Data Analytics Assignment

March 27, 2023

1 DA Assignment: Abhishek Akkewar

2 Importing Libraries

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

3 Loading Dataset

[2]:
```
# Loading Data
df =  pd.read_csv("Data_Set.csv")
```

4 Exploratory Data Analysis and Data Cleaning

[3]:
```
df.head(5)
```

[3]:
```
      Segment  Country    Product   Discount Band   Units Sold  \
0  Government   Canada  Carretera            None       1618.5
1  Government  Germany  Carretera            None       1321.0
2   Midmarket   France  Carretera            None       2178.0
3   Midmarket  Germany  Carretera            None        888.0
4   Midmarket   Mexico  Carretera            None       2470.0

   Manufacturing Price  Sale Price  Gross Sales  Discounts        Sales    \
0                $3.00      $20.00   $32,370.00        $-     $32,370.00
1                $3.00      $20.00   $26,420.00        $-     $26,420.00
2                $3.00      $15.00   $32,670.00        $-     $32,670.00
3                $3.00      $15.00   $13,320.00        $-     $13,320.00
4                $3.00      $15.00   $37,050.00        $-     $37,050.00

   Cost of Goods Sold      Profit        Date  Month Number  Month Name   \
0          $16,185.00  $16,185.00  01-01-2014             1     January
1          $13,210.00  $13,210.00  01-01-2014             1     January
2          $21,780.00  $10,890.00  01-06-2014             6        June
```

```
3           $8,880.00     $4,440.00   01-06-2014             6        June
4          $24,700.00    $12,350.00   01-06-2014             6        June

   Year
0  2014
1  2014
2  2014
3  2014
4  2014
```

[4]:
```
df.tail(5)
```

[4]:
```
              Segment                   Country    Product   Discount Band   \
695    Small Business                    France   Amarilla            High
696    Small Business                    Mexico   Amarilla            High
697        Government                    Mexico    Montana            High
698        Government                    Canada      Paseo            High
699  Channel Partners  United States of America        VTT            High

     Units Sold  Manufacturing Price   Sale Price     Gross Sales   \
695      2475.0              $260.00      $300.00    $7,42,500.00
696       546.0              $260.00      $300.00    $1,63,800.00
697      1368.0                $5.00        $7.00       $9,576.00
698       723.0               $10.00        $7.00       $5,061.00
699      1806.0              $250.00       $12.00      $21,672.00

         Discounts          Sales   Cost of Goods Sold        Profit   \
695   $1,11,375.00   $6,31,125.00         $6,18,750.00    $12,375.00
696     $24,570.00   $1,39,230.00         $1,36,500.00     $2,730.00
697      $1,436.40      $8,139.60            $6,840.00     $1,299.60
698        $759.15      $4,301.85            $3,615.00       $686.85
699      $3,250.80     $18,421.20            $5,418.00    $13,003.20

           Date  Month Number  Month Name   Year
695  01-03-2014             3       March   2014
696  01-10-2014            10     October   2014
697  01-02-2014             2    February   2014
698  01-04-2014             4       April   2014
699  01-05-2014             5         May   2014
```

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

[5]:
```
(700, 16)
```

[6]:
```
df.columns
```

[6]:
```
Index(['Segment', 'Country', ' Product ', ' Discount Band ', 'Units Sold',
       ' Manufacturing Price ', ' Sale Price ', ' Gross Sales ', ' Discounts ',
       '  Sales ', ' Cost of Goods Sold ', ' Profit ', 'Date', 'Month Number',
       ' Month Name ', 'Year'],
      dtype='object')
```

[7]:
```
#checking which columns name contain white space
[x for x in df.columns if x.endswith(' ') or x.startswith(' ')]
```

[7]:
```
[' Product ',
 ' Discount Band ',
 ' Manufacturing Price ',
 ' Sale Price ',
 ' Gross Sales ',
 ' Discounts ',
 '  Sales ',
 ' Cost of Goods Sold ',
 ' Profit ',
 ' Month Name ']
```

[8]:
```
#removing white space form all column names
df.columns = df.columns.str.strip()
```

[9]:
```
df.columns
```

[9]:
```
Index(['Segment', 'Country', 'Product', 'Discount Band', 'Units Sold',
       'Manufacturing Price', 'Sale Price', 'Gross Sales', 'Discounts',
       'Sales', 'Cost of Goods Sold', 'Profit', 'Date', 'Month Number',
       'Month Name', 'Year'],
      dtype='object')
```

[10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Segment              700 non-null    object
 1   Country              700 non-null    object
 2   Product              700 non-null    object
 3   Discount Band        700 non-null    object
 4   Units Sold           700 non-null    float64
 5   Manufacturing Price  700 non-null    object
 6   Sale Price           700 non-null    object
 7   Gross Sales          700 non-null    object
 8   Discounts            700 non-null    object
 9   Sales                700 non-null    object
 10  Cost of Goods Sold   700 non-null    object
 11  Profit               700 non-null    object
 12  Date                 700 non-null    object
 13  Month Number         700 non-null    int64
 14  Month Name           700 non-null    object
 15  Year                 700 non-null    int64
dtypes: float64(1), int64(2), object(13)
memory usage: 87.6+ KB
```

[11]:

```
df['Date']= pd.to_datetime(df['Date'])
```

[12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Segment              700 non-null    object
 1   Country              700 non-null    object
 2   Product              700 non-null    object
 3   Discount Band        700 non-null    object
 4   Units Sold           700 non-null    float64
 5   Manufacturing Price  700 non-null    object
 6   Sale Price           700 non-null    object
 7   Gross Sales          700 non-null    object
 8   Discounts            700 non-null    object
 9   Sales                700 non-null    object
 10  Cost of Goods Sold   700 non-null    object
 11  Profit               700 non-null    object
 12  Date                 700 non-null    datetime64[ns]
 13  Month Number         700 non-null    int64
 14  Month Name           700 non-null    object
 15  Year                 700 non-null    int64
dtypes: datetime64[ns](1), float64(1), int64(2), object(12)
memory usage: 87.6+ KB
```

