

- **LINF0 1103**
 - BA1, +200 étudiants
 - Une section par TP
 - Ouverture par section plutôt que par tâche
 - Utilisation « classique »
 - Pour chaque question d'implémentation, un template est mis à disposition des étudiants avec 2-3 tests
 - Ces tests doivent passer avant d'avoir plus de feedback
 - Examen
 - Modèle comme pour les projets de LGBIO 2010
- **LGBIO 2010**
 - MA(1), ~50 étudiants
 - 4 sections
 - Introduction
 - Ouvert en permanence
 - Projet 1
 - Projet 2
 - Projet 3



- Question de **validation**
 - Feedback complet, y compris la réponse attendue
 - Objectif : s'assurer que l'exercice soit bien compris sur une petite instance
 - Aussi parfois sous-problème des questions évaluées dans la suite de la tâche
 - Idéalement l'exercice est résolu à la main
 - Connaître le nombre de soumissions déjà effectuées pour donner la réponse après 2-3 essais ?
- Question **évaluée**
 - Tests/feedback **avant deadline** :
 - Syntaxe, type de retour, ...
 - Objectif : s'assurer que l'étudiant soit évalué sur son travail et pas sur sa compréhension de la tâche à effectuer
 - Tests/feedback **après la deadline** :
 - Si tests pré-deadline réussis : évaluation des attentes
- *Toujours mettre à disposition les paramètres et les résultats générés et attendus (au moins pour une instance fausse)*

- Par tâche
 - Une ou deux questions de **validation**
 - Quelques questions **évaluées**
 - Toujours des questions directement liées
 - Si implémentation, l'étudiant devrait pouvoir vérifier grâce aux inputs et outputs (connus et valides) de la/des questions de validation
 - Parfois questions en tiroirs
- Juste avant la deadline, l'étudiant devrait avoir 100%
- Dans certains cas, les tâches sont directement liées
 - Essayer d'évaluer les fonctions plutôt que les données produites

- Pour chaque tâche, config.json
 - **Validation:** T/F
 - **Grade:** T/F
 - **Weight:** 1.0
 - Relatif
 - **Prerequis:** []
 - Liste des questions qui doivent être réussies pour que la questions soit vérifiée
 - **Checks:** []
 - Liste des tests avant deadline
 - **Tests:** []
 - Liste des tests post deadline (ceux qui donnent des points)
 - **Options:** []
 - Si nécessaires (par exemple si il faut renvoyer une valeur dans un certain interval
- **Soumission évaluée = dernière**
 - + rerun post deadline

There are some errors in your answer. Your score is 96.67%. [Submission #6375fc1580607efdfdbcacfd]

Evaluated questions: 2/3 succeeded.

Your grade is 96.67%.

Question 1: Relative frequencies

✖ Your answer does not match the expected results.

Consider inspecting your answer:

- Your code does not respect the specifications.

Your R object does not match the expected object:

- Number of non-matching values: 109/5929

The expected file can be downloaded [here](#) for comparisons purposes. The file is an RDS file and should be opened with the `readRDS()` function. Observe that ordering of rows and columns can be different between your submission and the expected result, but while grading, this has been taken into account. In other words, you haven't been penalized if your result is following a different order (regarding columns and rows) from the expected one.

Grade: 0.9/1.0

- Réf. relative pour les images/scripts/src...

1 On s'intéresse à l'algorithme **Surprise** décrit dans le pseudo-code ci-dessous.

2

3 .. image:: https://inginius.info.ucl.ac.be/course/LINF01103/S02_1_1_surprise/Surprise.png

4 :height: 5cm

5 :align: center

6

- Archive des cours passés

LINFO1103-0620	Hidden course - Administrator
LINFO1103-0621	Hidden course - Administrator
LINFO1103-0621PEPS	Hidden course - Administrator
LINFO1103-0622	Hidden course - Administrator
LINFO1103-0820	Hidden course - Administrator
LINFO1103-0821	Hidden course - Administrator
LINFO1103-0822	Hidden course - Administrator

- Javascript pour randomizer des données
- Production d'instances spécifiques par groupe
 - Récupérer dans le conteneur l'info du groupe (numéro)