In [ ]:

```
import pandas as pd
import requests
from google.colab import auth
auth.authenticate_user()

import gspread
from google.auth import default
creds, _ = default()

gc = gspread.authorize(creds)
```

In [ ]:

```
worksheet = gc.open_by_url('https://docs.google.com/spreadsheets/d/1hNJzHEkYi5l9vVy5ba0

type(worksheet)

rows = worksheet.sheet1.get_all_values()

df = pd.DataFrame.from_records(rows)



df.columns = df.iloc[0]
```

In [ ]:

```
df.head(3)
```

Out[ ]:

	Timestamp	What is your age?	What gender do you identify with?	Are you currently employed?	What's your income?	How many hours on average per day do you spend browsing products online?	What category of products do you typically purchase?	(Optional) Do you shop more online or in-person? Or browse online first before shopping in-person?	Which e-commerce websites do you mostly browse and shop on?	W
0	Timestamp	What is your age?	What gender do you identify with?	Are you currently employed?	What's your income?	How many hours on average per day	What category of products do you typically pur...	(Optional) Do you shop more online or in-perso...	Which e-commerce websites do you mostly browse...	V

						do you spend...			
1	3/15/2023 10:22:12	22	Man	No	N/A	0-1	Electronics and gadgets		I don't really shop
2	3/15/2023 10:22:38	20	Woman	No	N/A	0-1	Clothing and accessories, Books and entertainment	I shop online	Amazon

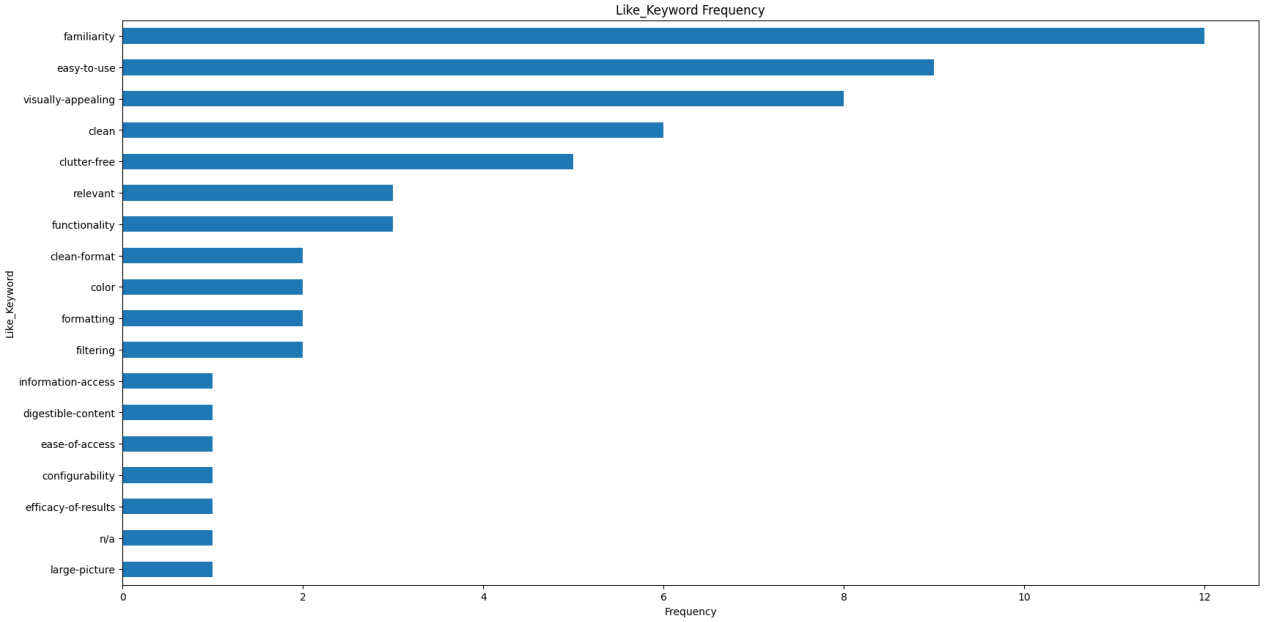
3 rows × 26 columns

In [ ]:

```
# get the most used keyword in Like_keywords

like_keywords = df['Like_Keyword']
# drop Like_Keyword word


like_keywords = like_keywords.to_frame()


like_keywords = like_keywords['Like_Keyword'].str.split(',', expand=True).stack().reset
like_keywords['Like_Keyword'] = like_keywords['Like_Keyword'].str.replace(',', '')
like_keywords['Like_Keyword'] = like_keywords['Like_Keyword'].str.strip()

like_keywords = like_keywords.drop(0)

ax = like_keywords['Like_Keyword'].value_counts().sort_values(ascending=True).plot(kind

# add axes
ax.set_ylabel("Like_Keyword")
ax.set_xlabel("Frequency")
ax.set_title("Like_Keyword Frequency")
```

Out[ ]:

```
Text(0.5, 1.0, 'Like_Keyword Frequency')
```

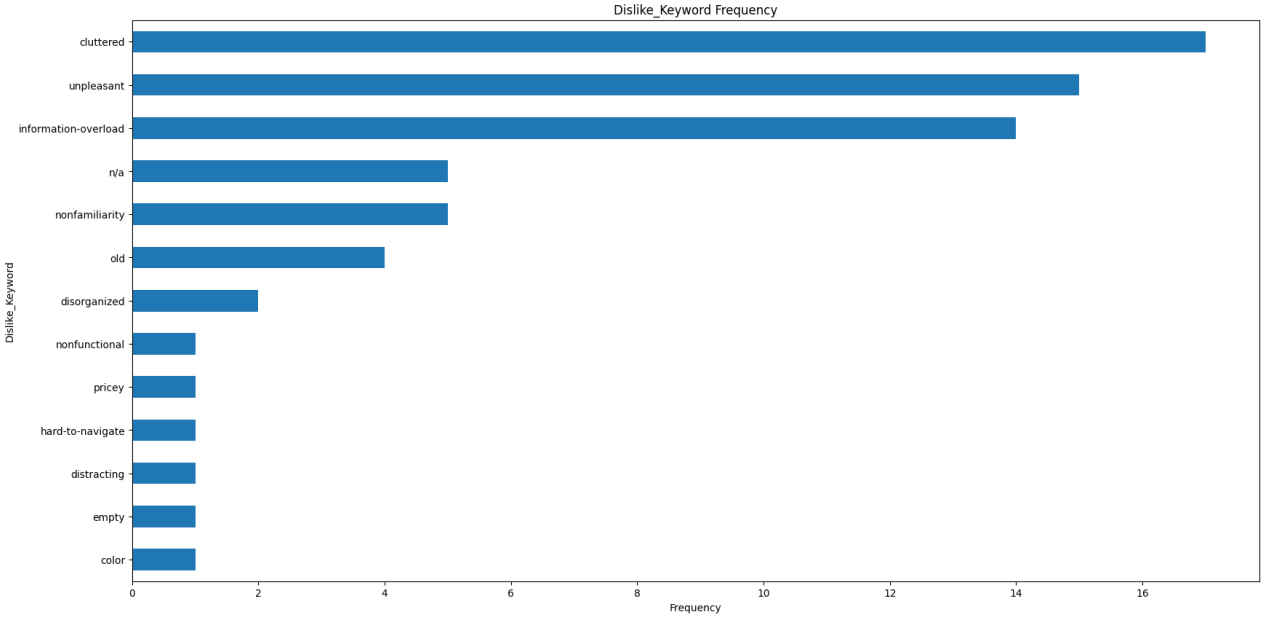


In [ ]:

```
# get the most used keywords in Dislike_Keyword

dislike_keywords = df['Dislike_Keyword']

dislike_keywords = dislike_keywords.to_frame()
dislike_keywords = dislike_keywords['Dislike_Keyword'].str.split(',', expand=True).stac
dislike_keywords['Dislike_Keyword'] = dislike_keywords['Dislike_Keyword'].str.replace('
dislike_keywords['Dislike_Keyword'] = dislike_keywords['Dislike_Keyword'].str.strip()

dislike_keywords = dislike_keywords.drop(0)

ax = dislike_keywords['Dislike_Keyword'].value_counts().sort_values(ascending=True).plo

# add axes
ax.set_ylabel("Dislike_Keyword")
ax.set_xlabel("Frequency")
ax.set_title("Dislike_Keyword Frequency")
```

Out[ ]:

```
Text(0.5, 1.0, 'Dislike_Keyword Frequency')
```

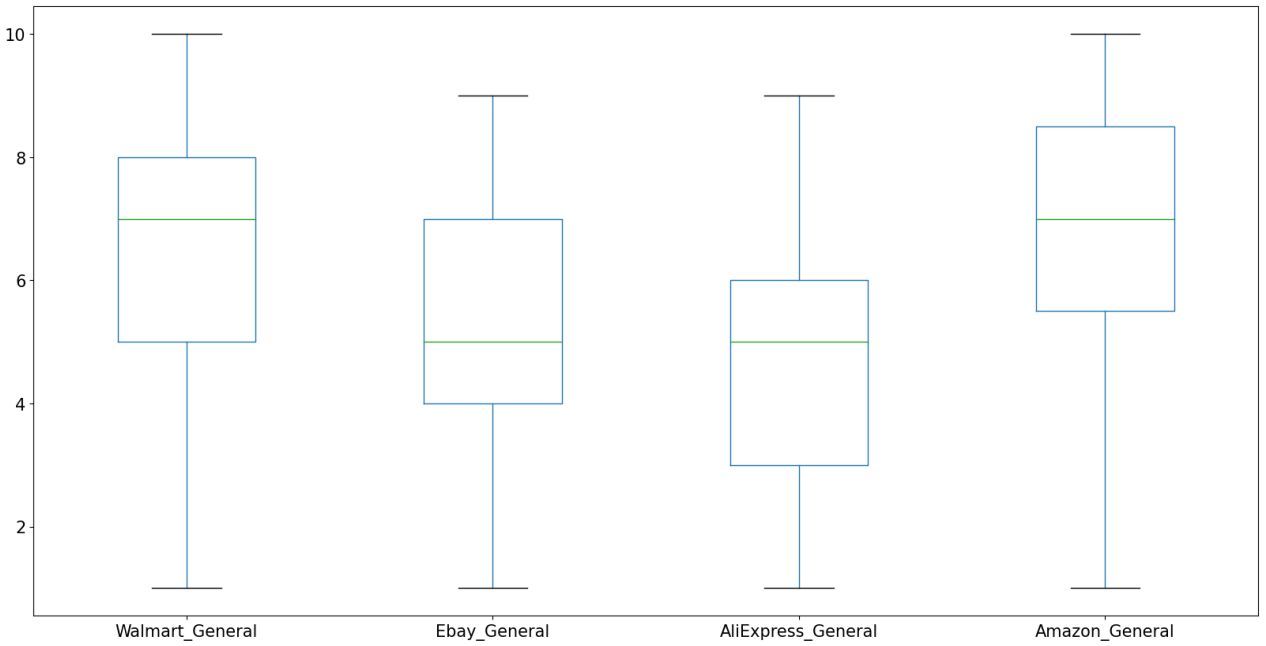


In [ ]:

```
df_general= df[['Walmart_General','Ebay_General', 'AliExpress_General', 'Amazon_General
# df_general.drop(0)
# df_general.boxplot(figsize=(20,10), column=['Walmart_General', 'Ebay_General', 'AliEx

# df_general =df_general.drop(0)

df_general.columns = df_general.iloc[0]
df_general = df_general[1:]

df_general = df_general.astype(int)
df_general.boxplot(figsize=(20,10), column=['Walmart_General', 'Ebay_General', 'AliExpr
```

Out[ ]:

```
<Axes: >
```



In [ ]:

```
import pandas as pd
from scipy.stats import f_oneway
from statsmodels.stats.multicomp import pairwise_tukeyhsd
```

```
import numpy as np

# Perform one-way ANOVA
fvalue, pvalue = f_oneway(df_general['Ebay_General'], df_general['Amazon_General'], df_

# Print the results
print('F-value:', fvalue)
print('P-value:', pvalue)

# Perform Tukey HSD test
data_general = np.concatenate([df_general['Ebay_General'], df_general['Amazon_General']
labels_general = ['Ebay'] * len(df_general) + ['Amazon'] * len(df_general) + ['Walmart'


# Perform Tukey test
tukey_results = pairwise_tukeyhsd(data_general, labels_general)

# Print the results
print(tukey_results)
```

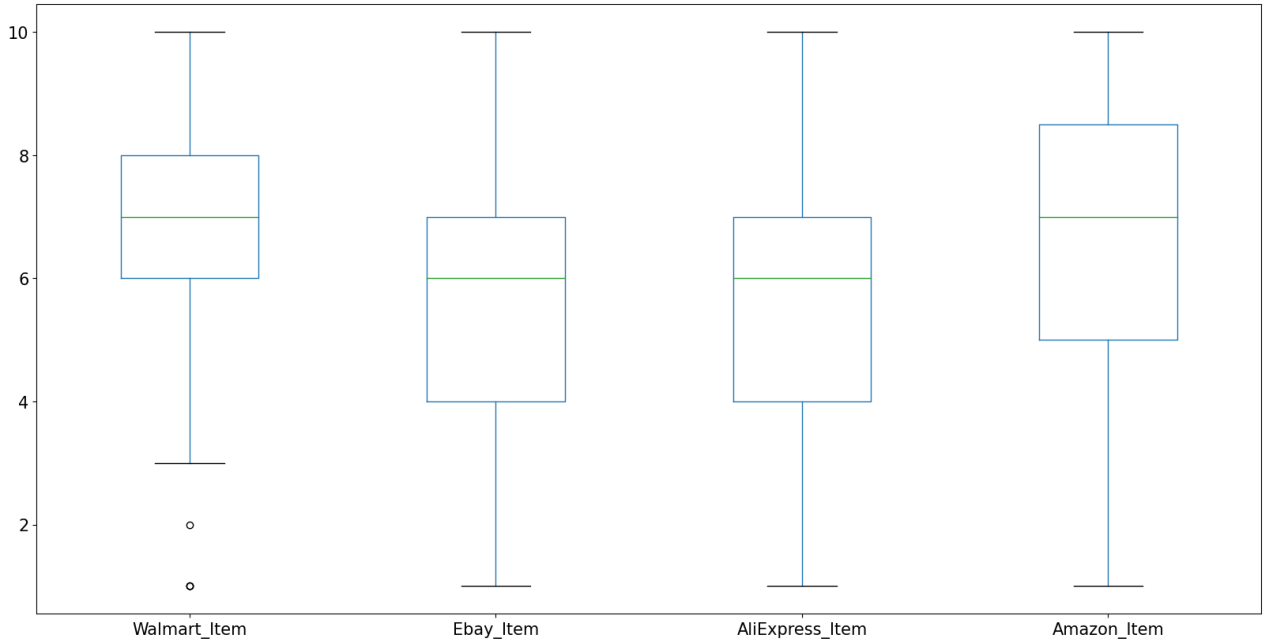
```
F-value: 10.4126652154737
P-value: 2.1361747930869428e-06
   Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
  group1    group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
AliExpress  Amazon   2.1765    0.0  1.0768  3.2761   True
AliExpress    Ebay   0.6667 0.3977  -0.433  1.7663  False
AliExpress Walmart   1.6078 0.0011  0.5082  2.7075   True
    Amazon    Ebay  -1.5098 0.0026 -2.6094 -0.4102   True
    Amazon Walmart  -0.5686 0.5388 -1.6683   0.531  False
      Ebay Walmart   0.9412  0.122 -0.1585  2.0408  False
-----------------------------------------------------
```

In [ ]:

```
df_item= df[['Walmart_Item','Ebay_Item', 'AliExpress_Item', 'Amazon_Item' ]]

df_item.columns = df_item.iloc[0]
df_item = df_item[1:]

df_item = df_item.astype(int)
df_item.boxplot(figsize=(20,10), column=['Walmart_Item', 'Ebay_Item', 'AliExpress_Item'

# TODO:ADD AXES
```

Out[ ]:

```
<Axes: >
```



In [ ]:

```
import pandas as pd
from scipy.stats import f_oneway
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Perform one-way ANOVA
fvalue, pvalue = f_oneway(df_item['Ebay_Item'], df_item['Amazon_Item'], df_item['Walmar

# Print the results
print('F-value:', fvalue)
print('P-value:', pvalue)

# Perform Tukey HSD test
data_item = np.concatenate([df_item['Ebay_Item'], df_item['Amazon_Item'], df_item['Walm
labels_general = ['Ebay'] * len(df_item) + ['Amazon'] * len(df_item) + ['Walmart'] * le


tukey_results_item = pairwise_tukeyhsd(data_item, labels_general)

# Print the results
print(tukey_results_item.summary())
```

```
F-value: 5.435884102030747
P-value: 0.0012967640192580608
   Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
  group1    group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
AliExpress  Amazon   1.3725 0.0107  0.2367  2.5084   True
AliExpress    Ebay    0.098  0.996 -1.0378  1.2339  False
AliExpress Walmart   1.2157 0.0307  0.0798  2.3516   True
    Amazon    Ebay  -1.2745  0.021 -2.4104 -0.1386   True
    Amazon Walmart  -0.1569 0.9843 -1.2927   0.979  False
      Ebay Walmart   1.1176 0.0556 -0.0182  2.2535  False
-----------------------------------------------------
```

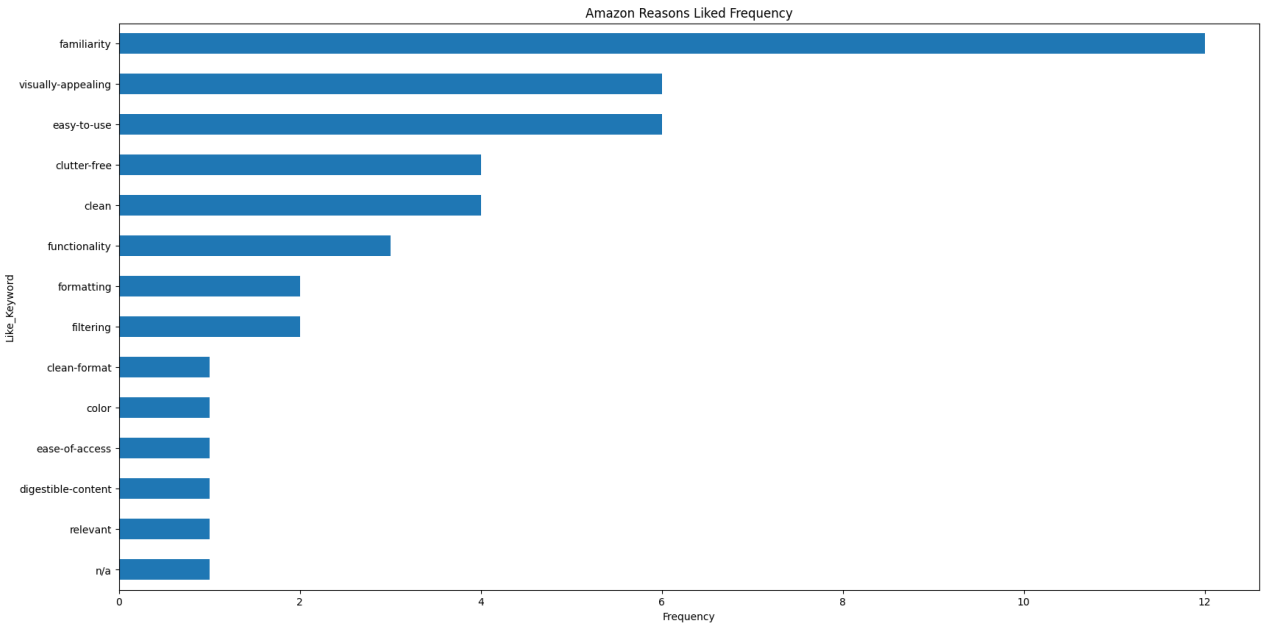
In [ ]:

```
df['Which interface did/would you rank the highest?'].value_counts()
```

Out[ ]:
```
Amazon                                                39
Walmart                                                9
Ebay                                                   2
Which interface did/would you rank the highest?        1
Ali-Express                                            1
Name: Which interface did/would you rank the highest?, dtype: int64
```

In [ ]:
```
# People who liked Amazon used these keywords
amazon_df = df[df['Which interface did/would you rank the highest?'] == 'Amazon']


amazon_reason_liked= amazon_df['Like_Keyword'].to_frame()
amazon_reason_liked = amazon_reason_liked['Like_Keyword'].str.split(',', expand=True).s
amazon_reason_liked['amazon_keyword'] = amazon_reason_liked['amazon_keyword'].str.repla
amazon_reason_liked['amazon_keyword'] = amazon_reason_liked['amazon_keyword'].str.strip

amz = amazon_reason_liked['amazon_keyword'].value_counts().sort_values(ascending=True).


amz.set_ylabel("Like_Keyword")
amz.set_xlabel("Frequency")
amz.set_title("Amazon Reasons Liked Frequency")
```

Out[ ]:
```
Text(0.5, 1.0, 'Amazon Reasons Liked Frequency')
```



In [ ]:
```
df['Which interface did/would you rank the lowest?'].value_counts()
```

Out[ ]:
```
Ali-Express                                           25
Ebay                                                  21
Walmart                                                4
Which interface did/would you rank the lowest?         1
Amazon                                                 1
Name: Which interface did/would you rank the lowest?, dtype: int64
```

In [ ]:
```
ali_df = df[df['Which interface did/would you rank the lowest?'] == 'Ali-Express']

print(ali_df['Dislike_Keyword'].value_counts())
```

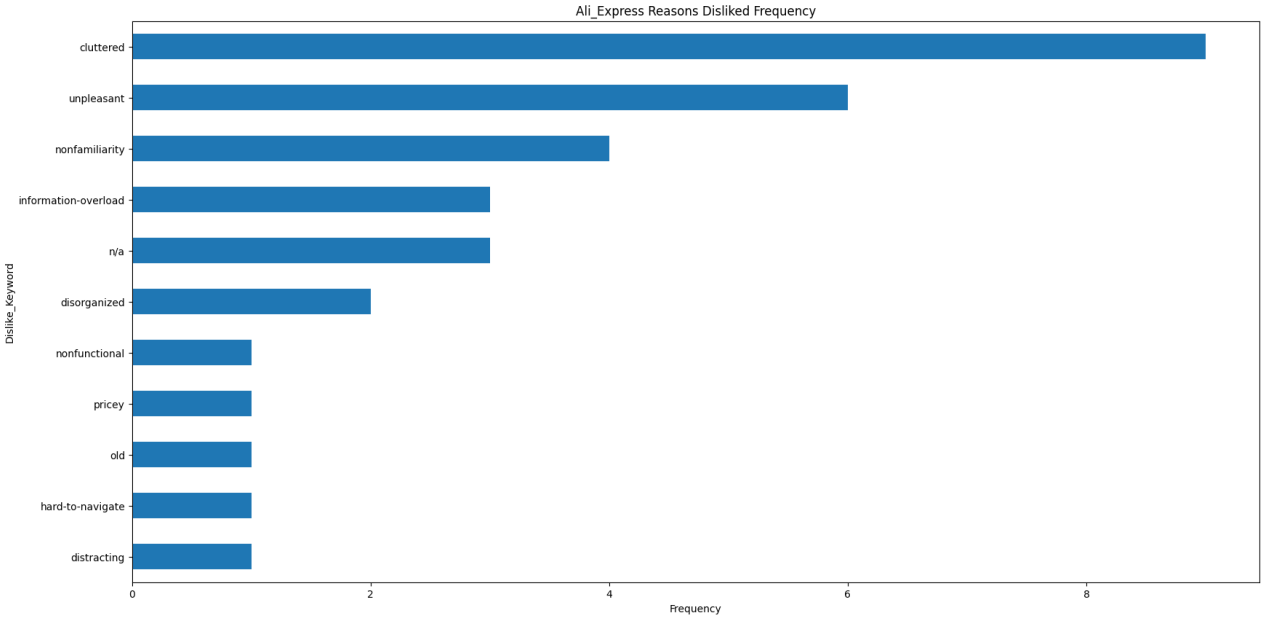
```
cluttered                          6
nonfamiliarity                     4
n/a                                3
unpleasant                         3
information-overload               2
cluttered,disorganized             1
cluttered,information-overload     1
distracting,disorganized           1
hard-to-navigate, unpleasant       1
cluttered, unpleasant, old         1
unpleasant, pricey                 1
nonfunctional                      1
Name: Dislike_Keyword, dtype: int64
```

In [ ]:

```
# People who hated Ali used these keywords
ali_df= ali_df['Dislike_Keyword'].to_frame()
ali_df = ali_df['Dislike_Keyword'].str.split(',', expand=True).stack().reset_index(leve
ali_df['ali_keyword'] = ali_df['ali_keyword'].str.replace(',', '')
ali_df['ali_keyword'] = ali_df['ali_keyword'].str.strip()

ali = ali_df['ali_keyword'].value_counts().sort_values(ascending=True).plot(kind='barh'

ali.set_ylabel("Dislike_Keyword")
ali.set_xlabel("Frequency")
ali.set_title("Ali_Express Reasons Disliked Frequency")
```

Out[ ]:

```
Text(0.5, 1.0, 'Ali_Express Reasons Disliked Frequency')
```



In [ ]:

```
df_general.describe()
```

Out[ ]:

	Walmart_General	Ebay_General	AliExpress_General	Amazon_General
count	51.000000	51.000000	51.000000	51.000000
mean	6.176471	5.235294	4.568627	6.745098
std	2.132659	1.934820	2.100047	2.381958
min	1.000000	1.000000	1.000000	1.000000

	Walmart_General	Ebay_General	AliExpress_General	Amazon_General
25%	5.000000	4.000000	3.000000	5.500000
50%	7.000000	5.000000	5.000000	7.000000
75%	8.000000	7.000000	6.000000	8.500000
max	10.000000	9.000000	9.000000	10.000000

In [ ]:
```
df_item.describe()
```

Out[ ]:

	Walmart_Item	Ebay_Item	AliExpress_Item	Amazon_Item
count	51.000000	51.000000	51.000000	51.000000
mean	6.725490	5.607843	5.509804	6.882353
std	2.333053	2.020677	2.138902	2.346462
min	1.000000	1.000000	1.000000	1.000000
25%	6.000000	4.000000	4.000000	5.000000
50%	7.000000	6.000000	6.000000	7.000000
75%	8.000000	7.000000	7.000000	8.500000
max	10.000000	10.000000	10.000000	10.000000