

# **BIG DATA ANALYTICS FOR OFFICIAL STATISTICS**

Dr. Abbas Maarooof  
aimaarooof@gmail.com  
29-03-2021



# WORKSHOP OUTLINE

---

1. Big Data Analytics-Day #1
2. Machine Learning Tools: R Programming language and Graphical Interfaces -Day#1
3. Data Mining Analytical Tools-Day#2
4. Data Visualization-Day#2
5. Digital Data Collection-Day#3
6. Drone Technology and Multispectral Sensor, Geospatial and Remote sensing Technology-Day#3





# **PART#1: BIG DATA ANALYTICS**



# Part#1: Presentation Outline

---

- Understanding Big Data: What is Big Data
- How big is the big data
- Big Data Types and Sources
- Application for Big Data Analytics
- Top Big Data Analytics Tools



# Big Data Interesting Facts

---

“From the dawn of civilization until 2003, humankind generated five Exabyte ( $10^{18}$  bytes) of data. Now we produce five Exabytes every two days....and the pace is accelerating”

Eric Schmidt  
Executive Chairman, Google

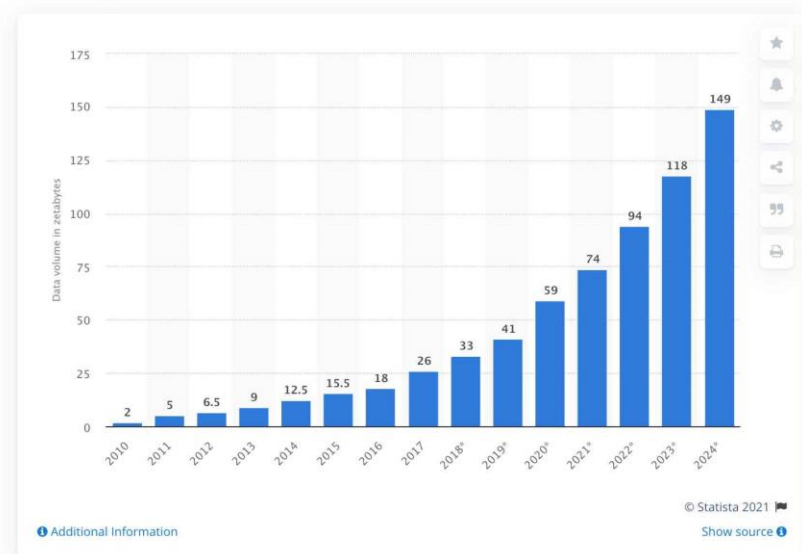
**90% of the world's data was created in the last few years alone! With new devices, sensors, and technologies emerging, the data growth rate will likely accelerate even more...**



# Big Data Interesting Facts

- Every minute we send 204 million emails, generate 1,8 million Facebook likes, send 278 thousand Tweets, and up-load 200 thousand photos to Facebook, etc.
- Google alone processes on average over 40 thousand search queries per second, making it over 3.5 billion in a single day
- Around 100 hours of video are uploaded to YouTube every minute and it would take you around 15 years to watch every video uploaded by users in one day
- Facebook processes 10 TB of data every day / Twitter 7 TB
- The total amount of data created, captured, copied, and consumed in the world is forecast to increase rapidly, reaching 149 zettabytes in 2024. The rapid development of digitalization contributes to the ever-growing global data sphere.

(in zettabytes)

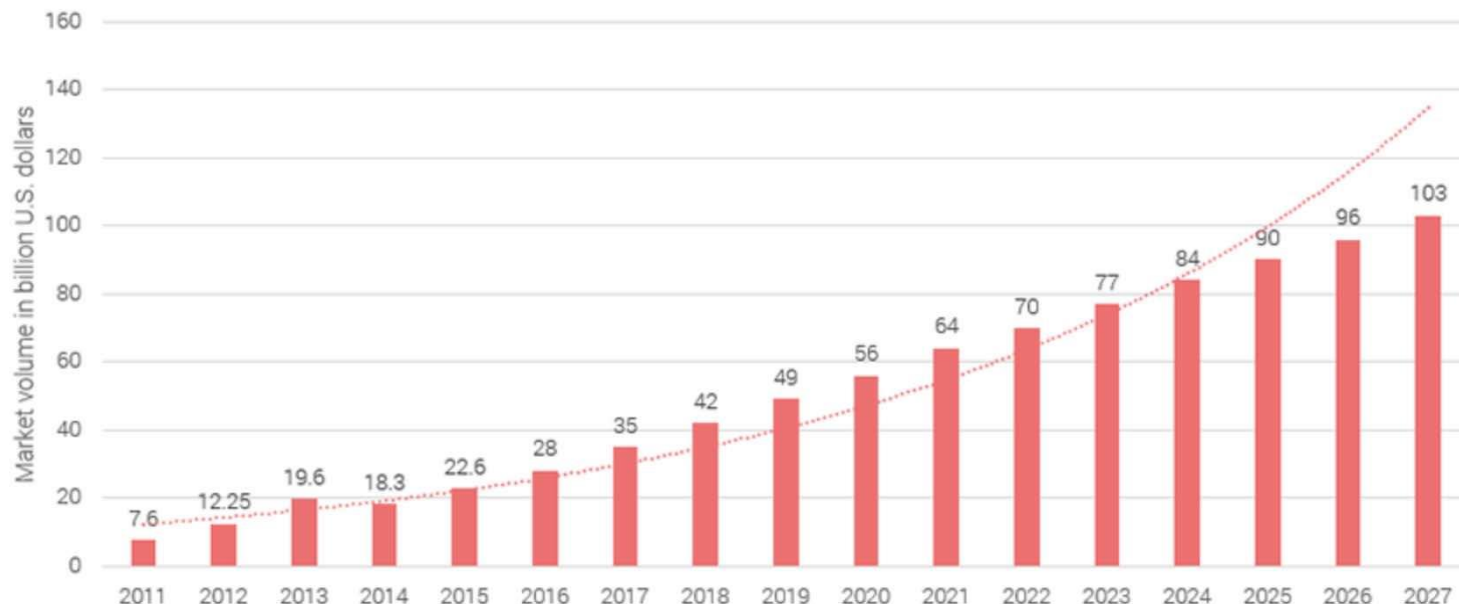




# Big Data Interesting Facts

- Bad data or poor data quality costs US businesses \$600 billion annually
- Today's data centres occupy an area of land equal in size to almost 6,000 football fields.

## Forecast of Big Data market size based on revenue (2015-2027)



Source: Statista



# Understanding Big Data: What is Big Data?

**Big data** is a buzzword; used to describe a massive volume of both structured and unstructured data that is so large it is difficult to process using traditional database and software techniques.

Relevance of big data occurs when large amounts of relevant data is analyzed, turned into information and then into knowledge.



It requires specific skill sets and attention towards processes and context.



# Understanding Big Data: What is Big Data?

## MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

*Google, Inc.*

### Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

Our implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: hundreds of MapReduce programs have been implemented and upwards of one thousand MapReduce jobs are executed on Google's clusters every day.

given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and hundreds of machines are needed to process a reasonable amount of data. The system must handle failures and manage the required inter-machine communication with these issues in mind.

As a result, the abstraction we use is simple and easy to use. It is inspired by the fact that many of the operations required to compute a large computation can be expressed as applying the same operation to many small pieces of data. This abstraction is the primary mechanism for fault tolerance.

The major contributions of this work are a simple and powerful interface that enables automatic parallelization and distribution of large-scale computations, combined

**MapReduce was first popularized as a programming model in 2004 by Jeffrey Dean and Sanjay Ghemawat of Google (Dean & Ghemawat, 2004). In their paper,**

**“MAPREDUCE: SIMPLIFIED DATA PROCESSING ON LARGE CLUSTERS,” they discussed Google's approach to collecting and analyzing website data for search optimizations. Google's proprietary MapReduce system ran on the Google File System (GFS). Apache, the open source organization, began using MapReduce**



# Big Data is characterized by the four “V’s

---

## 4 V's of Big Data

### Volume

- Data quantity

### Velocity

- Data Speed

### Variety

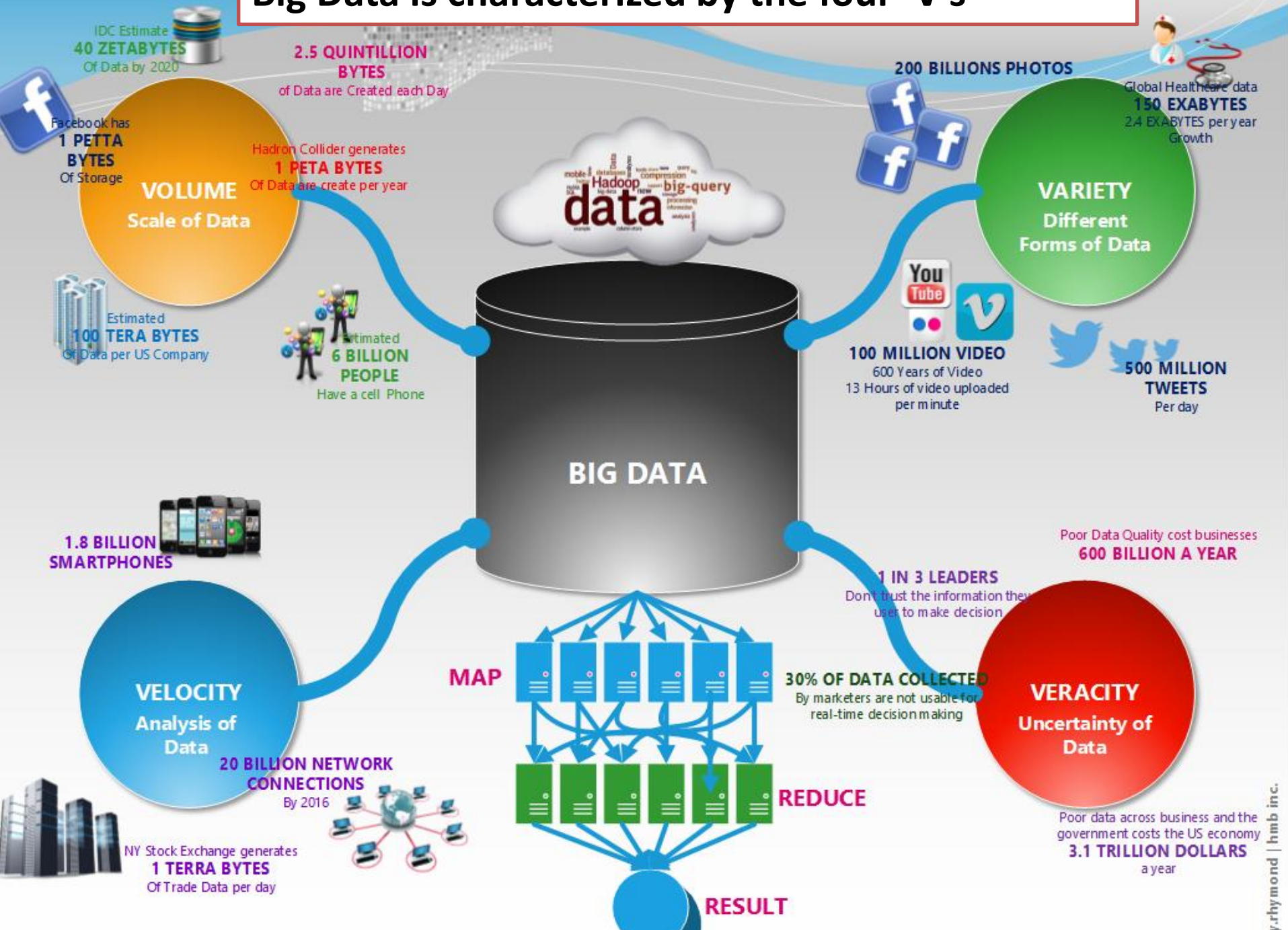
- Data Types

### Veracity

- Messiness



# Big Data is characterized by the four 'V's





# Typical Hadoop Cluster



## Who uses MapReduce and for What

### □ At Google:

1. Index construction for Google Search
2. Article clustering for Google News
3. Statistical machine translation

### □ At Yahoo!:

1. "Web map" powering Yahoo! Search
2. Spam detection for Yahoo! Mail

### □ At Facebook:

1. Data mining
2. Ad optimization
3. Spam detection

### □ At New York Times

1. Moving typeset into PDF

**If you don't own a warehouses like this, go to the Cloud!**



# How Big is Big Data

Byte (1): One grain of rice

Kilobyte ( $10^3$ ): cup of rice



Hobbyist

Megabyte ( $10^6$ ): 8 bags of rice

Gigabyte ( $10^9$ ): 3 Semi trucks of rice



Desktop

Terabyte ( $10^{12}$ ): 2 Container Ships of rice

Petabyte ( $10^{15}$ ): Blankets Manhattan of rice



Internet

Exabyte ( $10^{18}$ ): Blankets west coast states of rice

Zettabyte ( $10^{21}$ ): Fills the Pacific Ocean with of rice



Big Data

Yottabyte ( $10^{24}$ ) : A Earth Size Rice Ball

The Future?



# Challenges Facing Big Data

---

- Data access and connectivity can be an obstacle
- The Big Data Talent Gap
- Getting Data into the Big Data Platform
- Synchronization across the Data Sources
- Getting Useful Information out of the Big Data Platform
- The technology landscape in the data world is evolving extremely fast





# **Big Data Types and Sources**



# Big Data Types and Sources

---

## 1. Structured Data

(i) **Computer- or Machine-Generated Structured Data:**

Sensor data, Web log data , Point-of-sale data, Financial data

(ii) **Human-Generated Data:** Input data, Click-stream data, and Gaming-related data

$fx$			
	A	B	C
1	first_name	last_name	order_id
2	Caroline	Forsey	124527
3			



# Big Data Types and Sources

## 2. Semi-Structured Data

Semi-structured data are increasingly occurring since the advent of the Internet

### Semi Structured Data Examples

1. Email
2. CSV, Extensible Markup Language ( XML ) and JSON (JavaScript Object Notation) documents
3. NoSQL databases
4. HTML
5. Electronic data interchange (EDI)
6. Resource Description Framework (RDF) s a framework for describing resources on the web

```
4
5 {
6     first_name  : "Caroline",
7     last_name   : "Forsey",
8     order_id    : "124527",
9
10 }
```

Example of JSON  
(JavaScript Object  
Notation)



# Big Data Types and Sources

## 3. Unstructured DATA

Unstructured information is typically **text-heavy**, but may contain data such as dates, numbers, and facts as well:

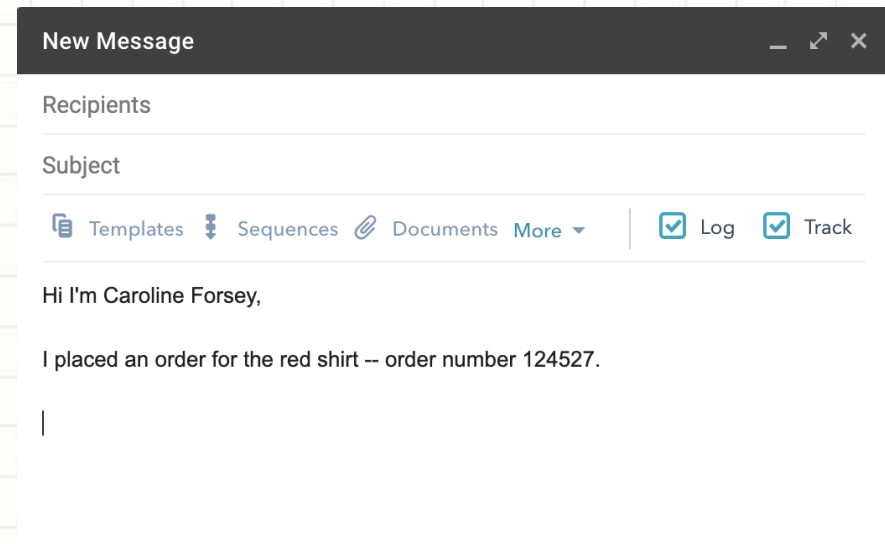
### (i) Machine-Generated Unstructured Data Examples:

Satellite images, Scientific data, Photographs and video and Radar or sonar data:

### (ii) Human-generated Unstructured Data Examples

Mobile and Voice data , Social media data ,Image and Video Data, Machine Data

An example of unstructured data includes email responses, like this one:



The screenshot shows a 'New Message' window. The 'Recipients' field is empty. The 'Subject' field is empty. Below the subject field, there are links for 'Templates', 'Sequences', 'Documents', and a 'More' dropdown menu. To the right of these links are two checkboxes: 'Log' and 'Track', both of which are checked. The body of the email contains the text: 'Hi I'm Caroline Forsey, I placed an order for the red shirt -- order number 124527.' followed by a vertical line.



# Top Big Data Applications

- Big Data in Retail
- Big Data in Healthcare
- Big Data in Education
- Big Data in E-commerce
- Big Data in Media and Entertainment
- Big Data in Finance
- Big Data in Travel Industry
- Big Data in Telecom
- Big Data in Automobile



# Big Data Analytics Methods

---

- (1) **Data Visualization**: Big data visualization
- (2) **Data Mining**: Social media(Facebook, Twitter, Instagram, etc.), text, email , and Internet (Google)...etc. (It can be done at a fraction of the cost and in real time).
- (3) **Predictive analytics**: Machine learning, predictive and statistical modeling
- (4) **Crowdsourcing**: Data from various sources
  - text messages
  - social media
  - blogs, etc.
- (5) **Internet of things**: Real-time data collection to computing systems by **sensors** and **actuators**
- (6) **Mobile analytics**: massive amounts of data that mobile companies gather about their users in terms of:
  - calling volume and pattern
  - location
  - privacy & ethical use challenges

**Applications of Big Data are Endless! It is just the beginning of a transformation into a big data economy.**



# Big Data Analytics Benefits

- Big data allows for better prediction of economic phenomena and improves causal inference
- Identify economic trends as they occur (“nowcasting”) to testing agents’ behaviour theories or creating a set of tools to manipulate and analyse these data
- Clustering and demand modelling
- Regularization to assist with variable selection in high-dimensional trade policy models
- Predictive analytics of the development of specific complex processes, e.g. climate, geological, natural, social, demographic, macroeconomic, etc





# **Top Big Data Analytics Tools**



# Top Big Data Analytics Tools

---

1. **Apache Hadoop** and Map Reduce Apache Hadoop is an open source software framework employed for handling of big data. It processes datasets of big data by means of the MapReduce programming model.
2. **Cloudera**- Distribution for Hadoop (CDH)
3. **Spark**- Apache Spark is an open source framework for data analytics
4. **Apache Cassandra** is free of cost and open-source distributed

**Top Cloud Computing;** such as Amazon Web Services (AWS), Google AI Platform, and Microsoft Azure





# **Big Data Analytics Case Studies**



## Case Study#1:

# Predict hotel bookings via user Behaviour

Machine learning, predictive and statistical modeling

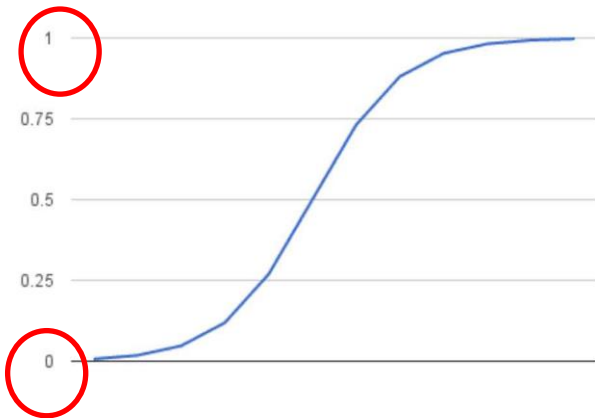
**Situation:** A search session describes a user's journey to find his ideal hotel, by including all his interactions. Given user search sessions, we are interested in predicting the outcome of these sessions based on the users' interactions; as well determining which of these interactions have the highest importance for this estimation.

The task is to train a machine learning model to estimate if a booking occurred – the training and target sets have been provided for you.



# Making Predictions with Logistic Regression:

To make predictions, we must train a final model. We can train models using train/test splits or k-fold cross validation of our data



Logistic Function

Number of raw booking  
data= **20,170,405** can  
not be used by Excel

```
In [38]: # example of training a final classification model
from sklearn.linear_model import LogisticRegression
from sklearn.datasets.samples_generator import make_blobs
# generate 2d classification dataset
X, y = make_blobs(n_samples=10, centers=2, n_features=2, random_state=1)
# fit final model
model = LogisticRegression()
model.fit(X, y)
# new instances where we do not know the answer
Xnew, _ = make_blobs(n_samples=10, centers=2, n_features=2, random_state=1)
# make a prediction
ynew = model.predict(Xnew)
# show the inputs and predicted outputs
for i in range(len(Xnew)):
    print("X=%s, Predicted=%s" % (Xnew[i], ynew[i]))

import pickle
filename = 'finalized_model.sav'
pickle.dump(model, open(filename, 'wb'))

X=[-10.17014071 -4.83120697], Predicted=1
X=[-11.09833168 -2.80862484], Predicted=1
X=[-9.95549876 -3.37053333], Predicted=1
X=[-8.86394306 -5.05323981], Predicted=1
X=[0.08525186 3.64528297], Predicted=0
X=[-0.79415228 2.10495117], Predicted=0
X=[-1.34052081 4.15711949], Predicted=0
X=[-10.32012971 -4.3374027 ], Predicted=1
X=[-2.18773166 3.33352125], Predicted=0
X=[-0.19745197 2.34634916], Predicted=0
```



## Case Study#2:

Use Hadoop to Derive Descriptive Statistics about patent data, and look for interesting, non-obvious, patterns



# Using Hadoop to Derive Descriptive Statistics about patent data, and look for interesting, non-obvious, patterns

## Source of Data, National Bureau of Economic Research

- <http://www.nber.org/data/>
- <http://data.nber.org/patents/>
- Download acite75\_99.zip (82MB) and apat63\_99.zip (56Mb)

Description	Documentation	Data -- Pkzipped	
		SAS .tpt	ASCII CSV
Overview	overview.txt	--	
Pairwise citations data	Cite75_99.txt	Cite75_99.zip -- (68 Mb)	acite75_99.zip -- (82 Mb)
Patent data, including constructed variables	pat63_99.txt	pat63_99.zip -- (90Mb)	apat63_99.zip -- (56Mb)
Assignee names	coname.txt	coname.zip -- (2Mb)	aconame.zip -- (2Mb)
Contains the match to CUSIP numbers	match.txt	match.zip -- (130Kb)	amatch.zip -- (98Kb)
Individual inventor records	inventor.txt	inventor.zip -- (98Mb)	ainventor.zip -- (82Mb)
Class codes with corresponding class names	classes.txt	--	
Country codes with corresponding country names	countries.txt		
Class, technological category, and technological	class_match.txt		

**Source:** USPTO, and Jaffe and Trajtenberg computations. The Pat63\_99 file includes all utility patents in the USPTO's TAF database granted during the period 1963 to December 1999. Classification information reflects the U.S. Patent Classification System as of December 31, 1999. No. of observations: 2,923,922



# Patent Citation Data, cite75\_99.txt File

- Source: US Patent office
- This file includes all US patent citations for utility patents granted in the period 1-Jan-75 to 31-Dec-99.
- No. of observations: 16,522,438
- Variable Name    variable type    Characters    Contents
- CITING            numeric            7            Citing Patent Number
- CITED             numeric            7            Cited Patent Number
- The file is sorted by Citing Patent Number.

"CITING", "CITED"

6009552,5278871

6009552,5598422

6009553,4131849

6009553,4517669

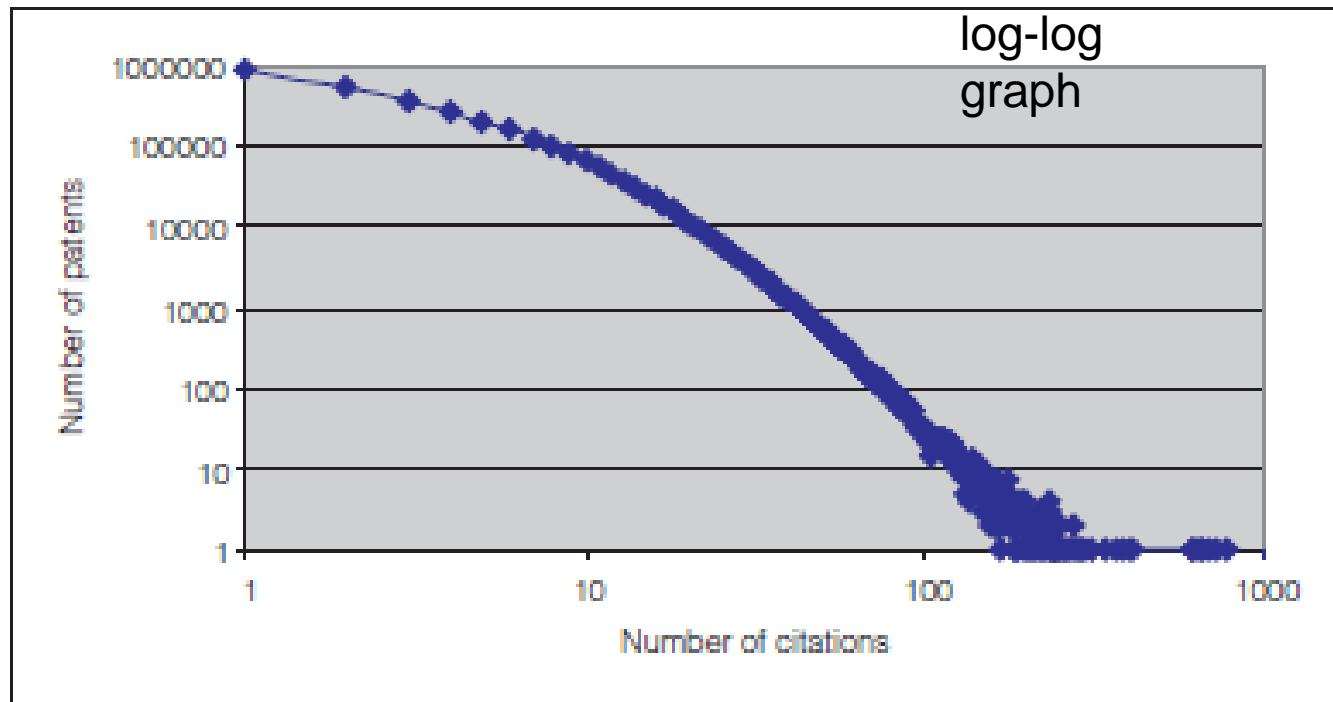
6009553,4519068

6009553,4590473

6009553,4636791

6009553,5671255

.....