[13]:

```
df.dtypes
```

[13]:

```
Segment                          object
Country                          object
Product                          object
Discount Band                    object
Units Sold                      float64
Manufacturing Price              object
Sale Price                       object
Gross Sales                      object
Discounts                        object
```

```
Sales                           object
Cost of Goods Sold              object
Profit                          object
Date                    datetime64[ns]
Month Number                     int64
Month Name                      object
Year                             int64
dtype: object
```

[14]:
```
df['Discount Band'] = df['Discount Band'].str.strip()
df.Product = df.Product.str.strip()
```

[15]:
```
cols_to_clean = ['Manufacturing Price', 'Sale Price', 'Gross Sales',␣
 ↪'Discounts','Sales', 'Cost of Goods Sold', 'Profit']
cols_to_clean
```

[15]:
```
['Manufacturing Price',
 'Sale Price',
 'Gross Sales',
 'Discounts',
 'Sales',
 'Cost of Goods Sold',
 'Profit']
```

[16]:
```
for col in cols_to_clean:
    df[col] = df[col].str.strip().str.replace('(', '').str.replace(')', '').str.
 ↪replace('$', '').str.replace('-', '').str.replace(',', '').replace('', np.
 ↪nan)
```

```
C:\Users\Security\AppData\Local\Temp\ipykernel_12404\1659008232.py:2:
FutureWarning: The default value of regex will change from True to False in a
future version. In addition, single character regular expressions will *not* be
treated as literal strings when regex=True.
  df[col] = df[col].str.strip().str.replace('(', '').str.replace(')',
'').str.replace('$', '').str.replace('-', '').str.replace(',', '').replace('',
np.nan)
```

[17]:
```
df[cols_to_clean] = df[cols_to_clean].astype(float)
```

[18]:
```
df.dtypes
```

[18]:
```
Segment                         object
Country                         object
Product                         object
Discount Band                   object
Units Sold                     float64
Manufacturing Price            float64
Sale Price                     float64
```

```
Gross Sales                   float64
Discounts                     float64
Sales                         float64
Cost of Goods Sold            float64
Profit                        float64
Date                   datetime64[ns]
Month Number                    int64
Month Name                     object
Year                            int64
dtype: object
```

[19]:

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

[19]:

```
Segment                 0
Country                 0
Product                 0
Discount Band           0
Units Sold              0
Manufacturing Price     0
Sale Price              0
Gross Sales             0
Discounts              53
Sales                   0
Cost of Goods Sold      0
Profit                  5
Date                    0
Month Number            0
Month Name              0
Year                    0
dtype: int64
```

[20]:

```
df = df.dropna()
```

[21]:

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

[21]:

```
Segment                0
Country                0
Product                0
Discount Band          0
Units Sold             0
Manufacturing Price    0
Sale Price             0
Gross Sales            0
Discounts              0
Sales                  0
Cost of Goods Sold     0
Profit                 0
```

```
Date                   0
Month Number           0
Month Name             0
Year                   0
dtype: int64
```

[22]:
```
df.head()
```

[22]:
```
       Segment Country Product Discount Band  Units Sold  Manufacturing Price  \
53  Government  France   Paseo           Low      3945.0                 10.0
54   Midmarket  France   Paseo           Low      2296.0                 10.0
55  Government  France   Paseo           Low      1030.0                 10.0
56  Government  France    Velo           Low       639.0                120.0
57  Government  Canada     VTT           Low      1326.0                250.0

    Sale Price  Gross Sales  Discounts     Sales  Cost of Goods Sold  \
53         7.0      27615.0     276.15  27338.85             19725.0
54        15.0      34440.0     344.40  34095.60             22960.0
55         7.0       7210.0      72.10   7137.90              5150.0
56         7.0       4473.0      44.73   4428.27              3195.0
57         7.0       9282.0      92.82   9189.18              6630.0

      Profit       Date  Month Number Month Name  Year
53   7613.85 2014-01-01             1    January  2014
54  11135.60 2014-01-02             2   February  2014
55   1987.90 2014-01-05             5        May  2014
56   1233.27 2014-01-11            11   November  2014
57   2559.18 2014-01-03             3      March  2014
```

[23]:
```
df.shape
```

[23]:
```
(642, 16)
```

[24]:
```
df = df.reset_index(drop=True)
```

[25]:
```
df.head()
```

[25]:
```
      Segment Country Product Discount Band  Units Sold  Manufacturing Price  \
0  Government  France   Paseo           Low      3945.0                 10.0
1   Midmarket  France   Paseo           Low      2296.0                 10.0
2  Government  France   Paseo           Low      1030.0                 10.0
3  Government  France    Velo           Low       639.0                120.0
4  Government  Canada     VTT           Low      1326.0                250.0

   Sale Price  Gross Sales  Discounts     Sales  Cost of Goods Sold    Profit  \
0         7.0      27615.0     276.15  27338.85             19725.0   7613.85
1        15.0      34440.0     344.40  34095.60             22960.0  11135.60
```

```
2          7.0        7210.0        72.10   7137.90              5150.0   1987.90
3          7.0        4473.0        44.73   4428.27              3195.0   1233.27
4          7.0        9282.0        92.82   9189.18              6630.0   2559.18

        Date  Month Number  Month Name  Year
0 2014-01-01             1     January  2014
1 2014-01-02             2    February  2014
2 2014-01-05             5         May  2014
3 2014-01-11            11    November  2014
4 2014-01-03             3       March  2014
```

[26]:
```
df.describe(include=object)
```

[26]:
```
           Segment                   Country Product Discount Band Month Name
count          642                       642     642           642        642
unique           5                         5       6             3         12
top     Government  United States of America   Paseo          High    October
freq           280                       132     184           245        136
```

[27]:
```
for col in df.describe(include=object).columns:
    print(col)
    print(df[col].unique())
    print('-'*50)
```

```
Segment
['Government' 'Midmarket' 'Channel Partners' 'Enterprise' 'Small Business']
--------------------------------------------------
Country
['France' 'Canada' 'United States of America' 'Mexico' 'Germany']
--------------------------------------------------
Product
['Paseo' 'Velo' 'VTT' 'Carretera' 'Montana' 'Amarilla']
--------------------------------------------------
Discount Band
['Low' 'Medium' 'High']
--------------------------------------------------
Month Name
[' January ' ' February ' ' May ' ' November ' ' March ' ' July '
 ' September ' ' October ' ' December ' ' April ' ' August ' ' June ']
--------------------------------------------------
```

[28]:
```
df.describe()
```

[28]:
```
        Units Sold  Manufacturing Price  Sale Price   Gross Sales  \
count   642.000000           642.000000  642.000000  6.420000e+02
mean   1608.270249            97.119938  120.526480  1.855083e+05
std     873.403353           108.568244  137.797292  2.571253e+05
min     200.000000             3.000000    7.000000  1.799000e+03
```

```
25%     887.250000             5.000000   12.000000  1.745175e+04
50%    1537.500000            10.000000   20.000000  3.900700e+04
75%    2259.500000           250.000000  300.000000  2.826750e+05
max    4492.500000           260.000000  350.000000  1.207500e+06

           Discounts         Sales  Cost of Goods Sold         Profit  \
count     642.000000  6.420000e+02          642.000000     642.000000
mean    14282.839984  1.712254e+05       147616.026480   26030.980981
std     23643.608658  2.383121e+05       206004.271685   40679.164643
min        18.410000  1.655080e+03          918.000000     285.600000
25%      1038.587500  1.598608e+04         7548.000000    3974.130000
50%      3083.175000  3.554020e+04        22985.000000   10911.900000
75%     19261.125000  2.620725e+05       247437.500000   23967.000000
max    149677.500000  1.159200e+06       950625.000000  262200.000000

       Month Number         Year
count    642.000000   642.000000
mean       7.981308  2013.739875
std        3.367685     0.439044
min        1.000000  2013.000000
25%        6.000000  2013.000000
50%        9.000000  2014.000000
75%       11.000000  2014.000000
max       12.000000  2014.000000
```

[29]:

```
# Assuming no outlier present in the dataset
```

[30]:

```
#Saving clean data

#df.to_csv("clean_data.csv", index = False)
```

5 Data Analysis and Visualization

[31]:

```
#To remove duplicates from column
#df.drop_duplicates(subset=['colname'],keep=False)
```

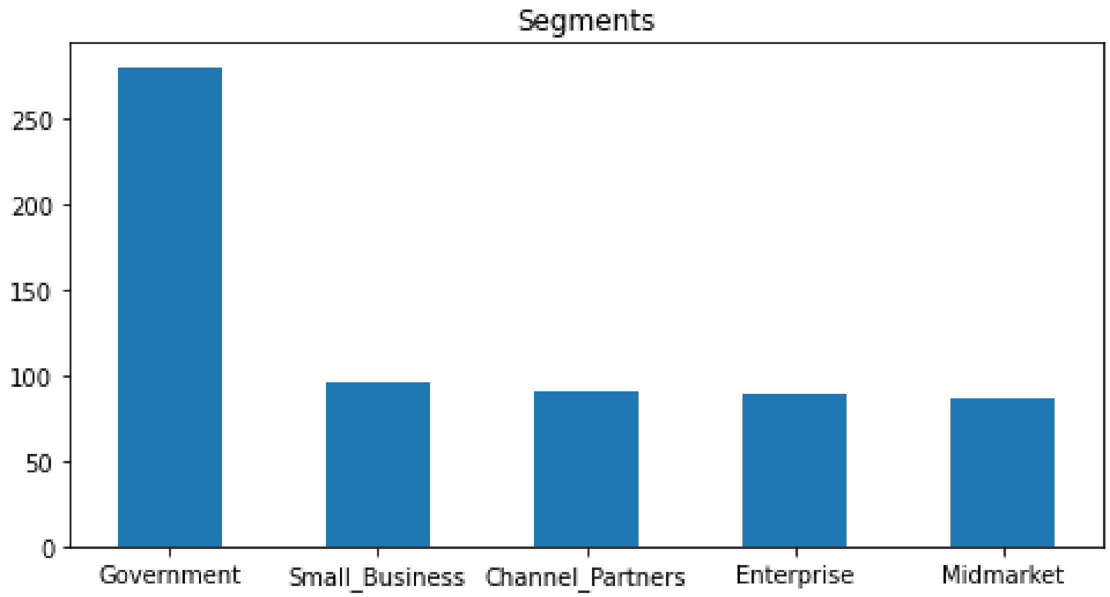
[32]:

```
seg = df['Segment'].value_counts(normalize=True)
print(seg)

plt.figure(figsize=(8,4))
plt.title('Segments')
plt.
  ↪bar(['Government','Small_Business','Channel_Partners','Enterprise','Midmarket'],df['Segment
  ↪value_counts(),width=0.5)
plt.show()
```

```
Government          0.436137
```

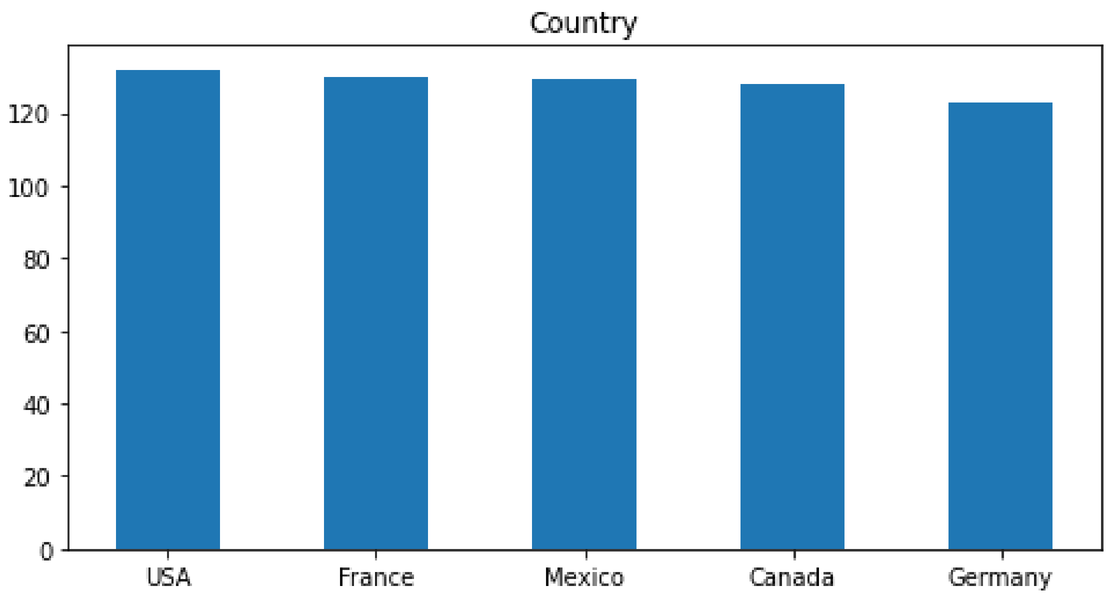
```
Small Business      0.149533
Channel Partners    0.140187
Enterprise          0.138629
Midmarket           0.135514
Name: Segment, dtype: float64
```



[33]:

```
con = df['Country'].value_counts(normalize=True)
print(con)
plt.figure(figsize=(8,4))
plt.title('Country')
plt.bar(['USA','France','Mexico','Canada','Germany'],df['Country'].
  ↪value_counts(),width=0.5)
plt.show()
```

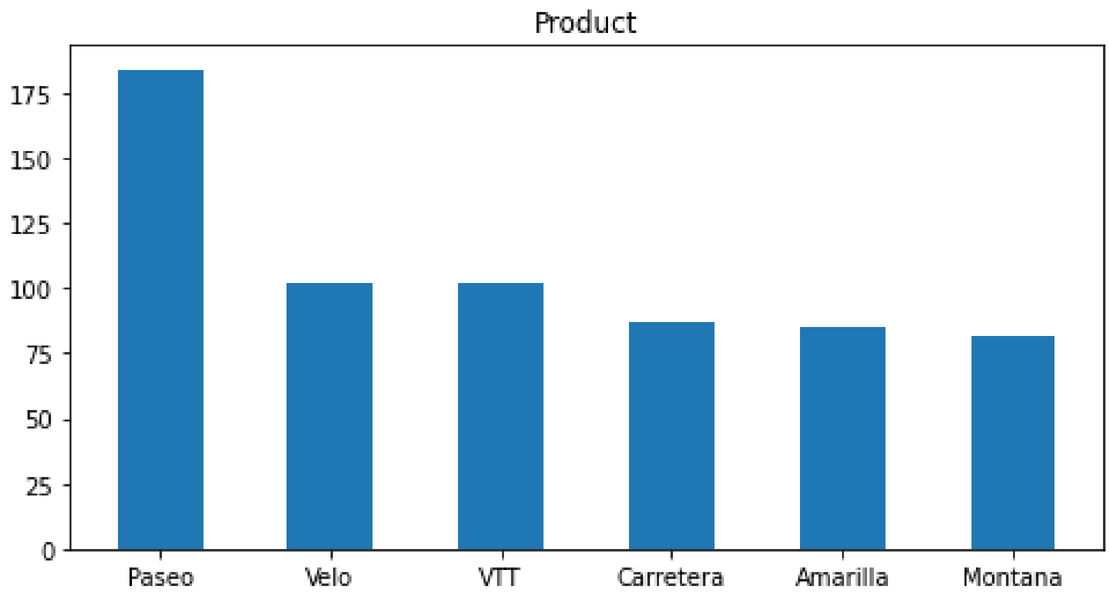
```
United States of America    0.205607
France                      0.202492
Mexico                      0.200935
Canada                      0.199377
Germany                     0.191589
Name: Country, dtype: float64
```



[34]:

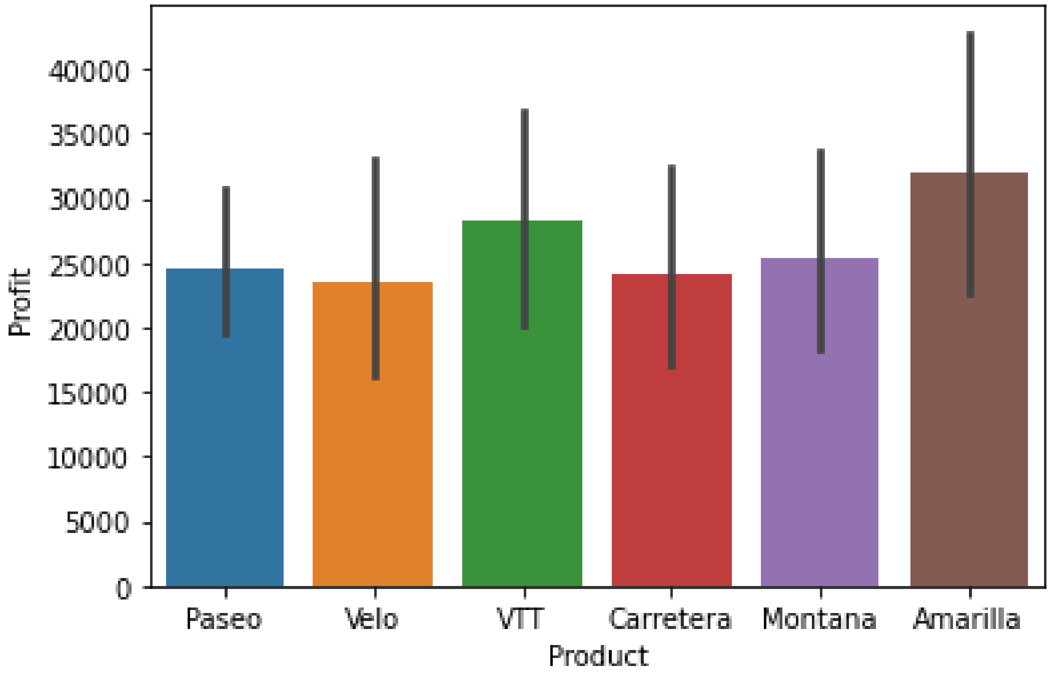
```
Pro = df['Product'].value_counts(normalize=True)
print(Pro)
plt.figure(figsize=(8,4))
plt.title('Product')
plt.bar(['Paseo','Velo','VTT','Carretera','Amarilla','Montana'],df['Product'].
  ↪value_counts(),width=0.5)
plt.show()
```

```
Paseo        0.286604
Velo         0.158879
VTT          0.158879
Carretera    0.135514
Amarilla     0.132399
Montana      0.127726
Name: Product, dtype: float64
```



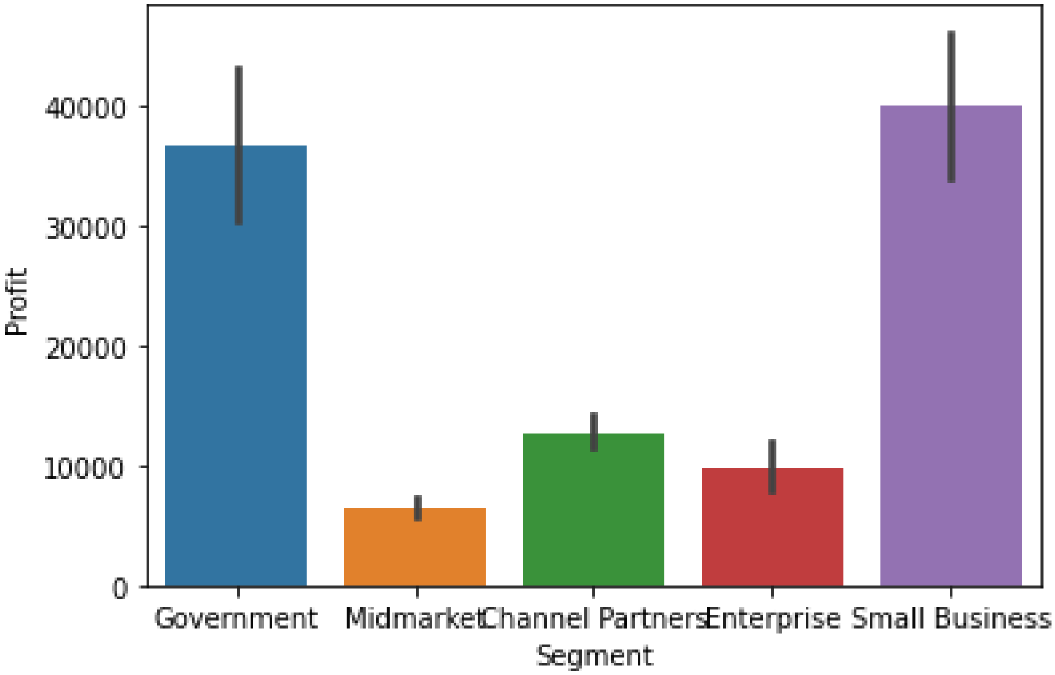
[35]:
```
sns.barplot(x='Product',y='Profit',data=df)
```

[35]:
```
<AxesSubplot:xlabel='Product', ylabel='Profit'>
```



[36]:
```
sns.barplot(x='Segment',y='Profit',data=df)
```

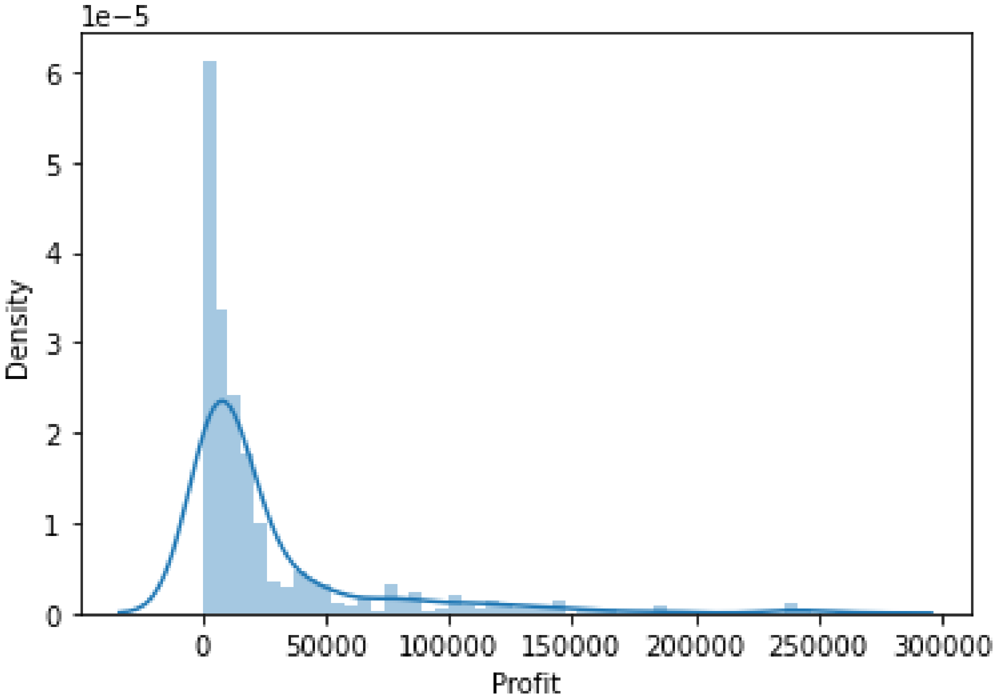
[36]:
```
<AxesSubplot:xlabel='Segment', ylabel='Profit'>
```



