03_Analisis_Exploratorio

September 4, 2024

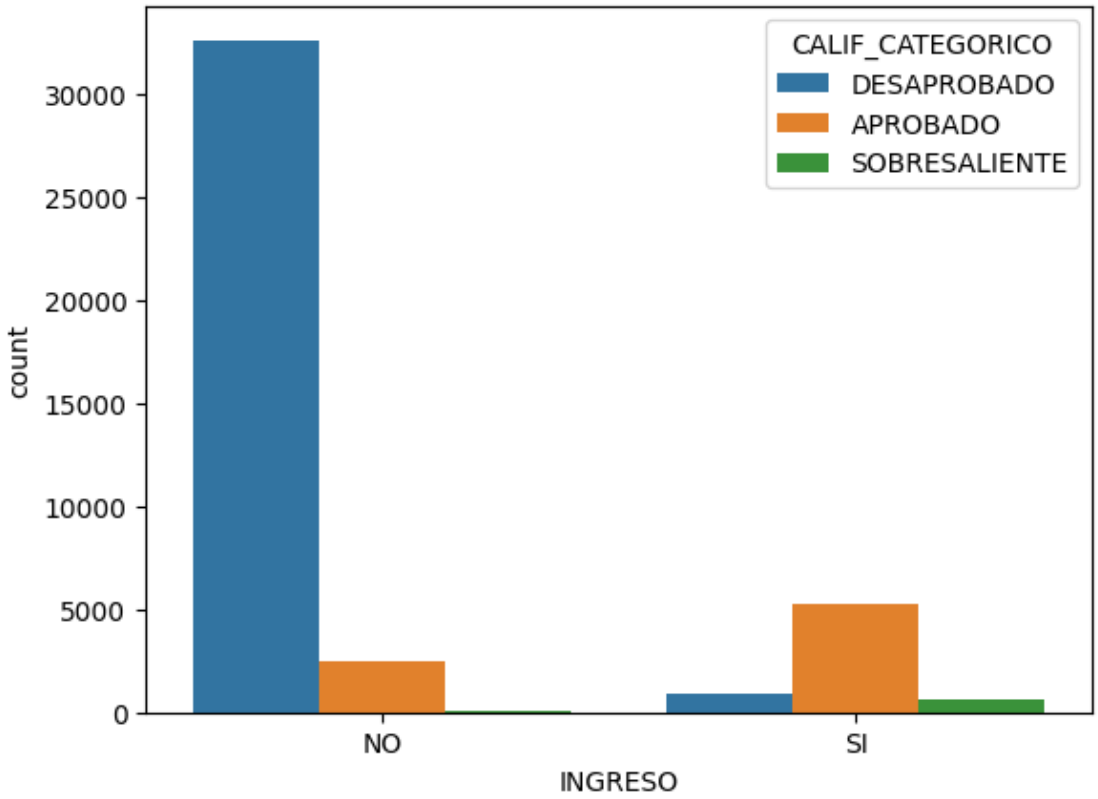
[ ]:
```
# Importamos librerías
import pandas as pd
import numpy as  np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[ ]:
```
# 1. Importamos la base de datos
## Cargamos la base de datos

df = pd.read_excel('data/data_preprocessed.xlsx')
df.head()
```

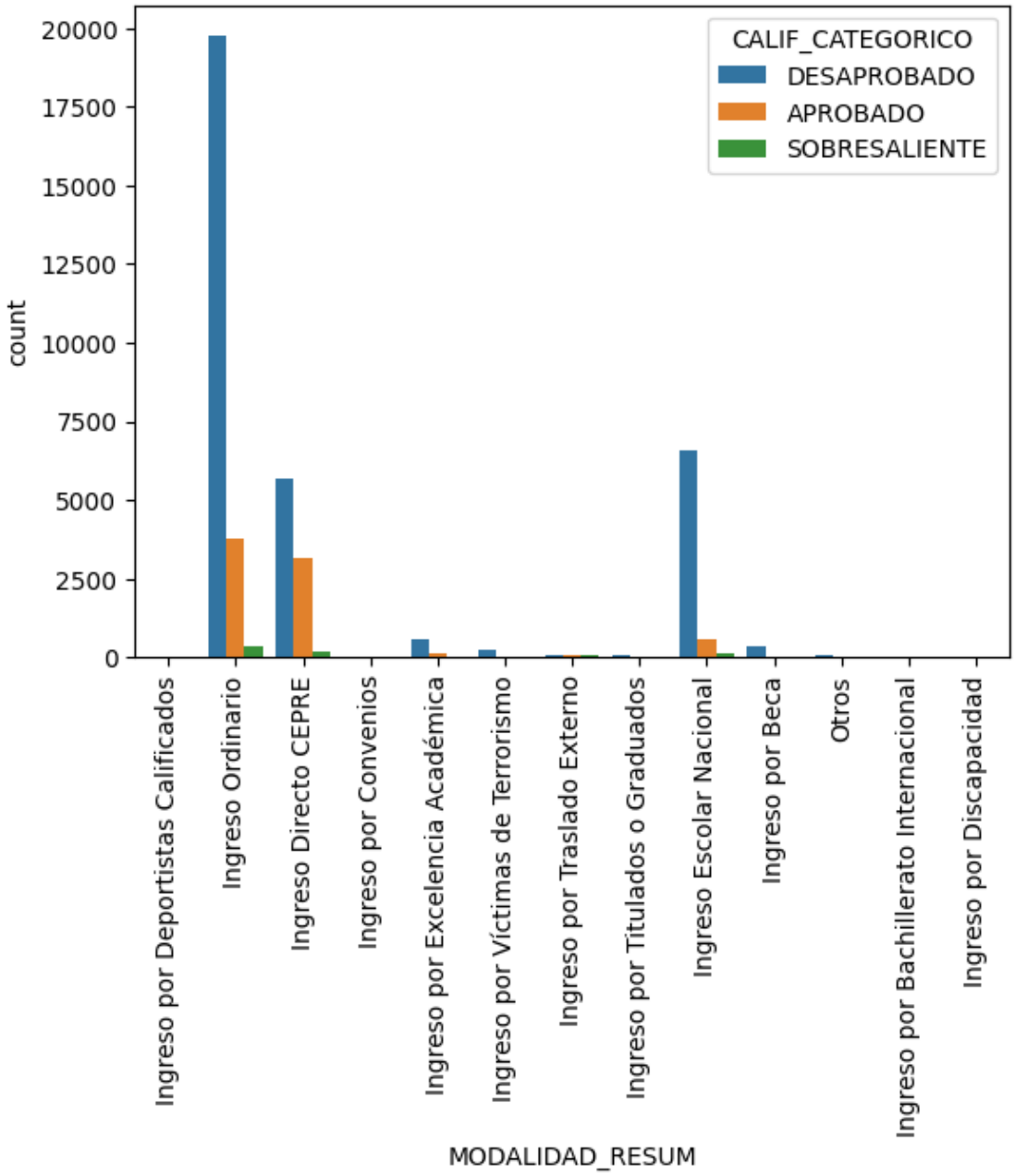
[ ]:
```
                                              IDHASH  \
0  E4287C2FE19F63C5E6641955147E36684A5A2FF8064676…
1  2DC37F0B9727B6591EC72D7A942647797A200F45D47C9E…
2  3B8677B90781D7BB8F2F967C05FA2DBBE153BBB682DF05…
3  FA366704D9E9F6FB5E5F55C1FB0CEEE973C626A5616F55…
4  E31CF8F30F3AE60B3D8A14F6E1020E9AD26EE975F0823B…

                    COLEGIO COLEGIO_DEPA COLEGIO_PROV  \
0       LA DIVINA PROVIDENCIA         LIMA         LIMA
1           86019 LA LIBERTAD       ANCASH       HUARAZ
2  0113 DANIEL ALOMIAS ROBLES         LIMA         LIMA
3     SEBASTIAN SALAZAR BONDY         LIMA         LIMA
4           TRILCE LOS OLIVOS         LIMA         LIMA

             COLEGIO_DIST COLEGIO_PAIS  COLEGIO_ANIO_EGRESO  \
0               SURQUILLO        PERÚ                 2020
1                  HUARAZ        PERÚ                 2017
2  SAN JUAN DE LURIGANCHO        PERÚ                 2016
3            SANTA ANITA         PERÚ                 2018
4    SAN MARTIN DE PORRES        PERÚ                 2020

                      ESPECIALIDAD  ANIO_POSTULA  CICLO_POSTULA  …  \
0           INGENIERÍA DE SISTEMAS          2021              1  …
1  INGENIERÍA DE TELECOMUNICACIONES          2021              1  …
```

```
2               INGENIERÍA MECÁNICA          2021              1  …
3            INGENIERÍA ELECTRÓNICA          2021              1  …
4                      ARQUITECTURA          2021              1  …

  NACIMIENTO_PAIS NACIMIENTO_DEPA NACIMIENTO_PROV    NACIMIENTO_DIST  \
0           PERÚ            LIMA            LIMA  VILLA EL SALVADOR
1           PERÚ          ÁNCASH          HUARAZ             HUARAZ
2           PERÚ            LIMA            LIMA        JESÚS MARÍA
3           PERÚ            LIMA            LIMA               LIMA
4           PERÚ            LIMA            LIMA              COMAS

       SEXO CALIF_FINAL INGRESO  \
0  MASCULINO        3.11      NO
1   FEMENINO        3.58      NO
2  MASCULINO        8.04      NO
3  MASCULINO       10.82      NO
4   FEMENINO        6.21      NO

                                          MODALIDAD  \
0  EXTRAORDINARIO1 - DEPORTISTAS CALIFICADOS DE A…
1                                          ORDINARIO
2                                          ORDINARIO
3           EXTRAORDINARIO2 - INGRESO DIRECTO CEPRE
4                                          ORDINARIO

                      MODALIDAD_RESUM  CALIF_CATEGORICO
0  Ingreso por Deportistas Calificados       DESAPROBADO
1                   Ingreso Ordinario       DESAPROBADO
2                   Ingreso Ordinario       DESAPROBADO
3               Ingreso Directo CEPRE       DESAPROBADO
4                   Ingreso Ordinario       DESAPROBADO

[5 rows x 24 columns]
```

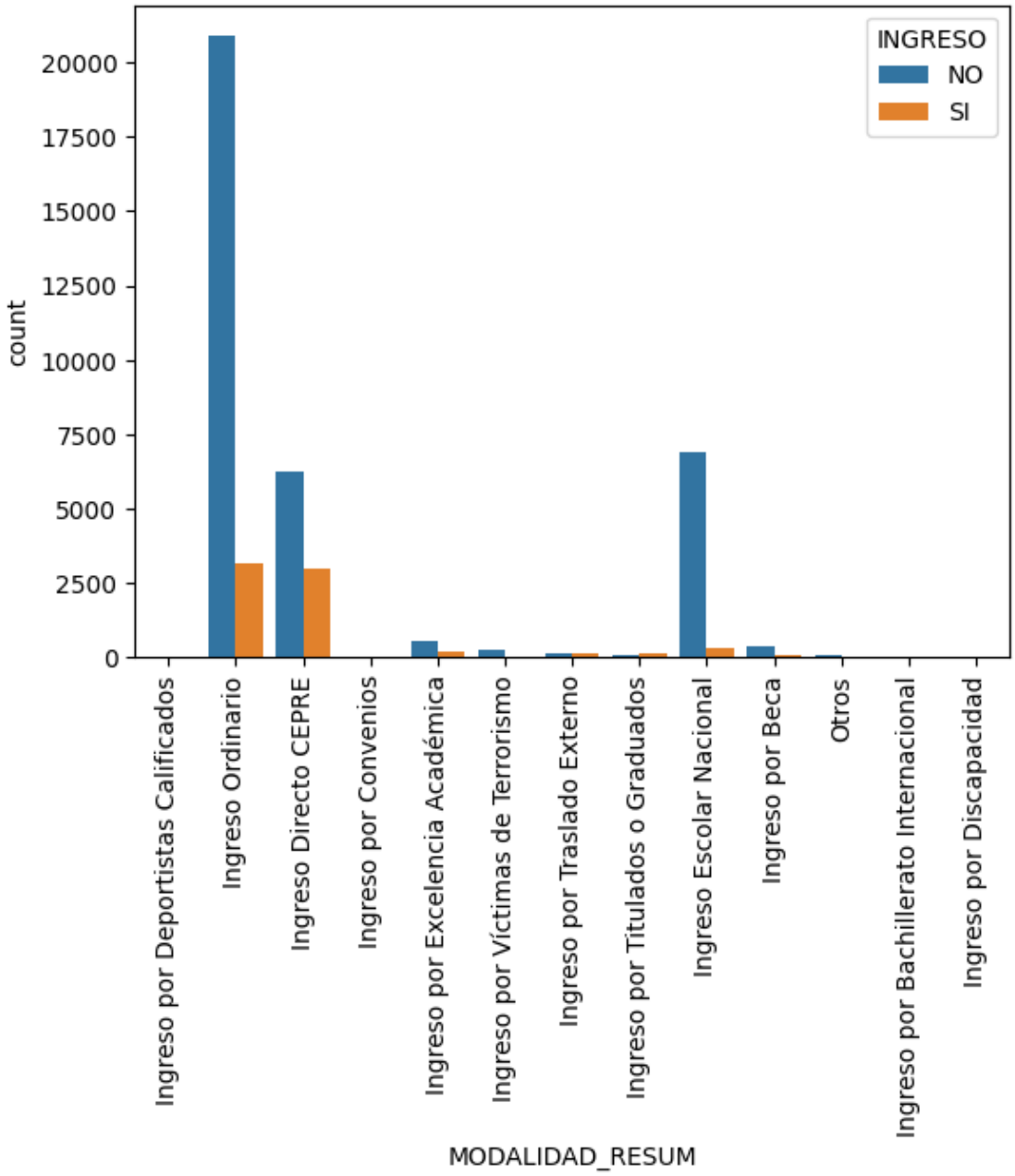
[ ]:
```
# 2. Analizamos gráficamente el dataframe

## Graficamos la dicotómica (Ingresó: Si o No) según la calificación categórica
sns.countplot(data=df, x="INGRESO", hue="CALIF_CATEGORICO")
plt.show()
```



[ ]:
```
## Graficamos la cantidad de estudiantes por modalidad y categoría de␣
 ↪calificación

