```
import pandas as pd
```

```
product_data = pd.read_csv('/content/assignment 3 data 1.csv')
review_data = pd.read_csv('/content/assignment 3 data 2 reviews.csv')
```

product_data and review_data is dataframe of product and review datasets

```
product_data.head()
```

	product_id	product_name	product_price	price_currency	product_availability	
0	103205	Hwipure, Disposable KF94 (N95 / KN95/ FFP2) ...	2.95	AUD	http://schema.org/InStock	http
1	101774	HIGUARD, Disposable KF94 (N95 / KN95/ FFP2) ...	2.95	AUD	http://schema.org/InStock	https:
2	101955	SunJoy, KN95, Professional Protective Disposab...	8.86	AUD	http://schema.org/InStock	htt
3	103838	Lozperi, Copper Mask, Adult, Black, 1 Mask	6.85	AUD	http://schema.org/InStock	htt
4	102734	Zidian, Disposable Protective Mask, 50 Pack	15.35	AUD	http://schema.org/InStock	h

```
product_data.info()
```

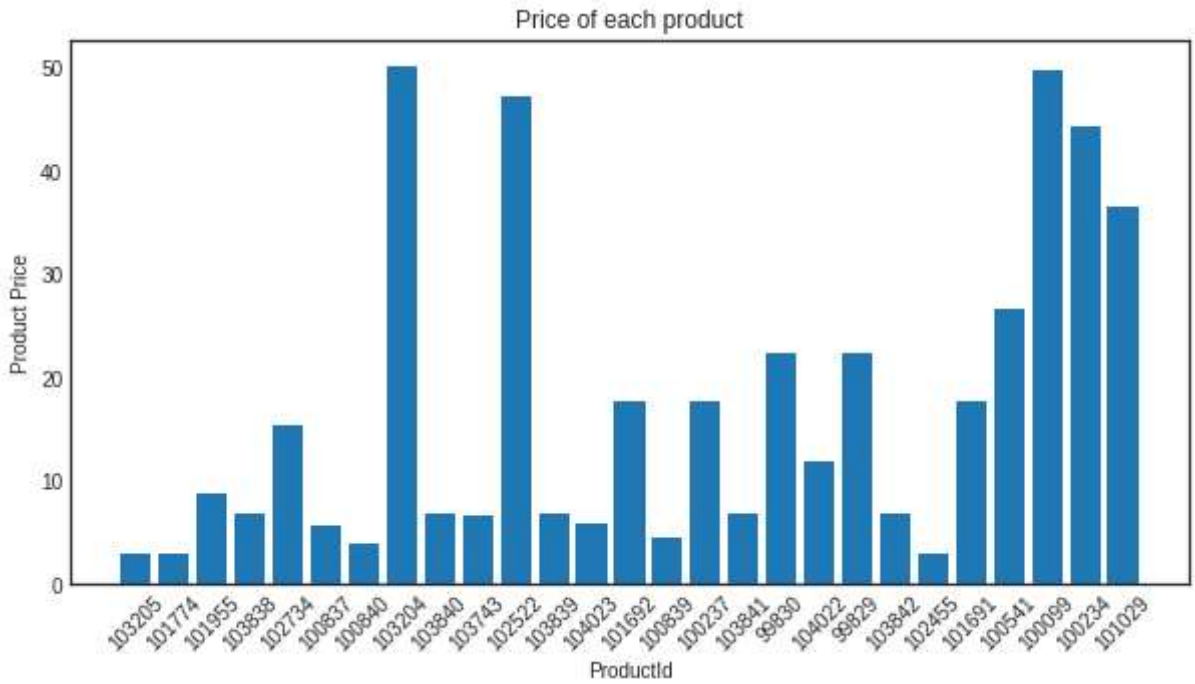
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   product_id            27 non-null     int64
 1   product_name          27 non-null     object
```

```
 2   product_price         27 non-null     float64
 3   price_currency        27 non-null     object
 4   product_availability  27 non-null     object
 5   product_url           27 non-null     object
 6   source_url            27 non-null     object
dtypes: float64(1), int64(1), object(5)
memory usage: 1.6+ KB
```

```
product_data.sort_values(by=['product_price'],ascending = False).head(5)
```

	product_id	product_name	product_price	price_currency	product_availability	
7	103204	Hwipure, Disposable KF94 (N95 / KN95/ FFP2) ...	50.19	AUD	http://schema.org/InStock	ht
24	100099	Luseta Beauty, Disposable Protection Face Mask...	49.61	AUD	http://schema.org/InStock	l
10	102522	Dr. Puri, Disposable KF94 (N95 / KN95/ FFP2)...	47.24	AUD	http://schema.org/InStock	h
25	100234	Luseta Beauty, Disposable Protection Face Mask...	44.31	AUD	http://schema.org/InStock	l
26	101029	Landsberg, 3 Ply Disposable Protective Face Ma...	36.54	AUD	http://schema.org/InStock	https

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,5))
plt.bar(x=product_data['product_id'].astype(str),height=product_data['product_price'])
plt.xticks(rotation=45)
plt.title('Price of each product')
plt.ylabel('Product Price')
plt.xlabel('ProductId')
plt.show()
```



There is a high amount of disparity in Product price as some masks' price is its unit price while some are sold as pack of 10 or 25 , which makes its price high as compared to others.

Removing unnecessary columns from product_data dataframe

```
columns = ['product_url','source_url','price_currency','product_availability','product_name']
product_data.drop(columns,axis=1,inplace = True)
```

```
product_data['productId'] = product_data['product_id']
```

```
product_data.drop('product_id',axis = 1,inplace = True)
```

```
product_data
```

	product_price	productId
0	2.95	103205
1	2.95	101774
2	8.86	101955
3	6.85	103838
4	15.35	102734
5	5.61	100837
6	3.93	100840
7	50.19	103204
8	6.85	103840
9	6.61	103743
10	47.24	102522
11	6.85	103839
12	5.91	104023
13	17.72	101692
14	4.49	100839
15	17.72	100237
16	6.85	103841
17	22.44	99830
18	11.81	104022

Exploring review_data

20	6.85	103842

```
review_data.head()
```

gcSummary.reviewCount	ratingValue	reviewText	reviewTitle	reviewed	score	langu
34.0	50	The mask quality and the color is good. It fit...	Dotted Pattern Is Nice	True	1614071051	
37.0	50	Внуку очень понравилось. Удобная маска.	Прекрасно!	False	1612659399	
3.0	40	Easy to put on & comfortable to wear.	Good	False	1612647603	
		Тонкая,				

```
review_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3849 entries, 0 to 3848
Data columns (total 19 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   abuseCount                         3849 non-null   int64
 1   customerNickname                   3849 non-null   object
 2   helpfulNo                          3849 non-null   int64
 3   helpfulYes                         3849 non-null   int64
 4   id                                 3849 non-null   object
 5   imagesCount                        3849 non-null   int64
 6   languageCode                       3849 non-null   object
 7   postedDate                         3849 non-null   object
 8   productId                          3849 non-null   int64
 9   profileInfo.ugcSummary.answerCount 3843 non-null   float64
 10  profileInfo.ugcSummary.reviewCount 3843 non-null   float64
 11  ratingValue                        3849 non-null   int64
 12  reviewText                         3849 non-null   object
 13  reviewTitle                        3849 non-null   object
 14  reviewed                           3849 non-null   bool
 15  score                              3849 non-null   int64
 16  languageCode.1                     3849 non-null   object
 17  translation.reviewText             1994 non-null   object
 18  translation.reviewTitle            1994 non-null   object
dtypes: bool(1), float64(2), int64(7), object(9)
memory usage: 545.1+ KB
```

Merging both review_data and product_data

```
data = pd.merge(product_data,review_data,on = 'productId' )
```

```
data.shape
```

```
(3849, 20)
```

Columns of combined dataset

```
data.columns
```

```
Index(['product_price', 'productId', 'abuseCount', 'customerNickname',
       'helpfulNo', 'helpfulYes', 'id', 'imagesCount', 'languageCode',
       'postedDate', 'profileInfo.ugcSummary.answerCount',
       'profileInfo.ugcSummary.reviewCount', 'ratingValue', 'reviewText',
       'reviewTitle', 'reviewed', 'score', 'languageCode.1',
       'translation.reviewText', 'translation.reviewTitle'],
      dtype='object')
```

Finding null values in the dataset

```
data.isnull().sum()
```

```
product_price                          0
productId                              0
abuseCount                             0
customerNickname                       0
helpfulNo                              0
helpfulYes                             0
id                                     0
imagesCount                            0
languageCode                           0
postedDate                             0
profileInfo.ugcSummary.answerCount     6
profileInfo.ugcSummary.reviewCount     6
ratingValue                            0
reviewText                             0
reviewTitle                            0
reviewed                               0
score                                  0
languageCode.1                         0
translation.reviewText              1855
translation.reviewTitle             1855
dtype: int64
```

OBSERVATIONS:

These null values in translation.reviewText and translation.reviewTitle is because reviews which are in English are left blank.

So next task is to join the reviews in English and reviews translated to English into one column, then

```
data['Final_Review'] = data['translation.reviewText'].fillna(data['reviewText'])
data
```

	product_price	productId	abuseCount	customerNickname	helpfulNo	helpfulYes	
0	2.95	103205	0	iHerb Customer	1	22	b1
1	2.95	103205	1	djagi	0	20	81

```
columns = ['reviewText','translation.reviewText','reviewTitle','translation.reviewTitle']
data = data.drop(columns,axis = 1)
data
```

	product_price	productId	abuseCount	customerNickname	helpfulNo	helpfulYes
0	2.95	103205	0	iHerb Customer	1	22
1	2.95	103205	1	djagi	0	20
2	2.95	103205	0	TMC	0	8
3	2.95	103205	1	INNAg	1	10

```
data.isnull().sum()
```

```
product_price                        0
productId                            0
abuseCount                           0
customerNickname                     0
helpfulNo                            0
helpfulYes                           0
id                                   0
imagesCount                          0
languageCode                         0
postedDate                           0
profileInfo.ugcSummary.answerCount   6
profileInfo.ugcSummary.reviewCount   6
ratingValue                          0
reviewed                             0
score                                0
languageCode.1                       0
Final_Review                         0
dtype: int64
```

```
data.shape
```

```
(3849, 17)
```

```
data = data.dropna()
```

```
data.shape
```

```
(3843, 17)
```

3849 rows × 17 columns

```
data['Date'] = data['postedDate'].str[:4]
data.drop('postedDate',axis = 1, inplace = True)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4913: SettingWithCopyWarning
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  errors=errors,
```

```
data
```

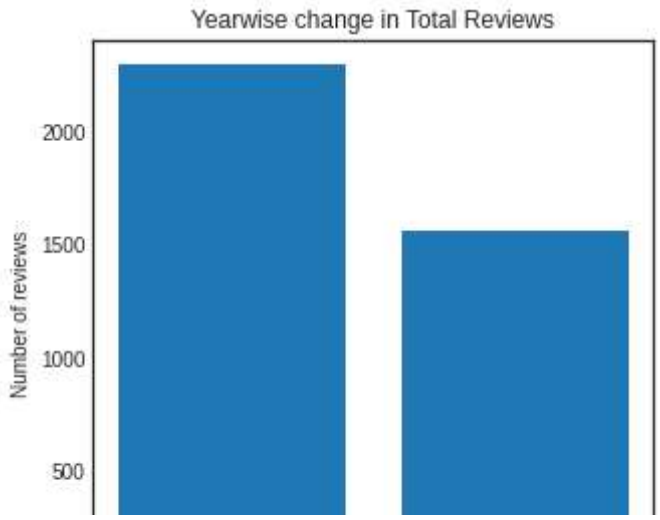
.eInfo.ugcSummary.answerCount	profileInfo.ugcSummary.reviewCount	ratingValue	review
0.0	24.0	50	Fal
0.0	179.0	50	Tr
282.0	108.0	50	Fal
396.0	119.0	50	Tr

```
df_for_year = data.groupby('Date').count()
df_for_year
```

gcSummary.answerCount	profileInfo.ugcSummary.reviewCount	ratingValue	reviewed	score
2286	2286	2286	2286	2286
1557	1557	1557	1557	1557

```
plt.figure(figsize=(5,5))
plt.bar(x=df_for_year.index.astype(str),height=df_for_year['reviewed'])

plt.title('Yearwise change in Total Reviews')
plt.ylabel('Number of reviews')
plt.xlabel('Year')
plt.show()
```



Observations:

The number of people reviewing has gone down in 2021.

Possible reasons could be:

- 2020 was mostly lockdown in most parts of world. So people were more concerned with quality of masks in 2020 as compared to 2021 when lockdown restrictions were reduced and vaccinations were available for general masses. Hence usage of masks might have decreased in many parts of world
- Another reason could be that people might have cared less about quality of mask by 2021.

```
data.describe()
```

	product_price	productId	abuseCount	helpfulNo	helpfulYes	imagesCount
count	3843.000000	3843.000000	3843.000000	3843.000000	3843.000000	3843.000000
mean	16.015798	101262.270362	0.045537	0.052823	0.604215	0.080926
std	10.042985	1168.439166	0.246279	0.345371	4.020542	0.428620
min	2.950000	99829.000000	0.000000	0.000000	0.000000	0.000000
25%	8.860000	100237.000000	0.000000	0.000000	0.000000	0.000000
50%	17.720000	101691.000000	0.000000	0.000000	0.000000	0.000000
75%	22.440000	101955.000000	0.000000	0.000000	0.000000	0.000000
max	50.190000	104023.000000	4.000000	9.000000	139.000000	5.000000

SENTIMENT ANALYSIS:

To get the most liked product we have to do sentiment analysis on the reviews of users.

The method I am using here is called TextBlob Analysis.

It gives a polarity score between -1 and +1 according to the review given by the users.

Here value close to -1 is a negative comment

Value close to +1 is a positive comment

Value close to 0 is a neutral comment

```
import spacy
from textblob import TextBlob
```

```
data['TextBlob_Subjectivity'] = data['Final_Review'].apply(lambda x: TextBlob(x).sentiment.su
data['TextBlob_Polarity'] = data['Final_Review'].apply(lambda x: TextBlob(x).sentiment.polari
data[['TextBlob_Subjectivity','TextBlob_Polarity']]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  """Entry point for launching an IPython kernel.
```

```
data['TextBlob_Analysis'] = data['TextBlob_Polarity'].apply(lambda x: 'negative' if x<0 else
data
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarni
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
  """Entry point for launching an IPython kernel.
```

	product_price	productId	abuseCount	customerNickname	helpfulNo	helpfulYes
0	2.95	103205	0	iHerb Customer	1	22
1	2.95	103205	1	djagi	0	20
2	2.95	103205	0	TMC	0	8

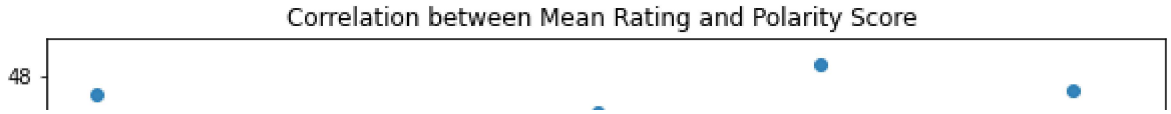
To get the product Id of most liked products by users

```
df = data.groupby(['productId']).mean()
df
```

	product_price	abuseCount	helpfulNo	helpfulYes	imagesCount	profileInfo.u
productId						
99829	22.44	0.046205	0.036304	0.521452	0.092409	
99830	22.44	0.077895	0.069474	1.006316	0.086316	
100099	49.61	0.170732	0.121951	1.731707	0.146341	
100234	44.31	0.020408	0.040816	0.448980	0.000000	
100237	17.72	0.041667	0.013258	0.219697	0.058712	
100541	26.57	0.096552	0.193103	0.613793	0.110345	
100837	5.61	0.035556	0.102222	0.360000	0.044444	
100839	4.49	0.080000	0.080000	0.480000	0.040000	
100840	3.93	0.044118	0.132353	0.750000	0.088235	
101029	36.54	0.162791	0.023256	0.209302	0.000000	
101691	17.72	0.000000	0.018692	0.233645	0.093458	
101692	17.72	0.023166	0.015444	0.166023	0.077220	
101774	2.95	0.083333	0.125000	0.880952	0.077381	
101955	8.86	0.010654	0.042618	0.627093	0.077626	
102455	2.95	0.029412	0.058824	1.661765	0.117647	

```
figure = plt.figure(figsize=(10,5))
plt.scatter(x=df.TextBlob_Polarity, y=df.ratingValue,alpha= 0.9, cmap='nipy_spectral')
plt.xticks(rotation = 45)
plt.title('Correlation between Mean Rating and Polarity Score ')
plt.ylabel('Mean Rating')
plt.xlabel('Polarity Score')
```

```
Text(0.5, 0, 'Polarity Score')
```



Observations from above graph:

- There is a very high positive correlation between Polarity score and Rating Value.

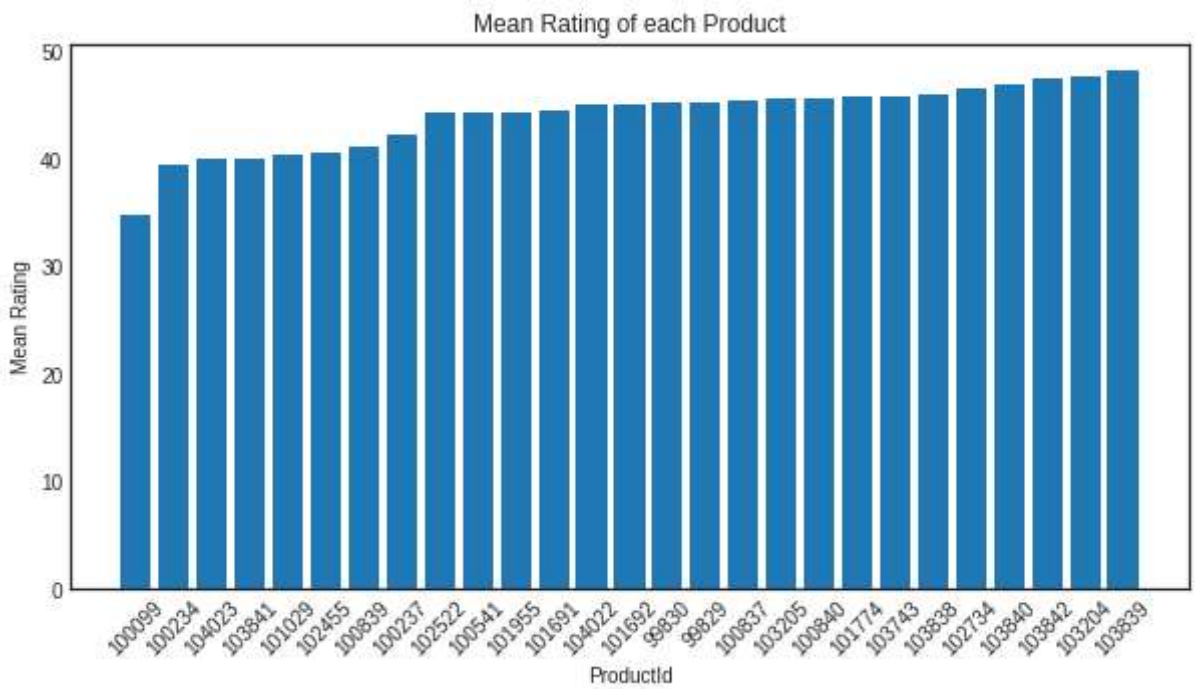
Conclusion:

- The product which has a high Polarity score as awell as a high Rating Value will be the product which is most liked by the users.

```
df_for_rating = df.sort_values(by = 'ratingValue')
```

```
plt.style.use('seaborn-white')
figure = plt.figure(figsize=(10,5))
plt.bar(x=df_for_rating.index.astype(str), height=df_for_rating.ratingValue)
plt.xticks(rotation = 45)
plt.title('Mean Rating of each Product')
plt.ylabel('Mean Rating')
plt.xlabel('ProductId')
```

```
Text(0.5, 0, 'ProductId')
```



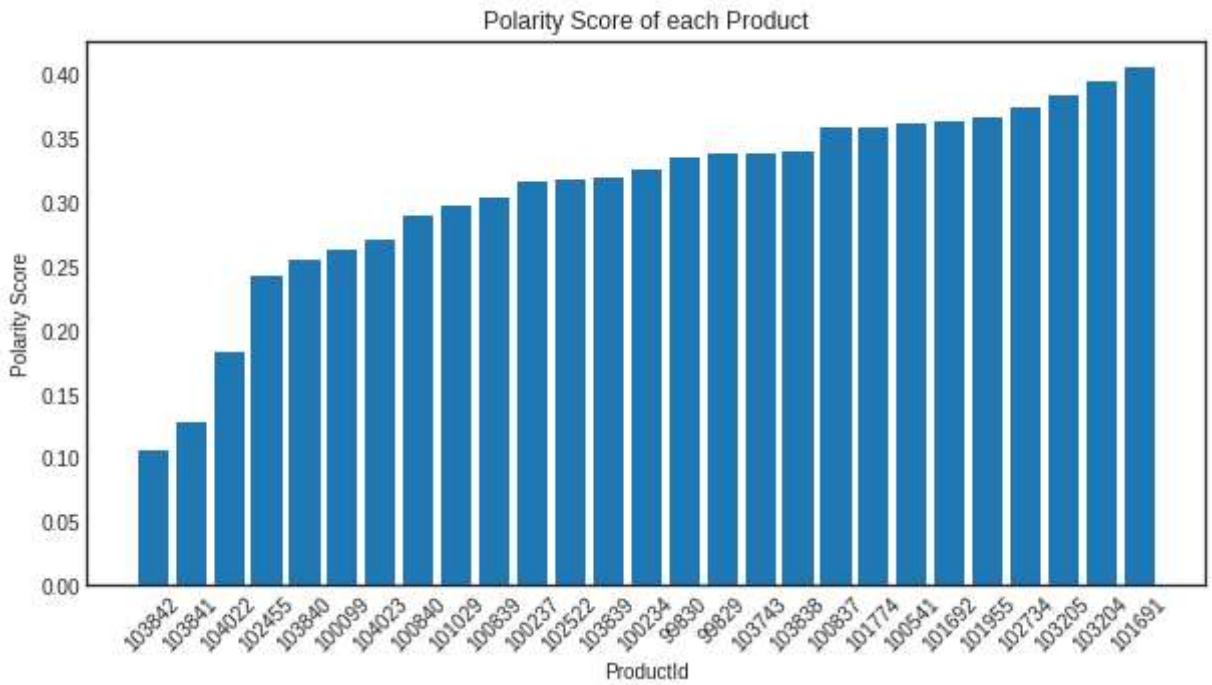
Top 5 products according to rating value

- 103839
- 103204
- 103842
- 103840
- 102734

```
df_for_polarity = df.sort_values(by = 'TextBlob_Polarity')
```

```
figure = plt.figure(figsize=(10,5))
plt.bar(x=df_for_polarity.index.astype(str), height=df_for_polarity.TextBlob_Polarity)
plt.xticks(rotation = 45)
plt.title('Polarity Score of each Product')
plt.ylabel('Polarity Score')
plt.xlabel('ProductId')
```

```
Text(0.5, 0, 'ProductId')
```



Top 5 products according to Mean of Polarity Score:

- 101691
- 103204
- 103205
- 102734
- 101955

In accordance with the above conclusion:

"Users like the product which has graeter average Rating Value as well as a greater mean of Polarity score".

Products liked most by customers are:

103204 -- Hwipure, Disposable KF94 (N95 / KN95/ FFP2) Mask, 25 Masks

102734 -- Zidian, Disposable Protective Mask, 50 Pack

To Check what is that the customers are liking most about these products

```
df_for_103204 = data['Final_Review'][(data['productId'] == 103204)  & (data['TextBlob_Analysi
df_for_102734 = data['Final_Review'][(data['productId'] == 102734)  & (data['TextBlob_Analysi
df_most_liked = df_for_103204.append(df_for_102734)
```

```
df_most_liked
```

```
     1624    I love them, you have to take into account tha...
     1625    These are good masks. They're packaged nicely ...
     1626                                     Very comfortable
     1627    Great! So iherb brought these in so we have ac...
     1628    It has more protection than the cloth masks.  ...
                                   ...
     1326                                      Wonderful masks
     1327    At first, the price pleased me, but the qualit...
     1328       good, a pack of 50 bibs at an unbeatable price
     1329    It's very comfortable and more stylish than th...
     1330    If you have any doubts to buy it. Don’t!! Actu...
     Name: Final_Review, Length: 352, dtype: object
```

```
from wordcloud import WordCloud
from wordcloud import ImageColorGenerator
from wordcloud import STOPWORDS
import matplotlib.pyplot as plt
text = " ".join(i for i in df_most_liked)
stopwords = set(STOPWORDS)
stopwords.add('mask')
stopwords.add('masks')
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Observations from word cloud:

The most recurring words are:

Comfortable , good , fit , great , face , breathe , quality , recommend , black (Assuming black coloured masks are more popular than other colours)

Things liked by customers about the most popular masks:

- The masks are Comfortable
- The masks fit well
- They fit well on the face
- They are comfortable to breathe

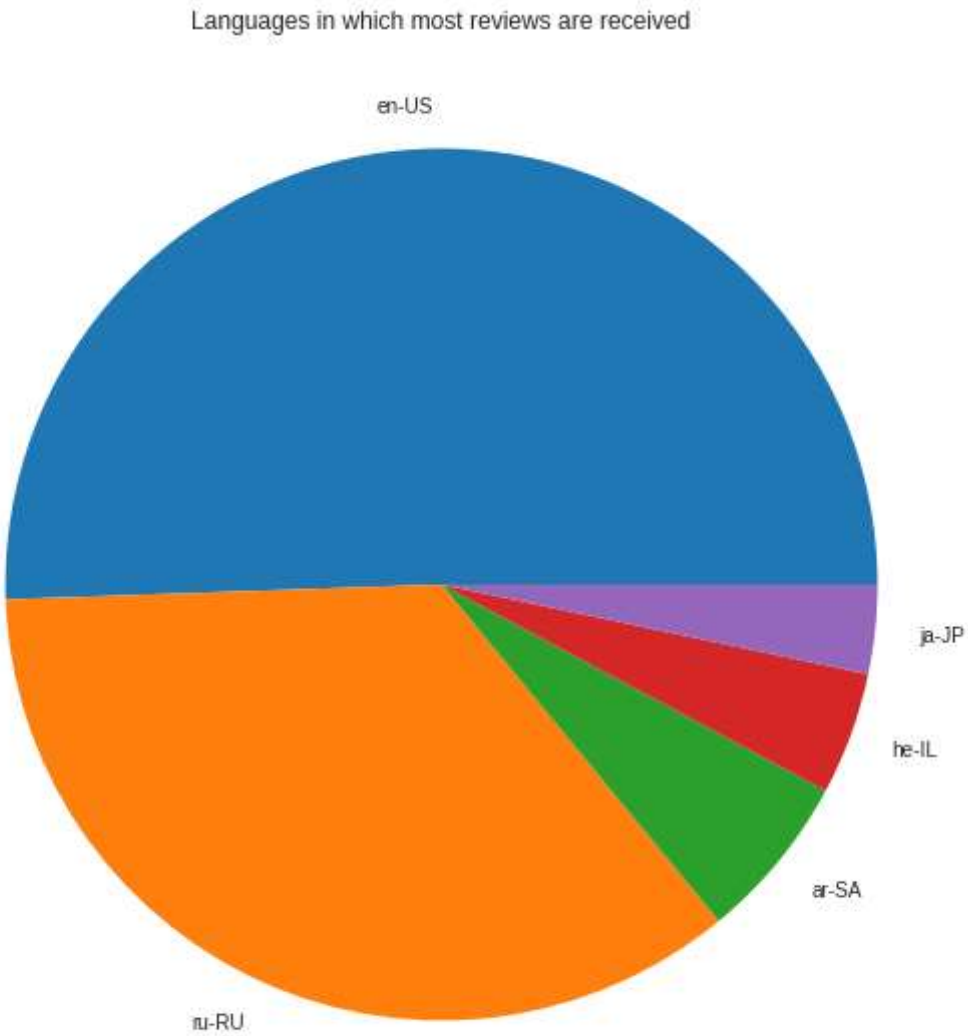
```
df_for_customer = review_data.groupby(['languageCode.1']).count()
df_for_customer = df_for_customer.sort_values(by=['reviewed'], ascending = False).head(5)
df_for_customer
```

	abuseCount	customerNickname	helpfulNo	helpfulYes	id	imagesCount
languageCode.1						
en-US	1855	1855	1855	1855	1855	1855
ru-RU	1300	1300	1300	1300	1300	1300
ar-SA	227	227	227	227	227	227

```
figure = plt.figure(figsize=(10,10))
plt.pie(labels = df_for_customer.index.astype(str), x = df_for_customer.reviewed)

plt.title('Languages in which most reviews are received')
```

```
Text(0.5, 1.0, 'Languages in which most reviews are received')
```



Conclusion:

The customers who are reviewing most are from either of the following countries:

- USA
- Russia
- South Africa
- Japan
- Israel

Double-click (or enter) to edit

✓ 0s completed at 8:44 PM