IFT 3913: Rapport TP3

Pour le 18 novembre 2022 à 23:59

Professeur: Michalis Famelis

Zi Kai Qin, 20191254 **Maxime Ton, 20143044**

1 Introduction :

Dans le cadre de ce travail pratique, nous allons étudier un échantillon de données portant sur la librarie JFreeChart composée des trois métriques suivantes :

— NoCom : nombre de commits

— NCLOC : nombre de lignes de code qui ne sont pas ni vides ni commentaires

— DCP : densité de commentaires (CLOC/LOC) donnée en pourcentage

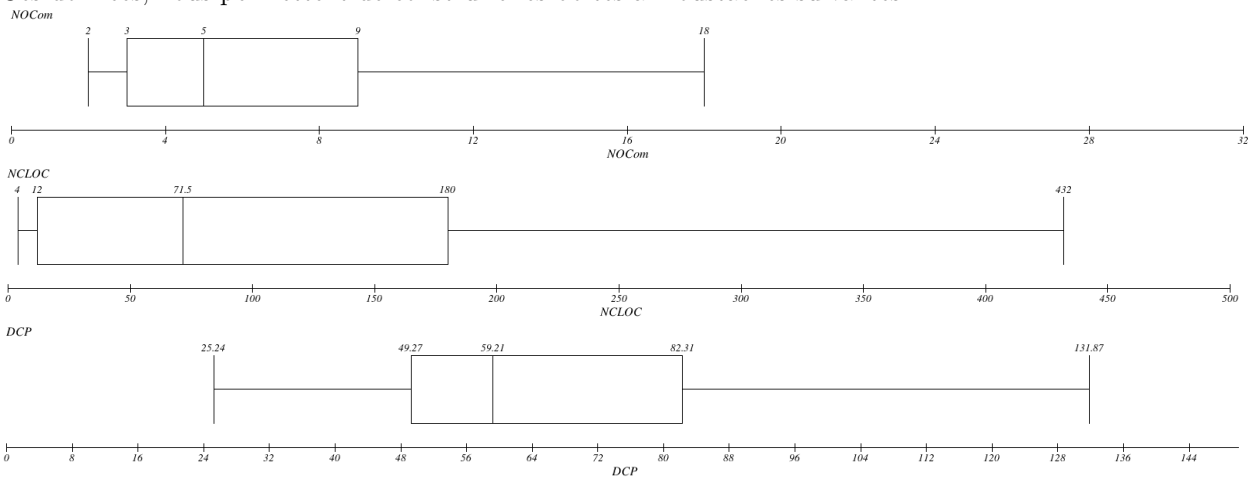
Nous nous servirons des résultats de cette analyse afin d'évaluer la validité d'une certaine hypothèse donnée.

2 Partie 1 – Boîtes à moustache :

Nous commençons par visualiser notre échantillon à l'aide de boîtes à moustaches. Ceci requiert quelques calculs, que nous effectuons à l'aide d'Excel [1], selon la technique d'analyse vue aux diapositives 7. Ces calculs nous donnent les valeurs suivantes :

	NoCom	NCLOC	DCP
m	5	71.5	59.21
u	9	180	82.31
l	3	12	49.27
d	6	168	33.04
s	18	432	131.87
Min	2	4	25.24
i	2(-6)	4(-240)	25.24(-0.29)
Max	32	2732	93.44

Ces données, nous permettent de construire les boîtes à moustaches suivantes [2] :



NoCom : La distribution est biaisée vers le bas avec un quartile supérieur deux fois plus long que le quartile inférieur. Plusieurs points extrêmes existent au-delà de l'intervalle théorique, particulièrement un point extrême maximum de 32.

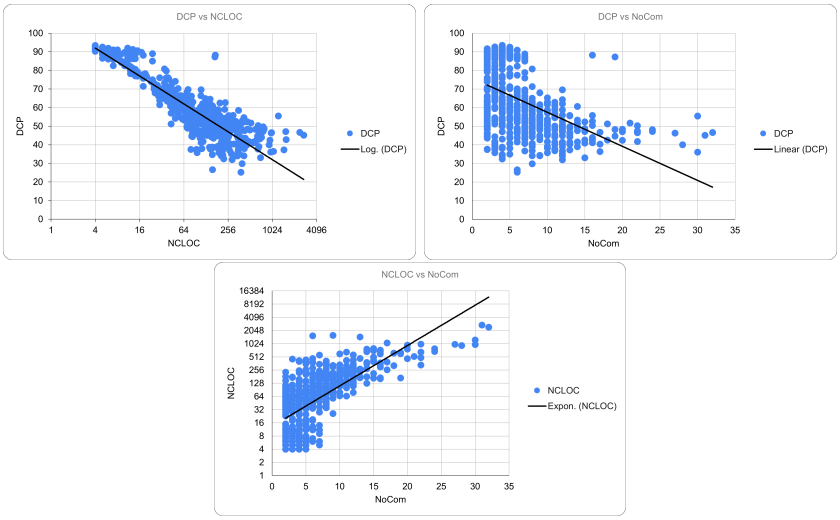
NCLOC : Tout comme NoCom, la distribution est biaisée vers le bas avec un quartile supérieur plus long que le quartile inférieur. La dispersion est cependant beaucoup plus élevée. Il y a plusieurs points extrêmes au-delà de la limite supérieure de 432, notamment un maximum de 2732.

DCP : Encore une fois, le quartile supérieur est beaucoup plus long que le quartile inférieur. Cependant, l'intervalle théorique est plus centré et les valeurs sont moins dispersées. Il n'y a pas points extrêmes.

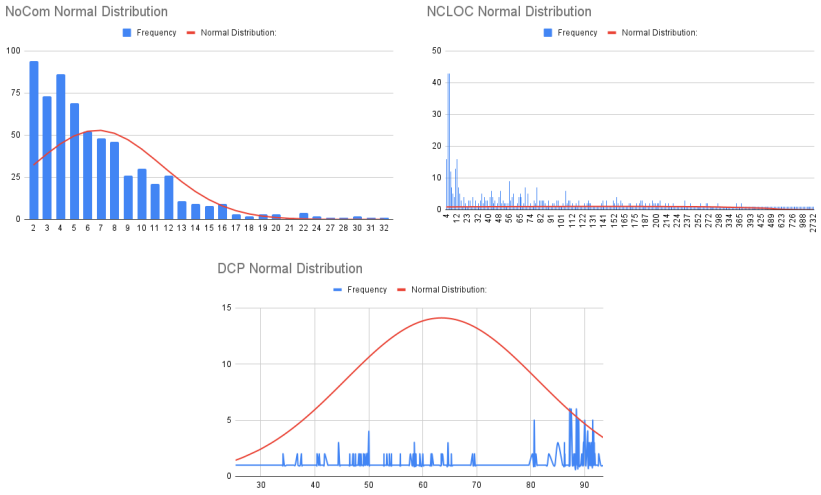
1. Voir jfreechart-stats.xlsx
2. Boîtes à moustaches créées sur le site https ://www.imathas.com/stattools/boxplot.html

3 Partie 2 – Études de corrélations :

Nous cherchons maintenant à étudier la corrélation entre les différentes métriques. L'outil *Trendline* de Google Sheets nous permet de générer les droites de régression :



Il nous faut maintenant déterminer si les variables sont normalement distribuées ou non. À l'aide de graphiques composés avec Google Sheets, nous comparons la fréquence des données à leur courbe normale :



Il clair que ces variables ne sont fort probablement pas normalement distribuées. Le coefficient de corrélation de Pearson (r) ne peut être utilisé, car il exige une distribution normale. Nous nous servirons donc du coefficient de corrélation de rang de Spearman (ρ), que nous calculons simplement en remplaçant nos valeurs par leur rang (RANK.AVG) et en calculant ensuite r sur ceux-ci en utilisant la fonction CORREL [3] :

NoCom vs NCLOC : $\rho = 0.69$ NoCom vs DCP : $\rho = -0.53$ NCLOC vs DCP : $\rho = -0.90$

Nous voyons donc que NoCom et NCLOC ont une bonne corrélation positive, que NoCom et DCP ont une corrélation négative modérée et que NCLOC et DCP ont une forte corrélation négative.

3. https ://support.google.com/docs/answer/3093990 ?hl=en

4 Partie 3 – Hypothèse :

Il nous faut finalement vérifier l'hypothèse suivante :

"*Les classes qui ont été modifiées plus de 10 fois sont mieux commentées que celles qui ont été modifiées moins de 10 fois.*"

4.1 Choix d'étude

Nous voulons vérifier l'existence d'une relation hypothétique entre le NoCom d'une classe et son DCP. Pour ce faire, nous allons mesurer le NoCom et DCP des classes d'une base de code non-triviale et nous allons comparer le DCP des classes dont le NoCom > 10 à celui des classes dont le NoCom ≤ 10. Le type d'étude que nous allons poursuivre est une quasi-expérience, puisqu'il est impossible de former un groupe de contrôle.

4.2 Énoncé des hypothèses

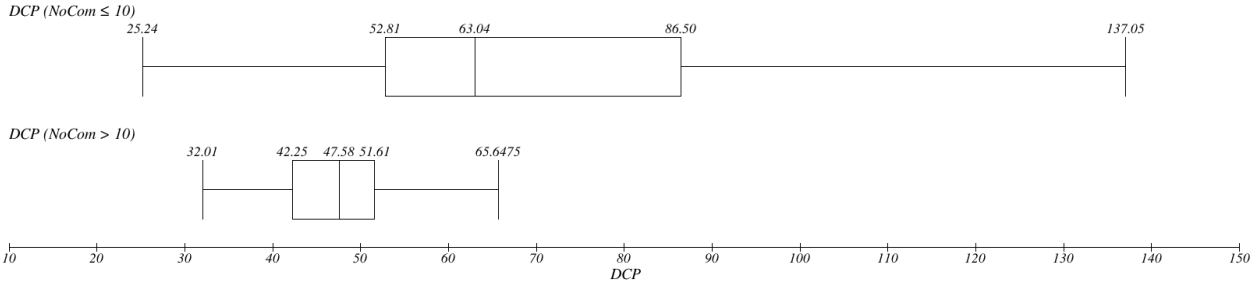
L'hypothèse que nous proposons est que dans une base de code, les classes dont le NoCom > 10 ont un DCP moyen significativement supérieur à celui des classes dont le NoCom ≤ 10.

4.3 Définition des variables

Nous cherchons à mesurer l'effet du NoCom sur le DCP. La variable indépendante est donc le NoCom des classes, et la variable dépendante est le DCP des classes.

4.4 Expérience et résultats

L'échantillon donné dans le fichier jfreechart-stats.csv contient déjà les valeurs de NoCom et DCP des classes d'une base de code de taille significative. Nous nous en sommes déjà servi pour calculer la corrélation entre le NoCom et le DCP à la partie 2. Nous effectuons des calculs supplémentaires dans Excel[1] afin de générer les boîtes à moustaches suivantes :



Celle du haut représente la distribution des valeurs de DCP des classes dont le NoCom est en-deçà de 10, et celle du bas représente la distribution du DCP où le NoCom dépasse 10.

4.5 Interprétation et généralisation des résultats

Malgré la faiblesse de la corrélation entre le DCP et le NoCom, sa droite de régression ainsi que son coefficient de Spearman (ρ) nous permet immédiatement de voir que le DCP tend plutôt vers le bas lorsque NoCom augmente, contrairement à l'hypothèse proposée.

De plus, à part pour les 4 plus grande valeurs, la totalité des valeurs de DCP où NoCom > 10 se trouve en-deçà de la médiane de l'ensemble des valeurs de DCP où NoCom ≤ 10. Ceci montre assez clairement que les classes qui ont été modifiées plus que 10 fois ne sont pas significativement mieux commentées que les

classes qui ont été modifiées moins que 10 fois. L'hypothèse ne peut donc être que fausse.

À partir de la droite de régression, nous observons plutôt la tendance suivante : au fur et à mesure que NoCom augmente, DCP semble tendre vers 50%. En effet, alors que le DCP moyen des classes dont le NoCom ≤ 10 est de 66.6%, celui des classes dont le NoCom > 10 est de 48.2%. Ceci peut s'expliquer par le fait que certains répositoires imposent ou encouragent un DCP minimum sur l'ensemble de leur code. Si le DCP minimum de la base de code de JFreeChart est de 50%, ceci expliquerait cette tendance.

4.6 Discusssion des menaces à la validité

Certains facteurs posent une menace à la validité de l'expérience. Un tel facteur est que l'échantillon est sélectionné parmi une seule base de code écrit en un seul langage de programmation. Ceci diminue considérablement la généralité des résultats, pour une multitude de raisons.

Nous savons, par exemple, que les langages de haut niveau permettent au programmeur d'encoder plus instructions par ligne que les langages de bas niveau. Alors, pour des fonctions équivalentes, un langage de bas niveau nécessite plus de lignes de code qu'un langage de haut niveau. Ainsi, même si les lignes de commentaires reste le même, le DCP du langage de bas niveau sera plus bas que celui du langage de haut niveau. Il est cependant clair que le code écrit dans le langage de haut niveau n'est tout de même pas mieux commenté que celui écrit dans un langage de bas niveau.