CARRERA: Computación
ESTUDIANTE: Christian Ronaldo Mocha

PRUEBA PRÁCTICA

ASIGNATURA: Simulación
TÍTULO: Prueba Práctica Unidad 1

ACTIVIDADES DESARROLLADAS

In [ ]:
```
!pip install datapane
!pip install bokeh
```

```
Requirement already satisfied: datapane in /usr/local/lib/python3.
7/dist-packages (0.12.3)
Requirement already satisfied: jsonschema<4.0.0,>=3.0.0 in /usr/lo
cal/lib/python3.7/dist-packages (from datapane) (3.2.0)
Requirement already satisfied: pandas<2.0.0,>=1.1.0 in /usr/local/
lib/python3.7/dist-packages (from datapane) (1.1.5)
Requirement already satisfied: munch<3.0.0,>=2.3.0 in /usr/local/l
ib/python3.7/dist-packages (from datapane) (2.5.0)
Requirement already satisfied: furl<3.0.0,>=2.0.0 in /usr/local/li
b/python3.7/dist-packages (from datapane) (2.1.3)
Requirement already satisfied: altair<5.0.0,>=4.0.0 in /usr/local/
lib/python3.7/dist-packages (from datapane) (4.1.0)
Requirement already satisfied: importlib_resources<6.0.0,>=3.0.0 i
n /usr/local/lib/python3.7/dist-packages (from datapane) (5.4.0)
Requirement already satisfied: glom<21.0.0,>=20.5.0 in /usr/local/
lib/python3.7/dist-packages (from datapane) (20.11.0)
Requirement already satisfied: datacommons<2.0.0,>=1.4.3 in /usr/l
ocal/lib/python3.7/dist-packages (from datapane) (1.4.3)
Requirement already satisfied: packaging<22.0.0,>=20.0.0 in /usr/l
ocal/lib/python3.7/dist-packages (from datapane) (21.3)
Requirement already satisfied: requests-toolbelt<0.10.0,>=0.9.1 in
/usr/local/lib/python3.7/dist-packages (from datapane) (0.9.1)
Requirement already satisfied: nbconvert<7.0.0,>=5.6.1 in /usr/loc
al/lib/python3.7/dist-packages (from datapane) (5.6.1)
Requirement already satisfied: click<9.0.0,>=7.1.0 in /usr/local/l
ib/python3.7/dist-packages (from datapane) (7.1.2)
Requirement already satisfied: colorlog<7.0.0,>=4.1.0 in /usr/loca
l/lib/python3.7/dist-packages (from datapane) (6.6.0)
Requirement already satisfied: click-spinner<0.2.0,>=0.1.8 in /usr
/local/lib/python3.7/dist-packages (from datapane) (0.1.10)
Requirement already satisfied: posthog<2.0.0,>=1.4.0 in /usr/local
/lib/python3.7/dist-packages (from datapane) (1.4.4)
Requirement already satisfied: requests<3.0.0,>=2.19.0 in /usr/loc
al/lib/python3.7/dist-packages (from datapane) (2.23.0)
```

```
Requirement already satisfied: chardet<5.0.0,>=3.0.4 in /usr/local
/lib/python3.7/dist-packages (from datapane) (3.0.4)
Requirement already satisfied: tabulate<0.9.0,>=0.8.0 in /usr/loca
l/lib/python3.7/dist-packages (from datapane) (0.8.9)
Requirement already satisfied: stringcase<2.0.0,>=1.2.0 in /usr/lo
cal/lib/python3.7/dist-packages (from datapane) (1.2.0)
Requirement already satisfied: vega-datasets<1.0.0,>=0.9.0 in /usr
/local/lib/python3.7/dist-packages (from datapane) (0.9.0)
Requirement already satisfied: PyYAML<6.0.0,>=5.1.0 in /usr/local/
lib/python3.7/dist-packages (from datapane) (5.4.1)
Requirement already satisfied: toolz<0.12.0,>=0.11.0 in /usr/local
/lib/python3.7/dist-packages (from datapane) (0.11.2)
Requirement already satisfied: datacommons-pandas<0.0.4,>=0.0.3 in
/usr/local/lib/python3.7/dist-packages (from datapane) (0.0.3)
Requirement already satisfied: pydantic<2.0.0,>=1.6.0 in /usr/loca
l/lib/python3.7/dist-packages (from datapane) (1.8.2)
Requirement already satisfied: Jinja2<4.0.0,>=2.11.0 in /usr/local
/lib/python3.7/dist-packages (from datapane) (2.11.3)
Requirement already satisfied: micawber>=0.5.0 in /usr/local/lib/p
ython3.7/dist-packages (from datapane) (0.5.4)
Requirement already satisfied: dacite<2.0.0,>=1.0.2 in /usr/local/
lib/python3.7/dist-packages (from datapane) (1.6.0)
Requirement already satisfied: lxml<5.0.0,>=4.0.0 in /usr/local/li
b/python3.7/dist-packages (from datapane) (4.2.6)
Requirement already satisfied: pyarrow<6.0.0,>=3.0.0 in /usr/local
/lib/python3.7/dist-packages (from datapane) (3.0.0)
Requirement already satisfied: boltons<22.0.0,>=20.0.0 in /usr/loc
al/lib/python3.7/dist-packages (from datapane) (21.0.0)
Requirement already satisfied: validators<0.19.0,>=0.18.0 in /usr/
local/lib/python3.7/dist-packages (from datapane) (0.18.2)
Requirement already satisfied: dominate<3.0.0,>=2.4.0 in /usr/loca
l/lib/python3.7/dist-packages (from datapane) (2.6.0)
Requirement already satisfied: entrypoints in /usr/local/lib/pytho
n3.7/dist-packages (from altair<5.0.0,>=4.0.0->datapane) (0.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/d
ist-packages (from altair<5.0.0,>=4.0.0->datapane) (1.19.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dis
t-packages (from datacommons<2.0.0,>=1.4.3->datapane) (1.15.0)
Requirement already satisfied: orderedmultidict>=1.0.1 in /usr/loc
al/lib/python3.7/dist-packages (from furl<3.0.0,>=2.0.0->datapane)
(1.0.1)
Requirement already satisfied: face>=20.1.0 in /usr/local/lib/pyth
on3.7/dist-packages (from glom<21.0.0,>=20.5.0->datapane) (20.1.1)
Requirement already satisfied: attrs in /usr/local/lib/python3.7/d
ist-packages (from glom<21.0.0,>=20.5.0->datapane) (21.2.0)
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/pytho
n3.7/dist-packages (from importlib_resources<6.0.0,>=3.0.0->datapa
ne) (3.6.0)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/
python3.7/dist-packages (from Jinja2<4.0.0,>=2.11.0->datapane) (2.
0.1)
Requirement already satisfied: importlib-metadata in /usr/local/li
b/python3.7/dist-packages (from jsonschema<4.0.0,>=3.0.0->datapane
```

```
) (4.8.2)
Requirement already satisfied: pyrsistent>=0.14.0 in /usr/local/li
b/python3.7/dist-packages (from jsonschema<4.0.0,>=3.0.0->datapane
) (0.18.0)
Requirement already satisfied: setuptools in /usr/local/lib/python
3.7/dist-packages (from jsonschema<4.0.0,>=3.0.0->datapane) (57.4.
0)
Requirement already satisfied: bleach in /usr/local/lib/python3.7/
dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane) (4.1.0)
Requirement already satisfied: testpath in /usr/local/lib/python3.
7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane) (0.5.0)
Requirement already satisfied: pygments in /usr/local/lib/python3.
7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane) (2.6.1)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/
lib/python3.7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapan
e) (1.5.0)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib
/python3.7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane)
(0.8.4)
Requirement already satisfied: defusedxml in /usr/local/lib/python
3.7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane) (0.7.1)
Requirement already satisfied: jupyter-core in /usr/local/lib/pyth
on3.7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane) (4.9.
1)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/py
thon3.7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane) (5.
1.1)
Requirement already satisfied: nbformat>=4.4 in /usr/local/lib/pyt
hon3.7/dist-packages (from nbconvert<7.0.0,>=5.6.1->datapane) (5.1
.3)
Requirement already satisfied: ipython-genutils in /usr/local/lib/
python3.7/dist-packages (from nbformat>=4.4->nbconvert<7.0.0,>=5.6
.1->datapane) (0.2.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/lo
cal/lib/python3.7/dist-packages (from packaging<22.0.0,>=20.0.0->d
atapane) (3.0.6)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/loca
l/lib/python3.7/dist-packages (from pandas<2.0.0,>=1.1.0->datapane
) (2.8.2)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/pyth
on3.7/dist-packages (from pandas<2.0.0,>=1.1.0->datapane) (2018.9)
Requirement already satisfied: backoff<2.0.0,>=1.10.0 in /usr/loca
l/lib/python3.7/dist-packages (from posthog<2.0.0,>=1.4.0->datapan
e) (1.11.1)
Requirement already satisfied: monotonic>=1.5 in /usr/local/lib/py
thon3.7/dist-packages (from posthog<2.0.0,>=1.4.0->datapane) (1.6)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/
local/lib/python3.7/dist-packages (from pydantic<2.0.0,>=1.6.0->da
tapane) (3.10.0.2)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/pyth
on3.7/dist-packages (from requests<3.0.0,>=2.19.0->datapane) (2.10
)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.
```

```
21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3.0.
0,>=2.19.0->datapane) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/li
b/python3.7/dist-packages (from requests<3.0.0,>=2.19.0->datapane)
(2021.10.8)
Requirement already satisfied: decorator>=3.4.0 in /usr/local/lib/
python3.7/dist-packages (from validators<0.19.0,>=0.18.0->datapane
) (4.4.2)
Requirement already satisfied: webencodings in /usr/local/lib/pyth
on3.7/dist-packages (from bleach->nbconvert<7.0.0,>=5.6.1->datapan
e) (0.5.1)
Requirement already satisfied: bokeh in /usr/local/lib/python3.7/d
ist-packages (2.3.3)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/pyth
on3.7/dist-packages (from bokeh) (5.4.1)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/pytho
n3.7/dist-packages (from bokeh) (2.11.3)
Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/pyt
hon3.7/dist-packages (from bokeh) (1.19.5)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/
lib/python3.7/dist-packages (from bokeh) (2.8.2)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/pyth
on3.7/dist-packages (from bokeh) (5.1.1)
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/p
ython3.7/dist-packages (from bokeh) (21.3)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from bokeh) (7.1.2)
Requirement already satisfied: typing-extensions>=3.7.4 in /usr/lo
cal/lib/python3.7/dist-packages (from bokeh) (3.10.0.2)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/
python3.7/dist-packages (from Jinja2>=2.9->bokeh) (2.0.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/lo
cal/lib/python3.7/dist-packages (from packaging>=16.8->bokeh) (3.0
.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.
7/dist-packages (from python-dateutil>=2.1->bokeh) (1.15.0)
```

