

Résumé

L'objectif de ce projet est la réalisation d'un logiciel présentant une boîte à outil permettant une étude interactive de séries chronologiques. Ce logiciel devra permettre la lecture des données, l'application de divers traitements (moyennes mobiles, désaisonnalisation, estimation de la tendance, différentiation,...), le calcul et l'affichage de prévision ainsi que l'utilisation de méthodes de mesure.

1 Brève description du logiciel

Le logiciel désiré sera complètement interactif. Il doit permettre les fonctionnalités suivantes.

- Permettre à l'utilisateur de charger des données à étudier : soit depuis un fichier, soit depuis un serveur http.
- Maintenir une liste de séries actives. À tout moment, l'utilisateur peut choisir d'afficher la(les) courbe(s) chronologique(s) d'une ou plusieurs des séries actives. L'abscisse donne le temps, l'ordonnée est affichée selon une échelle calculée dynamiquement.
- Appliquer un traitement à une des séries actives. Un traitement peut être :
 - une **transformation** de la série, par laquelle on obtient une nouvelle série. Une transformation peut demander un ou plusieurs paramètres de type numérique ou chronologique.
 - une **analyse** qui effectue une mesure sur une série (par exemple le calcul des autocorrélations). Cette **analyse** ne produit pas une nouvelle série mais donne à l'utilisateur la ou les valeurs calculées
 - une **prévision** à partir de la série actuelle jusqu'à une date spécifiée.

Le résultat d'une transformation ou d'une prévision est une nouvelle série qui sera ajoutée à la liste des séries actives.

- L'utilisateur peut à tout moment supprimer des séries actives de la liste.
- L'utilisateur peut à tout moment choisir la période affichée, soit en choisissant une date de début et une date de fin, soit en choisissant une date de début et une durée.
- L'utilisateur peut à tout moment exporter les données d'une série vers un fichier CSV.
- Éventuellement, l'utilisateur peut demander de visualiser les données d'une série sous forme tabulaire.

2 Zoom sur le fonctionnalités

2.1 Charger des données

Le logiciel permettra à minima de lire des données d'un fichier csv de deux colonnes (colonnes date et valeur). En option, le logiciel permettra de lire des données depuis une adresse http.

2.2 Afficher des courbes

L'affichage de toutes les courbes se fera dans le même panel à l'aide de la bibliothèque JavaFX Charts. Lorsque les courbes de plusieurs séries sont superposées, elles utilisent différentes couleurs et/ou types de ligne. Une légende permet de savoir quelle courbe correspond à quelle série.

2.3 Fonctionnalités liées à la liste de séries actives

La liste est constituée de séries chargées depuis un fichier, et de série résultat de transformations ou prévisions.

Chaque série a un nom, et si c'est une série résultat de transformation, on peut savoir quelle est la transformation qui l'a produite (série source, paramètres de la transformation).

À tout moment on voit dans la liste quelles sont les séries actuellement affichées, et on peut choisir d'en afficher de nouvelles, ou de cacher des séries actuellement affichées.

On peut également supprimer des séries de la liste des séries actives.

L'utilisateur peut demander que les données d'une des séries actives soient sauvegardées dans un fichier CSV.

Fonctionnalité optionnelle : l'utilisateur peut demander de visualiser les données d'une série sous forme tabulaire.

2.4 Traitements sur les séries

Voici la liste des traitements des séries chronologiques qu'il vous est demandé d'implémenter.

2.4.1 Les transformations

Les transformations suivantes ont pour but de stabiliser la variance de vos données :

1. la transformation logarithme $Y_t = \log(X_t)$,
2. la transformation de Box-Cox :

$$Y_t = g_\lambda(X_t) = \begin{cases} \frac{X_t^\lambda - 1}{\lambda} & \text{si } \lambda > 0 \\ \log(X_t) & \text{si } \lambda = 0 \end{cases}$$

Il faut donner le choix du λ à l'utilisateur.

3. la transformation logistique pour les données qui varient dans l'intervalle $]0, 1[$

$$Y_t = \log\left(\frac{X_t}{(1 - X_t)}\right).$$

Il faudra tester la validité de cette transformation avant son application.

D'autres transformations permettent d'estimer la tendance et la saisonnalité :

1. Lissage à l'aide d'une moyenne mobile simple d'ordre h . L'utilisateur choisit l'ordre. Si h est impair, alors

$$M_h(t) = X_t^* = \frac{1}{h}(X_{t-(h-1)/2} + \dots + X_t + \dots + X_{t+(h-1)/2})$$

et si h est pair alors

$$M_h(t) = Y_t^* = \frac{1}{h}(X_{t-h/2/2} + X_{t+1-h/2} + \dots + X_{t+1+h/2} + X_{t+h/2/2})$$

2. Lissage à l'aide d'une moyenne mobile pondérée d'ordre h . Dans ce cas, l'utilisateur doit fournir la longueur h et les pondérations w_1, \dots, w_h (prévoir la possibilité d'utiliser des pondérations par défaut).
3. L'estimation de la saisonnalité à partir de la série $X_t - X_t^*$, puis le calcul de la série X_t^{**} désaisonnalisée.
4. L'estimation d'une tendance linéaire à l'aide d'une droite de régression linéaire de la forme

$$X_t = at + b + \epsilon_t.$$

Dans ce cas l'utilisateur doit pouvoir choisir l'intervalle de temps sur lequel la droite est ajustée.

5. L'application d'un opérateur de différentiation B_h où h l'ordre de la différentiation. Dans ce cas, l'utilisateur doit rentrer l'ordre de la différentiation.

2.4.2 Les analyses

Les analyses portent essentiellement sur la notion de qualité d'ajustement et sur l'analyse des résidus. Si Y_t est la série de départ (éventuellement transformée), et qu'elle a été décomposée sous la forme $Y_t = m + g(t) + S(t) + \epsilon_t$, où $g(t)$ représente la tendance et $S(t)$ la saisonnalité, alors la qualité de la modélisation est donnée par l'analyse des résidus.

Les quantités d'intérêts sont

1. le graphe des résidus pour déterminer les valeurs exceptionnelles, l'existence d'un mouvement saisonnier, etc...
2. La variance résiduelle qui indique la qualité de la modélisation,
3. les autocorrélations des résidus $Cov(\epsilon_t, \epsilon_{t+h})$ qui permettent de tester l'existence d'une corrélation temporelle du bruit.

2.4.3 Les prévisions

Les méthodes de prévisions sont des méthodes empiriques qui extrapolent l'avenir en accordant plus ou moins d'importance aux valeurs passées. Pour toutes les méthodes, l'utilisateur doit définir l'horizon H auquel il veut obtenir des prévisions. Les principales méthodes de prévisions sont

1. Le lissage exponentiel simple, l'utilisateur choisit le coefficient de lissage β .
2. Le lissage exponentiel double. ici aussi, l'utilisateur choisit le coefficient de lissage.
3. La méthode de Holt-Winters avec ou sans saisonnalité.

3 Architecture du programme

L'application devra être construite suivant le pattern de conception MVC (c'est à dire avec une séparation claire des préoccupations liées à l'affichage de celles liées au traitement de l'information (algorithmes, persistance).

D'autres patrons de conceptions (*Design Patterns*) pourraient vous être utiles dans ce projet (ex : *Adapter*, *Bridge*, *Decorator*, *Facade*, *Strategy*). Vous consulterez les ressources disponibles sur internet pour comprendre lesquels et comment les appliquer.

4 Données de test

Vous trouverez de nombreux exemples de séries chronologiques sur internet, en particulier sur le site de l'Insee.

Vous pouvez aussi vous intéresser aux données financières. Quelques liens utiles pour télécharger des données boursières sont les suivants :

- les données fournies par Yahoo (<http://fr.finance.yahoo.com/>),
- un amateur qui fournit des feuilles excel <http://www.gummy-stuff.org/download-stock-prices.htm>,
- ABC bourse propose deux ans d'historique <http://www.abcbourse.com/download/historiques.aspx?f=ex>.

Ce type de séries chronologiques est toutefois plus difficiles à traiter, en particulier, elles n'exhibent pas de saisonnalité en général.

5 Exigences et livrables

5.1 Évaluation

Le projet contribuera à trois modules : COA (UE31), Modélisation (UE32), et Projet (UE33). Chacune des notes sera basée sur des aspects différents du projet. Attention, même si certaines évaluations n'auront lieu qu'à la fin du projet, il faut les prendre en considération dès le début car elles peuvent porter sur l'historique de votre code. De toutes façons, vous n'aurez pas le temps matériel de corriger tout votre code à la fin (livrable 3). Prenez donc le temps de lire en détail tout l'énoncé dès maintenant et de vous faire une petite liste récapitulative des choses importantes. Eventuellement désignez un Responsable Qualité qui vérifiera périodiquement et souvent que les consignes sont bien respectées.

Conception Objet Avancée. La note prendra en compte :

MVC : Application structurée selon les principes du MVC (voir le document "Est-ce que mon projet est MVC ?" sur Moodle) ;

Tests : existence de tests, de qualité, et pertinants. Par exemple :

- tests couvrant toutes les parties importantes du programme, notamment le modèle du MVC ;
- tests développés en accompagnement du code et non pas la dernière semaine pour satisfaire aux exigences d'évaluation.

Patrons de Conceptions : Utilisation de patrons de conception autres que le MVC ; documentation de cette utilisation (indiquez clairement dans la documentation accompagnant le rendu quel(s) patron(s) a(ont) été utilisé(s) et où).

Orienté Objet : Conception/programmation correcte selon les principes de l'orienté-objet. Par exemple :

- respect des conventions d'encapsulation (attributs "cachés", utilisation de getters/setters, respect de la loi de Déméter) ;
- utilisation réfléchie (donc parcimonieuse) de `static` ;
- respect des conventions de nommage (noms significatifs, majuscule/minuscule) ;
- + les recommandations habituelles sur le code bien commenté, bien structuré, bien indenté, ...

Modélisation. La note prendra en compte le nombre de traitements et transformations implémentées sur les courbes, et la correction de celles-ci.

Projet. C'est la note avec le plus grand poids. Elle prendra en compte :

- le respect de la spécification : le logiciel demandé correspond à ce qui est demandé ;
- le nombre de fonctionnalités différentes implémentées, parmi celles demandées ;
- les aspects liés au travail en équipe : partage des tâches, participation de tous les membres de l'équipe ;
- le respect des contraintes de calendrier et de forme du rendu (telles qu'elles seront données sur Moodle).

Concernant le travail en équipe, l'évaluation sera basée sur l'activité de chaque étudiant sur GitLab. C'est pourquoi vous devez vous assurer que chacun *committe* son travail. Aucune excuse liées au mauvais fonctionnement de git ne sera acceptée, c'est de votre responsabilité de mettre en place un bon environnement de travail sur tous les postes que vous utilisez.

5.2 Délivrables

Il vous est demandé de produire trois versions du logiciel. La première implémente seulement certaines des fonctionnalités principale. La dernière est la version complète telle que décrite ci-dessus.

Des détails seront donnés sur Moodle concernant la forme de rendu de chaque livrable, et la date limite.

5.2.1 Livrable 1

- pouvoir charger une série depuis un fichier CSV ;
- pouvoir afficher la courbe d'une série dans un panel graphique. Il n'est pas demandé de faire une fenêtre d'application complète, mais une fenêtre très simple qui contient uniquement la courbe ;
- pouvoir calculer (et afficher) le résultat d'une des transformation listées en 2.4.1,
- pouvoir calculer (et afficher) une moyenne mobile d'ordre h ,
- pouvoir calculer (et afficher) les résidus entre la série de départ et la série lissée par une moyenne mobile, c'est à dire la série $\epsilon_t = X_t - M_h(t)$.

Concrètement, le programme

- demande à l'utilisateur le fichier qui contient les données en entrée, le type de transformation ou analyse (à choisir dans une liste), et les paramètres de la transformation ou analyse sélectionnée précédemment,
- si une transformation a été demandée, donne le choix entre afficher le résultat ou le sauvegarder dans un fichier,
- calcule et affiche la série résultat ou le résultat numérique (si une analyse a été demandée).

5.2.2 Livrable 2

Vous devez implémenter l'interface du logiciel avec ses différents éléments : espace d'affichage des courbes, espace d'affichage de la liste des courbes actives, espace (ou menu) de choix des traitements à appliquer.

L'interface est fonctionnelle et permet les interactions décrites dans les fonctionnalités du logiciel.

Il est possible d'afficher plusieurs courbes au même temps.

En plus des traitements demandés au Livrable 1, vous devez permettre les traitements suivants :

En plus des traitements demandés au Livrable 1, vous devez permettre les traitements suivants :

- désaisonnalisation,
- ajustement de tendance par régression linéaire,
- affichage du graphe (série) des résidus.

5.2.3 Livrable 3

Vous devez ajouter la possibilité d'ajouter dynamiquement de nouveaux traitements par un système de plugins.

De plus, vous devez implémenter les traitements liés aux prévisions.

Finalement, vous ajouterez :

- la possibilité de faire un zoom sur les courbes affichées en spécifiant une période ,
- (optionnel) le calcul des auto-corrélations des résidus,
- (optionnel) de charger des données depuis un serveur http.

Vous pouvez également ajouter d'autres fonctionnalités non listées ici : ou bien des traitements vus en cours de Modélisation et non listé ici, ou bien des traitements qui vous semblent intéressants, ou bien des améliorations de l'interface graphique. Cela apporterait un bonus sur votre note.