualization (plot)

```
> points(kmeans.result$centers[,c("Sepal.Length", "Sepal.Width")], col=1:3, pch=8, cex=2)
```

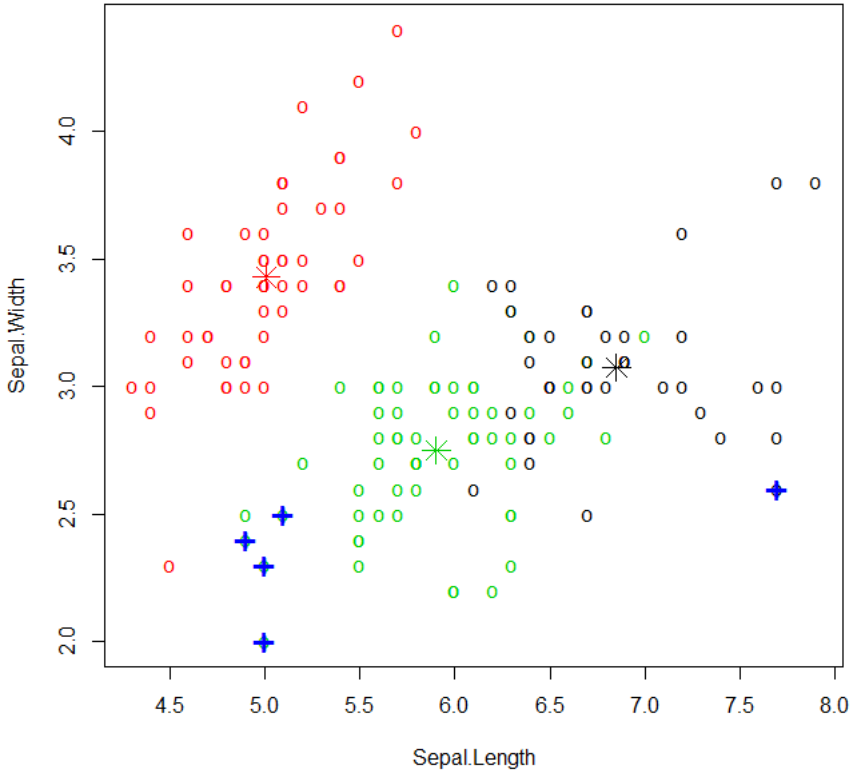



***[9] Zhao,'R and Data Mining Examples and Case Studies', 2013***

# Distance-based – Using K-means Example in R (9)

- Add top 5 outlier candidates (plus signs) to the visualization (plot)

```
> points(irisdata[outliers, c("Sepal.Length", "Sepal.Width")], pch="+", col=4, cex=2)
```



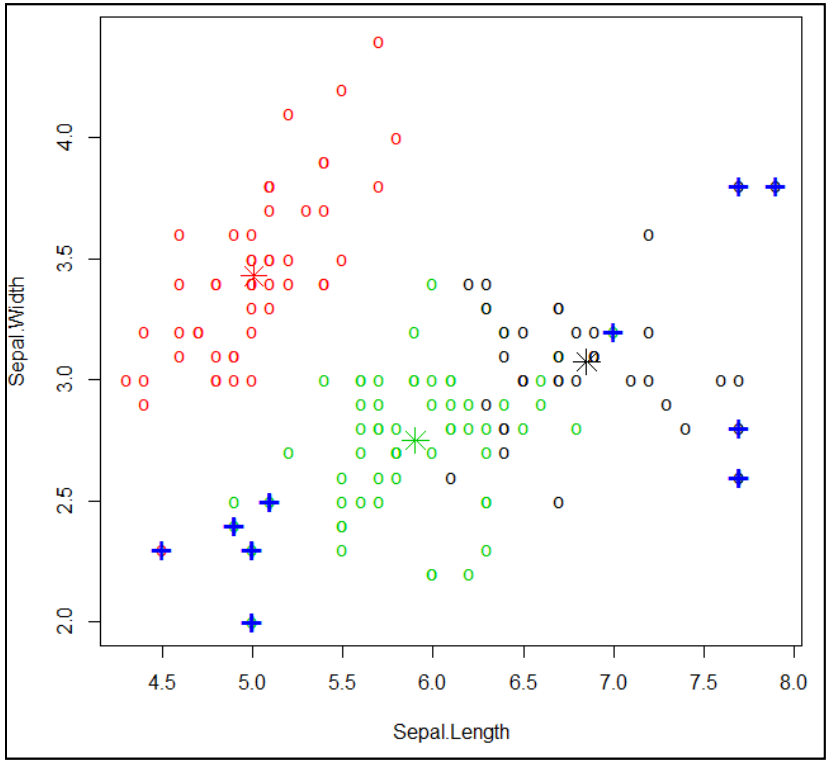

***[9] Zhao,'R and Data Mining Examples and Case Studies', 2013***

# Distance-based – Using K-means Example in R (10)

- Identify top 10 outliers
  - Visualize them in the plot

```
> outliers <- order(distances, decreasing=T)[1:10]
> outliers
 [1]  99  58  94  61 119 118 132 123  42  51
```

```
 [1]  99  58  94  61 119 118 132 123  42  51
> points(irisdata[outliers, c("Sepal.Length", "Sepal.Width")], pch="+", col=4, cex=2)
```



