

# Les 12 Travaux d'AstéRix

Jean Souris

18/12/2020

## Contents

<b>Mes 5 critères permettant d'évaluer les travaux :</b>	<b>2</b>
<b>Partie 1 : Les travaux basés sur R :</b>	<b>3</b>
Travail N°1 : HAKAM Tarik et son Etude Ggplot2 ; . . . . .	3
Synthèse du travail . . . . .	3
Commentaire des parties intéressantes . . . . .	3
Evaluation selon mes critères . . . . .	4
Conclusion . . . . .	5
Travail N°2 : MARIE Théo et son Etude Caret ; . . . . .	5
Synthèse du travail . . . . .	5
Commentaire des parties intéressantes . . . . .	5
Evaluation selon mes critères . . . . .	6
Conclusion . . . . .	6
Travail N°3 : OBERTELLI Vincent et son Etude Caret ; . . . . .	6
Synthèse du travail . . . . .	6
Commentaire des parties intéressantes . . . . .	6
Evaluation selon mes critères . . . . .	7
Conclusion . . . . .	7
Travail N°4 : GARCIA Victor et son Etude Ggplot2 ; . . . . .	7
Synthèse du travail . . . . .	7
Commentaire des parties intéressantes . . . . .	8
Evaluation selon mes critères . . . . .	11
Conclusion . . . . .	11
Travail N°5 : LEANDRE Teddy et son Etude Caret ; . . . . .	12
Synthèse du travail . . . . .	12
Commentaire des parties intéressantes . . . . .	12
Evaluation selon mes critères . . . . .	14
Conclusion . . . . .	14

<b>Partie 2 : Les travaux sur les aspects mathématiques :</b>	<b>15</b>
Travail N°6 : SERVANT Thomas et son Etude rSymPy ; . . . . .	15
Synthèse du travail . . . . .	15
Commentaire des parties intéressantes . . . . .	15
Evaluation selon mes critères . . . . .	15
Conclusion . . . . .	16
Travail N°7 : MARIE Théo et son Etude rSymPy ; . . . . .	16
Synthèse du travail . . . . .	16
Commentaire des parties intéressantes . . . . .	16
Evaluation selon mes critères . . . . .	16
Conclusion . . . . .	17
Travail N°8 : OBERTELLI Vincent et son Etude Pracma ; . . . . .	17
Synthèse du travail . . . . .	17
Commentaire des parties intéressantes . . . . .	17
Evaluation selon mes critères . . . . .	19
Conclusion . . . . .	19
Travail N°9 : HAKAM Tarik et son Etude Pracma ; . . . . .	19
Synthèse du travail . . . . .	19
Commentaire des parties intéressantes . . . . .	20
Evaluation selon mes critères . . . . .	21
Conclusion . . . . .	21
Travail N°10 : ZHAO Wenjun et son Etude rSymPy . . . . .	21
Synthèse du travail . . . . .	21
Commentaire des parties intéressantes . . . . .	21
Evaluation selon mes critères . . . . .	22
Conclusion . . . . .	22

<b>Partie 3 : Auto-evaluation objective</b>	<b>22</b>
---	-----------

## Mes 5 critères permettant d'évaluer les travaux :

Voici les 5 critères qui me permettront d'évaluer les travaux de mes camarades (triés par ordre non-exhaustifs) :

- Mon premier critère sera l'aspect visuel du document, à savoir les différentes polices utilisés, mais aussi l'aspect "ordonné" de leurs documents ;
- Mon second critère sera sur les différentes fonctions utilisées dans les packages ;
- Mon troisième critère sera la qualité de l'explication des différentes fonctions et des packages, voir si l'étudiant a porté attention à ce que son document soit compris de tout le monde ;

- Mon quatrième critère sera la qualité des graphiques ou des expressions mathématiques LaTeX utilisés lors des études ;
- Mon dernier critère sera basé sur les potentiels bugs que peut contenir le document, voir si celui ci est bel et bien réalisable ou non.

---

## Partie 1 : Les travaux basés sur R :

### Travail N°1 : HAKAM Tarik et son Etude Ggplot2 ;

#### Synthèse du travail

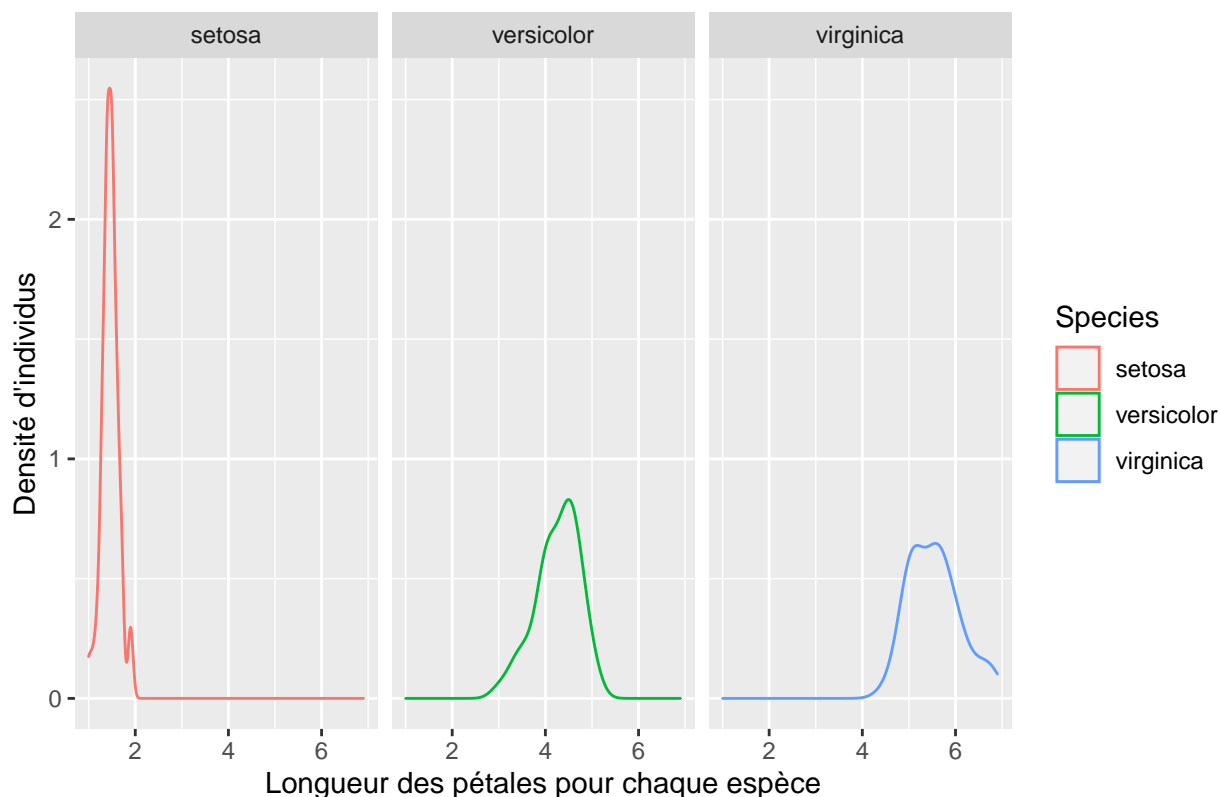
Dans son travail, Tarik nous montre différents types de graphiques que nous pouvons effectuer avec le package **ggplot2**. Il nous montre tout d'abord de quel dataset il va se servir. Par la suite, nous aurons l'occasion de voir 8 graphiques composés de fonctions du package **ggplot2**. Ensuite, je me contenterai d'énoncer les différentes fonctions utilisées par Tarik dans ses différents plots. Le premier graphique représente la longueur des sepals en fonction de celle des pétales ; celui-ci a été réalisé sans le package **ggplot2** mais en utilisant la fonction `plot`. Dans les 7 autres graphiques, nous aurons l'occasion de voir 5 fonctions du package en question. La première, **geom\_point()**, nous montre que l'on peut effectuer un graphique à base de nuage de points, mais Tarik ne s'est pas contenté de ça, il a en effet réussi à nous montrer les différentes espèces de plantes dans ce graphique en utilisant la fonction **aes()**, ce qui met en valeur en quoi ces espèces diffèrent. La seconde fonction qu'il a utilisé est **geom\_histogram()**, qui lui a permis de trier le nombre d'individu en fonction de la longueurs de leurs pétales. La troisième fonction se nomme **geom\_density()**, et comme son nom l'indique nous permet de distinguer la différence de densité entre chacune des espèces des plantes présentent dans sa base de données. La quatrième fonction utilisée est du domaine statistique, en nous montrant la répartition des longueurs des pétales en fonction des individus et en nous disant sur quel critère se pencher pour voir si un comportement est "anormal". Pour faire cela, Tarik a modélisé une boîte à moustache avec la fonction **geom\_boxplot()**. La cinquième et dernière fonction est **geom\_bar()** et modélise grâce à cela un histogramme.

#### Commentaire des parties intéressantes

La partie la plus intéressante de son travail se trouve, à mon humble avis, au niveau de la fonction "density" de son travail. Premièrement car cette fonction m'était encore inconnue, mais aussi car dans le dernier graphique, Tarik arrive à séparer les différentes espèces de plantes en 3 graphiques distincts grâce au terme **facet\_wrap()** comme montré dans son code :

```
library(ggplot2)
ggplot(iris, aes(x = Petal.Length, color = Species)) + geom_density() + facet_wrap(~ Species) +
  xlab("Longueur des pétales pour chaque espèce") + ylab("Densité d'individus") +
  ggtitle("Fig 7. Fonction de densité de la longueur des pétales par espèce")
```

Fig 7. Fonction de densité de la longueur des pétales par espèce



Ainsi, au lieu d'avoir les courbes superposées, qui dans un premier temps nous permet en effet de comparer les espèces telles quelles, mais dans un second temps rend plus difficile leur étude séparée, nous avons l'occasion de voir le comportement de chaque plante indépendamment les une des autres. Une autre partie de son étude qui m'a interpellé (même si elle n'est pas dans son exercice à proprement parler) est la qualité de sa biographie, non pas ce qu'il y cite, mais la manière dont il le fait :

@online{ggplot2, author = {Eric Matzner-Lober}, title = {Présentation de ggplot2}, url = {<https://calcul.math.cnrs.fr/attachments/spip/IMG/pdf/presentationggplot.pdf>} }

@online{CRAN, author = {Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington}, title = {Package 'ggplot2'}, year = {June, 2020}, url = {<https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf>} }

Elle ne s'affiche pas de la même manière que la sienne car dans ses métadonnées il a inclus :

“bibliography : gr02\_hakam\_tarik\_ggplot2\_bibliography.bib”

### Evaluation selon mes critères

En ce qui concerne mes critères cités plus haut, le visuel du document est globalement bon, l'étude est agréable à lire, malgré une ou deux fautes d'orthographe qui ne portent pas préjudice à sa qualité. Au niveau des fonctions utilisées, la diversité de celles-ci est belle et bien de mise puisque pas moins de 5 fonctions ont été illustrées dans cette étude, recouvrant ainsi une majorité des fonctions du package **ggplot2**. De plus, la qualité des graphiques allant de paire avec sa quantité permet une compréhension optimale des différentes parties de code qui pouvaient nous sembler floues. En plus de ses graphiques, les explications allant avec les fonctions sont claires, nous permettant de les reproduire à notre convenance. En ce qui concerne les potentiels “bugs” au niveau du code il n'y en a pas.

## Conclusion

Pour conclure, ce travail est globalement bon, puisqu'il nous permet de bien cerner ce qu'est le package **ggplot2** et ainsi nous pouvons en connaître les bases et les appliquer en suivant ses conseils.

## Travail N°2 : MARIE Théo et son Etude Caret ;

### Synthèse du travail

Dans ce travail, Théo nous montre les bases du package **Caret**. Il commence tout d'abord à nous expliquer à quoi sert **caret**, puis comment l'installer. Il nous explique ensuite lors d'un atelier comment nous pouvons obtenir le pourcentage de prédiction d'un algorithme en différentes étapes. La première étape est la mise en place de sa sélection et la mise en place d'un tableau de data, avec la fonction **createDataPartition**. Dans cette fonction, nous allons nous servir de 80% des données de ce tableau. Ensuite, Théo nous explique comment générer l'effet aléatoire de cet algorithme avec la fonction **set.seed**. Pour finir avec la mise en place de cet algorithme, Théo nous montre comment créer un dataset test avec le terme **trainIndex**. Il nous définit ensuite la cible sur laquelle il va se concentrer pendant cet exercice, puis il définit le modèle que nous allons utiliser : les arbres de décisions. Enfin, pour conclure sur le taux de fiabilité de l'algorithme, Théo va nous montrer qu'il "superpose" les tableaux trouvés avec ses arbres de probabilité et fait ensuite une règle de 3 pour nous en montrer la fiabilité.

### Commentaire des parties intéressantes

La première partie de code intéressante dans ce document est celle-ci :

```
library(caret)

## Loading required package: lattice

set.seed(333)
trainIndex <- createDataPartition(iris$Species, p = 0.8,
                                   list = FALSE,
                                   times = 1)
irisEntrainement <- iris[ trainIndex,]
irisTest <- iris[-trainIndex,]
```

Ici, Théo nous montre comment créer une variable qui aura pour proportion 80% du dataset qu'il utilise lors de son exercice. Nous voyons bien, comme il l'explique où sont les termes importants dans ce chunk.

La seconde partie intéressante de son code est celle-ci :

```
library(ggplot2)
library(lattice)
library(caret)
library(mlbench)
Theo <- train(Species ~ .,
              data = irisEntrainement,
              method = "rf",) ##il faut ici installer le package randomforest
```

Grâce au commentaire rajouté dans ce chunk, Théo nous informe que l'on a besoin d'un autre package afin de réussir à reproduire son exercice tout en utilisant la dataframe de départ.

## Evaluation selon mes critères

En ce qui concerne le visuel du document, globalement, Théo a bien fait attention à la police utilisée. Il utilise différentes couleurs afin de mettre certains termes en valeur, ce qui rend le document agréable à lire. Au niveau des différentes fonctions utilisées, bien que peu nombreuses, elles sont assez bien expliquées pour que l'on les comprenne et que l'on puisse reproduire l'exercice chez nous. Par contre, malheureusement il n'y a aucun graphique pour étayer ses propos, ce qui, pour les personnes plus "visuelles" peut compromettre la compréhension de son exercice.

## Conclusion

Pour conclure, Théo a bien expliqué l'exercice qu'il souhaitait nous montrer, le document est très lisible mais il manque de tableaux, ce qui fait défaut à son travail.

## Travail N°3 : OBERTELLI Vincent et son Etude Caret ;

### Synthèse du travail

Dans son étude, Vincent va nous montrer le principe de **Caret** et nous expliquer comment il l'utilise. Premièrement il va importer une base de données depuis un fichier .csv. Par la suite, il va afficher un tableau regroupant toutes les colonnes de sa dataframe, grâce à la fonction **str()**, afin de voir sur quels critères nous pourrions commencer la prédiction. Ensuite, grâce à la fonction **head()**, il va pouvoir nous afficher les 6 premières lignes de chaque colonne, ce qui est une méthode pour ordonner les données. Pour finir, il va créer une seed ainsi qu'une partition de données pour tester son algorithme. Malheureusement la fin de son exercice ne s'est pas montrée concluante puisqu'il nous annonce que les codes qu'il a essayé n'ont pas été concluant.

### Commentaire des parties intéressantes

Une des parties intéressante de son code se trouve au début, lors de l'importation de base de données depuis un fichier .csv :

```
library(caret)

# Import du dataset
orange <- read.csv('https://raw.githubusercontent.com/selva86/datasets/master/orange_juice_withmissing.
```

C'est une partie intéressante car personnellement je n'avais pas réussi à importer un fichier .csv sur RStudio ; ceci me montre donc quelle fonction utiliser, à savoir : **read.csv()**.

La seconde partie intéressante de son code est le moment où il ordonne son tableau avec la fonction **head()** :

```
# Structure de la dataframe
str(orange)

## 'data.frame':  1070 obs. of  18 variables:
## $ Purchase      : chr  "CH" "CH" "CH" "MM" ...
## $ WeekofPurchase: int   237 239 245 227 228 230 232 234 235 238 ...
## $ StoreID       : int    1 1 1 1 7 7 7 7 7 7 ...
## $ PriceCH       : num   1.75 1.75 1.86 1.69 1.69 1.69 1.69 1.75 1.75 1.75 ...
```

```
## $ PriceMM      : num  1.99 1.99 2.09 1.69 1.69 1.99 1.99 1.99 1.99 1.99 ...
## $ DiscCH       : num  0 0 0.17 0 0 0 0 0 0 0 ...
## $ DiscMM       : num  0 0.3 0 0 0 0 0.4 0.4 0.4 0.4 ...
## $ SpecialCH    : int   0 0 0 0 0 0 1 1 0 0 ...
## $ SpecialMM    : int   0 1 0 0 0 1 1 0 0 0 ...
## $ LoyalCH      : num  0.5 0.6 0.68 0.4 0.957 ...
## $ SalePriceMM  : num  1.99 1.69 2.09 1.69 1.69 1.99 1.59 1.59 1.59 1.59 ...
## $ SalePriceCH  : num  1.75 1.75 1.69 1.69 1.69 1.69 1.69 1.75 1.75 1.75 ...
## $ PriceDiff    : num  0.24 -0.06 0.4 0 0 0.3 -0.1 -0.16 -0.16 -0.16 ...
## $ Store7       : chr   "No" "No" "No" "No" ...
## $ PctDiscMM    : num  0 0.151 0 0 0 ...
## $ PctDiscCH    : num  0 0 0.0914 0 0 ...
## $ ListPriceDiff : num  0.24 0.24 0.23 0 0 0.3 0.3 0.24 0.24 0.24 ...
## $ STORE        : int   1 1 1 1 0 0 0 0 0 0 ...
```

```
# afficher du début à la 6ème ligne et les 10 colonnes
head(orange[, 1:10])
```

## Evaluation selon mes critères

En termes de visuel, le document de Vincent est agréable à lire, le titre est bien centré, tout comme l'auteur. Malheureusement il n'annonce pas de parties, même si cela n'influence pas la compréhension de son document. Au niveau des fonctions utilisées, Vincent utilise les fonctions basiques à l'élaboration d'une prédiction Caret, donc je n'ai rien à redire. De plus, les tableaux qu'il a affichés sont clairs, surtout celui affiché de manière réorganisée. Pour finir, comme il l'annonce en fin de document, il n'a pas réussi à afficher sa fonction correctement, mon dernier critère n'est donc pas rempli.

## Conclusion

Pour conclure, ce document nous montre bel et bien les bases afin d'arriver à accomplir une tâche **Caret**, avec toutes les fonctions nécessaires pour y parvenir, bien que son document soit incomplet, mais c'est le principe de l'apprentissage : tomber pour mieux se relever.

## Travail N°4 : GARCIA Victor et son Etude Ggplot2 ;

### Synthèse du travail

Dans son devoir, Victor nous explique le principe et l'utilité de **ggplot2** à travers une multitude de fonctions et de graphiques. Son devoir peut être décomposé en 2 parties : La première partie serait l'explication et la mise en situation des différentes fonctions de **ggplot2** telles que **geom\_point()**, **geom\_histogram()**, **geom\_boxplot()** ou encore **geom\_violin()**. Toutes ces fonctions sont expliquées de manière textuelles et graphiques afin que l'on puisse les reproduire sans trop de difficulté. La seconde partie de son devoir mêle le package **ggplot2** à celui de **dplyr** afin d'effectuer des corrélations plus ou moins complexe dans une dataframe et de les représenter graphiquement. Il nous explique dans un premier temps comment lire un fichier .csv et remodeler ses informations telles que le nom de ses colonnes afin qu'il soit plus lisible, ou encore le format de la date par exemple. Par la suite il filtrera ses données afin d'obtenir l'échantillon qu'il souhaite en utilisant une fonction de **dplyr** : **dplyr::filter()**. Enfin, il nous montrera dans sa dernière partie comment corréler différentes variables d'une dataframe en utilisant la matrice de corrélation et en nous en faisant un graphique.

## Commentaire des parties intéressantes

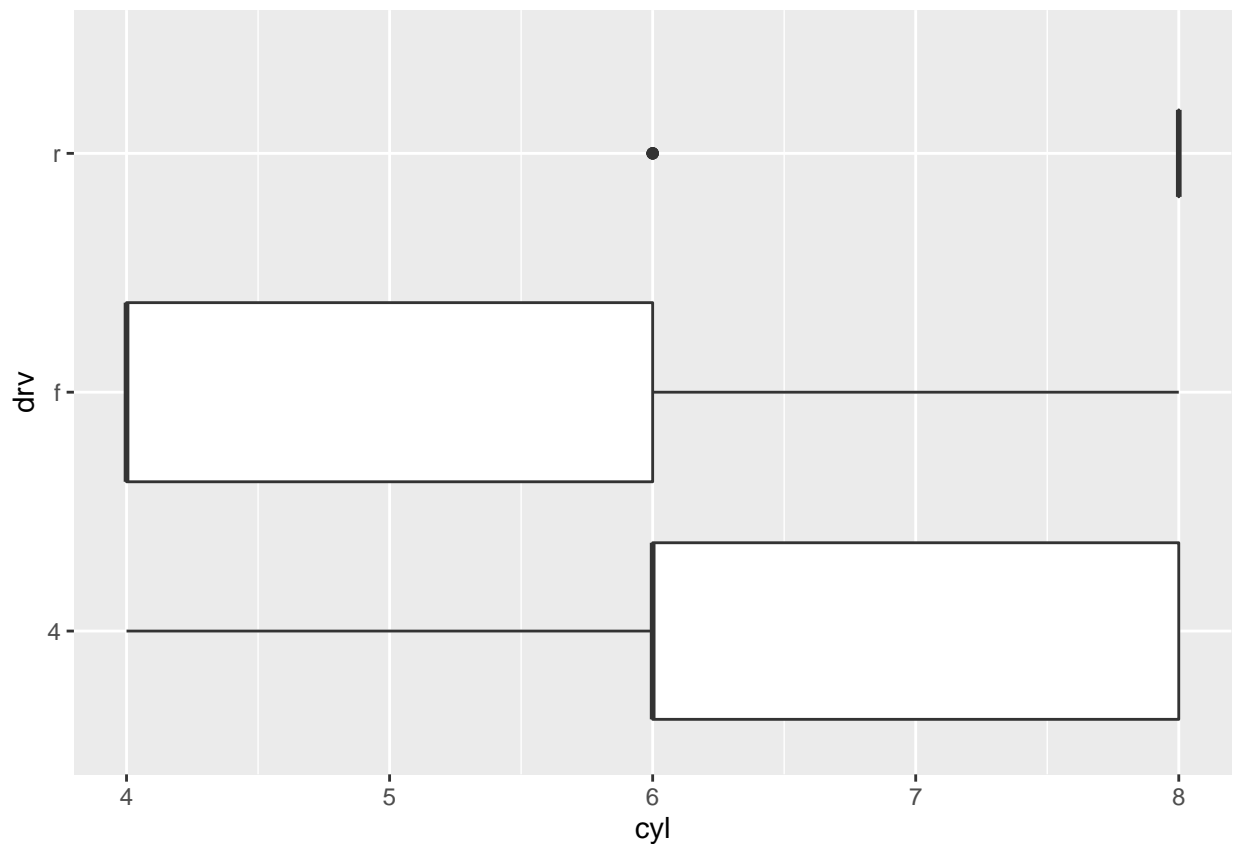
Pour commencer, je vais appeler les éléments dont j'aurais besoin afin que ses graphiques s'affichent :

```
library(ggplot2)
head(mpg)
```

```
data_covid<-read.table("donneesaplatcovid2.csv",header=TRUE,sep=';',dec='.')
#install.packages('igraph')
```

Son code présente plusieurs parties intéressantes. Pour commencer la première partie ou plutôt le premier graphique intéressant est sa variante de la boîte à moustache, avec la fonction **varwidth()** qu'il a rajouté afin d'obtenir ce résultat :

```
ggplot(mpg) + geom_boxplot(aes(x = cyl, y = drv), varwidth = TRUE)
```



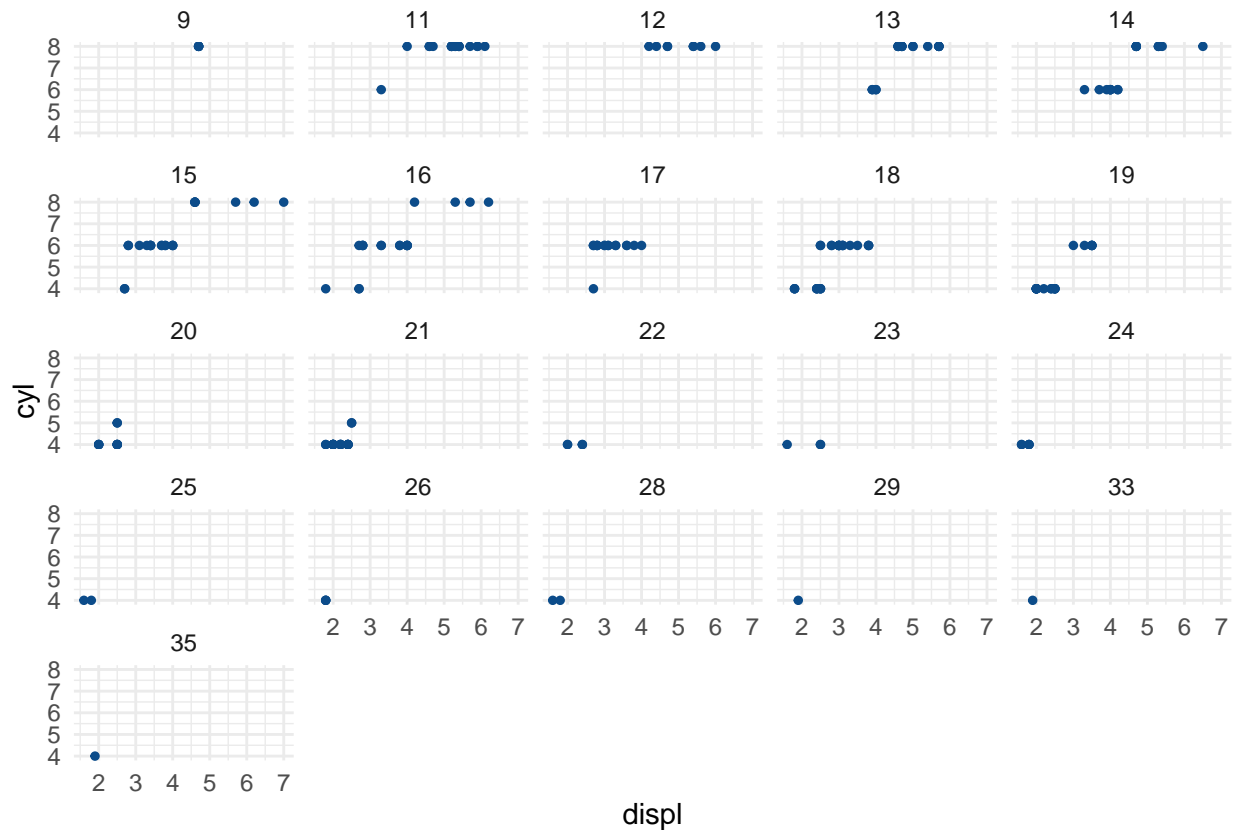
Cette fonction nous permet de varier les longueurs des boîtes à moustaches en fonction de leur effectif, afin que ce modèle soit encore plus représentatif de l'échantillon.

La seconde partie intéressante de son travail est lorsqu'il nous montre un graphique utilisant la fonction **facet\_wrap()** :

```
dataframe <- data.frame(mpg) #definition du DataFrame
ggplot(dataframe) +
  aes(x = displ, y = cyl) +
```



```
geom_point(size = 1L, colour = "#0c4c8a") +
theme_minimal() +
facet_wrap(vars(cty))
```



Comme on a pu le voir, cette fonction nous permet de séparer un graphique en plusieurs plus petits graphiques afin de voir le comportement individuel de chaque variable.

La dernière partie mais pas des moindres est lorsqu'il utilise la fonction **cor()** afin d'effectuer des corrélations entre différentes variables d'une dataframe, puis lorsqu'il crée une matrice à partir de cela :

```
cor(mpg$hwy,mpg$cty)
```

```
## [1] 0.9559159
```

```
df <- data.frame(mpg$displ,mpg$cyl,mpg$cty,mpg$hwy)#creation d'un DataFrame avec les variables numériques
mc <- cor(df)#utilisation de ma fonction correlation pour obtenir la matrice de correlation
mm <- as.matrix(mc* mc) # La matrice au carée permet de n'avoir que des valeurs positives.
print(mm)
```

```
##          mpg.displ  mpg.cyl  mpg.cty  mpg.hwy
## mpg.displ 1.0000000 0.8653225 0.6376405 0.5867867
## mpg.cyl   0.8653225 1.0000000 0.6492676 0.5805104
## mpg.cty   0.6376405 0.6492676 1.0000000 0.9137752
## mpg.hwy   0.5867867 0.5805104 0.9137752 1.0000000
```

Après avoir obtenu sa matrice, il va filtrer les données supérieures à 0.6 afin de faciliter les interprétations comme il l'explique.

```
mm[mm>0.7] <- 1
mm[mm<=0.7] <- 0
print(mm)
```

```
##           mpg.displ mpg.cyl mpg.cty mpg.hwy
## mpg.displ         1         1         0         0
## mpg.cyl           1         1         0         0
## mpg.cty           0         0         1         1
## mpg.hwy           0         0         1         1
```

Après avoir créé sa matrice, il va en faire une représentation graphique pour le moins surprenante :

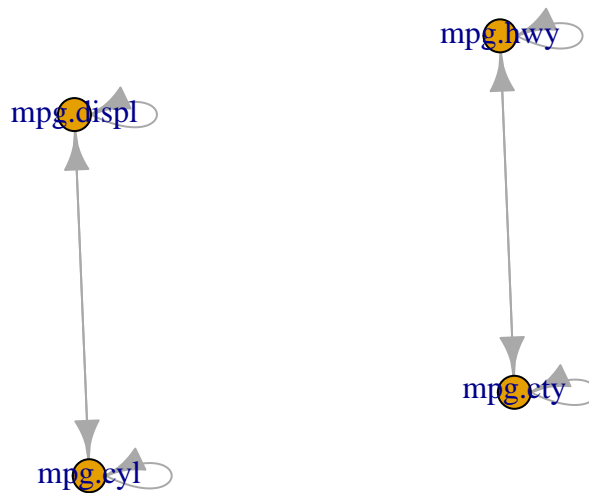
```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##   decompose, spectrum

## The following object is masked from 'package:base':
##
##   union
```

```
network <- graph_from_adjacency_matrix(mm)
plot(network)
```



Il m'est impossible de citer son travail sans aussi en citer sa Source principale.

### Evaluation selon mes critères

Maintenant passons à l'évaluation selon mes critères :

Visuellement, son PDF paraît basique de par les polices utilisées, qui ne sont pas spécialement agréables à lire. Bien qu'il ait tout de même trié son document en utilisant des titres ainsi que des sous-titres. Au niveau des fonctions utilisées, son document en est très riche, puisqu'il nous montre les fonctions "basiques" que l'on peut utiliser dans **ggplot2** en en poussant le potentiel, comme par exemple avec la boîte à moustache présentée dans la partie précédente. De plus, il ne se contente pas simplement de nous montrer des fonctions de **ggplot2**, mais il nous en montre aussi des packages **dplyr** ainsi que "igraph", ce qui rend son document encore plus consistant. Au niveau de ses explications, dans son étude l'élève a réussi à "vulgariser" des notions qui pourraient paraître obscures pour certains étudiants. Il explique donc très bien ces notions, en en donnant des exemples clairs et en énumérant toutes les étapes pour y parvenir pour pouvoir les reproduire chez soi. Au niveau de ses graphiques, ils sont très bien réalisés, puisqu'il en modifie les termes pour montrer les différentes variantes qu'ils peuvent prendre ainsi les différentes fonctions que nous pouvons y incorporer pour les rendre encore plus riches et précis. Pour finir, son code ne présente aucune erreur apparente, puisqu'il s'exécute de manière tout à fait normale sur RStudio, ce critère est donc rempli.

### Conclusion

Pour conclure, cette étude est bien poussée, ce qui rend un travail globalement bon qui serait encore meilleur en faisant attention à son esthétique.

## Travail N°5 : LEANDRE Teddy et son Etude Caret ;

### Synthèse du travail

Dans son travail, Teddy va nous montrer son usage de **Caret** sur une base de données sur les composants d'une voiture. Il va tout d'abord installer puis charger le package "Caret" ainsi que son dataset "cars" avec la fonction `data()`. Par la suite, après avoir élaborer sa seed, il va créer une partition de données avec la fonction `createDataPartition()`. Il va par la suite nous montrer la contenu de sa table avec la fonction `str()`, qui nous montre donc les 18 variables de sa base de données. Il va dans une troisième étape établir une partition de test et d'entraînement. Après il nous montre qu'il va utiliser la méthode d'arbre de decision random forest (rf) afin d'exécuter ses tests tout en mesurant le temps que la requête va prendre avec la fonction `Sys.time()`. Enfin, grâce au package "Ggplot2", Teddy va nous montrer l'écart qu'il peut y avoir entre la prédiction du prix des voitures et leur véritable valeur.

### Commentaire des parties intéressantes

Une des parties intéressante de son code est lorsqu'il vérifie le contenu de sa base de donnée grâce à la fonction `str()` :

```
carsTrain <- cars[trainIndex,]
carsTest  <- cars[-trainIndex,]
data("cars")
str(cars)
```

```
## 'data.frame':      804 obs. of  18 variables:
## $ Price       : num  22661 21725 29143 30732 33359 ...
## $ Mileage     : int   20105 13457 31655 22479 17590 23635 17381 27558 25049 17319 ...
## $ Cylinder    : int    6 6 4 4 4 4 4 4 4 4 ...
## $ Doors       : int    4 2 2 2 2 2 2 2 2 4 ...
## $ Cruise      : int    1 1 1 1 1 1 1 1 1 1 ...
## $ Sound       : int    0 1 1 0 1 0 1 0 0 0 ...
## $ Leather     : int    0 0 1 0 1 0 1 1 0 1 ...
## $ Buick       : int    1 0 0 0 0 0 0 0 0 0 ...
## $ Cadillac    : int    0 0 0 0 0 0 0 0 0 0 ...
## $ Chevy       : int    0 1 0 0 0 0 0 0 0 0 ...
## $ Pontiac     : int    0 0 0 0 0 0 0 0 0 0 ...
## $ Saab        : int    0 0 1 1 1 1 1 1 1 1 ...
## $ Saturn      : int    0 0 0 0 0 0 0 0 0 0 ...
## $ convertible: int    0 0 1 1 1 1 1 1 1 0 ...
## $ coupe       : int    0 1 0 0 0 0 0 0 0 0 ...
## $ hatchback   : int    0 0 0 0 0 0 0 0 0 0 ...
## $ sedan       : int    1 0 0 0 0 0 0 0 0 1 ...
## $ wagon       : int    0 0 0 0 0 0 0 0 0 0 ...
```

La seconde partie intéressante est lorsqu'il calcul le temps que va mettre sa requête à s'élaborer après avoir choisi sa méthode de randomisation :

```
set.seed(300)

trainIndex <- createDataPartition(cars$Price, p = .8,
                                   list = FALSE,
                                   times = 1)
```

```
carsTrain <- cars[trainIndex,]
carsTest <- cars[-trainIndex,]

fitControl <- trainControl(## 5-fold CV
  method = "repeatedcv",
  number = 5,

  repeats = 5)
set.seed(300)
t_before <- Sys.time()
Fit1 <- train(Price ~ ., data = carsTrain,
  method = "rf", # methode d'arbre de décision random forest(rf), on peut tester "nnet" aus
  trControl = fitControl,
  verbose = FALSE)
t_after <- Sys.time()
duree <- t_after - t_before
print(duree)
```

```
## Time difference of 40.03601 secs
```

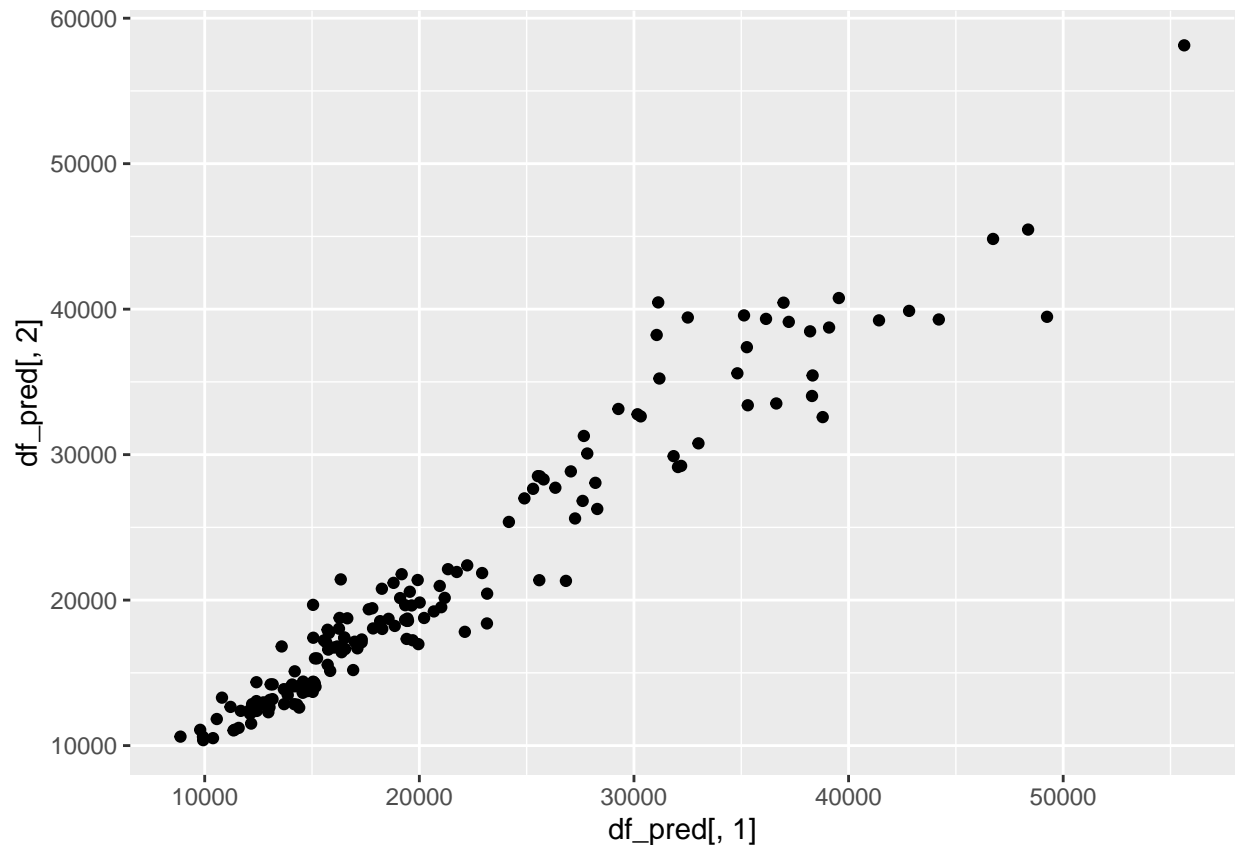
C'est interessant de voir combien de temps met une requête à s'exécuter, car cela varie en fonction du nombre de données traitées ainsi que la complexité de la tâche demandée.

Enfin, la dernière partie intéressante de son étude est lorsqu'il montre, grâce à un nuage de points les différents résultats des prédictions qu'il a effectué :

```
prediction_carsTest <- predict(Fit1, carsTest)
df_pred <- data.frame(carsTest$Price,prediction_carsTest )
ecart <- df_pred[,1] == df_pred[,2]

library(ggplot2)

qplot(df_pred[,1],df_pred[,2])
```



### Evaluation selon mes critères

En termes de visuel, le PDF n'est pas désagréable à regarder, car le titre n'est pas trop gros et il y a des couleurs. En revanche, ces dites couleurs apparaissent car son document est composé de 2 chunks R ; il n'y a pas de titre ni de sous-titre. Malgré cela, les idées restent ordonnées grâce aux commentaires qu'il dispose sur certaines d'entre-elles pour expliquer son fil de pensée. Ce qui me fait passer à son explication lors de son travail ; Ici, Teddy nous donne les étapes afin de réussir à reproduire son code et en variant les données, malheureusement certains termes importants ne sont pas expliqués, ce qui pourrait laisser un lecteur commun un peu dans le brouillard. Au niveau de ses graphiques, ils sont clairs et expriment bien ce que Teddy voulait nous montrer lors de son étude à savoir la différence flagrante entre la prédiction du prix d'une voiture et son prix réel.

Au niveau des potentiels bugs, il n'y en a pas de flagrant, ce critère est donc rempli.

### Conclusion

Pour conclure, le travail de Teddy atteint son but, qui est de montrer une méthode de prédiction en utilisant la méthode d'arbre de décision "random forest", mais il devrait faire plus attention à la structure de son devoir qui pourrait lui porter préjudice.

## Partie 2 : Les travaux sur les aspects mathématiques :

### Travail N°6 : SERVANT Thomas et son Etude rSymPy ;

#### Synthèse du travail

L'étude de Thomas consiste globalement à nous montrer le potentiel de la librairie **rSymPy**. Il nous explique qu'avant toute chose il faut télécharger bons nombres de package afin de pouvoir executer les calculs à venir. Ensuite, Thomas nous montre différents types de fonctions que l'on peut executer telle que la création de variable, et les calculs simples qui en découlent, le calcul des limites d'une expression, pour voir vers quelle valeur tend "x" par rapport à sa fonction associée, mais encore il nous montre que la dérivation de tous types de fonctions est possible sur **rSymPy**. De plus nous pouvons voir que nous pouvons simplifier, développer ou encore factoriser une expression grâce aux fonctions respectives `sympy("simplify()")`, `sympy(expand())` et `sympy("factor()")`. Outre les calculs d'avant, il nous montre que nous pouvons obtenir par exemple n'importe quelle décimale de  $\pi$  grâce à la fonction `sympy("pi.evalf()")`.

#### Commentaire des parties intéressantes

Une partie intéressante de son code est lorsqu'il nous montre les décimales de  $\pi$  :

```
library(rJava)
library(rJython)
```

```
## Loading required package: rjson
```

```
library(rSymPy)
sympy("pi.evalf(120)") #nous permet d'afficher les 120 premières décimales de Pi
```

```
## [1] "3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342"
```

Grâce à cette expression nous pouvons donc obtenir autant de décimale de  $\pi$  que l'on souhaite, ce qui montre le potentiel de **rSymPy**.

Une seconde partie intéressante est lorsqu'il résoud une equation du second degré comme ceci :

```
#sympy("solve(x**2 - 2, x)") #on résout ici x^2-2=0
```

La fonction `sympy("solve()")` peut s'avérer très pratique dans bons nombres cas de figure.

Enfin, une dernière partie intéressante de son étude a été le développement d'une expression, qui peut s'avérer très pratique :

```
#sympy("expand((x + 2)*(x - 3))")
```

#### Evaluation selon mes critères

En termes de visuel, le PDF de Thomas est structuré, avec un titre ainsi que des sous-titres, ce qui permet de bien voir les différentes parties qui composent son devoir, néanmoins la police n'a pas été soignée, ce qui rend le visuel moins agréable à lire. Au niveau des fonctions utilisées, elles sont multiples et bien illustrées par les chunk qu'il a incorporé à son code, mais, même si cela ne semble pas nécessaire vu la complexité des calculs, très peu voire pas d'explications sont fournies, ce qui pourrait rendre confus un lecteur lambda. De plus, malheureusement, aucune expression LaTeX n'a été représenté, ce qui aurait peut être aidé à mieux visualiser les fonctions utilisées. Pour finir, aucun bug n'est apparent dans le code ni aucun message ne s'affiche dans le PDF, il n'y a donc pas de soucis à ce niveau là.

## Conclusion

Pour conclure, ce travail survole bien les bases des calculs et expressions que l'on peut obtenir sur **rSymPy**, malgré l'absence que formules LaTeX qui se fait ressentir.

## Travail N°7 : MARIE Théo et son Etude rSymPy ;

### Synthèse du travail

Dans son travail, Théo nous explique les bases de **rSymPy** ainsi que son utilité. Tout d'abord, il nous liste les packages que nous devons installer pour le bon fonctionnement de notre code. Il nous liste ensuite des fonctions que nous pouvons réaliser sur **rSymPy** telles que des fonctions logarithme, mais aussi les fonctions trigonométriques telles que cosinus et sinus, mais encore des simplifications de nombres décimaux avec la fonction `sympy("nsimplify()")`, et même des limites de fonctions avec `sympy("limit()")`.

### Commentaire des parties intéressantes

Les parties intéressantes du devoir de Théo sont son introduction, qui nous explique bien le but de **rSymPy**, et les étapes à effectuer afin de l'appeler correctement et donc effectuer nos calculs sans soucis, tel que créer une variable par exemple.

La seconde partie intéressante est lorsqu'il nous montre que l'on peut simplifier des expressions grâce à ce package tel que :

```
library(rSymPy)
sympy("nsimplify(4.242640687119286)")
```

```
## [1] "3*2**(1/2)"
```

Ici, nous pouvons donc voir que ce chiffre a été simplifié grâce à la fonction `sympy("nsimplify()")`.

De plus, il nous montre que nous pouvons calculer des Sinus ou Cosinus en utilisant simplement `sympy()` :

```
library(rSymPy)
sympy("var('x')")
```

```
## [1] "x"
```

```
sympy("y = log(x)")
```

```
## [1] "log(x)"
```

```
sympy("y = x <- 0.5 + 0.25*sin(x) + 1/3*cos(x) - 1/3*sin(2*x) - 0.25*cos(2*x)")
```

```
## [1] "x < -0.5 + 0.25*sin(x) - 0.25*cos(2*x)"
```

### Evaluation selon mes critères

En termes de visuel, le travail de Théo est bien organisé, ce qui facilite la lecture de son document. En revanche, les fonctions utilisées sont peu nombreuses, donc les explications qu'il peut y apporter sont limitées, tout comme les graphiques ou autres expressions LaTeX qui sont absentes. En revanche, aucun bug n'est présent, ce qui remplit mon dernier critère.



## Conclusion

Pour conclure, bien que son devoir soit bien présenté, il reste peu riche en fonctions, je ne peux donc pas bien voir les autres facettes de ce package, bien qu'il m'ait expliqué correctement comment l'installer et le faire fonctionner.

## Travail N°8 : OBERTELLI Vincent et son Etude Pracma ;

### Synthèse du travail

Dans son devoir, Vincent nous montre comment installer le package **pracma** pour commencer, puis il décide directement de nous montrer le potentiel de ce package en nous montrant une rosace pour le moins surprenante.

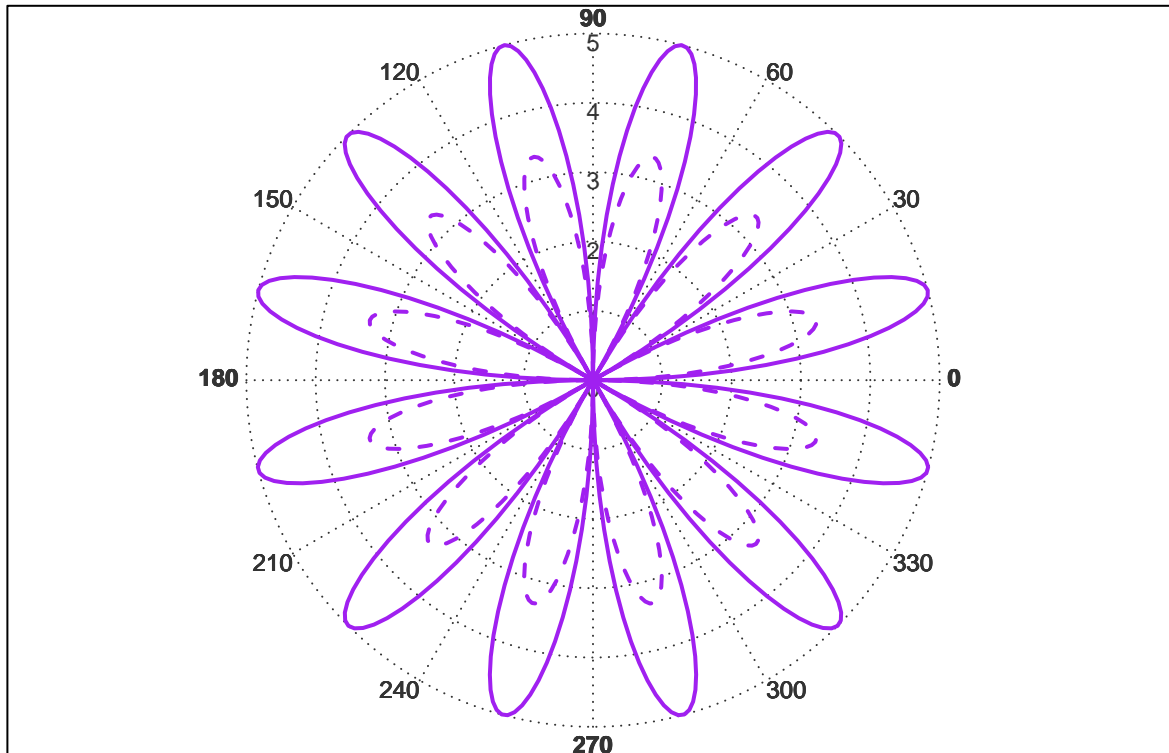
Par la suite, il nous listera différentes fonctions issu de sa même source et y décrivant chaque arguments. Il commence par la **fonction abm3pc()** qui est une fonction d'approche prédictive, puis nous la montre visuellement dans un graphique. Il nous montre ensuite la fonction **agmean()**, qui nous permet de calculer la moyenne, suivi de **accumarray()**, qui permet d'effectuer une fonction de groupe sur un dataset en regroupant les différentes données. Enfin, il décide de nous montrer les fonctions **and** et **or**, en voulant les comparer avec celles de Python qui peut nous sembler familière et en conclue que ces fonctions n'ont pas la même finalité.

### Commentaire des parties intéressantes

La première partie que j'ai trouvé intéressante a été la rosace :

```
library(pracma)
t=seq(0,2*pi, length=360)
f=function(t){5*sin(6*t)}
par(mar=c(1,1,2,1))
polar(t,f(t),grcol="gray20",bxcol="black",col="purple",
lwd=2,main=expression(5*sin(6*theta)~ ~"and"~ ~
2(5*sin(6*theta))/3))
polar(t,2*f(t)/3,col="purple",lty=2,lwd=2,add=TRUE)
```

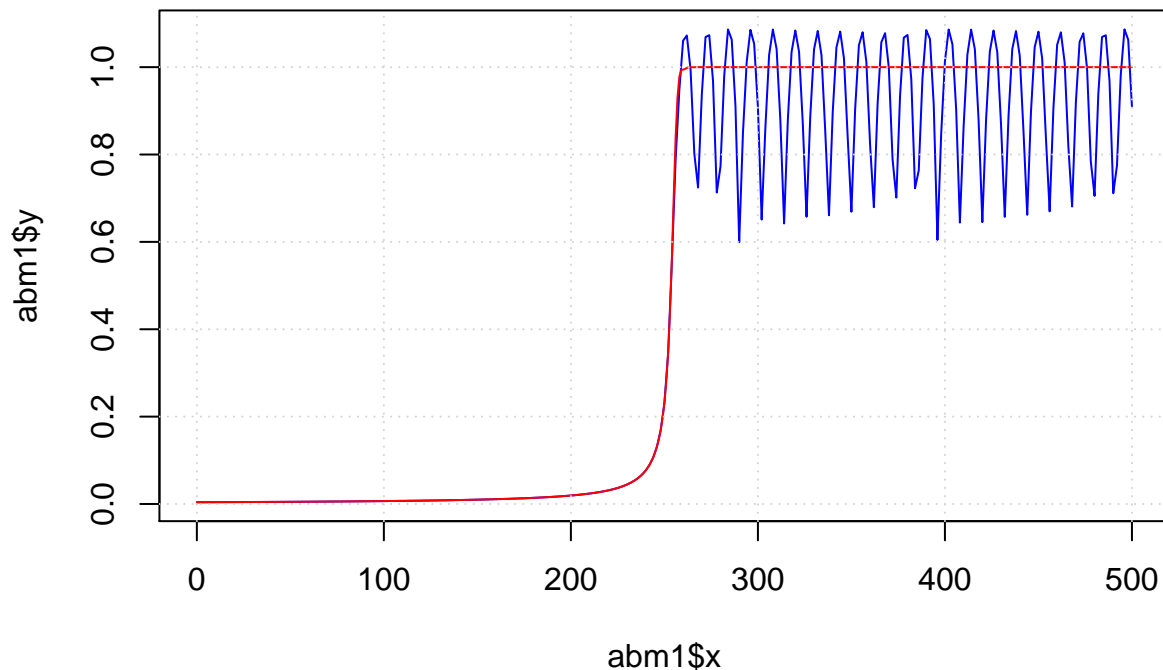
$$5\sin(6\theta) \text{ and } 2(5\sin(6\theta))/3$$



Cette rosace montre la complexité ainsi que le potentiel du package **pracma** en termes d'expression mathématiques.

La seconde partie intéressante de son devoir est la fonction **abm3pc()**, et plus précisément la graphique qui en découle :

```
library(pracma)
# y' = y^2 - y^3, y(0) = d, 0 <= t <= 2/d, d = 0.01
f <- function(t, y) y^2 - y^3
d <- 1/250
abm1 <- abm3pc(f, 0, 2/d, d, n = 1/d)
abm2 <- abm3pc(f, 0, 2/d, d, n = 2/d)
## Not run:
plot(abm1$x, abm1$y, type = "l", col = "blue")
lines(abm2$x, abm2$y, type = "l", col = "red")
grid()
```



Encore une fois, cette fonction nous montre que nous pouvons modéliser et calculer des expressions d'une certaine complexité et avec un certain potentiel prédictif, ce qui s'avère très utile de nos jours.

### Evaluation selon mes critères

En termes de visuel, le titre est bien centré par rapport au document, mais au niveau de la structure il y a certaines lacunes, car ses parties ne sont pas à proprement parler séparées par des titres, bien que le document reste assez clair à lire. Au niveau de la qualité des fonctions utilisées, Vincent nous a présenté des fonction un peu poussées, mais assez compréhensible pour qu'il puisse nous les expliquer et en expliquer chaque termes, ce qui nous aide tout au long de son devoir. Pour les graphiques, il nous montre là aussi des graphiques originaux, qui sont reproductible grâce à la définition de chacun de ces termes par Vincent. Le document ne présente pas de bugs apparant, ce critère est donc rempli sans soucis.

### Conclusion

Pour conclure, le travail de Vincent est bon mais présente certaines lacunes telles que le manque de structure, et aussi un léger manque d'explication sur le premier graphique qu'il nous montre.

## Travail N°9 : HAKAM Tarik et son Etude Pracma ;

### Synthèse du travail

Dans son etude, Tarik va nous exposer 2 fonctions mathématiques : la **Quadrature Hermite-Gauss** ainsi que l'**Approximation de Tchebychev**. La première est utilisée pour calculer des intégrales, qui est le

volume présent sous une fonction sur un intervalle pré-défini ; il nous explique alors comment la calculer avec le langage R, puis arrive à nous démontrer que cette intégrale est égale à la racine carré de  $\pi$ . Puis, afin d'aller plus loin Tarik a essayé de nous montrer l'usage de Tchebychev, qui globalement traite des polynômes. Pour étayer ses propos, Tarik nous montre une évaluation polynomiale faite sur R en approximant  $\sin(x)$  avec un polynôme. Afin d'effectuer son approximation, 2 étapes vont se dérouler : - Calcul de l'écart entre la fonction  $\sin(x)$  et le polynôme ; - Calcul de l'écart entre la fonction  $\sin(x)$  et la fonction de Tchebychev. En comparant les résultats de ces 2 calculs, il nous démontre que l'approximation de Tchebychev est plus efficace que celle du polynôme. Il finit son étude en nous montrant la fonction sinusoidale dessinée.

## Commentaire des parties intéressantes

Cette étude comporte plusieurs parties qui ont retenues mon attention.

Premièrement ce fût la richesse et la complexité des fonctions LaTeX qu'il a retranscrit, telles que :

$$\int_{-\infty}^{\infty} \cos(x) \exp(-x^2) dx = \frac{\sqrt{\pi}}{\exp(1)^{1/4}} \quad (1)$$

Mais encore, et l'expression la plus "impressionnante" de son devoir à mon avis :

$$w_i = -\frac{A_{n+1}\gamma_n}{A_n H_n^{(x_i)} H'_{n+1}(x_i)} = \frac{A_n}{A_{n-1}} \frac{\gamma_{n-1}}{H_{n-1}(x_i) H'_n(x_i)} \quad (2)$$

Cette fonction, de part ses différents termes et sa récurrence a dû mettre du temps à être composée.

Enfin, la dernière que j'ai sélectionnée et pas des moindres :

$$w_i = -\frac{2^{n+1}n!\sqrt{(\pi)}}{H_{n+1}(x_i)H'_n(x_i)} = \frac{2^n(n-1)!\sqrt{(\pi)}}{H_{n-1}(x_i)H'_n(x_i)} = \frac{2^{n+1}n!\sqrt{(\pi)}}{[H'_n(x_i)]^2} = \frac{2^{n+1}n!\sqrt{(\pi)}}{[H_{n+1}(x_i)]^2} = \frac{2^{n-1}n!\sqrt{(\pi)}}{n^2[H_{n-1}(x_i)]^2} \quad (3)$$

Néanmoins, il n'y a pas seulement les fonctions LaTeX que j'ai trouvées remarquables, il y a aussi les calculs d'intégrales qu'il effectue, tels que :

```
library("pracma")
f <- gaussHermite(17)

# Integrate x^2 exp(-x^2)
sum(f$x * f$x^2) #=> 0.88622692545276 == sqrt(pi)/2
```

```
## [1] 0.8862269
```

```
# Integrate cos(x) * exp(-x^2)
sum(f$x * cos(f$x)) #=> 1.38038844704314 == sqrt(pi)/exp(1)^0.25
```

```
## [1] 1.380388
```

## Evaluation selon mes critères

Mon premier critère étant l'aspect visuel de son devoir, je n'ai rien à redire, puisque Tarik a bien ordonnées les différents sujets qu'il présente, le titre du document est centré et mis en valeur, et sa bibliographie est bien faite. Au niveau des fonctions utilisées, Tarik n'a pas choisi les plus faciles, ce qui est tout à son honneur, néanmoins leurs explications, ou du moins pour la Quadrature Hermite-Gauss laisse à désirer, car des recherches supplémentaires sur Internet sont de mise si l'on souhaite comprendre pleinement son utilité et son fonctionnement. Au niveau de la qualité de ses graphiques ou expression LaTeX, elles sont très bien, et nombreuses, je n'ai donc rien à redire à ce niveau là, tout comme les potentiels bugs qui sont absents de son travail.

## Conclusion

Pour conclure, Tarik ne s'est pas laissé les fonctions les plus faciles à nous montrer, et les expose très bien au biais de ses expressions LaTeX, mais leurs explications pèchent, ce qui peut porter préjudice à son devoir.

## Travail N°10 : ZHAO Wenjun et son Etude rSymPy

### Synthèse du travail

Dans son travail, Wenjun nous présente les bases du package **rSymPy**, ainsi que son mode d'installation. Puis Wenjun nous montre les possibles créations que **rSymPy** permet telles que la création de variable, avec la fonction `sympy("var()")`, puis les calculs possibles en y incluant ces variables. Elle nous montre également que nous pouvons effectuer des calculs trigonométriques mais aussi de limites de fonctions et d'intégrale. Enfin, Wenjun nous montre comment s'affiche une matrice et nous apprend qu'elle ne s'affiche pas comme les matrices dans R, qui elles sont en colonne.

### Commentaire des parties intéressantes

Les parties intéressantes de son code apparaissent lorsque Wenjun nous montre quels packages ils faut installer, puis lorsqu'elle effectue des calculs cités plus haut tels que :

Un calcul d'intégral :

```
library(rSymPy)
# create a SymPy variable called x
sympy("var('x')")
```

```
## [1] "x"
```

```
sympy("integrate(exp(-x), (x, 0, oo))")
```

```
## [1] "1"
```

ou encore un calcul de limite de fonction :

```
sympy("limit(1/x, x, oo)")
```

```
## [1] "0"
```

Sa création de matrice reste la partie la plus intéressante de son travail à mon avis :

```
sympy("y = x*x")
```

```
## [1] "x**2"
```

```
sympy("y")
```

```
## [1] "x**2"
```

```
cat(sympy("A = Matrix([[1,x], [y,1]])"), "\n")
```

```
## [ 1, x]
```

```
## [x**2, 1]
```

```
cat(sympy("A**2"), "\n")
```

```
## [1 + x**3, 2*x]
```

```
## [ 2*x**2, 1 + x**3]
```

### Evaluation selon mes critères

En termes de visuel, le document de Wenjun est agréable à lire, le titre est centré et la police soignée. De plus, elle nous montre différentes fonctions basiques dans R. Au niveau des explications, elle nous explique bien comment installer le package **rSymPy** ainsi que tous ceux qui y sont corrélés tels que **rJython** ou encore **rJava**, et n'oublie pas de nous raconter qu'elle a réussi à corriger une erreur dans son code avec 'output=FALSE" au cas où nous rencontrions le même à un moment donné. Malheureusement son document ne comporte pas de graphique ni d'expression LaTeX, ce qui est dommage pour un devoir portant sur un package basé sur des expressions mathématiques. Enfin, en termes de potentiels bug, Wenjun nous alerte d'un bug mineur qui l'empêchait de kniter son document correctement et a donc mit une expression en mode commentaire, elle a donc résolu le problème "à la source" si je puis dire.

### Conclusion

Pour conclure, le devoir de Wenjun est un devoir correct pour commencer à survoler les bases de **RSympy**, mais manque de structure tels que des sous-titres entre les fonctions utilisées ainsi que d'expressions LaTeX qui auraient été un réel plus à son devoir.

---

## Partie 3 : Auto-evaluation objective

Voici les 2 travaux que je vais évaluer : Mon étude Ggplot2 ainsi que mon étude Pragma. (Vous pouvez retrouver tous mes autres travaux, dont celui-ci sur mon GitHub )

Après avoir lu en détail 10 autres travaux de mes camarades, j'ai pu me faire une idée approximative que ce qu'était un travail complet sur tous les points, que ce soit visuel, pédagogique ou encore poussé.

- Je vais donc évaluer objectivement mon travail **ggplot2**, puis “**pracma**” :

En ce qui concerne mon travail “**ggplot2**”, j’explique avec plus ou moins de précision les fonctions que je souhaite montrer à savoir “**geom\_boxplot**” et “**geom\_violin**”, mais je n’explique pas l’usage que l’on peut faire du package “**ggplot2**”, ni les étapes à réaliser pour l’installer. Bien que ces étapes soient présente je ne les explique pas, ce qui peut laisser un lecteur lambda dans l’incompréhension. De plus, l’esthétique de mon document laisse à désirer, bien que les idées soient structurées, le document n’est pas si agréable à lire. En revanche, au niveau des fonctions que j’ai choisi, ou du moins la fonction “**geom\_boxplot**”, j’explique de façon assez claire ce que représente cette fonction et explique quelques variantes que celle-ci peut prendre afin qu’un lecteur puisse en modifier les termes sans être perdu. Pour ma fonction “**geom\_violin**”, bien que j’en explique la signification, je me suis contenté d’en modifier les couleurs alors que biens d’autres arguments plus “fonctionnels” étaient disponible. En somme mon travail ne peut pas être considéré comme mauvais, car j’ai expliqué au mieux l’usage et la signification des fonctions, mais j’aurais pu m’attarder un peu plus sur les détails de type visuels pour l’ensemble du document et technique sur la fonction “**geom\_violin**”.

- Je vais maintenant parler de mon document sur le package **pracma** :

En termes de visuel, tout comme mon précédent devoir, bien que les idées soient structurées pour ne pas perdre le lecteur, je ne lui ai accordé que trop peu d’importance. En revanche, j’ai pris plus le temps d’expliquer en introduction l’utilité que pouvait apporter ce package, bien que je ne décrive pas toutes les étapes de son installation même si le code y est présent. Au niveau de l’exercice que j’ai décidé de présenter, je pense avoir bien détaillé toutes les étapes, des différents calculs, en ayant inclu différentes expressions LaTeX (bien que les 2 premières ne se sont pas insérées dans le texte comme je le souhaitais). Cependant, je me suis rendu compte que détailler toutes ces étapes était certes essentiel à la compréhension de l’exercice, mais je n’ai malheureusement pas toujours poussé mon analyse, au niveau du comportement de la fonction par exemple.

Pour conclure, mes 2 travaux sont loin d’être parfait, puisqu’il y a un très net manque d’attention porté sur les détails, mais j’estime qu’ils restent compréhensible pour tous types de lecteurs car un aspect assez pédagogique y est présent.

---

Mes Sources supplémentaires :

Je tenais à remercier :

- Le tuto de monashbioinformaticsplatform ainsi que Bookmark.org qui m’ont aidé à la mise en page de ce document ;
- Liu Jiayue et son tuto sur les formes de LaTeX ;