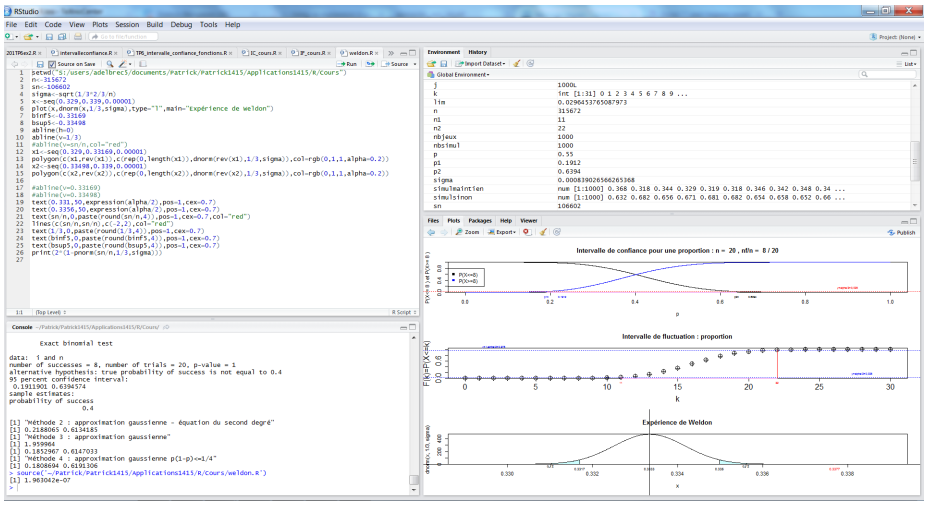
Prise en main de RStudio et de R

RStudio est un environnement de développement intégré (IDE) facilitant la saisie, l'exécution de code et la visualisation des résultats. Il comprend quatre zones (*panels* en anglais).

1. Une zone permet l'*édition de fichiers* source R (*the source*).
   File → *New File* → *R Script*
2. Une autre zone affiche la *console* avec la session R en cours d'exécution (*the console*).
3. Une troisième zone permet de basculer entre l'affichage des objets de l'espace de travail en cours (Workspace) et l'historique des commandes exécutées (*the environment or history panel*).
4. Une quatrième zone (*the panel for help, plots and others*) permet de basculer entre :
   - un navigateur d'aide qui permet à la fois la navigation dans l'aide en ligne intégrée à R et l'affichage des pages d'aide des différentes fonctions,
   - la fenêtre d'affichage et d'export des graphiques,
   - une liste des extensions installées, qui permet de les charger en mémoire ou d'en installer de nouvelles,
   - un navigateur de fichiers.

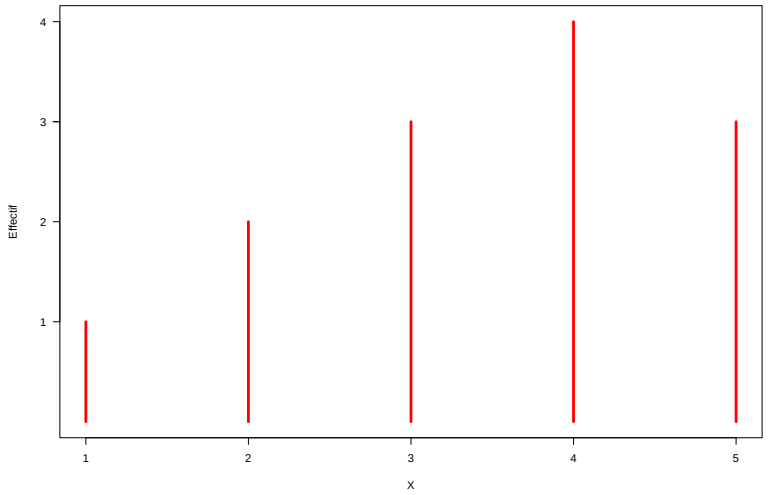
Pour visualiser les différentes parties : *View* → *Panes* → *Show All Panes* ou différents modes d'affichage : *View...*



La console fonctionne comme un calculateur, avec un rappel possible des commandes grâce aux flèches *haut* et *bas*.
Il vaut mieux écrire des *scripts* dans la zone d'édition. Ils peuvent être exécutés ligne à ligne (*Ctrl R* ou *Run*) ou globalement (*Source* ou *Source on save*).
Si le bouton *Run* ou *Source on save* n'apparaissent plus : *Code -> Source with echo.*
Effacer l'affichage de l'espace de travail : *Edit* → *Clear Console* ou encore *Ctrl L*
Quelques indications pour débuter :

1. Affectation des variables : <- ou =
   Exemples
   mavariable<- 3
   autre = 4

2. Commentaires : #
3. Opérateurs de comparaison : ==, ! =, <, <=, >, >=.
4. Exemples de **script** :
   - Exemple 1
     s<-0
     for (i in 1:25) {
     s<-s+i^2
     }
     print(s)
   - Exemple 1
     #Diagramme en bâtons
     x<-c(1,2,2,3,3,3,4,4,4,4,5,5,5)
     tabx<-table(x)
     plot(tabx,lwd=4,col="red",yaxt="n",xlab="X",ylab="Effectif",main="Diagramme en bâtons")



5. *Export* → *Save as PDF* (ou *Image*).
6. Pour connaître le répertoire de travail utilisé par R : getwd()
7. Changement de répertoire
   - setwd()
     Exemple : setwd("C :/documents/Applications/R/TP")
     Attention avec le répertoire de travail : si on l'écrit setwd('...'), les anti-slash windows (\) doivent être remplacés par des slash (/)
     Remarque : utiliser / ou // pour remplacer \.
     dir() pour lister les fichiers du répertoire de travail.
   - autre possibilité : Session → Set Working Directory → To Source File Location
8. Un **objet** est un espace dans lequel on peut stocker des éléments.
   - Un **vecteur** est un objet d'un même *mode* (numérique, caractère, logique, vide) pour toutes les valeurs qui le constituent. Il est constitué de composantes.
   - Une **matrice** est un objet d'un même mode (numérique, caractère, logique, vide) qui peut contenir m lignes et n colonnes.
   - Une **liste** est un objet permettant de stocker des objets qui peuvent être hétérogènes, c'est-à-dire qui n'ont pas tous le même mode ou la même longueur.

- Un **tableau de données** ou *data.frame* est une liste particulière dont les composantes sont de même longueur et dont les modes peuvent être différents. Il s'agit d'un tableau à double entrée : les lignes sont les individus sur lesquels les mesures sont faites et les colonnes sont les variables.
- Un **facteur** est un vecteur particulier qui permet de manipuler des variables qualitatives.

9. **Vecteurs** (*vector*)
   - Création avec c(), : ou seq
     Exemples
     vecteur1<-c(2,6,8,9)
     vecteur2<-2:9
     vecteur3<-seq(1,10,2)
   - Nombre de termes dans un vecteur : *length*.
   - Renvoi des indices pour lesquels le résultat d'une opération logique est vrai : *which*
     Exemple
     ve<-(0:10)
     which(ve==4)
     renvoie 5
   - Initialisation :
     u<-vector() #initialisation du vecteur ou encore u<-numeric() ou u<-rep(x=NA,times=6)
     Exemples
     u<-vector()
     for (i in seq(0,5,1)){
     u<-c(u,3+i*4)
     }
     print(u)
     print(sum(u))
10. **Indicateurs statistiques**
    - de positionnement : mean(), median(), min(), max(), quantile().
      Remarque : 9 types de quantiles existent dans R. Par défaut, le type 7 est utilisé par la fonction quantile. Préciser type=2 pour la définition du cours.
    - de dispersion : var(), sd(). Il s'agit de la variance : $\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})^2$ et de l'écart-type : $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})^2}$ corrigés ou non biaisés (*sample variance* et *sample standard deviation*).
      Remarque : la variance $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^2$ et l'écart-type σ de la population sont données par popvar() et popsd() de la librairie rafalib.
    - La fonction *summary* permet d'obtenir un résumé dans un tableau de valeurs : moyenne, min, max...
      print(summary(x,quantile.type=2))

11. **Lois de probabilité**
    - Fonctions d, p, q, r
      Ü Si *nomloi* désigne sous une loi de probablité alors *dnomloi()*, *pnomloi()*, *qnomloi()* et *rnomloi()* représentent respectivement la fonction de densité de probablité, la fonction de répartition, la réciproque de cette dernière et la fonction de génération aléatoire de cette loi.
      Exemple : loi normale notée *norm*
      - ☐ dnorm() représente la fonction de densité de probabilité de la loi normale (d pour densité) .
      - ☐ pnorm() représente la fonction de répartition de la loi normale (p pour probabilité).
      - ☐ qnorm() représente la fonction réciproque de la fonction de répartition de la loi normale (q pour quantile).
      - ☐ rnorm() représente la fonction permettant de faire des tirages aléatoire selon une loi normale (r pour random).

Loi	nom R	d	p	q	r
Uniforme	unif	dunif	punif	qunif	runif
Binomiale	binom	dbinom	pbinom	qbinom	rbinom
Poisson	pois	poisd	poisp	poisq	poisr
Exponentielle	exp	expd	expp	expq	expr
Normale	norm	dnorm	pnorm	qnorm	rnorm

    - Arguments
      Le premier argument des fonctions *nomloi* est nommé de la façon suivante :
      - ☐ dnomloi(x) avec x comme dans la fonction de densité de probabilité f(x) un vecteur de valeurs possibles pour une variable aléatoire suivant la loi *nomloi*.
      - ☐ pnomloi(q) avec q pour quantile, un vecteur de valeurs possibles pour une variable aléatoire suivant la loi *nomloi*.
      - ☐ qnomloi(p) avec p pour probabilité, un vecteur de probabilités.
      - ☐ rnomloi(n) avec n un entier donnant le nombre total de tirages aléatoire voulu.

      Exemple 1 : loi de probabilité de la loi binomiale B(4,0.25)
      dbinom(0 :4,4,0.25)
      Exemple 2 : quantile associé à 0,975 pour la loi normale centrée réduite
      qnorm(0.975)
      ou encore
      qnorm(0.975,0,1)

12. **Fonctions R**
    nom <- function(arg1, arg2, ...) expression
    Exécution : nom(arg1, arg2, ...)
    Exemple 1
    f <- function(x) { x^2 }
    Exemple 2
    #Calcul de la moyenne
    moyenne<-function(x){# x : vecteur
    n<-length(x) #longueur du vecteur x
    moy<-sum(x)/n
    moy #la fonction retourne le résultat

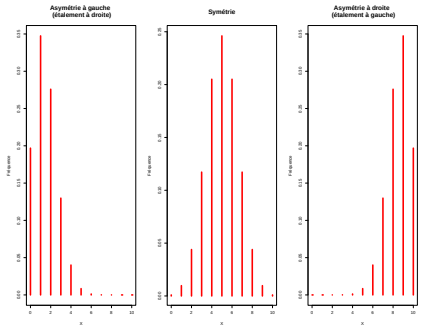
}

13. **Fonctions mathématiques**

Nom	Fonction
ln	log()
exponentielle	exp()
factorielle	factorial()
coefficient binomial $C_n^k = \begin{pmatrix} n \\ k \end{pmatrix}$	choose(n,k)

14. **Représentation graphique d'une fonction mathématique**
La représentation graphique d'une fonction mathématique est possible en R avec la fonction *curve*.
Exemples
curve(expr = log)
curve(expr = x^2 + 5*x - 25, from = -23, to = 18, n = 200)
qui correspond à
x <- seq(from = -23, to = 18, length = 200)
plot(x = x, y = x^2 + 5*x - 25, type = "l",col="red")#couleur rouge pour différencier

15. **Représentation graphique**
plot(), hist(), boxplot(), barplot().
Exemple :
x<-sample(1 :10,100) #x<-runif(50, min=40, max=60) autre exemple
hist(x) # histogramme
plot(ecdf(x)) pour représenter la distribution des données d'un vecteur x et les effectifs cumulés croissants (*empirical cumulative distribution function*).
On peut représenter plusieurs graphiques sur une même image : par().
Exemple : par(mfrow = c(1,3)) permet l'affichage de trois graphiques à l'horizontale.
Pour réinitialiser à un seul graphique par image : par(mfrow = c(1,1))



Remarques pour boxplot :
En posant b<-boxplot(x,type=2), b$stats et b$n permettent de connaître les quartiles et le nombre de valeurs utilisées par boxplot.
On peut retrouver ces éléments par print(boxplot(x,type=2,plot=FALSE).
Pour pouvoir avoir une boîte à moustaches avec les quantiles de type 2, il faut utiliser la fonction qboxplot qui nécessite le package du même nom. On peut lister les valeurs de la boîte à moustache par qboxplot.stats.

16. **Intégration**
integrate(f,lower=...,upper=...)

Exemple 1 :
f <- function(x) { x^2 }
integrate(f,lower=0,upper=1)
Exemple 2 : fonction de répartition de la loi normale
integrate(dnorm, mean=0, sd=1, lower= -Inf, upper= 1.96, abs.tol = 0)
integrate(dnorm, mean=100, sd=110, lower= -Inf, upper= 110, abs.tol = 0)
#absolute accuracy requested
integrate(dnorm, mean=100, sd=110, lower= 100, upper= 110, abs.tol = 0)
Remarque : la sortie n'est pas numérique (valeur et précision).
Pour pouvoir récupérer la valeur numérique de l'intégrale et faire des calculs, par exemple :
temp<-integrate(f,lower=0,upper=1)
temp$value #valeur numérique de l'intégrale

17. **Paquets et librairies** (*package*, *library*)
    On peut télécharger des packages (bibliothèques externes) et les installer sur l'ordinateur
    (a) avec la fonction *install.packages()* dans la zone *source*
    (b) ou *Tools -> Install packages*

    La fonction ***library()*** permet de charger le package et rendre les fonctionnalités disponibles dans R (le faire à chaque ouverture de R).
    Exemple : *swirl* (paquet de R pour apprendre R)
    install.packages("swirl")
    library(swirl)
    swirl() # pour appeler le paquet *swirl* et démarrer.
    Autres exemples : *e1071* (pour le calcul des moments centrés : moment(x, order=3, center=TRUE)), *dplyr* (paquet de R contenant de nombreuses fonctions facilitant la manipulation de données).
18. *R.Version()* pour avoir des informations sur la version de R.

Sources :
www.quanti.hypotheses.org/488/
www.edutechwiki.unige.ch/fr/Premiers_pas_avec_R
www.r-tutor.com/elementary-statistics
Initiation à la statistique avec R - Frédéric Bertrand - Myriam Maumy-Bertrand - Dunod