

```
In [3]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.cluster import KMeans
```

```
In [3]: df = pd.read_csv(r"C:\Users\Surendranath TV\Desktop\customer satisfaction\data\c
```

```
In [3]: import pandas as pd

# Load dataset
df = pd.read_csv(r"C:\Users\Surendranath TV\Desktop\customer satisfaction\data\c

# Check shape
print("Rows:", df.shape[0], " Columns:", df.shape[1])

# Peek at first 5 rows
df.head()
```

Rows: 8469 Columns: 17

Out[3]:

	Ticket ID	Customer Name	Customer Email	Customer Age	Customer Gender	Product Purchased	D Pur
0	1	Marisa Obrien	carrollallison@example.com	32	Other	GoPro Hero	202
1	2	Jessica Rios	clarkeashley@example.com	42	Female	LG Smart TV	202
2	3	Christopher Robbins	gonzalestracy@example.com	48	Other	Dell XPS	202
3	4	Christina Dillon	bradleyolson@example.org	27	Female	Microsoft Office	202
4	5	Alexander Carroll	bradleymark@example.com	67	Female	Autodesk AutoCAD	202

```
In [5]: df.isna().sum()
```

```
Out[5]: Ticket ID          0  
Customer Name           0  
Customer Email          0  
Customer Age             0  
Customer Gender          0  
Product Purchased        0  
Date of Purchase         0  
Ticket Type              0  
Ticket Subject            0  
Ticket Description         0  
Ticket Status             0  
Resolution                5700  
Ticket Priority           0  
Ticket Channel            0  
First Response Time       2819  
Time to Resolution        5700  
Customer Satisfaction Rating 5700  
dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 8469 entries, 0 to 8468  
Data columns (total 17 columns):  
 #   Column           Non-Null Count  Dtype    
 ---  --    
 0   Ticket ID        8469 non-null    int64    
 1   Customer Name    8469 non-null    object   
 2   Customer Email   8469 non-null    object   
 3   Customer Age     8469 non-null    int64    
 4   Customer Gender  8469 non-null    object   
 5   Product Purchased 8469 non-null    object   
 6   Date of Purchase 8469 non-null    object   
 7   Ticket Type      8469 non-null    object   
 8   Ticket Subject    8469 non-null    object   
 9   Ticket Description 8469 non-null    object   
 10  Ticket Status     8469 non-null    object   
 11  Resolution        2769 non-null    object   
 12  Ticket Priority   8469 non-null    object   
 13  Ticket Channel    8469 non-null    object   
 14  First Response Time 5650 non-null    object   
 15  Time to Resolution 2769 non-null    object   
 16  Customer Satisfaction Rating 2769 non-null    float64  
dtypes: float64(1), int64(2), object(14)  
memory usage: 1.1+ MB
```

```
In [9]: df.describe()
```

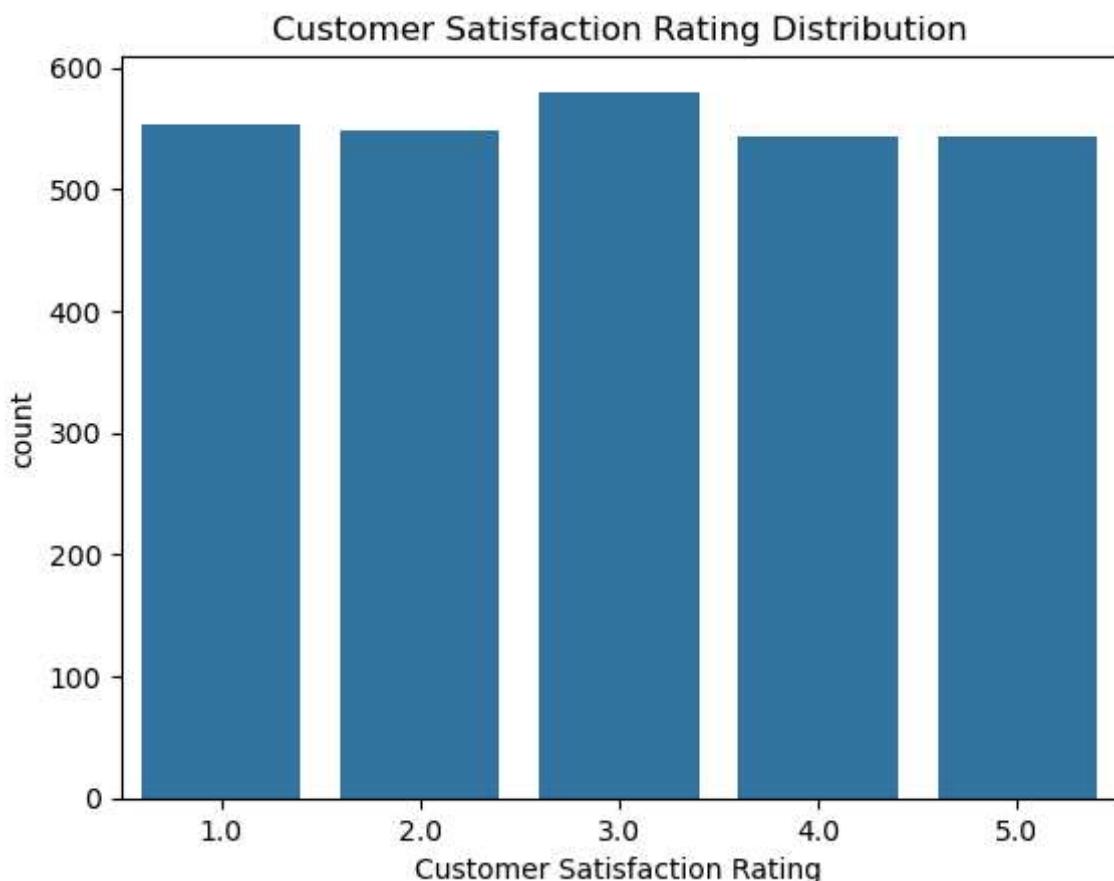
Out[9]:

	Ticket ID	Customer Age	Customer Satisfaction Rating
count	8469.000000	8469.000000	2769.000000
mean	4235.000000	44.026804	2.991333
std	2444.934048	15.296112	1.407016
min	1.000000	18.000000	1.000000
25%	2118.000000	31.000000	2.000000
50%	4235.000000	44.000000	3.000000
75%	6352.000000	57.000000	4.000000
max	8469.000000	70.000000	5.000000

