Data Insights

- By - Gautam Sharma

Import all required libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime
from matplotlib.ticker import FuncFormatter
import math as ma
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data = pd.ExcelFile('KPMG_VI_New_raw_data_update_final.xlsx')
```

In [3]:

```
data.sheet_names
```

Out[3]:

```
['Title Sheet',
 'Transactions',
 'NewCustomerList',
 'CustomerDemographic',
 'CustomerAddress']
```

In [4]:

```
df = pd.read_excel('KPMG_VI_New_raw_data_update_final.xlsx',3)
```

In [5]:

```
1 df.head()
```

Out[5]:

	Note: The data and information in this document is reflective of a hypothetical situation and client. This document is to be used for KPMG Virtual Internship purposes only.	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed 5
0	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	DOE
1	1	Laraine	Medendorp	F	93	1953-10-12 00:00:00
2	2	Eli	Bockman	Male	81	1980-12-16 00:00:00
3	3	Arlin	Dearle	Male	61	1954-01-20 00:00:00
4	4	Talbot	NaN	Male	33	1961-10-03 00:00:00

5 rows × 26 columns

In [6]:

```
1 df.isnull().sum()
```

Out[6]:

```
Note: The data and information in this document is reflective of a hyp
othetical situation and client. This document is to be used for KPMG V
irtual Internship purposes only.        0
Unnamed: 1
0
Unnamed: 2
125
Unnamed: 3
0
Unnamed: 4
0
Unnamed: 5
87
Unnamed: 6
506
Unnamed: 7
656
Unnamed: 8
0
Unnamed: 9
0
Unnamed: 10
302
Unnamed: 11
0
Unnamed: 12
87
Unnamed: 13
4001
Unnamed: 14
4001
Unnamed: 15
4001
Unnamed: 16
4001
Unnamed: 17
4001
Unnamed: 18
4001
Unnamed: 19
4001
Unnamed: 20
4001
Unnamed: 21
4001
Unnamed: 22
4001
Unnamed: 23
4001
Unnamed: 24
4001
Unnamed: 25
4001
dtype: int64
```

In [7]:

```
# rename for easier analysis
df.rename(columns={"Note: The data and information in this document is reflecti
df.rename(columns={"Unnamed: 1":"fname",
                   "Unnamed: 2":"lname",
                   "Unnamed: 3":"gender",
                   "Unnamed: 4":"3y_bike_purchases",
                   "Unnamed: 5":"DOB",
                   "Unnamed: 6":"JT"}, inplace = True)
df.rename(columns={"Unnamed: 7":"Category",
                   "Unnamed: 8":"wealth_segement",
                   "Unnamed: 9":"D_Indicator",
                   "Unnamed: 10":"default",
                   "Unnamed: 11":"owns_car",
                   "Unnamed: 12":"tencure"}, inplace = True)
df=df.iloc[1:]
df.head()
```

Out[7]:

	customer_id	fname	lname	gender	3y_bike_purchases	DOB	JT	Catego
1	1	Laraine	Medendorp	F	93	1953-10-12 00:00:00	Executive Secretary	Hea
2	2	Eli	Bockman	Male	81	1980-12-16 00:00:00	Administrative Officer	Financ Servic
3	3	Arlin	Dearle	Male	61	1954-01-20 00:00:00	Recruiting Manager	Prope
4	4	Talbot	NaN	Male	33	1961-10-03 00:00:00	NaN	
5	5	Sheila-kathryn	Calton	Female	56	1977-05-13 00:00:00	Senior Editor	Na

5 rows × 26 columns

In [8]:

```
df.columns
```

Out[8]:

```
Index(['customer_id', 'fname', 'lname', 'gender', '3y_bike_purchases',
'DOB',
       'JT', 'Category', 'wealth_segement', 'D_Indicator', 'default',
       'owns_car', 'tencure', 'Unnamed: 13', 'Unnamed: 14', 'Unnamed:
15',
       'Unnamed: 16', 'Unnamed: 17', 'Unnamed: 18', 'Unnamed: 19',
       'Unnamed: 20', 'Unnamed: 21', 'Unnamed: 22', 'Unnamed: 23',
       'Unnamed: 24', 'Unnamed: 25'],
      dtype='object')
```

In [9]:

```
df = df.drop(['Unnamed: 13', 'Unnamed: 14', 'Unnamed: 15','Unnamed: 16', 'Unnam
```

In [10]:

```
df.head()
```

Out[10]:

	customer_id	fname	lname	gender	3y_bike_purchases	DOB	JT	Catego
1	1	Laraine	Medendorp	F	93	1953-10-12 00:00:00	Executive Secretary	Hea
2	2	Eli	Bockman	Male	81	1980-12-16 00:00:00	Administrative Officer	Financ Servic
3	3	Arlin	Dearle	Male	61	1954-01-20 00:00:00	Recruiting Manager	Prope
4	4	Talbot	NaN	Male	33	1961-10-03 00:00:00	NaN	
5	5	Sheila-kathryn	Calton	Female	56	1977-05-13 00:00:00	Senior Editor	Na

In [11]:

```
df.shape
```

Out[11]:

```
(4000, 13)
```

In [12]:

```
df.isnull().sum()
```

Out[12]:

```
customer_id            0
fname                  0
lname                125
gender                 0
3y_bike_purchases      0
DOB                   87
JT                   506
Category             656
wealth_segement        0
D_Indicator            0
default              302
owns_car               0
tencure               87
dtype: int64
```

In [13]:

```
# Analyze the gender VS bike bought
gender = [0,0,0]
for each in df['gender']:
    if each[0] == 'F':
        gender[0] += 1
    elif each[0] == "M":
        gender[1] += 1
    else:
        gender[2] += 1

print(gender)
```

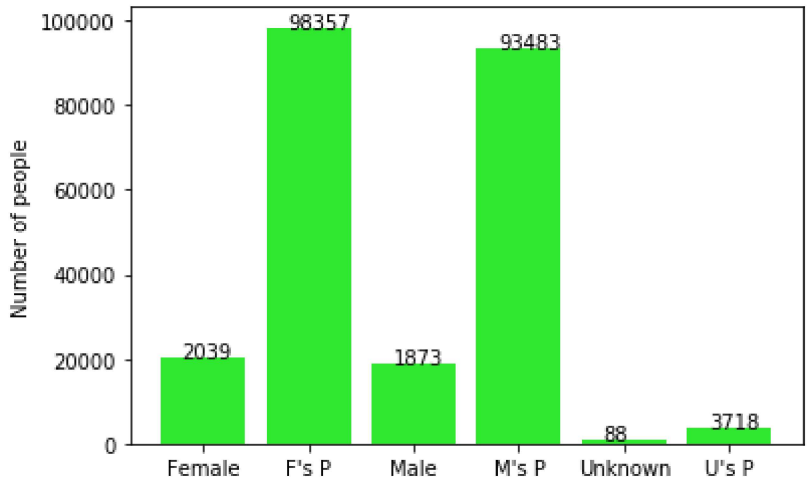
```
[2039, 1873, 88]
```

In [14]:

```
def gf(x, pos):
    #'The two args are the value and gender'
    return int(x)
```

In [15]:

```
bike = [0,0,0]
bike[0] += df['3y_bike_purchases'][df['gender'] == 'Female'].sum() + df['3y_bik
bike[1] += df['3y_bike_purchases'][df['gender'] == 'Male'].sum() + df['3y_bike_
bike[2] += df['3y_bike_purchases'][df['gender'] == 'U'].sum()

grapho = [gender[0], bike[0], gender[1], bike[1], gender[2], bike[2]]
graphl = [gender[0]*10, bike[0], gender[1]*10, bike[1], gender[2]*10, bike[2]]
colorr = (0.1,0.9,0.1,0.9)
#scale up gender by 10 for easier visualization
formatter = FuncFormatter(gf)
x = np.arange(6)
fig, ax = plt.subplots()
ax.set_ylabel('Number of people')
ax.yaxis.set_major_formatter(formatter)
plt.bar(x, graphl, color = colorr)
for i in range(len(gender*2)):
    plt.text(x = i-0.2, y = graphl[i]+0.1, s = grapho[i], size = 10)