sns.countplot(data=df, x="MODALIDAD_RESUM", hue="CALIF_CATEGORICO")
plt.xticks(rotation=90)
plt.show()
```

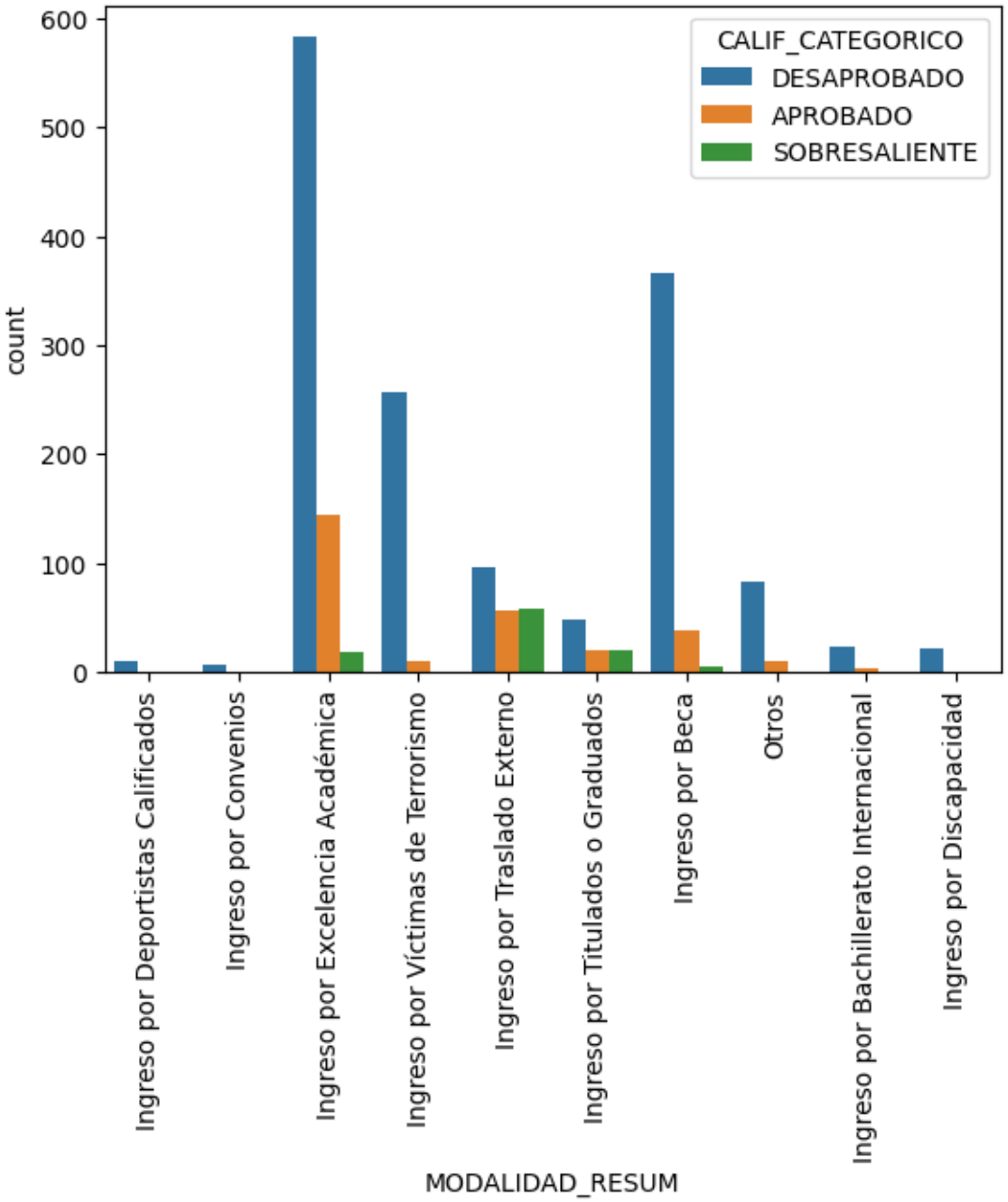


[ ]:
```
## Graficamos la cantidad de estudiantes por modalidad y dicotómica (Ingresó: Si o No)
sns.countplot(data=df, x="MODALIDAD_RESUM", hue="INGRESO")
plt.xticks(rotation=90)
plt.show()
```



[ ]:
```
# 3.  Filtramos las modalidades de ingreso más comunes para analizar a detalle␣
 ↪las de menor ingreso
filtered_df = ~df['MODALIDAD_RESUM'].isin(['Ingreso Ordinario', 'Ingreso␣
 ↪Directo CEPRE', 'Ingreso Escolar Nacional'])
filtered_df = pd.DataFrame(df[filtered_df])

## Graficamos la cantidad de estudiantes por modalidad y categoría de␣
 ↪calificación
```

```
sns.countplot(data=filtered_df, x="MODALIDAD_RESUM", hue="CALIF_CATEGORICO")
plt.xticks(rotation=90)
plt.show()
```



[ ]:
```
## Graficamos la cantidad de estudiantes por modalidad y dicotómica (Ingresó: Si o No)
sns.countplot(data=filtered_df, x="MODALIDAD_RESUM", hue="INGRESO")
```

```
plt.xticks(rotation=90)
plt.show()
```