

Electivo Profesional Salud Digital

Archivos, DataFrames y Visualización II

**Prof. Mauricio Cerda
Álvaro López Acevedo**

Programa de Biología Integrativa, I.C.B.M.,
Centro de Informática Médica y Telemedicina
Facultad de Medicina, Universidad de Chile



- Ya aprendiste lo que es un *Dataframe*
- Ahora extraigamos información de nuestra base de datos
- Métodos básicos para explorar



```
df.head()  
df.tail()  
df.info()  
df.describe()  
df.mean()
```

- Supongamos que deseas saber cuál es el valor mínimo de la variable numérica “Edad”

```
edad_minima = df['Edad'].min() #Entrega un valor
```



- Supongamos que quieres saber qué diagnósticos se repiten más

```
print(df['Diagnóstico'].value_counts())
```

Hipertensión	5
Diabetes Tipo 2	4
Colesterol alto	4

```
print(df['Diagnóstico'].value_counts().max())
```

- Supongamos que deseas saber cuál es el valor mínimo de la variable numérica “Edad”

```
edad_minima = df['Edad'].min() #Entrega un valor
```

- Supongamos que quieres saber qué diagnósticos se repiten más

```
print(df['Diagnóstico'].value_counts())
```

Hipertensión	5
Diabetes Tipo 2	4
Colesterol alto	4

```
print(df['Diagnóstico'].value_counts().max())
```



Pandas

- Supongamos que deseas saber cuál es la media de “Edad” de los pacientes que tienen Hipertensión

```
edad_mínima_HTA = df[df['Diagnóstico'] == 'Hipertensión']  
['Edad'].mean() #Entrega un valor
```



- Ahora quieres saber el porcentaje de los pacientes con Hipotiroidismo que pertenecen a FONASA

```
print(df[df['Diagnóstico'] == 'Hipotiroidismo'] ['Previsión']  
.value_counts(normalize=True).get('FONASA')*100)
```

50

Matplotlib

- Ahora aprendamos a visualizar

```
import matplotlib as plt
```

```
n_dgs = df['Diagnóstico'].value_counts() #Creamos el df con el n° de repeticiones
```

```
plt.figure(figsize=(10, 6)) # Definir el tamaño de la figura  
plt.barh(n_dgs.index, #Definir qué tipo de figura (barras), Define eje y  
n_dgs.values, #Define el eje x  
color='blue', edgecolor='black') # Color de barras y de bordes
```

