Programmation R sous Spark avec SparkR

Installation du framework Spark sous Windows

La librairie MLlib de Spark pour le Machine Learning

Programmation R avec SparkR

Le framework Spark – Objectif du tutoriel

Apache Spark est un framework open source de calcul distribué dédié au Big Data. Sa particularité est qu'il est capable de travailler en mémoire vive. Il est très performant pour les opérations nécessitant plusieurs itérations sur les mêmes données, exactement ce dont ont besoin les algorithmes de machine learning.

Spark peut fonctionner sans Hadoop, mais il a besoin d'un gestionnaire de clusters (qu'il a en interne) et d'un système de fichiers distribués (qu'il n'a pas), ce que peut lui fournir Hadoop avec respectivement Hadoop Yarn et HDFS (Hadoop Distributed File System). De fait, les faire fonctionner ensemble est très avantageux (Hadoop, stockage ; Spark, calculs).

Au-delà des API (modules de classes et fonctions) standards, Spark intègre des librairies additionnelles : Streaming, traitement des données en flux ; SQL, accès aux données Spark avec des requêtes SQL ; GraphX, traitement des graphes ; MLlib, types de données et algorithmes pour le machine learning.

SparkR est un package qui permet de manipuler les types de données et méthodes de **MLlib** (*pas toutes, le portage est en cours*) en programmation R, et de bénéficier directement des avantages de Spark (gestion de la volumétrie, calcul distribué). Ce tutoriel a pour objectif de s'initier à l'utilisation de SparkR en traitant un exemple typique d'analyse prédictive.

Plan

1. Installation de Spark sous Windows
2. Installation de l'environnement de développement – R et RStudio
3. Programmation R avec SparkR
4. Références

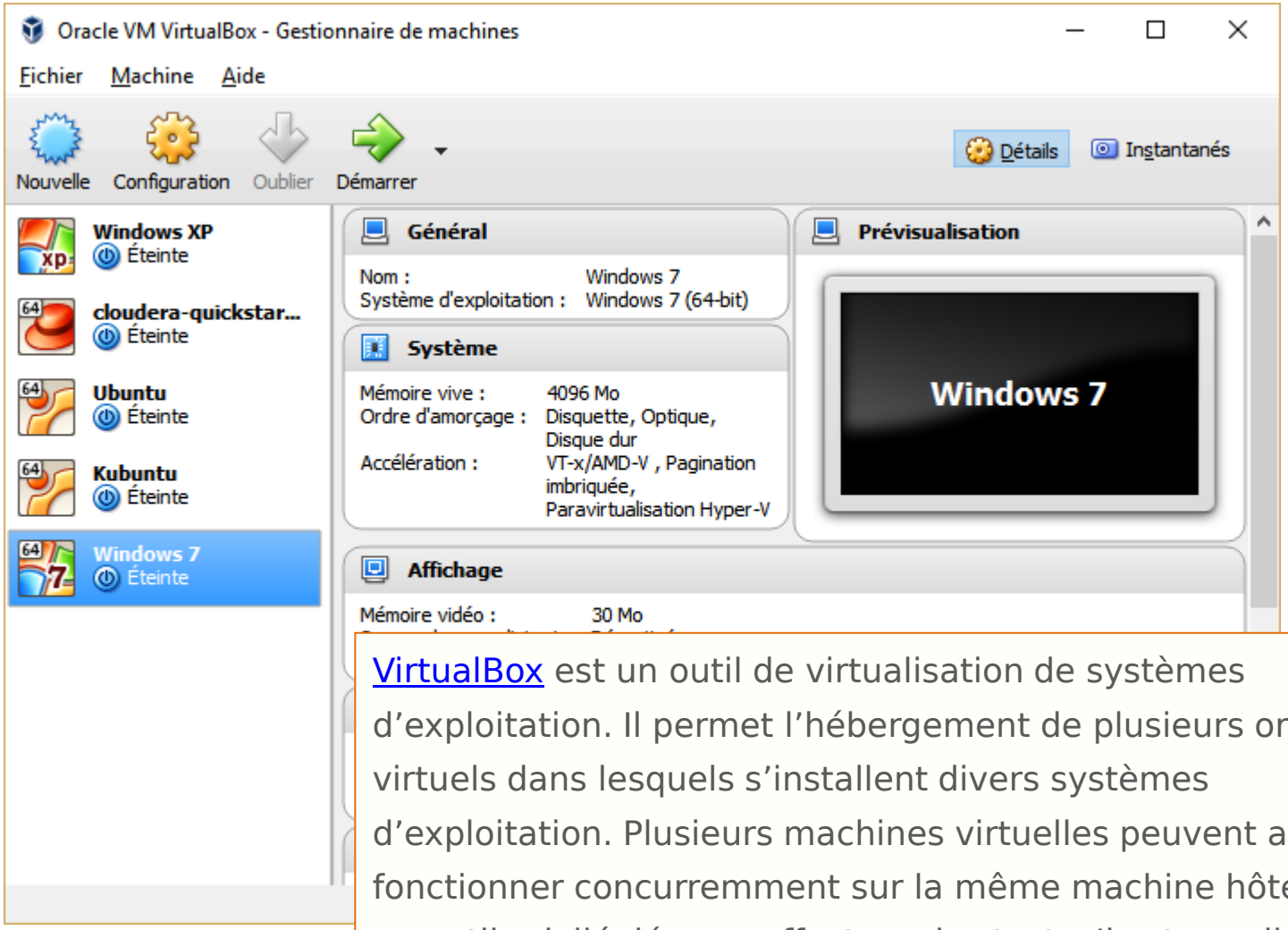
Installation et configuration du framework Spark

Plusieurs pistes sont possibles. Certains éditeurs proposent des systèmes complets (système d'exploitation + Spark) déjà configurés que l'on peut installer directement sur une machine ou une machine virtuelle (ex. Cloudera).

Mais nous pouvons également installer le framework sur un système d'exploitation préexistant. C'est le choix que nous avons fait dans ce tutoriel. Nous nous appuyons sur Windows 7 64 bits (Edition familiale)

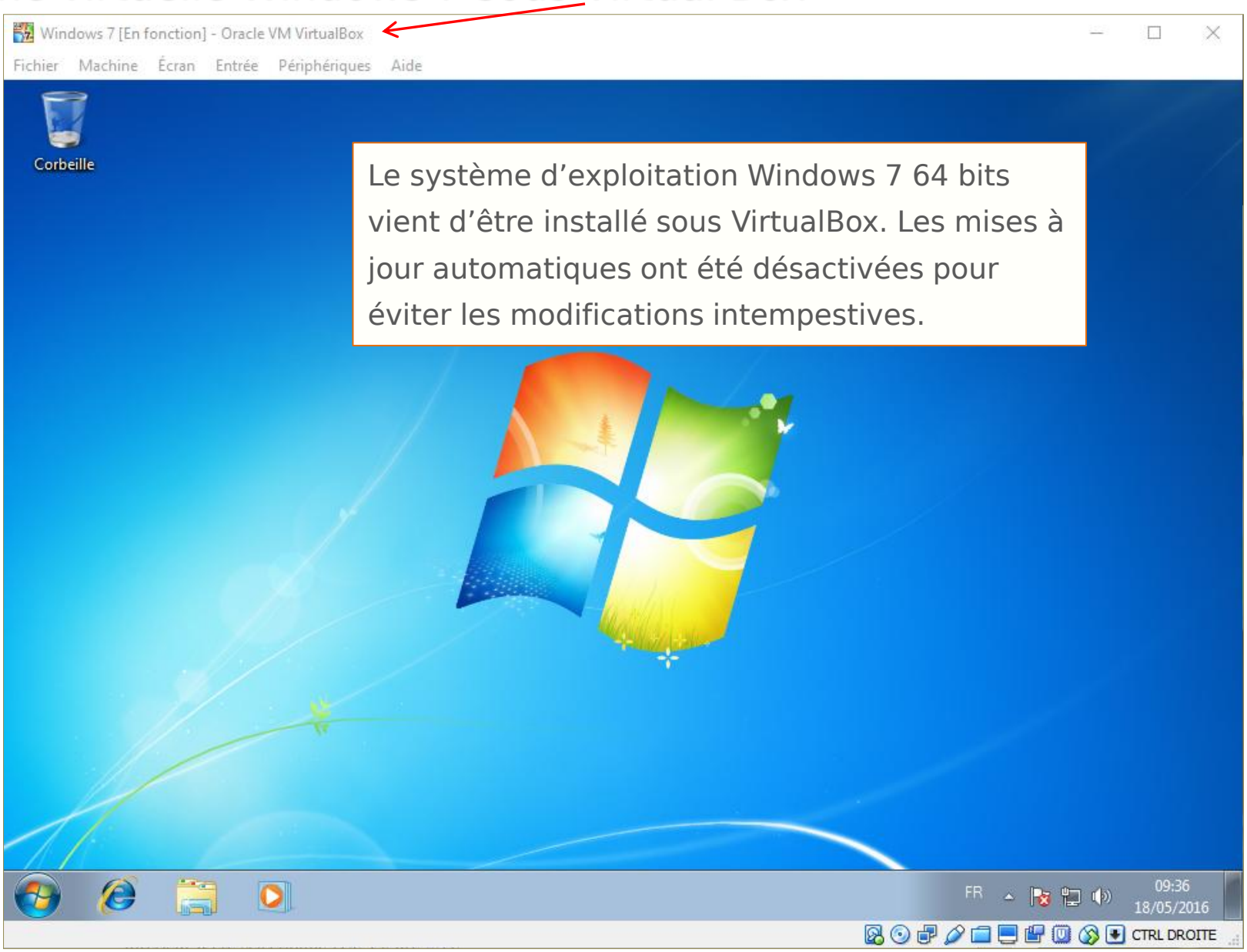
Pour éviter les interférences, nous partons d'une machine virtuelle vierge hébergée par Virtual Box, un outil de virtualisation libre.

VirtualBox



VirtualBox est un outil de virtualisation de systèmes d'exploitation. Il permet l'hébergement de plusieurs ordinateurs virtuels dans lesquels s'installent divers systèmes d'exploitation. Plusieurs machines virtuelles peuvent ainsi fonctionner concurremment sur la même machine hôte. C'est un outil privilégié pour effectuer des tests. Il est possible d'archiver différents états de la même machine.

Machine virtuelle Windows 7 sous Virtual Box



Installation de Java JDK version 8 (64 bits)

Java SE Development Kit 8 - Downloads - Windows Internet Explorer

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Fichier Edition Affichage Favoris Outils ?

Favoris | Sites suggérés | Galerie de composants ...

Java SE Development Kit 8 - Downloads

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.72 MB	jdk-8u91-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.69 MB	jdk-8u91-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.74 MB	jdk-8u91-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u91-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u91-linux-x64.rpm
Linux x64	172.97 MB	jdk-8u91-linux-x64.tar.gz
Mac OS X	227.29 MB	jdk-8u91-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.59 MB	jdk-8u91-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.95 MB	jdk-8u91-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.29 MB	jdk-8u91-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u91-solaris-x64.tar.gz
Windows x86	182.11 MB	jdk-8u91-windows-i586.exe
Windows x64	187.41 MB	jdk-8u91-windows-x64.exe

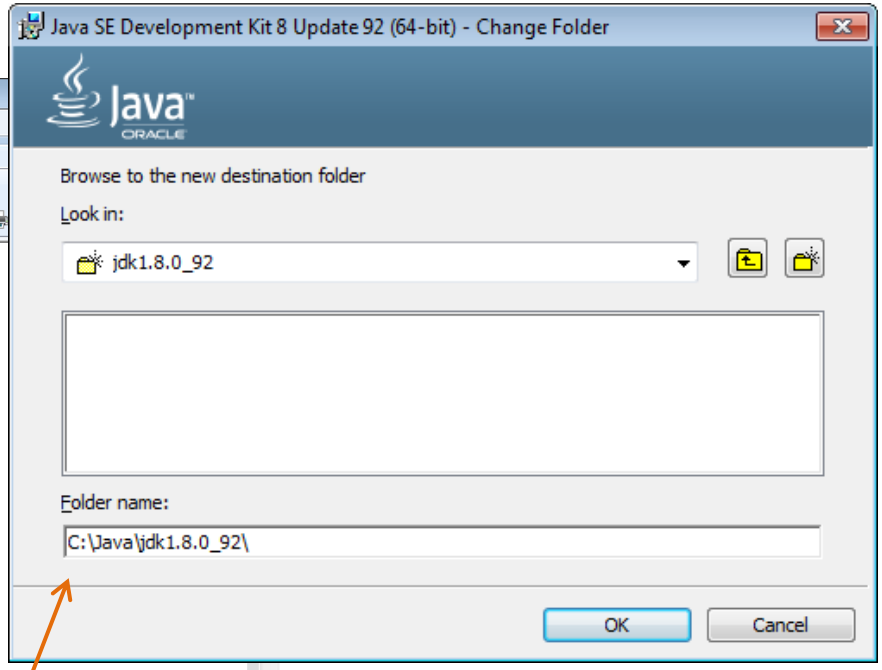
Java SE Development Kit 8u92

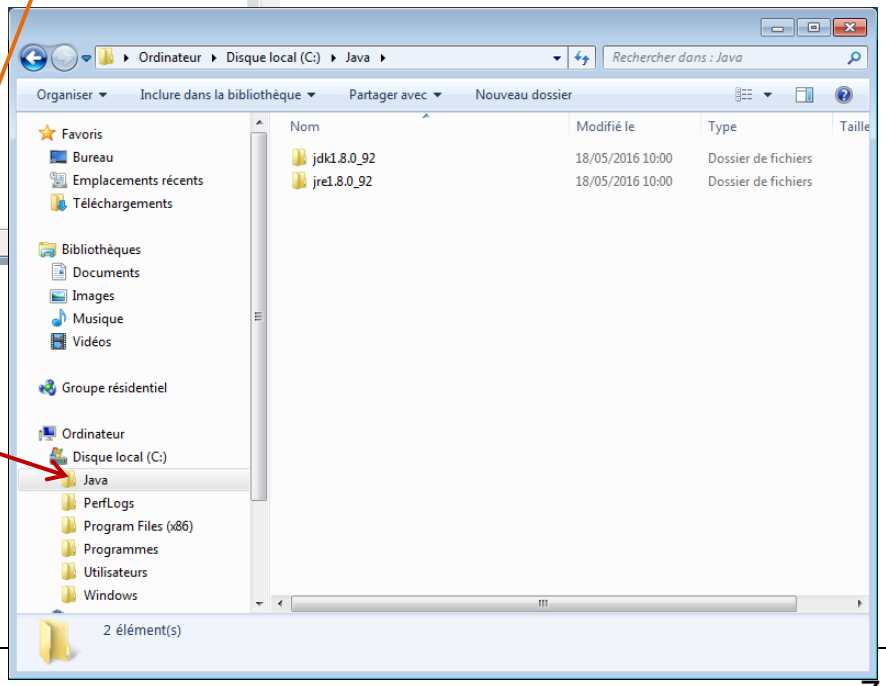
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	160.26 MB	jdk-8u92-linux-i586.rpm
Linux x86	174.94 MB	jdk-8u92-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u92-linux-x64.rpm
Linux x64	172.99 MB	jdk-8u92-linux-x64.tar.gz
Mac OS X	227.32 MB	jdk-8u92-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.47 MB	jdk-8u92-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.93 MB	jdk-8u92-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.35 MB	jdk-8u92-solaris-x64.tar.Z
Solaris x64	96.76 MB	jdk-8u92-solaris-x64.tar.gz
Windows x86	188.43 MB	jdk-8u92-windows-i586.exe
Windows x64	193.66 MB	jdk-8u92-windows-x64.exe

Terminé, mais il existe des erreurs sur la page. Internet | Mode protégé : activé





Java a été installé à la racine, dans le dossier « Java ». 2 sous-répertoires ont été créés pour le JDK et le JRE.

L'utilitaire WinUtils

Hadoop 2.6.0 Windows 64

https://www.barik.net/archive/2015/01/19/172716/

Titus Barik in General

Hadoop 2.6.0 Windows 64-bit Binaries

The official release of Apache Hadoop 2.6.0 does not include the required binaries (e.g., `winutils.exe`) necessary to run hadoop. In order to use Hadoop on Windows, it must be compiled from source. This takes a bit of effort, so I've provided a pre-compiled, unofficial distribution below:

- hadoop-2.6.0.tar.gz
  SHA1: `205b235d77213b958f126647241a5092845a0ff8`
- hadoop-dist-2.6.0-javadoc.jar
  SHA1: `cbccbc5d7eeb03261e533cbe0b1b367fce83b181`
- hadoop-2.6.0-src.tar.gz
  SHA1: `4a1dfa9bd34d5efb7f2a0cd4dcf03db3eab46a5d`

Plusieurs utilitaires, dont ''winutils'', sont nécessaires pour pouvoir faire fonctionner Hadoop sous Windows.

La distribution Spark que nous allons utiliser est adossée à Hadoop.

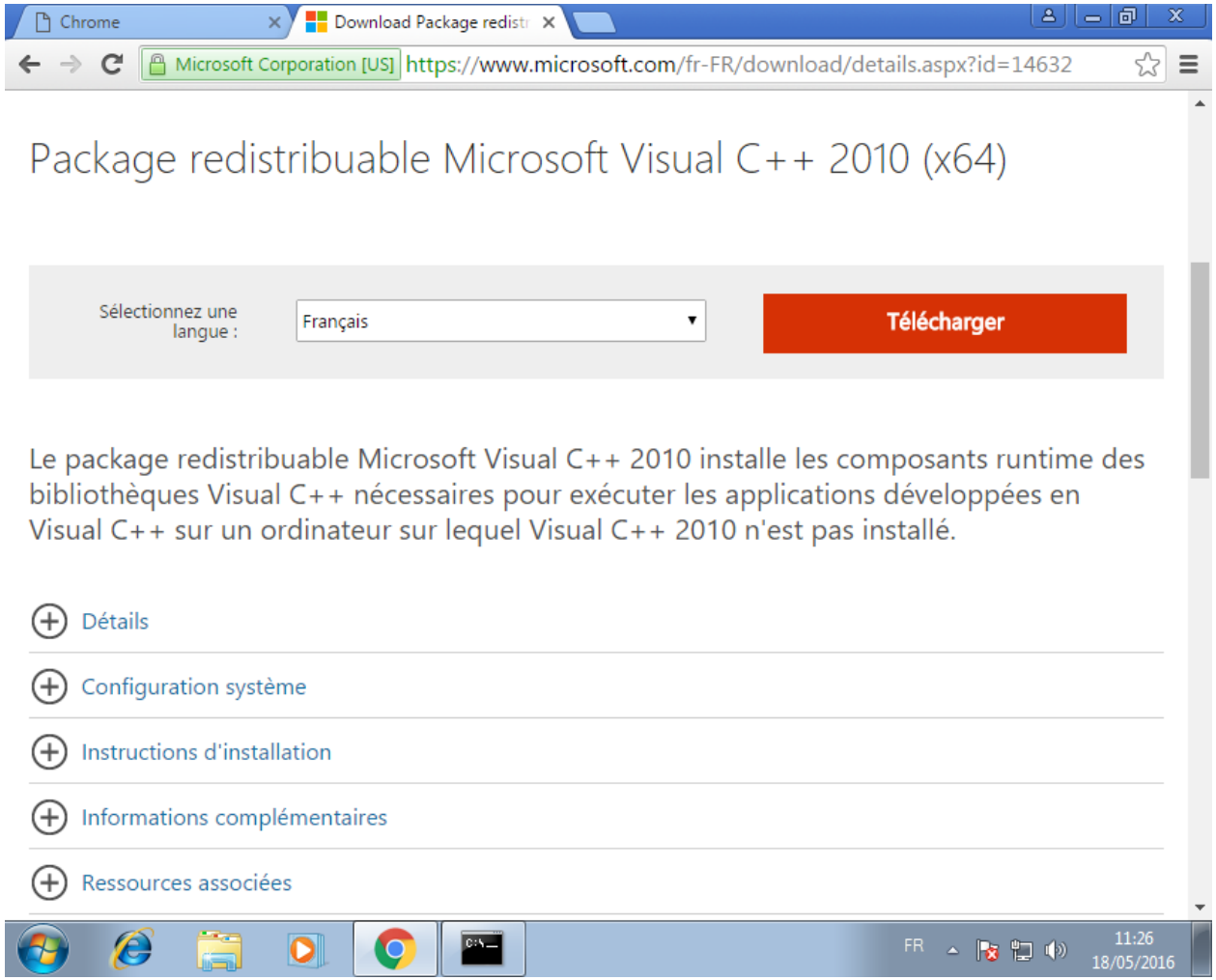
Contenu du fichier archive.

Nous désarchivons le fichier dans le dossier « c:\winutils »

Ci-contre le contenu du sous-répertoire « c:\wintils\bin »

Package redistribuable MS Visual C++ 2010

Ce package doit être installé. Il est nécessaire au bon fonctionnement de « winutils ».



Choix de la version de Spark et installation

Downloads | Apache Spar

spark.apache.org/downloads.html

Spark — Lightning-fast cluster computing

Download | Libraries | Documentation | Examples | Community | FAQ | Apache Software Foundation

Latest News

- Spark Summit (June 6, 2016, San Francisco) agenda posted (Apr 17, 2016)
- Spark 1.6.1 released (Mar 09, 2016)
- Submission is open for Spark Summit San Francisco (Feb 11, 2016)
- Spark Summit East (Feb 16, 2016, New York) agenda posted (Jan 14, 2016)

Download Apache Spark™

Our latest version is Spark 1.6.1, released on March 9, 2016 (release notes) (git ta

1. Choose a Spark release: 1.6.1 (Mar 09 2016)
2. Choose a package type: Pre-built for Hadoop 2.6 and later
3. Choose a download type: Direct Download
4. Download Spark: spark-1.6.1-bin-hadoop2.6.tgz
5. Verify this release using the 1.6.1 signatures and checksums.

Note: Scala 2.11 users should download the Spark source package and build with S

Chargement de la version pour Hadoop 2.6.

Nous désarchivons simplement le fichier dans un répertoire dédié. Ici : « c:\spark-1.6.1-bin-hadoop2.6 »

Ordinateur ▸ Disque local (C:) ▸ — Rechercher dans : Disque local (C:)

Organiser | Partager avec | Nouveau dossier

Favoris: Bureau, Emplacements récents, Téléchargements

Bibliothèques: Documents, Images, Musique, Vidéos

Groupe résidentiel

Ordinateur: Disque local (C:), Lecteur de CD (D:) VirtualBox Guest Additio, Echange (\\vboxsrv) (E:)

Réseau

Nom	Modifié le	Type
Java	18/05/2016 09:59	Dossier de fichiers
PerfLogs	14/07/2009 05:20	Dossier de fichiers
Program Files (x86)	18/05/2016 10:15	Dossier de fichiers
Programmes	18/05/2016 10:34	Dossier de fichiers
spark-1.6.1-bin-hadoop2.6	18/05/2016 11:03	Dossier de fichiers
Utilisateurs	18/05/2016 06:42	Dossier de fichiers
Windows	18/05/2016 09:54	Dossier de fichiers
winutils	18/05/2016 10:39	Dossier de fichiers

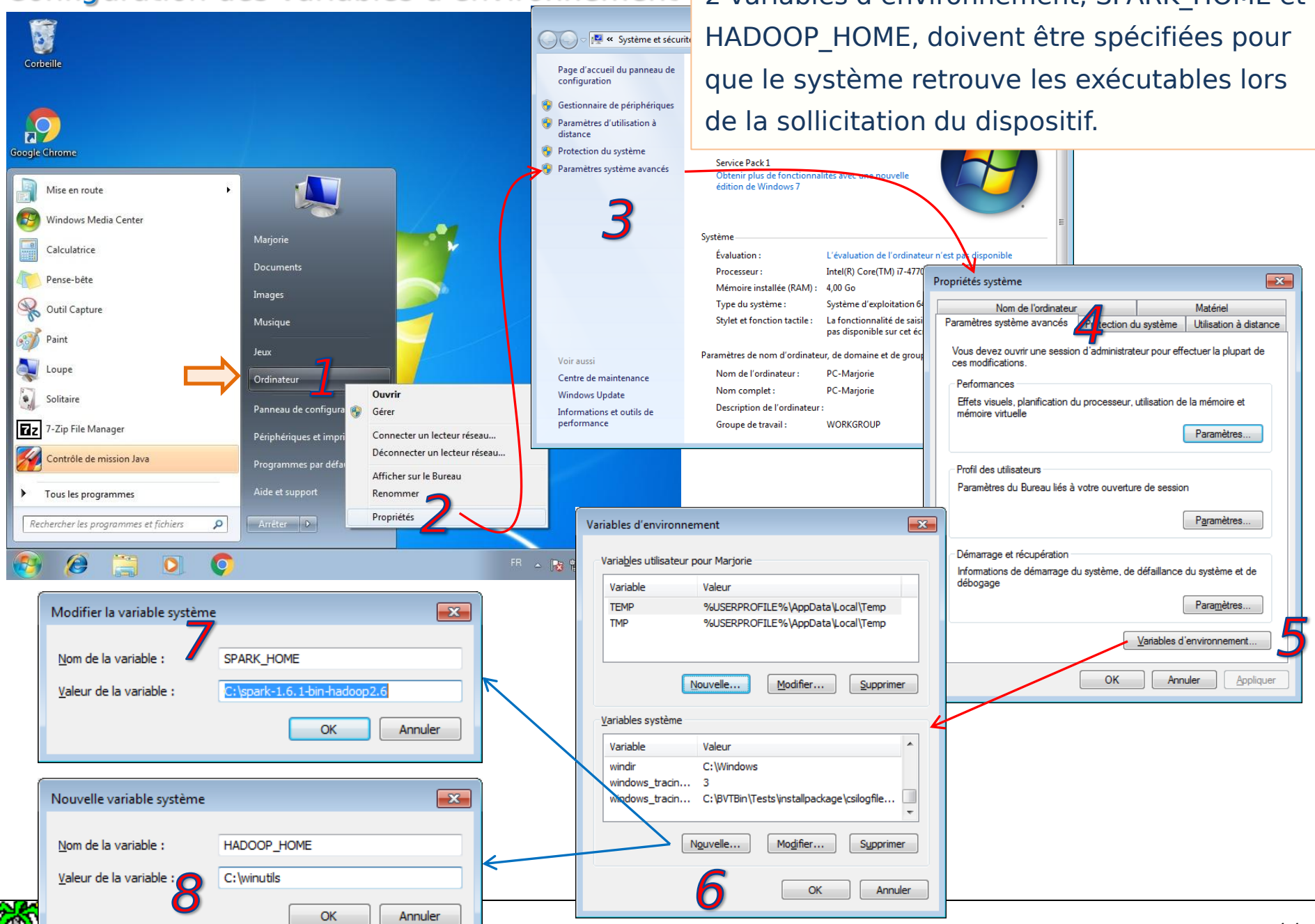
8 élément(s)

Voici l'organisation des dossiers de notre machine à ce stade.

FR 11:04 18/05/2016

Configuration des variables d'environnement

2 variables d'environnement, SPARK_HOME et HADOOP_HOME, doivent être spécifiées pour que le système retrouve les exécutables lors de la sollicitation du dispositif.

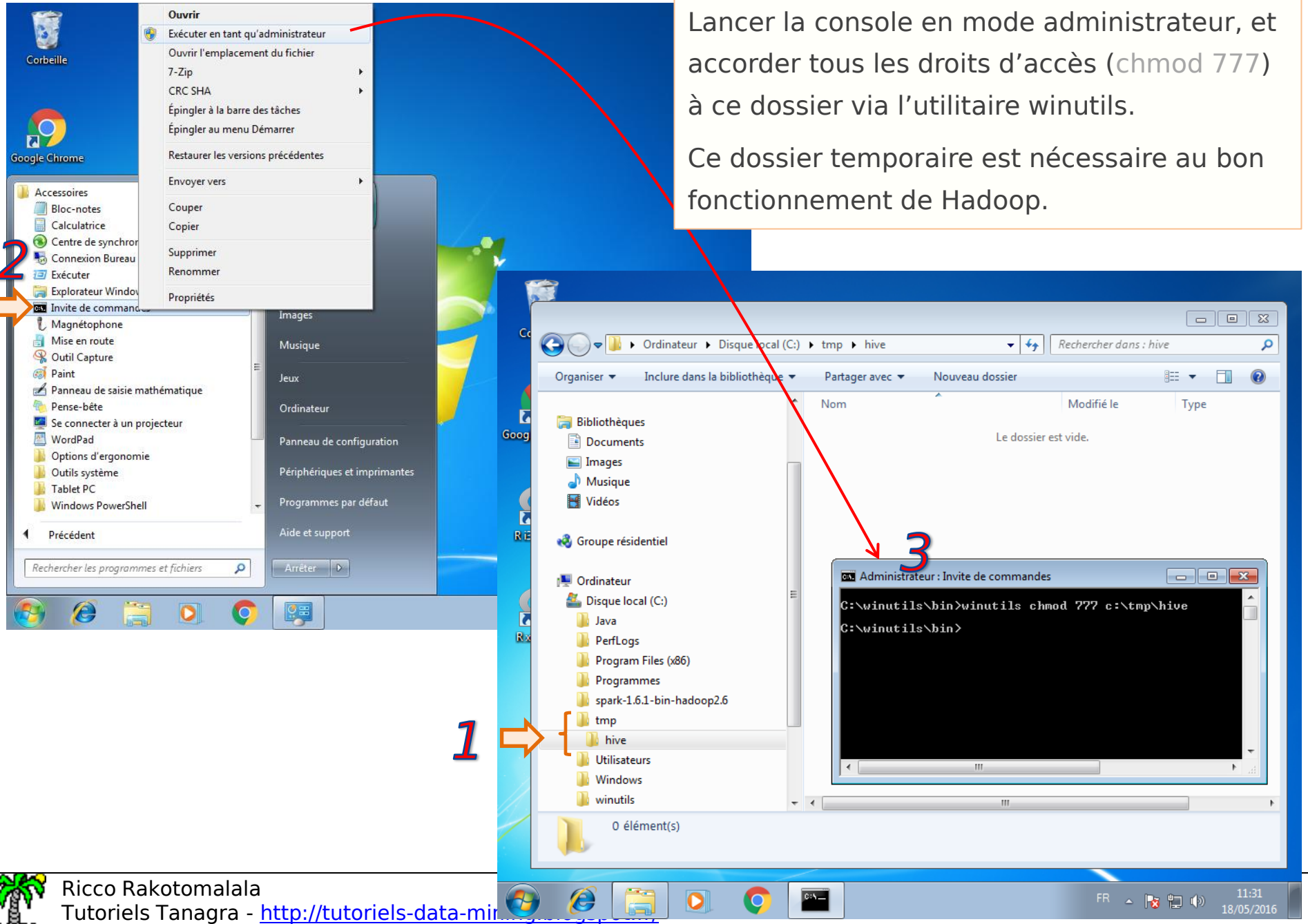


Configuration de Hadoop via winutils

Créer le dossier « c:\tmp\hive » au préalable.

Lancer la console en mode administrateur, et accorder tous les droits d'accès (chmod 777) à ce dossier via l'utilitaire winutils.

Ce dossier temporaire est nécessaire au bon fonctionnement de Hadoop.



Lancer Spark

Administrateur : Invite de commandes

```
C:\spark-1.6.1-bin-hadoop2.6\bin>dir
 Le volume dans le lecteur C n'a pas de nom.
 Le numéro de série du volume est 7882-C8DD

 Répertoire de C:\spark-1.6.1-bin-hadoop2.6\bin

18/05/2016  11:36    <REP>          .
18/05/2016  11:36    <REP>          ..
27/02/2016  07:02             1 099 beeline
27/02/2016  07:02               932 beeline.cmd
19/05/2016  12:03            21 118 derby.log
27/02/2016  07:02             1 910 load-spark-env.cmd
27/02/2016  07:02             2 143 load-spark-env.sh
19/05/2016  12:02    <REP>          metastore_db
27/02/2016  07:02             3 459 pyspark
27/02/2016  07:02             1 000 pyspark.cmd
27/02/2016  07:02             1 486 pyspark2.cmd
27/02/2016  07:02             2 384 run-example
27/02/2016  07:02             1 012 run-example.cmd
27/02/2016  07:02             2 682 run-example2.cmd
27/02/2016  07:02             2 858 spark-class
27/02/2016  07:02             1 010 spark-class.cmd
27/02/2016  07:02             2 365 spark-class2.cmd
27/02/2016  07:02             3 026 spark-shell
27/02/2016  07:02             1 008 spark-shell.cmd
27/02/2016  07:02             1 528 spark-shell2.cmd
27/02/2016  07:02             1 075 spark-sql
27/02/2016  07:02             1 050 spark-submit
27/02/2016  07:02             1 010 spark-submit.cmd
27/02/2016  07:02             1 126 spark-submit2.cmd
27/02/2016  07:02             1 049 sparkR
27/02/2016  07:02               998 sparkR.cmd
27/02/2016  07:02             1 010 sparkR2.cmd
               24 fichier(s)          58 338 octets
                3 Rép(s)  257 175 998 464 octets libres

C:\spark-1.6.1-bin-hadoop2.6\bin>spark-shell
```

1

Administrateur : Invite de commandes - spark-shell

```
-rdbms-3.2.9.jar" is already registered, and you are trying to register an ident
ical plugin located at URL "file:/C:/spark-1.6.1-bin-hadoop2.6/bin/../lib/datanu
cleus-rdbms-3.2.9.jar."
16/05/19 12:01:31 WARN General: Plugin (Bundle) "org.datanucleus" is already reg
istered. Ensure you dont have multiple JAR versions of the same plugin in the cl
asspath. The URL "file:/C:/spark-1.6.1-bin-hadoop2.6/bin/../lib/datanucleus-core
-3.2.10.jar" is already registered, and you are trying to register an identical
plugin located at URL "file:/C:/spark-1.6.1-bin-hadoop2.6/lib/datanucleus-core-3
.2.10.jar."
16/05/19 12:01:31 WARN General: Plugin (Bundle) "org.datanucleus.api.jdo" is alr
eady registered. Ensure you dont have multiple JAR versions of the same plugin i
n the classpath. The URL "file:/C:/spark-1.6.1-bin-hadoop2.6/bin/../lib/datanucl
eus-api-jdo-3.2.6.jar" is already registered, and you are trying to register an
identical plugin located at URL "file:/C:/spark-1.6.1-bin-hadoop2.6/lib/datanucl
eus-api-jdo-3.2.6.jar."
16/05/19 12:01:32 WARN Connection: BoneCP specified but not present in CLASSPATH
 (or one of dependencies)
16/05/19 12:01:33 WARN Connection: BoneCP specified but not present in CLASSPATH
 (or one of dependencies)
16/05/19 12:02:03 WARN ObjectStore: Version information not found in metastore.
hive.metastore.schema.verification is not enabled so recording the schema versio
n 1.2.0
16/05/19 12:02:04 WARN ObjectStore: Failed to get database default, returning No
SuchObjectException
16/05/19
istered.
asspath.
-3.2.10.
plugin l
.2.10.ja
16/05/19
 already
in in th
-rdbms-3
ical plu
cleus-rdbms-3.2.9.jar."
16/05/19 12:02:23 WARN General: Plugin (Bundle) "org.datanucleus.api.jdo" is alr
eady registered. Ensure you dont have multiple JAR versions of the same plugin i
he classpath. The URL "file:/C:/spark-1.6.1-bin-hadoop2.6/bin/../lib/datanucl
-api-jdo-3.2.6.jar" is already registered, and you are trying to register an
ntical plugin located at URL "file:/C:/spark-1.6.1-bin-hadoop2.6/lib/datanucl
-api-jdo-3.2.6.jar."
05/19 12:02:23 WARN Connection: BoneCP specified but not present in CLASSPATH
r one of dependencies)
05/19 12:02:24 WARN Connection: BoneCP specified but not present in CLASSPATH
r one of dependencies)
 context available as sqlContext.

scala>
```

2

Mais pour l'exploiter, il faudrait savoir coder en « scala », ce qui n'est pas mon cas (Remarque : exit permet de sortir de l'interpréteur de commandes).

Via le terminal de commande lancé en mode administrateur. Liste des exécutables.

On peut lancer Spark avec « spark-shell »

Installation et configuration de l'environnement de développement

Installation de R (obligatoire)

Installation de RStudio (c'est mieux)

Installation de R

The Comprehensive R Archive Network - Windows Internet Explorer

https://pbil.univ-lyon1.fr/CRAN/

Fichier Edition Affichage Favoris Outils ?

Favoris Sites suggérés Galerie de composants ...

The Comprehensive R Archive Network

Page Sécurité Outils

R logo

R-3.3.0 for Windows (32/64 bit)

Download R 3.3.0 for Windows (62 megabytes, 32/64 bit)

Installation and other instructions

New features in this version

If you want to double-check that the package you have downloaded exactly matc
distributed by R, you can compare the md5sum of the .exe to the true fingerprint.
of md5sum for windows: both graphical and command line versions are available.

Frequently asked question

- Does R run under my version of Windows?
- How do I update packages in my previous version of R?
- Should I run 32-bit or 64-bit R?

Please see the R FAQ for general information about R and the R Windows FAQ
information.

Other builds

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

Internet | Mode protégé : activé

« R ▸ R-3.3.0 ▸

Rechercher dans : R-3.3.0

Organiser Inclure dans la bibliothèque Partager avec »

Ordinateur
Disque local (C:)
data
Java
PerfLogs
Program Files (x86)
ProgramData
Programmes
7-Zip
Common Files
DVD Maker
Internet Explorer
Microsoft Games
MSBuild
Oracle
R
R-3.3.0
bin
doc
etc
include
library
modules
share
src
Tcl
tests

Nom

bin
doc
etc
include
library
modules
share
src
Tcl
tests
CHANGES
COPYING
MD5
README
README.R-3.3.0
unins000.dat
unins000.exe

17 élément(s)

Installation de la version la plus récente de R pour Windows (3.3.0 au 19/05/2016).

Première utilisation de SparkR

Administrateur : Invite de commandes

```
C:\spark-1.6.1-bin-hadoop2.6\bin>dir
 Le volume dans le lecteur C n'a pas de nom.
 Le numéro de série du volume est 7882-C8DD

 Répertoire de C:\spark-1.6.1-bin-hadoop2.6\bin

18/05/2016  11:36    <REP>          .
18/05/2016  11:36    <REP>          ..
27/02/2016  07:02             1 099 beeline
27/02/2016  07:02               932 beeline.cmd
19/05/2016  12:20            21 118 derby.log
27/02/2016  07:02             1 910 load-spark-env.cmd
27/02/2016  07:02             2 143 load-spark-env.sh
19/05/2016  12:19    <REP>          metastore_db
27/02/2016  07:02             3 459 pyspark
27/02/2016  07:02             1 000 pyspark.cmd
27/02/2016  07:02             1 486 pyspark2.cmd
27/02/2016  07:02             2 384 run-example
27/02/2016  07:02             1 012 run-example.cmd
27/02/2016  07:02             2 682 run-example2.cmd
27/02/2016  07:02             2 858 spark-class
27/02/2016  07:02             1 010 spark-class.cmd
27/02/2016  07:02             2 365 spark-class2.cmd
27/02/2016  07:02             3 026 spark-shell
27/02/2016  07:02             1 008 spark-shell.cmd
27/02/2016  07:02             1 528 spark-shell2.cmd
27/02/2016  07:02             1 075 spark-sql
27/02/2016  07:02             1 050 spark-submit
27/02/2016  07:02             1 010 spark-submit.cmd
27/02/2016  07:02             1 126 spark-submit2.cmd
27/02/2016  07:02             1 049 sparkR
27/02/2016  07:02               998 sparkR.cmd
27/02/2016  07:02             1 010 sparkR2.cmd
               24 fichier(s)           58 338 octets
                3 Rép(s)  257 165 078 528 octets libres

C:\spark-1.6.1-bin-hadoop2.6\bin>sparkR
```

On peut déjà fonctionner directement en lançant ''sparkR'' dans le terminal de commande. Attention ! Il faut que le chemin de R soit ajouté à la variable d'environnement PATH de Windows.

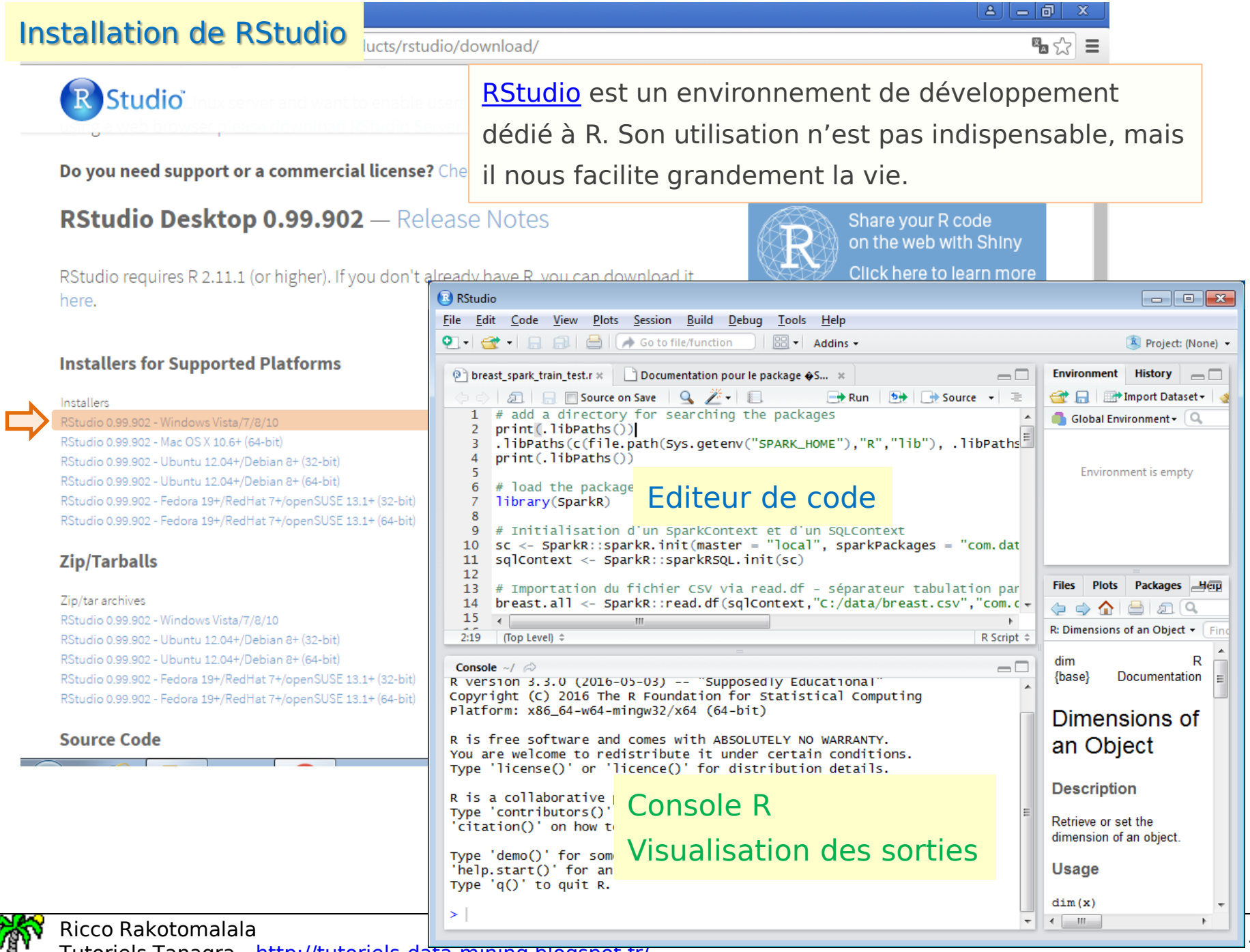
Rterm (64-bit)

```
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more in
fo.
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
16/05/19 12:42:07 INFO SparkContext: Running Spark version 1.6.1
16/05/19 12:42:07 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
16/05/19 12:42:07 INFO SecurityManager: Changing view acls to: Marjorie
16/05/19 12:42:07 INFO SecurityManager: Changing modify acls to: Marjorie
16/05/19 12:42:07 INFO SecurityManager: SecurityManager: authentication disabled
; ui acls disabled; users with view permissions: Set(Marjorie); users with modif
y permissions: Set(Marjorie)
16/05/19 12:42:08 INFO Utils: Successfully started service 'sparkDriver' on port
 49328.
16/05/19 12:42:08 INFO Slf4jLogger: Slf4jLogger started
16/05/19 12:42:08 INFO Remoting: Starting remoting
16/05/19 12:42:08 INFO Utils: Successfully started service 'sparkDriverActorSyst
em' on port 49341.
16/05/19 12:42:08 INFO Remoting: Remoting started; listening on addresses :[akka
.tcp://sparkDriverActorSystem@10.0.2.15:49341]
16/05/19 12:42:08 INFO SparkEnv: Registering MapOutputTracker
16/05/19 12:42:08 INFO SparkEnv: Registering BlockManagerMaster
16/05/1
jorie\A
16/05/1
16/05/1
16/05/1
0.
16/05/1
16/05/1
16/05/1
work.netty.NettyBlockTransferService' on port 49348.
16/05/19 12:42:09 INFO NettyBlockTransferService: Server created on 49348
16/05/19 12:42:09 INFO BlockManagerMaster: Trying to register BlockManager
16/05/19 12:42:09 INFO BlockManagerMasterEndpoint: Registering block manager loc
alhost:49348 with 517.4 MB RAM, BlockManagerId(driver, localhost, 49348)
16/05/19 12:42:09 INFO BlockManagerMaster: Registered BlockManager

 Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version  1.6.1
      /_/


 Spark context is available as sc, SQL context is available as sqlContext
> ls()
[1] "sc"         "sqlContext"
>
```

On peut insérer directement ici les commandes R. Mais ce n'est pas très convivial, j'en conviens.

Installation de RStudio

RStudio est un environnement de développement dédié à R. Son utilisation n'est pas indispensable, mais il nous facilite grandement la vie.



Programmation R avec SparkR

Exploiter via R la librairie MLlib de machine learning pour Spark dans un schéma d'analyse prédictive très classique

Pour éviter toutes ambigüités, toutes les fonctions SparkR seront préfixées par le nom de la librairie dans le code R (SparkR::)

1 MLlib est une librairie de machine learning pour Spark. Il intègre les algorithmes usuels de fouille de données (classement, régression, clustering – Voir MLlib Guide).

Il permet l'exploitation des capacités de Spark en matière de manipulation et de traitement des gros volumes sans que l'on ait quelque chose de particulier à faire, sauf à connaître les structures de données et les commandes dédiées.

2 SparkR est un package spécifique qui fournit des outils permettant d'exploiter les fonctionnalités de Spark et MLlib à partir de R.

Nous programmons toujours en langage R, mais de nouvelles structures de données et jeux d'instructions sont à notre disposition pour exploiter pleinement la puissance de Spark pour le traitement des données massives.

Données : Breast Cancer Wisconsin (Serveur UCI)

Il s'agit d'un fichier texte avec séparateur tabulation. La dernière colonne ''target'' représente la variable cible.

Le fichier « breast.csv » a été placé dans le dossier « c:\data »

C:\data\breast.csv - Notepad++

	clump	ucellsize	ucellshape	mgadhesion	sepics	bnu
2	4 2	2 1	2 1	2 1	1	begnin
3	1 1	1 1	2 1	2 1	1	begnin
4	2 1	1 1	2 1	2 1	1	begnin
5	10 6	6 2	4 10	9 7	1	malignant
6	4 1	1 1	2 1	2 1	1	begnin
7	1 1	1 1	2 1	1 1	1	begnin
8	1 1	1 1	2 1	2 1	1	begnin
9	5 1	1 1	2 1	2 1	1	begnin
10	3 1	1 1	2 1	2 1	1	begnin
11	1 1	1 1	2 4	2 1	1	begnin
12	5 3	3 2	3 1	3 1	1	begnin
13	4 2	2 1	2 1	2 1	1	begnin
14	1 1	1 1	2 1	2 1	1	begnin
15	2 1	1 1	2 1	2 1	1	begnin

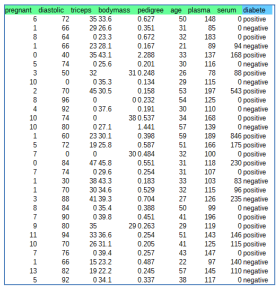
Etapes de la modélisation prédictive et de son évaluation

Y : variable cible (target)

X1, X2, … : variables explicatives (clump, …, mitoses)

f(.) une fonction qui essaie d'établir la relation Y = f(X1, X2, …)

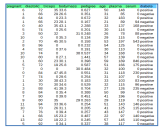
f(.) doit être « aussi précise que possible »…



pregnant	diastolic	triceps	bodymass	pedigree	age	plasma	serum	diabete
6	72	35	33.6	0.627	50	148		0 positive
1	66	29	26.6	0.351	31	85		0 negative
8	64	0	23.3	0.672	32	183		0 positive
1	66	23	28.1	0.167	21	89		94 negative
0	40	35	43.1	2.288	33	137		168 positive
5	74	0	25.6	0.201	30	116		0 negative
3	50	32		31 0.248	26	78		88 positive
10	0	0	35.3	0.134	29	115		0 negative
2	70	45	30.5	0.158	53	197		543 positive
8	96	0		0 0.232	54	125		0 positive
4	92	0	37.6	0.191	30	110		0 negative
10	74	0		38 0.537	34	168		0 positive
10	80	0	27.1	1.441	57	139		0 negative
1	60	23	30.1	0.398	59	189		846 positive
5	72	19	25.8	0.587	51	166		175 positive
7	0	0		30 0.484	32	100		0 positive
0	84	47	45.8	0.551	31	118		230 positive
7	74	0	29.6	0.254	31	107		0 positive
1	30	38	43.3	0.183	33	103		83 negative
1	70	30	34.6	0.529	32	115		96 positive
3	88	41	39.3	0.704	27	126		235 negative
8	84	0	35.4	0.388	50	99		0 negative
7	90	0	39.8	0.451	41	196		0 positive
9	80	35		29 0.263	29	119		0 positive
11	94	33	36.6	0.254	51	143		146 positive
10	70	26	31.1	0.205	41	125		115 positive
7	76	0	39.4	0.257	43	147		0 positive
1	66	15	23.2	0.487	22	97		140 negative
13	82	19	22.2	0.245	57	145		110 negative
5	92	0	34.1	0.337	38	117		0 negative

Ensemble de données (dataset)

Ensemble d'apprentissage

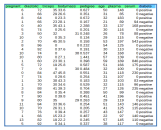


Construction de la fonction f(.) à partir des données d'apprentissage

$$Y = f(X1,X2,\ldots) + \varepsilon$$

Application du modèle (prédiction) sur l'ensemble de test

Ensemble de test



$$(Y, \hat{Y})$$

Y : valeurs observées

Y^ : valeurs prédites par f(.)

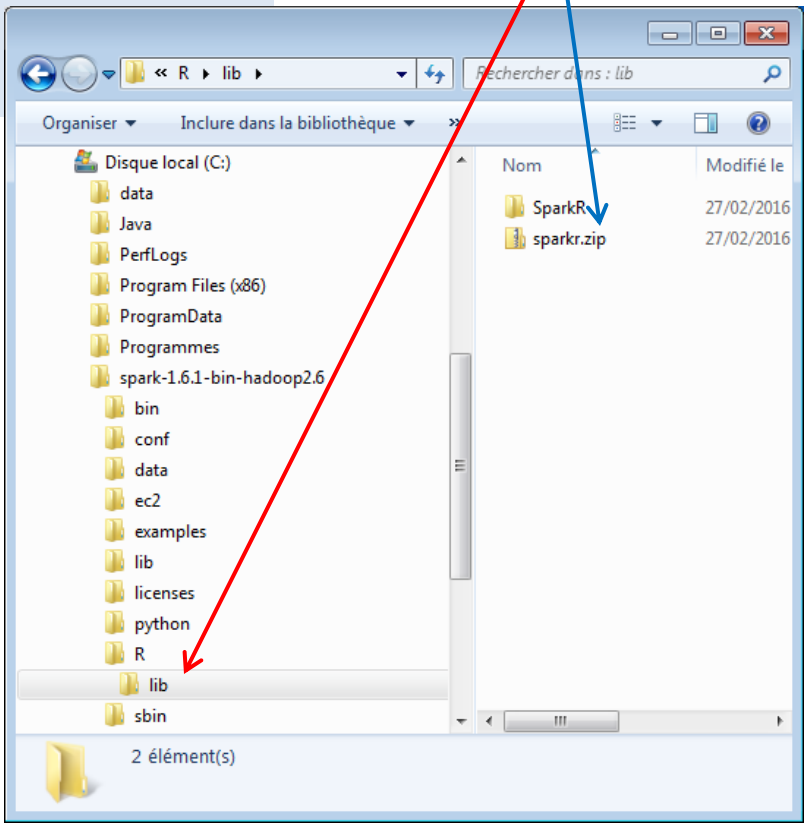
Mesures de **performances** par confrontation entre Y et Y^ : matrice de confusion + mesures

Etape 1 : Modification des chemins d'accès aux packages – Chargement du package

```
# vérifier les chemins d'accès aux packages
print(.libPaths())
# ajouter le chemin du package SparkR
.libPaths(c(file.path(Sys.getenv("SPARK_HOME"),"R","lib"), .libPaths()))
#re-vérifier
print(.libPaths())

# chargement du package sparkR
library(SparkR)
```

```
#1er affichage (print)
[1] "C:/Users/Marjorie/Documents/R/win-library/3.3" "C:/Program Files/R/R-3.3.0/library"

#2nd affichage (print)
[1] "c:/spark-1.6.1-bin-hadoop2.6/R/lib"
"C:/Users/Marjorie/Documents/R/win-library/3.3"
[3] "C:/Program Files/R/R-3.3.0/library"
```

Il faut indiquer à R l'endroit où est situé le package SparkR



Etape 2 : Chargement des données

```
# initialisation d'un SparkContext c.-à-d. le "moteur" Spark est démarré. spark.stop() l'éteindra
# dans le gestionnaire de tâches de Windows, on voit 3 nouveaux processus arriver (java, cmd, cmd)
sc <- SparkR::sparkR.init(master = "local", sparkPackages = "com.databricks:spark-csv_2.10:1.4.0")
# puis d'un SQLContext, point d'entrée pour toutes les fonctionnalités Spark
# http://spark.apache.org/docs/latest/sql-programming-guide.html#starting-point-sqlcontext
sqlContext <- SparkR::sparkRSQL.init(sc)
# importation du fichier CSV via read.df - séparateur tabulation (delimiter), en-tête (header) : noms des variables
# breast.all est de type DataFrame spécifique à SparkR
breast.all <- SparkR::read.df(sqlContext,"C:/data/breast.csv","com.databricks.spark.csv", header="true",delimiter="\t")
# affichage du schéma du DataFrame
SparkR::printSchema(breast.all)
# affichage des 10 premières observations (lignes)
SparkR::showDF(breast.all,10)
```

```
# affichage de la structure (schéma)
# du tableau de données
root
|-- clump: string (nullable = true)
|-- ucellsize: string (nullable = true)
|-- ucellshape: string (nullable = true)
|-- mgadhesion: string (nullable = true)
|-- sepics: string (nullable = true)
|-- bnuclei: string (nullable = true)
|-- bchromatin: string (nullable = true)
|-- normnucl: string (nullable = true)
|-- mitoses: string (nullable = true)
|-- target: string (nullable = true)
```

Affichage des 10 premières lignes du DataFrame ''breast.all''

```
Console ~/
> SparkR::showDF(breast.all,10)
+-----+---------+----------+----------+------+-------+----------+---------+-------+---------+
|clump|ucellsize|ucellshape|mgadhesion|sepics|bnuclei|bchromatin|normnucl|mitoses|   target|
+-----+---------+----------+----------+------+-------+----------+---------+-------+---------+
|    4|        2|         2|         1|     2|      1|         2|        1|      1|   begnin|
|    1|        1|         1|         1|     2|      1|         2|        1|      1|   begnin|
|    2|        1|         1|         1|     2|      1|         2|        1|      1|   begnin|
|   10|        6|         6|         2|     4|     10|         9|        7|      1|malignant|
|    4|        1|         1|         1|     2|      1|         2|        1|      1|   begnin|
|    1|        1|         1|         1|     2|      1|         1|        1|      1|   begnin|
|    1|        1|         1|         1|     2|      1|         2|        1|      1|   begnin|
|    5|        1|         1|         1|     2|      1|         2|        1|      1|   begnin|
|    3|        1|         1|         1|     2|      1|         2|        1|      1|   begnin|
|    1|        1|         1|         1|     2|      4|         2|        1|      1|   begnin|
+-----+---------+----------+----------+------+-------+----------+---------+-------+---------+
only showing top 10 rows
>
```

Etape 3 : Conversion de types

```
# conversions de types pour les descripteurs de breast.all
breast.all$clump <- SparkR::cast(breast.all$clump, "double")
breast.all$ucellsize <- SparkR::cast(breast.all$ucellsize, "double")
breast.all$ucellshape <- SparkR::cast(breast.all$ucellshape, "double")
breast.all$mgadhesion <- SparkR::cast(breast.all$mgadhesion, "double")
breast.all$sepics <- SparkR::cast(breast.all$sepics, "double")
breast.all$bnuclei <- cast(breast.all$bnuclei, "double")
breast.all$bchromatin <- cast(breast.all$bchromatin, "double")
breast.all$normnucl <- cast(breast.all$normnucl, "double")
breast.all$mitoses <- cast(breast.all$mitoses, "double")
# affichage du nouveau schéma du DataFrame
SparkR::printSchema(breast.all)
```

Les descripteurs "clump"..."mitoses" ont été reconnues comme « string ». Cela ne convient pas pour les calculs (régression logistique). Il faut les convertir en réel (double) dans le DataFrame ''breast.all''

```
# nouveau schéma
root
|-- clump: double (nullable = true)
|-- ucellsize: double (nullable = true)
|-- ucellshape: double (nullable = true)
|-- mgadhesion: double (nullable = true)
|-- sepics: double (nullable = true)
|-- bnuclei: double (nullable = true)
|-- bchromatin: double (nullable = true)
|-- normnucl: double (nullable = true)
|-- mitoses: double (nullable = true)
|-- target: string (nullable = true)
```

''target'' est la variable cible qualitative, on peut la laisser au format « string »

Etape 4 : Subdivision des données en échantillons d'apprentissage et de test

Pour disposer d'une mesure honnête des performances du modèle dans la population, il faut l'évaluer sur un échantillon qui n'a pas pris part à sa construction. Habituellement, on scinde en 2 les données disponibles : la première sert à l'apprentissage, la seconde (test) sert à l'évaluation.

```
# compter le nombre de lignes
n <- SparkR::nrow(breast.all)
print(n) # 699

# ajouter dans le data.frame une colonne de valeurs aléatoires U(0, 1)
# on va s'en servir pour scinder les données
breast.all$alea <- SparkR::rand(n)
SparkR::printSchema(breast.all)

# extraction de l'échantillon d'apprentissage à partir d'une condition sur ''alea''
# approximativement 2/3 des individus disponibles
# un nouveau DataFrame SparkR nommé ''breast.train'' est créé
# la colonne additionnelle ''alea'' n'est pas incluse (1:10)
breast.train <- SparkR::subset(breast.all, breast.all$alea <= 0.667, 1:10)
SparkR::printSchema(breast.train)
print(SparkR::nrow(breast.train)) # 455

# échantillon test (les autres)
breast.test <- SparkR::subset(breast.all, breast.all$alea > 0.667, 1:10)
print(SparkR::nrow(breast.test)) # 244
```

Etape 5 : Modélisation et affichage des résultats – Régression logistique

```
# construction d'un modèle de régression logistique
# target vs. toutes les autres variables du DataFrame ‘’breast.train’’
modele <- SparkR::glm(target ~ ., family = "binomial", data = breast.train)
# affichage
print(modele)
# affichage du summary
print(SparkR::summary(modele))
```

```
# print(modele)
An object of class "PipelineModel »
Slot "model": Java ref type
org.apache.spark.ml.PipelineModel id 290

#affichage du summary
$coefficients Estimate
(Intercept)-10.9809891
clump       0.5609039
ucellsize -0.1896514
ucellshape 0.3820318
mgadhesion 0.3394497
sepics      -0.0805315
bnuclei     0.5316810
bchromatin 0.4646330
normnucl    0.4338739
mitoses     0.8756312
```

Pas grand-chose dans le print(), summary() en revanche fournit les coefficients de l’équation LOGIT

Etape 6 : Prédiction sur l'échantillon test

''**target**'', cible observée sur l'échantillon test ; ''**label**'', target recodée en 0/1 ; ''**rawPrediction**'', valeur du logit ; ''**probability**'', probabilité d'appartenance aux classes, transformation du logit via la fonction logistique ; ''**prediction**'', prédiction du modèle : proba < 0.5 alors 0 (begnin) sinon 1 (malignant)

```
+------+-----+--------------------+--------------------+----------+
|target|label|       rawPrediction|         probability|prediction|
+------+-----+--------------------+--------------------+----------+
|begnin|  0.0|[7.27886612233866...|[0.99931050826784...|       0.0|
|begnin|  0.0|[7.74349908156457...|[0.99956663600131...|       0.0|
|begnin|  0.0|[7.27886612233866...|[0.99931050826784...|       0.0|
|begnin|  0.0|[6.15705834996786...|[0.99788600252317...|       0.0|
|begnin|  0.0|[6.81423316311276...|[0.99890316947111...|       0.0|
|begnin|  0.0|[3.83546639202643...|[0.97886506390269...|       0.0|
|begnin|  0.0|[6.81423316311276...|[0.99890316947111...|       0.0|
|begnin|  0.0|[6.40323491823817...|[0.99834654747738...|       0.0|
|begnin|  0.0|[7.66296757701046...|[0.99953030986949...|       0.0|
|begnin|  0.0|[6.81423316311276...|[0.99890316947111...|       0.0|
+------+-----+--------------------+--------------------+----------+
only showing top 10 rows
```

```r
# prédiction sur l'échantillon test
pred <- SparkR::predict(modele, newData = breast.test)
# structure (schéma) de pred qui est un DataFrame
SparkR::printSchema(pred)
# affichage des valeurs pour les 10 premiers
SparkR::showDF(SparkR::select(pred,c('target','label','rawPrediction','probability','prediction')),10)

# recodage de la prédiction pour être conforme avec la description de nos données (type target = string)
# la nouvelle colonne ''predTarget'' est insérée dans le DataFrame ''pred''
pred$predTarget <- SparkR::ifelse(pred$prediction == 0, "begnin", "malignant")
#affichage des valeurs pour les 10 premiers
SparkR::showDF(SparkR::select(pred,c('target','label','rawPrediction','probability','prediction','predTarget')),10)
```

```
+------+-----+--------------------+--------------------+----------+----------+
|target|label|       rawPrediction|         probability|prediction|predTarget|
+------+-----+--------------------+--------------------+----------+----------+
|begnin|  0.0|[7.27886612233866...|[0.99931050826784...|       0.0|    begnin|
|begnin|  0.0|[7.74349908156457...|[0.99956663600131...|       0.0|    begnin|
|begnin|  0.0|[7.27886612233866...|[0.99931050826784...|       0.0|    begnin|
|begnin|  0.0|[6.15705834996786...|[0.99788600252317...|       0.0|    begnin|
|begnin|  0.0|[6.81423316311276...|[0.99890316947111...|       0.0|    begnin|
|begnin|  0.0|[3.83546639202643...|[0.97886506390269...|       0.0|    begnin|
|begnin|  0.0|[6.81423316311276...|[0.99890316947111...|       0.0|    begnin|
|begnin|  0.0|[6.40323491823817...|[0.99834654747738...|       0.0|    begnin|
|begnin|  0.0|[7.66296757701046...|[0.99953030986949...|       0.0|    begnin|
|begnin|  0.0|[6.81423316311276...|[0.99890316947111...|       0.0|    begnin|
+------+-----+--------------------+--------------------+----------+----------+
only showing top 10 rows
```

La colonne ''predTarget'' a été accolée au DataFrame, on aura alors à confronter ''target'' (Y) et ''predTarget'' (Y^) (cf. page 21)

Etape 7 : Matrice de confusion et taux d'erreur

```
# 1. récupérer les infos dans un data.frame local (type R)
pred.mem <- SparkR::collect(select(pred,c('target','predTarget')))

# 2. matrice de confusion
mc <- base::table(pred.mem$target,pred.mem$predTarget)
print(mc)

# 3. taux d'erreur
print(1-sum(diag(mc))/sum(mc))

# fin de session sparkR, ne pas oublier (!)
sparkR.stop()
```

On utilise un artifice pour pouvoir élaborer la matrice de confusion : (**1**) on monte les colonnes ''target'' et ''predTarget'' dans un data.frame (type R) en mémoire locale [c'est le rôle de SparkR::collect()], (**2**) à partir de là on les croise avec base::table() [fonction de « base » de R], (**3**) puis on calcule le ratio « taux d'erreur ».

```
           begnin malignant
begnin        162         6
malignant       5        71
```

$$\text{taux d'erreur} = \frac{5+6}{244} = 0.045$$

Matrice de confusion sur l'échantillon test et taux d'erreur.

Références

Références

Machine Learning Library (MLlib) Guide

Liste des méthodes de Machine Learning disponibles dans MLlib.

SparkR (R on Spark)

Structures et algorithmes disponibles. Pour l'heure (ver. 1.6.1), seuls les régressions linéaires et logistiques sont disponibles.

Daniel Emaasit, « Installing and Starting SparkR Locally on Windows OS and Rstudio », R-bloggers, July 2015.

Détaille le processus d'installation de Spark / SparkR sous Windows + un exemple très simple d'accès aux données.

Alban Phelip, « Découvrez SparkR, la nouvelle API de Spark », Blog Xebia, Sep. 2015.

La trame de ma démo est très proche de la sienne, à la différence que d'autres données sont utilisées, la subdivision aléatoire (apprentissage / test) des données est réalisée dans le programme R, et que la matrice de confusion est calculée explicitement.