In [1]:
```
!pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\acer\anaconda3\lib\site-packages (0.18.0)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\acer\anaconda3\lib\site-packages (from mlxtend) (3.3.2)
Requirement already satisfied: scipy>=1.2.1 in c:\users\acer\anaconda3\lib\site-packages (from mlxtend) (1.5.2)
Requirement already satisfied: setuptools in c:\users\acer\anaconda3\lib\site-packages (from mlxtend) (50.3.1.post202
01107)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\acer\anaconda3\lib\site-packages (from mlxtend) (0.2
3.2)
Requirement already satisfied: numpy>=1.16.2 in c:\users\acer\anaconda3\lib\site-packages (from mlxtend) (1.19.2)
Requirement already satisfied: joblib>=0.13.2 in c:\users\acer\anaconda3\lib\site-packages (from mlxtend) (0.17.0)
Requirement already satisfied: pandas>=0.24.2 in c:\users\acer\anaconda3\lib\site-packages (from mlxtend) (1.1.3)
Requirement already satisfied: cycler>=0.10 in c:\users\acer\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlx
tend) (0.10.0)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\acer\anaconda3\lib\site-packages (from matplotlib>=3.
0.0->mlxtend) (2020.6.20)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\acer\anaconda3\lib\site-packages
(from matplotlib>=3.0.0->mlxtend) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\acer\anaconda3\lib\site-packages (from matplotlib>=3.0.0
->mlxtend) (1.3.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\acer\anaconda3\lib\site-packages (from matplotlib>=3.0.0->ml
xtend) (8.0.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\acer\anaconda3\lib\site-packages (from matplotlib>=3.
0.0->mlxtend) (2.8.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\acer\anaconda3\lib\site-packages (from scikit-learn>=
0.20.3->mlxtend) (2.1.0)
Requirement already satisfied: pytz>=2017.2 in c:\users\acer\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxten
d) (2020.1)
Requirement already satisfied: six in c:\users\acer\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib>=3.0.0
->mlxtend) (1.15.0)
```

In [2]:
```
import pandas as pd
from mlxtend.frequent_patterns import apriori,association_rules
from mlxtend.preprocessing import TransactionEncoder
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
book = pd.read_csv("C:/Users/acer/Sandesh Pal/Data Science Assgn/ASS rule/book.csv")
```

In [5]:
```
book.head()
```

Out[5]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	ItalArt	Florence
0	0	1	0	1	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	1	0	1	0	0	0	0
4	0	0	1	0	0	0	1	0	0	0	0

In [6]:

```
book.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ChildBks   2000 non-null   int64
 1   YouthBks   2000 non-null   int64
 2   CookBks    2000 non-null   int64
 3   DoItYBks   2000 non-null   int64
 4   RefBks     2000 non-null   int64
 5   ArtBks     2000 non-null   int64
 6   GeogBks    2000 non-null   int64
 7   ItalCook   2000 non-null   int64
 8   ItalAtlas  2000 non-null   int64
 9   ItalArt    2000 non-null   int64
 10  Florence   2000 non-null   int64
dtypes: int64(11)
memory usage: 172.0 KB
```

Apriori Algorithm with 10 % minimum support

In [7]:

```
frequent_book1 = apriori(book, min_support=0.1, use_colnames=True)
frequent_book1
```

Out[7]:

	support	itemsets
0	0.4230	(ChildBks)
1	0.2475	(YouthBks)

	support	itemsets
2	0.4310	(CookBks)
3	0.2820	(DoItYBks)
4	0.2145	(RefBks)
5	0.2410	(ArtBks)
6	0.2760	(GeogBks)
7	0.1135	(ItalCook)
8	0.1085	(Florence)
9	0.1650	(ChildBks, YouthBks)
10	0.2560	(CookBks, ChildBks)
11	0.1840	(ChildBks, DoItYBks)
12	0.1515	(ChildBks, RefBks)
13	0.1625	(ChildBks, ArtBks)
14	0.1950	(ChildBks, GeogBks)
15	0.1620	(CookBks, YouthBks)
16	0.1155	(YouthBks, DoItYBks)
17	0.1010	(YouthBks, ArtBks)
18	0.1205	(GeogBks, YouthBks)
19	0.1875	(CookBks, DoItYBks)
20	0.1525	(CookBks, RefBks)
21	0.1670	(CookBks, ArtBks)
22	0.1925	(CookBks, GeogBks)
23	0.1135	(CookBks, ItalCook)
24	0.1055	(RefBks, DoItYBks)
25	0.1235	(ArtBks, DoItYBks)
26	0.1325	(GeogBks, DoItYBks)

	support	itemsets
27	0.1105	(RefBks, GeogBks)
28	0.1275	(GeogBks, ArtBks)
29	0.1290	(CookBks, ChildBks, YouthBks)
30	0.1460	(CookBks, ChildBks, DoItYBks)
31	0.1225	(CookBks, ChildBks, RefBks)
32	0.1265	(CookBks, ChildBks, ArtBks)
33	0.1495	(CookBks, ChildBks, GeogBks)
34	0.1045	(ChildBks, GeogBks, DoItYBks)
35	0.1020	(ChildBks, GeogBks, ArtBks)
36	0.1015	(CookBks, ArtBks, DoItYBks)
37	0.1085	(CookBks, GeogBks, DoItYBks)
38	0.1035	(CookBks, GeogBks, ArtBks)

In [9]:

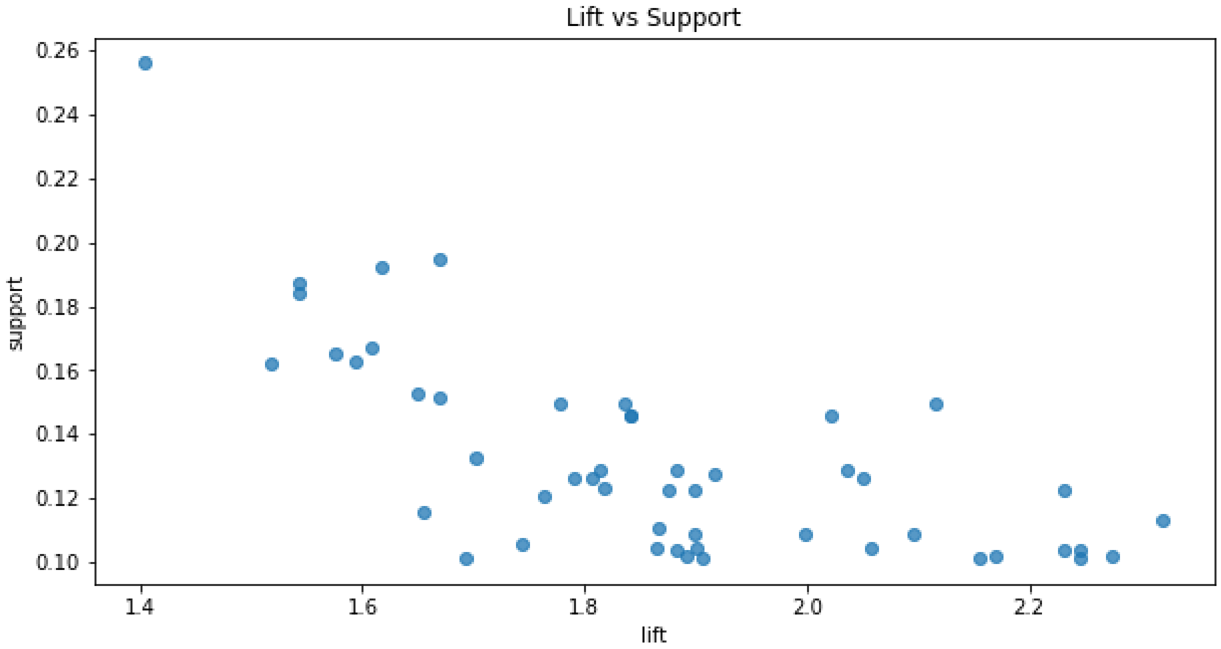
```
# using Lift as metric and keeping the value as maximum 1
rules1a = association_rules(frequent_book1, metric="lift", min_threshold=1)
rules1a.sort_values('lift',ascending = False).head(10)
```

Out[9]:

