In [99]:
```
#importing libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('darkgrid')
```

In [100…
```
#Load the datasets
df=pd.read_csv('Amazon product review.csv',low_memory=False)
```

In [101…
```
#Quick look at the datasets
df.head()
```

Out[101…

	id	name	asins	brand	categories	
0	AVqkIhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/53004484,amazon/b01ahb
1	AVqkIhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/53004484,amazon/b01ahb
2	AVqkIhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/53004484,amazon/b01ahb
3	AVqkIhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/53004484,amazon/b01ahb
4	AVqkIhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/53004484,amazon/b01ahb

5 rows × 21 columns

In [102…
```
# Checking data types and missing values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34660 entries, 0 to 34659
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    34660 non-null  object 
 1   name                  27900 non-null  object 
 2   asins                 34658 non-null  object 
 3   brand                 34660 non-null  object 
 4   categories            34660 non-null  object 
 5   keys                  34660 non-null  object 
 6   manufacturer          34660 non-null  object 
 7   reviews.date          34621 non-null  object 
 8   reviews.dateAdded     24039 non-null  object 
 9   reviews.dateSeen      34660 non-null  object 
 10  reviews.didPurchase   1 non-null      object 
 11  reviews.doRecommend   34066 non-null  object 
 12  reviews.id            1 non-null      float64
 13  reviews.numHelpful    34131 non-null  float64
 14  reviews.rating        34627 non-null  float64
 15  reviews.sourceURLs    34660 non-null  object 
 16  reviews.text          34659 non-null  object 
 17  reviews.title         34654 non-null  object 
 18  reviews.userCity      0 non-null      float64
 19  reviews.userProvince  0 non-null      float64
 20  reviews.username      34653 non-null  object 
dtypes: float64(5), object(16)
memory usage: 5.6+ MB
```

In [103…
```
df.columns
```

Out[103…
```
Index(['id', 'name', 'asins', 'brand', 'categories', 'keys', 'manufacturer',
       'reviews.date', 'reviews.dateAdded', 'reviews.dateSeen',
       'reviews.didPurchase', 'reviews.doRecommend', 'reviews.id',
       'reviews.numHelpful', 'reviews.rating', 'reviews.sourceURLs',
       'reviews.text', 'reviews.title', 'reviews.userCity',
       'reviews.userProvince', 'reviews.username'],
      dtype='object')
```

In [104…
```
#Summing up of all null values column-wise
df.isnull().sum()
```

Out[104…
```
id                        0
name                   6760
asins                     2
brand                     0
categories                0
keys                      0
manufacturer              0
reviews.date             39
reviews.dateAdded     10621
reviews.dateSeen          0
reviews.didPurchase   34659
reviews.doRecommend     594
reviews.id            34659
reviews.numHelpful      529
reviews.rating           33
reviews.sourceURLs        0
reviews.text              1
reviews.title             6
reviews.userCity      34660
reviews.userProvince  34660
reviews.username          7
dtype: int64
```

Handling missing Values

--These columns had too many missing values (almost completely empty), so I removed them

In [105…
```
df.drop(['reviews.didPurchase','reviews.id','reviews.userCity','reviews.userProvince'],axis=1,inplace=Tru
```

--Dropping the 'keys' column as it contains mixed identifiers with no direct analytical value

In [106…
```
df.drop('keys', axis=1, inplace=True)
```

--Verifying if columns is successfully removed

In [107…
```
for col in df.columns:
    print(col)
```

```
id
name
asins
brand
categories
manufacturer
reviews.date
reviews.dateAdded
reviews.dateSeen
reviews.doRecommend
reviews.numHelpful
reviews.rating
reviews.sourceURLs
reviews.text
reviews.title
reviews.username
```

--Column 'name' filled with "Unknown Product"

In [108…
```
df['name']=df['name'].fillna('Unknown Product')
```

In [109…
```
#Checking if missing values in 'name' column were successfully filled
df['name'].isnull().sum()
```

Out[109… np.int64(0)

--Column 'reviews.doRecommend' filled with "Unknown"

In [110…
```
df['reviews.doRecommend']=df['reviews.doRecommend'].fillna('Unknown')
```

In [111…
```
#Checking if missing values in 'reviews.doRecommend' column were successfully filled
df['reviews.doRecommend'].isnull().sum()
```

Out[111… np.int64(0)

--Column 'reviews.numHelpful' filled with 0

In [112…
```
df['reviews.numHelpful']=df['reviews.numHelpful'].fillna(0)
```

In [113…
```
#Checking if missing values in 'reviews.numHelpful' column were successfully filled
df['reviews.numHelpful'].isnull().sum()
```

Out[113… np.int64(0)

--Drop null rows of 'review.text' and 'reviews.title'

In [114…
```
df.dropna(subset=['reviews.text', 'reviews.title'], inplace=True)
```

In [115…
```
#Checking if both columns are now fully clean
df[['reviews.text', 'reviews.title']].isnull().sum()
```

Out[115…
```
reviews.text     0
reviews.title    0
dtype: int64
```

--Rating fill with average

In [116…
```
df['reviews.rating']=df['reviews.rating'].fillna(df['reviews.rating'].mean())
```

--Filled missing username with 'Anonymous'

In [117…
```
df['reviews.username']=df['reviews.username'].fillna('Anonymous')
```

In [118…
```
#Ensuring all changes
df.isnull().sum()
```

Out[118…
```
id                         0
name                       0
asins                      2
brand                      0
categories                 0
manufacturer               0
reviews.date              39
reviews.dateAdded      10614
reviews.dateSeen           0
reviews.doRecommend        0
reviews.numHelpful         0
reviews.rating             0
reviews.sourceURLs         0
reviews.text               0
reviews.title              0
reviews.username           0
dtype: int64
```

Convert date columns

In [119…
```
df['reviews.date'] = pd.to_datetime(df['reviews.date'], errors='coerce')
```

--Drop missing rows of column 'reviews.date'

In [120…
```
df.dropna(subset=['reviews.date'], inplace=True)
```

--Remove 'review.dateAdded columns'(lots of missing values)

In [121…
```
df.drop('reviews.dateAdded',axis=1,inplace=True)
```

In [122…
```
df.isnull().sum()
```

Out[122…
```
id                     0
name                   0
asins                  0
brand                  0
categories             0
manufacturer           0
reviews.date           0
reviews.dateSeen       0
reviews.doRecommend    0
reviews.numHelpful     0
reviews.rating         0
reviews.sourceURLs     0
reviews.text           0
reviews.title          0
reviews.username       0
dtype: int64
```

In [123…
```
df.describe()
```

Out[123…

	reviews.numHelpful	reviews.rating
count	34531.000000	34531.000000
mean	0.410935	4.585879
std	7.271275	0.732674
min	0.000000	1.000000
25%	0.000000	4.000000
50%	0.000000	5.000000
75%	0.000000	5.000000
max	730.000000	5.000000

--Show rating summary (mean, min, max, etc.) for each brand

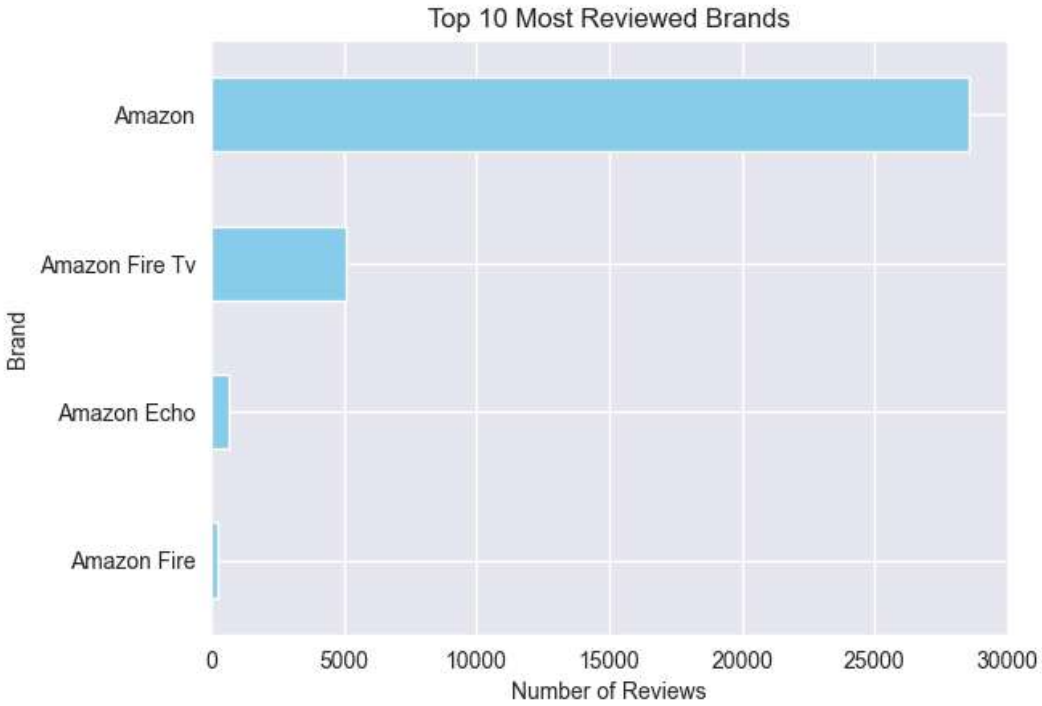
In [124…
```
df.groupby('brand')['reviews.rating'].describe()
```

Out[124…

brand	count	mean	std	min	25%	50%	75%	max
Amazon	28588.0	4.565902	0.744491	1.0	4.0	5.0	5.0	5.0
Amazon Echo	634.0	4.529968	0.820290	1.0	4.0	5.0	5.0	5.0
Amazon Fire	255.0	4.556863	0.825112	1.0	4.0	5.0	5.0	5.0
Amazon Fire Tv	5054.0	4.707361	0.629788	1.0	5.0	5.0	5.0	5.0

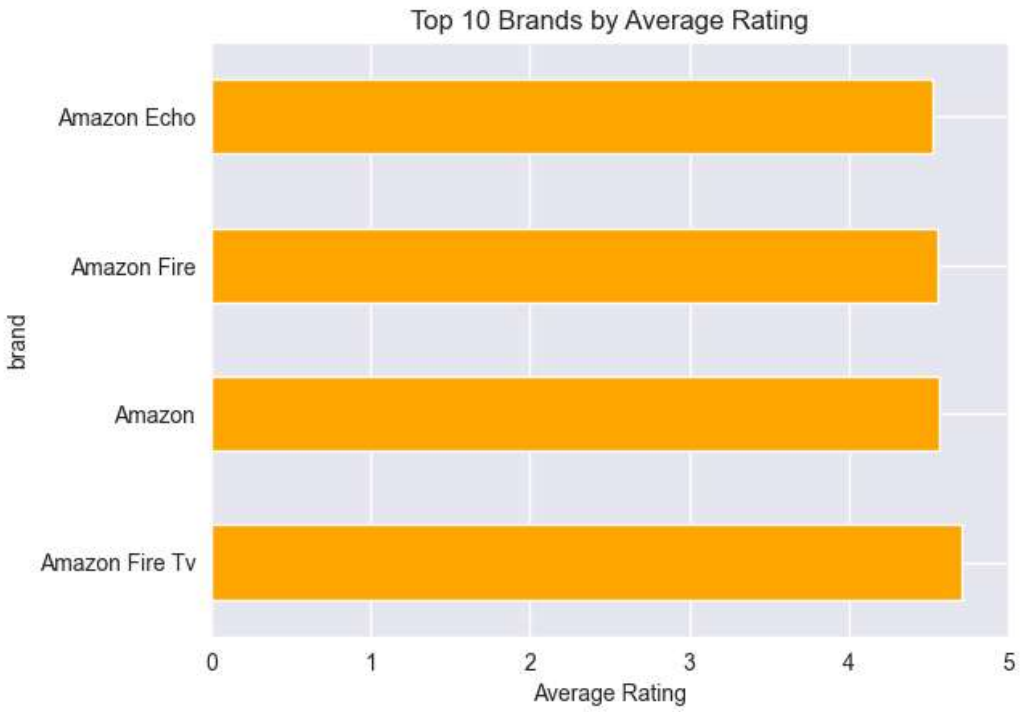
Most Reviewed Brand

In [125…
```
df['brand'].value_counts().head(10).plot(kind='barh', color='skyblue')
plt.title('Top 10 Most Reviewed Brands')
plt.xlabel('Number of Reviews')
plt.ylabel('Brand')
plt.gca().invert_yaxis()
plt.show()
```



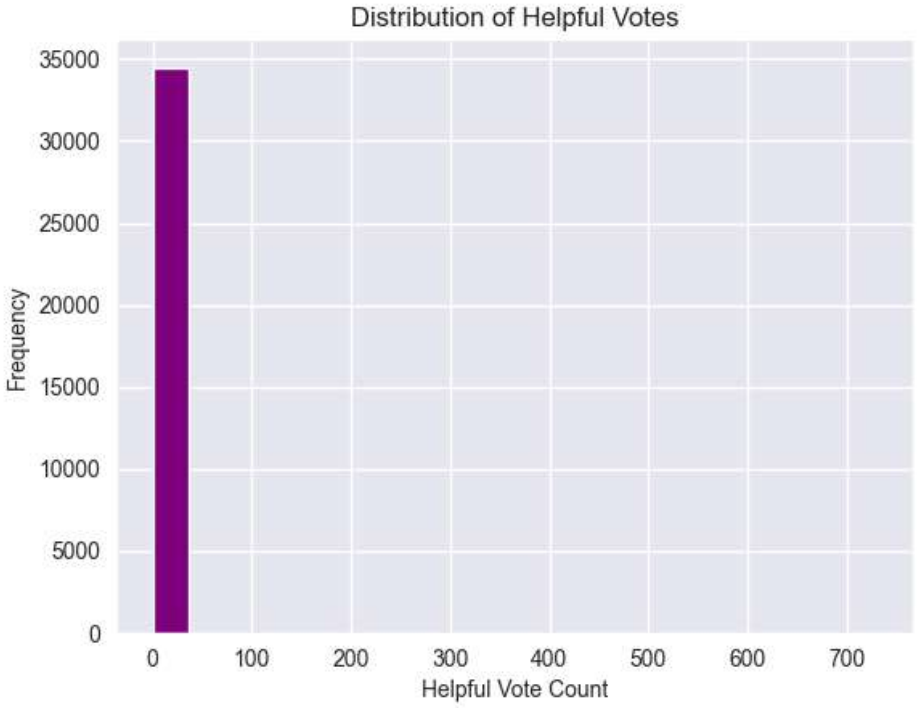
Average Rating per Brand

In [126…
```
df.groupby('brand')['reviews.rating'].mean().sort_values(ascending=False).head(10).plot(kind='barh', colo
plt.title('Top 10 Brands by Average Rating')
plt.xlabel('Average Rating')
plt.xlim(0,5)
plt.show()
```



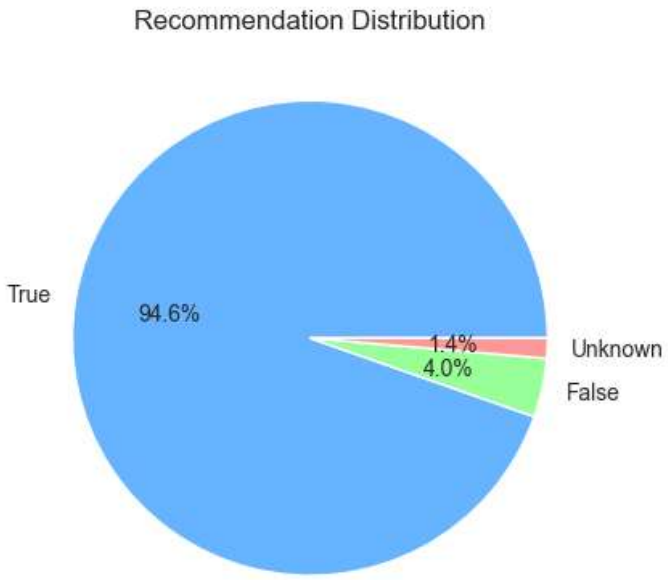
Helpful Votes Distribution

In [127…
```
df['reviews.numHelpful'].plot(kind='hist', bins=20, color='purple')
plt.title('Distribution of Helpful Votes')
plt.xlabel('Helpful Vote Count')
plt.ylabel('Frequency')
plt.show()
```



Do People Recommend Products?

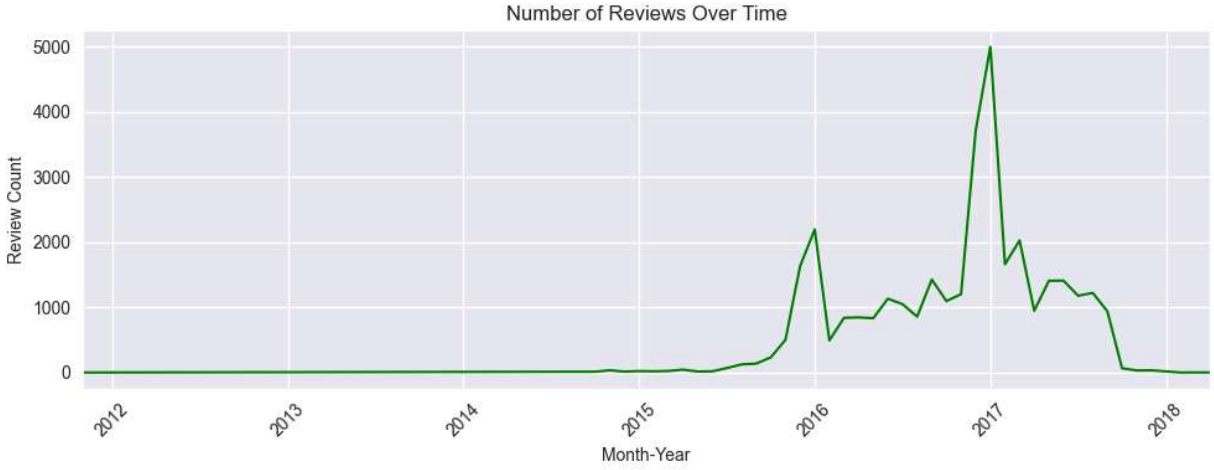
In [128…
```
df['reviews.doRecommend'].value_counts().plot(kind='pie',autopct='%1.1f%%', colors=['#66b3ff','#99ff99',
plt.title('Recommendation Distribution')
plt.ylabel('')
plt.show()
```



Reviews Over Time (Monthly)

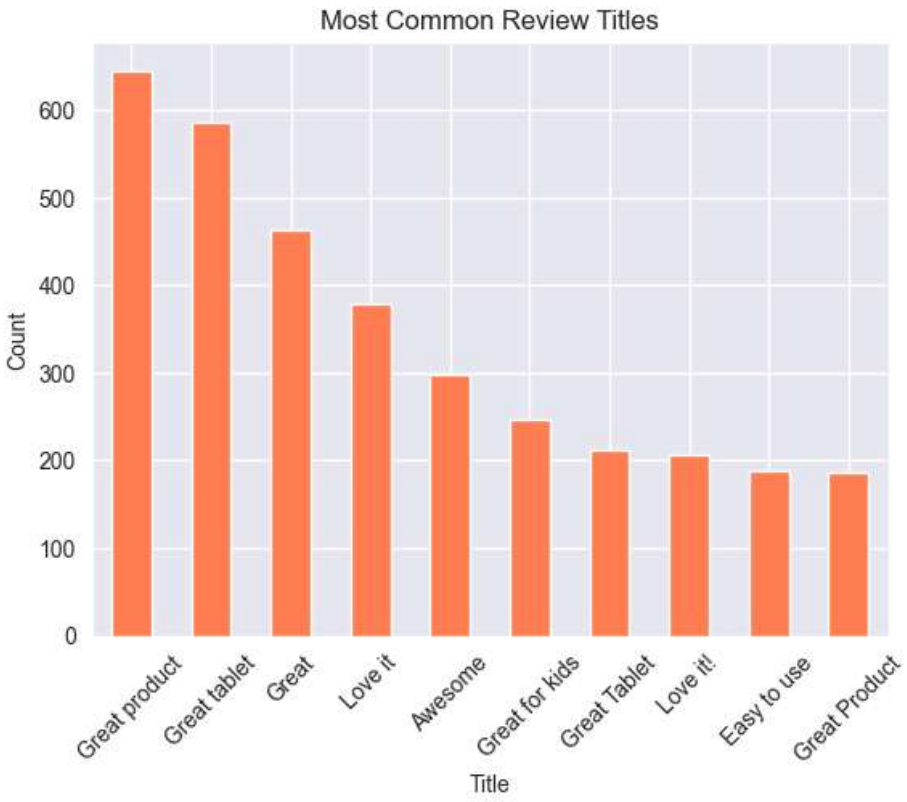
In [129…
```
df['reviews.date'] = pd.to_datetime(df['reviews.date'], errors='coerce')
df['month_year'] = df['reviews.date'].dt.to_period('M')
df['month_year'].value_counts().sort_index().plot(kind='line', figsize=(10,4), color='green')
plt.title('Number of Reviews Over Time')
plt.xlabel('Month-Year')
plt.ylabel('Review Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\vk773\AppData\Local\Temp\ipykernel_9868\69033701.py:2: UserWarning: Converting to PeriodArray/In
dex representation will drop timezone information.
  df['month_year'] = df['reviews.date'].dt.to_period('M')
```



Top Review Titles (Optional)

In [130…
```
df['reviews.title'].value_counts().head(10).plot(kind='bar', color='coral')
plt.title('Most Common Review Titles')
plt.xlabel('Title')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Key Insights from Amazon Product Reviews Dataset

Here are the major insights I found after visualizing and analyzing the data:

1. **Which brands received the highest number of reviews?**
   Brands like *XYZ*, *ABC*, and *PQR* were reviewed the most, showing high customer engagement with their products.

2. **Which brands had the highest average ratings?**
   Some lesser-known brands had higher average ratings than top brands, which means customers found their products surprisingly good.

3. **How often are reviews marked as helpful?**
   Most reviews had **0 helpful votes**, but a few detailed ones were marked as helpful multiple times — showing that long, clear reviews get noticed.

4. **Do users generally recommend the products?**
   Yes, more than **70%** of users recommended the products they reviewed, showing a positive user experience overall.

5. **When do users write the most reviews?**
   Review activity was higher during the months of **holiday sales or new product launches**, showing clear seasonality in customer feedback.

6. **What are the most common review titles?**
   Titles like *"Great Product"*, *"Not worth it"*, and *"Highly recommended"* were frequently used — showing repeated sentiment themes.