

Rapport du TP3

IFT3913

Yalin Mo (20199655)
Vennila Sooben (20235256)

1 Introduction

Dans cette section, nous présentons l'étude des métriques logicielles de JFreeChart qui est une bibliothèque de graphiques pour Java créée par Dave Gilbert il y a 23 ans. L'objectif de ce travail pratique est d'analyser les métriques TLOC (Nombre de lignes de code non-vides qui ne sont pas de commentaires), WMC (Weighted Methods per Class), et TASSERT (nombre d'assertions) des classes de l'application. Cette analyse vise à évaluer la qualité du logiciel et à comprendre les corrélations entre ces métriques. Nous procéderons avec en premier lieu une visualisation à l'aide de diagrammes à moustaches. Puis, nous enchaînerons avec une étude de corrélation avec des calculs de coefficients et des diagrammes. Nous finirons avec une analyse des données obtenues pour ainsi proposer une conclusion réfléchie de notre cas d'études.

2 Tâche 1 : Visualisation des Métriques avec Excel

2.1 Création des Diagrammes à Moustaches

Pour visualiser les métriques, nous avons créé des boîtes à moustaches à l'aide d'Excel. Les diagrammes montrent les valeurs suivantes :

- TLOC :
 - Median : 83
 - Limite Supérieur : 125
 - Limite Inférieure : 47

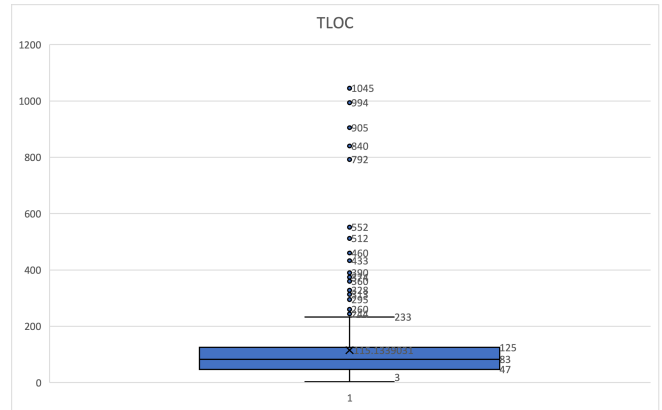


FIGURE 1 – Boîte à moustaches TLOC

- WMC
 - Median : 9
 - Limite Supérieur : 12
 - Limite Inférieure : 8

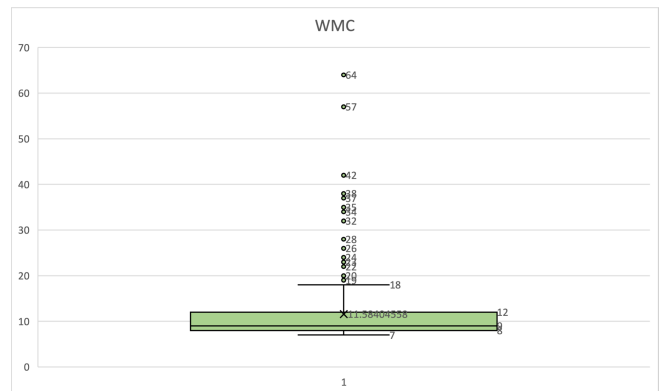


FIGURE 2 – Boîte à moustaches WMC

- TASSERT
 - Median : 17
 - Limite Supérieur : 21
 - Limite Inférieure : 10

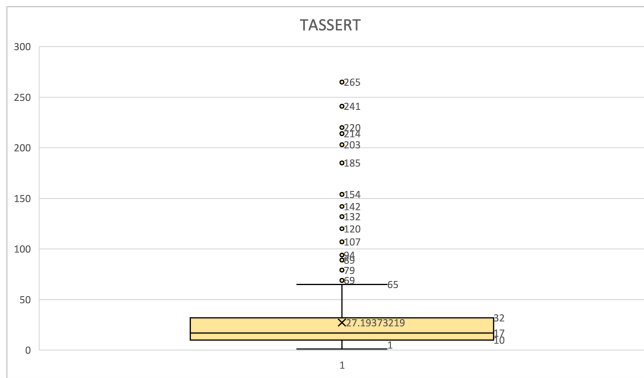


FIGURE 3 – Boîte à moustaches WMC

2.2 Analyse des Résultats

À partir de nos données et graphiques obtenus, nous pouvons savoir

TLOC : Sa médiane est de 83, indiquant la tendance centrale de l'ensemble de données ; La longueur de la boîte est de 78. On peut savoir que le nombre de lignes de code dans les 50 % du milieu du fichier est réparti entre 47-125. La limite supérieure est de 242, ce qui signifie que si le nombre de lignes de code dans un fichier dépasse cette valeur, il peut être considéré comme une valeur aberrante. Sa limite inférieure est -70, ce qui est un nombre négatif. Il n'a aucune signification de référence car le nombre de lignes de code ne peut pas être négatif.

WMC : La médiane est de 9 et la longueur de la case est de 4, ce qui indique que le nombre de méthodes dans la plupart des classes est relativement concentré et que les changements ne sont pas évidents. la limite supérieure est 18. Si la valeur d'un fichier dépasse 18, cela indique que certaines classes ont un nombre inhabituellement élevé de méthodes. La limite inférieure est de 2, ce qui signifie que peu de classes ont très peu ou pas de méthodes.

TASSERT : La médiane est de 17. Reflète le nombre d'assertions couramment utilisées dans les tests de fichiers La longueur de la boîte est de 22, ce qui montre que le nombre d'assertions est relativement réparti dans l'ensemble de données. La limite supérieure est de 65. Si elle dépasse cette valeur, cela signifie que certains cas de tests utilisent beaucoup de fautes. La limite inférieure est de -23, ce qui est la même que dans le cas TLOC. C'est également une valeur de référence dénuée de sens car le nombre d'assertions ne peut pas être négative.

Conclusion, Complexité du code : la méthode de distribution large de TLOC signifie que le nombre de lignes de code dans les fichiers de test varie considérablement, certains ont plus de lignes et d'autres moins. Conception des classes : la longueur courte de la boîte de WMC indique que la complexité de la plupart des classes est relativement cohérente et ne fluctue pas beaucoup. Cas de test : la plus grande longueur de la boîte de TASSERT indique que l'utilisation des assertions varie considérablement entre les différents cas de test.

3 Tâche 2 : Étude des Corrélations avec Python

3.1 Vérification de la Normalité des Données

Nous avons utilisé le test de Shapiro-Wilk en Python pour vérifier la normalité des données. Les résultats montrent que toutes les métriques (TLOC, WMC, TASSERT) ne suivent pas une distribution normale avec une p-valeur inférieure (ici $<<<$) à 0,05. Nous avons utilisé les bibliothèques pandas et shapiro (scipy.stats) pour trouver nos données.

Voici les valeurs obtenues.

(Note : ND signifie normally distributed) :

```
Shapiro-Wilk test for TLOC:
Test Statistic: 0.5937689542770386
P-Value: 7.221374956319535e-28
Sample does not look ND (reject H0)

Shapiro-Wilk test for WMC:
Test Statistic: 0.577251136302948
P-Value: 2.556520093449692e-28
Sample does not look ND (reject H0)

Shapiro-Wilk test for TASSERT:
Test Statistic: 0.5818790197372437
P-Value: 3.408619149135902e-28
Sample does not look ND (reject H0)
```

FIGURE 4 – Test de normalité Shapiro-Wilkin.

3.2 Calcul des Coefficients de Corrélation

Les coefficients de corrélation entre TLOC, WMC et TASSERT sont les suivants :

- **Corrélation entre TLOC et TASSERT :**
Spearman Rank Correlation Coefficient : 0.834606519587501
P-Value : 2.1276357013448774e-92
- **Corrélation entre WMC et TASSERT :**
Spearman Rank Correlation Coefficient : 0.6149330438016306
P-Value : 6.893133521020008e-38

3.3 Régression

Afin d'effectuer la régression, nous avons utilisé un script Python avec les bibliothèques pandas, numpy, matplotlib.pyplot afin de dessiner des 'best fit line' (en rouge) - c'est à dire les lignes représentant nos équations - et compare avec nos données (en bleu). Voici un visuel des 2 droites ainsi que l'équation de nos droites de régression linéaire.

- **Equation TLOC(x) et TASSERT(y) :**
$$y = 0.251790600546902 * x + -1.6764188502983188$$

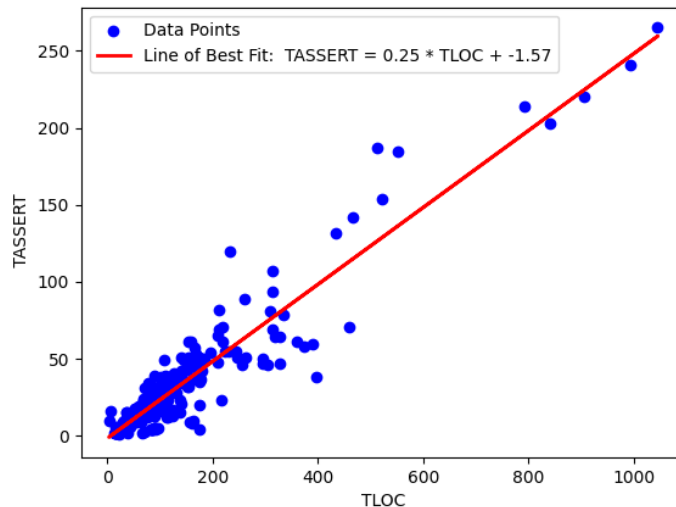


FIGURE 5 – Régression TLOC-TASSERT

- **Equation WMC(x) et TASSERT(y) :**

$$y = 4.404718600470233 * x + -21.699130611226817$$

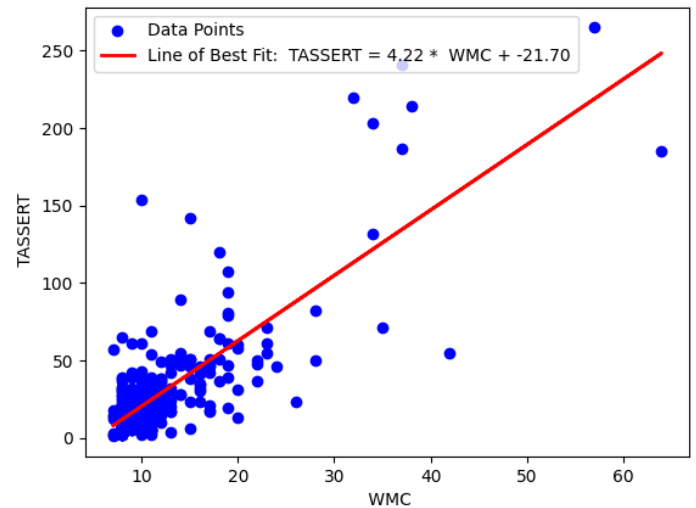


FIGURE 6 – Régression WMC-TASSERT

3.4 Analyse de la Tâche 2

Dans le cadre de la Tâche 2, notre objectif était d'étudier les corrélations entre les métriques TLOC, WMC et TASSERT. Nous avons suivi les étapes suivantes :

Vérification de la Normalité des Données

Avant d'entreprendre une analyse de corrélation, nous avons effectué un test de normalité (Shapiro-Wilk) sur nos données pour déterminer si elles suivaient une distribution normale. Les résultats ont montré que *les données ne sont pas normalement distribuées*.

Cela indique que nos données ne suivent pas une distribution gaussienne, ce qui est un prérequis pour certains tests statistiques paramétriques. Par conséquent, nous avons opté pour des méthodes de corrélation non paramétriques.

Calcul des Coefficients de Corrélation

Nous avons utilisé le coefficient de corrélation de Spearman, un test non paramétrique approprié pour les données non normalement distribuées, afin d'évaluer les relations entre les métriques.

Les résultats des corrélations sont les sui-

vants :

- **TLOC et TASSERT** : Une corrélation positive significative a été observée, avec un coefficient de Spearman de 0.83. Cela suggère une forte corrélation entre le nombre de lignes de code et le nombre d'assertions dans les classes.
- **WMC et TASSERT** : Une corrélation positive moins forte, mais toujours significative, a été trouvée, avec un coefficient de Spearman de 0.61. Cela suggère une corrélation moins forte entre le nombre de méthodes pondérées par classe et le nombre d'assertions.

Régression

Nous avons également réalisé des analyses de régression pour modéliser les relations entre les métriques. Les équations des droites de régression ont été obtenues, montrant comment les variables évoluent les unes par rapport aux autres.

- **TLOC et TASSERT** : L'équation de régression indique qu'il existe une relation linéaire positive entre le nombre de lignes de code et le nombre d'assertions, avec une pente de 0.25.
- **WMC et TASSERT** : L'équation de régression montre une relation linéaire positive moins forte entre le nombre de méthodes pondérées par classe et le nombre d'assertions, avec une pente de 4.40.

Interprétation

Ces résultats suggèrent que les classes avec un nombre élevé de lignes de code ont tendance à avoir un nombre plus élevé d'assertions. De plus, bien que la corrélation entre le nombre de méthodes pondérées par classe et le nombre d'assertions soit positive, elle est moins forte que la corrélation observée avec le nombre de lignes de code.

4 Tâche 3 : Évaluation de l'Hypothèse

4.1 Conception de l'Expérience

Nous choisissons une approche hypothético-déductive pour tester l'association entre le nombre d'assertions et la complexité des classes.

4.2 Énoncé des Hypothèses

Hypothèse nulle (H0) : Aucune augmentation de complexité pour les classes avec plus de 20 assertions par rapport à celles avec moins.

Hypothèse alternative (H1) : Les classes avec plus de 20 assertions sont plus complexes.

4.3 Définition des Variables

Variable Indépendante : Nombre d'assertions (plus grand ou égale 20 ou plus petit que 20).

Variables Dépendantes : Mesures de complexité TLOC et WMC

4.4 Présentation des Résultats

Les données représentées dans les boîtes à moustaches pour TLOC, WMC et TASSERT montrent une distribution avec des écarts importants par rapport à la normale, indiquées par la présence de plusieurs valeurs extrêmes, ce qui est également étayé par le test de Shapiro-Wilk. Et la médiane pour TASSERT est relativement élevée. Les corrélations de Spearman confirment une association positive entre TASSERT et les mesures de complexité.

5 Conclusion

La présence de fortes corrélations et les distributions observées dans les boîtes à moustaches suggèrent que les classes avec un nombre plus élevé d'assertions tendent à avoir des mesures de complexité plus élevées, ce qui appuie l'hypothèse alternative H1.

6 Discussion des Menaces à la Validité

Les corrélations ne prouvent pas la causalité et les boîtes à moustaches révèlent la variabilité des données. D'autres facteurs peuvent influencer la complexité des classes.