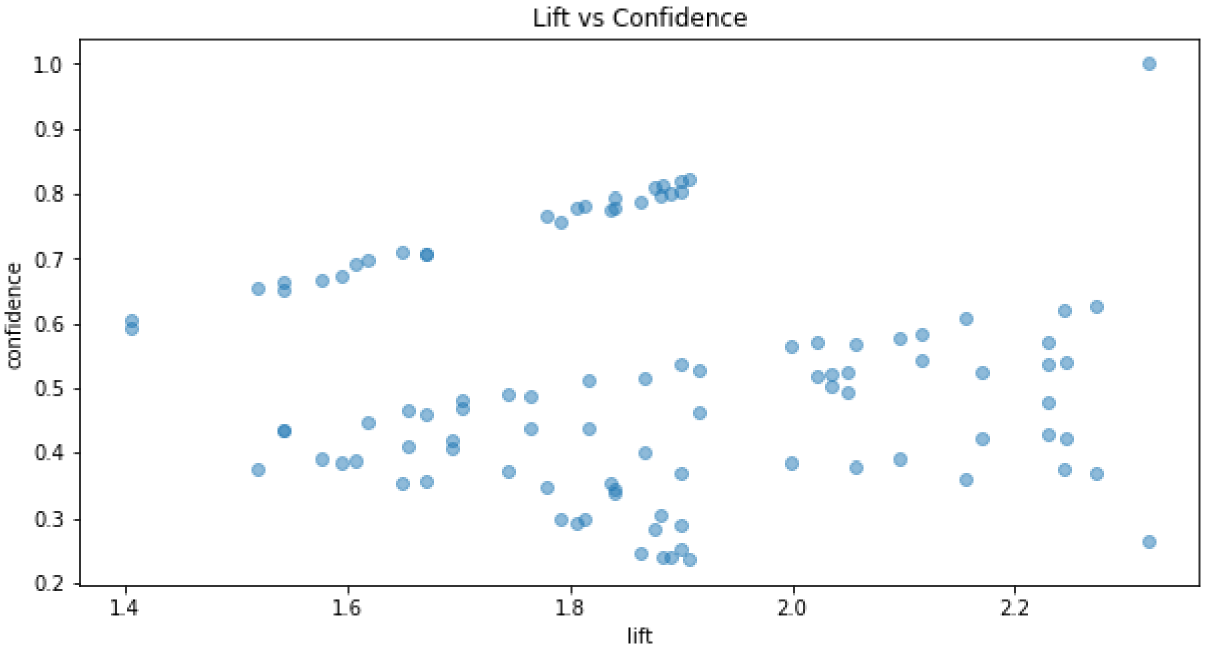
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
28	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.064582	1.203406
29	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582	inf
77	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150	1.944628
80	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.057150	1.328448
86	(ArtBks)	(CookBks, DoItYBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.056313	1.403674
83	(CookBks, DoItYBks)	(ArtBks)	0.1875	0.2410	0.1015	0.541333	2.246196	0.056313	1.654797
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408	1.332800
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057408	1.904063

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
99	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107	1.415327
94	(CookBks, GeogBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107	1.641657

In [10]:
```
plt.figure(figsize=(10,5))
plt.scatter(rules1a['lift'], rules1a['support'], alpha=0.5)
plt.xlabel('lift')
plt.ylabel('support')
plt.title('Lift vs Support')
plt.show()
```



In [11]:
```
plt.figure(figsize=(10,5))
plt.scatter(rules1a['lift'], rules1a['confidence'], alpha=0.5)
plt.xlabel('lift')
plt.ylabel('confidence')
plt.title('Lift vs Confidence')
plt.show()
```



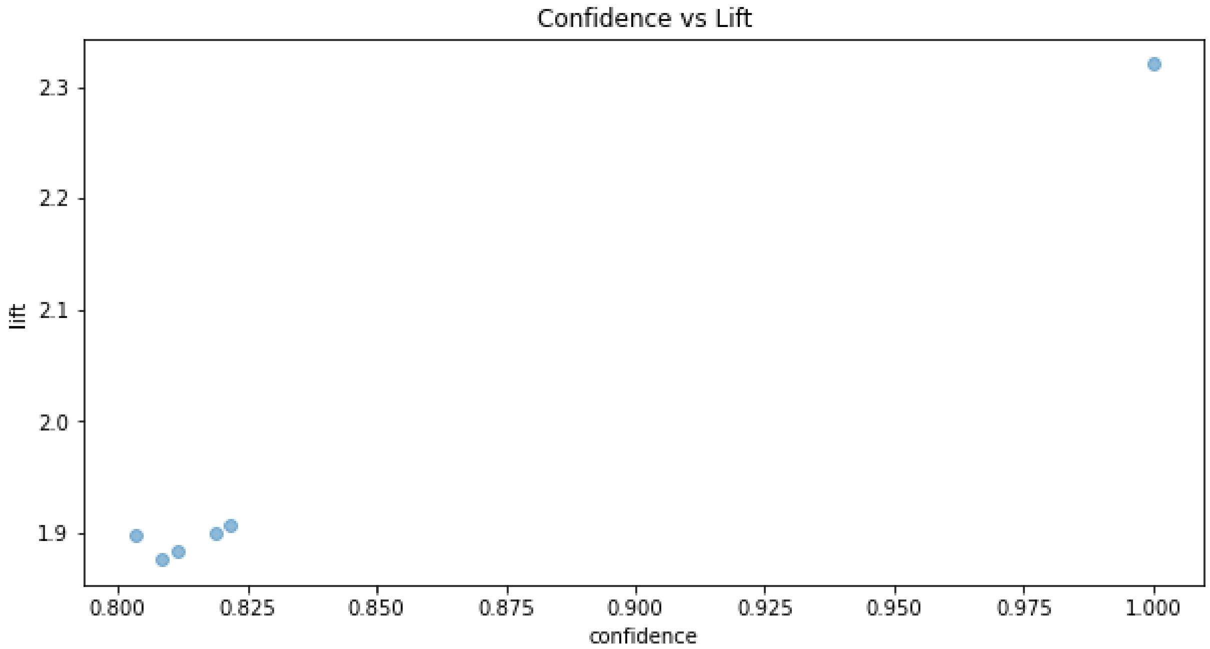
In [12]:
```
# using confidence as metric and keeping the value as 0.8
rules1b = association_rules(frequent_book1, metric="confidence", min_threshold=0.8)
rules1b.sort_values('lift',ascending = False).head(10)
```

Out[12]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ItalCook)	(CookBks)	0.1135	0.431	0.1135	1.000000	2.320186	0.064582	inf
3	(DoItYBks, ArtBks)	(CookBks)	0.1235	0.431	0.1015	0.821862	1.906873	0.048272	3.194159
4	(GeogBks, DoItYBks)	(CookBks)	0.1325	0.431	0.1085	0.818868	1.899926	0.051392	3.141354
1	(CookBks, RefBks)	(ChildBks)	0.1525	0.423	0.1225	0.803279	1.899004	0.057993	2.933083
5	(GeogBks, ArtBks)	(CookBks)	0.1275	0.431	0.1035	0.811765	1.883445	0.048547	3.022812
2	(ChildBks, RefBks)	(CookBks)	0.1515	0.431	0.1225	0.808581	1.876058	0.057204	2.972534

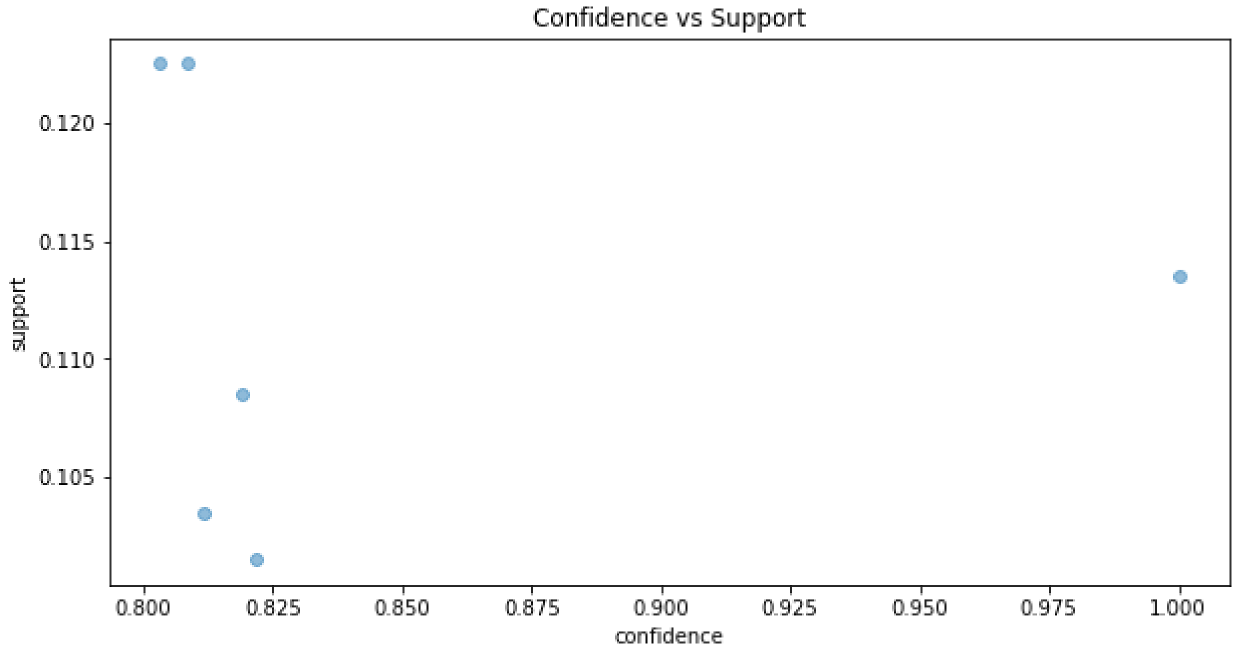
In [13]:
```
plt.figure(figsize=(10,5))
plt.scatter(rules1b['confidence'], rules1b['lift'], alpha=0.5)
plt.xlabel('confidence')
```

```
plt.ylabel('lift')
plt.title('Confidence vs Lift')
plt.show()
```



In [14]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules1b['confidence'], rules1b['support'], alpha=0.5)
plt.xlabel('confidence')
plt.ylabel('support')
plt.title('Confidence vs Support')
plt.show()
```



In [15]:

```
# using support as metric and keeping the value as .1
rules1c = association_rules(frequent_book1, metric="support", min_threshold=.1)
rules1c.sort_values('lift',ascending = False).head(10)
```

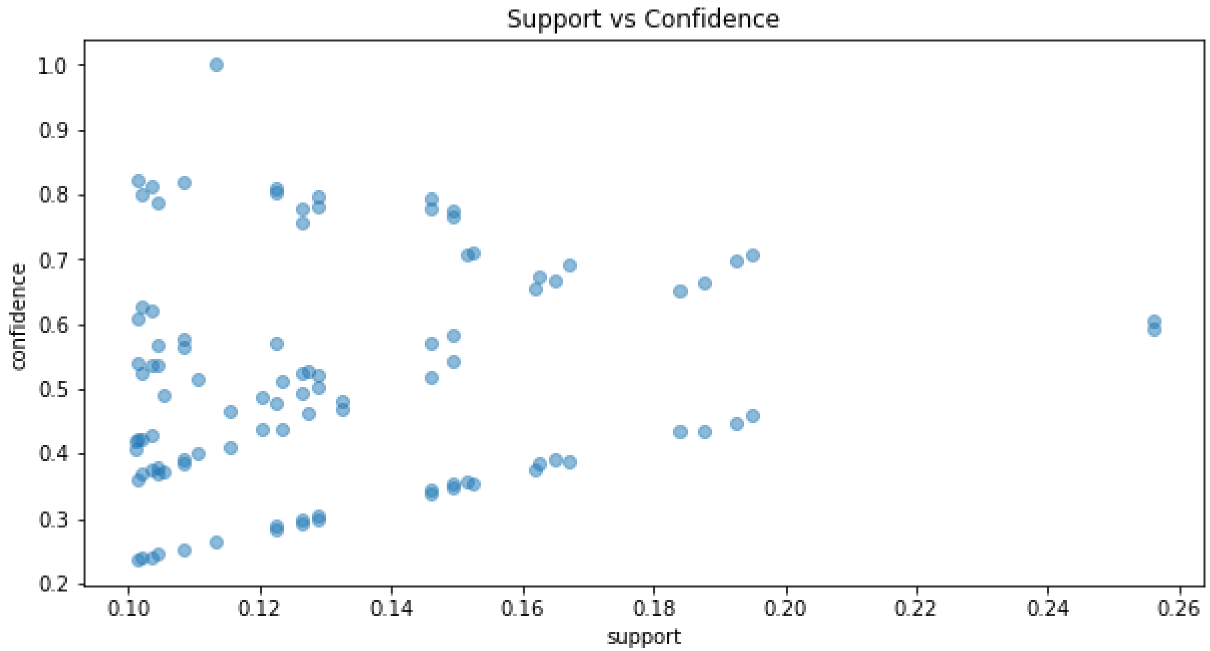
Out[15]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
28	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.064582	1.203406
29	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582	inf
77	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150	1.944628
80	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.057150	1.328448
86	(ArtBks)	(CookBks, DoItYBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.056313	1.403674
83	(CookBks, DoItYBks)	(ArtBks)	0.1875	0.2410	0.1015	0.541333	2.246196	0.056313	1.654797
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408	1.332800
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057408	1.904063
99	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107	1.415327

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
94	(CookBks, GeogBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107	1.641657

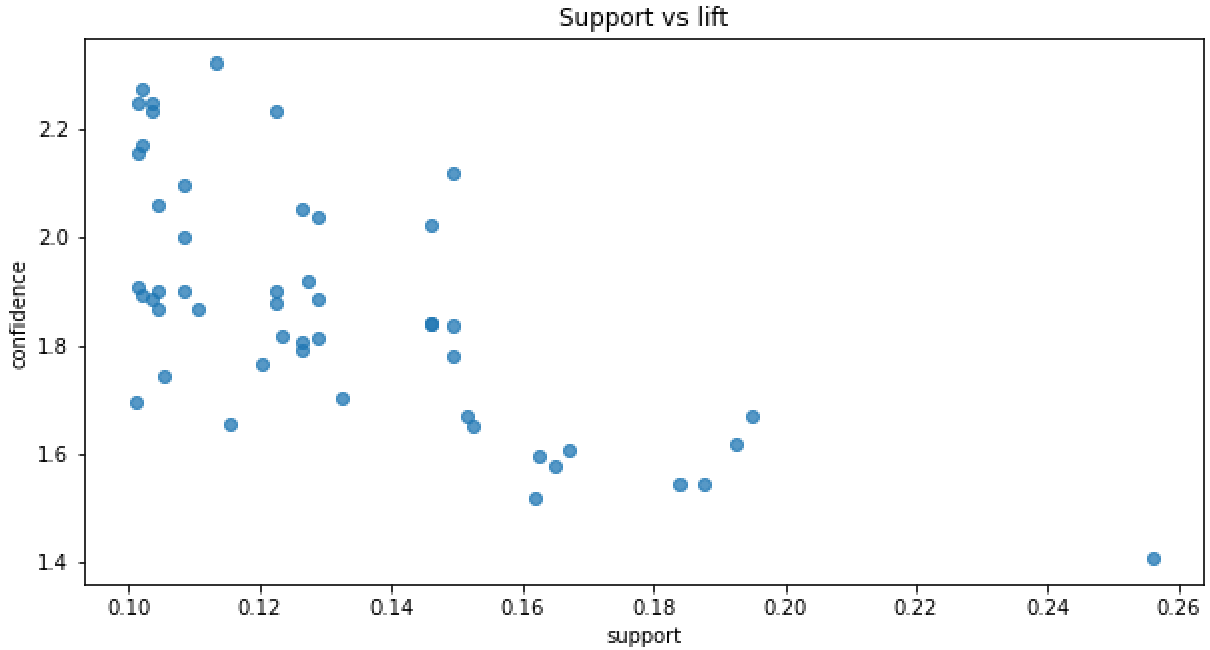
In [16]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules1c['support'], rules1c['confidence'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs Confidence')
plt.show()
```



In [17]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules1c['support'], rules1c['lift'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs lift')
plt.show()
```



Apriori Algorithm with 20 % minimum support

In [18]:

```
frequent_book2 = apriori(book, min_support=0.2, use_colnames=True)
frequent_book2
```

Out[18]:

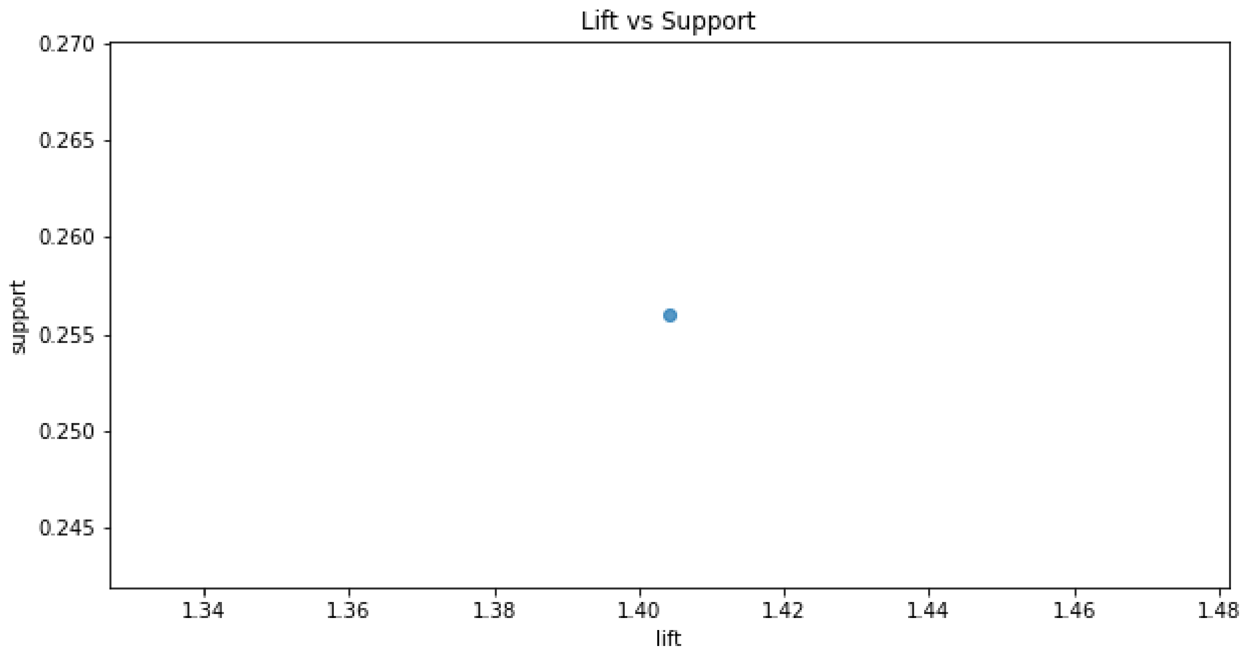
	support	itemsets
0	0.4230	(ChildBks)
1	0.2475	(YouthBks)
2	0.4310	(CookBks)
3	0.2820	(DoItYBks)
4	0.2145	(RefBks)
5	0.2410	(ArtBks)
6	0.2760	(GeogBks)
7	0.2560	(CookBks, ChildBks)

In [19]:
```
# using Lift as metric and keeping the value as 0.5
rules2a = association_rules(frequent_book2, metric="lift", min_threshold=0.5)
rules2a.sort_values('lift',ascending = False).head(10)
```

Out[19]:

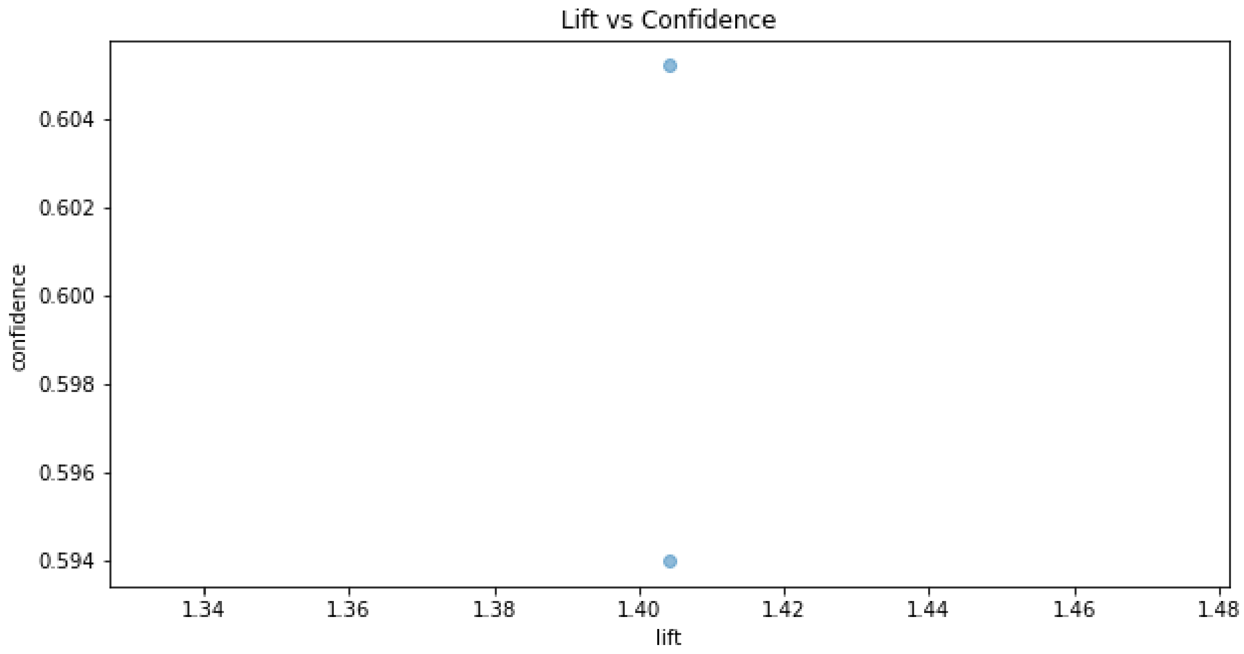
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(CookBks)	(ChildBks)	0.431	0.423	0.256	0.593968	1.404179	0.073687	1.421069
1	(ChildBks)	(CookBks)	0.423	0.431	0.256	0.605201	1.404179	0.073687	1.441240

In [20]:
```
plt.figure(figsize=(10,5))
plt.scatter(rules2a['lift'], rules2a['support'], alpha=0.5)
plt.xlabel('lift')
plt.ylabel('support')
plt.title('Lift vs Support')
plt.show()
```



In [21]:
```
plt.figure(figsize=(10,5))
```

```
plt.scatter(rules2a['lift'], rules2a['confidence'], alpha=0.5)
plt.xlabel('lift')
plt.ylabel('confidence')
plt.title('Lift vs Confidence')
plt.show()
```



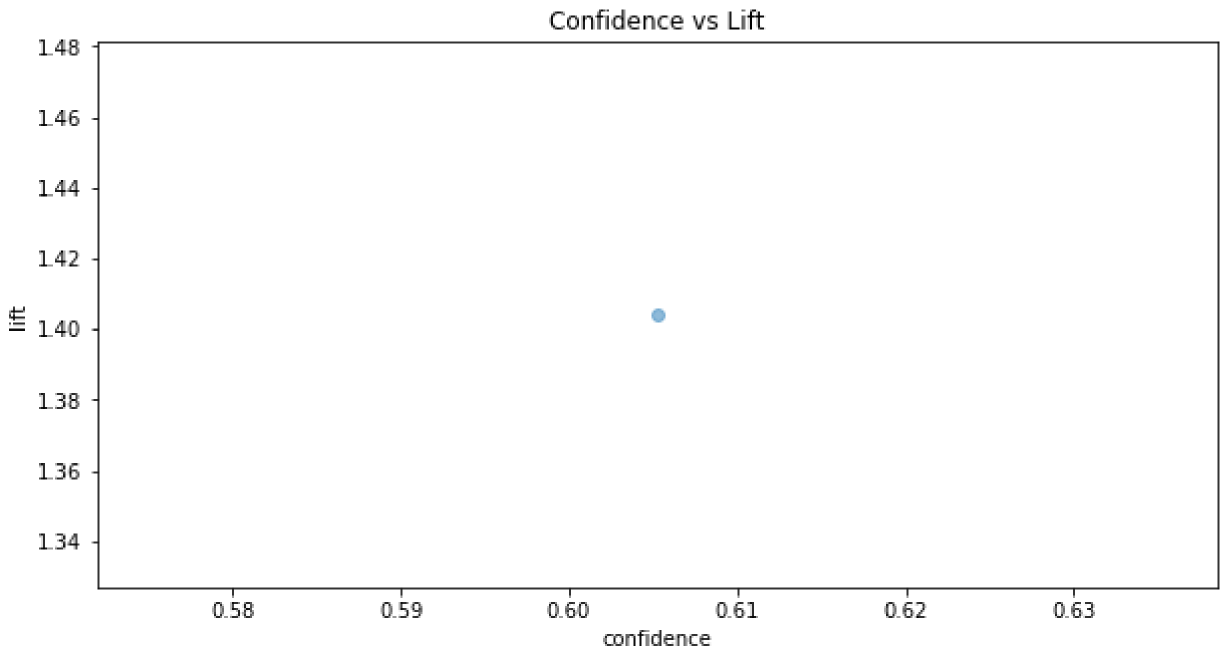
In [22]:
```
# using confidence as metric and keeping the value as 0.6
rules2b = association_rules(frequent_book2, metric="confidence", min_threshold=.6)
rules2b.sort_values('lift',ascending = False).head(10)
```

Out[22]:

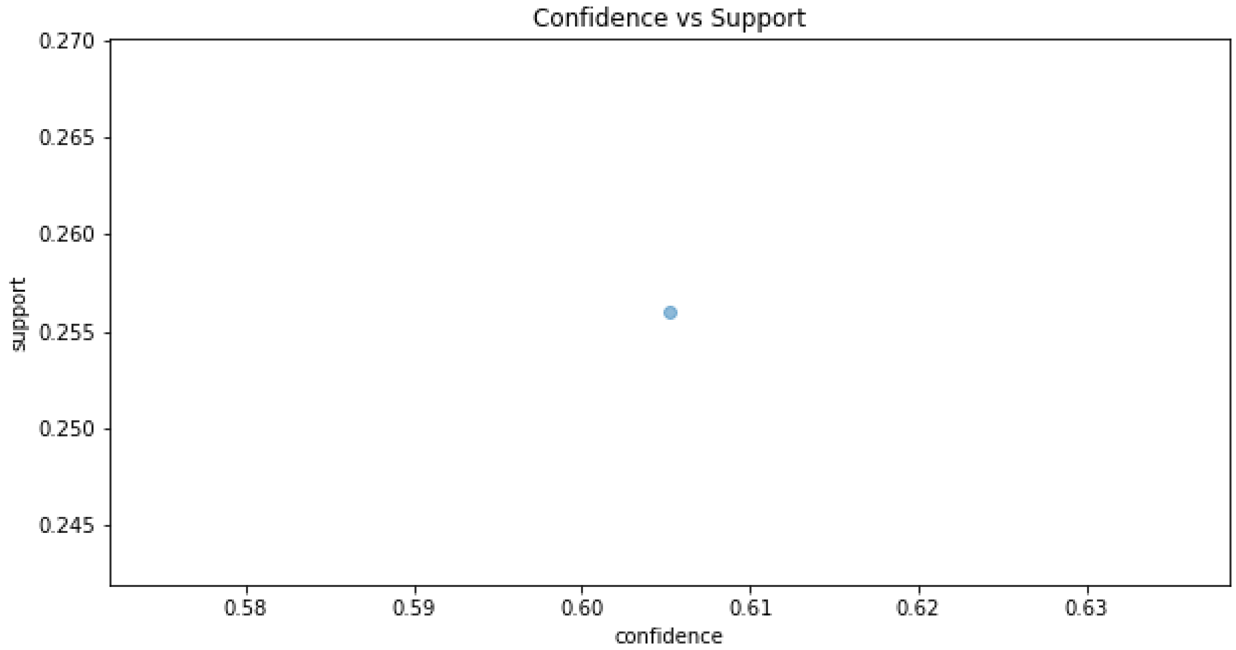
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ChildBks)	(CookBks)	0.423	0.431	0.256	0.605201	1.404179	0.073687	1.44124

In [23]:
```
plt.figure(figsize=(10,5))
plt.scatter(rules2b['confidence'], rules2b['lift'], alpha=0.5)
plt.xlabel('confidence')
plt.ylabel('lift')
```

```
plt.title('Confidence vs Lift')
plt.show()
```



In [25]:
```
plt.figure(figsize=(10,5))
plt.scatter(rules2b['confidence'], rules2b['support'], alpha=0.5)
plt.xlabel('confidence')
plt.ylabel('support')
plt.title('Confidence vs Support')
plt.show()
```

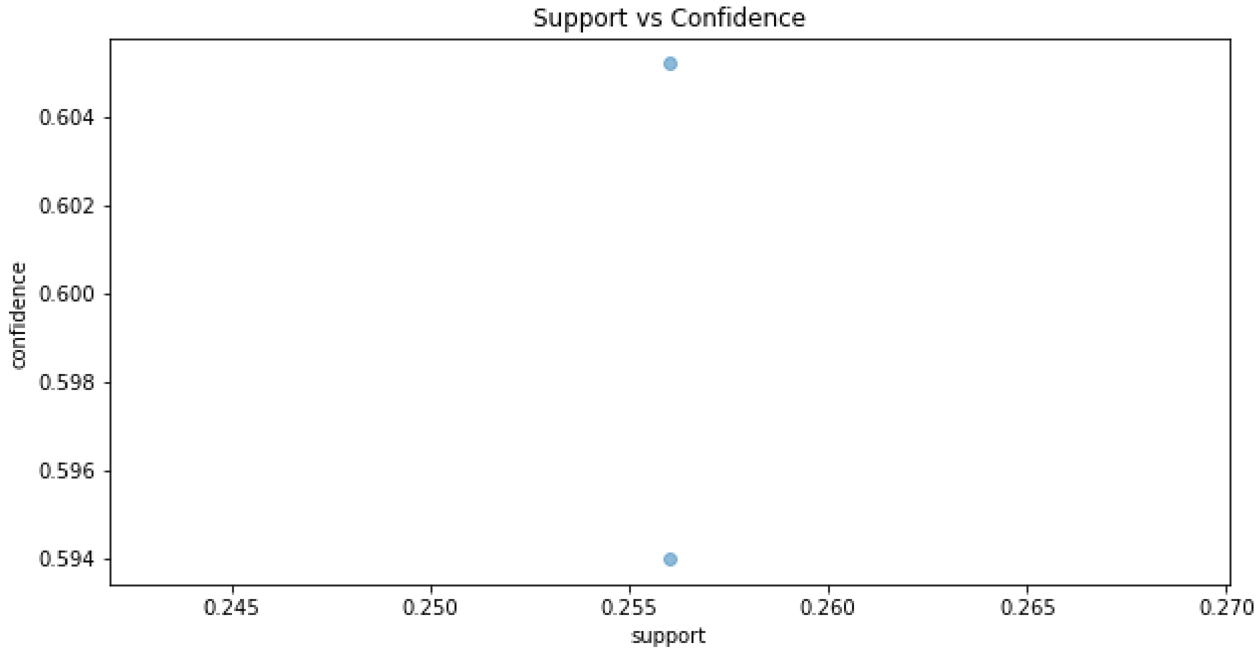


In [26]:
```
# using support as metric and keeping the value as .2
rules2c = association_rules(frequent_book2, metric="support", min_threshold=.2)
rules2c.sort_values('lift',ascending = False).head(10)
```

Out[26]:

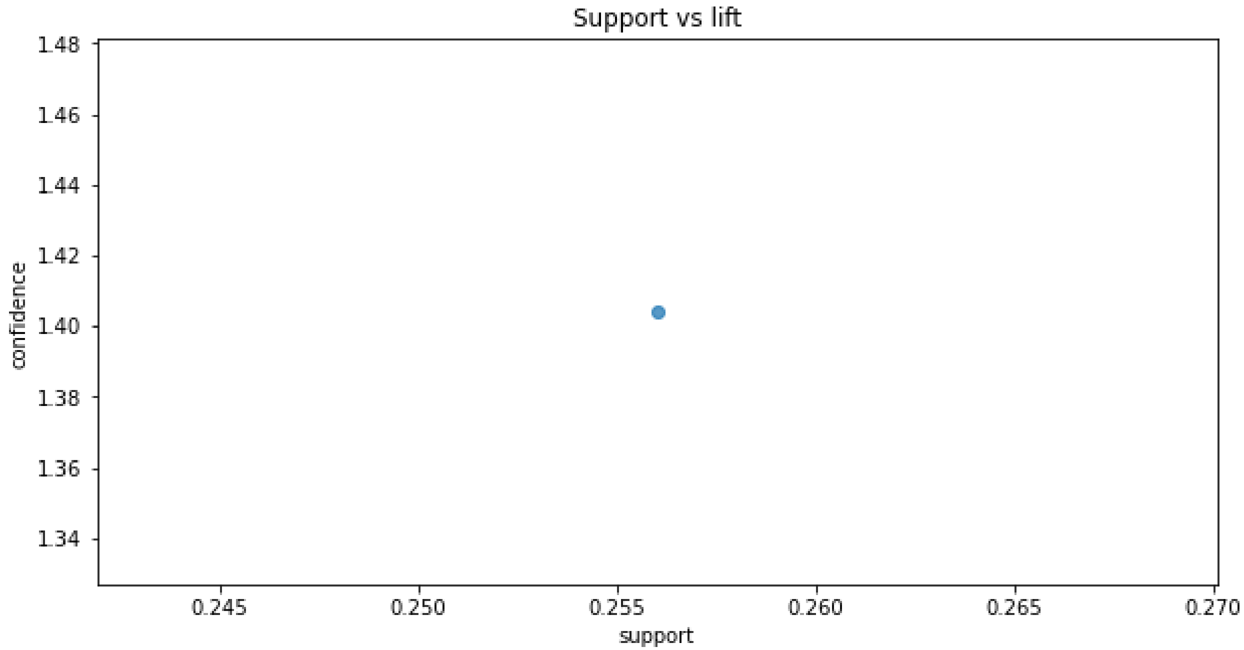
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(CookBks)	(ChildBks)	0.431	0.423	0.256	0.593968	1.404179	0.073687	1.421069
1	(ChildBks)	(CookBks)	0.423	0.431	0.256	0.605201	1.404179	0.073687	1.441240

In [27]:
```
plt.figure(figsize=(10,5))
plt.scatter(rules2c['support'], rules2c['confidence'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs Confidence')
plt.show()
```



In [28]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules2c['support'], rules2c['lift'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs lift')
plt.show()
```



Apriori Algorithm with 15 % minimum support

In [29]:
```
frequent_book3 = apriori(book, min_support=0.15, use_colnames=True)
frequent_book3
```

Out[29]:

	support	itemsets
0	0.4230	(ChildBks)
1	0.2475	(YouthBks)
2	0.4310	(CookBks)
3	0.2820	(DoItYBks)
4	0.2145	(RefBks)
5	0.2410	(ArtBks)
6	0.2760	(GeogBks)
7	0.1650	(ChildBks, YouthBks)

	support	itemsets
8	0.2560	(CookBks, ChildBks)
9	0.1840	(ChildBks, DoItYBks)
10	0.1515	(ChildBks, RefBks)
11	0.1625	(ChildBks, ArtBks)
12	0.1950	(ChildBks, GeogBks)
13	0.1620	(CookBks, YouthBks)
14	0.1875	(CookBks, DoItYBks)
15	0.1525	(CookBks, RefBks)
16	0.1670	(CookBks, ArtBks)
17	0.1925	(CookBks, GeogBks)

In [30]:

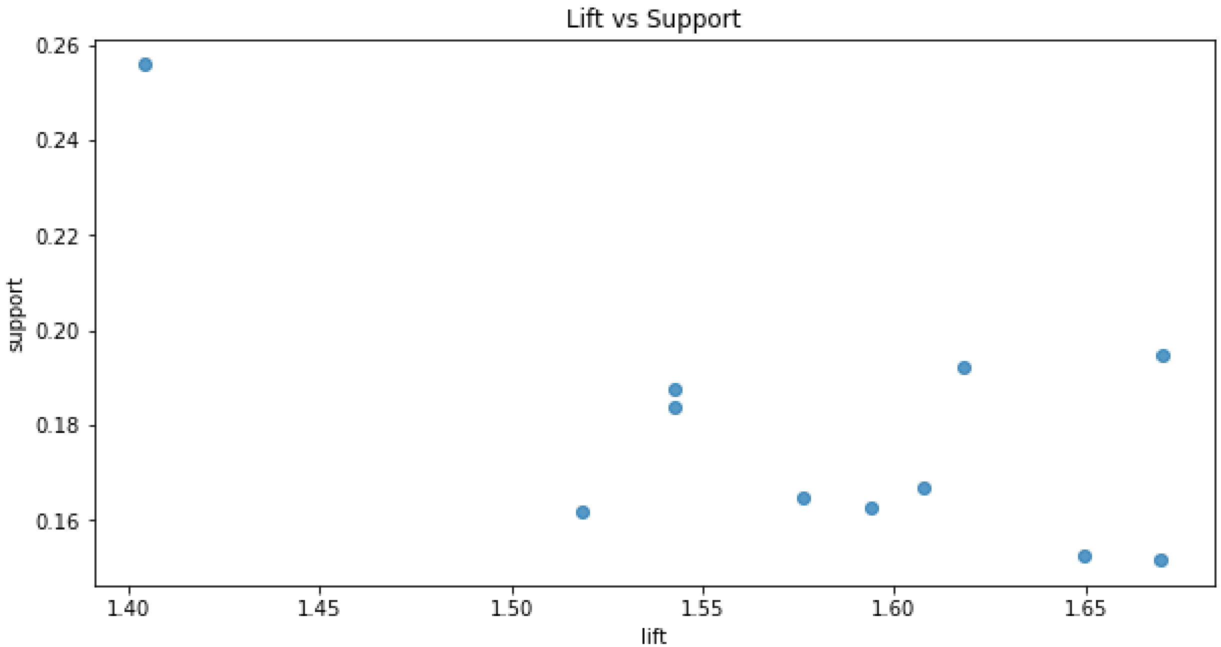
```
# using Lift as metric and keeping the value as 0.75
rules3a = association_rules(frequent_book3, metric="lift", min_threshold=0.75)
rules3a.sort_values('lift',ascending = False).head(10)
```

Out[30]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
10	(ChildBks)	(GeogBks)	0.4230	0.2760	0.1950	0.460993	1.670264	0.078252	1.343211
11	(GeogBks)	(ChildBks)	0.2760	0.4230	0.1950	0.706522	1.670264	0.078252	1.966074
6	(ChildBks)	(RefBks)	0.4230	0.2145	0.1515	0.358156	1.669725	0.060767	1.223818
7	(RefBks)	(ChildBks)	0.2145	0.4230	0.1515	0.706294	1.669725	0.060767	1.964548
17	(RefBks)	(CookBks)	0.2145	0.4310	0.1525	0.710956	1.649549	0.060050	1.968556
16	(CookBks)	(RefBks)	0.4310	0.2145	0.1525	0.353828	1.649549	0.060050	1.215621
20	(CookBks)	(GeogBks)	0.4310	0.2760	0.1925	0.446636	1.618245	0.073544	1.308361
21	(GeogBks)	(CookBks)	0.2760	0.4310	0.1925	0.697464	1.618245	0.073544	1.880766
18	(CookBks)	(ArtBks)	0.4310	0.2410	0.1670	0.387471	1.607763	0.063129	1.239125
19	(ArtBks)	(CookBks)	0.2410	0.4310	0.1670	0.692946	1.607763	0.063129	1.853095

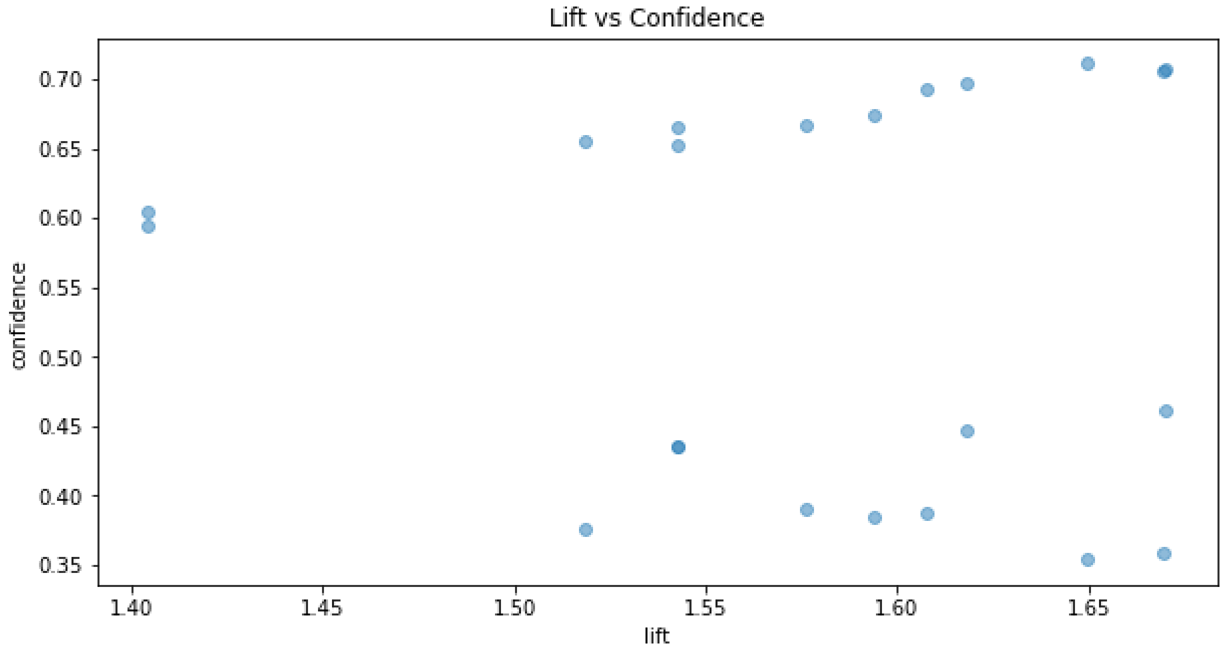
In [31]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules3a['lift'], rules3a['support'], alpha=0.5)
plt.xlabel('lift')
plt.ylabel('support')
plt.title('Lift vs Support')
plt.show()
```



In [32]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules3a['lift'], rules3a['confidence'], alpha=0.5)
plt.xlabel('lift')
plt.ylabel('confidence')
plt.title('Lift vs Confidence')
plt.show()
```



In [33]:

```
# using confidence as metric and keeping the value as 0.3
rules3b = association_rules(frequent_book3, metric="confidence", min_threshold=0.30)
rules3b.sort_values('lift',ascending = False).head(10)
```

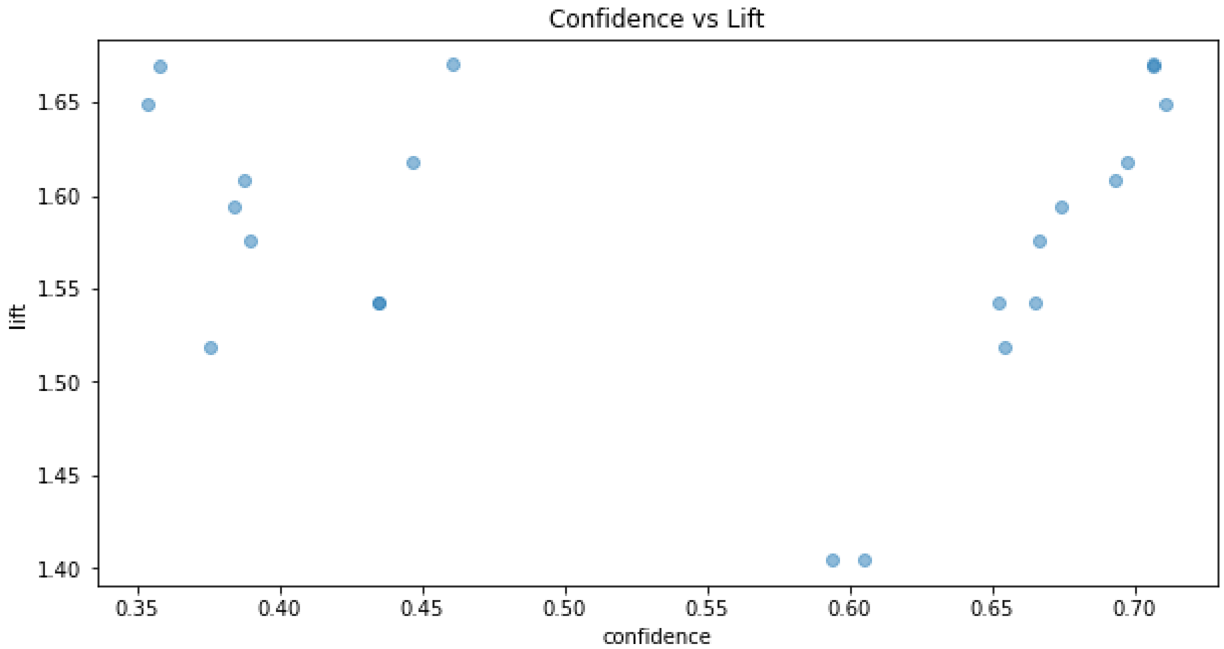
Out[33]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
10	(ChildBks)	(GeogBks)	0.4230	0.2760	0.1950	0.460993	1.670264	0.078252	1.343211
11	(GeogBks)	(ChildBks)	0.2760	0.4230	0.1950	0.706522	1.670264	0.078252	1.966074
6	(ChildBks)	(RefBks)	0.4230	0.2145	0.1515	0.358156	1.669725	0.060767	1.223818
7	(RefBks)	(ChildBks)	0.2145	0.4230	0.1515	0.706294	1.669725	0.060767	1.964548
17	(RefBks)	(CookBks)	0.2145	0.4310	0.1525	0.710956	1.649549	0.060050	1.968556
16	(CookBks)	(RefBks)	0.4310	0.2145	0.1525	0.353828	1.649549	0.060050	1.215621
20	(CookBks)	(GeogBks)	0.4310	0.2760	0.1925	0.446636	1.618245	0.073544	1.308361
21	(GeogBks)	(CookBks)	0.2760	0.4310	0.1925	0.697464	1.618245	0.073544	1.880766
18	(CookBks)	(ArtBks)	0.4310	0.2410	0.1670	0.387471	1.607763	0.063129	1.239125

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
19	(ArtBks)	(CookBks)	0.2410	0.4310	0.1670	0.692946	1.607763	0.063129	1.853095

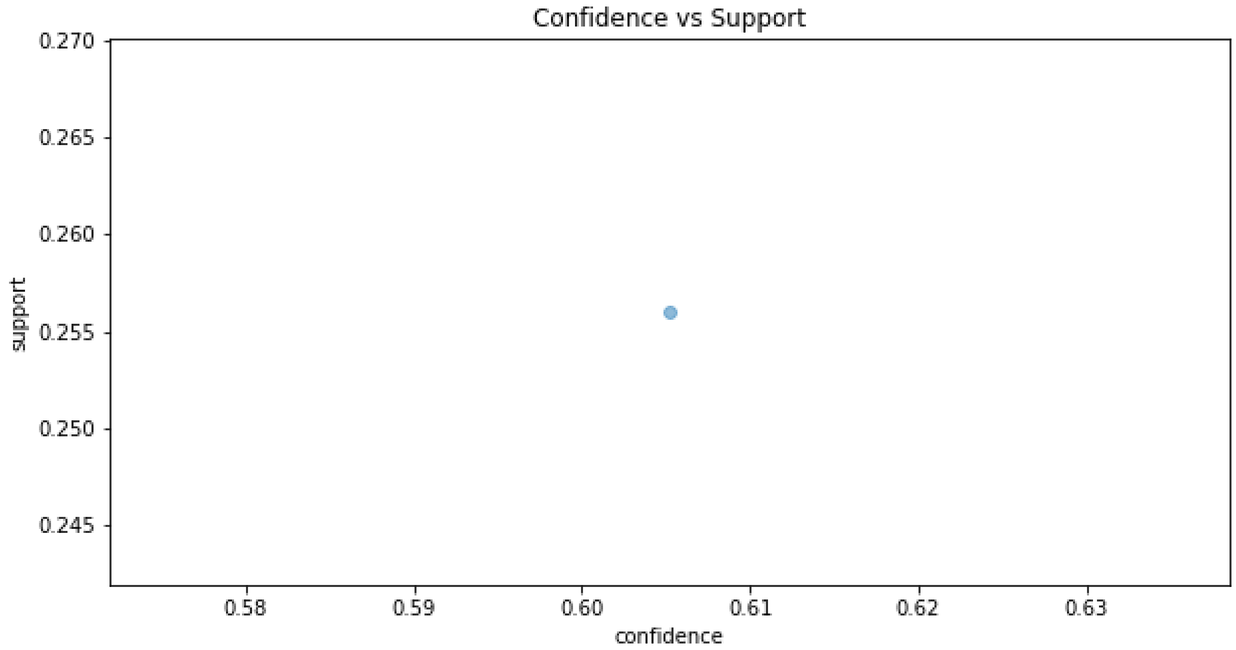
In [34]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules3b['confidence'], rules3b['lift'], alpha=0.5)
plt.xlabel('confidence')
plt.ylabel('lift')
plt.title('Confidence vs Lift')
plt.show()
```



In [35]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules2b['confidence'], rules2b['support'], alpha=0.5)
plt.xlabel('confidence')
plt.ylabel('support')
plt.title('Confidence vs Support')
plt.show()
```



In [36]:

```
# using support as metric and keeping the value as .15
rules3c = association_rules(frequent_book3, metric="support", min_threshold=.15)
rules3c.sort_values('lift',ascending = False).head(10)
```

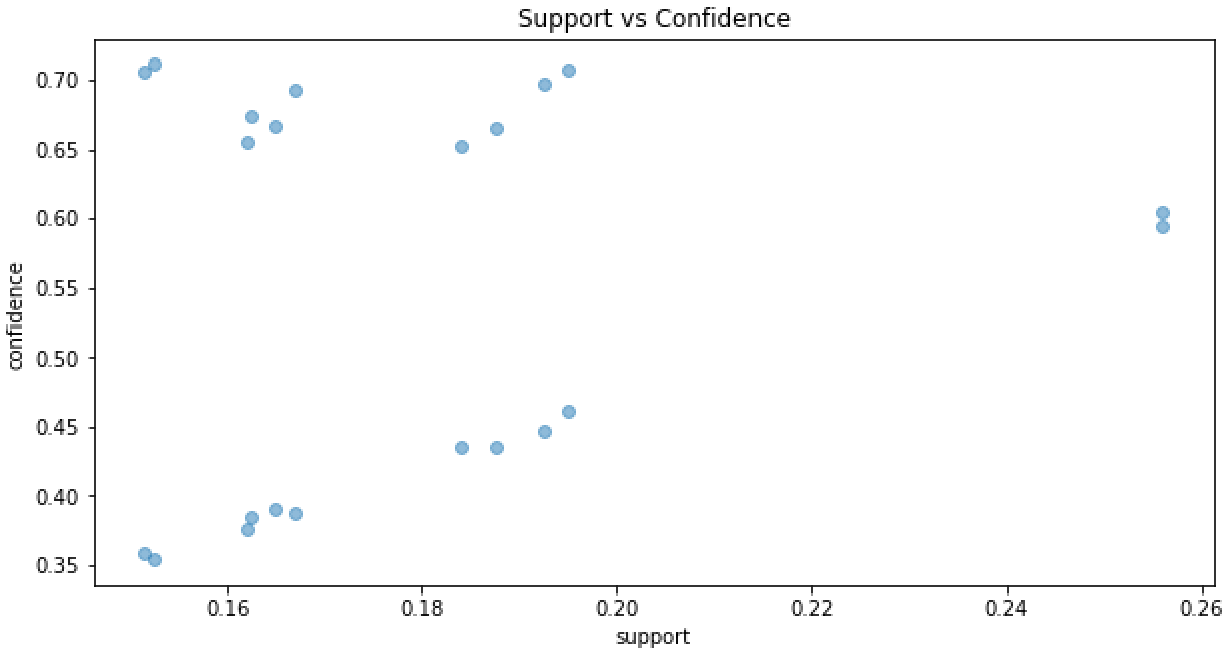
Out[36]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
10	(ChildBks)	(GeogBks)	0.4230	0.2760	0.1950	0.460993	1.670264	0.078252	1.343211
11	(GeogBks)	(ChildBks)	0.2760	0.4230	0.1950	0.706522	1.670264	0.078252	1.966074
6	(ChildBks)	(RefBks)	0.4230	0.2145	0.1515	0.358156	1.669725	0.060767	1.223818
7	(RefBks)	(ChildBks)	0.2145	0.4230	0.1515	0.706294	1.669725	0.060767	1.964548
17	(RefBks)	(CookBks)	0.2145	0.4310	0.1525	0.710956	1.649549	0.060050	1.968556
16	(CookBks)	(RefBks)	0.4310	0.2145	0.1525	0.353828	1.649549	0.060050	1.215621
20	(CookBks)	(GeogBks)	0.4310	0.2760	0.1925	0.446636	1.618245	0.073544	1.308361
21	(GeogBks)	(CookBks)	0.2760	0.4310	0.1925	0.697464	1.618245	0.073544	1.880766
18	(CookBks)	(ArtBks)	0.4310	0.2410	0.1670	0.387471	1.607763	0.063129	1.239125

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
19	(ArtBks)	(CookBks)	0.2410	0.4310	0.1670	0.692946	1.607763	0.063129	1.853095

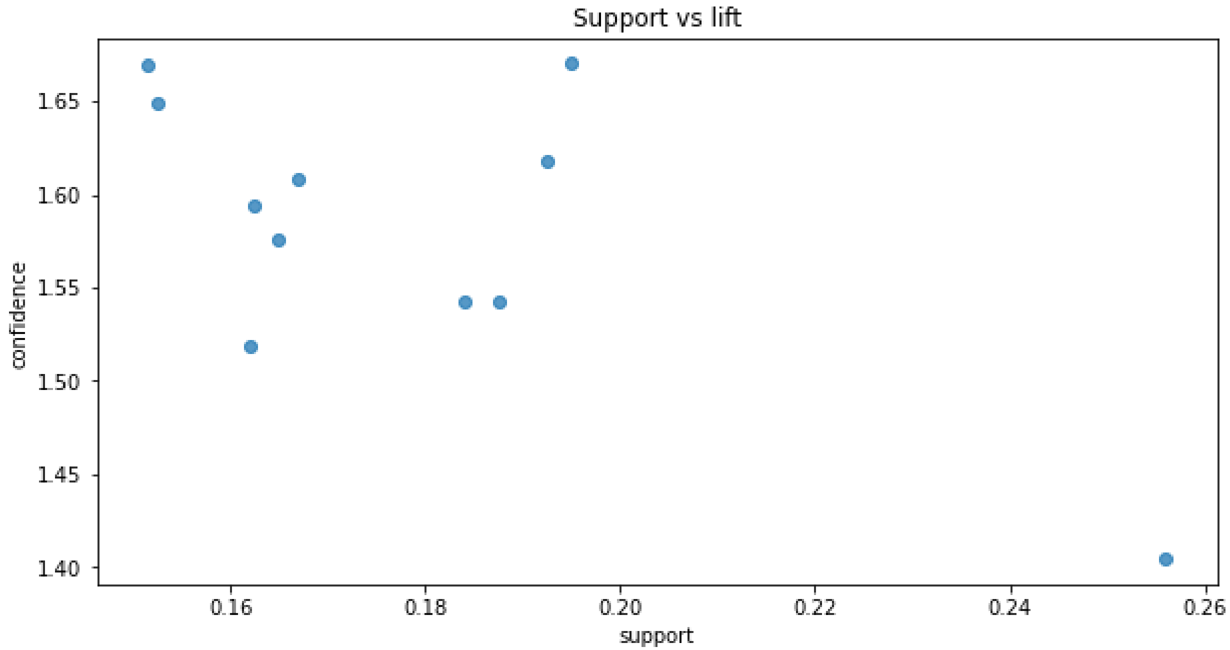
In [37]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules3c['support'], rules3c['confidence'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs Confidence')
plt.show()
```



In [38]:

```
plt.figure(figsize=(10,5))
plt.scatter(rules3c['support'], rules3c['lift'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs lift')
plt.show()
```



Support vs lift
confidence
support
1.65
1.60
1.55
1.50
1.45
1.40
0.16
0.18
0.20
0.22
0.24
0.26

In [ ]: