Cahier de charge

1. Preview du dataset

	login	name	company	location	total_stars	nb_repos_fetched	languages_list
1	sindresorhus	Sindre Sorhus			457146	5	JavaScript, TypeScript
2	kamranahmedse	Kamran Ahmed	roadmap.sh	United Kingdom	426260	5	JavaScript, Shell, TypeScript
3	torvalds	Linus Torvalds	Linux Foundation	Portland, OR	212950	5	C, OpenSCAD
4	getify	Kyle Simpson	Getify Solutions	Austin, TX	203886	5	JavaScript
5	openai	OpenAI			174874	5	Jupyter Notebook, Python, Rust, TypeScript

On voit les variables suivantes :

- `login` **: identifiant GitHub**
- `name` **: nom de la personne/organisation**
- `company` **: entreprise ou projet associé**
- `location` **: localisation géographique**
- `total_stars` **: nombre total d'étoiles GitHub sur ses dépôts**
- `nb_repos_fetched` **: nombre de dépôts pris en compte dans le calcul**
- `languages_list` **: principales langues de programmation utilisées (liste séparée par des virgules)**

2. Contexte et objectif

- Thème : analyse statistique de développeurs/projets GitHub les plus populaires.

- Objectif général : étudier les caractéristiques des profils GitHub très étoilés et voir comment se répartissent la popularité (`total_stars`) **selon :**
    - **les localisations,**
    - **les entreprises,**
    - **les langages de programmation.**

3. Questions de recherche possibles

- **Comment se distribue le nombre total d'étoiles (`total_stars`) parmi ces profils ?**
- **Les profils d'organisations (ex. OpenAI, roadmap.sh) se distinguent-ils des profils individuels ?**
- **Certaines localisations (pays / régions) concentrent-elles plus de profils très étoilés ?**
- **Les profils qui utilisent certains langages (ex. JavaScript, Python, C, Rust) ont-ils des niveaux de `total_stars` différents ?**
- **Y a-t-il une relation entre `nb_repos_fetched` et `total_stars` (plus de dépôts = plus d'étoiles ?) ?**

4. Données disponibles

- Source : extrait de profils GitHub très populaires (dataset propriétaire/local, pas issu de Kaggle).
- Format : fichier CSV `profiles_index.csv`**.**
- **Observations : ~30 lignes (profils).**

- **Variables :**
    - **Qualitatives nominales :** `login`, `name`, `company`, `location`, `languages_list`.
    - **Quantitatives :** `total_stars`, `nb_repos_fetched`.

5. Méthodologie d'analyse (dans R / Quarto / Colab)

1. Import et préparation

- Importer le CSV dans R (`read.csv`).
- **Vérifier les types des variables.**
- **Nettoyer :**
    - **les valeurs manquantes pour** `company` **et** `location`.
    - **transformer** `languages_list` **en variables analytiques (par exemple, indicateurs binaires par langage : JavaScript, Python, etc.).**
    - **éventuellement regrouper** `location` **par pays/régions (Europe, US, etc.).**

2. Analyse descriptive

- **Statistiques descriptives de** `total_stars` **et** `nb_repos_fetched` **(moyenne, médiane, min, max, etc.).**
- **Fréquences pour** `company`, `location`, **langages principaux.**
- **Visualisations :**
    - **histogramme ou boxplot de** `total_stars`,
    - **barplots pour la présence de certains langages,**
    - **boxplots de** `total_stars` **par localisation ou par présence d'un langage (ex. JavaScript vs pas JavaScript).**

3. **Analyse bivariée / inférentielle (simple, vu la taille)**
    - **Corrélation ou régression simple entre `total_stars` et `nb_repos_fetched`.**
    - **Comparaison de moyennes de `total_stars` entre 2 groupes**
4. **Présentation avec Quarto (ça sera un plus)**

    Création d' un document `.qmd` :

    - introduction (contexte GitHub, objectifs),
    - description des données,
    - résultats (tableaux + graphiques),
    - discussion (interprétation, limites, pistes futures).

6. Livrables

- Colab contenant :
    - import des données,
    - nettoyage,
    - analyses descriptives et graphiques,
    - tests/statistiques simples.
- Rapport Quarto(plus):
    - structuré en sections (Introduction, Méthodes, Résultats, Discussion).