[37]:
```
sns.distplot(df['Profit'])
```

```
C:\Users\Security\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[37]:
```
<AxesSubplot:xlabel='Profit', ylabel='Density'>
```



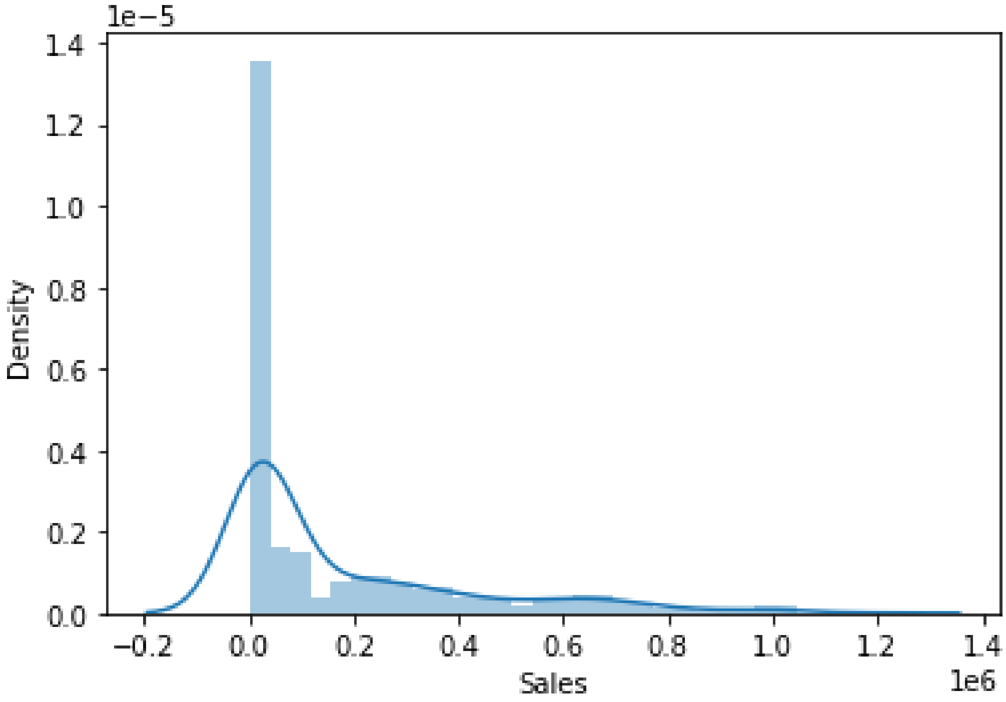
[38]:

```
sns.distplot(df['Sales'],kde=True,bins=30)
```

```
C:\Users\Security\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

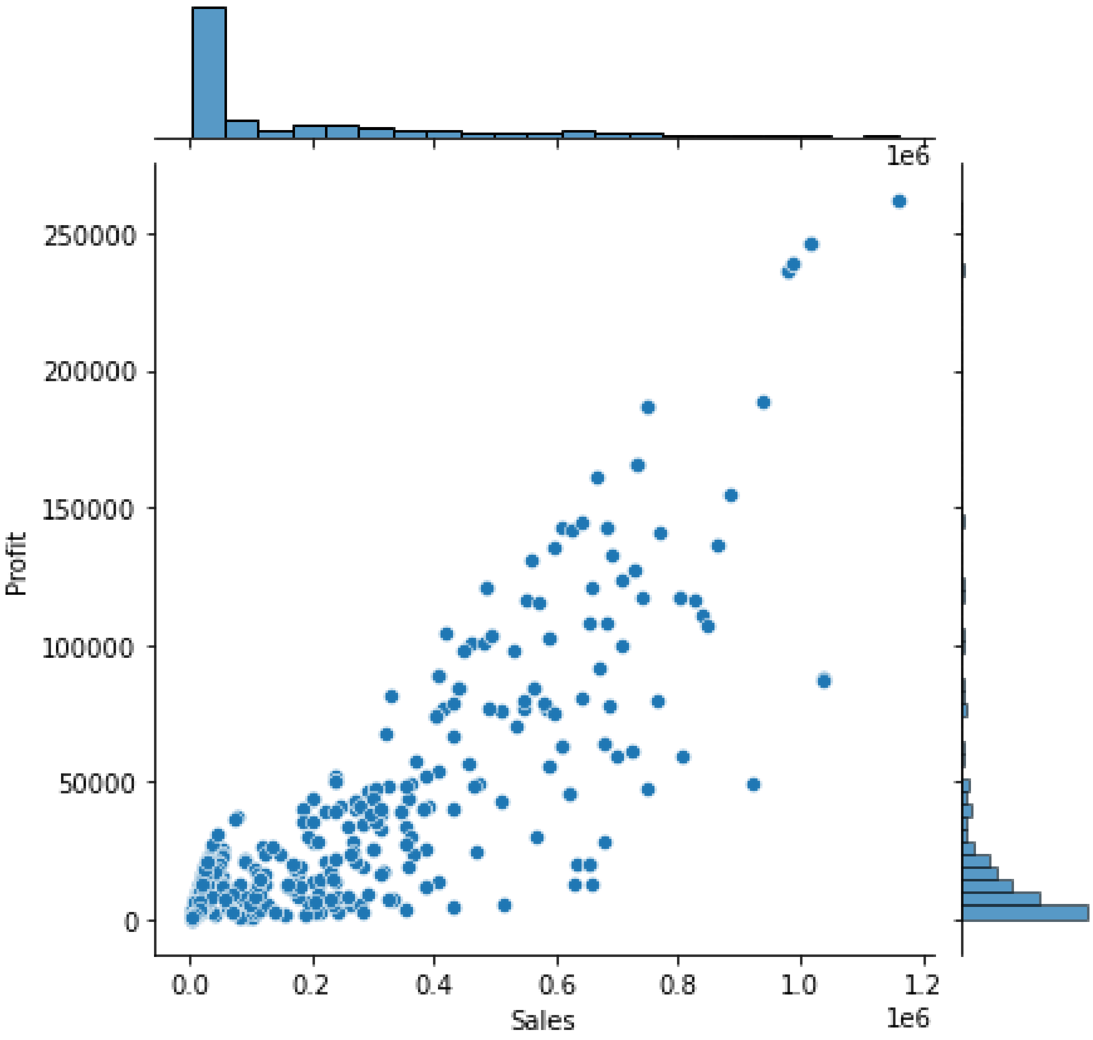
[38]:

```
<AxesSubplot:xlabel='Sales', ylabel='Density'>
```



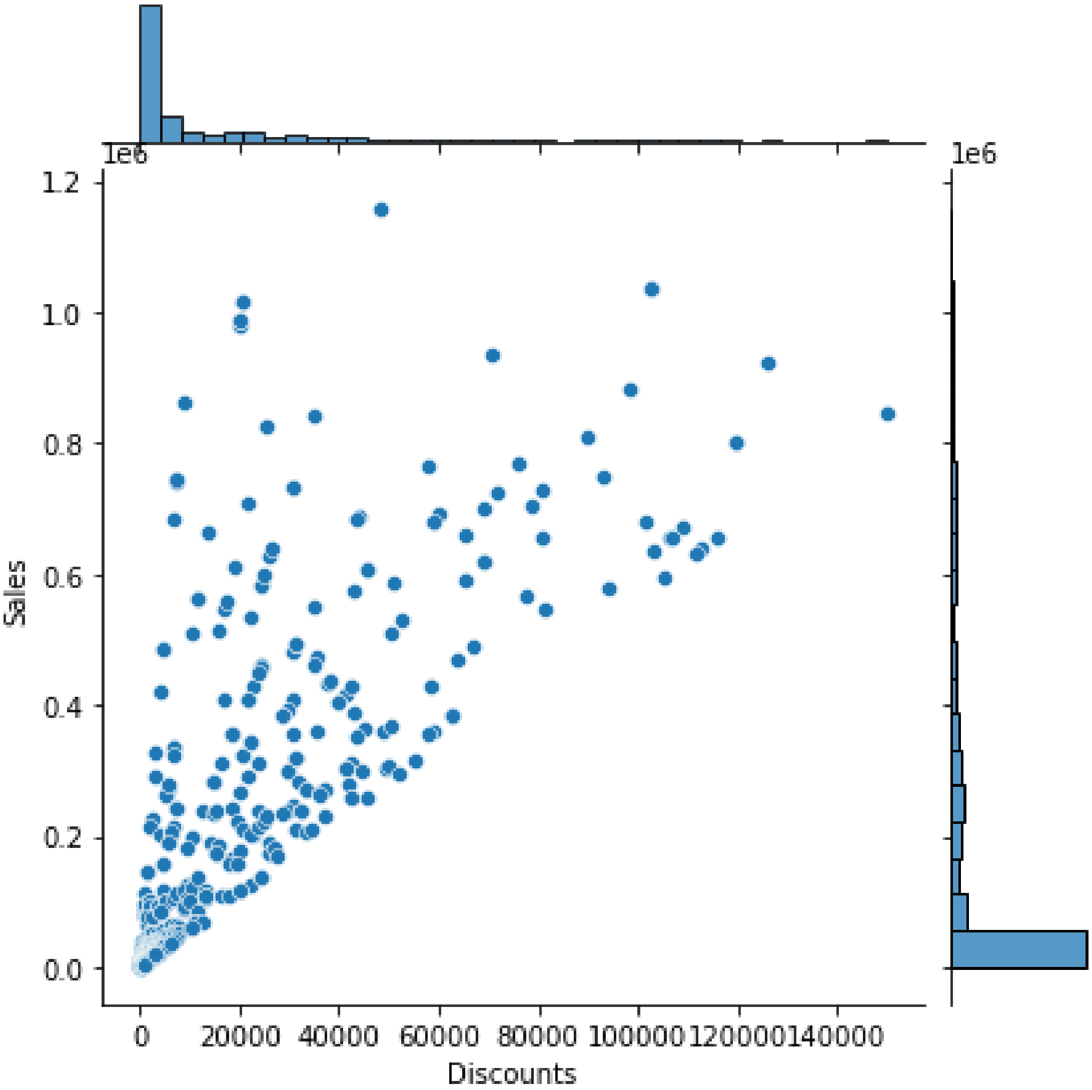
[39]:
```
sns.jointplot(x='Sales',y='Profit',data=df,kind='scatter')
```

[39]:
```
<seaborn.axisgrid.JointGrid at 0x1b009c5cfa0>
```



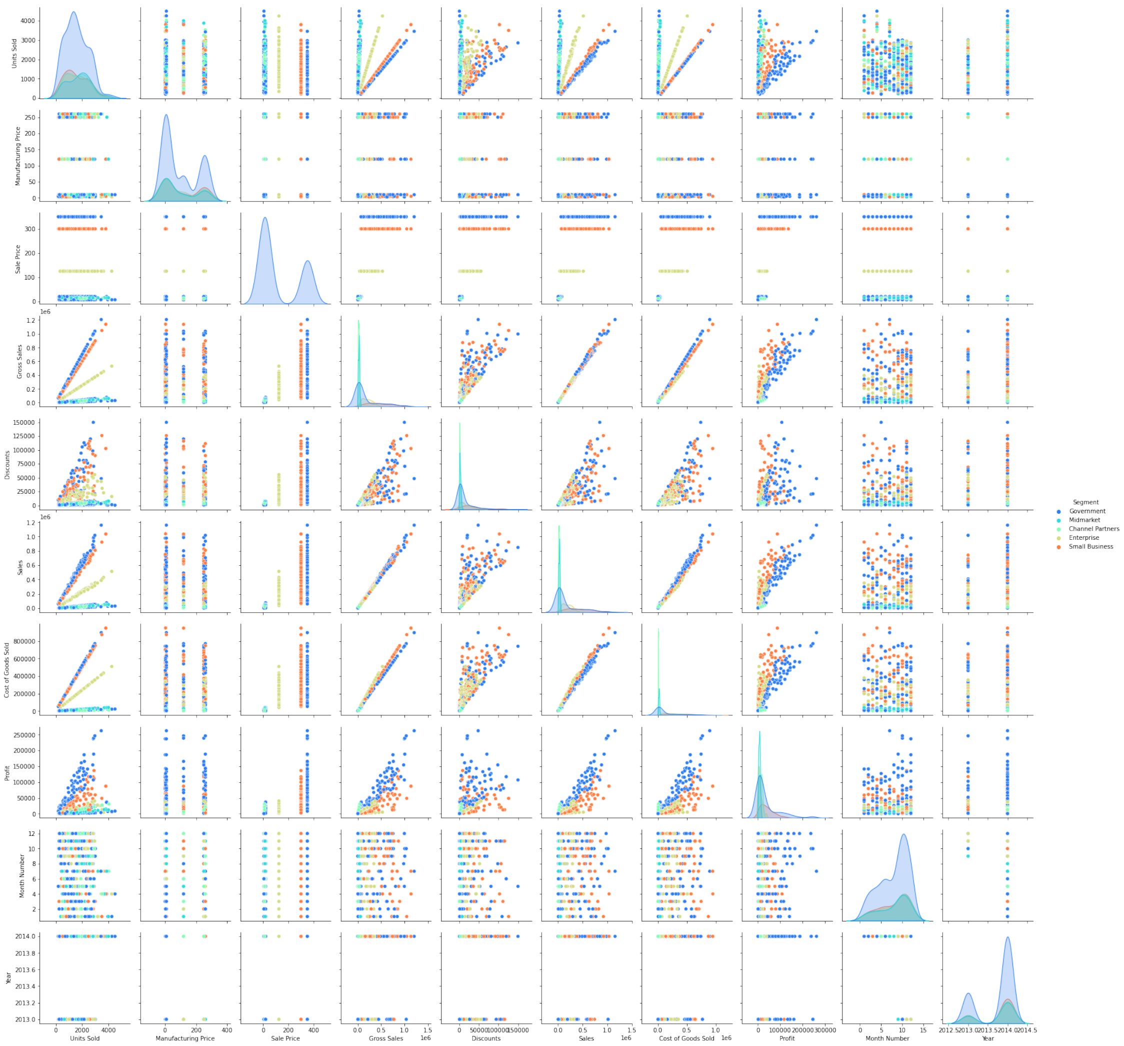
```
[40]: sns.jointplot(x='Discounts',y='Sales',data=df,kind='scatter')
```

```
[40]: <seaborn.axisgrid.JointGrid at 0x1b009731f10>
```



```
[41]: sns.pairplot(df,hue='Segment',palette='rainbow')
```

```
[41]: <seaborn.axisgrid.PairGrid at 0x1b00adbe670>
```



[42]:
```
df.corr()
```

[42]:
```
                     Units Sold  Manufacturing Price  Sale Price  Gross Sales  \
Units Sold             1.000000            -0.052763   -0.069328     0.327200
Manufacturing Price   -0.052763             1.000000    0.070428     0.036657
Sale Price            -0.069328             0.070428    1.000000     0.803442
Gross Sales            0.327200             0.036657    0.803442     1.000000
Discounts              0.265492             0.019295    0.665243     0.812565
Sales                  0.326690             0.037637    0.800867     0.998327
Cost of Goods Sold     0.330028             0.033304    0.795471     0.994724
Profit                 0.282140             0.054576    0.665323     0.831533
Month Number          -0.096963             0.017163   -0.015177    -0.041849
Year                   0.049838            -0.008836    0.017920     0.054330

                     Discounts     Sales  Cost of Goods Sold    Profit  \
Units Sold            0.265492  0.326690            0.330028  0.282140
Manufacturing Price   0.019295  0.037637            0.033304  0.054576
Sale Price            0.665243  0.800867            0.795471  0.665323
Gross Sales           0.812565  0.998327            0.994724  0.831533
Discounts             1.000000  0.777499            0.813402  0.480415
Sales                 0.777499  1.000000            0.992551  0.849513
Cost of Goods Sold    0.813402  0.992551            1.000000  0.783844
Profit                0.480415  0.849513            0.783844  1.000000
Month Number         -0.071387 -0.038070           -0.042479 -0.002449
Year                  0.036024  0.055045            0.056406  0.033304

                     Month Number      Year
Units Sold              -0.096963  0.049838
Manufacturing Price      0.017163 -0.008836
Sale Price              -0.015177  0.017920
Gross Sales             -0.041849  0.054330
Discounts               -0.071387  0.036024
Sales                   -0.038070  0.055045
Cost of Goods Sold      -0.042479  0.056406
Profit                  -0.002449  0.033304
Month Number             1.000000 -0.428507
Year                    -0.428507  1.000000
```

[ ]: