Assemble the Data

Project Description

After having major success on the European market, the board of Fond Rouge, a shoe company from France, decides it is time to venture to the other side of the pond and take on North America's footwear industry.

Initially, everything goes according to plan and they even manage to scale the retail locations faster than expected. In the optimism of this continuous progress, though, the company's business leads have missed an important trend that started to arise in their data. After the Marketing department of Fond Rouge noticed a stream of negative reviews online (primarily on social media) - they realised there is a problem.

Today, you are in the meeting room with Fond Rouge's VP of BizDev and Head of Engineering. They believe there are counterfeit products of lesser quality interfering with the US sales and resulting in negative reviews online. Fond Rouge would like us to consult them on the best way to narrow the affected locations, identify the problem(s) and propose a solution.

Business Task

Here is what your first task looks like:

Setup a standard office package software. If you already have something like MS Office (Excel, Word & PPT), OpenOffice or LibreOffice installed - then you can skip this step. If not - please find a link to an installation guide for LibreOffice in the Resources section.

Get acquainted with the most common data types clients like Fond Rouge would usually collect. In the Resources section below, you will find samples of several different types of data (these are only samples of the formats Fond Rouge stores data in, please don't try to solve the client's problem based only on this data).

- Sales - example for sales transaction
- Returns - example of order return transactions
- Sentiment - a score of 0-100 (0-64 is considered negative, 65-84 neutral, 85-100 positive) generated from a combination of the NPS the client measures + sentiment scores of public mentions on social media (like twitter)
- Server Errors - a sample of the server error log

Determine what's the right data to help you troubleshoot the client's problem.

Take a quiz to solidify your learnings.

Recommendation for Action

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
import statsmodels.formula.api as smf
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')

#Webscraping
#import requests
#from bs4 import BeautifulSoup

# 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.discretisation import EqualWidthDiscretiser, EqualFrequencyDiscretiser, DecisionTreeDiscre
#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]:

```
sales_df = pd.read_csv("Sales_sample_data.csv", parse_dates=["Date"])
```

In [3]:

```
sales_df.head()
```

Out[3]:

	OrderID	Date	Country	City	Latitude	Longtitude	ProductCategoryName	ProductID	Product	Quantity	Discount	Pri
0	order-fd7c3c2a-f481-4d58-91b6-06a9e2de0724	2020-11-07	Denmark	Copenhagen	55.68	12.56	Platforms	product-b7c06d0a-977d-497b-ae3e-95b58985cafd	Amélie	7.00	2.65	40.
1	order-74ff7daf-2d0a-4dae-b132-58a12fc42d24	2020-11-07	Denmark	Copenhagen	55.68	12.56	Stilettos	product-7bef3e02-033c-4259-93da-a25f4f7169be	Claudette	5.00	6.21	102.
2	order-0587ebd4-4a63-45f8-8d86-a8d2e814a10a	2020-11-19	Denmark	Copenhagen	55.68	12.56	Platforms	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	2.00	0.58	39.
3	order-bf7d5289-d4c1-4cca-b054-e4a4817eb8f8	2020-11-07	Denmark	Copenhagen	55.68	12.56	Platforms	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	10.00	1.70	46.
4	order-1c0c517f-2d62-4278-940b-f8023b19a4e3	2020-11-15	Denmark	Copenhagen	55.68	12.56	Stilettos	product-fa4a41fc-4a31-44b5-953f-8e2a45b43673	Cecile	1.00	1.03	78.

In [4]:

```
sales_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 168 entries, 0 to 167
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   OrderID              99 non-null     object
 1   Date                 99 non-null     datetime64[ns]
 2   Country              99 non-null     object
 3   City                 99 non-null     object
 4   Latitude             99 non-null     float64
 5   Longtitude           99 non-null     float64
 6   ProductCategoryName  99 non-null     object
 7   ProductID            99 non-null     object
 8   Product              99 non-null     object
 9   Quantity             99 non-null     float64
 10  Discount             99 non-null     float64
 11  Price                99 non-null     float64
 12  Status               99 non-null     object
dtypes: datetime64[ns](1), float64(5), object(7)
memory usage: 17.2+ KB
```

In [5]:

```
sales_df.describe(include='all')
```

Out[5]:

	OrderID	Date	Country	City	Latitude	Longtitude	ProductCategoryName	ProductID	Product	Quantity	Discount	Price
count	99	99	99	99	99.00	99.00	99	99	99	99.00	99.00	99.0
unique	99	84	3	4	NaN	NaN	4	12	12	NaN	NaN	NaN
top	order-b0639802-c23d-49f7-81e5-65c576fd5d9b	2019-11-11 00:00:00	United States	New York	NaN	NaN	Brogues	product-20700833-fc84-4340-9a59-669fe6acc94b	Antoine	NaN	NaN	NaN
freq	1	3	48	48	NaN	NaN	28	12	12	NaN	NaN	NaN
first	NaN	2019-02-10 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
last	NaN	2021-02-28 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	47.39	-32.82	NaN	NaN	NaN	5.66	2.49	55.11
std	NaN	NaN	NaN	NaN	6.69	40.36	NaN	NaN	NaN	2.76	1.91	20.4
min	NaN	NaN	NaN	NaN	40.69	-73.92	NaN	NaN	NaN	1.00	0.07	35.9
25%	NaN	NaN	NaN	NaN	40.69	-73.92	NaN	NaN	NaN	3.00	0.99	40.0
50%	NaN	NaN	NaN	NaN	51.51	-2.98	NaN	NaN	NaN	5.00	2.13	45.95
75%	NaN	NaN	NaN	NaN	54.54	6.22	NaN	NaN	NaN	8.00	3.50	68.7
max	NaN	NaN	NaN	NaN	55.68	12.56	NaN	NaN	NaN	10.00	10.15	102.9

In [6]:

```
sales_df.columns
```

Out[6]:

```
Index(['OrderID', 'Date', 'Country', 'City', 'Latitude', 'Longtitude', 'ProductCategoryName', 'ProductID', 'Pro
duct', 'Quantity', 'Discount', 'Price', 'Status'], dtype='object')
```

In [7]:

```
sales_df.groupby('Status').mean()
```

Out[7]:

	Latitude	Longtitude	Quantity	Discount	Price
Status					
COMPLETED	47.39	-32.82	5.66	2.49	55.11

In [8]:

```
return_df = pd.read_csv("Returns sample.csv")
```

In [9]:

```
return_df.head()
```

Out[9]:

	OrderID	Status
0	order-8cacb48a-6a1e-42de-abe2-cf6092a48af2	RETURNED
1	order-f696bafe-acd9-4985-a4ad-05da86cc8268	RETURNED
2	order-3c18c650-b023-4b3e-bc03-e6ae455329ff	RETURNED
3	order-6b940800-f65e-4717-ac18-bd12e0e91400	RETURNED
4	order-766719c3-2fe4-4fc9-8d44-26c10a575ea2	RETURNED

In [10]:

```
df3 = pd.merge(left=sales_df,right=return_df, on="OrderID", how='inner')
```

In [11]:

```
df3
```

Out[11]:

	OrderID	Date	Country	City	Latitude	Longtitude	ProductCategoryName	ProductID	Product	Quantity	Discount	Price
0	order-fc3e5cf3-267e-49fa-af66-f9f86c341d1c	2020-11-09	Denmark	Copenhagen	55.68	12.56	Platforms	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	3.00	0.30	46.93
1	order-68b41bfc-ab5f-461f-bbfd-1b4576e79fc2	2020-11-20	Denmark	Copenhagen	55.68	12.56	Platforms	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	9.00	3.75	39.95

In [12]:

```
sentiment_df = pd.read_csv("Sentiment_sample_data.csv", parse_dates=["Year_Month"])
```

In [13]:

```
sentiment_df.head()
```

Out[13]:

	Year_Month	Location_ID	ProductID	Product	Sentiment	Class
0	2019-03-01	Belgium, Brussels	product-20700833-fc84-4340-9a59-669fe6acc94b	Antoine	92	POS
1	2019-03-01	Belgium, Brussels	product-124ef52a-c7c3-48af-b315-33a14b2f6e1d	François	87	POS
2	2019-03-01	Belgium, Brussels	product-a19d1434-d5f2-4a2a-9fe0-7d70f63e391e	Denis	83	NEU
3	2019-03-01	Belgium, Brussels	product-9f6a916a-271c-4d78-9e5f-f802bbcf6548	Adele	87	POS
4	2019-03-01	Belgium, Brussels	product-f709c12a-ffe5-48b1-a3d2-f247acf8e176	Danielle	90	POS

In [14]:

```
sentiment_df.groupby("Class").mean()
```

Out[14]:

	Sentiment
Class	
NEG	44.07
NEU	83.36
POS	88.14

In [15]:

```
negative_sentiment_df = sentiment_df[sentiment_df["Class"]== "NEG"]
```

In [16]:

```
negative_sentiment_df
```

Out[16]:

	Year_Month	Location_ID	ProductID	Product	Sentiment	Class
69	2020-12-01	United States, San Diego	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	40	NEG
70	2020-12-01	United States, San Diego	product-b7c06d0a-977d-497b-ae3e-95b58985cafd	Amélie	48	NEG
71	2020-12-01	United States, San Diego	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	42	NEG
72	2020-12-01	United States, San Diego	product-f7e07591-2598-4615-82c2-34f6ad248e50	Eloise	42	NEG
73	2020-12-01	United States, San Diego	product-fa4a41fc-4a31-44b5-953f-8e2a45b43673	Cecile	49	NEG
74	2020-12-01	United States, San Diego	product-7bef3e02-033c-4259-93da-a25f4f7169be	Claudette	48	NEG
75	2021-01-01	United States, San Diego	product-20700833-fc84-4340-9a59-669fe6acc94b	Antoine	41	NEG
76	2021-01-01	United States, San Diego	product-124ef52a-c7c3-48af-b315-33a14b2f6e1d	François	41	NEG
77	2021-01-01	United States, San Diego	product-a19d1434-d5f2-4a2a-9fe0-7d70f63e391e	Denis	46	NEG
78	2021-01-01	United States, San Diego	product-9f6a916a-271c-4d78-9e5f-f802bbcf6548	Adele	40	NEG
79	2021-01-01	United States, San Diego	product-f709c12a-ffe5-48b1-a3d2-f247acf8e176	Danielle	40	NEG
80	2021-01-01	United States, San Diego	product-0a97c64c-582b-41a9-b367-2a4e081cf3d5	Estelle	47	NEG
81	2021-01-01	United States, San Diego	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	43	NEG
82	2021-01-01	United States, San Diego	product-b7c06d0a-977d-497b-ae3e-95b58985cafd	Amélie	44	NEG
83	2021-01-01	United States, San Diego	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	45	NEG
84	2021-01-01	United States, San Diego	product-f7e07591-2598-4615-82c2-34f6ad248e50	Eloise	41	NEG
85	2021-01-01	United States, San Diego	product-fa4a41fc-4a31-44b5-953f-8e2a45b43673	Cecile	41	NEG
86	2021-01-01	United States, San Diego	product-7bef3e02-033c-4259-93da-a25f4f7169be	Claudette	40	NEG
87	2021-02-01	United States, San Diego	product-20700833-fc84-4340-9a59-669fe6acc94b	Antoine	46	NEG
88	2021-02-01	United States, San Diego	product-124ef52a-c7c3-48af-b315-33a14b2f6e1d	François	45	NEG
89	2021-02-01	United States, San Diego	product-a19d1434-d5f2-4a2a-9fe0-7d70f63e391e	Denis	43	NEG
90	2021-02-01	United States, San Diego	product-9f6a916a-271c-4d78-9e5f-f802bbcf6548	Adele	49	NEG
91	2021-02-01	United States, San Diego	product-f709c12a-ffe5-48b1-a3d2-f247acf8e176	Danielle	49	NEG
92	2021-02-01	United States, San Diego	product-0a97c64c-582b-41a9-b367-2a4e081cf3d5	Estelle	44	NEG
93	2021-02-01	United States, San Diego	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	42	NEG
94	2021-02-01	United States, San Diego	product-b7c06d0a-977d-497b-ae3e-95b58985cafd	Amélie	49	NEG
95	2021-02-01	United States, San Diego	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	45	NEG
96	2021-02-01	United States, San Diego	product-f7e07591-2598-4615-82c2-34f6ad248e50	Eloise	44	NEG
97	2021-02-01	United States, San Diego	product-fa4a41fc-4a31-44b5-953f-8e2a45b43673	Cecile	43	NEG
98	2021-02-01	United States, San Diego	product-7bef3e02-033c-4259-93da-a25f4f7169be	Claudette	45	NEG

In [17]:

```
df4 = pd.merge(left=df3, right=negative_sentiment_df, on="ProductID", how="inner")
```

In [18]:

```
df4
```

Out[18]:

	OrderID	Date	Country	City	Latitude	Longtitude	ProductCategoryName	ProductID	Product_x	Quantity	Discount	Pri
0	order-fc3e5cf3-267e-49fa-af66-f9f86c341d1c	2020-11-09	Denmark	Copenhagen	55.68	12.56	Platforms	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	3.00	0.30	46.9
1	order-fc3e5cf3-267e-49fa-af66-f9f86c341d1c	2020-11-09	Denmark	Copenhagen	55.68	12.56	Platforms	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	3.00	0.30	46.9
2	order-fc3e5cf3-267e-49fa-af66-f9f86c341d1c	2020-11-09	Denmark	Copenhagen	55.68	12.56	Platforms	product-642f72ba-c5d6-4126-be0f-a22fe4e9fbb6	Bella	3.00	0.30	46.9
3	order-68b41bfc-ab5f-461f-bbfd-1b4576e79fc2	2020-11-20	Denmark	Copenhagen	55.68	12.56	Platforms	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	9.00	3.75	39.9
4	order-68b41bfc-ab5f-461f-bbfd-1b4576e79fc2	2020-11-20	Denmark	Copenhagen	55.68	12.56	Platforms	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	9.00	3.75	39.9
5	order-68b41bfc-ab5f-461f-bbfd-1b4576e79fc2	2020-11-20	Denmark	Copenhagen	55.68	12.56	Platforms	product-98f22154-ee97-4ef8-be84-7283cec0ebad	Bridgette	9.00	3.75	39.9

Data Analysis Result

Based on analysis, product ordered in Denmark and returned with negative sentiment occurred on 9 Nov 2020 and 20 Nov 2020. Product name Bella and Bridgette affected.

Python code done by Dennis Lam