

```
import os
```

## Section 2: Data Health Review

In [2]:

os.chdir(r'H:\IDMA')  
os.getcwd()

Out[2]:

'H:\\IDMA'

In [3]:

import pandas as pd  
df=pd.read\_excel('Case\_Study.xlsx')  
df

Out[3]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	NaN	Variables	Description
1	NaN	gender	Gender of Customer
2	NaN	age	Age of Customer
3	NaN	gross_income	Annual income
4	NaN	segment	Segment as specified by the bank
5	NaN	num_credit_cards	Number of credit cards issued
6	NaN	active_inactive_start	Customer inactivity flag at the beginning of L...
7	NaN	active_inactive_end	Customer inactivity flag at the end of the per...
8	NaN	num_products	Total number of financial products that the cu...
9	NaN	num_loans	Total number loans disbursed to the customer
10	NaN	duration	Number of days since customer

In [4]:

ds=pd.read\_excel('Case\_Study.xlsx',sheet\_name='Data')  
ds

Out[4]:

	gender	age	gross_income	segment	num_credit_cards	active_inactive_start	active_inactive_end	num_products	num_loans	duration
0	M	21.0	79070.91	Individuals	0.0	I	0.0	1.0	0.0	1035.0
1	M	23.0	178270.68	College_Graduated	0.0	I	0.0	1.0	0.0	1097.0
2	M	24.0	31243.56	College_Graduated	0.0	I	1.0	1.0	0.0	1866.0
3	M	24.0	130740.54	College_Graduated	0.0	I	0.0	1.0	0.0	1440.0
4	M	31.0	112975.17	Individuals	0.0	I	0.0	0.0	0.0	376.0
...	...	...	...	...	...	...	...	...	...	...
54025	M	24.0	47237.01	College_Graduated	0.0	I	1.0	1.0	0.0	1147.0
54026	F	46.0	106930.05	Individuals	1.0	A	1.0	8.0	0.0	6061.0
54027	M	24.0	185323.05	College_Graduated	0.0	I	0.0	0.0	0.0	1817.0
54028	F	27.0	65246.79	College_Graduated	0.0	I	0.0	1.0	0.0	1447.0
54029	M	20.0	180678.48	College_Graduated	0.0	A	0.0	1.0	0.0	366.0

54030 rows × 10 columns

In [5]:

ds.shape

Out[5]:

(54030, 10)

## 01: Type of Variables

In [6]:

ds.info()

Out[6]:

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 54030 entries, 0 to 54029  
Data columns (total 10 columns):  
# Column Non-Null Count Dtype   
... ...  
0 gender 54030 non-null object   
1 age 54030 non-null float64  
2 gross\_income 54030 non-null float64  
3 segment 54030 non-null object   
4 num\_credit\_cards 54030 non-null float64  
5 active\_inactive\_start 54030 non-null object   
6 active\_inactive\_end 54030 non-null float64  
7 num\_products 54030 non-null float64  
8 num\_loans 54030 non-null float64  
9 duration 54030 non-null float64  
dtypes: float64(7), object(3)  
memory usage: 4.1+ MB

## 02: Any Missing Values

In [7]:

ds.isnull().sum()

Out[7]:

gender 0  
age 0  
gross\_income 0  
segment 0  
num\_credit\_cards 0  
active\_inactive\_start 0  
active\_inactive\_end 0  
num\_products 0  
num\_loans 0  
duration 0  
dtype: int64

In [8]:

ds.columns

Out[8]:

Index(['gender', 'age', 'gross\_income', 'segment', 'num\_credit\_cards',  
 'active\_inactive\_start', 'active\_inactive\_end', 'num\_products',  
 'num\_loans', 'duration'],  
 dtype='object')

## 05: Duplicate Records

In [9]:

ds1= ds.drop\_duplicates()

Out[9]:

ds1

In [10]:

ds1.shape

Out[10]:

(52839, 10)

In [11]:

ds1

Out[11]:

	gender	age	gross_income	segment	num_credit_cards	active_inactive_start	active_inactive_end	num_products	num_loans	duration
0	M	21.0	79070.91	Individuals	0.0	I	0.0	1.0	0.0	1035.0
1	M	23.0	178270.68	College_Graduated	0.0	I	0.0	1.0	0.0	1097.0
2	M	24.0	31243.56	College_Graduated	0.0	I	1.0	1.0	0.0	1866.0
3	M	24.0	130740.54	College_Graduated	0.0	I	0.0	1.0	0.0	1440.0
4	M	31.0	112975.17	Individuals	0.0	I	0.0	0.0	0.0	376.0
...	...	...	...	...	...	...	...	...	...	...
54024	M	24.0	75405.60	College_Graduated	0.0	I	0.0	1.0	0.0	1073.0
54025	M	24.0	47237.01	College_Graduated	0.0	I	1.0	1.0	0.0	1147.0
54026	F	46.0	106930.05	Individuals	1.0	A	1.0	8.0	0.0	6061.0
54028	F	27.0	65246.79	College_Graduated	0.0	I	0.0	1.0	0.0	1447.0
54029	M	20.0	180678.48	College_Graduated	0.0	A	0.0	1.0	0.0	366.0

52839 rows × 10 columns

In [12]:

ds1['active\_inactive\_end'].value\_counts()

Out[12]:

0.0 28824  
1.0 24815  
Name: active\_inactive\_end, dtype: int64

In [13]:

ds1['active\_inactive\_start'].value\_counts()

Out[13]:

I 28335  
A 24584  
Name: active\_inactive\_start, dtype: int64

## 04: Cleaning of Variables

In [14]:

ds1['active\_inactive\_start'].replace(to\_replace=['I','A'],value=['Inactive','Active'],inplace=True)

Out[14]:

C:\Users\HOME\anaconda3\lib\site-packages\pandas\core\generic.py:6619: SettingWithCopyWarning:  
A value is trying to be set on a copy of a DataFrame  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
return self.\_update\_inplace(result)

In [15]:

ds1['active\_inactive\_end'].replace(to\_replace=(0,0,1,0),value=['Inactive','Active'],inplace=True)

Out[15]:

ds1

In [16]:

ds1

Out[16]:

	gender	age	gross_income	segment	num_credit_cards	active_inactive_start	active_inactive_end	num_products	num_loans	duration
0	M	21.0	79070.91	Individuals	0.0	Inactive	Inactive	1.0	0.0	1035.0
1	M	23.0	178270.68	College_Graduated	0.0	Inactive	Inactive	1.0	0.0	1097.0
2	M	24.0	31243.56	College_Graduated	0.0	Inactive	Active	1.0	0.0	1866.0
3	M	24.0	130740.54	College_Graduated	0.0	Inactive	Inactive	1.0	0.0	1440.0
4	M	31.0	112975.17	Individuals	0.0	Inactive	Inactive	0.0	0.0	376.0
...	...	...	...	...	...	...	...	...	...	...
54024	M	24.0	75405.60	College_Graduated	0.0	Inactive	Inactive	1.0	0.0	1073.0
54025	M	24.0	47237.01	College_Graduated	0.0	Inactive	Active	1.0	0.0	1147.0
54026	F	46.0	106930.05	Individuals	1.0	Active	Active	8.0	0.0	6061.0
54028	F	27.0	65246.79	College_Graduated	0.0	Inactive	Inactive	1.0	0.0	1447.0
54029	M	20.0	180678.48	College_Graduated	0.0	Active	Inactive	1.0	0.0	366.0

52839 rows × 10 columns

In [17]:


import seaborn as sns  
import matplotlib.pyplot as plt

Out[17]:

In [18]:

plt.figure(figsize=(10,8))  
f=sns.countplot(x=ds1['active\_inactive\_start']);  
for i in f.containers:  
 r.bar\_label(i)

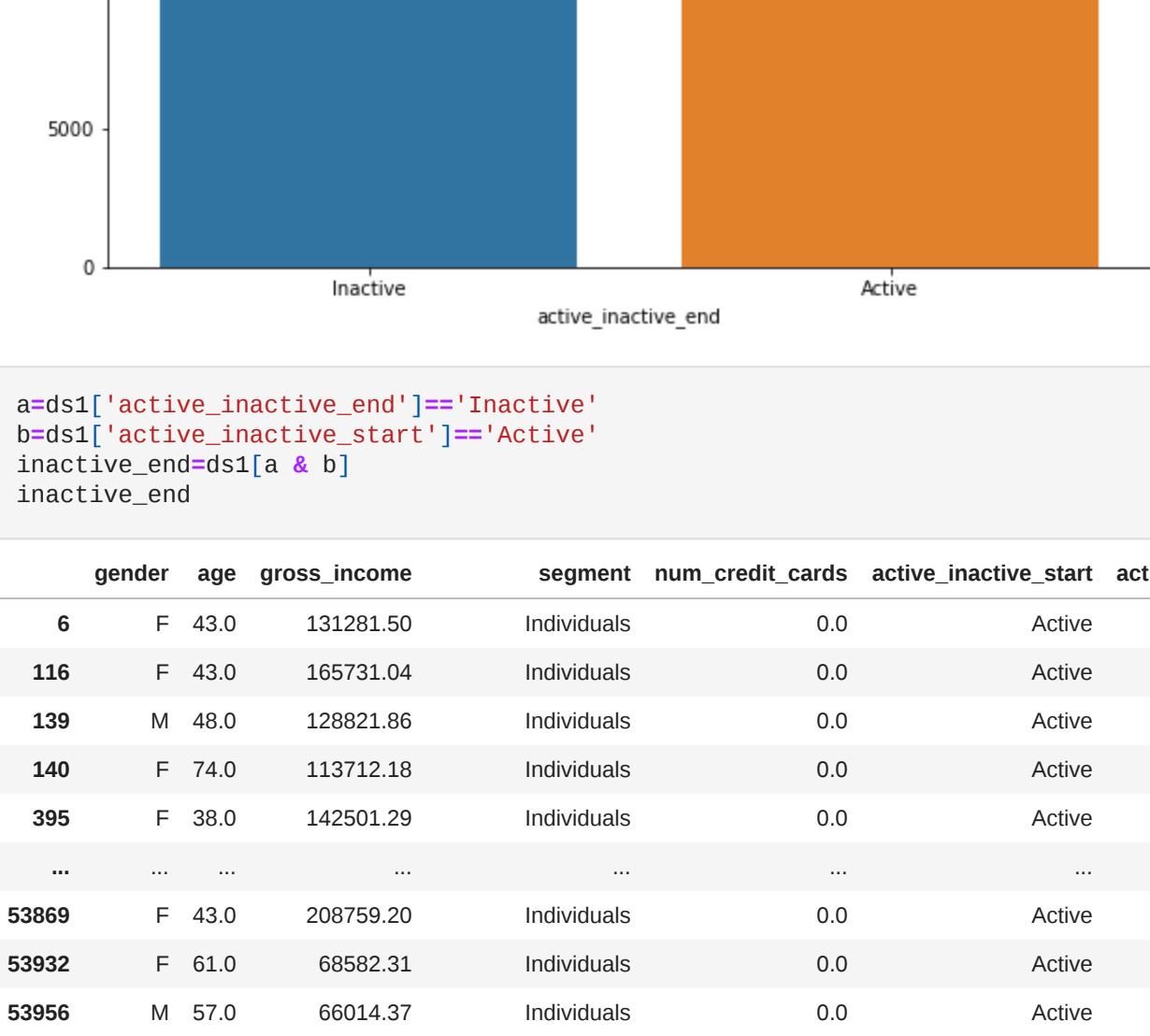
Out[18]:



In [19]:

plt.figure(figsize=(10,8))  
f=sns.countplot(x=ds1['active\_inactive\_end']);  
for i in f.containers:  
 r.bar\_label(i)

Out[19]:



In [20]:

a=ds1['active\_inactive\_end']=='Inactive'  
b=ds1['active\_inactive\_start']=='Active'  
inactive\_end=ds1[a & b]  
inactive\_end

Out[20]:

	gender	age	gross_income	segment	num_credit_cards	active_inactive_start	active_inactive_end	num_products	num_loans	duration
6	F	43.0	131281.50	Individuals	0.0	Active	Inactive	1.0	0.0	5022.0
116	F	43.0	165731.04	Individuals	0.0	Active	Inactive	3.0	0.0	5585.0
139	M	48.0	128821.86	Individuals	0.0	Active	Inactive	2.0	0.0	4996.0
140	F	74.0	113712.18	Individuals	0.0	Active	Inactive	1.0	0.0	7256.0
395	F	38.0	142501.29	Individuals	0.0	Active	Inactive	1.0	0.0	4241.0
...	...	...	...	...	...	...	...	...	...	...
53869	F	43.0	208759.20	Individuals	0.0	Active	Inactive	1.0	0.0	5389.0
53932	F	61.0	68582.31	Individuals	0.0	Active	Inactive	0.0	0.0	4650.0
53956	M	57.0	66014.37	Individuals	0.0	Active	Inactive	2.0	0.0	5487.0
53968	F	22.0	70219.17	College_Graduated	0.0	Active	Inactive	1.0	0.0	737.0
54029	M	20.0	180678.48	College_Graduated	0.0	Active	Inactive	1.0	0.0	366.0

2068 rows × 10 columns

In [21]:

c=ds1['active\_inactive\_start']=='Inactive'  
d=ds1['active\_inactive\_end']=='Active'  
active\_end=ds1[c & d]  
active\_end

Out[21]:

	gender	age	gross_income	segment	num_credit_cards	active_inactive_start	active_inactive_end	num_products	num_loans	duration
2	M	24.0	31243.56	College_Graduated	0.0	Inactive	Active	1.0	0.0	1866.0
9	M	26.0	27311.31	College_Graduated	0.0	Inactive	Active	1.0	0.0	1458.0
97	F	84.0	71984.22	Individuals	0.0	Inactive	Active	1.0	0.0	2974.0
119	M	49.0	314065.11	Individuals	0.0	Inactive	Active	2.0	0.0	6418.0
165	M	25.0	50303.79	College_Graduated	0.0	Inactive	Active	1.0	0.0	1133.0
...	...	...	...	...	...	...	...	...	...	...
53955	F	45.0	183191.15	Individuals	0.0	Inactive	Active	1.0	0.0	3582.0
53979	M	44.0	136109.01	Individuals	0.0	Inactive	Active	1.0	0.0	1070.0
53982	M	45.0	255592.53	Individuals	0.0	Inactive	Active	2.0	0.0	5835.0
54013	F	35.0	41178.45	Individuals	0.0	Inactive	Active	2.0	0.0	1189.0
54025	M	24.0	47237.01	College_Graduated	0.0	Inactive	Active	1.0	0.0	1147.0

2379 rows × 10 columns

In [22]:

inactive\_end['segment'].value\_counts(normalize=True)

Out[22]:

Individuals 0.722921  
College\_Graduated 0.262089  
VIP 0.014990  
Name: segment, dtype: float64

In [23]:

inactive\_end['gender'].value\_counts(normalize=True)

Out[23]:

F 0.687834  
M 0.302166  
Name: gender, dtype: float64

In [24]:

active\_end['gender'].value\_counts(normalize=True)

Out[24]:

F 0.532156  
M 0.467844  
Name: gender, dtype: float64

In [25]:

active\_end['segment'].value\_counts(normalize=True)

Out[25]:

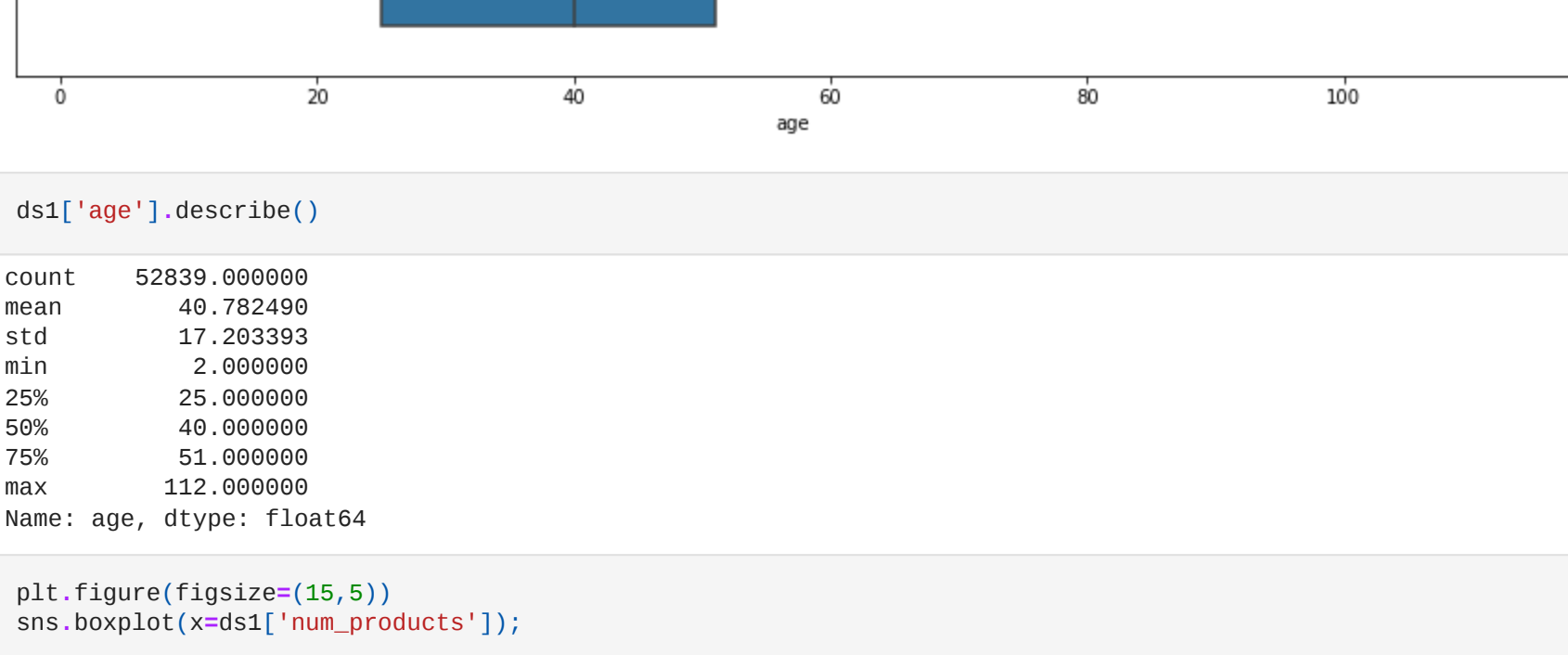
Individuals 0.664456  
College\_Graduated 0.394704  
VIP 0.000841  
Name: segment, dtype: float64

## 03: Outliers of Variables

In [26]:

plt.figure(figsize=(15,5))  
sns.boxplot(x=ds1['age']);

Out[26]:



In [27]:

ds1['age'].describe()

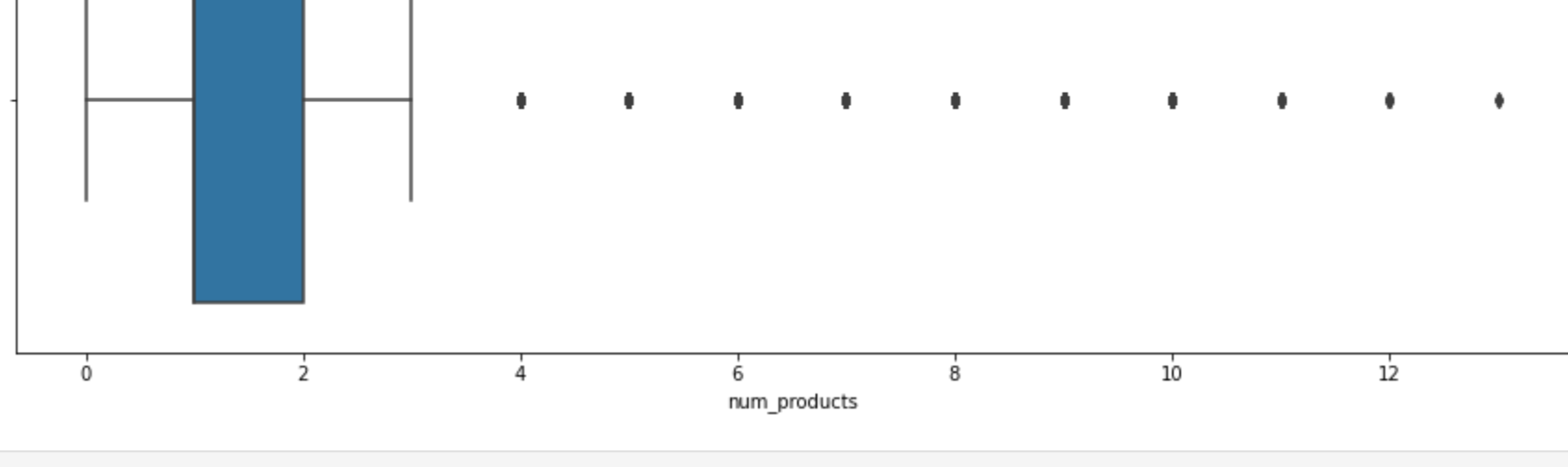
Out[27]:

count 52839.000000  
mean 40.782498  
std 17.263393  
min 2.000000  
25% 25.000000  
50% 40.000000  
75% 51.000000  
max 112.000000  
Name: age, dtype: float64

