

Challenge : Analyser de la data

Pouvoir d'achat et employabilité

Description des données

Le projet porte sur des données concernant le pouvoir d'achat et l'employabilité des américains en 2012.

Pour ce faire, deux jeux de données :

- all-ages.csv qui correspond aux données d'emploi par domaine d'étude pour tous les âges
- recent-grads.csv qui correspond aux données d'emploi par domaine d'étude pour les diplômé universitaire récent, y compris par sexe et par type d'emploi qu'ils ont obtenu

Description de quelques colonnes

recent-grads.csv :

- Rank : classement des domaines d'étude par rapport aux salaires médian des diplômé.
- Major_code : Code numérique du domaine d'étude
- Major : Description du diplôme.
- Major_category : Catégorie du diplôme.
- Total : nombre total de personne qui on suivie ce diplôme.
- Sample_size : échantillon des étudiant à temps plein.
- Men : nombre d'hommes qui on suivit ce diplôme.
- Women : nombre de femmes qui on suivit ce diplôme.
- ShareWomen : Proportion de femmes qui ont suivit le diplôme.
- Employed : nombre de personnes qui ont obtenue un emploie après leurs diplôme.
- Low_wage_jobs : nombre d'emploi présentant des bas salaires

Introduction au dataset

- Lire all-ages.csv dans un DataFrame et l'assigner à la variable all_ages.
- Lire recent-grads.csv dans un DataFrame et l'assigner à la variable recent_grads.
- Afficher les 5 premières lignes de all_ages et recent_grads.

Nombre d'étudiants par catégorie de Major

- Retourner les valeurs uniques de Major_category.
 - Utiliser la méthode Series.unique() (équivalent du distinct en SQL) pour retourner les valeurs uniques d'une Series.
- Pour chaque valeur unique (utilisation d'une boucle for?):
 - Retourner toutes les lignes où Major_category vaut cette valeur unique.
 - Calculer le nombre total d'étudiants représentant cette catégorie de major (colonne Total à sommer).
 - Vous garderez en mémoire ce résultat sous la forme d'un dictionnaire contenant une Major_category en clé et le nombre d'étudiants en valeur.

Nombre d'étudiants par catégorie de Major

Mission:

- Créer une fonction dans laquelle vous utiliserez la colonne Total pour calculer le nombre d'étudiants pour chaque catégorie de Major (Major_category) dans chaque dataset.
 - Stocker le résultat dans 2 dictionnaires distincts
 - La clé pour chaque dictionnaire sera Major_category et la valeur le total d'étudiants
 - Pour le dataset all_ages, stocker le résultat dans un dictionnaire qu'on nommera aa_car_counts
 - Pour le dataset recent_grads, stocker le résultat dans un dictionnaire qu'on nommera rg_cat_counts
- Autre méthode: Utiliser un pivot de table.
- Afficher les 2 dictionnaires.

Taux de jobs à faible salaire

- Utiliser les colonnes "Low_wage_jobs" et "Total" pour calculer la proportion de jeunes diplômés qui ont du trouver des jobs à faible salaire (recent_grads).
 - Souvenez-vous que vous pouvez utiliser la méthode `Series.sum()` pour retourner la somme des valeurs d'une colonne
- Stocker le résultat dans la variable `low_wage_proportion` et afficher le.

Comparer des datasets

- Utiliser une boucle for pour parcourir toutes les majors.
 - Pour chaque Major et chaque DataFrame, filtrer seulement les lignes du DataFrame correspondant à cette major
 - Comparer les valeurs pour la colonne "Unemployment_rate" pour voir lequel des 2 DataFrames possède la valeur la plus basse
 - Incrémenter (cad ajouter 1) à la variable rg_lower_count si la valeur pour Unemployment_rate est plus petite dans le dataframe recent_grads que dans le dataframe all_ages
- Afficher le résultat rg_lower_count.