Hi, my name is Omotosho Esther, i worked on this report which attempts to answer the following questions from a dataset on chocholate bars obtained from various countries: What is the average rating of these chcholates by country of origin? How many bars were reviewed for each of those countries? Create plots to visualize findings for questions 1 and 2. Is the cacao bean's origin an indicator of quality? How does cocoa content relate to rating? What is the average cocoa content for bars with higher ratings (above 3.5)? Your research indicates that some consumers want to avoid bars with lecithin. Compare the average rating of bars with and without lecithin (L in the ingredients). Summarize your findings.

In [1]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import pandas as pd
import numpy as np
data = pd.read_csv('chocolate_bars.csv')
```

In [2]:
```
data
#so the first thing i want to do is to see my dataset
```

Out[2]:

	id	manufacturer	company_location	year_reviewed	bean_origin	bar_name	cocoa_per
0	2454	5150	U.S.A.	2019	Tanzania	Kokoa Kamili, batch 1	
1	2458	5150	U.S.A.	2019	Dominican Republic	Zorzal, batch 1	
2	2454	5150	U.S.A.	2019	Madagascar	Bejofo Estate, batch 1	
3	2542	5150	U.S.A.	2021	Fiji	Matasawalevu, batch 1	
4	2546	5150	U.S.A.	2021	Venezuela	Sur del Lago, batch 1	
...	...	...	...	...	...	...	
2525	1205	Zotter	Austria	2014	Blend	Raw	
2526	1996	Zotter	Austria	2017	Colombia	APROCAFA, Acandi	
2527	2036	Zotter	Austria	2018	Blend	Dry Aged, 30 yr Anniversary bar	
2528	2170	Zotter	Austria	2018	Congo	Mountains of the Moon	
2529	2170	Zotter	Austria	2018	Belize	Maya Mtn	

2530 rows × 11 columns

In [3]:
```
data.columns.to_list()
#haven done that, i want to check my columns
```

Out[3]:
```
['id',
 'manufacturer',
 'company_location',
 'year_reviewed',
 'bean_origin',
 'bar_name',
 'cocoa_percent',
 'num_ingredients',
 'ingredients',
 'review',
 'rating']
```

In [4]:
```
data.isnull().sum()
#then i am moving on to the data cleaning stage, i want to actually check if my d
```

Out[4]:
```
id                  0
manufacturer        0
company_location    0
year_reviewed       0
bean_origin         0
bar_name            0
cocoa_percent       0
num_ingredients    87
ingredients        87
review              0
rating              0
dtype: int64
```

In [5]:
```
data.manufacturer.unique()
```

Out[5]:
```
array(['5150', 'A. Morin', 'Acalli', 'Adi aka Fijiana (Easy In Ltd)',
       'Aelan', 'Aequare (Gianduja)', 'Ah Cacao', "Akesson's (Pralus)",
       'Alain Ducasse', 'Alexandre', 'Altus aka Cao Artisan', 'Amano',
       'Amatller (Simon Coll)', 'Amazing Cacao', 'Amazona', 'Ambrosia',
       'Amedei', 'AMMA', 'Anahata', 'Animas', 'Ara', 'Arete', 'Argencove',
       'Artisan du Chocolat', 'Artisan du Chocolat (Casa Luker)',
       'Aruntam', 'Askinosie', 'Atypic', 'Auro', 'Avanaa', 'Bahen & Co.',
       'Baiani', 'Bakau', 'Bankston', 'Bar Au Chocolat', "Baravelli's",
       'Batch', 'Bean', 'Beau Cacao', 'Beehive', 'Belcolade',
       'Bellflower', 'Belvie', 'Belyzium', 'Benns', 'Benoit Nihant',
       'Bernachon', 'Beschle (Felchlin)', 'Bisou', 'Bitacora',
       'Bittersweet Origins', 'Bixby', 'Black Mountain',
       'Black River (A. Morin)', 'Black Sheep', 'Blanxart',
       'Blue Bandana', 'Boho', 'Bonaterra', 'Bonnat',
       'Bouga Cacao (Tulicorp)', 'Bowler Man', 'Brasstown',
       "Brasstown aka It's Chocolate", 'Brazen', 'Breeze Mill', 'Bright',
       'Britarev', 'Bronx Grrl Chocolate', 'Bullion', 'Burnt Fork Bend',
       'By Cacao', 'Cacai Cacao', 'Cacao 70', 'Cacao Arabuco',
       'Cacao Atlanta', 'Cacao Barry', 'Cacao Betulia', 'Cacao de Origen',
```

In [6]:
```
pd.options.display.max_rows = 9999
data
```

Out[6]:

	id	manufacturer	company_location	year_reviewed	bean_origin	bar_n
0	2454	5150	U.S.A.	2019	Tanzania	Kokoa Kamili, ba
1	2458	5150	U.S.A.	2019	Dominican Republic	Zorzal, ba
2	2454	5150	U.S.A.	2019	Madagascar	Bejofo Estate, ba
3	2542	5150	U.S.A.	2021	Fiji	Matasawalevu, ba
4	2546	5150	U.S.A.	2021	Venezuela	Sur del Lago, ba
5	2546	5150	U.S.A.	2021	Uganda	Semuliki Forest, ba
6	2542	5150	U.S.A.	2021	India	Anamalai, ba
7	797	A. Morin	France	2012	Bolivia	B
8	797	A. Morin	France	2012	Peru	

In [7]:
```
data.describe(include = 'object')
```

Out[7]:

	manufacturer	company_location	bean_origin	bar_name	ingredients	review
count	2530	2530	2530	2530	2443	2530
unique	580	67	62	1605	21	2487
top	Soma	U.S.A.	Venezuela	Madagascar	B,S,C	spicy, cocoa
freq	56	1136	253	55	999	4

In [8]:
```
data.describe()
#this table gives me important information about my data set such as the mean, st
```

Out[8]:

	id	year_reviewed	cocoa_percent	num_ingredients	rating
count	2530.000000	2530.000000	2530.000000	2443.000000	2530.000000
mean	1429.800791	2014.374308	71.639723	3.041343	3.196344
std	757.648556	3.968267	5.616724	0.913728	0.445321
min	5.000000	2006.000000	42.000000	1.000000	1.000000
25%	802.000000	2012.000000	70.000000	2.000000	3.000000
50%	1454.000000	2015.000000	70.000000	3.000000	3.250000
75%	2079.000000	2018.000000	74.000000	4.000000	3.500000
max	2712.000000	2021.000000	100.000000	6.000000	4.000000

In [9]:
```
data.num_ingredients.dtype
```

Out[9]:
```
dtype('float64')
```

In [10]:
```
data["ingredients"].dtype
```

Out[10]:
```
dtype('O')
```

In [11]:
```
data["num_ingredients"].mean()
```

Out[11]:
```
3.0413426115431847
```

In [12]:
```
data["ingredients"].mode()
```

Out[12]:
```
0    B,S,C
Name: ingredients, dtype: object
```

In [13]:
```
#i did some of these processes above to check fill in the missing values, so for
#ingredients, i filled them with the mode(the highest occuring ingredient), i am
#values in my num_ingredients with the mean.
data['ingredients'].fillna(value = "B,S,C", inplace = True)
```

In [14]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2530 entries, 0 to 2529
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                2530 non-null   int64  
 1   manufacturer      2530 non-null   object 
 2   company_location  2530 non-null   object 
 3   year_reviewed     2530 non-null   int64  
 4   bean_origin       2530 non-null   object 
 5   bar_name          2530 non-null   object 
 6   cocoa_percent     2530 non-null   float64
 7   num_ingredients   2443 non-null   float64
 8   ingredients       2530 non-null   object 
 9   review            2530 non-null   object 
 10  rating            2530 non-null   float64
dtypes: float64(3), int64(2), object(6)
memory usage: 217.5+ KB
```

In [15]:
```
data['num_ingredients'].fillna(value =3.0, inplace = True)
```

In [16]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2530 entries, 0 to 2529
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                2530 non-null   int64  
 1   manufacturer      2530 non-null   object 
 2   company_location  2530 non-null   object 
 3   year_reviewed     2530 non-null   int64  
 4   bean_origin       2530 non-null   object 
 5   bar_name          2530 non-null   object 
 6   cocoa_percent     2530 non-null   float64
 7   num_ingredients   2530 non-null   float64
 8   ingredients       2530 non-null   object 
 9   review            2530 non-null   object 
 10  rating            2530 non-null   float64
dtypes: float64(3), int64(2), object(6)
memory usage: 217.5+ KB
```

In [17]:
```
data.isnull().sum()
```

Out[17]:
```
id                  0
manufacturer        0
company_location    0
year_reviewed       0
bean_origin         0
bar_name            0
cocoa_percent       0
num_ingredients     0
ingredients         0
review              0
rating              0
dtype: int64
```

data analysis and visualization stage

What is the average rating by country of origin?

In [18]:
```
r = data.groupby('bean_origin')[['rating']].mean()
r
```

Out[18]:

bean_origin	rating
Australia	3.250000
Belize	3.233553
Blend	3.038462
Bolivia	3.181250
Brazil	3.262821
Burma	3.000000
Cameroon	3.083333
China	3.500000
Colombia	3.196203
Congo	3.318182
Costa Rica	3.151163
Cuba	3.291667
DR Congo	3.000000
Dominican Republic	3.215708
Ecuador	3.164384
El Salvador	3.000000
Fiji	3.062500
Gabon	3.250000
Ghana	3.134146
Grenada	3.026316
Guatemala	3.258065
Haiti	3.266667
Honduras	3.240000
India	3.164286
Indonesia	3.112500
Ivory Coast	2.857143
Jamaica	3.197917
Liberia	3.083333
Madagascar	3.266949
Malaysia	3.093750
Martinique	2.750000
Mexico	3.168182
Nicaragua	3.255000

bean_origin	rating
Nigeria	3.000000
Panama	3.111111
Papua New Guinea	3.280000
Peru	3.197746
Philippines	3.125000
Principe	2.750000
Puerto Rico	2.714286
Samoa	3.083333
Sao Tome	3.071429
Sao Tome & Principe	3.500000
Sierra Leone	2.750000
Solomon Islands	3.450000
Sri Lanka	2.875000
St. Lucia	2.950000
St.Vincent-Grenadines	2.750000
Sulawesi	3.250000
Sumatra	3.000000
Suriname	3.250000
Taiwan	2.875000
Tanzania	3.234177
Thailand	3.300000
Tobago	3.625000
Togo	3.083333
Trinidad	3.244048
U.S.A.	3.242424
Uganda	3.065789
Vanuatu	3.115385
Venezuela	3.231225
Vietnam	3.287671

In [19]:
```
#if it is by company location
data.groupby('company_location')[['rating']].mean()
```

Out[19]:

company_location	rating
Amsterdam	3.312500
Argentina	3.305556
Australia	3.358491
Austria	3.258333
Belgium	3.103175
Bolivia	3.250000
Brazil	3.280000
Canada	3.303672
Chile	3.750000
Colombia	3.198276
Costa Rica	3.138889
Czech Republic	3.000000
Denmark	3.338710
Dominican Republic	3.113636
Ecuador	3.038793
El Salvador	3.000000
Fiji	3.250000
Finland	3.250000
France	3.258523
Germany	3.208333
Ghana	2.750000
Grenada	2.833333
Guatemala	3.350000
Honduras	3.208333
Hungary	3.221154
Iceland	3.312500
India	2.625000
Ireland	2.900000
Israel	3.250000
Italy	3.230769
Japan	3.129032
Lithuania	3.125000

company_location	rating
Madagascar	3.147059
Malaysia	2.833333
Martinique	2.750000
Mexico	3.100000
Netherlands	3.125000
New Zealand	3.212963
Nicaragua	3.100000
Norway	3.333333
Peru	3.076087
Philippines	3.150000
Poland	3.375000
Portugal	2.750000
Puerto Rico	2.625000
Russia	3.250000
Sao Tome	2.875000
Sao Tome & Principe	2.812500
Scotland	3.272727
Singapore	3.200000
South Africa	2.750000
South Korea	3.181818
Spain	3.263889
St. Lucia	2.750000
St.Vincent-Grenadines	2.750000
Suriname	3.250000
Sweden	3.000000
Switzerland	3.318182
Taiwan	3.100000
Thailand	3.300000
U.A.E.	3.400000
U.K.	3.069549
U.S.A.	3.190801
Vanuatu	2.750000
Venezuela	3.112903
Vietnam	3.359375
Wales	2.750000

In [20]:
```
#2. how many bars were reviewed per country
review = data.groupby('bean_origin')[['review']].count()
```

In [21]:
```
review
#this gives us the number of bars reviewed per each country.
```

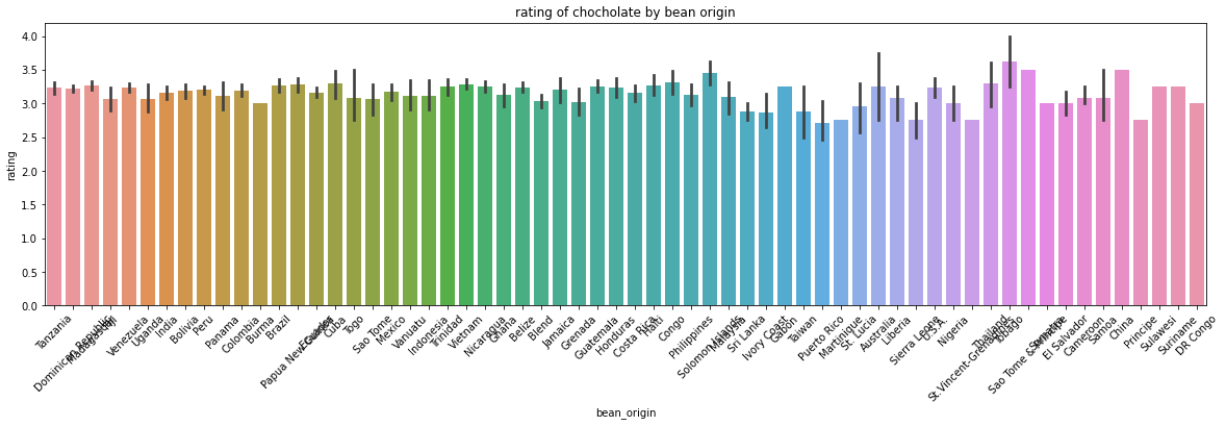
Out[21]:

bean_origin	review
Australia	3
Belize	76
Blend	156
Bolivia	80
Brazil	78
Burma	1
Cameroon	3
China	1
Colombia	79
Congo	11
Costa Rica	43
Cuba	12
DR Congo	1
Dominican Republic	226
Ecuador	219
El Salvador	6
Fiji	16
Gabon	1
Ghana	41
Grenada	19
Guatemala	62
Haiti	30
Honduras	25
India	35
Indonesia	20
Ivory Coast	7
Jamaica	24
Liberia	3
Madagascar	177
Malaysia	8
Martinique	1
Mexico	55
Nicaragua	100

bean_origin	review
Nigeria	3
Panama	9
Papua New Guinea	50
Peru	244
Philippines	24
Principe	1
Puerto Rico	7
Samoa	3
Sao Tome	14
Sao Tome & Principe	2
Sierra Leone	4
Solomon Islands	10
Sri Lanka	2
St. Lucia	10
St.Vincent-Grenadines	1
Sulawesi	1
Sumatra	1
Suriname	1
Taiwan	2
Tanzania	79
Thailand	5
Tobago	2
Togo	3
Trinidad	42
U.S.A.	33
Uganda	19
Vanuatu	13
Venezuela	253
Vietnam	73

No 3 question i am asked to create plots to visualize findings for questions 1 and 2. so, i am just gonna be trying out different plots and see the one from which i can extract loads of inference from

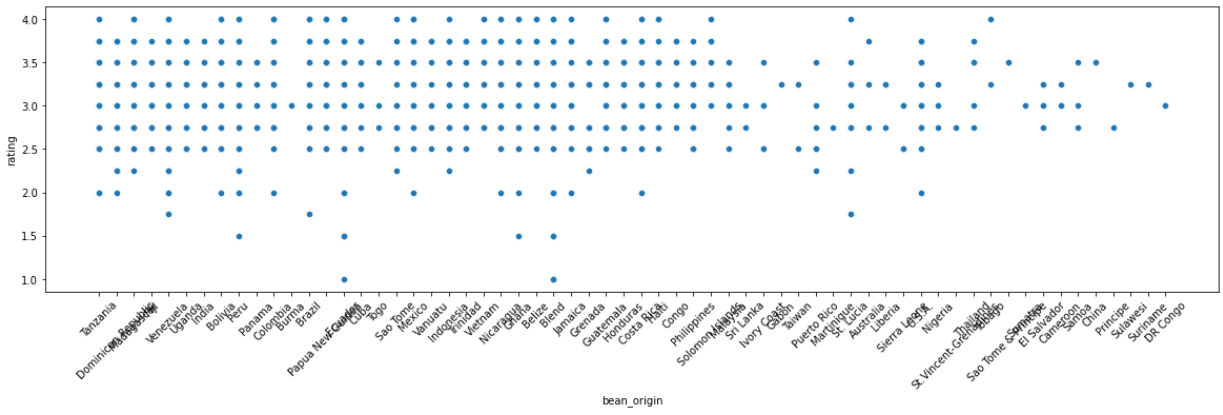
In [44]:
```
p = sns.barplot( x = "bean_origin", y = "rating", data = data)
plt.rcParams["figure.figsize"] = [20,5]
p.set(title = 'rating of chocholate by bean origin')
p.set_xticklabels(p.get_xticklabels(), rotation = 45);
```



the plot above gives me the rating of chocholate bars based on the cocaoa beans origin, we can infer from this plot that St Vincent- Grenadines has cocoa bean produces chocholates with the highest rating

In [23]:
```
sp = sns.scatterplot(x= 'bean_origin', y = 'rating', data = data)
sp.set_xticklabels(p.get_xticklabels(), rotation = 45);
```

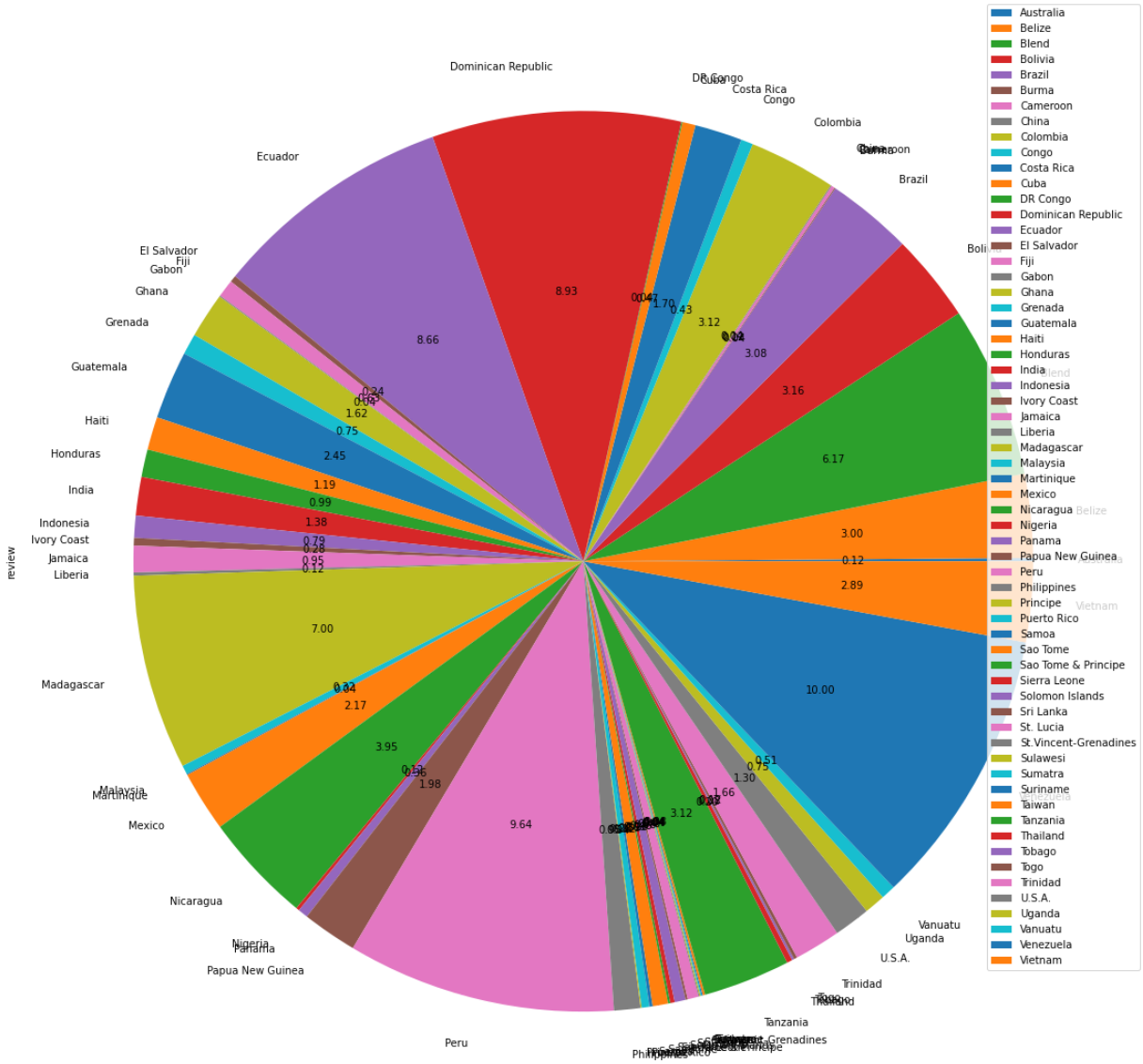
```
C:\Users\USER\AppData\Local\Temp\ipykernel_8472\782286473.py:2: UserWarning: Fi
xedFormatter should only be used together with FixedLocator
  sp.set_xticklabels(p.get_xticklabels(), rotation = 45);
```



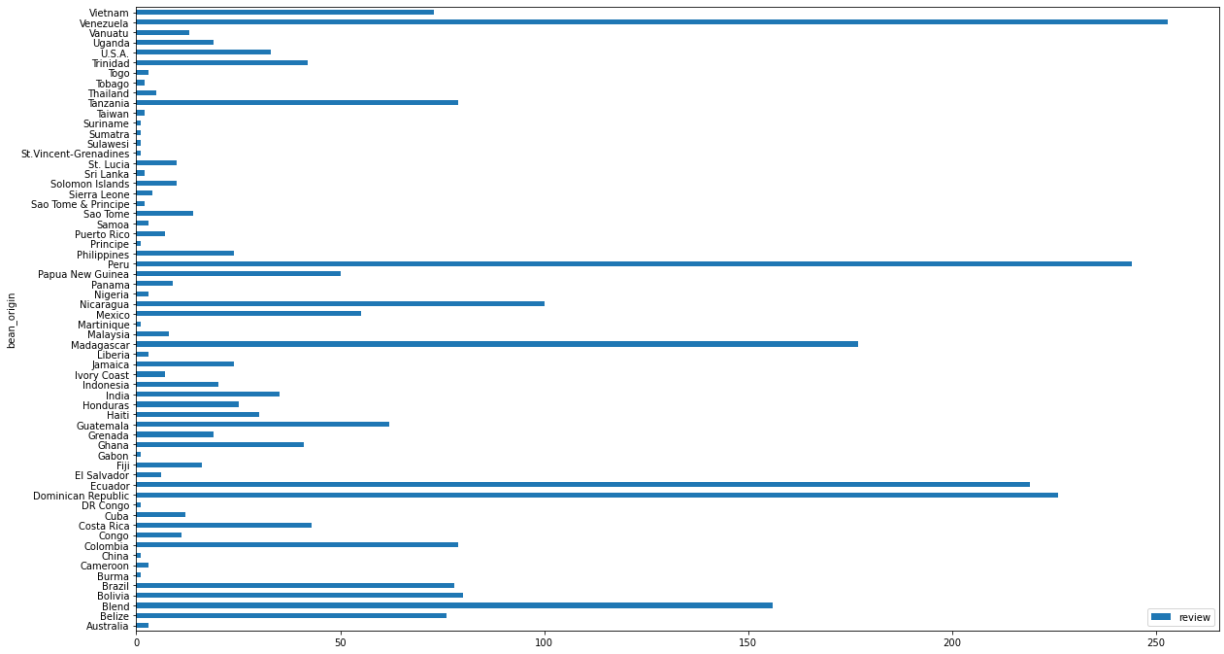
in visualizing question 2 on how many bars were reviewed per country, i will also use a bar chart to do that

In [24]:
```
data.groupby('bean_origin')[['review']].count().plot.pie(autopct='%.2f', figsize=
```

Out[24]:
```
array([<AxesSubplot:ylabel='review'>], dtype=object)
```



In [25]:
```
r= data.groupby('bean_origin')[['review']].count().plot(kind= 'barh')
r.figure.set_figheight(11.8)
```



the pie chart and the barh shows me that venezuela had the most chocholate bars which were reviewed holding 10% of the total chocholate bars reviewed as shown on the pie chart and having about 250(253 as shown in the table) reviewed chocholates as shown on the barh

question 3. the next question is about showing if the cocoa bean's origin is an indicator of quality?

so, first i wouuld say the best prove of quality would be the ratings of this chocholate bars so, in checking if there id any correlation between cocoa bean's origin and the quality of the chocholate bars i will be checking the correlation between bean origin and the ratings

In [26]:
```
#so, what we can use to check for any relationship would be
# so we actually have a very large dataset, so to prove if there is any relations
#the top 10 countries producing cocoa beans
# This will be
print(data['bean_origin'].value_counts().head(10))
```

```
Venezuela             253
Peru                  244
Dominican Republic    226
Ecuador               219
Madagascar            177
Blend                 156
Nicaragua             100
Bolivia                80
Tanzania               79
Colombia               79
Name: bean_origin, dtype: int64
```

In [27]:
```
#so, i have gotte my top ten cocoa bean producers,
best_ratings = []
for index, row in data.iterrows():
    if row['rating'] >= 4:
        best_ratings.append(row)
best_ratings = pd.DataFrame(best_ratings)
```

In [28]:
```
best_ratings
```

Out[28]:

	id	manufacturer	company_location	year_reviewed	bean_origin	bar_na
18	1015	A. Morin	France	2013	Venezuela	Chu
19	1019	A. Morin	France	2013	Peru	Chanchamayo Provi
24	1319	A. Morin	France	2014	Peru	Pabl
32	2648	A. Morin	France	2021	Mexico	La J
79	470	Amano	U.S.A.	2010	Ecuador	Gua
80	725	Amano	U.S.A.	2011	Papua New Guinea	Mor
111	572	AMMA	Brazil	2010	Brazil	Monte Alegre, 3 plantati
129	1598	Arete	U.S.A.	2015	Nicaragua	Chu
141	1908	Arete	U.S.A.	2016	Costa Rica	Coto Brus, Terciop
142	1924	Arete	U.S.A.	2016	Peru	Phant

In [29]:
```
origin_rating = best_ratings['bean_origin'].value_counts().head(10)
origin_rating
```

Out[29]:
```
Venezuela           20
Peru                19
Madagascar          11
Ecuador              8
Blend                7
Brazil               5
Bolivia              5
Colombia             5
Mexico               5
Papua New Guinea     4
Name: bean_origin, dtype: int64
```

In [30]:
```
#this is already showing me that okay for venezula, only 20 of their chocholate b
#my next stop is to convert this my output to a dataframe
origin_rating = pd.DataFrame(origin_rating). reset_index()
```

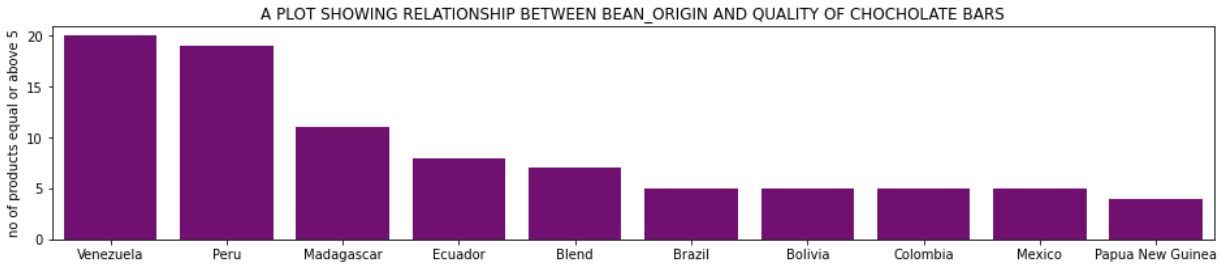
In [31]:
```
origin_rating
```

Out[31]:

	index	bean_origin
0	Venezuela	20
1	Peru	19
2	Madagascar	11
3	Ecuador	8
4	Blend	7
5	Brazil	5
6	Bolivia	5
7	Colombia	5
8	Mexico	5
9	Papua New Guinea	4

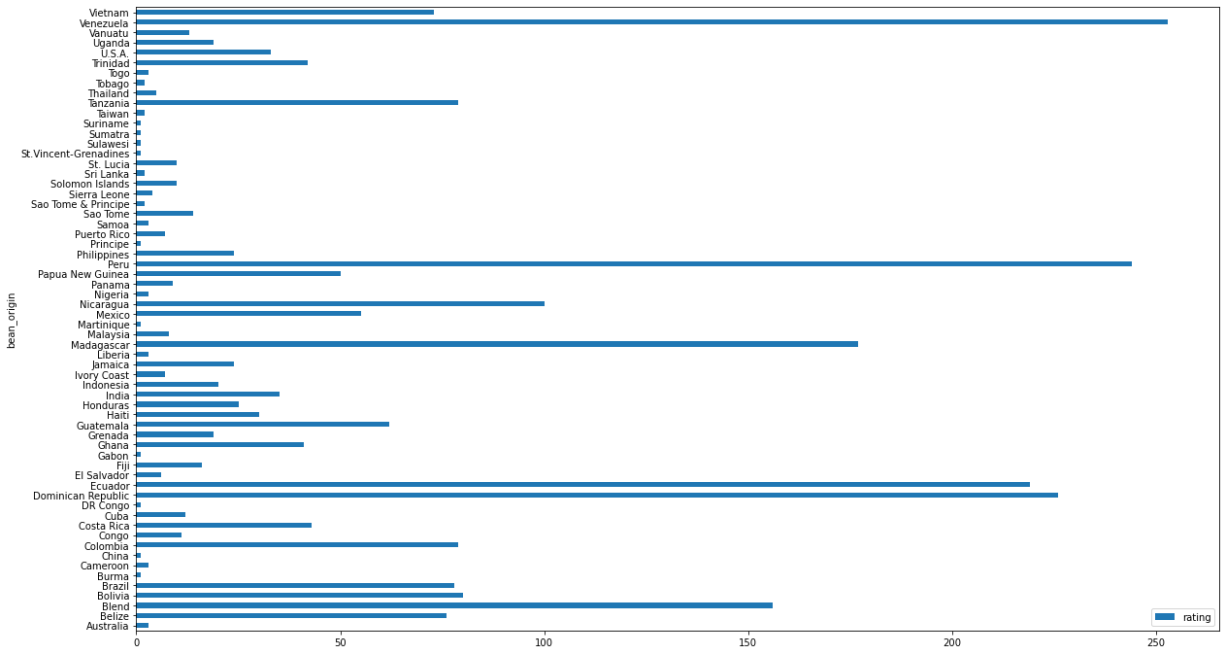
In [32]:
```
#okay then i am now going to have a bar plot to show the relationship
fig, ax = plt.subplots(figsize =(16,3))
sns.barplot(x= 'index', y = 'bean_origin', data = origin_rating, color='purple',
ax.set_title('A PLOT SHOWING RELATIONSHIP BETWEEN BEAN_ORIGIN AND QUALITY OF CHOC
ax.set(xlabel= '', ylabel= 'no of products equal or above 5')
```

Out[32]:
```
[Text(0.5, 0, ''), Text(0, 0.5, 'no of products equal or above 5')]
```

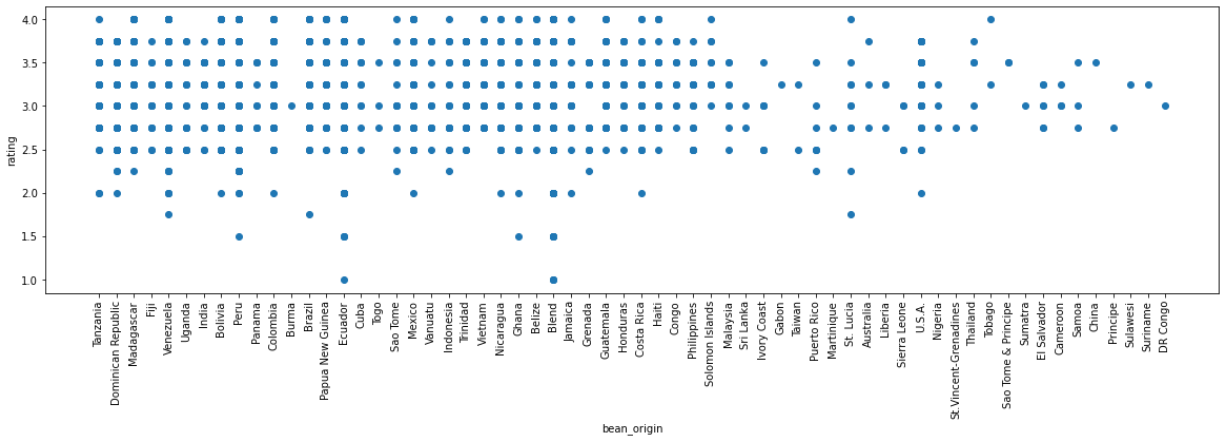


venezuela being the largest producer of cocoa beans and having the highest products with best ratings could mean something however, let us check a scatter plot to see if there is actually any relationship between bean_origin and quality

In [33]:
```
data1 = data.groupby('bean_origin')[['rating']].count().plot(kind= 'barh')
data1.figure.set_figheight(11.8)
```



In [34]:
```
data2 = plt.scatter( x = "bean_origin", y = "rating", data = data)
plt.xlabel("bean_origin")
plt.ylabel("rating")
plt.xticks(rotation = 90);
```



from the scatter plot, it is evident that there is no relationship between the bean origin and the quality of the chocholate bars thus bean origin is not an indicator of quality

In [35]:
```
# so moving on to the next question, the next question is talking about if cocoa
#the percentage of cocoa and ratings are continous variables, i can check for the
#variables to deduce their relationship
correlation = data.corr(method='pearson')
#so here, i basically did the correlation of my entire table columns
```

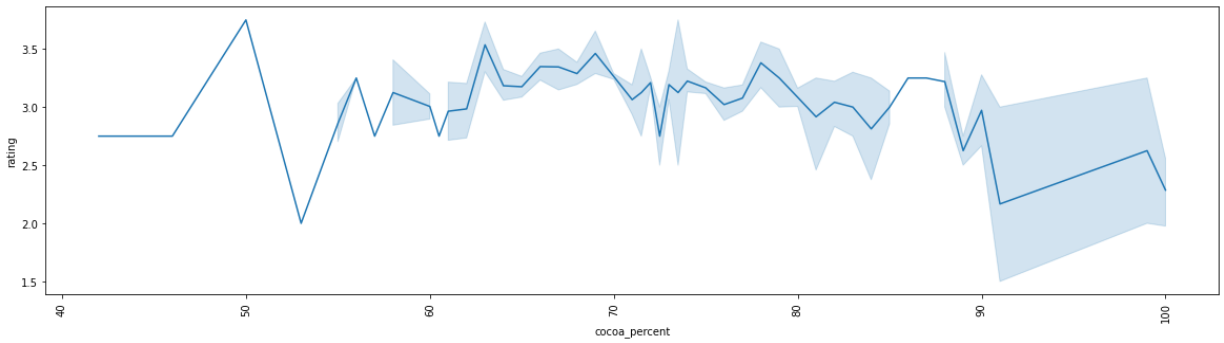
In [36]:
```
correlation
```

Out[36]:

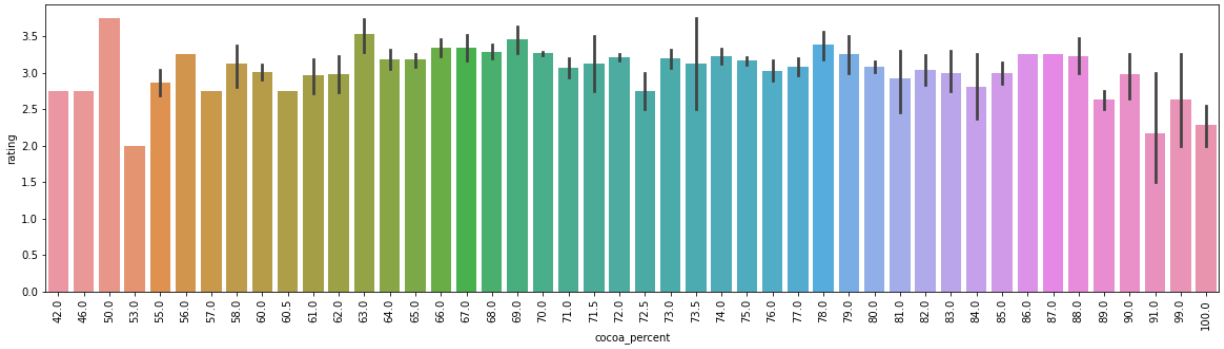
	id	year_reviewed	cocoa_percent	num_ingredients	rating
id	1.000000	0.993126	0.015835	-0.348381	0.113398
year_reviewed	0.993126	1.000000	0.016484	-0.354428	0.116256
cocoa_percent	0.015835	0.016484	1.000000	-0.170263	-0.146690
num_ingredients	-0.348381	-0.354428	-0.170263	1.000000	-0.092046
rating	0.113398	0.116256	-0.146690	-0.092046	1.000000

from the above i can see that the pearson correlation coefficient r is -0.146690, this suggests a weak or negligible negative association between cocoa percentage in chocholate bars and the rating(which in here is used to measure quality).

In [37]:
```
#so now i want to actually create a line graph to have a visual picture of my inf
sns.lineplot(x='cocoa_percent', y='rating', data= data)
plt.xticks(rotation=90);
```



In [38]:
```
sns.barplot(x='cocoa_percent', y='rating', data=data)
plt.xticks(rotation=90);
```



from the above figure, we can see that there isn't visual relationship between both variables

so let me move on to the next question. the next question is asking for the average cocoa content for bars with higher ratings (above 3.5)

In [39]:
```
data[data['rating']>3.5].cocoa_percent.mean()
```

Out[39]:
```
70.94781553398059
```

this showed that the average cocoa content in the chocholate bars with ratings > 3.5 is 70.9%

finally, the last question is asking, Your research indicates that some consumers want to avoid bars with lecithin. Compare the average rating of bars with and without lecithin (L) in the ingredients

In [40]:
```
data.groupby('ingredients')[['rating']].count()
```

Out[40]:

ingredients	rating
B	6
B,C	1
B,S	718
B,S*	31
B,S*,C	12
B,S*,C,L	2
B,S*,C,Sa	20
B,S*,C,V	7
B,S*,Sa	1
B,S*,V,L	3
B,S,C	1086
B,S,C,L	286
B,S,C,L,Sa	1
B,S,C,Sa	5
B,S,C,V	141
B,S,C,V,L	184
B,S,C,V,L,Sa	4
B,S,C,V,Sa	6
B,S,L	8
B,S,V	3
B,S,V,L	5

In [49]:
```
data['ingredients']= data['ingredients'].str.strip('')
ingredients1 = data['ingredients'].str.get_dummies(sep ='')
chocholate_lecithin = pd.concat([data,ingredients1['L']], axis =1)
chocholate_has_lecithin =chocholate_lecithin[chocholate_lecithin['L']==1]
chocholate_has_no_lecithin = chocholate_lecithin[chocholate_lecithin['L']==0]
#so, what i did here was simple, i first had to strip off the quotation'', why yo
#ingredients values. then i then used the eries.str.get_dummies(sep='')function,
#it seperates each string, remember i already split my ingredients so each word i
#splits each of my string then returns the values of each strings back to me in o
# after doing that. i then concat my data and the newly created dataframe column
# LECITHIN, YOU GET) together

# after doing that, i then said okay, let me group my data right. so i said if ch
#what it should do for me is that it should check through my cocat dataframe, if
#get enlisted under chocholate_has_lecithin and viceversa
```

In [43]:
```
chocholate_has_lecithin
```

Out[43]:

	id	manufacturer	company_location	year_reviewed	bean_origin	bar_name
7	797	A. Morin	France	2012	Bolivia	Bolivia
8	797	A. Morin	France	2012	Peru	Peru
9	1011	A. Morin	France	2013	Panama	Panama
10	1015	A. Morin	France	2013	Colombia	Colombie
11	1011	A. Morin	France	2013	Madagascar	Madagascar, Criollo
12	1015	A. Morin	France	2013	Burma	Birmanie
13	1011	A. Morin	France	2013	Brazil	Brazil
14	1015	A. Morin	France	2013	Papua New Guinea	Papua New Guinea
15	1019	A. Morin	France	2013	Peru	Piura
16	1019	A. Morin	France	2013	Peru	Chanchamayo

In [46]:
```
ingredients1
```

Out[46]:

	*	,	B	C	L	S	V	a
0	0	1	1	1	0	1	0	0
1	0	1	1	1	0	1	0	0
2	0	1	1	1	0	1	0	0
3	0	1	1	1	0	1	0	0
4	0	1	1	1	0	1	0	0
5	0	1	1	1	0	1	0	0
6	0	1	1	1	0	1	0	0
7	0	1	1	1	1	1	0	0
8	0	1	1	1	1	1	0	0
9	0	1	1	1	1	1	0	0
10	0	1	1	1	1	1	0	0
11	0	1	1	1	1	1	0	0

In [48]:
```
chocholate_lecithin = pd.concat([data,ingredients1['L']], axis =1)
chocholate_lecithin
```

Out[48]:

	id	manufacturer	company_location	year_reviewed	bean_origin	bar_r
0	2454	5150	U.S.A.	2019	Tanzania	Kokoa Kamili, ba
1	2458	5150	U.S.A.	2019	Dominican Republic	Zorzal, ba
2	2454	5150	U.S.A.	2019	Madagascar	Bejofo Estate, ba
3	2542	5150	U.S.A.	2021	Fiji	Matasawalevu, ba
4	2546	5150	U.S.A.	2021	Venezuela	Sur del Lago, ba
5	2546	5150	U.S.A.	2021	Uganda	Semuliki Forest, ba
6	2542	5150	U.S.A.	2021	India	Anamalai, ba
7	797	A. Morin	France	2012	Bolivia	B
8	797	A. Morin	France	2012	Peru	

In [50]:
```
rating_chocholate_has_lecithin = chocholate_has_lecithin['rating'].mean()
rating_chocholate_has_no_lecithin = chocholate_has_no_lecithin['rating'].mean()
#this is basically me then finiding the mean rating of the group i like lecithin

print("the average rating of chocholates with lecithin is", rating_chocholate_has
print("the average rating of chocholates without lecithin is", rating_chocholate_
```

```
the average rating of chocholates with lecithin is 3.150608519269777
the average rating of chocholates without lecithin is 3.2074128620520375
```

this shows that the average rating of chocholate without lecithin is higher than chocholates with lecithin

summary: in summary my findings as i analysed this data includes; venezuela had the most chocholate bars which were reviewed holding 10% of the total chocholate bars reviewed as shown on the pie chart and having about 250(253 as shown in the table) reviewed chocholates as shown on the barh followed by peru then dominican republic.

```
2.venezuela being the largest producer of cocoa beans also had the highe
st products with best ratings
however, there was no relationship found between cocoa bean origin and q
uality

3. there was no relationship between cocoa percent and the quality of ch
ocholate bars using ratings as the judge of quality

4. the average cocoa percent in chocholate bars that had ratings above
 3.5 was 70.94%

5. the average rating of bars with and without lecithin (L) in the ingre
dients were compared and it was found
out that; the average rating of chocholates with lecithin was 3.15060851
9269777
```

the average rating of chocholates without lecithin was 3.2074128620520375, showing that the average rating of chocholate without lecithin is higher than chocholates with lecithin.

In [ ]: