Statistique : devoir, octobre 2022

Le sujet est composé de **5 exercices indépendants**. Vous devrez répondre aux questions sur un document quarto (ou éventuellement Markdown) avec une sortie au format **html**. Ce document devra afficher les codes R ainsi que les sorties qui permettent de répondre aux questions. A la fin de l'épreuve vous enverrez

- le **fichier de sortie compilé correctement au format html** ainsi que le répertoire créé par quarto (vous pouvez le zipper)
- le **fichier source au format qmd**

par email à laurent.rouviere@univ-rennes2.fr. La qualité du document quarto sera prise en compte dans le barème tout comme la structure et l'élégance des codes **R**.

Exercice 1 (Intervalles de confiance)**.** On considère les données sur les iris de Fisher. Construire un intervalle de confiance de niveau 95% pour les paramètres suivants :

1. La longueur de Pétales moyenne

```
data(iris)
t.test(iris$Petal.Length,conf.level=0.95)$conf.int
## [1] 3.473185 4.042815
## attr(,"conf.level")
## [1] 0.95
```

2. La largeur de Sépales moyenne de l'espèce Setosa

```
sep_set <- iris |> filter(Species=="setosa") |> select(Sepal.Width)
t.test(sep_set,conf.level=0.95)$conf.int
## [1] 3.320271 3.535729
## attr(,"conf.level")
## [1] 0.95
#ou
iris |> filter(Species=="setosa") |>
  summarize(bi=t.test(Sepal.Width,conf.level=0.95)$conf.int[1],
            bs=t.test(Sepal.Width,conf.level=0.95)$conf.int[2])
```

```
##         bi       bs
## 1 3.320271 3.535729
```

3. La longueur de Sépales moyenne des espèces Versicolor-Virginica, c'est-à-dire de tous les iris excepté l'espèce Setosa.

```
sep_vervin <- iris |>
  filter(Species=="versicolor" | Species =="virginica") |>
  select(Sepal.Length)
t.test(sep_vervin,conf.level=0.95)$conf.int
## [1] 6.130479 6.393521
## attr(,"conf.level")
## [1] 0.95
```

Exercice 2 (Calcul de probabilités avec R)**.** Cet exercice est consacré à des calculs de probabilités et des représentations de lois classiques.

1. On considère X une variable qui suit une loi Binomiale $\mathcal{B}(30, 0.6)$.

   a) Calculer les probabilités suivantes (on donnera les résultats sans utiliser de fonctions **R** mais en justifiant brièvement).

$$\mathbf{P}(X = -2) \quad \text{et} \quad \mathbf{P}(X \geq 0).$$

X prend toutes ses valeurs entre 0 et 30. Par conséquent :

$$\mathbf{P}(X = -2) = 0 \quad \textit{et} \quad \mathbf{P}(X \geq 0) = 1.$$

   b) Calculer les probabilités (avec **R**)

$$\mathbf{P}(X \leq 15) \quad \text{et} \quad \mathbf{P}(X > 15).$$

```
pbinom(15,30,0.6)
## [1] 0.1753691
1-pbinom(15,30,0.6)
## [1] 0.8246309
```

2. On considère ici Y une variable de loi normale d'espérance -2 et de variance 1 (notée $\mathcal{N}(-2, 1)$).

a) Calculer les probabilités (sans le logiciel **R** et en justifiant brièvement)

$$\mathbf{P}(Y = 0) \quad \text{et} \quad \mathbf{P}(Y \leq -2).$$

La première est nulle puisque Y est une variable continue. La seconde vaut 0.5 puisque Y est centrée en -2.

b) Calculer (avec **R**) les probabilités

$$\mathbf{P}(Y \leq 0),\ \mathbf{P}(Y < 0) \quad \text{et} \quad \mathbf{P}(Y > 0).$$

```
pnorm(0,-2,1)
## [1] 0.9772499
pnorm(0,-2,1)
## [1] 0.9772499
1-pnorm(0,-2,1)
## [1] 0.02275013
```

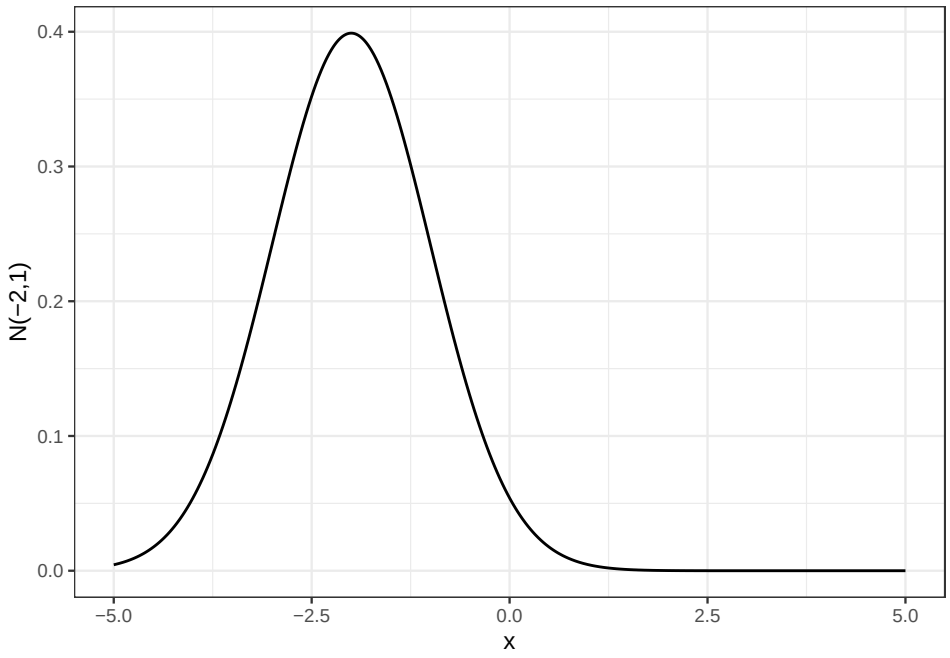
c) Calculer les probabilités

$$\mathbf{P}(-1 \leq Y \leq 1) \quad \text{et} \quad \mathbf{P}(Y \leq -3 \text{ ou } Y \geq 3).$$

```
pnorm(1,-2,1)-pnorm(-1,-2,1)
## [1] 0.1573054
pnorm(-3,-2,1)+(1-pnorm(3,-2,1))
## [1] 0.1586555
```

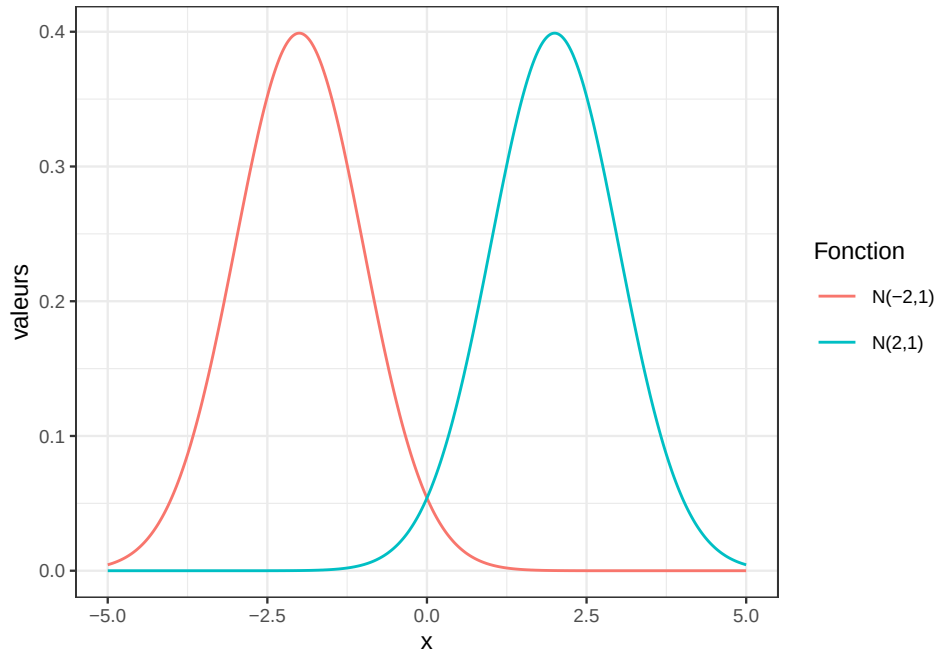
3. A l'aide de `ggplot`, représenter la densité de la loi de Y (la densité de $\mathcal{N}(-2, 1)$). On prendra des valeurs entre -5 et 5 sur l'axe des abscisses.

```
tbl <- tibble(x=seq(-5,5,by=0.01)) |> mutate(`N(-2,1)`=dnorm(x,-2,1))
ggplot(tbl)+aes(x=x,y=`N(-2,1)`)+geom_line()
```



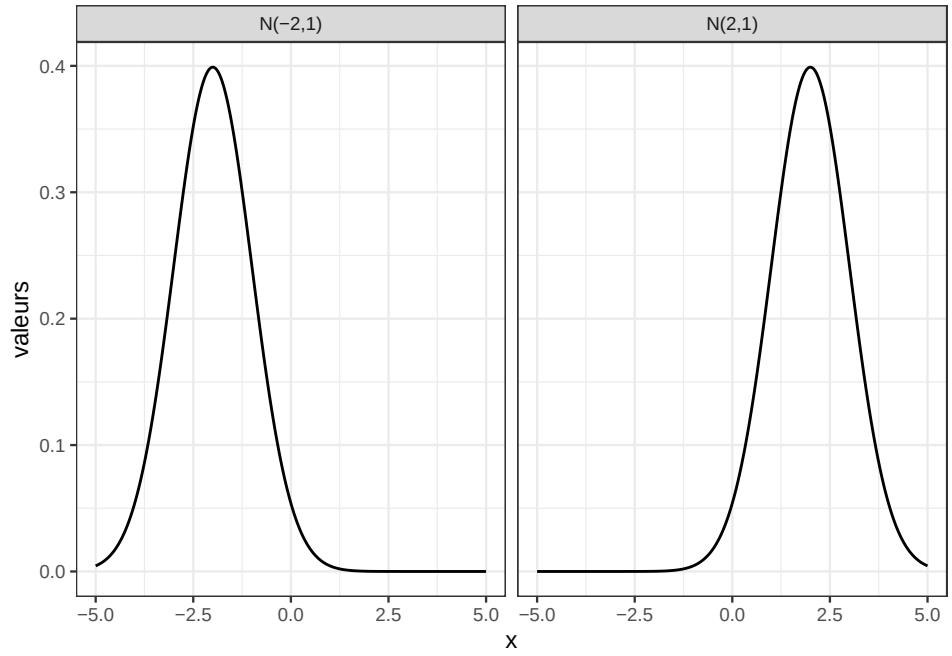
4. Refaire le graphe précédent an ajoutant la densité de la loi normale d'espérance 2 et de variance 1 ($\mathcal{N}(2,1)$). On utilisera deux couleurs différentes et on affichera une légende qui permet de distinguer les deux lois.

```
tbl1 <- tbl |> mutate(`N(2,1)`=dnorm(x,2,1)) |>
  pivot_longer(-x,names_to = "Fonction",values_to = "valeurs")
ggplot(tbl1)+aes(x=x,y=valeurs,color=Fonction)+geom_line()
```



5. Refaire la question précédente en représentant les deux densités sur deux graphes séparés. On utilisera `facet_wrap`.

```
ggplot(tbl1)+aes(x=x,y=valeurs)+geom_line()+facet_wrap(~Fonction)
```



Exercice 3 (Les tirs au but en Allemagne)**.** Le package `footballpenaltiesBL`

```
library(footballpenaltiesBL)
data(penalties)
```

propose une base de données `penalties` qui contient des informations sur 4599 penaltys tirés dans le championnat d'Allemagne entre 1963 et 2017. La base de données contient 4599 lignes (individus) et 15 colonnes (variables). On pourra trouver un descriptif des variables avec

```
help(penalties)
```

Les principales variables sont :

- `result` : le résultat du penalty (marqué, arrêté par le goel, à coté…)
- `goalkeeper` : le nom du gardien de but
- `penaltytaker` : le nom du tireur du penalty
- `season` : la saison où le penalty a été tiré
- `ptclub` : le club d'appartenance du tireur du penalty
- …

On répondra aux questions suivantes en se basant sur les grammaires `dplyr` (pour les calculs d'indicateurs) et `ggplot` (pour les représentations graphiques).

1. Sur les 4599 penaltys de la base, combien ont été marqués ?

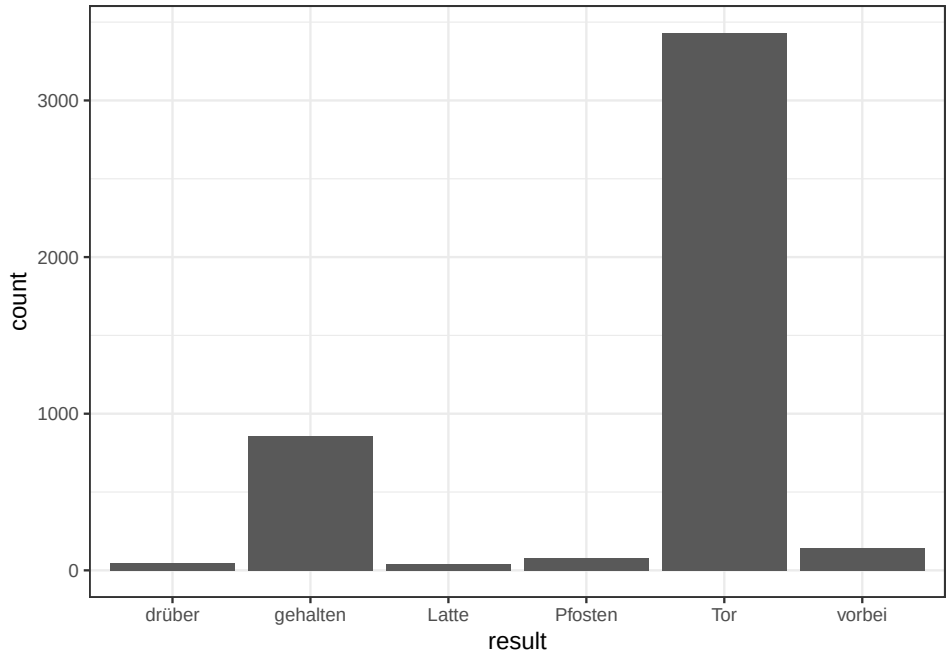
```
penalties |> filter(result=="Tor") |> summarize("marque"=n())
##   marque
## 1   3432
```

2. Calculer le nombre de penaltys marqués, arrêtés, tirés sur le poteau, la barre… Il s'agit de compter les effectifs de chaque modalités de la variable `result`. On pourra utiliser la fonction `count`. On pourra afficher les résultats en effectifs et en pourcentage.

```
penalties |> count(result) |> mutate(prop=round(n/4599*100,2))
##     result    n  prop
## 1   drüber   49  1.07
## 2 gehalten  856 18.61
## 3    Latte   39  0.85
## 4  Pfosten   79  1.72
## 5      Tor 3432 74.62
## 6   vorbei  144  3.13
```

3. Visualiser les effectifs calculés à la question précédente à l'aide d'un diagramme en barres.

```
ggplot(penalties)+aes(x=result)+geom_bar()
```



4. Quel gardien a les meilleurs résultats en terme de penaltys manqués (un penalty est manqué à partir du moment où il n'a pas été marqué).

```
penalties |> group_by(goalkeeper) |>
  summarize(manques=sum(result!="Tor")) |>
  arrange(desc(manques)) |> slice(1)
## # A tibble: 1 x 2
##   goalkeeper     manques
##   <fct>            <int>
## 1 Kargus, Rudolf      29
```

5. Quels sont les 3 joueurs qui ont marqué le plus de penaltys. On affichera le nom des joueurs et le nombre de penaltys marqués.

```
penalties |> group_by(penaltytaker) |>
  summarize(marque=sum(result=="Tor")) |>
  arrange(desc(marque)) |> slice(1:3)
## # A tibble: 3 x 2
##   penaltytaker   marque
##   <fct>           <int>
## 1 Kaltz, Manfred     53
```

```
## 2 Müller4, Gerd      51
## 3 Zorc, Michael      49
```

6. Construire un tibble a 3 colonnes qui contient, pour chaque gardien de but :
   - le nombre de penaltys tirés contre lui
   - le nombre de penaltys qui n'ont pas été marqués contre lui

```
df <- penalties |> group_by(goalkeeper) |>
  summarize(total=n(),manques=sum(result!="Tor"))
df
## # A tibble: 353 x 3
##    goalkeeper         total manques
##    <fct>              <int>   <int>
##  1 Adler, Rene           30      10
##  2 Albustin, Thorsten     1       1
##  3 Alter, André           2       0
##  4 Amsif, Mohamed         3       0
##  5 Aumann, Raimond       14       4
##  6 Bade, Alexander       10       3
##  7 Bahr, Nils             2       0
##  8 Bailly, Logan          9       1
##  9 Basikow, Klaus         1       0
## 10 Baumann, Oliver       24       5
## # ... with 343 more rows
```

7. Déduire de la question précédente quel gardien, parmi les gardiens ayant été concernés par strictement plus de 30 penaltys, celui qui a la meilleure proportion de penaltys manqués ? Cette proportion est égale au nombre de penaltys manqués divisé par le nombre de penaltys tirés face à ce gardien.

```
df |> filter(total>30) |> mutate(prop=manques/total) |>
  arrange(desc(prop)) |> slice(1)
## # A tibble: 1 x 4
##   goalkeeper        total manques  prop
##   <fct>             <int>   <int> <dbl>
## 1 Radenkovic, Petar    37      15 0.405
```

8. Quel club a obtenu le plus de penaltys au cours de la saison 2015/2016 ?

```
penalties |> filter(season=="15/16") |> group_by(ptclub) |>
  summarize(total=n()) |> arrange(desc(total)) |> slice(1)
## # A tibble: 1 x 2
##   ptclub           total
```

```
##   <fct>            <int>
## 1 FC Ingolstadt 04    10
```

Exercice 4 (Intervalles de confiance - la suite)**.** On considère toujours les ris de Fisher. On note μ_X la (vraie mais inconnue) longueur de sépales moyenne de l'espèce setosa et μ_Y la (vraie mais inconnue) longueur de sépales moyenne de l'espèce versicolor. On répondra aux questions suivantes à partir des mesures présentes dans le jeu de données `iris`.

1. Calculer un intervalle de confiance de niveau 90% pour le paramètre $\mu_X - \mu_Y$

```
sep_set <- iris |> filter(Species=="setosa") |> select(Sepal.Length)
sep_ver <- iris |> filter(Species=="versicolor") |> select(Sepal.Length)
t.test(sep_set$Sepal.Length,sep_ver$Sepal.Length,
       conf.level = 0.9)$conf.int
## [1] -1.0769698 -0.7830302
## attr(,"conf.level")
## [1] 0.9
```

2. A la lecture du résultat de la question précédente, que pouvez-vous intuiter ?

   0 n'étant pas dans l'intervalle de confiance, on peut penser que les valeurs de μ_X et μ_Y sont significativement différentes.

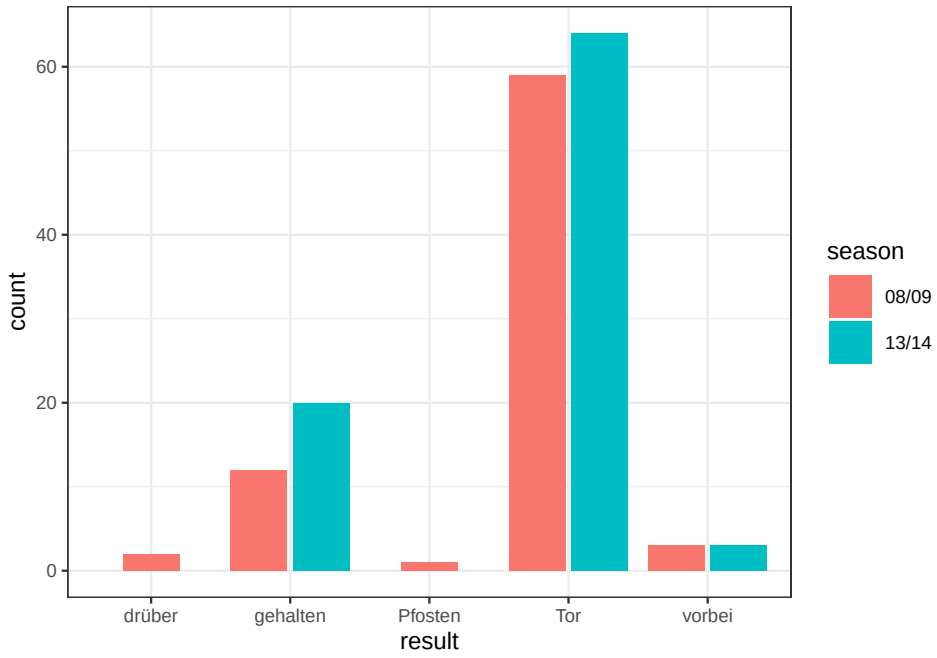
3. Calculer un intervalle de confiance de niveau 90% pour le paramètre $\mu_X + \mu_Y$

```
t.test(sep_set$Sepal.Length,-sep_ver$Sepal.Length,
       conf.level = 0.9)$conf.int
## [1] 10.79503 11.08897
## attr(,"conf.level")
## [1] 0.9
```

Exercice 5 (Les tirs au but en Allemagne - toujours)**.** On souhaite ici comparer différentes variables entre les saisons 2008-2009 et 2013-2014. Proposer différents graphiques permettant d'établir cette comparaison. On pourra par exemple :

- comparer la distribution de la variable `result` ;
- comparer le nombre de penaltys obtenu par chaque club ;
- ...

```
pen2 <- penalties |> filter(season=="08/09" | season=="13/14")
ggplot(pen2)+aes(x=result,fill=season)+
  geom_bar(position = position_dodge2(preserve = "single"))
```




```
pen3 <- pen2 |> group_by(season,ptclub) |>
  summarize(nb_pen=n())
ggplot(pen3)+aes(y=ptclub,x=nb_pen,fill=season)+
  geom_bar(stat="identity")
```