Ejercicio 1

Solución 1

import pandas as pd

import matplotlib.pyplot as plt

def grafico\_sectores\_ventas(ventas, titulo):

'''Función que construye un diagrama de sectores con las ventas de un trimestre y lo guarda con un nombre dado.

Parámetros:

- ventas: Es una serie de Pandas con las ventas del trimestre, siendo el índice los meses.

- titulo: Es una cadena con el título.

'''

# Definimos la figura y los ejes del gráfico con Matplotlib

fig, ax = plt.subplots()

# Dibujamos los sectores con las verntas a partir de la serie

ventas.plot(kind = 'pie', ax = ax)

# Eliminamos el título del eje y

plt.ylabel('')

# Añadimos el título

plt.title(titulo)

# Guardamos el gráfico con el nombre dado en formato png

plt.savefig(titulo + '.png')

return

ventas = {'Enero':200, 'Febrero':240, 'Marzo':310}

s\_ventas = pd.Series(ventas)

grafico\_sectores\_ventas(s\_ventas, 'Ventas primer trimestre')

Ejercicio 2

import pandas as pd

import matplotlib.pyplot as plt

def grafico\_evolucion\_ventas(ventas, tipo):

'''Función que construye un gráfico del tipo indicado con la evolución de las ventas por años.

Parámetros:

- ventas: Es un dataframe de Pandas con las ventas, siendo el índice los años.

- tipo: Es una cadena con el tipo de gráfico a dibujar: lineas, barras, pie o areas.

Salida:

Un gráfico del tipo indicado con la evolución de las ventas.

'''

# Definimos un diccionario con los tipos de gráficos

graficos = {'lineas':'line', 'barras':'bar', 'pie':'pie', 'area':'area'}

# Definimos la figura y los ejes del gráfico con Matplotlib

fig, ax = plt.subplots()

# Dibujamos las series de líneas con los ingresos y los gastos

ventas.plot(kind = graficos[tipo], ax = ax)

# Añadimos el título

plt.title('Evolución del número de ventas')

# Devolvemos el objeto con los ejes y el gráfico que contienten

return ax

df\_ventas = pd.Series([1200, 840, 1325, 1280, 1500], index = [2000, 2001, 2002, 2003, 2004])

grafico\_evolucion\_ventas(df\_ventas, 'lineas')

plt.show()

grafico\_evolucion\_ventas(df\_ventas, 'area')

plt.show()

grafico\_evolucion\_ventas(df\_ventas, 'barras')

plt.show()

grafico\_evolucion\_ventas(df\_ventas, 'sectores')

plt.show()

Ejercicio 3

import pandas as pd

import matplotlib.pyplot as plt

def grafico\_lineas\_ingresos\_gastos(datos):

'''Función que construye un gráfico de lineas con los ingresos y gastos de un cuatrimestre.

Parámetros:

- datos: Es un dataframe de Pandas con dos columnas, una para los ingresos y otra para los gastos, y como índice los meses.

Salida:

Un gráfico de líneas con los ingresos y los gastos dados.

'''

# Definimos la figura y los ejes del gráfico con Matplotlib

fig, ax = plt.subplots()

# Dibujamos las series de líneas con los ingresos y los gastos

datos.plot(ax = ax)

# Añadimos la escala del eje y

ax.set\_ylim([0, max(datos.Ingresos.max(), datos.Gastos.max()) + 500])

# Añadimos el título

plt.title('Evolución de ingresos y gastos')

# Devolvemos el objeto con los ejes y el gráfico que contienten

return ax

datos = {'Mes':['Ene', 'Feb', 'Mar', 'Abr'], 'Ingresos':[4500, 5200, 4800, 5300], 'Gastos':[2300, 2450, 2000, 2200]}

df\_datos = pd.DataFrame(datos).set\_index('Mes')

grafico\_lineas\_ingresos\_gastos(df\_datos)

plt.show()

Ejercicio 4

import pandas as pd

import matplotlib.pyplot as plt

# Creamos un dataframe a partir del archivo csv

df\_titanic = pd.read\_csv('titanic.csv')

# Creamos la figura y los ejes

fig, ax = plt.subplots()

# Gráfico pie de falleccidos y supervivientes

df\_titanic.Survived.value\_counts().plot(kind = "pie", labels = ["Muertos", "Supervivientes"], title = "Distribución de supervivientes")

plt.show()

# Histograma de edades

df\_titanic.Age.plot(kind = "hist", title = "Histograma de edades")

plt.show()

# Gráfico de barras con el número de personas de cada clase

df\_titanic.Pclass.value\_counts().plot(kind = "bar", title = "Número de personas por clase")

plt.show()

# Otra forma

df\_titanic.groupby("Pclass").size().plot(kind = "bar", title = "Número de personas por clase")

plt.show()

# Gráfico de barras con el número de personas fallecidas y supervivientes de cada clase

df\_titanic.groupby(["Pclass", "Survived"]).size().unstack().plot(kind = "bar", title = "Número de personas fallecidas y supervivientes por clase")

plt.show()

# Gráfico de barras con el número de personas fallecidas y supervivientes acumuladas de cada clase

df\_titanic.groupby(["Pclass", "Survived"]).size().unstack().plot(kind = "bar", stacked = True, title = "Número de personas fallecidas y supervivientes por clase")

plt.show()