In [11]:

```
import seaborn as sns
import matplotlib.pyplot as plt

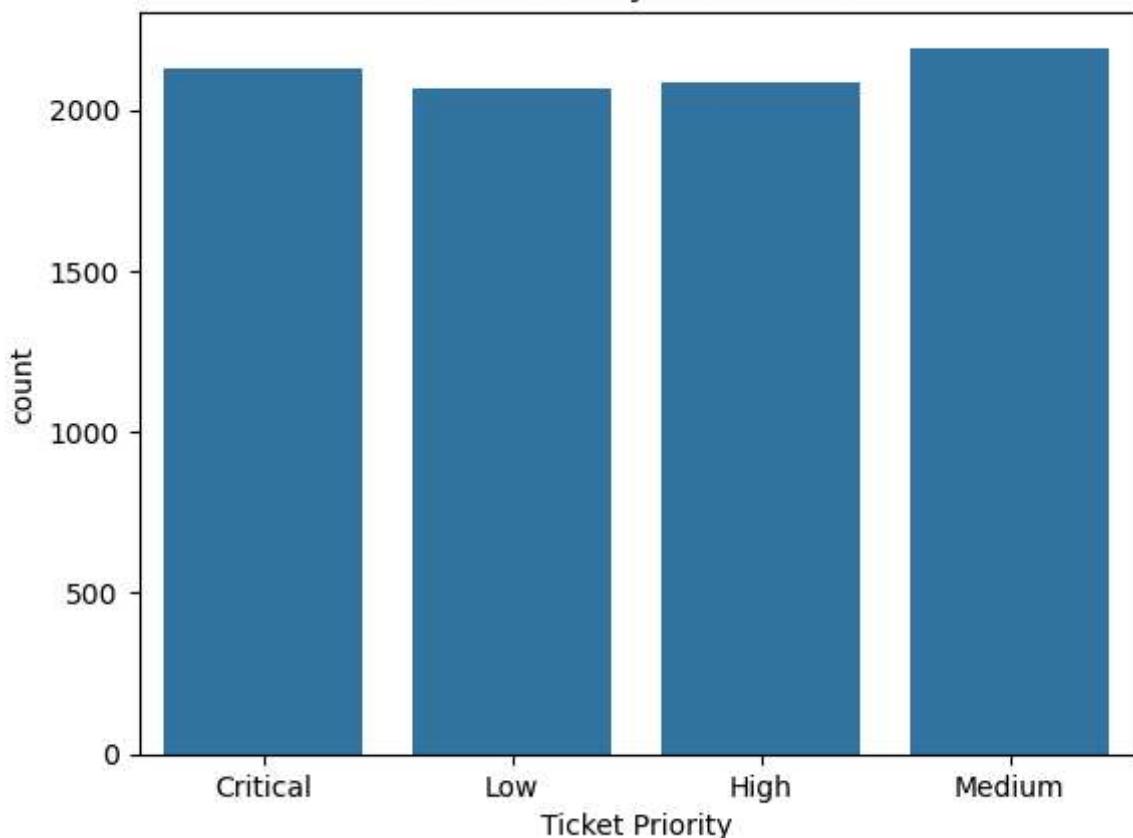
sns.countplot(x='Customer Satisfaction Rating', data=df)
plt.title("Customer Satisfaction Rating Distribution")
plt.show()
```



In [13]:

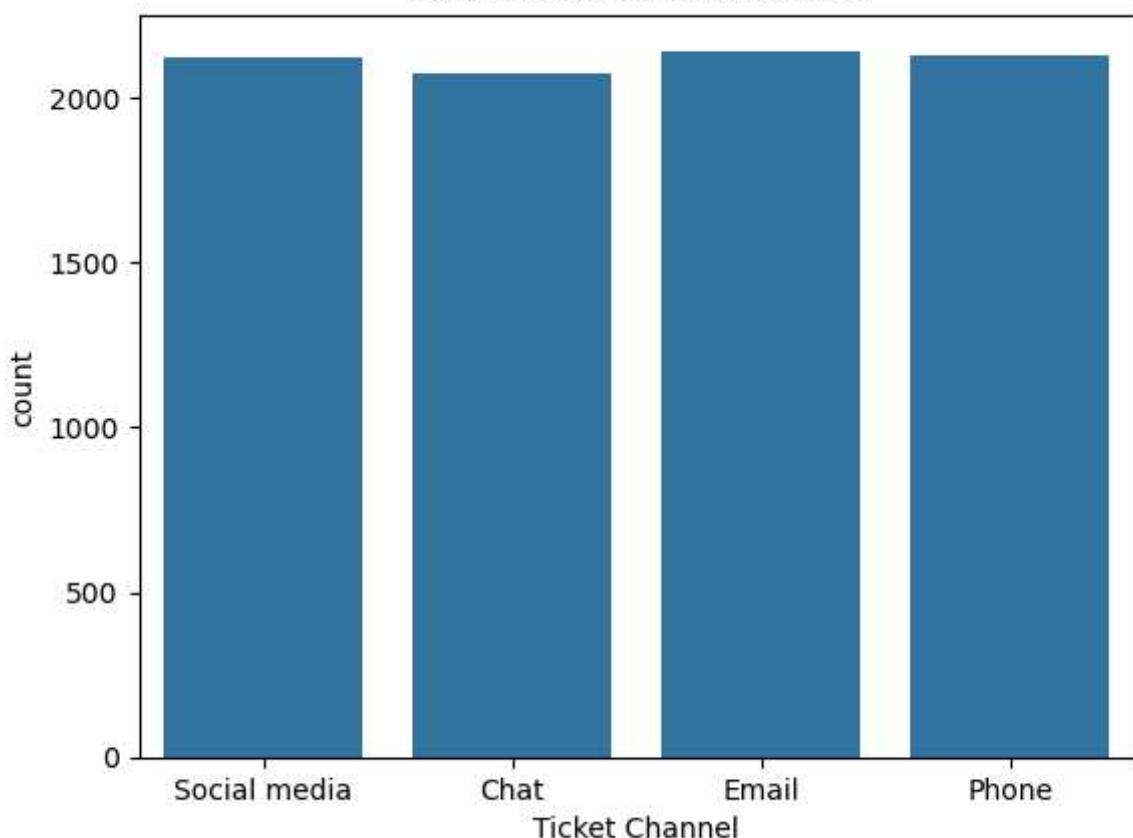
```
sns.countplot(x='Ticket Priority', data=df)
plt.title("Ticket Priority Distribution")
plt.show()
```

Ticket Priority Distribution

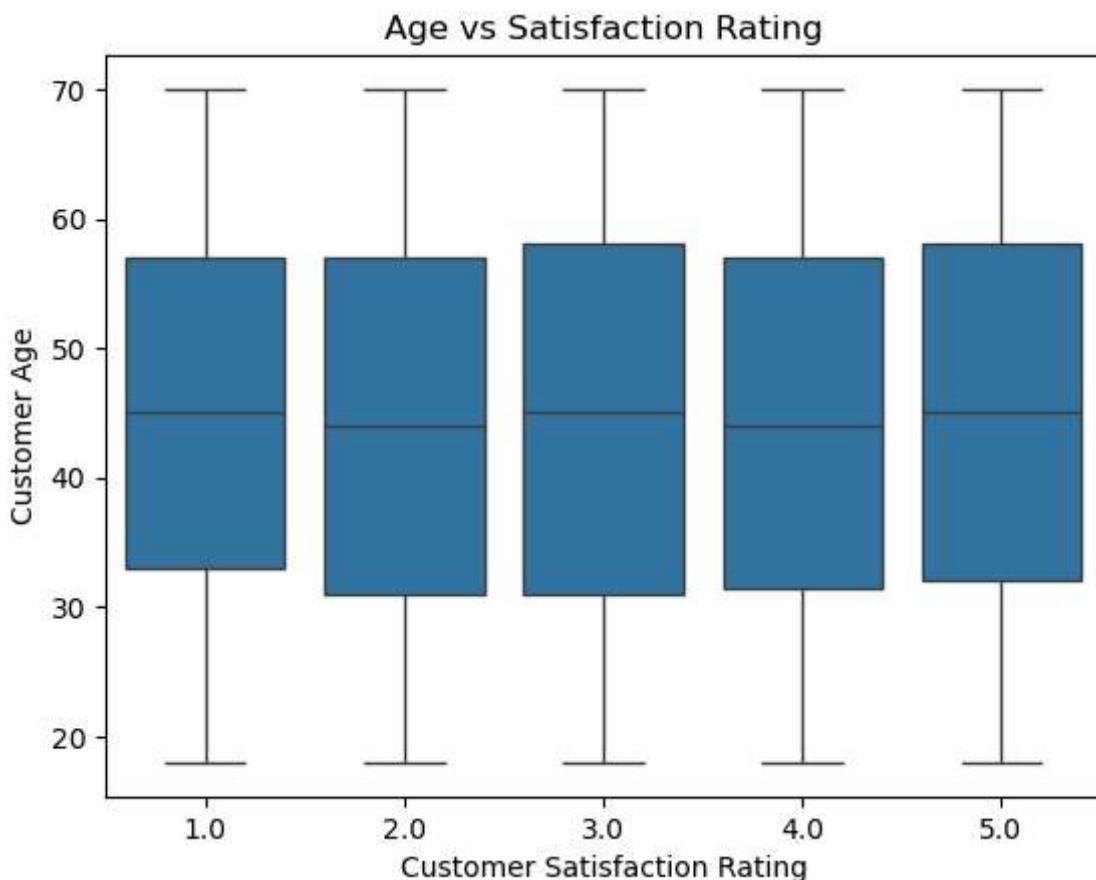


```
In [15]: sns.countplot(x='Ticket Channel', data=df)
plt.title("Ticket Channel Distribution")
plt.show()
```

Ticket Channel Distribution



```
In [17]: sns.boxplot(x='Customer Satisfaction Rating', y='Customer Age', data=df)
plt.title("Age vs Satisfaction Rating")
plt.show()
```



```
In [19]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_ma
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [3]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.cluster import KMeans
```

```
In [ ]:
```

In []:

In [9]: `import pandas as pd`

```
# Load your dataset
df = pd.read_csv(r"C:\Users\Surendranath TV\Desktop\customer satisfaction\data\customers.csv")
```

```
# Explore the dataset
```

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

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

```
RangeIndex: 8469 entries, 0 to 8468
```

```
Data columns (total 17 columns):
```

#	Column	Non-Null Count	Dtype
0	Ticket ID	8469 non-null	int64
1	Customer Name	8469 non-null	object
2	Customer Email	8469 non-null	object
3	Customer Age	8469 non-null	int64
4	Customer Gender	8469 non-null	object
5	Product Purchased	8469 non-null	object
6	Date of Purchase	8469 non-null	object
7	Ticket Type	8469 non-null	object
8	Ticket Subject	8469 non-null	object
9	Ticket Description	8469 non-null	object
10	Ticket Status	8469 non-null	object
11	Resolution	2769 non-null	object
12	Ticket Priority	8469 non-null	object
13	Ticket Channel	8469 non-null	object
14	First Response Time	5650 non-null	object
15	Time to Resolution	2769 non-null	object
16	Customer Satisfaction Rating	2769 non-null	float64

```
dtypes: float64(1), int64(2), object(14)
```

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

```
None
```

	Ticket ID	Customer Age	Customer Satisfaction Rating
count	8469.000000	8469.000000	2769.000000
mean	4235.000000	44.026804	2.991333
std	2444.934048	15.296112	1.407016
min	1.000000	18.000000	1.000000
25%	2118.000000	31.000000	2.000000
50%	4235.000000	44.000000	3.000000
75%	6352.000000	57.000000	4.000000
max	8469.000000	70.000000	5.000000

In [15]: `print(df.columns)`

```
Index(['Ticket ID', 'Customer Name', 'Customer Email', 'Customer Age',
       'Customer Gender', 'Product Purchased', 'Date of Purchase',
       'Ticket Type', 'Ticket Subject', 'Ticket Description', 'Ticket Status',
       'Resolution', 'Ticket Priority', 'Ticket Channel',
       'First Response Time', 'Time to Resolution',
       'Customer Satisfaction Rating'],
      dtype='object')
```

In [27]: `common_issues = df['Ticket Subject'].value_counts().head(10)`
`print("Top 10 Common Issues:")`
`print(common_issues)`
`# Plotting ticket trends over time`

```

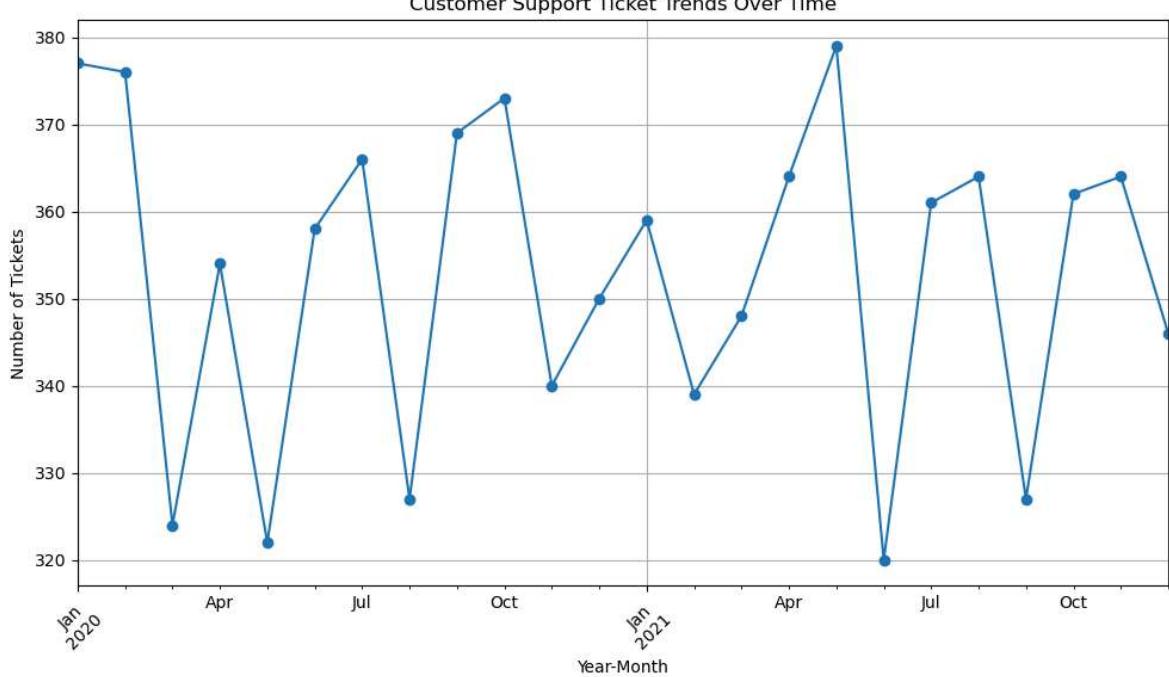
df['Date of Purchase'] = pd.to_datetime(df['Date of Purchase'])
df['YearMonth'] = df['Date of Purchase'].dt.to_period('M')
ticket_trends = df.groupby('YearMonth').size()
plt.figure(figsize=(10, 6))
ticket_trends.plot(kind='line', marker='o')
plt.title('Customer Support Ticket Trends Over Time')
plt.xlabel('Year-Month')
plt.ylabel('Number of Tickets')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

Top 10 Common Issues:

Ticket Subject	
Refund request	576
Software bug	574
Product compatibility	567
Delivery problem	561
Hardware issue	547
Battery life	542
Network problem	539
Installation support	530
Product setup	529
Payment issue	526

Name: count, dtype: int64



```

In [31]: # Segment customers
# Segment based on ticket types
ticket_type_segmentation = df.groupby('Ticket Type').size()
print("\nSegmentation based on Ticket Types:")
print(ticket_type_segmentation)
# Segment based on satisfaction levels
satisfaction_segmentation = df.groupby('Customer Satisfaction Rating').size()
print("\nSegmentation based on Customer Satisfaction Levels:")
print(satisfaction_segmentation)

```

Segmentation based on Ticket Types:

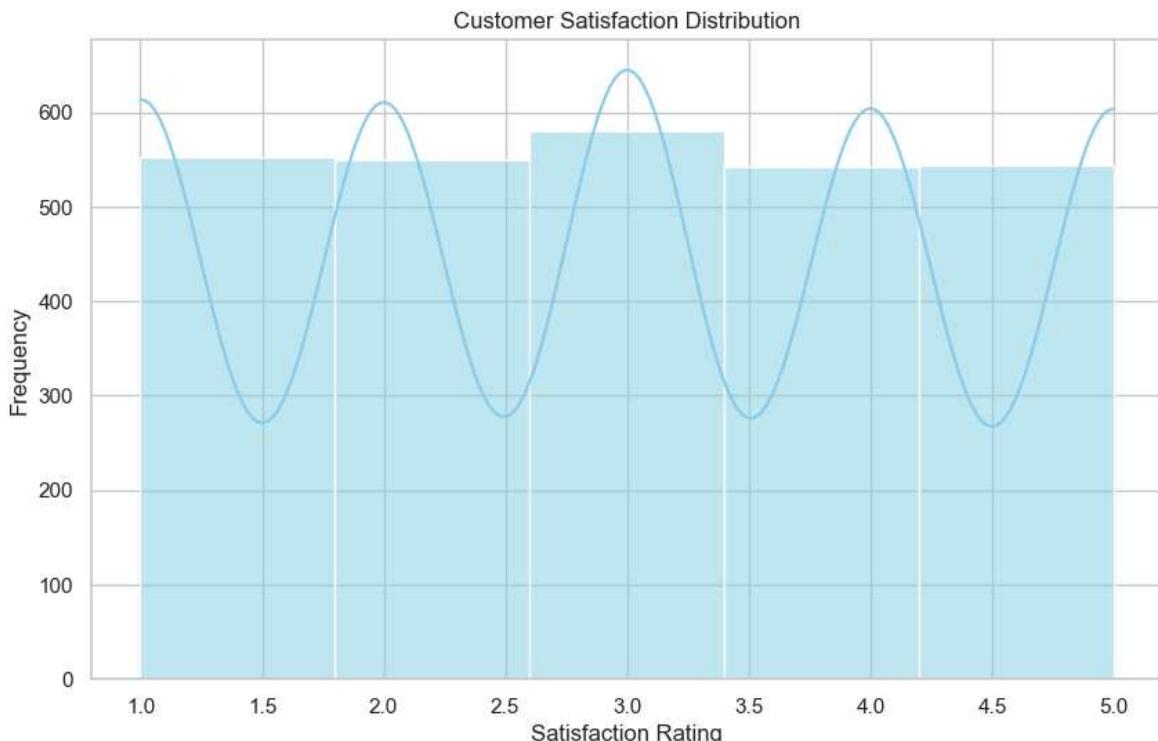
```
Ticket Type
Billing inquiry      1634
Cancellation request 1695
Product inquiry      1641
Refund request       1752
Technical issue     1747
dtype: int64
```

Segmentation based on Customer Satisfaction Levels:

Customer Satisfaction Rating

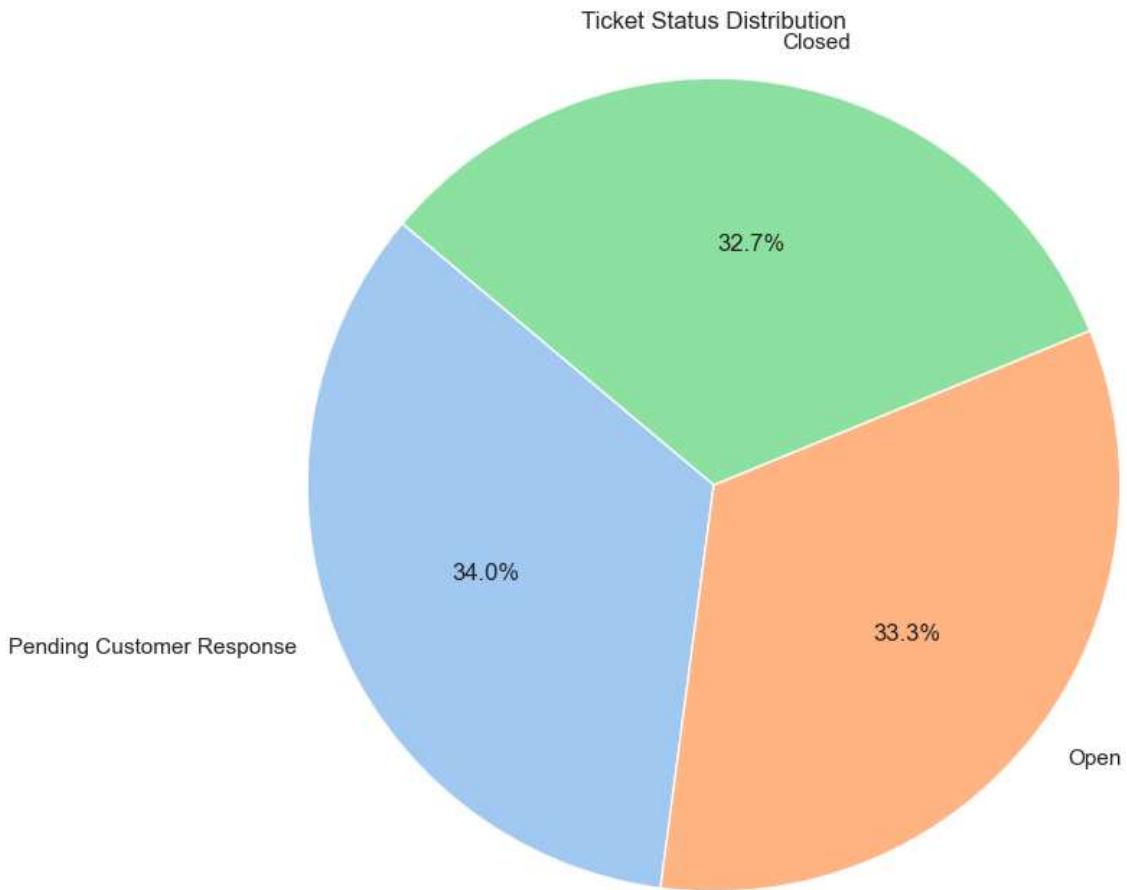
```
1.0    553
2.0    549
3.0    580
4.0    543
5.0    544
dtype: int64
```

```
In [37]: # Set up the plotting aesthetics
sns.set(style="whitegrid")
#Customer Satisfaction Distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['Customer Satisfaction Rating'], bins=5,
kde=True, color='skyblue')
plt.title('Customer Satisfaction Distribution')
plt.xlabel('Satisfaction Rating')
plt.ylabel('Frequency')
plt.show()
```

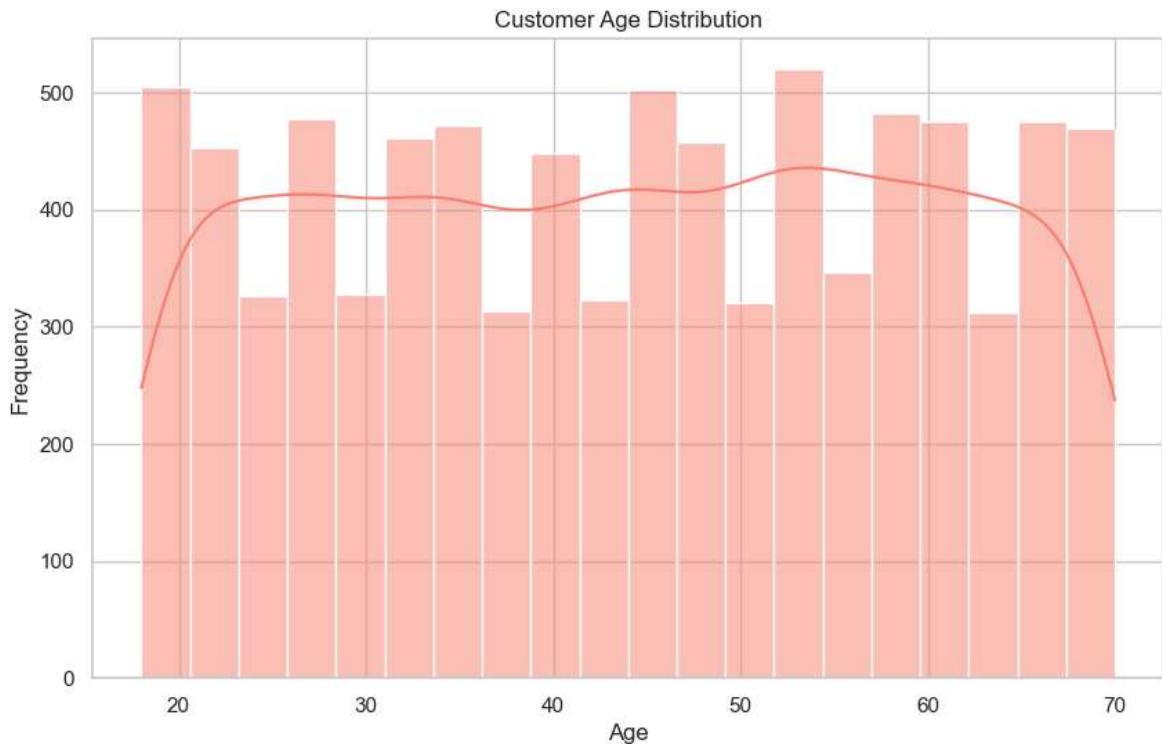


```
In [41]: #Ticket Status Distribution
ticket_status_distribution = df['Ticket Status'].value_counts()
plt.figure(figsize=(8, 8))
plt.pie(ticket_status_distribution,
labels=ticket_status_distribution.index, autopct='%1.1f%%',
colors=sns.color_palette('pastel'), startangle=140)
plt.title('Ticket Status Distribution')
```

```
plt.axis('equal')
plt.show()
```

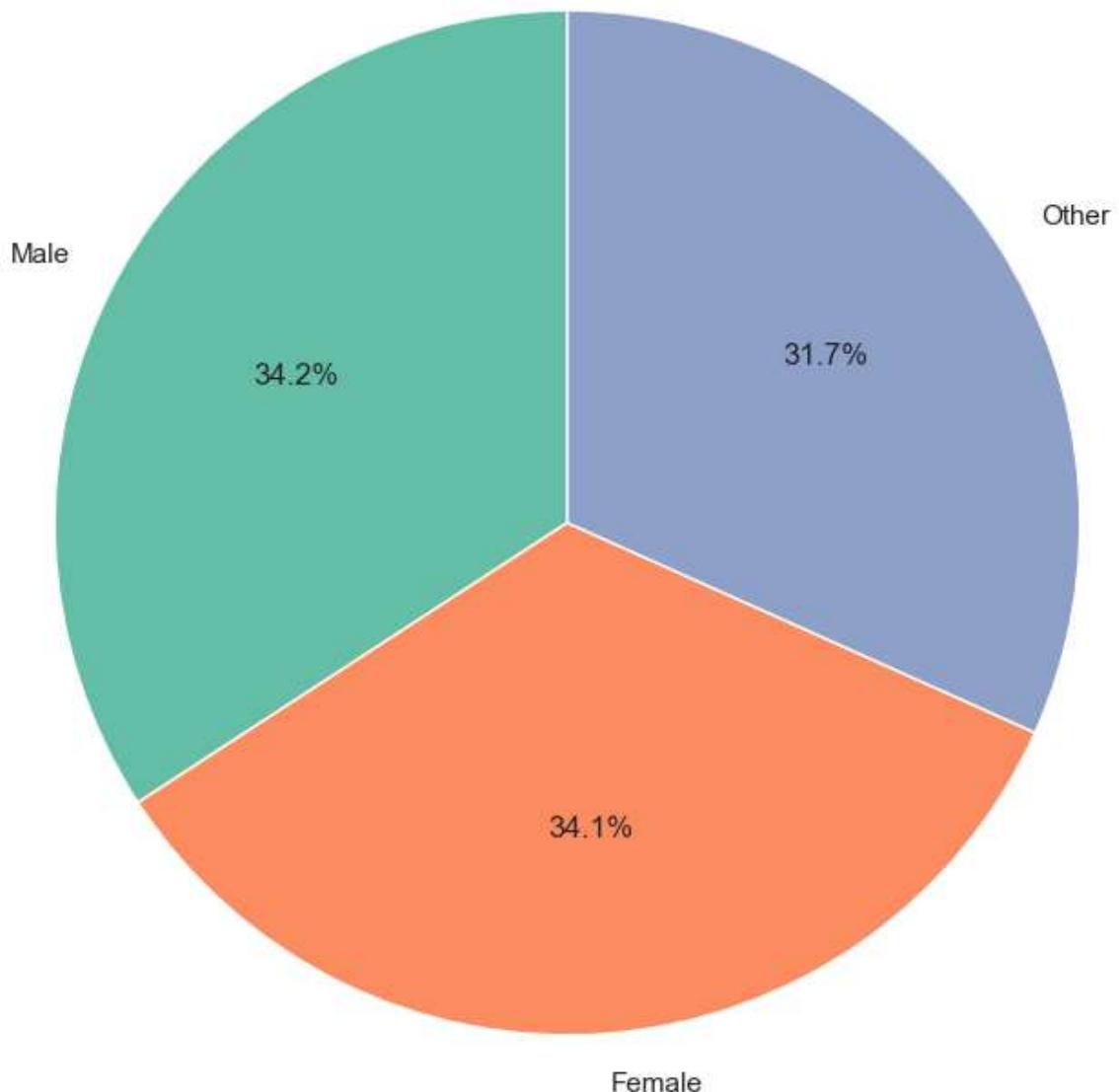


```
In [45]: #Customer Age Distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['Customer Age'], bins=20, kde=True,
color='salmon')
plt.title('Customer Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



```
In [47]: #Customer Gender Distribution
customer_gender_distribution=df['Customer Gender'].value_counts()
plt.figure(figsize=(8,8))
plt.pie(customer_gender_distribution,
labels=customer_gender_distribution.index, autopct='%1.1f%%',
colors=sns.color_palette('Set2'), startangle=90)
plt.title('CustomerGenderDistribution')
plt.axis('equal')
plt.show()
```

CustomerGenderDistribution



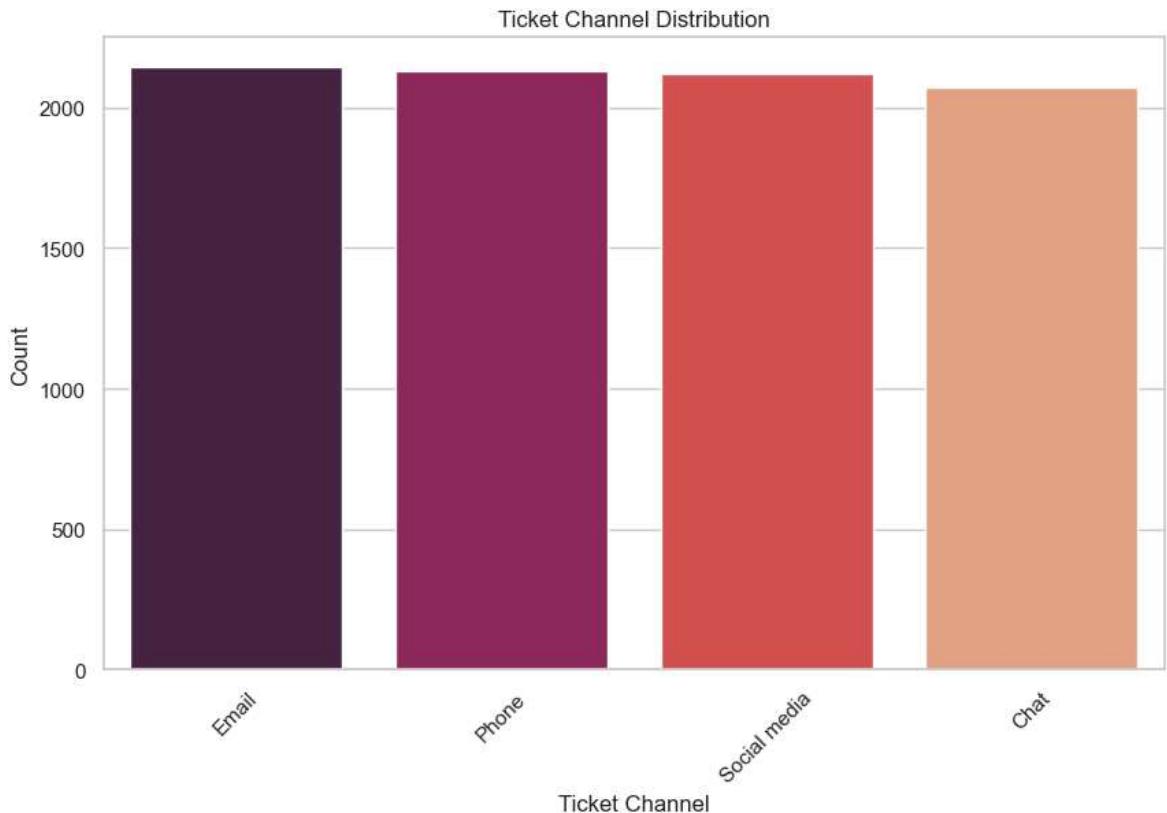
```
In [51]: #Ticket Channel Distribution
```

```
plt.figure(figsize=(10, 6))
ticket_channel_distribution = df['Ticket Channel'].value_counts()
sns.barplot(x=ticket_channel_distribution.index,
y=ticket_channel_distribution, palette='rocket')
plt.title('Ticket Channel Distribution')
plt.xlabel('Ticket Channel')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\Surendranath TV\AppData\Local\Temp\ipykernel_10540\4266043028.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.

```
    sns.barplot(x=ticket_channel_distribution.index,
```



In []:

In []:

```
In [13]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load data
df = pd.read_csv(r"C:\Users\Surendranath TV\Desktop\customer satisfaction\data\customer_satisfaction.csv")

# Check exact column names
print(df.columns.tolist())

# Group by gender and get average satisfaction
average_satisfaction = df.groupby('Customer Gender')[['Customer Satisfaction Rating']].mean()

# Plot
plt.figure(figsize=(8,6))
sns.barplot(
    x='Customer Gender',
    y='Customer Satisfaction Rating',
    data=average_satisfaction,
    palette='muted',
    order=['Male', 'Female', 'Other']
)

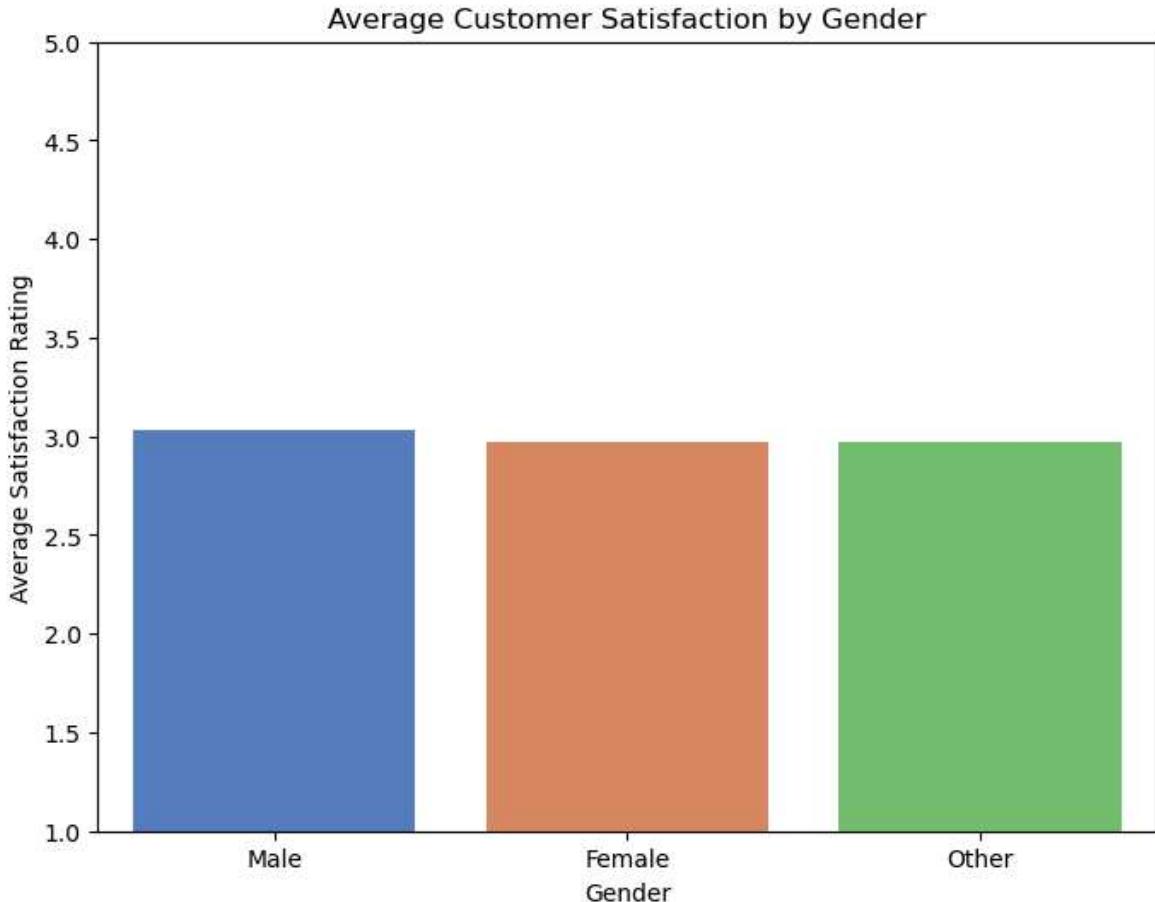
plt.title('Average Customer Satisfaction by Gender')
plt.xlabel('Gender')
plt.ylabel('Average Satisfaction Rating')
plt.ylim(1,5) # if ratings are from 1 to 5
plt.show()
```

```
[ 'Ticket ID', 'Customer Name', 'Customer Email', 'Customer Age', 'Customer Gender', 'Product Purchased', 'Date of Purchase', 'Ticket Type', 'Ticket Subject', 'Ticket Description', 'Ticket Status', 'Resolution', 'Ticket Priority', 'Ticket Channel', 'First Response Time', 'Time to Resolution', 'Customer Satisfaction Rating']
```

C:\Users\Surendranath TV\AppData\Local\Temp\ipykernel_3596\2248928348.py:16: 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(
```



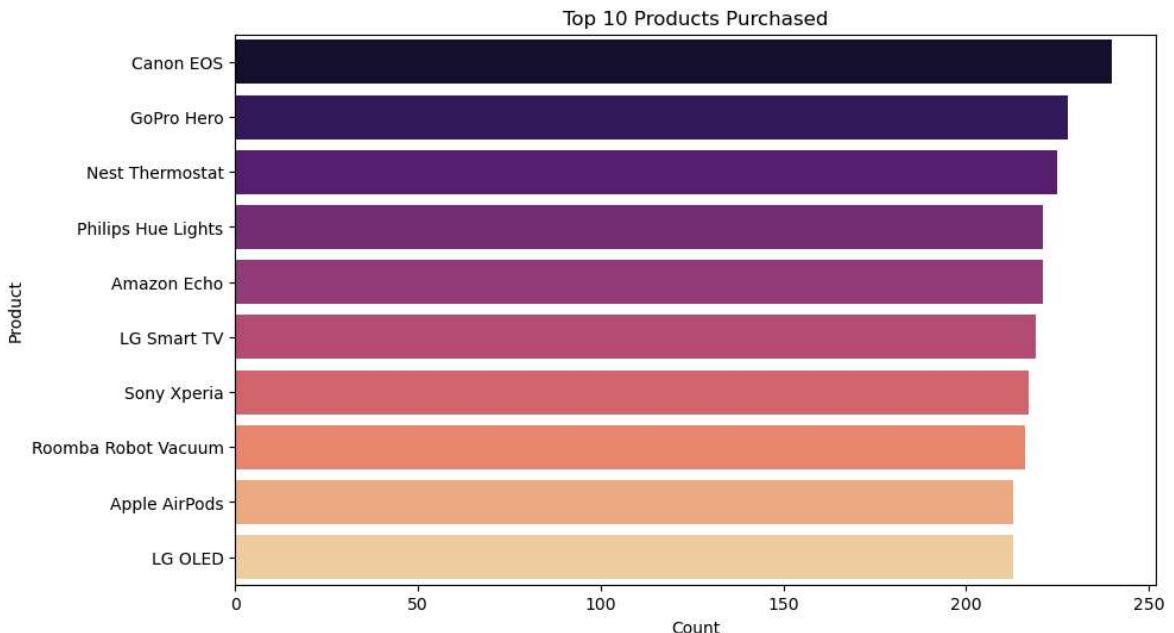
In [17]:

```
#Product Purchased Distribution
plt.figure(figsize=(10, 6))
product_purchased_distribution = df['Product Purchased'].value_counts().head(10)
sns.barplot(y=product_purchased_distribution.index,x=product_purchased_distribution)
plt.title('Top 10 Products Purchased')
plt.xlabel('Count')
plt.ylabel('Product')
plt.show()
```

