

Gaspard_Palay_CreateMapRaster_ThomasMasse

Gaspard PALAY

1/30/2021

1. Critères d'évaluation

1. Comportement du Rmd lors de son exécution
2. Qualité rédactionnelle du dossier
3. Didactisme et pertinence du dossier
4. Qualité et lisibilité du Rmarkdown
5. Qualité des applications permettant d'illustrer le package

2. Lien vers le document commenté

En cliquant **ici**, vous trouverez le lien menant au GitHub de Thomas Massé qui héberge les travail réalisé.

3. Auteurs du document commenté

Le document évalué dans le cadre de ce rendu a été produit par Thomas Massé, étudiant en MSc Data Management à Paris School of Business.

4. Synthèse du document

Le document illustre l'utilisation du package Raster et s'intéresse plus particulièrement à la création d'une carte grâce au package.

A travers ce document, l'auteur commence tout d'abord par expliquer que plusieurs package permettent de créer des carte. Il explique que représenter des données géographiques directement sur une carte est bien plus pertinent et permet d'éblouir le lecteur grace à cette data vizualisation. Il décompose deux types de cartes :

- Vectorielles : comme celle créées grâce au package rayShader par exemple
- Raster : les images

Il décompose son document de la manière suivante :

1. L'installation du package
2. La focntion GetData
3. Les différentes cartes
4. La création des contours de la carte
5. la carte topographique

Le document est bien structuré.

5. Extrait commenté des parties de code

```
install.packages("raster", repos = "http://cran.us.r-project.org")
install.packages("rgdal", repos = "http://cran.us.r-project.org")
install.packages("RCurl", repos = "http://cran.us.r-project.org")
#Définition du répertoire de travail
setwd("C:/R/Maps")
```

Ce code est très bien structuré, assez didactique et parle au lecteur.

L’auteur expose ensuite une fonction très importante pour la suite de la réalisation de son travail à savoir **getData** tout en citant bien sa source.

L’auteur est très clair dans l’explication de la fonction et de ses paramètres :

La fonction **getData** permet d’obtenir des données géographiques de n’importe où dans le monde. Elle possède 3 arguments principaux :

- *name* : c’est le nom du jeu de données
- *download* : si on télécharge le jeu de données ou non
- *path* : le chemin dans lequel stocker les données
- ainsi que des arguments additionnels selon le jeu de données choisi.

L’argument “name” peut prendre les valeurs suivantes :

- *alt*
- *GADM*
- *worldclim*
- *SRTM*

Si le nom “alt” ou “GADM” est utilisé il faut également donner un argument “country”.

Ensuite, l’auteur explique au lecteur comment générer les contours d’un pays pour pouvoir travailler dessus ensuite, grâce à la fonction **GADM**

Le document, comme expliqué au préalable, montre les différents types de cartes disponibles :

```
maCarteAlt <- getData(name="alt", country="GBR")
plot(maCarteAlt, main="Carte alt")
```

Génère une carte Alt de la grande bretagne grace au code ISO GBR.

```
maCarteGadm <- getData(name="GADM", country="GBR", level=1)
plot(maCarteGadm, main="Carte GADM")
```

Génère une carte GADM de la grande bretagne grace au code ISO GBR.

```
maCarteWorldclim <- getData('worldclim', var='tmin', res=0.5, lon=0, lat=55)
plot(maCarteWorldclim, main="Carte Worldclim")
```

Génère plusieurs cartes du monde avec les climats et la correspondance couleur (température)

```
maCarteSrtm <- getData("SRTM", lon=-2, lat=57)
plot(maCarteSrtm, main="Carte SRTM")
```

Génère l'altitude des différentes régions de l'Ecosse.

L'auteur génère ensuite les contours de la grande Bretagne en les récupérant sur un URL. Il pense même à print un message d'erreur si l'adresse URL est indisponible...

Il exprime la manière d'extraire les données non visibles des contours de la carte, ses attributs, via les fonctions et chunk suivants :

```
maCarteGadm$NAME_0
```

```
maCarteGadm$NAME_1
```

```
maCarteGadm$NAME_2
```

Dans un dernier temps, l'auteur crée une carte topographique qu'il choisit lui-même. Il montre au lecteur comment ajouter des Points d'intérêts.

```
plot(maCarteCrop, main="Ma Carte avec 1 POI")
plot(maCarte, add=TRUE)
points(-2.1, 57.1, pch=21, col="black", bg="red")
text(-2.1, 57.1, labels="Aberdeen", cex=1, pos=3)
```

Puis l'ajout de plusieurs POI grace aux vecteurs :

```
plot(maCarteCrop, main="Ma Carte avec plusieurs POI")
plot(maCarte, add=TRUE)
p.lon <- c(-2.1, -3.2)
p.lat <- c(57.1, 55.95)
p.lbl <- c("Aberdeen", "Edinburgh")

points(p.lon, p.lat, pch=21, col="black", bg="red")
text(p.lon, p.lat, labels=p.lbl, cex=.8, pos=3)
points(-4.25, 55.8, pch=22, col="black", bg="blue")
text(-4.25, 55.8, labels="Glasgow", cex=1, pos=1)
```

6. Evaluation du travail suivant les 5 critères

1. Comportement du Rmd lors de son exécution

L'exécution du code Rmd se réalise parfaitement et sans conditions. Le code RMD appelle bien les librairie nécessaire et les installe aussi si l'utilisateur ne les possède pas au préalable. L'exécution du code est longue sur une machine peu performante du à la grande quantité de données appelées.

2. Qualité de la rédaction du dossier

La rédaction de ce document est de très bonne qualité. Il ne comporte aucune faute d'orthographe et syntaxique.

Le document est très bien rédigé.

3. Didactisme et pertinence du dossier

Le dossier se retrouve facilement sur son github qui est d'ailleurs très bien structuré. Le fichier ne comporte pas de table des matières.

Les descriptions sont bien explicitées et bien illustrées.

L'auteur arrive à faire adhérer le lecteur à sa production. Il transmet une connaissance nécessaire à la reproduction de son travail.

L'objectif de ce document est rempli.

4. Qualité et lisibilité du RMarkdown

Le RMarkdown est bien écrit. Les tableaux sont bien structurés et les cartes se génèrent facilement.

Il est lisible et les chunk sont très bien paramétrés. Autre point d'intention : l'auteur a pris le temps de créer son propre style de RMD qu'il applique à toutes ses productions. Ce style est beau, bien créé, met en exergue les chunks grâce à un effet de contouring. Ce travail qui ne lui était pas demandé mérite d'être valorisé car il rend le PDF et RMD très visible et agréable à la lecture.

5. Qualité des applications permettant d'illustrer le package

L'auteur s'appuie sur différents tutoriels github. Les exemples de cartes utilisés sont différents et reprennent plusieurs pays ce qui change. L'auteur va au bout des choses en illustrant ses propos avec une très grande quantité d'exemple.

7. Conclusion

Il s'agit d'un très bon travail selon moi. L'auteur est arrivé au bout de son travail de manière très visuelle. Les termes qu'il adopte sont très bien choisis et illustrent avec pertinence ses propos. On y apprend beaucoup de choses. Le document est enregistré en local sur mon ordinateur et me servira très probablement à la réalisation de travaux personnels. Il m'aurait été d'une très grande utilité pour la création de mon tutoriel sur RayShader.

Vous retrouvez ce document sur mon **GitHub**.