IE517_HWK3

September 10, 2021

0.1 EDA: Corporate Bond Set

Perform an exploratory data analysis (EDA) on the "High Yield Corporate Bond" dataset. Use the code listings presented in Bowles Chapter 2 to guide you.

You do NOT have to do any other datasets (sonar_all_data or "rocks_v_mines", abalone, winequality or glass)!

Figure out how best to include the information, visually and quantitatively. Refer to the code in listing 2-1 through 2-10 and also the Datacamp assignments.

[1]:
```
import pandas as pd
df = pd.read_csv('HY_Universe_corporate bond.csv', header=0)

df.head()
```

[1]:
```
       CUSIP  Ticker  ... weekly_mean_ntrades weekly_median_ntrades
0  000324AA1  FLECIN  ...            3.541176                     1
1  00080QAB1     RBS  ...           18.412903                     3
2  00081TAD0    ACCO  ...            6.477612                     1
3  00081TAH1    ACCO  ...           27.038043                     1
4  00081TAJ7    ACCO  ...            9.238095                     1

[5 rows x 37 columns]
```

[2]:
```
#Listing 2-1, modified for df: find shape of dataframe
df.shape
```

[2]:
```
(2721, 37)
```

[3]:
```
#Listing 2-2, column data types
df.dtypes
```

[3]:
```
CUSIP                            object
Ticker                           object
Issue Date                       object
Maturity                         object
1st Call Date                    object
Moodys                           object
S_and_P                          object
Fitch                            object
Bloomberg Composite Rating       object
Coupon                          float64
```

```
Issued Amount                  float64
Maturity Type                   object
Coupon Type                     object
Maturity At Issue months       float64
Industry                        object
LiquidityScore                 float64
Months in JNK                   object
Months in HYG                   object
Months in Both                  object
IN_ETF                          object
LIQ SCORE                      float64
n_trades                         int64
volume_trades                  float64
total_median_size              float64
total_mean_size                float64
n_days_trade                     int64
days_diff_max                    int64
percent_intra_dealer           float64
percent_uncapped               float64
bond_type                        int64
Client_Trade_Percentage        float64
weekly_mean_volume             float64
weekly_median_volume           float64
weekly_max_volume              float64
weekly_min_volume              float64
weekly_mean_ntrades            float64
weekly_median_ntrades            int64
dtype: object
```

[4]:

```
#Listing 2-3 (actually 2-5), adapted for dataframe (summary statistics)
df.describe(include = 'all')
```

[4]:

```
            CUSIP Ticker  ... weekly_mean_ntrades weekly_median_ntrades
count        2721   2721  ...         2721.000000           2721.000000
unique       2721    870  ...                 NaN                   NaN
top     28368EAE6    LEH  ...                 NaN                   NaN
freq            1     45  ...                 NaN                   NaN
mean          NaN    NaN  ...           21.598988              2.471885
std           NaN    NaN  ...           32.901129              5.581749
min           NaN    NaN  ...            1.000000              1.000000
25%           NaN    NaN  ...            4.046154              1.000000
50%           NaN    NaN  ...           10.821429              1.000000
75%           NaN    NaN  ...           24.526316              2.000000
max           NaN    NaN  ...          513.769231            160.000000

[11 rows x 37 columns]
```

[5]:

```
#drop unique identifiers
df = df.drop(columns = ['CUSIP', 'Ticker', 'Issue Date', 'Maturity'])
```

[6]:

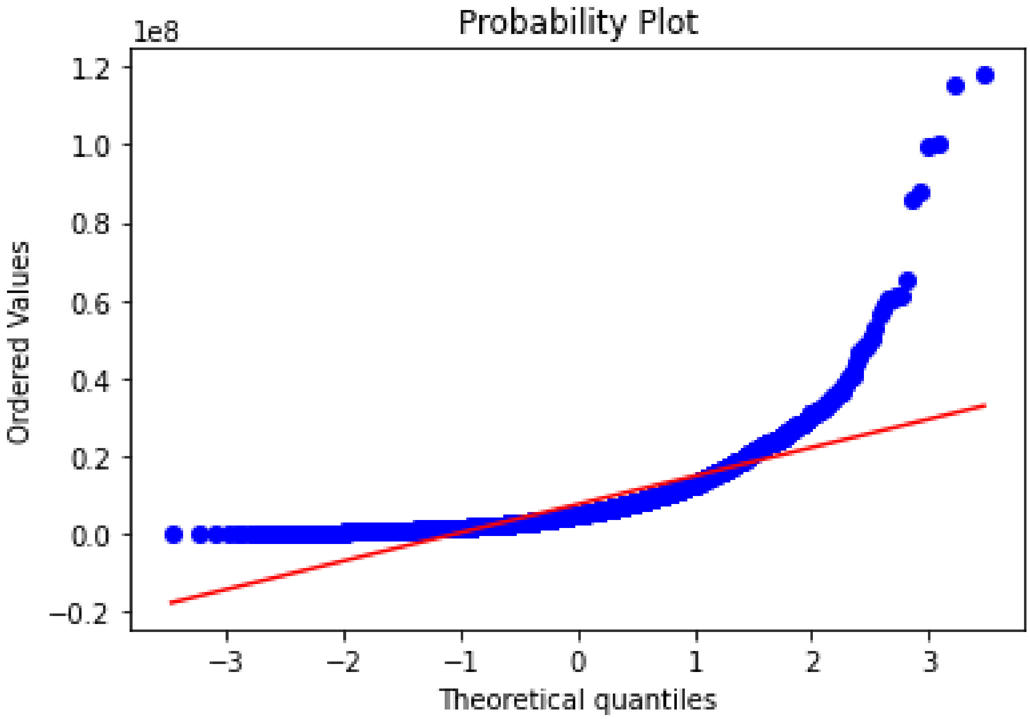
```
df. count()
```

[6]:

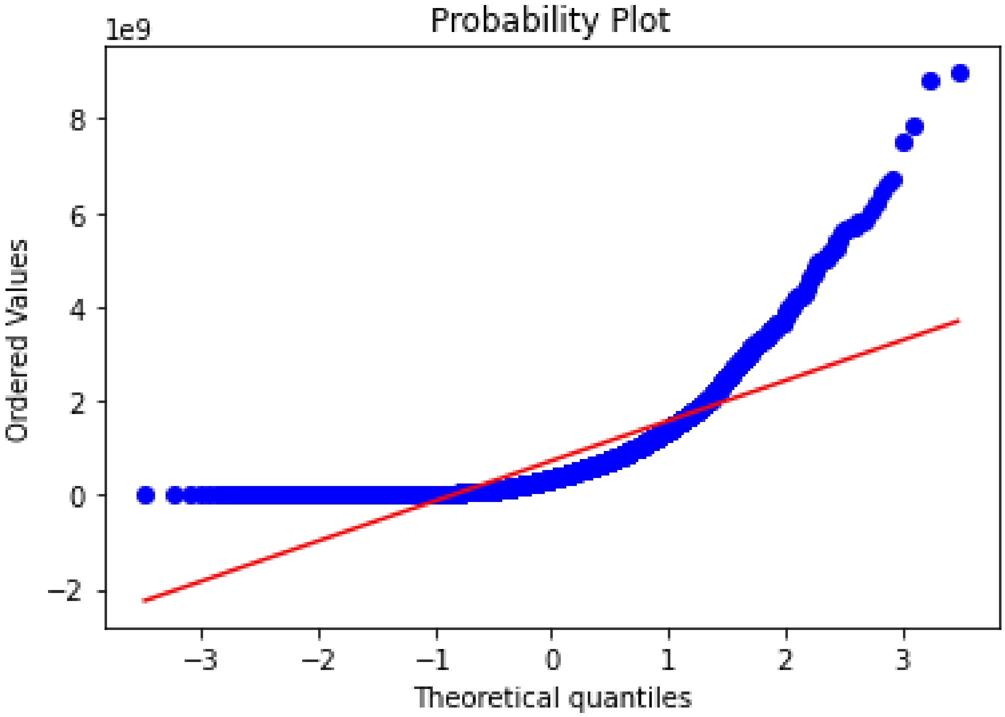
```
1st Call Date                 2721
Moodys                        2721
S_and_P                       2721
Fitch                         2721
Bloomberg Composite Rating    2721
Coupon                        2721
Issued Amount                 2721
Maturity Type                 2721
Coupon Type                   2721
Maturity At Issue months      2721
Industry                      2721
LiquidityScore                2721
Months in JNK                 2721
Months in HYG                 2721
Months in Both                2721
IN_ETF                        2721
LIQ SCORE                     2721
n_trades                      2721
volume_trades                 2721
total_median_size             2721
total_mean_size               2721
n_days_trade                  2721
days_diff_max                 2721
percent_intra_dealer          2721
percent_uncapped              2721
bond_type                     2721
Client_Trade_Percentage       2721
weekly_mean_volume            2721
weekly_median_volume          2721
weekly_max_volume             2721
weekly_min_volume             2721
weekly_mean_ntrades           2721
weekly_median_ntrades         2721
dtype: int64
```

[7]:

```
#Listing 2-4: Q-Q Plot
import scipy.stats as stats
import pylab
stats.probplot(df['weekly_mean_volume'], dist = "norm", plot = pylab)
pylab.show()
```



[8]:
```
#Listing 2-4: Q-Q Plot
import scipy.stats as stats
import pylab
stats.probplot(df['volume_trades'], dist = "norm", plot = pylab)
pylab.show()
```

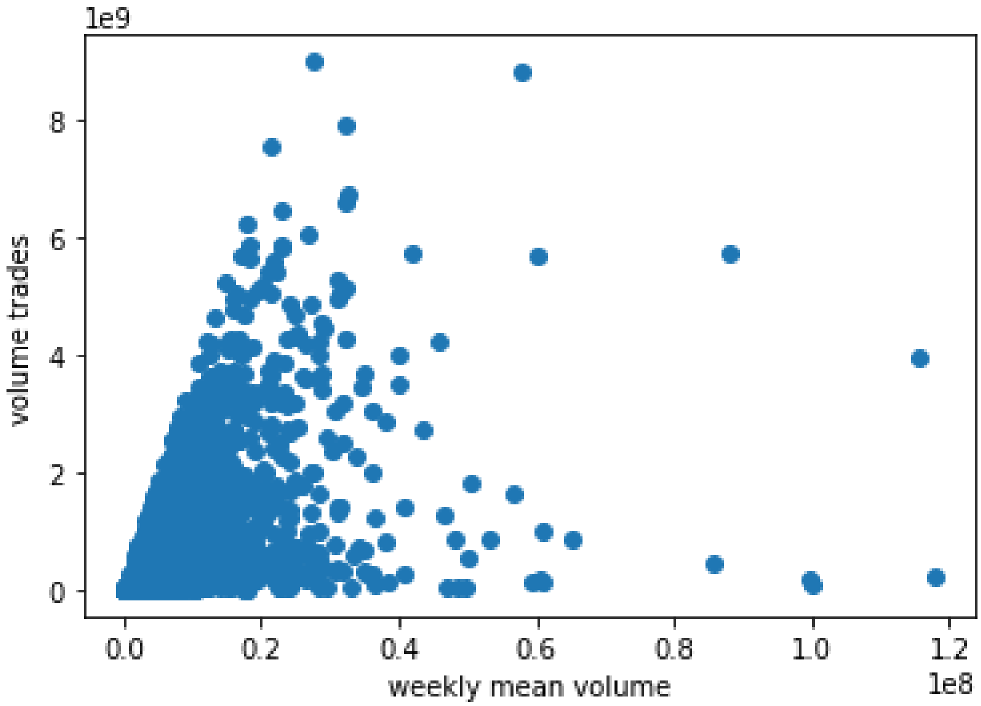


[9]:

```
#Listing 2-7
import matplotlib.pyplot as plt
plt.scatter(df['weekly_mean_volume'], df['volume_trades'])
plt.xlabel('weekly mean volume')
plt.ylabel('volume trades')
```

[9]:

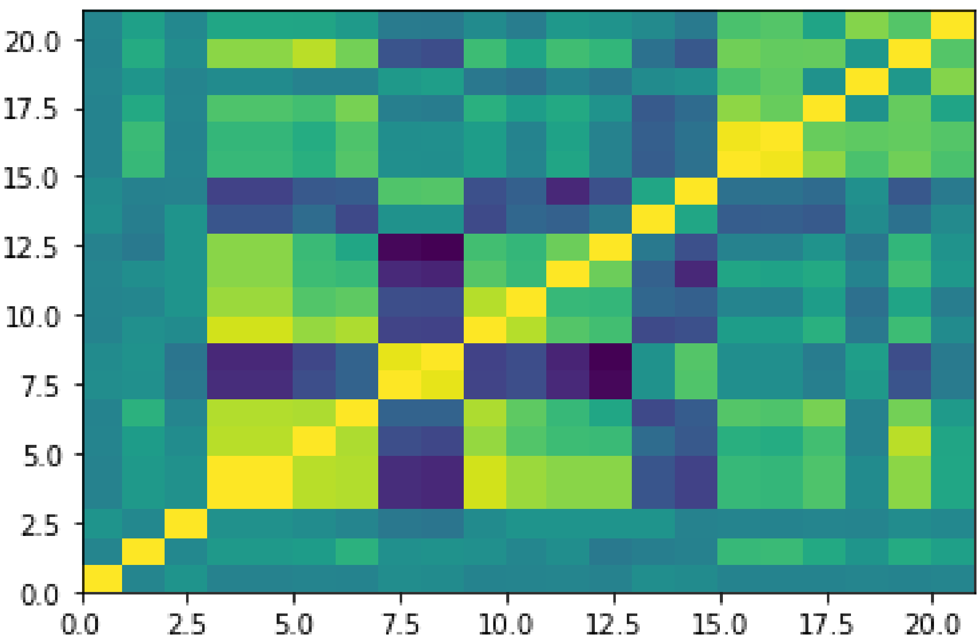
```
Text(0, 0.5, 'volume trades')
```



[10]:

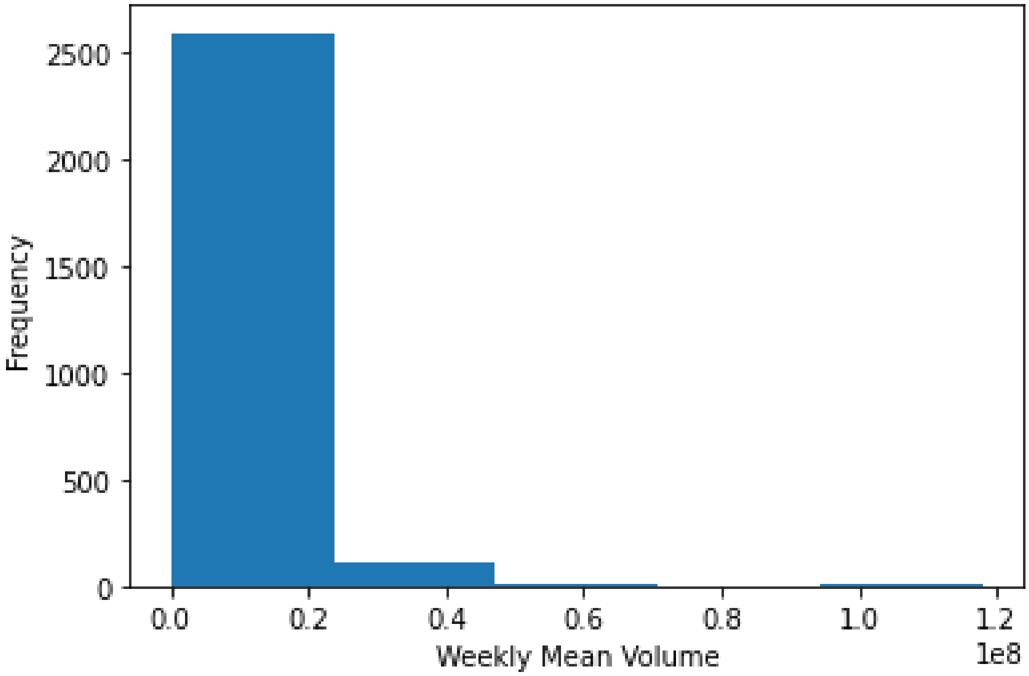
```
#Listing 2-10
from pandas import DataFrame
corMat = DataFrame(df.corr())

plt.pcolor(corMat)
plt.show()
```



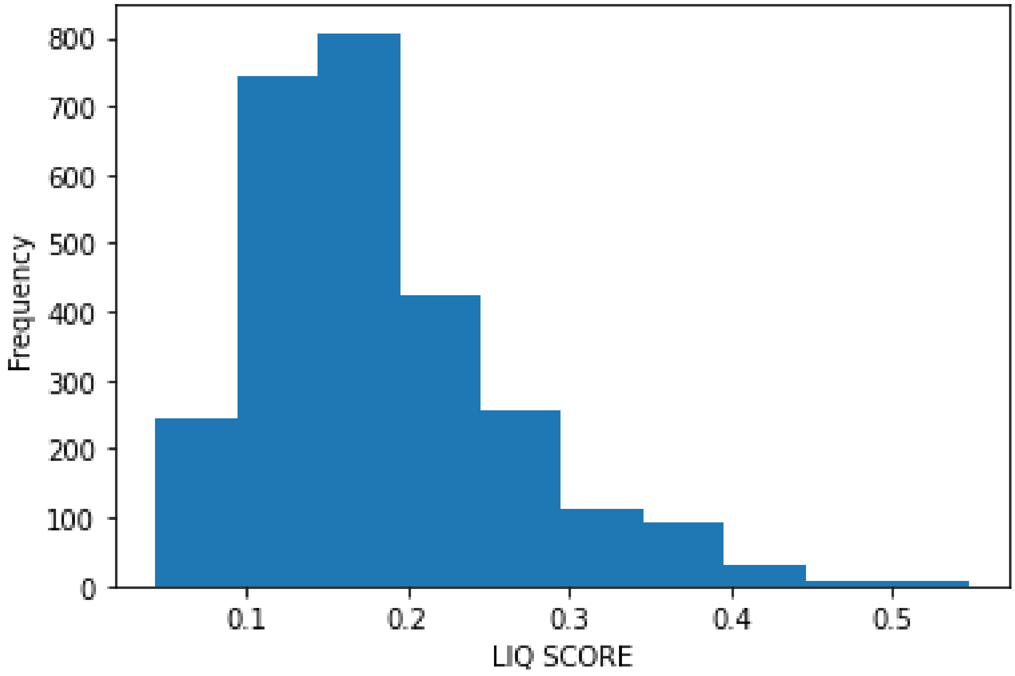
[11]:
```
#Histogram
plt.hist(df['weekly_mean_volume'], bins = 5)
plt.xlabel('Weekly Mean Volume')
plt.ylabel('Frequency')
```

[11]:
```
Text(0, 0.5, 'Frequency')
```



[12]:
```
#Histogram
plt.hist(df['LIQ SCORE'])
plt.xlabel('LIQ SCORE')
plt.ylabel('Frequency')
```

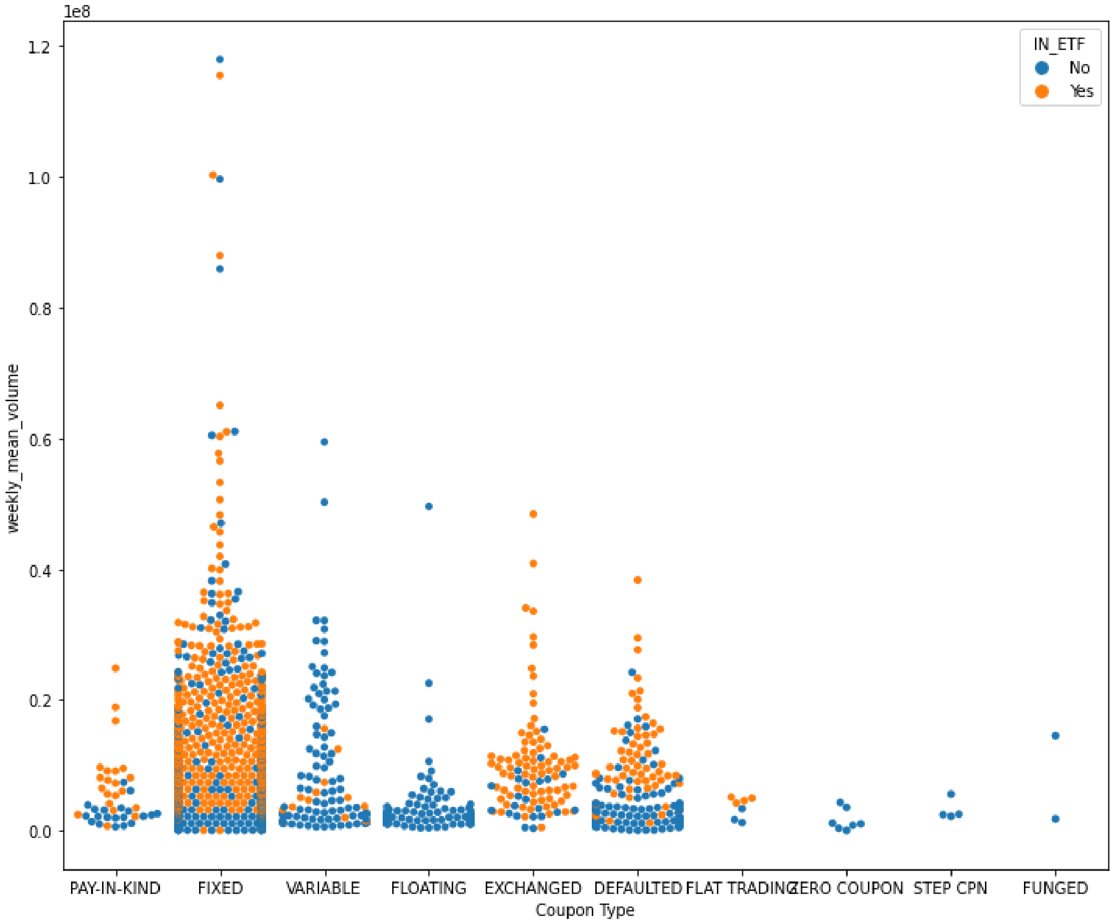
[12]:
```
Text(0, 0.5, 'Frequency')
```



[15]:
```
# BEE Swarm
import seaborn as sns
fig_dims = (12, 10)
fig,ax = plt.subplots(figsize = fig_dims)
s = sns.swarmplot(x = 'Coupon Type', y = 'weekly_mean_volume', hue = 'IN_ETF',␣
 ↪ax = ax, data = df)
```

