

Grille d'évaluation d'un programme					
Structure du programme	Excellent 5	Très bien 4	Bien 2	Passable 1	Insuffisant 0
<p>Division des scripts en algorithmes et fonctions</p> <p>Le code est généralisé Les fonctions sont autoportantes (elles dépendent que des arguments) Le processus est contrôlé par un script principal (peut être un script R ou Rmd)</p>		<p>Les opérations sont regroupées en fonction et chaque script contient une seule fonction ou un seul algorithme. Un script décrit le processus et fait les appels aux fonctions qui s'occupent de la lecture à la préparation des données. Les fonctions ne dépendent que d'arguments.</p>	<p>Les opérations sont regroupées en fonction et chaque script contient parfois plusieurs fonctions ou algorithmes. Un script décrit le processus et fait les appels aux fonctions qui s'occupent de la lecture à la préparation des données ou les fonctions contiennent des valeurs qui ne sont pas passées comme arguments.</p>	<p>Les opérations sont parfois regroupées en fonction. Les scripts décrivent les étapes de la lecture à la préparation des données. Les fonctions contiennent des valeurs qui ne sont pas passées comme arguments.</p>	<p>Les opérations sont dans un seul script ou dans plusieurs sans organisation évidente. Les opérations ne sont pas organisées sous forme de fonctions. Absence d'un script principal pour exécuter les tâches.</p>
Fonctionnement du programme	5	4	3	2	1
<p>Le programme s'exécute sans intervention de l'utilisateur à partir du fichier principal</p> <p>Un dataframe par table de la base de données est retourné</p> <p>Toutes les données sont assemblées dans ces dataframes</p> <p>Le format des données est approprié (dates, taxonomie, etc)</p> <p>Le type de données (booléen, numérique, etc) pour chaque colonne est juste</p> <p>Une série de validation est exécutée pour valider l'intégrité des données</p> <p>Les erreurs sont détectées et corrigées automatiquement par l'action de fonctions</p>		<p>Le programme s'exécute de manière entièrement automatisée à partir du fichier principal, sans besoin d'intervention de l'utilisateur.</p> <p>Chaque dataframe retourné correspond précisément à une table de la base de données, avec toutes les données correctement assemblées.</p> <p>Le format des données est approprié pour toutes les colonnes, suivant les meilleures pratiques.</p> <p>Les types de données pour chaque colonne sont corrects dans tous les cas, assurant une interprétation précise et cohérente des données.</p> <p>Une série exhaustive de validation est exécutée à chaque étape pour garantir l'intégrité des données.</p> <p>Toutes les erreurs sont automatiquement détectées et corrigées par l'action de fonctions, garantissant la qualité et la fiabilité des résultats.</p>	<p>Le programme s'exécute sans intervention de l'utilisateur à partir du fichier principal de manière fluide.</p> <p>Chaque dataframe retourné correspond à une table spécifique de la base de données, avec toutes les données correctement assemblées.</p> <p>Le format des données est uniforme pour la plupart, suivant les meilleures pratiques.</p> <p>Les types de données pour chaque colonne sont corrects dans la plupart des cas, assurant une interprétation précise des données.</p> <p>Une série de validation est exécutée pour garantir l'intégrité des données à chaque étape.</p> <p>La plupart des erreurs sont détectées et corrigées automatiquement par l'action de fonctions, assurant la fiabilité des résultats.</p>	<p>Le programme peut s'exécuter sans intervention de l'utilisateur à partir du fichier principal, bien que cela ne soit pas toujours fluide.</p> <p>Chaque dataframe retourné correspond généralement à une table de la base de données, mais quelques-uns peuvent contenir des données de plusieurs tables.</p> <p>La plupart des données sont assemblées dans les dataframes, bien que des lacunes subsistent.</p> <p>Le format des données est uniforme pour la plupart des colonnes, mais certaines diffèrent des meilleures pratiques.</p> <p>Les types de données pour chaque colonne sont corrects dans la plupart des cas, mais des erreurs persistent.</p> <p>Certaines vérifications sont effectuées pour valider l'intégrité des données, mais elles ne sont pas exhaustives.</p> <p>Quelques erreurs sont détectées et corrigées automatiquement par l'action de fonctions, mais pas toutes.</p>	<p>Le programme nécessite une intervention de l'utilisateur pour s'exécuter à partir du fichier principal.</p> <p>Certains dataframes retournés contiennent les données de plusieurs tables de la base de données.</p> <p>Les données sont partiellement assemblées dans les dataframes.</p> <p>Le format des données diffère souvent des meilleures pratiques.</p> <p>Les types de données pour chaque colonne sont incorrects dans plusieurs cas.</p> <p>Aucune série de validation n'est exécutée pour garantir l'intégrité des données.</p> <p>Les erreurs ne sont pas automatiquement détectées ni corrigées par l'action de fonctions.</p>
Bonnes pratiques de programmation	5	4	3	2	1

<p>Le code est structuré en étapes et est facile à lire (indentation, une commande par ligne)</p> <p>Le code est commenté Les variables et arguments sont nommés de façon explicite</p> <p>Les opérations qui doivent être répétés sont enchâssés dans des boucles</p>		<p>Le code est structuré en étapes claires et est très facile à lire en raison d'une indentation impeccable et d'une organisation soignée. Chaque commande est correctement espacée sur sa propre ligne.</p> <p>Les commentaires sont abondants et détaillés, fournissant une explication exhaustive du fonctionnement du code.</p> <p>Les variables et les arguments sont nommés de manière très explicite, ce qui rend le code transparent et compréhensible pour toute personne qui le consulte.</p> <p>Les opérations répétées sont systématiquement enchâssées dans des boucles, ce qui réduit au minimum la duplication de code et favorise une approche plus efficace et élégante.</p>	<p>Le code est structuré en étapes claires et est facile à lire grâce à une indentation appropriée et à une organisation cohérente. Chaque commande est sur une ligne distincte.</p> <p>Les commentaires sont présents et fournissent des explications adéquates sur le fonctionnement du code, facilitant sa compréhension.</p> <p>Les variables et les arguments sont nommés de manière explicite, ce qui rend le code plus compréhensible pour les autres programmeurs.</p> <p>Les opérations répétées sont souvent enchâssées dans des boucles, réduisant ainsi la redondance de code et améliorant sa maintenabilité.</p>	<p>Le code est structuré en étapes, mais la lisibilité pourrait être améliorée avec une meilleure indentation et une organisation plus claire. Il y a une commande par ligne.</p> <p>Certains commentaires sont présents, mais ils ne sont pas toujours suffisamment détaillés pour expliquer pleinement le code.</p> <p>Les variables et les arguments sont nommés de manière relativement explicite, mais il y a place à amélioration pour une meilleure compréhension du code.</p> <p>Quelques opérations répétées sont enchâssées dans des boucles, mais cela n'est pas systématique.</p>	<p>Le code manque de structure en étapes et n'est pas facile à lire en raison d'une indentation et d'une organisation insuffisantes. Il peut y avoir plusieurs commandes par ligne.</p> <p>Les commentaires sont absents ou insuffisants pour expliquer le fonctionnement du code.</p> <p>Les variables et les arguments sont mal nommés, ce qui rend difficile la compréhension du code sans une analyse approfondie.</p> <p>Les opérations répétées ne sont pas enchâssées dans des boucles, ce qui entraîne une redondance inutile de code.</p>
--	--	---	--	---	--