Project 13-- SEYI OGUNMODEDE

Instructor Help: Dr. Ward

- Help with figuring out how to write a function.

Question 1

In [1]:
```
import pandas as pd

finefood = pd.read_csv("/anvil/projects/tdm/data/amazon/amazon_fine_food_reviews.csv")
```

In [2]:
```
finefood.head()
```

Out[2]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0

In [3]:
```
# from the data,HelpfulnessNumerator and HelpfulnessDenominator seems to have values e
# or not always equal and HelpfulnessNumerator is never larger than HelpfulnessDenomir
# HelpfulnessNumerator seems to be number of people who rated the review to be helpful
# HelpfulnessDenominator seems to be total number of people who rated the review wheth
```

In [4]:
```
# what is the user id of review number 23789?
# You can use iloc to locate the location
```

In [5]:
```
finefood.iloc[23789]
```

Out[5]:
```
Id                                                              23790
ProductId                                                  B0013NUGDE
UserId                                                 ADAASOYZ1TOMW
ProfileName                                SammySosa21 "sammysosa21"
HelpfulnessNumerator                                              33
HelpfulnessDenominator                                            35
Score                                                              5
Time                                                      1211414400
Summary                                                Fantastic chips!!!
Text                    I want to start out by saying that i thought a...
Name: 23789, dtype: object
```

In [6]:
```
finefood.iloc[23788]
```

Out[6]:
```
Id                                                              23789
ProductId                                                  B0013NUGDE
UserId                                                 A1HBC0NBQJHT7X
ProfileName                                                 Bookphile
HelpfulnessNumerator                                               73
HelpfulnessDenominator                                             77
Score                                                              5
Time                                                      1265760000
Summary                 Not exactly like traditional potato chips but ...
Text                    One of my biggest frustrations with doing Weig...
Name: 23788, dtype: object
```

In [7]:
```
# How many duplicate ProfileName values are there?
```

In [8]:
```
# checking with first five for duplication

finefood.ProfileName.duplicated().head()
# They are not.
```

Out[8]:
```
0    False
1    False
2    False
3    False
4    False
Name: ProfileName, dtype: bool
```

In [9]:
```
# remember that when we sum False and True values, False become zero True becomes 1,
# so we could just sum() to get the number of true values

finefood.ProfileName.duplicated().sum()
```

Out[9]:
```
350058
```

Immediately we see two columns that might be interesting: HelpfulnessNumerator and HelpfulnessDenominator. What do you think those mean, and what would they (potentially) be used for?

What is the user id of review number 23789?

How many duplicate ProfileName values are there? (I am not asking for which values are duplicated but just the total number of duplicated ProfileName values; it is helpful to explain your answer for this one.)

Question 2

In [10]:
```
#for a histogram
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import plotly.express as px

fig = px.histogram(finefood, x="Score")
fig.update_traces(marker_color="turquoise",marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5)
fig.update_layout(title_text='product rating')
fig.show()
```

In [11]:
```
#for a histogram
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import plotly.express as px

fig = px.histogram(finefood, x="ProductId")
fig.update_traces(marker_color="turquoise",marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5)
fig.update_layout(title_text='product Info' )
fig.show()
```

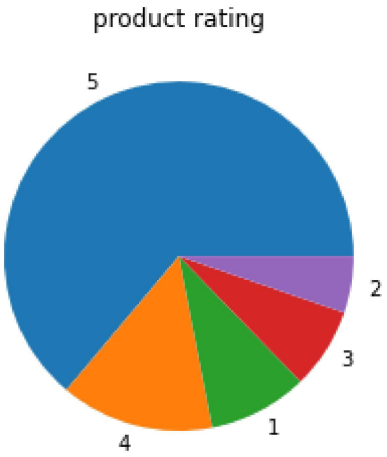
In [12]:
```
#for a histogram
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import plotly.express as px

fig = px.histogram(finefood, x="Time")
fig.update_traces(marker_color="turquoise",marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5)
fig.update_layout(title_text='timing' )
fig.show()
```

In [13]:
```
#for a piechart
import matplotlib.pyplot as plt
rating_counts = finefood["Score"].value_counts()
```

In [14]:
```
rating_counts
```

Out[14]:
```
5    363122
4     80655
1     52268
3     42640
2     29769
Name: Score, dtype: int64
```

In [15]:
```
plt.pie(rating_counts, labels=rating_counts.index)
plt.title("product rating")
plt.show()
```

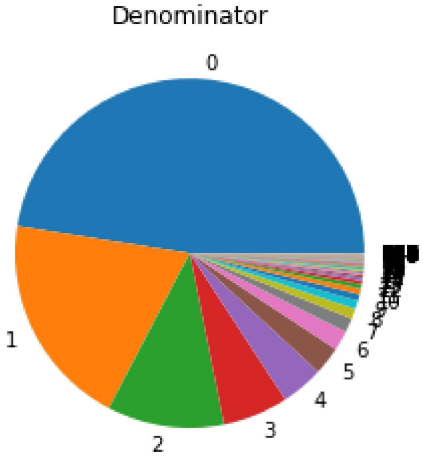


In [16]:
```
#for a piechart
import matplotlib.pyplot as plt
rating_counts = finefood["HelpfulnessDenominator"].value_counts()
```

In [17]:
```
rating_counts
```

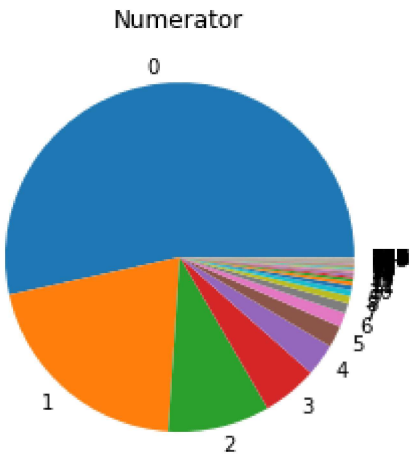
Out[17]:
```
0      270052
1      112753
2       61482
3       34394
4       22306
        ...
197         1
488         1
383         1
219         1
815         1
Name: HelpfulnessDenominator, Length: 234, dtype: int64
```

In [18]:
```
plt.pie(rating_counts, labels=rating_counts.index)
plt.title("Denominator")
plt.show()
```



In [19]:
```
#for a piechart
import matplotlib.pyplot as plt
rating_counts = finefood["HelpfulnessNumerator"].value_counts()
```

In [20]:
```
plt.pie(rating_counts, labels=rating_counts.index)
plt.title("Numerator")
plt.show()
```



In [21]:
```
# Histogram and pie chart visualisation was use for the score data
# They were chosen because they are numerical data
# For the score there rae lot of people that score 5, then follow by 4,1,3,2 and there
```

Now we are going to focus on three more columns:

Score : customer's product rating

Text : the full review written by the customer

We can see that the rating system is a numerical value in the range 0-5. A rating of 0 is the worst rating available and 5 is the best rating available. We want to start by getting a feel for the ratings, e.g., do we have more negative than positive reviews? The easiest way to see this is to plot the data.

What type of visualization did you choose to represent the score data?

Why did you choose it?

What do you notice about the results?

Question 3

In [22]:
```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/x-sogunmod/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[22]:
```
True
```

In [23]:
```
mystopwords=stopwords.words('english') + ["br", "href","b","r"]
```

In [24]:
```
finefood.Text[0]
```

Out[24]:
```
'I have bought several of the Vitality canned dog food products and have found them a
ll to be of good quality. The product looks more like a stew than a processed meat an
d it smells better. My Labrador is finicky and she appreciates this product better th
an  most.'
```

In [25]:
```
# i can first put it in lower case
# everything will be in lower case all through

finefood.Text[0].lower()
```

Out[25]:
```
'i have bought several of the vitality canned dog food products and have found them a
ll to be of good quality. the product looks more like a stew than a processed meat an
d it smells better. my labrador is finicky and she appreciates this product better th
an  most.'
```

In [26]:
```
# then i can split it into list of individual words

mywords=finefood.Text[0].lower().split()
```

In [27]:
```
# for words in my words, if the words is not in mystopwords from up, i want to print t

myfilteredwords=[word for word in mywords if word not in mystopwords ]
```

In [28]:
```
# then i will join them together again

onefilteredreview=' '.join(myfilteredwords)
```

In [29]:
```
onefilteredreview
```

Out[29]:
```
'bought several vitality canned dog food products found good quality. product looks l
ike stew processed meat smells better. labrador finicky appreciates product better mo
st.'
```

In [30]:
```
 myfilteredreviews=[]
```

In [31]:
```
# now we want to go do this for all of our reviews

for myreview in finefood.Text:
    mywords=myreview.lower().split()
    myfilteredwords=[word for word in mywords if word not in mystopwords]
    onefilteredreview=' '.join(myfilteredwords)
    myfilteredreviews.append(onefilteredreview)
```

In [32]:
```
myfilteredreviews[0]
```

Out[32]:
```
'bought several vitality canned dog food products found good quality. product looks l
ike stew processed meat smells better. labrador finicky appreciates product better mo
st.'
```

In [33]:
```
# seeing from zero through 5

myfilteredreviews[0:5]
```

Out[33]:
```
['bought several vitality canned dog food products found good quality. product looks
like stew processed meat smells better. labrador finicky appreciates product better m
ost.',
 'product arrived labeled jumbo salted peanuts...the peanuts actually small sized uns
alted. sure error vendor intended represent product "jumbo".',
 'confection around centuries. light pillowy citrus gelatin nuts - case filberts. cut
tiny squares liberally coated powdered sugar. tiny mouthful heaven. chewy flavorful.
highly recommend yummy treat. familiar story c.s. lewis\' "the lion witch wardrobe" -
treat seduces edmund selling brother sisters witch.',
 'looking secret ingredient robitussin believe found it. got addition root beer extra
ct ordered (which good) made cherry soda. flavor medicinal.',
 'great taffy great price. wide assortment yummy taffy. delivery quick. taffy lover d
eal.']
```

Markdown notes and sentences and analysis written here.

Question 4

In [34]:
```
# I will make one huge string by joining everything together in myfilteredreviews by s

myhugestring=' '.join(myfilteredreviews)
```

In [35]:
```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

In [36]:
```
# then build a world cloud and generate them from our hugestring

wordcloud=WordCloud(width=800, height=800, background_color='white', min_font_size=10)
```

In [37]:
```
plt.figure(figsize=(8,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show
```

Out[37]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

In [38]:
```
from collections import Counter
```

In [39]:
```
mycounts=Counter(myhugestring.lower().split())
```

In [40]:
```
mycounts.most_common(20)
```

Out[40]:
```
[('/><br', 257585),
 ('like', 243401),
 ('good', 162845),
 ('one', 155725),
 ('taste', 140018),
 ('great', 139303),
 ('coffee', 130651),
 ('love', 123318),
 ('would', 121974),
 ('product', 112594),
 ('flavor', 109469),
 ('tea', 108961),
 ('get', 106067),
 ('food', 99274),
 ('really', 99058),
 ('much', 84569),
 ('little', 82138),
 ('also', 80274),
 ('it.', 78711),
 ('use', 78126)]
```

In [41]:
```
mystopwords=stopwords.words('english') + ["br", "href","b","r",'/><br','like','product
```

In [42]:
```
for myreview in finefood.Text:
    mywords=myreview.lower().split()
    myfilteredwords=[word for word in mywords if word not in mystopwords]
    onefilteredreview=' '.join(myfilteredwords)
    myfilteredreviews.append(onefilteredreview)
```

In [43]:
```
myhugestring=' '.join(myfilteredreviews)
```

In [44]:
```
wordcloud=WordCloud(width=800, height=800, background_color='white', min_font_size=10)
```

In [45]:
```
plt.figure(figsize=(8,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show
```

Out[45]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

In [ ]:
```
# Observation was that, there was font reduction of the words displayed.
```

Create a wordcloud from the column "Text" that should have all the stop words taken out of it.

Are there any additional "stop words" or words that are unimportant to your analysis that you could take out (an example could be cant, gp, br, hef, etc)?

Take out those additional stop words and then create a new wordcloud. What do you notice?

Question 5

In [46]:
```
# code here
```

Markdown notes and sentences and analysis written here.

Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together – We are Purdue.