**EJERCICIOS DE ETL con Pandas & MathPlotLib**

**Indicaciones**

* Modo de entrega: Individual, en aula de MIND HUB, Entrega clase 12
* Plazo: Hasta las 20:00 del 19 de abril
* Adjuntos: Este documento, más archivos PY/IPYNB y CSV de origen que correspondan

**Grupo N° 3:**

**Integrantes:**

* Felipe Bravo
* Nathalia Cerda
* Camila Valdivia
* Natalia Guajardo

**Desarrollo**

* Determinar qué datos son requeridos para visualizar dichos gráficos
* Evaluar si los datos deben ser procesados e indicar qué decide en base a su análisis
* Exportar los campos necesario a un archivo CSV, para ser importado por Pandas

#Explique cómo realizó el análisis y sus conclusiones

Para determinar los datos a utilizar para realizar los gráficos, el equipo realizó un minucioso análisis del archivo Excel enviado investigando el contenido de cada hoja.

La principal discusión estaba en ocupar 4 archivos csv (uno por año, “Finanzas 201X”) o utilizar un solo archivo csv con la información consolidada de los 4 años en una sola tabla (“Administración y finanzas”). En un principio habíamos decidido ocupar los archivos por separado porque eran menos datos con los que trabajar, pero luego, al empezar a codificar nos percatamos de que trabajar con un solo archivo era más producente.

Antes de traspasar la tabla de Excel a un csv se redondearon las cifras, para no tener problemas con las comas(,) de los números, al momento de convertir el archivo. Una vez el archivo csv estuvo listo, lo abrimos con un editor de texto y reemplazamos los ; por , para cumplir con el enunciado.

* Crear un programa en Python que consulte por teclado los nombres de los archivos CSV de origen, para ser cargados a los dataframes que usted estime conveniente

#Código utilizado

import pandas as pd

import matplotlib.pyplot as plt

csv = ["finanzasGenerales.csv"]

def read\_option():

try:

return pd.read\_csv(csv[0], sep=",", encoding= 'unicode\_escape')

except IndexError:

print("La opción ingresada no se encuentra disponible")

print(''''

##########Bienvenido############

En el siguiente CSV se despliega:

-La información de finanzas de

la empresa entre años 2016 a 2019.-

################################

''')

df\_years = read\_option()

#para visualizar el dataframe creado

df\_years

#Transformar cadena a entero en las columnas requeridas

df\_years["INGRESOS"] = df\_years.INGRESOS.str.strip('$')

df\_years["INGRESOS"] = df\_years["INGRESOS"].str.replace(".",'').astype(int)

df\_years["UTILIDAD"] = df\_years.UTILIDAD.str.strip('$')

df\_years["UTILIDAD"] = df\_years["UTILIDAD"].str.replace(".",'').astype(int)

df\_years['MES'] = df\_years.MES.str.rstrip(' ')

df\_years["SUELDOS"] = df\_years.SUELDOS.str.strip('$')

df\_years["SUELDOS"] = df\_years["SUELDOS"].str.replace(".",'').astype(int)

#para visualizar los cambios

df\_years

#Eliminamos los espacios de los valores en la columna mes

df\_years['MES'] = df\_years.MES.str.rstrip(' ')

#Agrupación de año y mes, para luego realizar una suma de sus ingresos pa

df\_years\_groups = df\_years.groupby(["AÑO","MES"], sort=False)[["UTILIDAD","INGRESOS"]].sum()

* Graficar
  1. Ingresos y Ganancias mediante gráfico de líneas
  2. Ventas, mediante gráfico de barras
  3. Costo de remuneraciones por año, mediante gráfico de torta

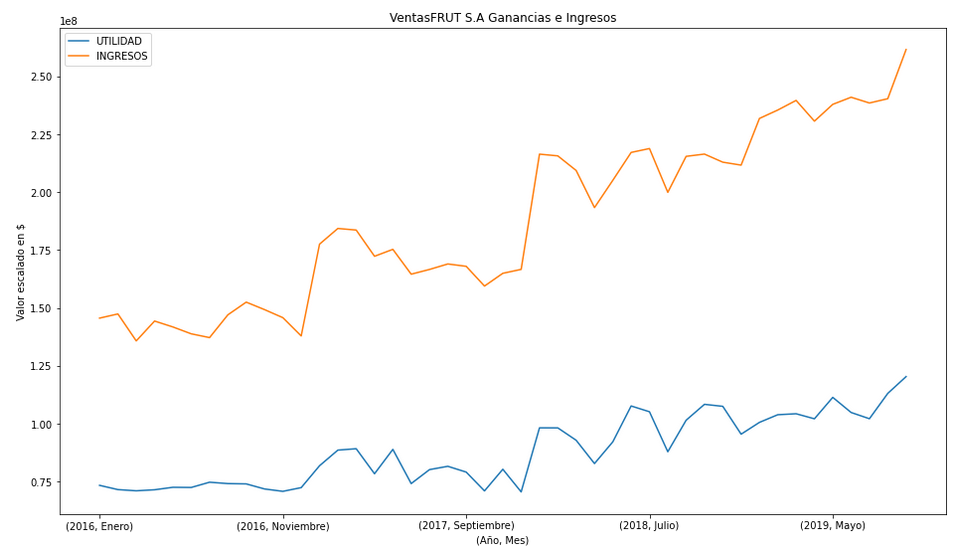
#Gráfico 1

#Ingresos y Ganancias mediante gráfico de líneas

graphic\_lines = df\_years\_groups.plot(kind='line', figsize=(16,9), xlabel='(Año, Mes)',

ylabel='Valor escalado en $',

title='VentasFRUT S.A Ganancias e Ingresos')

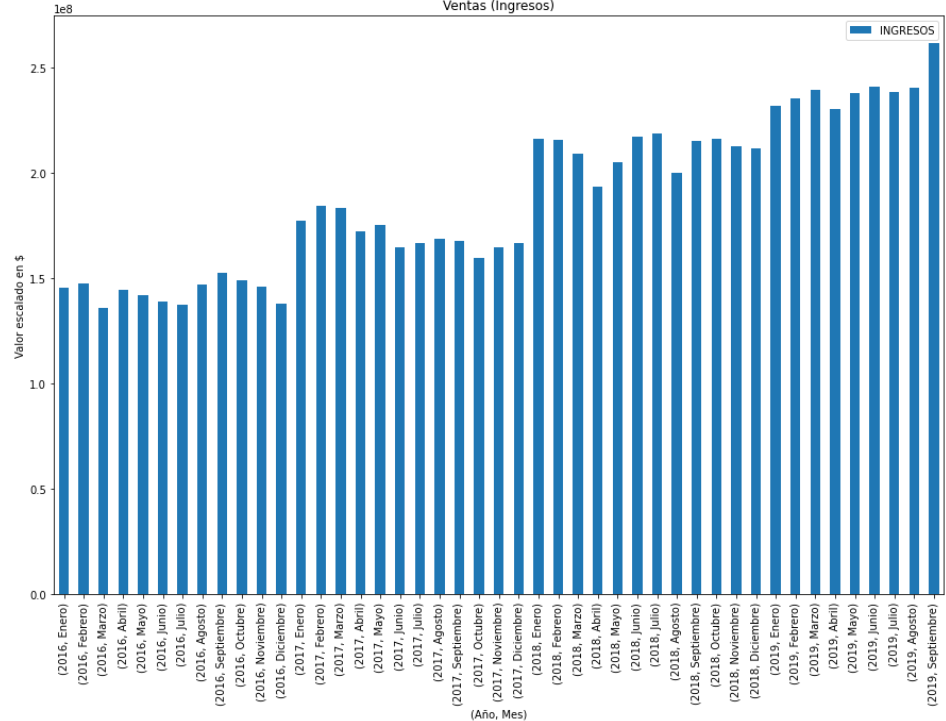


#Gráfico 2- de barras

df\_years\_i = df\_years.groupby(["AÑO","MES"], sort = False)[["INGRESOS"]].sum()

df\_years\_i

grafico = df\_years\_i.plot(kind = 'bar', figsize = (15,10), xlabel = '(Año, Mes)', ylabel = 'Valor escalado en $', title = 'Ventas (Ingresos)')



#Gráfico 3

remuneraciones = df\_years.groupby(['AÑO']).SUELDOS.sum()

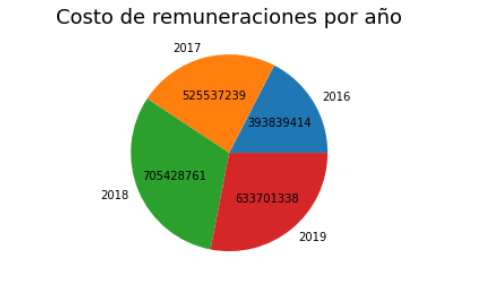
ano = df\_years['AÑO'].unique()

plt.pie(remuneraciones, labels=ano, autopct=lambda p:f'{p\*sum(remuneraciones)/100 :.0f}')

plt.title(f'Costo de remuneraciones por año', size = 18)

plt.show()

print(remuneraciones)



* Adaptar programa anterior para cargar los datos directamente desde Excel, usando la librería XLRD de Python.

#Código utilizado

import pandas as pd

import matplotlib.pyplot as plt

from xlrd import open\_workbook

file\_name = open\_workbook("Proyecto\_VentasFRUT.xlsx")

new\_array = file\_name.sheet\_names()

def return\_array():

n = 0

for f in new\_array:

print(f'{new\_array.index(f)}-{f}')

print('##########Opciones#############')

return\_array()

print('###############################')

try:

option = int(input("Ingrese una opción para mostrar: "))

except ValueError:

print("Error, ingrese un valor valido")

if option == 1:

df\_years = pd.read\_excel("Proyecto\_VentasFRUT.xlsx", sheet\_name=new\_array[option], header = 3)

df\_years

else:

print("Error, La unica opción disponible por el momento es la numero 1 !!!!!!!!!!!!!!")

##Grafico de lineas##

#Eliminamos los espacios de los valores en la columna mes

df\_years['MES'] = df\_years.MES.str.rstrip(' ')

#Agrupación de año y mes, para luego realizar una suma de sus ingresos pa

df\_years\_groups = df\_years.groupby(["AÑO","MES"], sort=False)[["UTILIDAD","INGRESOS"]].sum()

#Ingresos y Ganancias mediante gráfico de líneas

graphic\_lines = df\_years\_groups.plot(kind='line', figsize=(16,9), xlabel='(Año, Mes)',

ylabel='Valor escalado en $',

title='VentasFRUT S.A Ganancias e Ingresos')

#Grafico de barras

df\_years\_i = df\_years.groupby(["AÑO","MES"], sort = False)[["INGRESOS"]].sum()

df\_years\_i

grafico = df\_years\_i.plot(kind = 'bar', figsize = (15,10), xlabel = '(Año, Mes)', ylabel = 'Valor escalado en $', title = 'Ventas (Ingresos)')

remuneraciones = df\_years.groupby(['AÑO']).SUELDOS.sum()

ano = df\_years['AÑO'].unique()

plt.pie(remuneraciones, labels=ano, autopct=lambda p:f'{p\*sum(remuneraciones)/100 :.0f}')

plt.title(f'Costo de remuneraciones por año', size = 18)

plt.show()

print(remuneraciones)