

Projet “Tinder”

Jedha FSDA

Contexte

Suite à l'organisation d'un speed dating, un sondage a été effectué auprès des participants.

Le but ? Essayer de juguler la baisse du nombre de matchs qui a été constaté sur Tinder.

Il en ressort un ensemble de données qui peuvent nous permettre de répondre à la question : qu'est ce qui fait que les personnes se plaisent mutuellement ?

Comment faire ? en récoltant ces données, en les analysant et obtenant des visualisations pertinentes.

Le principe

on confronte le nombre de match (colonne "match") avec des indicateurs pertinents issues des réponses au questionnaire.

On en dégage des possibles influences ou inférences.

quels indicateurs garder / exclure ?

- Exclus d'office : "attr1_1" et suivantes (attractivités recherchés chez l'autre), trop qualitatif pour être analysé

- le goal → **Oui mais...**
- l'âge → **Non**
- L'income → **Oui mais...**
- les préférences → **OK**

RGPD

Dans la mesure où les données de Speed+Dating+Data.csv ont été collectées en dehors de l'UE, les données personnelles peuvent être traitées

Dictionnaires de données

Même si pas très structuré, le dictionnaire de données donne des infos précieuses sur la signification des colonnes, permettant ainsi de lire correctement les résultats des analyses

La démarche

On confronte le nombre de match (colonne "match") avec des indicateurs pertinents issues des réponses au questionnaire. On en dégage des possibles influences ou inférences

Corrélation entre le goal et les matches

Le nombre de match dépend-il du "goal" que l'on a lorsqu'on se rend au speed dating ?

colonne : "goal"

"What is your primary goal in participating in this event?"

"Seemed like a fun night out=1

To meet new people=2

To get a date=3

Looking for a serious relationship=4

To say I did it=5

Other=6"

Il est possible de répondre à cette question mais avec précaution.
Pourquoi ?

Corrélation entre le goal et les matches

Sanity check 1 : 79 lignes de "goal" sont vides, il faudrait les remplir avec le mode (utile pour les valeurs déclaratives).

Sanity check 2 : La variabilité est médiocre. L'écart type est de 1.4 pour une moyenne de 2,12 sur des valeurs allant de 1 à 6 : il se situe à 12% par rapport à la moyenne.

– > cet indicateur est donc à manier avec précaution pour cette raison

```
#remplir les cellules vides avec le mode récupéré
df_filled = df1.fillna({'goal': mode_value})
df1 = df_filled
```

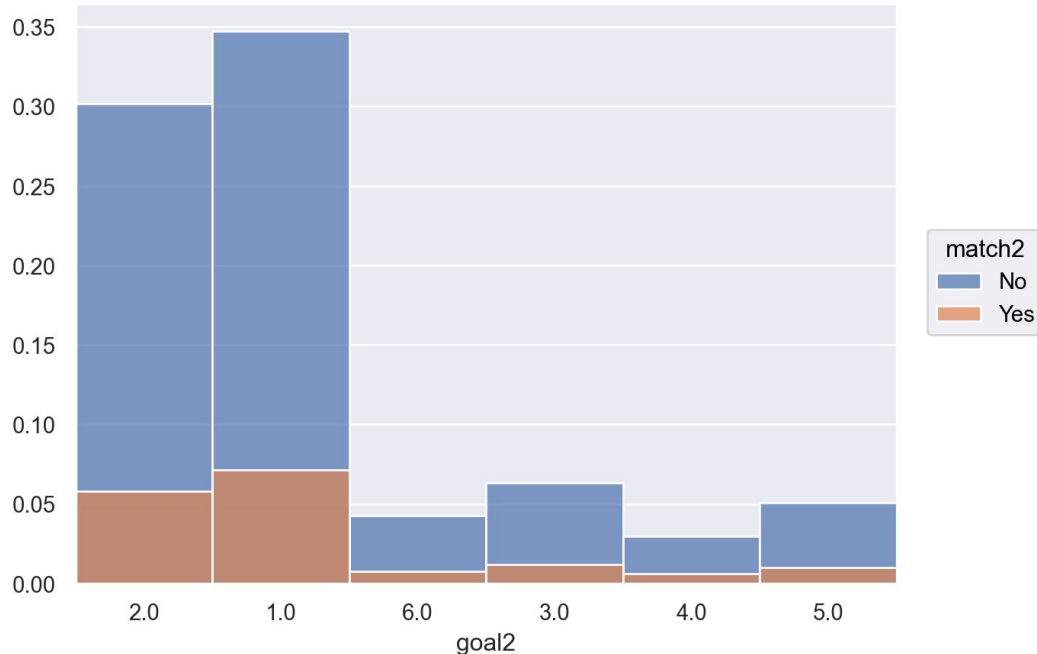
```
#Sanity check #2 : vérifier la bonne variabilité des données
df2.describe()
```

[33] ✓ 0.0s

| | match | goal |
|-------|-------------|-------------|
| count | 8378.000000 | 8299.000000 |
| mean | 0.164717 | 2.122063 |
| std | 0.370947 | 1.407181 |
| min | 0.000000 | 1.000000 |
| 25% | 0.000000 | 1.000000 |
| 50% | 0.000000 | 2.000000 |
| 75% | 0.000000 | 2.000000 |
| max | 1.000000 | 6.000000 |

Corrélation entre le goal et les matches

#Dataviz qui montre la relation entre les matchs et les réponses de "Goal"
#Utilisation de SO pour les histogrammes, plus lisibles

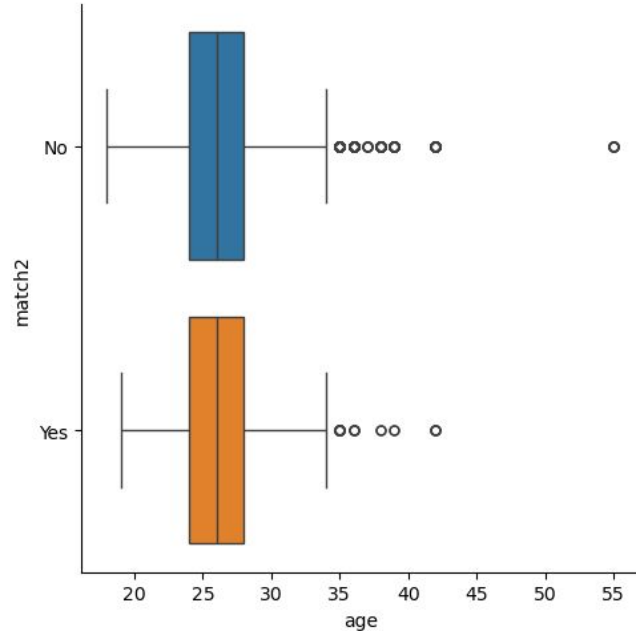


Conclusion : la corrélation entre les matchs et la valeur 1 de Goal "Seemed like a fun night out" est la plus forte. Cette prédiction est à nuancer vu la médiocre variabilité.

Corrélation avec l'âge

#la variabilité est trop faible (trop de gens ont le même âge), et l'écart type (5,45 % avec les outliers, 8,8 sans ceux là) est trop réduit par rapport à la moyenne

```
count    8283.000000
mean      26.358928
std        3.566763
min       18.000000
25%       24.000000
50%       26.000000
75%       28.000000
max       55.000000
Name: age, dtype: float64
```



Conclusion : par sa très faible variabilité , cet indicateur sera exclu des prédictions.

Corrélation avec le revenu ("income")

#Sanity check 1 : complétude des données. La moitié des données sont manquantes (4279 sur 8378)

On peut quand même se baser sur la population renseignée si les données manquantes ne sont pas liés à la variable cible (le match).

```
#Vérification des opérations de conversion en float et de l'exclusion des NaN
df1IncomeNotNull["income"]

[27] ✓ 0.0s

... 0      69487.00
     1      69487.00
     2      69487.00
     3      69487.00
     4      69487.00
     ...
    8351     55138.00
    8352     55138.00
    8353     55138.00
    8354     55138.00
    8355     55138.00
Name: income, Length: 4279, dtype: object
```

Voyons si c'est le cas !

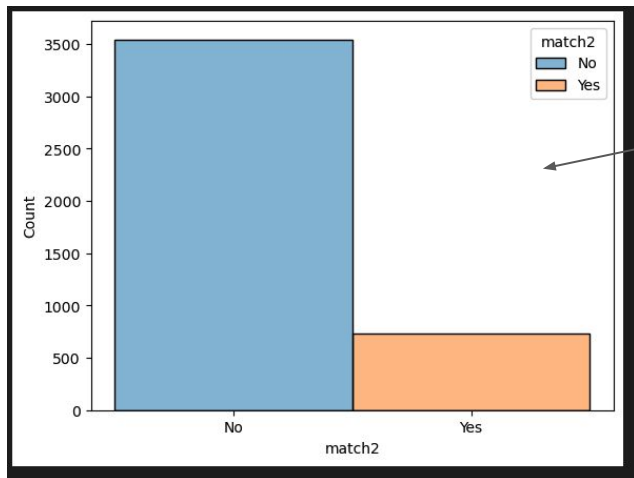
Si on démontre que les matchs sont équivalents dans la population renseignée et dans la population non renseignée, alors cet indicateur sera ok.

Corrélation avec le revenu ("income")

On voit que les match YES et les match NO sont comparables de manière satisfaisante

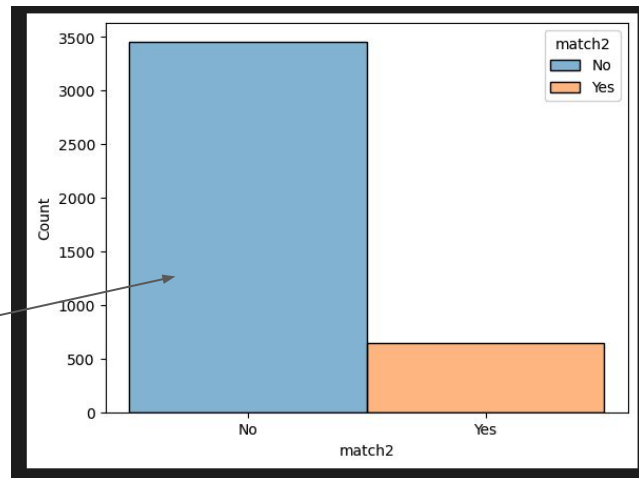
La différence de YES est de 14 %, ceux des NO de 2,5 %

| | countMatchYesIncomeNotNull | countMatchNoIncomeNotNull | countMatchYesIncomeNull | countMatchNoIncomeNull | percent_diff_Yes | percent_diff_No |
|---|----------------------------|---------------------------|-------------------------|------------------------|------------------|-----------------|
| 0 | 644 | 3455 | 736 | 3543 | -14.285714 | -2.547033 |



Population
non
renseignée

Population
renseignée



Corrélation avec le revenu (“income”)

#Sanity check 2 : variabilité. Sur 4279 income renseigné, seuls 261 sont uniques, le reste a la même valeur. Cela reste suffisant pour dresser des tendances

```
totalcountincome = df1IncomeNotNull["income"].count()
uniquecountincome = df1IncomeNotNull["income"].nunique()
uniquecountincomeduplicated = totalcountincome - uniquecountincome
print(totalcountincome)
print(uniquecountincome)
print(uniquecountincomeduplicated)
```

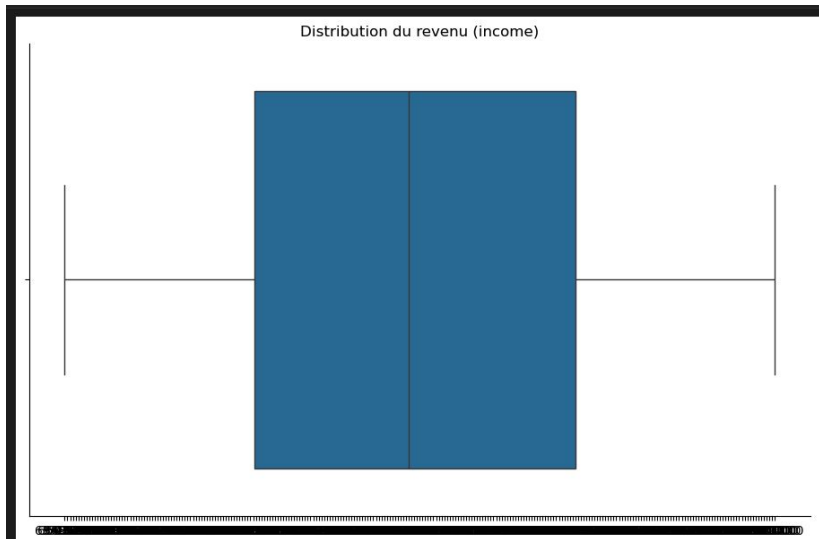
```
4279
261
4018
```

Voyons maintenant si ces valeurs sont suffisamment bien distribués !

Corrélation avec le revenu (“income”)

Variabilité suffisante : écart type qui représente **38 % de la moyenne**, distribution relativement étalée

```
...   count      4279.000000  
      mean      44887.606450  
      std       17206.920962  
      min       8607.000000  
      25%       31516.000000  
      50%       43185.000000  
      75%       54303.000000  
      max       109031.000000  
      Name: income, dtype: float64
```



Et aucun outliers !
On va pouvoir voir la
corrélation entre
match et income.
**Mais comment le
faire efficacement ?**

Corrélation avec le revenu (“income”)

#Création de catégories d'income pour une meilleure lisibilité (on passe du numérique au catégoriel)

```
df1IncomeNotNullUnique["income"] = df1IncomeNotNullUnique["income"].astype("float64")
df1IncomeNotNullUnique2 = df1IncomeNotNullUnique
df1IncomeNotNullUnique2["income"] = df1IncomeNotNullUnique2["income"].apply([
    lambda x: "low" if x <= 31000
    else "mediumlow" if 31000 < x <= 43000
    else "mediumhigh" if 43000 < x <= 54000
    else "high"
```

```
#Filtre sur march = yes
maskNotNullMatched = (df1IncomeNotNull2["match2"] == "Yes")
```

[48] ✓ 0.0s

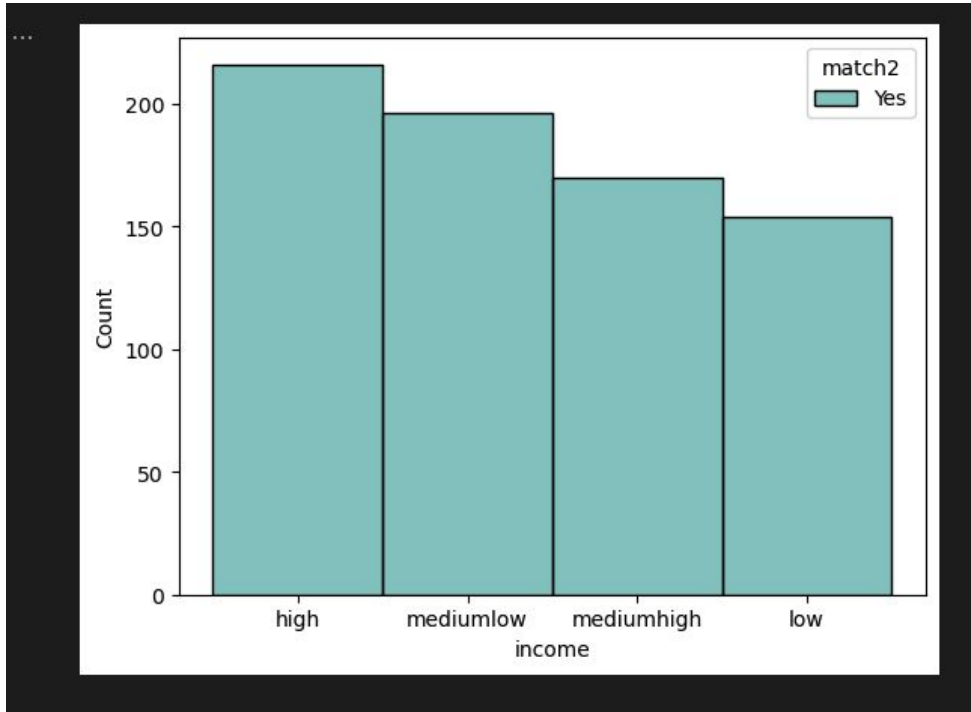
```
#Vérification du filtrage
df1IncomeNotNull2Matched = df1IncomeNotNull2.loc[maskNotNullMatched]
df1IncomeNotNull2Matched["match2"]
```

[49] ✓ 0.0s

#On focus sur les “match = yes” (4 fois moins nombreux) pour une échelle lisible

Corrélation avec le revenu ("income")

#Création d'un histplot pour voir la corrélation



Conclusion : les revenus les plus élevés sont plus susceptibles d'obtenir un match que les revenus faibles. Les revenus moyens sont intermédiaires

Corrélation avec les préférences d'activités

#De quoi parle-t-on ?

- des colonnes 50:67 du dataset.

12. How interested are you in the following activities, on a scale of 1-10?

sports: Playing sports/ athletics

tvsports: Watching sports

excercise: Body building/exercising

dining: Dining out

museums: Museums/galleries

art: Art

hiking: Hiking/camping

gaming: Gaming

clubbing: Dancing/clubbing

reading: Reading

tv: Watching TV

theater: Theater

movies: Movies

concerts: Going to concerts

music: Music

shopping: Shopping

yoga: Yoga/meditation

#que cherche t on à savoir :

Quelle relation entre le nombre de match et les activités préférées?

Est ce que cet indicateur est fiable ?

Corrélation avec les préférences d'activités

#Sanity check 1

- vérif de la complétude.

Seuls 79 lignes contiennent des valeurs NaN, soit 0.94% des données

Marginal sur l'inférence, la complétude est donc satisfaisante

- Sanity check 2 : pas d'analyse à faire sur la variabilité, les données récoltées sont subjectives et non démographiques .

Indicateur OK !

Corrélation avec les préférences d'activités


#Arrangement des données (filtre : match = yes)

- je n'aime pas ou pas beaucoup = 0 (si score entre 1 et 5),
- j'aime bien ou beaucoup = 1 (si score entre 6 et 10)

```
for col in dflactivities.columns:  
    dflactivities[col] = dflactivities[col].apply(lambda x: 0 if 1 <= x <= 5 else 1)
```

#on crée un dataset de count de chaque match par activité distribué par préférence 0 ou 1

```
results = []  
for col in dflactivities.columns[:-1]: # Exclure la colonne "match"  
    count_0 = len(dflactivities[(dflactivities[col] == 0) & (dflactivities["match"] == 1)])  
    count_1 = len(dflactivities[(dflactivities[col] == 1) & (dflactivities["match"] == 1)])  
    results.append({"Activité": col, "0": count_0, "1": count_1})  
  
results_df = pd.DataFrame(results)
```



| | Activité | 0 | 1 |
|----|----------|-----|------|
| 0 | sports | 458 | 922 |
| 1 | tvsports | 842 | 538 |
| 2 | exercise | 489 | 891 |
| 3 | dining | 154 | 1226 |
| 4 | museums | 292 | 1088 |
| 5 | art | 352 | 1028 |
| 6 | hiking | 562 | 818 |
| 7 | gaming | 982 | 398 |
| 8 | clubbing | 466 | 914 |
| 9 | reading | 143 | 1237 |
| 10 | tv | 693 | 687 |
| 11 | theater | 368 | 1012 |
| 12 | movies | 131 | 1249 |
| 13 | concerts | 308 | 1072 |
| 14 | music | 138 | 1242 |
| 15 | shopping | 614 | 766 |
| 16 | yoga | 813 | 567 |

Corrélation avec les préférences d'activités

Besoin de matplotlib et de numpy pour répondre au besoin : avec notamment "np.arange" pour extraire les valeurs d'une seule colonne et en faire un array affiché en X et "bottom" pour pouvoir empiler 2 colonnes sur un seul histogramme :

```
x = np.arange(len(results_df["Activité"]))

ax.bar(x, results_df["0"], width=bar_width, color="#ff9999", label="N'aime pas (0)")
ax.bar(x, results_df["1"], bottom=results_df["0"], width=bar_width, color="#66b3ff", label="Aime (1)")

ax.set_xticks(x)
ax.set_xticklabels(results_df["Activité"], rotation=45, ha="right")
ax.set_ylabel("Nombre de matchs")
ax.set_title("Impact de la préférence pour une activité sur le nombre de matchs")
ax.legend()

plt.tight_layout()
plt.show()
```

Corrélation avec les préférences d'activités

