

OBJECTIF

Dans ce TP, vous allez créer un prototype d'analyse automatisée, basé sur le logiciel que vous avez créé pour TP1. Puis vous allez l'utiliser pour étudier la corrélation entre deux métriques.

TÂCHE 1 (25%)

Créez le programme **proto**. Il prend comme entrée le URL d'un dépôt Git et produit comme sortie un fichier CSV (« *comma separated values* », valeurs séparées par des virgules) avec les colonnes : `id_version`, `NC`.

- **id_version** : identificateur hexadécimal d'une version. Votre programme peut faire appel à la commande `git rev-list master` pour trouver l'ensemble des identificateurs des versions dans l'historique de la dernière version du branchement principal.
- **NC** : nombre de classes (pour nos besoins : nombre de fichiers *.java) dans une version. Votre programme peut faire appel à la commande `git reset --hard <HEX>` pour amener le dépôt git local à la version avec l'identificateur <HEX>.

TÂCHE 2 (25%)

Ajoutez fonctionnalité à **proto** pour ajouter à sa sortie les colonnes `mWMC`, `mcBC`:

- **mWMC** : moyenne de la métrique WMC («weighed methods per class»).
- **mcBC** : moyenne de la métrique `classe_BC`, définie dans TP1.

Vous êtes encouragés à réutiliser votre implémentation du TP1 pour calculer les valeurs WMC et `classe_BC` pour les classes de chaque version. Idéalement, **proto** devrait faire appel à votre programme du TP1 comme commande externe. Ici, nous n'évaluerons pas la correction de votre implémentation du TP1. Mais votre code de TP1 doit au minimum réussir à générer les fichiers avec les données requises. Au pire cas, utilisez un *mock* pour une pénalité de 15%.

TÂCHE 3 (20%)

Appliquez **proto** à l'historique du code de <https://github.com/jfree/jfreechart>. Dès le 15 février 2022, jfreechart comptait 4209 commit. Concentrez sur un échantillon aléatoire de 5% du total. (Bonus 5% si vous analysez l'historique entière du projet.) Soumettez dans votre ZIP le fichier CSV généré par **proto**.

TÂCHE 4 (10%)

Montrez visuellement l'évolution des métriques `NC`, `mWMC`, et `mcBC` à travers l'histoire de jfreechart et essayez d'expliquer de façon qualitative ce que vous observez.

TÂCHE 5 (20%)

Évaluez l'hypothèse selon laquelle pour les versions du jfreechart il existe une **corrélation** entre les métriques **NC** et **mWMC** et entre les métriques **NC** et **mcBC**. Basez votre analyse sur l'hypothèse que leurs valeurs **ne sont pas** normalement distribuées.

CONDITIONS DE RÉALISATION

- Travaillez en groupes de 2. Soumettez un zip nommé comme suit : *prenom1_nom1_prenom2_nom2.zip*
- Travail à remettre le **vendredi 22 avril 23h59** via StudiUM.
- Le membre de l'équipe qui ne soumet pas le fichier Zip principal de la soumission doit soumettre un fichier avec le même nom, contenant les noms de l'équipe, comme dans les autres TP.
- Vous pouvez utiliser votre IDE et langage préféré (python, groovy, bash, java, ...). Vous êtes encouragés à utiliser les outils standards de UNIX (ls, shuf, echo, grep, wc, awk, ...) et vous pouvez mélanger des langages (par exemple : un script bash qui fait appel à un script R pour des calculs numériques). Cependant, nous n'offrons pas de conseils sur l'utilisation des IDE ou sur le scripting.
- Décrivez brièvement votre travail dans votre rapport.
- Expliquez toute hypothèse et décrivez comment exécuter *proto*.
- Mettez le code/script du *proto*, le fichier CSV et votre rapport en format PDF dans le Zip de votre soumission.
- Vous pouvez utiliser n'importe quel logiciel qui appuie les opérations statistiques requises (Excel, R, Octave, SPSS, Python, Java, ...). Cependant, nous n'offrons pas de conseils sur l'utilisation des logiciels statistiques. **Ajoutez votre code/script/fichiers dans le Zip de votre soumission.**
- Rapports de plus de 10 pages (12p, simple espace) vont être éliminés d'office.
- Assurez-vous de communiquer clairement, la qualité de votre rapport est **très importante**. Un mauvais rapport pourrait causer une déduction significative.