# Les tests d'hypothèses

Vincent Guillemot

Jeudi

# Rappels

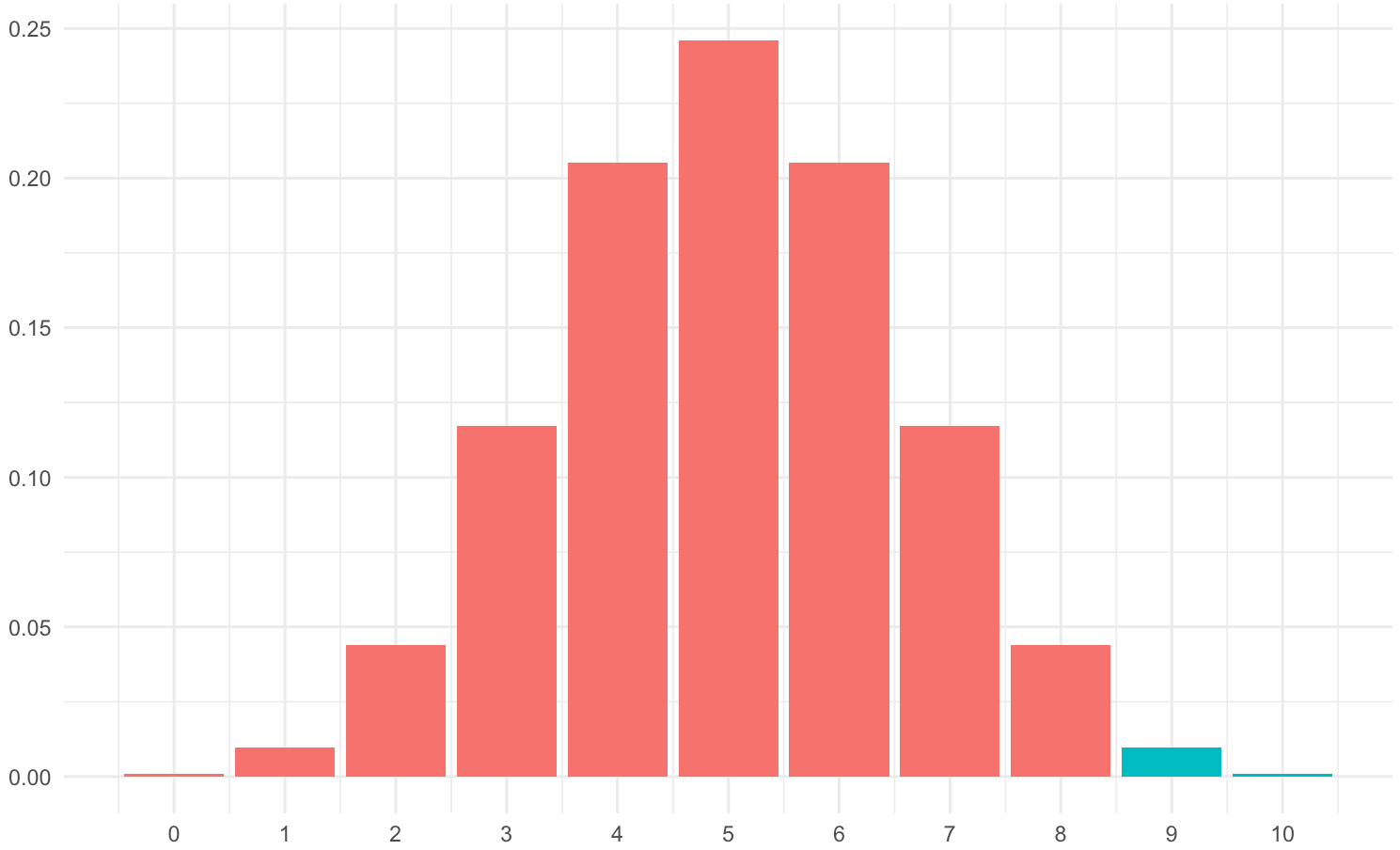
- Hypothèse nulle $H_0$, c'est l'hypothèse du *statu quo*
- Hypothèse alternative $H_1$, c'est la situation intéressante ! (signal)
- $\alpha$ : risque de première espèce, rejeter $H_0$ lorsqu'elle est vraie ("erreur de détection")
- $\beta$ : risque de deuxième espèce, ne pas rejeter $H_0$ alors que $H_1$ est vraie ("rater un signal")
- Puissance : $1 - \beta$

| Décision / Verité | Non rejet de $H_0$ | Rejet $H_0$ |
|---|---|---|
| $H_0$ | Confiance | Erreur de 1ère esp. |
| $H_1$ | Erreur de 2ème esp. | Puissance |

# Expérience de Chastaing (1958)

Keewee ou Koowoo ?

# Les résultats



# Rappel 1 : variable aléatoire du $\chi^2$

Une variable suivant une loi du khi-deux à $k$ degrés de liberté ($\chi^2(k)$) est la somme des carrés de $k$ variable normales indépendantes :

$$\sum_{i=1}^{k} Z_i^2 \sim \chi^2(k)$$

Remarques :

- parfois on note une telle variable $X^2$
- en pratique, ces “carrés” sont souvent des “variances”

# Rappel 2 : variable aléatoire de Student

Une variable obtenue en divisant un variable normale par la racine carrée d'une variable du khi-deux (indépendante de la première) elle-même normalisée par son dégré de liberté $d$ suit une loi de Student à $d$ degrés de liberté :

$$\frac{Z}{\sqrt{\frac{1}{d}X}} \sim T(d)$$

En pratique :

$$\frac{\text{moyenne}}{\frac{1}{\sqrt{\text{taille}}}\,\text{écart-type}} \sim T(\text{taille} - 1)$$

# Rappel 3 : variable aléatoire de Fisher

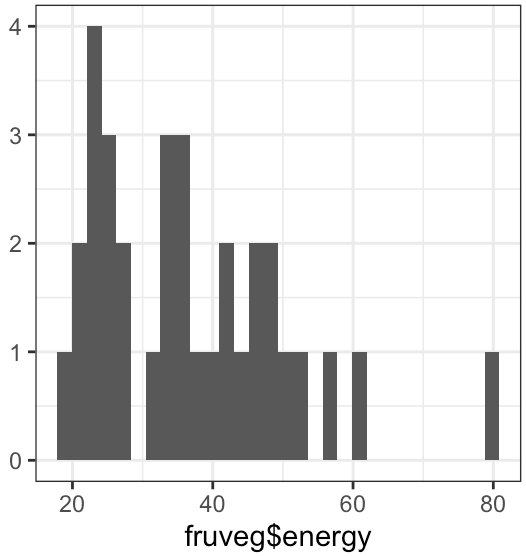
Le ratio de deux variables indépendantes du khi-deux à $d_1$ et $d_2$ degrés de liberté est une variable aléatoire de Fisher à $d_1$ et $d_2$ degrés de liberté :

$$\frac{X_1^2}{X_2^2} \sim F(d_1, d_2)$$

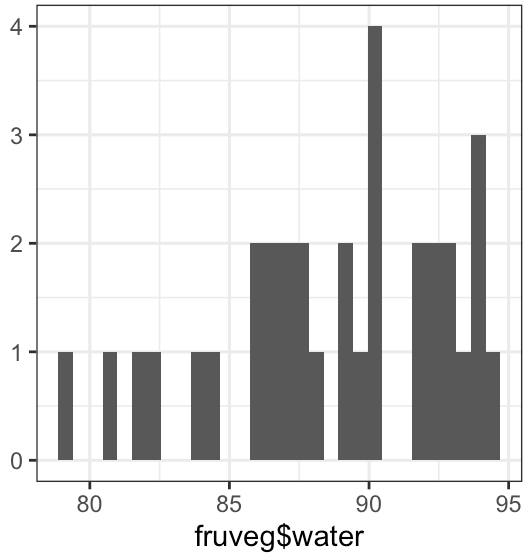
En pratique : des ratios de variance !

# Intermède : Création d'un exemple

```
qplot(fruveg$energy)
```




```
qplot(fruveg$water)
```




```
energiequal <- cut(fruveg$energy,
                   c(0, 35, 80))
```

```
eauqual <- cut(fruveg$water,
                   c(0, 90, 100))
```

# Table de contingence

Une table de contingence, ou table de comptage, est un tableau croisé (de comptage) entre deux variables qualitatives ou plus.

```
(tab <- table(energiequal, eauqual))
#>            eauqual
#> energiequal (0,90] (90,100]
#>     (0,35]       4       12
#>     (35,80]     15        2
```

On peut aussi calculer les proportions

```
prop.table(tab)
#>            eauqual
#> energiequal     (0,90]   (90,100]
#>     (0,35]  0.12121212 0.36363636
#>     (35,80] 0.45454545 0.06060606
```

# Profils lignes et profils colonnes

Proportions conditionnellement aux lignes :

```
prop.table(tab, margin = 1)
#>             eauqual
#> energiequal    (0,90]  (90,100]
#>     (0,35]  0.2500000 0.7500000
#>     (35,80] 0.8823529 0.1176471
```

Proportions conditionnellement aux colonnes :

```
prop.table(tab, margin = 2)
#>             eauqual
#> energiequal    (0,90]  (90,100]
#>     (0,35]  0.2105263 0.8571429
#>     (35,80] 0.7894737 0.1428571
```

# Comparer des proportions

Avec la fonction `prop.test` :

```
prop.test(table(energiequal, eauqual))
#>
#>  2-sample test for equality of proportions with continuity correction
#>
#> data:  table(energiequal, eauqual)
#> X-squared = 11.029, df = 1, p-value = 0.0008971
#> alternative hypothesis: two.sided
#> 95 percent confidence interval:
#>  -0.9546901 -0.3100158
#> sample estimates:
#>    prop 1    prop 2
#> 0.2500000 0.8823529
```

Attention, le test des proportions a besoin de données de comptage, pour lui :

$$\frac{2}{4} \neq \frac{50}{100}$$

# La fonction `prop.test`

- Accepte des tables de contingences,
- Ou bien deux vecteurs : `x` pour les “succès”, `n` pour le nombre total,
- Éventuellement un vecteur de proportions de référence `p`

Un des exemples de la fonction (cf. `?prop.test`) :

```
smokers  <- c( 83, 90, 129, 70 )
patients <- c( 86, 93, 136, 82 )
prop.test(smokers, patients)
#>
#>  4-sample test for equality of proportions without continuity correction
#>
#> data:  smokers out of patients
#> X-squared = 12.6, df = 3, p-value = 0.005585
#> alternative hypothesis: two.sided
#> sample estimates:
#>    prop 1    prop 2    prop 3    prop 4
#> 0.9651163 0.9677419 0.9485294 0.8536585
```

# Test du “khi-deux”

Avec la fonction `chisq.test` :

```
chisq.test(energiequal, eauqual)
#>
#>  Pearson's Chi-squared test with Yates' continuity correction
#>
#> data:  energiequal and eauqual
#> X-squared = 11.029, df = 1, p-value = 0.0008971
```

# La fonction `chisq.test`

- Accepte deux variables qualitatives,
- Ou une table de contingence

Un des exemples de la fonction (cf. `?chisq.test`) :

```
M <- as.table(rbind(c(762, 327, 468), c(484, 239, 477)))
dimnames(M) <- list(gender = c("F", "M"),
                    party = c("Democrat","Independent", "Republican"))
(Xsq <- chisq.test(M))  # Prints test summary
#>
#>  Pearson's Chi-squared test
#>
#> data:  M
#> X-squared = 30.07, df = 2, p-value = 2.954e-07
Xsq$expected   # expected counts under the null
#>       party
#> gender Democrat Independent Republican
#>      F 703.6714    319.6453   533.6834
#>      M 542.3286    246.3547   411.3166
```

# La statistique du test du $\chi^2$

Elle compare les fréquences observées aux fréquences attendues. Les fréquences attendues sont calculées à partir des fréquences marginales sous hypothèse d'indépendance.

$$X^2 = \sum \frac{\left(n_{ij} - \frac{n_{i.} n_{.j}}{n}\right)^2}{\frac{n_{i.} n_{.j}}{n}},$$

avec $n_{ij}$ l'effectif observé, $n_{i.}$ l'effectif marginal ligne, $n_{.j}$ l'effectif marginal colonne et $n$ l'effectif total.

Rappel : quand $A$ et $B$ son indépendants, $P(A \cap B) = P(A)P(B)$.

# Test exact de Fisher

Avec la fonction `fisher.test` :

```
fisher.test(energiequal, eauqual)
#>
#>  Fisher's Exact Test for Count Data
#>
#> data:  energiequal and eauqual
#> p-value = 0.0003614
#> alternative hypothesis: true odds ratio is not equal to 1
#> 95 percent confidence interval:
#>  0.003955024 0.353541685
#> sample estimates:
#> odds ratio
#>  0.0507175
```

# La fonction `fisher.test`

- Accepte deux variables qualitatives,
- Ou une table de contingence.

Un des exemples de la fonction (cf. `?fisher.test`) :

```
Convictions <- matrix(
  c(2, 10, 15, 3),
  nrow = 2,
  dimnames = list(
    c("Dizygotic", "Monozygotic"),
    c("Convicted", "Not convicted")))
fisher.test(Convictions, alternative = "less")
#>
#>  Fisher's Exact Test for Count Data
#>
#> data:  Convictions
#> p-value = 0.0004652
#> alternative hypothesis: true odds ratio is less than 1
#> 95 percent confidence interval:
#>  0.0000000 0.2849601
#> sample estimates:
#> odds ratio
#> 0.04693661
```

# Comparer des moyennes

Avec la fonction `t.test` :

```
t.test(fruveg$vitaminC ~ eauqual)
#>
#>  Welch Two Sample t-test
#>
#> data:  fruveg$vitaminC by eauqual
#> t = 1.3836, df = 25.372, p-value = 0.1785
#> alternative hypothesis: true difference in means between group (0,90] and group (90,100
#> 95 percent confidence interval:
#>  -8.481789 43.287653
#> sample estimates:
#>   mean in group (0,90] mean in group (90,100]
#>               33.59579                16.19286
```

# Les formules

Les formules permettent à l'utilisateur de décrire un modèle :

$$Y = X_1 + X_2 + X_3 + X_2 * X_3 + X_3 * X_4$$

deviendra

```
y ~ x1 + x2 * x3 + x3:x4
```

Repérez le tilde sur votre clavier, il est très important en R !

Exemple :

```
y ~ x + age + sex + SCL:disease
```

# La fonction `t.test`

- Accepte une formule,
- Ou bien deux vecteurs contenant respectivement les deux groupes de valeurs à comparer,
- L'argument `paired = TRUE` pour des données appariées,
- Ou bien un seul vecteur (pour un test sur une moyenne),

Un des exemples de la fonction (cf. `?t.test`) :

```
t.test(extra ~ group, data = sleep)
#>
#>  Welch Two Sample t-test
#>
#> data:  extra by group
#> t = -1.8608, df = 17.776, p-value = 0.07939
#> alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
#> 95 percent confidence interval:
#>  -3.3654832  0.2054832
#> sample estimates:
#> mean in group 1 mean in group 2
#>            0.75            2.33
```

# Equivalent non-paramétrique

L'équivalent non-paramétrique du test de Student est le test de Wilcoxon-Mann-Whitney :

```
wilcox.test(fruveg$vitaminC ~ eauqual)
#> Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot compute exa
#>
#>  Wilcoxon rank sum test with continuity correction
#>
#> data:  fruveg$vitaminC by eauqual
#> W = 163, p-value = 0.2825
#> alternative hypothesis: true location shift is not equal to 0
```

# Remarque sur ces fonctions

L'objet retourné est une liste qui contient les deux éléments les plus intéressants (en général) : `statistic` et `p.value`.

Exemple de récupération de la P-value :

```
res.ttest <- t.test(fruveg$vitaminC ~ eauqual)
pval <- res.ttest$p.value
```

# ANOVA

Faire une ANOVA en R n'est pas une mince affaire !

```
summary(aov(energy ~ group, data = fruveg))
#>             Df Sum Sq Mean Sq F value   Pr(>F)
#> group        2   3160  1580.0   16.82 1.26e-05 ***
#> Residuals   30   2818    93.9
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Et récupérer la P-value est ridiculement difficile :

```
res <- summary(aov(energy ~ group, data = fruveg))
res[[1]]$`Pr(>F)`[1]
#> [1] 1.260605e-05
```

# ANOVA non paramétrique

La syntaxe est un peu différente pour l'équivalent non-paramétrique : le test de Kruskal-Wallis :

```
kruskal.test(energy ~ group, data = fruveg)
#>
#>  Kruskal-Wallis rank sum test
#>
#> data:  energy by group
#> Kruskal-Wallis chi-squared = 16.918, df = 2, p-value = 0.000212
```

Récupérer la p-valeur s'effectue de la même façon que pour un test de Student.

# Corrélation

Avec la fonction `cor.test`. Exemple :

```
cor.test(fruveg$iron, fruveg$calcium)
#>
#>  Pearson's product-moment correlation
#>
#> data:  fruveg$iron and fruveg$calcium
#> t = 4.1218, df = 31, p-value = 0.0002601
#> alternative hypothesis: true correlation is not equal to 0
#> 95 percent confidence interval:
#>  0.3162937 0.7791490
#> sample estimates:
#>       cor
#> 0.5949944
```

# La fonction `cor.test`

- Accepte deux vecteurs `x` et `y` de même longueur,
- Permet de tester les trois types de corrélation (Pearson, Sparman et Kendall)

Un des exemples de la fonction (cf. `?cor.test`) :

```
x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
y <- c( 2.6,  3.1,  2.5,  5.0,  3.6,  4.0,  5.2,  2.8,  3.8)

cor.test(x, y, method = "kendall", alternative = "greater")
#>
#>  Kendall's rank correlation tau
#>
#> data:  x and y
#> T = 26, p-value = 0.05972
#> alternative hypothesis: true tau is greater than 0
#> sample estimates:
#>       tau
#> 0.4444444
```

# Modèles linéaires

Avec la fonction `lm`. Exemple :

```
res.lm <- lm(energy ~ proteins + sugar + fibers + water,
             data = fruveg)
summary(res.lm)
#>
#> Call:
#> lm(formula = energy ~ proteins + sugar + fibers + water, data = fruveg)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -8.2934 -2.0977  0.1793  1.9007  9.6594
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 218.9887    39.1097   5.599 5.41e-06 ***
#> proteins      1.6572     1.0722   1.546 0.133447
#> sugar         1.8294     0.4358   4.198 0.000247 ***
#> fibers       -2.0362     0.5227  -3.896 0.000556 ***
#> water        -2.1116     0.4012  -5.263 1.35e-05 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 3.607 on 28 degrees of freedom
#> Multiple R-squared:  0.9391, Adjusted R-squared:  0.9304
#> F-statistic: 107.9 on 4 and 28 DF,  p-value: < 2.2e-16
```

# Pour aller plus loin

# Analyse en composantes principales

Les packages

- `FactomineR`
- `factoextra`

Les références :

- le site de François Husson https://husson.github.io/,
- la page sur l'ACP https://husson.github.io/MOOC_AnaDo/ACP.html
- Une référence en anglais par Hervé Abdi

# Les modèles linéaires à effets mixte

Les packages

- `lme4`
- `lmerTest`
- `multcomp`

Une référence (parmi d'autres) : Mixed Models with R