# Density-based – LOF Algorithm

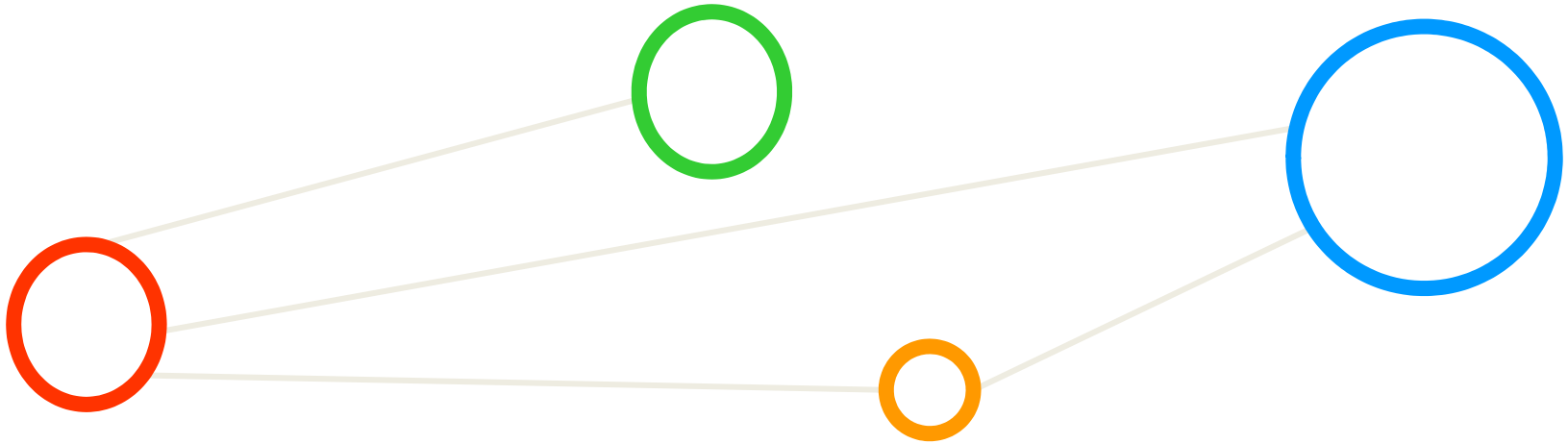
- Local Outlier Factor (LOF) algoritm
  - Identifies density-based local outliers

*[8] Breunig,' LOF: Identifying Density-Based Local Outliers ' 2000*

- Approach
  - 'Local density' of a point is compared with the density of its neighbors
- If the former density of a point is significantly lower than the latter density of a point (with an LOF value greater than one)
  - The point is in a sparser region than its neighbors
  - That suggests the point to be an outlier
- Packages DMwR and dprep (additional package to R)
  - Function lofactor(data, k) calculates local outlier factors using the LOF algorithm
  - k is the number of neighbors used in the calculation of the local outlier factors

*[7] Zhao,'R and Data Mining' 2014*

# Lecture Bibliography



# Lecture Bibliography


- [1] Engineering Statistics Handbook, ‘Detection of Outliers‘, Online: http://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm
- [2] Boris Iglewicz and David Hoaglin , "Volume 16: How to Detect and Handle Outliers", *The ASQC Basic References in Quality Control: Statistical Techniques*, Edward F. Mykytka, Ph.D., 1993
- [3] YouTube Video, ‘Excel 2010: Creating a Normal Probability Plot‘, Online: http://www.youtube.com/watch?v=1Ts2lYrXenE
- [4] E. Knorr and R. Ng. Algorithms for Mining Distance-Based Outliers in Large Datasets. In *Proceedings of VLDB’98*, pages 392–403, 1998.
- [5] E. Lozano and E. Acuna. Parallel Algorithms for Distance-Based and Density-Based Outliers. In *Proceedings of ICDM’05*, pages 729–732, 2005.
- [6] E. Hung and D. Cheung. Parallel Mining of Outliers in Large Database. *Distrib. Parallel Databases*, 12:5–26, 2002.
- [7] Y. Zhao, ‘R and Data Mining: Outlier Detection‘, Online: http://www.rdatamining.com/examples/outlier-detection
- [8] Markus Breunig , Hans-Peter Kriegel , Raymond T. Ng , Jörg Sander , ‘LOF: Identifying Density-Based Local Outliers’, in Proceedings of the 2000 ACM SigMod International Conference on Management of Data, Online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.8948
- [9] Yanchang Zhao, ‘R and Data Mining: Examples and Case Studies’, 2013, Online: http://cran.r-project.org/doc/contrib/Zhao_R_and_data_mining.pdf
- [10] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., ‘A density-based algorithm for discovering clusters in large spatial databases with noise’, In KDD, pages 226 – 231, 1996


Data
Big
Science
Research
science
methods
simulations
big
Centre
E.g
Computational
resources
Computing
climate
nodes
centers
PB
community
Environment
Enable
Modelling
technologies
processing
computing
European
TB
Scientific
data-intensive
Supercomputing
analysis
databases
computational
research
HPC
Juelich
datasets
performance
management
access
storage
ENES
EUDAT
increasing
brain
Project
DLCL
JSC
Climate
modelling
System
structures
DLCLs
Hardware
Technologies
Earth
Simulation
Energy
systems
Provide
NASA
Germany
Understanding
AR5
day
Health
Processing
directory
distributed
large-scale
cores
compute
project
General
Structure
current
hundreds
Nutshell
Services
network
Landscape
system
Concrete
Matters
Service
Summary
Large
Work
images
often
Using
Methods
IPCC
Common
physical
via
GB
stored
Resources
analyze
Level
use
codes
MapReduce
create
References
engineering
SMAQ
international
model
formats
Cross-Disciplinary
like
year
forms
cross-disciplinary
Policy-based
services
measurement
within
fusion
new
funding
concepts
device
Analysis
Computer
expertise
disciplines
results
Approaches
construction
Fusion
Test
High
EU
etc
DKRZ
Use
Key
scientific
important
basis
Infrastructure
files
different
acquisition
e.g
Lecture BDA6 – Outlier Detection
32 / 32