plt.xticks(x, ('Female', "F's P", 'Male', "M's P", 'Unknown', "U's P"))
plt.show()
```

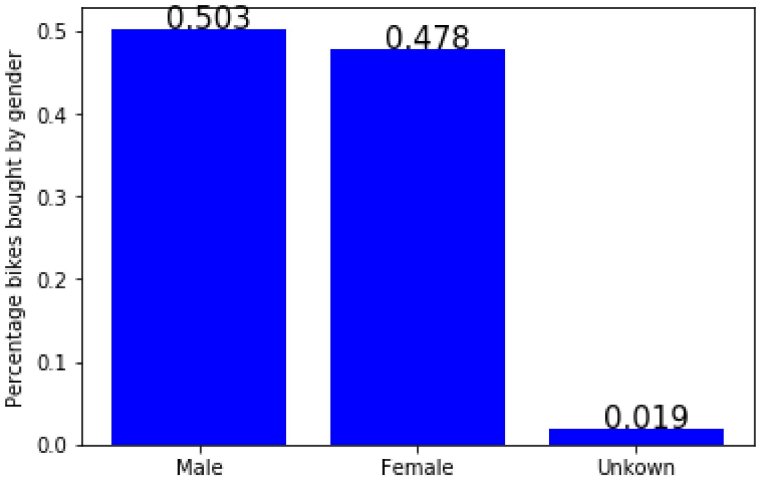


In [16]:

```
# Percentage of bike bought male vs female and other

avg = [0,0,0]
ss = sum(bike)

for i in range(len(avg)):
    avg[i] += round(bike[i]/ss, 3)

```

In [17]:

```
x1 = np.arange(3)
fig, ax1 = plt.subplots()
ax1.set_ylabel('Percentage bikes bought by gender')
plt.bar(x1,avg,color='blue')
for i in range(len(avg)):
    plt.text(x = i-0.15, y=avg[i],s=avg[i],size=15)

plt.xticks(x1,('Male','Female','Unkown'))
plt.show()
```



In [18]:

```
# Analyze on the age VS bikes bought
# Need to transform

print(df['DOB'][1].ctime().split(" ")[4])
df['Age'] = 0
lenn = len(df["DOB"])
k = 0
```

```
1953
```

In [19]:

```
for i in range(1, lenn):
    if isinstance(df["DOB"][i], datetime.date):
        t1 = len(df["DOB"][i].ctime().split(" "))
        df['Age'][i] += int(2019-int(df['DOB'][i].ctime().split(' ')[t1-1]))
    elif isinstance(df['DOB'][i],str):
        t1 = len(df['DOB'][i].split("-"))
        df['Age'][i] += int(209 - int(df['DOB'][i].split('-')[t1-1]))
```

In [ ]:

```
print(k)
```

In [20]:

```
df.head()
```

Out[20]:

er_id	fname	lname	gender	3y_bike_purchases	DOB	JT	Category	wealth_s
1	Laraine	Medendorp	F	93	1953-10-12 00:00:00	Executive Secretary	Health	Mass (
2	Eli	Bockman	Male	81	1980-12-16 00:00:00	Administrative Officer	Financial Services	Mass (
3	Arlin	Dearle	Male	61	1954-01-20 00:00:00	Recruiting Manager	Property	Mass (
4	Talbot	NaN	Male	33	1961-10-03 00:00:00	NaN	IT	Mass (
5	Sheila-kathryn	Calton	Female	56	1977-05-13 00:00:00	Senior Editor	NaN	Affluent (

In [21]:

