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

```
df = pd.read_csv('Python Project Data - Supermarket Sales.csv')
df
```

	Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Paym
0	750-67-8428	A	1	0	0	Normal	Male	Health and beauty	74.69	7	26.1415	NaN	1/5/2019	13:08	Ew
1	226-31-3081	C	0	1	0	Normal	Male	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	C
2	631-41-3108	A	1	0	0	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Cr
3	123-19-1176	A	1	0	0	Normal	Male	Health and beauty	58.22	8	NaN	489.0480	1/27/2019	8 - 30 PM	Ew
4	373-73-7910	A	1	0	0	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ew
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1001	861-77-0145	C	0	1	0	Member	Male	Electronic accessories	81.97	10	40.9850	860.6850	3/3/2019	14:30	C

```
print(df.head())
```

```
     Invoice ID Branch  Yangon  Naypyitaw  Mandalay Customer type Gender  \
0  750-67-8428      A       1          0         0        Normal   Male
1  226-31-3081      C       0          1         0        Normal   Male
2  631-41-3108      A       1          0         0        Normal   Male
3  123-19-1176      A       1          0         0        Normal   Male
4  373-73-7910      A       1          0         0        Normal   Male

               Product line Unit price  Quantity   Tax 5%     Total       Date  \
0       Health and beauty      74.69         7  26.1415       NaN   1/5/2019
1  Electronic accessories      15.28         5   3.8200   80.2200   3/8/2019
2      Home and lifestyle      46.33         7  16.2155  340.5255   3/3/2019
3       Health and beauty      58.22         8      NaN  489.0480  1/27/2019
4       Sports and travel      86.31         7  30.2085  634.3785   2/8/2019

         Time      Payment  Rating
0      13:08      Ewallet     9.1
1      10:29         Cash     9.6
2      13:23  Credit card     7.4
3  8 - 30 PM      Ewallet     8.4
4      10:37      Ewallet     5.3
```

Start coding or generate with AI.

Start coding or generate with AI.

```
print(df.head())
```

```
     Invoice ID Branch  Yangon  Naypyitaw  Mandalay Customer type Gender  \
0  750-67-8428      A       1          0         0        Normal   Male
1  226-31-3081      C       0          1         0        Normal   Male
2  631-41-3108      A       1          0         0        Normal   Male
3  123-19-1176      A       1          0         0        Normal   Male
4  373-73-7910      A       1          0         0        Normal   Male

               Product line Unit price  Quantity   Tax 5%     Total       Date  \
0       Health and beauty      74.69         7  26.1415       NaN   1/5/2019
1  Electronic accessories      15.28         5   3.8200   80.2200   3/8/2019
2      Home and lifestyle      46.33         7  16.2155  340.5255   3/3/2019
3       Health and beauty      58.22         8      NaN  489.0480  1/27/2019
4       Sports and travel      86.31         7  30.2085  634.3785   2/8/2019
```

```
           Time      Payment  Rating
0      13:08      Ewallet     9.1
1      10:29         Cash     9.6
2      13:23  Credit card     7.4
3  8 - 30 PM      Ewallet     8.4
4      10:37      Ewallet     5.3
```

```
print(df.describe())
```

```
             Yangon    Naypyitaw     Mandalay     Quantity      Tax 5%  \
count  1006.000000  1006.000000  1006.000000  1006.000000  997.000000
mean      0.338966     0.329026     0.332008     5.469185   15.479682
std       0.473594     0.470093     0.471168     3.014153   11.728320
min       0.000000     0.000000     0.000000    -8.000000    0.508500
25%       0.000000     0.000000     0.000000     3.000000    5.986500
50%       0.000000     0.000000     0.000000     5.000000   12.227500
75%       1.000000     1.000000     1.000000     8.000000   22.720500
max       1.000000     1.000000     1.000000    10.000000   49.650000

             Total       Rating
count  1003.000000  1006.000000
mean    322.734689     7.056163
std     245.865964     3.318751
min      10.678500     4.000000
25%     123.789750     5.500000
50%     254.016000     7.000000
75%     471.009000     8.500000
max    1042.650000    97.000000
```

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1006 entries, 0 to 1005
Data columns (total 16 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice ID     1006 non-null   object
 1   Branch         1006 non-null   object
 2   Yangon         1006 non-null   int64
 3   Naypyitaw      1006 non-null   int64
 4   Mandalay       1006 non-null   int64
 5   Customer type  1006 non-null   object
 6   Gender         1006 non-null   object
 7   Product line   1006 non-null   object
 8   Unit price     1006 non-null   object
 9   Quantity       1006 non-null   int64
 10  Tax 5%         997 non-null    float64
 11  Total          1003 non-null   float64
 12  Date           1006 non-null   object
 13  Time           1006 non-null   object
 14  Payment        1006 non-null   object
 15  Rating         1006 non-null   float64
dtypes: float64(3), int64(4), object(9)
memory usage: 125.9+ KB
None
```

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