Importanción de librerias

In [ ]:
```
import pandas as pd
import numpy as np
from datetime import datetime
from matplotlib import pyplot as plt
import matplotlib.patches as mpatches
from datetime import datetime,timedelta
from sklearn.model_selection import train_test_split as tts
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
import plotly.express as plotly
import datapane as dp
```

• Diseñe y desarrolle un modelo y/o script que permita describir el siguiente caso aplicado:
◦ Se tiene los datos del ecuador (https://www.ecuadorencifras.gob.ec/documentos/web-inec/EMPLEO/2021/Octubre-2021/202110_Tabulados_Mercado_Laboral_CSV.zip (https://www.ecuadorencifras.gob.ec/documentos/web-inec/EMPLEO/2021/Octubre-2021/202110_Tabulados_Mercado_Laboral_CSV.zip)): En base a ello obtener lo siguiente:
▪ Generar graficas para entender y procesar los datos:
• Generar graficas y reportes del total de personas empleadas y desempleadas por año.

Leer el dataSet

In [ ]:
```
poblacion = pd.read_csv('/content/drive/MyDrive/decimo ciclo/qusis/Prueba1/1.Poblaciones.csv',encoding='latin1', sep=';',skiprows=2)
poblacion = poblacion.drop(poblacion.columns[[8,9,10,11,12,13,14,15,16,17,18]],axis=1)
poblacion.rename(columns ={'Unnamed: 0':'Encuesta','Unnamed: 1':'Periodo','Unnamed: 2':'Indicadores'}, inplace = True)
poblacion
```

Out[ ]:

	Encuesta	Periodo	Indicadores	Total	Urbana	Rural	Hombre	M
0	ENEMDU	dic-07	Población Total	13.682.302	9.066.209	4.616.093	6.768.646	6.913
1	ENEMDU	dic-07	Población menor de 15 años	4.372.812	2.723.124	1.649.688	2.226.618	2.146
2	ENEMDU	dic-07	Población en Edad de Trabajar (PET)	9.309.490	6.343.085	2.966.404	4.542.028	4.767
3	ENEMDU	dic-07	Población Económicamente Activa	6.336.029	4.227.702	2.108.328	3.777.232	2.558
4	ENEMDU	dic-07	Empleo	6.019.332	3.971.040	2.048.292	3.632.314	2.387
...	...	...	...	...	...	...	...	
895	ENEMDU*	oct-21	Desempleo Abierto	336.101	291.606	44.495	150.276	185
896	ENEMDU*	oct-21	Desempleo Oculto	48.103	27.600	20.503	25.975	22
897	ENEMDU*	oct-21	Desempleo Cesante	298.846	257.856	40.991	140.223	158
898	ENEMDU*	oct-21	Desempleo Nuevo	85.358	61.351	24.007	36.028	49
899	ENEMDU*	oct-21	Población Económicamente Inactiva	4.330.241	3.307.420	1.022.821	1.324.745	3.005

900 rows × 8 columns

Eliminar espacios

In [ ]:
```
def eliminarPuntos(x):
  return int(x.replace(".",""))
```

In [ ]:
```
poblacion["Total"] =  poblacion["Total"].apply(eliminarPuntos)
```

In [ ]:
```
def anio(x):
  nu= x.split("-")
  return nu[1]
```

In [ ]:
```
poblacion["Periodo"] = poblacion["Periodo"].apply(anio)
poblacion
```

Out[ ]:

	Encuesta	Periodo	Indicadores	Total	Urbana	Rural	Hombre	Mu
0	ENEMDU	07	Población Total	13682302	9.066.209	4.616.093	6.768.646	6.913.(
1	ENEMDU	07	Población menor de 15 años	4372812	2.723.124	1.649.688	2.226.618	2.146.
2	ENEMDU	07	Población en Edad de Trabajar (PET)	9309490	6.343.085	2.966.404	4.542.028	4.767.(
3	ENEMDU	07	Población Económicamente Activa	6336029	4.227.702	2.108.328	3.777.232	2.558.7
4	ENEMDU	07	Empleo	6019332	3.971.040	2.048.292	3.632.314	2.387.(
...	...	...	...	...	...	...	...	
895	ENEMDU*	21	Desempleo Abierto	336101	291.606	44.495	150.276	185.(
896	ENEMDU*	21	Desempleo Oculto	48103	27.600	20.503	25.975	22.
897	ENEMDU*	21	Desempleo Cesante	298846	257.856	40.991	140.223	158.(
898	ENEMDU*	21	Desempleo Nuevo	85358	61.351	24.007	36.028	49.(
899	ENEMDU*	21	Población Económicamente Inactiva	4330241	3.307.420	1.022.821	1.324.745	3.005.(

900 rows × 8 columns

In [ ]:
```
dataPoblacion = poblacion[['Periodo','Indicadores','Total']].groupby(['Periodo','Indicadores'], as_index=False).mean()
dataPoblacion
```

Out[ ]:

	Periodo	Indicadores	Total
0	07	Desempleo	316697.0
1	07	Desempleo Abierto	193225.0
2	07	Desempleo Cesante	190044.0
3	07	Desempleo Nuevo	126653.0
4	07	Desempleo Oculto	123472.0
...	...	...	...
266	21	Población en Edad de Trabajar (PET)	12655772.8
267	21	Población menor de 15 años	5150507.2
268	21	Subempleo	1933593.1
269	21	Subempleo por insuficiencia de ingresos	211003.9
270	21	Subempleo por insuficiencia de tiempo de trabajo	1722589.2

271 rows × 3 columns

Extraer los datos de empleo y desempleo

In [ ]:
```
indicadorEmpleo = dataPoblacion[dataPoblacion['Indicadores'] == "Empleo"]
indicadorDesempleo = dataPoblacion[dataPoblacion['Indicadores'] == "Desempleo"]
indicadorEmpleo
indicadorDesempleo
```

Out[ ]:

	Periodo	Indicadores	Total
0	07	Desempleo	316697.00
18	08	Desempleo	362084.50
36	09	Desempleo	423802.00
54	10	Desempleo	365672.50
72	11	Desempleo	302996.00
90	12	Desempleo	279372.50
108	13	Desempleo	281348.00
126	14	Desempleo	304555.00
144	15	Desempleo	324618.00
162	16	Desempleo	423871.75
180	17	Desempleo	358466.50
199	18	Desempleo	330265.75
217	19	Desempleo	365105.75
235	20	Desempleo	456457.50
253	21	Desempleo	431402.70

Personas con empleo por año

In [ ]:
```
def graficaempleo():
  fig, ax = plt.subplots(figsize =(12, 7))
  plt.bar( indicadorEmpleo["Periodo"],indicadorEmpleo["Total"], col
or='Beige')

  try:
      plt.ticklabel_format(axis='y', style='plain')
  except AttributeError:
      print('')
  plt.title('Empleados por Año')
  plt.xlabel('Año')
  plt.ylabel('Cantidad de Empleados')
  plt.grid(linestyle='--', linewidth=0.4)
  for index,data in enumerate(indicadorEmpleo["Total"]):
      plt.text(x=index , y =data+1 , s=f"{data}" , fontdict=dict(fo
ntsize=8), ha='center', color='green', va='bottom')
  plt.tight_layout()
  plt.show()
  return fig
```

Personas con desempleo por año

In [ ]:
```
def graficaDesempleo():
  fig, ax = plt.subplots(figsize =(16, 9))
  plt.bar(indicadorDesempleo["Periodo"],indicadorDesempleo["Total"]
, color='red')

  try:
      plt.ticklabel_format(axis='y', style='plain')
  except AttributeError:
      print('')
  plt.title('Personas con desempleo por Año')
  plt.xlabel('Año')
  plt.ylabel('Cantidad de dempleados')
  plt.grid(linestyle='--', linewidth=0.4)
  for index,data in enumerate(indicadorDesempleo["Total"]):
      plt.text(x=index , y =data+1 , s=f"{data}" , fontdict=dict(fo
ntsize=10), ha='center', color='green', va='bottom')
  plt.tight_layout()
  plt.show()
  return fig
```

• Generar grafico de pie por personas basadas en la sectorización de empleo.

In [ ]:
```
def getString(x):
  return str(x)
def getInt(x):
  return int(x)
```

In [ ]:
```
def obtSectores(anio):
   dataSet = poblacion;
   dataSet['Periodo'] = dataSet['Periodo'].apply(getInt);
   dataSet = dataSet.query('Periodo == '+str(anio)+'');
   dataSet['Periodo'] = dataSet['Periodo'].apply(getString);
   return dataSet;
```

In [ ]:
```
def graficaSectores():
  Urbano = obtSectores(8).iloc[-18:]
  Urbano['Urbana'] = Urbano['Urbana'].apply(eliminarPuntos);

  vecDatos=[]
  vecEtiquetas=[]
  for x in Urbano['Urbana']:
    vecDatos.append(x);
  for y in Urbano['Indicadores']:
    vecEtiquetas.append(y)

  fig = plotly.pie(values=vecDatos, names=vecEtiquetas)
  fig.show()
  return fig
```

• Generar histogramas subempleo, empleo pleno y empleo no pleno por año.

In [ ]:
```
subEmpleo = dataPoblacion[dataPoblacion['Indicadores'] == "Subemple
o"]
empleoPleno = dataPoblacion[dataPoblacion['Indicadores'] == "Empleo
Adecuado/Pleno"]
empleoNopleno = dataPoblacion[dataPoblacion['Indicadores'] == "Otro
Empleo no pleno"]
```

In [ ]:
```
def geficaSubEmNoEm():
  barWidth = 0.3
  fig = plt.subplots(figsize =(18, 8))

  IT = np.array(subEmpleo["Total"])
  ECE = np.array(empleoPleno["Total"])
  CSE = np.array(empleoNopleno["Total"])

  br1 = np.arange(len(IT))
  br2 = [x + barWidth for x in br1]
  br3 = [x + barWidth for x in br2]

  plt.bar(br1, IT, color ='y', width = barWidth,
          edgecolor ='grey', label ='Subempleo')
  plt.bar(br2, ECE, color ='c', width = barWidth,
          edgecolor ='grey', label ='Empleo Pleno')
  plt.bar(br3, CSE, color ='m', width = barWidth,
          edgecolor ='grey', label ='Empleo No pleno')

  plt.xlabel('Año', fontweight ='bold', fontsize = 15)
  plt.ylabel('Datos mostrando en millones', fontweight ='bold', fon
tsize = 15)
  plt.title('Personas con Subempleo, Empleo pleno y empleo no pleno
',fontweight ='bold',)
  plt.xticks([r + barWidth for r in range(len(IT))],
          np.array(empleoNopleno["Periodo"]))

  plt.legend()
  plt.show()
  return plt
```

▪ Generar un reporte parametrizado que permita ingresar los datos de las fechas inicio y fin para obtener la información de las graficas vistas en el primer punto.

- Metodo para tomar los datos de una fecha de inicio y fecha de fin ingresada por teclado

In [ ]:
```
def getDataset(fechaInicio, fechaFin):
  dateInicio = int(fechaInicio)
  dateFIn = int(fechaFin)
  if dateInicio>=7 and dateFIn <=21:
    dataSet = poblacion;
    dataSet['Periodo'] = dataSet['Periodo'].apply(getInt);
    dataSet = dataSet.query('Periodo >='+fechaInicio+'& Periodo <=
'+fechaFin)
    dataSet['Periodo'] = dataSet['Periodo'].apply(getString);

    return dataSet;
  else:
    print("Error el rango invalido, permito es: 7 a 21, su rango fu
e " + fechaInicio + " a " + fechaFin)
```

In [ ]:
```
print("Buscar Personas Empleadas y Desempleadas")
print("Ingrese el rango en el que quiera buscar (Años)")
print("7 --> 2007 8 --> 2008")
print("Ingrese fecha de inicio")
fechaInicio = input();
print("Ingrese fecha de fin")
fechaFin = input();

empleados = getDataset(fechaInicio, fechaFin);
empleadoss = empleados[empleados['Indicadores'] == "Empleo"]
desempleados = empleados[empleados['Indicadores'] == "Desempleo"]
```

```
Buscar Personas Empleadas y Desempleadas
Ingrese el rango en el que quiera buscar (Años)
7 --> 2007 8 --> 2008
Ingrese fecha de inicio
7
Ingrese fecha de fin
21
```

Grafica de personas con empleo y personas sin empleo, las variables son ingresadas por teclado

In [ ]:
```
def grafporAnio():
  fig, ax = plt.subplots(figsize =(20, 9))
  empleado = ['Empl ']
  desempleado = ['Des ']
  grafporAnio = plt.bar(empleado + empleadoss["Periodo"],empleadoss
["Total"], label = 'Empleo', width = 0.5, color='lightblue')
  plt.bar(desempleado + desempleados["Periodo"],desempleados["Total
"], label = 'Desempleo', width = 0.5, color='orange')


  try:
      plt.ticklabel_format(axis='y', style='plain')
  except AttributeError:
      print('')
  plt.title('Personas con Empleo y Desempleo por Año con datos de l
as fechas inicio y fin')
  plt.xlabel('Año')
  plt.ylabel('Cantidad de dempleados')
  plt.grid(linestyle='--', linewidth=0.4)
  plt.tight_layout()
  plt.show()
  return fig
```

▪ Describir con estadística descriptiva los datos que se encuentran en el archivos.

In [ ]:
```
df = pd.DataFrame(indicadorEmpleo.describe(), columns = ['Total'])
df1 = pd.DataFrame(desempleo.describe(), columns = ['Total'])
df2 = pd.DataFrame(subEmpleo.describe(), columns = ['Total'])
df3 = pd.DataFrame(empleoPleno.describe(), columns = ['Total'])
df4 = pd.DataFrame(empleoNopleno.describe(), columns = ['Total'])

print("Empleo \n ", df, "\n")
print("Desempleo \n ", df1, "\n")
print("SubEmpleo \n ", df2, "\n")
print("Empleo Adecuado/Pleno \n ", df3, "\n")
print("Otro Empleo no pleno \n ", df4, "\n")
```

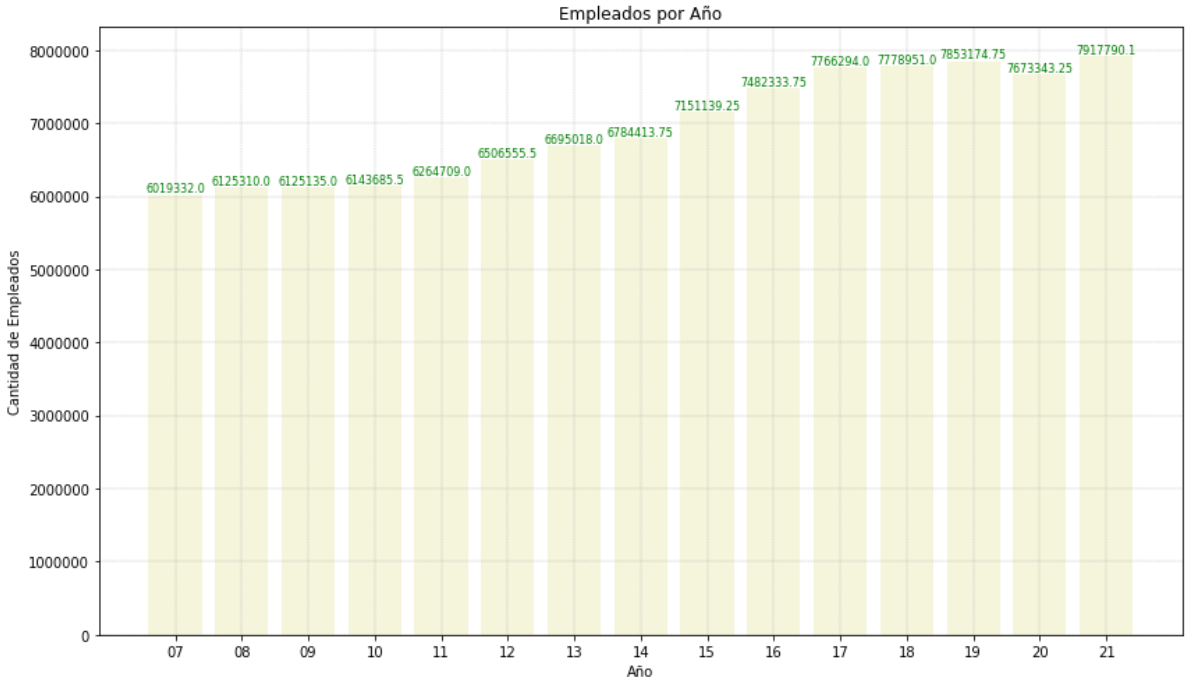
```
Empleo
                Total
count   1.500000e+01
mean    6.952479e+06
std     7.362216e+05
min     6.019332e+06
25%     6.204197e+06
50%     6.784414e+06
75%     7.719819e+06
max     7.917790e+06
```

```
Desempleo
                Total
count      15.000000
mean   355114.363333
std     56871.589568
min    279372.500000
25%    310626.000000
50%    358466.500000
75%    394737.250000
max    456457.500000

SubEmpleo
                Total
count  1.500000e+01
mean   1.229065e+06
std    4.250236e+05
min    6.344360e+05
25%    9.278525e+05
50%    1.071615e+06
75%    1.531731e+06
max    1.978117e+06

Empleo Adecuado/Pleno
                Total
count  1.500000e+01
mean   2.984756e+06
std    3.148057e+05
min    2.395944e+06
25%    2.766348e+06
50%    3.111499e+06
75%    3.219239e+06
max    3.404390e+06

Otro Empleo no pleno
                Total
count  1.500000e+01
mean   1.991426e+06
std    2.186774e+05
min    1.504000e+06
25%    1.886346e+06
50%    2.040986e+06
75%    2.122606e+06
max    2.265990e+06
```

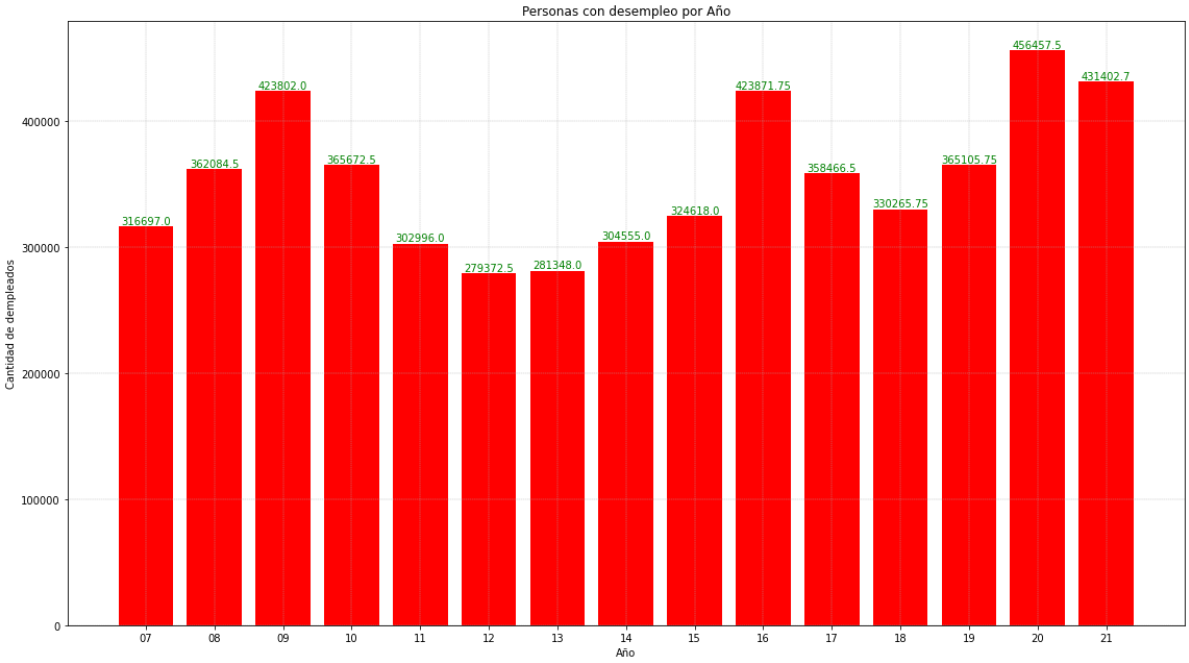
REPORTE

In [ ]:
```
graficaempleo = graficaempleo()
graficaDesempleo = graficaDesempleo()
graficaSectores = graficaSectores()
geficaSubEmNoEm = geficaSubEmNoEm()
grafporAnio = grafporAnio()
```

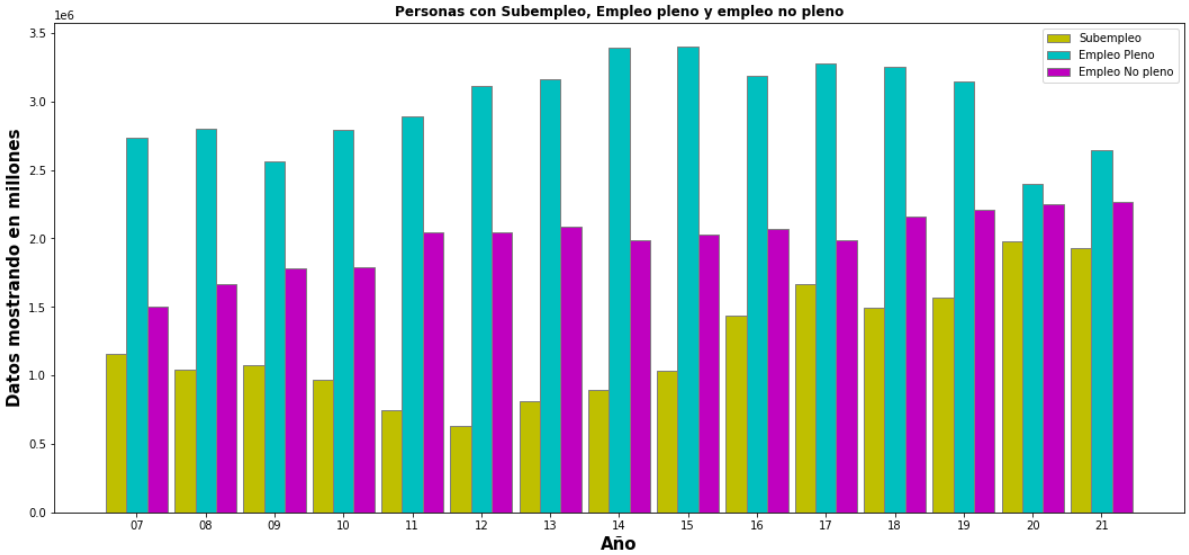
```
[05:57:52] [INFO ] Using categorical units to plot a list of strin
gs that are all parsable as floats or dates. If these strings shou
ld be plotted as numbers, cast to the appropriate data type before
plotting.
[05:57:52] [INFO ] Using categorical units to plot a list of strin
gs that are all parsable as floats or dates. If these strings shou
ld be plotted as numbers, cast to the appropriate data type before
plotting.
```

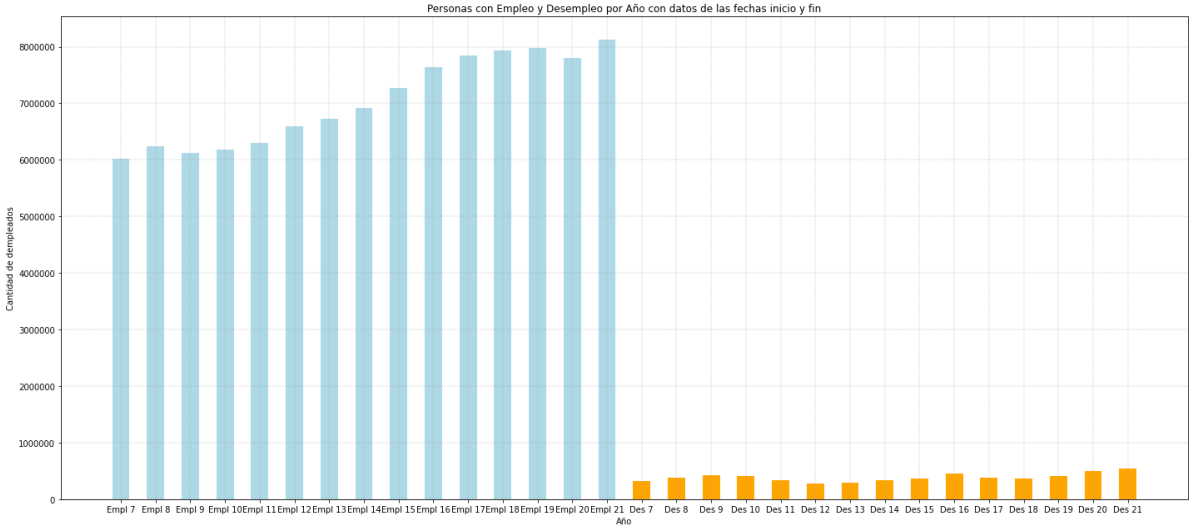



```
[05:57:53] [INFO ] Using categorical units to plot a list of strin
gs that are all parsable as floats or dates. If these strings shou
ld be plotted as numbers, cast to the appropriate data type before
plotting.
[05:57:53] [INFO ] Using categorical units to plot a list of strin
gs that are all parsable as floats or dates. If these strings shou
ld be plotted as numbers, cast to the appropriate data type before
plotting.
```




```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy
```


Personas con Subempleo, Empleo pleno y empleo no pleno
1e6
Datos mostrando en millones
Año
Subempleo
Empleo Pleno
Empleo No pleno
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21



Reporte de las graficas.

In [ ]:
```
report = dp.Report(dp.Text("""## Descripción de datos"""), dp.Text(
"""## Empleo"""), dp.Plot(graficaempleo), dp.Text("""## Desempleo""
"), dp.Plot(graficaDesempleo), dp.Text("""## Sectores"""), dp.Plot(
graficaSectores),  dp.Text("""## Empleo y desempleo con fecha de in
icio y fin"""), dp.Plot(grafporAnio), )
file_name = "reportePrueba.html"
report.save(path=file_name,open=True)
```

```
[05:59:33] [DEBUG] Saved object to /tmp/dp-tmp-uzlm_v0f/dp-tmp-4n_
levt8.svg (52930 bytes)
[05:59:33] [DEBUG] Saved object to /tmp/dp-tmp-uzlm_v0f/dp-tmp-4yt
r22cf.svg (49804 bytes)
[05:59:33] [DEBUG] Saved object to /tmp/dp-tmp-uzlm_v0f/dp-tmp-dpq
woq4x.pl.json (8704 bytes)
[05:59:33] [DEBUG] Saved object to /tmp/dp-tmp-uzlm_v0f/dp-tmp-jmz
4ys9r.svg (83060 bytes)
[05:59:33] [DEBUG] Successfully Built Report

Report saved to ./reportePrueba.html. To upload and share your rep
ort, create a free Datapane account by running `datapane signup`.
```

Analisis

• Con respecto a las dos primeras graficas de empleos y desempleos por año existe de un 20% a 30% mas personas con empleos que personas desempleadas.
• Con respecto al diagrama de paste se muestra que existe varios sectores y de esta forma es facil indetificar los valores altos y los bajos.

Concluciones

• Este trabajo que se realizo fue muy util ya que con esto se pudo profundisar en temas importantes como son las graficas, tomar datos de un csv, etc.
• Todos los datos que se obtuvo se logra concluir que los empleados en los ultimos 4 periodos se tiene un aumento del total de la población a comparacion de los desempleados se tiene una gran baja en el empleo.

Recomendaciones

• Se recomienda trabajar con todas las librerias actualizadas para evitar errores.
• Se recomienda analizar el dataSet antes de trabajarlo por ejemplo en el data set que se trabajo existia columnas sin nombre.

Enlace del GoogleColab:
https://drive.google.com/file/d/198h0oNvictLyUOSQVpODumV2f6ZB5pJz/view?usp=sharing (https://drive.google.com/file/d/198h0oNvictLyUOSQVpODumV2f6ZB5pJz/view?usp=sharing)

In [ ]:
```
!jupyter nbconvert --to html /content/PruebaUnidad1.ipynb
```