République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieure et de la Recherche Scientifique
Université des sciences et de la Technologie Houari Boumediene

Faculté de Mathématique

Département de Recherche Opérationnelle

Rapport

Méthodes de Prévision
Lissage exponentiel

- **Présenté par :**
  - BERRIOUECHE Sabrina
- **Professeur du module :**
  - Mr CHAABANE Djamal

2021 - 2022

Première partie

SUR LE PLAN THÉORIQUE

0.0.1 Lissage exponentiel simple :

Avec l'absence de la tendance et de la saisonnalité.

- **Algorithme** :

Étape 01 : (Initialisation)

```
        La valeur initiale du lissage est fixée par
            x̂1 = x1 ou x̂1 = x̄
        Lire α
```

Étape 02 :

```
        Pour j = 2 à N
            on calcule x̂1 = x̂j−1+(1 − α)(xj − x̂j−1)
        Fin pour;
```

- **Justification de l'algorithme** :

La valeur $\widehat{X}_N(h) =$ fournie par le modèle du lissage exponentiel simple avec la constante de lissage α (entre 0 et 1) est

$$\widehat{X}_N(h) = \widehat{X}_N + h = (1-\alpha)\cdot\sum_{j=0}^{N-1}\alpha^j X_{N-j} \tag{1}$$

On remplace N par $N-1$ dans (1)

$$\widehat{X}_{N-1}(h) = \widehat{X}_{N-1} + h = (1-\alpha)\cdot\sum_{j=0}^{N-1}\alpha^j X_{N-1-j}$$

On multiplie par α

$$\alpha\cdot\widehat{X}_{N-1}(h) = \widehat{X}_{N-1} + h = (1-\alpha)\cdot\sum_{j=0}^{N-2}\alpha^{j+1} X_{N-(j+1)}$$

$$\alpha\cdot\widehat{X}_{N-1}(h) = (1-\alpha)\sum_{j=0}^{N-2}\alpha^{j'} X_{N-j'} \tag{2}$$

On fait la soustraction de $(1)-(2)$, on aura

$$\begin{aligned}
\widehat{X}_N - \alpha\cdot\widehat{X}_{N-1}(h) &= (1-\alpha)\cdot\sum_{j=0}^{N-1}\alpha^j X'_{N-j} - (1-\alpha)\sum_{j=0}^{N-2}\alpha^{j'} X_{N-j'} \\
&= (1-\alpha)\cdot(\sum_{j=0}^{N-1}\alpha^j X'_{N-j} - \sum_{j=0}^{N-2}\alpha^{j'} X_{N-j'}) \\
&= (1-\alpha)\cdot X_N
\end{aligned}$$

Qui donne la formule de lissage simple suivante :

$$\widehat{X}_N = \widehat{X}_{N-1} + (1-\alpha)(X_N - \widehat{X}_{N-1})$$

Et la valeur de premier lissage $\widehat{X}_1 = X_1$.

0.0.2 Lissage exponentiel double :

Avec la présence de la tendance .

- **Algorithme** :

Étape 01 : (Initialisation)

$S_1(1) = X$ et $S_2(1) = (1-\alpha)^2 \cdot X_1$

Étape 02 :

Pour $j = 2$ à N

$S_1(j) = \alpha \cdot S_1(j-1) + (1-\alpha) \cdot X_j$

$S_2(j) = \alpha \cdot S_2(j-1) + (1-\alpha) \cdot S_1(j)$

Fin pour ;

Étape 03 :

On calcule

$\widehat{\alpha}_1(N) = 2S_1(N) - S_2(N)$

$\widehat{\alpha}_2(N) = \frac{(1-\alpha)}{\alpha}(S_1(N) - S_2(N))$

Étape 04 :

Prévision

$\widehat{X}_{N+h} = \widehat{\alpha}_1(N) + \widehat{\alpha}_2(N) + h$

- **Justification** :

Le modèle de lissage double est donné par la formule suivante :

$$\widehat{X}_{N+h} = \widehat{\alpha}_1(N) + \widehat{\alpha}_2(N) + h$$

On cherche $\widehat{\alpha}_1(N)$ et $\widehat{\alpha}_2(N)$ par

$$\min_{\widehat{\alpha}_1,\widehat{\alpha}_2} S^2 = ||\alpha(X_t - \widehat{\alpha}_1 - \widehat{\alpha}_2(t-N))||^2 \Longrightarrow Somme = \sum_{j=0}^{N-1} \alpha^j (X_{N-j} - \widehat{\alpha}_1 + \widehat{\alpha}_2 \times j)^2$$

On dérive la formule "somme" par rapport à $\widehat{\alpha}_1$ et $\widehat{\alpha}_2$, on ontient

$$\begin{cases} \frac{\partial S^2}{\partial \widehat{\alpha}_1} = \frac{\partial}{\partial \widehat{\alpha}_1}(\sum_{j=0}^{N-1} \alpha^j (X_{N-j} - \widehat{\alpha}_1 + \widehat{\alpha}_2 \times j)^2) = -2\sum_{j=0}^{N-1} \alpha^j (X_{N-j} - \widehat{\alpha}_1 + \widehat{\alpha}_2 \times j)^2 \\ \\ \frac{\partial S^2}{\partial \widehat{\alpha}_2} = \frac{\partial}{\partial \widehat{\alpha}_2}(\sum_{j=0}^{N-1} \alpha^j (X_{N-j} - \widehat{\alpha}_1 + \widehat{\alpha}_2 \times j)^2) = -2\sum_{j=0}^{N-1} j\alpha^j (X_{N-j} - \widehat{\alpha}_1 + \widehat{\alpha}_2 \times j)^2 \end{cases}$$

On s'annule les deux dérivées, on aura

$$\begin{cases} \frac{\partial S^2}{\partial \widehat{\alpha}_1} = 0 \\ \\ \frac{\partial S^2}{\partial \widehat{\alpha}_2} = 0 \end{cases} \Longrightarrow \begin{cases} -2\sum_{j=0}^{N-1} \alpha^j (X_{N-j} - \widehat{\alpha}_1 + \widehat{\alpha}_2 \times j)^2 = 0 \\ \\ -2\sum_{j=0}^{N-1} j\alpha^j (X_{N-j} - \widehat{\alpha}_1 + \widehat{\alpha}_2 \times j)^2 = 0 \end{cases}$$

$$\Longrightarrow \begin{cases} \widehat{\alpha}_1 \sum_{j=0}^{N-1} \alpha^j - \widehat{\alpha}_2 \sum_{j=0}^{N-1} j\alpha^j = \sum_{j=0}^{N-1} \alpha^j X_{N-j} \\ \\ \widehat{\alpha}_1 \sum_{j=0}^{N-1} j\alpha^j - \widehat{\alpha}_2 \sum_{j=0}^{N-1} j^2\alpha^j = \sum_{j=0}^{N-1} j\alpha^j X_{N-j} \end{cases} \quad (3)$$

Quand N tend vers ∞, on aura

$$\begin{aligned}\sum_{j=0}^{N-1}\alpha^j &= \frac{1}{1-\alpha}\\ \sum_{j=0}^{N-1}j\alpha^j &= \frac{\alpha}{(1-\alpha)^2}\\ \sum_{j=0}^{N-1}j^2\alpha^j &= \frac{\alpha(\alpha+1)}{(1-\alpha)^3}\end{aligned}$$

On remplace (3)

$$\begin{cases}\widehat{\alpha}_1-\widehat{\alpha}_2\frac{1}{1-\alpha}=(1-\alpha)\sum_{j=0}^{N-1}\alpha^jX_{N-j}\\ \widehat{\alpha}_1\alpha-\widehat{\alpha}_2\frac{\alpha(\alpha+1)}{(1-\alpha)}=(1-\alpha)^2\sum_{j=0}^{N-1}j\alpha^jX_{N-j}\end{cases}$$

On définie les lissage comme suit

$$\begin{cases}S_1(N)=(1-\alpha)\sum_{j=0}^{N-1}\alpha^jX_{N-j}\\ S_1(N)=(1-\alpha)\sum_{j=0}^{N-1}\alpha^jS_1(N-j)\end{cases}\Rightarrow\begin{cases}S_1(N)=(1-\alpha)\sum_{j=0}^{N-1}\alpha^jX_{N-j}\\ S_1(N)=(1-\alpha)\sum_{j=0}^{N-1}j\alpha^jX_{N-j}+(1-\alpha)S_1(N)\end{cases}$$

Qui donne le système d'équations linéaire suivant

$$S_1(N)=\widehat{\alpha}_1-\widehat{\alpha}_2\frac{1}{1-\alpha}$$

$$S_2(N)-(1-\alpha)S_1(N)=\widehat{\alpha}_1\alpha-\widehat{\alpha}_2\frac{\alpha(\alpha+1)}{(1-\alpha)}$$

On détermine $\widehat{\alpha}_1$ et $\widehat{\alpha}_2$

$$\begin{aligned}\widehat{\alpha}_1 &= 2\cdot S_1(N)-S_2(N)\\ \widehat{\alpha}_2 &= \frac{1}{1-\alpha}(S_1(N)-S_2(N))\end{aligned}$$

Cela implique

$$\begin{aligned}S_1(N) &= \alpha\cdot S_1(N-1)+(1-\alpha)\cdot X_N\\ S_2(N) &= \alpha\cdot S_2(N-1)+(1-\alpha)\cdot S_1(N)\end{aligned}$$

Et les premiers valeurs de $S_1(N)$ et $S_2(N)$ sont X_1 et $(1-\alpha)^2X_1$ respevtivement.

Deuxième partie

SUR LE PLAN PRATIQUE

0.0.3 Implémentation des algorithmes en MATLAB :

Le code de lissage simple :

```
%code de lissage exponentiel simple
%Faire entrer la série chronologique X_N et alpha
%Etape 01 de lissage simple : Initialisation
%La première valeur de lissage
    lissage_simple(1)= X_N(1);
    lissage_initial(1)=lissage_simple(1);
 %Etape 02 de lissage simple : calcul de lissage
    for i = 2 :N+1
        lissage_simple(i) = lissage_initial(i-1)+(1-alpha)*(X_N(i-1)-lissage_initial(i-1));
        lissage_initial(i)=lissage_simple(i);
    end
 %Affichage des résultats
    lissage_simple;
    lissage_simple([1])=[]
```

Le code de lissage double :

```
%Code de lissage exponentiel double
    %Faire entrer la série chronologique X_N et alpha et h
    %Etape 01 : initialisation de lissage
        lissage_double(1)= X_N(1);
        S_1(1)=lissage_double(1);
        S_2(1)=(1-alpha)^2*X_N(2);
    %Etape 02 : calcul de lissage
        for i = 2 :N
            %le premier lissage
            S_1(i) = alpha*S_1(i-1)+(1-alpha)*X_N(i);
            %le deuxième lissage
            S_2(i) = alpha*S_2(i-1)+(1-alpha)*S_1(i);
        end
         S_1
        S_2
     %Etape 03 : calcul des alpha's
        alpha_chap_1(N) = 2*S_1(N)-S_2(N);
        alpha_chap_1([1 :N-1])=[]
        alpha_chap_2(N) = ((1-alpha)/alpha)*(S_1(N)-S_2(N));
        alpha_chap_2([1 :N-1])=[]
     %Etape 04 : prévision pour la période h
        X_chap_(N+h) = alpha_chap_1+alpha_chap_2+h;
        X_chap_(N+h)
    %Affichage des résultats
        S_1 = transpose(S_1)
        S_2 = transpose(S_2)
         alpha_chap_1 = transpose(alpha_chap_1)
         alpha_chap_2 = transpose(alpha_chap_2)
        X_chap = transpose(X_chap)
```

Code de génération des séries chronologiques aléatoires :

La fonction suivante prend "d_start" et "d_end" comme des entrées et sert à générer le temps :

```
%Génération de temps pour les séries chronologique
    function [ m ] = SerieChronologique(d_start,d_end)
        %La première date de la série
        date_start = datevec(d_start) ;
        %La dernière date de la série
         date_end = datevec(d_end) ;
        tmp = (1 :[date_end-date_start]*[12 1 0 0 0 0]')'-1 ;
        u = ones(size(tmp)) ;
        m = datenum([date_start(1)*u, date_start(2)+tmp, u*[1 0 0 0]]) ;
    end
```

Puis, on exécute le code suivant pour avoir les données d'une série chronologique sans tendance :

```
%Génération : Série chronologoique avec tendance
    %Génération : Temps
    %dd-mmm-d : format de la date entrée
    %Choisir une date de début et une date de fin
        date = datestr(SerieChronologique('d_start','d_end'),'dd-mmm-yy') ;
     %Génération : Données
        A = randi([1200 1950],24,1) ;
        val = sort(A) ;
     %Génération : Série chronologique
        esp=' '; espace=repmat(esp,24,1) ;
        time_serie = strcat([date,espace,int2str(val)])
```

Et on exécute le code suivant pour avoir les données d'une série chronologique avec tendance :

```
%Génération : Série chronologoique avec tendance
    %Génération : Temps
    %dd-mmm-d : format de la date entrée
    %Choisir une date de début et une date de fin
        date = datestr(SerieChronologique('d_start','d_end'),'dd-mmm-yy') ;
     %Génération : Données
        val = randi([1200 1950],24,1) ;
     %Génération : Série chronologique
        esp=' '; espace=repmat(esp,24,1) ;
        time_serie = strcat([date,espace,int2str(val)])
```

0.0.4 Applications :

1ère application : lissage exponentiel simple

Pour "$N = 12$", "$d\_start = 1 - jan - 2020$" et "$d\_end = 1 - jan - 2021$", on applique le code de prévision suivant pour :

- Générer une série chronologique sans tendance.
- Choisir une meilleure valeur de la constante α.

- Calculer le lissage simple.

```
display('1\LISSAGEEXPONENTIELSIMPLE : Sriechronologiquesanstendance')
%Donner une valeur à N : N désigne le nombre des élements de la série
    N = 12 ;
    %Génération : Série chronologoique sans tendance
        %Génération : Temps
        %'dd-mm-yy' : la forme de la date entrée
        %Choisir une date de début et une date de fin
            date = datestr(SerieChronologique('1-jan-2019','1-jan-2020'),'dd-mmm-yy') ;
        %Génération : Données
            val = randi([950 2000],N,1) ;
        %Affichage : Série chronologique
            esp = ' '; espace = repmat(esp,N,1) ;
            serie_chronologique = strcat([date,espace,int2str(val)])
    %Lissage simple et le choix de la meilleure constante d'alpha
        %Déclaration des variables
            alpha = 0.01 ; j = 1 ;
            X_N = val ;
        %Initialisation de min de l'erreur carré et de la meilleure constante
        %d'alpha
            min_Err_carre = inf ;
            meilleur_alpha = 0.01 ;
        %faire varier le alpha
            while(alpha <= 0.99)
                %Etape 01 de lissage simple : Initialisation
                 %La première valeur de lissage
                lissage_simple(1) = X_N(1) ;
                lissage_initial(1) = lissage_simple(1) ;
                %Etape 02 de lissage simple : calcul de lissage
                for i = 2 :N+1
                    %Calcul : lissage simple
                    lissage_simple(i) = lissage_initial(i-1)+(1-alpha)*(X_N(i-1)-lissage_initial(i-
1)) ;

                    lissage_initial(i) = lissage_simple(i) ;
                    %Calcul : l'erreur
                    Err(i)=X_N(i-1)-lissage_simple(i) ;
                    %Calcul : l'erreur carré
                    Err_carre(i) = Err(i)^2 ;
                    %Calcul : la moyenne de l'erreur carré
                    Moyenne_Err_carre(j) = sum(Err_carre)/N ;
                end
                %La condition pour que alpha soit meilleure
                if (Moyenne_Err_carre(j)<min_Err_carre)
                    min_Err_carre = Moyenne_Err_carre(j) ;
                    meilleur_alpha = alpha ;
                end
                %Incrémentation
                alpha = alpha + 0.01 ;
                 j = j + 1 ;
                lissage_simple([1]) = [] ;
                Err([1]) = [] ;
```

```
        Err_carre([1]) = [];
    end
%Affichage des résultats
    lissage_exponentiel_simple = transpose(lissage_simple)
    Erreur = transpose(Err)
    Erreur_carre = transpose(Err_carre)
    meilleur_alpha
```

On aura les résultas suivants :

```
>> Prevision_lissage_simple
1\ LISSAGE EXPONENTIEL SIMPLE : Série chronologique sans tendance

serie_chronologique =

01-Jan-19    1547
01-Feb-19    1443
01-Mar-19     962
01-Apr-19    1304
01-May-19    1120
01-Jun-19    1784
01-Jul-19    1277
01-Aug-19    1505
01-Sep-19    1124
01-Oct-19    1582
01-Nov-19    1226
01-Dec-19    1637
```

```
lissage_exponentiel_simple =

   1.0e+03 *

    1.5470
    1.5449
    1.5333
    1.5287
    1.5205
    1.5258
    1.5208
    1.5205
    1.5126
    1.5139
    1.5082
    1.5108
```

```
Erreur =

         0
 -101.9200
 -571.2616
 -224.6764
 -400.5028
  258.2272
 -243.7973
  -15.4814
 -388.5518
   68.0593
 -282.1819
  126.2417
```

```
Erreur_carre =

   1.0e+05 *

         0
    0.1039
    3.2634
    0.5048
    1.6040
    0.6668
    0.5944
    0.0024
    1.5097
    0.0463
    0.7963
    0.1594


meilleur_alpha =

    0.0100

>>
```

<u>2ème application</u> : lissage exponentiel double

Pour "$N = 24$", "$d\_start = 1 - jan - 2020$" , "$d\_end = 1 - jan - 2020$" et $h = 9$, on applique le code de prévision suivant pour :

- Générer une série chronologique avec tendance.
- Choisir une meilleure valeur de la constante α.
- Calculer le lissage double.

---

display('1\$LISSAGEEXPONENTIELDOUBLE'$)
display(' Série chronologique avec tendance')
%Choisir une valeur pour N et h
N = 24 ; h = 9 ;
%Génération : Série chronologoique avec tendance
%Génération : Temps
%dd-mmm-d : format de la date entrée
%Choisir une date de début et une date de fin
date = datestr(SerieChronologique('1-jan-2020','1-jan-2022'),'dd-mmm-yy') ;
%Génération : Données
A = randi([950 2000],N,1) ;
val = sort(A) ;
%Affichage : Série chronologique

```
            esp = ' '; espace = repmat(esp,N,1);
            serie_chronologique = strcat([date,espace,int2str(val)])
%Lissage double et le choix de la meilleure constante d'alpha
      %Déclaration des variables
            alpha = 0.01; j = 1;
            X_N = val;
      %Initialisation de min de l'erreur carré et de la meilleure constante
       %d'alpha
            min_Err_carre = inf;
            meilleur_alpha = 0.01;
      %faire varier le alpha
      while(alpha <= 0.99)
      %Etape 01 : initialisation de lissage
            lissage_double(1 )= X_N(1);
            S_1(1) = lissage_double(1);
      S_2(1) = (1-alpha)^2*X_N(1);
      %Etape 02 : calcul de lissage
            for i = 2 :N
                  %le premier lissage
                  S_1(i) = alpha*S_1(i-1)+(1-alpha)*X_N(i);
                  %le deuxième lissage
                  S_2(i) = alpha*S_2(i-1)+(1-alpha)*S_1(i);
            end
      %Etape 03 : calcul des alpha's
            for i = 1 :N
                  alpha_chap_1(i) = 2*S_1(i)-S_2(i);
                  alpha_chap_2(i) = ((1-alpha)/alpha)*(S_1(i)-S_2(i));
            end
       %Etape 04 : prévision pour la période h
            for i = 1 :N
                  X_chap(i) = alpha_chap_1(i)+(alpha_chap_2(i)*h);
                   %Calcul : l'erreur
                  Err(i) = X_N(i)-X_chap(i);
                  %Calcul : l'erreur carré
                  Err_carre(i) = Err(i)^2;
            end
            %Calcul : la moyenne de l'erreur carré
                  Moyenne_Err_carre(j) = sum(Err_carre)/N;
            %La condition pour que alpha soit meilleure
                  if (Moyenne_Err_carre(j)<min_Err_carre)
                        min_Err_carre = Moyenne_Err_carre(j);
                        meilleur_alpha = alpha;
                  end
            %Incrémentation d'alpha
                  alpha = alpha + 0.01;
                  j = j + 1;
      end
       %Affichage des résultats
            S_1 = transpose(S_1)
            S_2 = transpose(S_2)
      alpha_chap_1 = transpose(alpha_chap_1)
      alpha_chap_2 = transpose(alpha_chap_2)
```

X_chap = transpose(X_chap)
Erreur = transpose(Err)
Erreur_carre = transpose(Err_carre)
meilleur_alpha

On aura les résultats suivants :

```
>> prevision_lissage_double
1\ LISSAGE EXPONENTIEL DOUBLE
   Série chronologique avec tendance

serie_chronologique =

01-Jan-20    1006
01-Feb-20    1029
01-Mar-20    1086
01-Apr-20    1156
01-May-20    1205
01-Jun-20    1213
01-Jul-20    1250
01-Aug-20    1317
01-Sep-20    1319
01-Oct-20    1349
01-Nov-20    1447
01-Dec-20    1507
01-Jan-21    1527
01-Feb-21    1546
01-Mar-21    1565
01-Apr-21    1597
01-May-21    1742
01-Jun-21    1745
01-Jul-21    1768
01-Aug-21    1805
01-Sep-21    1823
01-Oct-21    1913
01-Nov-21    1926
01-Dec-21    1931
```

```
S_1 =

   1.0e+03 *

    1.0060
    1.0065
    1.0081
    1.0110
    1.0149
    1.0189
    1.0235
    1.0293
    1.0351
    1.0414
    1.0495
    1.0587
    1.0680
    1.0776
    1.0874
    1.0975
    1.1104
    1.1231
    1.1360
    1.1494
    1.1629
    1.1779
    1.1928
    1.2076
```

```
S_2 =

    0.4024
   20.5236
   40.2741
   59.6888
   78.7928
   97.5940
  116.1116
  134.3763
  152.3915
  170.1720
  187.7591
  205.1775
  222.4348
  239.5381
  256.4944
  273.3154
  290.0577
  306.7190
  323.3051
  339.8270
  356.2879
  372.7197
  389.1220
  405.4916
```

```
alpha_chap_1 =

   1.0e+03 *

    2.0116
    1.9924
    1.9758
    1.9623
    1.9510
    1.9401
    1.9308
    1.9243
    1.9179
    1.9127
    1.9113
    1.9122
    1.9137
    1.9157
    1.9182
    1.9218
    1.9308
    1.9395
    1.9487
    1.9590
    1.9695
    1.9830
    1.9966
    2.0097
```

```
alpha_chap_2 =

   20.5224
   20.1212
   19.7505
   19.4147
   19.1040
   18.8012
   18.5176
   18.2647
   18.0152
   17.7805
   17.5871
   17.4184
   17.2573
   17.1034
   16.9562
   16.8210
   16.7423
   16.6613
   16.5860
   16.5219
   16.4609
   16.4317
   16.4024
   16.3696
```

```
X_chap =

   1.0e+03 *

    2.1963
    2.1735
    2.1536
    2.1371
    2.1229
    2.1093
    2.0975
    2.0887
    2.0800
    2.0727
    2.0696
    2.0689
    2.0690
    2.0696
    2.0708
    2.0732
    2.0815
    2.0895
    2.0980
    2.1077
    2.1176
    2.1309
    2.1442
    2.1570
```

```
Erreur =

   1.0e+03 *

   -1.1903
   -1.1445
   -1.0676
   -0.9811
   -0.9179
   -0.8963
   -0.8475
   -0.7717
   -0.7610
   -0.7237
   -0.6226
   -0.5619
   -0.5420
   -0.5236
   -0.5058
   -0.4762
   -0.3395
   -0.3445
   -0.3300
   -0.3027
   -0.2946
   -0.2179
   -0.2182
   -0.2260
```

```
Erreur_carre =

   1.0e+06 *

    1.4168
    1.3099
    1.1397
    0.9625
    0.8426
    0.8034
    0.7183
    0.5955
    0.5792
    0.5237
    0.3876
    0.3158
    0.2937
    0.2742
    0.2558
    0.2267
    0.1153
    0.1187
    0.1089
    0.0916
    0.0868
    0.0475
    0.0476
    0.0511
```

```
meilleur_alpha =

    0.9800

>>
```