## **Case Study#3:**

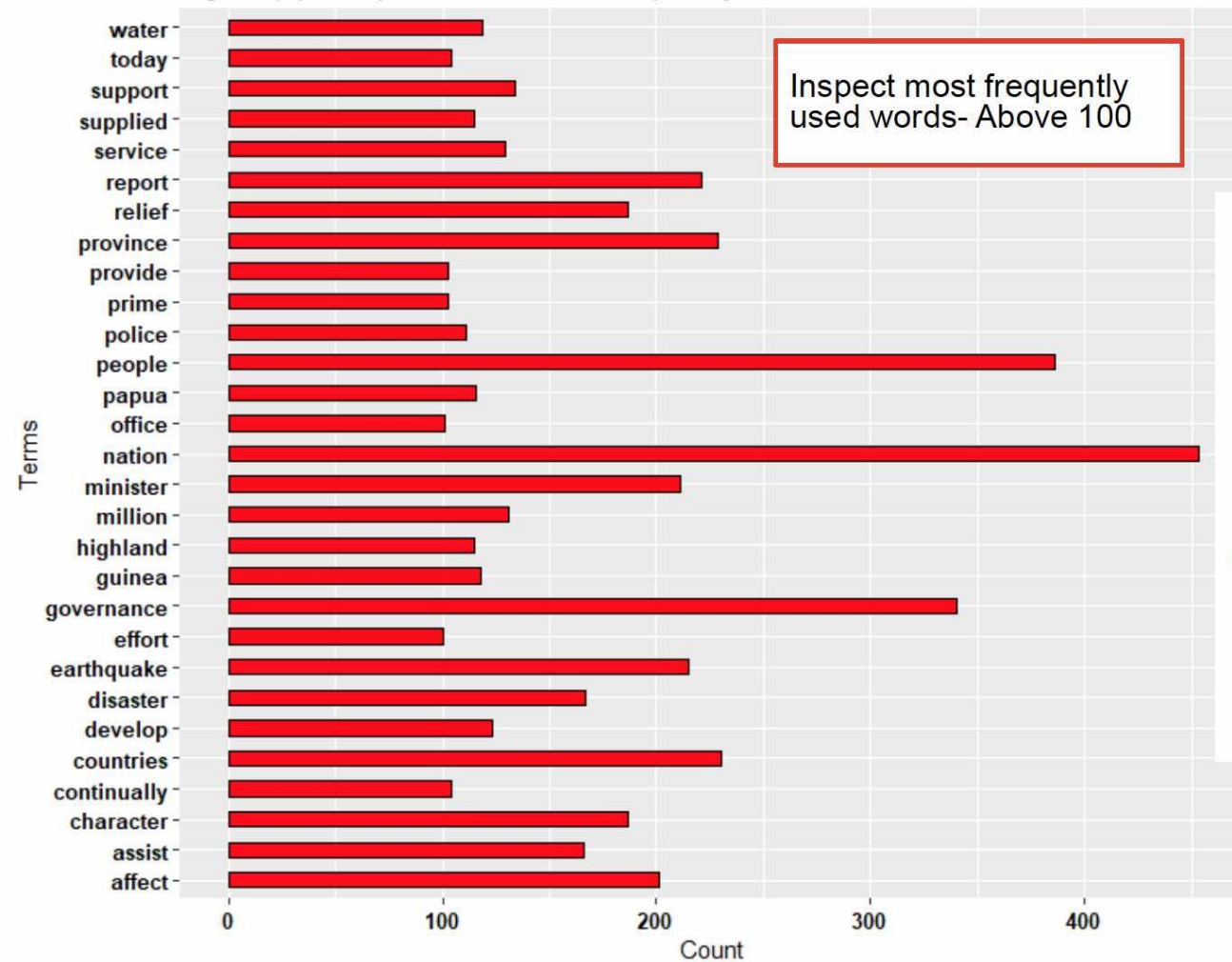
Data (Text) Mining of Papua New Guinea NBC National News Videos:  
Voice recognition algorithms  
technology



- Traditional statistics, household surveys and census data have been effective in tracking medium to long-term development trends, but are less effective in generating a real-time snapshot for policymakers , while **Data Mining method can be done at a fraction of the cost and in real time.**
- There is an ocean of data (Big Data)— generated by citizens in both developed and developing countries—that did not exist even a few years ago. Videos, Audios, Mobile phones, social media and Internet searches all leave digital traces that, **when anonymized, aggregated and analyzed, can reveal significant insights that help governments make faster and more informed decisions.**



Figure (1): Frequent words- Low frequency 100 times.



Inspect most frequently used words- Above 100







1. The figure shows how topics within a document are distributed according to the model.
2. In the current model all three documents show at least a small percentage of each topic.
3. However, two to three topics dominate each document.

Let us first take a look at the contents of *three* sample documents:  
[1,10,20]

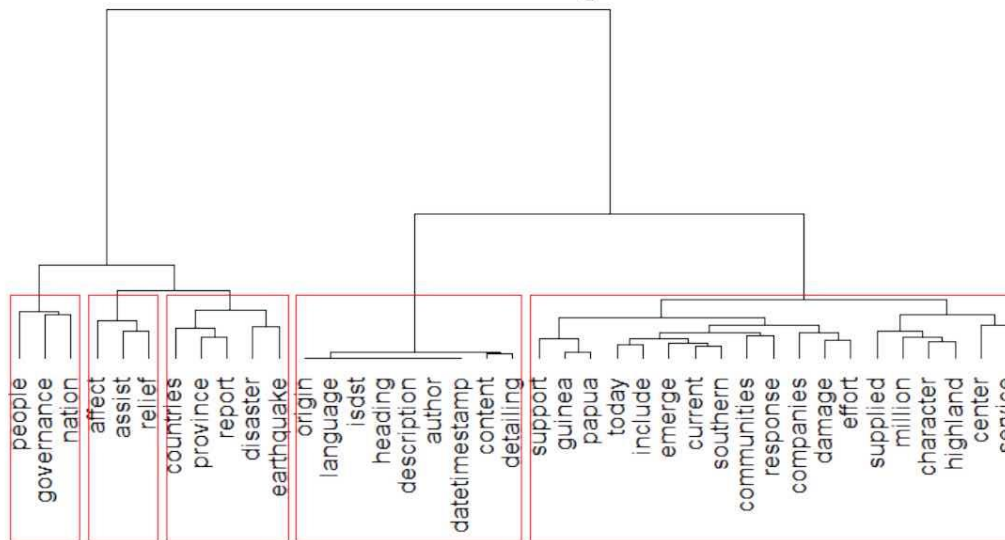
## Topic distribution



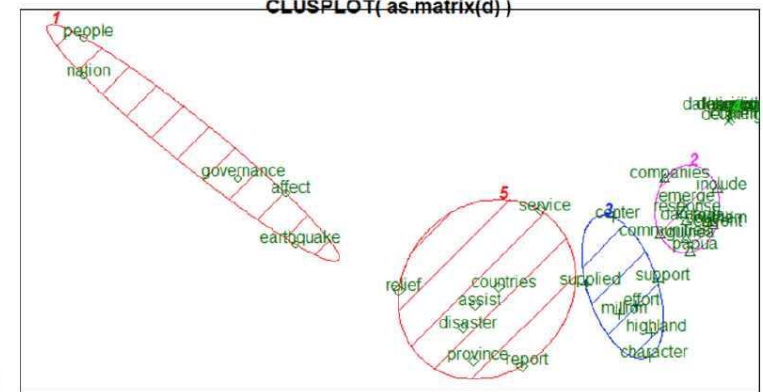
# Clustering by Term Similarity




Cluster Dendrogram



CLUSPLOT(as.matrix(d))







## **Case Study#4:**

# **Big Data Analytics in Public Health - Maternal Mortality the case of Indonesia-D3.js code**



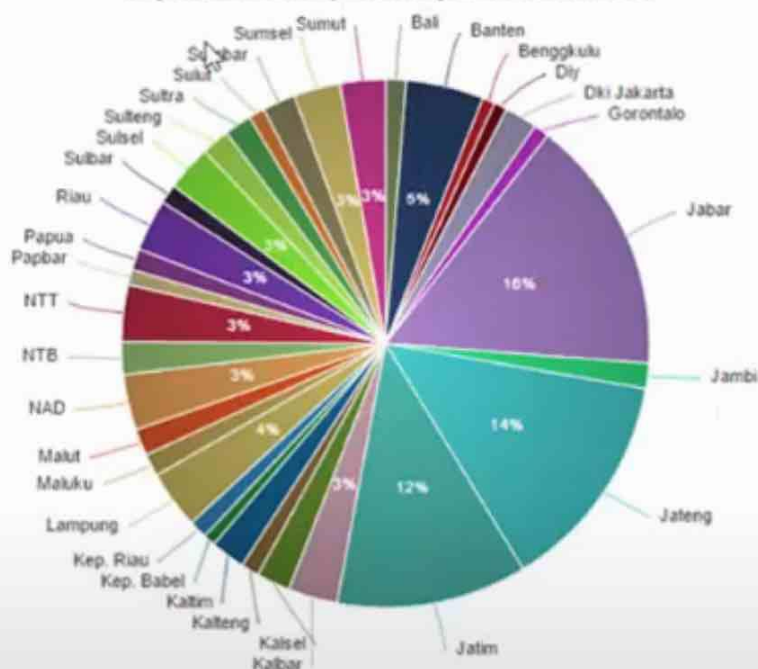
[https://www.youtube.com/watch?v=coanS5RgKxs&feature=em-upload\\_owner](https://www.youtube.com/watch?v=coanS5RgKxs&feature=em-upload_owner)



# D3 Visualization for the Maternal Mortality Dataset as a D3 Pie

## Maternal Mortality in Indonesia

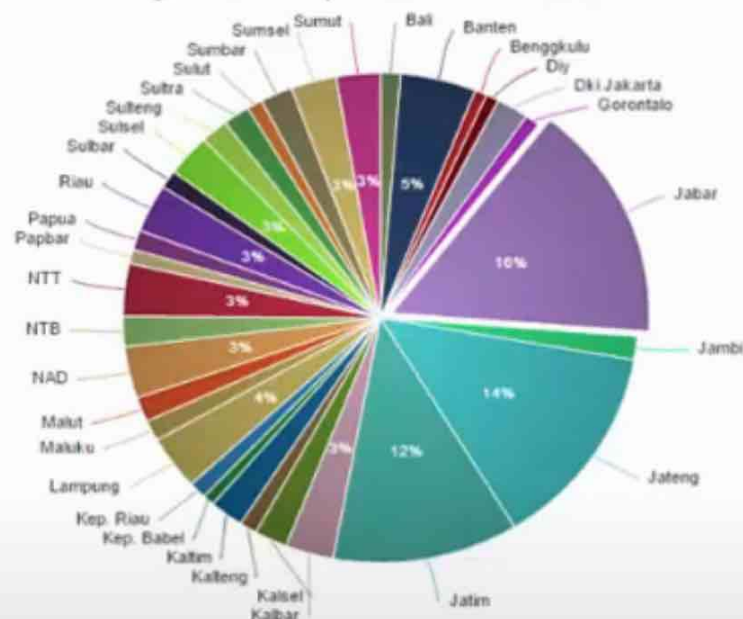
Range of Maternal Mortality in Different provinces of Indonesia-2012



**Figure 3(a): Full pie chart  
dynamic portion when is off**

## Maternal Mortality in Indonesia

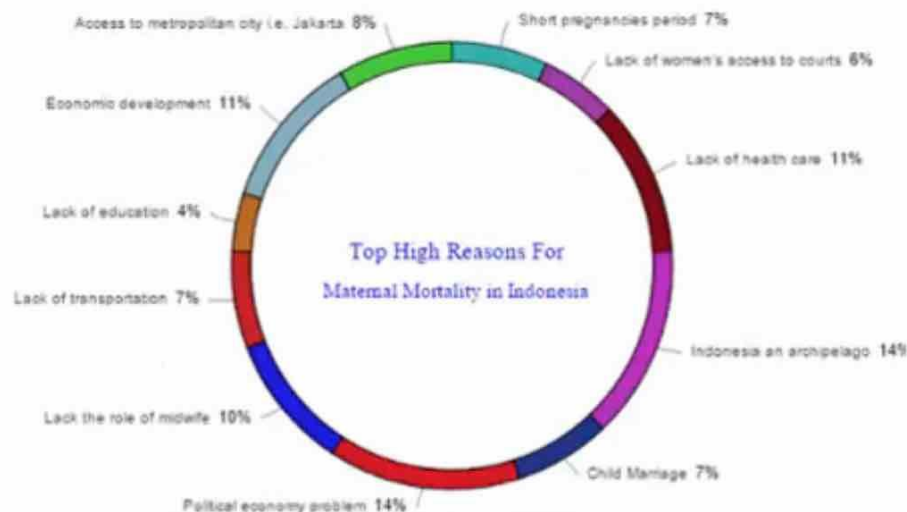
Range of Maternal Mortality in Different Provinces of Indonesia-2012



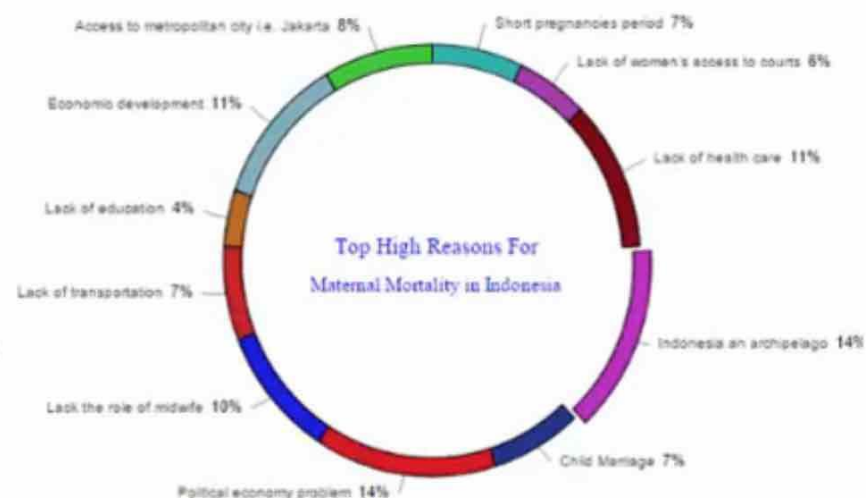
**Figure 3(b): Full pie chart  
dynamic portion when is on**



# D3 Visualization of General Reasons for High Maternal Mortality in Indonesia as a D3 Donut Pie




**Figure 4(a): D3 donut chart dynamic portion when is off**



**Figure 4(b): D3 donut chart dynamic portion when is on**





# **PART#2. MACHINE LEARNING TOOLS: R-PROGRAMMING LANGUAGE AND GRAPHICAL INTERFACES**



# Programming Tools and Types

---

## Text Interfaces

These are generally programming languages that use written commands:

1. One of the most fundamental tools in data mining is the statistical **programming language R-Free-Open Source**
2. **Programming language Python 3-Free- Open Source**

## Graphical Interfaces

These include specialized applications that use menus, widgets, virtual connections, and it's really easy to see the process:

1. **RapidMiner**- There is a free version and there is a paid version- You can download the free version
2. **KNIME**- You can download it for free
3. **Orange**-You can download it for free
4. **BigML** on Server free for small task but they charge with big data analytics-Nice way to work with.





# **PART#2: MACHINE LEARNING TOOLS: R PROGRAMMING LANGUAGE**





# **Why Learn R programming language ?**



# R Processing more than just statistics

---

R was developed by statisticians to make statistical processing easier. This heritage continues, making R a very powerful tool for performing virtually any statistical computation.

**The result is that R is now eminently suitable for a wide variety of nonstatistical tasks, including** Data processing, Graphic visualization, and analysis of all sorts

## R is being used in

- The fields of finance,
- Natural language processing, Genetics,
- Biology,
- Market research, to name just a few

Which means that you can use R alone to program anything you want

## Running code without a compiler

R is an interpreted language, which means that — contrary to compiled languages like C and Java — you don't need a compiler to first create a program from your code before you can use it



# Why Learn R programming language ?

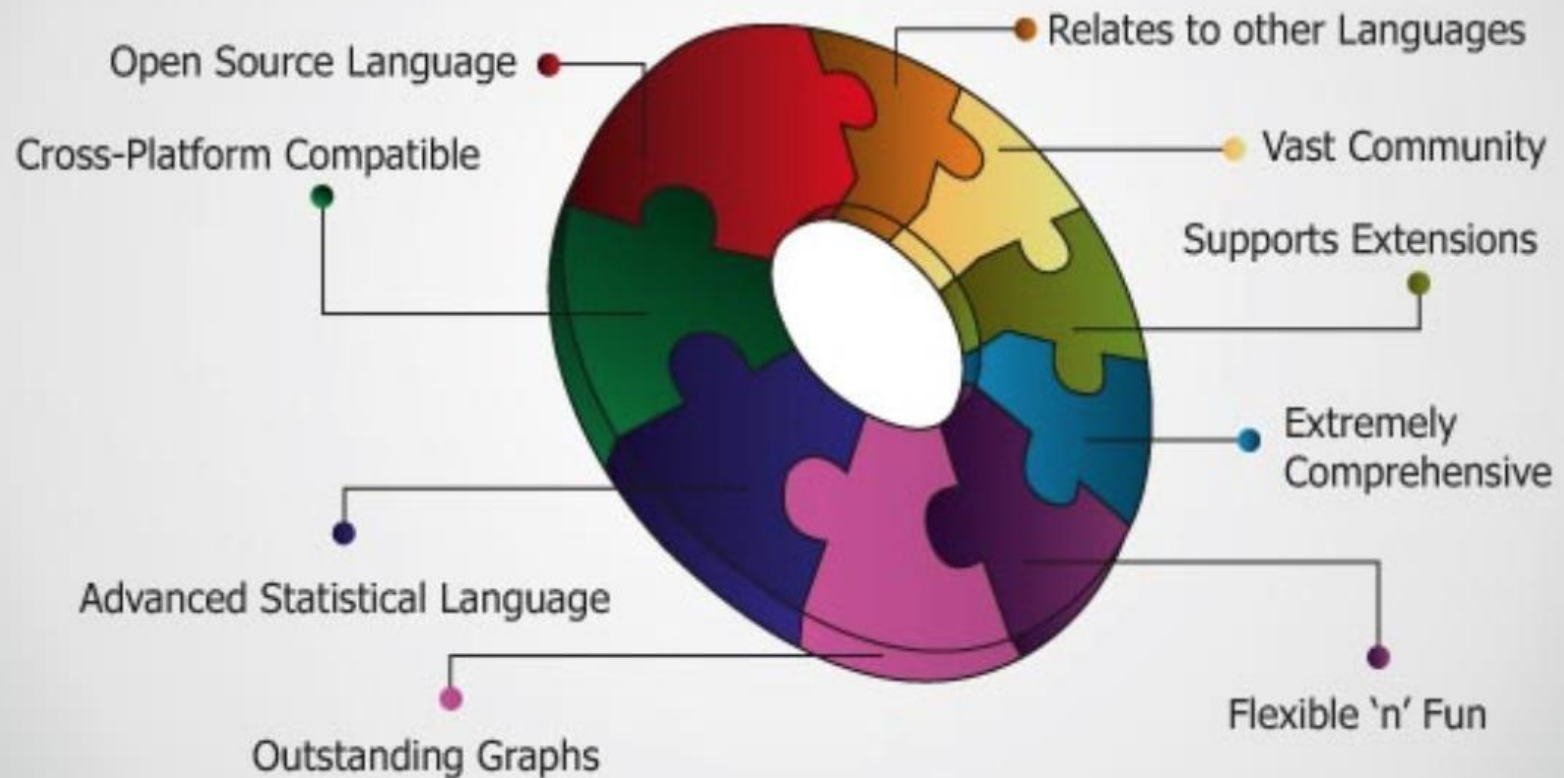
---

1. R-the programming language favoured by many statisticians because facilitate matrix arithmetic - carrying out complex, often automated calculations on data which is held in a grid of rows and columns.
2. The style of coding is quite easy.
3. It's open source. No need to pay any subscription charges.
4. Availability of instant access to over 7800 packages customized for various computation tasks.
5. The community support is overwhelming. There are numerous forums to help you out.
6. Online, R code is everywhere although you won't see it, as it's always hidden behind pretty graphical interfaces. But when you use **Google, Facebook or Twitter** you are almost certainly executing R code running on the servers of those organizations.
7. It is also capable of executing code written in other languages such as C++ or Java, so resources coded in those languages can be made available. Because it can be compiled to run on any major operating system, R code can easily be ported between Unix, Windows or Mac environments.
8. With a reported more than **two million** users worldwide, and thousands of deployed applications created using it, **R is undoubtedly one of the backbone technologies of the Big Data revolution.**



# Why Learn R programming language ?

## Why Learn R?





# Things R does and What R does not do

---

R does	R does not
<ul style="list-style-type: none"><li>• Data handling and storage: numeric, textual</li><li>• Matrix algebra</li><li>• Has tables and regular</li><li>• Expressions</li><li>• high-level data analytic and statistical functions</li><li>• classes (“Object Oriented”)</li><li>• Graphics</li><li>• programming language: loops, branching, Subroutines</li></ul>	<ul style="list-style-type: none"><li>• Is not a database, but connects to DBMSs</li><li>• has no graphical user interfaces, however it connects to Java, TclTk and it has <b>R Studio</b></li><li>• language interpreters are not fast. However, R could be extended by compiled C/C++ code</li><li>• No spreadsheet view of data, but connects to MS Excel.</li><li>• No professional / commercial Support</li></ul>



# Comparing R with the classic Statistical Tools

Features	Stata	SPSS	SAS	R
Data extensions	*.dta	*.sav, *.por (portable file)	*.sas7bcat, *.sas#bcat, *.xpt (xport files)	*.Rdata
User interface	Programming/point-and-click	Mostly point-and-click	Programming	Programming
Data manipulation	Very strong	Moderate	Very strong	Very strong
Data analysis	Powerful	Powerful	Powerful/versatile	Powerful/versatile
Graphics	Very good	Very good	Good	Excellent
Cost	Affordable (perpetual licenses, renew only when upgrade)	Expensive (but not need to renew until upgrade, long term licenses)	Expensive (yearly renewal)	Open source
Program extensions	*.do (do-files)	*.sps (syntax files)	*.sas	*.txt (log files)
Output extension	*.log (text file, any word processor can read it), *.smcl (formatted log, only Stata can read it).	*.spo (only SPSS can read it)	(various formats)	*.R, *.txt(log files, any word processor can read)



# Useful R Packages

---

Out of ~7800 packages listed on CRAN, I've listed some of the most powerful and commonly used packages in predictive modeling in this article. Since, I've already explained the method of installing packages, you can go ahead and install them now. Sooner or later you'll need them.

**Importing Data:** R offers wide range of packages for importing data available in any format such as .txt, .csv, .json, .sql etc. To import large files of data quickly, it is advisable to install and use `data.table`, `readr`, `RMySQL`, `sqldf`, `jsonlite`.

**Data Visualization:** R has in built plotting commands as well. They are good to create simple graphs. But, becomes complex when it comes to creating advanced graphics. Hence, you should install `ggplot2`.

**Data Manipulation:** R has a fantastic collection of packages for data manipulation. These packages allows you to do basic & advanced computations quickly. These packages are *dplyr*, *plyr*, *tidyr*, *lubridate*, *stringr*. Check out this [complete tutorial](#) on data manipulation packages in R.

**Modeling / Machine Learning:** For modeling, *caret* package in R is powerful enough to cater to every need for creating machine learning model. However, you can install packages algorithms wise such as *randomForest*, *rpart*, *gbm* etc

**Note:** I've only mentioned the commonly used packages. You might check this interesting [Link](#) on complete list of useful R packages. And functionality

[https://rstudio.com/wp-content/uploads/2019/01/Cheatsheets\\_2019.pdf](https://rstudio.com/wp-content/uploads/2019/01/Cheatsheets_2019.pdf)





# **R Programming Functionality: Insight/Analysis**



# 1. Data Analysis

Exploratory data analysis is a term minted in data analysis using R. This is an approach for data analysis which includes a variety of techniques such as:

1. Extraction of important variables
2. Test underlying assumptions
3. Maximising insights into the dataset, etc.

## 2. Data wrangling

Data wrangling is the process of cleaning messy and complex data sets to enable convenient consumption and further analysis . The following packages can be used to do three main parts, import, tidy and transform:

1. **dplyr Package** –dplyr is best known for its data exploration and transformation capabilities and highly adaptive chaining syntax.
2. **data.table Package** – It allows for faster manipulation of data set with minimum coding. It simplifies data aggregation and drastically reduces the compute time.
3. **readr Package** – ‘readr’ helps in reading various forms of data into R. By not converting characters into factors it performs the task at 10x faster speed



### **3. Machine learning**

At some point in data science, a programmer may need to train the algorithm and bring in automation and learning capabilities to make predictions possible. R provides ample tools to developers to train and evaluate an algorithm and predict future events. Thus, R makes machine learning (a branch of data science) lot more easy and approachable. The list of R packages for machine learning is really extensive. R machine learning packages include MICE (to take care of missing values), rpart & PARTY (for creating data partitions), CARET (for classification and regression training), randomFOREST (for creating decision trees) and much more.

### **4. Data visualization**

Data visualization is the visual representation of data in graphical form. This allows analyzing data from angles which are not clear in unorganized or tabulated data. R has many tools that can help in data visualization, analysis, and representation. The R packages ggplot2 and ggedit for have become the standard plotting packages. While the ggplot2 package is focused on visualizing data, ggedit helps users bridge the gap between making a plot and getting all of those pesky plot aesthetics precisely correct

### **5. Specificity**

R is a language designed especially for statistical analysis and data reconfiguration. All the R libraries focus on making one thing certain – to make data analysis easier, more approachable and detailed. Any new statistical method is first enabled through R libraries. This makes R a perfect choice for data analysis and projection. Members of the R community are very active and supporting and they have a great knowledge of statistics as well as programming. This all gives R a special edge, making it a perfect choice for data science projects.



