

TP01- AJUSTEMENT LINÉAIRE

Objectif : ajustement linéaire, méthode des moindres carrés (MCO), coefficient de détermination

1. EXEMPLE FIL ROUGE

On cherche à « expliquer » les variations d'une variable quantitative Y (le poids de naissance de l'enfant, noté BWT) par une variable explicative X également quantitative (le poids de la mère noté LWT). Le modèle s'écrit sous la forme : $\hat{Y} = \beta_0 + \beta_1 X$

Les paramètres (inconnus) du modèle de régression sont β_0 , β_1 et σ^2 .

```
#-----
#           Exemple fil rouge
#-----
## Chargement des données
mesdonnees <- read.table("E:/Google Drive/Projects/IUT/MesCours/M3103-ModeleLineaire-
15h/TP15h/DataForTP/Poids_naissance.csv",header=TRUE,sep="\t")
mesdonnees <- transform(mesdonnees,LWT=LWT*0.4535923) # transformation du data.frame
des données pour recoder cette variable en kilogrammes (1 livre = 0.453 592 37 kg)
attach(mesdonnees) # Accès au nom des variables.

## Inspection graphique
plot(BWT~LWT,xlab="Poids de la mère", ylab="Poids de naissance de l'enfant")
# Nous observons une légère tendance d'augmentation du poids de l'enfant avec le poids
de la mère, même si cette relation n'est pas très nette.

## Calcul du coefficient de corrélation
cor(BWT,LWT, method="pearson")

#Estimation des paramètres
modele1 <- lm(BWT ~ LWT,data=mesdonnees) # On obtient un objet de classe "lm"
modele1
#La sortie R ci-dessus fournit les estimations par moindres carrés de  $\beta_0$  et de  $\beta_1$ . On
trouve dans l'exemple ci-dessus  $\hat{\beta}_0 = 2\,369.672$  et  $\hat{\beta}_1 = 9.765$ .
#Ce qui nous interesse :
#• Coefficients: ce tableau comprend quatre colonnes :
#- Estimate correspond aux estimations des paramètres de la droite de régression;
#- Std. Error correspond à l'estimation de l'écart type des estimateurs de la droite de
#régression ;
#• Residual standard error: une estimation de l'écart type du bruit  $\sigma$  est fournie ainsi
#que le degré de liberté associé  $n-2$ .
#• Multiple R-squared: valeur du coefficient de détermination  $R^2$  (pourcentage de
#variance expliqué par la régression).

# Représentation la droite de régression sur le nuage de points
plot(BWT~LWT,xlab="Poids de la mère",ylab="Poids de l'enfant")
abline(modele1,col="blue")
```

1.1. Exercice

Nous donnons les couples d'observations suivants :

x_i	18	7	14	31	21	5	11	16	26	29
y_i	55	17	36	85	62	18	33	41	63	87

1. Enregistrer les dans un format adapté pour une lecture par la suite avec le logiciel R.
2. Tracer le diagramme de dispersion des couples (x_i, y_i) . A la vue de ce diagramme, pouvons-nous soupçonner une liaison linéaire entre ces deux variables ?
Le coefficient de corrélation mesure la relation entre deux variable
Calculer le coefficient de corrélation de Pearson entre ces deux variables. (Corrélation linéaire)
Calculer le coefficient de corrélation de Spearman entre ces deux variables. (Corrélation linéaire et monotone)
Calculer le coefficient de corrélation de Kendall entre ces deux variables
3. Etudier la symétrie des trois coefficients de corrélation précédents, c'est-à-dire lors de la saisie des données inverser le rôle de x et y. Etait-ce prévisible ?
4. Déterminer pour ces observations la droite des moindres carrés, c'est-à-dire donner les coefficients a et b de la droite des moindres carrés ($y = b + ax$). Utiliser la fonction *lm()* et stocker son résultat dans une variable appelée DroiteMCO_YparX
5. De la même manière, déterminer les coefficients de la droite des moindres carrés dans le cas ou X est la variable à expliquer, stocker ce résultat dans une variable appelée DroiteMCO_XparY
Ces coefficients sont-ils différents de ceux donnés à la question précédente ? Pourquoi ?
6. Donner les ordonnées des y_i calculés par la droite des moindres carrés fournie par DroiteMCO_YparX correspondant aux différentes valeurs des x_i . Utiliser la fonction *fitted()*.
7. Tracer ensuite la droite fournie par DroiteMCO_YparX sur le même graphique. Utiliser la fonction *abline()*.
8. Quelle est une estimation plausible de Y à $x_i = 21$ (la 5^{ème} valeur de la série x) ?
9. Quel est l'écart entre la valeur observée de Y à $x_i = 21$ et la valeur estimée avec la droite des moindres carrés ? Cet écart s'appelle un résidu (un résidu c'est le reste, l'erreur d'approximation)
10. Est-ce que la droite des moindres carrés obtenue en 2. passe par le point (\bar{x}, \bar{y}) (\bar{x} est la moyenne de la série x)? Pouvons-nous généraliser cette conclusion à n'importe laquelle droite de régression ?

1.2.Exercice (Etude sur l'intima-média)

Dans l'étude « Intima-média », on cherche à étudier la relation entre la mesure de l'épaisseur de l'intima-média et l'âge.

1. Récupérez le fichier de données sur l'intima-média. Décrivez les données
2. Tracez le nuage de points de la variable mesure en fonction de la variable AGE. Décrivez-le.
3. Existe-t-il une liaison entre ces deux variables ? Précisez l'indicateur de liaison qui permet de mesurer l'intensité entre ces deux variables.
4. On cherche maintenant à ajuster une droite de régression sur ce nuage de points :
 - Proposez un modèle de régression et estimez les paramètres du modèle ;
 - Tracez la droite obtenue sur le nuage de points.

1.3.Exercice

On considère 5 groupes de femmes âgées respectivement de 35, 45, 55, 65 et 75 ans. Dans chaque groupe, on a mesuré la tension artérielle en mm de mercure de chaque femme et on a calculé la valeur moyenne pour chaque groupe. On définit donc les variables :

Y : tension moyenne en mm(Hg) 114 124 143 158 166
Z : âge du groupe considéré 35 45 55 65 75

0. Plotter les données

1. Décrire la série en utilisant les principaux indicateurs statistique (fonction *summary()*)

2. Calculer le coefficient de corrélation de Pearson

3. Effectuer la régression linéaire simple de la tension moyenne en fonction de l'âge des échantillons, donner l'équation de prédiction du modèle et déterminer les coefficients de la droite des moindres canés.

4. Donner la qualité de l'estimation de ce modèle.

5. Représenter le nuage de points et la droite de régression associée.

1.4.Exercice (Droite de régression et points atypiques)

Douze personnes sont inscrites à une formation. Au début de la formation, ces stagiaires subissent une épreuve A notée sur 20. A la fin de la formation, elles subissent une épreuve B de niveau identique. Les résultats sont donnés dans le tableau suivant :

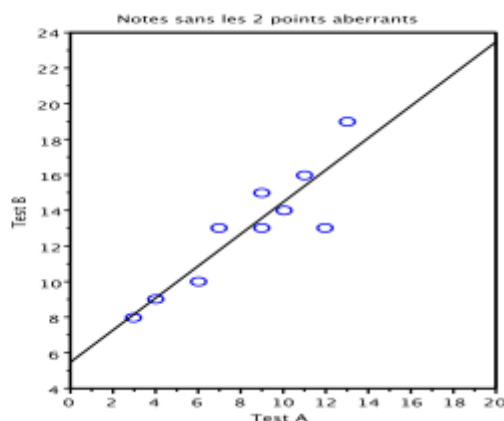
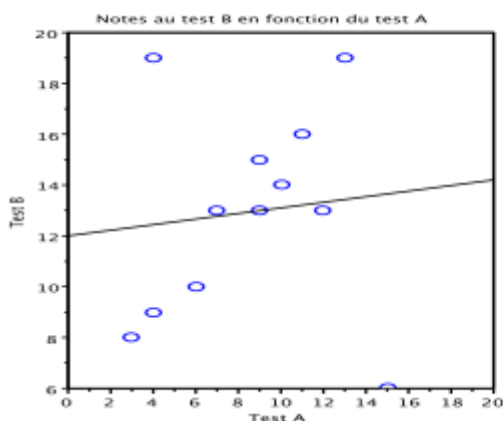
Epreuve A : 3 4 6 7 9 10 9 11 12 13 15 4

Epreuve B : 8 9 10 13 15 14 13 16 13 19 6 19

1. Représenter le nuage de points. Déterminer la droite de régression. Calculer le coefficient de détermination. Commenter.

2. Deux stagiaires semblent se distinguer des autres. Les supprimer et déterminer la droite de régression sur les dix points restants. Calculer le coefficient de détermination. Commenter.

3. Réaliser les figures ci-dessous dans une même figure



TP02- SERIES CHRONOLOGIQUES : PHASE EXPLORATOIRE

Objectif : série chronologique : phase exploratoire,

AIDE SOUS R pour les séries chronologiques

```
?ts #information sur objet « time-séries »
data() #tous les jeux de données disponibles en R
library() #nous allons utiliser souvent caschrono
data(package = .packages(all.available = TRUE))#jeux de données groupés par package
data(package = "MASS")
```

Décomposition de séries

```
is.ts(co2)
time(co2) #les dates de la série
frequency(co2) #la fréquence de la série
require(graphics) #appel d'une librairie
#decompose(co2, type = c("additive", "multiplicative"), filter = NULL)
m<-decompose(co2) # décompose une ts avec fréquence en 3 composantes : tendance,
saisonnier, résidus.
plot(m) #m$figure
tsp(co2) #Obtenir le début, la fin et la fréquence de la série
frequency(co2) #Fréquence de la série des temps
plot.ts(co2) #Représenter la série
matrix(co2,12,10) #Représentations annuelles de la série
matrix(co2,10,12) #Comparez
ts.plot(matrix(co2,12,10)) #Graphes
ts.plot(matrix(co2,10,12)) #Comparez
XX=window(co2,c(1970,4),1980) #Extraire une sous-série
```

Etude des résidus

```
re=m$random #appel des résidus généré par la fonction decompose
plot(m$random) #plot des résidus
res=CVS-CVS.lm$fitted.values #Définition des résidus
res=res/sqrt(var(res)) #Définition des résidus réduits
acf(res) #Corrélogramme des résidus
hist(res) #Histogramme des résidus
qqnorm(res) #Comparaison des quantiles des résidus
abline(0,1) # et des quantiles d'une loi normale
no=stl(co2, "per")
plot(no)
nx=no$time.s[, "remainder"]
acf(nx, na.action=na.pass)
```

Estimation de la tendance

```
y=time(CVS)
z=time(CVS)^2
CVS.lm=lm(CVS~y+z) #Estimation de la tendance :  $f(t) = a + bt + ct^2$ .
CVS.lm$coefficients #Valeurs des coefficients a, b et c
ts.plot(CVS)
ts.plot(time(CVS), CVS.lm$fitted.values)
```

1. Exercice (Tache solaires)

1. Charger les données *sunspots*. Décrire les données (sujet, type de données, nombre d'entrées, pas de temps, début, fin)

```
Data=sunspots
?sunspots
is.ts(Data)
time(Data)
start(Data)
end(Data)
frequency(Data)
```

2. Plotter la série (avec titre et libellés des axes).

3. Observez-vous une tendance ? une saisonnalité ?

4. Décomposer la série par un modèle additif (utiliser la fonction *decompose()*) et dresser le graphique de la décomposition de la série

5. On veut établir la droite de régression linéaire qui approxime les valeurs de la tendance.

Etablir la régression linéaire (fonction *lm()*) de la tendance par le temps (ici obtenue par la variable *time(ResultatDeLaDecomposition\$trend)*)

En vous appuyant sur les coefficients de la droite de régression donner une estimation de la tendance sur ces 11 années

6. Décomposer la série par un modèle multiplicatif (utiliser la fonction *decompose()*) et dresser le graphique de la décomposition de la série.

2. Exercice (Volume de fret pour les aéroports de Paris)

On considère le fichier *fret.txt* qui contient le volume de fret, exprimé en millions de tonnes, pour les aéroports de Paris entre janvier 1982 et décembre 2005 (source INSEE).

1. Charger les données. Réaliser la description de chacune des variables présentes dans ce fichier (type et codage, colonnes, début, fin, pas de temps).

```
library(readr)
data=read.table("E:/Google Drive/Projects/IUT/MesCours/M2102-Ajustement de courbes et
séries/TP10h/DataForTP/fret.txt", sep=";")
View(data)
data[1:10,]
```

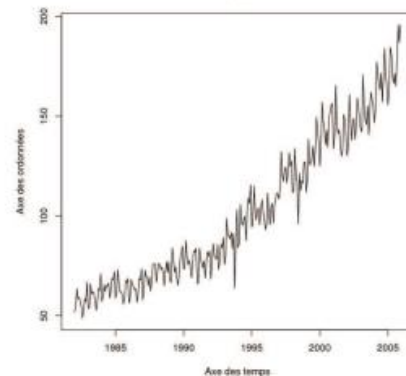
2. On souhaite sélectionner les données de fret (colonne 4, commande *Serie=data[,4]*), et les placer dans une variable nommée *série*. Comme les données originales vont de la valeur la plus récente à la valeur la plus ancienne, il faut « inverser » la série (commande *rev()*), vous stockerez cette série ordonnée dans une variable *SerieOrd*.

Après inversion, vérifier que la 100^{ème} valeur de votre série est 79.2

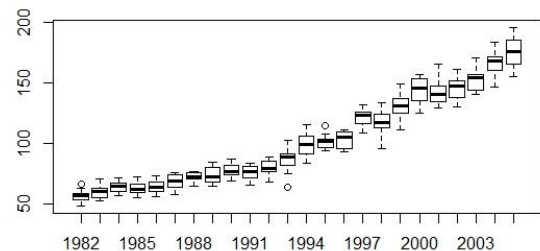
3. Pour visualiser une chronique, il convient dans un premier temps de transformer l'objet série de type *vector* en un objet de type *ts* (pour *time séries*). Ecrire et expliquer (les options) la ligne de code suivante :

```
série.ts=ts(série, start=c(1982,1), frequency=12)
```

4. Comparer le contenu des objets *sérieOrd* et *série.ts*. (regarder dans environnement). Quelle est la différence ?
5. Donner les indicateurs de statistique univariées classiques (moyenne, médiane, 1^{er} et 3^{ème} quartiles, écart-type, fonction *summary()* et *sd()*). Quelle est la moyenne mensuelle du fret (en millions de tonnes) sur l'ensemble des années d'étude.
6. Représenter la série comme ci-contre (fonction *plot.ts()*). Ajouter un titre.



7. On souhaite pour chaque année obtenir la représentation d'un boxplot. Afin de faciliter les comparaisons, on impose que les boxplots soient représentées sur le même graphique en parallèle. Ajouter titre et légende des axes.



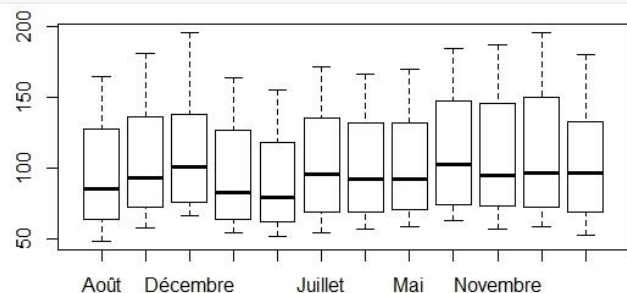
8. On veut également obtenir des informations quantitatives (moyennes, médianes, quartiles et extrêmes) concernant chaque année d'étude. Ecrire et expliquer les lignes de code suivantes :

```
for (i in 1982:2005){
  cat("année :", i, "\n")
  print(summary(data[data[,3]==i,4]))
}
```

9. On envisage de faire un travail similaire pour les mois.

Réaliser le boxplot de toutes les valeurs pour chaque mois (figure ci-contre). Ajouter titre et libellés des axes.

Comment pourrait-on améliorer cette figure ?



10. On veut compléter cette représentation graphique avec des informations quantitatives. S'inspirer de la boucle écrite en 8, pour générer une boucle donnant les principaux indicateurs statistiques pour chaque mois.

3. Exercice (Ventes)

1. Charger les données de vente *BjSales* et les expliquer (quoi ? début ? fin ? fréquence?)

```
Data= BjSales
?BjSales
```

2. Plotter la série (avec titre et libellés des axes).
3. Observez-vous une tendance ? une saisonnalité ?
4. Décomposer la série par un modèle additif (utiliser la fonction *decompose()*) et dresser le graphique de la décomposition de la série
5. Décomposer la série par un modèle multiplicatif (utiliser la fonction *decompose()*) et dresser le graphique de la décomposition de la série.

TP03- SERIES CHRONOLOGIQUES : MOYENNES MOBILES, DECOMPOSITION, MODELISATION, PREDICTION

Objectif : série chronologique : moyennes mobiles, décompositions, modélisation, prédiction

1. Exercice 1 (Volume de fret pour les aéroports de Paris)

On considère le fichier fret.txt qui contient le volume de fret, exprimé en millions de tonnes, pour les aéroports de Paris entre janvier 1982 et décembre 2005 (source INSEE).

A - Phase exploratoire

1. Charger le code ci-dessous. A la fin de chaque bloc, vous expliquerez ce que le bloc de code réalise.

```
-----Bloc 1-----
library(readr)
data=read.table("E:/Google Drive/Projects/IUT/MesCours/M2102-Ajustement de courbes et
séries/TP10h/DataForTP/fret.txt", sep=";")
View(data)
data[1:10,]
-----Bloc 2-----
série=data[,4]
série=rev(série)
série[100]
-----Bloc 3-----
série.ts=ts(série,start=c(1982,1),frequency=12)
-----Bloc 4-----
#série est simplement un vecteur
#série.ts est un objet spécifique de R « time-séries object » il s'agit d'une série
#chronologique indexée sur le temps
-----Bloc 5-----
sd(série.ts)
summary(série.ts)
-----Bloc 6-----
plot.ts(série.ts)
-----Bloc 7-----
boxplot(data[,4]~ data[,3])
-----Bloc 8-----
for (i in 1982:2005){
  cat("année :",i,"\n")
  print(summary(data[data[,3]==i,4]))
}
-----Bloc 9-----
boxplot(data[,4]~data[,2])
#On pourrait améliorer le graphique en organisant les abscisses selon la progression
usuelle des mois de l'année et non l'ordre alphabétique
-----Bloc 10-----
for (k in levels(data[,2])){
  cat("Mois :",k,"\n")
  print(summary( data[data[,2]==k,4]))
}
```

B - Phase de modélisation : modèle affine

1. On veut modéliser la série à l'aide d'un modèle affine de la forme $y_t = at + b$. Pour cela, on définit l'objet t qui contiendra les entiers entre 1 et 288 correspondant aux 288 mois d'étude (nombre de mois que l'on peut obtenir). Ecrire et expliquer les lignes de code suivantes:

```
length(série)      #Explication
t=1:288            #Explication
modell=lm(série~t)  #Explication (?lm)
```

Remarque: l'objet *modell* contient un certain nombre d'informations sur la modélisation. Pour obtenir l'ensemble de ces informations, regroupées dans différents objets, il suffit d'écrire *attributes(modell)*

2. Donner une estimation des paramètres a et b du modèle. Commenter la valeur de a .

3. On souhaite voir sur un graphique la série initiale et la modélisation de la composante tendancielle. Ecrire le code suivant :

```
tendance1.ts=ts(modell$fitted.values,start=c(1982,1),frequency=12)
ts.plot(série.ts,tendance1.ts,lty=c(1,2))
```

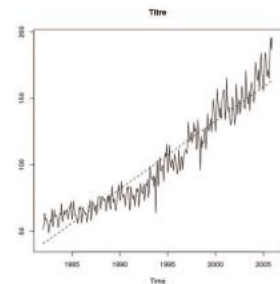
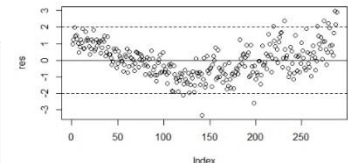


Fig. 6 : Modèle affine

4. Représenter le graphique des résidus normalisés ci-contre (fonction `plot()` et `abline()`). Les résidus normalisés sont générés de la façon suivante :

```
#génération des résidus normalisés
n=length(série)
s=sqrt((n-1)*var(modell$residuals)/n)
res=modell$residuals/s
```



C - Phase de modélisation : modèle exponentiel

1. On veut maintenant modéliser la série à l'aide d'un modèle exponentiel de la forme $y_t = \exp(at + b)$. Quel changement de variable doit-on réaliser afin de se ramener à un modèle affine ?

2. Calculer le logarithme de la série en écrivant le code suivant :

```
log.série=log(série)
```

3. Via la fonction `lm` écrire une ligne de code permettant de réaliser l'ajustement affine de `log.série` par rapport au temps t . On appellera ce nouveau modèle `model2`. Donner une estimation des paramètres a et b .

4. Comment obtient-on les valeurs estimées \hat{y}_t associées au modèle ?

5. Calculer l'exponentielle des valeurs estimées, puis représenter sur un même graphique la série initiale et la courbe de tendance obtenue via le modèle exponentiel. Ajouter le titre et les libelles de axes au graphique

```
tendance2=exp(model2$fitted.values);
tendance2.ts=ts(tendance2,start=c(1982,1),frequency=12)
ts.plot(série.ts,tendance2.ts,lty=c(1,2))
```

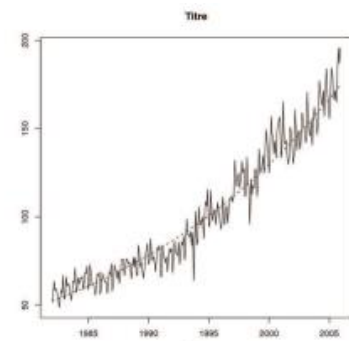
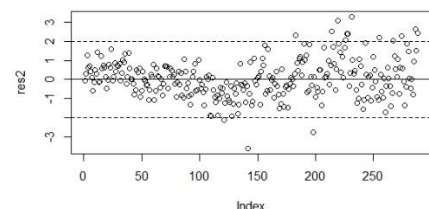


Fig. 7 : Modèle exponentiel

6. Comparer les représentations graphiques des modèles affine et exponentiel.

7. En vous inspirant de la question B4, sans aucune transformation (sauf une normalisation judicieuse) étudier les résidus associés à cette seconde modélisation (`residus2=série-tendance2`).



D - Modèle quadratique

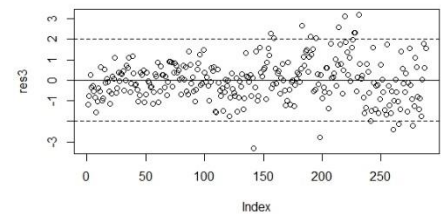
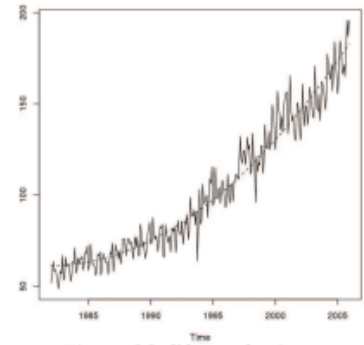
On se propose de modéliser la série via un modèle quadratique de la forme $y_t = at^2 + b$.

1. la fonction `lm()` écrire une ligne de code permettant de réaliser l'ajustement quadratique de séries par rapport au temps t . On appellera ce nouveau modèle `model3` (faire usage de la notation t^2 pour symboliser t^2).

2. Donner une estimation des paramètres a et b

3. Visualiser sur un même graphique la série initiale et l'ajustement de sa composante tendancielle via le modèle quadratique. Ajouter le titre et les libelles de axes au graphique.

4. En vous inspirant de la question B4, étudier les résidus associés à cette seconde modélisation (`residus3=série-tendance3`). Ajouter le titre et les libelles de axes au graphique.



E - Sélection du meilleur modèle

1. D'après la phase précédente, on note que trois modélisations ont été envisagées pour ajuster la composante tendancielle de la série. Il convient dès lors de choisir sur la base d'un critère adéquat le meilleur modèle. Par rapport au critère visuel, quel modèle semble le meilleur ?

2. L'un des critères souvent utilisé est celui de la somme des résidus carrés. Pour chacun des modèles, déterminer cette somme des résidus carrés (faire usage de la fonction `sum()`).

3. Comparer les valeurs et dire quel modèle est le meilleur selon ce critère.

2. Exercice (Moyenne mobile et décomposition)

Dans cet exercice, nous travaillerons avec le jeu de données *elecequip*. Il s'agit de données de demande de production européenne d'équipements électriques (ordinateurs, produits électroniques, produits optiques) – source Eurostat.

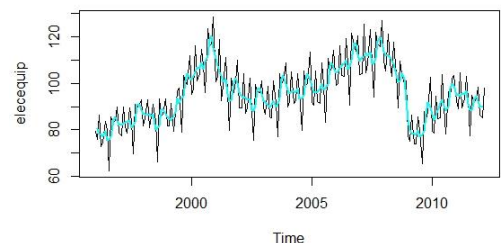
1. Décrire les données (sujet, type de données, nombre d'entrées, pas de temps), les tracer (fonction `plot.ts()`)

2. Lissage par moyenne mobile

Une première méthode de lissage, dite non-paramétrique, est l'utilisation d'une moyenne mobile (MA=Moving Average) dont le but est d'approcher la tendance en atténuant les fluctuations irrégulières tout en éliminant la composante saisonnière.

21. Utilisez la fonction `ma()` pour calculer la moyenne mobile centrée d'ordre $k=3$ ou 5 ou 7 afin d'approcher la tendance et gommer la saisonnalité de ces données mensuelles.

22. Visualisez le résultat grâce à un graphique. Jouez avec l'ordre de la moyenne mobile pour en observer l'effet. Pourquoi les données aux extrémités sont égales à des NAs ?



3. Décomposition via un modèle additif

31. Calculer la décomposition du signal *elecequip* selon un modèle additif. Tracer le graphique qui résume la décomposition par ce modèle additif

32. Calculer la décomposition du signal *elecequip* selon un modèle multiplicatif. Tracer le graphique qui résume la décomposition par ce modèle multiplicatif.

33. Quelles composantes des décompositions varient selon un modèle ou un autre ?

34. Est-ce que les composantes de tendance et saisonnière expliquent bien le signal ?

3. Exercice (Décomposition manuelle de séries : modèle additif)

La production de bière est un bon exemple de série chronologique additive. Lorsque les valeurs métriques augmentent, la saisonnalité reste relativement constante.

1. Charger les données et les tracer.

```
install.packages("fpp")
library(fpp)
install.packages("forecast")
library(forecast)
data(ausbeer)
timeserie_beer = tail(head(ausbeer, 17*4+2), 17*4-4)
plot(as.ts(timeserie_beer))
```

2. Détection de la tendance.

La production australienne de bière suit clairement la saisonnalité annuelle. Comme elle est enregistrée trimestriellement, il y a 4 points de données enregistrés par an, et nous utilisons une fenêtre moyenne mobile de 4. Utiliser la fonction *ma()*. Tracer la série et la tendance sur un même graphique (fonctions *plot()* et *lines()*).

3. Calcul de la série sans tendance.

En supprimant la tendance précédemment calculée de la série chronologique, on obtient une nouvelle série chronologique qui expose clairement la saisonnalité. En faisant la différence de la série originelle moins la tendance, calculer la série sans tendance, puis la tracer.

4. Calcul de la saisonnalité

A partir des séries chronologiques sans tendances, nous calculons la saisonnalité moyenne. Nous additionnons les variations saisonnières et les divisons par la période de variation saisonnière. Techniquement parlant, pour faire la moyenne des séries chronologiques, nous les introduisons dans une matrice. Ensuite, nous transformons la matrice pour que chaque colonne contienne des éléments de la même période (même jour, même mois, même trimestre, etc...). Enfin, nous calculons la moyenne de chaque colonne.

Ici, nous avons une saisonnalité trimestrielle : nous utilisons une matrice de 4 lignes. La saisonnalité moyenne est répétée 16 fois pour créer le graphique à comparer plus tard.

```
m_beer = t(matrix(data = detrend_beer, nrow = 4))
seasonal_beer = colMeans(m_beer, na.rm = T)
plot(as.ts(rep(seasonal_beer, 16)))
```

5. Calcul du bruit aléatoire restant

La formule additive est la suivante :

"Série chronologique = Saisonnier + Tendance + Aléatoire" \Leftrightarrow "Aléatoire = Série chronologique - Saisonnier - Tendance". Calculer l'erreur et la tracer.

6. Utilisation de la fonction *decompose()*

Pour faciliter la vie, certains paquets R fournissent une décomposition avec une seule ligne de code. Comme prévu, notre décomposition étape par étape donne les mêmes résultats que la méthode *decompose()*. La seule exigence : la saisonnalité est trimestrielle (fréquence = 4). Utiliser la fonction *decompose()* pour retrouver les résultats précédents

```
ts_beer = ts(timeserie_beer, frequency = 4)
decompose_beer = decompose(ts_beer, "additive")
```

Tracer les graphiques de la tendance, de la saisonnalité et du bruit séparément, puis dans une même fenêtre

4. Exercice (Décomposition manuelle de séries : modèle multiplicatif)

Les chiffres mensuels sur les passagers des compagnies aériennes sont un bon exemple de série chronologique multiplicative. Plus il y a de passagers, plus on observe de saisonnalité.

1. Charger les données et les tracer.

```
install.packages("fpp")
library(fpp)
install.packages("forecast")
library(forecast)
data(AirPassengers)
timeserie_air = AirPassengers
plot(as.ts(timeserie_air))
```

2. Détection de la tendance

La saisonnalité du nombre de passagers des compagnies aériennes semble annuelle. Toutefois, elle est enregistrée mensuellement, de sorte que nous choisissons une fenêtre moyenne mobile centrée de 12. Utiliser la fonction *ma()*. Tracer la série et la tendance sur un même graphique (fonctions *plot.ts()* et *lines()*).

3. Calcul de la série sans tendance.

En supprimant la tendance précédemment calculée de la série chronologique, on obtient une nouvelle série chronologique qui expose clairement la saisonnalité. En faisant la division de la série originelle par la tendance, calculer la série sans tendance, puis la tracer.

4. Calcul de la saisonnalité

A partir des séries chronologiques sans tendances, nous calculons la saisonnalité moyenne. Nous additionnons les variations saisonnières et les divisons par la période de variation saisonnière. Techniquement parlant, pour faire la moyenne des séries chronologiques, nous les introduisons dans une matrice. Ensuite, nous transformons la matrice pour que chaque colonne contienne des éléments de la même période (même jour, même mois, même trimestre, etc...). Enfin, nous calculons la moyenne de chaque colonne.

Ici, nous avons une saisonnalité mensuelle : nous utilisons une matrice de 12 lignes. La saisonnalité moyenne est répétée 12 fois pour créer le graphique que nous comparerons plus tard

```
m_air = t(matrix(data = detrend_air, nrow = 12))
seasonal_air = colMeans(m_air, na.rm = T)
plot(as.ts(rep(seasonal_air, 12)))
```

5. Calculer du bruit aléatoire restant

La formule multiplicative est :

"Série chronologique = Saisonnier * Tendance * Aléatoire" \Leftrightarrow "Aléatoire = Série chronologique / (Tendance * Saisonnier)". Calculer l'erreur et la tracer.

6. Utilisation de la fonction *decompose()*

Pour faciliter la vie, certains paquets R fournissent une décomposition avec une seule ligne de code. Comme prévu, notre décomposition étape par étape donne les mêmes résultats que la méthode *decompose()*. La seule exigence : la saisonnalité est mensuelle (fréquence = 12). Utiliser la fonction *decompose()* pour retrouver les résultats précédents

```
ts_air = ts(timeserie_air, frequency = 12)
decompose_air = decompose(ts_air, "multiplicative")
```

Tracer les graphiques de la tendance, de la saisonnalité et du bruit séparément, puis dans une même fenêtre

TP04- LISSAGE EXPONENTIEL

Objectif : lissage exponentiel simple, double ou de type Holt-Winters

AIDE SOUS R

```
#Découper une série temporelle de c(i,j) à c(k,l) :
xd<-window(x,c(i,j),c(k,l))

#un lissage exponentiel simple :
xlisce <- HoltWinters(x,alpha=α,beta=FALSE,gamma=FALSE),
# un lissage exponentiel double paramètre 1-α0 :
xlisce <- HoltWinters(x,alpha=α, beta=β, gamma=FALSE) #avec α =1-(α0)^2,
β=(1-α0)/(1+α0)
#un lissage de Holt-Winters sans composante saisonnière :
xlisce <- HoltWinters(x,alpha=α,beta=β,gamma=FALSE),
#un lissage Holt-Winters additif :
xlisce <- HoltWinters(x,alpha=α,beta=β,gamma=γ,seasonal="add"),
# un lissage Holt-Winters multiplicatif :
xlisce <- HoltWinters(x,alpha=α,beta=β,gamma=γ,seasonal="mul")
#HoltWinters ne sauve pas les residues, mais on peut les recuper avec
res= xlisce - xlisce $fitted"
#Remarque : lorsqu'aucune valeur n'est précisée pour les constantes de lissage, un
#algorithme interne à la procédure HoltWinters se charge d'estimer la meilleur
constante #possible à partir de #la série des observations

#Les prévisions à l'horizon h sont réalisées à l'aide de la fonction predict :
p<-predict(xlisce,n.ahead=h)
#on peut aussi utiliser library(forecast)
library(forecast)#install.packages("forecast",repos="http://R-Forge.R-project.org")
rs2 <- forecast.HoltWinters(rsH, h=14)
plot.forecast(rs2)

# Tracer la prédiction et la série sur le même graphique :
xlisce=HoltWinters(x,...);
p=predict(xlisce,n.ahead=50); plot(xlisce,p)

par(mfrow=c(n,p)) #afficher n*p graphique sur la même page
```

1. Exercice (CAC40)

On considère le CAC40. Le but est d'essayer de prédire par lissage exponentiel les valeurs de clôture de 1998 en utilisant les valeurs de clôture de 1991 à 1997

1. Charger les données de CAC40, les plotter, les décrire (de quoi s'agit-il ? début ? fin ? fréquence ?)

```
library(readr)
library(datasets)
#EuStockMarkets <- read_csv("E:/Google Drive/Projects/IUT/MesCours/M2102-Ajustement
#de courbes et séries/TP10h/DataForTP/EuStockMarkets.csv")
View(EuStockMarkets)
CAC=EuStockMarkets[,3]
```

2. Découper le jeu de données en deux séries : la série y : du début jusqu'à fin 1997 et la série x les données depuis 1998. Utiliser la fonction `window()`

3. Lissage exponentiel simple

31. Faire un lissage exponentiel simple via la fonction `HoltWinters()` avec les options `beta=FALSE` et `gamma=FALSE`.

Donner le coefficient α donné par l'algorithme et la somme finale des erreurs au carré obtenues lors de l'optimisation.

Plotter la série en noire, le lissage exponentiel en rouge.

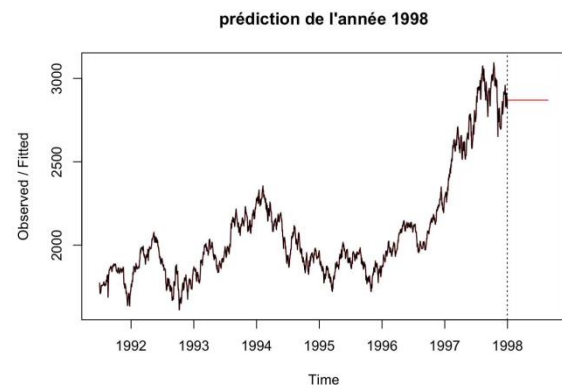
32. Faire une prédiction pour l'année 1998 avec la fonction `predict()` et l'option `n.ahead=longueur de 1998`

33. Plotter la série en noire, le lissage exponentiel en rouge, et la prédiction en bleu.

34. Calculer l'erreur de prédiction.

Plotter l'erreur de prédiction pour les 10 premières valeurs à partir de 1998

Calculer l'erreur quadratique moyenne entre le prédiction et la réalité observée en 1998



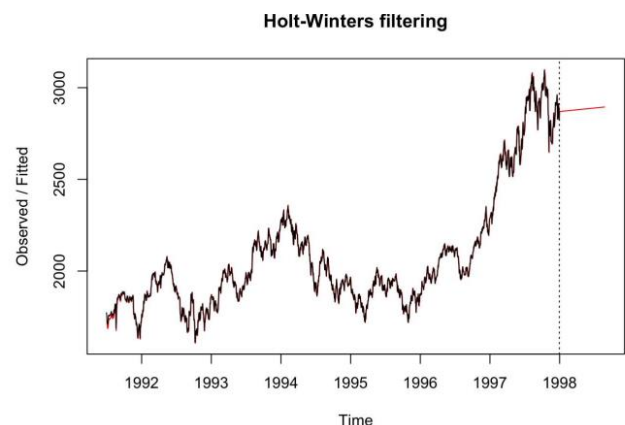
4. Lissage double

41. Faire un lissage exponentiel simple via la fonction `HoltWinters()` avec les options `beta=NULL` et `gamma=FALSE`.

Plotter la série en noire, le lissage exponentiel en rouge.

Donner les coefficient α et β donnés par l'algorithme, la somme finale des erreurs au carré obtenues lors de l'optimisation.

42. Faire une prédiction pour l'année 1998 avec la fonction `predict()` et l'option `n.ahead=longueur de 1998`



43. Ploter la série en noire, le lissage exponentiel en rouge, et la prédiction en bleu.

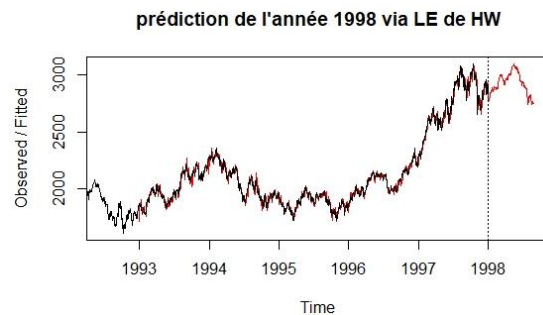
5. Lissage de HoltWinter

51. Faire un lissage exponentiel simple via la fonction `HoltWinters()` avec les options `beta=NULL` et `gamma=NULL`.

Donner les coefficient α et β donnés par l'algorithme, la somme finale des erreurs au carré obtenues lors de l'optimisation.

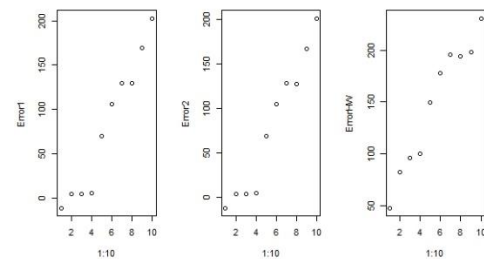
52. Faire une prédiction pour l'année 1998 avec la fonction `predict()` et l'option `n.ahead=longueur de 1998`

53. Ploter la série en noire, le lissage exponentiel en rouge, et la prédiction en bleu.



6. Quelle est la différence entre les trois prédictions : une parait mieux adaptée, une autre constate et l'autre seulement linéaire.

En comparant les graphes des erreurs ploté aux question 33 et 43 que pouvez-vous dire de plus ?



2. Exercice (Pluie à Londres)

1. Charger les données de `precip1.dat`, les stocker dans une variable `rain`, puis dans une série temporelle appelée `rst`, les plotter, les décrire (de quoi s'agit-il ? début ? fin ?)

```
#rain <- scan("E:/Google Drive/Projects/IUT/MesCours/M2102-Ajustement de courbes et
#séries/TP10h/DataForTP/precip1.dat", skip=1)
rain <- scan("http://robjhyndman.com/tsdldata/hurst/precip1.dat", skip=1)
```

2. Faire un lissage exponentiel simple

3. Tracer le résultat de ce lissage

4. Donner le coefficient α donné par l'algorithme, la somme finale des erreurs au carré obtenues lors de l'optimisation.

5. Sauvegarder les résidus (c.à.d. la série d'origine moins les résidus), les plotter.

6. Charger la librairie `forecast`. Etablir des prévisions aux horizons 15 et 30 en utilisant la fonction `forecast()`

3. Exercice (Concentration en co2 à Hawaï)

Le fichier de données `co2` contenu dans R contient les concentrations en CO2 à proximité du volcan Mauna Loa (Hawaï) de 1959 à 1997.

1. Décrire et représenter graphiquement ces données

2. Quel modèle de lissage exponentiel simple vous semble le mieux approprié ?
3. Afin de valider ce modèle, tester la prédiction des données de 1990 à 1997 en utilisant celles de 1959 à 1989.
31. Découper le jeu de donner Data1=données de 1959 à 1989 et DataVal=données de 1990 à 1997
32. Etablir un modèle LES basé sur les données de 1959 à 1989, et sa prévision associée des années 1990 à 1997
33. Calculer l'erreur quadratique moyenne (RMSE) entre les données prédites et DataVal
4. Si le modèle prévisionnel établi vous semble correct vous semble graphiquement correct, utilisez cette méthode pour prédire les concentrations en CO2 de 1997 à 2007.
5. Tester d'autres méthodes de lissage exponentiel double et Holt-Winter