UNIVERSIDAD DE GUANAJUATO
DCI

Trabajo Final Programacion en Python

Student name: *Rogelio Hashimoto Campos*

Course: *HIGI (Herramientas de la informacion)* – Professor: *Dra. Alma Xochitl Gonzalez Morales*
Due date: *Junio 17, 2021*

Abstract. The task which was given to us is to use python for the data analysis of 2 subjects, first one being free subject, I chose the analysis of the sales of a bicycle shop, the other one was about the National Electoral Institue, and what we had to do was predicting how many ballouts we need for the next 2 months for the next elections. In the case of the Bicycle shop, I calculated what was the total unity cost and follow a tendency as well as calculating which age group is more likely to buy at the shop nd what is the revenue of each group. In the NEI we had to read 16 different data sheets and concentrate all that info, to then make our ballout numbers prediction for January and February of the year 2020.

Analisis del archivo de ventas sobre bicicleta

Descripcion del Codigo

Listing 1: Librerias usadas.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

%matplotlib inline
```

Esto fueron las librerias que importe para empezar el trabajo, una vez importando todo lo que necesitaba empece a programar, cheque que se vieran todos los datos, use el codigo:

Listing 2: Extraer y leer documento .csv

```
sales = pd.read_csv(
'data/sales_data.csv',
parse_dates=['Date'])

sales.head()

sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113036 entries, 0 to 113035
Data columns (total 18 columns):
#   Column            Non-Null Count   Dtype
---  ------            --------------   -----
0   Date              113036 non-null  datetime64[ns]
1   Day               113036 non-null  int64
2   Month             113036 non-null  object
3   Year              113036 non-null  int64
4   Customer_Age      113036 non-null  int64
5   Age_Group         113036 non-null  object
6   Customer_Gender   113036 non-null  object
7   Country           113036 non-null  object
8   State             113036 non-null  object
9   Product_Category  113036 non-null  object
10  Sub_Category      113036 non-null  object
11  Product           113036 non-null  object
12  Order_Quantity    113036 non-null  int64
13  Unit_Cost         113036 non-null  int64
14  Unit_Price        113036 non-null  int64
15  Profit            113036 non-null  int64
16  Cost              113036 non-null  int64
17  Revenue           113036 non-null  int64
dtypes: datetime64[ns](1), int64(9), object(8)
```

En la siguiente imagen se puede observar Que en verdad si esta leyendo los datos y los esta separando.

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Product_Category	Sub_Category	Product	Order_(
0	2013-11-26	26	November	2013	19	Youth (<25)	M	Canada	British Columbia	Accessories	Bike Racks	Hitch Rack - 4-Bike	8
1	2015-11-26	26	November	2015	19	Youth (<25)	M	Canada	British Columbia	Accessories	Bike Racks	Hitch Rack - 4-Bike	8
2	2014-03-23	23	March	2014	49	Adults (35-64)	M	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	23
3	2016-03-23	23	March	2016	49	Adults (35-64)	M	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	20
4	2014-05-15	15	May	2014	47	Adults (35-64)	F	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	4

En este proyecto hice analisis y visualizacion de la tabla de Unidad de costo.

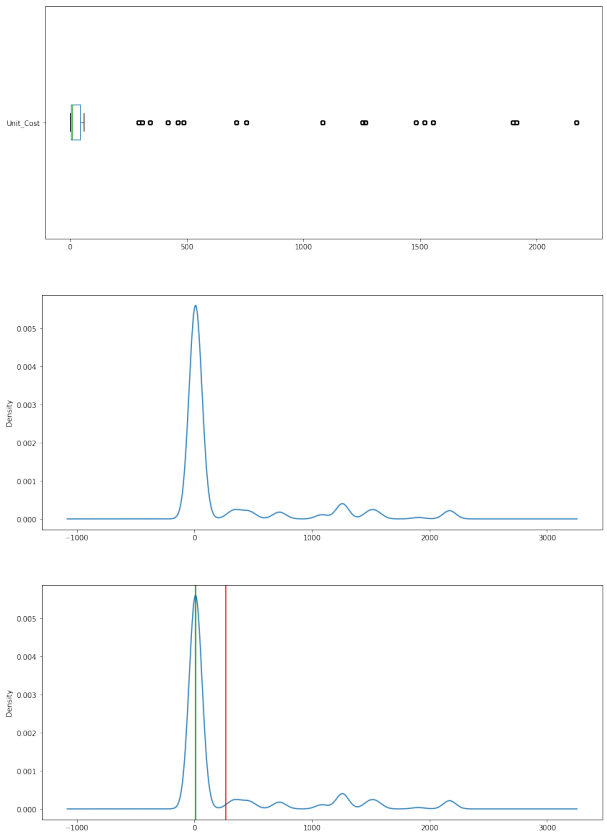
Listing 3: Librerias usadas.

```
In[8]: sales['Unit_Cost'].describe()

Out[8]:

count    113036.000000
mean        267.296366
std         549.835483
min           1.000000
```

```
25%              2.000000
50%              9.000000
75%             42.000000
max           2171.000000
Name: Unit_Cost, dtype: float64

In [9]:

sales['Unit_Cost'].mean()

Out[9]:

267.296365759581

In [10]:

sales['Unit_Cost'].median()

Out[10]:

9.0
```

Obtuve la infromacion de la columna unidad de costo, nos da el promedio que es 267.2963, tambien nos arroja desviacion estanda, mediana y maximo.

En las siguientes graficas nos mestra la interaccion entre la densidad que se tiene del producto y su unidad de costo, en otra grafica se tiene lo mismo pero se mustra la linea del promedio y de la mediana.

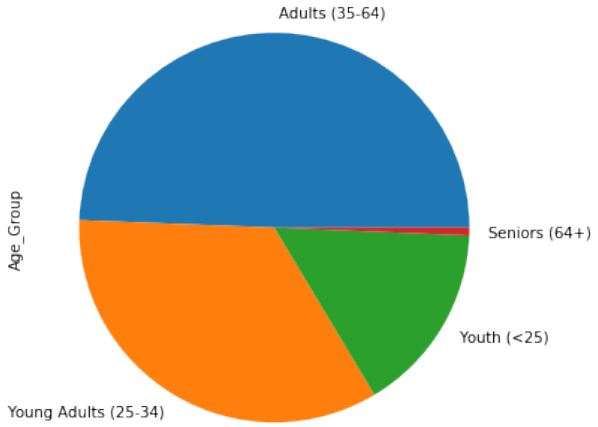


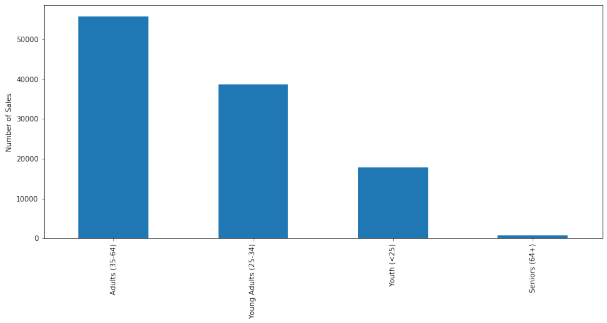
A continuacion se hizo el analisis con el grupo de edades, sobre el consumo que tienen en la tienda.

Listing 4: Librerias usadas.

```
In [15]:

sales['Age_Group'].value_counts()

Out[15]:

Adults (35-64)             55824
Young Adults (25-34)       38654
Youth (<25)                17828
Seniors (64+)                730
Name: Age_Group, dtype: int64

In [17]:

sales['Age_Group'].value_counts().plot(kind='pie', figsize=(6,6))

Out[17]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8ade8bc2e0>
```

Este codigo nos arrojo 2 graficas:





En las siguientes Lineas de codigo, jugue con la informacion y queria ver la correlacion entre los datos.

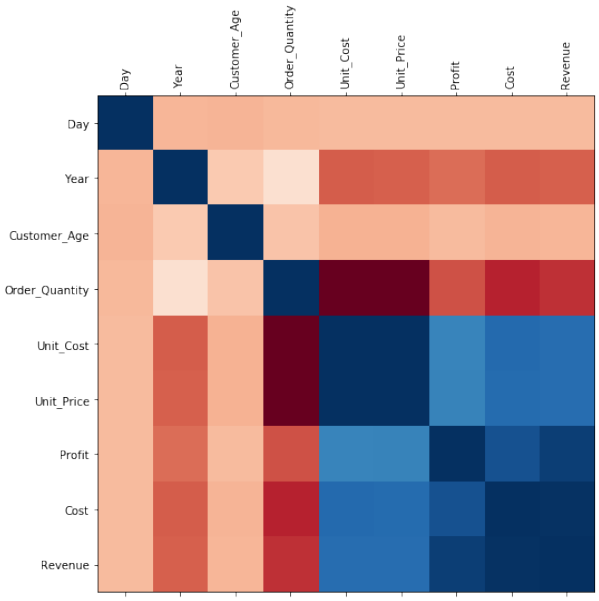
```
In [19]: corr = sales.corr()
         corr
```

Out[19]:

	Day	Year	Customer_Age	Order_Quantity	Unit_Cost	Unit_Price	Profit	Cost	Revenue
Day	1.000000	-0.007635	-0.014296	-0.002412	0.003133	0.003207	0.004623	0.003329	0.003853
Year	-0.007635	1.000000	0.040994	0.123169	-0.217575	-0.213673	-0.181525	-0.215604	-0.208673
Customer_Age	-0.014296	0.040994	1.000000	0.026887	-0.021374	-0.020262	0.004319	-0.016013	-0.009326
Order_Quantity	-0.002412	0.123169	0.026887	1.000000	-0.515835	-0.515925	-0.238863	-0.340382	-0.312895
Unit_Cost	0.003133	-0.217575	-0.021374	-0.515835	1.000000	0.997894	0.741020	0.829869	0.817865
Unit_Price	0.003207	-0.213673	-0.020262	-0.515925	0.997894	1.000000	0.749870	0.826301	0.818522
Profit	0.004623	-0.181525	0.004319	-0.238863	0.741020	0.749870	1.000000	0.902233	0.956572
Cost	0.003329	-0.215604	-0.016013	-0.340382	0.829869	0.826301	0.902233	1.000000	0.988758
Revenue	0.003853	-0.208673	-0.009326	-0.312895	0.817865	0.818522	0.956572	0.988758	1.000000

Listing 5: Grafica de correlacion de todos los datos.

```
fig = plt.figure(figsize=(8,8))
plt.matshow(corr, cmap='RdBu', fignum=fig.number)
plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical');
plt.yticks(range(len(corr.columns)), corr.columns);
```



Por ultimo saque las ganancias que se tienen por cada grupo de edad.

Listing 6: Grafica de correlacion de todos los datos.

```
Get the mean revenue of the Adults (35-64) sales group
In [39]:

sales.loc[sales['Age_Group'] == 'Adults (35-64)', 'Revenue'].mean()

Out[39]:

762.8287654055604

How many records belong to Age Group Youth (<25) or Adults (35-64)?
In [43]:

sales.loc[(sales['Age_Group'] == 'Youth (<25)') | (sales['Age_Group'] == 'Adults (35-64)

Out[43]:

73652
```

```
Get the mean revenue of the sales group Adults (35-64) in United States
In [44]:

sales.loc[(sales['Age_Group'] == 'Adults (35-64)') & (sales['Country'] == 'United States

Out[44]:

726.7260473588342

Increase the revenue by 10% to every sale made in France
In [45]:

sales.loc[sales['Country'] == 'France', 'Revenue'].head()

Out[45]:

50      787
51      787
52     2957
53     2851
60      626
Name: Revenue, dtype: int64

In [46]:

#sales.loc[sales['Country'] == 'France', 'Revenue'] = sales.loc[sales['Country'] == 'Fra

sales.loc[sales['Country'] == 'France', 'Revenue'] *= 1.1

In [47]:

sales.loc[sales['Country'] == 'France', 'Revenue'].head()

Out[47]:

50      865.7
51      865.7
52     3252.7
53     3136.1
60      688.6
Name: Revenue, dtype: float64
```

Analisis del archivo de INE y prediccion de Casillas.

Se usaron las siguientes librerias:

Listing 7: Grafica de correlacion de todos los datos.

```
##Importar librerias a utilizar
import numpy as np
import matplotlib.pyplot as plt
#from scipy.optimize import minimize
import os
import glob
import pandas as pd
import re
import math
```

Y se usaron las siguientes lineas de codigo para poder leer los documentos .csv:

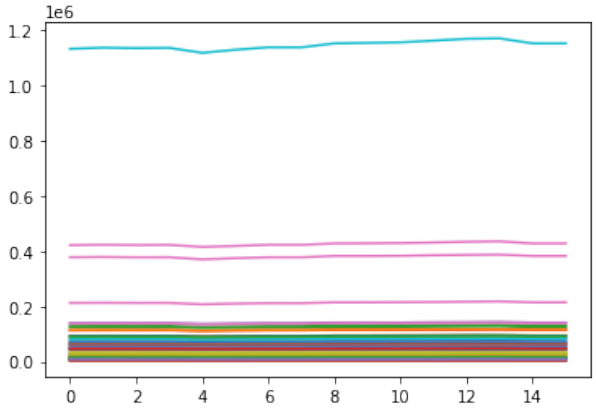
Listing 8: Grafica de correlacion de todos los datos.

```
import glob
files=glob.glob("./higi/*.txt")

date=[]
date_=[]
files_=[]

for i,file in enumerate(files):
date.append(re.findall(r'\d+',file)[0])


temp=sorted(range(len(date)), key=date.__getitem__)

for i in temp:
date_.append(date[i])
print(date[i],files[i])
files_.append(files[i])
```

Se uso el codigo para poder acomodar los archivos por meses y asi extraerlos en orden. Se obtuvo la lista nominal y la lista nacional de cada seccion de Guanajuato, y se grafico.

Listing 9: Grafica de correlacion de todos los datos.

```
for i,file in enumerate(files_):
data=pd.read_csv(file)
data=data[1:]
data=data[data['ENTIDAD']==11][1:]
mpo=data.groupby(['MUNICIPIO']).sum()
if i==0 :
if 'LISTA_NAL' in mpo.columns:
df_mpo = pd.DataFrame(mpo['LISTA_NAL'])
if 'LISTA_NACIONAL' in mpo.columns:
df_mpo = pd.DataFrame(mpo['LISTA_NACIONAL'])
if 'LISTA' in mpo.columns:
df_mpo = pd.DataFrame(mpo['LISTA'])
else:
if 'LISTA_NAL' in mpo.columns:
df_mpo[date_[i]]=mpo['LISTA_NAL']
if 'LISTA_NACIONAL' in mpo.columns:
df_mpo[date_[i]]=mpo['LISTA_NACIONAL']
if 'LISTA' in mpo.columns:
df_mpo[date_[i]]=mpo['LISTA']ing}

plt.figure(figsize=(6,6))

for i in range(46):
plt.plot(df_mpo.iloc[i])
plt.yscale('log')
```



Listing 10: Grafica de correlacion de todos los datos.

```
fits=[]
prediction_lnal=[]

for i in range(len(municipios)):
xx=np.arange(len(municipios[i]))
ma, ba = np. polyfit(xx, municipios[i],1,w=municipios[i])
fits.append([ma,ba])
pred=ma*(xx[-12]+12)+ba
#if pred < municipios[i][-1]:
#     pred=municipios[i][-1]

prediction_lnal.append(pred)

prediccion=df_mpo['Prediction_LNAL']
prediccion.sum()
prediccion.sum()/750

for i,file in enumerate(files_):
data=pd.read_csv(file)
data=data[1:]
data=data[data['ENTIDAD']==11][1:]
mpo=data.groupby(['SECCION']).sum()
if i==0 :
if 'LISTA_NAL' in mpo.columns:
df_SEC = pd.DataFrame(mpo['LISTA_NAL'])
if 'LISTA_NACIONAL' in mpo.columns:
df_SEC = pd.DataFrame(mpo['LISTA_NACIONAL'])
if 'LISTA' in mpo.columns:
df_SEC = pd.DataFrame(mpo['LISTA'])
else:
if 'LISTA_NAL' in mpo.columns:
df_SEC[date_[i]]=mpo['LISTA_NAL']
if 'LISTA_NACIONAL' in mpo.columns:
df_SEC[date_[i]]=mpo['LISTA_NACIONAL']
if 'LISTA' in mpo.columns:
df_SEC[date_[i]]=mpo['LISTA']
```

ICIPIO	LISTA	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	Prediction LNAL
1	67427	67541	67450	67463	65505	66245	66750	66736.0	68057	68093	68244	68241	68422	68611	68052	68054	68460.251924
2	94040	94139	93825	93970	91712	92759	93343	93340.0	94967	95009	95209	95595	96005	95934	94952	94967	95772.011451
3	128446	128937	128731	128978	125884	127461	128544	128543.0	130720	130992	131172	131903	132669	133489	130708	130717	132561.633385
4	49656	49787	49699	49818	48907	49311	49596	49596.0	50283	50287	50324	50647	50867	50877	50273	50281	50745.340228
5	67868	68120	67992	68218	66789	67553	68063	68058.0	69125	69132	69287	69745	70072	70194	69114	69124	69994.162832

SECCION	LISTA	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	Prediction SEC
1	1539	1545	1539	1538	1493	1515	1527	1527.0	1552	1557.0	1562.0	1561.0	1565.0	1571.0	1552.0	1552	1564.929138
2	1998	2008	2004	2005	1939	1963	1977	1977.0	2026	2024.0	2034.0	2028.0	2028.0	2033.0	2026.0	2026	2035.501966
3	1522	1522	1525	1520	1457	1477	1493	1493.0	1531	1529.0	1533.0	1528.0	1534.0	1537.0	1531.0	1531	1534.396793
4	945	949	949	952	915	932	938	938.0	957	959.0	961.0	959.0	962.0	957.0	957.0	957	961.691591
5	1067	1065	1065	1064	1026	1036	1047	1045.0	1072	1071.0	1076.0	1073.0	1067.0	1068.0	1072.0	1072	1072.347876

```
In [39]: total=sum(pr)
         math.ceil(total)
Out[39]: 6028
```

SE obtuvieron los numeros de personas que son potenciales votadores, y se dividio todo entre 750, que son el numero de personas por casilla.

En total nos dio 6028 de prediccion.