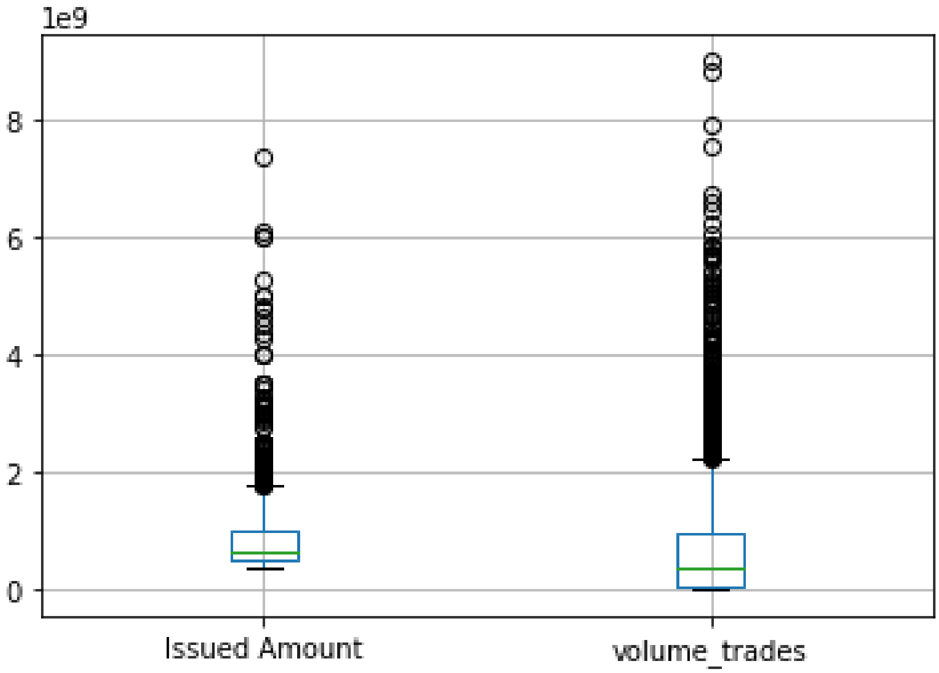
```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning:
85.2% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning:
18.9% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning:
```

```
57.3% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning:
6.9% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning:
38.6% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



[26]:
```
#Boxplots
df.boxplot(column = ['Issued Amount', 'volume_trades'])
```

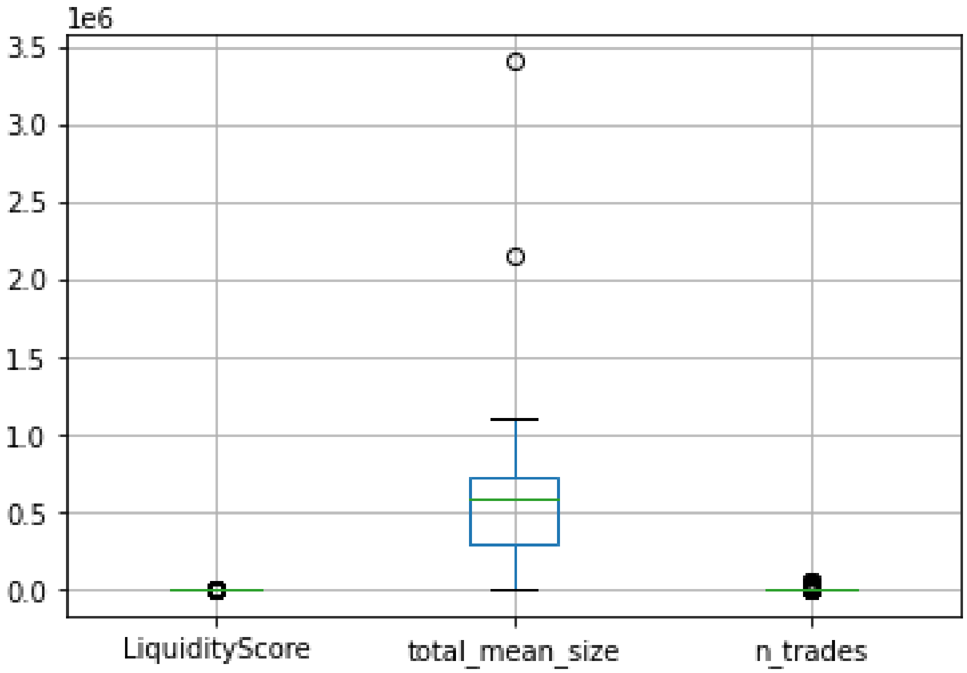
[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc1f167a2d0>
```



[28]:
```
df.boxplot(column = ['LiquidityScore', 'total_mean_size', 'n_trades'])

#obvious in this plot we have a scale problem. Can see it in data, but boxplots␣
↪highlight that some are 10^6, 10^9, or 10^1 numeric values, so we need to␣
↪address this when
# getting data ready to fit model
```

[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc1e64db350>
```



[29]:

```
print("My name is Emma Mayes")
print("My NetID is: eemayes2")
print("I hereby certify that I have read the University policy on Academic␣
 ↪Integrity and that I am not in violation.")
```

```
My name is Emma Mayes
My NetID is: eemayes2
I hereby certify that I have read the University policy on Academic Integrity
and that I am not in violation.
```

[ ]:

```
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('IE517_HWK3.ipynb')
```

```
--2021-09-10 22:11:42--  https://raw.githubusercontent.com/brpy/colab-
pdf/master/colab_pdf.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)...
185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1864 (1.8K) [text/plain]
Saving to: colab_pdf.py

colab_pdf.py        100%[===================>]   1.82K  --.-KB/s    in 0s
```

```
2021-09-10 22:11:43 (21.2 MB/s) - colab_pdf.py saved [1864/1864]
```

```
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
```

```
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
```