

Prérequis R pour la formation "Science reproductible"

Tristan Mary-Huard

13/01/2020

Ce document a pour objectif d'évaluer son niveau de maîtrise du logiciel R. Les trois exercices couvrent différents types de compétence qu'il est utile de maîtriser pour pouvoir suivre la formation sans difficulté :

- connaissance de quelques outils de base d'analyse de données en R,
- manipulation d'un jeu de données,
- programmation.

Les deux premiers exercices requièrent des notions minimales, qui seront nécessaires pour suivre la formation proposée. Le troisième exercice requiert un peu de programmation, qui n'est pas nécessaire pour suivre la formation mais qui ouvre la possibilité de traiter certains sujets de projets lors de la session 2 (voir programme).

Pour chaque question la réponse (ou un indice) est donné pour permettre de vérifier son code. Il n'est pas demandé d'obtenir un résultat formellement identique, le résultat de chaque question pouvant être obtenu de différentes manières, et sous différentes formes (une matrice, un data.frame, un tibble).

Exercice 1 : Analyse de données

On considère le jeu de données "ChickWeight" disponible dans le logiciel R. Ce jeu de données consiste en une série de mesures de prise de poids réalisées sur des poulets à différents stades de leur croissance, les poulets étant soumis à différents régimes. On souhaite ici faire une analyse de la variance afin de tester si le régime a une influence après 8 jours de suivi. Les données peuvent être obtenues de la manière suivante :

```
data("ChickWeight")
head(ChickWeight)
```

	weight	Time	Chick	Diet
1	42	0	1	1
2	51	2	1	1
3	59	4	1	1
4	64	6	1	1
5	76	8	1	1
6	93	10	1	1

Le modèle à ajuster est le suivant :

$$Y_{ij} = \mu + \alpha_i + E_{ij}, \quad E_{ij} \sim \mathcal{N}(0, \sigma^2), \text{ i.i.d.}$$

1/ Créer une table ne contenant que les observations relatives au temps 8.

Indice: la table obtenu doit contenir 49 lignes.

2/ Transformer la variable "Diet" qui indique le type de régime suivi en facteur.

3/ Ajuster le modèle d'analyse de la variance.

Indice: utiliser la fonction **lm**.

4/ Tester l'effet du régime avec la fonction “anova”.

Réponse:

Analysis of Variance Table

Response: weight

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
factor(Diet)	3	5114.1	1704.69	10.167	3.128e-05 ***
Residuals	45	7545.0	167.67		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Exercice 2 : Manipulation d'un jeu de données

On considère le jeu de données “nycflights13”, qui se trouve dans le package du même nom (téléchargeable sur le CRAN). Chaque ligne du tableau de données correspond à un vol réalisé en 2013, au départ de New-York. Chaque vol est décrit par 19 variables, on s'intéressera ici uniquement aux variables suivantes :

- *carrier* : nom de la compagnie ayant réalisé le vol,
- *arr-delay* : retard évalué à l'arrivée du vol,

5/ Obtenir le nombre de vols réalisé par compagnie.

```
# A tibble: 16 x 2
  carrier NbFlPerCarrier
  <chr>      <int>
1 9E         18460
2 AA         32729
3 AS          714
4 B6         54635
5 DL         48110
6 EV         54173
7 F9          685
8 FL         3260
9 HA         342
10 MQ        26397
11 OO          32
12 UA        58665
13 US        20536
14 VX         5162
15 WN        12275
16 YV         601
```

6/ Sélectionner les compagnies ayant réalisé au moins 10,000 vols en 2013.

```
# A tibble: 325,980 x 8
# Groups:   carrier [9]
```

```

      month dep_time arr_delay carrier origin dest time_hour
      <int>   <int>   <dbl> <chr>   <chr> <chr> <dtm>
1      1      517      11 UA      EWR   IAH   2013-01-01 05:00:00
2      1      533      20 UA      LGA   IAH   2013-01-01 05:00:00
3      1      542      33 AA      JFK   MIA   2013-01-01 05:00:00
4      1      544     -18 B6      JFK   BQN   2013-01-01 05:00:00
5      1      554     -25 DL      LGA   ATL   2013-01-01 06:00:00
6      1      554      12 UA      EWR   ORD   2013-01-01 05:00:00
7      1      555      19 B6      EWR   FLL   2013-01-01 06:00:00
8      1      557     -14 EV      LGA   IAD   2013-01-01 06:00:00
9      1      557      -8 B6      JFK   MCO   2013-01-01 06:00:00
10     1      558       8 AA      LGA   ORD   2013-01-01 06:00:00
# i 325,970 more rows
# i 1 more variable: NbFlPerCarrier <int>

```

7/ Pour les compagnies sélectionnées, calculer le retard moyen à l'arrivée par compagnie.

```

# A tibble: 9 x 2
  carrier AvDelayPerMonth
  <chr>      <dbl>
1 9E          7.38
2 AA          0.364
3 B6          9.46
4 DL          1.64
5 EV         15.8
6 MQ         10.8
7 UA          3.56
8 US          2.13
9 WN          9.65

```

8/ Calculer ce même retard moyen à l'arrivée, par compagnie et par mois.

'summarise()' has grouped output by 'carrier'. You can override using the '.groups' argument.

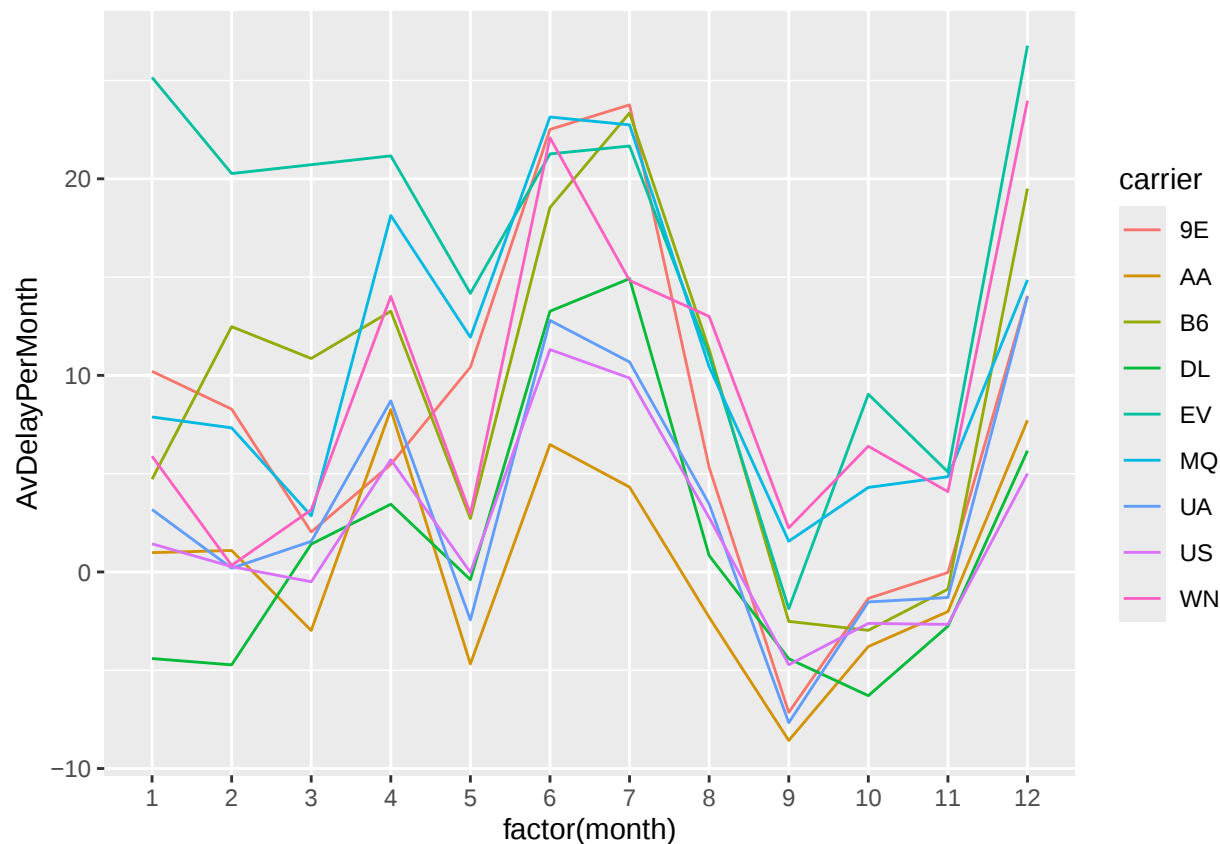
```

# A tibble: 108 x 3
# Groups:   carrier [9]
  carrier month AvDelayPerMonth
  <chr>   <int>      <dbl>
1 9E       1      10.2
2 9E       2       8.28
3 9E       3       2.03
4 9E       4       5.47
5 9E       5      10.4
6 9E       6      22.5
7 9E       7      23.8
8 9E       8       5.31
9 9E       9      -7.14
10 9E      10      -1.35
# i 98 more rows

```

9/ Realiser le graphique représentant le retard moyen par mois pour chaque compagnie.

Indice : Le résultat devrait ressembler au graphe ci-dessous



Exercice 3 : Programmation

On s'intéresse ici à l'énigme de la suite de Syracuse, qui s'énonce de la manière suivante :

- on prend un nombre entier, noté N_0 ,
- si ce nombre est pair le diviser par 2, si ce nombre est impair le multiplier par 3 puis ajouter 1 pour obtenir N_1 ,
- recommencer l'opération précédente avec N_1 pour obtenir N_2 puis $N_3, N_4 \dots$ et ainsi de suite.

Remarquons que si lors de la $n^{ième}$ étape on obtient $N_n = 1$ alors la suite "boucle" : on a $N_{n+1} = 4$, $N_{n+2} = 2$, $N_{n+3} = 1$ puis les valeurs successives 4, 2 et 1 se répètent à l'infini.

10/ Ecrire une fonction qui prend en entrée un nombre entier quelconque, et qui calcule les 2,000 premières itérations de la suite. La fonction doit prendre en compte le fait que si le résultat obtenu à une itération est 1 il faut arrêter d'itérer (le reste de la suite étant connu). Le résultat final renvoyé par la fonction est donc un vecteur de taille au plus 2,000.

11/ Appliquer la fonction aux 1,000 premiers entiers.

Indice : Voici ce que doit renvoyer la fonction lorsqu'appliquée aux valeurs initiales 1, 2, 3, 4, 5 :

```
[[1]]
[1] 1

[[2]]
[1] 2 1
```

[[3]]

[1] 3 10 5 16 8 4 2 1

[[4]]

[1] 4 2 1

[[5]]

[1] 5 16 8 4 2 1

12/ Pour combien de valeurs d'initialisation observe-t-on que la fonction boucle ?

Réponse : pour l'ensemble des 1,000 initialisation ! En effet le nombre maximal d'itérations obtenu en partant avec une valeur initiale comprise entre 1 et 1,000 est 178. On n'atteint donc jamais les 2,000 itérations, signe que la fonction s'arrête avant parce que le chiffre 1 est obtenu lors d'une itération.