

epiconcept

smart health



plateforme R

R Statistics and Reporting module for Voozanoo: Reference Sheets

December 2019

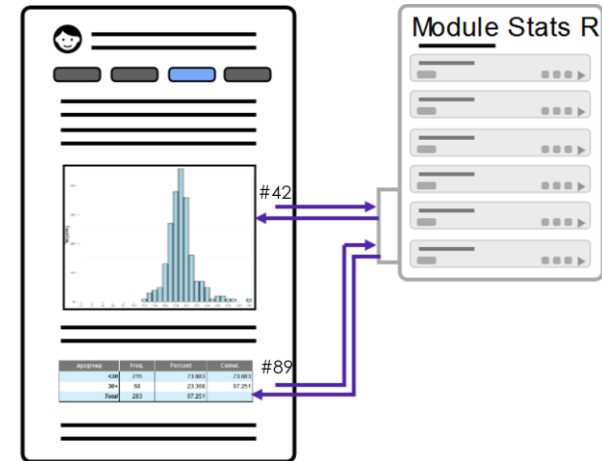
Editor:	EpiConcept, 47 rue de Charenton, 75012 Paris, France www.epiconcept.fr
Title:	R Statistics and Reporting module for Voozanoo: Reference Sheets
Current version:	Version 3: 12 December 2019
Previous versions:	Version 1.01 (19 Nov 2015) Version 1.1 (21 Dec 2015) version 2 (14 Nov 2017)
Applies to:	R Statistics and Reporting module version 19w38
License:	Copyright © EpiConcept 2019
Author(s):	Curtis Broderick
Distribution:	Public
Other languages:	Available in French
Corollary documents:	Voozanoo 4 Reference Sheets
Other information:	
Feedback:	Please signal any errors or problems with this document to c.broderick@epiconcept.fr

ABOUT THIS DOCUMENT	2
TABLE OF CONTENTS	3
OVERVIEW	4
GLOSSARY AND MARKDOWN	5
HOME SCREEN: LIST OF ANALYSIS PLANS	6
AN ANALYSIS PLAN	7
CREATING A NEW ANALYSIS PLAN (1 OF 2)	8
CREATING A NEW ANALYSIS PLAN (2 OF 2)	9
CALLING A PLAN FROM A VOO4 PROJECT: WIDGET <STAT>	10
PASSING VALUES TO YOUR ANALYSIS SCRIPT	11
DYNAMIC CARTOGRAPHIC DATA	12
ANNEX 1 INCLUDED LIBRARIES	13

This document describes the organisation and use of EpiConcept's **R Statistics and Reporting module**. This application was created with Voozanoo 4 and is supported by a dedicated server running the R language and executing the scripts. The module can be used directly as an application, but is used more typically indirectly as a web service for another Voozanoo application to run statistical analysis, reporting, and cartographic data representations. This module is not accessible to all users of a Voozanoo information system (even if that system calls upon this module to produce analyses and reports). Using this module implies that the user has privileged access to his/her Voozanoo application and that he/she is a proficient R programmer.

Conceptual Overview

A Voozanoo 4 application can show a user all of its data through listings and filters (based of course on the user's access rights). However there is no built-in system to display data graphically or to mathematical analyse the data (such as statistical analyses). The idea is that one creates R scripts in this module and then, within the user's Voozanoo application, these specific scripts are called



upon and executed by a dedicated R server. That means that the executing script must have access to the application and have a specific data request so that it can retrieve those data, do the processing and then return the required graphic, map, or report in html form.

In general, it works as follows. A given script such as #42 or

#89 have been pre-configured (when they were written) with a specific data request to the user's application and with specific R scripts (in this example, a graphic and a listing with calculations respectively). The page is configured in Epicraft with some text and buttons and then a call to analysis #42, some more text and then a call to analysis #89. The stats module, having access to the application's data, reads and processes the data, and then returns an html object which the page then displays.

Glossary

Script: Also known as an R Script. It is written in the R language and stored in the stats module.

An Analysis: Another name for a Script

Analysis Plan: A grouping of R scripts. It is an analysis plan (multiple R scripts) that is executed when called from a Voozanoo application, not just a single analysis.

Source: The data source that the R script will need in order to execute its analysis and/or graphing. Data sources may be a Voozanoo 4 application, a third-party application, or an uploaded CSV file.

Tag: any keyword/s that you want to give to an Analysis Plan to help you more easily find it among the other plans in the list.

Markdown language

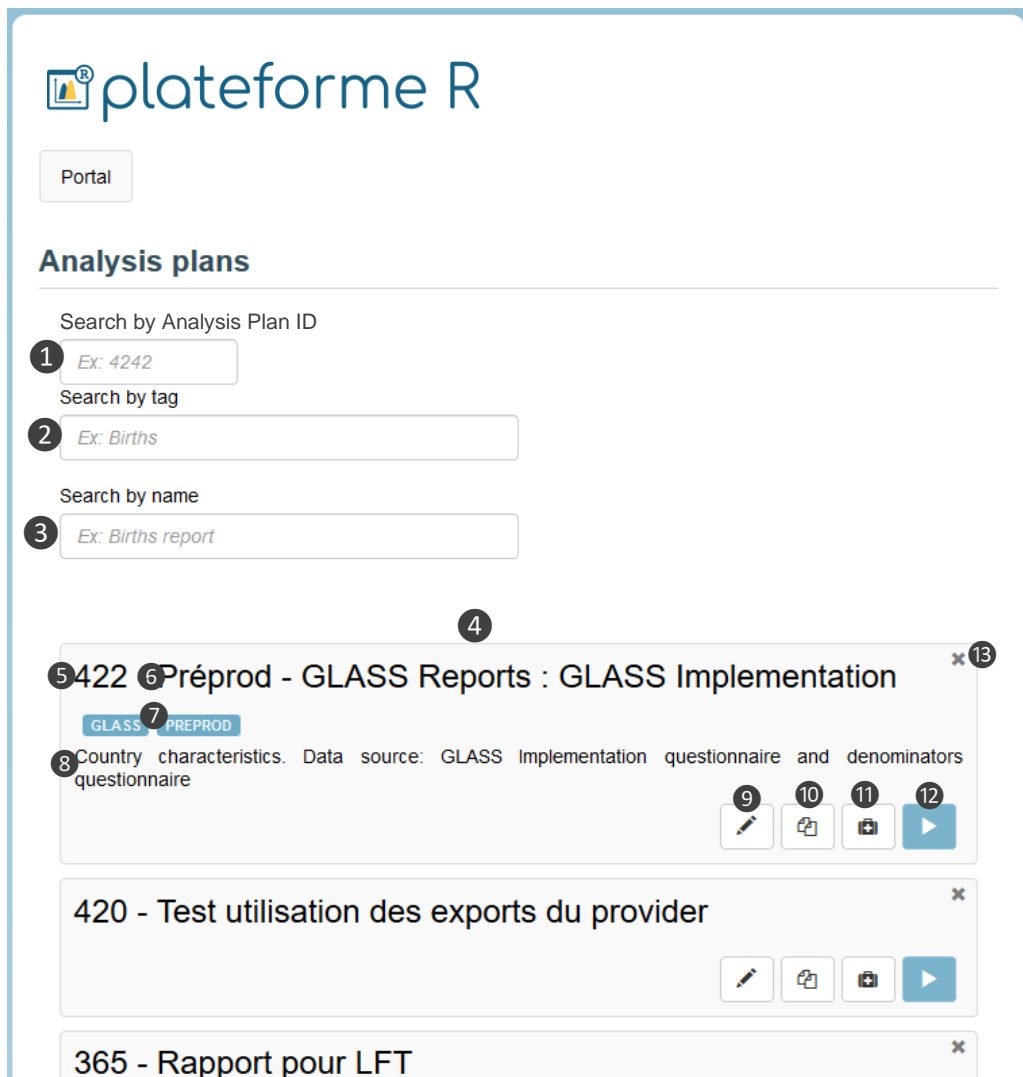
M↓ The R application which runs your scripts will interpret the markdown language. At the beginning of every script include ````{r echo=FALSE, message=FALSE}` to indicate "read the following as the R language". Simply add the line ````` (three grave accents) to indicate "read the following as Markdown text". Then to return to R interpretation, add the line ````{r message=FALSE}`.

Find the help sheet here: <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

```
```{r echo=FALSE, message=FALSE}
This line is an R comment line
Nothing will be displayed because R just ignores
comment lines
```

Now we are in Markdown mode. Did you notice the three
little grave accents above?
## When I use the hashtags now, they are interpreted as
headers H1, H2, H3 ...
```{r message=FALSE}
Now we are back to R interpretation. The hashtags are
for R comments once again.
2+3
We'll see that this expression gets evaluated and
displayed as 5.
```

When you first enter the Stats module, you are presented with a list of the analysis plans that you have the right to access (your own or your group's plans). Each of these contains one or more analyses (R scripts) inside. When a plan is called by the user's application, all of the scripts within a given plan are executed in order.



### 1. Search by Analysis ID

Search for a specific Analysis Plan ID

### 2. Search by tag

Filters the displayed list to those plans that include the entered characters within their tags.

### 3. Search by name

Filters the displayed list to those plans that include the entered characters within their names.

### 4. An analysis plan

5. The ID number that will need to be used to call it from one's Voozanoo application.

6. A good descriptive name for the plan

7. Tags to help one easily find it in the list of plans

8. A more detailed explanation of what the plan does.

9. Open a plan to view or modify it.

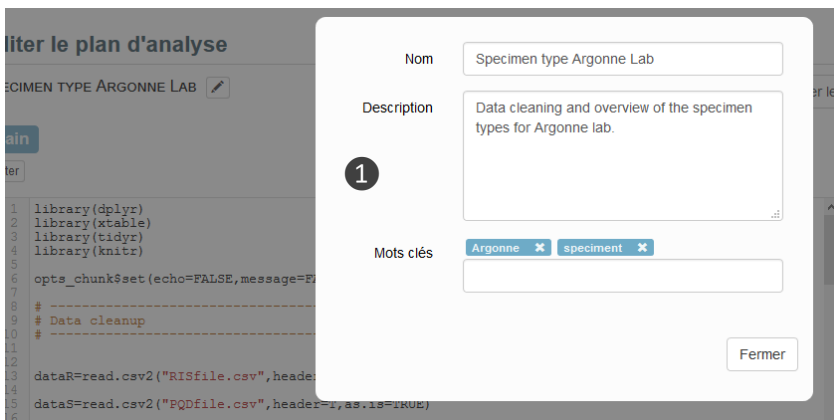
10. Make a copy of this plan.

11. View a log of activity for the plan.

12. Run this plan. It will run each of scripts in order and output an HTML file.

13. Delete this Analysis Plan and all its scripts.

Inside an analysis plan, one can add or delete data sources (existing data requests or uploaded CSV files) and manage the various R scripts that comprise the plan.

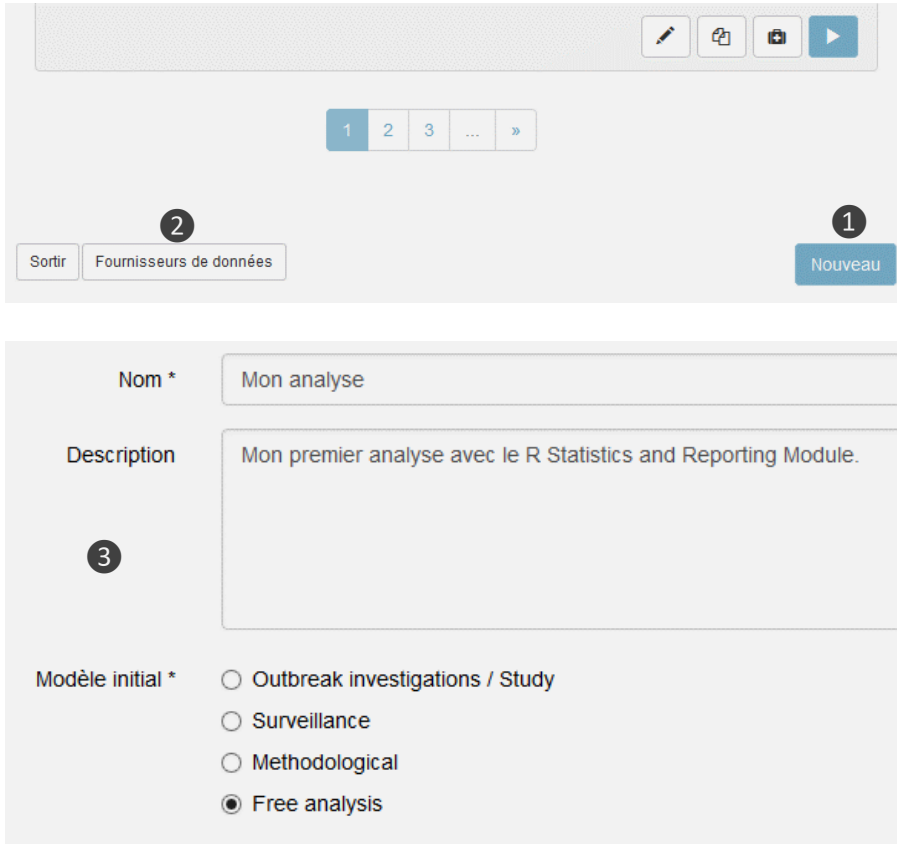


## Explanation

### Buttons

1. Modify the name, description, or tags of the plan. Tags cannot contain spaces.
2. Add, delete, or modify a data source.
3. Download all the scripts of this plan as a single text file.
4. Review any of your data sources.
  - Browse Data:** displays the data as a table.
  - Show Data Summary:** gives a statistical summary of each variable (mean, variance, number of missing, etc.)
5. **Group membership:** you can modify the group that this analysis plan belongs to.
6. **Analysis R script:** the script written in the R language that is to be executed.
7. **Analysis list:** list of the individual analyses (R scripts) which comprise the plan. When the ID of this plan is called from the user's Voozanoo 4 application, the scripts are executed in the ordered displayed (top to bottom). The results are then sent back to the application as HTML and the end-user then sees them displayed.
8. **Add:** Add another script to the plan. They can be dragged upwards or downwards to change the order of execution.
9. **Save:** Do not forget to SAVE YOUR WORK before returning to the home screen.

When you create a new plan, you'll need to specify if it will follow a template or not, and where the data for this plan will come from (a Voozanoo 4 application, a web service, a locally uploaded data file, or a combination of these).



1

2

Sortir Fournisseurs de données

Nouveaux

Nom \*

Mon analyse

Description

3

Mon premier analyse avec le R Statistics and Reporting Module.

Modèle initial \*

Outbreak investigations / Study

Surveillance

Methodological

Free analysis

### 1. New

On the bottom right of the home screen, create a new Analysis Plan.

### 2. Data sources

Displays a list of the data sources that you have to right to access.

### 3. Template

After giving your new plan a name and description, you can choose one of the standard epidemiological methods for treating data (specifically named and ordered analytical steps) or create and name your analyses freely with “**Free analysis**”.



Choose the data source or sources that you will need for the new Analysis Plan.

**Ajouter / Modifier la source de données**

1 +

Annuler

**Ajouter un fournisseur de données pour le w**

2 Nom de l'utilisateur Voozanoo 4 stats\_module

3 Mot de passe Voozanoo 4 .....

4 URL de l'application https://argonne.voozanoo.net

Local

**Add / Edit datasource**

WebService Voozanoo4 WebService Files

1 +

2 +

Data provider https://sandbox.preprod.voozanoo.net/fourmi ::

5 Data query données grippe

Alias grippe

6 Dico format Code

Raw  
Code  
Short label  
Label

Cancel

7

**Sélectionner le fichier de données**

Parcourir...

Fichier	Action	Alias
sk_rat_chemtraces.csv	Télécharger le fichier - Supprimer	skopia_traces

8

WebService Voozanoo 4 WebService Fichiers

1 +

Nom de l'utilisateur

Mot de passe

### Data source: Webservice Voozanoo 4

Click the + button 1 to define a new data source. If a **Data Provider** is available, you can simply choose one from the drop-down list. For the first analysis however, you will need to add a user 2 which has the right to retrieve data. In your Voozanoo 4 application, you should have already created a specific user and role for this Stats module to use, for example a user called **stats\_module**. Enter that user's **password** 3 and the **URL** 4 of your application. **Local** is for advanced environments (typically not selected).

Once the **Data Provider** is established, that user will have access to the various exports viewable by his role and group. Select the desired Export 5 with the **Data Query** drop-down menu. Then, give this data query an **Alias** (local name) to be used in your R scripts (referenced as *alias\_name.csv*), and finally choose the form 6 that you want for your dictionary variables.

### Data source: local File

You can upload CSV files 7 to your Analysis plan to be used as your main data source or as an auxiliary set of data which may be needed to help process the data retrieved by the web service above. The alias will be used in your R scripts (referenced in this example as **skopia\_traces.csv**).

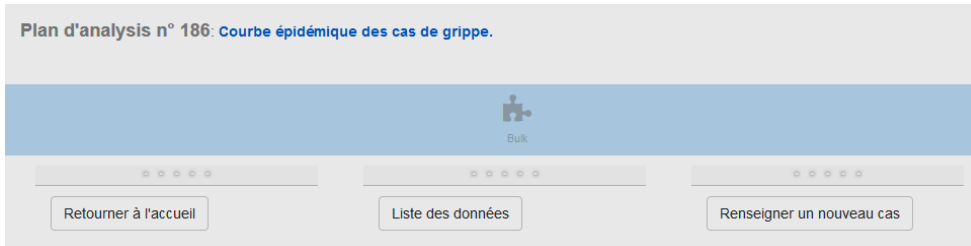
### Data source: Webservice

8 This can be used to allow other non-Voozanoo 4 web applications to call an analysis plan and receive its output.

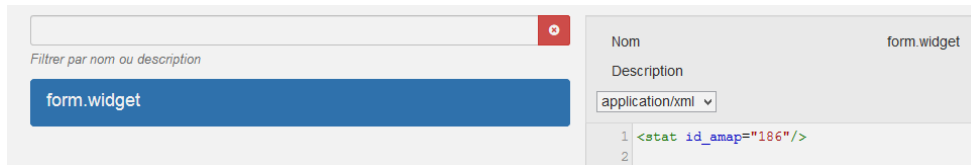
As of the writing of this guide, there is not yet a simple component in Epicraft called “graph” or “Map” to place on a page. Thus, it is by personalised properties in a **Bulk** component that an analysis is called and then displayed.

Here is an example of adding an analysis plan that makes an epidemic curve.

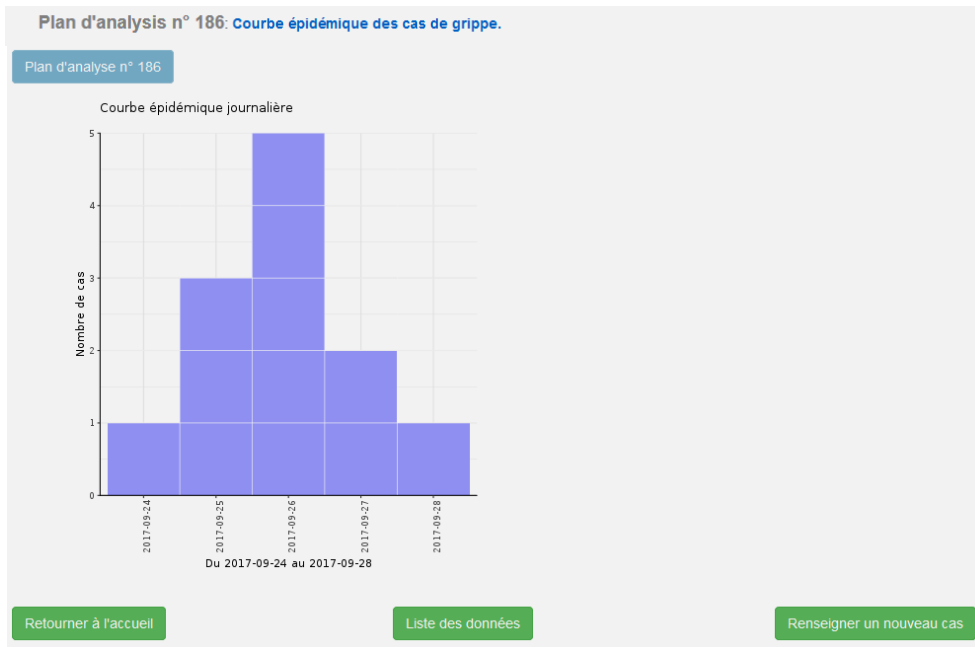
1



2



4



3

1. In Epicraft, add a bulk component to the page where you want the output of the plan to be displayed.

2. Open the personalised properties and add a new one called **form.widget**. Add the simple XML line

**<stat id\_ama="186">**

(replacing the **186** plan number with the one you have already created in the stats module and want to call).

3. Often, one does not want the analysis plan to start running as soon as the page displays (as there may be numerous plans to be displayed on the page). By using the following XML, you can delay its display by creating a button on the page to launch the execution of the plan.

```
<form_row>
 <button action="stat:exec186" class="btn btn-primary" label="Plan n° 186"/>
</form_row>

<stat id_ama="186">
 <option output="html" option_name="exec_event" value="stat:exec186"/>
 <option output="html" option_name="update_disabled" value="true"/>
</stat>
```

Again, replacing **186** by the id number of your analysis plan. When the button 4 is clicked, the Analysis plan will be launched and displayed.

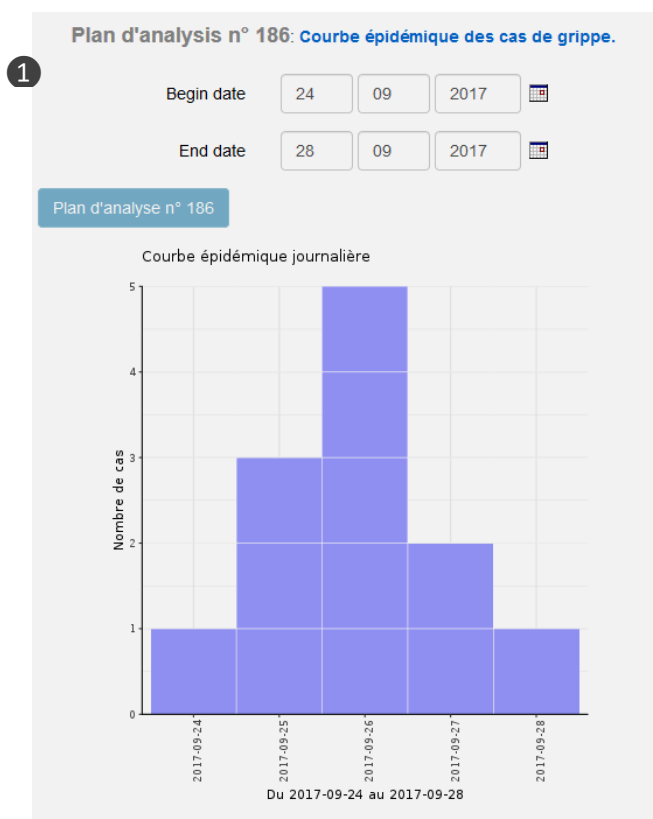
Often, you'd like to be able to filter the data that will be displayed by the plan. You can do this by passing values to an analysis R script.

2

```

form.dataset
application/xml
1 <dataset id="filter">
2 <metadata>
3 <fields>
4 <field id="begin" type="date" default_label="Begin date" mandatory="
5 <field id="end" type="date" default_label="End date" mandatory="fals
6 </fields>
7 </metadata>
8 </dataset>

```



3

```

1 <form_row>
2 <label dataset="filter" field="begin"/>
3 <value dataset="filter" field="begin" mode="rw"/>
4 </form_row>
5 <form_row>
6 <label dataset="filter" field="end"/>
7 <value dataset="filter" field="end" mode="rw"/>
8 </form_row>
9
10 <form_row>
11 <button action="stat:exec186" class="btn btn-primar
12 </form_row>
13
14 <stat id_amap="186" dataset_filter="filter">
15 <option output="html" option name="exec event" value="

```

4

```

1 ``{r, results="asis", echo=FALSE, mes
2
3 #library(dplyr)
4 #library(sp)
5 #library(gridExtra)
6 #library(ggplot2)
7 options(xtable.floating = FALSE)
8 options(xtable.timestamp = "")
9
10 begin = JSON$params$begin
11 end = JSON$params$end
12

```

## Explanation

1. In this example, we'll ask the user to enter starting and ending dates and pass those values to the analysis plan to have the plan limit which data will be graphed.
2. Add a "filter" dataset to the page's personalised properties named **form.dataset**.

```

<dataset id="filter">
 <metadata>
 <fields>
 <field id="begin" type="date" default_label="Begin date"
 mandatory="false"/>
 <field id="end" type="date" default_label="End date" mandatory="false"/>
 </fields>
 </metadata>
</dataset>

```

Set the mandatory attribute to "true" if your script requires the variables to be non-null.

3. Add entry fields to the bulk and add the attribute **dataset\_filter="filter"** to the **<stat>** tag.

```

<form_row>
 <label dataset="filter" field="begin"/>
 <value dataset="filter" field="begin" mode="rw"/>
</form_row>
<form_row>
 <value dataset="filter" field="end" mode="rw"/>
</form_row>

<stat id_amap="186" dataset_filter="filter">

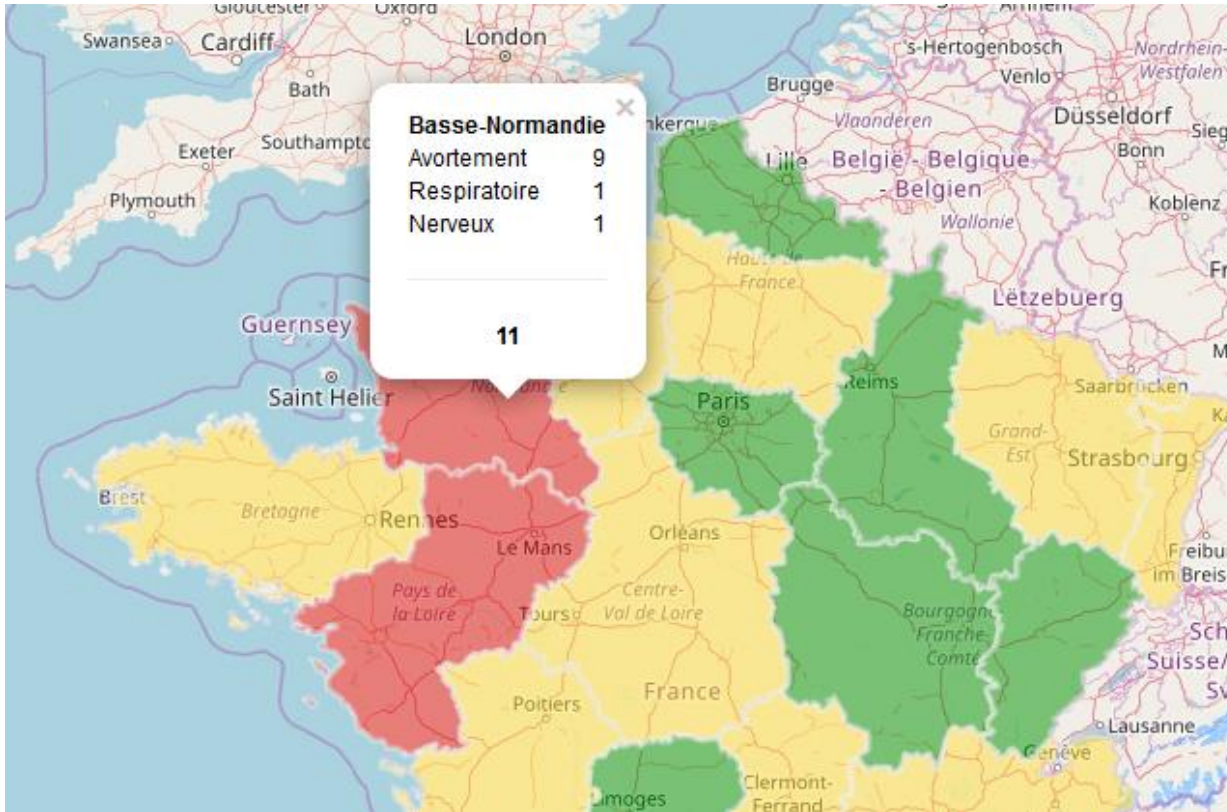
```

4. To use the values within your R analysis script, you will use **JSON\$params\$xxxx** where the xxxx is the name of your variable coming from the Voozanoo application. In this example the script will use the **begin** and **end** variables to limit which data is used for the analysis.

There exists a Voozanoo 4 widget called <carto> which is used similarly to the <stat> widget (p.10) but includes features that allow additional dynamic data display such as map colourisation based on aggregate data or lists of values for a given country, region, county, or town.

### Map your data

For more details on putting cartographic data displays into your Voozanoo 4 application, contact an Epiconcept project Director or your normal Epiconcept interlocutor.



The R server for Voozanoo includes the following libraries.

abind	compiler	flashClust	htmlwidgets	markdown	plotrix	regtest	shiny	timeDate
acepack	cowplot	flextable	httpuv	MASS	plyr	ReporteRs	slam	timeline
assertthat	crayon	foreach	httr	Matrix	png	ReporteRsjars	sna	timereg
backports	crayon	foreign	igraph	MatrixModels	png	reshape	snow	timeROC
base	curl	formatR	ipred	memoise	polyspline	reshape2	snowfall	timeSeries
base64enc	CVST	Formula	irlba	methods	polyclip	rgdal	sourcetools	tinytex
BH	Daim	fracdiff	ISOweek	mfp	polyCub	rgeos	sp	tm
BiasedUrn	data.table	GADMTools	iterators	mgcv	praise	RgoogleMaps	SparseM	tools
bindr	dataframes2xls	gdata	jpeg	mice	pROC	riverplot	spatial	treemap
bindrcpp	datasets	gdtools	jsonlite	mime	prodlim	rJava	spatstat	TSA
binom	DBI	geosphere	kernlab	minqa	proto	rjson	spatstat.utils	tseries
binomTools	ddalpha	ggmap	KernSmooth	missMDA	purrr	RJSONIO	splines	TTR
bit	deldir	ggplot2	knitr	ModelMetrics	qcc	rlang	stabledist	utils
bitops	DEoptimR	git2r	labeling	msm	quadprog	rmarkdown	statnet.common	uuid
blockrand	devtools	glmnet	lattice	multcomp	quantmod	rms	stats	viridis
boot	dichromat	glue	latticeExtra	munsell	quantreg	robustbase	stats4	viridisLite
car	digest	gofest	lava	mvtnorm	R.methodsS3	ROCR	stringi	whisker
caret	dimRed	gower	lazyeval	network	R.oo	rpart	stringr	withr
caret	docopt	gpclib	leaps	nlme	R.utils	rpart.plot	surveillance	xlsx
caTools	dplyr	gplots	littler	nloptr	R6	rprojroot	survival	xlsxjars
CCP	DRR	graphics	lme4	NLP	randomForest	RSclient	tabplot	XML
checkmate	DT	graphicsQC	lmtree	nnet	randomizeR	Rserve	tcltk	xml2
chron	e1071	grDevices	locfit	numDeriv	raster	RServerUtils	tensor	xtable
chron	ellipse	grid	lsmeans	officer	RColorBrewer	rstudioapi	testthat	xts
class	Epiconcepts	gridBase	lubridate	Openssl	Rcpp	rvg	TH.data	yaml
classInt	epiR	gridExtra	magick	outliers	RcppArmadill	sandwich	tibble	zip
classInt	estimability	gss	magrittr	pander	o	scales	tidyr	zoo
cluster	evaluate	gtable	mapproj	parallel	RcppEigen	scatterplot3d	tidyselect	
clusteval	Expm	gtools	maps	pbkrtest	RcppRoll	sfsmisc		
coda	FactoMineR	highr	maptools	pec	recipes			
codetools	fastmatch	Hmisc		pixmap				
colorRamps	fBasics	htmlTable		pkgconfig				
colorspace	ff	htmltools		plogr				
colorspace	ffbase							