**Chocolate Bar Ratings**

Chocolate is one of the most popular candies in the world. Each year, residents of the United States collectively eat more than 2.8 billions pounds. However, not all chocolate bars are created equal! This dataset contains expert ratings of over 1,700 individual chocolate bars, along with information on their regional origin, percentage of cocoa, the variety of chocolate bean used and where the beans were grown.

**Flavors of Cacao Rating System:**

* 5= Elite (Transcending beyond the ordinary limits)
* 4= Premium (Superior flavor development, character and style)
* 3= Satisfactory(3.0) to praiseworthy(3.75) (well made with special qualities)
* 2= Disappointing (Passable but contains at least one significant flaw)
* 1= Unpleasant (mostly unpalatable)

**Data description**

* **Company (Maker-if known)** - Name of the company manufacturing the bar.
* **Specific Bean Origin or Bar Name** - The specific geo-region of origin for the bar.
* **REF** - Help us describe this column... **What is it?**
* **Review Date** - Date of publication of the review.
* **Cocoa Percent** - Cocoa percentage (darkness) of the chocolate bar being reviewed.
* **Company Location** - Manufacturer base country.
* **Rating** - Expert rating for the bar.
* **Bean Type** - The variety (breed) of bean used, if provided.
* **Broad Bean Origin** - The broad geo-region of origin for the bean.

**Table of contents**

1. **Data preparation and the Exploratory Data Analysis**

Load and Cleaning of Data

First of all these are the libraries that we have import for the analysis;

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Warnings
* re
* os

*#imports*

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**

**import** **warnings**

**import** **re**

**import** **os**

The data here is provided to us in a csv file and kept on the desktop directory, which is inside the current working directory. I just simply load it here using the read csv function from the pd.read\_excel

*#load data*

data = pd.read\_excel("C:/Users/MPortal/Desktop/flavors\_of\_cacao.xlsx")

Now that we have read in the data we should explore the data set to try and get an understanding about how the data is stored for us.

We can check the dimensions of the data.

**Data Preparation**

*#explore the first 5 rows*

data.head().T

V

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| **CompanyÂ \n(Maker-if known)** | A.M0rin | A.M0rin | A.M0rin | A.M0rin | A.M0rin |
| **Specific Bean Origin\nor Bar Name** | Agua Grande | Kpime | Atsane | Akata | Quilla |
| **REF** | 1876 | 1676 | 1676 | 1680 | 1704 |
| **Review\nDate** | 2016 | 2015 | 2015 | 2015 | 2015 |
| **Cocoa\nPercent** | 2015 | 0.7 | 0.7 | 0.7 | 0.7 |
| **Company\nLocation** | France | France | France | France | France |
| **Rating** | 3.75 | 2.75 | 3.0 | 3.5 | 3.5 |
| **Bean\nType** | Â | Â | Â | Â | Â |
| **Broad Bean\nOrigin** | Sao Tome | Togo | Togo | Togo | Peru |

*#explore the last 5 rows*

data.tail().T

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1791** | **1792** | **1793** | **1794** |
| **CompanyÂ \n(Maker-if known)** | Zotter | Zotter | Zotter | Zotter |
| **Specific Bean Origin\nor Bar Name** | Congo | Kerala State | Kerala State | Brazil, Mitzi Blue |
| **REF** | 749 | 749 | 781 | 486 |
| **Review\nDate** | 2011 | 2011 | 2011 | 2010 |
| **Cocoa\nPercent** | 0.65 | 0.65 | 0.65 | 0.65 |
| **Company\nLocation** | Austria | Austria | Austria | Austria |
| **Rating** | 3.0 | 3.5 | 3.25 | 3.0 |
| **Bean\nType** | Forastero | Forastero | Â | Â |
| **Broad Bean\nOrigin** | Congo | India | India | Brazil |

*#explore the last 5 rows*

data.tail().T

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **count** | **unique** | **top** | **freq** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **CompanyÂ \n(Maker-if known)** | 1795 | 416 | Soma | 47 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **Specific Bean Origin\nor Bar Name** | 1795 | 1039 | Madagascar | 57 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **REF** | 1795.0 | NaN | NaN | NaN | 1035.904735 | 552.886365 | 5.0 | 576.0 | 1069.0 | 1502.0 | 1952.0 |
| **Review\nDate** | 1795.0 | NaN | NaN | NaN | 2012.325348 | 2.92721 | 2006.0 | 2010.0 | 2013.0 | 2015.0 | 2017.0 |
| **Cocoa\nPercent** | 1795.0 | NaN | NaN | NaN | 0.716983 | 0.063231 | 0.42 | 0.7 | 0.7 | 0.75 | 1.0 |
| **Company\nLocation** | 1795 | 60 | U.S.A. |  | 764 | NaN | NaN | NaN | NaN | NaN | NaN |
| **Rating** | 1795.0 |  | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **Bean\nType** | 1795 | 41 | Â | 887 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **Broad Bean\nOrigin** | 1794 | 100 | Venezuela | 214 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

*#explore datatypes*

data.dtypes

CompanyÂ \n(Maker-if known) object

Specific Bean Origin\nor Bar Name object

REF int64

Review\nDate int64

Cocoa\nPercent float64

Company\nLocation object

Rating float64

Bean\nType object

Broad Bean\nOrigin object

dtype: object

*##Before we continue let us rename some columns,*

original\_colnames = data.columns

new\_colnames = ['company', 'species', 'REF', 'review\_year', 'cocoa\_p',

'company\_location', 'rating', 'bean\_typ', 'country']

data = data.rename(columns=dict(zip(original\_colnames, new\_colnames)))

*#explore the shape*

data.shape

(1795, 9)

*# Explore description*

data.describe(include='all').T

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **count** | **unique** | **top** | **freq** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **company** | 1795 | 416 | Soma | 47 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **species** | 1795 | 1039 | Madagascar | 57 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **REF** | 1795.0 | NaN | NaN | NaN | 1035.904735 | 552.886365 | 5.0 | 576.0 | 1069.0 | 1502.0 | 1952.0 |
| **Review\_year** | 1795.0 | NaN | NaN | NaN | 2012.325348 | 2.92721 | 2006.0 | 2010.0 | 2013.0 | 2015.0 | 2017.0 |
| **Cocoa\nPercent** | 1795.0 | NaN | NaN | NaN | 0.716983 | 0.063231 | 0.42 | 0.7 | 0.7 | 0.75 | 1.0 |
| **Company\nLocation** | 1795 | 60 | U.S.A. |  | 764 | NaN | NaN | NaN | NaN | NaN | NaN |
| **Rating** | 1795.0 |  | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **Bean\nType** | 1795 | 41 | Â | 887 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **Broad Bean\nOrigin** | 1794 | 100 | Venezuela | 214 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

*## Look at most frequent species*

data['species'].value\_counts().head(10)

Madagascar 57

Peru 45

Ecuador 42

Dominican Republic 37

Venezuela 21

Chuao 19

Sambirano 19

Ocumare 17

Papua New Guinea 15

Ghana 15

Name: species, dtype: int64

*## Is where any N/A values in origin country?*

data['country'].isnull().value\_counts()

False 1794

True 1

Name: country, dtype: int64

*## Replace origin country*

data['country'] = data['country'].fillna(data['species'])

data['country'].isnull().value\_counts()

False 1795

Name: country, dtype: int64

*## Look at most frequent origin countries*

data['country'].value\_counts().head(10)

Venezuela 214

Ecuador 193

Peru 165

Madagascar 146

Dominican Republic 141

Â  73

Nicaragua 60

Brazil 58

Bolivia 57

Belize 49

Name: country, dtype: int64

*## We see that a lot of countries have ' ' value - means that this is 100% blend. Let's look at this*

data[data['country'].str.len()==1]['species'].unique()

array([], dtype=object)

*## Is there another way to determine blends?*

data[data['species'].str.contains(',')]['species'].nunique()

533

*## Is there any misspelling/reduction?*

data['country'].sort\_values().unique()

array(['Africa, Carribean, C. Am.', 'Australia', 'Belize', 'Bolivia',

'Brazil', 'Burma', 'Cameroon', 'Carribean',

'Carribean(DR/Jam/Tri)', 'Central and S. America', 'Colombia',

'Colombia, Ecuador', 'Congo', 'Cost Rica, Ven', 'Costa Rica',

'Cuba', 'DR, Ecuador, Peru', 'Dom. Rep., Madagascar',

'Domincan Republic', 'Dominican Rep., Bali', 'Dominican Republic',

'Ecuador', 'Ecuador, Costa Rica', 'Ecuador, Mad., PNG',

'El Salvador', 'Fiji', 'Gabon', 'Ghana', 'Ghana & Madagascar',

'Ghana, Domin. Rep', 'Ghana, Panama, Ecuador',

'Gre., PNG, Haw., Haiti, Mad', 'Grenada',

'Guat., D.R., Peru, Mad., PNG', 'Guatemala', 'Haiti', 'Hawaii',

'Honduras', 'India', 'Indonesia', 'Indonesia, Ghana',

'Ivory Coast', 'Jamaica', 'Liberia', 'Mad., Java, PNG',

'Madagascar', 'Madagascar & Ecuador', 'Malaysia', 'Martinique',

'Mexico', 'Nicaragua', 'Nigeria', 'PNG, Vanuatu, Mad', 'Panama',

'Papua New Guinea', 'Peru', 'Peru(SMartin,Pangoa,nacional)',

'Peru, Belize', 'Peru, Dom. Rep', 'Peru, Ecuador',

'Peru, Ecuador, Venezuela', 'Peru, Mad., Dom. Rep.',

'Peru, Madagascar', 'Philippines', 'Principe', 'Puerto Rico',

'Samoa', 'Sao Tome', 'Sao Tome & Principe', 'Solomon Islands',

'South America', 'South America, Africa', 'Sri Lanka', 'St. Lucia',

'Suriname', 'Tanzania', 'Tobago', 'Togo', 'Trinidad',

'Trinidad, Ecuador', 'Trinidad, Tobago', 'Trinidad-Tobago',

'Uganda', 'Vanuatu', 'Ven, Bolivia, D.R.',

'Ven, Trinidad, Ecuador', 'Ven., Indonesia, Ecuad.',

'Ven., Trinidad, Mad.', 'Ven.,Ecu.,Peru,Nic.',

'Venez,Africa,Brasil,Peru,Mex', 'Venezuela',

'Venezuela, Carribean', 'Venezuela, Dom. Rep.', 'Venezuela, Ghana',

'Venezuela, Java', 'Venezuela, Trinidad', 'Venezuela/ Ghana',

'Vietnam', 'West Africa', 'Â\xa0'], dtype=object)

*## Text preparation (correction) func*

**def** txt\_prep(text):

replacements = [

['-', ', '], ['/ ', ', '], ['/', ', '], ['\(', ', '], [' and', ', '], [' &', ', '], ['\)', ''],

['Dom Rep|DR|Domin Rep|Dominican Rep,|Domincan Republic', 'Dominican Republic'],

['Mad,|Mad$', 'Madagascar, '],

['PNG', 'Papua New Guinea, '],

['Guat,|Guat$', 'Guatemala, '],

['Ven,|Ven$|Venez,|Venez$', 'Venezuela, '],

['Ecu,|Ecu$|Ecuad,|Ecuad$', 'Ecuador, '],

['Nic,|Nic$', 'Nicaragua, '],

['Cost Rica', 'Costa Rica'],

['Mex,|Mex$', 'Mexico, '],

['Jam,|Jam$', 'Jamaica, '],

['Haw,|Haw$', 'Hawaii, '],

['Gre,|Gre$', 'Grenada, '],

['Tri,|Tri$', 'Trinidad, '],

['C Am', 'Central America'],

['S America', 'South America'],

[', $', ''], [', ', ', '], [', ,', ', '], ['**\xa0**', ' '],[',\s+', ','],

[' Bali', ',Bali']

]

**for** i, j **in** replacements:

text = re.sub(i, j, text)

**return** text

data['country'].str.replace('.', '').apply(txt\_prep).unique()

<ipython-input-19-2093e08cee79>:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will\*not\* be treated as literal strings when regex=True.

data['country'].str.replace('.', '').apply(txt\_prep).unique()

array(['Sao Tome', 'Togo', 'Peru', 'Venezuela', 'Cuba', 'Panama',

'Madagascar', 'Brazil', 'Ecuador', 'Colombia', 'Burma',

'Papua New Guinea', 'Bolivia', 'Fiji', 'Mexico', 'Indonesia',

'Trinidad', 'Vietnam', 'Nicaragua', 'Tanzania',

'Dominican Republic', 'Ghana', 'Belize', 'Â ', 'Jamaica',

'Grenada', 'Guatemala', 'Honduras', 'Costa Rica', 'Haiti', 'Congo',

'Philippines', 'Malaysia', 'Dominican Republic,Bali',

'Venezuela,Africa,Brasil,Peru,Mexico', 'Gabon', 'Ivory Coast',

'Carribean', 'Sri Lanka', 'Puerto Rico', 'Uganda', 'Martinique',

'Sao Tome,Principe', 'Vanuatu', 'Australia', 'Liberia',

'Ecuador,Costa Rica', 'West Africa', 'Hawaii', 'St Lucia',

'Costa Rica,Venezuela', 'Peru,Madagascar', 'Venezuela,Trinidad',

'Trinidad,Tobago', 'Venezuela,Trinidad,Ecuador',

'South America,Africa', 'India',

'Africa,Carribean,Central America', 'Tobago',

'Venezuela,Indonesia,Ecuador', 'Peru,Ecuador,Venezuela',

'Venezuela,Dominican Republic', 'Colombia,Ecuador',

'Solomon Islands', 'Nigeria', 'Peru,Belize',

'Peru,Madagascar,Dominican Republic',

'Papua New Guinea,Vanuatu,Madagascar', 'El Salvador',

'South America', 'Samoa', 'Ghana,Dominican Republic',

'Trinidad,Ecuador', 'Cameroon', 'Venezuela,Java',

'Venezuela,Ghana', 'Indonesia,Ghana',

'Peru,SMartin,Pangoa,nacional', 'Principe',

'Central,South America', 'Venezuela,Trinidad,Madagascar',

'Carribean,Dominican Republic,Jamaica,Trinidad',

'Ghana,Madagascar', 'Venezuela,Ecuador,Peru,Nicaragua',

'Madagascar,Ecuador',

'Guatemala,Dominican Republic,Peru,Madagascar,Papua New Guinea',

'Peru,Dominican Republic', 'Dominican Republic,Madagascar',

'Grenada,Papua New Guinea,Hawaii,Haiti,Madagascar',

'Madagascar,Java,Papua New Guinea',

'Venezuela,Bolivia,Dominican Republic',

'Dominican Republic,Ecuador,Peru', 'Suriname', 'Peru,Ecuador',

'Ecuador,Madagascar,Papua New Guinea', 'Ghana,Panama,Ecuador',

'Venezuela,Carribean'], dtype=object)

*## Looks better*

data['country'].value\_counts().tail(10)

Madagascar & Ecuador 1

Dom. Rep., Madagascar 1

Venezuela, Dom. Rep. 1

Burma 1

Ven, Bolivia, D.R. 1

Africa, Carribean, C. Am. 1

Nigeria 1

Peru(SMartin,Pangoa,nacional) 1

Guat., D.R., Peru, Mad., PNG 1

Peru, Dom. Rep 1

Name: country, dtype: int64

*## Is there any misspelling/reduction in company location?*

data['company\_location'].sort\_values().unique()

array(['Amsterdam', 'Argentina', 'Australia', 'Austria', 'Belgium',

'Bolivia', 'Brazil', 'Canada', 'Chile', 'Colombia', 'Costa Rica',

'Czech Republic', 'Denmark', 'Domincan Republic', 'Ecuador',

'Eucador', 'Fiji', 'Finland', 'France', 'Germany', 'Ghana',

'Grenada', 'Guatemala', 'Honduras', 'Hungary', 'Iceland', 'India',

'Ireland', 'Israel', 'Italy', 'Japan', 'Lithuania', 'Madagascar',

'Martinique', 'Mexico', 'Netherlands', 'New Zealand', 'Niacragua',

'Nicaragua', 'Peru', 'Philippines', 'Poland', 'Portugal',

'Puerto Rico', 'Russia', 'Sao Tome', 'Scotland', 'Singapore',

'South Africa', 'South Korea', 'Spain', 'St. Lucia', 'Suriname',

'Sweden', 'Switzerland', 'U.K.', 'U.S.A.', 'Venezuela', 'Vietnam',

'Wales'], dtype=object)

*## We need to make some replacements*

data['company\_location'] = data['company\_location']\

.str.replace('Amsterdam', 'Holland')\

.str.replace('U.K.', 'England')\

.str.replace('Niacragua', 'Nicaragua')\

.str.replace('Domincan Republic', 'Dominican Republic')

data['company\_location'].sort\_values().unique()

<ipython-input-22-1da09fd50d69>:2: FutureWarning: The default value of regex will change from True to False in a future version.

data['company\_location'] = data['company\_location']\

Out[22]:

array(['Argentina', 'Australia', 'Austria', 'Belgium', 'Bolivia',

'Brazil', 'Canada', 'Chile', 'Colombia', 'Costa Rica',

'Czech Republic', 'Denmark', 'Dominican Republic', 'Ecuador',

'England', 'Eucador', 'Fiji', 'Finland', 'France', 'Germany',

'Ghana', 'Grenada', 'Guatemala', 'Holland', 'Honduras', 'Hungary',

'Iceland', 'India', 'Ireland', 'Israel', 'Italy', 'Japan',

'Lithuania', 'Madagascar', 'Martinique', 'Mexico', 'Netherlands',

'New Zealand', 'Nicaragua', 'Peru', 'Philippines', 'Poland',

'Portugal', 'Puerto Rico', 'Russia', 'Sao Tome', 'Scotland',

'Singapore', 'South Africa', 'South Korea', 'Spain', 'St. Lucia',

'Suriname', 'Sweden', 'Switzerland', 'U.S.A.', 'Venezuela',

'Vietnam', 'Wales'], dtype=object)

In [23]:

*## Is there any misspelling/reduction in company name?*

data['company'].str.lower().sort\_values().nunique() == data['company'].sort\_values().nunique()

Out[23]:

True

**Data Visualization**

*# what is the greatest amount of chocolate used in terms of percentage?*

fig, ax = plt.subplots(figsize=[16,4])

sns.histplot(data['cocoa\_p'], ax=ax)

ax.set\_title('Cocoa %, Distribution')

plt.show()

*#what is the highest rating in the dataset?*

fig, ax = plt.subplots(figsize=[16,4])

sns.histplot(data['rating'], ax=ax)

ax.set\_title('Rating, Distribution')

plt.show()

*#which company is the highest producer of the chocolate?*

fig, ax = plt.subplots(figsize=[16,4])

sns.histplot(data['company'], ax=ax)

ax.set\_title('Company Production Distribution')

plt.show()

*## Look at boxplot over the countries, even Blends*

fig, ax = plt.subplots(figsize=[6, 16])

sns.boxplot(

data=data,

y='country',

x='rating'

)

ax.set\_title('Boxplot, Rating for countries (+blends)')

Text(0.5, 1.0, 'Boxplot, Rating for countries (+blends)')

*## But how can we see what country is biggest contributor in rating?*

data\_ = pd.concat([pd.Series(row['rating'], row['country'].split(',')) **for** \_, row **in** data.iterrows()]

).reset\_index()

data\_.columns = ['country', 'rating']

data\_['mean\_rating'] = data\_.groupby(['country'])['rating'].transform('mean')

*## Look at boxplot over the countries (contributors in blends)*

fig, ax = plt.subplots(figsize=[6, 16])

sns.boxplot(

data=data\_.sort\_values('mean\_rating', ascending=**False**),

y='country',

x='rating'

)

ax.set\_title('Boxplot, Rating for countries (contributors)')

Text(0.5, 1.0, 'Boxplot, Rating for countries (contributors)')

*#what country is the biggest contributor in rating?*

data\_.groupby(['country'])['rating'].mean().sort\_values(ascending=**False**).head(10)

country

Haw. 4.000

Gre. 4.000

Haiti 4.000

Dom. Rep. 4.000

D.R. 4.000

Dom. Rep 4.000

Guat. 4.000

Bolivia 4.000

Peru 3.875

Mad 3.875

Name: rating, dtype: float64

*#what is the relationship between the rating, REF, the review and the cocoa percentage?*

correlation = data.corr()

sns.heatmap(correlation, xticklabels =correlation.columns, yticklabels= correlation.columns

, annot=**True**)

<AxesSubplot:>

*## Look at rating in terms of company location*

fig, ax = plt.subplots(figsize=[6, 16])

sns.boxplot(

data=data,

y='company\_location',

x='rating'

)

ax.set\_title('Boxplot, Rating by Company location')

Text(0.5, 1.0, 'Boxplot, Rating by Company location')

# Conclusions

* Over time the average rating isn’t increasing but the instances of extreme ratings is decreasing, resulting in a smaller spread of ratings
* The country of manufacture doesn’t appear to have a large impact on rating, but both the maker and company can be seen to impact the average rating of a chocolate
* There isn’t a strong relationship between the cocoa percent and rating of the chocolate. Fitting a linear model showed us that we would need to include more features to build a good prediction model
  + The addition of year to the model as a factor improved the model, but it is still poor
  + Using the trends spotted in our EDA, the inclusion of company location improved our predictive power further
  + A random forest model seems to perform better overall than a simple linear model
* Broad bean origin seems to be well represented by the big five:
  + Venezuela
  + Ecuador
  + Peru
  + Madagascar
  + Dominican Republic
* However, the broad bean origin of the bean doesn’t seem to determine the quality, or rating, of the chocolate