TD1\_bases\_R

2023-10-11

# Exemple de fonction

# Fonction-----------  
  
# function with various descriptive statistics  
fxdescribe = function(x){  
 c(obs = length(x),  
 missing = sum(is.na(x),na.rm=T),  
 min = min(x,na.rm=T),  
 max = max(x,na.rm=T),  
 median = median(x,na.rm=T),  
 q1 = quantile(x,na.rm=T,c(.25)),  
 q3 = quantile(x,na.rm=T,c(.75)),  
 mean = mean(x,na.rm=T),  
 sd = sd(x,na.rm=T),  
 `95lci` = mean(x,na.rm=T)-(sd(x,na.rm = T)\*1.96/sqrt(length(x))),  
 `95hci` = mean(x,na.rm=T)+(sd(x,na.rm = T)\*1.96/sqrt(length(x)))  
 )  
}  
# e.g. =   
# fxdescribe(c(NA,NA,2,5,6))

# Installer un paquet

# install.packages("psych") # installe le paquet  
library("psych") # charge le paquet et ses fonctions

# Exemples de commandes

# Lancer des commandes------------  
  
5+5

## [1] 10

5\*2

## [1] 10

20/2

## [1] 10

sqrt(40)

## [1] 6.324555

2^4

## [1] 16

# Exercice 1

# Exercice 1----------  
  
seq(from = 0, to = 100, by = 20)

## [1] 0 20 40 60 80 100

c(1:10)\*2

## [1] 2 4 6 8 10 12 14 16 18 20

rep(x = c(1,2,3), times = 3)

## [1] 1 2 3 1 2 3 1 2 3

rep(x = c(1,2,3), times = 3, each = 3)

## [1] 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3

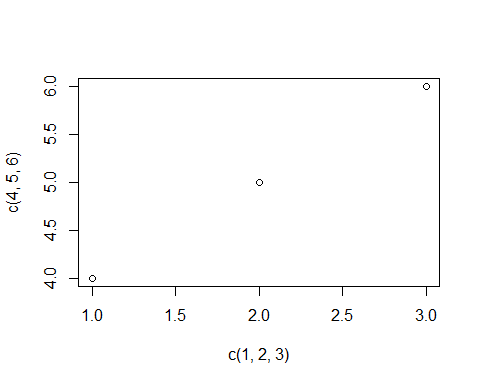
sum(1:3)

## [1] 6

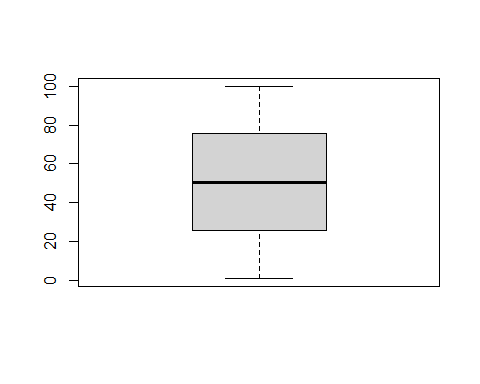
sum(is.na(c(NA,NA,1,NA)))

## [1] 3

plot(x = c(1,2,3), y = c(4,5,6))



boxplot(1:100)



summary(c(1,8,9,7))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 5.50 7.50 6.25 8.25 9.00

# Exercice 2

# Exercice 2----------  
  
nom\_vache1 <- "marguerite"  
nom\_vache1

## [1] "marguerite"

(nom\_vache1 <- "marguerite") # quelle différence observez-vous ?

## [1] "marguerite"

nombre\_vache\_par\_champ <- c(1,5,4,2) # c signifie "combine"  
nombre\_vache\_par\_champ

## [1] 1 5 4 2

print(nombre\_vache\_par\_champ) # print est généralement superflu

## [1] 1 5 4 2

names(nombre\_vache\_par\_champ) <- c("champ1", "champ2", "champ3", "champ4") # donne des noms pour chaque champ  
nombre\_vache\_par\_champ

## champ1 champ2 champ3 champ4   
## 1 5 4 2

mean(nombre\_vache\_par\_champ)

## [1] 3

median(nombre\_vache\_par\_champ)

## [1] 3

sd(nombre\_vache\_par\_champ)

## [1] 1.825742

# Objet simple

objet1 <- 10  
objet2 <- "Michel est dans le garage"  
  
objet1

## [1] 10

objet2

## [1] "Michel est dans le garage"

# Vecteur

# Vecteur--------------  
  
nom\_vaches <- c("marguerite","zoé", "léa")  
str(nom\_vaches)

## chr [1:3] "marguerite" "zoé" "léa"

# characters  
  
nombres\_vaches <- c(1:3)   
str(nombres\_vaches)

## int [1:3] 1 2 3

# integers (chiffres ronds)  
  
nombres\_vaches <- c(1:3/pi)   
str(nombres\_vaches)

## num [1:3] 0.318 0.637 0.955

# numeric (chiffres à virgules)  
  
# les données doivent être du même type   
noms\_et\_nombres\_vaches = c(nom\_vaches, nombres\_vaches)  
noms\_et\_nombres\_vaches

## [1] "marguerite" "zoé" "léa"   
## [4] "0.318309886183791" "0.636619772367581" "0.954929658551372"

# changer noms d'un vecteur  
  
nombre\_vache\_par\_champ <- c(1,5,4,2)   
names(nombre\_vache\_par\_champ) <- c("champ1", "champ2", "champ3", "champ4") # donne des noms pour chaque champ  
nombre\_vache\_par\_champ

## champ1 champ2 champ3 champ4   
## 1 5 4 2

# Facteur

nom\_vaches <- c("marguerite","zoé", "léa")  
nom\_vaches <- as.factor(nom\_vaches)  
nom\_vaches

## [1] marguerite zoé léa   
## Levels: léa marguerite zoé

# Matrice

# Matrice--------------  
# exemple de matrice avec deux vecteurs  
  
v1 <- c(1,5,4,2,5,6)  
v2 <- c(0,5,7,2,0,5)  
# on combine ces deux vecteurs avec la fonction cbind()   
# qui veut dire : column bind (combiner colonnes)  
MAT <- cbind(v1,v2)  
MAT

## v1 v2  
## [1,] 1 0  
## [2,] 5 5  
## [3,] 4 7  
## [4,] 2 2  
## [5,] 5 0  
## [6,] 6 5

# DataFrame

# DataFrame--------------  
v3 <- c("a","a","b","b","c","c")  
DF <- data.frame(v1,v2,v3)  
DF

## v1 v2 v3  
## 1 1 0 a  
## 2 5 5 a  
## 3 4 7 b  
## 4 2 2 b  
## 5 5 0 c  
## 6 6 5 c

# Liste

# Liste--------------  
LS <- list(nom\_vaches, MAT, DF)  
LS

## [[1]]  
## [1] marguerite zoé léa   
## Levels: léa marguerite zoé  
##   
## [[2]]  
## v1 v2  
## [1,] 1 0  
## [2,] 5 5  
## [3,] 4 7  
## [4,] 2 2  
## [5,] 5 0  
## [6,] 6 5  
##   
## [[3]]  
## v1 v2 v3  
## 1 1 0 a  
## 2 5 5 a  
## 3 4 7 b  
## 4 2 2 b  
## 5 5 0 c  
## 6 6 5 c

# Exercice 3

animaux <- c("koala", "chat", "poule", "serpent")  
pays <- c("chine", "japon", "irlande", "canada")  
poids <- c(10,4,3,1)  
(MAT <- matrix(cbind(animaux,pays)))

## [,1]   
## [1,] "koala"   
## [2,] "chat"   
## [3,] "poule"   
## [4,] "serpent"  
## [5,] "chine"   
## [6,] "japon"   
## [7,] "irlande"  
## [8,] "canada"

(DF <- data.frame(animaux, pays, poids))

## animaux pays poids  
## 1 koala chine 10  
## 2 chat japon 4  
## 3 poule irlande 3  
## 4 serpent canada 1

(LS <- list(animaux, DF))

## [[1]]  
## [1] "koala" "chat" "poule" "serpent"  
##   
## [[2]]  
## animaux pays poids  
## 1 koala chine 10  
## 2 chat japon 4  
## 3 poule irlande 3  
## 4 serpent canada 1

# Importation des données

Connaître son répertoire de travail :

getwd()

## [1] "C:/Users/remi.DESKTOP-UI81QOM/Desktop/VERSIONS PROJETS TT/COURS v10tt/6. Stat R analyse linguistique 23-24/COURS"

Lire ReadingSkills à partir d’un sous-fichier

ReadingSkills = readxl::read\_xlsx("ReadingSkills/ReadingSkills.xlsx")

# with csv file  
ReadingSkills <- read.csv2("C:/Users/rlafitte/Desktop/COURS v7chu/6. Stat R analyse linguistique 23-24/COURS/ReadingSkills.csv")  
# View(ReadingSkills)  
  
# read.csv2("ReadingSkills.csv")  
# read.csv("ReadingSkills.csv", sep = ";")  
  
# with excel file  
# install.packages("readxl") # procède à l'installation ; à ne faire qu'une fois  
library(readxl) # procède au chargement de la librairie ; à faire tout le temps  
ReadingSkills <- readxl::read\_xlsx("ReadingSkills.xlsx")

# Visualiser les données importées

head(ReadingSkills, n = 6) # affiche les 6 premières lignes

## # A tibble: 6 x 3  
## accuracy dyslexia iq  
## <dbl> <chr> <dbl>  
## 1 0.884 no 0.827  
## 2 0.765 no 0.59   
## 3 0.915 no 0.471  
## 4 0.984 no 1.14   
## 5 0.884 no -0.676  
## 6 0.709 no -0.795

str(ReadingSkills) # affiche le format de chaque colonne (variable)

## tibble [44 x 3] (S3: tbl\_df/tbl/data.frame)  
## $ accuracy: num [1:44] 0.884 0.765 0.915 0.984 0.884 ...  
## $ dyslexia: chr [1:44] "no" "no" "no" "no" ...  
## $ iq : num [1:44] 0.827 0.59 0.471 1.144 -0.676 ...

summary(ReadingSkills) # stats descriptives pour chaque colonne

## accuracy dyslexia iq   
## Min. :0.4593 Length:44 Min. :-1.7450000   
## 1st Qu.:0.6185 Class :character 1st Qu.:-0.7950000   
## Median :0.7059 Mode :character Median :-0.1225000   
## Mean :0.7728 Mean :-0.0000227   
## 3rd Qu.:0.9900 3rd Qu.: 0.6197500   
## Max. :0.9900 Max. : 1.8560000

# Exercice 4

Importez le fichier “lexdec.csv” disponible sur <https://github.com/lafitter/M2-Science-du-Langage>

Examinez le type de variables composant ce DF ; voyez vous quelle erreur s’est glissée ?!

DF=read.csv2("lexdec.csv")  
# dataframe lexdec : Lexical decision latencies elicited from 21 subjects for 79 English concrete nouns, with variables linked to subject or word  
str(DF)

## 'data.frame': 1659 obs. of 10 variables:  
## $ Subject : chr "A1" "A1" "A1" "A1" ...  
## $ RT : num 6.34 6.31 6.35 6.19 6.03 ...  
## $ Trial : int 23 27 29 30 32 33 34 38 41 42 ...  
## $ Sex : chr "F" "F" "F" "F" ...  
## $ NativeLanguage: chr "English" "English" "English" "English" ...  
## $ Correct : chr "correct" "correct" "correct" "correct" ...  
## $ Word : chr "owl" "mole" "cherry" "pear" ...  
## $ Frequency : chr "4,859812" "4,60517" "4,997212" NA ...  
## $ Length : int 3 4 6 4 3 10 10 8 6 6 ...  
## $ Class : chr "animal" "animal" "plant" "plant" ...

La catégorie Frequency est codée en *caractère* et non pas en *numérique* ! :-C

**Voici l’astuce pour y remédier** :

DF=read.csv2("lexdec.csv", na.strings = c("NA","pas fini"))  
str(DF)

## 'data.frame': 1659 obs. of 10 variables:  
## $ Subject : chr "A1" "A1" "A1" "A1" ...  
## $ RT : num 6.34 6.31 6.35 6.19 6.03 ...  
## $ Trial : int 23 27 29 30 32 33 34 38 41 42 ...  
## $ Sex : chr "F" "F" "F" "F" ...  
## $ NativeLanguage: chr "English" "English" "English" "English" ...  
## $ Correct : chr "correct" "correct" "correct" "correct" ...  
## $ Word : chr "owl" "mole" "cherry" "pear" ...  
## $ Frequency : num 4.86 4.61 5 NA NA ...  
## $ Length : int 3 4 6 4 3 10 10 8 6 6 ...  
## $ Class : chr "animal" "animal" "plant" "plant" ...

# Objet et environnement

# rm(list = ls()) # vide TOUT l'environnement (rm = remove)  
rm(objet1) # supprime un objet  
ls() # liste des objets

## [1] "animaux" "DF" "fxdescribe"   
## [4] "LS" "MAT" "nom\_vache1"   
## [7] "nom\_vaches" "nombre\_vache\_par\_champ" "nombres\_vaches"   
## [10] "noms\_et\_nombres\_vaches" "objet2" "pays"   
## [13] "poids" "ReadingSkills" "v1"   
## [16] "v2" "v3"

ls(".GlobalEnv") # liste des objets (idem)

## [1] "animaux" "DF" "fxdescribe"   
## [4] "LS" "MAT" "nom\_vache1"   
## [7] "nom\_vaches" "nombre\_vache\_par\_champ" "nombres\_vaches"   
## [10] "noms\_et\_nombres\_vaches" "objet2" "pays"   
## [13] "poids" "ReadingSkills" "v1"   
## [16] "v2" "v3"

search() # montre environnement et paquets

## [1] ".GlobalEnv" "package:stats" "package:graphics"   
## [4] "package:grDevices" "package:utils" "package:datasets"   
## [7] "package:methods" "Autoloads" "package:base"

Extraire une variable d’un DF (ReadingSkill)

head(ReadingSkills)

## # A tibble: 6 x 3  
## accuracy dyslexia iq  
## <dbl> <chr> <dbl>  
## 1 0.884 no 0.827  
## 2 0.765 no 0.59   
## 3 0.915 no 0.471  
## 4 0.984 no 1.14   
## 5 0.884 no -0.676  
## 6 0.709 no -0.795

accuracy # ne fonctionne pas...

## Error in eval(expr, envir, enclos): objet 'accuracy' introuvable

ReadingSkills$accuracy # fonctionne !

## [1] 0.88386 0.76524 0.91508 0.98376 0.88386 0.70905 0.77148 0.99000 0.99000  
## [10] 0.99000 0.99000 0.99000 0.99000 0.99000 0.99000 0.99000 0.70281 0.99000  
## [19] 0.66535 0.99000 0.95878 0.99000 0.73402 0.64662 0.99000 0.57794 0.64038  
## [28] 0.45932 0.65286 0.60916 0.60916 0.54048 0.57170 0.70281 0.56546 0.53424  
## [37] 0.57794 0.69032 0.54673 0.68408 0.59043 0.62165 0.67159 0.66535

ReadingSkills["accuracy"] # fonctionne !

## # A tibble: 44 x 1  
## accuracy  
## <dbl>  
## 1 0.884  
## 2 0.765  
## 3 0.915  
## 4 0.984  
## 5 0.884  
## 6 0.709  
## 7 0.771  
## 8 0.99   
## 9 0.99   
## 10 0.99   
## # ... with 34 more rows

ReadingSkills[,"accuracy"] # fonctionne !

## # A tibble: 44 x 1  
## accuracy  
## <dbl>  
## 1 0.884  
## 2 0.765  
## 3 0.915  
## 4 0.984  
## 5 0.884  
## 6 0.709  
## 7 0.771  
## 8 0.99   
## 9 0.99   
## 10 0.99   
## # ... with 34 more rows

Transformer ReadingSkills en environnement

attach(ReadingSkills) # transforme le DF en environnement  
search()

## [1] ".GlobalEnv" "ReadingSkills" "package:stats"   
## [4] "package:graphics" "package:grDevices" "package:utils"   
## [7] "package:datasets" "package:methods" "Autoloads"   
## [10] "package:base"

accuracy

## [1] 0.88386 0.76524 0.91508 0.98376 0.88386 0.70905 0.77148 0.99000 0.99000  
## [10] 0.99000 0.99000 0.99000 0.99000 0.99000 0.99000 0.99000 0.70281 0.99000  
## [19] 0.66535 0.99000 0.95878 0.99000 0.73402 0.64662 0.99000 0.57794 0.64038  
## [28] 0.45932 0.65286 0.60916 0.60916 0.54048 0.57170 0.70281 0.56546 0.53424  
## [37] 0.57794 0.69032 0.54673 0.68408 0.59043 0.62165 0.67159 0.66535

detach(ReadingSkills) # dé-transforme le DF en simple DF

# Exercice 5

accuracy = "papi"  
attach(ReadingSkills)

## The following object is masked \_by\_ .GlobalEnv:  
##   
## accuracy

accuracy

## [1] "papi"

print(accuracy)

## [1] "papi"

attach(ReadingSkills)

## The following object is masked \_by\_ .GlobalEnv:  
##   
## accuracy

## The following objects are masked from ReadingSkills (pos = 3):  
##   
## accuracy, dyslexia, iq

attach(ReadingSkills)

## The following object is masked \_by\_ .GlobalEnv:  
##   
## accuracy  
##   
## The following objects are masked from ReadingSkills (pos = 3):  
##   
## accuracy, dyslexia, iq

## The following objects are masked from ReadingSkills (pos = 4):  
##   
## accuracy, dyslexia, iq

search()

## [1] ".GlobalEnv" "ReadingSkills" "ReadingSkills"   
## [4] "ReadingSkills" "package:stats" "package:graphics"   
## [7] "package:grDevices" "package:utils" "package:datasets"   
## [10] "package:methods" "Autoloads" "package:base"

detach(ReadingSkills)   
detach(ReadingSkills)   
search()

## [1] ".GlobalEnv" "ReadingSkills" "package:stats"   
## [4] "package:graphics" "package:grDevices" "package:utils"   
## [7] "package:datasets" "package:methods" "Autoloads"   
## [10] "package:base"

# Coordonnées

Vecteur

nom\_vaches <- c("marc","jacques", "timéo")  
nom\_vaches[1]

## [1] "marc"

Dataframe

Comment extraire la 2ème ligne du DF ? Comment extraire la 2ème colonne du DF ?

DF <- as.data.frame(ReadingSkills)  
DF[2,]

## accuracy dyslexia iq  
## 2 0.76524 no 0.59

DF[,2] # renvoie un vecteur

## [1] "no" "no" "no" "no" "no" "no" "no" "no" "no" "no" "no" "no"   
## [13] "no" "no" "no" "no" "no" "no" "no" "no" "no" "no" "no" "no"   
## [25] "no" "yes" "yes" "yes" "yes" "yes" "yes" "yes" "yes" "yes" "yes" "yes"  
## [37] "yes" "yes" "yes" "yes" "yes" "yes" "yes" "yes"

DF[2] # renvoie un DF

## dyslexia  
## 1 no  
## 2 no  
## 3 no  
## 4 no  
## 5 no  
## 6 no  
## 7 no  
## 8 no  
## 9 no  
## 10 no  
## 11 no  
## 12 no  
## 13 no  
## 14 no  
## 15 no  
## 16 no  
## 17 no  
## 18 no  
## 19 no  
## 20 no  
## 21 no  
## 22 no  
## 23 no  
## 24 no  
## 25 no  
## 26 yes  
## 27 yes  
## 28 yes  
## 29 yes  
## 30 yes  
## 31 yes  
## 32 yes  
## 33 yes  
## 34 yes  
## 35 yes  
## 36 yes  
## 37 yes  
## 38 yes  
## 39 yes  
## 40 yes  
## 41 yes  
## 42 yes  
## 43 yes  
## 44 yes

Comment extraire la 2ème valeur de la variable iq ?

DF[2,"iq"]

## [1] 0.59

DF[2,3]

## [1] 0.59

DF$iq[2]

## [1] 0.59

attach(DF)

## The following object is masked \_by\_ .GlobalEnv:  
##   
## accuracy

## The following objects are masked from ReadingSkills:  
##   
## accuracy, dyslexia, iq

iq[2]

## [1] 0.59

detach(DF)

Filtrer des données

# différentes stratégies pour créer un DF avec seulement dyslexia et iq :  
(DFiq <- DF[c("dyslexia","iq")] )  
(DFiq <- DF[c(2,3)])  
(DFiq <- DF[2:3])  
(DFiq <- DF[,-c(1)])  
  
(DFsujet\_1\_et\_3 <- DF[c(1,3),])  
(DFsans\_sujet\_4 <- DF[-4,])