EXAMEN PRACTICO UNISAD UF1468

Carlos Pamias Mora

El código de Python generado es:

Importamos la biblioteca pandas para poder realizar el ejercicio y cargamos los datos del ejercicio:

import pandas as pd

df = pd.read_csv('empleados.csv',delimiter=',')

Obtenemos los 5 primeros registros

df.head()

7	Código_empleado	Nombre	Apellidos	Dirección	Teléfono	Edad	Casado	Fecha_nacimiento	Salario	Comentarios	E
0	1	Juan	García	Calle Falsa 123	123 456 789	28	No	1995-05-12	1500.50	Ninguno	
1	2	Ana	López	Av. Siempre Viva 742	987 654 321	35	Sí	1988-08-23	2500.00	NaN	
2	3	Pedro	Rodríguez	Plaza Mayor 5	555 555 555	40	Sí	1983-02-14	3000.75	Promoción reciente	
3	4	María	Pérez	Calle Luna 9	444 444 444	22	No	2001-11-01	1200.00	Nuevo empleado	
4	5	Luis	Fernández	Calle Sol 15	333 333 333	31	Sí	1992-03-09	2800.20	Trabaja medio tiempo	

Analisis de datos

df.info() # Informacion general del DataFame

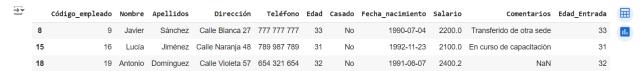
₹	Range	<pre><class 'pandas.core.frame.dataframe'=""> RangeIndex: 20 entries, 0 to 19 Data columns (total 10 columns): # Column Non-Null Count Dtype</class></pre>									
	0	Código empleado	20 non-null	int64							
	1	Nombre	20 non-null	object							
	2	Apellidos	20 non-null	object							
	3	Dirección	20 non-null	object							
	4	Teléfono	20 non-null	object							
	5	Edad	20 non-null	int64							
	6	Casado	20 non-null	object							
	7	Fecha_nacimiento	20 non-null	object							
	8	Salario	20 non-null	float64							
	9	Comentarios	9 non-null	object							
	<pre>dtypes: float64(1), int64(2), object(7)</pre>										
	memo	ry usage: 1.7+ KB									

df.describe() # Muesta las extadisticas descriptivas de las columnas numericas

	Código_empleado	Edad	Salario	Ħ
count	20.00000	20.000000	20.000000	11.
mean	10.50000	33.750000	2430.237500	
std	5.91608	6.025167	645.947757	
min	1.00000	22.000000	1200.000000	
25%	5.75000	29.000000	1887.550000	
50%	10.50000	33.500000	2550.250000	
75%	15.25000	38.250000	2912.687500	
max	20.00000	45.000000	3500.000000	

"""Filtrar datos"""

Se muestran los empleados que tienen ,as de 30 años y no estan casados df[(df['Edad']>30) & (df['Casado']=='No')]



Se muestran los empleados cuyo nombre comienza con A df[df['Nombre'].str.startswith('A')]

	Código_empleado	Nombre	Apellidos	Dirección	Teléfono	Edad	Casado	Fecha_nacimiento	Salario	Comentarios	Edad_Entrada	H
1	2	Ana	López	Av. Siempre Viva 742	987 654 321	35	Sí	1988-08-23	2500.0	NaN	35	11.
14	15	Andrés	Romero	Calle Gris 45	654 456 654	34	Sí	1989-05-05	2600.5	NaN	34	
18	19	Antonio	Domínguez	Calle Violeta 57	654 321 654	32	No	1991-06-07	2400.2	NaN	32	

"""Creacion de nuevas columnas"""

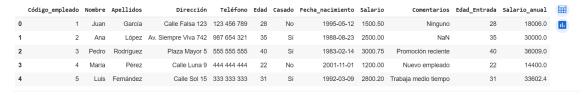
Convertir la columna de fecha de nacimiento a formato datetime df['Fecha_nacimiento'] = pd.to_datetime(df['Fecha_nacimiento'])

Definir el 1 de enero de 2023 como referencia Fecha_empresa = pd.Timestamp('2023-01-01')

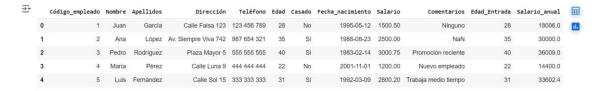
Calcular la edad (diferencia en años) al inicio de 2023

df['Edad_Entrada'] = Fecha_empresa.year - df['Fecha_nacimiento'].dt.year

df.head()



df['Salario_anual'] = df['Salario'] * 12
df.head()



```
"""Agrupacion de datos"""
```

Agrupar los empleados por tipo

df_agrupado = df.groupby('Casado')['Salario'].mean()
df_agrupado

Casado

No	1827.933333					
Sí	2923.031818					

Agrupar los empleados por edad y calcular

df_agrupado = df.groupby('Edad')['Salario'].sum()
df_agrupado

Salario

Edad	
22	1200.00
26	1600.50
27	1700.00
28	1500.50
29	3750.20
31	4900.20
32	2400.20
33	2200.00
34	2600.50
35	2500.00
36	2800.75
37	2700.40
38	2950.75
39	2900.00
40	3000.75
41	3300.00

Exportar DataFrame a un archivo CSV df.to_csv('empleados_modificados.csv')