C:\Users\Surendranath TV\AppData\Local\Temp\ipykernel_3596\131736119.py:4: FutureWarning:

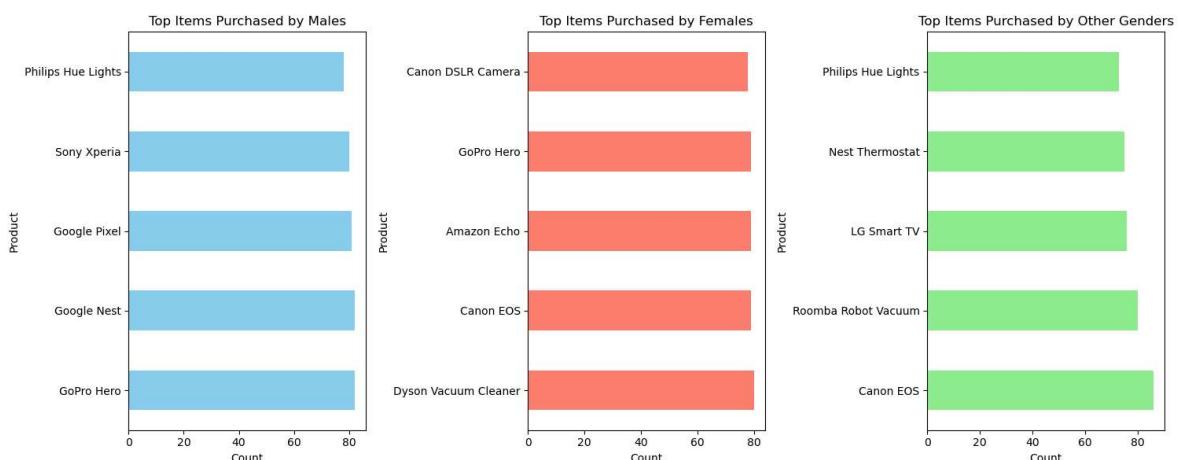
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(y=product_purchased_distribution.index,x=product_purchased_distribution, palette='magma')
```

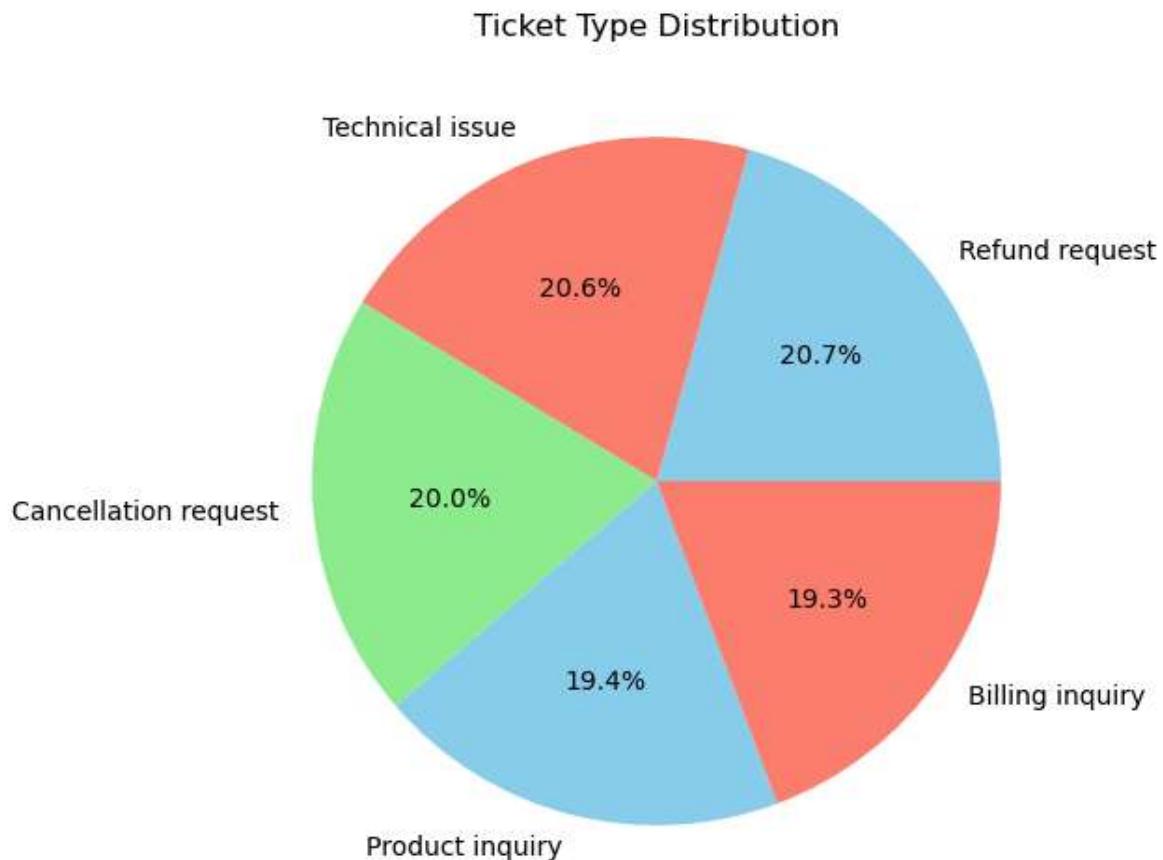


In [23]:

```
# Chart 2: Top Items Purchased by Gender (Horizontal Bar Chart)
plt.figure(figsize=(15, 6))
# Top Items Purchased by Males
plt.subplot(1, 3, 1)
top_items_male = df[df['Customer Gender'] == 'Male']['Product Purchased'].value_
top_items_male.plot(kind='barh', color='skyblue')
plt.title('Top Items Purchased by Males')
plt.xlabel('Count')
plt.ylabel('Product')
# Top Items Purchased by Females
plt.subplot(1, 3, 2)
top_items_female = df[df['Customer Gender'] == 'Female']['Product Purchased'].va_
top_items_female.plot(kind='barh', color='salmon')
plt.title('Top Items Purchased by Females')
plt.xlabel('Count')
plt.ylabel('Product')
# Top Items Purchased by Other Gender
plt.subplot(1, 3, 3)
top_items_other = df[df['Customer Gender'] == 'Other']['Product Purchased'].valu_
top_items_other.plot(kind='barh', color='lightgreen')
plt.title('Top Items Purchased by Other Genders')
plt.xlabel('Count')
plt.ylabel('Product')
plt.tight_layout()
plt.show()
```

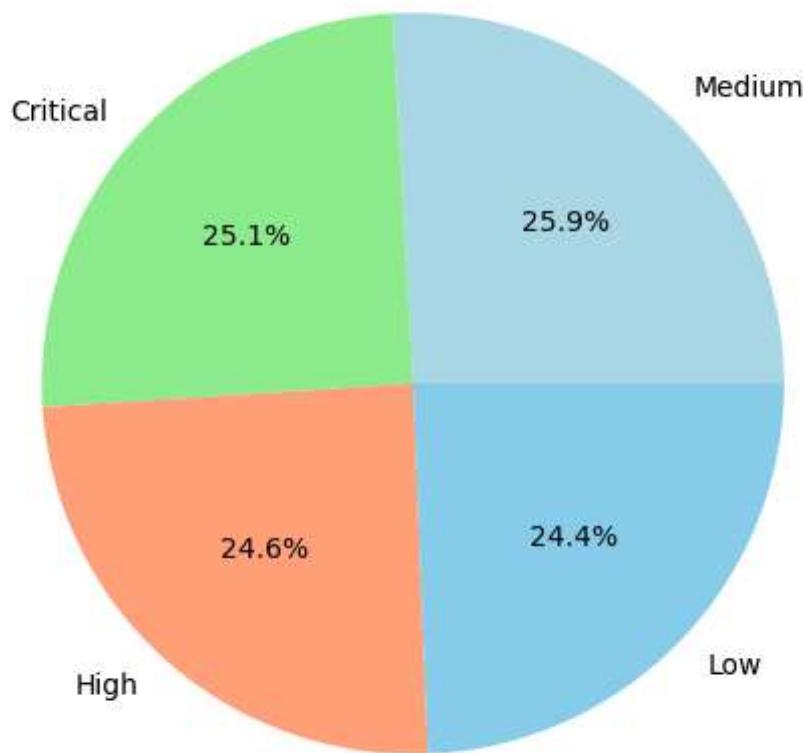


```
In [31]: # Count ticket types
ticket_type_distribution = df['Ticket Type'].value_counts()
# Plot
plt.figure(figsize=(8, 6))
ticket_type_distribution.plot(kind='pie', autopct='%1.1f%%', colors=['skyblue', 'lightgreen', 'lightsalmon', 'lightblue'])
plt.title('Ticket Type Distribution')
plt.ylabel('')
plt.show()
```



```
In [33]: # Count ticket priorities
priority_distribution = df['Ticket Priority'].value_counts()
# Plot
plt.figure(figsize=(8, 6))
priority_distribution.plot(kind='pie', autopct='%1.1f%%',
                           colors=['lightblue', 'lightgreen', 'lightsalmon', 'skyblue'])
plt.title('Priority Level Distribution')
plt.ylabel('')
plt.show()
```

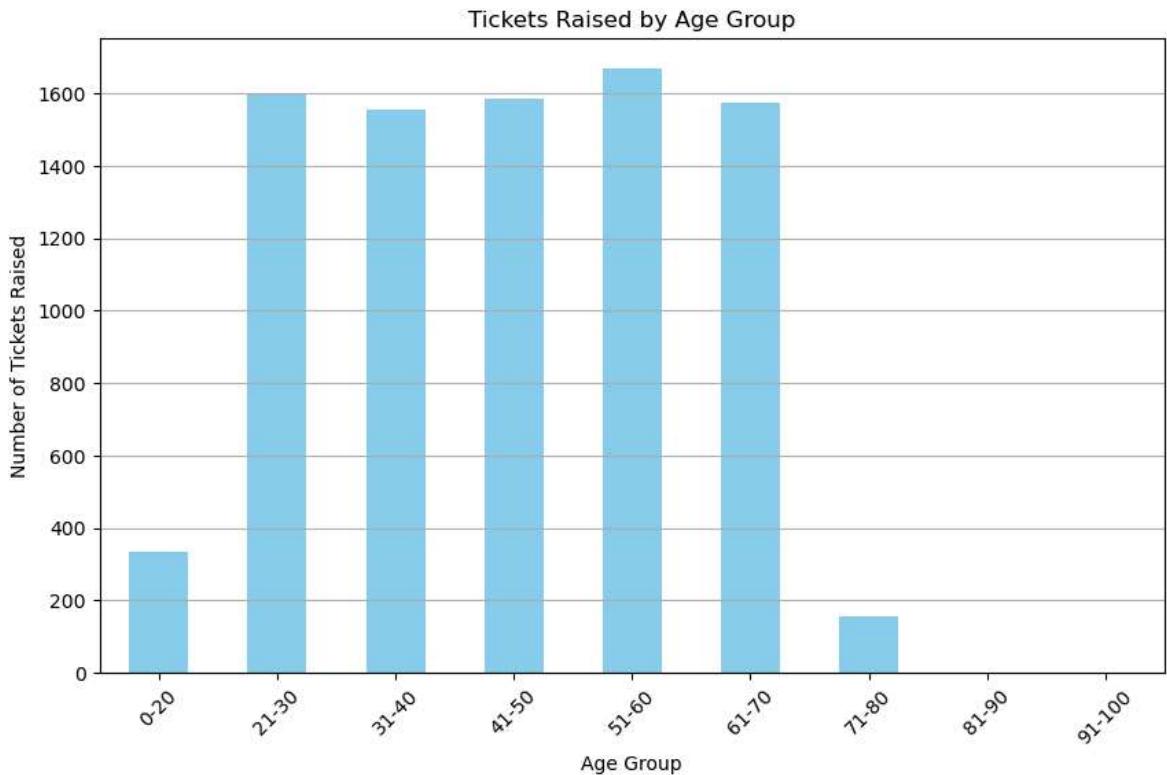
Priority Level Distribution



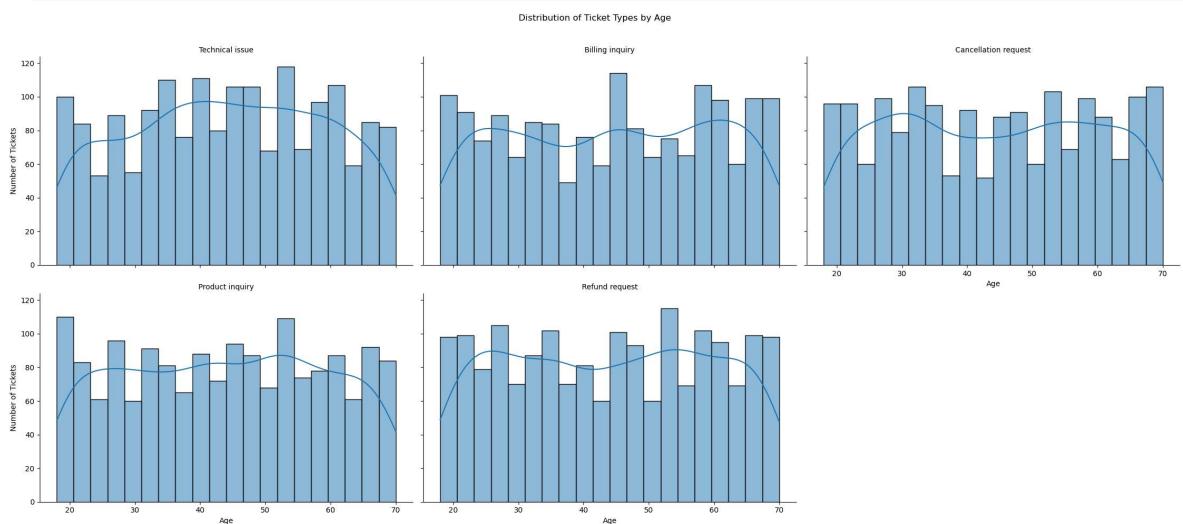
```
In [39]: # Define age groups
bins = [0, 20, 30, 40, 50, 60, 70, 80, 90, 100]
labels = ['0-20', '21-30', '31-40', '41-50', '51-60', '61-70', '71-80', '81-90',
# Categorize customers into age groups
df['Age Group'] = pd.cut(df['Customer Age'], bins=bins, labels=labels, right=False)
# Calculate number of tickets raised by each age group
tickets_by_age_group = df.groupby('Age Group').size()
# Plot
plt.figure(figsize=(10, 6))
tickets_by_age_group.plot(kind='bar', color='skyblue')
plt.title('Tickets Raised by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Number of Tickets Raised')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

C:\Users\Surendranath TV\AppData\Local\Temp\ipykernel_3596\1306694313.py:7: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
tickets_by_age_group = df.groupby('Age Group').size()
```



```
In [45]: import numpy as np
# Replace inf values with NaN
df.replace([np.inf,-np.inf], np.nan, inplace=True)
# Create a facet grid for each ticket type
g = sns.FacetGrid(df, col='Ticket Type', col_wrap=3, height=5, aspect=1.5)
g.map(sns.histplot, 'Customer Age', bins=20, kde=True)
# Set titles and labels
g.set_titles('{col_name}')
g.set_axis_labels('Age', 'Number of Tickets')
# Adjust layout
plt.subplots_adjust(top=0.9)
g.fig.suptitle('Distribution of Ticket Types by Age')
# Show plot
plt.show()
```



In []:

In []:

In []:	
In []:	