In [28]:

plt.figure(figsize=(15,5))  
sns.boxplot(x=ds1['num\_products']);

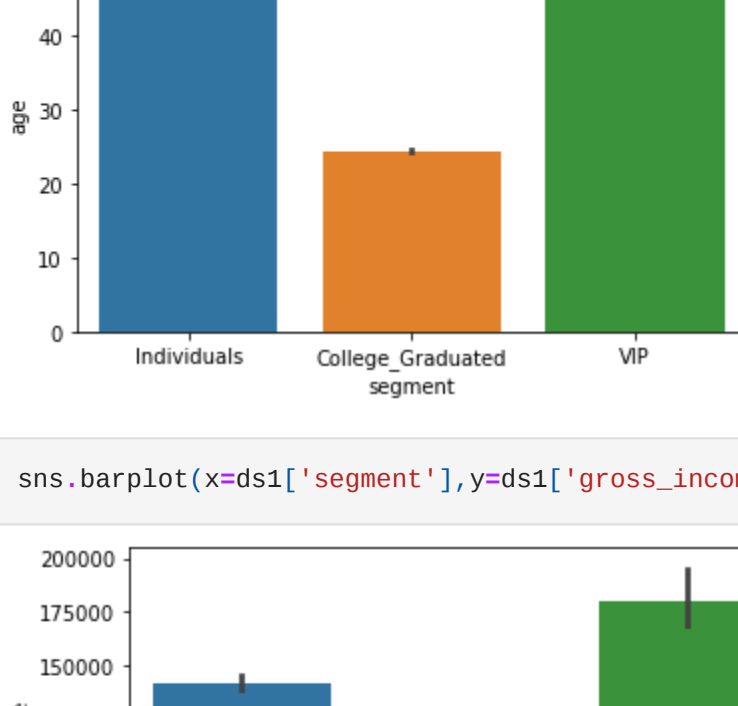
Out[28]:



In [29]:

sns.barplot(x=ds1['segment'],y=ds1['age']);

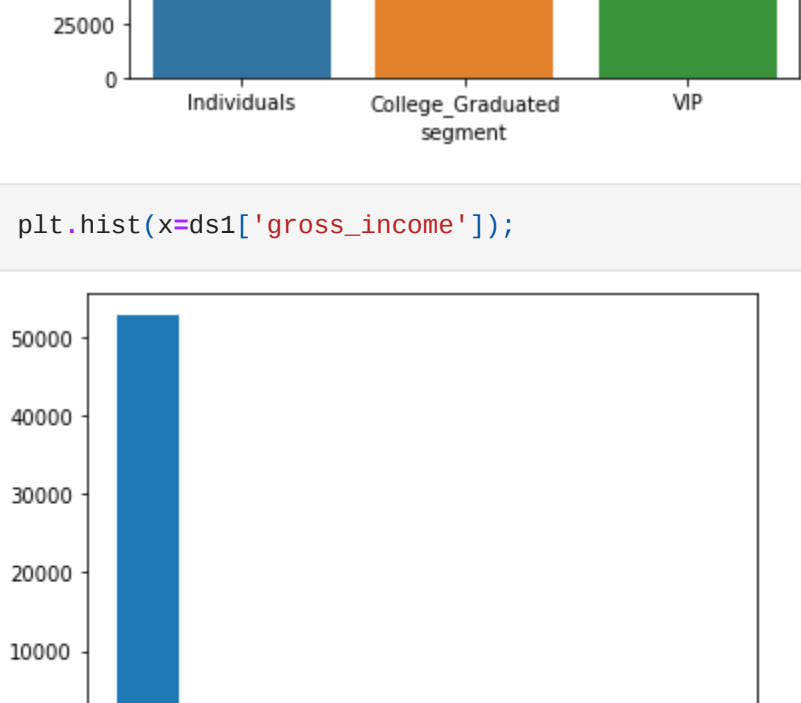
Out[29]:



In [30]:

sns.barplot(x=ds1['segment'],y=ds1['gross\_income']);


Out[30]:



In [31]:

plt.hist(x=ds1['gross\_income']);

Out[31]:



## Scatter Plot

In [35]:

plt.figure(figsize=(20,8))  
plt.scatter(data=ds1,x='age',y='gross\_income');

Out[35]:



In [38]:

plt.figure(figsize=(10,7))  
plt.scatter(data=ds1,x='segment',y='gross\_income');

Out[38]:

