TP statistiques univariées

Eric Marcon

14 février 2024

Statistiques descriptives

Données

Enquête de vie 2003 de l'INSEE

```
library("questionr")
data(hdv2003)
```

☺ Afficher les tableaux avec `View()`

Moyenne, écart-type, médiane

Statistiques sur l'âge des personnes interrogées

```
mean(hdv2003$age)
```

```
## [1] 48.157
```

```
sd(hdv2003$age)
```

```
## [1] 16.94181
```

```
var(hdv2003$age)
```

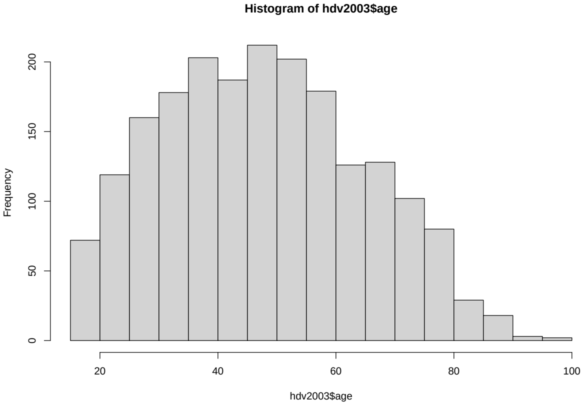
```
## [1] 287.0249
```

```
median(hdv2003$age)
```

```
## [1] 48
```

Histogramme

```
hist(hdv2003$age)
```



AgroParisTech

TP statistiques univariées

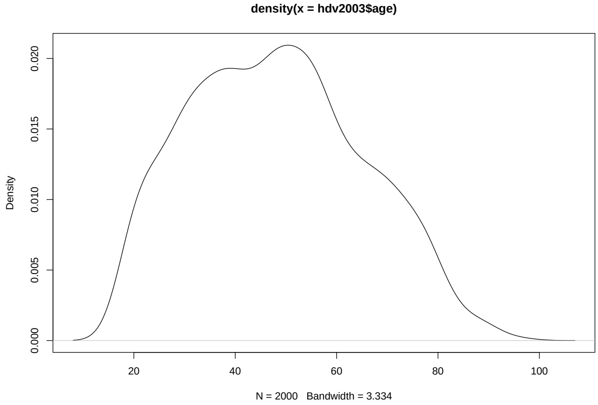
Eric Marcon

Statistiques descriptives

Lois de Probabilités

Densité de probabilité

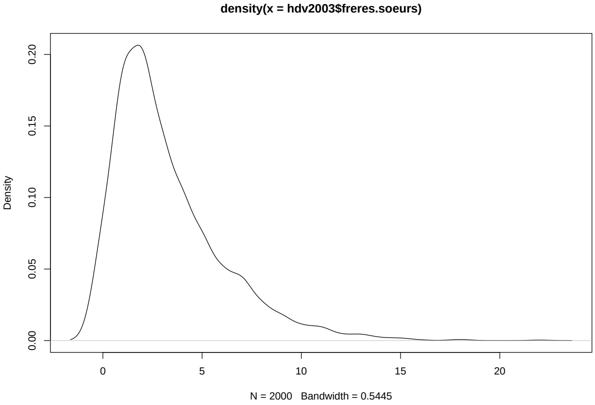
```
plot(density(hdv2003$age))
```



Densité sans réflexion

La densité n'est pas bornée

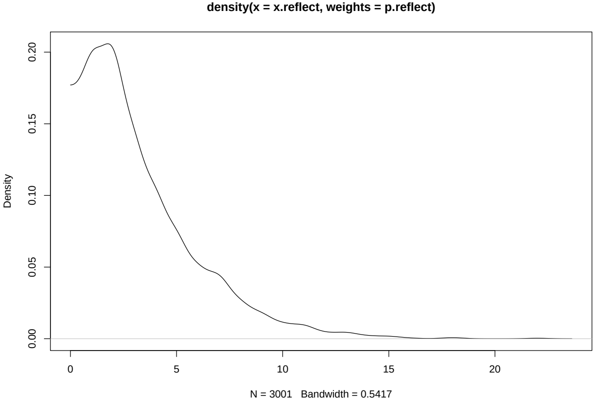
```
plot(density(hdv2003$freres.soeurs))
```



Densité avec réflexion

Utiliser le package *GoFKernel*

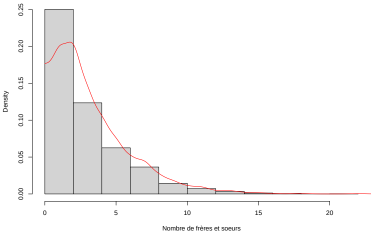
```
library("GoFKernel")
plot(
  density.reflected(hdv2003$freres.soeurs, lower = 0)
)
```



Histogramme lissé

Histogramme des probabilités

```
hist(
  hdv2003$freres.soeurs,
  prob = TRUE,
  main = "",
  xlab = "Nombre de frères et soeurs"
)
lines(
  density.reflected(hdv2003$freres.soeurs, lower = 0),
  col = "red"
)
```



Quantiles

```
summary(hdv2003$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   35.00   48.00   48.16   60.00   97.00
```

```
quantile(hdv2003$age, probs = c(0.025, 0.975))
```

```
##  2.5% 97.5%
##    20    81
```

AgroParisTech

TP statistiques univariées

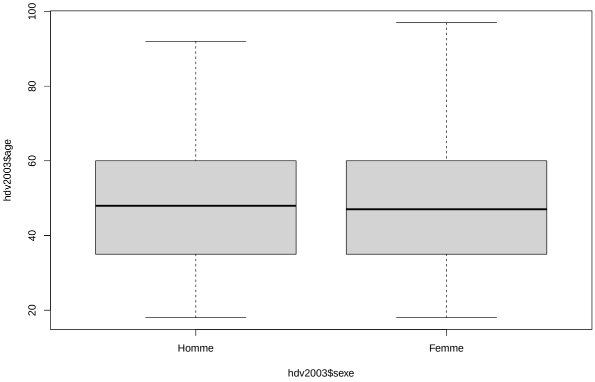
Eric Marcon

Statistiques descriptives

Lois de Probabilités

Boîte à moustaches

```
boxplot(hdv2003$age ~ hdv2003$sexe)
```



Comptages

Pour les variables discrètes.

```
table(hdv2003$sexe)
```

```
##
## Homme Femme
##   899  1101
```

Lois de Probabilités

Distributions classiques

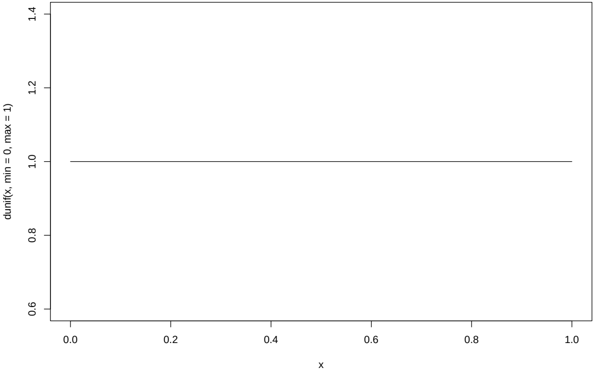
Incontournables:

- loi uniforme
- loi de Bernoulli, loi binomiale
- loi de Poisson
- loi normale (gaussienne)

Loi uniforme

Densité de probabilité:

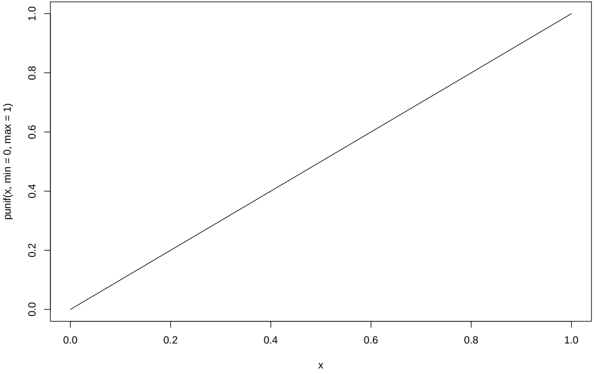
```
curve(dunif(x, min = 0, max = 1), from = 0, to = 1)
```



Loi uniforme

Fonction cumulative :

```
curve(punif(x, min = 0, max = 1), from = 0, to = 1)
```



Loi uniforme

Fonction quantile:

```
qunif(p = 0.95, min = 0, max = 2)
```

```
## [1] 1.9
```

Loi uniforme

Tirage:

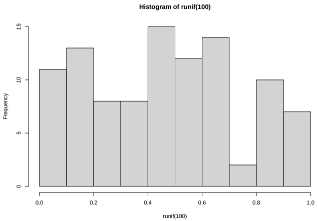
```
runif(n = 5)
```

```
## [1] 0.0482511 0.2186699 0.5021846 0.4053148
## [5] 0.3986500
```

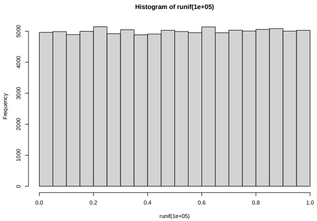
Toutes les distributions de probabilité ont des fonctions `d`, `p`, `q` et `r`.

Loi des grands nombres

```
hist(runif(100))
```




```
hist(runif(100000))
```



La distribution des tirages tend vers la loi quand le nombre de tirages augmente.

AgroParisTech

TP statistiques univariées

Eric Marcon

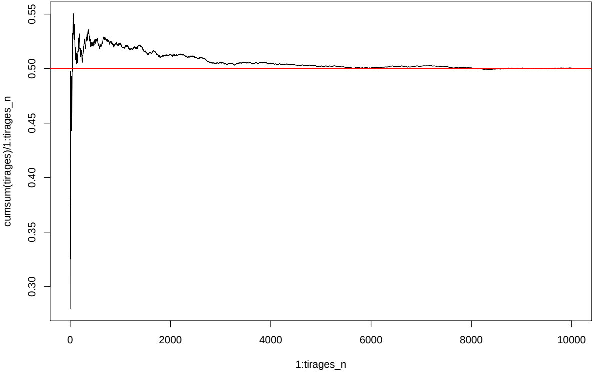
Statistiques descriptives

Lois de Probabilités

Loi des grands nombres

La moyenne tend aussi vers l'espérance:

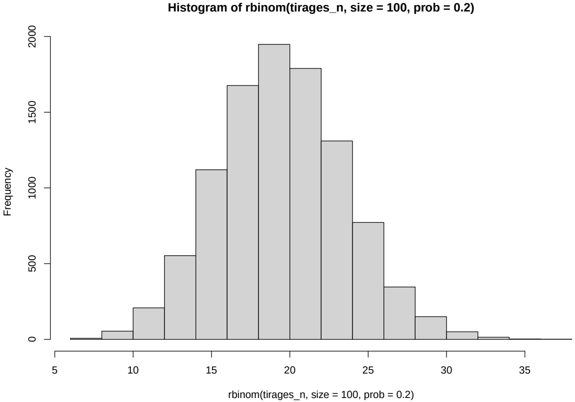
```
tirages_n <- 10000
tirages <- runif(tirages_n)
plot(x = 1:tirages_n, y = cumsum(tirages) / 1:tirages_n, type = "l")
abline(h = 0.5, col = "red")
```



Loi binomiale

Nombre de succès d'une épreuve répétée `size` fois avec la probabilité de succès `prob`.

```
hist(rbinom(tirages_n, size = 100, prob = 0.2))
```

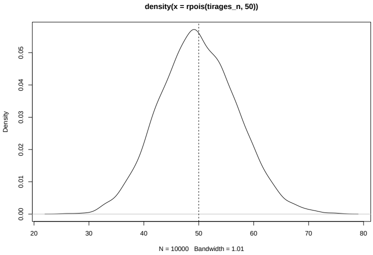


Loi de Poisson

Loi binomiale dont la probabilité de succès tend vers 0 et le nombre d'épreuves vers $+\infty$.

Ex.: combien d'arbres se trouvent dans 1000 m^2 de forêt avec une densité de 500/ha?

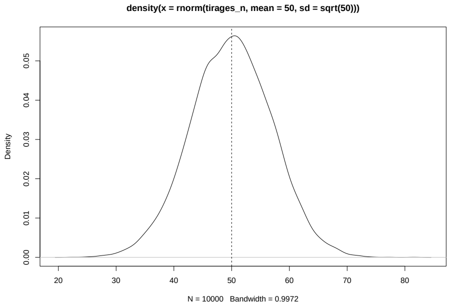
```
# 10000 tirages, espérance = 500 * 0.1
plot(density(rpois(tirages_n, 50)))
abline(v = 50, lty = 2)
```



Loi normale

Distribution de la moyenne de nombreuses variables aléatoires.

```
# 10000 tirages, espérance = 500 * 0.1
plot(density(rnorm(tirages_n, mean = 50, sd = sqrt(50))))
abline(v = 50, lty = 2)
```



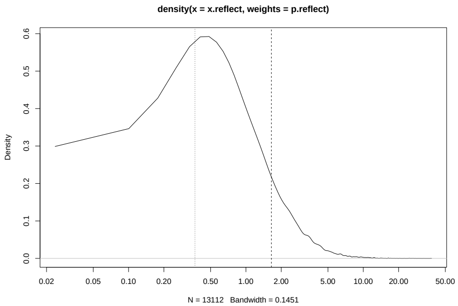
A comparer avec la loi de Poisson

Loi log-normale

Loi dont le logarithme est normal.

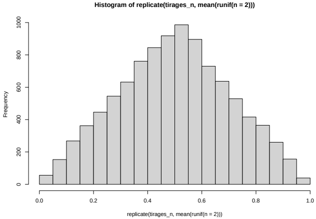
Distribution du produit de nombreuses variables aléatoires.

```
plot(
  density.reflected(rlnorm(tirages_n, meanlog = 0, sdlog = 1), lower = 0
  log ="x"
)
abline(v = exp(1/2), lty = 2)   # Espérance
abline(v = exp(-1), lty = 3)    # Mode
```

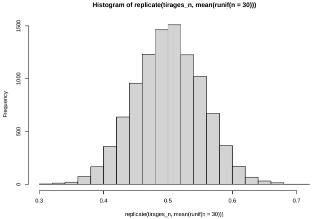


Théorème de la limite centrale

```
hist(
  replicate(
    tirages_n,
    mean(runif(n = 2))
  )
)
```




```
hist(
  replicate(
    tirages_n,
    mean(runif(n = 30))
  )
)
```



La distribution de la moyenne de n variables uniformes tend vers la loi normale. Sa variance est celle de la loi uniforme $(1/12)$ divisée par n.

Intervalle de confiance

α est le seuil de risque, en général 5%.

$1 - \alpha$ est le seuil de confiance, en général 95%.

95% des **tirages** d'une loi normale sont situés à moins de 1.96 écarts-types (σ) de l'espérance.

```
qnorm(0.975)
```

```
## [1] 1.959964
```

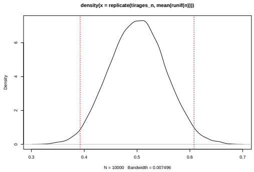
Intervalle de confiance

La **moyenne** de n variables aléatoires tend vers une loi normale.
95% de ses réalisations sont situés à moins de $1,96\sigma/\sqrt{n}$ de l'espérance.
Précisément 1,96 est le 97,5ème centile de la loi de Student avec un très grand nombre de degrés de liberté.

```
alpha <- 0.05
qt(1 - alpha / 2, df = 1E6)
```

```
## [1] 1.959966
```

```
n = 30
plot(density(
  replicate(
    tirages_n,
    mean(runif(n))
  )
))
ci <- qt(1-alpha/2, df = n - 1) /
  sqrt(12) / sqrt(n)
abline(v = 0.5 + c(ci, -ci),
       col = "red", lty = 2)
```



Test contre une valeur

Combien de temps regarde-t-on la TV par jour ?

```
(tv_mean <- mean(hdv2003$heures.tv, na.rm = TRUE))
```

```
## [1] 2.246566
```

n mesures individuelles, loi inconnue. La moyenne tend vers une loi normale.

```
n <- sum(!is.na(hdv2003$heures.tv))
tv_sd <- sd(hdv2003$heures.tv, na.rm = TRUE)
ci <- qt(1 - alpha / 2, df = n - 1) * tv_sd / sqrt(n)
paste("Intervalle de confiance:", tv_mean - ci, "-", tv_mean + ci)
```

```
## [1] "Intervalle de confiance: 2.16859286989944 - 2.32453996218076"
```

On regarde la TV plus de 2 heures par jour (95% de confiance).

Méthode de Monte-Carlo

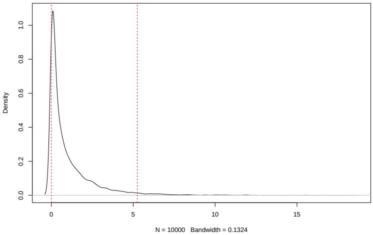
Si la loi est inconnue mais l'algorithme de simulation disponible.

Exemple : carré d'une distribution normale.

```
dist <- rnorm(tirages_n)^2
(dist_q <- quantile(dist, c(0.025, 0.975)))
```

```
##        2.5%       97.5%
## 0.001017103 5.249021141
```

```
plot(density(dist), main = "")
abline(v = dist_q, col = "red", lty = 2)
```



Méthode de Monte-Carlo

... mais on connaît souvent les distributions.

Le carré d'une loi normale est une loi du χ^2 à 1 degré de liberté, identique à une loi Γ de forme $1/2$ et d'échelle 2.

```
qchisq(.075, df = 1)
```

```
## [1] 0.008861853
```

```
qchisq(.975, df = 1)
```

```
## [1] 5.023886
```

```
qgamma(0.975, shape = 1/2, scale = 2)
```

```
## [1] 5.023886
```

$\rightarrow$ lire l'aide `?qchisq`, Wikipedia, Google...