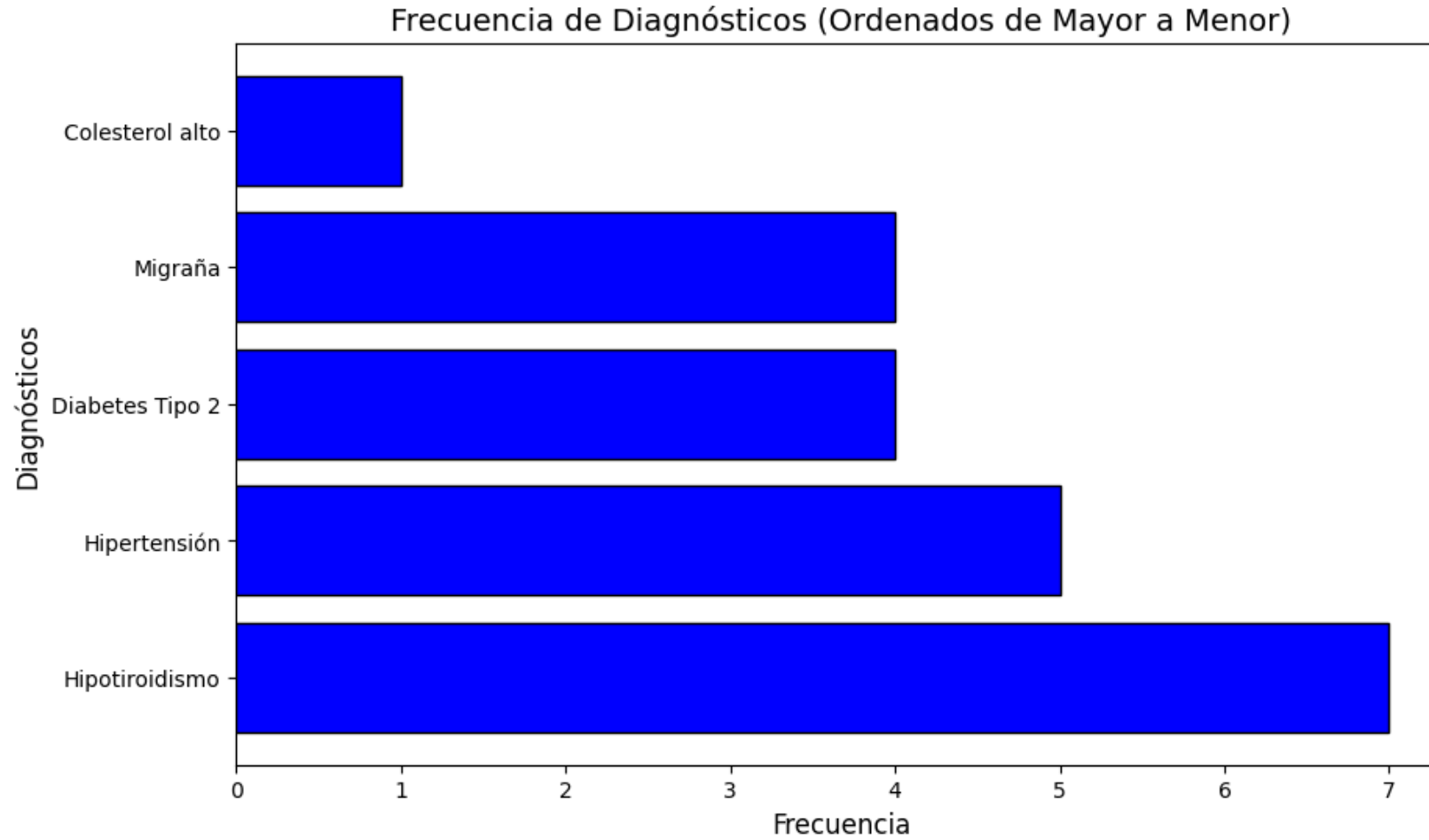
```
# Etiquetar el eje x con el texto "Frecuencia"  
plt.xlabel('Frecuencia', fontsize=12)
```

```
# Etiquetar el eje y con el texto "Diagnósticos"  
plt.ylabel('Diagnósticos', fontsize=12)
```

```
# Título del gráfico  
plt.title('Frecuencia de Diagnósticos (Ordenados de Mayor a Menor)', fontsize=14)
```

```
# Mostrar el gráfico  
plt.show()
```





- Supongamos ahora que quieres un gráfico lineal que muestre el número de consultas diarias de su servicio de urgencia

```
import datetime
df['FECHA_ING_URG'] = pd.to_datetime(df['FECHA_ING_URG'])

diarios =
df['FECHA_ING_URG'].groupby(df['FECHA_ING_URG'].dt.to_period('D')).count()
```

2019-06-01	84
2019-06-02	75
2019-06-03	109
...	

- Ahora aprendamos a visualizar

```
import matplotlib as plt
```

```
#Parámetros del nuevo plot, cuál será el índice, valores, tipo de línea
```

```
plt.plot(diarios.index.astype(str), diarios.values, marker='o', linestyle='-',  
color='b')
```



```
plt.title("Cantidad de Personas Atendidas Diariamente")
```

```
plt.xlabel("Fecha")
```

```
plt.ylabel("Cantidad de Personas")
```

```
plt.xticks(rotation=45)
```

```
plt.grid(visible=True, linestyle='--', alpha=0.7)
```

```
# Mostrar el gráfico
```

```
plt.show()
```

