Chocolate Bars Analysis

Markdown Guides

> This is a blockquote.

Some of these words *are emphasized*.

Use two asterisks for **strong emphasis**.

- Another item in the list.

This is an example link.

$x = x + y$

text to appear as link

Images inline image

Project Description

As part of the data science team at Gourmet Analytics, you use data analytics to advise companies in the food industry. You clean, organize, and visualize data to arrive at insights that will benefit your clients. As a member of a collaborative team, sharing your analysis with others is an important part of your job.

Your current client is Chocolate and Tea, an up-and-coming chain of cafés.

The eatery combines an extensive menu of fine teas with chocolate bars from around the world. Their diverse selection includes everything from plantain milk chocolate, to tangerine white chocolate, to dark chocolate with pistachio and fig. The encyclopedic list of chocolate bars is the basis of Chocolate and Tea's brand appeal. Chocolate bar sales are the main driver of revenue.

Chocolate and Tea aims to serve chocolate bars that are highly rated by professional critics. They also continually adjust the menu to make sure it reflects the global diversity of chocolate production. The management team regularly updates the chocolate bar list in order to align with the latest ratings and to ensure that the list contains bars from a variety of countries.

They've asked you to collect and analyze data on the latest chocolate ratings. In particular, they'd like to know which countries produce the highest-rated bars of super dark chocolate (at least 80% cocoa). This data will help them create their next chocolate bar menu.

Your team has received a dataset that features the latest ratings for thousands of chocolates from around the world. Click here to access the dataset. Given the data and the nature of the work you will do for your client, your team agrees to use R for this project.

Data Dictionary

Field	Description
Company (Maker-if known)	Name of the company manufacturing the bar.
Specific Bean Originor Bar Name	The specific geo-region of origin for the bar.
REF	A value linked to when the review was entered in the database. Higher = more recent.
ReviewDate	Date of publication of the review.
CocoaPercent	Cocoa percentage (darkness) of the chocolate bar being reviewed.
CompanyLocation	Manufacturer base country.
Rating	Expert rating for the bar.
BeanType	The variety (breed) of bean used, if provided.
Broad BeanOrigin	The broad geo-region of origin for the bean.

Data sources used

Chocolate Bar Ratings dataset from Kaggle. 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.

Business Task

Analyze chocolate bars to find best ratings and recommend which ones for sales. Chocolate Bar Ratings dataset containing 1700 versions are used.

Metrics Used

Only chocolate bars with minimum ratings 4.0 and above will be considered and 80% cocoa content

Assumptions

All data is normally distributed.

Data Tasks

- Load data and perform simple statistical analysis
- Use Groupings to look at mean ratings
- Cleaning data work in filling missing values, check duplicates and outliers
- Perform hypothesis testing
- Plot graphs
- Focus on company requirements

Summary

Only two companies qualified for supplying high quality chocolate bars: **Pralus and Soma**

Recommendation for Action

- Either reduce the rating to 3.75 for more suppliers to chose
- Search for other datasets on chocolate bar ratings

Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import random

import statsmodels.api as sm
from statsmodels.formula.api import ols

import datetime
from datetime import datetime, timedelta

import scipy.stats

import pandas_profiling
from pandas_profiling import ProfileReport

%matplotlib inline
#sets the default autosave frequency in seconds
%autosave 60
sns.set_style('dark')
sns.set(font_scale=1.2)

plt.rc('axes', titlesize=9)
plt.rc('axes', labelsize=14)
plt.rc('xtick', labelsize=12)
plt.rc('ytick', labelsize=12)

import warnings
warnings.filterwarnings('ignore')

# Use Folium library to plot values on a map.
#import folium

# Use Feature-Engine library
#import feature_engine
#import feature_engine.missing_data_imputers as mdi
#from feature_engine.outlier_removers import Winsorizer
#from feature_engine import categorical_encoders as ce
#from feature_engine.discretization import EqualWidthDiscretiser, EqualFrequencyDiscretiser, DecisionTreeDis
#from feature_engine.encoding import OrdinalEncoder

pd.set_option('display.max_columns',None)
#pd.set_option('display.max_rows',None)
pd.set_option('display.width', 1000)
pd.set_option('display.float_format','{:.2f}'.format)

random.seed(0)
np.random.seed(0)
np.set_printoptions(suppress=True)
```

Autosaving every 60 seconds

Exploratory Data Analysis

In [2]:
```
df = pd.read_csv("flavors_of_cacao.csv")
```

In [3]:
```
df
```

Out[3]:

	Company	SpecificBeanOrigin	REF	ReviewDate	CocoaPercent	CompanyLocation	Rating	BeanType	BroadBeanOrigin
0	A. Morin	Agua Grande	1876	2016	0.63	France	3.75	NaN	Sao Tome
1	A. Morin	Kpime	1676	2015	0.70	France	2.75	NaN	Togo
2	A. Morin	Atsane	1676	2015	0.70	France	3.00	NaN	Togo
3	A. Morin	Akata	1680	2015	0.70	France	3.50	NaN	Togo
4	A. Morin	Quilla	1704	2015	0.70	France	3.50	NaN	Peru
...	...	...	...	...	...	...	...	...	...
1790	Zotter	Peru	647	2011	0.70	Austria	3.75	NaN	Peru
1791	Zotter	Congo	749	2011	0.65	Austria	3.00	Forastero	Congo
1792	Zotter	Kerala State	749	2011	0.65	Austria	3.50	Forastero	India
1793	Zotter	Kerala State	781	2011	0.62	Austria	3.25	NaN	India
1794	Zotter	Brazil, Mitzi Blue	486	2010	0.65	Austria	3.00	NaN	Brazil

1795 rows × 9 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1795 entries, 0 to 1794
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Company             1795 non-null   object
 1   SpecificBeanOrigin  1795 non-null   object
 2   REF                 1795 non-null   int64
 3   ReviewDate          1795 non-null   int64
 4   CocoaPercent        1795 non-null   float64
 5   CompanyLocation     1795 non-null   object
 6   Rating              1795 non-null   float64
 7   BeanType            907 non-null    object
 8   BroadBeanOrigin     1721 non-null   object
dtypes: float64(2), int64(2), object(5)
memory usage: 126.3+ KB
```

In [5]:
```
df.describe()
```

Out[5]:

	REF	ReviewDate	CocoaPercent	Rating
count	1795.00	1795.00	1795.00	1795.00
mean	1035.90	2012.33	0.72	3.19
std	552.89	2.93	0.06	0.48
min	5.00	2006.00	0.42	1.00
25%	576.00	2010.00	0.70	2.88
50%	1069.00	2013.00	0.70	3.25
75%	1502.00	2015.00	0.75	3.50
max	1952.00	2017.00	1.00	5.00

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Company', 'SpecificBeanOrigin', 'REF', 'ReviewDate', 'CocoaPercent', 'CompanyLocation', 'Rating', 'B
eanType', 'BroadBeanOrigin'], dtype='object')
```

Groupby Function

In [7]:
```
df.groupby("Company")["CocoaPercent","Rating"].mean()
```

Out[7]:

Company	CocoaPercent	Rating
A. Morin	0.69	3.38
AMMA	0.69	3.50
Acalli	0.70	3.75
Adi	0.75	3.25
Aequare (Gianduja)	0.62	2.88
...	...	...
hello cocoa	0.66	3.12
hexx	0.71	3.05
iQ Chocolate	0.72	2.88
organicfair	0.72	2.85
twenty-four blackbirds	0.73	3.04

416 rows × 2 columns

In [8]:
```
df.groupby("SpecificBeanOrigin")["CocoaPercent","Rating"].mean()
```

Out[8]:

SpecificBeanOrigin	CocoaPercent	Rating
"heirloom", Arriba Nacional	0.79	3.50
100 percent	1.00	1.50
2009 Hapa Nibby	0.70	3.75
A case of the Xerces Blues, triple roast	0.70	3.50
ABOCFA Coop	0.70	4.00
...	...	...
black label	0.72	3.00
la Amistad	0.70	3.50
one hundred	1.00	2.00
single estate	0.80	2.75
the lost city, gracias a dias, batch 362	0.72	3.50

1039 rows × 2 columns

In [9]:
```
df.groupby("ReviewDate")["CocoaPercent","Rating"].mean()
```

Out[9]:

ReviewDate	CocoaPercent	Rating
2006	0.71	3.12
2007	0.72	3.16
2008	0.73	2.99
2009	0.70	3.07
2010	0.71	3.15
2011	0.71	3.26
2012	0.72	3.18
2013	0.72	3.20
2014	0.72	3.19
2015	0.72	3.25
2016	0.72	3.23
2017	0.72	3.31

In [10]:
```
df.groupby("CompanyLocation")["CocoaPercent","Rating"].mean()
```

Out[10]:

CompanyLocation	CocoaPercent	Rating
Amsterdam	0.73	3.50
Argentina	0.73	3.31
Australia	0.70	3.36
Austria	0.72	3.24
Belgium	0.72	3.09
Bolivia	0.73	3.25
Brazil	0.70	3.40
Canada	0.72	3.32
Chile	0.70	3.75
Colombia	0.66	3.17
Costa Rica	0.72	3.14
Czech Republic	0.63	2.75
Denmark	0.70	3.28
Domincan Republic	0.67	3.20
Ecuador	0.72	3.01
Eucador	0.75	3.00
Fiji	0.75	3.25
Finland	0.70	3.25
France	0.72	3.25
Germany	0.73	3.18
Ghana	0.80	2.75
Grenada	0.71	2.83
Guatemala	0.70	3.35
Honduras	0.75	3.21
Hungary	0.72	3.20
Iceland	0.69	3.42
India	0.61	2.50
Ireland	0.79	2.81
Israel	0.71	3.25
Italy	0.71	3.33
Japan	0.71	3.09
Lithuania	0.72	3.17
Madagascar	0.72	3.15
Martinique	0.42	2.75
Mexico	0.72	2.69
Netherlands	0.70	3.50
New Zealand	0.73	3.19
Niacragua	0.75	2.75
Nicaragua	0.72	3.00
Peru	0.69	2.90
Philippines	0.65	3.50
Poland	0.72	3.38
Portugal	0.76	2.75
Puerto Rico	0.71	2.62
Russia	0.70	3.25
Sao Tome	0.82	2.94
Scotland	0.73	3.33
Singapore	0.71	3.33
South Africa	0.71	2.67
South Korea	0.70	3.20
Spain	0.73	3.27
St. Lucia	0.65	3.00
Suriname	0.70	3.25
Sweden	0.70	3.05
Switzerland	0.71	3.34
U.K.	0.74	3.05
U.S.A.	0.72	3.15
Venezuela	0.69	3.17
Vietnam	0.75	3.41
Wales	0.80	2.75

In [11]:
```
df.groupby("BeanType")["CocoaPercent","Rating"].mean()
```

Out[11]:

BeanType	CocoaPercent	Rating
Amazon	0.70	3.25
Amazon mix	0.74	3.75
Amazon, ICS	0.68	3.62
Beniano	0.71	3.58
Blend	0.71	3.35
Blend-Forastero,Criollo	0.70	3.75
CCN51	0.65	3.50
Criollo	0.72	3.24
Criollo (Amarru)	0.70	3.25
Criollo (Ocumare 61)	0.73	3.25
Criollo (Ocumare 67)	0.70	4.00
Criollo (Ocumare 77)	0.70	3.75
Criollo (Ocumare)	0.80	3.00
Criollo (Porcelana)	0.70	3.38
Criollo (Wild)	0.68	4.00
Criollo, +	0.65	3.50
Criollo, Forastero	0.77	3.62
Criollo, Trinitario	0.72	3.29
EET	0.71	3.58
Forastero	0.72	3.10
Forastero (Amelonado)	0.70	3.75
Forastero (Arriba)	0.73	2.83
Forastero (Arriba) ASS	0.69	2.88
Forastero (Arriba) ASSS	0.70	3.50
Forastero (Catongo)	0.72	3.38
Forastero (Nacional)	0.73	3.27
Forastero (Parazinho)	0.69	3.53
Forastero(Arriba, CCN)	0.75	3.00
Forastero, Trinitario	0.70	3.00
Matina	0.74	3.42
Nacional	0.71	3.00
Nacional (Arriba)	0.73	3.25
Trinitario	0.72	3.25
Trinitario (85% Criollo)	0.70	3.88
Trinitario (Amelonado)	0.70	3.00
Trinitario (Scavina)	0.70	3.50
Trinitario, Criollo	0.68	3.03
Trinitario, Forastero	0.71	3.00
Trinitario, Nacional	0.70	3.75
Trinitario, TCGA	0.72	3.75

In [12]:
```
df.groupby("BroadBeanOrigin")["CocoaPercent","Rating"].mean()
```

Out[12]:

BroadBeanOrigin	CocoaPercent	Rating
Africa, Carribean, C. Am.	0.74	2.75
Australia	0.69	3.25
Belize	0.73	3.23
Bolivia	0.72	3.20
Brazil	0.71	3.28
...	...	...
Venezuela, Java	0.70	4.00
Venezuela, Trinidad	0.62	2.50
Venezuela/ Ghana	0.85	3.50
Vietnam	0.74	3.32
West Africa	0.73	2.58

99 rows × 2 columns

In [13]:
```
df.Rating.value_counts()
```

Out[13]:
```
3.50    392
3.00    341
3.25    303
2.75    259
3.75    210
2.50    127
4.00     98
2.00     32
2.25     14
1.50     10
1.00      4
1.75      3
5.00      2
Name: Rating, dtype: int64
```

Pandas-Profiling Reports

In [14]:
```
#profile = ProfileReport(df=df, title='Cocao Report', minimal=True)
```

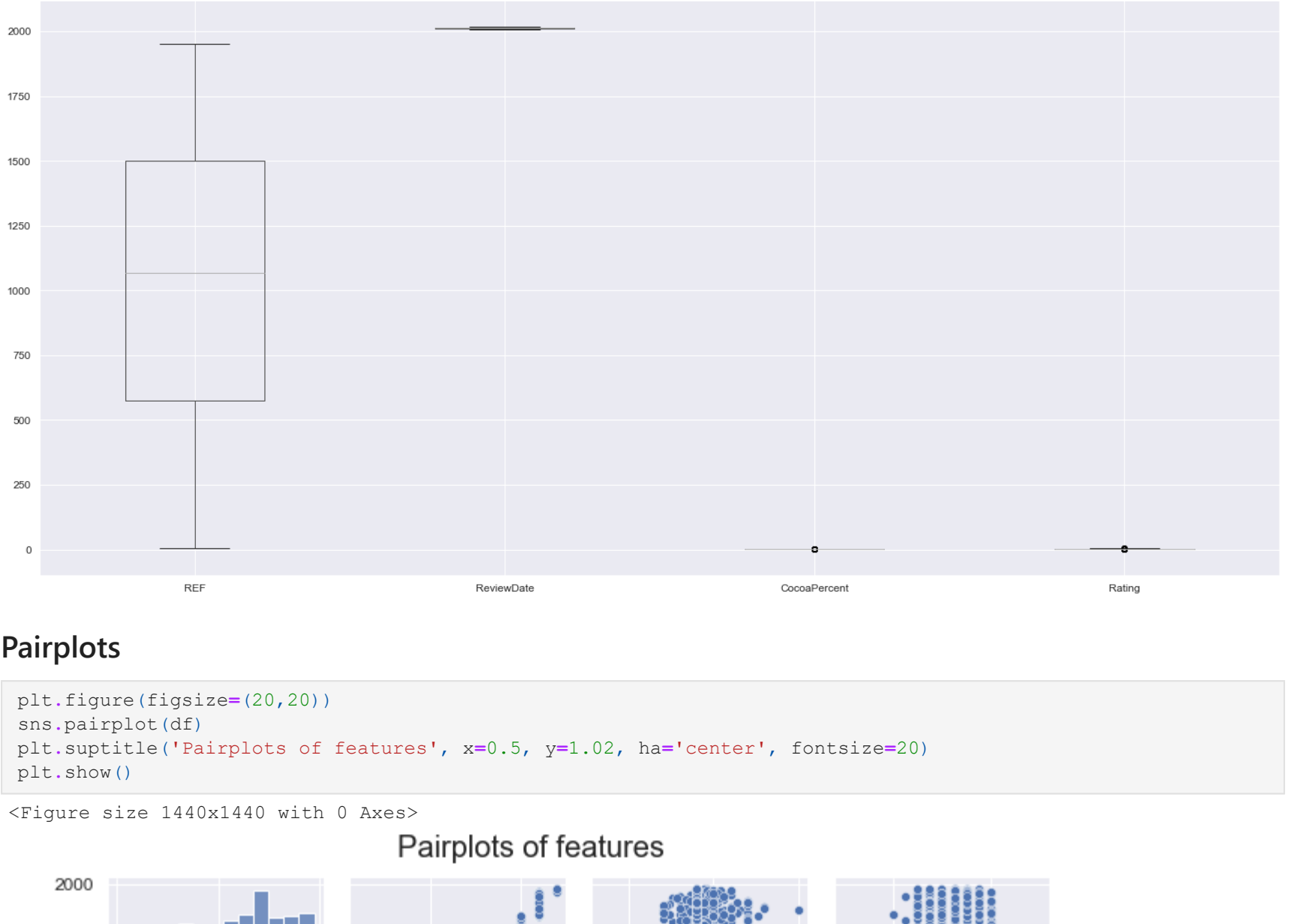
In [15]:
```
#profile.to_notebook_iframe()
```

In [16]:
```
#profile.to_file("Cocao_Report.html")
```

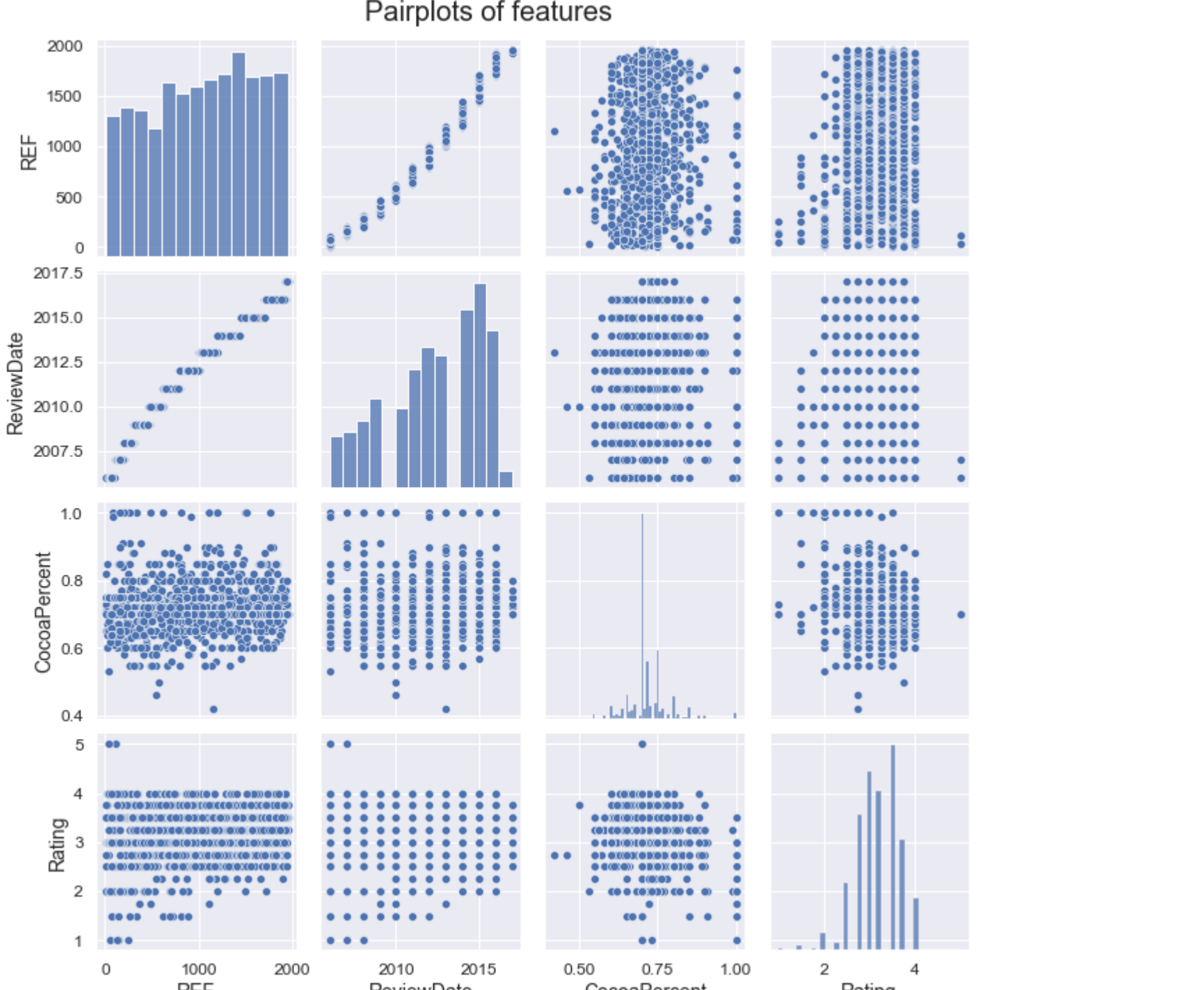
Data Visualization

Univariate Data Exploration

In [17]:
```
df.hist(bins=50, figsize=(20,10))
plt.suptitle('Feature Distribution', x=0.5, y=1.02, ha='center', fontsize=20)
plt.tight_layout()
plt.show()
```



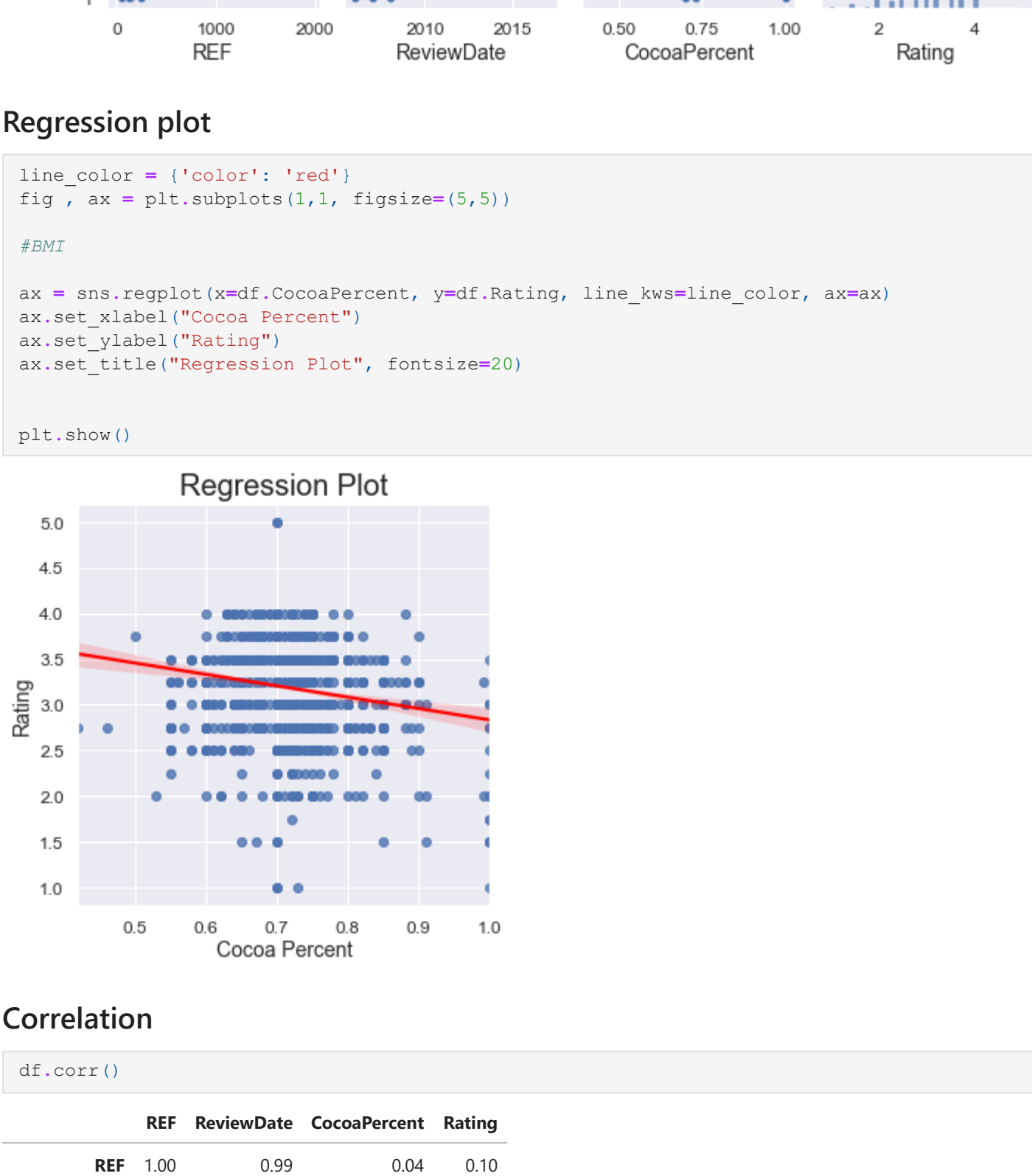
In [18]:
```
df.boxplot(figsize=(20,10))
plt.suptitle('BoxPlot', x=0.5, y=1.02, ha='center', fontsize=20)
plt.tight_layout()
plt.show()
```



Pairplots

In [19]:
```
plt.figure(figsize=(20,20))
sns.pairplot(df)
plt.suptitle('Pairplots of features', x=0.5, y=1.02, ha='center', fontsize=20)
plt.show()
```

```
<Figure size 1440x1440 with 0 Axes>
```



Regression plot

In [20]:
```
line_color = {'color': 'red'}
fig , ax = plt.subplots(1,1, figsize=(5,5))

#BMI

ax = sns.regplot(x=df.CocoaPercent, y=df.Rating, line_kws=line_color, ax=ax)
ax.set_xlabel("Cocoa Percent")
ax.set_ylabel("Rating")
ax.set_title("Regression Plot", fontsize=20)


plt.show()
```

Correlation

In [21]:
```
df.corr()
```

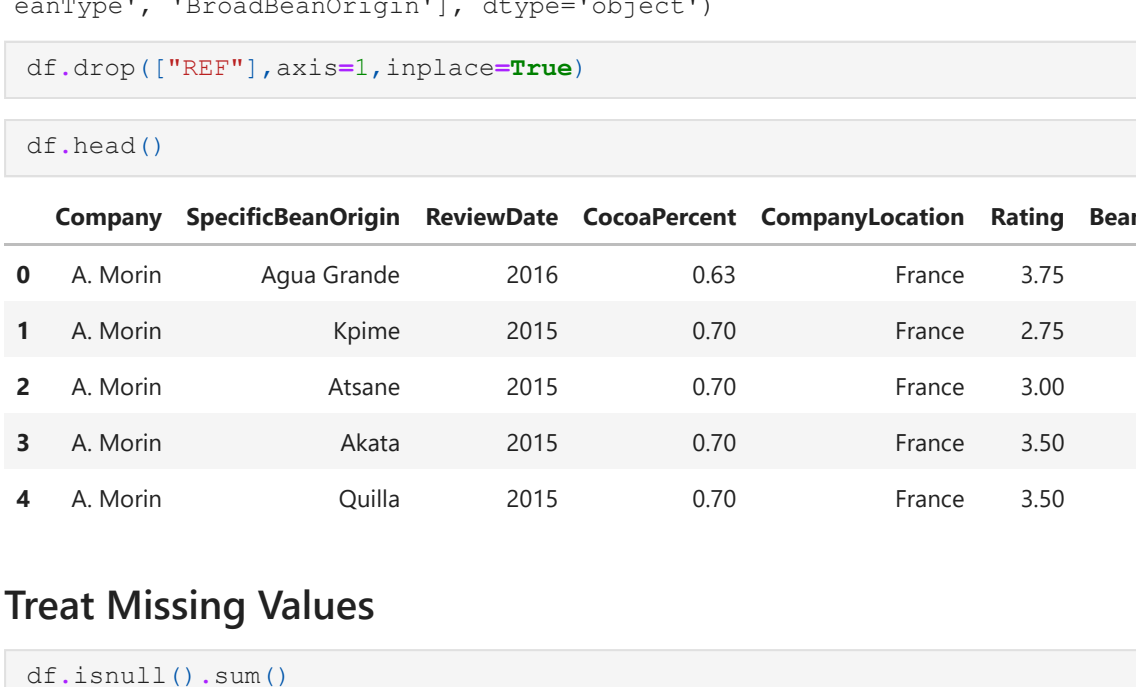
Out[21]:

	REF	ReviewDate	CocoaPercent	Rating
REF	1.00	0.99	0.04	0.10
ReviewDate	0.99	1.00	0.04	0.10
CocoaPercent	0.04	0.04	1.00	-0.17
Rating	0.10	0.10	-0.17	1.00

In [22]:
```
df.corr()["Rating"].sort_values()
```

Out[22]:
```
CocoaPercent   -0.17
ReviewDate      0.10
REF             0.10
Rating          1.00
Name: Rating, dtype: float64
```

In [23]:
```
plt.figure(figsize=(9,5))
sns.heatmap(df.corr(),cmap="coolwarm",annot=True,fmt='.2f',linewidths=2)
plt.title("", fontsize=20)
plt.show()
```



Data Preprocessing

In [24]:
```
df.head()
```

Out[24]:

	Company	SpecificBeanOrigin	REF	ReviewDate	CocoaPercent	CompanyLocation	Rating	BeanType	BroadBeanOrigin
0	A. Morin	Agua Grande	1876	2016	0.63	France	3.75	NaN	Sao Tome
1	A. Morin	Kpime	1676	2015	0.70	France	2.75	NaN	Togo
2	A. Morin	Atsane	1676	2015	0.70	France	3.00	NaN	Togo
3	A. Morin	Akata	1680	2015	0.70	France	3.50	NaN	Togo
4	A. Morin	Quilla	1704	2015	0.70	France	3.50	NaN	Peru

Drop unwanted features

In [25]:
```
df.columns
```

Out[25]:
```
Index(['Company', 'SpecificBeanOrigin', 'REF', 'ReviewDate', 'CocoaPercent', 'CompanyLocation', 'Rating', 'B
eanType', 'BroadBeanOrigin'], dtype='object')
```

In [26]:
```
df.drop(["REF"],axis=1,inplace=True)
```

In [27]:
```
df.head()
```

Out[27]:

	Company	SpecificBeanOrigin	ReviewDate	CocoaPercent	CompanyLocation	Rating	BeanType	BroadBeanOrigin
0	A. Morin	Agua Grande	2016	0.63	France	3.75	NaN	Sao Tome
1	A. Morin	Kpime	2015	0.70	France	2.75	NaN	Togo
2	A. Morin	Atsane	2015	0.70	France	3.00	NaN	Togo
3	A. Morin	Akata	2015	0.70	France	3.50	NaN	Togo
4	A. Morin	Quilla	2015	0.70	France	3.50	NaN	Peru

Treat Missing Values

In [28]:
```
df.isnull().sum()
```

Out[28]:
```
Company                 0
SpecificBeanOrigin      0
ReviewDate              0
CocoaPercent            0
CompanyLocation         0
Rating                  0
BeanType              888
BroadBeanOrigin        74
dtype: int64
```

In [29]:
```
df['BeanType'] = df['BeanType'].replace(np.nan,"Missing")
```

In [30]:
```
df['BroadBeanOrigin'] = df['BroadBeanOrigin'].replace(np.nan,"Missing")
```

In [31]:
```
df.isnull().sum()
```

Out[31]:
```
Company               0
SpecificBeanOrigin    0
ReviewDate            0
CocoaPercent          0
CompanyLocation       0
Rating                0
BeanType              0
BroadBeanOrigin       0
dtype: int64
```

In [32]:
```
df
```

Out[32]:

	Company	SpecificBeanOrigin	ReviewDate	CocoaPercent	CompanyLocation	Rating	BeanType	BroadBeanOrigin
0	A. Morin	Agua Grande	2016	0.63	France	3.75	Missing	Sao Tome
1	A. Morin	Kpime	2015	0.70	France	2.75	Missing	Togo
2	A. Morin	Atsane	2015	0.70	France	3.00	Missing	Togo
3	A. Morin	Akata	2015	0.70	France	3.50	Missing	Togo
4	A. Morin	Quilla	2015	0.70	France	3.50	Missing	Peru
...	...	...	...	...	...	...	...	...
1790	Zotter	Peru	2011	0.70	Austria	3.75	Missing	Peru
1791	Zotter	Congo	2011	0.65	Austria	3.00	Forastero	Congo
1792	Zotter	Kerala State	2011	0.65	Austria	3.50	Forastero	India
1793	Zotter	Kerala State	2011	0.62	Austria	3.25	Missing	India
1794	Zotter	Brazil, Mitzi Blue	2010	0.65	Austria	3.00	Missing	Brazil

1795 rows × 8 columns

Treat Duplicate Values