## 6. Big Data Analytics

1. **RHIPE stands for R and Hadoop Integrated Programming Environment**. It is a software package which allows the R user to create MapReduce jobs that work entirely within the R environment using R expressions. The package uses the Divide and Recombine technique to perform data analytics over Big Data. This integration with R is a transformative change to MapReduce as it allows an analyst to quickly specify Maps and Reduces using the full power, flexibility, and expressiveness of the R interpreted language. Hadoop is the go-to big data technology for storing large quantities of data at economical costs, and R programming language is the go-to data science tool for statistical data analysis and visualization. R and Hadoop combined prove to be an excellent data crunching tool for some seriously big data analytics for business.

2. **ORCH** stands for Oracle R Connector for Hadoop is a collection of R packages which provides predictive analytic techniques, written in R or Java as Hadoop MapReduce jobs, that can be applied to data in HDFS files. It also provides interfaces to work with Hive tables, the Apache Hadoop compute infrastructure, the local R environment, and Oracle database tables. There are several analytic algorithms in ORCH such as linear regression, neural networks for prediction, clustering, matrix completion using low-rank matrix factorization, and non-negative matrix factorization.





# **Statistical Models in R - Some Examples**

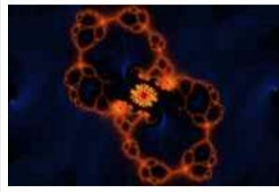


# R Tutorial

An R Introduction to Statistics

[HOME](#)
[DOWNLOAD](#)
[SALES](#)
[EBOOK](#)
[SITE MAP](#)

## Elementary Statistics with R



Ever wonder how to finish your statistics homework real fast? Or you just want a quick way to verify your tedious calculations in your statistics class assignment. We provide an answer here by solving statistics exercises with R.

Here, you will find statistics problems similar to those found in popular college textbooks. The R solutions are short, self-contained and requires minimal R skill. Most of them are just a few lines in length. With simple modifications, the code samples can be turned into homework answers. In addition to helping with your homework, the tutorials will give you a taste of working with statistics software in general, and it will prove invaluable in the success of your career.

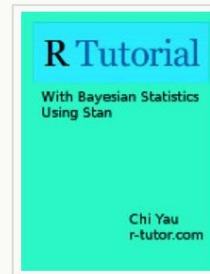
We have included separate introductory tutorials for basic R concepts. The topics are by no means comprehensive. Nevertheless, even if you are not familiar with R, you can go through just the first *R Introduction* page. Then go straight to the statistics tutorials, and only come back for reference as needed.

Please find the topics you are interested in via the [Site Map](#). If you still cannot find what you are looking for, please [contact us](#) and let us know.

- ▶ [Qualitative Data](#)
- ▶ [Quantitative Data](#)
- ▶ [Numerical Measures](#)
- ▶ [Probability Distributions](#)
- ▶ [Interval Estimation](#)
- ▶ [Hypothesis Testing](#)
- ▶ [Type II Error](#)
- ▶ [Inference About Two Populations](#)
- ▶ [Goodness of Fit](#)
- ▶ [Analysis of Variance](#)
- ▶ [Non-parametric Methods](#)
- ▶ [Simple Linear Regression](#)
- ▶ [Multiple Linear Regression](#)
- ▶ [Logistic Regression](#)

Search this site:

[R Tutorial eBook](#)



### R Tutorials

- ⊕ [R Introduction](#)
- ⊖ [Elementary Statistics with R](#)
  - ⊕ [Qualitative Data](#)
  - ⊕ [Quantitative Data](#)
  - ⊕ [Numerical Measures](#)
  - ⊕ [Probability Distributions](#)
  - ⊕ [Interval Estimation](#)
  - ⊕ [Hypothesis Testing](#)
  - ⊕ [Type II Error](#)
  - ⊕ [Inference About Two Populations](#)
  - ⊕ [Goodness of Fit](#)
  - ⊕ [Analysis of Variance](#)
  - ⊕ [Non-parametric Methods](#)
  - ⊕ [Simple Linear Regression](#)

[Qualitative Data](#) >

<http://www.r-tutor.com/elementary-statistics>



# Case Study#1

## **Study the contribution of Fiscal Decentralization on the Local Economic Growth in PNG:**

### **Aim.**

Examine the relationship between fiscal decentralization policy, which has been implemented in all Provinces, and local economic performance.

Is Fiscal decentralization contributes to economic growth?



## Empirical model

$$GRDP_i = \alpha_0 + \alpha_1 FD_i + X_i \beta + \varepsilon_i, i = 1, \dots, 34,$$

Where *i* refers to Province or state

**GRDP** represents growth rate, which is represented by Per capita Gross Domestic Product

**FD** represents indicators of fiscal decentralization in PNG

**Xi** is control variables for economic growth.

The parameters  $\alpha_0$  and  $\alpha_1$  are scalars,  $\beta$  represents a parameter vector to be estimated and

$\varepsilon_i$  is an error term, which is assumed to be normally distributed, homoscedastic, and independent across observations. .

Moreover, this work will consider, population, employment and human capital as controlling variables. For human capital, it is used average years of schooling as a proxy. Therefore, the growth regression can be modified as:

$$GRDP_{(i,t)} = \beta_0 + \beta_1 FD_{(i,t)} + \beta_2 (1 - GDRP_{(i,t)}) + \beta_3 GINI + \beta_4 Pop_{(i,t)} + \beta_5 Emplo_{(i,t)} + \beta_6 Educ_{(i,t)} + \beta_7 Location_{(i,t)} + \varepsilon_i,$$

Where:

- GRDP Per capita Gross Domestic Product
- FD Fiscal Decentralization indicator which involves three Fiscal Decentralization indicators (FD 1, FD 2, FD3) or more
- Initial GRDP Initial level of per capita GRDP each region during period t-1
- **GINI** Gini coefficient
- Pop The number of population



## Case Study#2

Plotting the binomial distribution for  $p = 0.3$ ,  $p = 0.5$  and  $p = 0.8$  and the total number of trials  $n = 60$  as a function of  $k$  the number of successful trials. For each value of  $p$ , determine 1st Quartile, median, mean, and standard deviation.



We begin by calculating the value of k

```
> k <- c(0:60); k
```

```
[1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26  
    27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53  
    54 55 56 57 58 59 60
```

Then, we calculate the distribution for each values of p by using

```
dbP0.3 <- dbinom(k, 60, 0.3); dbP0.3
```

```
dbP0.5 <- dbinom(k, 60, 0.5); dbP0.5
```

```
dbP0.8 <- dbinom(k, 60, 0.8); dbP0.8
```

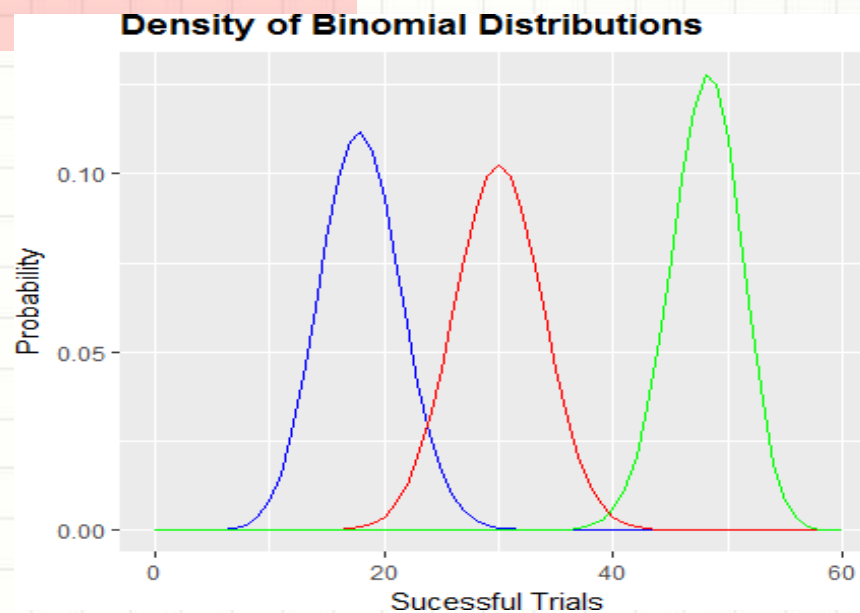
After calculating the value of binomial distributions for each p, we can create the plots.



```

k <- c(0:60); k
dbP0.3 <- dbinom(k, 60, 0.3); dbP0.3
dbP0.5 <- dbinom(k, 60, 0.5); dbP0.5
dbP0.8 <- dbinom(k, 60, 0.8); dbP0.8
df<- data.frame (k, dbP0.3,dbP0.5,dbP0.8);
df ggplot(df, aes(x = k)) +
  geom_line(aes(y = dbP0.3), colour="blue") +
  geom_line(aes(y = dbP0.5), colour = "red") +
  geom_line(aes(y = dbP0.8), colour = "green")
+ ylab(label="Probability") +
  xlab("Sucessful Trials") +
  ggtitle("Density of Binomial Distributions") +
  theme(plot.title = element_text(lineheight=.8,
  face="bold"))

```





For each value of p, determine 1st Quartile, median, mean, standard deviation and the 3rd Quartile.

```
> quantile(dbP0.3)
      0%      25%      50%      75%     100%
4.239116e-32 7.460887e-13 8.357380e-06 9.613404e-03
1.118036e-01
```

The 1st Quartile is 7.460887e-13

```
> quantile(dbP0.5)
      0%      25%      50%      75%     100%
8.673617e-19 3.349811e-10 4.613852e-05 1.227688e-02
1.025782e-01
```

The 1st Quartile is 3.349811e-10

```
> quantile(dbP0.8)
      0%      25%      50%      75%     100%
1.152922e-42 6.585109e-20 1.572006e-07 5.842579e-03
1.278228e-01
```

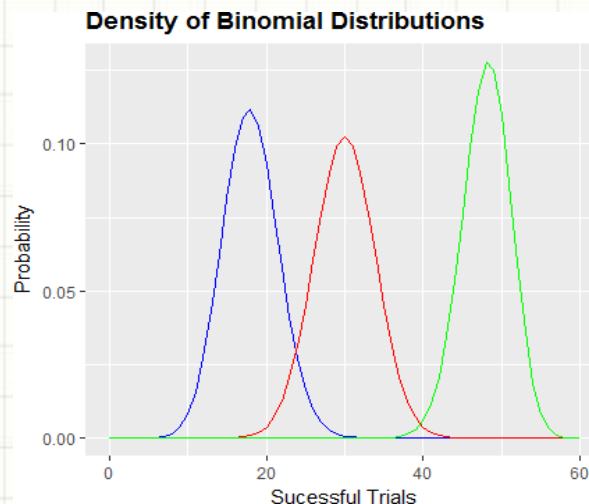
The 1st Quartile is 6.585109e-20

```
median
>median(dbP0.3)
>[1] 8.35738e-06
>median(dbP0.5)
>[1] 4.613852e-05
>median(dbP0.8)
> [1] 1.572006e-07
```

```
mean
>mean(dbP0.3)
> [1] 0.01639344
>mean(dbP0.5)
> [1] 0.01639344
```

```
> mean(dbP0.8)
[1] 0.01639344
```

```
standard deviation
>sd(dbP0.3) [1] 0.03239755
>sd(dbP0.5) [1] 0.03062992
>sd(dbP0.8) [1] 0.03527981
```





## Case Study#3

# Using R Programming for Sustainable Development Goals Examples




# Data Methodology Mapping Results of All SDGs Indicators Relevant for Iraq

IAEG- Category	National Methodology	Data	Number	Percentage
<b>Tier (1)-Ready-Feasible with more effort</b>	Methodology exists	1.Data only available at national level, not at subnational level (e.g. district, city or village level) 2. Indicators require data that need to be improved, adjusted or modified based on existing national data.	94	38%
<b>Tier (2)-Not Ready-Feasible with strong effort</b>	Methodology does not exist	1. Data is not available 2. The indicators require data collected using a new methodology or approach	153	62%
			247	100%



Goal 9: Industry, Innovation and Infrastructure	Indicator 9.5.1. Research and development expenditure as a proportion of GDP
<b>Global Methodology</b>	<p>Research and development expenditure as a proportion of GDP (<math>R\&amp;D_{intensity}</math>) is calculated as:</p> $R\&D_{intensity} = \frac{\text{The total intramural expenditure on R\&D (GERD)}}{GDP} \times 100$
<b>Global Indicator Variables</b>	Computation of the indicator Research and development (R&D) expenditure as a proportion of Gross Domestic Product (GDP) is self-explanatory, using readily available GDP data as denominator.
<b>Global Methodology Data Disaggregation</b>	R&D expenditure can be disaggregated by sector of performance, source of funds, field of R&D, type of research and type of cost. The Frascati Manual provides more details related to these breakdowns (what these breakdowns/classifications are, the purposes, including user needs, the main criteria that are applied, etc).
<b>Nationally Adapted Methodology</b>	NIL
<b>Nationally Adapted Variables</b>	NIL
<b>Level and frequency of collection</b>	NIL
<b>Values per year</b>	NIL
<b>Missing Variables (Data gaps) against Global Methodology</b>	<ol style="list-style-type: none"> <li>1. Total intramural expenditure on R&amp;D</li> <li>2. GDP</li> </ol>
<b>Year data was last collected, frequency of collection</b>	NIL
<b>Proposed Data Source</b>	<ol style="list-style-type: none"> <li>1. Central Statistical Organization (CSO)</li> <li>2. Ministry of Higher Education and Scientific Research</li> </ol>
<b>Potential specialized agencies</b>	UNESCO-UIS
<b>Partner agencies</b>	NIL
<b>Proposed Data Collection methods and Advanced Analytical Tools</b>	<ol style="list-style-type: none"> <li>1. Advanced Mobile Based Data Collection tools support capturing data as text, numbers, dates, GPS, photos, video and audio such as Open Data Kit and Kobo Toolbox. They allow for offline data collection with mobile devices in remote areas with advanced data curation and visualization on the device.</li> <li>2. R and Python are programming languages and free modern software environment for data Science, statistical computing and data visualization. The R and python languages have become very popular among statisticians and data miners (Big Data analytics) and is widely used for advanced data analysis in statistical methodology, survey creation, data cleaning and data analysis.</li> <li>3. Data Visualization tools to visualize large amounts of complex structure and unstructured data and to identify areas that need attention or improvement such as Tableau, Highcharts, Power BI, etc.</li> <li>4. Graphical User Interface softwares; these include specialized applications that use menus, widgets, virtual connections, no programming skills required Applying a hybrid of powerful analytical capabilities Text analysis, combined with predictive modelling and data visualization such as RapidMiner, KNIME, and Orange</li> <li>5. Big Data Analytics and Data Mining tools for large volumes of data using Machine Learning modelling.</li> </ol>
<b>Links</b>	<p><b>Official SDG Metadata URL</b></p> <p><a href="https://unstats.un.org/sdgs/metadata/files/Metadata-09-05-01.pdf">https://unstats.un.org/sdgs/metadata/files/Metadata-09-05-01.pdf</a></p>





# **PART#2. GRAPHICAL INTERFACES- RAPIDMINER SOFTWARE**



# Why RapidMiner Studio?

---

**RapidMiner Studio** is a comprehensive data science platform with visual workflow design and full automation. Data science software provides an integrated environment for data preparation, machine learning, deep learning, **text mining**, and predictive analytics

Drag and drop application in RapidMiner. RapidMiner's a very popular program, and there are several, very expensive commercial versions, but there's also a free community version

The community version can only handle 10,000 rows of data





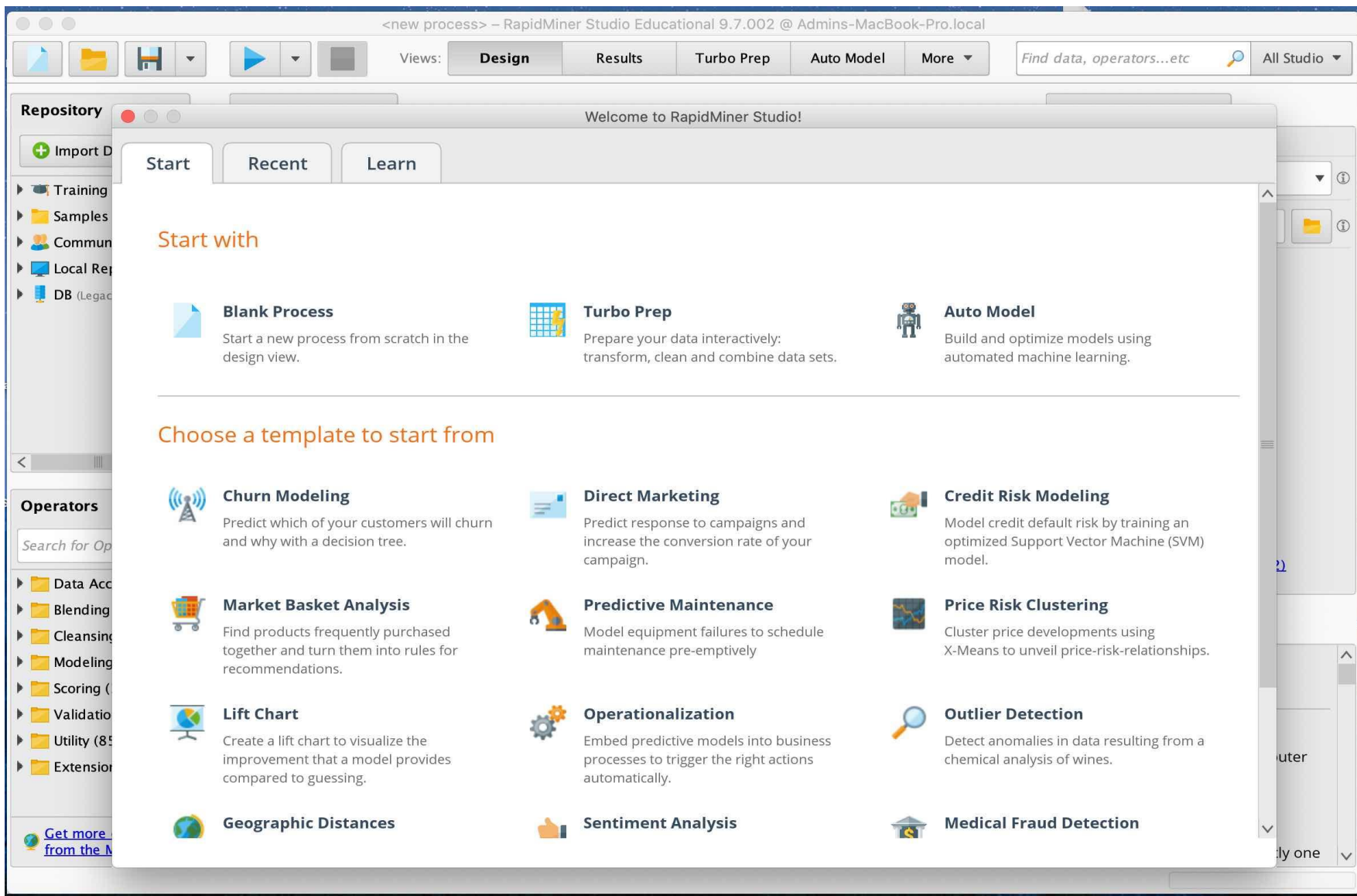
# Studio

Version 9.8

**Loading Advanced File Connectors Extension**  
Educational Edition registered to Abbas Maarroof  
Copyright (C) 2001–2020 RapidMiner GmbH









# Why RapidMiner Studio?

## 1. Visual Workflow Designer

The screenshot displays the RapidMiner Studio Visual Workflow Designer interface. The main workspace shows a workflow for "CHURN MODELING". The workflow consists of the following steps:

- Step 1:** Load a customer dataset that contains customer attributes like:
  - Age
  - Technology used (4G, fiber, etc.)
  - Date since he/she is a customer
  - Average bill last year
  - Number of support calls
  - Did he/she abandon last year?
- Step 2:** Edit, transform & learn (ETL) and prepare data: Mark the target label column (i.e. the churn indicator) and convert the numerical churn column to binary.
- Step 3:** Model validation is key! This cross-validation splits the dataset for training and, then, for independent testing. This splitting is done several times to get a better performance estimate.

The workflow diagram shows the following operators: **Retrieve Customer**, **Set Role**, **Numerical to Binary**, and **Cross Validation**. The **Cross Validation** operator is highlighted, and its details are shown in the right-hand pane.

**Cross Validation**  
Consistency

**Synopsis**  
This Operator performs a cross validation to estimate the statistical performance of a learning model.  
[Jump to Tutorial Project](#)

**Description**  
It is mainly used to estimate how accurately a model (learned by a particular learning Operator) will perform in practice.

The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model. The trained model is then applied in the Testing subprocess. The performance of the model is measured during the Testing phase.

The input ExampleSet is partitioned into  $k$  subsets of equal size. Of the  $k$  subsets, a single subset is retained as the test data set (i.e. input of the Testing subprocess). The remaining  $k - 1$  subsets are used as training data set (i.e. input of the Training subprocess). The cross validation process is then repeated  $k$  times, with each of the  $k$  subsets used exactly once as the test data. The  $k$  results from the  $k$  iterations are averaged (or otherwise combined) to produce a single estimation. The value  $k$  can be adjusted using the number of folds parameter.

The evaluation of the performance of a model on independent test sets yields a good estimation of the performance on unseen data sets. It also shows if "overfitting" occurs. This means that the model represents the training data very well, but it does not generalize well for new data. Thus, the performance can be much worse on test data.

**Differentiation**  
[Split Validation](#)

This Operator is similar to the Cross Validation Operator but only splits the data into one training and one test set. Hence it is similar to one iteration of the cross validation.

[Split Data](#)

This Operator splits an ExampleSet into different subsets. It can be used to manually perform a validation.

**Visual workflow designer**



# Why RapidMiner Studio?

## 2. Connect to Any Data Source:

Edit connection - dwh2

Info Setup Advanced Driver Sources

Database system PostgreSQL

User testaccount

Password \*\*\*\*\*

☒ Configure URL automatically

Host 192.162.1.10

Port 8888

Database dwh

URL jdbc:postgresql://192.162.1.10:8888/dwh

☐ Configure URL manually

Set injected parameters An injected parameter is a parameter whose value is provided by an external source.

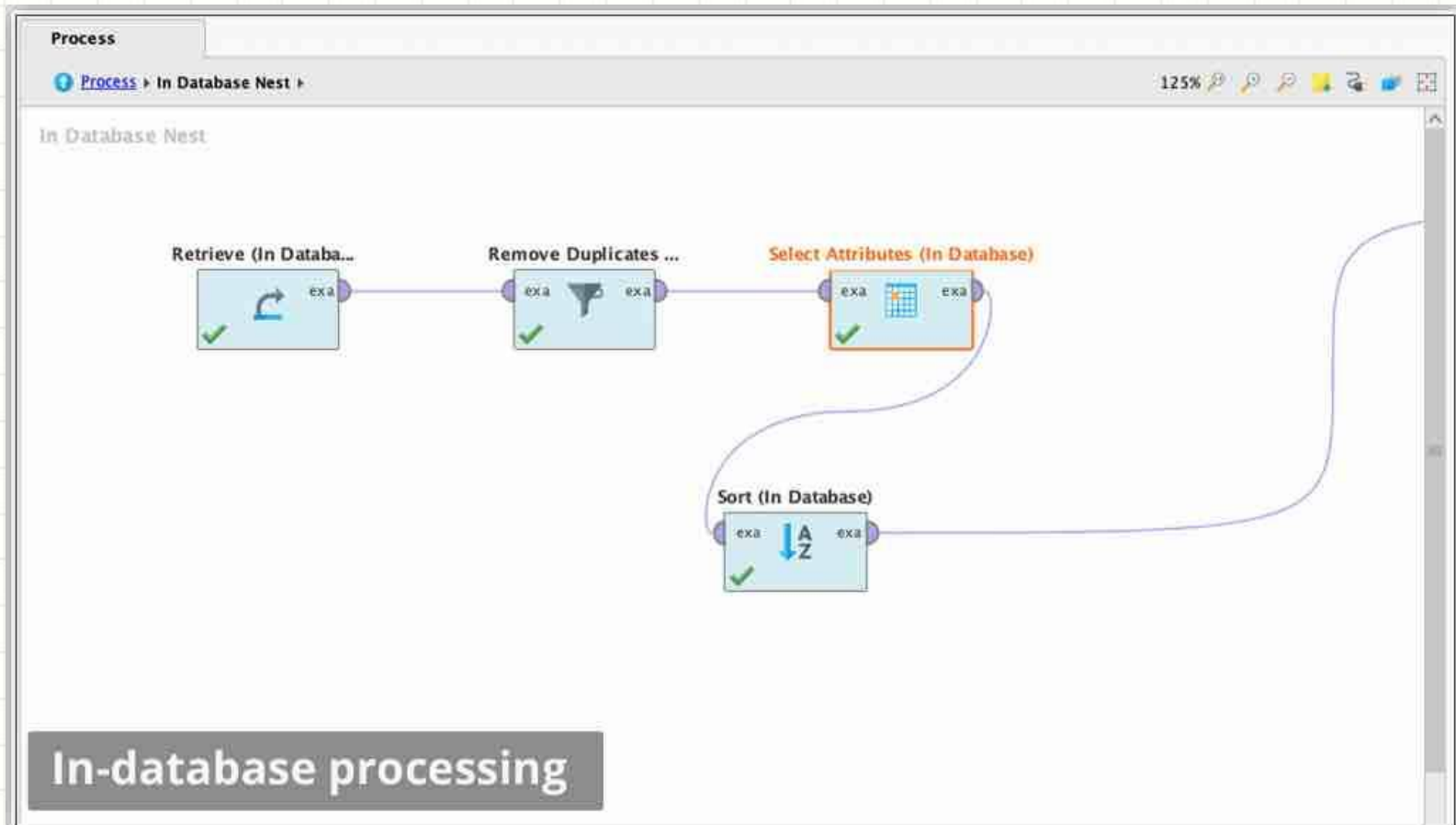
Test connection

**Create database connections** Save Cancel



# Why RapidMiner Studio?

## 3. Automated In-Database Processing





## 4. Data Visualization & Exploration





# Why RapidMiner Studio?

## 5.Data Prep & Blending

The screenshot shows the RapidMiner Studio interface in the 'Turbo Prep' tab. On the left, under 'Data Sets', three data sources are listed: 'Labor-Negotiations', 'Counterparty Risk Data' (selected), and 'Customer Data'. The 'Counterparty Risk Data' is highlighted with a red bar. The main area displays the details for 'Counterparty Risk Data', including a table of data and a list of actions (Transform, Cleanse, Generate, Pivot, Merge).

**Counterparty Risk Data**

Add new data sets on the left. Details for the selected data are shown below. You can change the data with the following actions.

✗ TRANSFORM ✎ CLEANSE 📊 GENERATE Σ PIVOT ➤ MERGE

Default Category	Long Term F... Number	Working Cap... Number	Debt Cash Fl... Number	Liability to E... Number	Net Debt to E... Number	Debt to Capit... Number	Long Term D... Number	Long Term Number
No	18.535	70.224	2.222	0.409	0.345	0.056	3.341	2.923
No	1.268	68.543	2.276	0.152	0.099	0.212	5.740	1.061
No	26.770	7.496	0.843	0.470	1.186	0.073	2.641	1.038
No	5.789	17.880	1.365	0.046	0.138	0.010	1.123	0.274
No	11.864	5.164	0.061	0.102	0.047	0.042	5.710	0.298
No	6.661	16.898	2.093	0.107	0.117	0.064	0.026	0.462
No	3.198	102.558	0.406	0.069	0.113	0.071	1.763	1.323
No	1.926	42.528	1.909	0.026	0.488	0.012	5.006	0.622
No	2.497	24.835	2.138	0.012	0.377	0.019	3.764	0.446
No	5.427	52.886	4.327	0.123	0.304	0.009	0.007	0.701
No	0.630	32.128	3.454	0.070	0.049	0.041	0.838	0.453
No	4.186	44.365	3.504	0.047	0.241	0.228	1.344	1.288
No	6.822	32.543	0.784	0.024	0.343	0.204	0.165	1.192



# Why RapidMiner Studio?

## 6. Visual & Automated Machine Learning






```
ExampleSet (/Data/CustomersData)
```

### Plot

### Plot type

 Histogram

### Value columns

Age

Color

Number of Bins

10

Plot style 

[Add new plot](#)

## General

### X-Axis

### Y-Axis

### Title

### Legend

### Tooltip

## Charts

## Maps

Select what plot to display. The selection may be limited if you have more than one plot!

## Scatter

### Scatter Matrix

### Scatter 3D

Bar (Column)

Bar (Horizontal)

## Streamgraph

### Histogram

### Boxplot

## Bell Curve

### Heatmap

### Treemap

## Sunburst

Pie

## Funnel

Pyramid

### Packed Bubble

Parliament

## Pareto

Range (Column)

Range (Error Bar)

Range (Line)

Range (Step)

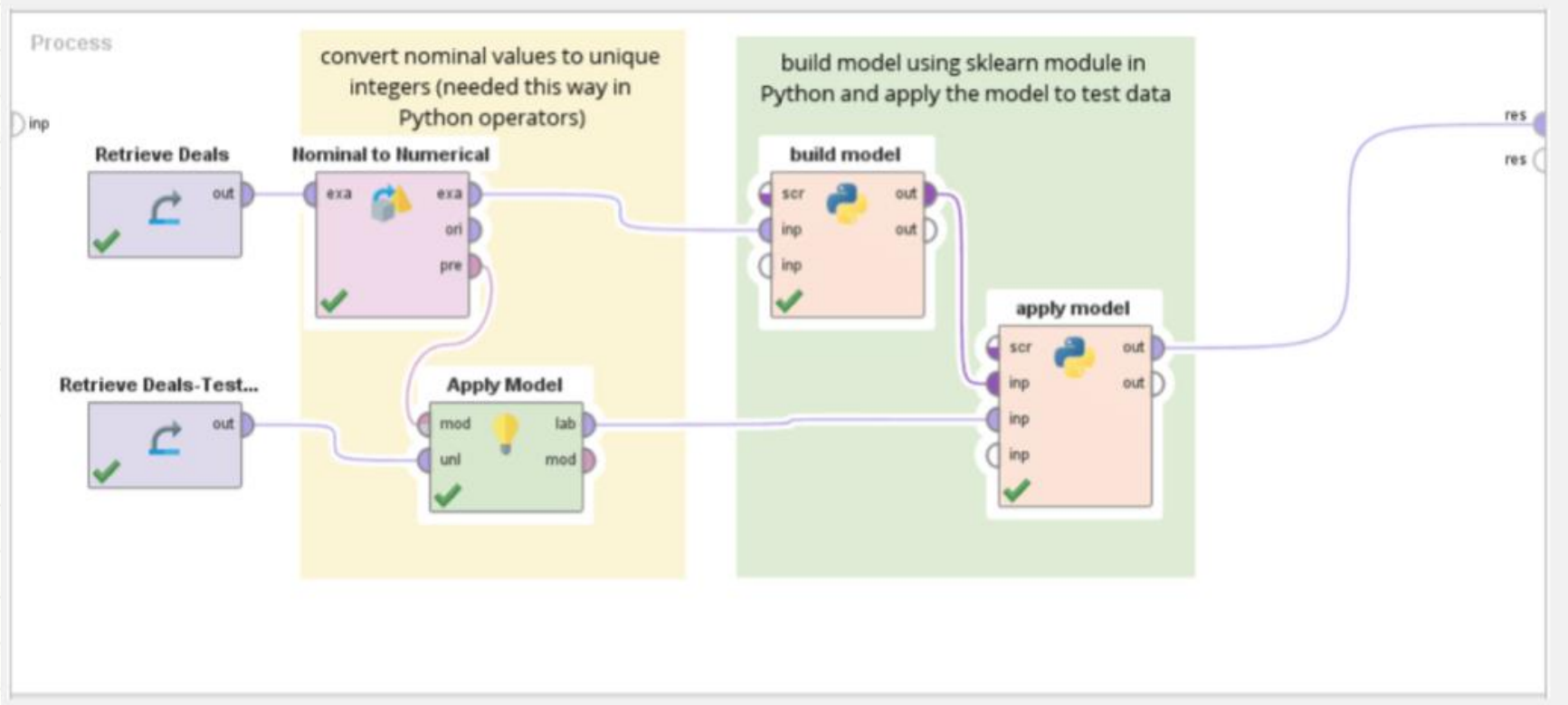
Range (Spline)

## Vector



# Why RapidMiner Studio?

## 7. Get More From R & Python Code





# Case Study#1

## customer churn rate

Customer churn rate is one of the most important metrics for businesses to track.

To identify your company's churn rate, choose a period of time you want to measure and identify the following values:





- Number of customers at the start of the period (X)
- Number of customers lost during that period (Y)

Then, use the following formula to determine your customer churn rate (Z) as a percentage

**Customer churn rate formula**

$$(Y/X) * 100 = Z$$





Repository

Import Data

Training Resources (connected)

Samples

Community Samples (connected)

Demo (Local)

Local Repository (Local)

DB (Legacy)

Operators

Search for Operators

Data Access (53)

Blending (82)

Cleansing (29)

Modeling (166)

Scoring (14)

Validation (30)

Utility (86)

Extensions (164)

[Get more operators from the Marketplace](#)

Import Data - Select the data location.

Select the data location.

Data Exmaples

Bookmarks	File Name	Size	Type	Last Modified
★ --- Last Directory	Customer+Data.xlsx	25 KB	Customer+Data.xlsx	Aug 25, 2020

Customer+Data.xlsx

All Files

The selected file will be imported as: Excel [Change](#)

Previous

Next

Cancel

Log verbosity level.



Result History

ExampleSet (//Demo/Data/Customer+Data)

ExampleSet (//Demo/Data/Customer+Data)

Repository

Import Data



Data



Statistics



Visualizations



Annotations

Open in Turbo Prep

Auto Model

Filter (996 / 996 examples):

all

Row No.	Churn	Gender	Age	Payment M...	LastTransa...
1	loyal	male	64	credit card	98
2	churn	male	35	cheque	118
3	loyal	female	25	credit card	107
4	?	female	39	credit card	177
5	loyal	male	39	credit card	90
6	churn	female	28	cheque	189
7	loyal	female	21	credit card	102
8	loyal	male	48	credit card	141
9	churn	female	70	credit card	153
10	loyal	male	36	credit card	46
11	loyal	male	22	credit card	51
12	?	female	53	cash	183
13	loyal	male	27	cash	137
14	loyal	male	22	cash	147
15	churn	female	49	credit card	158
16	churn	female	24	cash	162
17	loyal	male	45	credit card	55
18	loyal	male	45	credit card	160

ExampleSet (996 examples, 1 special attribute, 4 regular attributes)

Why there is special attribute -This column need to model to predict



RapidMiner Studio

File Edit Process View Connections Settings Extensions Help

<new process\*> – RapidMiner Studio Educational 9.7.002 @ Admins-MacBook-Pro.local

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc

All Studio

Result History

ExampleSet (//Demo/Data/Customer+Data)

ExampleSet (//Demo/Data/Customer+Data)

Data

Statistics

Visualizations

Annotations

Name	Type	Missing	Statistics	Filter (5 / 5 attributes):	
Label Churn	Polynomial	96	Least churn (322)	Most loyal (578)	Values loyal (578), churn (322)
Gender	Polynomial	0	Least female (448)	Most male (548)	Values male (548), female (448)
Age	Integer	0	Min 17	Max 91	Average 45.616
Payment Method	Polynomial	0	Least cheque (68)	Most credit card (649)	Values credit card (649), cash (2
LastTransaction	Integer	0	Min 1	Max 223	Average 111.072

Showing attributes 1 – 5

Examples: 996 Special Attributes: 1 Regular Attributes: 4

pository

Training Res

Samples

Community

Demo (Local)

Connections

Data

Custom

Processes

Textanah

Local Repos

Temporary

DB (Legacy)



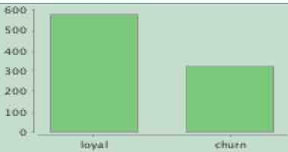
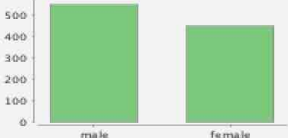
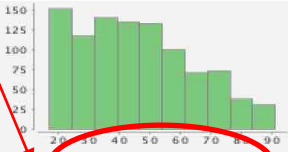

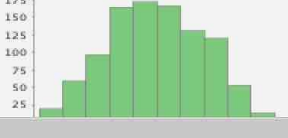
# Hyper Link

RapidMiner Studio File Edit Process View Connections Settings Extensions Help

<new process\*> - RapidMiner Studio Educational 9.7.002 @ Admins-MacBook-Pro.local

Views: Design Results Turbo Prep Auto Model Deployments Find data, operators...etc All Studio

Result History ExampleSet (//Demo/Data/Customer+Data) ExampleSet (//Demo/Data/Customer+Data)

Name	Type	Missing	Statistics	Filter (5 / 5 attributes): Search for Attribute.
Churn	Polynomial	96	 Least churn (322) Most loyal (578) <a href="#">Open visualizations</a>	
Gender	Polynomial	0	 Least female (448) Most male (548) <a href="#">Open visualizations</a>	
Age	Integer	0	 Min 17 Max 91 Average 45.61 <a href="#">Open visualizations</a>	
Payment Method	Polynomial	0	 Least cheque (68) Most credit card (64) <a href="#">Open visualizations</a>	
LastTransaction	Integer	0	 Min 1 Max 223 Average 111.0	

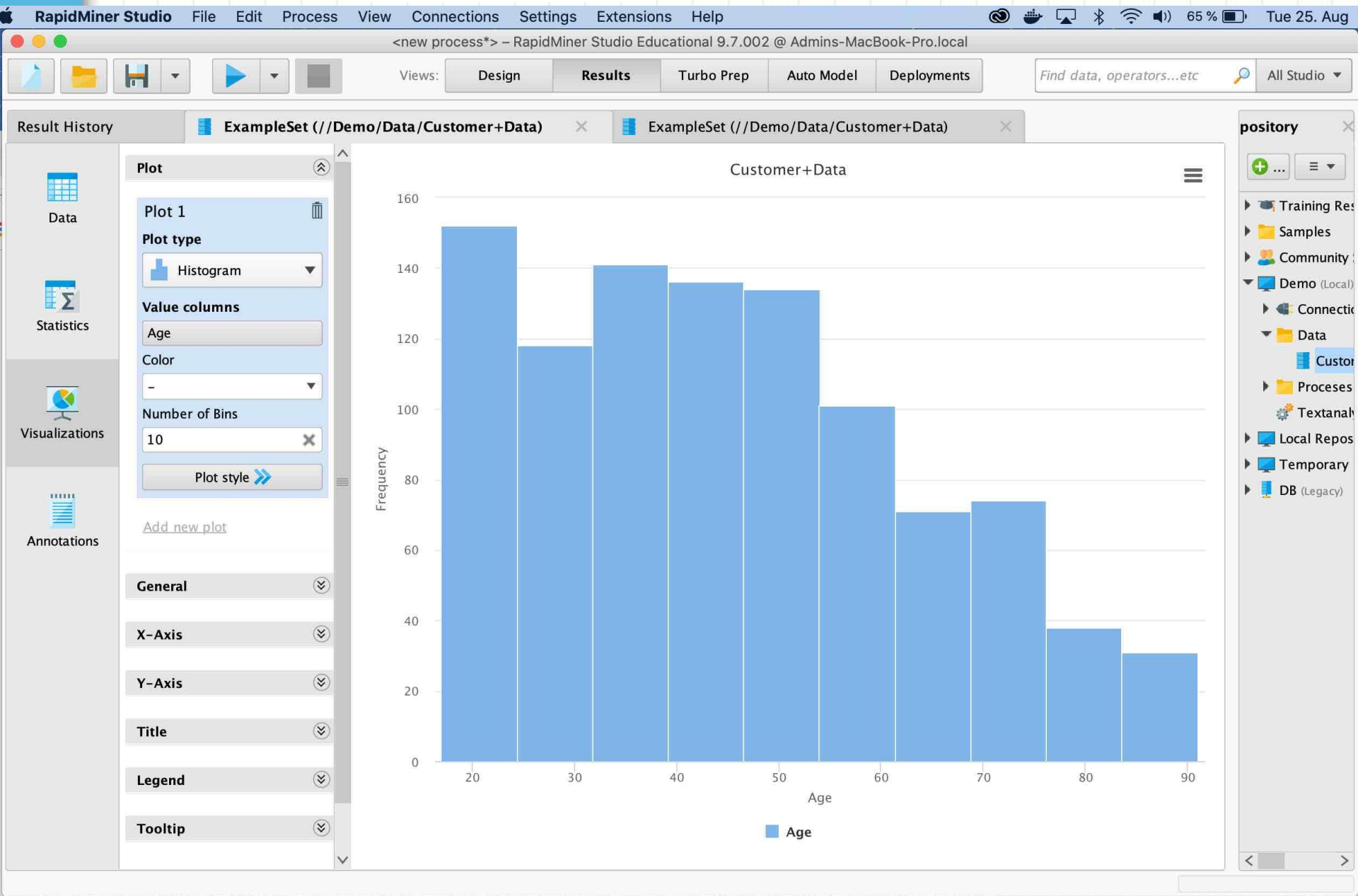
Showing attributes 1 - 5 Examples: 996 Special Attributes: 1 Regular Attributes: 4

pository

- Training Res
- Samples
- Community
- Demo (Local)
  - Connectic
  - Data
    - Custom
  - Processes
  - Textanah
- Local Repos
- Temporary
- DB (Legacy)



# Visualize the data





# Visualize the data

RapidMiner Studio File Edit Process Views: Design Results Turbo Prep Auto Model Deployments Find data, operators...etc All Studio

Result History ExampleSet (/Data/Date (Customer Data)) ExampleSet (/Data/Date (Customer Data))

**Plot**

Plot 1

Plot type

Histogram

Value columns

Age

Color

-

Number of Bins

10

Plot style >>

Add new plot

**General**

X-Axis

Y-Axis

Title

Legend

Tooltip

**Charts**

Select what plot to display. The selection may be limited if you have more than one plot!

Line Step Line Spline Area Step Area Spline Area

Scatter Scatter Matrix Scatter 3D Bar (Column) Bar (Horizontal) Streamgraph

**Histogram** Boxplot Bell Curve Heatmap Treemap Sunburst

Pie Funnel Pyramid Packed Bubble Parliament Pareto

Range (Column) Range (Error Bar) Range (Line) Range (Step) Range (Spline) Vector

**Maps**

pository

- Training Res
- Samples
- Community
- Demo (Local)
  - Connectio
  - Data
    - Custom
  - Processes
  - Textanah
- Local Repos
- Temporary
- DB (Legacy)



Views:

Design

Results

Turbo Prep

Auto Model

Deployments

Find data, operators...etc

All Studio

Result History

ExampleSet (//Demo/Data/Custom+Data)

ExampleSet (//Demo/Data/Custom+Data)

pository



Data



Statistics



Visualizations



Annotations

Plot

Plot 1

Plot type

Scatter

X-Axis column

Age

Value column

LastTransaction

Color

Churn

Size

-

Jitter

Regression interpolation

None

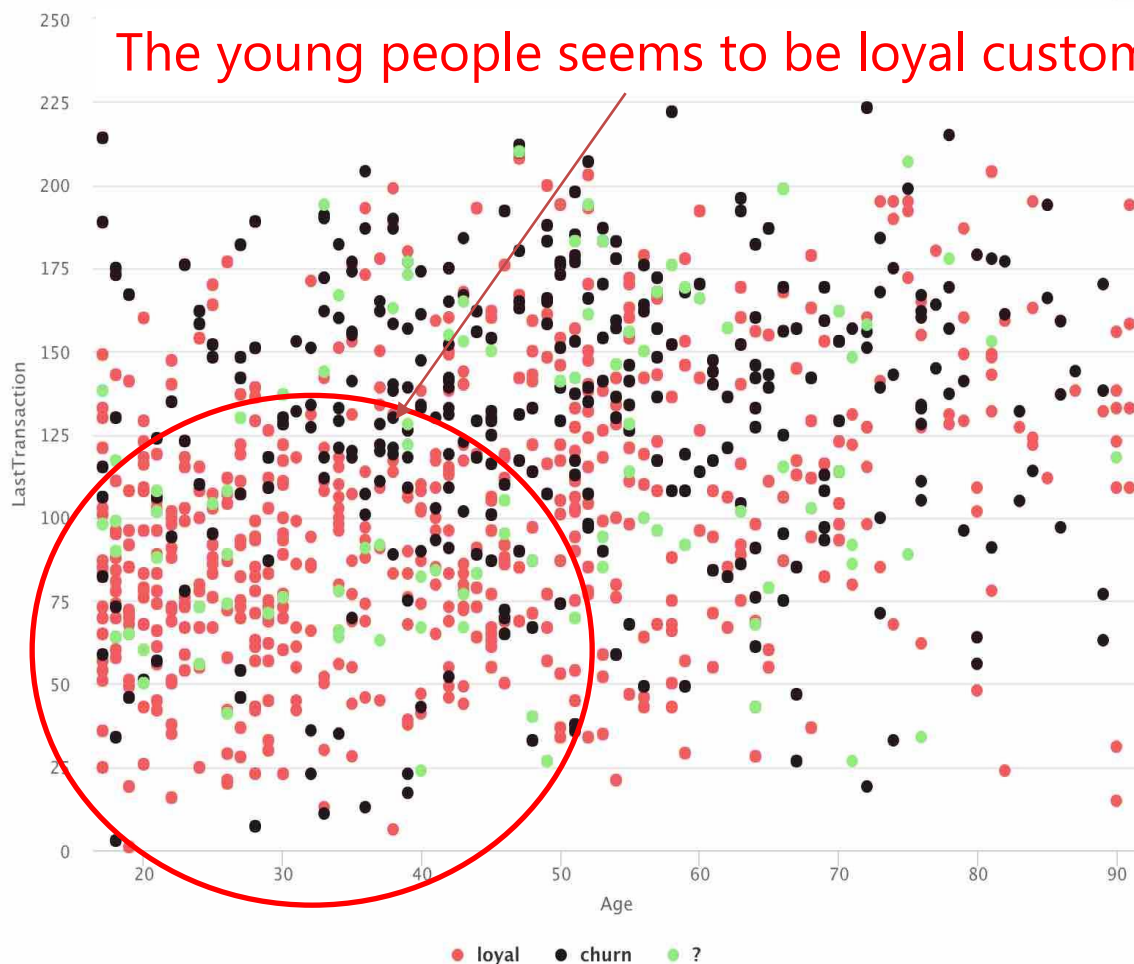
Plot style

General

X-Axis

Y-Axis

Customer+Data



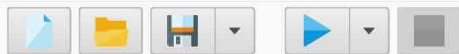


## Case Study#2

# Twitter Data Analysis Using RapidMiner

<https://twitter.com/RapidMiner>





Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

### Repository

+ Import Data

- Community Samples (connected)
- Demo (Local)
- Import Text (Local)
- Linear Regression (Local)
- Local Repository (Local)
- Temporary Repository (Local)
- Twitter Analysis (Local)
  - Connections
    - Twitter (8/26/20 11:37 PM - 1 kB)
  - Data

### Operators

Twitter

- Data Access (4)
- Applications (4)
  - Twitter (4)
    - Search Twitter
    - Get Twitter User Statuses
    - Get Twitter User Details
    - Get Twitter Relations

No results were found.

### Process

Process

Get Twitter User Statuses

Process diagram showing a flow from 'Inp' to 'Get Twitter User Statuses' (with a green checkmark) and then to 'res'.

#### Recommended Operators

Operator	Popularity
Search Twi...	31%
Write Excel	26%
Select Attri...	22%

### Parameters

#### Get Twitter User Statuses

connection source: repository

connection entry: inections/

query type: name

user: **Rapidminer**

limit: 100

since id:

max id:

[Hide advanced parameters](#)

[Change compatibility \(9.6.000\)](#)

### Help

#### Get Twitter User Statuses

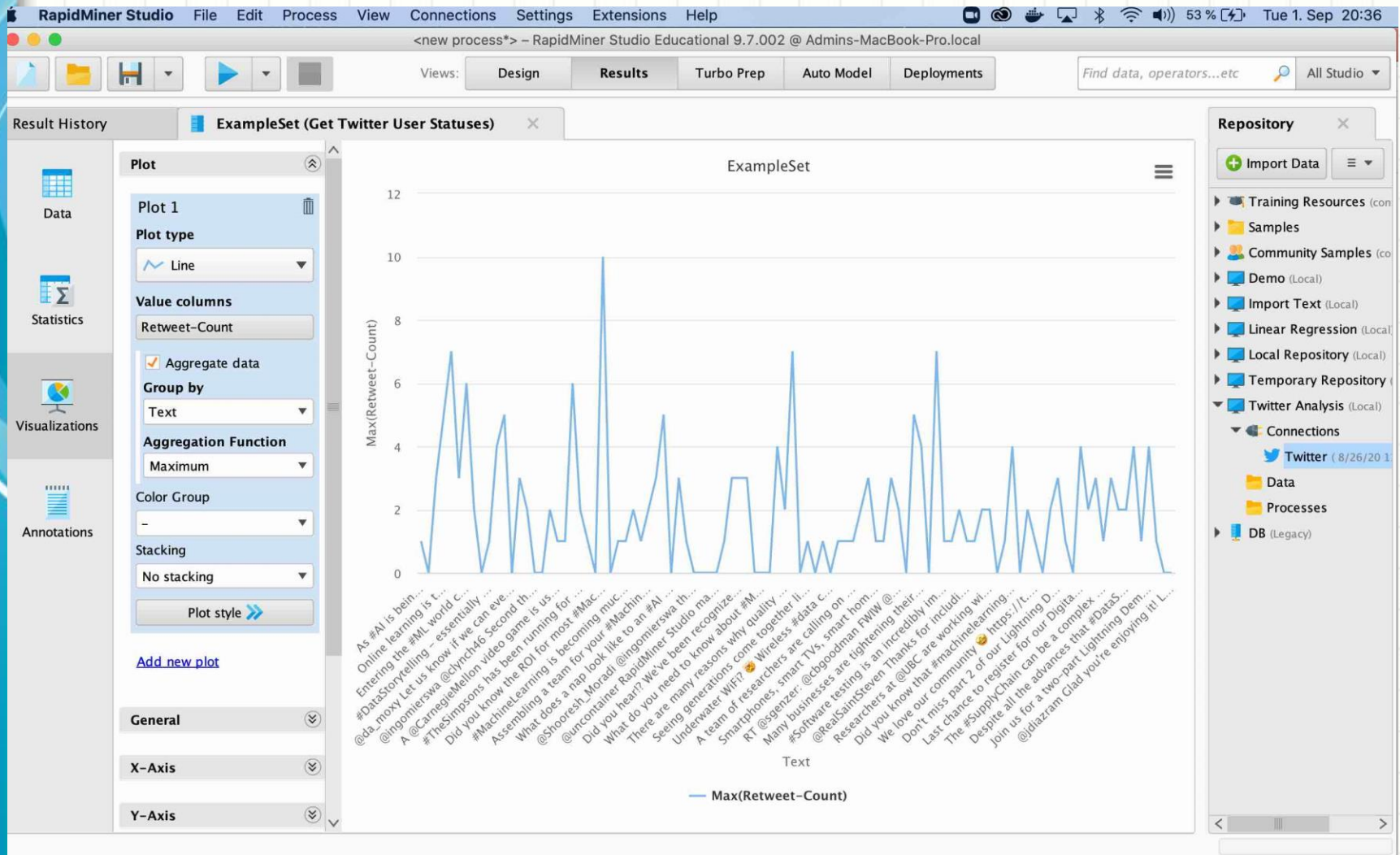
Social Media

Tags: [Connection](#), [Twitter](#)

#### Synopsis

This operator searches for Twitter statuses of a specific user.









**Thank you**