



plateforme R

Module R Statistiques et Reporting pour Voozаноо : Feuilles de référence

Editeur :	Epiconcept, 47 rue de Charenton, 75012 Paris, France - www.epiconcept.fr
Titre :	Module R Statistiques et Reporting : Feuilles de référence
Version actuelle :	Version 3 (13 déc 2019)
Versions précédentes :	Version 1.01 (19 nov 2015) Version 1.1 (21 déc 2015) Version 2 (14 nov 2017)
Concerne :	Module R Statistiques et Reporting version 19w38
Licence :	Copyright © Epiconcept 2019
Auteur(s) :	Curtis Broderick
Distribution :	Publique
Autres langues :	Disponible en anglais
Documents corollaires :	Fiches de référence Voozanoo 4
Autres informations :	
Commentaires :	Merci de bien vouloir adresser vos remarques concernant ce document à c.broderick@epiconcept.fr

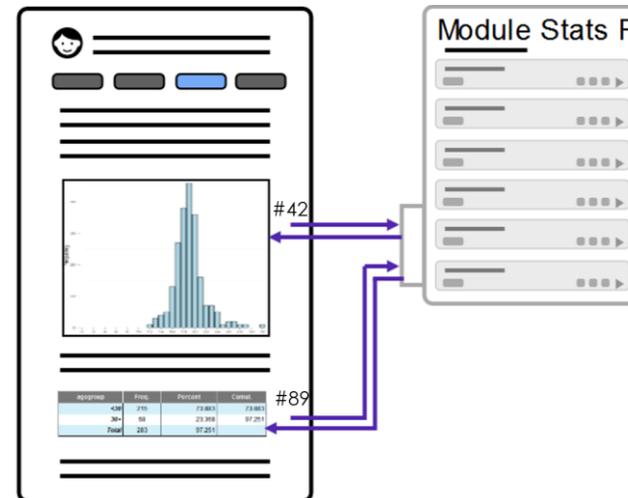
INFORMATIONS CONCERNANT CE DOCUMENT	2
TABLE DE MATIÈRE	3
VUE D'ENSEMBLE	4
GLOSSAIRE ET LANGAGE MARKDOWN	5
PAGE D'ACCUEIL : LA LISTE DES PLANS D'ANALYSES	6
LE PLAN D'ANALYSE	7
CREER UN NOUVEAU PLAN D'ANALYSE (1 DE 2)	8
CREER UN NOUVEAU PLAN D'ANALYSE (2 DE 2)	9
EXECUTION DE SCRIPTS DEPUIS VOO4 : WIDGET <STAT>	10
PASSER DES VARIABLES A VOTRE SCRIPT R	11
DONNEES DYNAMIQUES POUR LA CARTOGRAPHIE	12
ANNEXE 1 LES BIBLIOTHÈQUES DANS R	13

Ce document décrit l'organisation et l'utilisation du **Module R Statistiques et Reporting** d'Epiconcept. Ce module est une application Voozadoo 4 et marche avec un serveur R dédié qui exécute les scripts. Le module peut être utilisé directement en tant qu'application, mais il est plus généralement utilisé indirectement comme service web pour une autre application Voozadoo afin d'exécuter des analyses statistiques, des rapports et des représentations de données cartographiques. Ce module n'est pas accessible à tous les utilisateurs d'un système d'information Voozadoo (même si ce système fait appel à ce module pour produire des analyses et des rapports). L'utilisation de ce module implique que l'utilisateur ait un accès privilégié à son application Voozadoo et qu'il / elle soit un programmeur R compétent.

Le concept

Une application Voozadoo 4 peut afficher à un utilisateur toutes ses données à travers des listings et des filtres (en fonction bien sûr des droits d'accès de l'utilisateur). Cependant, il n'y a pas de système intégré pour afficher graphiquement les données ou pour analyser mathématiquement les données (telles que les analyses statistiques). L'idée est que l'on crée des scripts R dans ce module puis, dans l'application Voozadoo de l'utilisateur, ces scripts spécifiques sont appelés et exécutés par un serveur R dédié. Cela signifie que le script d'exécution doit avoir accès à l'application et avoir une requête de données spécifique pour pouvoir récupérer ces données, effectuer le traitement, puis renvoyer le graphique, la carte ou le rapport requis au format HTML.

En général, cela fonctionne comme suit : un script donné tel que # 42 ou # 89 a



été préconfiguré (quand il a été écrit) avec une requête de données spécifique à l'application de l'utilisateur et avec des scripts R spécifiques (dans cet exemple respectivement, un graphique et une liste avec de calculs). La page est configurée dans Epicraft avec du texte et des

boutons, puis un appel à lancer l'analyse n ° 42, un peu plus de texte et un appel à lancer l'analyse n ° 89. Le module stats, ayant accès aux données de l'application (par biais d'un utilisateur spécifiquement créé pour le plan d'analyse), lit et traite les données, puis renvoie un objet html que la page affiche ensuite.

Glossaire

Script : Aussi connu sous le nom de Script R. Il est écrit dans le langage R et stocké dans le module de statistiques.

Une analyse : autre nom pour un script.

Plan d'analyse : un regroupement de scripts R. Il s'agit d'un plan d'analyse exécuté à partir d'une application Voozadoo, et non d'une analyse individuelle.

Source : La source de données, dont le script R aura besoin pour exécuter son analyse et / ou sa représentation graphique. Les sources de données peuvent être une application Voozadoo 4, une application tierce ou un fichier CSV téléchargé.

Balise (tag) : tout mot-clé que vous souhaitez donner à un plan d'analyse pour vous aider à le retrouver plus facilement parmi les autres plans de la liste.

Langage Markdown

M↓ L'application R qui exécute vos scripts interprétera le langage de Markdown. Au début de chaque script, ajoutez ````${r echo = FALSE, message = FALSE}` pour indiquer "lire ce qui suit comme langage R". Il suffit d'ajouter la ligne ````` (trois accents) pour indiquer "lire ce qui suit comme du texte Markdown". Ensuite, pour revenir à l'interprétation R, insérer la ligne ``${r message = FALSE}`. Retrouver l'aide-mémoire du langage Markdown ici :

<https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

```
```${r echo=FALSE, message=FALSE}
Ceci est une ligne de commentaire R
Rien n'est affiché parce que R ignore ces lignes
```

Maintenant c'est du Markdown. Avez-vous vu ces trois
petits accents graves avant cette ligne ?
## Quand j'utilise les dièses, ils sont interprétés
comme les niveaux d'en-tête H1, H2, H3 ...
```${r message=FALSE}
Au retour au script, nous utilisons le dièse pour les
commentaires R encore
2+3
on verra l'expression évaluée et affichée, 5.
```

Lorsque vous entrez dans le module « Stats », vous obtenez une liste des plans d'analyse auxquels vous avez le droit d'accéder. Chacun d'entre eux contient une ou plusieurs analyses (scripts R) à l'intérieur. Lorsque l'un de ces plans est appelé par l'application de l'utilisateur, tous les scripts du plan donné sont exécutés dans l'ordre.

The screenshot shows the 'plateforme R' interface. At the top left is the logo and 'FR' language selector. Below is a 'Portail' button. The main section is titled 'Plans d'analyse'. It features three search filters: 'Recherche par identifiant du plan d'analyse' (with example '4242'), 'Recherche par tag' (with example 'Age'), and 'Recherche par nom' (with example 'rapport sur les naissances'). Below these are three plan cards. The first card, '228 - Test sources de données', has tags 'test' and 'source' and a detailed description 'Voici un petit test sur les sources de données'. The second card is '218 - Test RBE' with tag 'test' and description 'Quelques tests temporaires'. The third card is '216 - Analyse sur SBD Amérique du Nord' with tags 'SBD', 'IA', and 'Argonne'. Each card has a set of icons for editing, copying, deleting, and running. Numbered callouts (1-13) point to various elements: 1 (search ID), 2 (search tag), 3 (search name), 4 (description), 5 (plan title), 6 (tag), 7 (tag), 8 (description), 9 (edit icon), 10 (copy icon), 11 (delete icon), 12 (run icon), and 13 (close icon).

### 1. Recherche par ID Plan d'analyse

Rechercher un plan d'analyse par son ID (numéro d'identification)

### 2. Rechercher par tags

Permet de filtrer la liste affichée en fonction des tags donnés aux plans.

### 3. Rechercher par nom

Permet de filtrer la liste des plans affichés par le nom d'après les caractères saisis.

### 4. Un plan d'analyse

5. Le numéro d'identification qui devra être utilisé pour l'appeler à partir de son application Voozadoo.
6. Un nom descriptif pour le plan.
7. Tags pour vous aider à le trouver facilement avec le filtre par tags.
8. Une explication plus détaillée de ce que fait le plan.
9. Ouvrez un plan pour l'afficher ou le modifier.
10. Créer une copie de ce plan.
11. Voir un journal d'activité du plan.
12. Exécuter ce plan. Il exécutera chacun des scripts dans l'ordre et produira un fichier HTML.
13. Supprimez ce plan d'analyse et tous ses scripts.

Dans un plan d'analyse, on peut ajouter ou supprimer des sources de données (des requêtes à la base de données ou fichiers CSV téléchargés) et gérer les divers scripts R qui composent le plan.

### Boutons

1. Modifiez le nom, la description ou les tags du plan. Les tags ne peuvent pas contenir d'espaces.
2. Ajoutez, supprimez ou modifiez une source de données.
3. Téléchargez tous les scripts de ce plan sous la forme d'un seul fichier texte.
4. Passez en revue l'une de vos sources de données.

Parcourir les données : affiche les données sous forme de tableau.

Afficher le descriptif des données : donne un résumé statistique de chaque variable (moyenne, variance, nombre de réponses vides, etc.)

5. Groupe : vous pouvez modifier le groupe auquel appartient ce plan d'analyse.

6. Analyse : le script écrit dans le langage R à exécuter.

7. Liste d'analyses : liste des analyses (scripts R) qui composent le plan. Lorsque l'ID de ce plan est appelé depuis l'application Voozano 4 de l'utilisateur, les scripts sont exécutés dans l'ordre affiché (de haut en bas). Les résultats sont ensuite renvoyés à l'application au format HTML et l'utilisateur final les voit ensuite s'afficher.

8. Ajouter : Ajouter un autre script au plan. Ils peuvent être déplacés vers le haut ou vers le bas pour changer l'ordre d'exécution.

9. Sauvegarder : **N'oubliez pas** de sauvegarder votre travail avant de retourner à l'écran d'accueil.

Lorsque vous créez un nouveau plan, vous devez spécifier s'il suivra ou non l'un des modèles, et d'où proviendront les données pour ce plan (d'une application Voozanoo 4, d'un service Web, d'un fichier de données téléchargé localement ou d'une combinaison de ceux-ci).



The screenshot shows the Voozanoo logo and a 'Portail' button. The main heading is 'Sélectionner le modèle initial'. The form contains the following fields:

- Nom \***: A text input field containing 'Ma première analyse'.
- Description**: A text area containing 'Ma première analyse sur les données XYZ avec le module R de statistiques et reporting.'.
- Modèle initial \***: A radio button selection with four options:
  - Outbreak investigations / Study
  - Surveillance
  - Methodological
  - Free analysis (marked with a circled 3)

At the bottom, there are 'Annuler' and 'Suivant' buttons.

## 1. Nouveau

En bas à droite de l'écran d'accueil, créez un nouveau plan d'analyse.

## 2. Sources de données

Afficher une liste des sources de données auxquelles vous avez le droit d'accéder.

## 3. Modèle

Après avoir donné un nom et une description à votre nouveau plan, vous pouvez choisir l'une des méthodes analytiques (spécifiquement nommées et ordonnées) ou créer et nommer vos analyses librement avec «Free analysis».

Choisissez-la ou les sources de données dont vous aurez besoin pour le nouveau plan d'analyse.

**Ajouter / Modifier la source de données**

1 +

Annuler

**Add / Edit datasource**

WebService Voozanoo4 WebService Files

1 +

Data provider:  +

5 Data query:

Alias:

6 Dico format:  (Raw, Code, Short label, Label)

Cancel

7

**Sélectionner le fichier de données**

Parcourir...

Fichier	Action	Alias
sk_rat_chemtraces.csv	Télécharger le fichier - Supprimer	<input type="text" value="skopia_traces"/>

8

WebService Voozanoo 4 Webservice Fichiers

1 +

Nom de l'utilisateur:

Mot de passe:

URL de l'application:

Local:

## Source de données : Webservice Voozanoo 4

Cliquez 1 sur le bouton + pour définir une nouvelle source de données. Si un fournisseur de données est disponible, vous pouvez simplement en choisir un dans la liste déroulante. Pour la première analyse cependant, vous devrez ajouter un utilisateur 2 qui a le droit de récupérer des données. Dans votre application Voozanoo 4, vous devriez déjà avoir créé un utilisateur et un rôle spécifique pour ce module Stats, par exemple un utilisateur appelé **stats\_module**. Entrez le **mot de passe** 3 de cet utilisateur et l'**URL** 4 de votre application.

**Local** est pour les environnements avancés (généralement non sélectionnés). Une fois le fournisseur de données établi, cet utilisateur aura accès aux différentes exportations visibles par son rôle et son groupe. Sélectionnez l'exportation 5 souhaitée avec le menu déroulant **Data Query**. Ensuite, donnez à cette requête de données un alias (nom local) à utiliser dans vos scripts R (référéncé comme **nom\_alias.csv**), et enfin choisissez le format 6 que vous voulez pour vos variables de dictionnaire.

## Source de données : fichier local

Vous pouvez télécharger des fichiers CSV 7 dans votre plan d'analyse pour les utiliser comme source de données principale ou comme données auxiliaires. Ces dernières peuvent être nécessaires pour traiter les données récupérées par le service Web ci-dessus. L'alias sera utilisé dans vos scripts R (référéncé dans cet exemple comme **skopia\_traces.csv**).

## Source de données : Webservice

8 Ceci peut être utilisé pour permettre à d'autres applications Web, non Voozanoo 4, d'appeler un plan d'analyse et de recevoir les résultats.

Au moment de la rédaction de ce guide, il n'y a pas encore de composant simple dans Epicraft appelé "graphique" ou "Carte" à placer sur une page. C'est donc par des propriétés personnalisées dans un composant Bulk qu'une analyse est appelée puis affichée.

Plan d'analyse n° 186: Courbe épidémique des cas de grippe.

1

2

3

4

Plan d'analyse n° 186: Courbe épidémique des cas de grippe.

Plan d'analyse n° 186

Courbe épidémique journalière

Date	Nombre de cas
2017-09-24	1
2017-09-25	3
2017-09-26	5
2017-09-27	2
2017-09-28	1

Du 2017-09-24 au 2017-09-28

Retourner à l'accueil

Liste des données

Renseigner un nouveau cas

Voici un exemple d'ajout d'un plan d'analyse qui fait une courbe épidémique.

1. Dans Epicraft, ajoutez le composant **Bulk** à la page où vous souhaitez afficher ce qui est produit par un de vos plan d'analyse.

2. Ouvrez les propriétés personnalisées et ajoutez-en une nouvelle appelée **form.widget**. Ajouter simplement la ligne XML : **<stat id\_ama = "186">**

(en remplaçant le numéro du plan 186 par celui que vous avez créé dans le module Stats et que vous souhaitez appeler).

3. Souvent, on ne souhaite pas que le plan d'analyse commence à s'exécuter dès que la page s'affiche (car il peut y avoir de nombreux plans à afficher sur la page). En utilisant le XML suivant, vous pouvez retarder son affichage en créant un bouton sur la page pour lancer l'exécution du 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>
```

Encore une fois, en remplaçant 186 par le numéro d'identification de votre plan d'analyse. Lorsque le bouton 4 est actionné, le plan d'analyse sera lancé et affiché.

Souvent, vous souhaitez pouvoir filtrer les données affichées par le plan. Vous pouvez le faire en transmettant des valeurs à un script d'analyse R.

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>

```

1

Plan d'analyse n° 186: Courbe épidémique des cas de grippe.

Begin date: 24 09 2017

End date: 28 09 2017

Plan d'analyse n° 186

Courbe épidémique journalière

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_ama="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

```

Dans cet exemple 1, nous demandons à l'utilisateur d'entrer les dates du début et de la fin et de transmettre ces valeurs au plan d'analyse afin que le plan limite les données qui seront représentées.

2. Ajoutez un jeu de données "filter" aux propriétés personnalisées de la page nommées **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>

```

Définissez l'attribut obligatoire sur "true" si votre script requiert que les variables soient non nulles.

3. Ajoutez des champs de saisie au Bulk et ajoutez l'attribut **dataset\_filter = "filter"** à la balise **<stat>**.

```

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

```

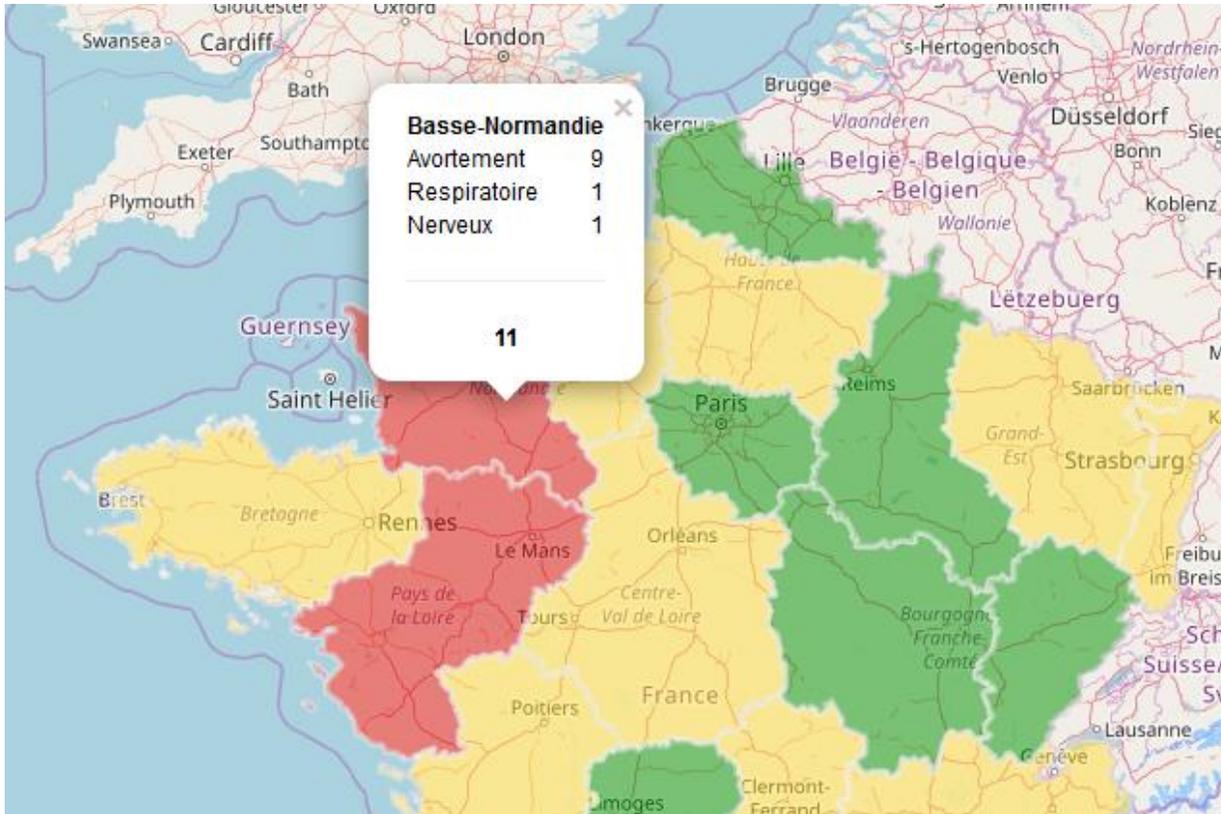
```

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

```

4. Pour utiliser les valeurs dans votre script d'analyse R, vous utiliserez **JSON\$params\$xxxx** où le xxxx est le nom établi de votre variable (ligne 10 et 11 à gauche) provenant de l'application Voozano. Dans cet exemple, le script utilisera les variables **begin** et **end** pour limiter les données utilisées dans l'analyse.

Il existe un widget Voozano 4 appelé <carto> qui est similaire au widget <stat> (p.10) mais qui inclut des fonctionnalités permettant d'afficher des données dynamiques supplémentaires telles que la coloration de la carte à partir de données ou de listes de valeurs pour un pays, région, comté ou ville donnés.



### Afficher vos données

Pour plus de détails sur la mise en place des affichages des données cartographiques dans votre application Voozano 4, contactez un des directeurs de projet Epiconcept ou votre interlocuteur Epiconcept habituel.

Les bibliothèques installées sur le serveur R

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	goftest	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	RScIlient	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							