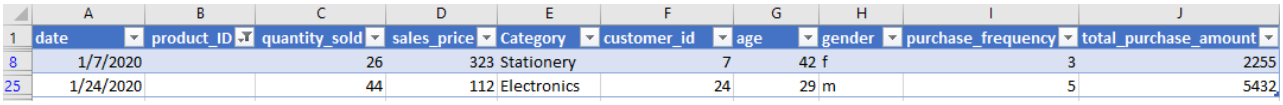
Basmalla Zakaria Sliem	**20210598**
Nada Mamdouh Mohamed	**20200593**
Mohamed Ahmed Abdelghani	**20200423**
Suhayla Ehab Gamal	**20200236**

Retail Sales Forecasting and Customer Segmentation

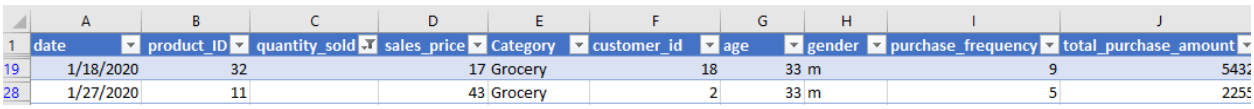
Data cleaning steps

-Handling missing values:

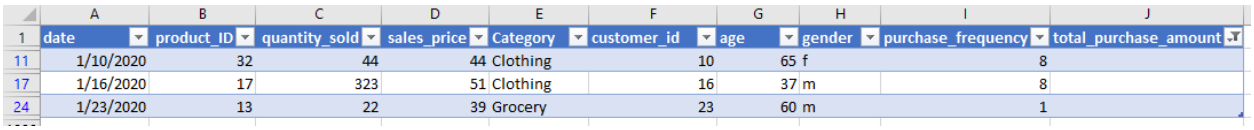
We find that there are missing values in 3 columns in the dataset (Product_ID, Quantity_sold, Total_purchase_amount)



	A	B	C	D	E	F	G	H	I	J
1	date	product_ID	quantity_sold	sales_price	Category	customer_id	age	gender	purchase_frequency	total_purchase_amount
8	1/7/2020		26	323	Stationery	7	42	f	3	2255
25	1/24/2020		44	112	Electronics	24	29	m	5	5432



	A	B	C	D	E	F	G	H	I	J
1	date	product_ID	quantity_sold	sales_price	Category	customer_id	age	gender	purchase_frequency	total_purchase_amount
19	1/18/2020	32		17	Grocery	18	33	m	9	5432
28	1/27/2020	11		43	Grocery	2	33	m	5	2255



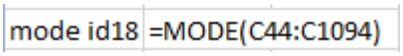
	A	B	C	D	E	F	G	H	I	J
1	date	product_ID	quantity_sold	sales_price	Category	customer_id	age	gender	purchase_frequency	total_purchase_amount
11	1/10/2020	32	44	44	Clothing	10	65	f	8	
17	1/16/2020	17	323	51	Clothing	16	37	m	8	
24	1/23/2020	13	22	39	Grocery	23	60	m	1	

-We have handled the missing values for product_ID column by calculating the average of the column by this function: 

```
=AVERAGE(Table1 [ product_ID] )
```

and replace blank cells with the average = 26 because this is the closest number to the rest of values of product ID with the same category and sales price.

- Handling missing values in quantity_sold column by finding the most repetitive value (mode) for the customer ID by this function:

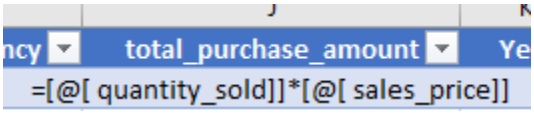


```
mode id18 =MODE(C44:C1094)
```

we find that mode for quantity sold for customer with ID (18) for grocery category = 55, it means that Iam looking for the most repetitive quantity sold from this category for this customer

And for customer with ID= 2 the most repetitive quantity sold of grocery category=26.

-The missing values in total purchase amount column filled automatically when we calculate it by multiplying sales_price by quantity_sold.



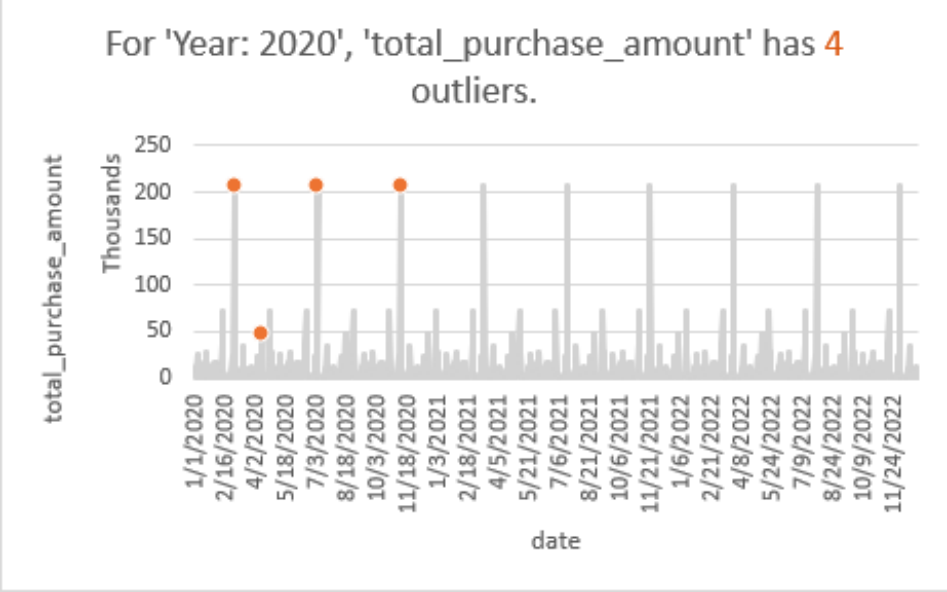
	J	K
	ncy	total_purchase_amount	Ye
	=[@[quantity_sold]]*[@[sales_price]]	

-Handling outliers:

we find outliers by using pivot tables and charts, handled the outliers by replacing them with the quartiles.

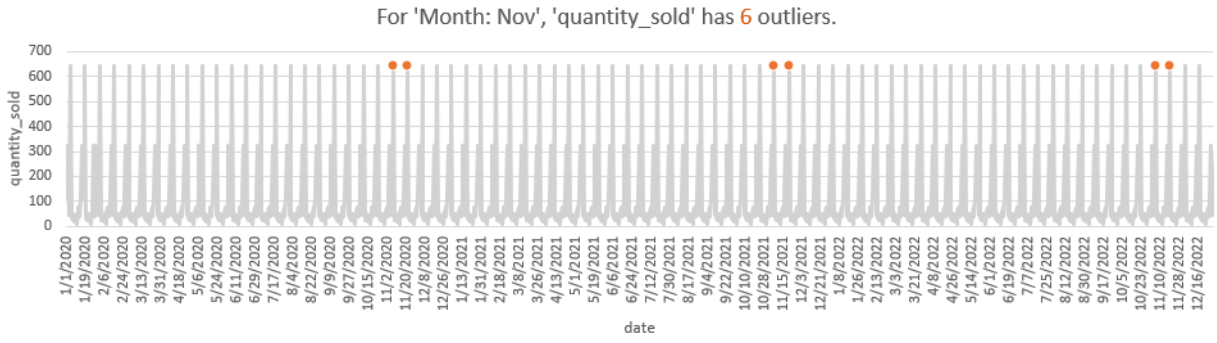
here are some outliers founded in total_purchase_amount:

At year 2020 we have outliers (47582, 207689)



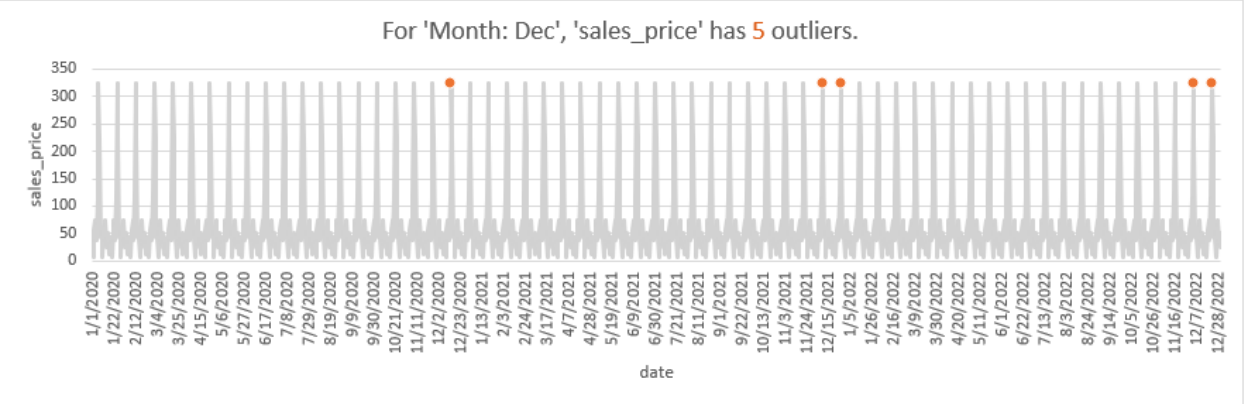
Outliers founded in quantity_sold:

For month November every year (643)



Outlier founded in sales_price:

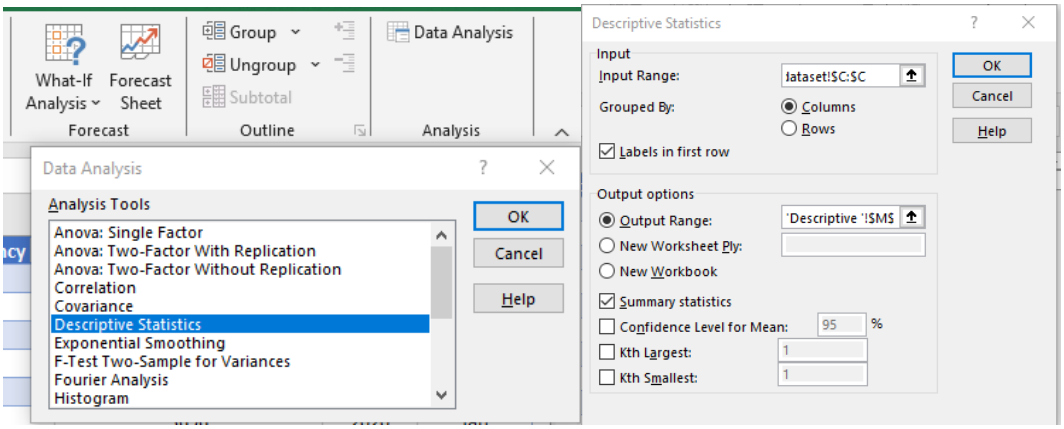
For month December every year (323)



Exploratory Data Analysis steps

We perform exploratory analysis to understand summary statistics for each numerical data by using descriptive statistics with excel

Go to data -> data analysis -> descriptive statistics then we put the input range and output range



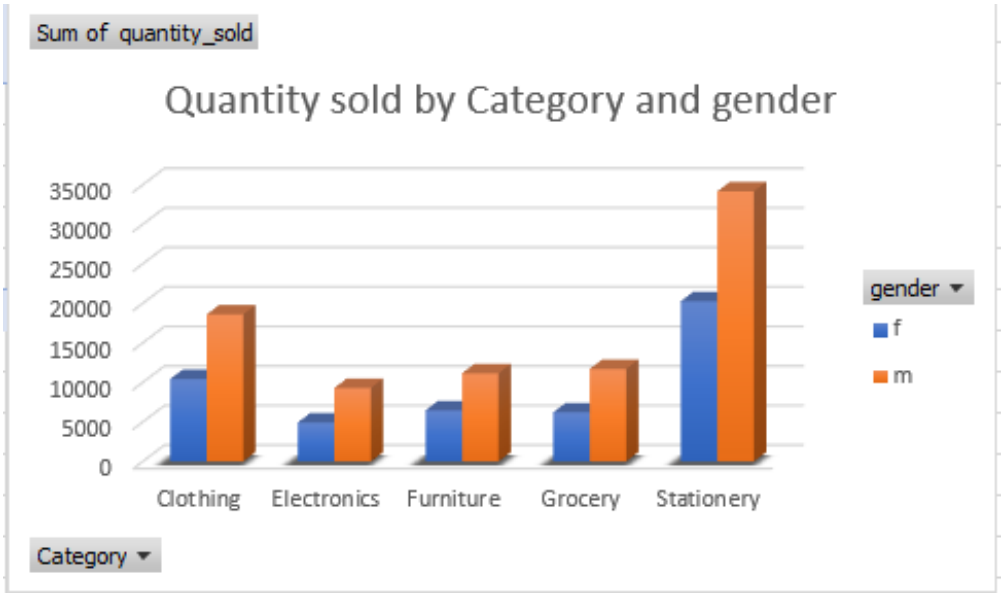
so, we have repeated these steps to make a summary statistic for (Quantity_sold, Sales_price, age, total_purchase_amount)

	A	B	C	D	E	F	G	H	I	J	K
1	*quantity_sold*			*sales_price*			*age*			*total_purchase_amount*	
2											
3	Mean	121.56		Mean	69.32		Mean	40.34		Mean	8412.87
4	Standard Error	5.06		Standard Error	2.36		Standard Error	0.41		Standard Error	651.58
5	Median	51.00		Median	44.00		Median	36.00		Median	2420.00
6	Mode	112.00		Mode	44.00		Mode	23.00		Mode	4928.00
7	Standard Deviation	167.67		Standard Deviation	78.23		Standard Deviation	13.74		Standard Deviation	21571.03
8	Sample Variance	28114.90		Sample Variance	6119.63		Sample Variance	188.73		Sample Variance	465309423.89
9	Kurtosis	3.91		Kurtosis	4.11		Kurtosis	-1.19		Kurtosis	58.53
10	Skewness	2.19		Skewness	2.22		Skewness	0.41		Skewness	6.99
11	Range	636.00		Range	316.00		Range	42.00		Range	207640.00
12	Minimum	7.00		Minimum	7.00		Minimum	23.00		Minimum	49.00
13	Maximum	643.00		Maximum	323.00		Maximum	65.00		Maximum	207689.00
14	Sum	133229.00		Sum	75976.00		Sum	44216.00		Sum	9220508.00
15	Count	1096.00		Count	1096.00		Count	1096.00		Count	1096.00

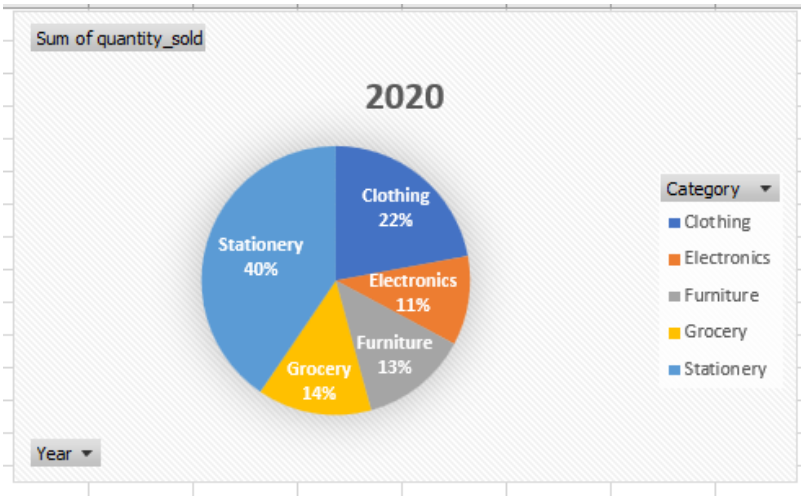
Data visualization steps:

In data visualization we use pivot charts in excel.

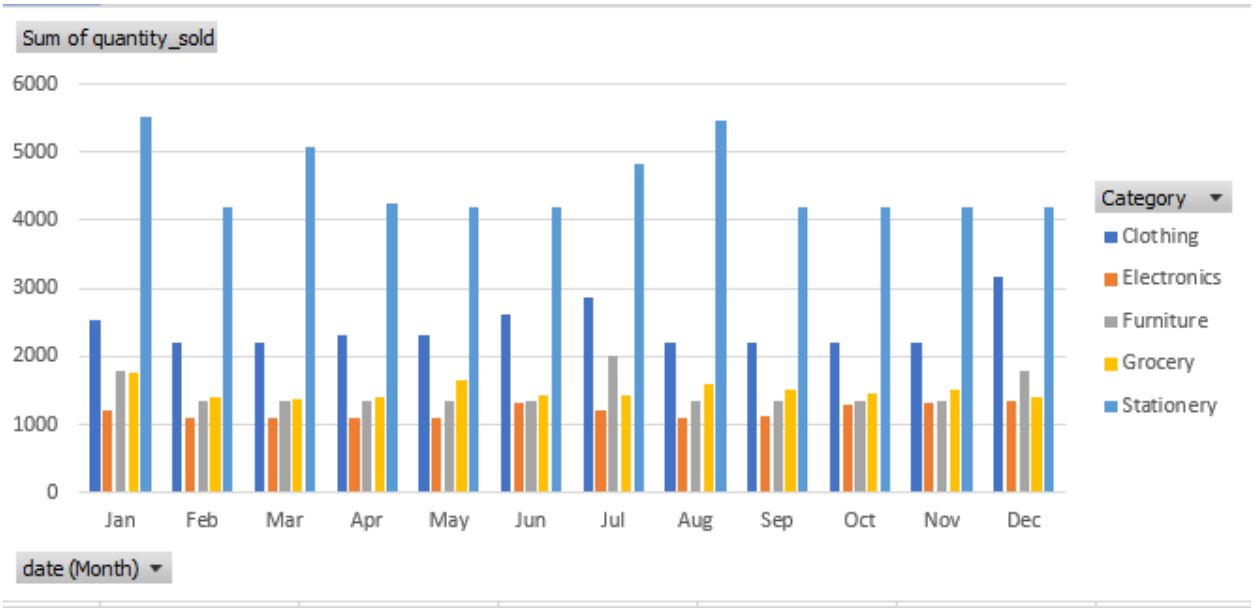
-We make a column chart to show the relation between the gender and the quantity sold for each category to this gender (male and female):



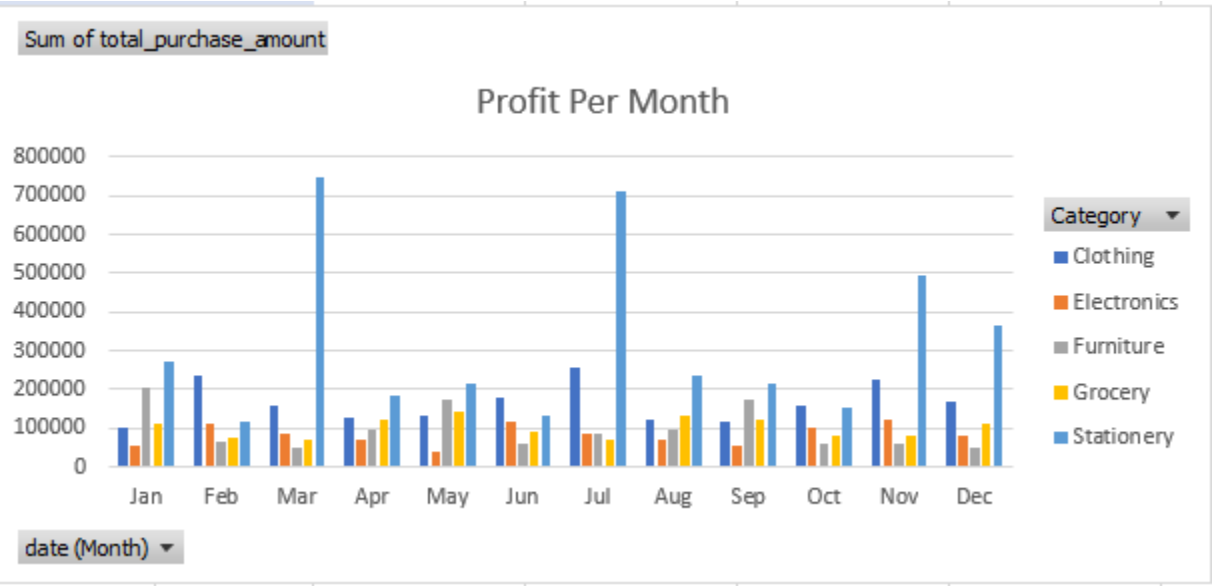
-a pie chart to show the percentage of the quantity sold for each category per year:



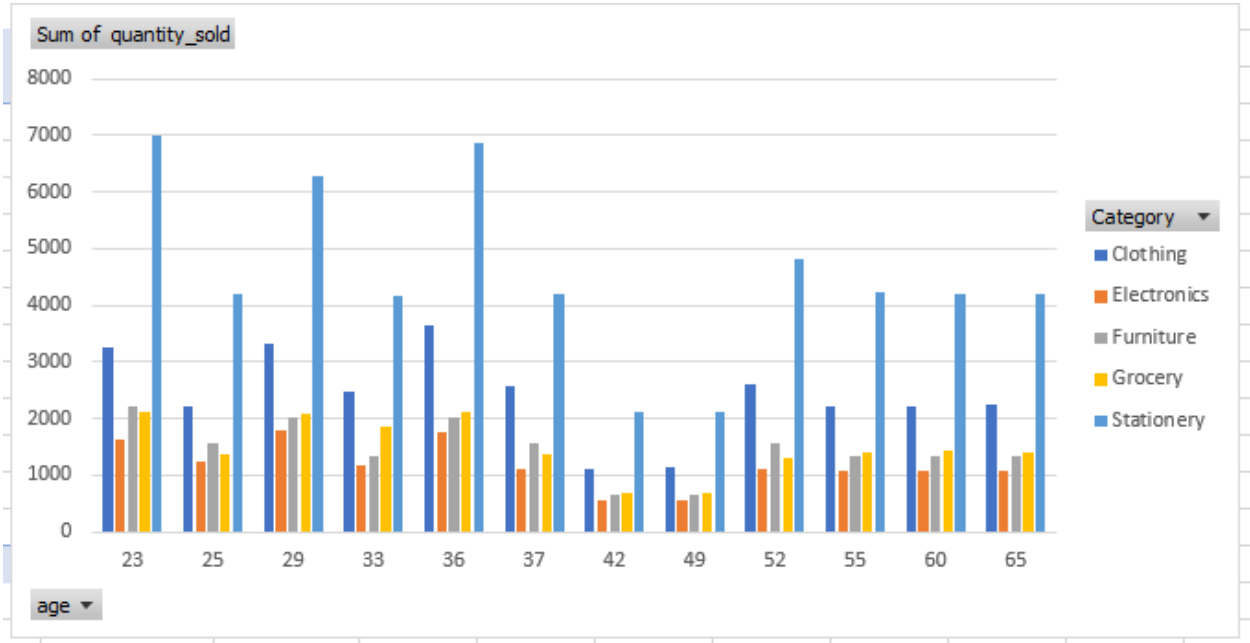
-column chart to show the quantity sold per month for each category:



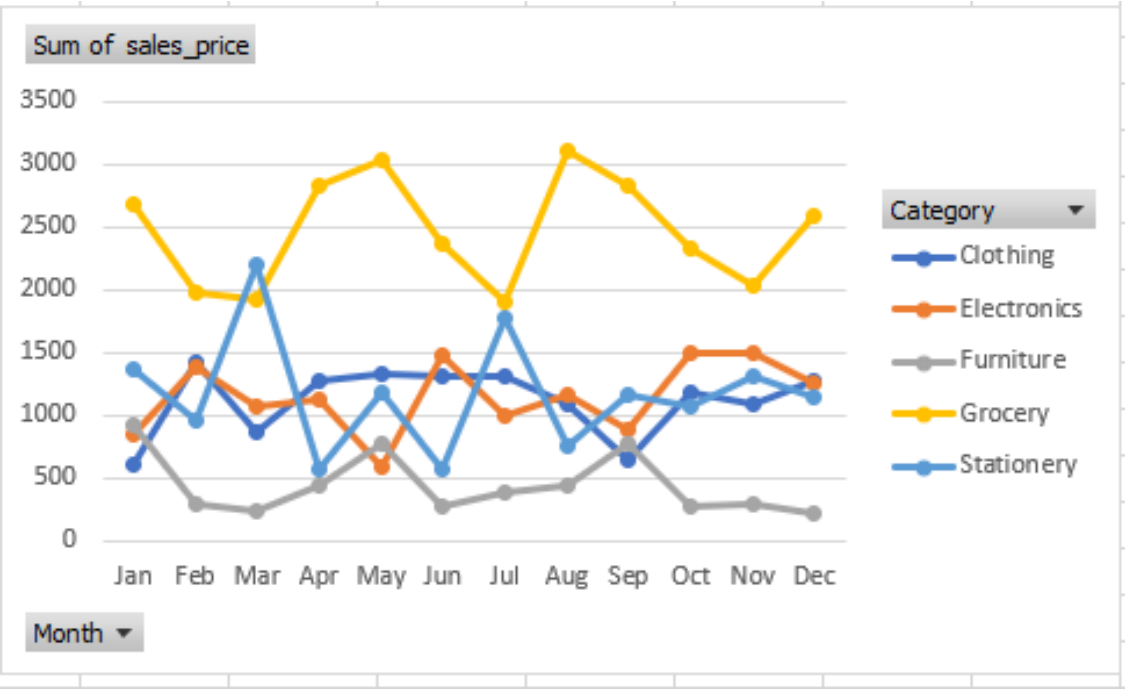
-column chart to show the total purchase amount per month for each category:



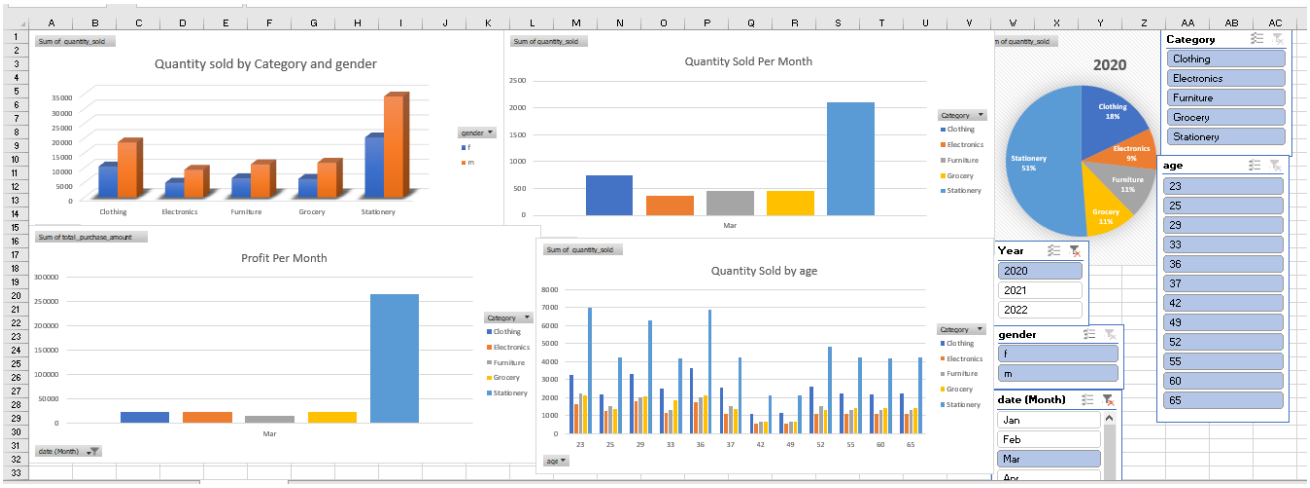
-column chart to show the relation between quantity sold for each category and the age:



-a line chart shows the trend of sales price of each category per month:



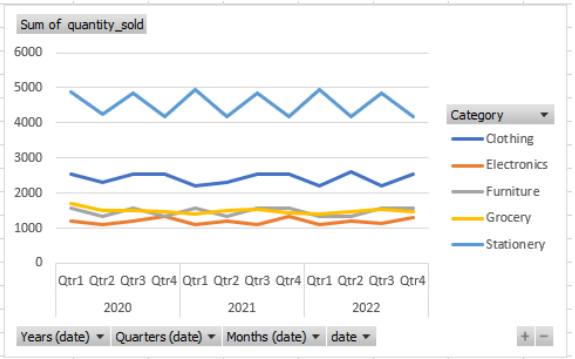
We created a dashboard containing a slicer for (date, category, gender and age)



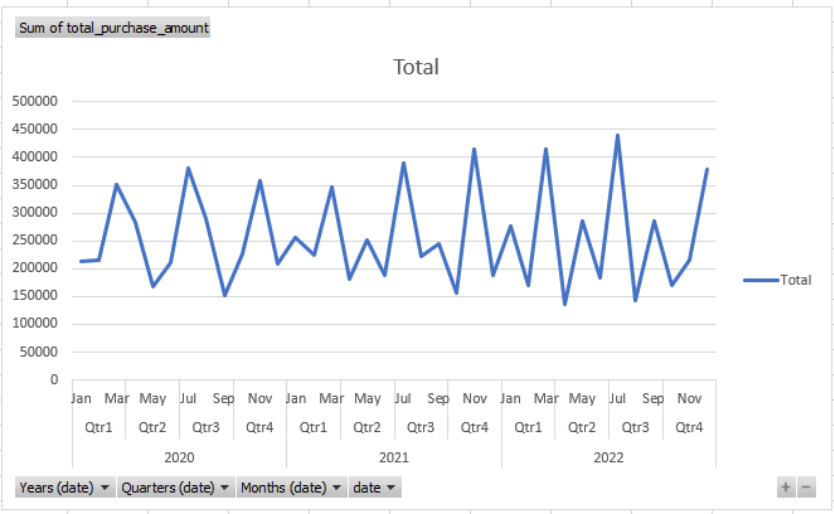
Time series forecasting

Time series analysis for quantity sold and total purchase amount:

Sum of quantity_sold Column Labels / Row Labels	Clothing	Electronics	Furniture	Grocery	Stationery	Grand Total
2020						
Qtr1	2531	1198	1561	1711	4890	11891
Qtr2	2298	1086	1338	1501	4239	10462
Qtr3	2531	1198	1561	1503	4831	11624
Qtr4	2531	1336	1338	1453	4188	10846
2021						
Qtr1	2208	1086	1561	1393	4941	11189
Qtr2	2298	1198	1338	1501	4188	10523
Qtr3	2531	1086	1561	1525	4831	11534
Qtr4	2531	1336	1561	1431	4188	11047
2022						
Qtr1	2208	1086	1338	1415	4941	10988
Qtr2	2621	1198	1338	1479	4188	10824
Qtr3	2208	1130	1561	1525	4831	11255
Qtr4	2531	1292	1561	1474	4188	11046
Grand Total	29027	14230	17617	17911	54444	133229



Row Labels	Sum of total_purchase_amount
2020	
Qtr1	
Jan	213016
Feb	215129
Mar	350395
Qtr2	
Apr	283516
May	167515
Jun	211503
Qtr3	
Jul	381713
Aug	291158
Sep	151818
Qtr4	
Oct	227351
Nov	357349
Dec	209267
2021	
Qtr1	
Jan	257102
Feb	224151
Mar	345827



Quantitative Forecasting

We Calculate Exponential Smoothing and Forecasted Quantity Sold and Sales Price ,and by Multiply them with each other to get Total Purchase Amount by this Function: `=M2+0.1*(C2-M2)` Ft+1= Ft+0.1*(At-Ft)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	date	product_ID	quantity_sold	sales_price	Category	customer_id	age	gender	purchase_frequency	total_purchase_amount	Year	Month	F(Quantity Sold)	F(Sales Price	F(Total Purchase)
2	1/1/2020	22	112	44	Electronics	1	25	m	1	4928	2020	Jan	112	44	4928
3	1/2/2020	32	323	7	Clothing	2	33	m	5	2261	2020	Jan	112	44	4928
4	1/3/2020	11	223	55	Furniture	3	52	f	6	12265	2020	Jan	133	40	5364
5	1/4/2020	33	43	74	Grocery	4	29	m	8	3182	2020	Jan	142	42	5935
6	1/5/2020	26	643	39	Stationery	5	36	m	5	25077	2020	Jan	132	45	5947
7	1/6/2020	51	33	112	Stationery	6	37	m	4	3696	2020	Jan	183	44	8136
8	1/7/2020	26	26	323	Stationery	7	42	f	3	8398	2020	Jan	168	51	8606
9	1/8/2020	17	51	223	Stationery	8	23	f	3	11373	2020	Jan	154	78	12065
10	1/9/2020	22	22	43	Grocery	9	23	m	2	946	2020	Jan	144	93	13337
11	1/10/2020	32	44	44	Clothing	10	65	f	8	1936	2020	Jan	132	88	11552
12	1/11/2020	11	7	7	Clothing	11	55	m	7	49	2020	Jan	123	83	10246

And to predict future sales for year 2023 based on historical trends:

1098	1/1/2023									0	2023	Jan	119	67	7920
1099	1/2/2023									0	2023	Jan	107	60	6416
1100	1/3/2023									0	2023	Jan	96	54	5197
1101	1/4/2023									0	2023	Jan	86	49	4209
1102	1/5/2023									0	2023	Jan	78	44	3409
1103	1/6/2023									0	2023	Jan	70	39	2762
1104	1/7/2023									0	2023	Jan	63	35	2237
1105	1/8/2023									0	2023	Jan	57	32	1812
1106	1/9/2023									0	2023	Jan	51	29	1468
1107	1/10/2023									0	2023	Jan	46	26	1189
1108	1/11/2023									0	2023	Jan	41	23	963
1109	1/12/2023									0	2023	Jan	37	21	780
1110	1/13/2023									0	2023	Jan	34	19	632
1111	1/14/2023									0	2023	Jan	30	17	512
1112	1/15/2023									0	2023	Jan	27	15	415
1113	1/16/2023									0	2023	Jan	24	14	336
1114	1/17/2023									0	2023	Jan	22	12	272
1115	1/18/2023									0	2023	Jan	20	11	220
1116	1/19/2023									0	2023	Jan	18	10	178
1117	1/20/2023									0	2023	Jan	16	9	145
1118	1/21/2023									0	2023	Jan	14	8	117
1119	1/22/2023									0	2023	Jan	13	7	95
1120	1/23/2023									0	2023	Jan	12	7	77
1121	1/24/2023									0	2023	Jan	11	6	62
1122	1/25/2023									0	2023	Jan	9	5	50
1123	1/26/2023									0	2023	Jan	9	5	41
1124	1/27/2023									0	2023	Jan	8	4	33
1125	1/28/2023									0	2023	Jan	7	4	27
1126	1/29/2023									0	2023	Jan	6	3	22
1127	1/30/2023									0	2023	Jan	6	3	18

Customer Segmentation:

Data Loading and Preprocessing

```
# Import necessary libraries
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report
```

✓ 0.0s

The initial step involves loading the data from the Excel sheet and extracting relevant features for analysis.

The features selected include ' Quantity_sold', ' sales_price', 'age', 'purchase_frequency', and 'total_purchase_amount'.

The target variable is 'Category', representing the category to which each customer belongs.

```
# Data preprocessing
X = data[[' Quantity_sold', ' sales_price', 'age', 'purchase_frequency', 'total_purchase_amount']]
y = data['Category']
```

✓ 0.0s

Feature Standardization:

To ensure consistent scaling.

```
# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

✓ 0.0s

Customer Segmentation using KMeans Clustering

```
# Customer Segmentation using KMeans clustering
kmeans = KMeans(n_clusters=3)
data['Cluster'] = kmeans.fit_predict(X_scaled)
```

✓ 0.0s

Data Splitting for Classification:

The dataset is split into training and testing sets to train and evaluate the neural network classifier.

Train set is 80% of the data

Test set is 20% of the data

The Targeted: y

The features: X_scaled

```
# Split the data into training and testing sets for classification
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=35)
```

✓ 0.0s

Neural Network Classification

A Multi-Layer Perceptron (MLP) classifier is employed to classify customers based on the provided features.

Using activation function: relu

The first hidden layer has 64 neurons.

The second hidden layer has 32 neurons.

```
# Neural Network Classification
classifier = MLPClassifier(hidden_layer_sizes=(64, 32), max_iter=100, activation='relu', random_state=35)
classifier.fit(X_train, y_train)
```

✓ 0.6s

Model Evaluation

The performance of the classifier is evaluated using the classification report.

Accuracy =0.72

```
# Predictions on the test set
y_pred = classifier.predict(X_test)
```

✓ 0.0s

```
# Evaluate the model
print("Classification Report:\n", classification_report(y_test, y_pred))
```

✓ 0.0s

```
Classification Report:
              precision    recall  f1-score   support

    Clothing       0.95      0.49      0.65        41
 Electronics       0.77      0.57      0.65        30
   Furniture       1.00      1.00      1.00        18
     Grocery       0.62      0.94      0.74        96
  Stationery       1.00      0.37      0.54        35

    accuracy                           0.72       220
   macro avg       0.87      0.67      0.72       220
weighted avg       0.79      0.72      0.70       220
```

Customer Segmentation Results

The cluster information obtained from KMeans clustering is appended to the predictions and displayed in the results dataframe.

```
# Append the cluster information to the predictions
X_test_with_cluster = pd.DataFrame(X_test, columns=X.columns)
X_test_with_cluster['Cluster'] = kmeans.predict(X_test)
```

✓ 0.0s

```
# Output the results to the console
result_df = pd.DataFrame({'Customer_ID': data['customer_id'], 'Category': y, 'Cluster': data['Cluster']})
print("\nCustomer Segmentation Results:\n", result_df)
```

✓ 0.0s

```
Customer Segmentation Results:
      Customer_ID     Category  Cluster
0               1  Electronics        1
1               2     Clothing        1
2               3    Furniture        0
3               4      Grocery        1
4               5   Stationery        2
...           ...          ...      ...
1091           17  Electronics        1
1092           18  Electronics        1
1093           19     Clothing        0
1094           20    Furniture        1
1095           21      Grocery        0

[1096 rows x 3 columns]
```