```
Invoice ID       0
Branch           0
Yangon           0
Naypyitaw        0
Mandalay         0
Customer type    0
Gender           0
Product line     0
Unit price       0
Quantity         0
Tax 5%           9
Total            3
Date             0
Time             0
Payment          0
Rating           0
dtype: int64
```

```
df.fillna(method='ffill', inplace=True)
```

```
<ipython-input-11-cc469a1f56c1>:1: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use c
  df.fillna(method='ffill', inplace=True)
```

```
df=df.interpolate(method='linear')
df
```

```
<ipython-input-24-827cffa7486e>:1: FutureWarning: DataFrame.interpolate with object dtype is deprecated and will raise in a future versi
  df=df.interpolate(method='linear')
```

	invoice_id	branch	yangon	naypyitaw	mandalay	customer_type	gender	product_line	unit_price	quantity	tax_5%	total	c
0	750-67-8428	A	1	0	0	Normal	Male	Health and beauty	74.69	7	26.1415	NaN	1/5/2
1	226-31-3081	C	0	1	0	Normal	Male	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2
2	631-41-3108	A	1	0	0	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2
3	123-19-1176	A	1	0	0	Normal	Male	Health and beauty	58.22	8	16.2155	489.0480	1/27/2
4	373-73-7910	A	1	0	0	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2
...	...	...	...	...	...	...	...	...	...	...	...	...	
995	233-67-5758	C	0	1	0	Normal	Male	Health and beauty	40.35	1	2.0175	42.3675	1/29/2
996	303-96-2227	B	0	0	1	Normal	Female	Home and lifestyle	97.38	10	48.6900	1022.4900	3/2/2

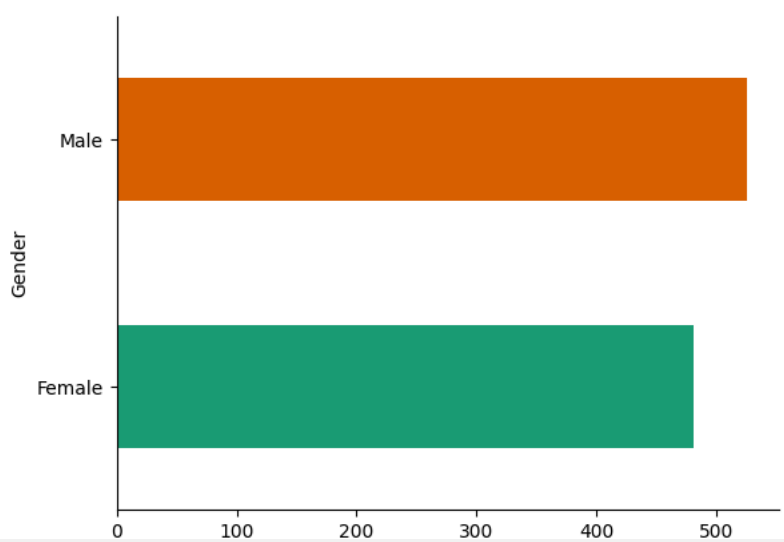
```
df.drop_duplicates(inplace=True)
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
```

```
f = pd.get_dummies(df, drop_first=True)
```

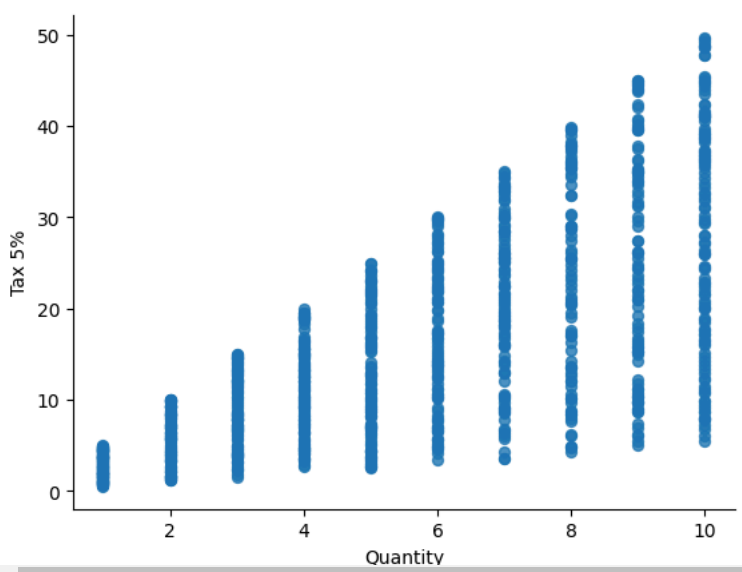
```
plt.figure(figsize=(10, 6))
```

```
<Figure size 1000x600 with 0 Axes>
```

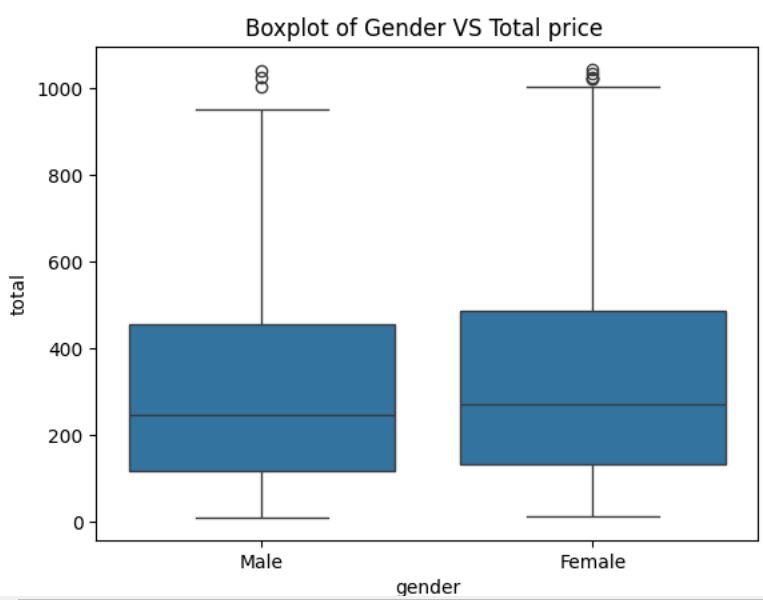
```
df.groupby('Gender').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```



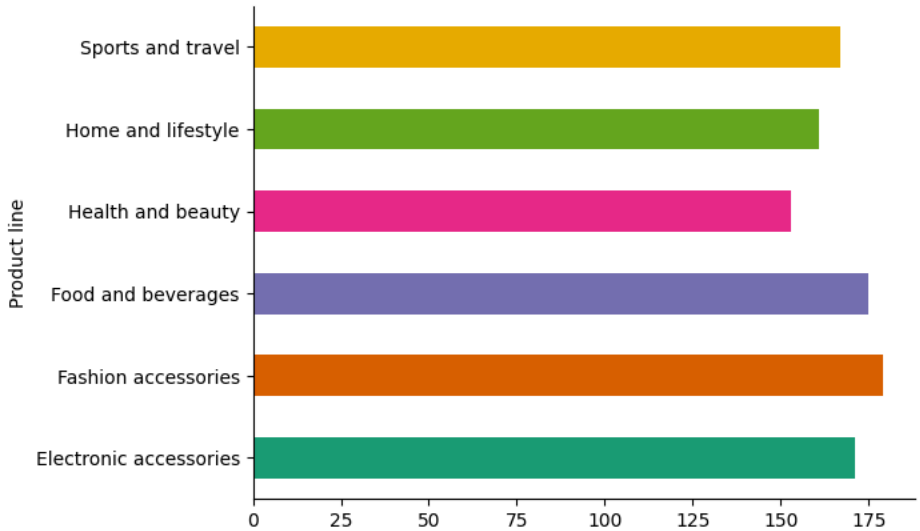

```
df.plot(kind='scatter', x='Quantity', y='Tax 5%', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
sns.boxplot(x='gender', y='total', data=df)
plt.title('Boxplot of Gender VS Total price ')
plt.show()
```




```
df.groupby('Product line').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
data_wrangling_report = """
Data Wrangling Report:
- Loaded dataset and checked for missing values
- Cleaned and transformed data
- Handled missing values and outliers
- Encoded categorical variables
- Saved cleaned dataset
"""


business_insights_report = """
Business Insights Report:
- Identified key trends and patterns
- Found correlations between variables
- Provided visual insights to support decision-making
"""


with open('data_wrangling_report.pdf', 'w') as f:
    f.write(data_wrangling_report)


with open('business_insights_report.pdf', 'w') as f:
    f.write(business_insights_report)
```