

Proyecto 14INV-457 – Programa de diagnóstico precoz del cáncer de próstata

Informe Técnico de Avance Núm. 7

RESULTADOS DE LAS HISTORIAS CLÍNICAS

1 Análisis de datos

El análisis de datos se realizó con Python 3.8 (Anaconda Distribution 2020-07, Anaconda, Inc., Austin, TX), usando las librerías *Pandas* y *Numpy* para el análisis estadístico, y las librerías *Matplotlib* y *Seaborn* para los gráficos. El reporte fue generado usando RStudio versión 1.3.1093 a través del paquete *Reticulate*. Los datos fueron analizados por el Investigador Principal del proyecto, Prof. Dr. Alcides Chaux.

- **Fecha de cierre del periodo de análisis:** 10 de marzo de 2019
- **Número de historias clínicas a la fecha:** 294 pacientes

```
library(reticulate)
use_condaenv("r-reticulate")

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_pickle('ITA_7.pkl')
```

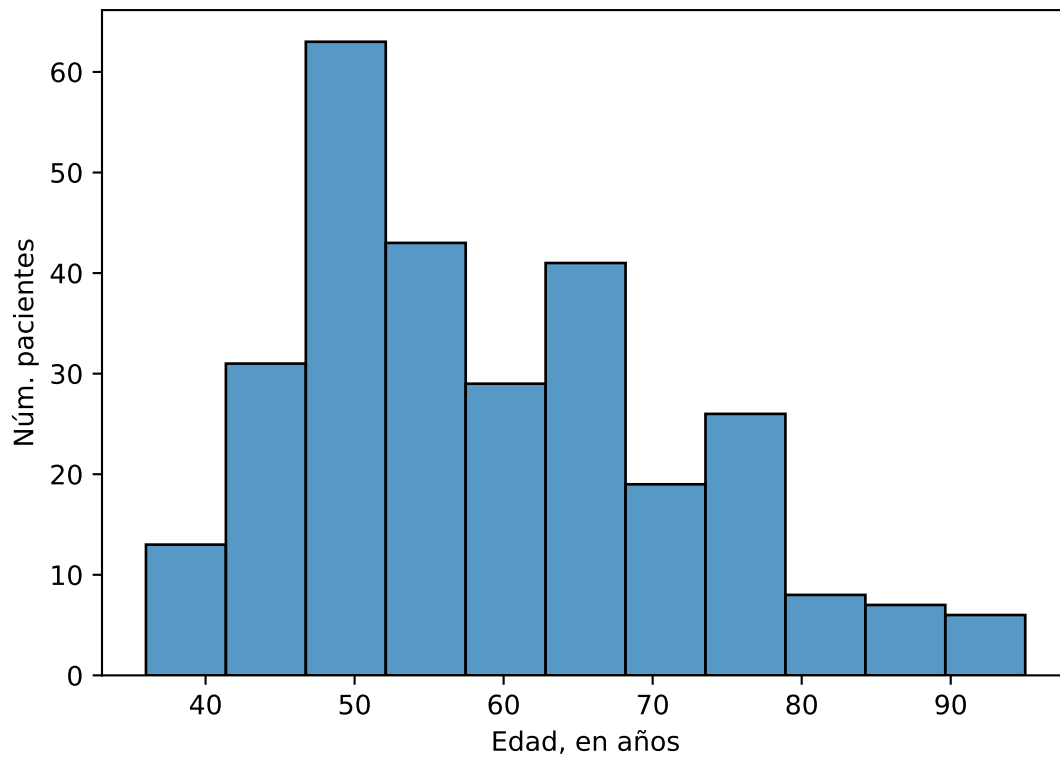
2 Datos personales

2.1 Edad

```
df['edad'].describe().round()

## count      286.0
## mean       59.0
## std        13.0
## min        36.0
## 25%        49.0
## 50%        57.0
## 75%        67.0
## max        95.0
## Name: edad, dtype: float64
```

```
sns.histplot(x='edad', data=df)
plt.xlabel('Edad, en años')
plt.ylabel('Núm. pacientes')
plt.show()
```



2.2 Procedencia

Las 10 ciudades de procedencia más frecuentes

```
df['ciudad'].value_counts().head(10)
```

```
## Asunción          94
## Luque             41
## Capiatá           18
## San Lorenzo       18
## Lambaré           17
## Ferndo de la Mora 14
## Villa Elisa       13
## Mariano Roque Alonso 12
## Itá               10
## Ñemby              7
## Name: ciudad, dtype: int64
```

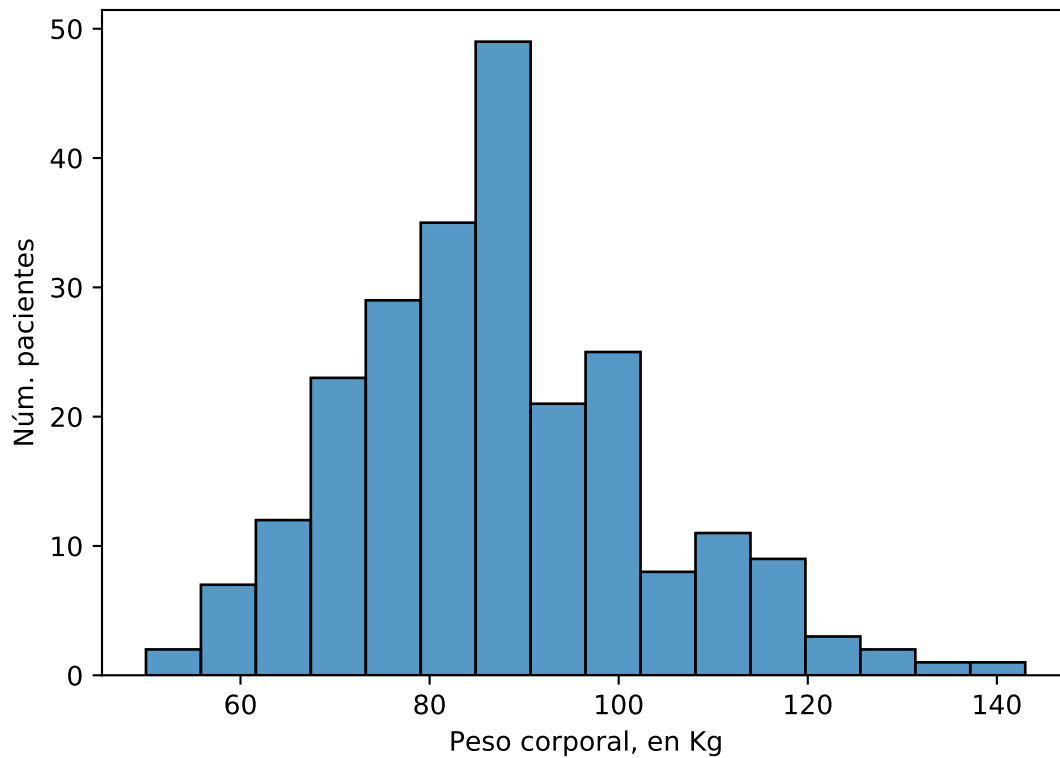
3 Datos antropométricos

3.1 Peso, en Kg

```
df['peso'].describe().round()
```

```
## count    238.0
## mean      87.0
## std       16.0
## min       50.0
## 25%       78.0
## 50%       85.0
## 75%       97.0
## max      143.0
## Name: peso, dtype: float64
```

```
sns.histplot(x='peso', data=df, edgecolor='black')
plt.xlabel('Peso corporal, en Kg')
plt.ylabel('Núm. pacientes')
plt.show()
```

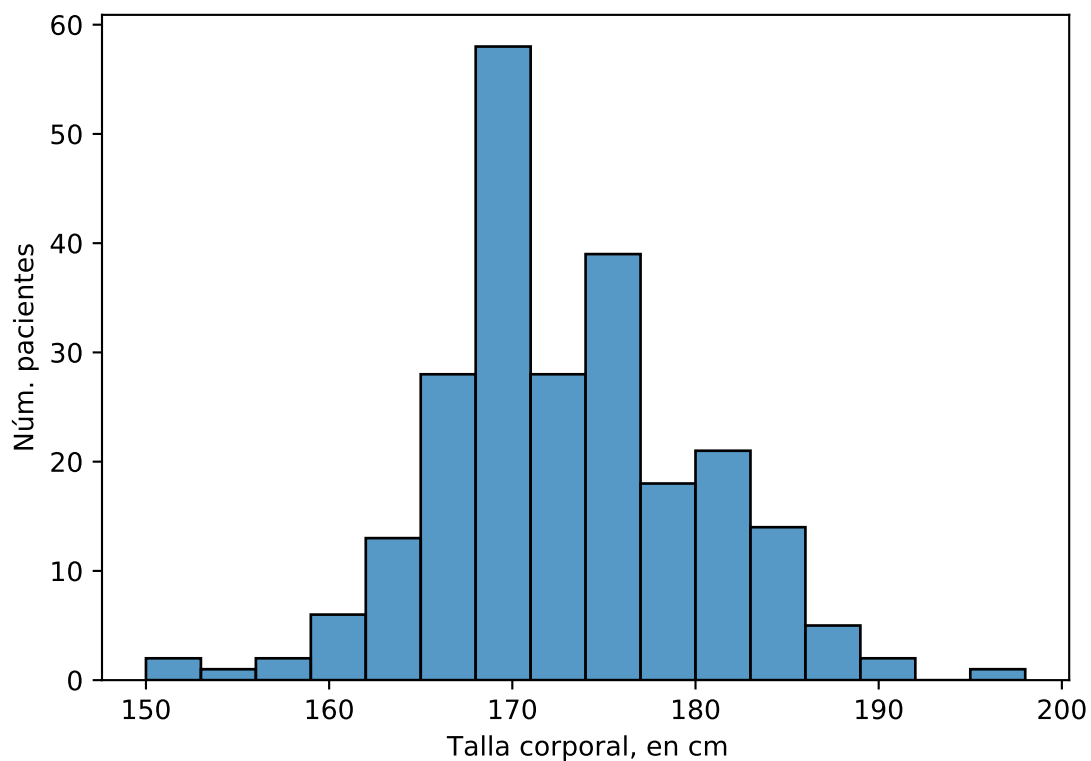


3.2 Talla, en cm

```
df['estatura'].describe().round()
```

```
## count    238.0
## mean     172.0
## std       7.0
## min      150.0
## 25%      168.0
## 50%      172.0
## 75%      178.0
## max      198.0
## Name: estatura, dtype: float64
```

```
sns.histplot(x='estatura', data=df, edgecolor='black')
plt.xlabel('Talla corporal, en cm')
plt.ylabel('Núm. pacientes')
plt.show()
```

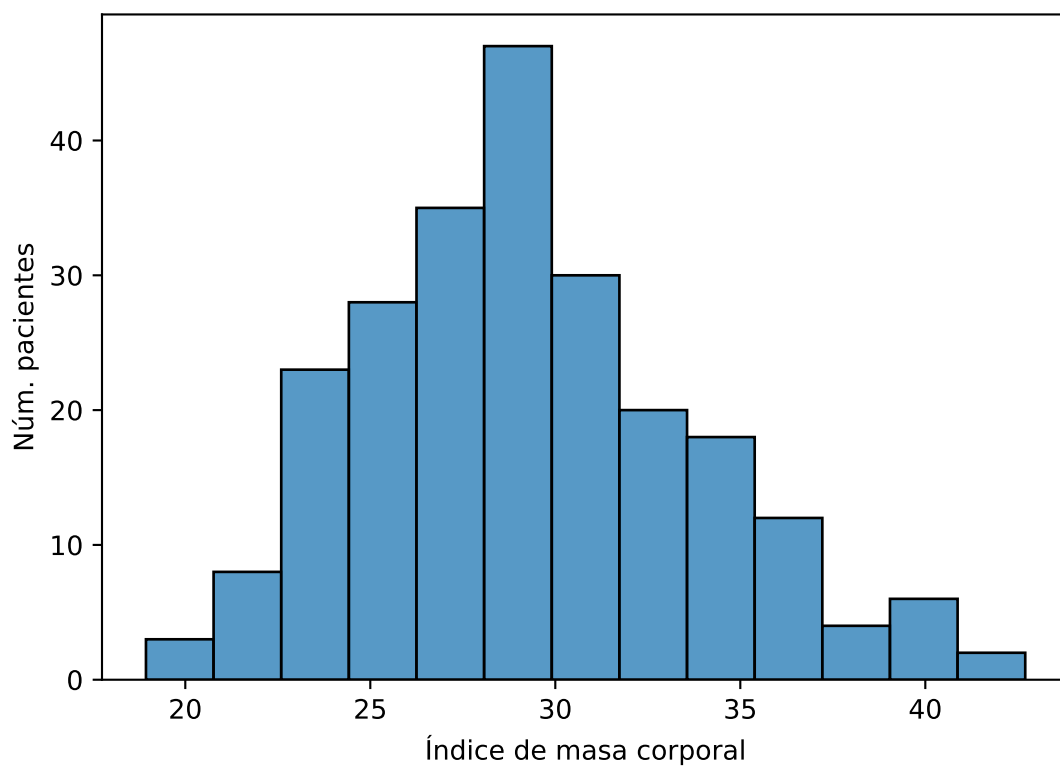


3.3 Índice de masa corporal

```
df['imc'] = df['peso']/(df['estatura']/100)**2
df['imc'].describe().round()
```

```
## count    236.0
## mean      29.0
## std       4.0
## min       19.0
## 25%       26.0
## 50%       29.0
## 75%       32.0
## max       43.0
## Name: imc, dtype: float64
```

```
sns.histplot(x='imc', data=df, edgecolor='black')
plt.xlabel('Índice de masa corporal')
plt.ylabel('Núm. pacientes')
plt.show()
```



4 Antecedentes patológicos

4.1 Hipertensión arterial

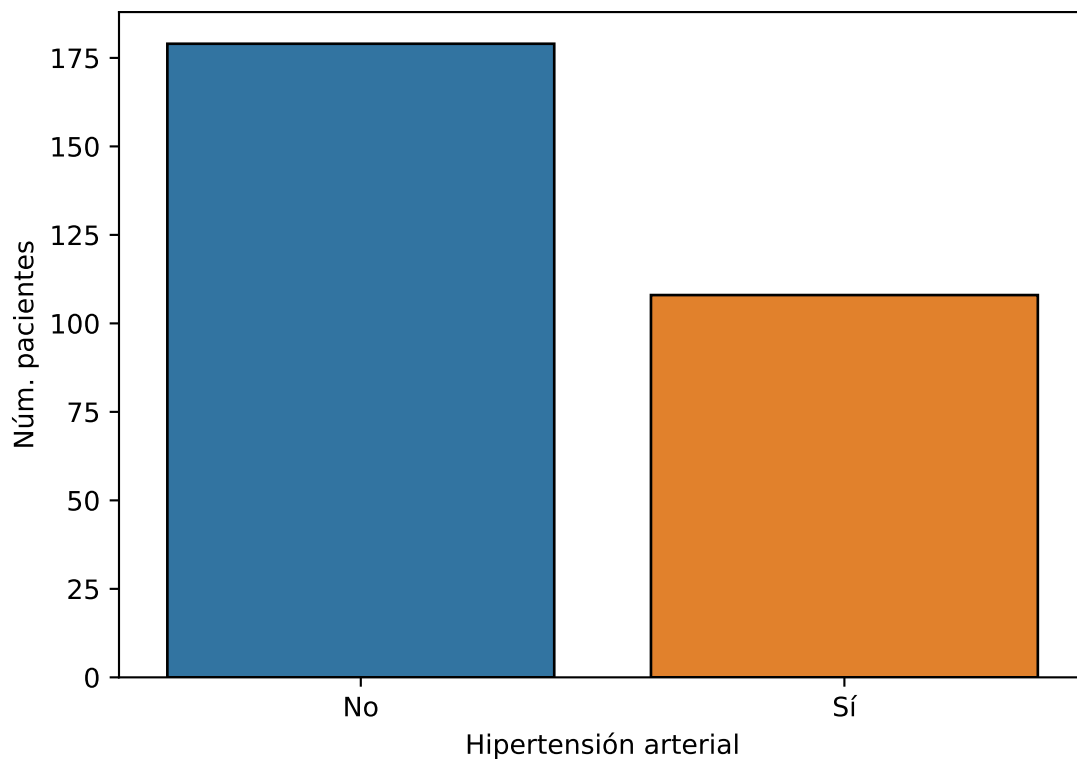
```
var = 'hta'  
df[var].value_counts()
```

```
## No      179  
## Sí      108  
## Name: hta, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.62  
## Sí      0.38  
## Name: hta, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Hipertensión arterial')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



4.2 Diabetes mellitus

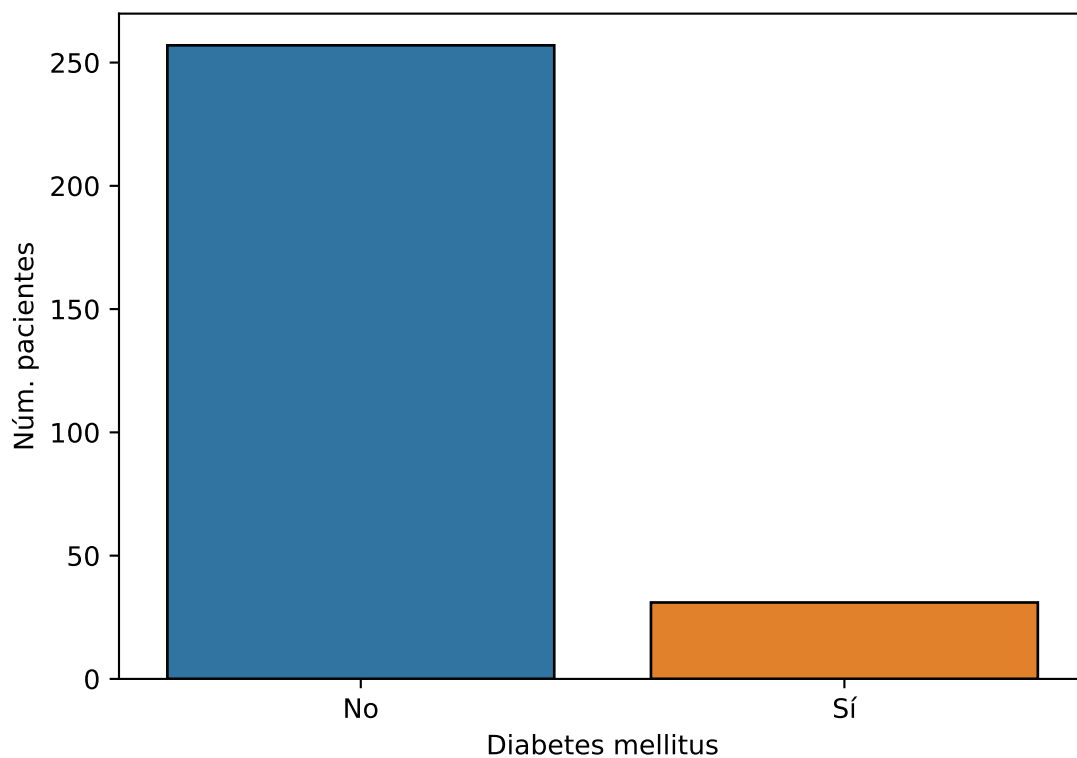
```
var = 'dm'  
df[var].value_counts()
```

```
## No      257  
## Sí       31  
## Name: dm, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.89  
## Sí      0.11  
## Name: dm, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Diabetes mellitus')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



4.3 Tumor no prostático

4.4 Hipertensión arterial

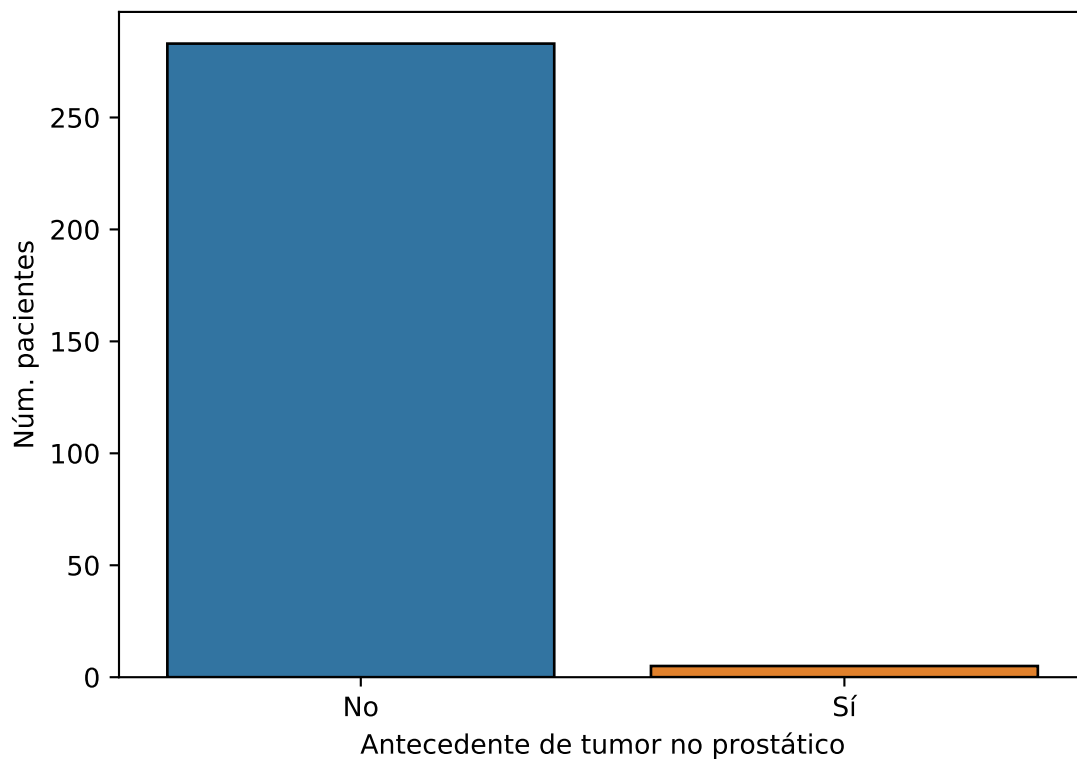
```
var = 'tumor'  
df[var].value_counts()
```

```
## No      283  
## Sí       5  
## Name: tumor, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.98  
## Sí      0.02  
## Name: tumor, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Antecedente de tumor no prostático')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



4.5 Tipo de tumor no prostático

```
df['tumor_tipo'].value_counts()
```

```
## Mieloma múltiple      1
## Tiroides              1
## Pólipos colónicos     1
## Name: tumor_tipo, dtype: int64
```

4.6 Hiperplasia prostática

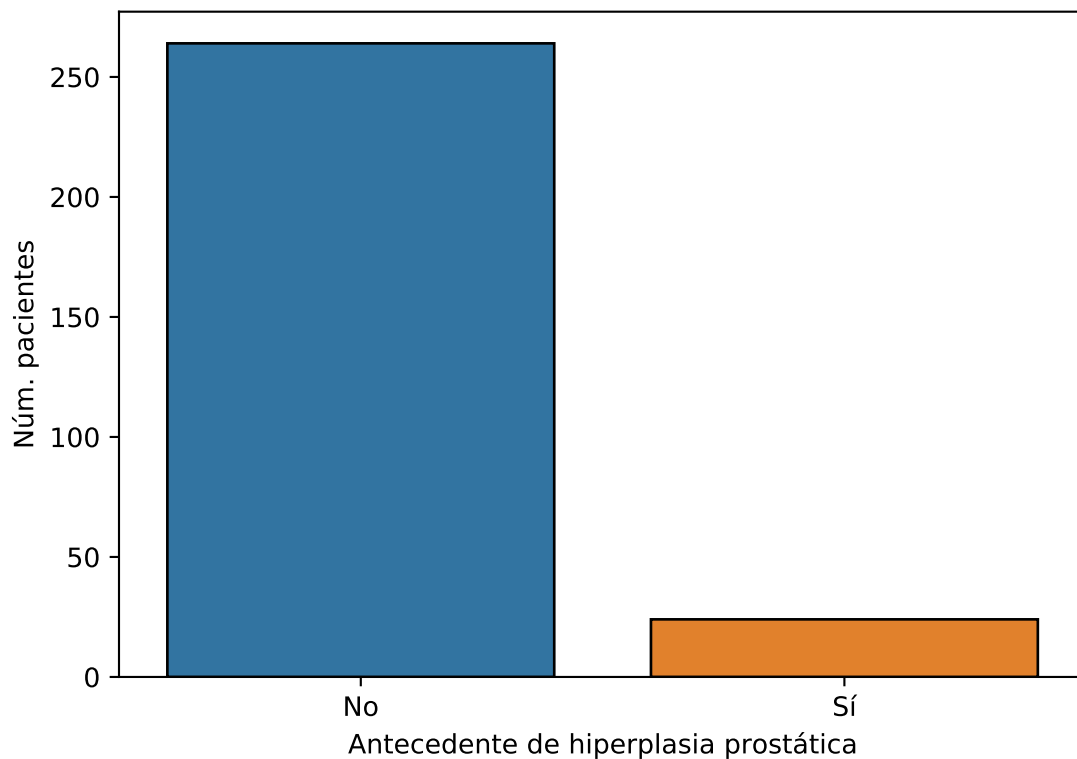
```
var = 'hp'
df[var].value_counts()
```

```
## No      264
## Si       24
## Name: hp, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.92
## Si      0.08
## Name: hp, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')
plt.xlabel('Antecedente de hiperplasia prostática')
plt.ylabel('Núm. pacientes')
plt.show()
```



4.7 Cáncer de próstata familiar

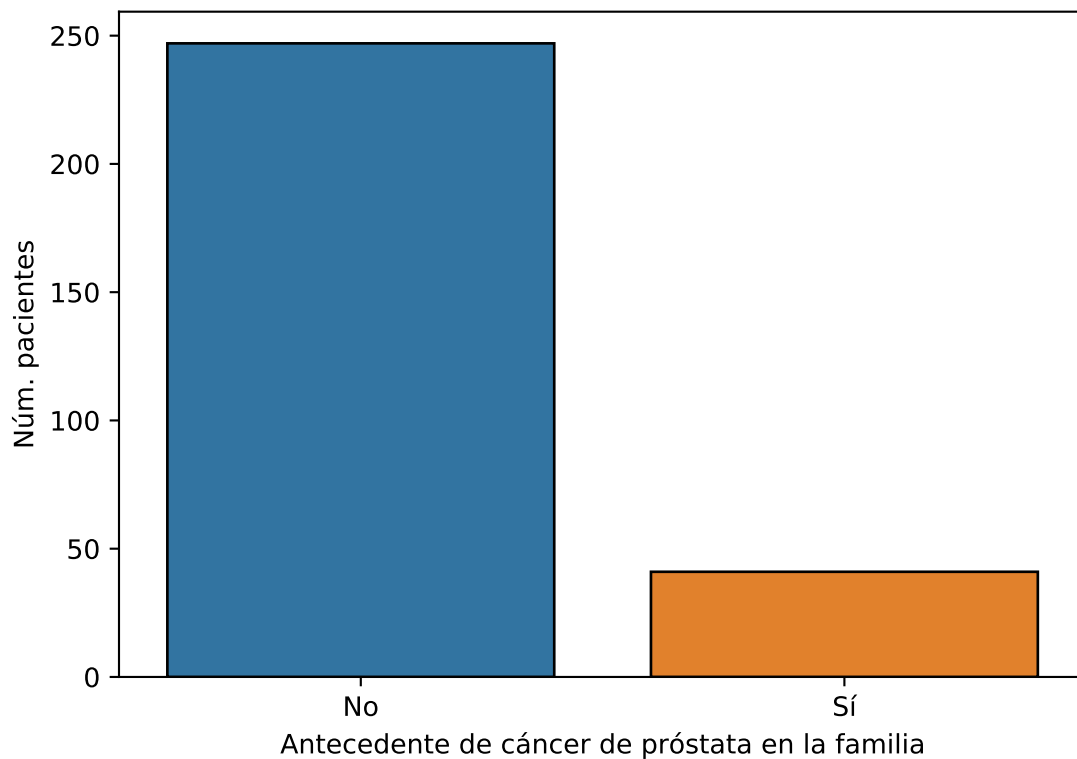
```
var = 'tumor_fam'  
df[var].value_counts()
```

```
## No      247  
## Sí       41  
## Name: tumor_fam, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.86  
## Sí      0.14  
## Name: tumor_fam, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Antecedente de cáncer de próstata en la familia')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



4.8 Familiar con cáncer de próstata

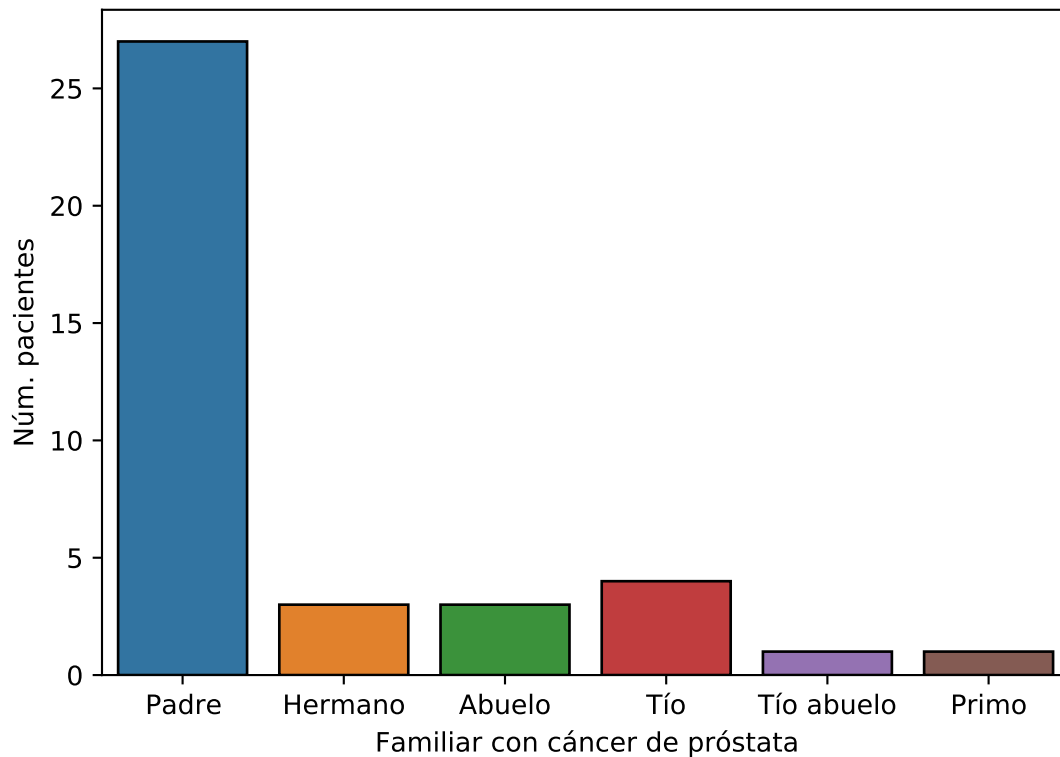
```
var = 'tumor_fam_tipo'
df[var].value_counts()
```

```
## Padre      27
## Tío        4
## Abuelo     3
## Hermano    3
## Primo      1
## Tío abuelo 1
## Name: tumor_fam_tipo, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## Padre      0.69
## Tío        0.10
## Abuelo     0.08
## Hermano    0.08
## Primo      0.03
## Tío abuelo 0.03
## Name: tumor_fam_tipo, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')
plt.xlabel('Familiar con cáncer de próstata')
plt.ylabel('Núm. pacientes')
plt.show()
```



5 Hábitos personales

5.1 Consumo de tabaco

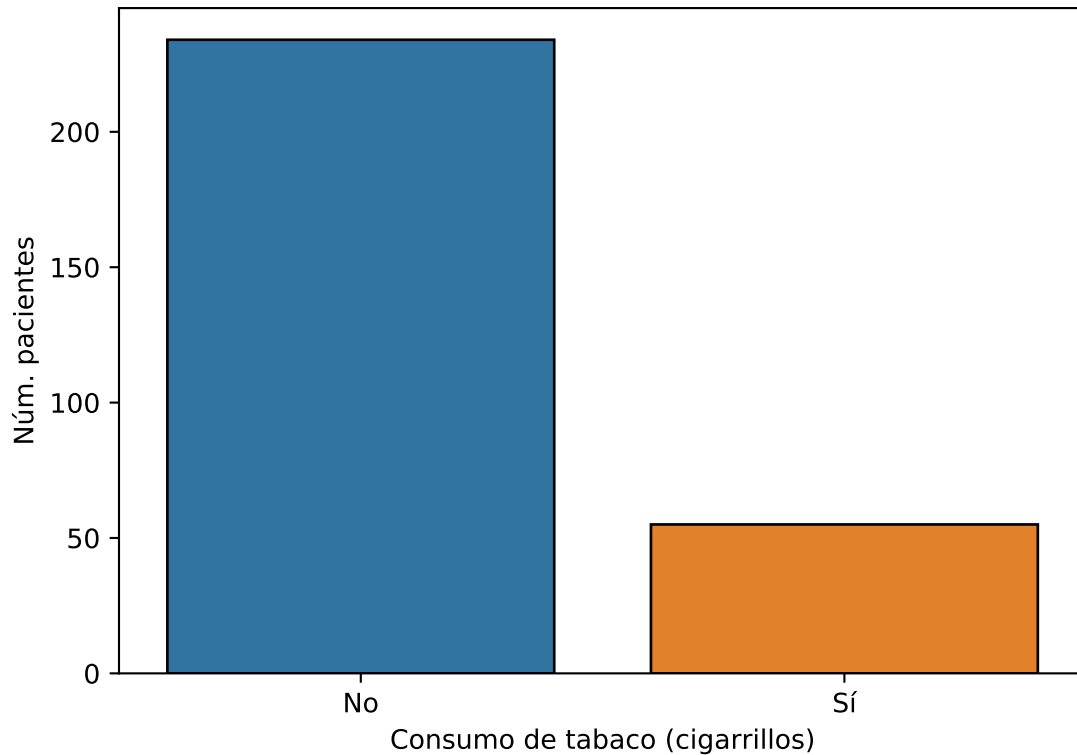
```
var = 'tabaco'
df[var].value_counts()
```

```
## No    234
## Sí     55
## Name: tabaco, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No    0.81
## Sí    0.19
## Name: tabaco, dtype: float64
```

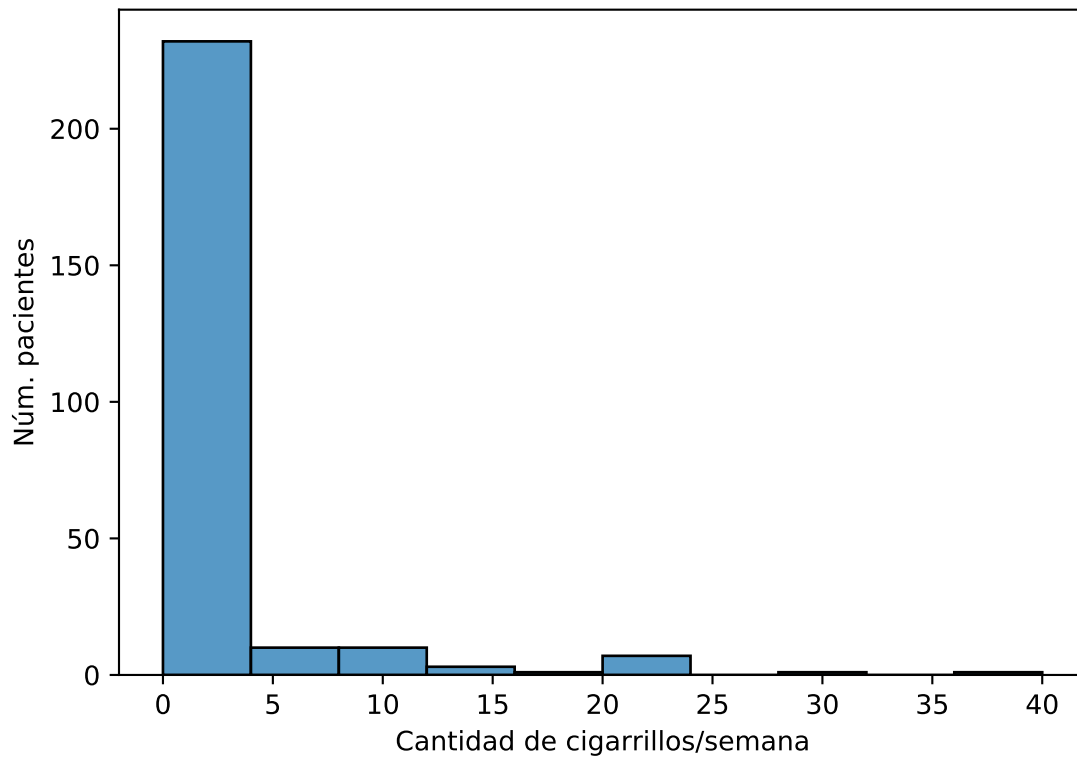
```
sns.countplot(x=var, data=df, edgecolor='black')
plt.xlabel('Consumo de tabaco (cigarrillos)')
plt.ylabel('Núm. pacientes')
plt.show()
```



```
var = 'tabaco_cantidad'
df[var].describe().round()
```

```
## count    265.0
## mean      2.0
## std       5.0
## min       0.0
## 25%       0.0
## 50%       0.0
## 75%       0.0
## max      40.0
## Name: tabaco_cantidad, dtype: float64
```

```
sns.histplot(x=var, data=df)
plt.xlabel('Cantidad de cigarrillos/semana')
plt.ylabel('Núm. pacientes')
plt.show()
```



5.2 Consumo de carne

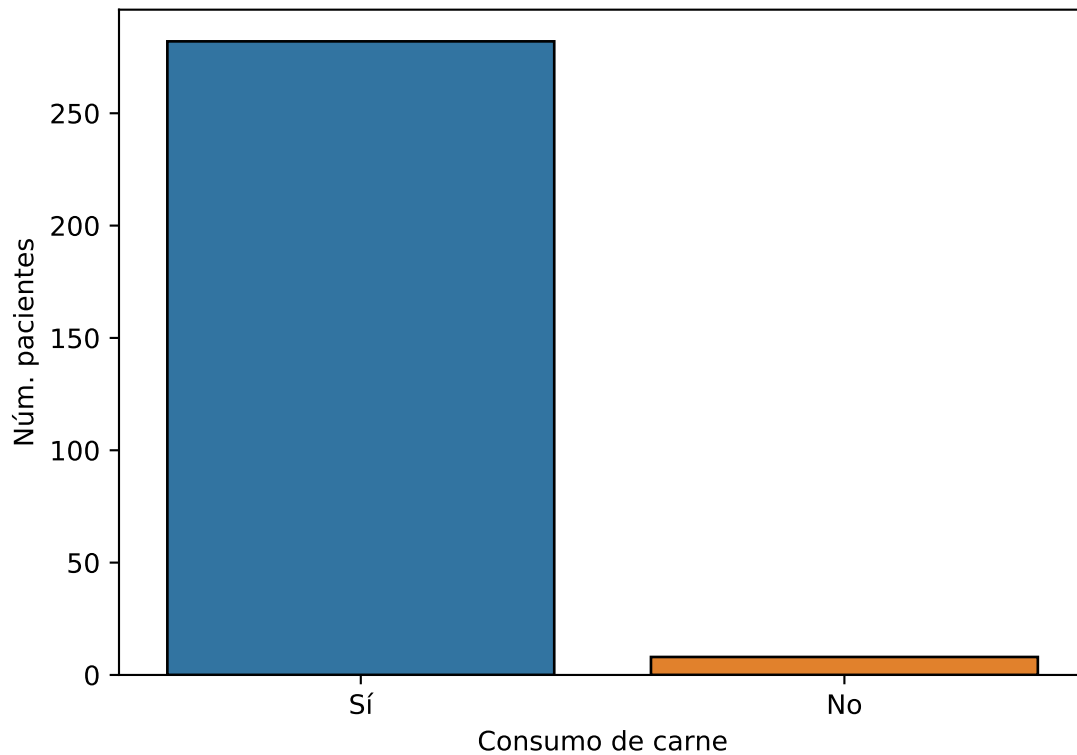
```
var = 'carne'
df[var].value_counts()
```

```
## Sí      282
## No        8
## Name: carne, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## Sí      0.97
## No      0.03
## Name: carne, dtype: float64
```

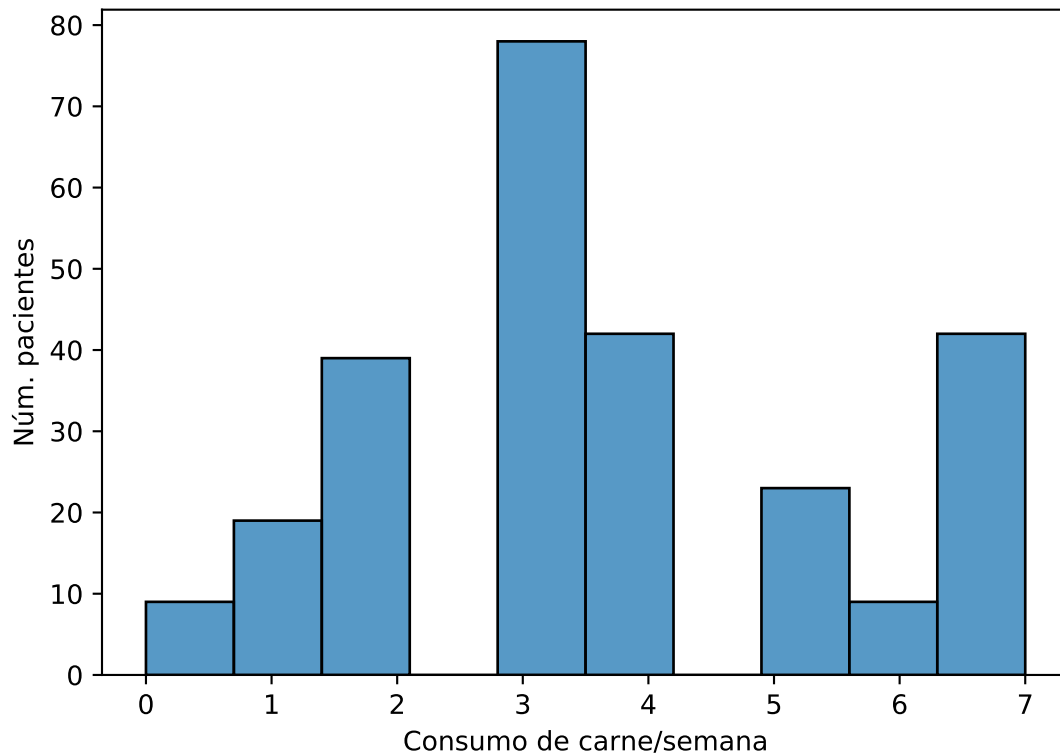
```
sns.countplot(x=var, data=df, edgecolor='black')
plt.xlabel('Consumo de carne')
plt.ylabel('Núm. pacientes')
plt.show()
```



```
var = 'carne_frec'  
df[var].describe().round()
```

```
## count    261.0  
## mean      4.0  
## std       2.0  
## min       0.0  
## 25%       2.0  
## 50%       3.0  
## 75%       5.0  
## max       7.0  
## Name: carne_frec, dtype: float64
```

```
sns.histplot(x=var, data=df)  
plt.xlabel('Consumo de carne/semana')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



5.3 Consumo de frutas/verduras

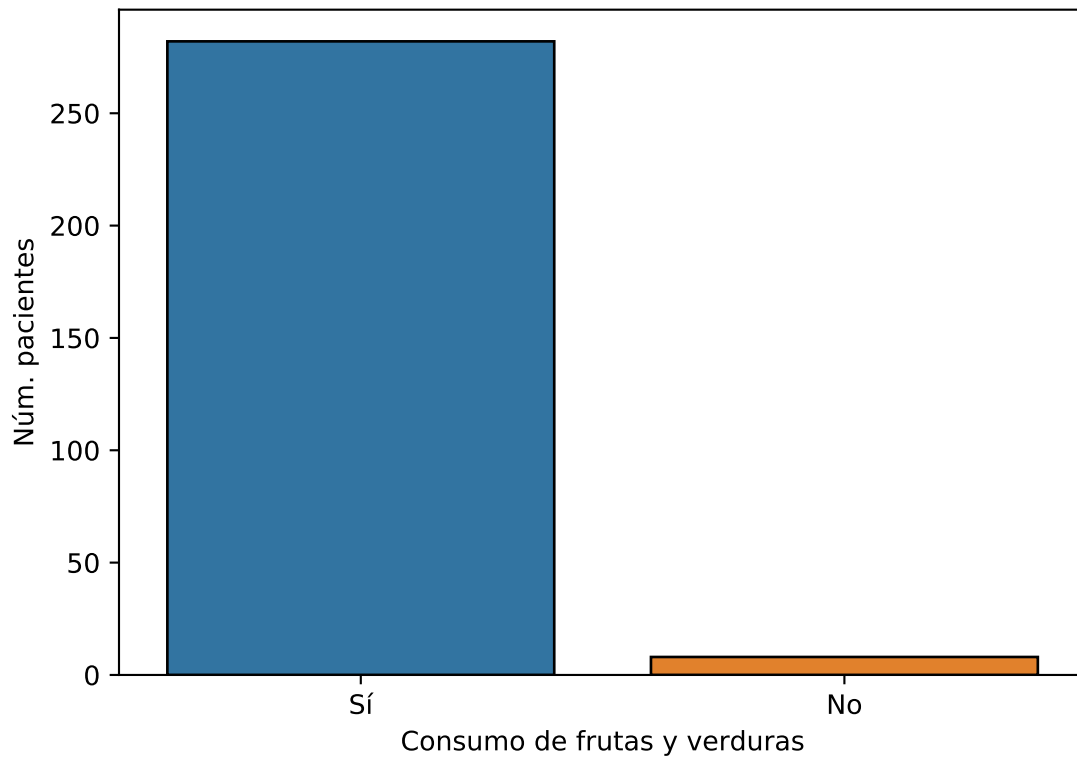
```
var = 'verdura'  
df[var].value_counts()
```

```
## Sí      282  
## No       8  
## Name: verdura, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## Sí      0.97  
## No      0.03  
## Name: verdura, dtype: float64
```

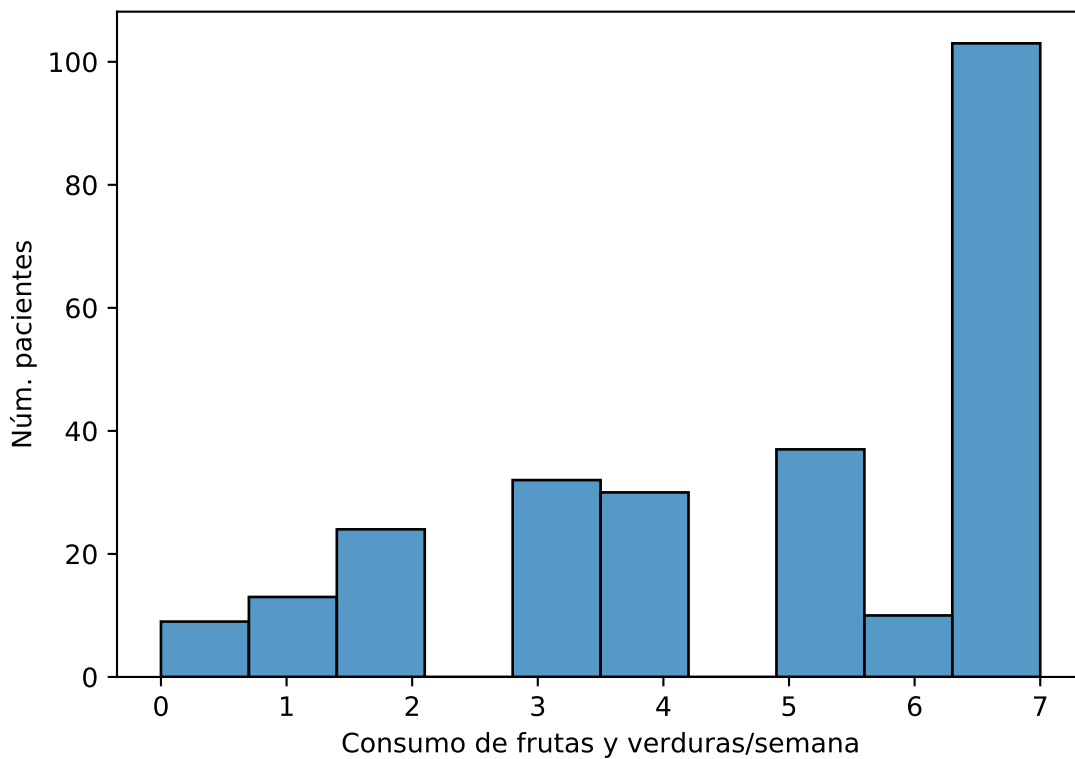
```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Consumo de frutas y verduras')  
plt.ylabel('Núm. pacientes')  
plt.show()
```

```
var = 'verdura_frec'  
df[var].describe().round()
```

```
## count    258.0  
## mean      5.0  
## std       2.0  
## min       0.0  
## 25%       3.0  
## 50%       5.0  
## 75%       7.0  
## max       7.0  
## Name: verdura_frec, dtype: float64
```

```
sns.histplot(x=var, data=df)  
plt.xlabel('Consumo de frutas y verduras/semana')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



5.4 Actividad física

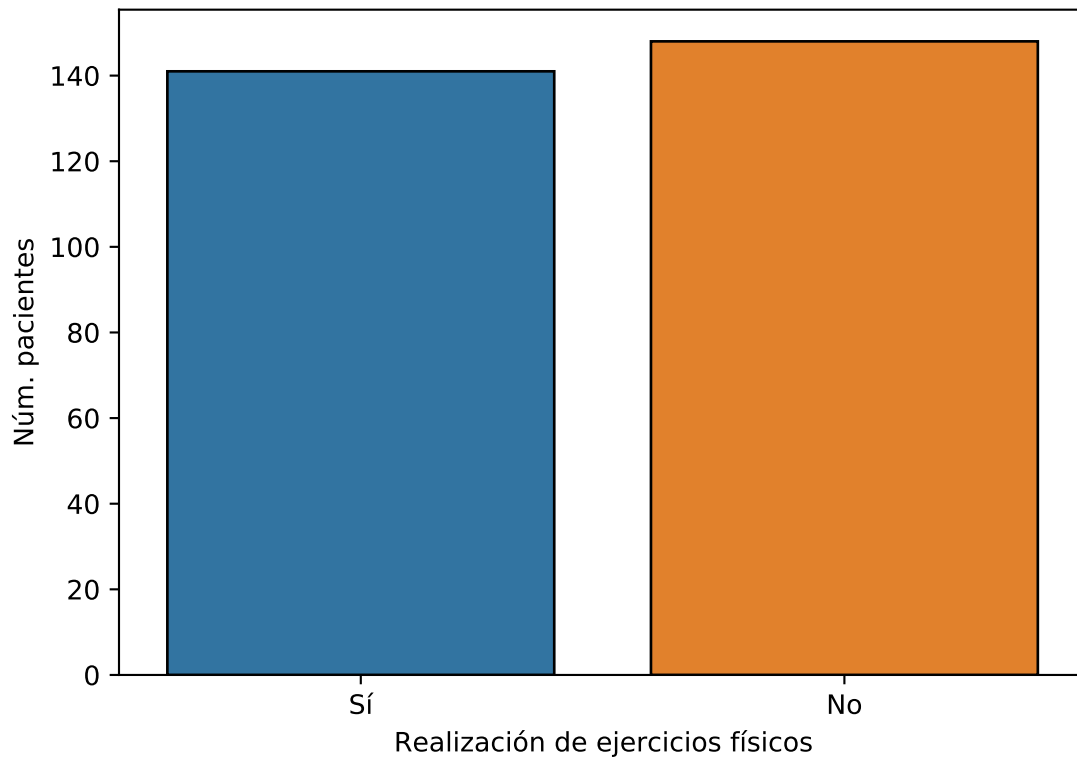
```
var = 'ejercicio'
df[var].value_counts()
```

```
## No      148
## Sí      141
## Name: ejercicio, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.51
## Sí      0.49
## Name: ejercicio, dtype: float64
```

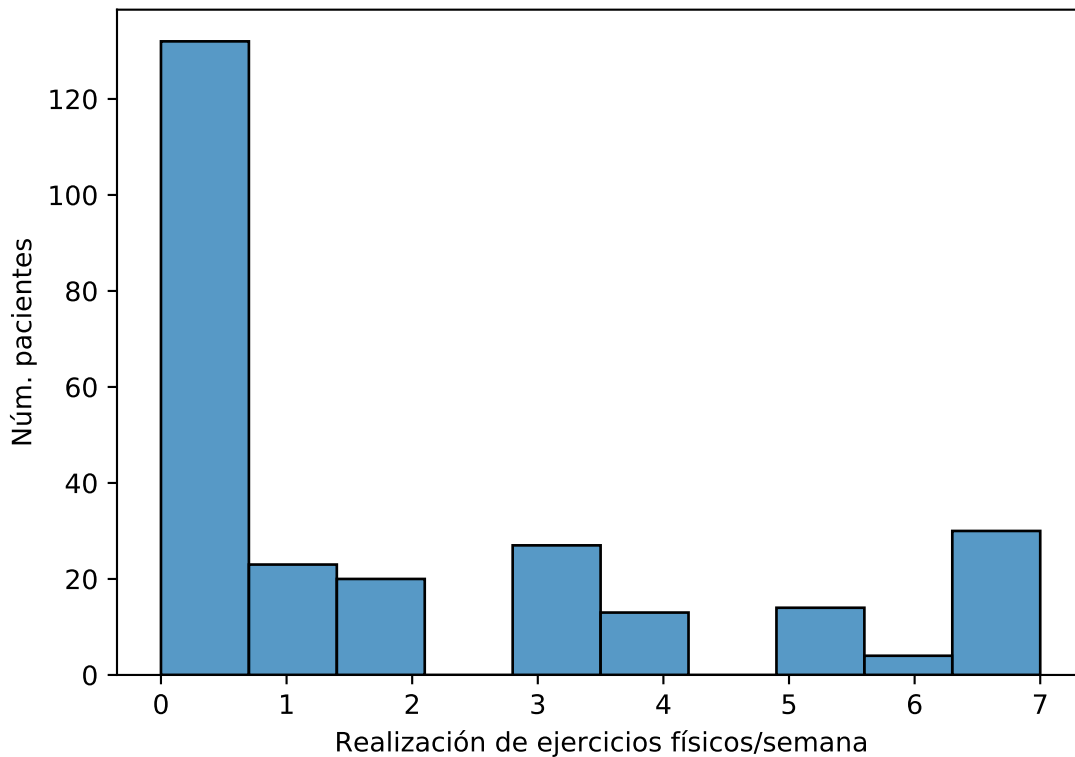
```
sns.countplot(x=var, data=df, edgecolor='black')
plt.xlabel('Realización de ejercicios físicos')
plt.ylabel('Núm. pacientes')
plt.show()
```



```
var = 'ejercicio_frec'  
df[var].describe().round()
```

```
## count    263.0  
## mean      2.0  
## std       2.0  
## min       0.0  
## 25%       0.0  
## 50%       0.0  
## 75%       3.0  
## max       7.0  
## Name: ejercicio_frec, dtype: float64
```

```
sns.histplot(x=var, data=df)  
plt.xlabel('Realización de ejercicios físicos/semana')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



6 Síntomas urinarios

6.1 Dificultad para empezar a orinar

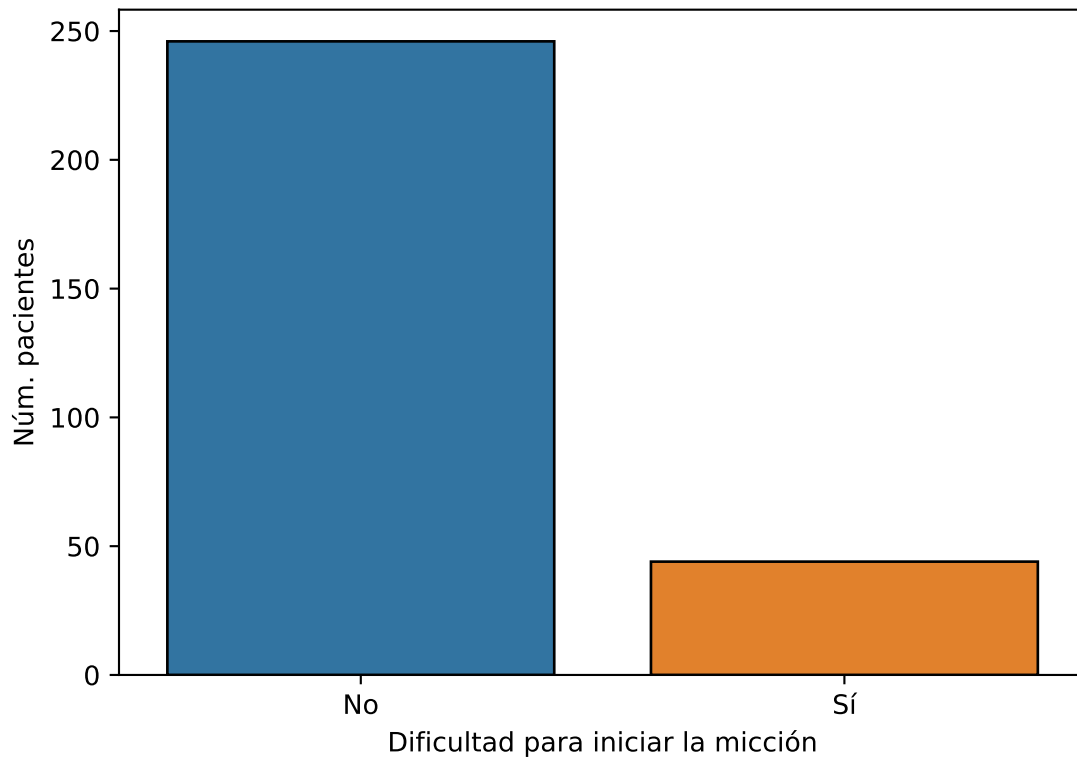
```
var = 'miccion_inicio'
df[var].value_counts()
```

```
## No      246
## Sí       44
## Name: miccion_inicio, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.85
## Sí      0.15
## Name: miccion_inicio, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')
plt.xlabel('Dificultad para iniciar la micción')
plt.ylabel('Núm. pacientes')
plt.show()
```



6.2 Chorro de orina lento o débil

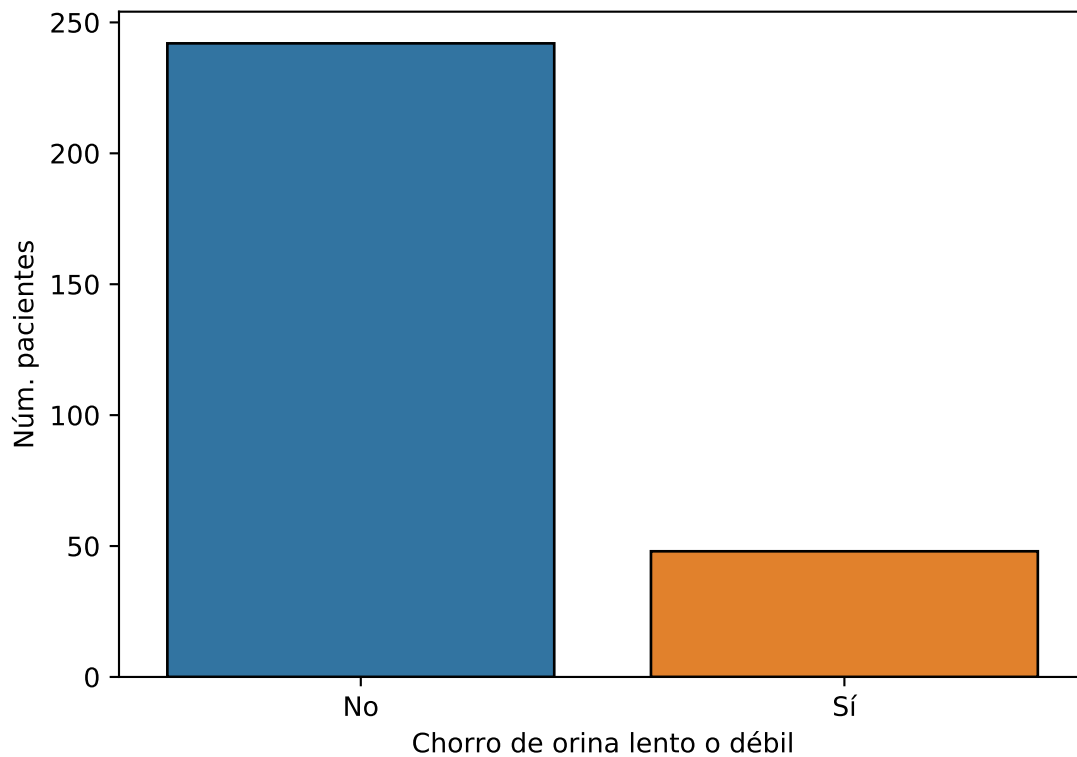
```
var = 'miccion_lenta'  
df[var].value_counts()
```

```
## No      242  
## Sí       48  
## Name: miccion_lenta, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.83  
## Sí      0.17  
## Name: miccion_lenta, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Chorro de orina lento o débil')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



6.3 Orinar frecuentemente (>10 veces/día)

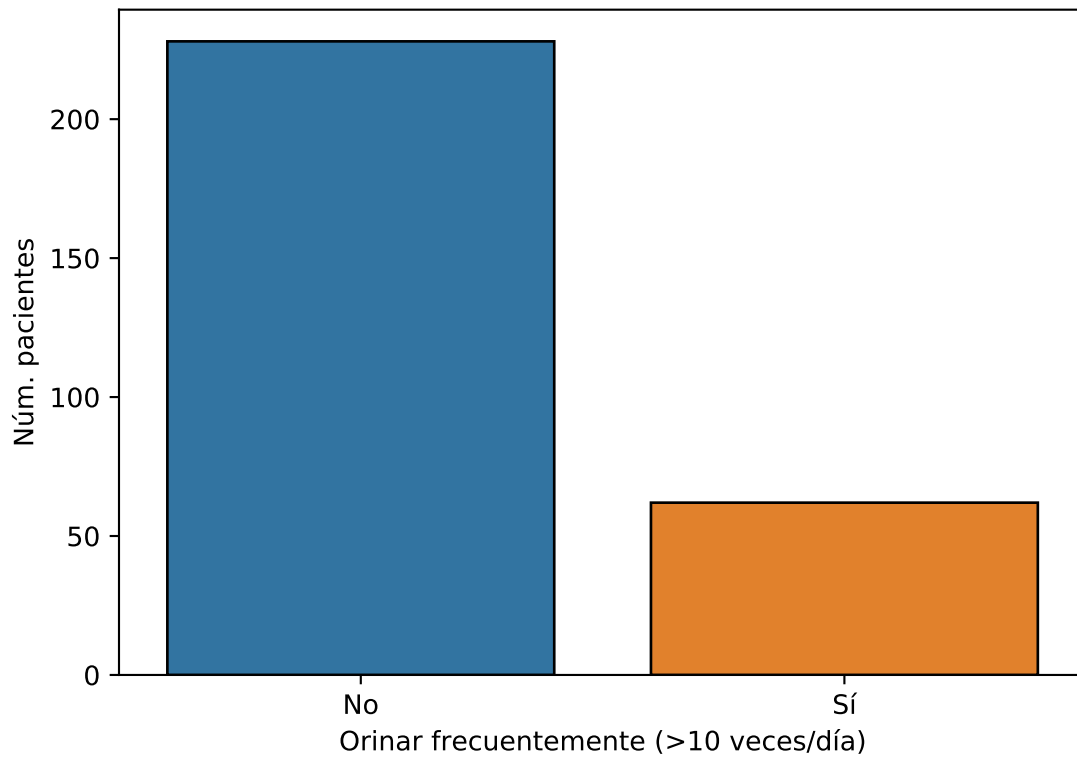
```
var = 'miccion_frec'
df[var].value_counts()
```

```
## No      228
## Sí       62
## Name: miccion_frec, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.79
## Sí      0.21
## Name: miccion_frec, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')
plt.xlabel('Orinar frecuentemente (>10 veces/día)')
plt.ylabel('Núm. pacientes')
plt.show()
```



6.4 Nicturia

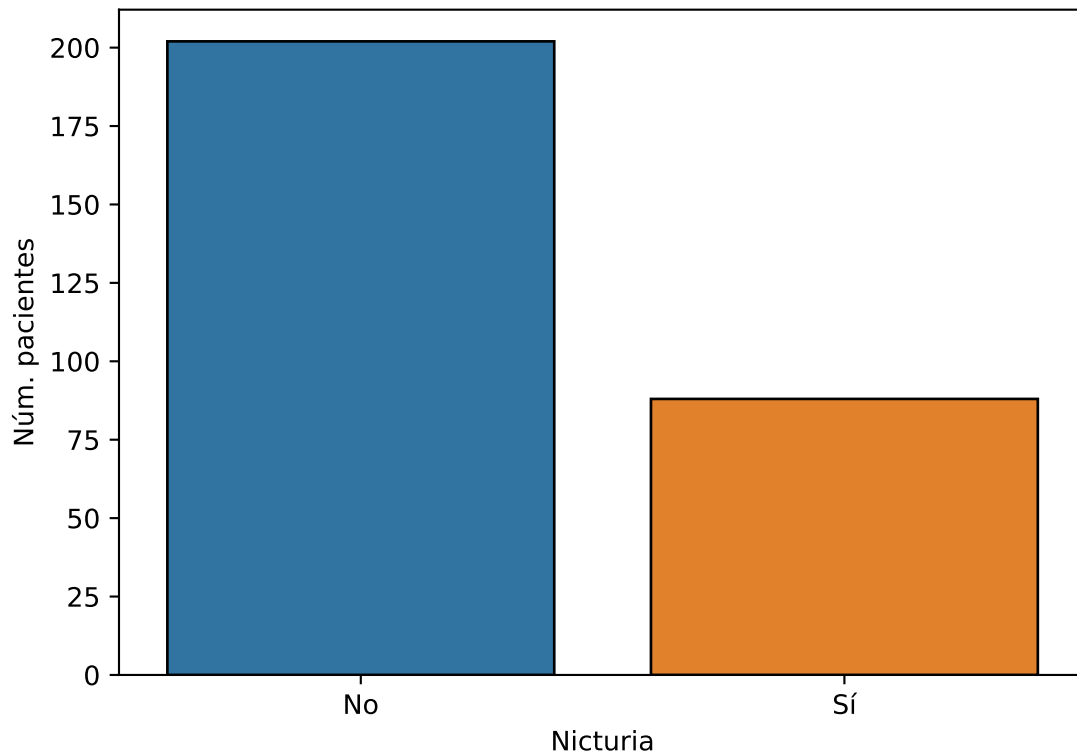
```
var = 'nicturia'  
df[var].value_counts()
```

```
## No    202  
## Sí     88  
## Name: nicturia, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No    0.7  
## Sí    0.3  
## Name: nicturia, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Nicturia')  
plt.ylabel('Núm. pacientes')  
plt.show()
```



6.5 Hematuria

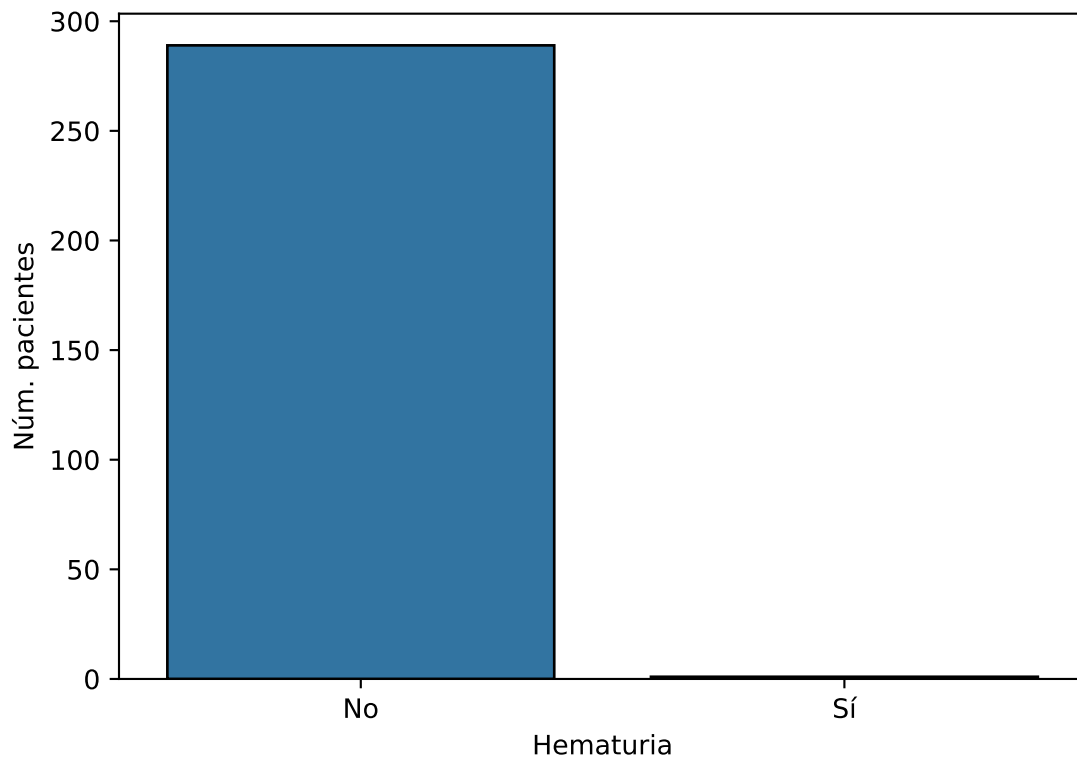
```
var = 'hematuria'  
df[var].value_counts()
```

```
## No      289  
## Sí       1  
## Name: hematuria, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      1.0  
## Sí      0.0  
## Name: hematuria, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Hematuria')  
plt.ylabel('Núm. pacientes')  
plt.show()
```

6.6 Dolor lumbar bajo

```
var = 'dolor_oseo'  
df[var].value_counts()
```

```
## No      232  
## Sí       58  
## Name: dolor_oseo, dtype: int64
```

```
df[var].value_counts(normalize=True).round(2)
```

```
## No      0.8  
## Sí      0.2  
## Name: dolor_oseo, dtype: float64
```

```
sns.countplot(x=var, data=df, edgecolor='black')  
plt.xlabel('Dolor en la región lumbar baja')  
plt.ylabel('Núm. pacientes')  
plt.show()
```

