

The logo consists of two white triangles pointing towards each other, forming a larger, irregular shape. The top triangle is smaller and positioned above the bottom triangle, which is larger and extends further to the right.

INSTITUT DES
ACTUAIRES



Data
Innovation Lab



Milliman

Predictive Analytics

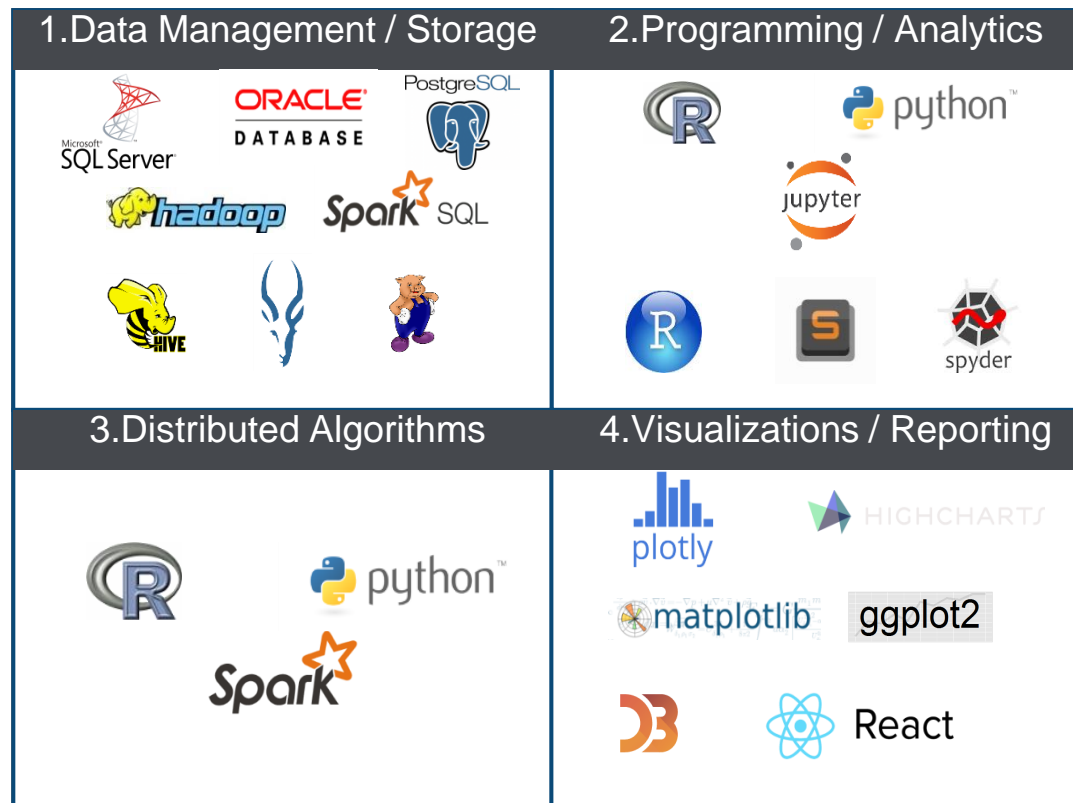
Which technologies? How to use them?
Which applications to insurance?

Eric.O Lebigot, Yves-Richard Hong T.H, Rémi Bellina, Antoine Ly

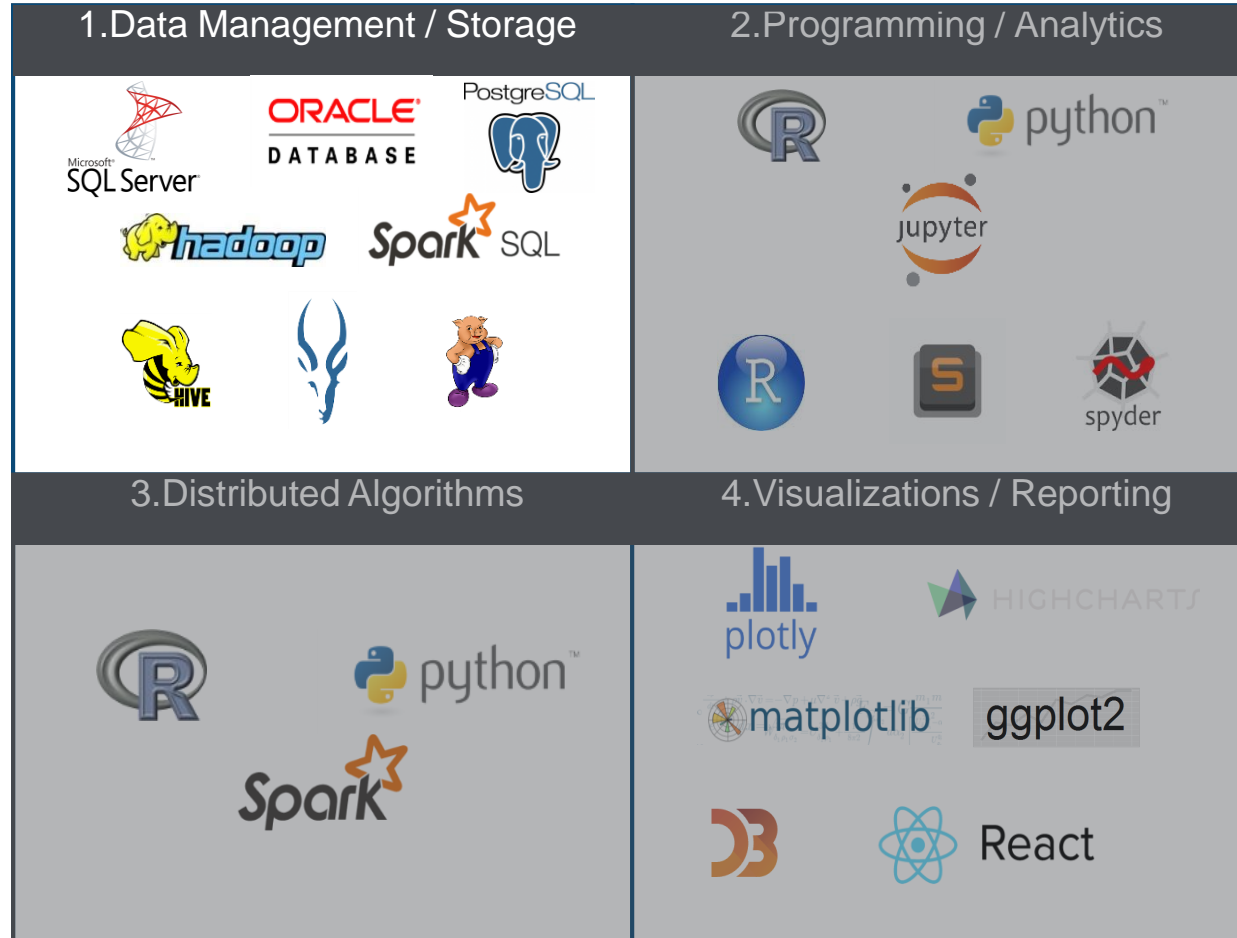
November, 8th 2016

Introduction

- Why those technologies ?
- Which one to use? When?



1. Data Management / Storage



1. Data Management / Storage

RDBMS vs distributed storage Hadoop

Query management	System management	Storage
SQL manager	System tables(meta data)	Data tables

RDBMS

- Everything integrated in one system
- Deals with relational databases

Technologies to set-up a RDBMS



How to manage your RDBM?

SQL (SQLite, Dbvisualizer etc.)

Query management	System management	Storage
Several devices to manage queries (SQL, noSQL)	Meta data on several devices	HDFS (redundant)

Hadoop

- Everything is distributed on several systems (quicker multi-accesses)
- Deals with different type of databases (structured or unstructured)
- Redundance, scalability, streaming

Technologies to set-up Hadoop



How to manage Hadoop?

SQL like (Hive, Impala, Pig) or use API (Python, R)

1. Data Management / Storage

Data Lake

Storage technologies made for Big-Data volumetry and fail-safe using redundant storage

Hadoop HDFS:

Limitless size storage thanks to a distributed storage, and fail-safe thanks to a redundant storage



Parquet:

Compressed file format working with Hadoop HDFS, data is analyzed and optimally compressed to optimize performance.



Data-sources



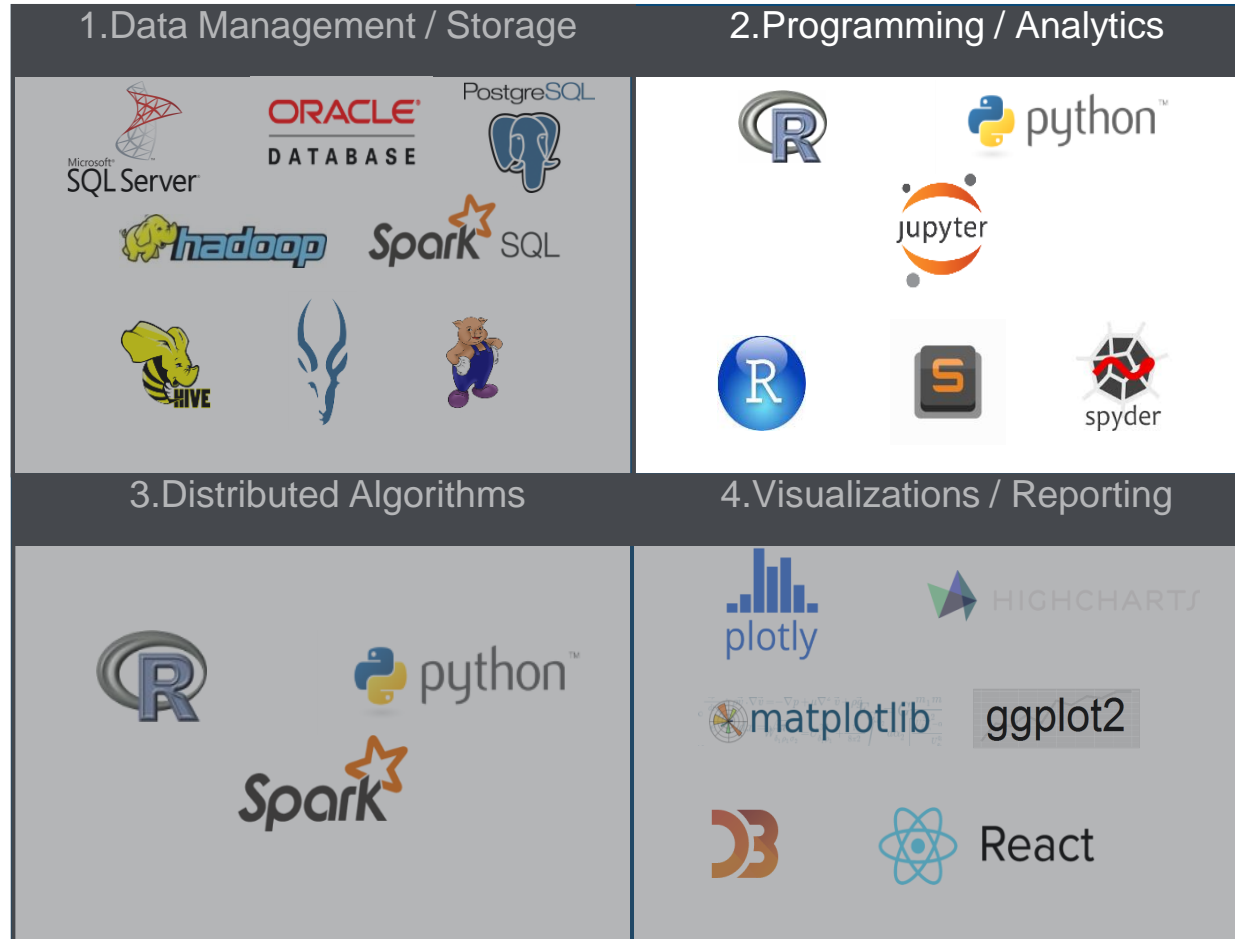
Data is imported in the Hadoop HDFS Data-Lake

Data-Lake



- All data-sources are merged into one single data storage system
- Unified storage support
- Prepare the data to be reshaped and enriched

2. Programming / Analytics



2. Programming / Analytics

R, Python

- What is R/Python compared to more traditional tools (SAS, SPSS etc.)? How to chose between these technologies?



Designed for **statistical computing** and graphics. Not the easiest to start programming

A large and rich diversity of statistical and machine learning modelling libraries.

Some extensions as RShiny allow you to prototype dashboard. Huge development with R could be less easy.

Not always optimized.



Learning curve



Modelling



Prototyping



Speed

Its design philosophy emphasizes **code readability** which makes Python a good starting point.

Less used for statistical modelling but largely adopted by the machine learning community, in particular in semantic and image analysis (deep neural network).

Object oriented language, you can easily design your own application and integrate it in an operational process.

Python presents better performance than R.

2. Programming / Analytics

Powerful tool to save time in analysis



Jupyter: Web user application for Julia Python and R



Save time: a unique tool to code, comment and plot



Share your work with multi export format (pdf, HTML, markdown)



Easily explore your data and test solutions

```

jupyter R_illustration_Propale Last Checkpoint: 01/19/2016 (unsaved changes)
File Edit View Insert Cell Kernel Help | R O
version: string 3. version 3.2.3 (2015-12-18)
nickname Wooden Christmas-Tree

Analyse des données
a. Analyses monovariées
b. Analyses multivariées
c. Réduction de dimension

Chargement des données
In [2]: df=read.csv("Base_test_outil_string.csv", sep=";", header=TRUE)
In [3]: df=data.frame(df)

Aperçu général des données
In [4]: summary(df)
Out[4]:
      V1          V2          V3          V4
Min.   :-3.19711  Min.   : 337.0  Min.   :0.000  Min.   :-5.061
1st Qu.:-0.63027  1st Qu.: 482.5  1st Qu.:2.000  1st Qu.: 5.372
Median : 0.04595  Median : 619.0  Median :4.000  Median :10.265
Mean   : 0.03559  Mean   : 638.9  Mean   :4.507  Mean   : 8.425
3rd Qu.: 0.68995  3rd Qu.: 782.0  3rd Qu.:7.000  3rd Qu.:12.787
Max.   : 3.35139  Max.   :1000.0  Max.   :9.000  Max.   :13.998

      V5          V6          V7          V8
Min.   : 0.000  Min.   :-0.04063  Min.   :-0.9986  Min.   :0.000
1st Qu.: 0.200  1st Qu.: 0.61649  1st Qu.:-0.3445  1st Qu.:5.000
Median : 0.900  Median : 1.00626  Median : 0.2314  Median :7.000
Mean   : 30.422  Mean   : 1.00681  Mean   : 0.1358  Mean   :6.202
3rd Qu.: 4.337  3rd Qu.: 1.41099  3rd Qu.: 0.6233  3rd Qu.:8.000
Max.   :10475.072  Max.   : 2.19571  Max.   : 0.9980  Max.   :9.000

      V9          V10         V11         V12         Y
Min.   :1.001  Min.   :0.004772  a:388  dimanche:142  Min.   : 0.0767
1st Qu.:1.302  1st Qu.:0.659379  c:611  jeudi   :143  1st Qu.: 0.0983
Median :1.670  Median :0.060583  jeudi  :143  Median : 0.1201
Max.   :1.970  Max.   :0.060583  jeudi  :143  Max.   : 0.1201
    
```

Notebook demo

2. Programming / Analytics

R, Python for Machine Learning

- Which one to choose to explore machine learning algorithm?



Each algorithm has its own libraries. You could then choose the implementation you prefer.

Libraries:

GLM: **stats**

CART: **tree, rpart**

Random Forest: **caret, randomForest**

Gradient boosting: **GBM, xgboost**
etc.

You not always know which one to choose.
Few centralized libraries (caret) with everything.

Not recommended for neural network



As in R many libraries exist. The one which makes reference in machine learning is **scikit-learn**. You can design quickly many algorithms and prepare your data with this package.

Packages:

Machine Learning (Preprocessing, clustering, Modelling, Dimensionality Reduction): **scikit-learn**

Statistical models: **statsmodels**

Neural Networks: **Keras, TensorFlow, Theano**

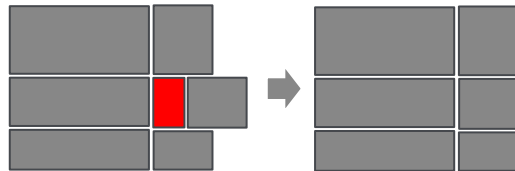


Modelling

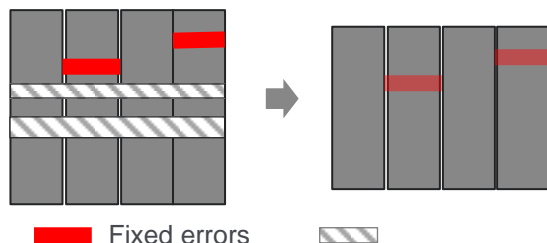
2. Programming / Analytics

Example of Python use for data-cleaning

1.1 Data structure



1.2 Data content



Data-cleaning

Identify and fix **data-structure** issues, such as **irregular column numbers**:

- This can occur during file merging phase if the initial files have different data-structures.
- Or mostly when working with CSV files due to CSV separators.

```
def clean_sep(line):
    ...
    return clean_line
```

Application of the *clean_sep* function on each line

Identify **data-content** issues:

- Identify types for each variable.
- Identify anomalies in the values taken by each variables. (“#Values!” Errors for instance)

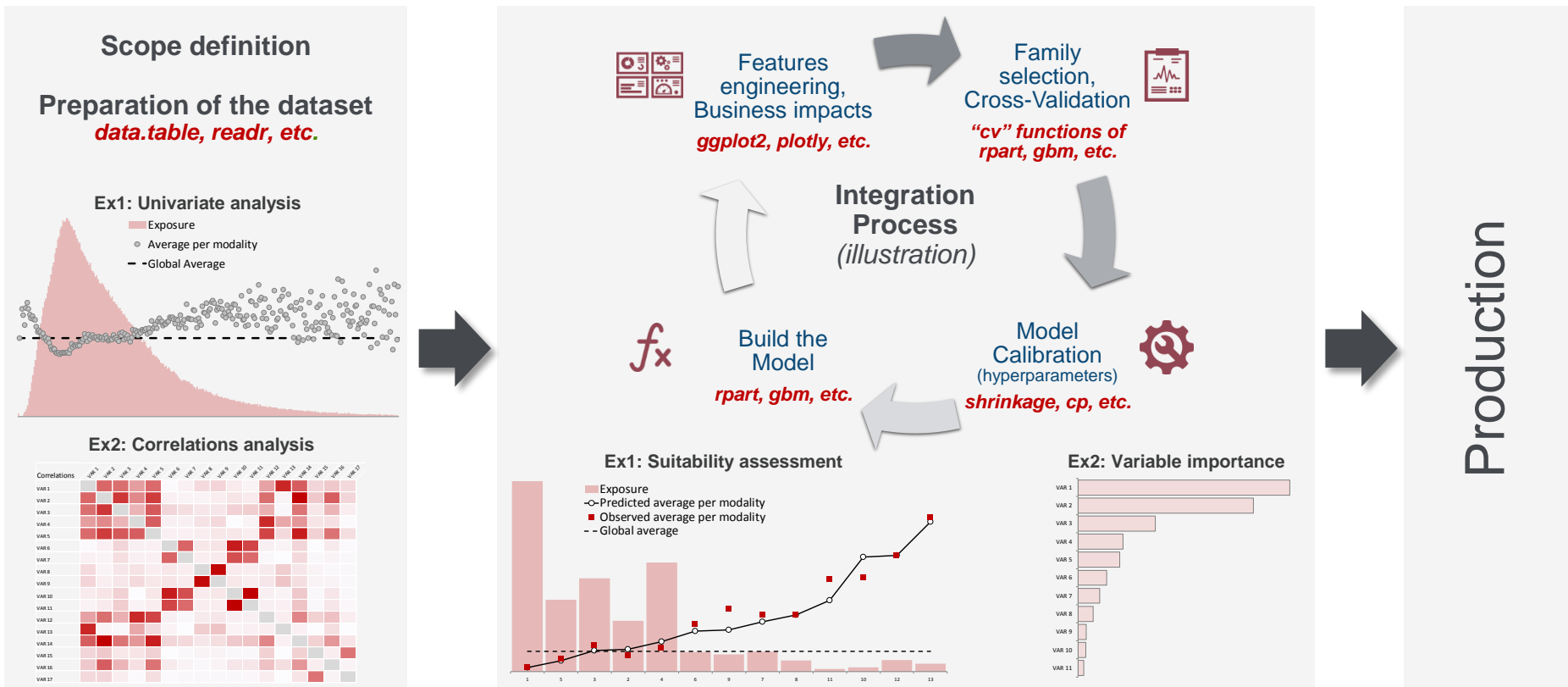
```
def detect_type(line):
    ...
    return typelist
```

Application of the *detect_type* function on each line

2. Programming / Analytics

Insurance case studies (1/2)

- How to choose and compute a model within **an insurance context using R?**
 - **Classical Statistics** (emphasize model) vs **Machine Learning** (distribution free, focused on making accurate predictions)
 - **Link between modeling and financial objectives**



2. Programming / Analytics

Insurance case studies (2/2)

- The **application domain** of Data Science is **very large within insurance**.

Binary Target

- Lapse Rate (churn, lapse between UC to Euro, etc.)
- Fraud analysis
- Etc.

Quantitative Target

- Pricing (pure premium)
- Customer Value
- Etc.

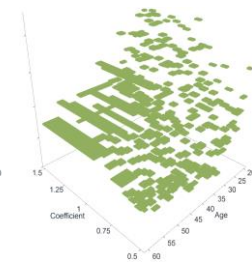
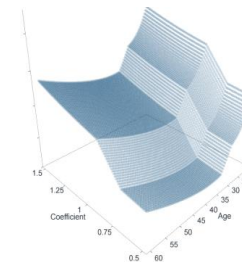
Specific Studies

- Multi equipment analysis
- Replace an existing pricing structure
- Claims development analysis
- Etc.

Illustration

Predict the **pure premium** within a **motor insurance** dataset (Property Damage Liability for instance)

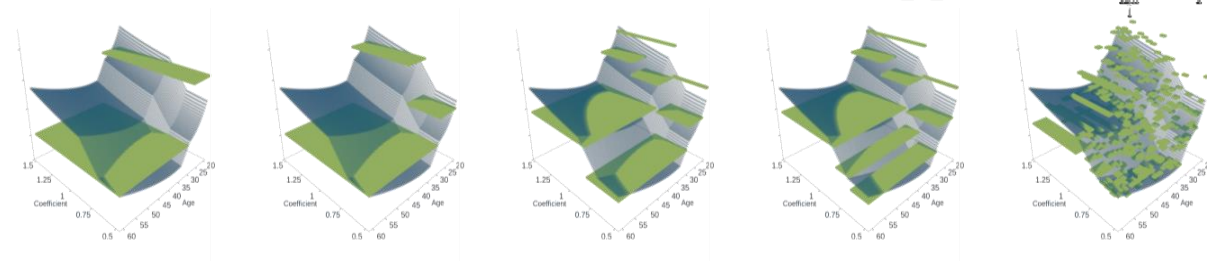
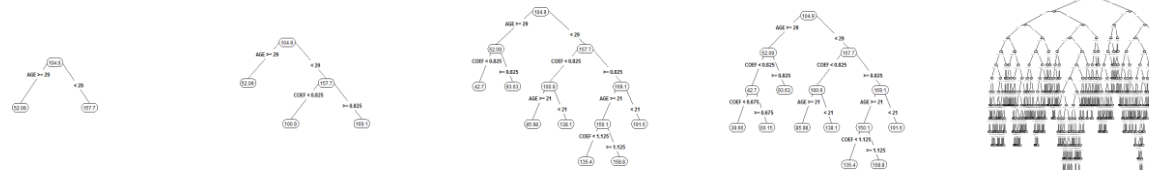
DATABASE				IDEAL
X_1	X_2	W	Y_{obs}	Y_{theo}
COEF	AGE	Weight	Target	Theoretical Pure Premium
0.50	34	0.5	23.1	17.7
0.85	32	0.8	6.5	60.9
1.05	20	1	287.8	172.2
1.00	20	1	150.3	170.5
0.50	36	1	12.8	19.1
0.50	36	0.5	123.6	16.7
⋮	⋮	⋮	⋮	⋮
0.50	44	1	20.9	23.7



What is **wanted**

What is only **seen**

How to **select the best model?**



2. Programming / Analytics

In real life: R, Python

- How to install / try them ?
 - Python: Anaconda distribution <https://www.continuum.io/downloads>
 - R: CRAN <https://www.r-project.org/>
- Use an IDE (Integrated Development Environment)
 - Python: Spyder (provided with Anaconda), PyCharm
 - R: Rstudio <https://www.rstudio.com/>
 - Other: Visual Studio, Eclipse etc.
- Advantages of IDE:
 - Structure your projects and save time
- Text editors + Terminal: Notepad++, Sublime Text



2. Programming / Analytics

In real life: Case study

- Automatic reporting thanks to Rstudio Knit()



HTML, Pdf, .doc



Script template (Rmarkdown)

```

1 title: "Test à mirror of Linear Regression"
2 output: html_document
3 ----
4
5 Load of the data set used for the test of the 'linear_model' algorithm in order to perform a mirror algorithm to compare the results from each method.
6
7 [ ]
8 [ ]
9
10 [ ]
11 source: "rmarkdown/antenne_1y/m11/mon_desc/test_result/2_algor/linear_regression/mon/ps1"
12 data_read_csv("C:/Users/.../R_data/tested_algo/data_linearreg_train.csv", header=TRUE, sep=";", stringsAsFactors=TRUE)
13 head(data)
14
15
16
17
18 Then we need to clean the data to scale and binarize it as we did with the tool:
19
20 [ ]
21
22 data[,c(2,3,4,7,8,9,10)] = scale(data[,c(2,3,4,7,8,9,10)], center=TRUE, scale=TRUE)
23 summary(data)
24
25 we load the test data set :
26
27 Then we can fit our 'linear_model' :
28
29 [ ]
30 linear_model = fit(data=data[,2:11], formula = Y ~ ., na.action = na.omit)
31 summary(linear_model)
32
33
34
35
36 [ ]
37 data_test_read_csv("C:/Users/.../R_data/tested_algo/data_linearreg_test.csv", header=TRUE, sep=";", stringsAsFactors=TRUE)

```



Daily reports Word

Rapport de [REDACTED]

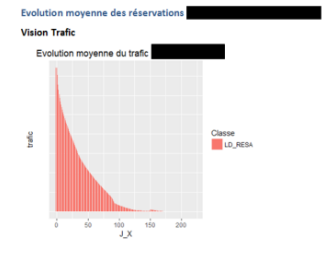
07 octobre, 2016, 16:38

Ce document a été généré automatiquement à partir des données consolidées sur l'entité. Merci de bien lire la documentation associée au Data Management avant l'utilisation de ce rapport.

Description des données utilisées pour la génération du rapport

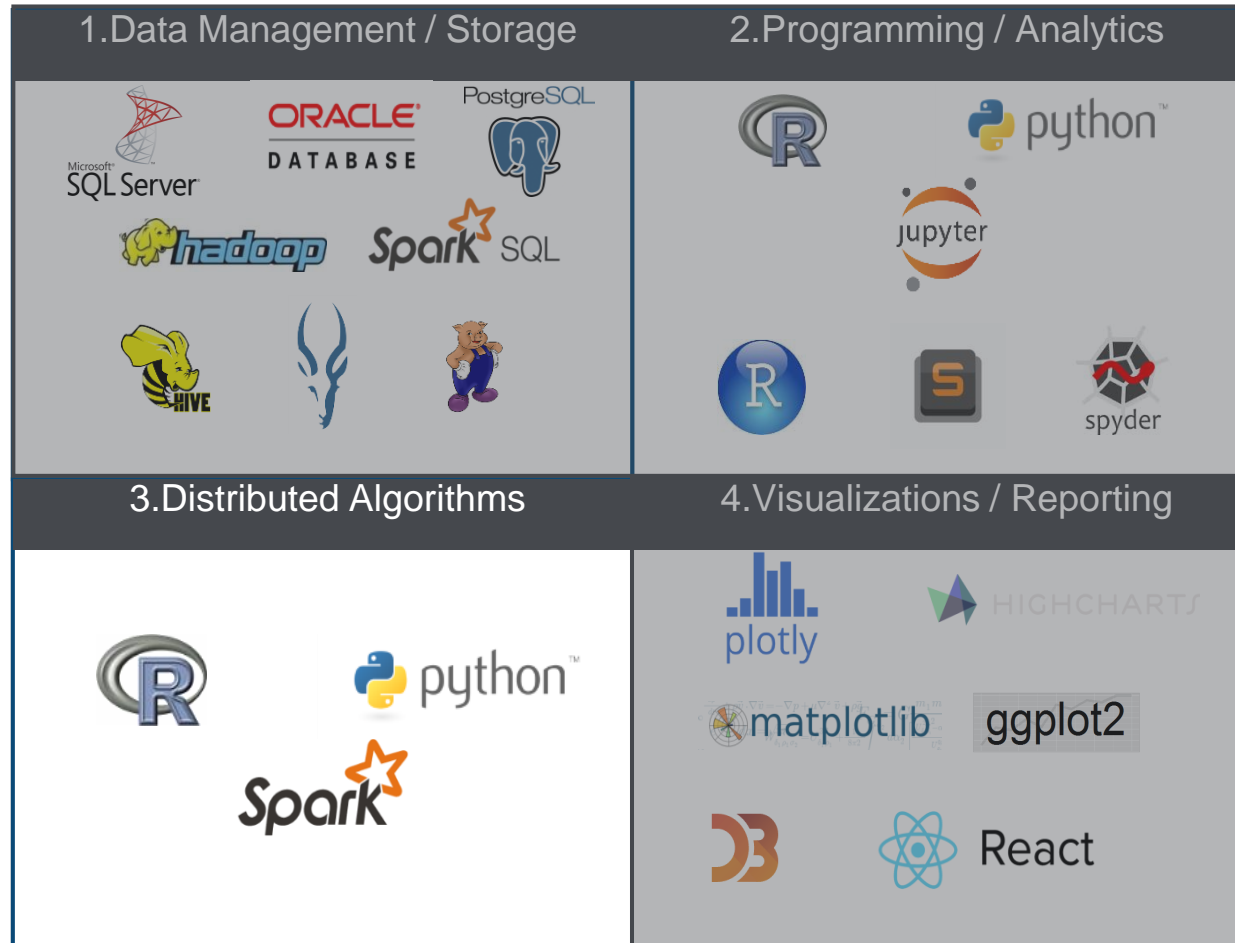
Début historique: 01/01/2012
 Fin historique: 07/10/2016
 Nb observations: 248714
 Début période d'intérêt: none
 Fin période d'intérêt: none

Min RESA: [REDACTED]
 MAX RESA: [REDACTED]
 MEAN RESA: [REDACTED]
 SD RESA: [REDACTED]
 Min REV: [REDACTED]
 MAX REV: [REDACTED]
 MEAN REV: [REDACTED]
 SD REV: [REDACTED]



- Weekly/monthly reporting on triangles in mortgage (credit default)

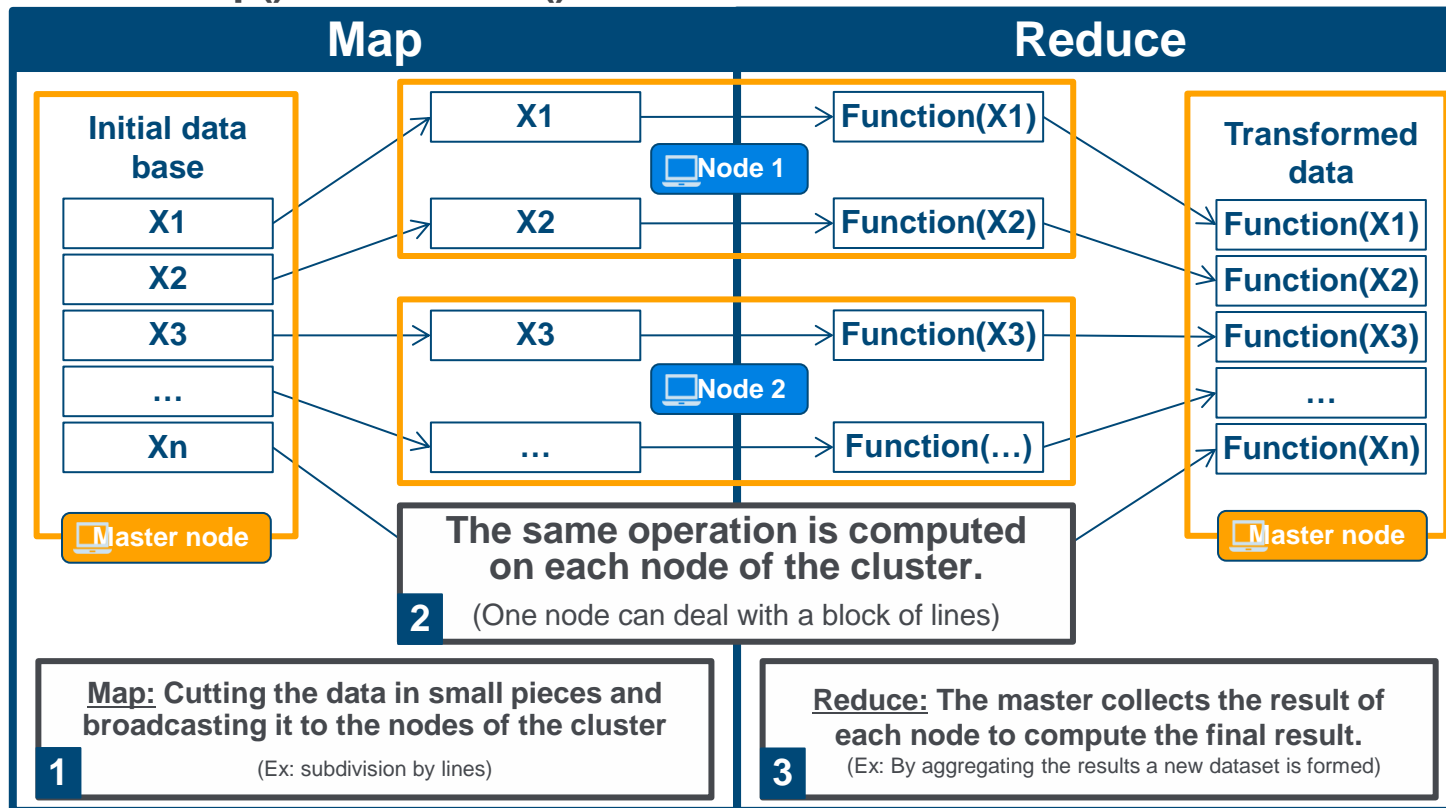
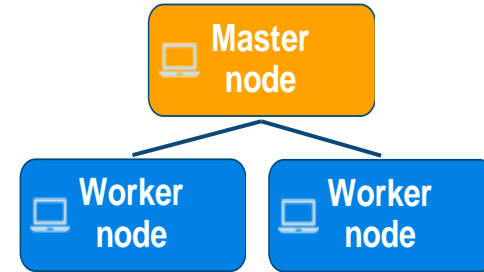
3. Distributed Algorithms



3. Distributed Algorithms

Analytics - Map-Reduce: How does it work ?

- **Concept** created by Google in order to deal with Big Data using a cluster architecture.
- **2 functions** Map() and Reduce()



3. Distributed Algorithms

Map Reduce in practice

How to compute quadratic error of prediction on a billion of lines?

We define for all $Y, \hat{Y} \in \mathcal{Y}^N$,

$$WSSSE(Y, \hat{Y}) = \sum_{i=1}^N error_2(Y_i, \hat{Y}_i)$$

reducer
mapper

Example with PySpark



```
# Evaluate clustering by computing Within Set Sum of Squared Errors
def error(point):
    center = clusters.centers[clusters.predict(point)]
    return sqrt(np.sum([x**2 for x in (point - center)]))
```

mapper

```
WSSSE = parsedData.map(lambda point: error(point)).reduce(lambda x, y: x + y)
print("Within Set Sum of Squared Error = " + str(WSSSE))
```

reducer

4. Visualizations / Reporting



4. Visualizations / Reporting

Visualization for exploration

Dynamic and modern visualization techniques in order to build business oriented visualizations

Matplotlib and ggplot:

Python and R graphic library, allowing graphical data export

Static

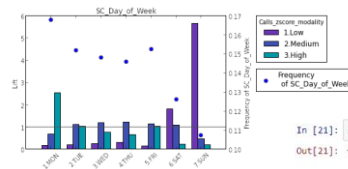


ggplot2

Dynamic

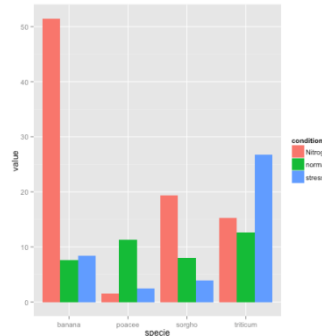
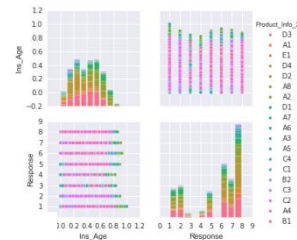
C3.js/D3.js:

Web-technology based on javascript allowing dynamic data-visualization, useful from the exploring to the reporting phase.



Seaborn (python)

```
In [21]: sns.pairplot(df[['Ins_Age', 'Response', 'Product_Info_2']], hue="")
Out[21]: <seaborn.axisgrid.PairGrid at 0x84b25c0>
```

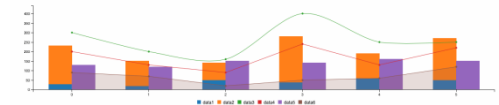
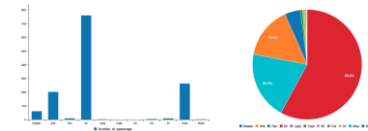


ggplot2

Usage:

- Matplotlib/ggplot works perfectly with Python/R and contains defaults plot styles with built-in code.

C3.js



```
var chart = c3.generate({
  data: {
    columns: [
      ['data1', 30, 20, 50, 40, 60, 80],
      ['data2', 200, 100, 90, 240, 150, 220],
      ['data3', 300, 200, 100, 400, 250, 200],
      ['data4', 200, 100, 90, 240, 150, 220],
      ['data5', 150, 100, 100, 140, 100, 150],
      ['data6', 90, 70, 20, 50, 60, 100],
    ],
  },
  type: 'bar',
  types: {
    data2: 'spline',
    data3: 'line',
    data6: 'area',
  },
  groups: [
    ['data1', 'data2']
  ]
});
```

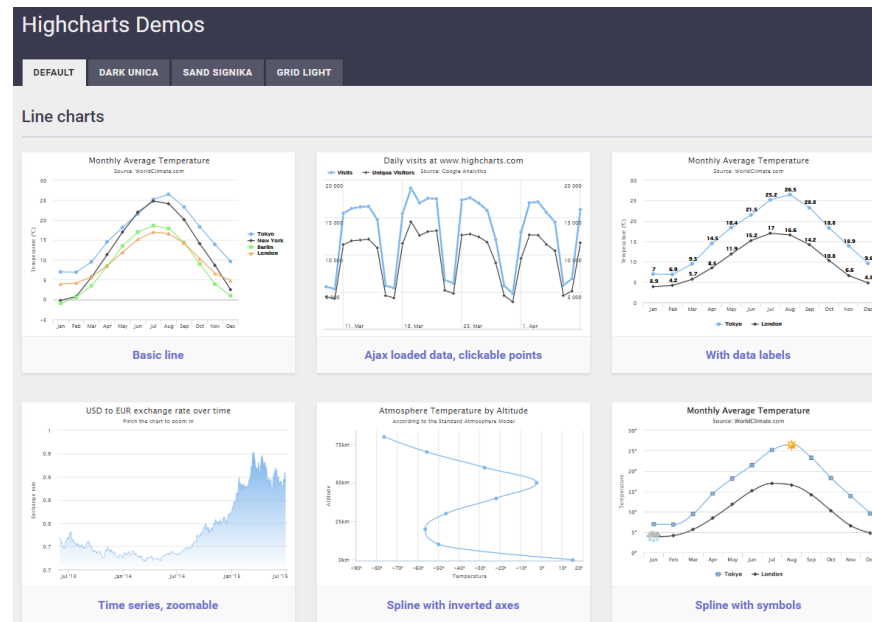
Usage:

- Use of C3.js and D3.js is recommended when dynamic interaction is required or when a web-integration is required.

4. Visualizations / Reporting

Build your own interactive dashboard (1/2)

- Highcharts: SaaS to generate interactive graphics then integrate them in a web page
 - <http://www.highcharts.com/>



- Plotly: library available for R/Python to produce interactive graphics
 - <https://plot.ly/python/offline/>

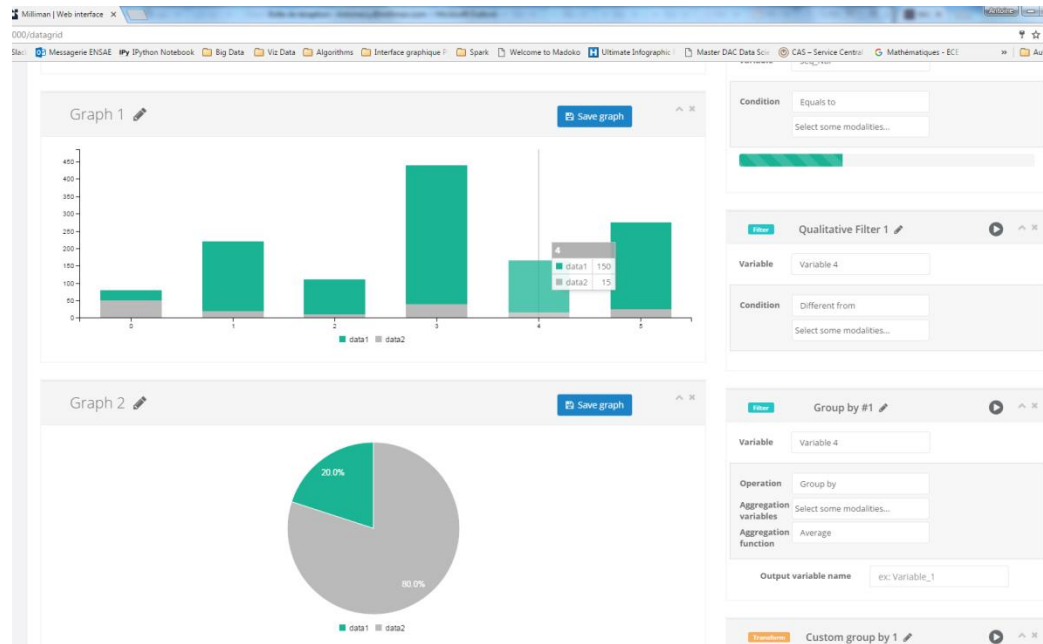
4. Visualizations / Reporting

Build your own interactive dashboard (2/2)

C3.js D3-based reusable chart library



React



Integrating different technologies in one solution

An example of workflow used by Milliman



Data-Lake / Storage



Clustered Computing



- Spark/Hadoop Users across all industries and Big-Data software vendors:

(source : <https://cwiki.apache.org/confluence/display/SPARK/Powered+By+Spark>)



(...)



Data
Innovation Lab



Milliman

Thank you

Eric. O Lebigot – eric.lebigot@axa.com

Rémi Bellina – remi.bellina@milliman.com

Yves-Richard Hong Tuan Ha – yves.hong@milliman.com

Antoine Ly – antoine.ly@milliman.com