212.3

Eric Yarger

Research Question

Using Market Basket Analysis, what are the top rules

and insights discovered from our patient's prescription history,

using a support minimum = .03, that can help guide our organization

to providing better patient care and organizational profitability?

In [1]:

```
# Import General Libraries
import pandas as pd
from scipy import stats
import missingno as msno
```

In [2]:

```
# Import Market Basket specific libraries
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

In [3]:

```
# Windows 10, Anaconda, JupyterLab, JupyterNotebook
# Jupyter environment version
!jupyter --version
```

```
jupyter core     : 4.6.3
jupyter-notebook : 6.0.3
qtconsole        : 4.7.2
ipython          : 7.13.0
ipykernel        : 5.1.4
jupyter client   : 6.1.2
jupyter lab      : 1.2.6
nbconvert        : 5.6.1
ipywidgets       : 7.5.1
nbformat         : 5.0.4
traitlets        : 4.3.3
```

In [4]:

```
# Python Environment version
import platform
print(platform.python_version())
```

```
3.7.7
```

In [5]:

```
# Read in data file
df = pd.read_csv('C:/Users/ericy/Desktop/medical_market_basket.csv')
```

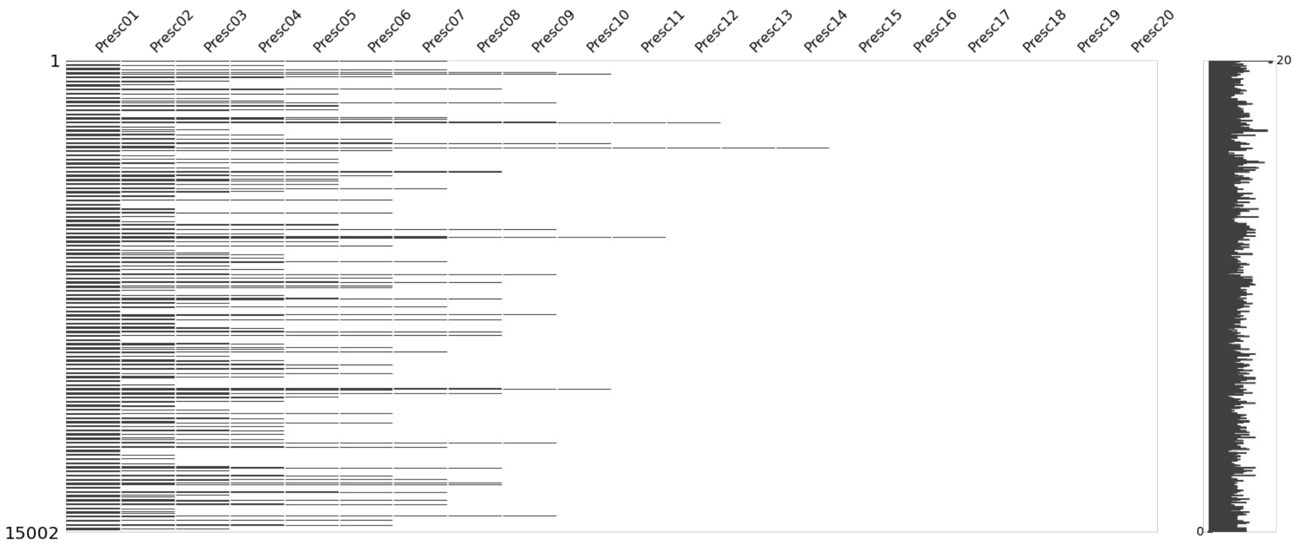
Initial Data Exploration

In [6]:

```
msno.matrix(df)
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29ae5078e08>
```



In [ ]:

```

```

Select relevant rows

In [7]:

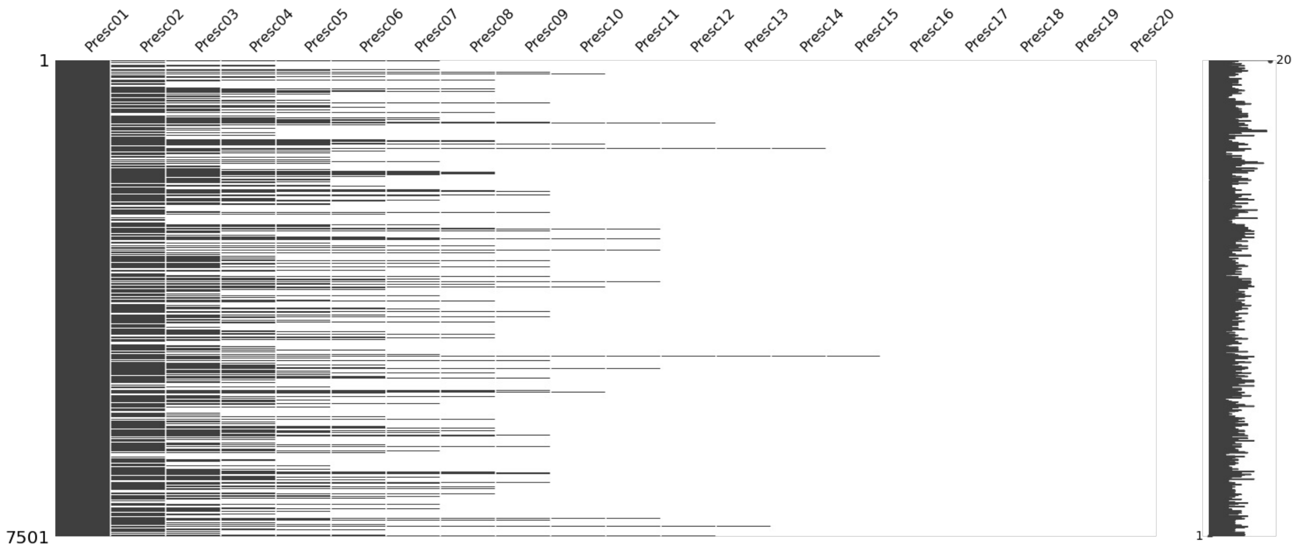
```
# Every other row is blank.
# Select every other row (that contains values).
# Update variable

df = df.iloc[1::2]
```

In [8]:

```
msno.matrix(df)
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29ae5196048>
```



In [9]:

```
# Read out intermediary file.
df.to_excel('C:/Users/ericy/Desktop/D212.3.xlsx', index=False)
```

In [10]:

```
# Read in intermediary file.
df = pd.read_excel('C:/Users/ericy/Desktop/D212.3.xlsx')
```

In [ ]:

```

```

In [ ]:

```

```

In [11]:

```
# Import Transaction Encoder
from mlxtend.preprocessing import TransactionEncoder
```

In [12]:

```
df_out = df.apply(lambda x: list(x.dropna().values), axis=1).tolist()
```

In [13]:

```
# Transaction Encoding
# This fits our df_out dataset and transforms to Boolean values.
# Eliminates NaN values, and prepares the dataset for Apriori Algorithm

# code reference (Raschka, n.d.a)
# code reference (Boston, 2019)

te = TransactionEncoder()
te_arr = te.fit(df_out).transform(df_out)
#te_arr
```

In [14]:

```
# Assign Transaction Encoder variable 'te' columns to variable
# Necessary for providing relevant, easy to read output from Apriori results

column_names = te.columns_
```

In [15]:

```
# Optional read out for integers instead of boolean array

# te_int = te_arr.astype('int')
# te_int
```

In [ ]:

```

```

In [16]:

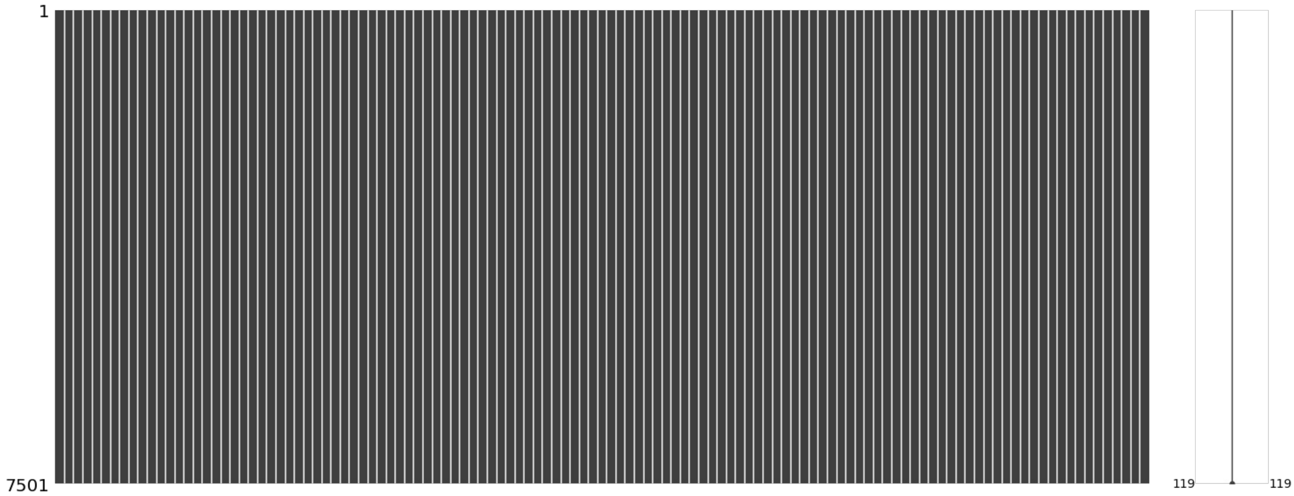
```
# Assign Boolean Array to pandas dataframe.

#Assign column name of drug from varible 'column_name'

dataset = pd.DataFrame(te_arr, columns=column_names)
```

In [17]:

```
msno.matrix(dataset)
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29ae61e89c8>
```



In [18]:

```
# Read out fully prepared dataset to Excel file.  No index necessary.

dataset.to_excel('C:/Users/ericy/Desktop/D212.3.clean.xlsx', index=False)
```

Market Basket Analysis using Apriori Algorithm

In [19]:

```
# Apriori Market Basket Analysis
# Code Reference (Raschka, n.d.b)

# Testing different min_support thresholds

itemsets = apriori(dataset, min_support=0.05, use_colnames=True)
```

In [20]:

```
itemsets
```

Out[20]:

	support	itemsets
0	0.238368	(abilify)
1	0.079323	(alprazolam)
2	0.071457	(amlodipine)
3	0.068391	(amphetamine salt combo)
4	0.179709	(amphetamine salt combo xr)
5	0.129583	(atorvastatin)
6	0.174110	(carvedilol)
7	0.076523	(cialis)
8	0.087188	(citalopram)
9	0.059992	(clopidogrel)
10	0.081056	(dextroamphetamine XR)
11	0.163845	(diazepam)
12	0.095054	(doxycycline hyclate)
13	0.080389	(ezetimibe)
14	0.051060	(fenofibrate)
15	0.052393	(furosemide)
16	0.065858	(glipizide)
17	0.170911	(glyburide)
18	0.063325	(levofloxacin)
19	0.098254	(lisinopril)
20	0.132116	(losartan)
21	0.050527	(metformin)
22	0.095321	(metoprolol)
23	0.058526	(naproxen)
24	0.062525	(paroxetine)
25	0.050927	(abilify, amphetamine salt combo xr)
26	0.059725	(abilify, carvedilol)
27	0.052660	(abilify, diazepam)

In [21]:

```
itemsets = apriori(dataset, min_support=0.08, use_colnames=True)
```

In [22]:

```
itemsets
```

Out[22]:

	support	itemsets
0	0.238368	(abilify)
1	0.179709	(amphetamine salt combo xr)
2	0.129583	(atorvastatin)
3	0.174110	(carvedilol)
4	0.087188	(citalopram)
5	0.081056	(dextroamphetamine XR)
6	0.163845	(diazepam)
7	0.095054	(doxycycline hyclate)
8	0.080389	(ezetimibe)
9	0.170911	(glyburide)
10	0.098254	(lisinopril)
11	0.132116	(losartan)
12	0.095321	(metoprolol)

In [23]:

```
itemsets = apriori(dataset, min_support=0.2, use_colnames=True)
```

In [24]:

```
itemsets
```

Out[24]:

	support	itemsets
0	0.238368	(abilify)

In [25]:

```
itemsets = apriori(dataset, min_support=0.015, use_colnames=True)
```

In [26]:

```
itemsets
```

Out[26]:

	support	itemsets
0	0.046794	(Premarin)
1	0.238368	(abilify)
2	0.015731	(acetaminophen)
3	0.020397	(albuterol aerosol)
4	0.033329	(allopurinol)
...	...	...
148	0.015998	(methylprednisone, lisinopril)
149	0.016931	(metoprolol, lisinopril)
150	0.015731	(abilify, atorvastatin, carvedilol)
151	0.015865	(diazepam, abilify, carvedilol)
152	0.017064	(abilify, lisinopril, carvedilol)

153 rows × 2 columns

In [27]:

```
freq_sets = apriori(dataset, min_support=0.03, use_colnames=True)
freq_sets['length'] = freq_sets['itemsets'].apply(lambda x: len(x))
freq_sets
```

Out[27]:

	support	itemsets	length
0	0.046794	(Premarin)	1
1	0.238368	(abilify)	1
2	0.033329	(allopurinol)	1
3	0.079323	(alprazolam)	1
4	0.071457	(amlodipine)	1
5	0.030129	(amphetamine)	1
6	0.068391	(amphetamine salt combo)	1
7	0.179709	(amphetamine salt combo xr)	1
8	0.129583	(atorvastatin)	1
9	0.174110	(carvedilol)	1
10	0.033729	(celecoxib)	1
11	0.076523	(cialis)	1
12	0.087188	(citalopram)	1
13	0.059992	(clopidogrel)	1
14	0.081056	(dextroamphetamine XR)	1
15	0.163845	(diazepam)	1
16	0.095054	(doxycycline hyclate)	1
17	0.080389	(ezetimibe)	1

18	0.051060	(fenofibrate)	1
19	0.031862	(fluconozole)	1
20	0.052393	(furosemide)	1
21	0.065858	(glipizide)	1
22	0.170911	(glyburide)	1
23	0.043061	(lantus)	1
24	0.063325	(levofloxacin)	1
25	0.098254	(lisinopril)	1
26	0.132116	(losartan)	1
27	0.050527	(metformin)	1
28	0.049460	(methylprednisone)	1
29	0.095321	(metoprolol)	1
30	0.047460	(metoprolol succinate XL)	1
31	0.032396	(metoprolol tartrate)	1
32	0.058526	(naproxen)	1
33	0.062525	(paroxetine)	1
34	0.030396	(pravastatin)	1
35	0.042528	(spironolactone)	1
36	0.050927	(abilify, amphetamine salt combo xr)	2
37	0.047994	(abilify, atorvastatin)	2
38	0.059725	(abilify, carvedilol)	2
39	0.052660	(abilify, diazepam)	2
40	0.033729	(doxycycline hyclate, abilify)	2
41	0.033729	(abilify, glyburide)	2
42	0.040928	(abilify, lisinopril)	2
43	0.031063	(losartan, abilify)	2
44	0.035729	(abilify, metoprolol)	2
45	0.030796	(amphetamine salt combo xr, atorvastatin)	2
46	0.036528	(amphetamine salt combo xr, carvedilol)	2
47	0.033196	(amphetamine salt combo xr, diazepam)	2
48	0.036395	(amphetamine salt combo xr, glyburide)	2
49	0.035462	(atorvastatin, carvedilol)	2
50	0.032129	(atorvastatin, diazepam)	2
51	0.039195	(diazepam, carvedilol)	2
52	0.039195	(lisinopril, carvedilol)	2
53	0.034395	(glyburide, diazepam)	2

In [28]:

```
freq_sets[ (freq_sets['length'] == 2) &
                   (freq_sets['support'] >= 0.03) ]
```

Out[28]:

	support	itemsets	length
36	0.050927	(abilify, amphetamine salt combo xr)	2
37	0.047994	(abilify, atorvastatin)	2
38	0.059725	(abilify, carvedilol)	2
39	0.052660	(abilify, diazepam)	2
40	0.033729	(doxycycline hyclate, abilify)	2
41	0.033729	(abilify, glyburide)	2
42	0.040928	(abilify, lisinopril)	2
43	0.031063	(losartan, abilify)	2
44	0.035729	(abilify, metoprolol)	2
45	0.030796	(amphetamine salt combo xr, atorvastatin)	2
46	0.036528	(amphetamine salt combo xr, carvedilol)	2
47	0.033196	(amphetamine salt combo xr, diazepam)	2
48	0.036395	(amphetamine salt combo xr, glyburide)	2
49	0.035462	(atorvastatin, carvedilol)	2
50	0.032129	(atorvastatin, diazepam)	2
51	0.039195	(diazepam, carvedilol)	2
52	0.039195	(lisinopril, carvedilol)	2
53	0.034395	(glyburide, diazepam)	2

In [29]:

```
# Code Reference (Brown, 2019)
rules = association_rules(freq_sets, min_threshold=.03)
rules
```

Out[29]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(abilify)	(amphetamine salt combo xr)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158
1	(amphetamine salt combo xr)	(abilify)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815
2	(abilify)	(atorvastatin)	0.238368	0.129583	0.047994	0.201342	1.553774	0.017105	1.089850
3	(atorvastatin)	(abilify)	0.129583	0.238368	0.047994	0.370370	1.553774	0.017105	1.209650
4	(abilify)	(carvedilol)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008
5	(carvedilol)	(abilify)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
6	(abilify)	(diazepam)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
7	(diazepam)	(abilify)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
8	(doxycycline hyclate)	(abilify)	0.095054	0.238368	0.033729	0.354839	1.488616	0.011071	1.180529
9	(abilify)	(doxycycline hyclate)	0.238368	0.095054	0.033729	0.141499	1.488616	0.011071	1.054100
10	(abilify)	(glyburide)	0.238368	0.170911	0.033729	0.141499	0.827912	-0.007011	0.965741
11	(glyburide)	(abilify)	0.170911	0.238368	0.033729	0.197348	0.827912	-0.007011	0.948894
12	(abilify)	(lisinopril)	0.238368	0.098254	0.040928	0.171700	1.747522	0.017507	1.088672
13	(lisinopril)	(abilify)	0.098254	0.238368	0.040928	0.416554	1.747522	0.017507	1.305401
14	(losartan)	(abilify)	0.132116	0.238368	0.031063	0.235116	0.986357	-0.000430	0.995748
15	(abilify)	(losartan)	0.238368	0.132116	0.031063	0.130313	0.986357	-0.000430	0.997927
16	(abilify)	(metoprolol)	0.238368	0.095321	0.035729	0.149888	1.572463	0.013007	1.064189
17	(metoprolol)	(abilify)	0.095321	0.238368	0.035729	0.374825	1.572463	0.013007	1.218270
18	(amphetamine salt combo xr)	(atorvastatin)	0.179709	0.129583	0.030796	0.171365	1.322437	0.007509	1.050423
19	(atorvastatin)	(amphetamine salt combo xr)	0.129583	0.179709	0.030796	0.237654	1.322437	0.007509	1.076009
20	(amphetamine salt combo xr)	(carvedilol)	0.179709	0.174110	0.036528	0.203264	1.167446	0.005239	1.036592
21	(carvedilol)	(amphetamine salt combo xr)	0.174110	0.179709	0.036528	0.209801	1.167446	0.005239	1.038081
22	(amphetamine salt combo xr)	(diazepam)	0.179709	0.163845	0.033196	0.184718	1.127397	0.003751	1.025603
23	(diazepam)	(amphetamine salt combo xr)	0.163845	0.179709	0.033196	0.202604	1.127397	0.003751	1.028711
24	(amphetamine salt combo xr)	(glyburide)	0.179709	0.170911	0.036395	0.202522	1.184961	0.005681	1.039640
25	(glyburide)	(amphetamine salt combo xr)	0.170911	0.179709	0.036395	0.212949	1.184961	0.005681	1.042232
26	(atorvastatin)	(carvedilol)	0.129583	0.174110	0.035462	0.273663	1.571779	0.012900	1.137061
27	(carvedilol)	(atorvastatin)	0.174110	0.129583	0.035462	0.203675	1.571779	0.012900	1.093043
28	(atorvastatin)	(diazepam)	0.129583	0.163845	0.032129	0.247942	1.513276	0.010898	1.111823
29	(diazepam)	(atorvastatin)	0.163845	0.129583	0.032129	0.196094	1.513276	0.010898	1.082736
30	(diazepam)	(carvedilol)	0.163845	0.174110	0.039195	0.239219	1.373952	0.010668	1.085581
31	(carvedilol)	(diazepam)	0.174110	0.163845	0.039195	0.225115	1.373952	0.010668	1.079070
32	(lisinopril)	(carvedilol)	0.098254	0.174110	0.039195	0.398915	2.291162	0.022088	1.373997
33	(carvedilol)	(lisinopril)	0.174110	0.098254	0.039195	0.225115	2.291162	0.022088	1.163716
34	(glyburide)	(diazepam)	0.170911	0.163845	0.034395	0.201248	1.228284	0.006393	1.046827
35	(diazepam)	(glyburide)	0.163845	0.170911	0.034395	0.209927	1.228284	0.006393	1.049383

In [30]:

```
# Uncomment below lines to sort by confidence, support, or lift in descending order
#rules.sort_values(by=['confidence'], ascending=False)
#rules.sort_values(by=['support'], ascending=False)
#rules.sort_values(by=['lift'], ascending=False)
```

In [ ]: