

# RESULTADOS DE LOS ANÁLISIS LABORATORIALES

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

### Informe técnico de avance núm. 8

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. Los datos fueron analizados por el investigador principal del proyecto Prof. Dr. Alcides Chaux.

- **Fecha de cierre del periodo de análisis:** 10 de septiembre de 2019
- **Número de análisis laboratoriales a la fecha:** 447 pacientes

```
In [1]: import numpy as np
import pandas as pd

from matplotlib import pyplot as plt
import seaborn as sns
sns.set_theme(rc={'figure.figsize': [12,8]})
```

```
In [2]: df = pd.read_pickle('../Datos/ITA_8.pkl')
```

```
In [3]: df.shape
```

```
Out[3]: (447, 37)
```

## Antígeno prostático específico (PSA)

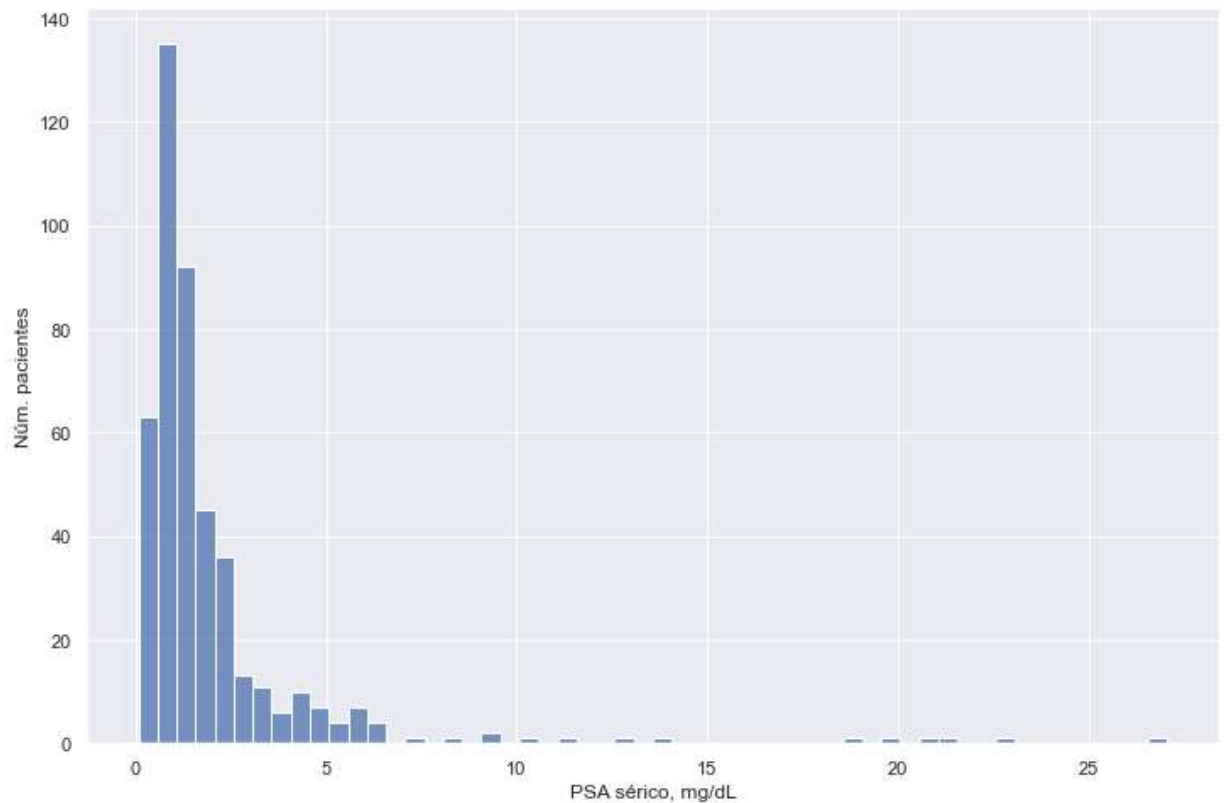
### Valor sérico

```
In [4]: df['psa_valor'].describe().round(2)
```

```
Out[4]: count    447.00
mean         2.00
std          2.91
min          0.07
25%          0.76
50%          1.19
75%          2.05
max          27.07
Name: psa_valor, dtype: float64
```

```
In [5]: sns.histplot(x='psa_valor', data=df, binwidth=0.5)
plt.xlabel('PSA sérico, mg/dL')
```

```
plt.ylabel('Núm. pacientes')  
plt.show()
```



## Categoría diagnóstica

```
In [6]: df['psa_categoria'] = df['psa_categoria'].astype('category')  
df['psa_categoria'].cat.reorder_categories(['En rango normal', 'Ligeramente elevado',  
df['psa_categoria'].value_counts(sort=False)
```

```
Out[6]: En rango normal      412  
Ligeramente elevado      25  
Marcadamente elevado     10  
Name: psa_categoria, dtype: int64
```

```
In [7]: df['psa_categoria'].value_counts(sort=False, normalize=True).round(2)
```

```
Out[7]: En rango normal      0.92  
Ligeramente elevado      0.06  
Marcadamente elevado     0.02  
Name: psa_categoria, dtype: float64
```

```
In [8]: sns.countplot(x='psa_categoria', data=df)  
plt.xlabel('Categoría diagnóstica del valor de PSA')  
plt.ylabel('Núm. pacientes')  
plt.show()
```

