

Data Test : BIFORA

ROBERT Wendy

19 août 2025

Table des matières

1	Traitement	2
1.1	Python	2
1.1.1	Chargement des bibliothèques	2
1.1.2	Lecture du fichier Excel	2
1.1.3	Préparation et nettoyage des données	2
1.1.4	Diagnostic initial des données	3
1.1.5	Suppression des doublons et fusion des bases	3
1.1.6	Renommage et sélection des variables pertinentes	3
1.1.7	Conversion des variables en numérique	3
1.2	R	4
1.2.1	Importation des données	4
1.2.2	Correction des types de variables	4
1.2.3	Correction des noms de variables	4
1.2.4	Fusion des deux bases	5
1.2.5	Exploration graphique	5
1.2.6	Export du jeu de données fusionné	5
1.3	SAS	5
1.3.1	Définition de la librairie et importation	5
1.3.2	Renommage des variables	6
1.3.3	Conversion des variables mal typées	6
1.3.4	Fusion des bases de données	7
1.3.5	Contrôle final	7
2	Analyse des données	7
2.1	Graphiques générés	8
2.2	Conclusion	11

Introduction

L'objectif de ce travail est d'analyser et de préparer un jeu de données issu d'un fichier Excel fourni. Ce fichier comporte deux feuilles principales (`Data_1` et `Data_2`) décrivant des informations sur une base de clients, ainsi qu'une troisième feuille (`TYPES VARIABLES`) servant de dictionnaire de données.

1 Traitement

1.1 Python

L'analyse débute par l'importation du fichier Excel `data_test_BIFORA.xlsx` contenant trois feuilles de données : `Data_1`, `Data_2` et `TYPES VARIABLES`. Cette opération est réalisée en Python grâce à la bibliothèque `pandas`, qui permet de lire directement des fichiers Excel.

1.1.1 Chargement des bibliothèques

Nous commençons par importer la bibliothèque `pandas` :

```
import pandas as pd
```

Listing 1 – Importation bibliothèque Pandas

1.1.2 Lecture du fichier Excel

Ensuite, nous spécifions le chemin du fichier et chargeons les trois feuilles de données :

```
# Importation des 3 feuilles
path = r"/Users/wendyr/Downloads/data_test_BIFORA (2) (1).xlsx"
data1 = pd.read_excel(path, sheet_name="Data_1")
data2 = pd.read_excel(path, sheet_name="Data_2")
dict_vars = pd.read_excel(path, sheet_name="TYPES VARIABLES ")
```

Listing 2 – Importation du fichier Excel

Une fois les données importées, nous pouvons examiner leurs dimensions, leurs colonnes et les premières lignes afin de comprendre leur structure :

Cette étape constitue la base de toute l'analyse, car elle permet de vérifier que les données sont bien accessibles et lisibles par Python avant de passer au diagnostic et au nettoyage.

1.1.3 Préparation et nettoyage des données

Certaines variables contiennent des erreurs de frappe ou des espaces parasites dans leur nom. Nous commençons par corriger ces problèmes et nettoyer les colonnes :

```
# Renommer correctement la variable identifiant
data1 = data1.rename(columns={"identiffiant ": "identifiant"})

# Nettoyer les noms des variables (suppression des espaces)
data1.columns = data1.columns.str.strip()
data2.columns = data2.columns.str.strip()
dict_vars["VARIABLES"] = dict_vars["VARIABLES"].str.strip()
```

Listing 3 – Renommage et nettoyage des colonnes

1.1.4 Diagnostic initial des données

Pour mieux comprendre la structure des jeux de données, nous définissons une fonction `diagnostic` qui affiche les types de variables, les valeurs manquantes et les doublons :

```
def diagnostic(df, name):
    print("\n--- Diagnostic "+name+" ---")
    print("Types de variables :\n", df.dtypes)
    print("\nValeurs manquantes :\n", df.isna().sum())
    print("\nDoublons :", df.duplicated().sum())

diagnostic(data1, "Data_1")
diagnostic(data2, "Data_2")
```

Listing 4 – Diagnostic des bases

1.1.5 Suppression des doublons et fusion des bases

Nous supprimons les doublons et fusionnons les deux jeux de données sur la variable `identifiant` :

```
data1 = data1.drop_duplicates()
data2 = data2.drop_duplicates()

merged = data1.merge(data2, on="identifiant", how="inner")

print("\nDimensions Data_1 :", data1.shape)
print("\nDimensions Data_2 :", data2.shape)
print("\nDimensions après fusion :", merged.shape)
```

Listing 5 – Nettoyage et fusion des données

1.1.6 Renommage et sélection des variables pertinentes

Pour harmoniser les noms de colonnes et conserver uniquement les variables nécessaires à l'analyse :

```
rename_map = {
    "AGE": "age",
    "SEXE": "sexe",
    "MARCHE": "marche",
    ...
}

merged = merged.rename(columns=rename_map)

needed = ["age", "marche", "csp", "type_client", "situation_famille", "
    epargne_financiere", "epargne_total"]
merged = merged[needed].copy()
```

Listing 6 – Renommage et sélection des variables

1.1.7 Conversion des variables en numérique

À l'aide du dictionnaire `TYPES VARIABLES`, nous convertissons les variables quantitatives. Les erreurs de conversion sont transformées en `NaN` :

```

for _, row in dict_vars.iterrows():
    var = row["VARIABLES"]
    if var in data1.columns and row["Quali / Quanti "] == "Quanti":
        data1[var] = pd.to_numeric(data1[var], errors="coerce")
    if var in data2.columns and row["Quali / Quanti "] == "Quanti":
        data2[var] = pd.to_numeric(data2[var], errors="coerce")

for col in ["age", "epargne_financiere", "epargne_total"]:
    if col in merged.columns:
        merged[col] = pd.to_numeric(merged[col], errors="coerce")

merged = merged.dropna(subset=["age", "epargne_total", "epargne_financiere"])

```

Listing 7 – Conversion numérique et suppression des valeurs manquantes

1.2 R

Les données ont également été traitées dans l'environnement RStudio. L'importation s'est faite de manière intuitive, l'interface de RStudio simplifiant le chargement manuel des feuilles `Data_1` et `Data_2` issues du fichier Excel.

1.2.1 Importation des données

L'importation des deux feuilles `Data_1` et `Data_2` a été effectuée directement via l'interface graphique de RStudio (menu Import Dataset). Les jeux de données on donc été disponibles dans l'environnement de travail sous les noms `data1` et `data2`.

1.2.2 Correction des types de variables

Certaines variables n'étaient pas reconnues avec le bon format. D'après le dictionnaire `TYPES VARIABLES`, les colonnes `epargne_financiere` et `epargne_totale` de `Data_1` devaient être numériques, alors qu'elles avaient été importées comme chaînes de caractères. De même, la variable `DATE_ERG` de `Data_2` devait être au format date.

Ces conversions ont été réalisées à l'importation du fichier mais peuvent se faire à la main :

```

data1$epargne_financiere <- as.numeric(data1$epargne_financiere)
data1$epargne_totale <- as.numeric(data1$epargne_totale)
data2$DATE_ERG <- as.Date(data2$DATE_ERG, format="%Y-%m-%d")

```

Listing 8 – Conversion des variables

1.2.3 Correction des noms de variables

Une erreur d'orthographe a été détectée dans la feuille `Data_1` : la variable `identifiant` était mal orthographiée `indentiffiant`. Cette variable constitue la clé commune aux deux bases de données. Afin de permettre leur fusion, le renommage a été effectué :

```

names(data_1)[names(data_1) == "indentiffiant"] <- "identifiant"

```

Listing 9 – Correction de l'orthographe d'une variable

1.2.4 Fusion des deux bases

Une fois les corrections effectuées, les deux jeux de données ont pu être fusionnés grâce à la clé commune `identifiant`.

```
data_merge <- merge(data_1, data_2, by = "identifiant", all = TRUE)
```

Listing 10 – Fusion des bases de données

Cette opération a permis de consolider l'information en un seul tableau prêt pour l'analyse.

1.2.5 Exploration graphique

Plusieurs tentatives de visualisation ont été effectuées avec la librairie `tidyverse` (notamment `ggplot2`). Cependant, les résultats obtenus se sont révélés moins pertinents que ceux obtenus en Python, en particulier pour la représentation des distributions et des regroupements par catégories.

Ainsi, si R a permis un contrôle efficace des types et de la structure des données, l'exploration graphique a finalement été poursuivie de manière plus approfondie avec Python.

1.2.6 Export du jeu de données fusionné

Enfin, afin de conserver une trace du nouveau jeu de données consolidé (`merged`), il est possible de l'exporter sous forme de fichier CSV. Cette opération permet de réutiliser la base nettoyée et fusionnée dans d'autres environnements (Python, SAS, Excel, ...) :

```
write.table(merged,
            "/Users/wendyr/Downloads/data_test.csv",
            sep = ",",
            row.names = FALSE)
```

Listing 11 – Exportation du jeu de données fusionné

1.3 SAS

En parallèle de Python et R, les données ont également été importées dans SAS afin de permettre une exploration complémentaire et des traitements statistiques.

1.3.1 Définition de la librairie et importation

Tout d'abord, une librairie pointant vers le fichier Excel a été définie. L'instruction `proc import` a ensuite été utilisée pour lire directement les feuilles `Data_1` et `Data_2` :

```
libname x xlsx '/home/u64300920/data_test_BIFORA (2) (1).xlsx';

/* Import feuille Data_1 */
proc import datafile='/home/u64300920/data_test_BIFORA (2) (1).xlsx'
    out=x.data1
    dbms=xlsx
    replace;
    sheet="Data_1";
    getnames=yes;
run;

/* Import feuille Data_2 */
```

```
proc import datafile='/home/u64300920/data_test_BIFORA (2) (1).xlsx'
    out=x.data2
    dbms=xlsx
    replace;
    sheet="Data_2";
    getnames=yes;
run;
```

Listing 12 – Importation des données dans SAS

1.3.2 Renommage des variables

Lors de l'importation, une erreur de frappe a été identifiée dans la variable clé `identifiant`, indispensable pour la fusion des deux bases. Afin d'assurer la cohérence et de permettre le merge, la variable `identiffiant` a été renommée correctement :

```
proc datasets library=x nolist;
    modify data1;
    rename identiffiant = identifiant;
quit;
```

Listing 13 – Correction d'une faute dans le nom de variable

1.3.3 Conversion des variables mal typées

Comme dans Python et R, certaines variables quantitatives ont été importées sous forme de texte, ce qui empêche les calculs statistiques. Un traitement a donc été appliqué pour convertir correctement ces colonnes (`Epargne_financiere`, `Epargne_totale`) en numérique, et la date `DATE_ERG` en format date SAS :

```
data x.data1_tmp;
    set x.data1;
    /* Conversion en numérique */
    Epargne_financiere_tmp = input(Epargne_financiere, best32.);
    Epargne_totale_tmp     = input(Epargne_totale, best32.);

    /* Conversion en format date */
    format DATE_ERG date9.;
    DATE_ERG = input(DATE_ERG, yymmdd10.);

    /* Nettoyage : suppression des anciennes versions */
    drop Epargne_financiere Epargne_totale;
    rename Epargne_financiere_tmp = Epargne_financiere
           Epargne_totale_tmp     = Epargne_totale;
run;

/* Remplacer l'ancienne table par la version corrigée */
proc datasets library=x nolist;
    delete data1;
    change data1_tmp = data1;
quit;
```

Listing 14 – Conversion des variables en SAS

1.3.4 Fusion des bases de données

Une fois les variables corrigées et renommées, les deux jeux de données peuvent être fusionnés sur la clé commune `identifiant`. Après un tri préalable des deux tables, la fusion est effectuée à l'aide de l'instruction `merge` :

```
proc sort data=x.data1;
    by identifiant;
run;

proc sort data=x.data2;
    by identifiant;
run;

data x.data_merge;
    merge x.data1(in=a) x.data2(in=b);
    by identifiant;
    if a and b; /* garde seulement les identifiants présents dans les deux
        tables */
run;
```

Listing 15 – Fusion des deux tables dans SAS

1.3.5 Contrôle final

Enfin, la structure des tables obtenues a été vérifiée grâce à la procédure `proc contents`, afin de s'assurer de la bonne prise en compte des types de variables et de la fusion :

```
proc contents data=x.data1;
    title "Structure du dataset DATA_1";
run;

proc contents data=x.data2;
    title "Structure du dataset DATA_2";
run;

proc contents data=x.data_merge;
    title "Structure du dataset fusionné";
run;
```

Listing 16 – Inspection des tables SAS

Ainsi, SAS a permis de répliquer les étapes de nettoyage et de transformation effectuées dans Python et R, et de générer un jeu de données consolidé (`data_merge`) utilisable pour des analyses statistiques plus poussées.

2 Analyse des données

Résumé du fichier

Le fichier contient trois feuilles principales :

- **Data_1** : informations individuelles des clients (âge, date de naissance, situation familiale), produits détenus (cartes, crédits, assurances), montants financiers (épargne, encours) et type de carte bancaire.
- **Data_2** : enrichit Data_1 avec des informations socio-démographiques et marketing : sexe, marché (particuliers, professionnels), type de client, secteurs géographiques, catégorie socio-professionnelle.
- **TYPES VARIABLES** : dictionnaire de données indiquant le type de chaque variable (qualitative/quantitative, discrète/continue).

Globalement, le fichier constitue une base clients d'un établissement bancaire ou d'assurance, combinant profil individuel, produits et montants financiers avec des informations socio-démographiques et de segmentation marketing.

2.1 Graphiques générés

Dans cette section, nous présentons les six graphiques réalisés afin d'explorer les liens entre les profils des individus et leurs niveaux d'épargne. Chaque graphique est suivi d'une analyse succincte.

1. Histogramme de l'épargne totale par rapport à l'âge

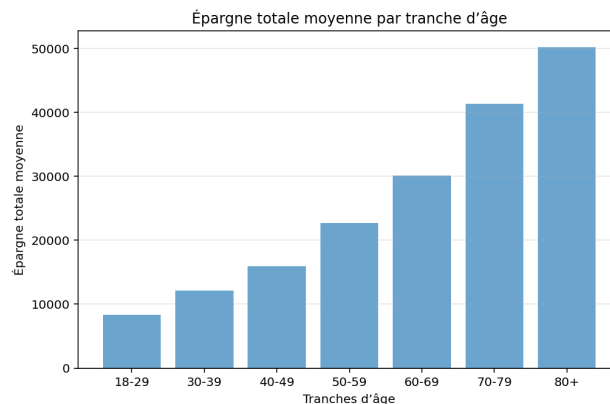


FIGURE 1 – Histogramme de l'épargne totale selon l'âge

On observe une tendance claire : plus l'âge augmente, plus la moyenne d'épargne totale croît. Cela suggère un effet d'accumulation patrimoniale au fil du temps.

2. Histogramme de l'épargne financière par rapport à l'âge

La même relation est visible que pour l'épargne totale : l'épargne financière progresse également avec l'âge, traduisant une capacité d'investissement plus forte chez les individus plus âgés.

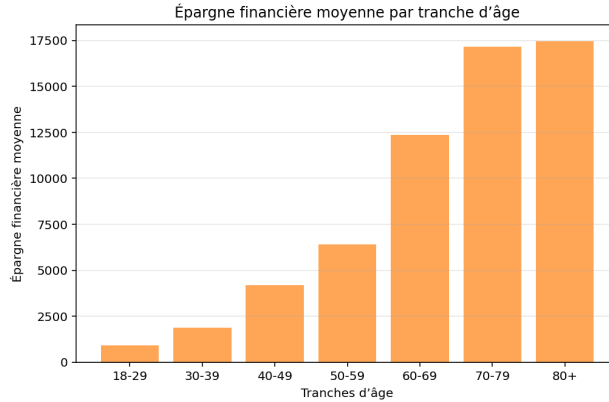


FIGURE 2 – Histogramme de l'épargne financière selon l'âge

3. Répartition de l'épargne totale par CSP

Répartition de l'épargne totale par CSP

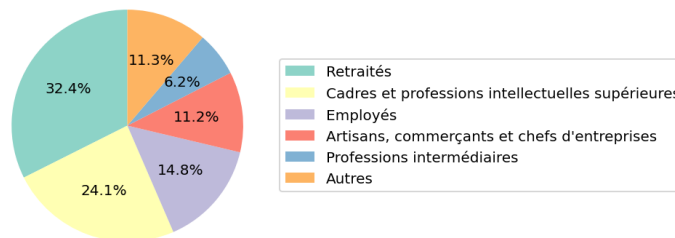


FIGURE 3 – Camembert de la répartition de l'épargne totale par CSP

La répartition montre que : - les **retraités** concentrent la part la plus importante (32,4%), - suivis des **cadres et professions intellectuelles supérieures** (24,1%), - puis des **employés** (14,8%), - et des **artisans, commerçants et chefs d'entreprises** (11,2%).

Les autres catégories représentent des parts plus réduites. Cela confirme que les retraités, bénéficiant souvent de temps d'accumulation et de revenus stables, détiennent l'épargne la plus importante.

4. Répartition de l'épargne financière par CSP

La structure est encore plus marquée que pour l'épargne totale : - **44,2% des retraités** concentrent à eux seuls près de la moitié de l'épargne financière, - suivis des **cadres et professions intellectuelles supérieures** (22,5%).

Les autres catégories (**artisans, employés, sans activité professionnelle**) se partagent le reste. Cette surreprésentation des retraités confirme leur rôle central dans l'épargne, probablement lié à une volonté de sécuriser le patrimoine.

5. Distribution des tranches d'épargne totale par marché

On constate une nette domination des particuliers, qui sont environ **cinq fois plus nombreux** que les professionnels.

Répartition de l'épargne financière par CSP

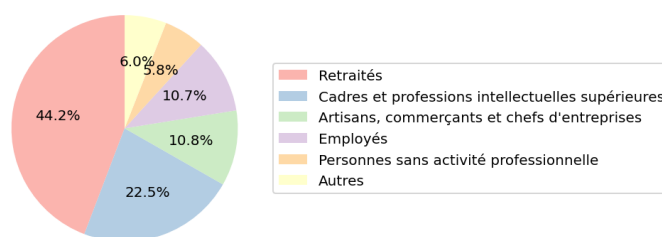


FIGURE 4 – Camembert de la répartition de l'épargne financière par CSP

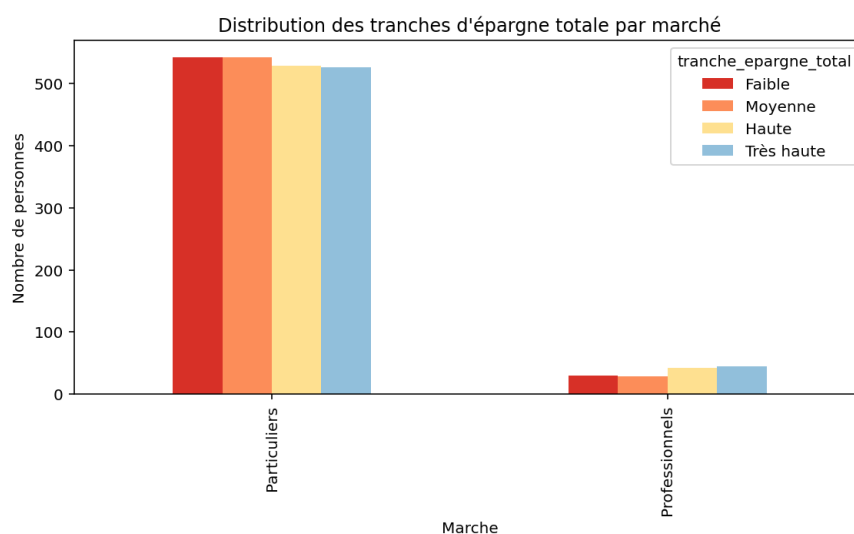


FIGURE 5 – Histogramme des tranches d'épargne totale par marché

6. Distribution des tranches d'épargne totale par type de client

L'analyse par type de client montre des contrastes importants :

- Les **adultes actifs non équipés** se concentrent surtout dans les tranches d'épargne faibles à moyennes
- Les **adultes actifs équipés**, en revanche, se distinguent par une forte présence dans les tranches hautes et très hautes, traduisant l'impact direct de l'équipement bancaire et assurantiel
- Les **adultes actifs équipés assurés** sont davantage répartis dans les niveaux faibles à moyens, ce qui suggère des profils plus hétérogènes
- Les **jeunes adultes actifs non équipés** disposent de très faibles niveaux d'épargne, quasiment inexistantes dans les tranches supérieures
- Les **jeunes adultes actifs équipés (assurés ou non)** présentent une légère amélioration, mais restent globalement cantonnés aux tranches faibles à moyennes
- Les profils **premium** montrent un contraste très marqué : les **premium actifs équipés** et les **premium inactifs** concentrent quasi exclusivement leur épargne dans la tranche *très haute*.

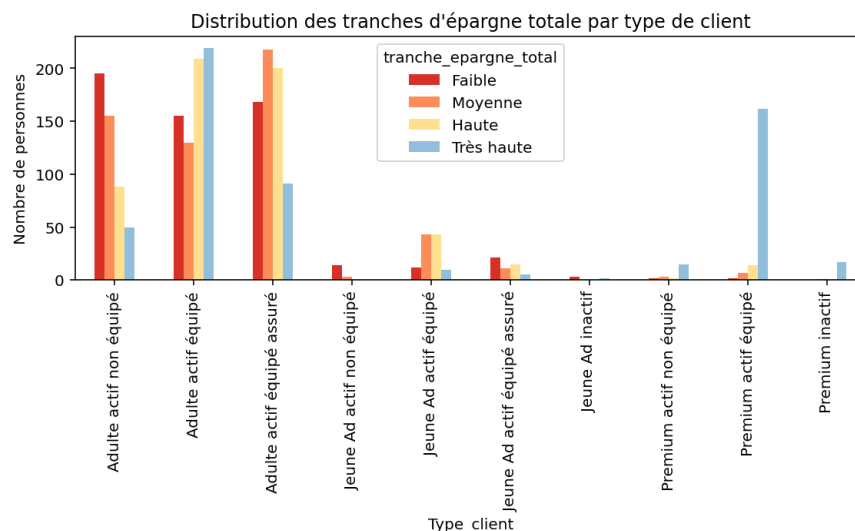


FIGURE 6 – Histogramme des tranches d'épargne totale par type de client

Cela confirme leur statut patrimonial privilégié, fortement surreprésenté dans les épargnes les plus élevées

Ainsi, l'équipement bancaire et la catégorie premium apparaissent comme des facteurs discriminants majeurs pour expliquer le niveau d'épargne.

2.2 Conclusion

L'analyse exploratoire des données permet de dégager plusieurs enseignements clés :

- **Âge** : l'épargne (totale et financière) augmente régulièrement avec l'âge, confirmant un mécanisme d'accumulation patrimoniale.
- **CSP** : les retraités dominent largement, suivis des cadres supérieurs, illustrant l'importance d'une carrière longue ou de revenus élevés dans la constitution d'un patrimoine.
- **Marché** : les particuliers sont beaucoup plus nombreux, mais certains professionnels concentrent des montants d'épargne plus élevés.
- **Type de client** : l'équipement joue un rôle central. Les adultes actifs équipés apparaissent comme les plus solides en termes d'épargne, tandis que les jeunes actifs non équipés restent vulnérables. Enfin, les profils premium sont quasi exclusivement positionnés dans les niveaux d'épargne les plus élevés, marquant une polarisation très forte.

En résumé, l'âge, le statut socio-professionnel, l'équipement bancaire et l'appartenance à la catégorie premium constituent les principaux facteurs explicatifs de l'épargne. Ces résultats offrent une base précieuse pour la mise en place d'un modèle de prédiction visant à anticiper le potentiel d'épargne d'un individu en fonction de ses caractéristiques.