

Trabajo autónomo 21



#SOLUCION ACTIVIDAD 01

```
def clasificar_grupo(edad):  
    if edad < 35:  
        return 'Joven'  
    elif 35 <= edad <= 59:  
        return 'Adulto'  
    else:  
        return 'Senior'  
  
df['Edad_str'] = df['Edad'].apply(clasificar_grupo)  
  
print(df)
```

```
***
```

	Paciente	Sexo	Edad	Mes	Tipo	Calorías	Proteínas	Edad_str
0	P007	F	40	10	Carnes	26546	568	Adulto
1	P012	M	72	11	Carnes	19688	314	Senior
2	P014	M	35	7	Verduras	28123	437	Adulto
3	P008	M	61	10	Cereales	13441	155	Senior
4	P012	F	40	11	Frutas	16103	535	Adulto
..
95	P013	F	66	2	Carnes	6636	334	Senior
96	P014	M	23	3	Verduras	16184	203	Joven
97	P010	F	22	1	Lácteos	11370	453	Joven
98	P003	M	78	5	Lácteos	7791	848	Senior
99	P011	F	64	2	Verduras	24695	456	Senior

```
[100 rows x 8 columns]
```



#SOLUCION ACTIVIDAD 2

```
filtro_senior = df['Edad_str'] == 'Senior'  
df_seniors = df[filtro_senior]  
  
promedio = df_seniors['Calorías'].mean()  
  
print(f"El promedio de calorías de los Seniors es: {promedio:.2f}")
```

```
*** El promedio de calorías de los Seniors es: 14988.03
```

```
# SOLUCION ACTIVIDAD 3
promedios = df.groupby('Tipo')['Calorías'].mean()

resultado = promedios.idxmax() # Queremos la etiqueta del máximo, más no el valor máximo (no .max())

print(f"El alimento con más calorías es: {resultado}")
```

El alimento con más calorías es: Verduras

```
▶ # SOLUCION ACTIVIDAD 4
condicion_tiempo = df['Mes'] <= 3
condicion_tipo = df['Tipo'] == 'Cereales'
condicion_sexo = df['Sexo'] == 'F'

df_filtrado = df[condicion_tiempo & condicion_tipo & condicion_sexo]

total = len(df_filtrado)

print(f"El total de registros encontrados es: {total}")
```

... El total de registros encontrados es: 5

```
▶ # SOLUCION ACTIVIDAD 5
pacientes = df.groupby('Paciente')['Calorías'].sum().reset_index()

pacientes = pacientes.sort_values('Calorías', ascending=False)

pacientes.to_csv('calorias_paciente.csv', index=False)
```

```
...
  Paciente  Calorías
13    P014    195663
0     P001    182752
5     P006    163741
10    P011    147160
11    P012    141996
7     P008    132156
6     P007    126201
1     P002     98367
9     P010     90504
14    P015     73299
8     P009     66916
2     P003     58757
3     P004     35597
4     P005     34070
12    P013     25493
```