Résumé Matlab

M8

```
%%% Manipulation d'objets %%%

vl = linspace(a, b, n)  % génère n points entre a et b
ma = ones(n,p)          % matrice n x p de 1
n = length(x)           % plus grande dimension de x
[n p] = size(x)         % hauteur et largeur de x

vc = x(:, p)            % pième colonne de la matrice
vl = x(n, :)            % nième ligne de la matrice
ma = x(3:4, 2:5)        % sous-matrice (2 x 3)
vc = x(1:2:end, 1); % matrice contenant un élément sur 2 de la 1ère colonne de x

vc = diag(x)            % matrice des éléments sur la diagonale de x
vc = find(x > a & x <= b)  % indices des éléments de x correspondants

%%% calculs courants %%%

n = mean(x)      % moyenne de x
n = median(x)    % mediana de x
n = mode(x)      % mode de x
n = var(x,1)     % variance de x
n = std(x,1)     % écart type de x
n = min(x)       % minimum de x
n = max(x)       % maximum de x
vc = sum(x)      % somme de x
vc = cumsum(x)   % somme cumulée de x

f = x/sum(x)
F = cumsum(f)
Fc = F - 1/2*( F - [0 F(1:n -1)]); % fct. de répartition empirique (cas continu)

%% matrice centrée réduite
xcr = (x-ones(n,1)*mean(x))./(ones(n,1)*std(x,1))

%% discretiser x en nbin intervales
d=linspace(min(x),max(x),nbin+1);
for i=1:nbin
        H(i)=length(find(x>d(i)&x<=d(i+1)));
end

%% nombre d'observations a partir d'un échantillon
n = length(a);
va=[]; % valeurs
na=[]; % nb occurences
for i=min(x):max(x)
    p=length(find(x==i));
    if p > 0
        va=[va i];
        na=[na p];
    end
end
```

```
%%% Affichage %%%

plot(x, y, 'o')     % affichage d'une courbe (x, y, paramètres)
bar(x, y)           % histogramme discret (x valeurs, y effectifs)
hist(x, n)          % histogramme continu de x en n espaces
boxplot(x)          % boite a moustache de x
ster(x)             % variation des résidus x

hold on/off         % continuer à dessiner par-dessus la figure
figure(n)           % affichage dans la figure n (évite de réécrire dessus)
subplot(n, p, num)  % met le prochain affichage dans la case num du subplot n×p
x/ylabel(text);     % titre des axes x et y
axis([minx maxx miny maxy]); % change les axes
title(text);        % titre de la figure

%%% ACP %%%

Xr = (X - e*mean(X))    % matrice centrée
Xn = Xr./(e*std(X,1))   % matrice centrée réduite
cov = 1/n*(Xr)'*(Xr)    % matrice de covariance
cor = 1/n*(Xn)'*(Xn)    % matrice de corrélation
[v,d] = eig(Xn'*Xn)     % l'ACP
u = Xn*v                % projections sur les axes de l'ACP
Vn = v*((d/n).^(1/2))   % role des variables

%%% Régression linéaire %%%

e = ones(n,1)                     % vecteur unitaire
a = (X'*X)\(X'*y)                 % coefs de la régression
z = X*a                           % prévision
r = y-z                           % résidus
SCM = sum((z-mean(y)).^2)
SCT = sum((y-mean(y)).^2)
R2 = SCM/SCT                      % R²
H = X*(X'*X)^(-1)*X'              % matrice des contributions
h = diag(H)                       % contributions des observations
s2_r = (1/(n-p))*r'*r             % variance estimée
r_s = r./((s2_r*(e-h)).^(1/2))    % résidus standardisés
c = h./(p*(e-h).^2).*r.^2/s2_r    % contributions
err_VC = sum((r.^2)./((e-h).^2))  % erreur de validation croisée

%%% Tests %%%

pval = cdf('norm', u, m, s2)    % loi normale : P(N(m, s2) < u)
pval = cdf('chi2',t,n))         % loi du chi2 : P(D_n < t)
pval = cdf('T',t,n))            % loi de student : P(T_n < t)
```