EDA Assignment 3

We are provided with two datasets :
1. A list of facemasks on online retailer iHerb with data around the product and its price, number of ratings
2. A list of consumer reviews of these face masks

Questions to answer:
1. Which are the most popular face masks out there?
2. What do consumers like about them? Why?
3. What different profiles of consumers buy masks?

In [1]:

```
# Import all libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import pycountry
from wordcloud import WordCloud, STOPWORDS
```

Lets start with exploratory analysis of the data first

Reviews

In [2]:

```
reviews = pd.read_excel('reviews.xlsx')
reviews.head()
```

Out[2]:

	abuseCount	customerNickname	helpfulNo	helpfulYes	id	imagesCount	languageCode	postedDate	pro
0	0	iHerb Customer	0	6	05c2b17e-c28d-4792-930d-27e787d8d4ad	1	en-US	2021-01-27T09:04:10.569Z	1
1	0	iHerb Customer	0	0	80e44af8-2edf-4b81-a80a-7e7888d03cc0	0	ru-RU	2021-02-07T00:56:39.055Z	1
2	0	iHerb Customer	0	0	9a76e047-21e4-4da3-8b50-9d2396519b6b	0	en-US	2021-02-06T21:40:02.886Z	1
3	0	Innalgorevna	0	0	2890ac54-8707-418e-be3e-8d46231e3672	0	ru-RU	2021-02-05T16:29:28.906Z	1
4	0	iHerb Customer	0	0	9db33354-0457-4efa-bc9c-b5f7ee0eff31	0	ru-RU	2021-02-05T09:43:42.367Z	1

In [3]:

```
reviews.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                            3848 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                1993 non-null   object
 18  translation.reviewTitle               1994 non-null   object
dtypes: bool(1), float64(2), int64(7), object(9)
memory usage: 545.1+ KB
```

Products

In [4]:

```
products = pd.read_excel('products.xlsx')
products.head()
```

Out[4]:

	product_id	product_name	product_price	price_currency	product_availability	product_url	
0	**103205**	**Hwipure, Disposable KF94 (N95 / KN95/ FFP2) ...**	**2.95**	**AUD**	**http://schema.org/InStock**	**https://au.iherb.com/pr/Hwipure-Disposable-KF9...**	**http**
1	**101774**	**HIGUARD, Disposable KF94 (N95 / KN95/ FFP2) ...**	**2.95**	**AUD**	**http://schema.org/InStock**	**https://au.iherb.com/pr/HIGUARD-Disposable-KF9...**	**http**
2	**101955**	**SunJoy, KN95, Professional Protective Disposab...**	**8.86**	**AUD**	**http://schema.org/InStock**	**https://au.iherb.com/pr/SunJoy-KN95-Profession...**	**http**
3	**103838**	**Lozperi, Copper Mask, Adult, Black, 1 Mask**	**6.85**	**AUD**	**http://schema.org/InStock**	**https://au.iherb.com/pr/Lozperi-Copper-Mask-Ad...**	**http**
4	**102734**	**Zidian, Disposable Protective Mask, 50 Pack**	**15.35**	**AUD**	**http://schema.org/InStock**	**https://au.iherb.com/pr/Zidian-Disposable-Prot...**	**http**

In [5]:

```
products.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
```

Visualisation and Analysis

In [6]:

```
products['product_id'].nunique(), reviews['productId'].nunique()
```

Out[6]:

```
(27, 27)
```

Product IDs seem to be common and can the 2 tables be combined using an outer joing over the reviews table

In [7]:

```
products['productId'] = products['product_id']
products.drop('product_id', axis=1, inplace=True)
combined_prod_id = reviews.merge(products[['productId','product_name','product_price']],
on='productId', how='left')
combined_prod_id.head()
```

Out[7]:

	abuseCount	customerNickname	helpfulNo	helpfulYes	id	imagesCount	languageCode	postedDate	pro
0	0	iHerb Customer	0	6	05c2b17e-c28d-4792-930d-27e787d8d4ad	1	en-US	2021-01-27T09:04:10.569Z	1
1	0	iHerb Customer	0	0	80e44af8-2edf-4b81-a80a-7e7888d03cc0	0	ru-RU	2021-02-07T00:56:39.055Z	1
2	0	iHerb Customer	0	0	9a76e047-21e4-4da3-8b50-9d2396519b6b	0	en-US	2021-02-06T21:40:02.886Z	1
3	0	Innalgorevna	0	0	2890ac54-8707-418e-be3e-8d46231e3672	0	ru-RU	2021-02-05T16:29:28.906Z	1
4	0	iHerb Customer	0	0	9db33354-0457-4efa-bc9c-b5f7ee0eff31	0	ru-RU	2021-02-05T09:43:42.367Z	1

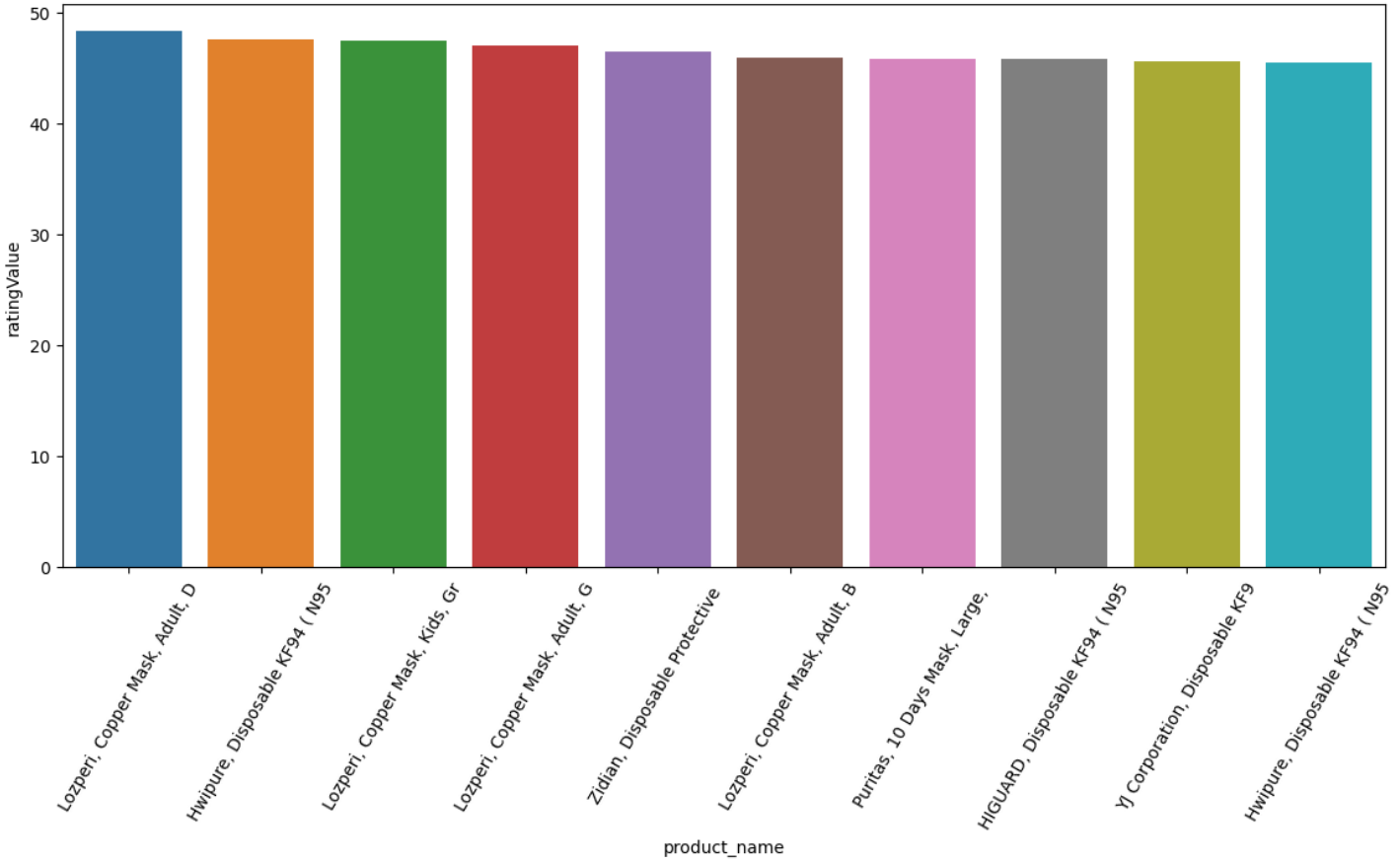
5 rows × 21 columns

Q1. What are the most popular facemasks?

By Raw Ratings

In [8]:

```
product_ratings_avg = combined_prod_id[['product_name','ratingValue']].groupby(['product_
name']).mean().sort_values(by='ratingValue', ascending=False)
plt.figure(figsize=(14,6))
ax = sns.barplot(y=product_ratings_avg.head(10)['ratingValue'], x=product_ratings_avg.in
dex[:10])
ax.set_xticklabels(product_ratings_avg.index[:10].map(lambda x:x[:30]), rotation=60)
pass
```

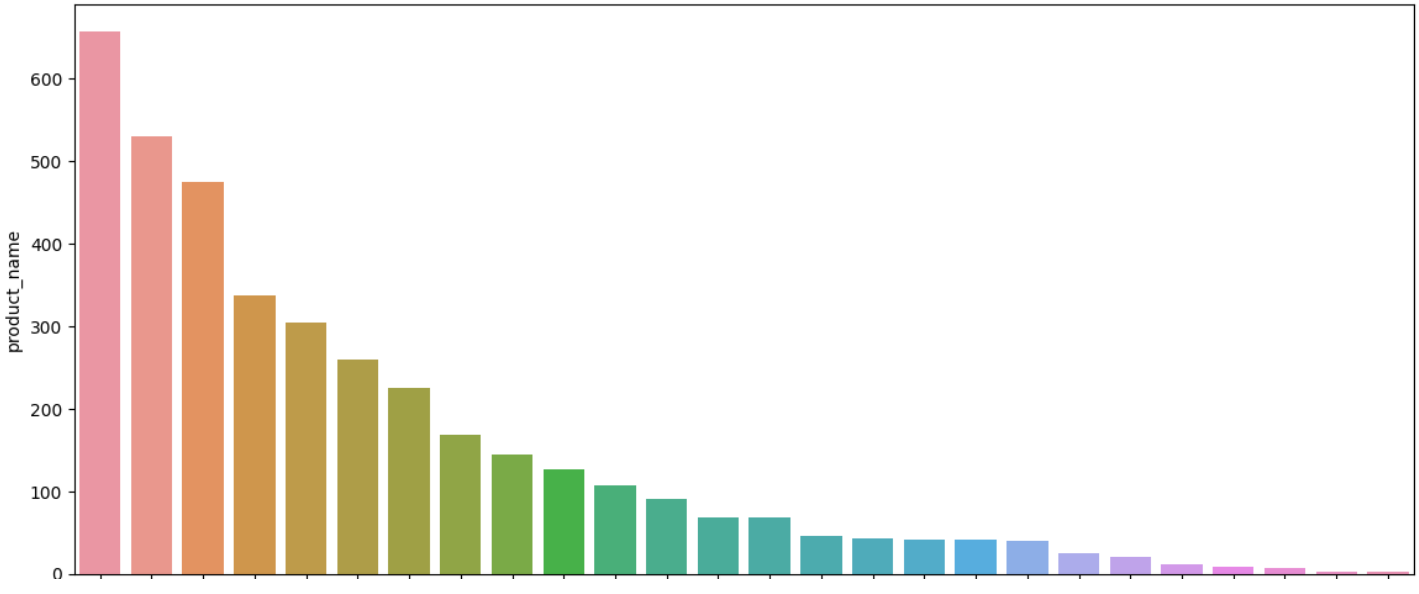


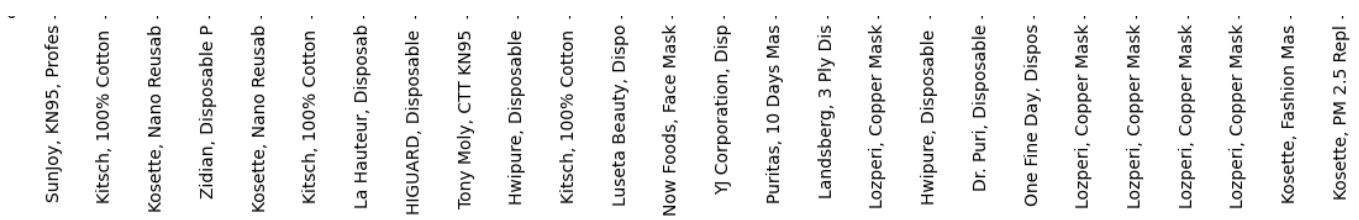
Lozperi Copper Mask seems to be the highest rated but this is not sufficient information. We also need to look at the sales numbers as well as different categories.

Sales numbers for each product

In [9]:

```
plt.figure(figsize=(14,6))
ax = sns.barplot(y=combined_prod_id['product_name'].value_counts(), x= combined_prod_id[
'product_name'].value_counts().index)
ax.set_xticklabels(pd.Series(combined_prod_id['product_name'].value_counts().index).map(
lambda x:x[:20]), rotation=90)
pass
```





Ratings for products with more than 100 units sold

In [10]:

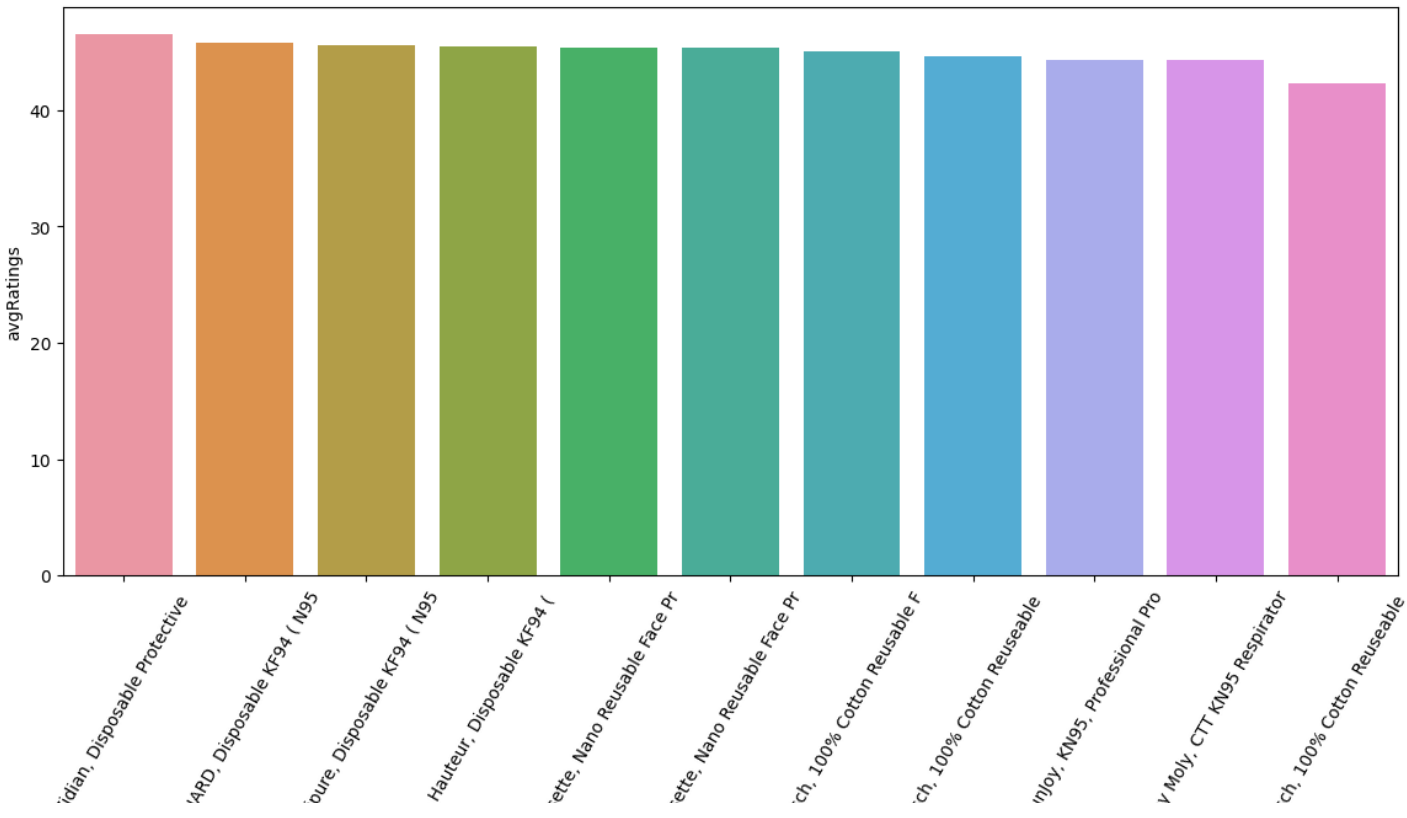
```
product_counts = combined_prod_id[['product_name','ratingValue']].groupby(by='product_name').count()
product_ratings_avg_100_plus = product_counts.merge(product_ratings_avg, on='product_name')
product_ratings_avg_100_plus.rename({'ratingValue_x':'unitCount', 'ratingValue_y':'avgRatings'}, inplace=True, axis=1)
product_ratings_avg_100_plus = product_ratings_avg_100_plus[product_ratings_avg_100_plus['unitCount'] > 100].sort_values(by='avgRatings', ascending=False)
product_ratings_avg_100_plus.head()
```

Out[10]:

product_name	unitCount	avgRatings
Zidian, Disposable Protective Mask, 50 Pack	337	46.498516
HIGUARD, Disposable KF94 (N95 / KN95/ FFP2) Mask, 1 Mask	168	45.833333
Hwipure, Disposable KF94 (N95 / KN95/ FFP2) Mask, 1 Mask	126	45.555556
La Hauteur, Disposable KF94 (N95 / KN95/ FFP2) Mask, 1 Mask	225	45.511111
Kosette, Nano Reusable Face Protection Mask, Large, 1 Mask	304	45.361842

In [11]:

```
plt.figure(figsize=(14,6))
ax = sns.barplot(y=product_ratings_avg_100_plus['avgRatings'], x=product_ratings_avg_100_plus.index)
ax.set_xticklabels(product_ratings_avg_100_plus.index.map(lambda x:x[:30]), rotation=60)
pass
```



product_name

Zidian Disposable mask appears to be the highest rated with more than 100 units sold. Brands like *Zidian, Kossette, Kitsch and Sunjoy* are popular among people.

Let's segregate the data *brandwise* and using *mask type* and look at popularity from that aspect.

In [12]:

```
combined_prod_id['BrandName'] = combined_prod_id['product_name'].apply(lambda x:x.split(
','))[0])
combined_prod_id['BrandName']
```

Out[12]:

```
0              Lozperi
1              Lozperi
2              Lozperi
3              Lozperi
4              Lozperi
             ...
3844     Luseta Beauty
3845     Luseta Beauty
3846     Luseta Beauty
3847     Luseta Beauty
3848     Luseta Beauty
Name: BrandName, Length: 3849, dtype: object
```

In [13]:

```
def getCategory(name):
    if 'Reus' in name or 'reus' in name:
        return 'Reuseable'
    elif 'Dispos' in name or 'Dispos' in name:
        return 'Disposable'
    else:
        return 'Other'
combined_prod_id['Category'] = combined_prod_id['product_name'].apply(getCategory)
combined_prod_id['Category'].head()
```

Out[13]:

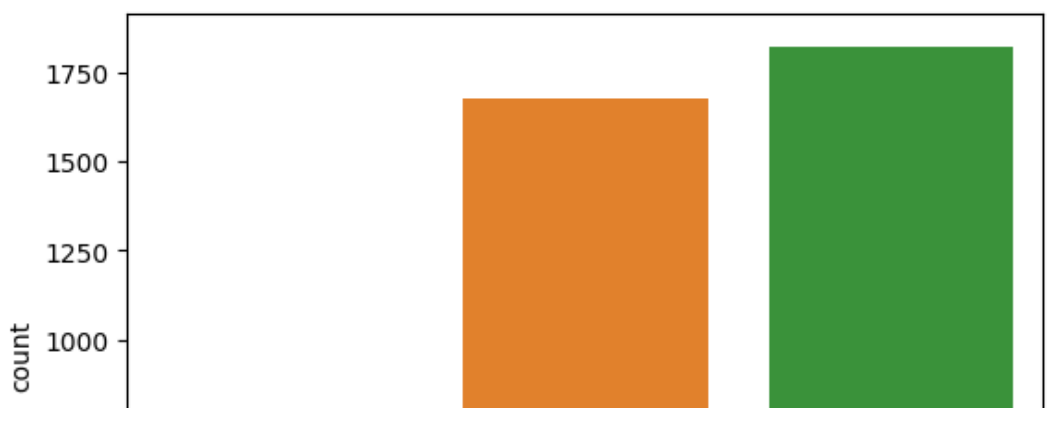
```
0    Other
1    Other
2    Other
3    Other
4    Other
Name: Category, dtype: object
```

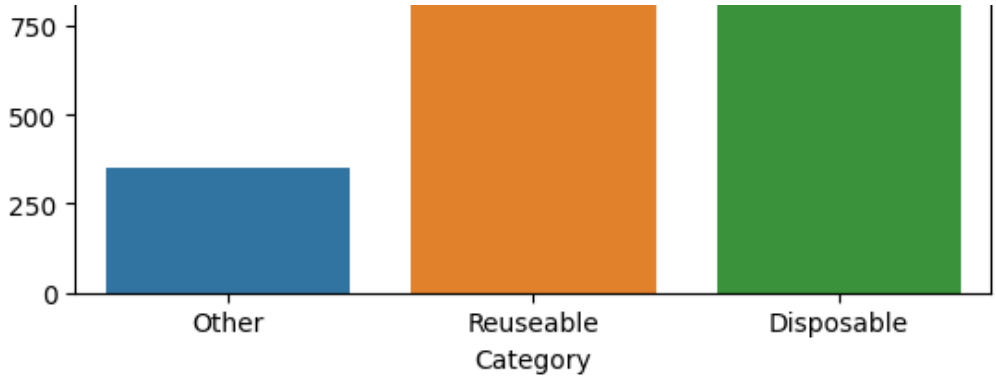
In [14]:

```
sns.countplot(data=combined_prod_id, x='Category')
```

Out[14]:

```
<AxesSubplot:xlabel='Category', ylabel='count'>
```





Disposable **masks are to be the most popular followed by** ***reuseable*** **masks.**

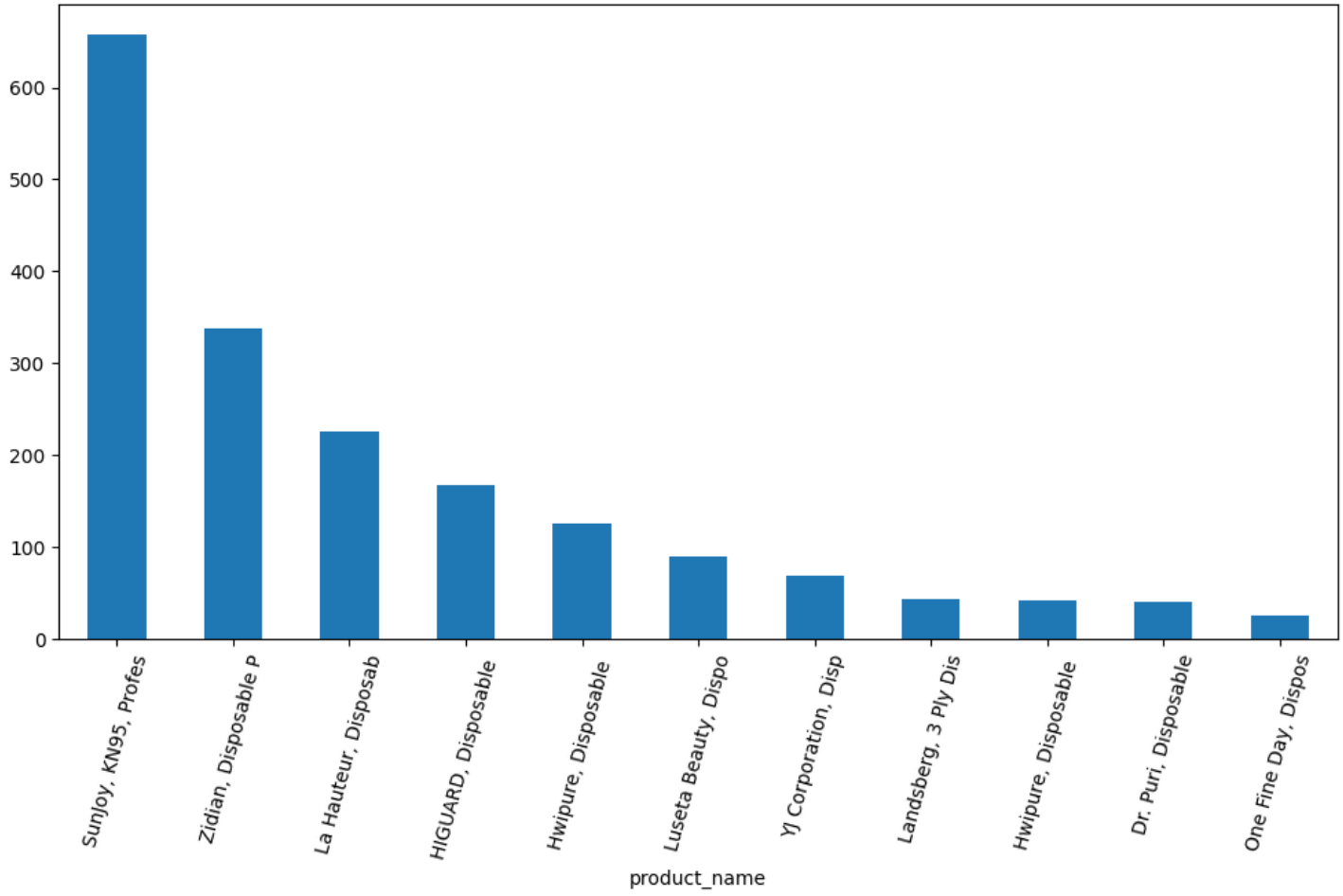
Let's look at the most popular brands categorywise.

In [15]:

```
category_group = combined_prod_id.groupby('Category')['product_name'].value_counts()
```

In [16]:

```
plt.figure(figsize=(12,6))
ax = category_group['Disposable'].plot(kind='bar')
ax.set_xticklabels(pd.Series(category_group['Disposable'].index).map(lambda x:x[:20]), r
otation=75)
pass
```

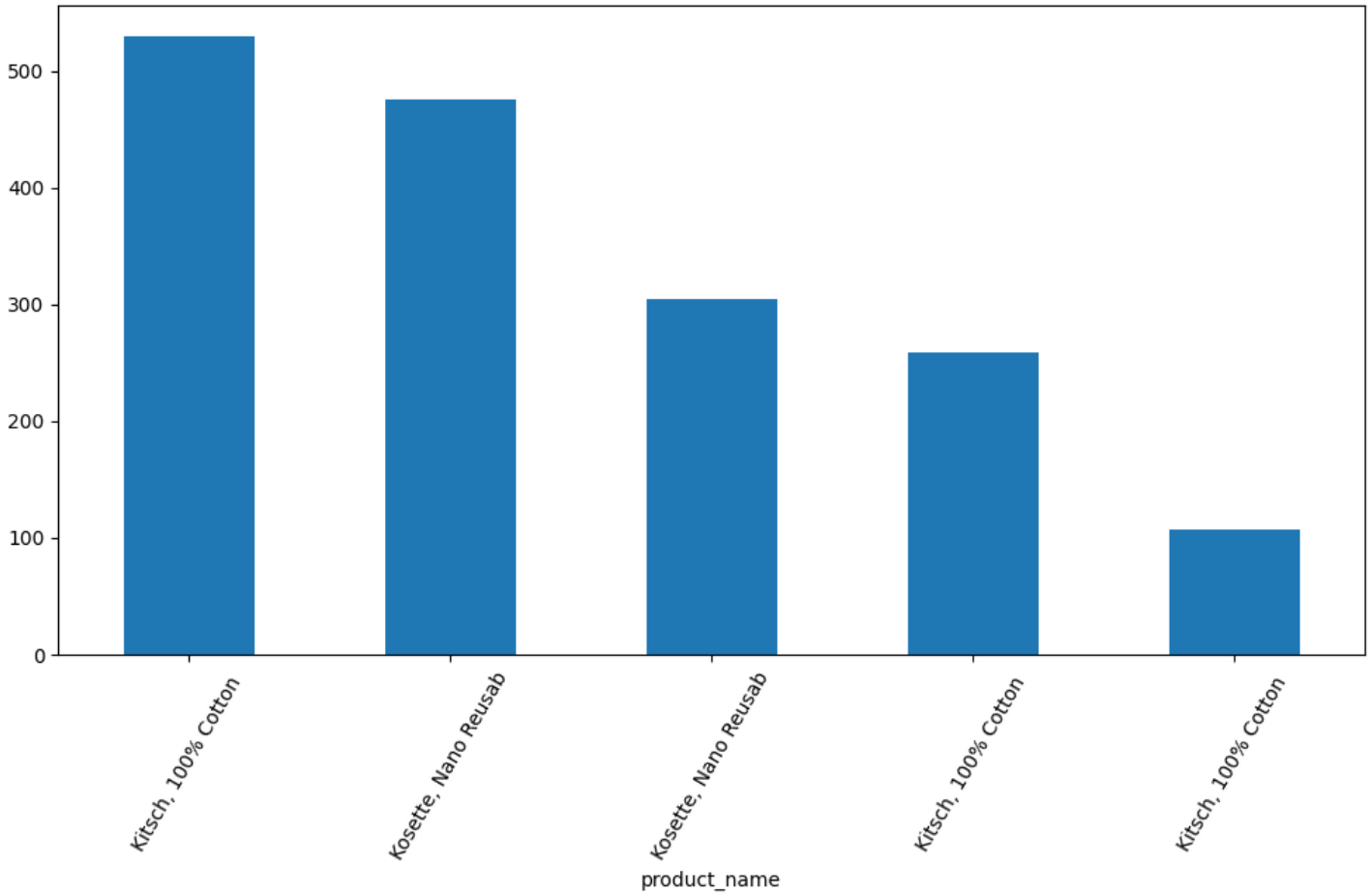


Sunjoy KN95 **masks are to the most popular in the** ***disposable*** **category.**

In [17]:

```
plt.figure(figsize=(12,6))
ax = category_group['Reuseable'].plot(kind='bar')
ax.set_xticklabels(pd.Series(category_group['Reuseable'].index).map(lambda x:x[:20]), ro
tation=60)
```

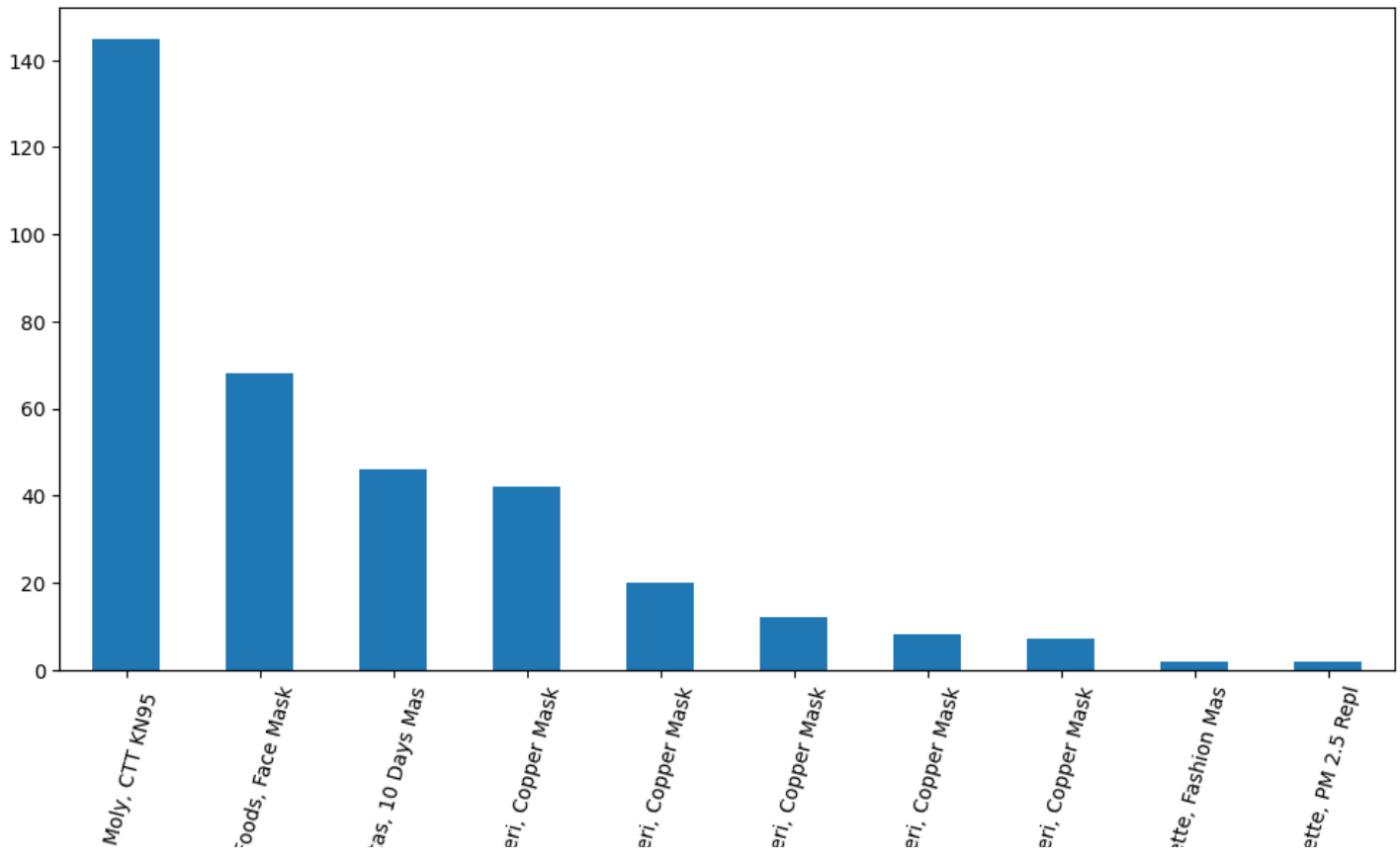
```
pass
```

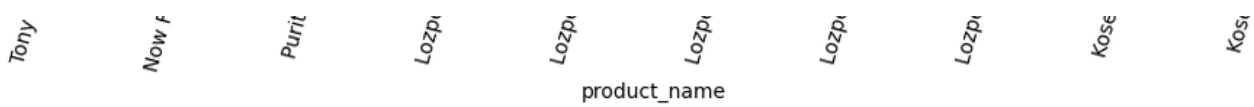


Kitsch Cotton **masks are to the most popular in the** ***reuseable*** **category.**

In [18]:

```
plt.figure(figsize=(12,6))
ax = category_group['Other'].plot(kind='bar')
ax.set_xticklabels(pd.Series(category_group['Other'].index).map(lambda x:x[:20]), rotation=75)
pass
```





Tony Moly **masks are popular in the** ***others*** **category.**

Q2. What do consumers like about them? Why?

Word cloud **is one of the most efficient and simple ways of identifying the keywords in a corpus of text. Lets generate a wordcloud for english translated reviews for each category of masks segregating only those reviews that consumers found helpful.**

In [19]:

```
combined_prod_id[['reviewText','translation.reviewText','languageCode.1']].fillna('', in
place=True)
combined_prod_id['english_reviews'] = combined_prod_id[['reviewText','translation.review
Text','languageCode.1']].apply(lambda x:x[0] if 'en' in x[2] else x[1], axis=1)
combined_prod_id['english_reviews'].head()
```

```
C:\Users\kishl\AppData\Local\Temp\ipykernel_25612\301186810.py:1: 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_g
uide/indexing.html#returning-a-view-versus-a-copy
  combined_prod_id[['reviewText','translation.reviewText','languageCode.1']].fillna('', i
nplace=True)
```

Out[19]:

```
0    The mask quality and the color is good. It fit...
1      The grandson really liked it. Comfortable mask.
2                 Easy to put on & comfortable to wear.
3    A thin mask that is pleasant to the body. I li...
4    Great mask! It suited me perfectly. There is a...
Name: english_reviews, dtype: object
```

In [20]:

```
def generate_wordcloud(frame):
    comment_words = ''
    stopwords = set(STOPWORDS)
    stopwords.update(['face', 'mask','masks','ð']) # Remove unnecessary words that creat
e noise

    # iterate through the csv file
    for val in frame['english_reviews']:

        # typecaste each val to string
        val = str(val)

        # split the value
        tokens = val.split()

        # Converts each token into lowercase
        for i in range(len(tokens)):
            tokens[i] = tokens[i].lower()

        comment_words += " ".join(tokens)+" "

    wordcloud = WordCloud(width = 800, height = 800,
                    background_color ='white',
                    stopwords = stopwords,
                    collocation_threshold = 3,
                    min_font_size = 10).generate(comment_words)

    # plot the WordCloud image
    plt.figure(figsize = (8, 8), facecolor = None)
```

```
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.tight_layout(pad = 0)

    plt.show()
```

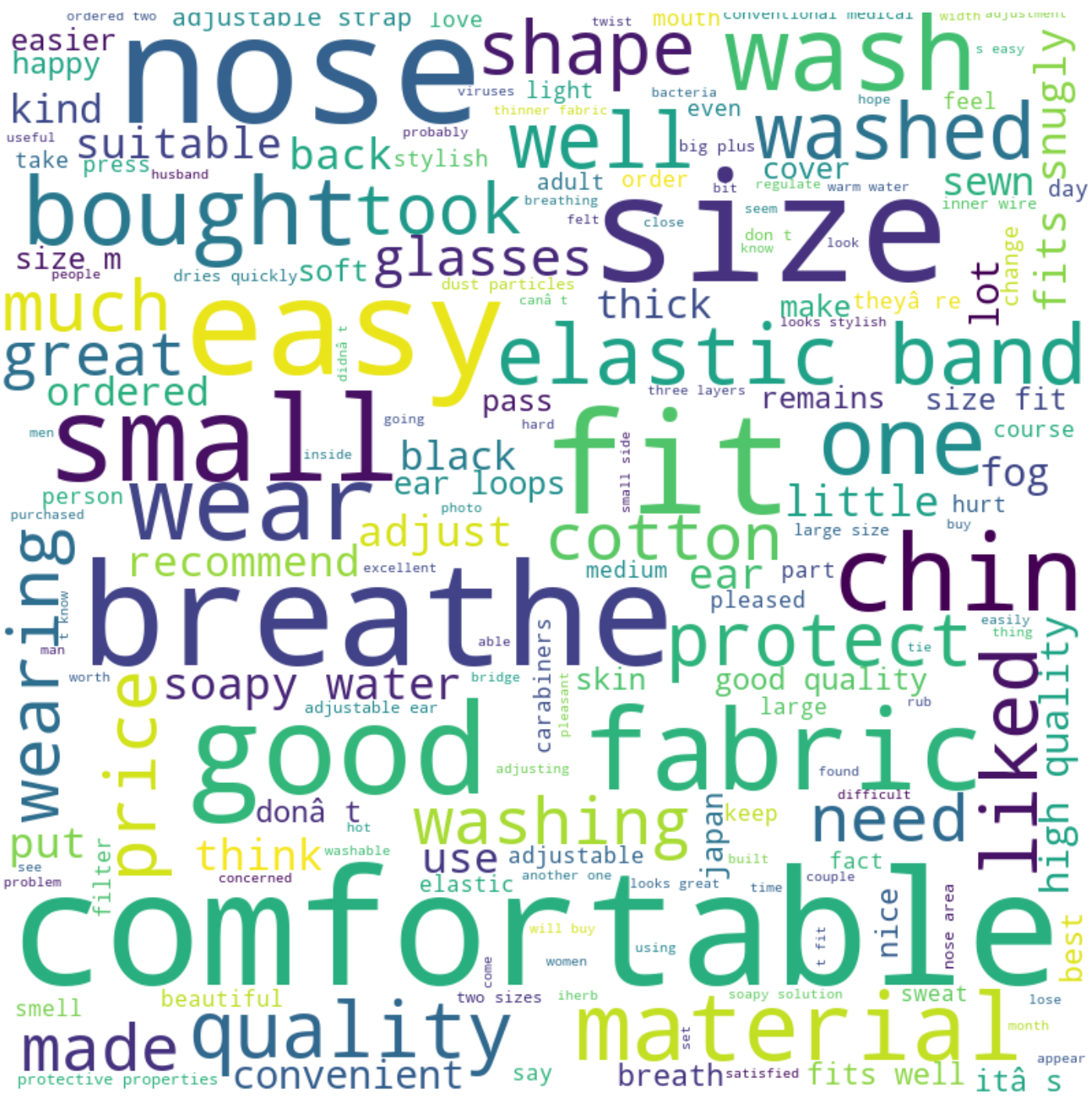
In [21]:

```
# Consider only those reviews which were helpful to consumers by using (combined_prod_id[
'helpfulYes'] > combined_prod_id['helpfulNo'])
disposables = combined_prod_id[(combined_prod_id['Category']=='Disposable')  & (combined
_prod_id['helpfulYes'] > combined_prod_id['helpfulNo'])]
generate_wordcloud(disposables)
```

While most of the reviews seem to be praising the comfort and quality of the products, some other things that stand out in this category are *filtration efficiency, cheaper, convenient and better fit.*

In [22]:

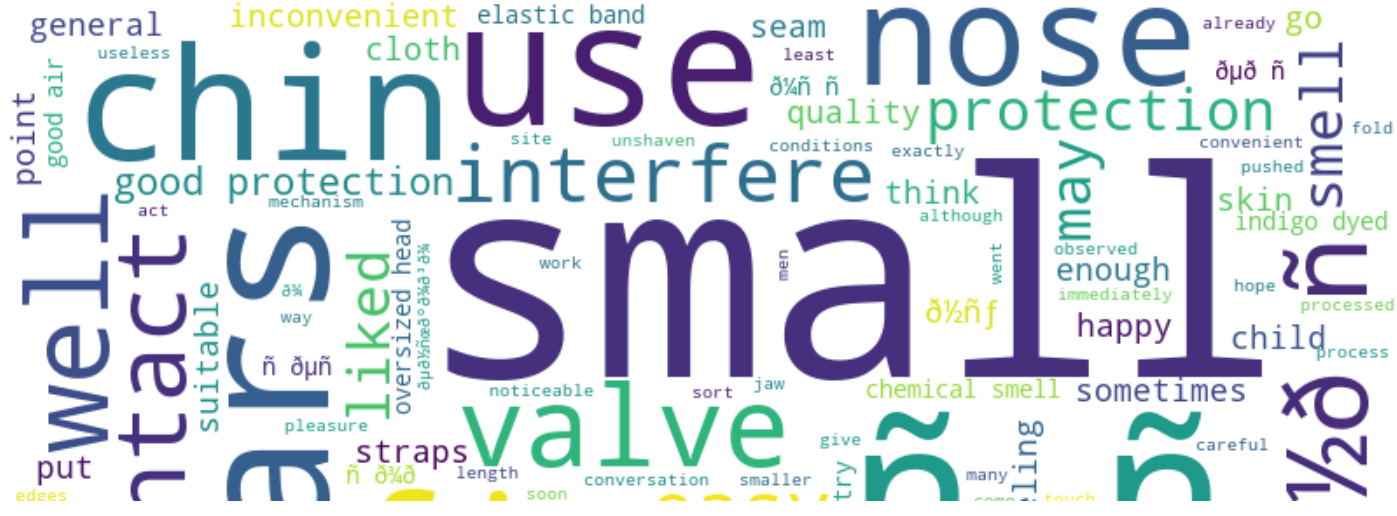
```
reuseables = combined_prod_id[(combined_prod_id['Category']=='Reuseable')  & (combined_p
rod_id['helpfulYes'] > combined_prod_id['helpfulNo'])]
generate_wordcloud(reuseables)
```

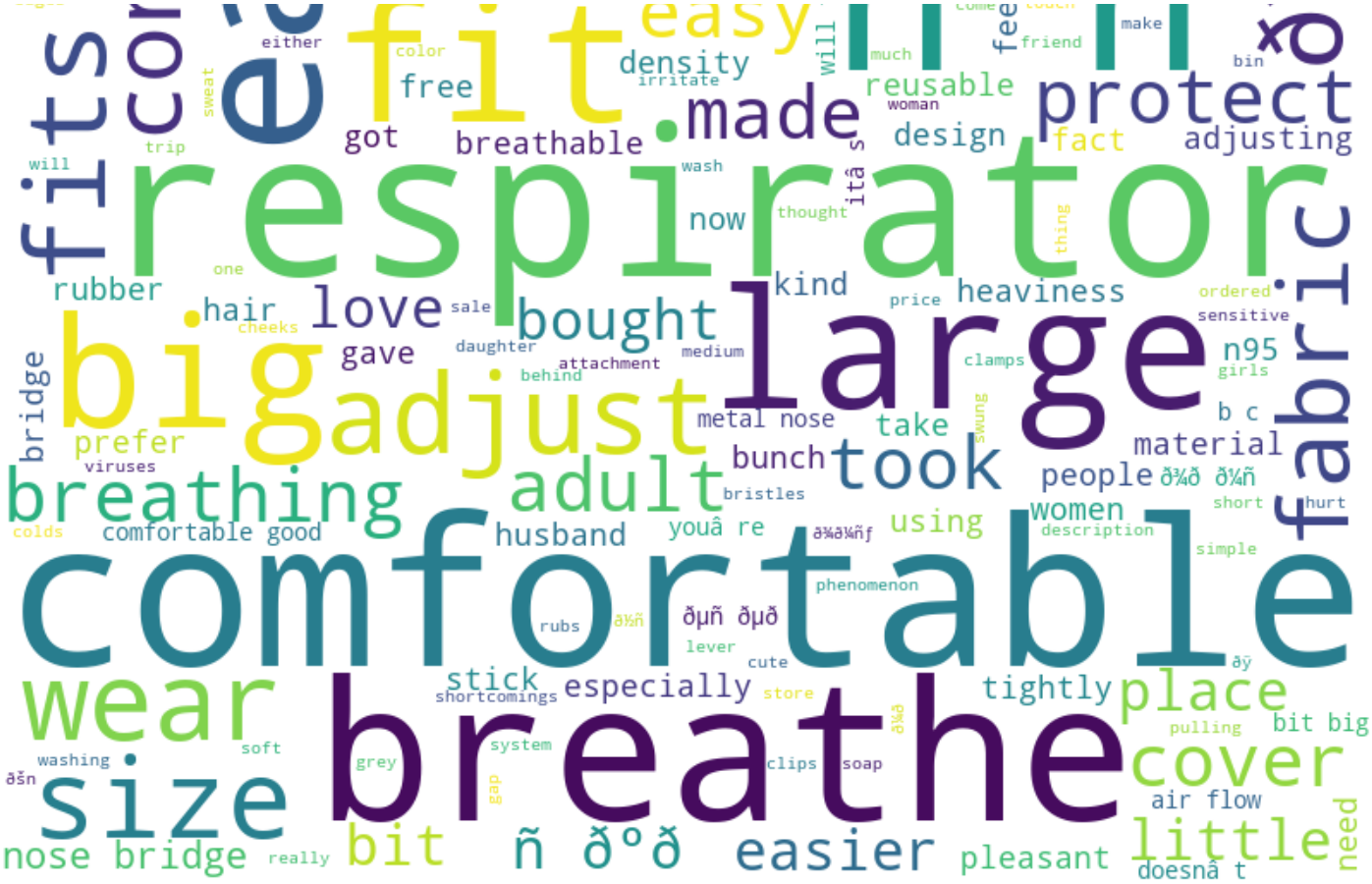


The things that stand out in the reuseables category are *looks great, fits well, washed(washable) and adjustable.*

In [23]:

```
others = combined_prod_id[(combined_prod_id['Category']=='Other')  & (combined_prod_id['
helpfulYes'] > combined_prod_id['helpfulNo'])]
generate_wordcloud(others)
```





Important keywords in the *Others* category are *respirator, adjustable, and valve.*

Q3. What different profiles of consumers buy masks?

Language data **can be used to segregate users based on their country.** ***Plotly Choropleth maps*** **can be used for this visualisation.**

In [24]:

```
combined_prod_id['alpha_3_code'] = combined_prod_id['languageCode.1'].apply(lambda x:pyc
ountry.countries.get(alpha_2=x[-2:]).alpha_3)
```

In [25]:

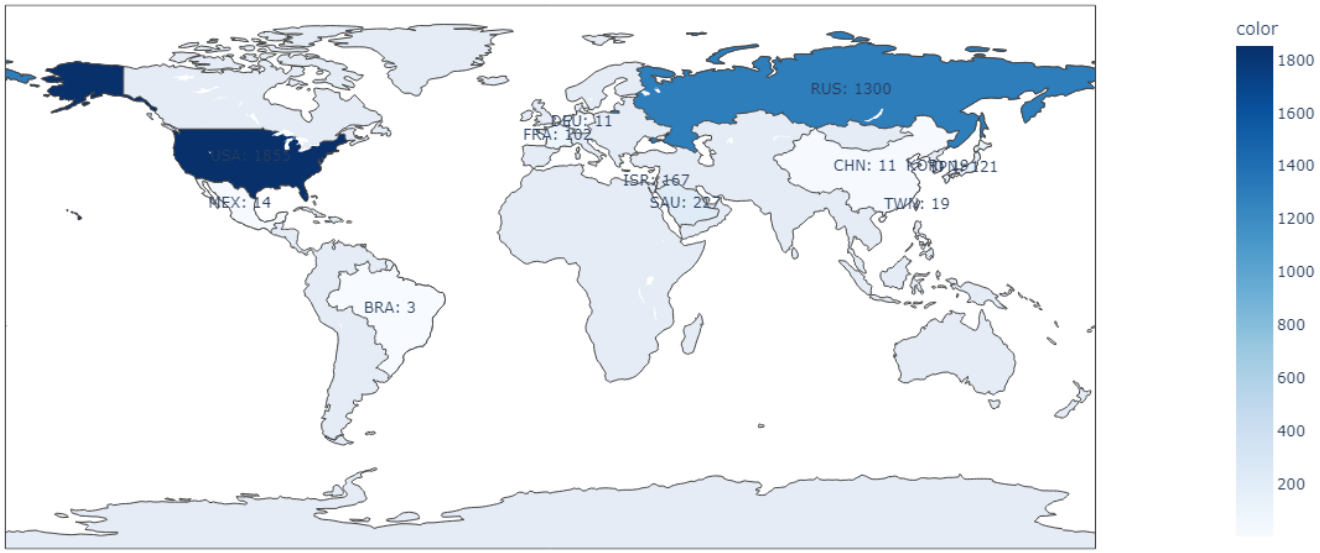
```
plt.figure(figsize=(14,8))
fig = px.choropleth(locations=combined_prod_id['alpha_3_code'].value_counts().index, loc
ationmode="ISO-3", color=combined_prod_id['alpha_3_code'].value_counts(), color_continuou
s_scale=px.colors.sequential.Blues)

fig.add_scattergeo(
    locations=combined_prod_id['alpha_3_code'].value_counts().index,
    locationmode="ISO-3",
    text=pd.Series(combined_prod_id['alpha_3_code'].value_counts().index) +': '+ pd.Seri
es(list(combined_prod_id['alpha_3_code'].value_counts().astype('str'))),
    mode='text',
    )

fig.update_layout(margin=dict(l=5, r=5, t=5, b=5))
fig.show()
```

```
<Figure size 1400x800 with 0 Axes>
```

In [26]:

```
# Had to display it using image since plotly graph is not converted to pdf
from IPython.display import Image
Image(filename='newplot.png')
```

Out[26]:



Russia and USA **seem to be the largest buyers of facemasks according to iHerb data. However, we don't have information about highly populous countries like India and China.**