



Mise en œuvre de solutions de reproductibilité à l'aide
des principaux notebook et d'un système de workflow

Antoine COSSA
Marine FIGAROL
Cécile MOUREAUX
Christelle VIGUIER



Galaxy / Europe

Analyze Data Workflow Visualize Shared Data Help Login or Register

Tools

search tools

[Support vector machines \(SVMs\) for classification](#)
[MINE - Maximal Information-based Nonparametric Exploration](#)
[Draw ROC plot on "Perform LDA" output](#)
[Perform LDA Linear Discriminant Analysis](#)
[Generate A Matrix for using PC and LDA](#)
[Count GFF Features](#)
[Correlation for numeric columns](#)
[Perform Best-subsets Regression](#)
[Principal Component Analysis](#)
[Kernel Principal Component Analysis](#)
[Kernel Canonical Correlation Analysis](#)
[Canonical Correlation Analysis](#)
[Wavelet variance using Discrete Wavelet Transforms](#)
[T Test for Two Samples](#)
[Nearest Neighbors Classification](#)
[Support vector machines \(SVMs\) for classification](#)
[Anova N-way anova. With ou Without interactions](#)
[Multivariate PCA, PLS and OPLS](#)
[Univariate Univariate statistics](#)
[JWTomics Plot with Threshold on Test Scale](#)
[JWTomics Test and Plot](#)

Kernel Principal Component Analysis (Galaxy Version 1.0.0) Options

Select data

No tabular dataset available.

Dataset missing? See TIP below.

Select columns containing input variables

☐ Select/Unselect all

Missing columns in referenced dataset.

Number of principal components to return

2

To return all, enter 0

Kernel function

Gaussian Radial Basis Function

sigma (inverse kernel width)

1.0

Execute

TIP: If your data is not TAB delimited, use *Edit Datasets*->*Convert characters*

What it does

This tool uses functions from 'kernlab' library from R statistical package to perform Kernel Principal Component Analysis (kPCA) on the input data. It outputs two files, one containing the summary statistics of the performed kPCA, and the other containing a scatterplot matrix of rotated values reported by kPCA.

Alexandros Karatzoglou, Alex Smola, Kurt Hornik, Achim Zeileis (2004). kernlab - An S4 Package for Kernel Methods in R. Journal of Statistical Software 11(9), 1-20. URL <http://www.jstatsoft.org/v11/i09/>

Note

This tool currently treats all variables as continuous numeric variables. Running the tool on categorical variables might result in incorrect results. Rows containing non-numeric (or missing) data in any of the chosen columns will be skipped from the analysis.

History

search datasets

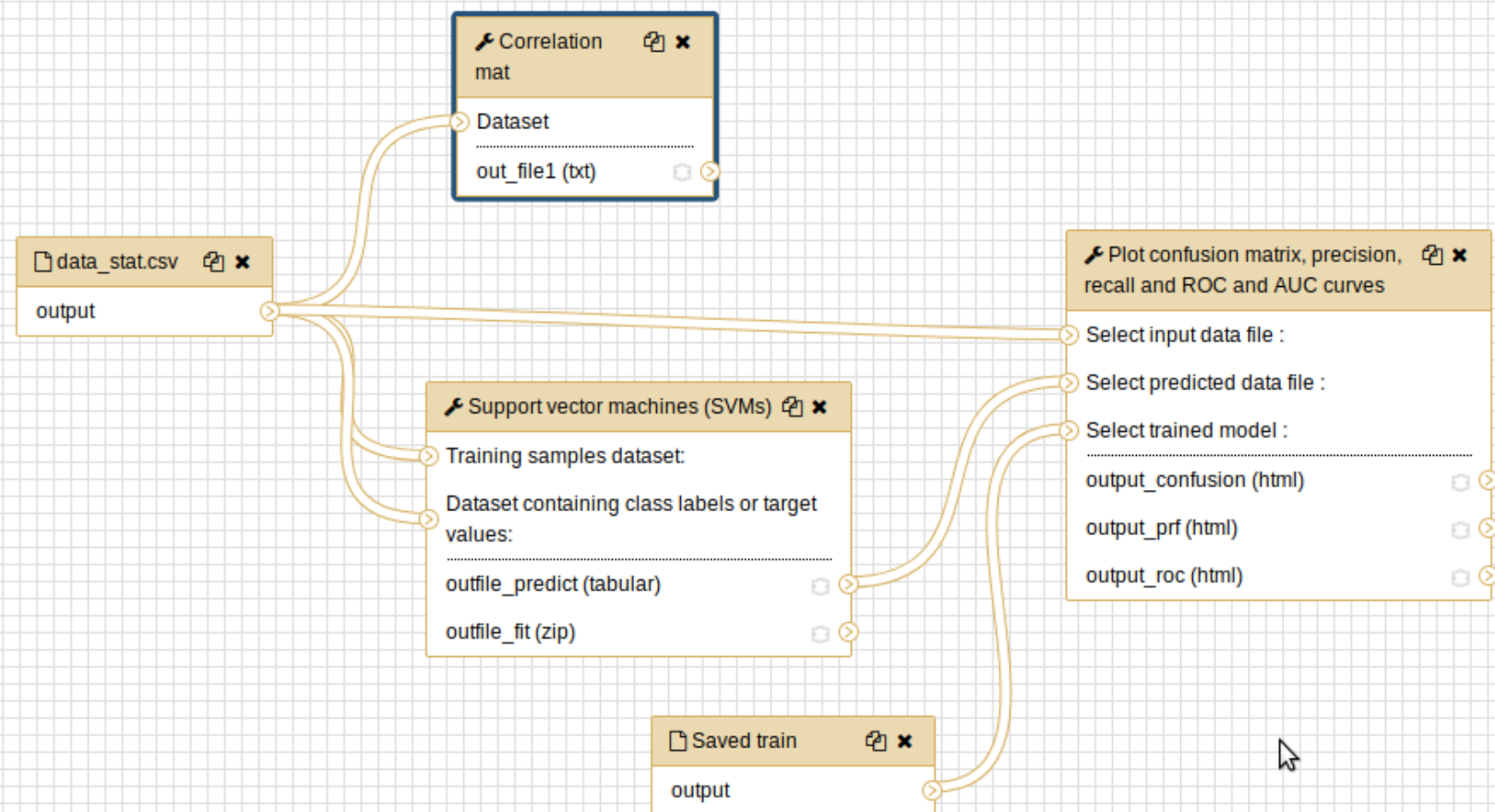
Unnamed history

(empty)

This history is empty. You can [load your own data](#) or [get data from an external source](#)

Inputs

- [Input dataset](#)
- [Input dataset collection](#)

FILE AND META TOOLS[Get Data](#)[Send Data](#)[Convert Formats](#)[Collection Operations](#)**GENERAL TEXT TOOLS**[Text Manipulation](#)[Filter and Sort](#)[Join, Subtract and Group](#)**GENOMICS, NGS**[Extract Features](#)[BED Tools](#)[Fetch Alignments](#)[Operate on Genomic Intervals](#)[FASTA/FASTQ manipulation](#)[Multiple Alignments](#)[FASTA/FASTQ manipulation](#)[Picard](#)[Quality Control](#)[Assembly](#)[Mapping](#)[Variant Calling](#)[Genome editing](#)

Notebooks

- Interface web + Kernel
- Combine texte formaté / code
- Multi-langage
- Portabilité / Reproductibilité



Apache Zeppelin

Installation



- Installation rapide sous Windows et Linux

```
python3 -m pip install --upgrade pip  
python3 -m pip install jupyter
```

- Lancement facile et rapide

```
jupyter notebook
```



Python nécessaire pour installer Jupyter Notebook



- Téléchargement : <https://www.rstudio.com/products/rstudio/download/>
- Plusieurs installations possibles. Assez rapide sous Linux ; très compliquée sous Windows



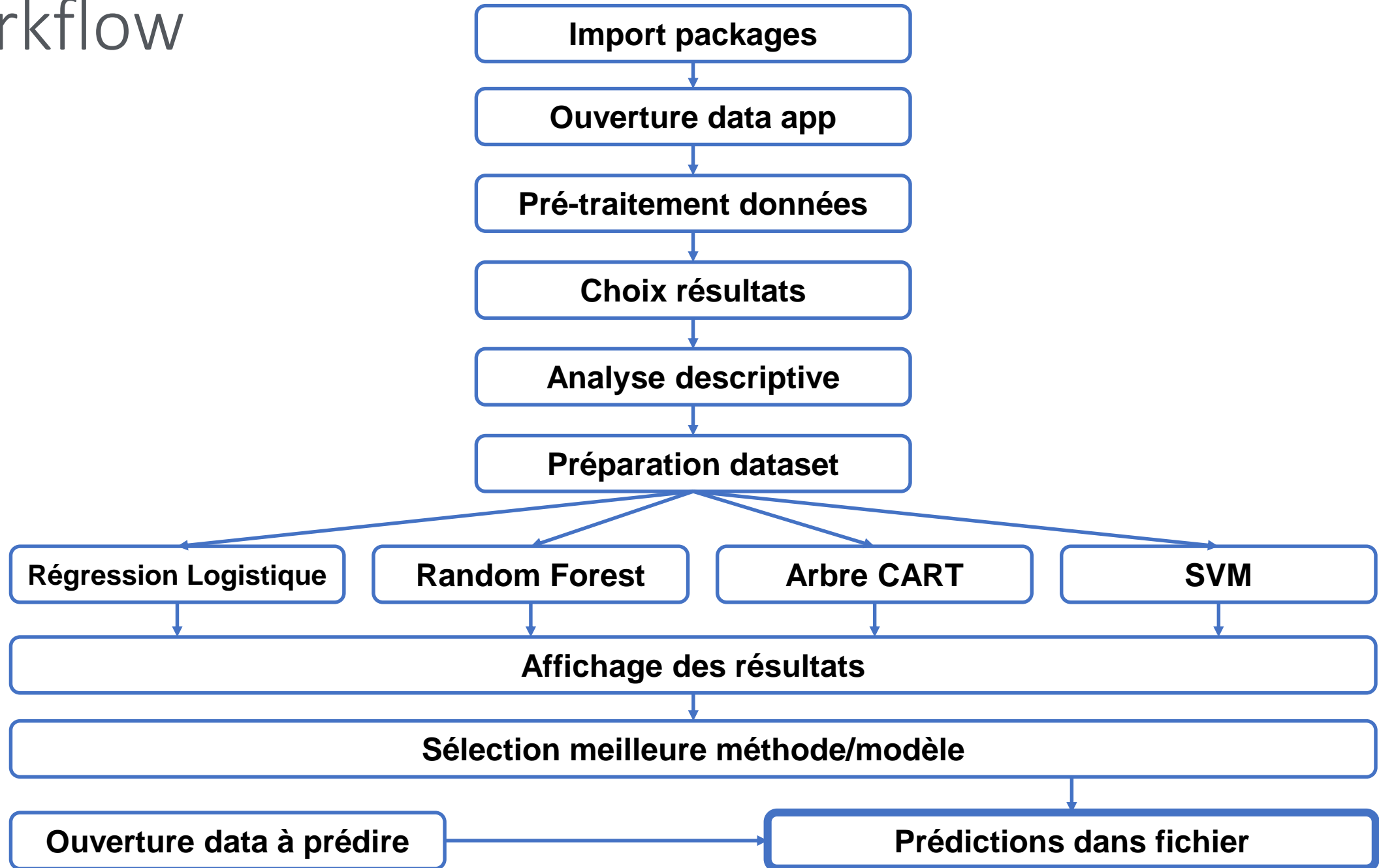
JAVA, Hadoop, Spark, Git, Scala, Maven nécessaires



- Lancement facile et rapide sous Linux et Windows

```
bin/zeppelin-daemon.sh start  
bin\zeppelin.cmd
```

Workflow



Dataset1 : Cancer du sein

Variable d'intérêt : rechute

Co variables : niveau d'expression de gènes

But : prédire la rechute ou non-rechute des patientes

Rechute	Non Rechute
1	-1

Dataset2 : Spam (test de la reproductibilité)

Variable d'intérêt : type

Co variables : fréquence de certains mots et caractères

But : prédire si il s'agit d'un spam ou non

Spam	Non spam
spam	nospam



Présentation générale

Interface

 Quit Logout

Files Running Clusters

Select items to perform actions on them. Upload New ▾ ↺

☐ 0 ▾

/ Desktop / Projet_Big_Data

Name ▾

<input type="checkbox"/>	..	il y a q	
<input type="checkbox"/>	BD.ipynb		kB
<input type="checkbox"/>	R_BD.ipynb	Actif il y a q	kB
<input type="checkbox"/>	COSSA-MOUREAUX.RMD		kB
<input type="checkbox"/>	donnees_liver.rda	il y a 8 jours	100 kB
<input type="checkbox"/>	spam.csv	il y a 8 jours	743 kB
<input type="checkbox"/>	train.txt	il y a 19 heures	93.5 kB

Notebook:

Python 3

R

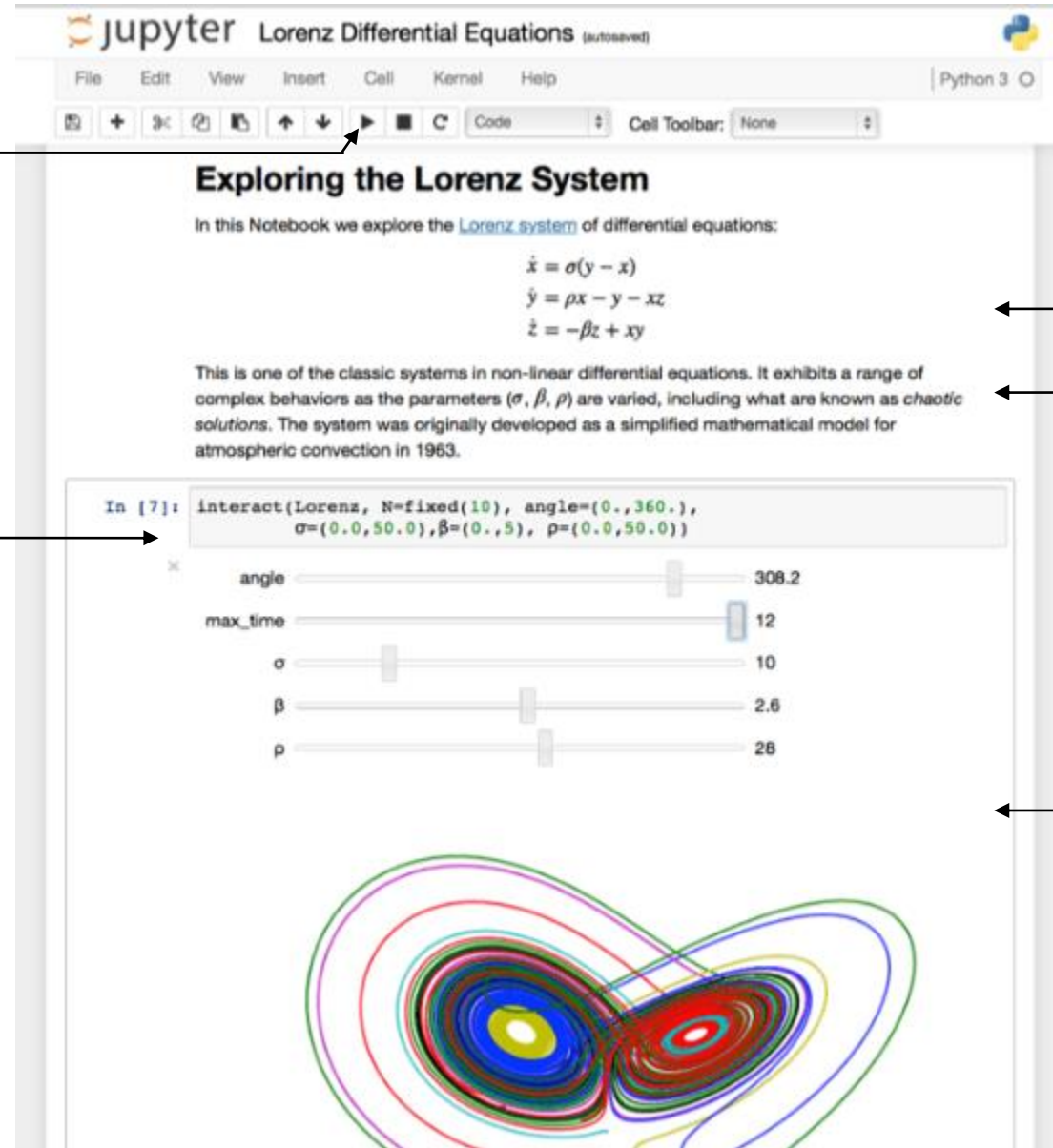
Other:

Text File

Folder

Terminal

Exemple



Exécution de la
cellule

Code

Langage utilisé

Texte enrichi

Texte brut

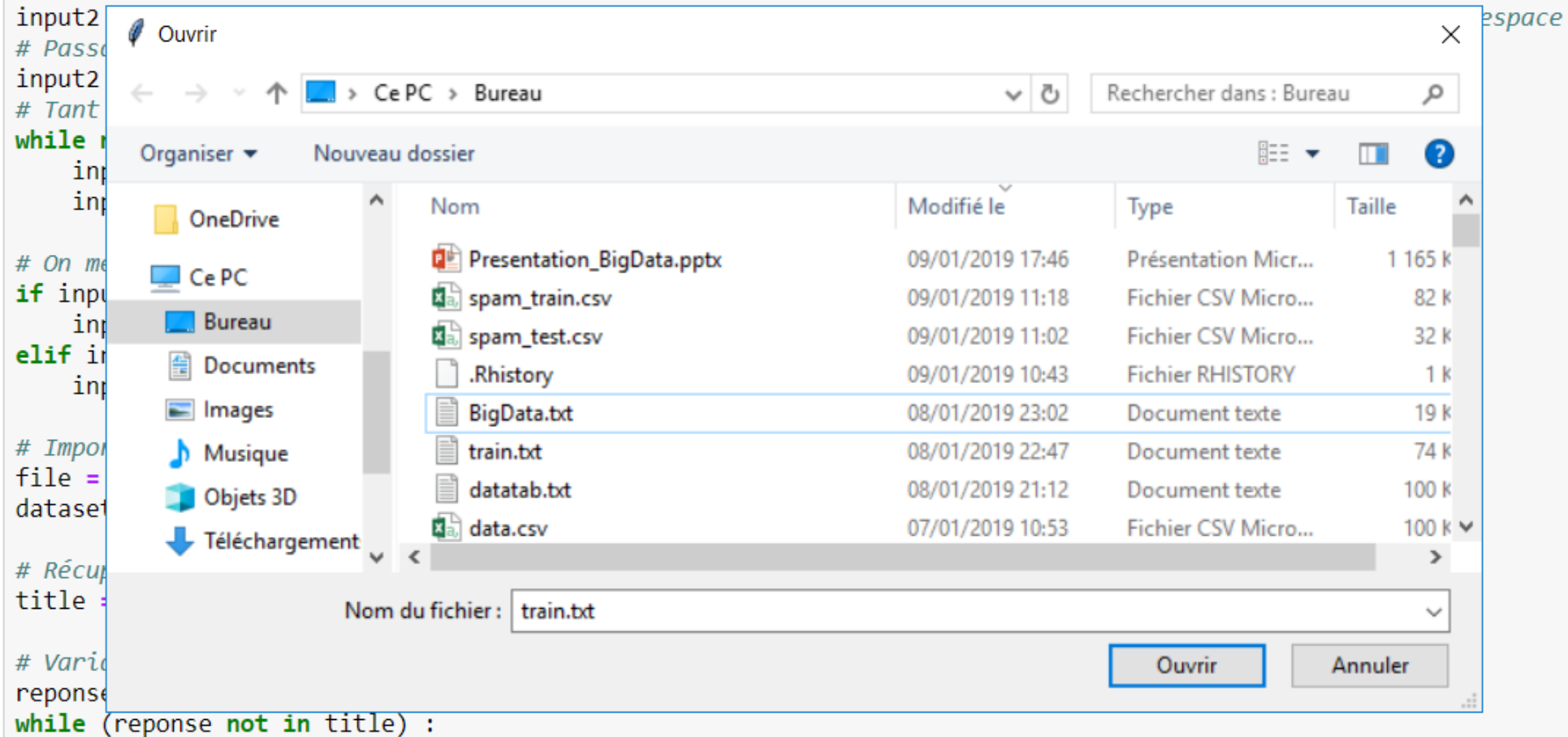
Sortie

Jupyter avec Python



Sélection dataset

```
# Permet à l'utilisateur de choisir un fichier
root = Tk()
root.withdraw()
root.call('wm', 'attributes', '.', '-topmost', True)
input1 = filedialog.askopenfilename()
%gui tk
```



Clé de la reproductibilité : les inputs

```
input2 = input("Indiquez la manière dont sont séparées vos variables (tabulation/espace/;) : ")  
# Passage en minuscule si l'utilisateur écrit en majuscule  
input2 = input2.lower() ;  
# Tant que l'utilisateur n'a pas rentré quelque chose de correct  
while not (input2 == 'tabulation' or input2 == 'espace' or input2 == ';') :  
    input2 = input("Veuillez entrer tabulation, espace ou ; : ")  
    input2 = input2.lower()
```

Indiquez la manière dont sont séparées vos variables (tabulation/espace/;) :

```
reponse = input("Indiquez le nom de la variable d'intérêt : ")  
while (reponse not in title) :  
    reponse = input("Cette variable n'existe pas ! Veuillez vérifier son nom et essayer à nouveau : ")
```

Indiquez le nom de la variable d'intérêt :

Statistiques descriptives

Entrée [4]: `dataset.head()`

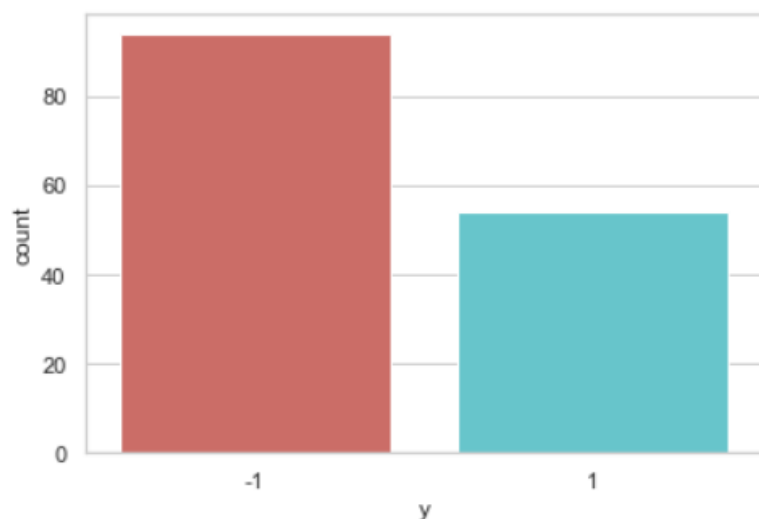
Out[4]:

	x10006_at	x10007_at	x100129361_at	x100130449_at	x100130633_at
0	8.44	7.84	9.23	2.20	4.93
1	7.93	8.47	7.77	2.48	6.70
2	7.53	7.49	9.49	5.66	5.99
3	7.72	8.38	8.03	5.22	6.95
4	7.89	7.75	6.99	2.20	7.06

5 rows × 101 columns

Répartition de la variable d'intérêt

Entrée [8]: `sns.countplot(x = reponse, data = dataset, palette = 'hls')`
`plt.show()`



Dimension de votre dataset

```
print(dataset.shape)
```

(148, 101)

Visualisation du résumé de la variable d'intérêt

```
y.describe()
classe = list(set(y))
nb_classe = len(list(set(y)))

print("Différentes classes de la variable d'intérêt : ", classe)
print("Nombre de classes : ", nb_classe)
```

Différentes classes de la variable d'intérêt : [1, -1]
Nombre de classes : 2

Analyses statistiques

Random Forest

```
Entrée [14]: rf_model = RandomForestClassifier()

# Etablissement des différents paramètres à tester
parameter_grid = {'n_estimators': [10, 25, 50, 100],
                  'criterion': ['gini', 'entropy'],
                  'max_features': [1, 2, 3, 4]}

# Cross Validation (on fait tourner 10 fois le modele sur differents decoupage)
cross_validation = StratifiedKFold(n_splits=10)

# Sélection des meilleurs paramètres
grid_search = GridSearchCV(rf_model,
                           param_grid=parameter_grid,
                           cv=cross_validation)

grid_search.fit(XTrain, yTrain_)

# Visualisation des meilleurs paramètres
print('Best parameters: {}'.format(grid_search.best_params_))

# Stockage des meilleurs paramètres
nestim_best = grid_search.best_params_['n_estimators']
criterion_best = grid_search.best_params_['criterion']
max_features_best = grid_search.best_params_['max_features']
```

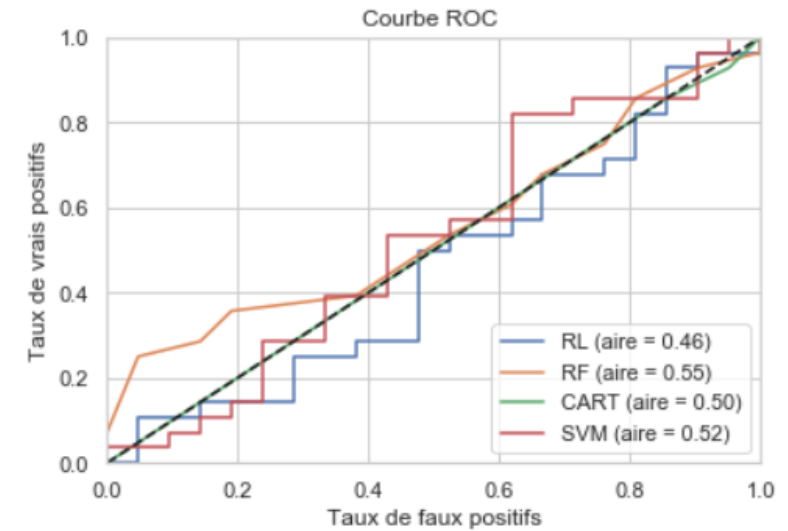

Affichage des résultats

Précision de la Régression Logistique : 57.1429 %

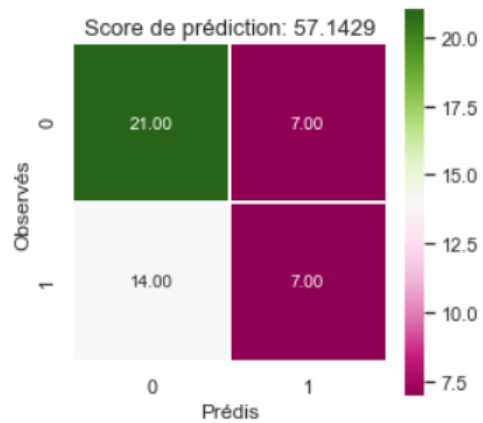
Précision du Random Forest : 57.1429 %

Précision de l'arbre CART : 55.102 %

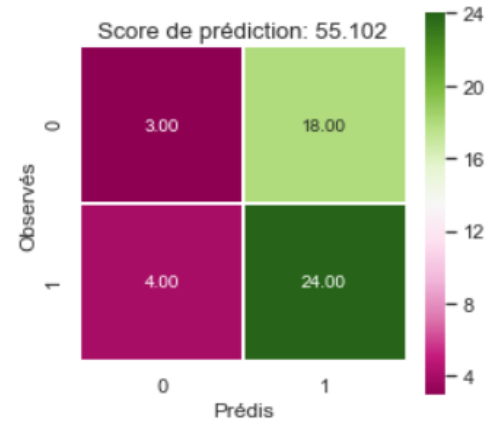
Précision de SVM : 57.1429 %



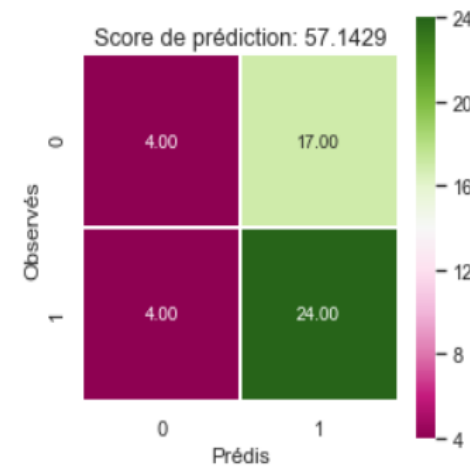
Matrice de confusion de la Régression Logistique:



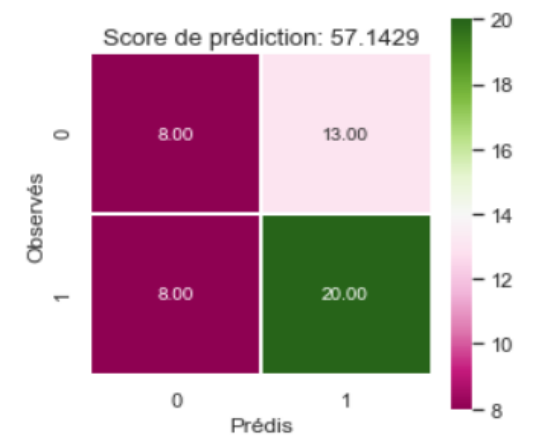
Matrice de confusion de CART:



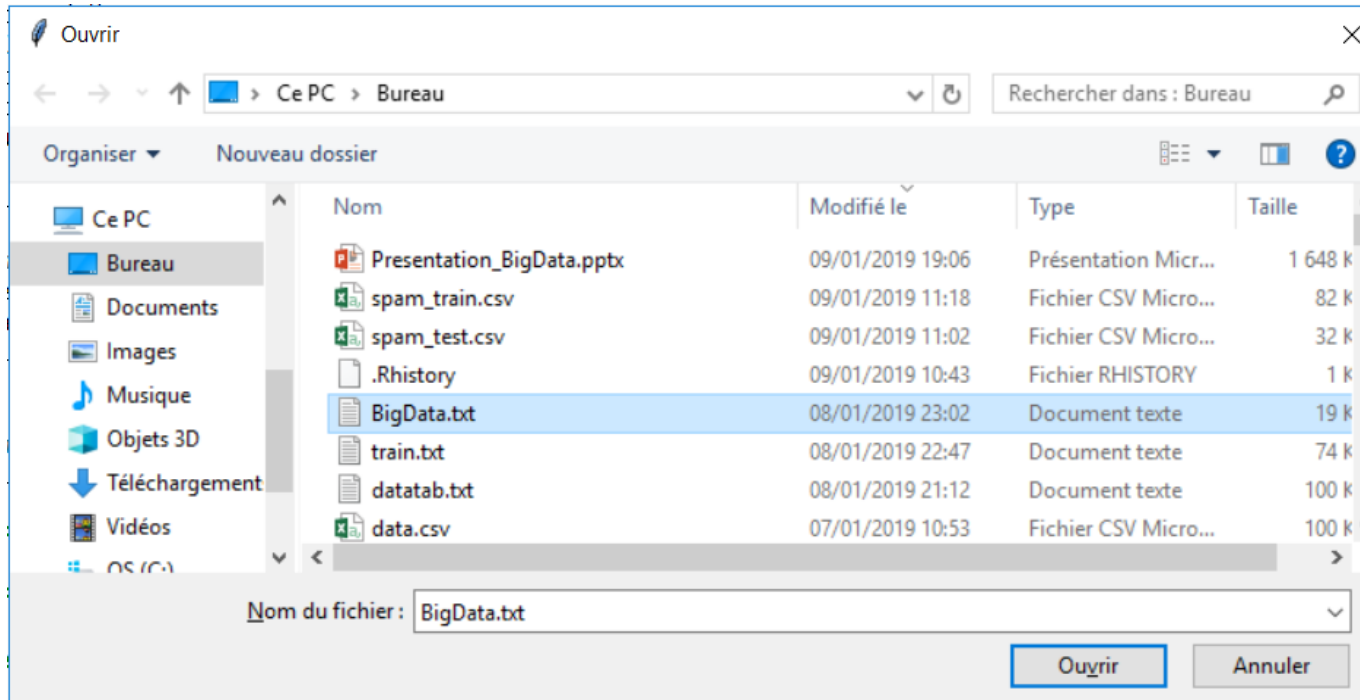
Matrice de confusion du Random Forest:



Matrice de confusion SVM :



Prédiction sur fichier à traiter



Enregistrer sortie
au format csv

```
# On cherche la méthode sélectionné et on effectue la prédiction à partir de ce modèle
if ind == 0 :
    pred = logit_model.predict(dataset2)
elif ind == 1 :
    pred = rf_model.predict(dataset2)
elif ind == 2 :
    pred = cart_model.predict(dataset2)
else :
    pred = svm_model.predict(dataset2)
```

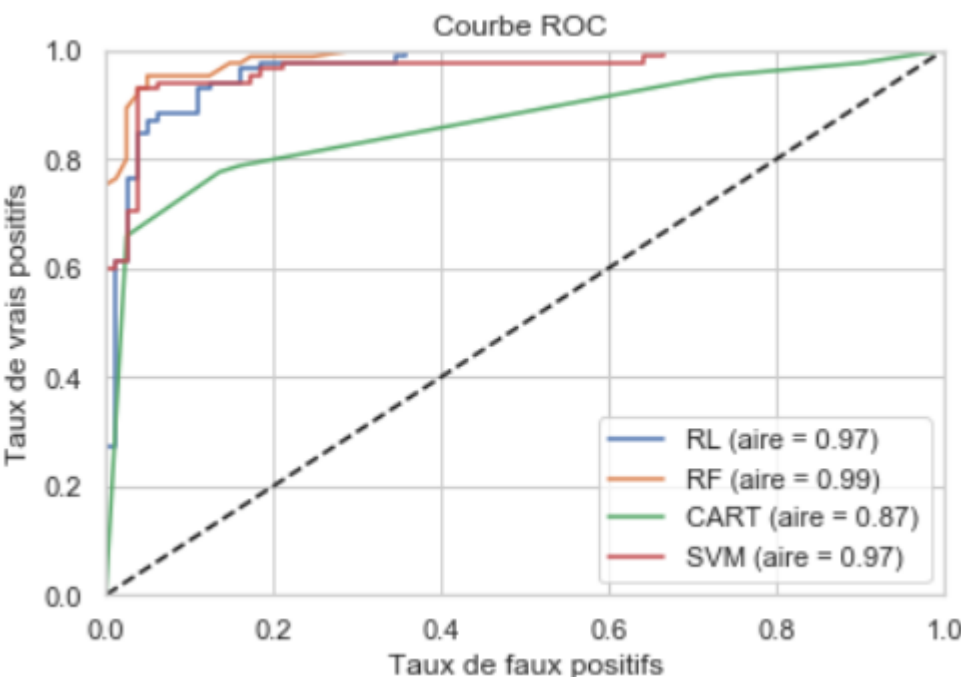
Reproductible ? Visualisation des résultats avec spam

Précision de la Régression Logistique : 90.3614 %

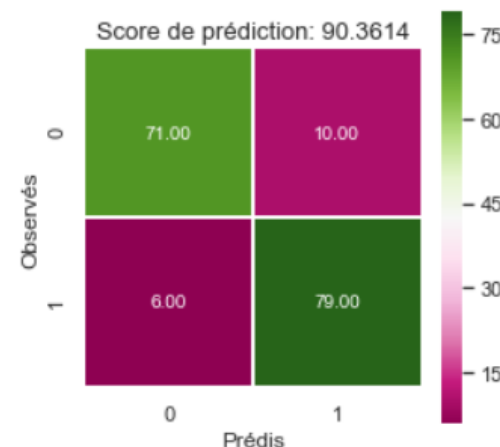
Précision du Random Forest : 95.1807 %

Précision de l'arbre CART : 81.3253 %

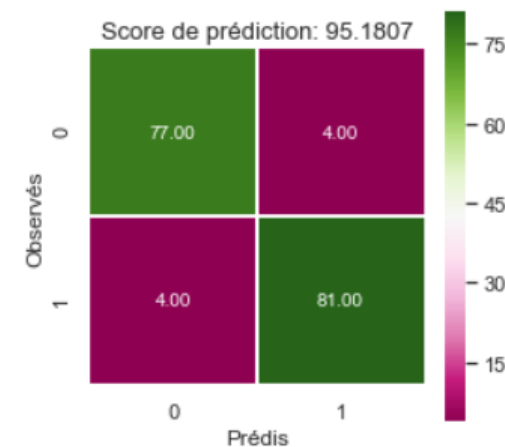
Précision de SVM : 93.3735 %



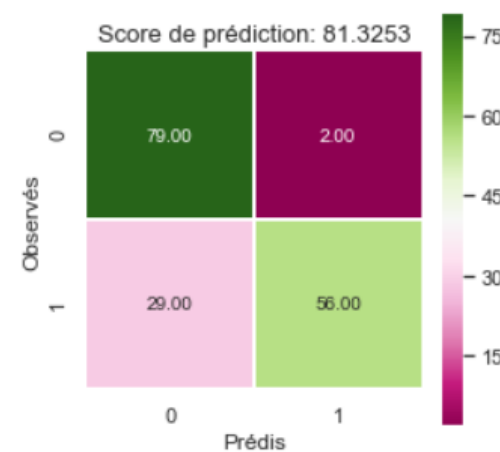
Matrice de confusion de la Régression Logistique:



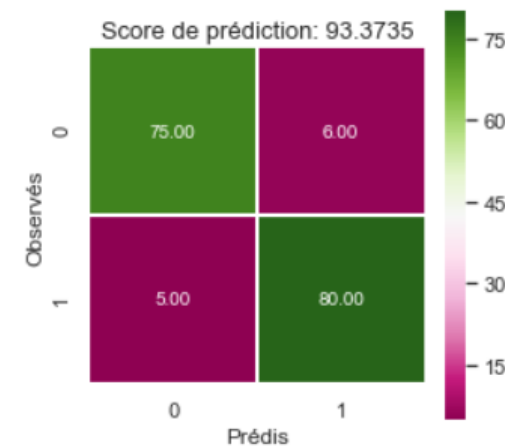
Matrice de confusion du Random Forest:



Matrice de confusion de CART:



Matrice de confusion SVM :



Jupyter avec
R



Installation IR Kernel

```
update.packages()  
install.packages('IRkernel')  
IRkernel::installspec(user = FALSE)
```

Installation IR Kernel

The screenshot shows the Anaconda Navigator application window. The title bar is blue with the Anaconda Navigator logo and window controls. Below the title bar is a menu bar with 'File' and 'Help'. The main interface has a sidebar on the left with icons for Home, Environments, Learning, and Community, and buttons for Documentation and Developer Blog. The main content area is divided into two panes. The left pane shows a list of environments: 'base (root)' and 'r-tensorflow'. The right pane shows the 'r-irkernel' package installed in the 'base (root)' environment. The package is listed in a table with columns for Name, Description, and Version. The version is 0.8.12. A status bar at the bottom of the right pane indicates '1 package available matching "r-irkernel"'. A search bar at the top of the right pane contains 'r-irkernel'.

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Search Environments

base (root)

r-tensorflow

Created

Clone

Import

Remove

Installed

Channels

Update index...

r-irkernel

Name	Description	Version
✓ r-irkernel		0.8.12

1 package available matching "r-irkernel"

Script interactif : boîtes de dialogue

jupyter R_BD Dernière Sauvegarde : il y a 20 heures (modifié) Logout

File Edit View Insert Cell Kernel Widgets Help De Confiance R

Exécuter

Importation des données :

```
Entrée [*]: filepath = file.choose(new = FALSE)
sep = readline(prompt = "Enter the type of separator")
sep = str_to_lower(sep)
while (sep != 'tab' && sep != 'space' && sep != ',') {
  sep = readline(prompt = "Enter the type of separator")
  sep = str_to_lower(sep)
}
if (sep == 'tab') {
  sep = '\t'
} else if (sep == 'space') {
  sep = ' '
}
print(sep)

Entrée [22]: #filepath = "spam.csv" # filepath of your data
data = read.table(file = filepath, header = TRUE, as.is = TRUE)

Entrée [23]: total_rows = nrow(data)
total_columns = ncol(data)
title = names(data) # Récupération du nom des colonnes

Entrée [24]: head(data)
summary(data)
```

7.53 7.49 9.49 5.66 5.99 8.87 3.48 8.02 10.73 4.07 ... 9.61

Select file

< > << Desktop > Projet_Big_Data Search Projet_Big_Data

Organise New folder

Name	Date modified	Type
Présentation1.pptx	10/01/2019 18:40	Présentation Mic
R_BD.ipynb	09/01/2019 22:02	IPYNB File
R_BD.r	09/01/2019 16:58	R File
R_BD.RMD	09/01/2019 17:01	RMD File
R_BD.zip	09/01/2019 17:04	Compressed (zip)
R_BD_RMD.RMD	09/01/2019 17:03	RMD File
spam.csv	09/01/2019 18:32	Fichier CSV Micro
spam_light.csv	09/01/2019 18:39	Fichier CSV Micro
spam_test.csv	09/01/2019 18:41	Fichier CSV Micro
spam_train.csv	09/01/2019 18:41	Fichier CSV Micro
train.txt	08/01/2019 21:48	Text Document

File name: spam_train.csv All files (*.*)

Open Cancel

Script interactif : inputs

Importation des donnees :

```
Entrée [*]: ▶ filepath = file.choose(new = FALSE)
sep = readline(prompt = "Enter the type of separator (;/,/tab/space) :")
sep = str_to_lower(sep)
while (sep != 'tab' && sep != 'space' && sep != ';' && sep != ',') {
  sep = readline(prompt = "Enter the type of separator (;/,/tab/space) :")
  sep = str_to_lower(sep)
}
if (sep == 'tab') {
  sep = '\t'
} else if (sep == 'space') {
  sep = ' '
}
print(sep)
```

Enter the type of separator (;/,/tab/space) :

Analyse descriptive

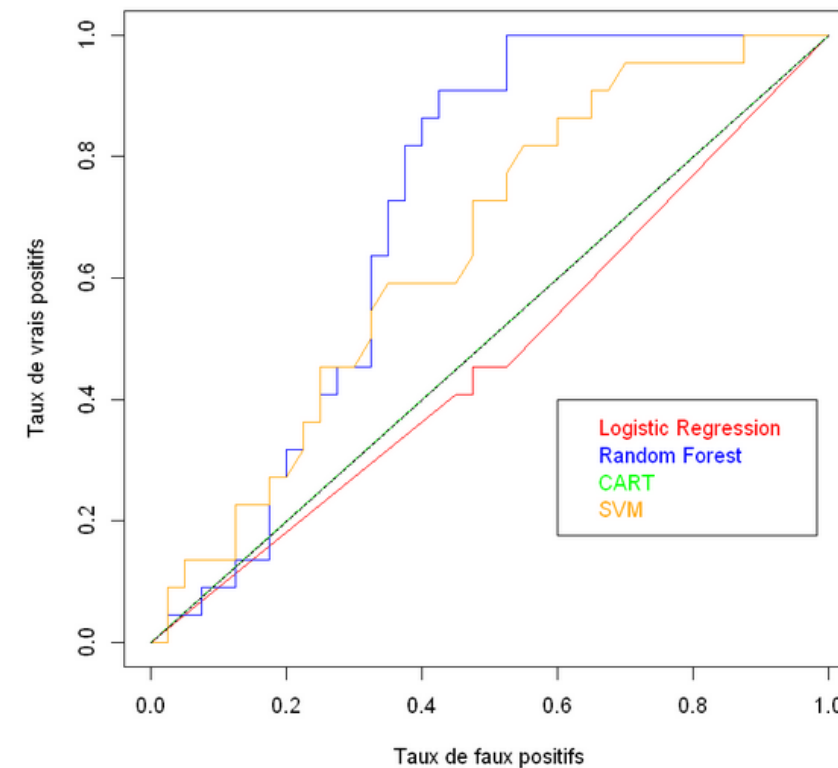
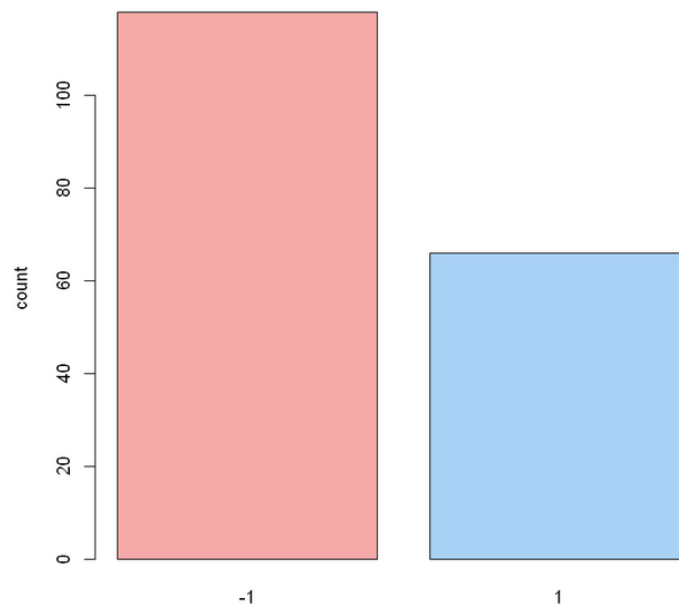
Entrée [76]: ▶ `head(data)`

x10006_at	x10007_at	x100129361_at	x100130449_at	x100130633_at	x100133941_at	x10014_at	x10016_at	x100188893_at	x10019_at	...	x10226_at	x10227_at
8.44	7.84	9.23	2.20	4.93	10.31	7.85	9.91	11.37	3.48	...	10.55	10.55
7.93	8.47	7.77	2.48	6.70	11.03	3.33	9.53	10.91	4.28	...	9.00	9.00
7.53	7.49	9.49	5.66	5.99	8.87	3.48	8.02	10.73	4.07	...	9.61	9.61
7.72	8.38	8.03	5.22	6.95	2.73	4.06	9.26	10.90	7.40	...	9.91	9.91
7.89	7.75	6.99	2.20	7.06	11.38	3.65	9.22	10.37	3.89	...	9.60	9.60
8.37	8.43	8.32	2.20	7.02	10.70	3.17	7.73	11.18	4.41	...	8.76	8.76

Courbes ROC

Répartition de la variable d'intérêt

Entrée [80]: ▶ `tbl <- with(data, table(y))`
`barplot(tbl, beside = TRUE, col = c("#F5A9A9", "#A9D0F5"), xlab = y, ylab = "count")`



Avantages/inconvénients

- Portabilité
- Possibilité d'exporter le code R mais pas RMD
- Variables en mémoire non visibles



Présentation générale

Interface

The screenshot displays the RStudio interface with three main components highlighted by text boxes:

- R script:** The script editor shows R code for biomass calculation per tree, including data loading and plotting functions.
- R console:** The console window shows the execution of R commands, including merging data and calculating biomass.
- Graphical output:** The Environment pane shows the Global Environment with variables like `hil.trees`, `kal.plot`, `kalimantan`, `lsi.plots`, `lsi`, `pub`, and `wei`. The Plots pane displays a box plot titled "Biomass estimation per plot with different models" showing biomass (Mg/ha) for different models.

R script

```
200
201 # Biomass calculation per tree
202 kalimantan$w.brown<-brown.moist.d(kalimantan$dbh)
203 kalimantan$w.yamakura<-yamakura.stem(kalimantan$dbh, kalimantan$h)+yamakura.branch(yamakura.stem(k
204 kalimantan$w.basuki<-basuki.mixed.d(kalimantan$dbh)
205 kalimantan$w.samalca<-samalca.d(kalimantan$dbh)
206 kalimantan$w.hashimoto<-hashimoto.d(kalimantan$dbh)
207 kalimantan$w.kenzo<-kenzo.d(kalimantan$dbh)
208 kalimantan$w.forda<-forda.d(kalimantan$dbh)
209 kalimantan$w.jaya<-jaya.d(kalimantan$dbh)
210 kalimantan$w.novita<-novita.d(kalimantan$dbh)
211 kalimantan$w.nugroho.d<-nugroho.d(kalimantan$dbh)
212 kalimantan$w.nugroho.d.h<
213
214 plot(kalimantan$dbh, kalimantan$w.brown, col="brown", xlab="DBH", ylab="Biomass",
215 points(kalimantan$dbh, kalimantan$w.yamakura, col="brown", xlab="DBH", ylab="Biomass",
216 points(kalimantan$dbh, kalimantan$w.basuki, col="brown", xlab="DBH", ylab="Biomass",
217 points(kalimantan$dbh, kalimantan$w.samalca, col="brown", xlab="DBH", ylab="Biomass",
218 points(kalimantan$dbh, kalimantan$w.hashimoto, col="brown", xlab="DBH", ylab="Biomass",
219 points(kalimantan$dbh, kalimantan$w.kenzo, col="brown", xlab="DBH", ylab="Biomass",
220 points(kalimantan$dbh, kalimantan$w.forda, col="brown", xlab="DBH", ylab="Biomass",
221 points(kalimantan$dbh, kalimantan$w.jaya, col="brown", xlab="DBH", ylab="Biomass",
222 points(kalimantan$dbh, kalimantan$w.novita, col="brown", xlab="DBH", ylab="Biomass",
223 points(kalimantan$dbh, kalimantan$w.nugroho.d, col="brown", xlab="DBH", ylab="Biomass",
224 points(kalimantan$dbh, kalimantan$w.nugroho.d.h, col="brown", xlab="DBH", ylab="Biomass",
225
226 legend(10,8000, c("Brown", "Yamakura", "Basuki", "Samalca", "Hashimoto", "Kenzo", "Forda", "Jaya",
227
228 # Summing all values per plot and nested plot
229 bio.plot.brown<-as.data.frame(tapply(kalimantan$w.brown, list(kalimantan$plot_id, kalimantan$subplot_id),
230
231
```

R console

```
> kal.plot<-merge(kal.plot, Dmed.Hmed.plot, by="Plot")
>
> # calculating the
> kal.plot$dg<-sqrt((4*kal.plot$w.brown))
>
> write.csv(kal.plot, "Kalimantan_Biomass.csv")
>
```

Graphical output

Biomass estimation per plot with different models

Biomass (Mg/ha)

The box plot shows the distribution of biomass (Mg/ha) for different models. The y-axis ranges from 100 to 500. The x-axis shows different models. The plot indicates that the biomass values are generally higher for the 'Kenzo' and 'Forda' models compared to the others.

RStudio avec
R



Avantages/Inconvénients

```
60
61 ~~~{r}
62 y = readline(prompt = "Enter the exact name of the response variable :")
63 while (!(y %in% title))
64 {
65   y <- readline(prompt = "This variable doesn't exist! Please make sure to enter the exact name: ")
66 }
67 print(y)
68 ~~~

69
70 ~~~{r}

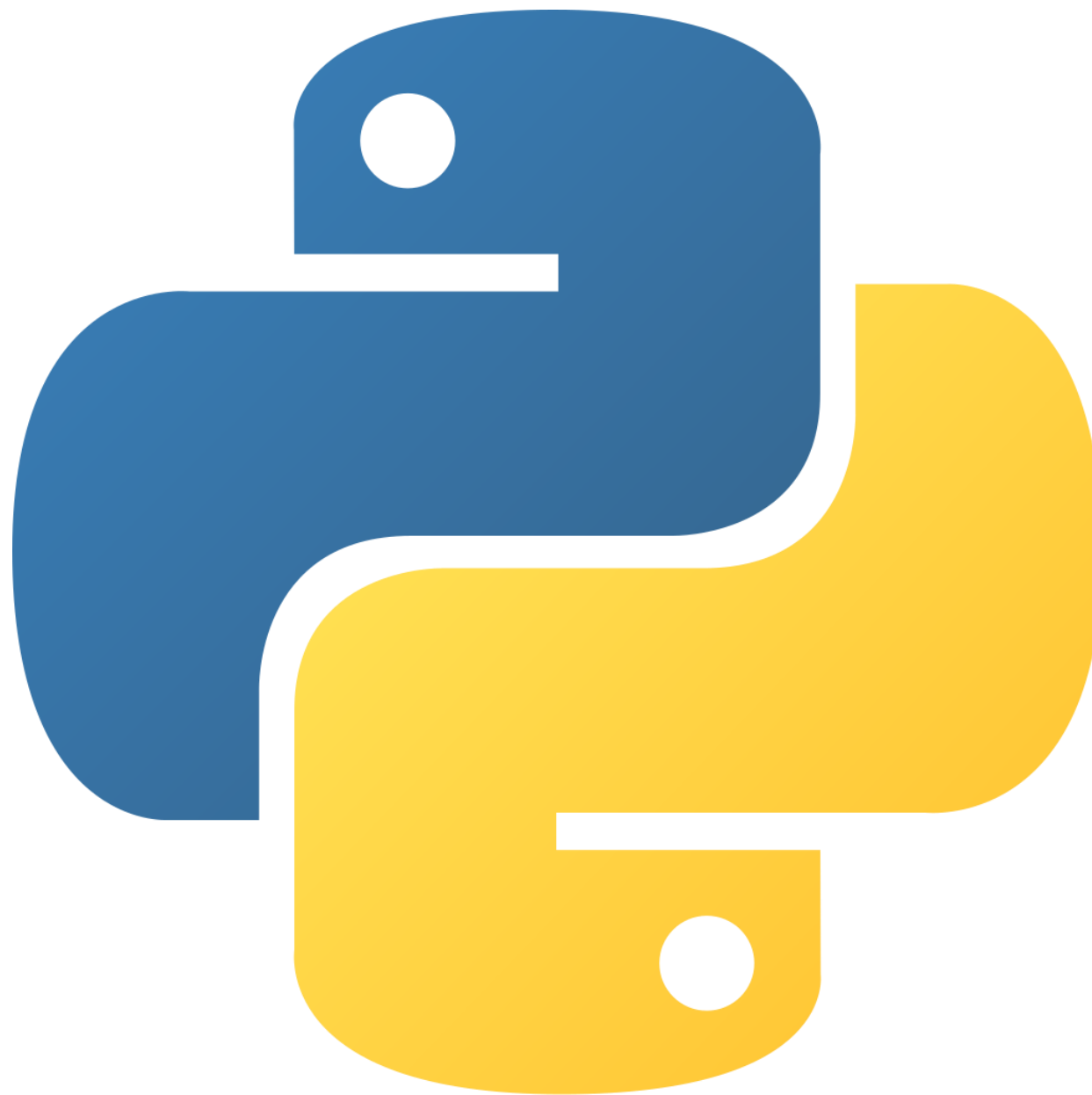
63:1 | [R] Chunk 7 | R Markdown
```

Console Terminal x

C:/Users/antoi/Desktop/Projet_Big_Data/R_BD/ ↗

Enter the exact name of the response variable :|

Rstudio avec Python



Inconvénient des inputs

```
'''{python}
input2 = input("Indiquez la manière dont sont séparées vos variables (tabulation/espace/;) : ")
'''
```

Indiquez la manière dont sont séparées vos variables (tabulation/espace/;) :



```
44 - '''{python}|
45
46 # Importation du fichier
47 file = open("train.txt", "r")
48 dataset = pd.read_csv(file, sep=" ")
49
50 # Récupération du noms des différentes variables
51 title = list(dataset.columns)
52
53 # variable d'intérêt
54 reponse = 'y'
55
```


Autre inconvénient

```
```{python}  
x = 5
print(x)|
```
```

5

```
```{python}  
print(x)
```
```

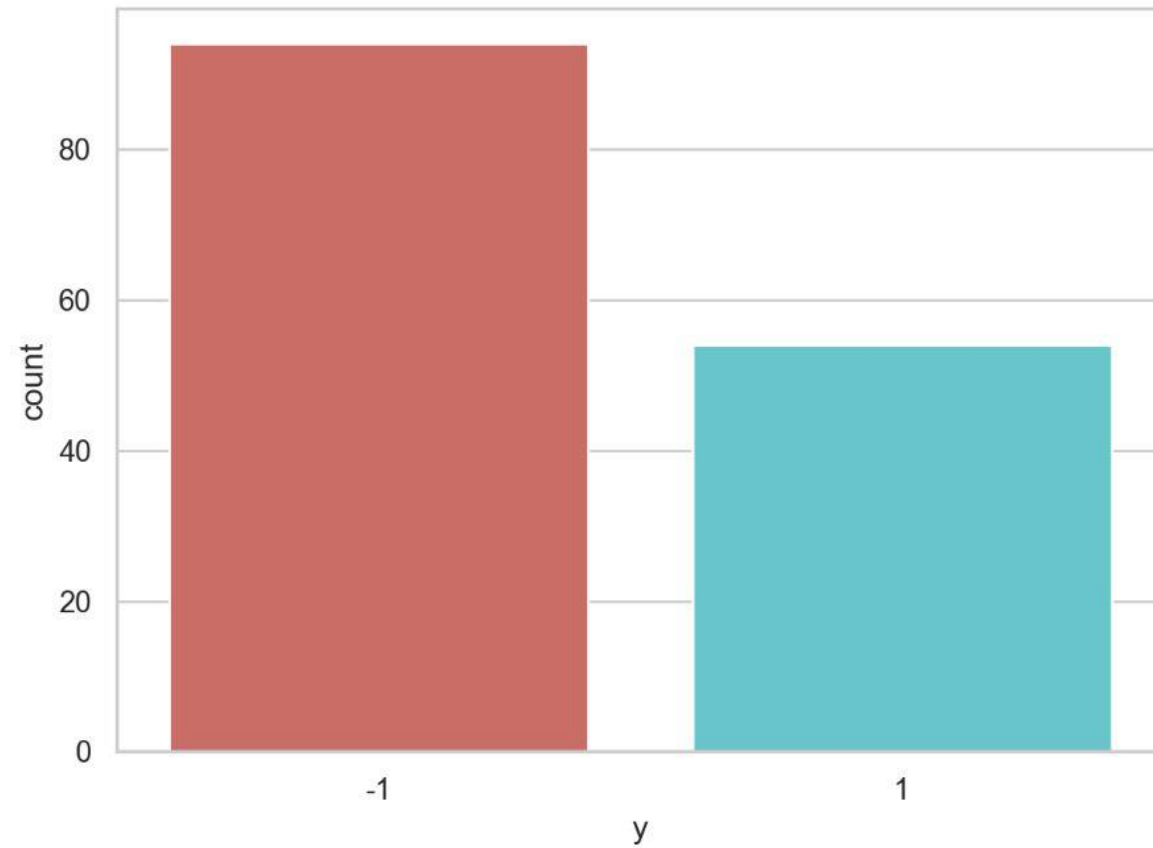
```
Traceback (most recent call last):  
  File "C:\Users\marin\AppData\Local\Temp\RtmpI1mbbh\chunk-code-3bd07a3d62d.txt", line 1, in <module>  
    print(x)  
NameError: name 'x' is not defined
```

« Solution »

```
print("Nombre de classes : ", nb_classe)
```

```
## Nombre de classes : 2
```

```
sns.countplot(x = reponse, data = dataset, palette = 'hls')  
plt.show()
```

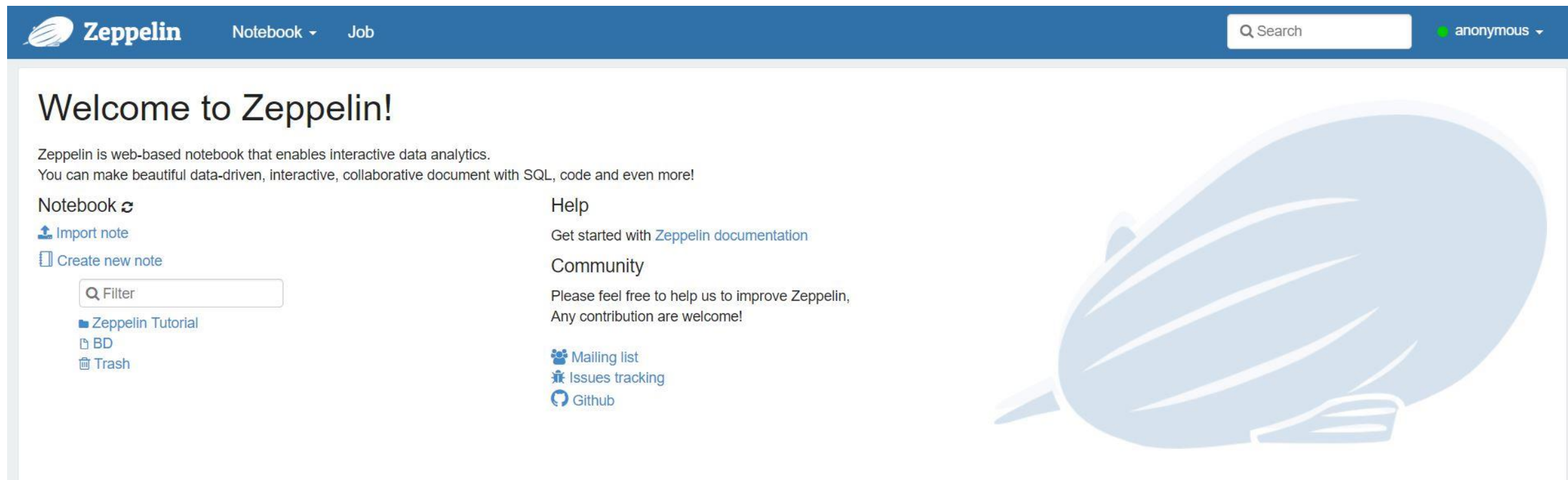




Apache Zeppelin

Présentation générale

Interface



Exemple

Zeppelin Notebook Job Search anonymous

BD [Icons] Head [Icons]

```
nb_classe = len(list(set(y))) # Taille de la liste contenant la variable d'intérêt sans doublon
```

Interpréteur utilisé

```
%python  
sns.countplot(x = reponse, data = dataset, palette = 'hls')  
plt.show()
```

code

Exécution de la cellule

FINISHED [Icons]

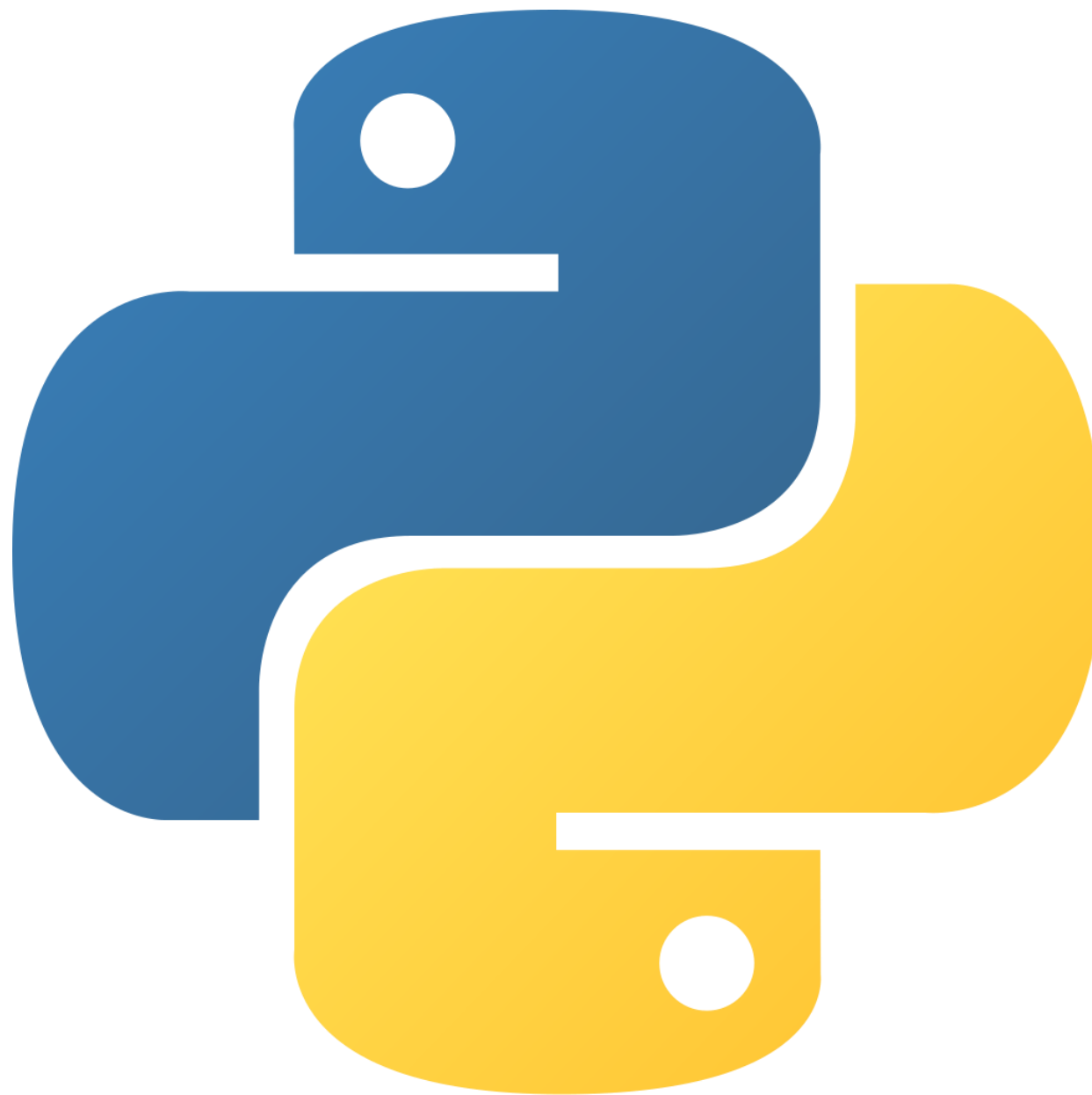
sorties



| Category | Count |
|----------|-------|
| -1 | 95 |
| 1 | 55 |

Took 0 sec. Last updated by anonymous at January 09 2019, 6:50:13 PM.

Zeppelin avec Python



Inconvénient des inputs

```
%python
input1 = z.input("Indiquez le chemin de votre fichier d'entrée : ")
#/home/tp-home008/mfigaro/Downloads/train.txt
```

Indiquez le chemin de votre fichier d'entrée :

Took 0 sec. Last updated by anonymous at January 09 2019, 2:53:27 PM.

```
%python
input2 = z.input("Indiquez la manière dont sont séparées vos variables (tabulation/espace/;) : ") # ici espace
# Passage en minuscule si l'utilisateur écrit en majuscule
input2 = input2.lower() ;
```

Indiquez la manière dont sont séparées vos variables (tabulation/espace/;) :

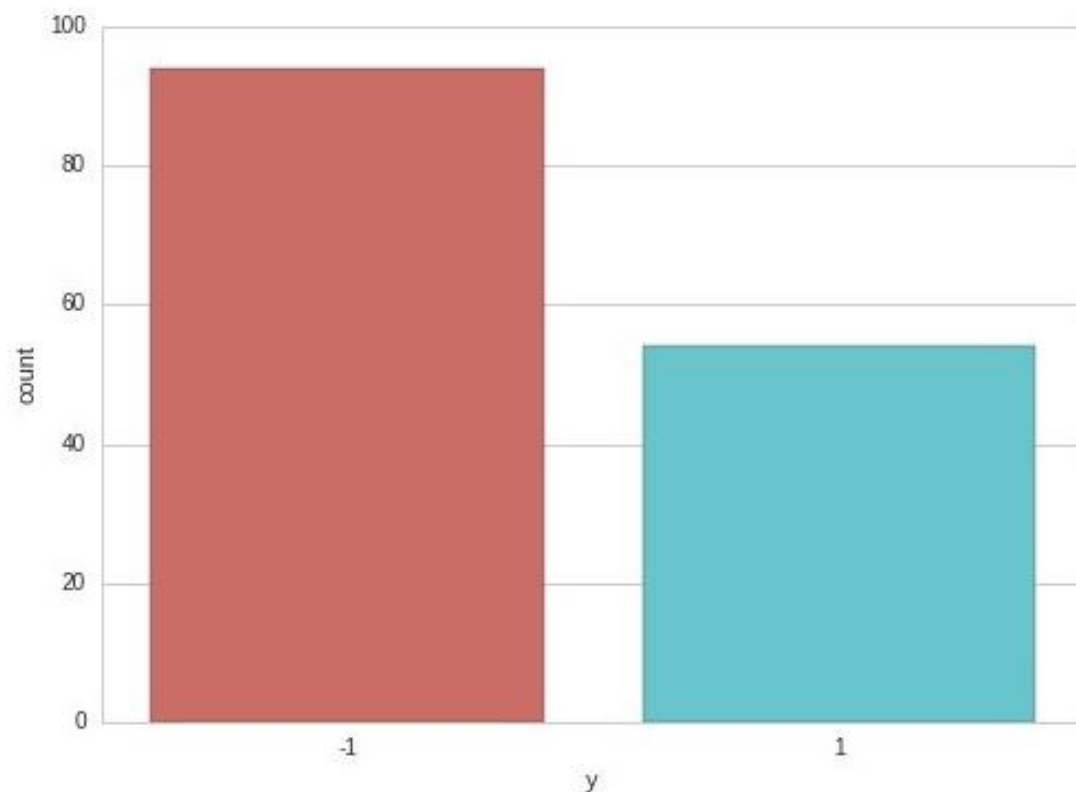
Took 0 sec. Last updated by anonymous at January 09 2019, 2:53:27 PM.

```
%python
y.describe()
classe = list(set(y))
nb_classe = len(list(set(y))) # Taille de la liste contenant la variable d'intérêt sans doublon

print "Differentes classes de la variable d'interet : ", classe
print "Nombre de classes : ", nb_classe
```

Differentes classes de la variable d'interet : [1, -1]
 Nombre de classes : 2

```
%python
sns.countplot(x = reponse, data = dataset, palette = 'hls')
plt.show()
```



```
%python
dataset.head()
```

| | x10006_at | x10007_at | x100129361_at | x100130449_at | x100130633_at | \ |
|---|-----------|-----------|---------------|---------------|---------------|---|
| 0 | 8.44 | 7.84 | 9.23 | 2.20 | 4.93 | |
| 1 | 7.93 | 8.47 | 7.77 | 2.48 | 6.70 | |
| 2 | 7.53 | 7.49 | 9.49 | 5.66 | 5.99 | |
| 3 | 7.72 | 8.38 | 8.03 | 5.22 | 6.95 | |
| 4 | 7.89 | 7.75 | 6.99 | 2.20 | 7.06 | |

| | x100133941_at | x10014_at | x10016_at | x100188893_at | x10019_at ... | \ |
|---|---------------|-----------|-----------|---------------|---------------|---|
| 0 | 10.31 | 7.85 | 9.91 | 11.37 | 3.48 ... | |
| 1 | 11.03 | 3.33 | 9.53 | 10.91 | 4.28 ... | |
| 2 | 8.87 | 3.48 | 8.02 | 10.73 | 4.07 ... | |
| 3 | 2.73 | 4.06 | 9.26 | 10.90 | 7.40 ... | |
| 4 | 11.38 | 3.65 | 9.22 | 10.37 | 3.89 ... | |

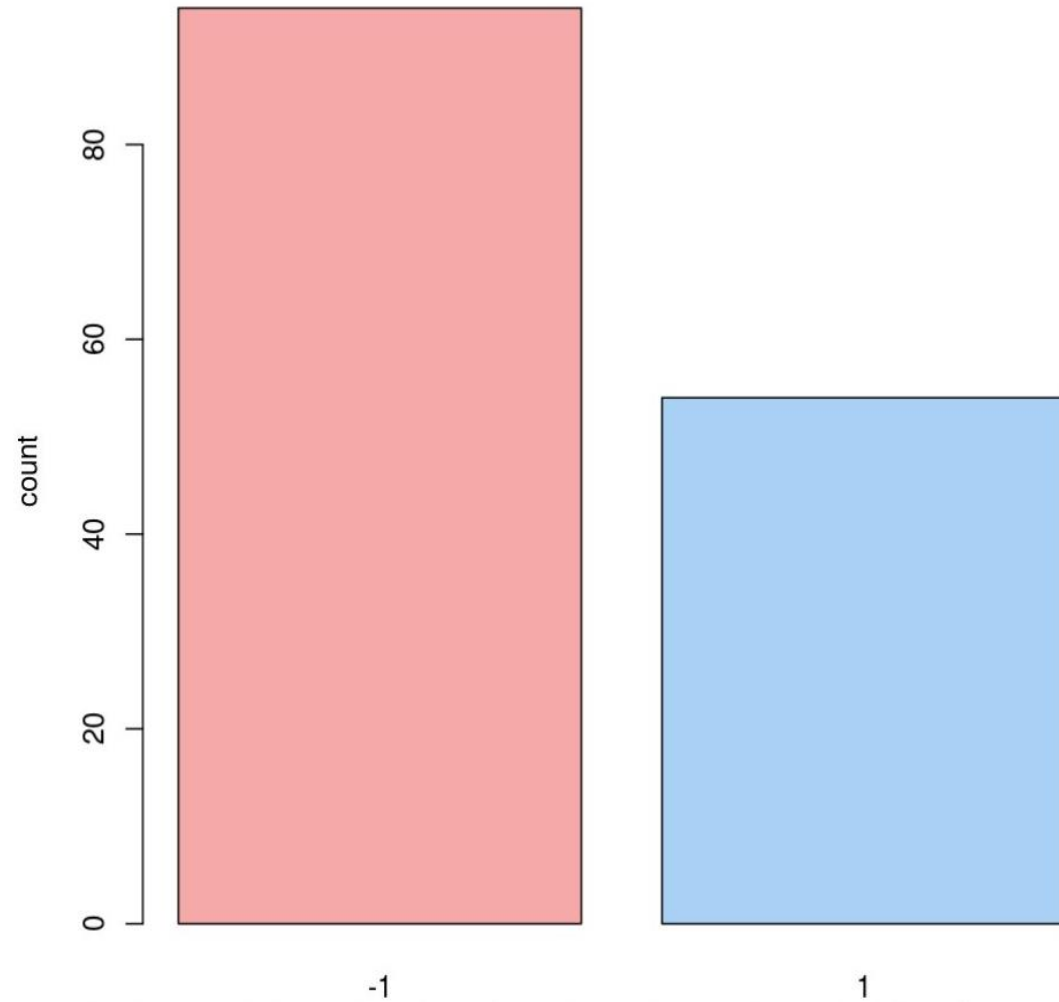
| | x10226_at | x10227_at | x1022_at | x10231_at | x10234_at | x10236_at | x10237_at | \ |
|---|-----------|-----------|----------|-----------|-----------|-----------|-----------|---|
| 0 | 10.55 | 7.82 | 7.87 | 2.97 | 4.48 | 9.61 | 12.82 | |
| 1 | 9.00 | 7.88 | 7.38 | 3.31 | 6.69 | 9.50 | 7.99 | |
| 2 | 8.64 | 8.33 | 6.74 | 3.33 | 3.33 | 8.33 | 8.33 | |

Zeppelin avec
R



```
%spark.r
tbl <- with(data, table(y))
barplot(tbl, beside = TRUE, col = c("#F5A9A9", "#A9D0F5"), xlab = y, ylab = "count")
```

FINISHED



```
%spark.r
head(data)
summary(data)
```

| | x10006_at | x10007_at | x100129361_at | x100130449_at | x100130633_at |
|---|-----------|-----------|---------------|---------------|---------------|
| 1 | 8.44 | 7.84 | 9.23 | 2.20 | 4.93 |
| 2 | 7.93 | 8.47 | 7.77 | 2.48 | 6.70 |
| 3 | 7.53 | 7.49 | 9.49 | 5.66 | 5.99 |
| 4 | 7.72 | 8.38 | 8.03 | 5.22 | 6.95 |
| 5 | 7.89 | 7.75 | 6.99 | 2.20 | 7.06 |
| 6 | 8.37 | 8.43 | 8.32 | 2.20 | 7.02 |

| | x100133941_at | x10014_at | x10016_at | x100188893_at | x10019_at | x1001_at |
|---|---------------|-----------|-----------|---------------|-----------|----------|
| 1 | 10.31 | 7.85 | 9.91 | 11.37 | 3.48 | 4.89 |
| 2 | 11.03 | 3.33 | 9.53 | 10.91 | 4.28 | 5.81 |
| 3 | 8.87 | 3.48 | 8.02 | 10.73 | 4.07 | 5.21 |
| 4 | 2.73 | 4.06 | 9.26 | 10.90 | 7.40 | 3.27 |
| 5 | 11.38 | 3.65 | 9.22 | 10.37 | 3.89 | 7.18 |
| 6 | 10.70 | 3.17 | 7.73 | 11.18 | 4.41 | 8.40 |

| | x10020_at | x10026_at | x100272147_at | x100287025_at | x100287552_at |
|---|-----------|-----------|---------------|---------------|---------------|
| 1 | 5.61 | 5.61 | 8.64 | 6.15 | 5.71 |
| 2 | 4.87 | 6.83 | 7.84 | 5.43 | 7.86 |

Conclusion

| | Jupyter | RStudio | Zeppelin |
|---------------------------|---------|---------------|-----------|
| Type d'application | Web | Web, Logiciel | Web |
| Nombre langages supportés | +++ | + | ++ |
| Installation | Facile | Facile | Difficile |
| Prise en main | Facile | Facile | Facile |
| Documentation | +++ | + | - |
| Support | +++ | ++ | + |
| Format d'export | ++ | + | - |