

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

```
%matplotlib inline
```

Data Cleaning

```
# reading the diwali sales csv file
```

```
df = pd.read_csv('Diwali Sales Data.csv', encoding='unicode_escape') #
To avoid encoding error, use 'unicode_escape'
```

```
df.head()
```

```
{
  "summary": {
    "name": "df",
    "rows": 11251,
    "fields": [
      {
        "column": "User_ID",
        "properties": {
          "dtype": "number",
          "std": 1716,
          "min": 1000001,
          "max": 1006040,
          "num_unique_values": 3755,
          "samples": [
            1005905,
            1003730,
            1005326
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Cust_name",
        "properties": {
          "dtype": "category",
          "num_unique_values": 1250,
          "samples": [
            "Nida",
            "Lacy",
            "Caudle"
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Product_ID",
        "properties": {
          "dtype": "category",
          "num_unique_values": 2351,
          "samples": [
            "P00224442",
            "P00205242",
            "P00347442"
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Gender",
        "properties": {
          "dtype": "category",
          "num_unique_values": 2,
          "samples": [
            "M",
            "F"
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Age Group",
        "properties": {
          "dtype": "category",
          "num_unique_values": 7,
          "samples": [
            "26-35",
            "0-17"
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Age",
        "properties": {
          "dtype": "number",
          "std": 12,
          "min": 12,
          "max": 92,
          "num_unique_values": 81,
          "samples": [
            18,
            28
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Marital_Status",
        "properties": {
          "dtype": "number",
          "std": 0,
          "min": 0,
          "max": 0,
          "num_unique_values": 1,
          "samples": [
            0
          ],
          "semantic_type": "",
          "description": ""
        }
      ]
    }
  }
}
```



```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 11251 entries, 0 to 11250
```

```
Data columns (total 15 columns):
```

#	Column	Non-Null Count	Dtype
0	User_ID	11251 non-null	int64
1	Cust_name	11251 non-null	object
2	Product_ID	11251 non-null	object
3	Gender	11251 non-null	object
4	Age Group	11251 non-null	object
5	Age	11251 non-null	int64
6	Marital_Status	11251 non-null	int64
7	State	11251 non-null	object
8	Zone	11251 non-null	object
9	Occupation	11251 non-null	object
10	Product_Category	11251 non-null	object
11	Orders	11251 non-null	int64
12	Amount	11239 non-null	float64
13	Status	0 non-null	float64
14	unnamed1	0 non-null	float64

```
dtypes: float64(3), int64(4), object(8)
```

```
memory usage: 1.3+ MB
```

```
df.describe()
```

```
{"summary":{"name": "df", "rows": 8, "fields": [{"column": "User_ID", "properties": {"dtype": "number", "std": 461310.51175439754, "min": 1716.125401923134, "max": 1006040.0, "num_unique_values": 8, "samples": [1003004.488134388, 1003065.0, 11251.0]}, "semantic_type": "", "description": ""}, {"column": "Age", "properties": {"dtype": "number", "std": 3965.0199871765367, "min": 12.0, "max": 11251.0, "num_unique_values": 8, "samples": [35.421207003821884, 33.0, 11251.0]}, "semantic_type": "", "description": ""}, {"column": "Marital_Status", "properties": {"dtype": "number", "std": 3977.6820425393917, "min": 0.0, "max": 11251.0, "num_unique_values": 5, "samples": [0.4203181939383166, 1.0, 0.4936319151275842]}, "semantic_type": "", "description": ""}, {"column": "Orders", "properties": {"dtype": "number", "std": 3977.0664408388902, "min": 1.0, "max": 11251.0, "num_unique_values": 8, "samples": [2.4892898409030306, 2.0, 11251.0]}, "semantic_type": "", "description": ""}]}}
```

```

n    },\n    {\n        \"column\": \"Amount\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 7024.070625946779, \n            \"min\": 188.0, \n            \"max\": 23952.0, \n            \"num_unique_values\": 8, \n            \"samples\": [\n                9453.610857727557, \n                8109.0, \n                11239.0\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"Status\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": null, \n            \"min\": 0.0, \n            \"max\": 0.0, \n            \"num_unique_values\": 1, \n            \"samples\": [\n                0.0\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"unnamed1\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": null, \n            \"min\": 0.0, \n            \"max\": 0.0, \n            \"num_unique_values\": 1, \n            \"samples\": [\n                0.0\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n        }\n    }\n]\n}, \"type\": \"dataframe\"}

```

```
df[df['Amount'].isnull()]
```

```

{\"summary\": {\n    \"name\": \"df[df['Amount']]\", \n    \"rows\": 12, \n    \"fields\": [\n        {\n            \"column\": \"User_ID\", \n            \"properties\": {\n                \"dtype\": \"number\", \n                \"std\": 1619, \n                \"min\": 1000326, \n                \"max\": 1005538, \n                \"num_unique_values\": 12, \n                \"samples\": [\n                    1004528, \n                    1004601, \n                    1002092\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n            }\n        },\n        {\n            \"column\": \"Cust_name\", \n            \"properties\": {\n                \"dtype\": \"string\", \n                \"num_unique_values\": 12, \n                \"samples\": [\n                    \"Anurag\", \n                    \"Gaurav\", \n                    \"Shivangi\"\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n            }\n        },\n        {\n            \"column\": \"Product_ID\", \n            \"properties\": {\n                \"dtype\": \"string\", \n                \"num_unique_values\": 12, \n                \"samples\": [\n                    \"P00338442\", \n                    \"P00014442\", \n                    \"P00273442\"\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n            }\n        },\n        {\n            \"column\": \"Gender\", \n            \"properties\": {\n                \"dtype\": \"category\", \n                \"num_unique_values\": 2, \n                \"samples\": [\n                    \"M\", \n                    \"F\"\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n            }\n        },\n        {\n            \"column\": \"Age Group\", \n            \"properties\": {\n                \"dtype\": \"category\", \n                \"num_unique_values\": 5, \n                \"samples\": [\n                    \"46-50\", \n                    \"26-35\"\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n            }\n        },\n        {\n            \"column\": \"Age\", \n            \"properties\": {\n                \"dtype\": \"number\", \n                \"std\": 8, \n                \"min\": 33, \n                \"max\": 61, \n                \"num_unique_values\": 11, \n                \"samples\": [\n                    33, \n                    34, \n                    35, \n                    36, \n                    37, \n                    38, \n                    39, \n                    40, \n                    41, \n                    42, \n                    43, \n                    44, \n                    45, \n                    46, \n                    47, \n                    48, \n                    49, \n                    50, \n                    51, \n                    52, \n                    53, \n                    54, \n                    55, \n                    56, \n                    57, \n                    58, \n                    59, \n                    60, \n                    61\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n            }\n        }\n    ]\n}\n}

```

```
[
    53,
    61,
    ],
    "semantic_type":
    "",
    "description":
    },
    "column":
    "Marital_Status",
    "properties": {
        "dtype": "number",
        "std": 0,
        "min": 0,
        "max": 1,
        "num_unique_values": 2,
        "samples": [
            1,
            0
        ],
        "semantic_type":
        "",
        "description":
        },
        "column":
        "State",
        "properties": {
            "dtype":
            "string",
            "num_unique_values": 7,
            "samples": [
                "Maharashtra",
                "Madhya Pradesh"
            ],
            "semantic_type":
            "",
            "description":
            },
            "column":
            "Zone",
            "properties": {
                "dtype": "category",
                "num_unique_values": 3,
                "samples": [
                    "Western",
                    "Central"
                ],
                "semantic_type":
                "",
                "description":
                },
                "column":
                "Occupation",
                "properties": {
                    "dtype": "string",
                    "num_unique_values": 6,
                    "samples": [
                        "IT Sector",
                        "Hospitality"
                    ],
                    "semantic_type":
                    "",
                    "description":
                    },
                    "column":
                    "Product_Category",
                    "properties": {
                        "dtype": "category",
                        "num_unique_values": 4,
                        "samples": [
                            "Footwear & Shoes",
                            "Food"
                        ],
                        "semantic_type":
                        "",
                        "description":
                        },
                        "column":
                        "Orders",
                        "properties": {
                            "dtype": "number",
                            "std": 1,
                            "min": 1,
                            "max": 4,
                            "num_unique_values": 4,
                            "samples": [
                                3,
                                4
                            ],
                            "semantic_type":
                            "",
                            "description":
                            },
                            "column":
                            "Amount",
                            "properties": {
                                "dtype": "number",
                                "std": null,
                                "min": null,
                                "max": null,
                                "num_unique_values": 0,
                                "samples": [],
                                "semantic_type":
                                "",
                                "description":
                                },
                                "column":
                                "Status",
                                "properties": {
                                    "dtype": "number",
                                    "std": null,
                                    "min": null,
                                    "max": null,
                                    "num_unique_values": 0,
                                    "samples": [],
                                    "semantic_type":
                                    "",
                                    "description":
                                    },
                                    "column":
                                    "unnamed1",
                                    "properties": {
                                        "dtype": "number",
                                        "std": null,
                                        "min": null,
                                        "max": null,
                                        "num_unique_values": 0,
                                        "samples": [],
                                        "semantic_type":
                                        "",
                                        "description":
                                        }
                                    }
                                }
                            ],
                        "type": "dataframe"
                    }
                ]
            ]
        ]
    ]

```

```
# dropping status and unnamed1 cols
```

```
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

```

-----
-----
KeyError                                Traceback (most recent call
last)
/tmp/ipython-input-19-3197098881.py in <cell line: 0>()
      1 # dropping status and unnamed1 cols
      2
----> 3 df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
      4 df.head()

/usr/local/lib/python3.11/dist-packages/pandas/core/frame.py in
drop(self, labels, axis, index, columns, level, inplace, errors)
    5579         weight  1.0      0.8
    5580         """
-> 5581         return super().drop(
    5582             labels=labels,
    5583             axis=axis,

/usr/local/lib/python3.11/dist-packages/pandas/core/generic.py in
drop(self, labels, axis, index, columns, level, inplace, errors)
    4786         for axis, labels in axes.items():
    4787             if labels is not None:
-> 4788                 obj = obj._drop_axis(labels, axis,
level=level, errors=errors)
    4789
    4790         if inplace:

/usr/local/lib/python3.11/dist-packages/pandas/core/generic.py in
_drop_axis(self, labels, axis, level, errors, only_slice)
    4828         new_axis = axis.drop(labels, level=level,
errors=errors)
    4829         else:
-> 4830         new_axis = axis.drop(labels, errors=errors)
    4831         indexer = axis.get_indexer(new_axis)
    4832

/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in
drop(self, labels, errors)
    7068         if mask.any():
    7069             if errors != "ignore":
-> 7070                 raise KeyError(f"{labels[mask].tolist()} not
found in axis")
    7071         indexer = indexer[~mask]
    7072         return self.delete(indexer)

KeyError: "['Status', 'unnamed1'] not found in axis"

df.head()

```

```

{"summary":{"\n  \"name\": \"df\", \n  \"rows\": 11251, \n  \"fields\": [\n    {\n      \"column\": \"User_ID\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 1716, \n        \"min\": 1000001, \n        \"max\": 1006040, \n        \"num_unique_values\": 3755, \n        \"samples\": [\n          1005905, \n          1003730, \n          1005326\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Cust_name\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 1250, \n        \"samples\": [\n          \"Nida\", \n          \"Lacy\", \n          \"Caudle\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Product_ID\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 2351, \n        \"samples\": [\n          \"P00224442\", \n          \"P00205242\", \n          \"P00347442\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Gender\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 2, \n        \"samples\": [\n          \"M\", \n          \"F\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Age Group\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 7, \n        \"samples\": [\n          \"26-35\", \n          \"0-17\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Age\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 12, \n        \"min\": 12, \n        \"max\": 92, \n        \"num_unique_values\": 81, \n        \"samples\": [\n          18, \n          28\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Marital_Status\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 0, \n        \"min\": 0, \n        \"max\": 1, \n        \"num_unique_values\": 2, \n        \"samples\": [\n          1, \n          0\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"State\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 16, \n        \"samples\": [\n          \"Maharashtra\", \n          \"Andhra Pradesh\", \n          \"u00a0Pradesh\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Zone\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 5, \n        \"samples\": [\n          \"Southern\", \n          \"Eastern\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      \"column\": \"Occupation\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 15, \n        \"samples\": [\n          \"Retail\", \n          \"Aviation\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }
    }
  ]
}

```

```

n    },\n    {\n        \"column\": \"Product_Category\",\n        \"properties\": {\n            \"dtype\": \"category\",\n            \"num_unique_values\": 18,\n            \"samples\": [\n                \"Auto\",\n                \"Hand & Power Tools\"\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"Orders\",\n        \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 1,\n            \"min\": 1,\n            \"max\": 4,\n            \"num_unique_values\": 4,\n            \"samples\": [\n                3,\n                4\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"Amount\",\n        \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 5222.355869186444,\n            \"min\": 188.0,\n            \"max\": 23952.0,\n            \"num_unique_values\": 6584,\n            \"samples\": [\n                19249.0,\n                13184.0\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n        }\n    }\n]\n}","type":"dataframe","variable_name":"df"}

```

```
df['Marital_Status'].unique()
```

```
array([0, 1])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 11251 entries, 0 to 11250
```

```
Data columns (total 13 columns):
```

#	Column	Non-Null Count	Dtype
0	User_ID	11251 non-null	int64
1	Cust_name	11251 non-null	object
2	Product_ID	11251 non-null	object
3	Gender	11251 non-null	object
4	Age Group	11251 non-null	object
5	Age	11251 non-null	int64
6	Marital_Status	11251 non-null	int64
7	State	11251 non-null	object
8	Zone	11251 non-null	object
9	Occupation	11251 non-null	object
10	Product_Category	11251 non-null	object
11	Orders	11251 non-null	int64
12	Amount	11239 non-null	float64

```
dtypes: float64(1), int64(4), object(8)
```

```
memory usage: 1.1+ MB
```

```
df['Amount'].isnull().sum()
```

```
np.int64(12)
```

```
df[df['Amount'].isnull()]
```



```
{
  "summary": {
    "name": "df[df['Amount']]",
    "rows": 12,
    "fields": [
      {
        "column": "User_ID",
        "dtype": "number",
        "std": 1619,
        "min": 1000326,
        "max": 1005538,
        "num_unique_values": 12,
        "samples": [
          1004528,
          1004601,
          1002092
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Cust_name",
        "dtype": "string",
        "num_unique_values": 12,
        "samples": [
          "Anurag",
          "Gaurav",
          "Shivangi"
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Product_ID",
        "dtype": "string",
        "num_unique_values": 12,
        "samples": [
          "P00338442",
          "P00014442",
          "P00273442"
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Gender",
        "dtype": "category",
        "num_unique_values": 2,
        "samples": [
          "M",
          "F"
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Age Group",
        "dtype": "category",
        "num_unique_values": 5,
        "samples": [
          "46-50",
          "26-35"
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Age",
        "dtype": "number",
        "std": 8,
        "min": 33,
        "max": 61,
        "num_unique_values": 11,
        "samples": [
          53,
          61
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Marital_Status",
        "dtype": "number",
        "std": 0,
        "min": 0,
        "max": 1,
        "num_unique_values": 2,
        "samples": [
          1,
          0
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "State",
        "dtype": "string",
        "num_unique_values": 7,
        "samples": [
          "Maharashtra",
          "Madhya Pradesh"
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Zone",
        "dtype": "category",
        "num_unique_values": 3,
        "samples": [
          "Western",
          "Central"
        ],
        "semantic_type": "\"",
        "description": ""
      },
      {
        "column": "Occupation",
        "dtype": "string",
        "num_unique_values": 6,
        "samples": [
          "IT Sector",
          "Hospitality"
        ],
        "semantic_type": "\"",
        "description": ""
      }
    ]
  }
}
```

```

{"semantic_type": "",
 "description": "",
 "column": "Product_Category",
 "properties": {
   "dtype": "category",
   "num_unique_values": 4,
   "samples": [
     "Footwear & Shoes",
     "Food"
   ],
   "semantic_type": "",
   "description": "",
   "column": "Orders",
   "properties": {
     "dtype": "number",
     "std": 1,
     "min": 1,
     "max": 4,
     "num_unique_values": 4,
     "samples": [
       3,
       4
     ],
     "semantic_type": "",
     "description": "",
     "column": "Amount",
     "properties": {
       "dtype": "number",
       "std": null,
       "min": null,
       "max": null,
       "num_unique_values": 0,
       "samples": []
     },
     "semantic_type": "",
     "description": ""
   }
 },
 "type": "dataframe"
}

```

dropping nan values

```
df.dropna(inplace=True)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 11239 entries, 0 to 11250
```

```
Data columns (total 13 columns):
```

#	Column	Non-Null	Count	Dtype
0	User_ID	11239	non-null	int64
1	Cust_name	11239	non-null	object
2	Product_ID	11239	non-null	object
3	Gender	11239	non-null	object
4	Age Group	11239	non-null	object
5	Age	11239	non-null	int64
6	Marital_Status	11239	non-null	int64
7	State	11239	non-null	object
8	Zone	11239	non-null	object
9	Occupation	11239	non-null	object
10	Product_Category	11239	non-null	object
11	Orders	11239	non-null	int64
12	Amount	11239	non-null	float64

```
dtypes: float64(1), int64(4), object(8)
```

```
memory usage: 1.2+ MB
```

```
df.shape
```

```
(11239, 13)
```

Chaging data type for Amount

```
df['Amount'] = df['Amount'].astype('int')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 11239 entries, 0 to 11250
```

```
Data columns (total 13 columns):
```

#	Column	Non-Null Count	Dtype
0	User_ID	11239 non-null	int64
1	Cust_name	11239 non-null	object
2	Product_ID	11239 non-null	object
3	Gender	11239 non-null	object
4	Age Group	11239 non-null	object
5	Age	11239 non-null	int64
6	Marital_Status	11239 non-null	int64
7	State	11239 non-null	object
8	Zone	11239 non-null	object
9	Occupation	11239 non-null	object
10	Product_Category	11239 non-null	object
11	Orders	11239 non-null	int64
12	Amount	11239 non-null	int64

```
dtypes: int64(5), object(8)
```

```
memory usage: 1.2+ MB
```

```
df.head()
```

```
{
  "summary": {
    "name": "df",
    "rows": 11239,
    "fields": [
      {
        "column": "User_ID",
        "properties": {
          "dtype": "number",
          "std": 1716,
          "min": 1000001,
          "max": 1006040,
          "num_unique_values": 3752,
          "samples": [
            1002014,
            1003491,
            1001842
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Cust_name",
        "properties": {
          "dtype": "category",
          "num_unique_values": 1250,
          "samples": [
            "Hallsten",
            "Shubham",
            "Riya"
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Product_ID",
        "properties": {
          "dtype": "category",
          "num_unique_values": 2350,
          "samples": [
            "P00133342",
            "P00302142",
            "P00227542"
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Gender",
        "properties": {
          "dtype": "category",
          "num_unique_values": 2,
          "samples": [
            "M",
            "F"
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Age Group",
        "properties": {
          "dtype": "category",
          "num_unique_values": 7,
          "samples": [
            "26-35",
            "0-17"
          ],
          "semantic_type": ""
        },
        "description": ""
      }
    ]
  }
}
```

```

\\",\n      \"description\": \"\"\n    }\n  },\n  {\n    \"column\": \"Age\",\n    \"properties\": {\n      \"dtype\":\n    \"number\",\n      \"std\": 12,\n      \"min\": 12,\n      \"max\": 92,\n      \"num_unique_values\": 81,\n      \"samples\":\n    [\n      38,\n      28\n    ],\n      \"semantic_type\":\n    \"\",\n    \"description\": \"\"\n  }\n},\n{\n  \"column\": \"Marital_Status\",\n  \"properties\": {\n    \"dtype\": \"number\",\n    \"std\": 0,\n    \"min\": 0,\n    \"max\": 1,\n    \"num_unique_values\": 2,\n    \"samples\":\n  [\n    1,\n    0\n  ],\n    \"semantic_type\":\n  \"\",\n  \"description\": \"\"\n},\n{\n  \"column\": \"State\",\n  \"properties\": {\n    \"dtype\":\n  \"category\",\n    \"num_unique_values\": 16,\n    \"samples\": [\n      \"Maharashtra\",\n      \"Andhra\\u00a0Pradesh\"\n    ],\n    \"semantic_type\": \"\",\n    \"description\": \"\"\n  },\n  {\n    \"column\": \"Zone\",\n    \"properties\": {\n      \"dtype\": \"category\",\n      \"num_unique_values\": 5,\n      \"samples\": [\n        \"Southern\",\n        \"Eastern\"\n      ],\n      \"semantic_type\": \"\",\n      \"description\": \"\"\n    },\n    {\n      \"column\": \"Occupation\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 15,\n        \"samples\": [\n          \"IT Sector\",\n          \"Hospitality\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      {\n        \"column\": \"Product_Category\",\n        \"properties\": {\n          \"dtype\": \"category\",\n          \"num_unique_values\": 18,\n          \"samples\": [\n            \"Auto\",\n            \"Hand & Power Tools\"\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\"\n        },\n        {\n          \"column\": \"Orders\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 1,\n            \"min\": 1,\n            \"max\": 4,\n            \"num_unique_values\": 4,\n            \"samples\": [\n              3,\n              4\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n          },\n          {\n            \"column\": \"Amount\",\n            \"properties\": {\n              \"dtype\": \"number\",\n              \"std\": 5222,\n              \"min\": 188,\n              \"max\": 23952,\n              \"num_unique_values\": 6583,\n              \"samples\": [\n                19247,\n                5293\n              ],\n              \"semantic_type\": \"\",\n              \"description\": \"\"\n            }\n          }\n        }\n      }\n    ],\n    \"type\": \"dataframe\", \"variable_name\": \"df\"}

```

```
df['Amount'].dtypes
```

```
dtype('int64')
```

```
df.columns
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
```

```

    'Marital_Status', 'State', 'Zone', 'Occupation',
    'Product_Category',
    'Orders', 'Amount'],
    dtype='object')

```

```

# Renaming column Marital_status as Shadi

```

```

df.rename(columns={'Marital_Status': 'Shaadi'})

```

```

{"summary":{"\n  \"name\": \"df\",\n  \"rows\": 11239,\n  \"fields\": [\n    {\n      \"column\": \"User_ID\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 1716,\n        \"min\": 1000001,\n        \"max\": 1006040,\n        \"num_unique_values\": 3752,\n        \"samples\": [\n          1002014,\n          1003491,\n          1001842\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Cust_name\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 1250,\n        \"samples\": [\n          \"Hallsten\",\n          \"Shubham\",\n          \"Riya\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Product_ID\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 2350,\n        \"samples\": [\n          \"P00133342\",\n          \"P00302142\",\n          \"P00227542\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Gender\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"M\",\n          \"F\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Age Group\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 7,\n        \"samples\": [\n          \"26-35\",\n          \"0-17\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 12,\n        \"min\": 12,\n        \"max\": 92,\n        \"num_unique_values\": 81,\n        \"samples\": [\n          38,\n          28\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Shaadi\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          1,\n          0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"State\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 16,\n        \"samples\": [\n          \"Maharashtra\",\n          \"Andhra Pradesh\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\":

```



```

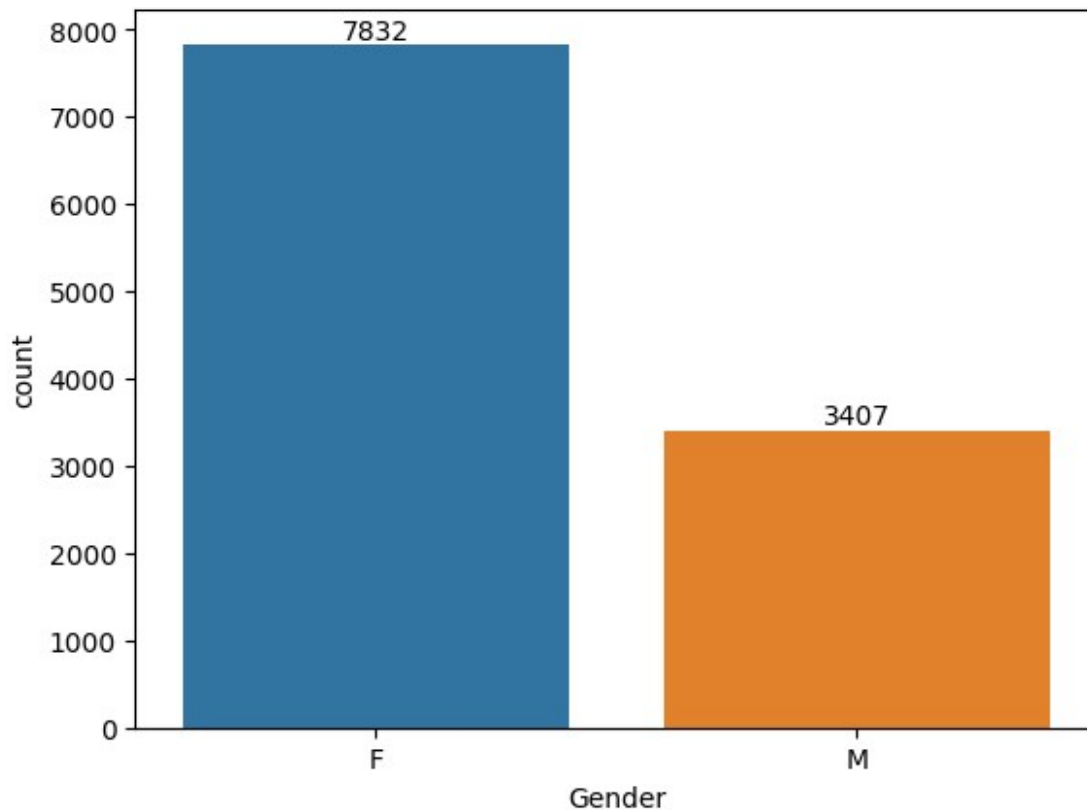
{"properties": {"dtype": "category", "num_unique_values": 7, "samples": ["26-35", "0-17", "18-25", "26-35", "36-45", "46-55", "56-65", "66-75", "76-85", "86-95"], "semantic_type": "Age", "description": "Age group"}, {"column": "Age", "properties": {"dtype": "number", "std": 12, "min": 12, "max": 92, "num_unique_values": 81, "samples": [38, 28, 18, 28, 38, 48, 58, 68, 78, 88], "semantic_type": "Age", "description": "Age group"}, {"column": "Marital_Status", "properties": {"dtype": "number", "std": 0, "min": 0, "max": 1, "num_unique_values": 2, "samples": [1, 0, 1, 0, 1, 0, 1, 0, 1, 0], "semantic_type": "Marital_Status", "description": "Marital status"}, {"column": "State", "properties": {"dtype": "category", "num_unique_values": 16, "samples": ["Maharashtra", "Andhra Pradesh", "Karnataka", "Tamil Nadu", "West Bengal", "Gujarat", "Kerala", "Odisha", "Rajasthan", "Uttar Pradesh", "Haryana", "Madhya Pradesh", "Chhatisgarh", "Jharkhand", "Assam", "Nagaland"], "semantic_type": "State", "description": "State"}, {"column": "Zone", "properties": {"dtype": "category", "num_unique_values": 5, "samples": ["Southern", "Eastern", "Northern", "Western", "Central"], "semantic_type": "Zone", "description": "Zone"}, {"column": "Occupation", "properties": {"dtype": "category", "num_unique_values": 15, "samples": ["IT Sector", "Hospitality", "Retail", "Manufacturing", "Healthcare", "Education", "Finance", "Government", "Non-Profit", "Freelance", "Consulting", "Marketing", "Sales", "Engineering", "Design"], "semantic_type": "Occupation", "description": "Occupation"}, {"column": "Product_Category", "properties": {"dtype": "category", "num_unique_values": 18, "samples": ["Auto", "Hand & Power Tools", "Home Appliances", "Electronics", "Clothing", "Footwear", "Books", "Stationery", "Garden Tools", "Sports Equipment", "Pet Supplies", "Baby Products", "Grocery", "Furniture", "Decor", "Toys", "Grooming Products", "Travel Accessories"], "semantic_type": "Product_Category", "description": "Product category"}, {"column": "Orders", "properties": {"dtype": "number", "std": 1, "min": 1, "max": 4, "num_unique_values": 4, "samples": [3, 4, 1, 2], "semantic_type": "Orders", "description": "Number of orders"}, {"column": "Amount", "properties": {"dtype": "number", "std": 5222, "min": 188, "max": 23952, "num_unique_values": 6583, "samples": [19247, 5293, 1234, 8765, 3456, 9876, 2109, 7654, 4321, 6789, 10101, 20202, 30303, 40404, 50505, 60606, 70707, 80808, 90909], "semantic_type": "Amount", "description": "Amount"}], "type": "dataframe", "variable name": "df"}

```

EDA(Exploratory Data Analysis)

Gender

```
df.columns  
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',  
      'Age',  
      'Marital_Status', 'State', 'Zone', 'Occupation',  
      'Product_Category',  
      'Orders', 'Amount'],  
      dtype='object')  
  
ax=sns.countplot(x='Gender', data=df,hue='Gender')  
  
for bars in ax.containers:  
    ax.bar_label(bars)
```

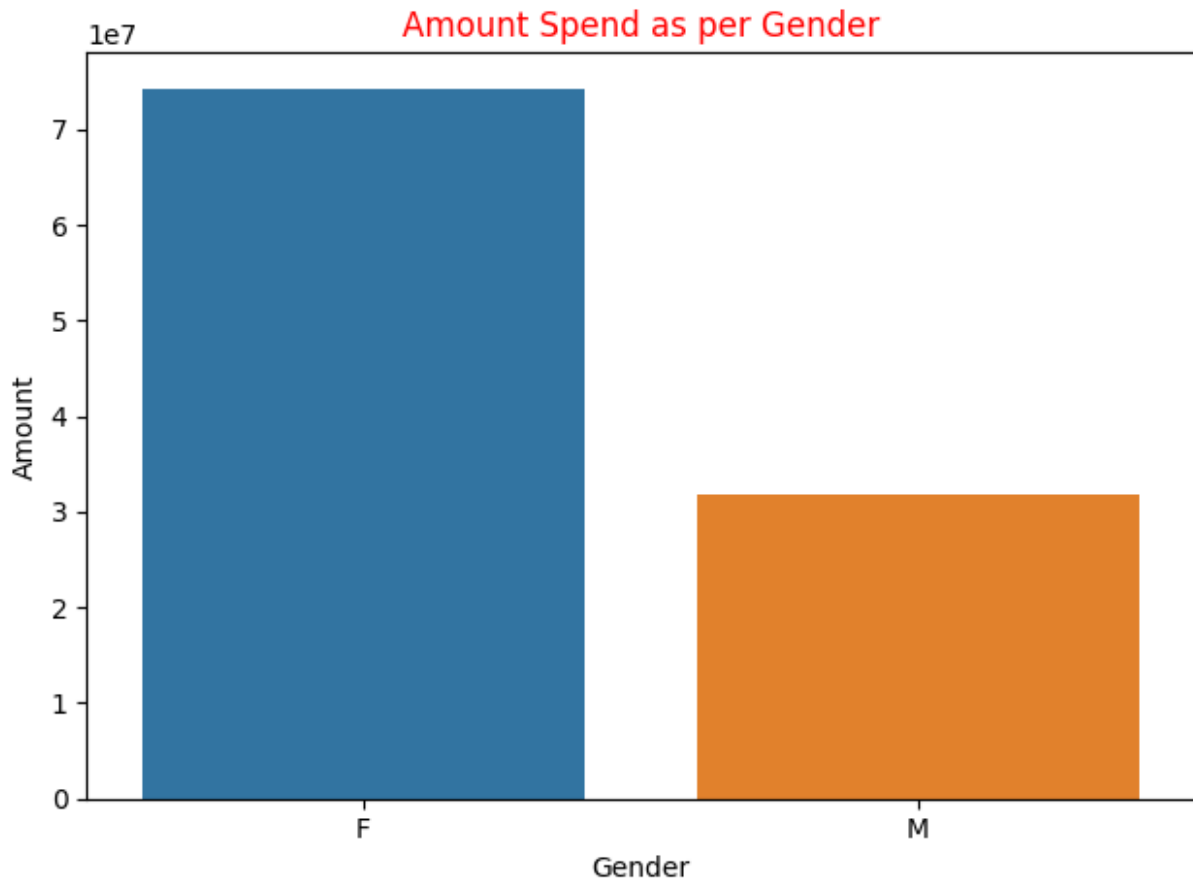


```
# now checking which femal and male spend maximum amount  
sales_gen = df.groupby(['Gender'], as_index=False)  
['Amount'].sum().sort_values(by='Amount', ascending=False)  
  
sns.barplot(x='Gender', y='Amount', data=sales_gen,hue='Gender')  
plt.title('Amount Spend as per Gender',color='red')
```



```
plt.tight_layout()
plt.show()

for bars in ax.containers:
    ax.bar_label(bars)
```



Insights and Observation

From above graphs, we can see that most of the buyers are females and even the purchasing power of females are greater than men

Age

```
df.columns
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',
      'Age',
      'Marital_Status', 'State', 'Zone', 'Occupation',
      'Product_Category',
      'Orders', 'Amount'],
      dtype='object')
```

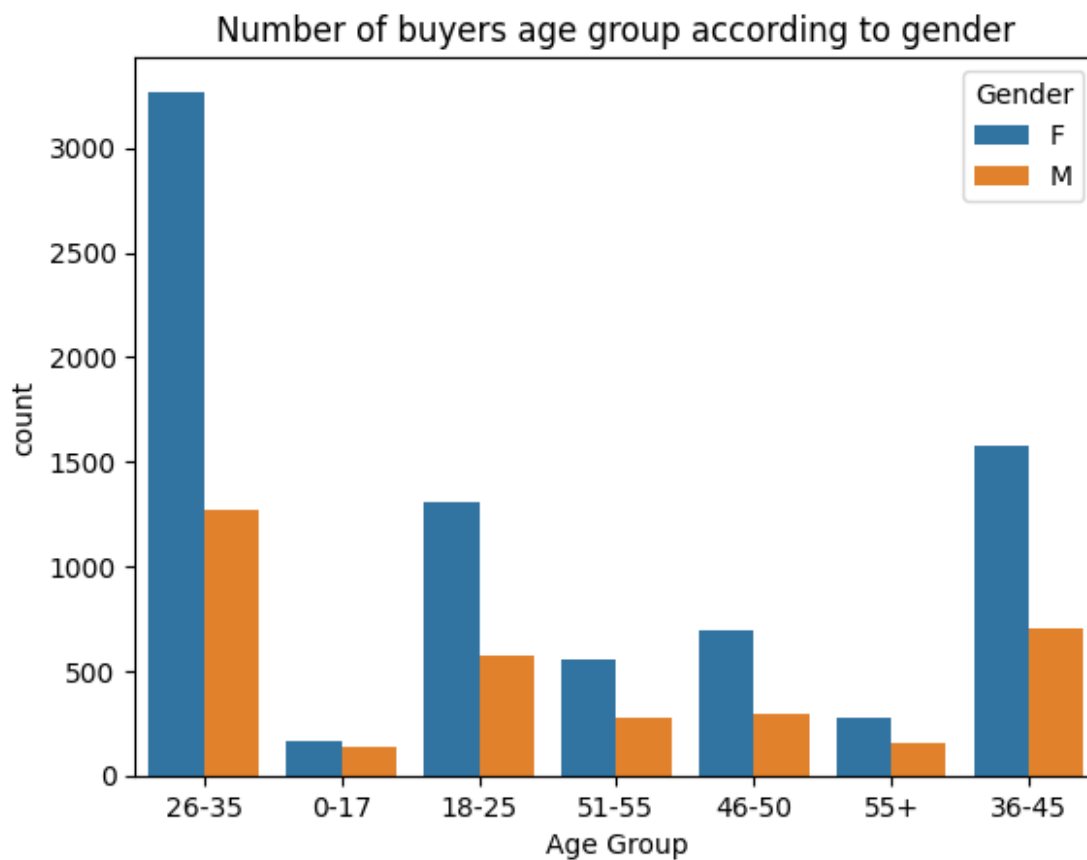
```

df['Age'].dtypes
dtype('int64')

for bars in ax.containers:
    ax.bar_label(bars)

ax=sns.countplot(x='Age Group',hue='Gender', data=df)
plt.title('Number of buyers age group according to gender')
plt.show()

```



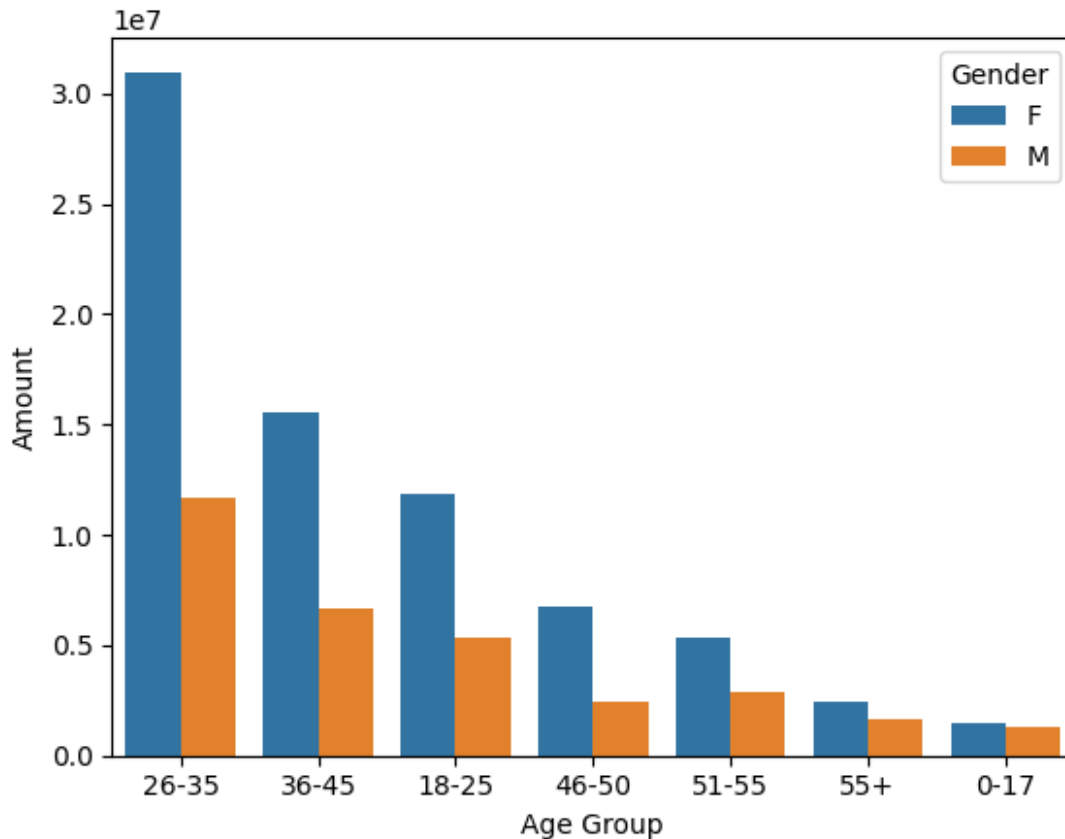
```

# Total Amount vs Total Age
sales_age = df.groupby(['Age Group', 'Gender'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.barplot(x='Age Group', y='Amount', data=sales_age, hue='Gender')

<Axes: xlabel='Age Group', ylabel='Amount'>

```



Insights and Observation

From above graphs we can see that most of the buyers are of age of 26-35 and Gender is Female

State

```
df.columns
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',  
      'Age',  
      'Marital_Status', 'State', 'Zone', 'Occupation',  
      'Product_Category',  
      'Orders', 'Amount'],  
      dtype='object')
```

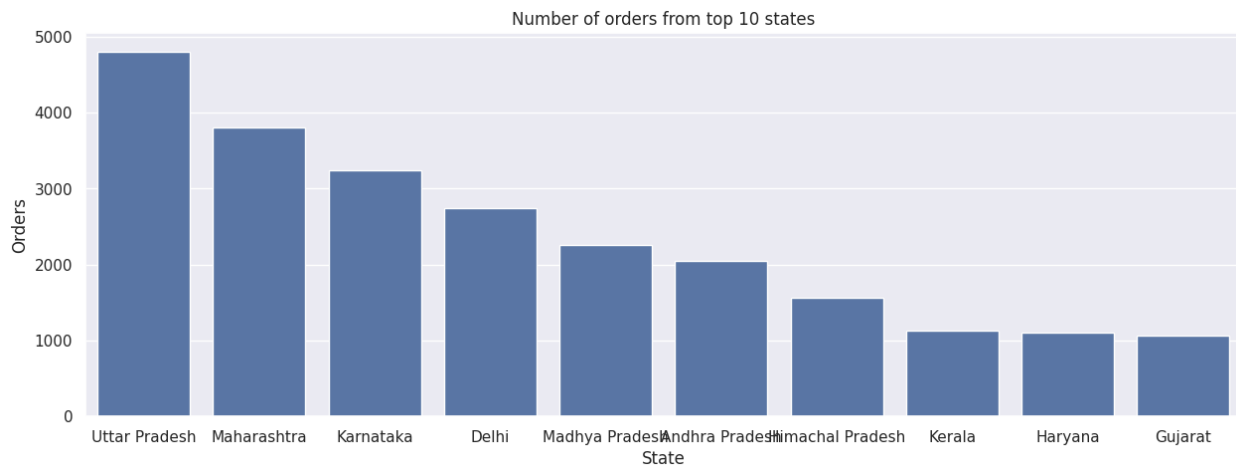
```
df['State'].unique()
```

```
array(['Maharashtra', 'Andhra Pradesh', 'Uttar Pradesh',  
      'Karnataka',  
      'Gujarat', 'Himachal Pradesh', 'Delhi', 'Jharkhand', 'Kerala',  
      'Haryana', 'Madhya Pradesh', 'Bihar', 'Rajasthan',  
      'Uttarakhand',  
      'Telangana', 'Punjab'], dtype=object)
```

```
# Total number of orders from 10 top state
```

```
sales_state=df.groupby(['State'], as_index=False)  
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
```

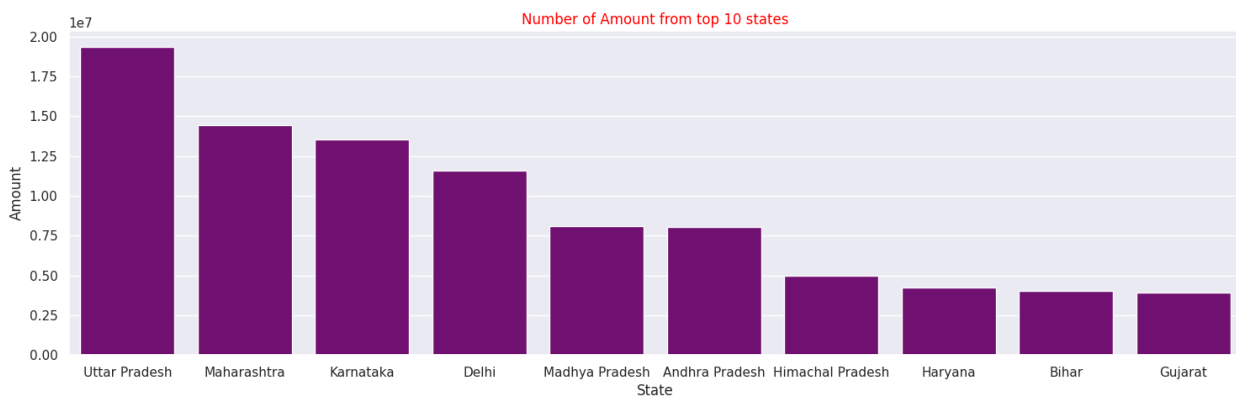
```
sales_state.style.background_gradient(cmap='orange')  
fig, ax = plt.subplots(figsize=(15, 5))  
sns.barplot(x='State', y='Orders', data=sales_state, ax=ax)  
plt.title('Number of orders from top 10 states')  
plt.show()
```



```
# Total amount/sales from top 10 states
```

```
sales_state=df.groupby(['State'], as_index=False)  
['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)
```

```
fig, ax = plt.subplots(figsize=(18, 5))  
sns.barplot(x='State', y='Amount', color='purple', data=sales_state,  
ax=ax)  
plt.title('Number of Amount from top 10 states', color='red')  
plt.show()
```

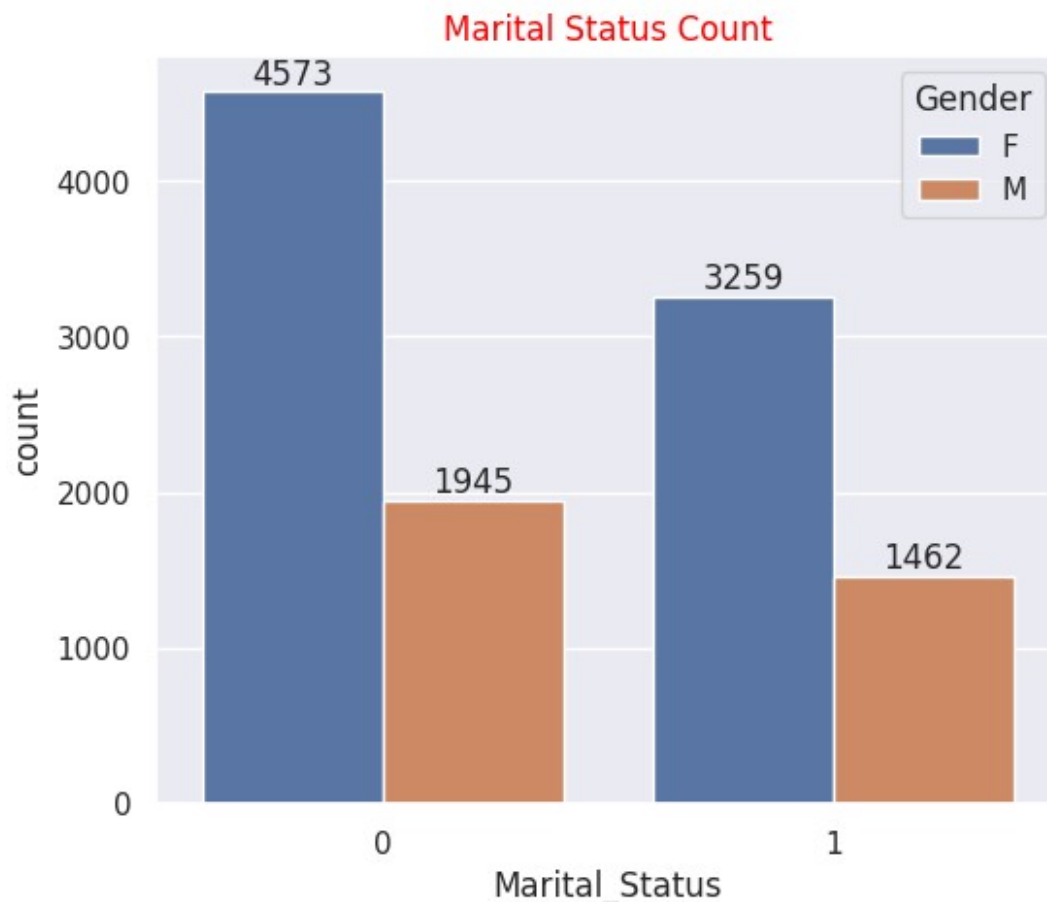


Insights and Observation from State

From above graphs we can see that most of the orders and total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

Marital_Status

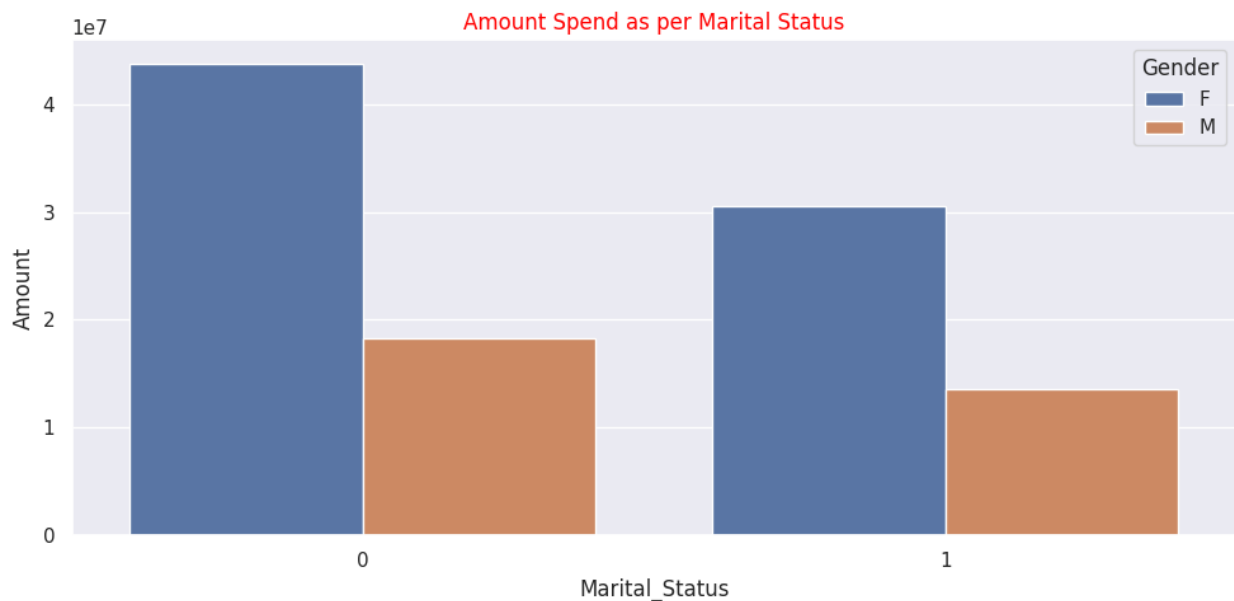
```
df.columns  
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',  
      'Age', 'Marital_Status', 'State', 'Zone', 'Occupation',  
      'Product_Category', 'Orders', 'Amount'],  
      dtype='object')  
  
ax=sns.countplot(x='Marital_Status',hue='Gender',data=df)  
  
sns.set(rc={'figure.figsize':(10,5)})  
plt.title('Marital Status Count',color='red')  
  
for bars in ax.containers:  
    ax.bar_label(bars)
```



```
# now checking based on amount
```

```
sales_amount=df.groupby(['Marital_Status','Gender'], as_index=False)  
['Amount'].sum().sort_values(by='Amount', ascending=False)
```

```
sns.barplot(x='Marital_Status', y='Amount',  
data=sales_amount,hue='Gender')  
plt.title('Amount Spend as per Marital Status',color='red')  
plt.tight_layout()  
plt.show()
```



Insights and Observation from Marital_Status

From above graphs we can see that most of the orders and total sales/amount are from Married People and in Married people mostly are females. Who spends most of the amount

Occupation

```
df.columns
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',  
      'Age',  
      'Marital_Status', 'State', 'Zone', 'Occupation',  
      'Product_Category',  
      'Orders', 'Amount'],  
      dtype='object')
```

```
df['Occupation'].unique()
```

```
array(['Healthcare', 'Govt', 'Automobile', 'Construction',  
      'Food Processing', 'Lawyer', 'Media', 'Banking', 'Retail',
```

```

        'IT Sector', 'Aviation', 'Hospitality', 'Agriculture',
        'Textile',
        'Chemical'], dtype=object)

```

```

# create count plot based on Occupation

```

```

sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(x='Occupation',data=df,palette='plasma')
plt.title('Occupation Count',color='red')
plt.tight_layout()

```

```

for bars in ax.containers:
    ax.bar_label(bars)

```

```

/tmp/ipython-input-155-205310627.py:4: FutureWarning:

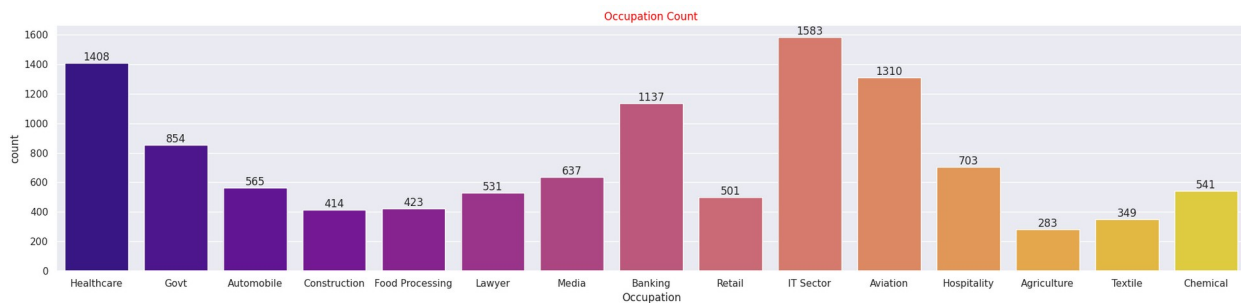
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```

ax=sns.countplot(x='Occupation',data=df,palette='plasma')

```



```

# Based on Occupation purchasing power checking

```

```

sales_amount=df.groupby(['Occupation'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)

```

```

sns.barplot(x='Occupation', y='Amount', data=sales_amount,
palette='hls')
plt.title('Amount Spend as per Occupation',color='red')
plt.tight_layout()
plt.show()

```

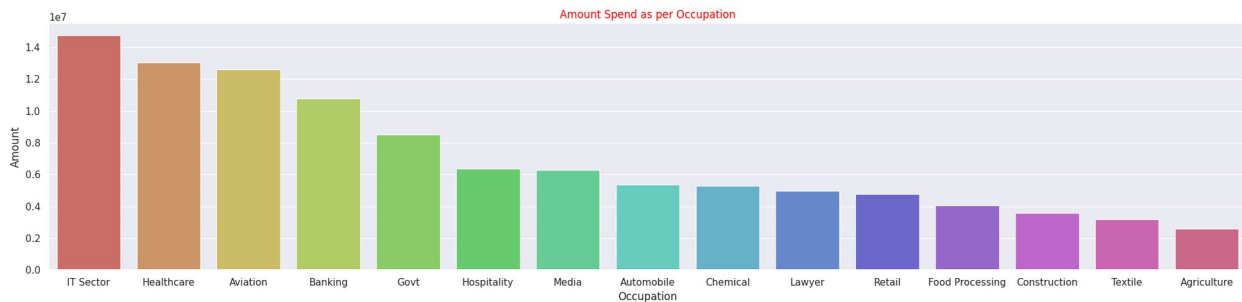
```

/tmp/ipython-input-148-2788981896.py:5: FutureWarning:

```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='Occupation', y='Amount', data=sales_amount,
palette='hls')
```



Insights and Observation

From the above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation **Sectors**

Seaborn Color Palettes

You can find a comprehensive list of all available color palettes in the official Seaborn documentation. This will allow you to see visual examples of each palette and choose the one that best suits your needs.

[Seaborn Color Palette Documentation](#)

Here are some of the most commonly used palettes:

- **Qualitative:** deep, muted, pastel, bright, dark, colorblind
- **Sequential:** viridis, plasma, inferno, magma, cividis
- **Diverging:** vlag, icefire, coolwarm, bwr

Product Category

```
df.columns
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',
      'Age',
      'Marital_Status', 'State', 'Zone', 'Occupation',
      'Product_Category',
      'Orders', 'Amount'],
      dtype='object')
```

```
df['Product_Category'].unique()
```

```
array(['Auto', 'Hand & Power Tools', 'Stationery', 'Tupperware',
      'Footwear & Shoes', 'Furniture', 'Food', 'Games & Toys',
      'Sports Products', 'Books', 'Electronics & Gadgets', 'Decor',
      'Clothing & Apparel', 'Beauty', 'Household items', 'Pet Care',
      'Veterinary', 'Office'], dtype=object)
```



```
# finding count based on Product category
```

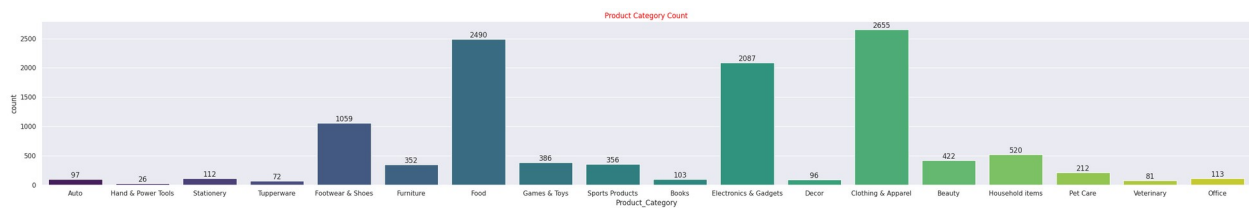
```
sns.set(rc={'figure.figsize':(30,5)})
ax=sns.countplot(x='Product_Category',data=df,palette='viridis')
plt.title('Product Category Count',color='red')
plt.tight_layout()
```

```
for bars in ax.containers:
    ax.bar_label(bars)
```

/tmp/ipython-input-162-1886573800.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

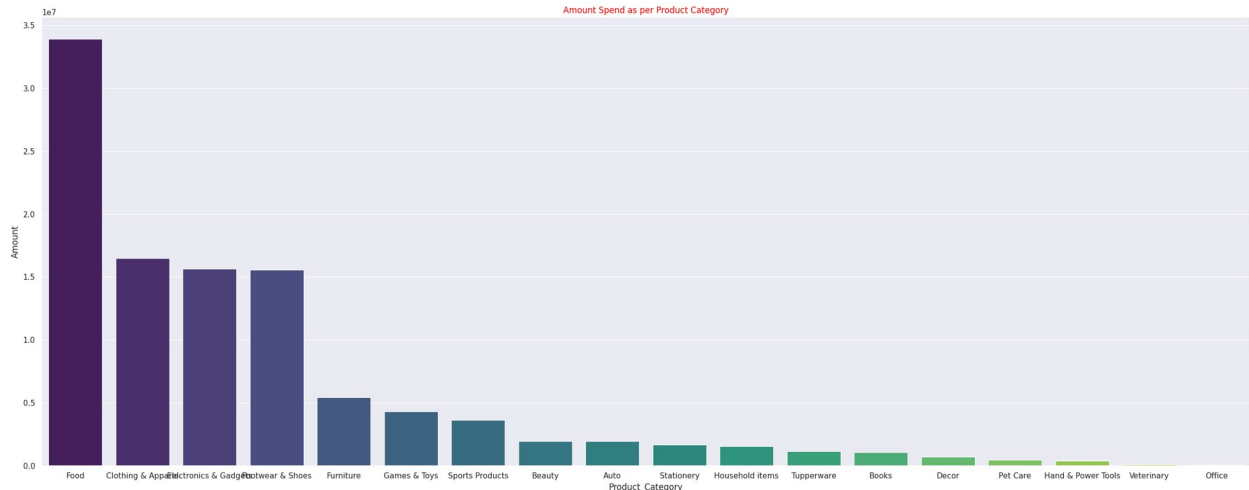
```
ax=sns.countplot(x='Product_Category',data=df,palette='viridis')
```



```
# checking amount spend based on product category
```

```
sales_amount=df.groupby(['Product_Category'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)
```

```
sns.set(rc={'figure.figsize':(25,10)})
sns.barplot(x='Product_Category', y='Amount', data=sales_amount,
palette='viridis',legend=False,hue='Product_Category')
plt.title('Amount Spend as per Product Category',color='red')
plt.tight_layout()
plt.show()
```



Insights and Observation from Product Category

From the above graphs we can see that most of orders done from Clothing and Apparel. While most of the amount spend on order done for Food product category

Product ID

```
df.columns
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',
      'Age',
      'Marital_Status', 'State', 'Zone', 'Occupation',
      'Product_Category',
      'Orders', 'Amount'],
      dtype='object')
```

```
df['Product_ID'].unique()
```

```
array(['P00125942', 'P00110942', 'P00118542', ..., 'P00307142',
      'P00044742', 'P00296942'], dtype=object)
```

```
df.head(2)
```

```
{"summary":{"name": "df", "rows": 11239, "fields":
[{"column": "User_ID", "properties": {"dtype": "number", "std": 1716, "min": 1000001, "max": 1006040, "num_unique_values": 3752, "samples": [1002014, 1003491, 1001842]}, "semantic_type": "\"", "description": "\"\""}], "column": "Cust_name", "properties": {"dtype": "category", "num_unique_values": 1250, "samples": ["Hallsten", "Shubham", "Riya"]}, "semantic_type": "\"", "description": "\"\""}], "column": "Product_ID", "properties": {"dtype":
```

```

\"category\", \n          \"num_unique_values\": 2350, \n
\"samples\": [ \n          \"P00133342\", \n          \"P00302142\", \n
\"P00227542\" \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\" \n          } \n          }, \n          { \n          \"column\":
\"Gender\", \n          \"properties\": { \n          \"dtype\":
\"category\", \n          \"num_unique_values\": 2, \n          \"samples\":
[ \n          \"M\", \n          \"F\" \n          ], \n
\"semantic_type\": \"\", \n          \"description\": \"\" \n          } \n
          }, \n          { \n          \"column\": \"Age Group\", \n
\"properties\": { \n          \"dtype\": \"category\", \n
\"num_unique_values\": 7, \n          \"samples\": [ \n          \"26-
35\", \n          \"0-17\" \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\" \n          } \n          }, \n          { \n
\"column\": \"Age\", \n          \"properties\": { \n          \"dtype\":
\"number\", \n          \"std\": 12, \n          \"min\": 12, \n
\"max\": 92, \n          \"num_unique_values\": 81, \n          \"samples\":
[ \n          38, \n          28 \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\" \n          } \n          }, \n          { \n
\"column\": \"Marital_Status\", \n          \"properties\": { \n
\"dtype\": \"number\", \n          \"std\": 0, \n          \"min\": 0, \n
\"max\": 1, \n          \"num_unique_values\": 2, \n          \"samples\":
[ \n          1, \n          0 \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\" \n          } \n          }, \n          { \n
\"column\": \"State\", \n          \"properties\": { \n          \"dtype\":
\"category\", \n          \"num_unique_values\": 16, \n
\"samples\": [ \n          \"Maharashtra\", \n          \"Andhra\\
u00a0Pradesh\" \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\" \n          } \n          }, \n          { \n          \"column\":
\"Zone\", \n          \"properties\": { \n          \"dtype\": \"category\", \n
\"num_unique_values\": 5, \n          \"samples\": [ \n
\"Southern\", \n          \"Eastern\" \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\" \n          } \n          }, \n          { \n
\"column\": \"Occupation\", \n          \"properties\": { \n          \"dtype\":
\"category\", \n          \"num_unique_values\": 15, \n          \"samples\": [ \n
\"IT
Sector\", \n          \"Hospitality\" \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\" \n          } \n          }, \n          { \n
\"column\": \"Product_Category\", \n          \"properties\": { \n          \"dtype\":
\"category\", \n          \"num_unique_values\": 18, \n          \"samples\": [ \n
\"Auto\", \n          \"Hand & Power Tools\" \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\" \n          } \n          }, \n          { \n
\"column\": \"Orders\", \n          \"properties\":
{ \n          \"dtype\": \"number\", \n          \"std\": 1, \n
\"min\": 1, \n          \"max\": 4, \n          \"num_unique_values\": 4, \n
\"samples\": [ \n          3, \n          4 \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\" \n          } \n          }, \n          { \n
\"column\": \"Amount\", \n          \"properties\":
{ \n          \"dtype\": \"number\", \n          \"std\": 5222, \n

```

```

{"min": 188, "max": 23952, "num_unique_values": 6583, "samples": [19247, 5293], "semantic_type": "\"", "description": "\"\""}
{"type": "dataframe", "variable_name": "df"}

```

```
df.groupby(['Product_ID', 'Product_Category'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
```

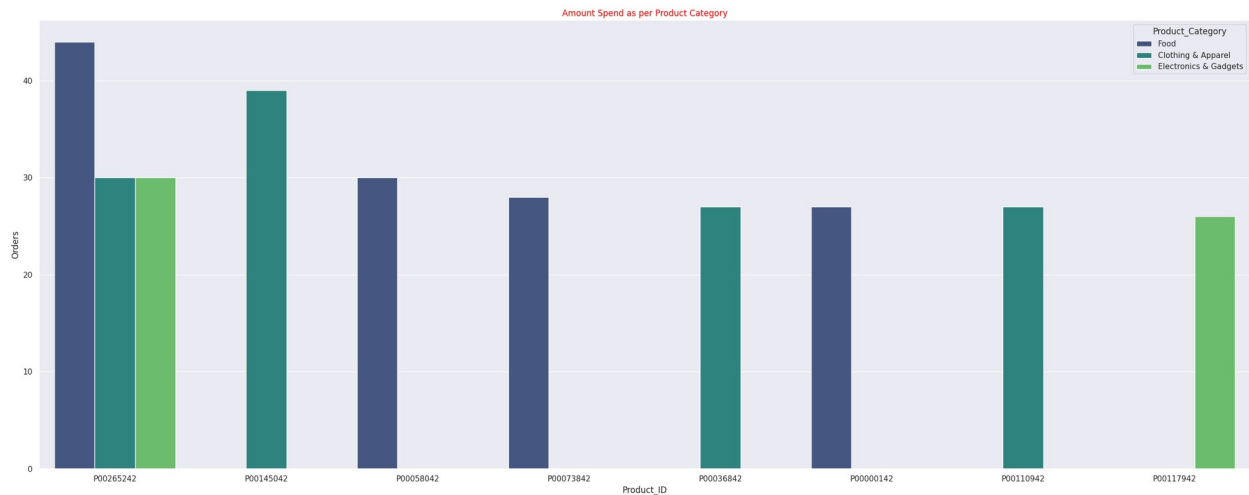
```
{
  "summary": {
    "name": "df",
    "rows": 10,
    "fields": [
      {
        "column": "Product_ID",
        "properties": {
          "dtype": "string",
          "num_unique_values": 8,
          "samples": [
            "P00145042",
            "P00000142",
            "P00265242"
          ],
          "semantic_type": "",
          "description": "",
          "column": "Product_Category",
          "properties": {
            "dtype": "category",
            "num_unique_values": 3,
            "samples": [
              "Food",
              "Clothing & Apparel",
              "Electronics & Gadgets"
            ],
            "semantic_type": "",
            "description": "",
            "column": "Orders",
            "properties": {
              "dtype": "number",
              "std": 5,
              "min": 26,
              "max": 44,
              "num_unique_values": 6,
              "samples": [
                44,
                39,
                26
              ],
              "semantic_type": "",
              "description": ""
            }
          }
        ]
      }
    ],
    "type": "dataframe"
  }
}
```

```
# based on product id group by orders and check count for top 10
product id
```

```
sales_amount=df.groupby(['Product_ID','Product_Category'],
as_index=False)['Orders'].sum().sort_values(by='Orders',
ascending=False).head(10)
```

```
# create plot
```

```
sns.set(rc={'figure.figsize':(25,10)})
sns.barplot(x='Product_ID', y='Orders', data=sales_amount,
palette='viridis',legend=True,hue='Product_Category')
plt.title('Amount Spend as per Product Category',color='red')
plt.tight layout()
```

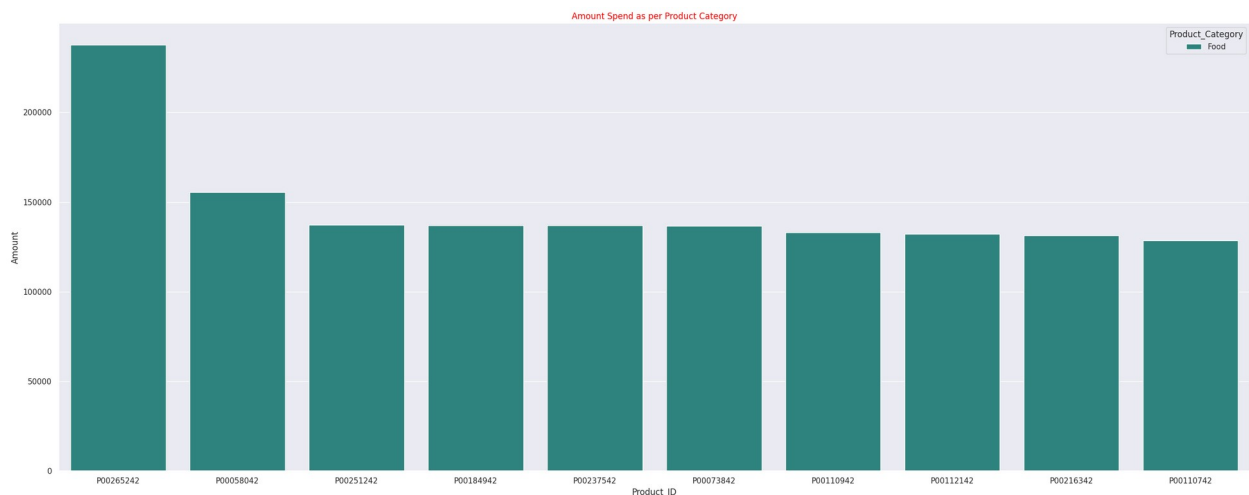


based on product id finding which product id use to spend the maximum amount

```
sales_amount=df.groupby(['Product_ID','Product_Category'],
as_index=False)['Amount'].sum().sort_values(by='Amount',
ascending=False).head(10)
```

create plot

```
sns.set(rc={'figure.figsize':(25,10)})
sns.barplot(x='Product_ID', y='Amount', data=sales_amount,
palette='viridis',legend=True,hue='Product_Category')
plt.title('Amount Spend as per Product Category',color='red')
plt.tight_layout()
```



Conclusion

Married women age group 26-35 yrs from UP, Maharastra and Karnatka working in IT, Healthcare and Aviation are more likely buy products from food, Clothing and Electronics category.