```
#same index as above
ngenage = [0,0,0]
j       = 1
ss      = 0
stdv    = []
for each in df['gender']:
    if each[0] == "F" and df['Age'][j] != 0:
        ss   += df['Age'][j]
        stdv.append((df['Age'][j]))
        ngenage[0] += 1
    elif each[0] == "M" and df['Age'][j] != 0:
        ss += df['Age'][j]
        stdv.append((df['Age'][j]))
        ngenage[1] += 1
    elif df['Age'][j] != 0:
        ss += df['Age'][j]
        stdv.append((df['Age'][j]))
        ngenage[2] += 1
    j += 1
```

In [22]:

```
print(ss)
print(ngenage) # as we can see, unknonwn gender will unlikely to have age, don'
#average age not counting 0 is
mean_val = round(ss/sum(ngenage),0)
print(mean_val)
stdv_val = round(ma.sqrt(1/(sum(ngenage)-1)*sum((stdv - (ss/sum(ngenage)))**2))
print(stdv_val)
```

```
164210
[2039, 1872, 1]
42.0
13.0
```

In [23]:

```
#pruchases from age 42 - 34/2, 42, 42 + 34/2
age_dict = {}
f1  = []
bf1 = []
f2  = []
bf2 = []
f3  = []
bf3 = []
f4  = []
bf4 = []
m1  = []
bm1 = []
m2  = []
bm2 = []
m3  = []
bm3 = []
m4  = []
bm4 = []
fq = mean_val - stdv_val/2
sq = mean_val
tq = mean_val + stdv_val/2
print(fq, sq, tq)
jjj = 1
for each in df['gender']:
    temp = int(df['Age'][jjj])
    bkt  = int(df['3y_bike_purchases'][jjj])
    if each[0] == "F" and temp != 0:
        if(temp <= fq):
            f1.append(temp)
            bf1.append(bkt)
        elif(fq < temp and temp <= sq):
            f2.append(temp)
            bf2.append(bkt)
        elif(sq < temp and temp <= tq):
            f3.append(temp)
            bf3.append(bkt)
        elif(tq < temp):
            f4.append(temp)
            bf4.append(bkt)
    elif each[0] == "M" and temp != 0:
        if(temp <= fq):
            m1.append(temp)
            bm1.append(bkt)
        elif(fq < temp and temp <= sq):
            m2.append(temp)
            bm2.append(bkt)
        elif(sq < temp and temp <= tq):
            m3.append(temp)
            bm3.append(bkt)
        elif(tq < temp):
            m4.append(temp)
            bm4.append(bkt)
    jjj += 1
dtt = {"Female1":f1,
       "Female2":f2,
       "Female3":f3,
       "Female4":f4,
       "Male1":m1,
       "Male2":m2,
```

```
        "Male3":m3,
        "Male4":m4,
       }
```

```
35.5 42.0 48.5
```

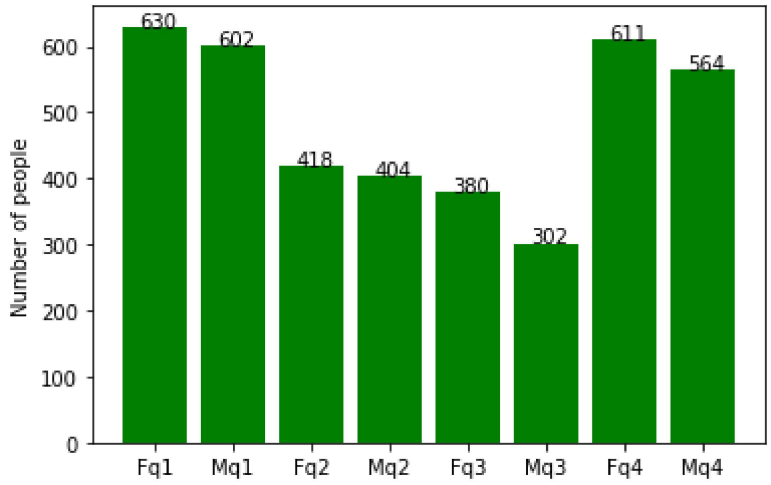
In [24]:

```
print(len(dtt["Female1"]), len(dtt["Female2"]), len(dtt["Female3"]), len(dtt["F
print(len(dtt["Male1"]), len(dtt["Male2"]), len(dtt["Male3"]), len(dtt["Male4"]
```

```
630 418 380 611
602 404 302 564
```

In [25]:

```
graphgen = [len(dtt["Female1"]), len(dtt["Male1"]), len(dtt["Female2"]), len(dt
colorr = (0.7,0.7,0.7,0.7)

x2 = np.arange(8)
fig2, ax2 = plt.subplots()
ax2.set_ylabel('Number of people')
plt.bar(x2, graphgen, color = 'green')
for i in range(len(graphgen)):
    plt.text(x = i-0.2, y = graphgen[i]+0.1, s = graphgen[i], size = 10)

plt.xticks(x2, ('Fq1', "Mq1", 'Fq2', "Mq2", 'Fq3', "Mq3", 'Fq4', "Mq4"))
plt.show()
```



In [26]:

```
#check_unique()["Category"]
df["Category"].value_counts()
```

Out[26]:

```
Manufacturing          799
Financial Services     774
Health                 602
Retail                 358
Property               267
IT                     223
Entertainment          136
Argiculture            113
Telecommunications      72
Name: Category, dtype: int64
```

In [27]:

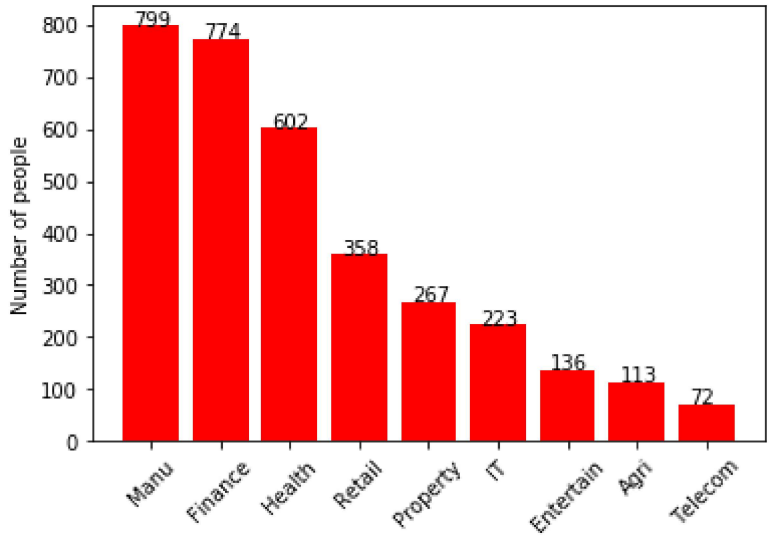
```
df.describe()
```

Out[27]:

	Age
count	4000.000000
mean	41.052500
std	14.103084
min	0.000000
25%	31.750000
50%	42.000000
75%	51.000000
max	188.000000

In [28]:

```
val = [799, 774, 602, 358, 267, 223, 136, 113, 72]
colorr = (0.2,0.3,0.4,0.5)

x3 = np.arange(9)
fig3, ax3 = plt.subplots()
ax3.set_ylabel('Number of people')
plt.bar(x3, val, color = 'red', width = 0.8)
for i in range(len(val)):
    plt.text(x = i-0.25, y = val[i]+0.1, s = val[i], size = 10)

plt.xticks(x3, ("Manu", "Finance", "Health", "Retail", "Property", "IT", "Enter
plt.show()
```



In [ ]:

```
df["wealth_segement"].value_counts()
```

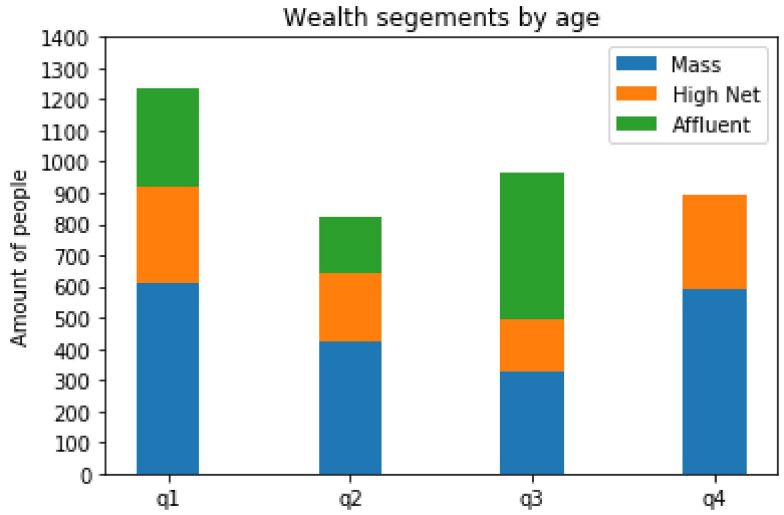
In [29]:

```
#split into 3: M (Mass), H(High), A(Affluent)
jjj = 1
wsm = {"q1":[], "q2":[], "q3":[], "q4":[]}
wsh = {"q1":[], "q2":[], "q3":[], "q4":[]}
wsa = {"q1":[], "q2":[], "q3":[], "q4":[]}

for each in df['wealth_segement']:
    temp = int(df['Age'][jjj])
    if each[0] == "M" and temp != 0:
        if(temp <= fq):
            wsm["q1"].append(temp)
        elif(fq < temp and temp <= sq):
            wsm["q2"].append(temp)
        elif(sq < temp and temp <= tq):
            wsm["q3"].append(temp)
        elif(tq < temp):
            wsm["q4"].append(temp)
    elif each[0] == "H" and temp != 0:
        if(temp <= fq):
            wsh["q1"].append(temp)
        elif(fq < temp and temp <= sq):
            wsh["q2"].append(temp)
        elif(sq < temp and temp <= tq):
            wsh["q3"].append(temp)
        elif(tq < temp):
            wsh["q4"].append(temp)
    elif each[0] == "A" and temp != 0:
        if(temp <= fq):
            wsa["q1"].append(temp)
        elif(fq < temp and temp <= sq):
            wsa["q2"].append(temp)
        elif(sq < temp and temp <= tq):
            wsa["q3"].append(temp)
        elif(tq < temp):
            wsa["q3"].append(temp)
    jjj += 1
print(len(wsm["q1"]), len(wsm["q2"]), len(wsm["q3"]), len(wsm["q4"]))
print(len(wsh["q1"]), len(wsh["q2"]), len(wsh["q3"]), len(wsh["q4"]))
print(len(wsa["q1"]), len(wsa["q2"]), len(wsa["q3"]), len(wsa["q4"]))
```

```
613 425 327 589
307 217 170 302
312 180 470 0
```

In [30]:

```
N = 4
wsmtp = [len(wsm["q1"]), len(wsm["q2"]), len(wsm["q3"]), len(wsm["q4"])]
wshtp = [len(wsh["q1"]), len(wsh["q2"]), len(wsh["q3"]), len(wsh["q4"])]
wsatp = [len(wsa["q1"]), len(wsa["q2"]), len(wsa["q3"]), len(wsa["q4"])]

bars = np.add(wsmtp, wshtp).tolist()
r = [0,1,2,3,4]

ind    = np.arange(N) # the x locations for the groups
width = 0.35       # the width of the bars: can also be len(x) sequence

p1 = plt.bar(ind, wsmtp, width)
p2 = plt.bar(ind, wshtp, width,  bottom=wsmtp)
p3 = plt.bar(ind, wsatp, width,  bottom=bars)

plt.ylabel('Amount of people')
plt.title('Wealth segements by age')
plt.xticks(ind, ('q1', 'q2', 'q3', 'q4'))
plt.yticks(np.arange(0, 1500, 100))
plt.legend((p1[0], p2[0], p3[0]), ('Mass', 'High Net', 'Affluent'))

plt.show()
```



In [31]:

```
df['owns_car'].value_counts()
```

Out[31]:

```
Yes    2024
No     1976
Name: owns_car, dtype: int64
```

In [ ]:

```

```