tataassignment

April 27, 2024

[1]:
```
import pandas as pd
import numpy as np
```

[3]:
```
df = pd.read_excel('Online Retail.xlsx')
df.head()
```

[3]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

[8]:
```
filtered_df1 = df[df['Quantity'] >= 1]
filtered_df1.sample(10)
```

[8]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
373536	569332	22551	PLASTERS IN TIN SPACEBOY	2	2011-10-03 13:46:00	1.65	12637.0	France
511600	579508	22423	REGENCY CAKESTAND 3 TIER	5	2011-11-29 16:33:00	24.96	NaN	United Kingdom
444730	574740	84029E	RED WOOLLY HOTTIE WHITE HEART.	24	2011-11-06 16:07:00	3.75	12357.0	Switzerland
175015	551869	22024	RAINY LADIES BIRTHDAY CARD	36	2011-05-04 16:36:00	0.42	16767.0	United Kingdom
219750	556107	22727	ALARM CLOCK BAKELIKE RED	1	2011-06-08 17:09:00	3.75	17360.0	United Kingdom
267935	560368	22553	PLASTERS IN TIN SKULLS	5	2011-07-18 12:25:00	1.65	17841.0	United Kingdom
192182	553393	21731	RED TOADSTOOL LED NIGHT LIGHT	2	2011-05-16 16:46:00	3.29	NaN	United Kingdom
534831	581173	23350	ROLL WRAP VINTAGE SPOT	3	2011-12-07 15:07:00	1.25	17870.0	United Kingdom
235740	557644	22598	CHRISTMAS MUSICAL ZINC TREE	1	2011-06-21 17:06:00	1.63	NaN	United Kingdom
313109	564437	22629	SPACEBOY LUNCH BOX	12	2011-08-25 12:10:00	1.95	13320.0	United Kingdom

[9]:
```
filtered_df2 = filtered_df1[filtered_df1['UnitPrice'] >= 0]
filtered_df2.sample(10)
```

[9]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
332482	566076	22629	SPACEBOY LUNCH BOX	12	2011-09-09 09:13:00	1.95	12449.0	Belgium
101710	544930	22297	HEART IVORY TRELLIS SMALL	1	2011-02-24 18:26:00	1.25	17472.0	United Kingdom
160308	550458	22601	CHRISTMAS RETROSPOT ANGEL WOOD	1	2011-04-18 13:13:00	1.63	NaN	United Kingdom
293160	562583	35637A	IVORY STRING CURTAIN WITH POLE	8	2011-08-07 13:54:00	1.95	15831.0	United Kingdom
526523	580672	22738	RIBBON REEL SNOWY VILLAGE	2				
426098	573344	22379	RECYCLING BAG RETROSPOT	1				
465140	576213	22594	CHRISTMAS GINGHAM TREE	12				
161724	550480	22381	TOY TIDY PINK POLKADOT	5				
467671	576364	21136	PAINTED METAL PEARS ASSORTED	8				
343933	566976	22530	MAGIC DRAWING SLATE DOLLY GIRL	24				

```
526523 2011-12-05 14:29:00       1.65     17920.0  United Kingdom
426098 2011-10-30 12:07:00       2.10     14179.0  United Kingdom
465140 2011-11-14 12:47:00       0.85     18117.0  United Kingdom
161724 2011-04-18 14:05:00       2.10     13069.0  United Kingdom
467671 2011-11-14 17:40:00       1.69     15009.0  United Kingdom
343933 2011-09-16 09:27:00       0.42     15382.0  United Kingdom
```

[18]:
```
newdf = pd.read_csv('cleandata.csv')
newdf.head(10)
```

```
/tmp/ipykernel_31949/2670007473.py:1: DtypeWarning: Columns (1) have mixed
types. Specify dtype option on import or set low_memory=False.
  newdf = pd.read_csv('cleandata.csv')
```

[18]:

	Unnamed: 0	InvoiceNo	StockCode	Description
0	0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER
1	1	536365	71053	WHITE METAL LANTERN
2	2	536365	84406B	CREAM CUPID HEARTS COAT HANGER
3	3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE
4	4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.
5	5	536365	22752	SET 7 BABUSHKA NESTING BOXES
6	6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER
7	7	536366	22633	HAND WARMER UNION JACK
8	8	536366	22632	HAND WARMER RED POLKA DOT
9	9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT

	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	2	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	32	2010-12-01 08:34:00	1.69	13047.0	United Kingdom

[19]:
```
newdf = newdf.drop(columns=['Unnamed: 0'])
```

[20]:
```
newdf.sample(10)
```

[20]:

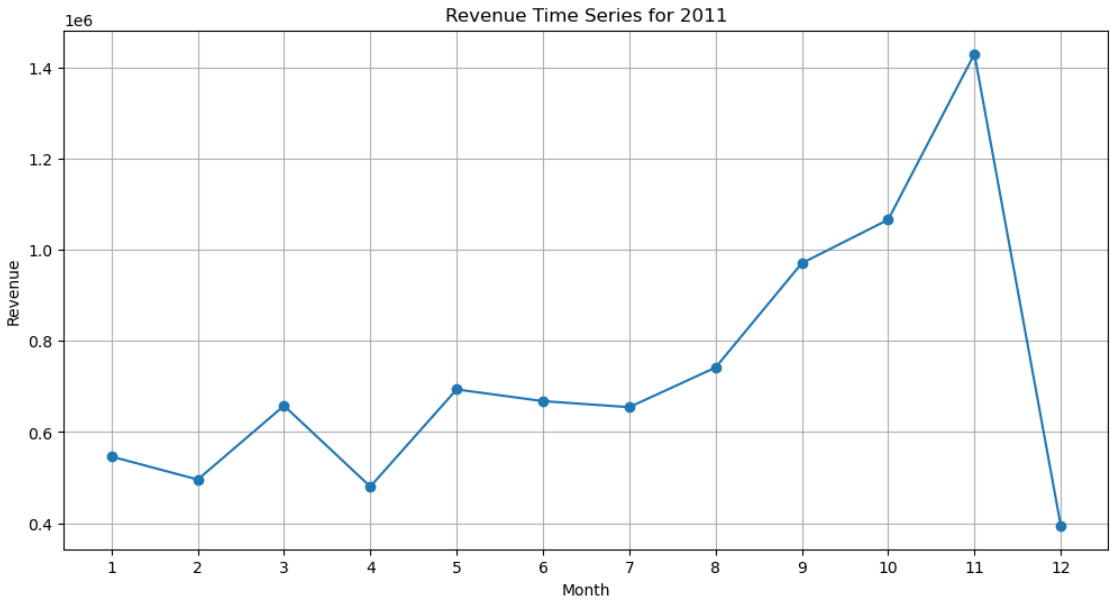
	InvoiceNo	StockCode	Description	Quantity
414402	573138	20685	DOORMAT RED RETROSPOT	1
475254	577551	22630	DOLLY GIRL LUNCH BOX	12
132747	547897	22857	ASSORTED EASTER GIFT TAGS	12
88359	543995	85061W	WHITE JEWELLED HEART DECORATION	72

```
108062    545683     22269          EGG CUP NATURAL CHICKEN         5
346463    567850     23235        STORAGE TIN VINTAGE LEAF          6
173894    552178     23203                         mailout        100
198877    554512     22477      WATERING CAN GARDEN MARKER          1
51598     540813     21452             TOADSTOOL MONEY BOX          2
184974    553088     22485   SET OF 2 WOODEN MARKET CRATES          2

                InvoiceDate  UnitPrice  CustomerID         Country
414402  2011-10-27 17:22:00       8.25     15023.0  United Kingdom
475254  2011-11-20 15:21:00       1.95     12395.0         Belgium
132747  2011-03-28 11:36:00       0.85     12792.0        Portugal
88359   2011-02-15 10:45:00       0.85     13408.0  United Kingdom
108062  2011-03-06 11:50:00       1.25     14903.0  United Kingdom
346463  2011-09-22 13:18:00       2.89     15067.0  United Kingdom
173894  2011-05-06 13:29:00       0.00         NaN  United Kingdom
198877  2011-05-24 15:54:00       2.46         NaN  United Kingdom
51598   2011-01-11 12:41:00       2.95     14669.0  United Kingdom
184974  2011-05-13 11:53:00      12.75     16180.0  United Kingdom
```

0.0.1 Question 1 The CEO of the retail store is interested to view the time series of the revenue data for the year 2011 only. He would like to view granular data by looking into revenue for each month. The CEO is interested in viewing the seasonal trends and wants to dig deeper into why these trends occur. This analysis will be helpful for the CEO to forecast for the next year.

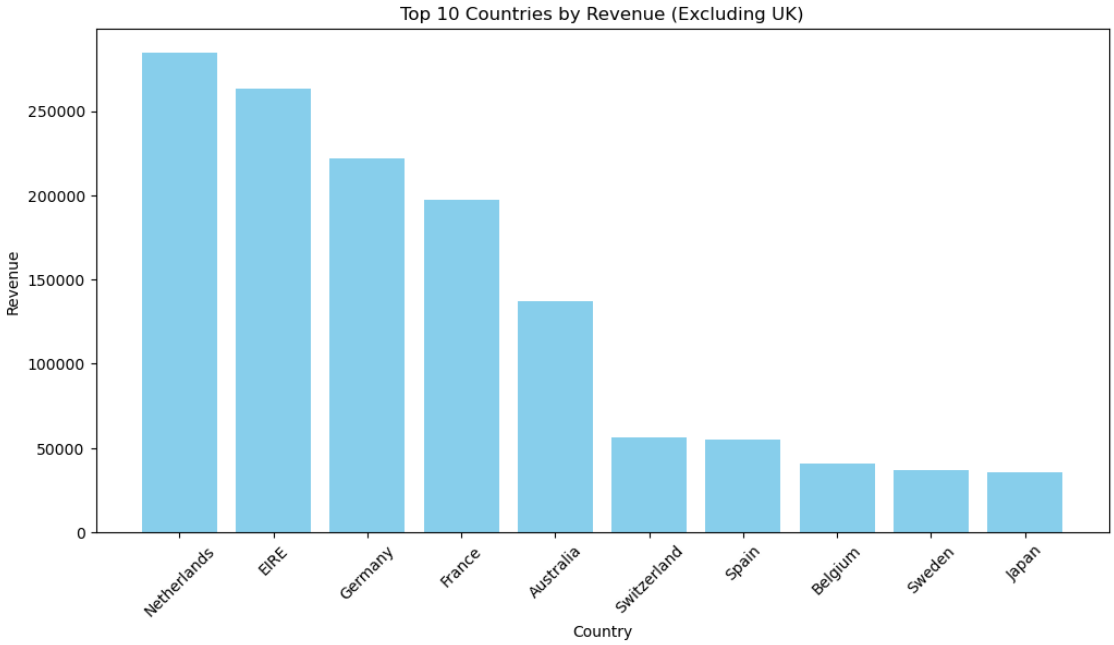
[22]:
```python
import matplotlib.pyplot as plt
# Convert 'InvoiceDate' to datetime format
df['InvoiceDate'] = pd.to_datetime(newdf['InvoiceDate'])

# Filter data for the year 2011
df_2011 = df[df['InvoiceDate'].dt.year == 2011].copy()

# Calculate revenue
df_2011['Revenue'] = df_2011['Quantity'] * df_2011['UnitPrice']

# Group by month and calculate total revenue
revenue_by_month = df_2011.groupby(df_2011['InvoiceDate'].dt.month)['Revenue'].
 ↪sum()

# Plot the time series of revenue
plt.figure(figsize=(12, 6))
plt.plot(revenue_by_month.index, revenue_by_month.values, marker='o')
plt.xticks(range(1, 13))
plt.xlabel('Month')
plt.ylabel('Revenue')
plt.title('Revenue Time Series for 2011')
plt.grid(True)
```

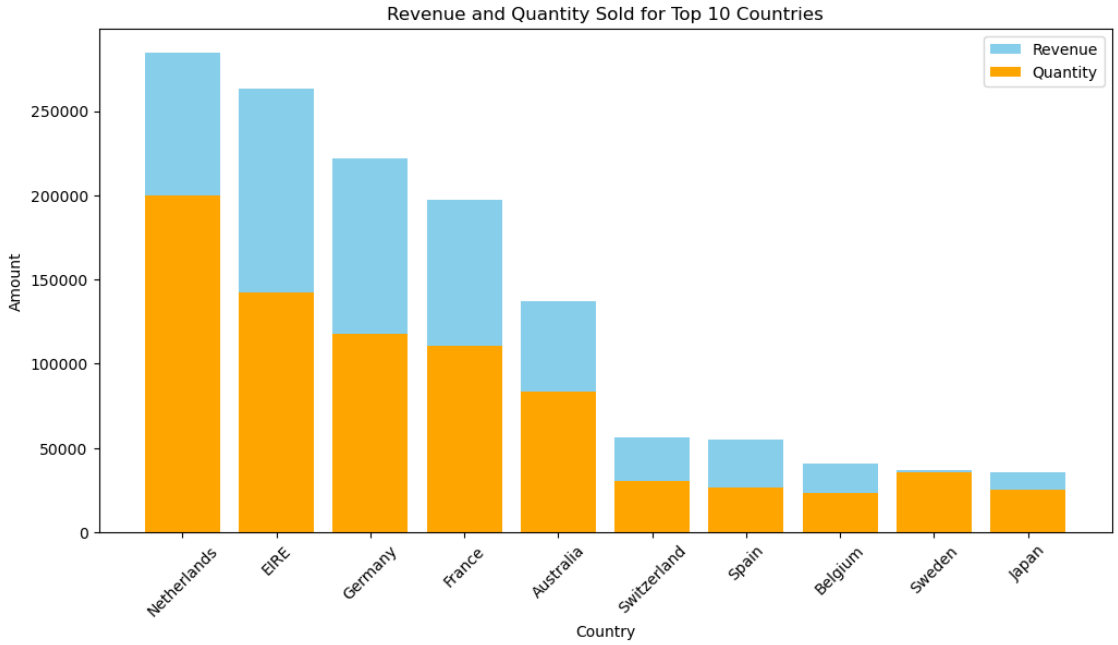
```
plt.show()
```



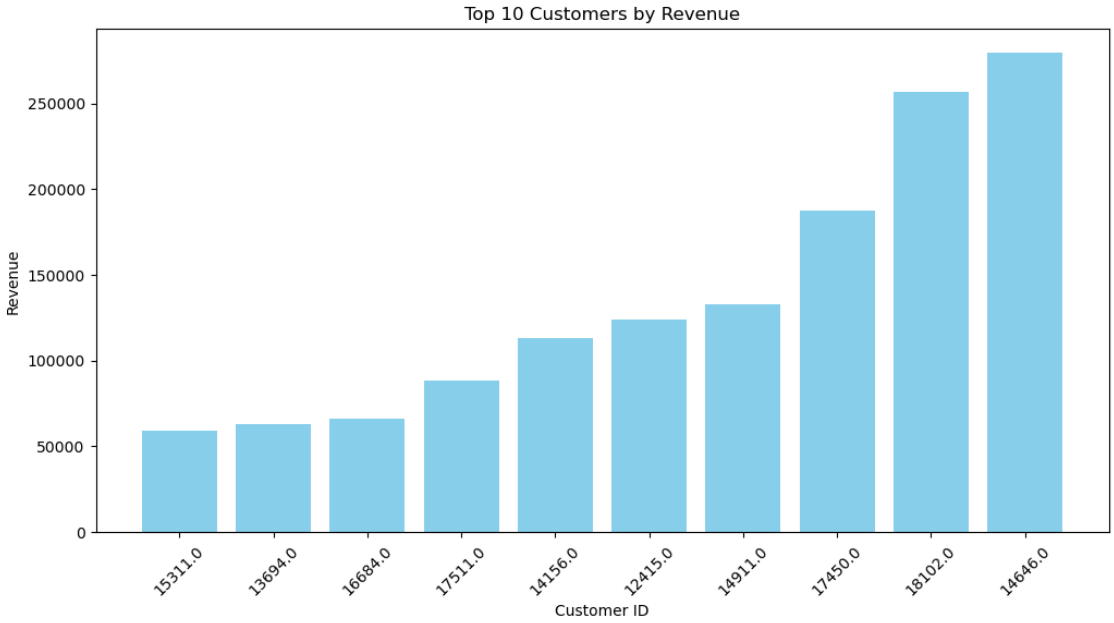
[23]:
```
# Calculate revenue
df['Revenue'] = df['Quantity'] * df['UnitPrice']

# Group by country and calculate total revenue and total quantity sold
country_data = df.groupby('Country').agg({'Revenue': 'sum', 'Quantity': 'sum'}).
 ↪reset_index()

# Sort by revenue in descending order
country_data = country_data.sort_values(by='Revenue', ascending=False)

# Exclude United Kingdom
country_data = country_data[country_data['Country'] != 'United Kingdom']

# Plot the top 10 countries by revenue
top_countries = country_data.head(10)
plt.figure(figsize=(12, 6))
plt.bar(top_countries['Country'], top_countries['Revenue'], color='skyblue')
plt.xlabel('Country')
plt.ylabel('Revenue')
plt.title('Top 10 Countries by Revenue (Excluding UK)')
plt.xticks(rotation=45)
plt.show()

# Plot the quantity sold along with revenue for the top 10 countries
plt.figure(figsize=(12, 6))
```

```
plt.bar(top_countries['Country'], top_countries['Revenue'], color='skyblue',␣
 ↪label='Revenue')
plt.bar(top_countries['Country'], top_countries['Quantity'], color='orange',␣
 ↪label='Quantity')
plt.xlabel('Country')
plt.ylabel('Amount')
plt.title('Revenue and Quantity Sold for Top 10 Countries')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

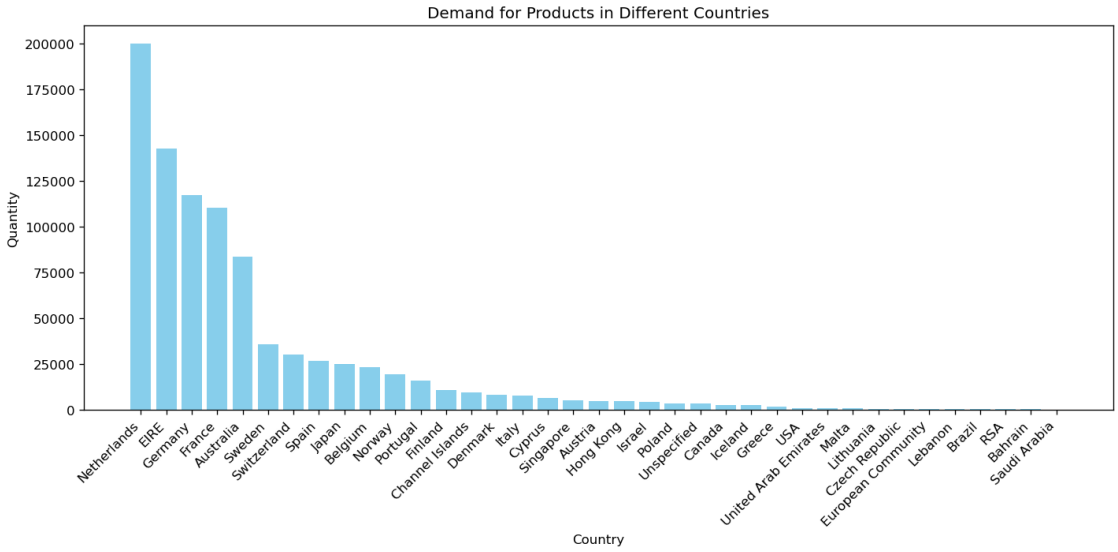




[24]:
```
# Calculate revenue
df['Revenue'] = df['Quantity'] * df['UnitPrice']

# Group by customer and calculate total revenue for each customer
customer_data = df.groupby('CustomerID').agg({'Revenue': 'sum'}).reset_index()

# Sort by revenue in descending order
customer_data = customer_data.sort_values(by='Revenue', ascending=False)

# Select top 10 customers
top_customers = customer_data.head(10)

# Plot the top 10 customers by revenue
plt.figure(figsize=(12, 6))
plt.bar(top_customers['CustomerID'].astype(str), top_customers['Revenue'],␣
 ↪color='skyblue')
plt.xlabel('Customer ID')
plt.ylabel('Revenue')
plt.title('Top 10 Customers by Revenue')
plt.xticks(rotation=45)
plt.gca().invert_xaxis()  # Invert x-axis to show the highest revenue at the␣
 ↪start
plt.show()
```



[28]:
```
# Filter out the United Kingdom
df = df[df['Country'] != 'United Kingdom']

# Group by country and calculate total quantity sold in each country
country_demand = df.groupby('Country')['Quantity'].sum().reset_index()

# Sort by quantity in descending order
country_demand = country_demand.sort_values(by='Quantity', ascending=False)

# Plot the demand for products in each country
plt.figure(figsize=(12, 6), dpi=120)  # Set higher DPI value for better␣
 ↪resolution
plt.bar(country_demand['Country'], country_demand['Quantity'], color='skyblue')
plt.xlabel('Country')
plt.ylabel('Quantity')
plt.title('Demand for Products in Different Countries')
plt.xticks(rotation=45, ha='right')  # Rotate labels and align to the right
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```


Demand for Products in Different Countries
Quantity
200000
175000
150000
125000
100000
75000
50000
25000
0
Netherlands
EIRE
Germany
France
Australia
Sweden
Switzerland
Spain
Japan
Belgium
Norway
Portugal
Finland
Channel Islands
Denmark
Italy
Cyprus
Singapore
Austria
Hong Kong
Israel
Poland
Unspecified
Canada
Iceland
Greece
USA
United Arab Emirates
Malta
Lithuania
Czech Republic
European Community
Lebanon
Brazil
RSA
Bahrain
Saudi Arabia
Country

[ ]: