

Data Science Hacking Basics

avec Linux, R, et Chrome

Master IM, Paris 5

@comeetie

Trouver et enrichir des données,

Scrapping, API et Base de données en ligne

Master IM, Paris 5

@comeetie

Où trouver des données sur le web

- instituts publics : [insee](#), [ign](#), ...
- portails open-data : [data.iledefrance.fr](#), [data.gouv.fr](#), ...
- sites collaboratifs : [wikipedia](#) ([dbpedia](#)), [openstreetmap](#), ...
- sites spécialisés : [météo](#), [sports](#), [logement](#), [annonces](#), ...
- réseaux sociaux : [twitter](#), [flickrR](#), [foursquare](#), ...
- moteur de recherche : [google](#), [yahoo](#), [bing](#), ...
- api spécialisées : [velib](#), ...

Où trouver des données sur le web

Jeux de données, déjà mis en forme

- instituts publics : [insee](#), [ign](#), ...
- portails open-data : [data.iledefrance.fr](#), [data.gouv.fr](#), ...
- sites collaboratifs : [wikipedia](#) ([dbpedia](#)), [openstreetmap](#)

Où trouver des données sur le web

Jeux de données à mettre en forme

Scrapping

- sites spécialisés : météo, sports, logement, annonces, ...

API

- sites collaboratifs : openstreetmap, ...
- réseaux sociaux : twitter, flickR, foursquare, ...
- moteur de recherche : google, yahoo, bing, ...
- api spécialisées : velib, ...

Scrapping

Extraire des informations spécifiques

d'une ou plusieurs pages web

en vu de constituer un jeu de données

Scrapping, le html

```
<!DOCTYPE html>
<head><meta charset="utf-8"></head>
<body>
<section style="padding-top:6em;text-align:center">
<h1 class="purple"> Scrapping </h1>
<h4 class="purple">Extraire des informations spécifiques</h4>
<h4 class="purple">d'une ou plusieurs pages web</h4>
<h4 class="purple">en vu de constituer un jeu de données</h4>
</section>
</body>
</html>
```

Scrapping, les package RCurl et XML

RCurl (Client URL Request Library)

le web en ligne de commande : **get**, **post**, **https**, **ftp**, ...

```
library(RCurl)
# récupérer la page
res = getURL("http://www.leboncoin.fr/jardinage/offres/centre/")
```

XML

htmlTreeParse, getNodeSet :

```
# parse du html
resp = htmlTreeParse(res,useInternal=T)
# fonction de haut niveau pour récupérer les tableaux
rest = readHTMLTable(resp)
# récupérer un noeud désiré (xpath)
node = getNodeSet(resp, '//nav/ul/')
```


Scrapping, les package RCurl et XML

Xpath, extraire des informations d'un arbre DOM

Syntaxe pour se promener dans l'arbre dom et en extraire des partie (noeuds, attributs, ...), plus détails sur [w3schools](https://www.w3schools.com/xpath/).

Expression	Description
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

Scrapping, les package RCurl et XML

Xpath, extraire des informations d'un arbre DOM

Syntaxe pour se promener dans l'arbre dom et en extraire des partie (noeuds, attributs, ...), plus détails sur [w3schools](#).

Expression	Description
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element.
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='en']	Selects all the title elements that have an attribute named lang with a value of 'en'
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00

Scraper leboncoin.fr

Ecrire un script R permettant de scraper le nombre d'annonce du site dans la catégorie "Jardinage" en région centre.

The screenshot shows the leboncoin.fr website interface. At the top, there's a navigation bar with the leboncoin.fr logo and the text 'JARDINAGE CENTRE OFFRES'. Below this, there's a search bar with the following elements:

- Search input field: []
- Category dropdown: Jardinage
- Region dropdown: Centre
- Postcode input field: Villes ou codes postaux
- Search button: CHERCHER

Below the search bar, there are filters and a 'Prix' (Price) section with 'Entre' (Between) and 'Prix min' (Minimum price) and 'Prix max' (Maximum price) dropdowns. There's also a checkbox for 'Recherche dans le titre uniquement' (Search only in title) and a checkbox for 'Annonces Urgentes uniquement' (Urgent ads only).

Below the search bar, there's a banner for Cofidis with the text 'Petite mensualité, budget rentrée respecté !' and a 'SERVICE CLIENT 2014' badge.

Below the banner, there's a section for 'PARTICULIERS' (Private) and 'PROFESSIONNELS' (Professional) with the number of ads: 9 231 annonces and 1 654 annonces respectively. There's also a 'Cacher les photos' (Hide photos) button and a 'Trier par prix' (Sort by price) button.

The main content area shows a list of results. The first result is a tractor for sale, titled 'Tracteur tondeuse: (casse) (pro)' and located in 'La Chapelle-Montmartin / Loir-et-Cher'. It has a photo of the tractor and a price of 10 885 €. The second result is a 'Un pressoir' (A press) for sale, located in 'La Chapelle-Montmartin / Loir-et-Cher'.

Scraper leboncoin.fr

```
library(RCurl)
library(XML)

# récupérer la page
res = getURL("http://www.leboncoin.fr/jardinage/offres/centre/")
# la parser
resp = htmlTreeParse(res,useInternal=T)
# récupérer le noeud désiré (xpath)
node = getNodeSet(resp, '//nav/ul/li/span/b')
# récupérer la valeur, supprimer l'espace et convertir en numérique
val = as.numeric(gsub(" ","",xmlValue(node[[1]])))
```

Scraper stackoverflow.com

Ecrire un script R permettant de scraper le nombre de question publier sur les sites ayant les tags : 'python','julia-lang','r','sas','matlab','ggplot2' et 'd3.js'. Réaliser un graphique à partir de ces données.

The screenshot displays the Stack Overflow homepage. At the top, there's a navigation bar with links for 'StackExchange', 'sign up', 'log in', 'tour', 'help', 'careers 2.0', and a search bar. Below this, the 'stackoverflow' logo is followed by tabs for 'Questions', 'Tags', 'Users', 'Badges', and 'Unanswered'. A 'Ask Question' button is also present. The 'Tagged Questions' section is active, showing a list of questions. The first question is 'Count number of same elements in the same position in two vectors' with 0 votes and 0 answers. The second is 'R program: i am not able to print "NA" if the values are not found/matching' with 0 votes and 0 answers. The third is 'R: how to replicate the <<- assignment with assign()?' with 1 answer. On the right, a sidebar titled 'Looking for a job?' lists several job openings, including 'Director, Mobile Lead Developer' at Wingit, 'Xebia', and 'Société Générale'.

StackExchange sign up log in tour help careers 2.0 [r]

stackoverflow Questions Tags Users Badges Unanswered Ask Question

Tagged Questions info newest 2 featured frequent votes active unanswered

R is a free, open-source programming language and software environment for statistical computing, bioinformatics, and graphics. Please supplement your question with a minimal reproducible example. For statistical questions please use stats.stackexchange.com.
[learn more...](#) | [top users](#) | [synonyms \(1\)](#)

0 votes 0 answers 5 views
Count number of same elements in the same position in two vectors
I have two vectors: `set.seed(12) a<-sample(c(0,1),10,replace=T) b<-sample(c(0,1),10,replace=T) > a [1] 0 1 0 0 0 1 0 0 > b [1] 0 1 0 0 0 0 1 1 0` I would like to count the ...
asked 3 mins ago user2733997 101 ● 7

0 votes 0 answers 8 views
R program: i am not able to print "NA" if the values are not found/matching
I am new to this "R", I have written a program in R which pulls the data from one flat file and matches the probe IDS (its microarray data) with another file which contains gene annotations(name, ...
asked 5 mins ago user3151268 1 ● 2

0 votes 1 answer 6 views
R: how to replicate the <<- assignment with assign()?
I need to assign a variable in a function, whose name is a parameter of the function, and I need to access it later on, outside the function. I think <<- would do it in another situation, but ...
asked 11 mins ago Peutch 183 ● 10

Looking for a job?

Director, Mobile Lead Developer (iOS, Android)
Wingit
Paris, France
[ios](#) [android](#)

expect(you).to.be.a('Xebian Front-End Developer')
Xebia
Paris, France
[javascript](#) [angularjs](#)

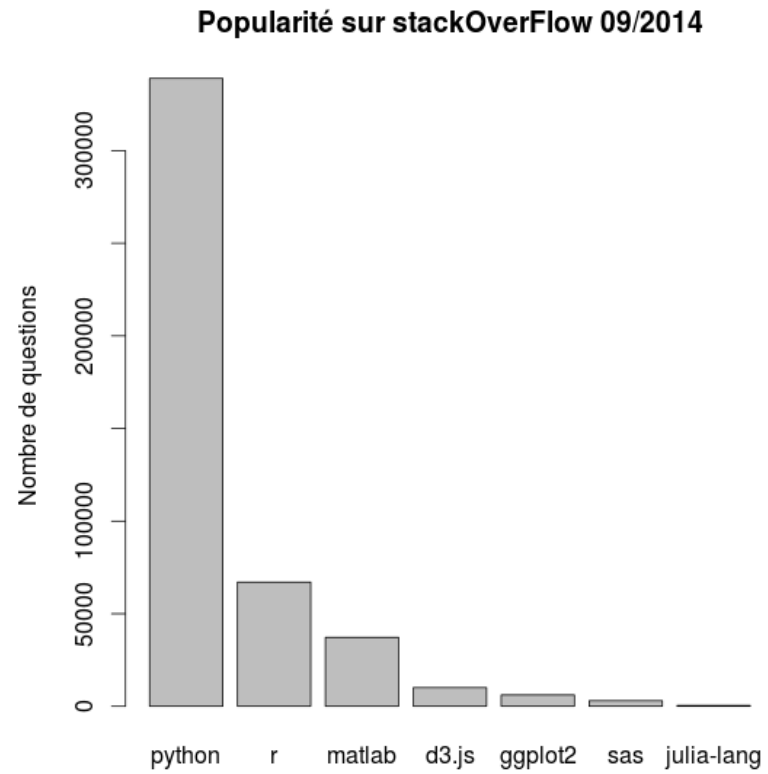
Leader technique JAVA
Société Générale
La Defense, France
[java](#) [xml](#)

Tech Lead Cloud Java H/F
SFEIR
Hauts-de-Seine, France
[java-ee](#) [google-app-engine](#)

Scraper stackoverflow.com

```
# definition des termes à scraper
languages=c('python','julia-lang','r','sas','matlab','ggplot2','d3.js')
# initialisation de la table
stackOF=data.frame(lang=languages,questions=NA)
# boucle sur les termes
for(i in 1:length(languages)){
  # récupérer la page
  base = "http://stackoverflow.com/questions/tagged/"
  res  = getURL(paste(base,stackOF[i,'lang'],sep=''))
  # la parser et récupérer le noeud désiré (xpath)
  resp = htmlTreeParse(res,useInternal=T)
  ns1  = getNodeSet(resp, "//div[@class='summarycount al']")
  # récupérer la valeur, supprimer la virgule et convertir en numérique
  stackOF[i,'questions'] = as.numeric(gsub(",","",xmlValue(ns1[[1]])))
}
# faire un graphique
stackOF=stackOF[order(stackOF$questions,decreasing=T),]
title="Popularité sur stackOverFlow 09/2014"
barplot(stackOF$questions,names.arg=stackOF$lang,main=title,ylab="Nombre de questions")
```

Scraper stackoverflow.com



Scraper les résultats de ligue 1

sur footballstats.fr

Récupérer les dix dernières années de résultats du championnat de france

Scraper les résultats de ligue 1

sur footballstats.fr

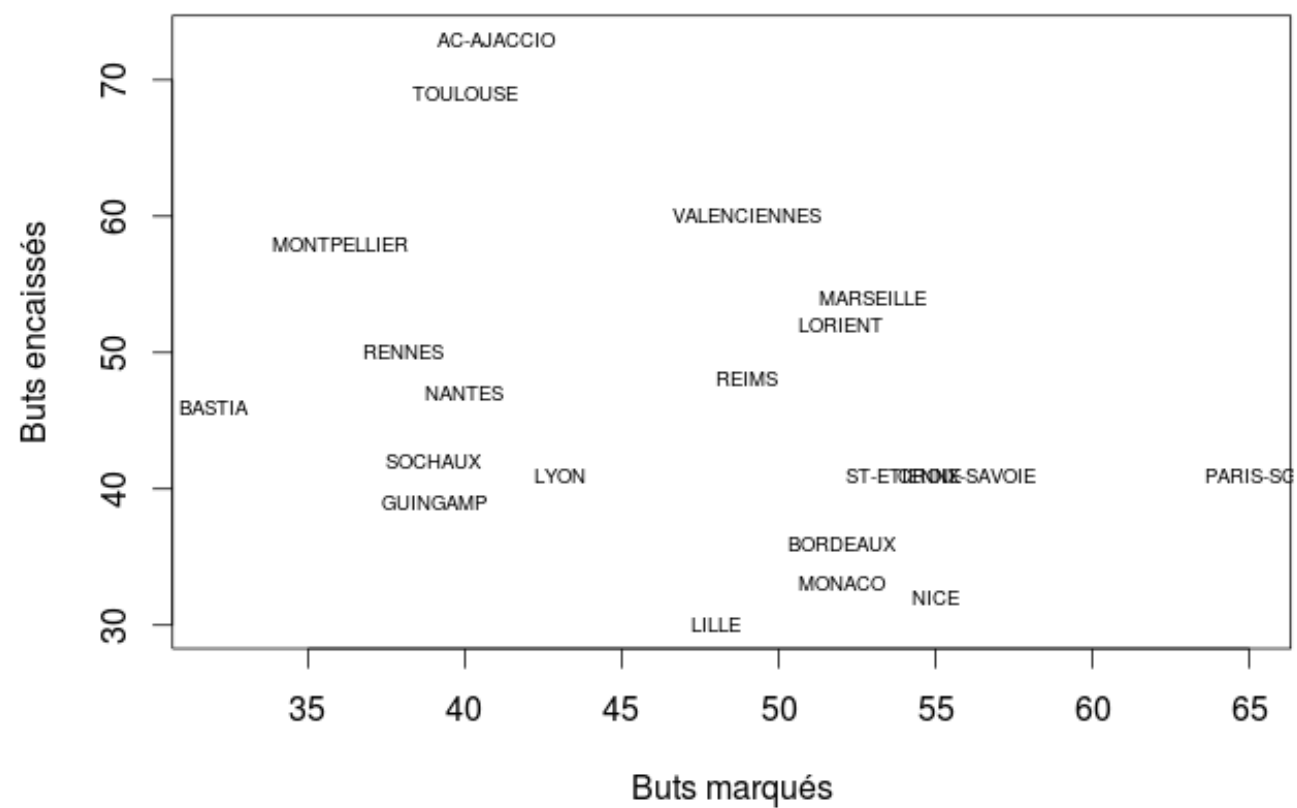
```
# récupérer la page et la parser
year = 2014
res = getURL(paste("http://www.footballstats.fr/resultat-ligue1-", year, ".html", sep=''))
resp = htmlTreeParse(res, useInternal=T)
# récupérer le bon tableau de la page
rest = readHTMLTable(resp)[[2]]
# le remettre légèrement en forme
rest = rest[!is.na(rest[,2]), 1:3]
names(rest) = c('locaux', 'visiteur', 'resultat')
rest$locaux = factor(as.character(rest$locaux), levels=unique(rest$locaux))
rest$visiteurs = factor(as.character(rest$visiteur), levels=unique(rest$locaux))
resm = matrix(unlist(strsplit(as.character(rest$resultat), '-')), 2)
rest$resultat.locaux = as.numeric(resm[1,])
rest$resultat.visiteurs = as.numeric(resm[2,])
```

Scraper les résultats de ligue 1

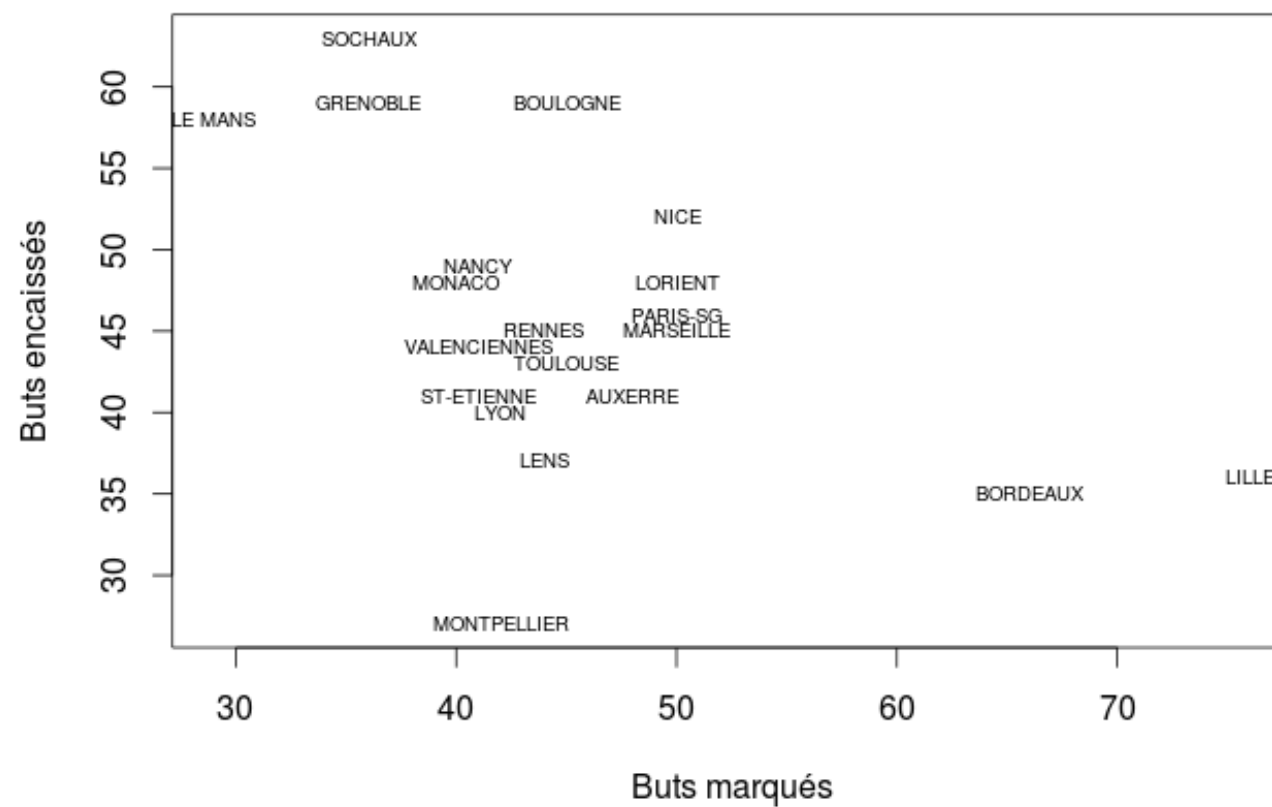
sur footballstats.fr

```
# récupérer la page et la parser
# calcul des totaux de buts marqués / encaissés
Abutadomicile = by(rest$resultat.locaux,rest$locaux,sum)
Abutalexterieur = by(rest$resultat.visiteur,rest$visiteur,sum)
Abut = Abutadomicile+Abutalexterieur
Dbutadomicile = by(rest$resultat.visiteur,rest$locaux,sum)
Dbutalexterieur = by(rest$resultat.locaux,rest$visiteur,sum)
Dbut = Dbutadomicile+Dbutalexterieur
# faire un graphique
ti = paste("Ligue 1, Saison",year)
xl = "Buts marqués"
yl = "Buts encaissés"
plot(Abut,Dbut,xlab=xl,ylab=yl,col="white",main=ti)
text(Abut,Dbut,levels(rest$locaux),cex=0.6)
```

Ligue 1, Saison 2014



Ligue 1, Saison 2010



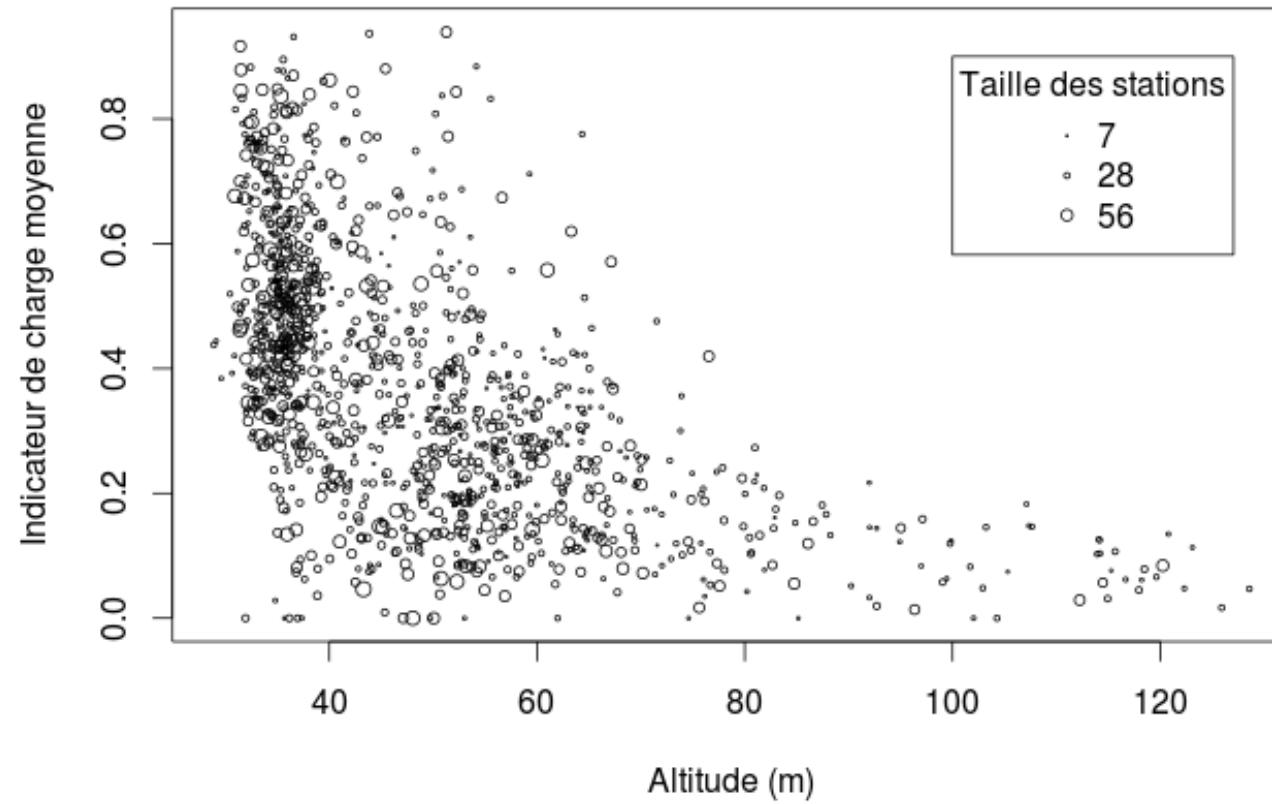
API

Application Programming Interface

Vélib' et altitude des stations

Utiliser les fichiers http://vlsstats.ifsttar.fr/data/input_Paris.json et http://vlsstats.ifsttar.fr/data/spatiotemporalstats_Paris.json ainsi que l'api google maps pour calculer un indicateur de charge moyenne des stations Vélib' et mettre celui-ci en relation avec l'altitude des stations.

Effet de l'altitude sur la charge des stations



Vélib' et altitude des stations

```
# récupérer la liste des stations et la mettre en forme
stationsList=fromJSON(file="http://vlsstats.ifsttar.fr/data/input_Paris.json")
data=sapply(stationsList,function(x){
  c(x$number,x$name,x$address,x$bike_stands,x$position$lat,x$position$lng)
})
stations=data.frame(id=data[1,],name=data[2,],adresse=data[3,],
nbdocks=as.numeric(data[4,]),lat=as.numeric(data[5,]),long=as.numeric(data[6,]),alt=NA)
```


Vélib' et altitude des stations

API google maps

```
# récupérer les altitudes
for (i in 1:ceiling(dim(stations)[1]/50)){
  system("sleep 0.5")
  print(i)
  ind  = ((i-1)*50):min((i*50),dim(stations)[1])
  query = paste(stations[ind,'lat'],stations[ind,'long'],sep=',',collapse='|')
  base  = "https://maps.googleapis.com/maps/api/elevation/json?locations="
  url   = paste(base,query,sep="")
  res   = fromJSON(getURL(url))
  stations$alt[ind]=unlist(lapply(res$results,function(x){x$elevation}))
}
```

Vélib' et altitude des stations

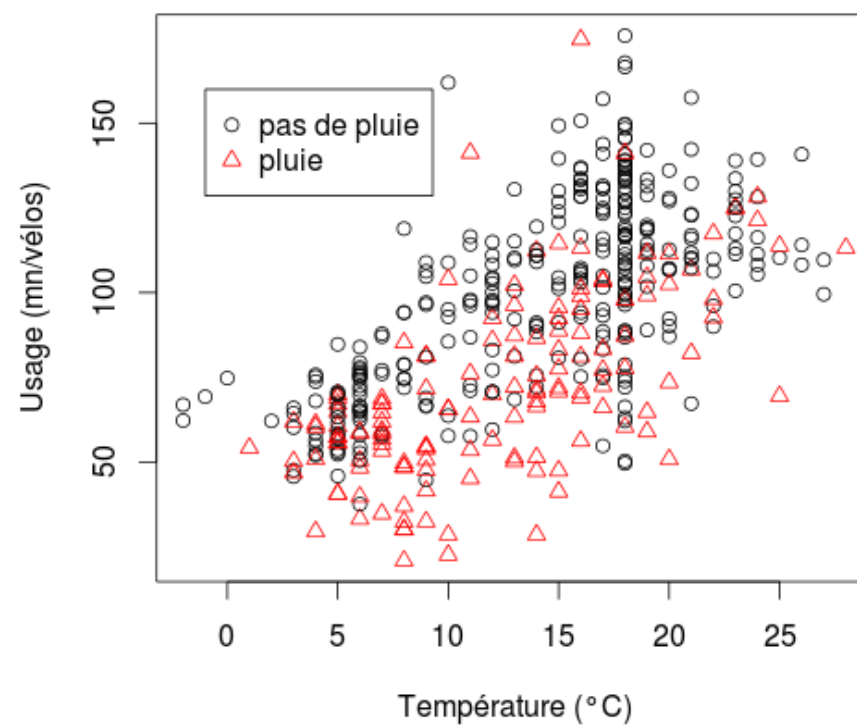
API google maps

```
# calculer l'indice de charge moyenne
url = "http://vlsstats.ifsttar.fr/data/spatiotemporalstats_Paris.json"
stationsData = fromJSON(file=url)
res = sapply(stationsData,function(x){c(x$'_id', mean(x$available_bikes))})
res = data.frame(t(res))
names(res) = c('id','mnbikes')
stations = merge(stations,res,by="id")
stations$loading = stations$mnbikes/stations$nb docks
ti = "Effet de l'altitude sur la charge des stations"
yl = "Indicateur de charge moyenne"
xl = "Altitude (m)"
plot(stations$alt,stations$loading,xlab=xl,ylab=yl,main=ti)
```

Vélib' et météo

Utiliser le fichiers http://vlsstats.ifsttar.fr/data/daystats_Paris.json et des données météo pour enrichir ces données d'informations sur la météo. Vous croiserez en particulier l'usage du service et la température.

Effet de la température sur l'usage



Vélib' et météo

```
# mettre en forme les données vélib  
daysData=fromJSON(file="http://vlsstats.ifsttar.fr/data/daystats_Paris.json")  
daysData=sapply(daysData,function(x){c(x$'_id',x$value$totaltime_used_bikes,x$value$max_available_bikes)  
daysData=data.frame(id=as.character(daysData[1,]),timeuse=as.numeric(daysData[2,]),nbikes=as.numeric(day
```



Vélib' et météo

```
# mettre en forme les données météo
base = "http://www.wunderground.com/weatherstation/WXDailyHistory.asp"
meteo2013=getURL(paste(base,"?ID=IILEDEFR16&year=2013&graphspan=year&format=1",sep=""))
meteo2014=getURL(paste(base,"?ID=IILEDEFR16&year=2014&graphspan=year&format=1",sep=""))
meteo=gsub("\n>br<", "", paste(meteo2013,meteo2014, sep="\n"))
meteo=read.table(text=meteo, sep=',', header=T, stringsAsFactors=F)

# créer une colonne pour faire la jointure
meteo$id=unlist(lapply(strsplit(meteo$Date, '-'), function(x){paste(x[3:1], collapse='/')}))
data = merge(daysData, meteo, by='id')
temp = as.numeric(data$TemperatureAvgC)
usage = data$timeuse/data$nbikes
pluie = as.numeric(data$PrecipitationSumCM.br.)>0
# visualiser
p = as.numeric(pluie)+1
xl='Température (°C)'
yl='Usage (mn/vélos)'
ti='Effet de la température sur l\'usage'
plot(temp, usage, col=c('black', 'red')[p], pch=c(1,2)[p], xlab=xl, ylab=yl, main=ti)
legend(-1, 160, legend=c('pas de pluie', 'pluie'), pch=c(1,2), col=c("black", "red"))
```

Mettez en forme un jeu de données sur les monuments historiques d'Indre et Loire

Celui-ci devra contenir tant que faire ce peut des informations sur leurs localisation (altitude/longitude) et une description iconographique.

Monuments historiques

```
# exo monuments historiques
library(rjson)
monum=read.table("exo5.csv",sep="\t",header=T,quote="",stringsAsFactors=F)
monum=monum[monum$DPT==37,]
monum$lat=rep(NA,dim(monum)[1])
monum$long=rep(NA,dim(monum)[1])
monum$geoquality=rep('NA',dim(monum)[1]);
```


Monuments historiques

```
# geocodage des adresse avec nominatim
for (i in 1:dim(monum)[1]){
  if(monum$ADRS[i]!=""){
    adrs = strsplit(monum$ADRS[i],';')[[1]][1]
    dec = strsplit(adrs,'[(,)]')[[1]]
    adrs = paste(dec[length(dec):1],collapse=' ')
    query = paste(adrs,monum$COM[i],'France',sep=', ')
    monum$geoquality[i] = 2
  }else{
    query = paste(monum$COM[i],'France',sep=', ')
    monum$geoquality[i] = 1
  }
  base = 'http://nominatim.openstreetmap.org/search?q='
  query = URLencode(query)
  query = paste(base,query,'&format=json&polygon=1&addressdetails=1',sep='')
  res = fromJSON(file=query)
  if(length(res)>0){
    print(paste(i, "Geocoding OK"))
    monum$lat[i] = as.numeric(res[[1]]$lat)
    monum$long[i] = as.numeric(res[[1]]$lon)
  }
}
```

Monuments historiques

```
# jointure avec les photos récupérées sur data.gouv
photos=read.table("exo5.photos.txt",sep="\t",header=T,quote="",stringsAsFactors=F)
monum$photos=photos$VIDE0.p[match(monum$REF,photos$LBASE)];

# recherche de photos en utilisant l'api flickr
for (i in 6:dim(monum)[1]){
  base  = 'https://api.flickr.com/services/rest/?method=flickr.photos.search'
  args  = '&api_key=edd8589d760e0b1d0bd00f0ac9c2f216&safe_search=1&per_page=1&radius=2&text='
  query = paste(URLEncode(monum$TIC0[i]), '&lat=', monum$lat[i], '&lon=', monum$long[i], sep='')
  res   = getURL(paste(base, args, query, sep=''))
  resp  = xmlTreeParse(res)
  pp    = xmlChildren(xmlRoot(resp))[[1]]
  if(length(pp)>0){
    photosFlickr=xmlAttrs(pp[[1]])
    print(paste(i, "Photo trouvée"))
    photoBurl= paste('https://farm', photosFlickr[5], '.staticflickr.com/', sep='')
    photoPath= paste(photosFlickr[4], '/', photosFlickr[1], '_', photosFlickr[3], '.jpg', sep='')
    monum$photos[i]=paste(photoBurl, photoPath, sep='')
  }
}
```

Monuments historiques



