Statistics Assignment: Exploratory Data Analysis (EDA)

Introduction

In statistics, exploratory data analysis (EDA) is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables. Here We perform Exploratory Data Analysis (EDA) on The data which is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (variable y).

Data Set Information:

This data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to assess if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

Attribute Information:

Bank client data:

1) Age (numeric)

2) job : type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')

3) Marital : marital status (categorical: 'divorced', 'married', 'single', 'unknown' ; note: 'divorced' means divorced or widowed)

4) Education (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree','unknown')

5) Default: has credit in default? (categorical: 'no', 'yes', 'unknown')

6) Housing: has a housing loan? (categorical: 'no', 'yes', 'unknown')

7) Loan: has personal loan? (categorical: 'no', 'yes', 'unknown'

8) Contact: contact communication type (categorical:'cellular','telephone')

9) Month: last contact month of year (categorical: 'jan', 'feb', 'mar',…, 'nov', 'dec')

10) Dayofweek: last contact day of the week (categorical:'mon','tue','wed','thu','fri')

11) Duration: last contact duration, in seconds (numeric). Important

note: this attribute highly affects the output target (e.g., ifduration=0 then y='no'). Yet, the duration is not known before a callis performed. Also, after the end of the call y is obviously known.

Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Other attributes:

12) Campaign: number of contacts performed during this campaign and forthis client (numeric, includes last contact)

13) Pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

14) Previous: number of contacts performed before this campaign and for this client (numeric)

15) Poutcome: outcome of the previous marketing campaign (categorical:'failure','nonexistent','success')

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df = pd.read_excel('Data.xlsx')
print(df)
```

```
             banking marketing Unnamed: 1                     Unnamed: 2  \
0      customer id and age.         NaN  Customer salary and balance.
1                customerid         age                         salary
2                         1          58                         100000
3                         2          44                          60000
4                         3          33                         120000
...                     ...         ...                            ...
45208                 45207          51                          60000
45209                 45208          71                          55000
45210                 45209          72                          55000
45211                 45210          57                          20000
45212                 45211          37                         120000

      Unnamed: 3                                       Unnamed: 4  \
0            NaN  Customer marital status and job with education...
1        balance                                          marital
2           2143                                          married
3             29                                           single
4              2                                          married
...          ...                                              ...
45208        825                                          married
45209       1729                                         divorced
45210       5715                                          married
45211        668                                          married
45212       2971                                          married

                    Unnamed: 5                               Unnamed: 6  \
0                          NaN  particular customer before targeted or not
1                       jobedu                                 targeted
2         management,tertiary                                      yes
3         technician,secondary                                     yes
4      entrepreneur,secondary                                      yes
...                        ...                                      ...
45208      technician,tertiary                                     yes
45209          retired,primary                                     yes
45210        retired,secondary                                     yes
45211    blue-collar,secondary                                     yes
45212   entrepreneur,secondary                                     yes

      Unnamed: 7                           Unnamed: 8 Unnamed: 9   Unnamed: 10
\
0            NaN  Loan types: loans or housing loans        NaN  Contact type
1        default                              housing       loan       contact
2             no                                  yes         no       unknown
3             no                                  yes         no       unknown
4             no                                  yes        yes       unknown
...          ...                                  ...        ...           ...
45208         no                                   no         no      cellular
45209         no                                   no         no      cellular
45210         no                                   no         no      cellular
45211         no                                   no         no     telephone
45212         no                                   no         no      cellular

      Unnamed: 11        Unnamed: 12          Unnamed: 13 Unnamed: 14  \
```

```
0              NaN  month of contact      duration of call         NaN
1              day             month              duration    campaign
2                5         may, 2017               261 sec           1
3                5         may, 2017               151 sec           1
4                5         may, 2017                76 sec           1
...            ...               ...                   ...         ...
45208           17         nov, 2017  16.2833333333333 min           3
45209           17         nov, 2017               7.6 min           2
45210           17         nov, 2017  18.7833333333333 min           5
45211           17         nov, 2017  8.46666666666667 min           4
45212           17         nov, 2017  6.01666666666667 min           2

      Unnamed: 15 Unnamed: 16               Unnamed: 17  \
0             NaN         NaN  outcome of previous contact
1           pdays    previous                  poutcome
2              -1           0                   unknown
3              -1           0                   unknown
4              -1           0                   unknown
...           ...         ...                       ...
45208          -1           0                   unknown
45209          -1           0                   unknown
45210         184           3                   success
45211          -1           0                   unknown
45212         188          11                     other

                                Unnamed: 18
0      response of customer after call happned
1                                  response
2                                        no
3                                        no
4                                        no
...                                     ...
45208                                   yes
45209                                   yes
45210                                   yes
45211                                    no
45212                                    no

[45213 rows x 19 columns]
```

In [3]:
```
df = pd.read_excel('Data.xlsx',skiprows = 2)
print(df)
```

```
       customerid   age  salary  balance   marital                  jobedu  \
0               1  58.0  100000     2143   married   management,tertiary
1               2  44.0   60000       29    single  technician,secondary
2               3  33.0  120000        2   married entrepreneur,secondary
3               4  47.0   20000     1506   married     blue-collar,unknown
4               5  33.0       0        1    single         unknown,unknown
...           ...   ...     ...      ...       ...                     ...
45206       45207  51.0   60000      825   married     technician,tertiary
45207       45208  71.0   55000     1729  divorced         retired,primary
45208       45209  72.0   55000     5715   married       retired,secondary
45209       45210  57.0   20000      668   married   blue-collar,secondary
45210       45211  37.0  120000     2971   married entrepreneur,secondary

      targeted default housing loan    contact  day      month  \
0          yes      no     yes   no    unknown    5  may, 2017
1          yes      no     yes   no    unknown    5  may, 2017
2          yes      no     yes  yes    unknown    5  may, 2017
3           no      no     yes   no    unknown    5  may, 2017
4           no      no      no   no    unknown    5  may, 2017
...        ...     ...     ...  ...        ...  ...        ...
45206      yes      no      no   no   cellular   17  nov, 2017
45207      yes      no      no   no   cellular   17  nov, 2017
45208      yes      no      no   no   cellular   17  nov, 2017
45209      yes      no      no   no  telephone   17  nov, 2017
45210      yes      no      no   no   cellular   17  nov, 2017

                   duration  campaign  pdays  previous poutcome response
0                   261 sec         1     -1         0  unknown       no
1                   151 sec         1     -1         0  unknown       no
2                    76 sec         1     -1         0  unknown       no
3                    92 sec         1     -1         0  unknown       no
4                   198 sec         1     -1         0  unknown       no
...                     ...       ...    ...       ...      ...      ...
45206  16.2833333333333 min         3     -1         0  unknown      yes
45207               7.6 min         2     -1         0  unknown      yes
45208  18.7833333333333 min         5    184         3  success      yes
45209  8.46666666666667 min         4     -1         0  unknown       no
45210  6.01666666666667 min         2    188        11    other       no

[45211 rows x 19 columns]
```

In [4]:
```
df.head()
```

Out[4]:

	customerid	age	salary	balance	marital	jobedu	targeted	default	housing	lo
0	1	58.0	100000	2143	married	management,tertiary	yes	no	yes	
1	2	44.0	60000	29	single	technician,secondary	yes	no	yes	
2	3	33.0	120000	2	married	entrepreneur,secondary	yes	no	yes	y
3	4	47.0	20000	1506	married	blue-collar,unknown	no	no	yes	
4	5	33.0	0	1	single	unknown,unknown	no	no	no	

In [5]:
```
df.shape
```

Out[5]: (45211, 19)

In [6]:
```
df.describe()
```

Out[6]:

	customerid	age	balance	day	campaign	pdays	p
count	45211.000000	45191.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211
mean	22606.000000	40.935651	1362.272058	15.806419	2.763841	40.197828	C
std	13051.435847	10.619198	3044.765829	8.322476	3.098021	100.128746	2
min	1.000000	18.000000	-8019.000000	1.000000	1.000000	-1.000000	C
25%	11303.500000	33.000000	72.000000	8.000000	1.000000	-1.000000	C
50%	22606.000000	39.000000	448.000000	16.000000	2.000000	-1.000000	C
75%	33908.500000	48.000000	1428.000000	21.000000	3.000000	-1.000000	C
max	45211.000000	95.000000	102127.000000	31.000000	63.000000	871.000000	275

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 19 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   customerid  45211 non-null  int64  
 1   age         45191 non-null  float64
 2   salary      45211 non-null  object 
 3   balance     45211 non-null  int64  
 4   marital     45211 non-null  object 
 5   jobedu      45211 non-null  object 
 6   targeted    45211 non-null  object 
 7   default     45211 non-null  object 
 8   housing     45211 non-null  object 
 9   loan        45211 non-null  object 
 10  contact     45211 non-null  object 
 11  day         45211 non-null  int64  
 12  month       45161 non-null  object 
 13  duration    45211 non-null  object 
 14  campaign    45211 non-null  int64  
 15  pdays       45211 non-null  int64  
 16  previous    45211 non-null  int64  
 17  poutcome    45211 non-null  object 
 18  response    45181 non-null  object 
dtypes: float64(1), int64(6), object(12)
memory usage: 6.6+ MB
```

In [8]:
```
# Drop the customer id as it is of no use.
df.drop('customerid', axis = 1, inplace = True)

#Extract job  & Education in newly from "jobedu" column.
df['job']= df["jobedu"].apply(lambda x: x.split(",")[0])
df['education']= df["jobedu"].apply(lambda x: x.split(",")[1])

# Drop the "jobedu" column from the dataframe.
df.drop('jobedu', axis = 1, inplace = True)
```

In [9]:
```
df
```

Out[9]:

	age	salary	balance	marital	targeted	default	housing	loan	contact	day	month	
0	58.0	100000	2143	married	yes	no	yes	no	unknown	5	may, 2017	
1	44.0	60000	29	single	yes	no	yes	no	unknown	5	may, 2017	
2	33.0	120000	2	married	yes	no	yes	yes	unknown	5	may, 2017	
3	47.0	20000	1506	married	no	no	yes	no	unknown	5	may, 2017	
4	33.0	0	1	single	no	no	no	no	unknown	5	may, 2017	
...	...	...	...	...	...	...	...	...	...	...	...	
45206	51.0	60000	825	married	yes	no	no	no	cellular	17	nov, 2017	1(
45207	71.0	55000	1729	divorced	yes	no	no	no	cellular	17	nov, 2017	
45208	72.0	55000	5715	married	yes	no	no	no	cellular	17	nov, 2017	1:
45209	57.0	20000	668	married	yes	no	no	no	telephone	17	nov, 2017	8.
45210	37.0	120000	2971	married	yes	no	no	no	cellular	17	nov, 2017	6.

45211 rows × 19 columns

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
age          20
salary        0
balance       0
marital       0
targeted      0
default       0
housing       0
loan          0
contact       0
day           0
month        50
duration      0
campaign      0
pdays         0
previous      0
poutcome      0
response     30
job           0
education     0
dtype: int64
```

In [11]:
```
# Dropping the records with age missing in data dataframe.
df = df[~df.age.isnull()].copy()

# Checking the missing values in the dataset.
df.isnull().sum()
```

Out[11]:
```
age           0
salary        0
balance       0
marital       0
targeted      0
default       0
housing       0
loan          0
contact       0
day           0
month        50
duration      0
campaign      0
pdays         0
previous      0
poutcome      0
response     30
job           0
education     0
dtype: int64
```

In [12]:
```
# Find the mode of month in data
month_mode = df.month.mode()[0]

# Fill the missing values with mode value of month in data.
df.month.fillna(month_mode, inplace = True)

# Let's see the null values in the month column.
df.month.isnull().sum()
```

Out[12]:
```
0
```

In [13]:
```
#drop the records with response missing in data.
df = df[~df.response.isnull()].copy()
# Calculate the missing values in each column of data frame
df.isnull().sum()
```

Out[13]:
```
age          0
salary       0
balance      0
marital      0
targeted     0
default      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
response     0
job          0
education    0
dtype: int64
```

Univariate Analysis:-

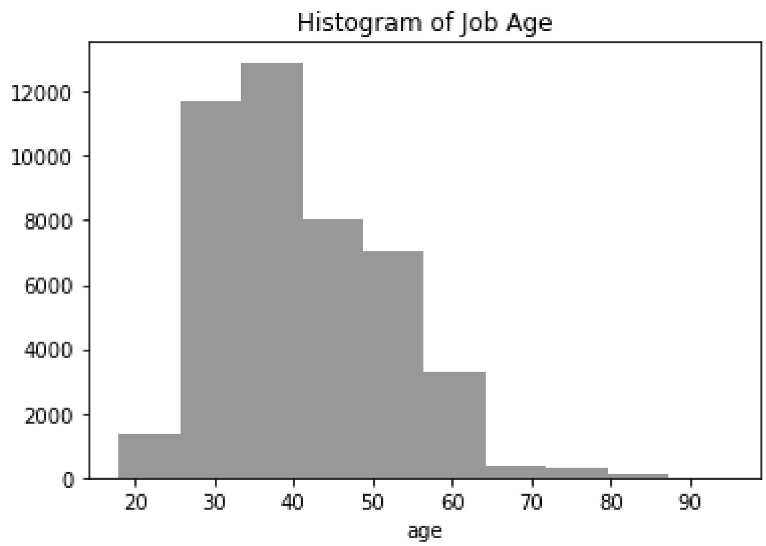
Categorical Unordered Univariate Analysis:

1. Histogram

In [14]:
```
sns.distplot(df['age'],kde=False,color='black',bins=10)
plt.title("Histogram of Job Age")
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future v
ersion. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```

Out[14]:
```
Text(0.5, 1.0, 'Histogram of Job Age')
```

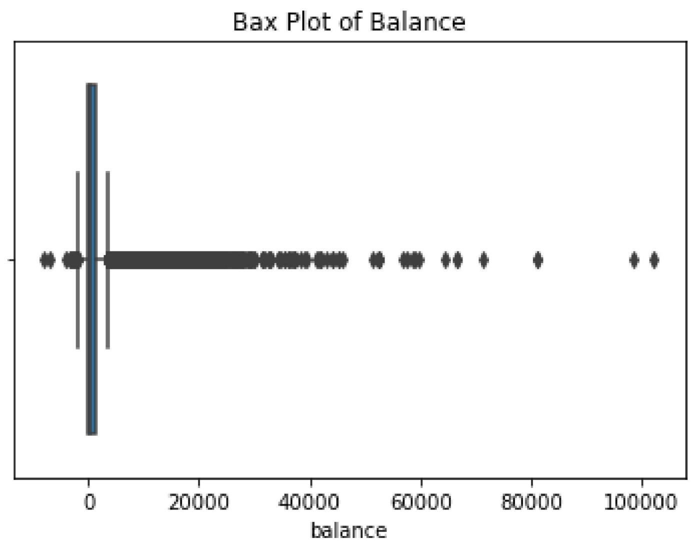


2. Box plot

In [15]:
```
sns.boxplot(df['balance'])
plt.title("Bax Plot of Balance")
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[15]:
```
Text(0.5, 1.0, 'Bax Plot of Balance')
```

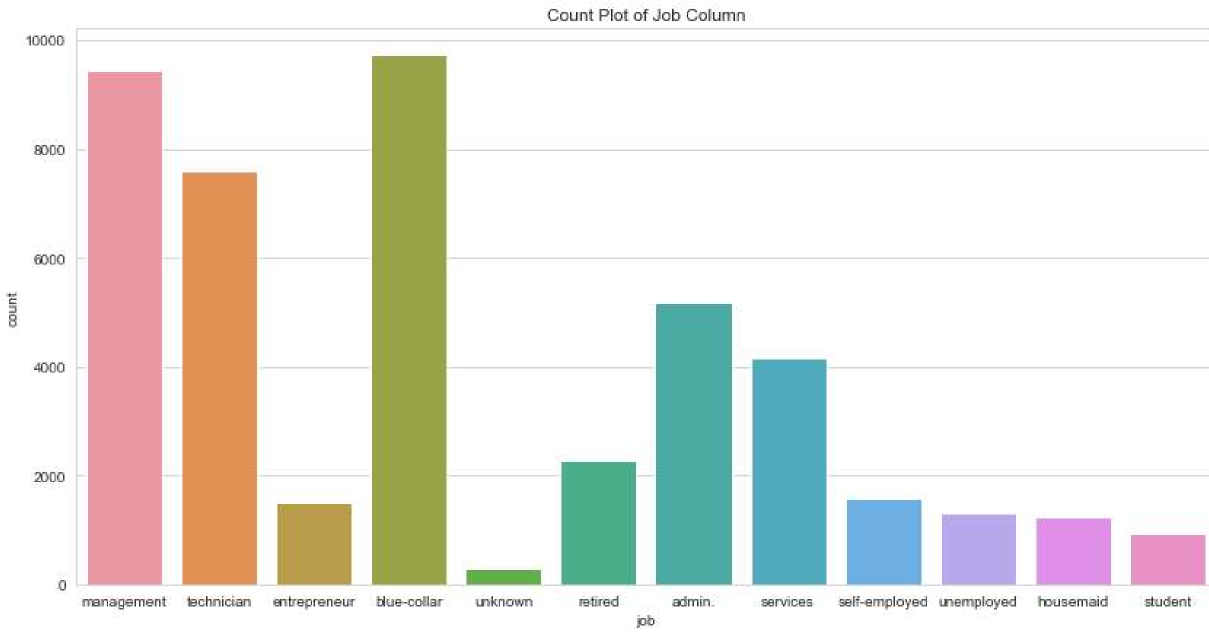


3. Count plot

In [16]:
```
sns.set_style("whitegrid")
plt.figure(figsize=(14,7))
sns.countplot(df['job'])
plt.title("Count Plot of Job Column")
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[16]:
```
Text(0.5, 1.0, 'Count Plot of Job Column')
```



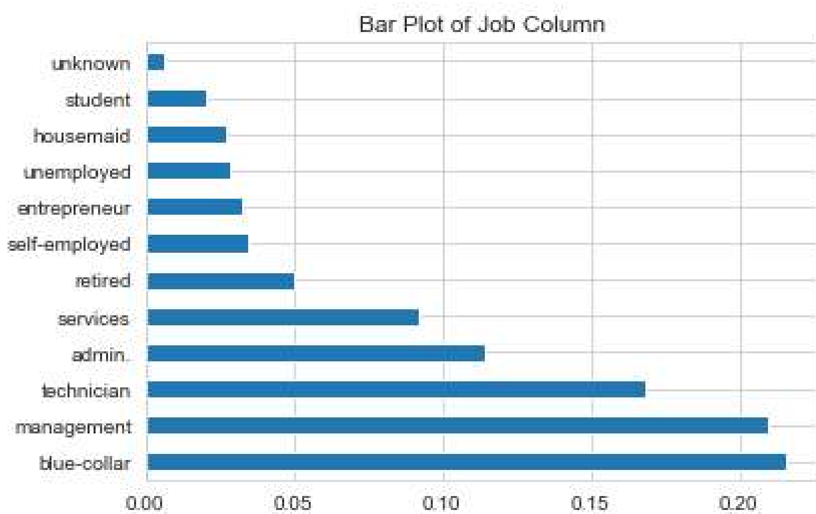
Job: from the visuals above, we can conclude that people with management jobs took part the most in the campaign.

4. Bar plot

In [17]:
```
# Let's calculate the percentage of each job status category.
df.job.value_counts(normalize=True)
```

Out[17]:
```
blue-collar      0.215274
management       0.209273
technician       0.168043
admin.           0.114369
services         0.091849
retired          0.050087
self-employed    0.034853
entrepreneur     0.032860
unemployed       0.028830
housemaid        0.027413
student          0.020770
unknown          0.006377
Name: job, dtype: float64
```

In [18]:
```
#plot the bar graph of percentage job categories
df.job.value_counts(normalize=True).plot.barh()
plt.title("Bar Plot of Job Column")
plt.show()
```



By the above bar plot, we can infer that the data set contains more number of blue-collar workers compared to other categories.

5. Heatmap

In [19]:
```
plt.figure(figsize=(14,7))
#sns.heatmap(data=df, annot=True)
cor = df.corr()
sns.heatmap(cor, annot=True)
plt.title("Heatmap")
plt.show()
```

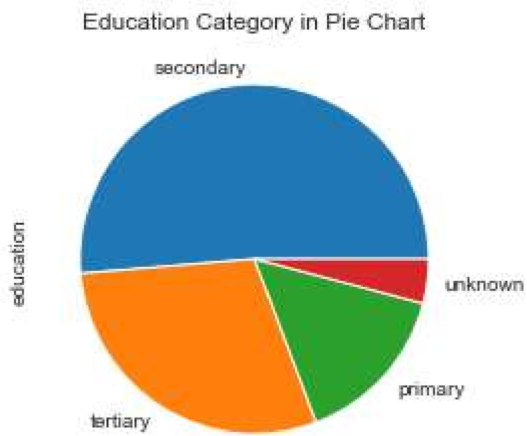
We can also use heat maps to visualize the correlation between the numerical values. It is quite evident from the graph that pdays and previous are highly correlated

Categorical Ordered Univariate Analysis:

In [20]:
```
#calculate the percentage of each education category.
df.education.value_counts(normalize=True)
```

Out[20]:
```
secondary    0.513275
tertiary     0.294192
primary      0.151436
unknown      0.041097
Name: education, dtype: float64
```

In [21]:
```
#plot the pie chart of education categories
df.education.value_counts(normalize=True).plot.pie()
plt.title("Education Category in Pie Chart")
plt.show()
```



By the above analysis, we can infer that the data set has a large number of them belongs to secondary education after that tertiary and next primary. Also, a very small percentage of them have been unknown

Bivariate Analysis:-

a) Numeric-Numeric Analysis:

Scatter Plot

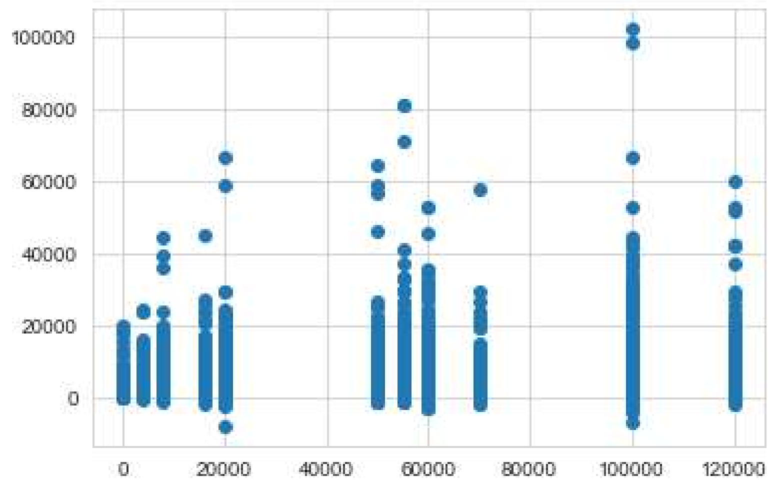
In [31]:
```
l=[]
for i in range(0,len(df.salary)):
    if i in df.index:
        a=df.salary[i]
        if a=='?':
            l.append(i)
```

In [32]:
```
for i in l:
    df.salary[i]=0
```

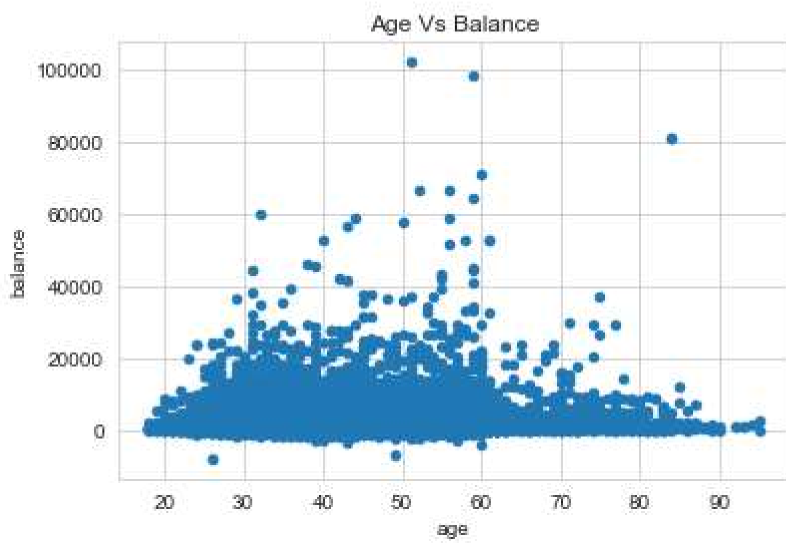
```
<ipython-input-32-34af87b7247d>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df.salary[i]=0
```

In [33]:
```
#df.salary=df.salary.replace({'?':0},regex=True)
df.salary=df.salary.astype(float)
```

In [34]:
```
#plot the scatter plot of balance and salary variable in data
plt.scatter(df['salary'], df['balance'])
plt.show()
```

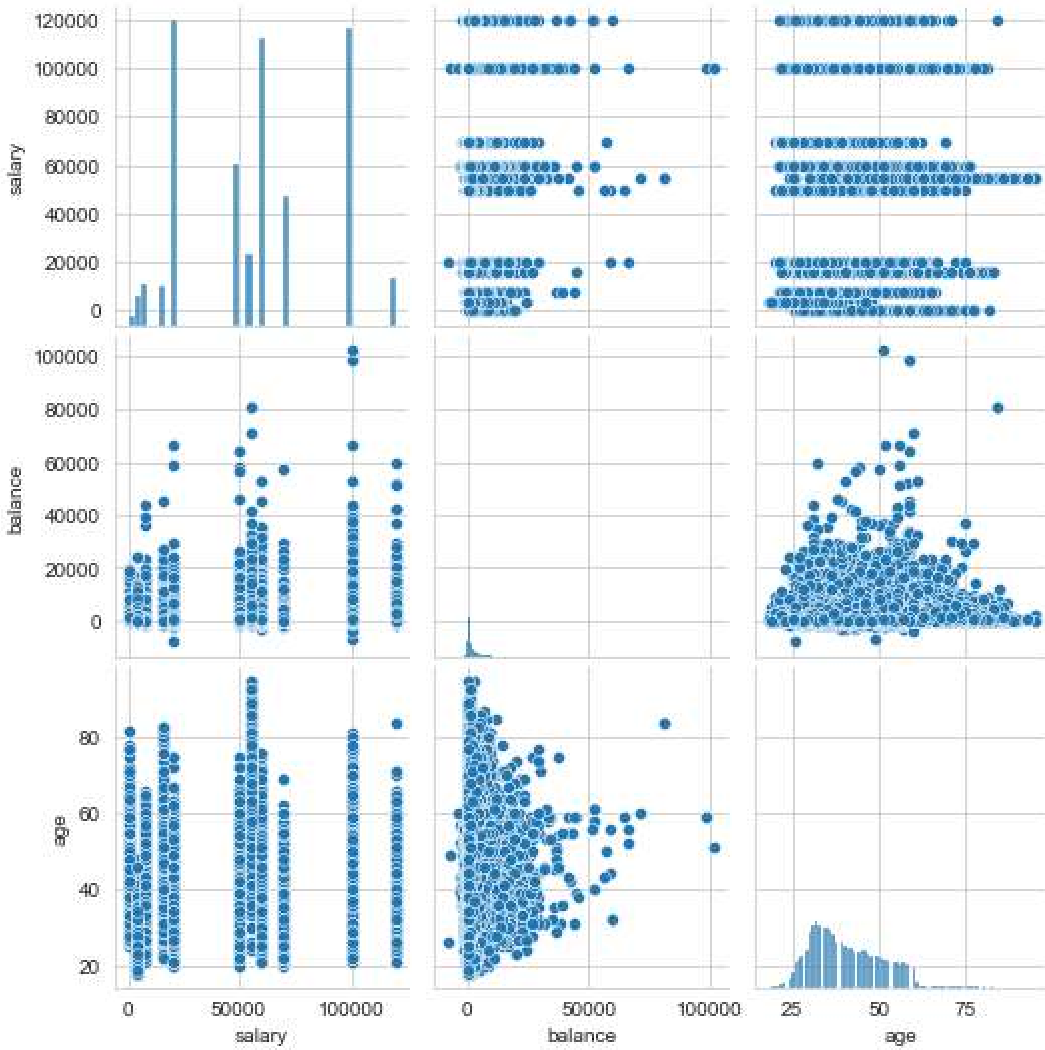


In [29]:
```
#plot the scatter plot of balance and age variable in data
df.plot.scatter(x="age",y="balance")
plt.title("Age Vs Balance")
plt.show()
```



Pair Plot

In [35]:
```
#plot the pair plot of salary, balance and age in data dataframe.
sns.pairplot(data=df, vars=['salary','balance','age'])
plt.show()
```



Correlation Matrix

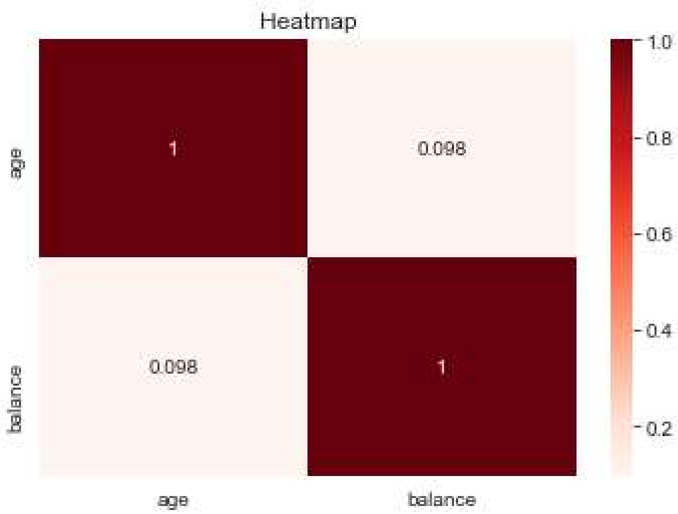
In [31]:
```
# Creating a matrix using age, salry, balance as rows and columns
df[['age','salary','balance']].corr()
```

Out[31]:

	age	balance
age	1.00000	0.09771
balance	0.09771	1.00000

Heatmap

In [32]:
```
#plot the correlation matrix of salary, balance and age in data dataframe.
sns.heatmap(df[['age','salary','balance']].corr(), annot=True, cmap = 'Reds')
plt.title("Heatmap")
plt.show()
```



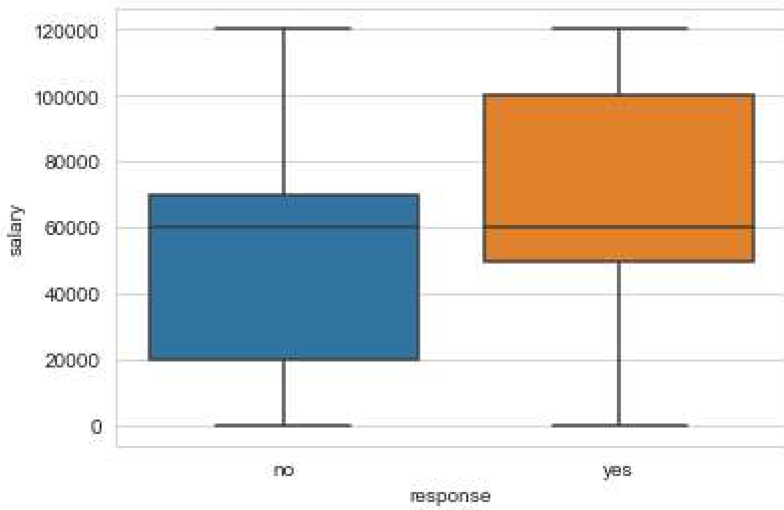
b) Numeric - Categorical Analysis

In [36]:
```
#groupby the response to find the mean of the salary with response no & yes separ
df.groupby('response')['salary'].mean()
```

Out[36]:
```
response
no     56769.510482
yes    58780.510880
Name: salary, dtype: float64
```

In [37]:
```
#plot the box plot of salary for yes & no responses.
sns.boxplot(df.response, df.salary)
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



As we can see, when we plot the Box Plot, it paints a very different picture compared to mean and median. The IQR for customers who gave a positive response is on the higher salary side.

This is how we analyze Numeric-Categorical variables, we use mean, median, and Box Plots to draw some sort of conclusions.

c) Categorical — Categorical Analysis

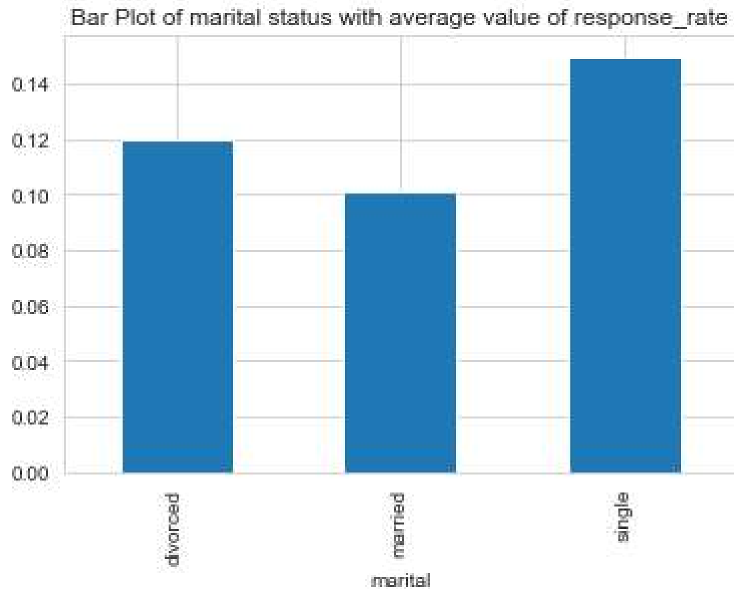
Since our target variable/column is the Response rate, we'll see how the different categories like Education, Marital Status, etc., are associated with the Response column. So instead of 'Yes' and 'No' we will convert them into '1' and '0', by doing that we'll get the "Response Rate".

In [35]:
```
#create response_rate of numerical data type where response "yes"= 1, "no"= 0
df['response_rate'] = np.where(df.response=='yes',1,0)
df.response_rate.value_counts()
```

Out[35]:
```
0    39876
1     5285
Name: response_rate, dtype: int64
```

Let's see how the response rate varies for different categories in marital status.

In [36]:
```
#plot the bar graph of marital status with average value of response_rate
df.groupby('marital')['response_rate'].mean().plot.bar()
plt.title("Bar Plot of marital status with average value of response_rate")
plt.show()
```



By the above graph, we can infer that the positive response is more for Single status members in the data set. Similarly, we can plot the graphs for Loan vs Response rate, Housing Loans vs Response rate, etc

Conclusion:-

This is how we'll do Exploratory Data Analysis. Exploratory Data Analysis (EDA) helps us to look beyond the data. The more we explore the data, the more the insights we draw from it. As a data analyst, almost 80% of our time will be spent understanding data and solving various business problems through EDA.