Importing and exploring the dataset

We will use the built-in dataset available in the pyECLAT module. Let us first import the pyECLAT module and the build-in dataset.

```
from pyECLAT import Example1
dataset = Example1().get()
dataset.head()
```

```
       0      1       2       3
0   milk   beer   bread  butter
1  coffe  bread  butter     NaN
2  coffe  bread  butter     NaN
3   milk  coffe   bread  butter
4   beer    NaN     NaN     NaN
```

Each row represents a customer's purchase at a supermarket in this dataset. For example, in row 1, the customer purchased only burgers, meatballs, and eggs. Let's get more information about the dataset by printing more details.

```
# printing the info
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   0       10 non-null     object
 1   1       5 non-null      object
 2   2       4 non-null      object
 3   3       2 non-null      object
dtypes: object(4)
memory usage: 448.0+ bytes
```

Visualizing the frequent items

To visualize the frequent items, let's load the dataset to the ECLAT class and generate binary DataFrame:

```
# importing the ECLAT module
from pyECLAT import ECLAT

# loading transactions DataFrame to ECLAT class
eclat = ECLAT(data=dataset)

# DataFrame of binary values
eclat.df_bin
```

```
   bean  bread  butter  rice  beer  coffe  milk
0     0      1       1     0     1      0     1
1     0      1       1     0     0      1     0
2     0      1       1     0     0      1     0
3     0      1       1     0     0      1     1
4     0      0       0     0     1      0     0
5     0      0       1     0     0      0     0
6     0      1       0     0     0      0     0
7     1      0       0     0     0      0     0
8     1      0       0     1     0      0     0
9     0      0       0     1     0      0     0
```

In this binary dataset, every row represents a transaction. Columns are possible products that might appear in every transaction. Every cell contains one of two possible values:

0 – the product was not included in the transaction

1 – the transaction contains the product

Now, we need to count items for every column in the DataFrame:

```
# count items in each column
items_total = eclat.df_bin.astype(int).sum(axis=0)

items_total
```

```
bean      2
bread     5
butter    5
rice      2
beer      2
coffe     3
milk      2
dtype: int64
```

```
# count items in each row
items_per_transaction = eclat.df_bin.astype(int).sum(axis=1)

items_per_transaction
```

```
0    4
1    3
2    3
3    4
4    1
5    1
6    1
7    1
8    2
9    1
dtype: int64
```

```
import pandas as pd

# Loading items per column stats to the DataFrame
df = pd.DataFrame({'items': items_total.index, 'transactions':
items_total.values})

# cloning pandas DataFrame for visualization purpose
df_table = df.sort_values("transactions", ascending=False)

#  Top 5 most popular products/items
df_table.head(5).style.background_gradient(cmap='Blues')
```

```
<pandas.io.formats.style.Styler at 0x2f5103ccd90>
```

```
# importing required module
import plotly.express as px

# to have a same origin
df_table["all"] = "Tree Map"

# creating tree map using plotly
fig = px.treemap(df_table.head(50), path=['all', "items"],
values='transactions',
                 color=df_table["transactions"].head(50),
hover_data=['items'],
                 color_continuous_scale='Blues',
                )
# ploting the treemap
fig.show()
```


```
{"data":[{"parents":["Tree Map","Tree Map","Tree Map","Tree Map","Tree
Map","Tree Map","Tree Map",""],"ids":["Tree Map/bean","Tree
Map/beer","Tree Map/bread","Tree Map/butter","Tree Map/coffe","Tree
Map/milk","Tree Map/rice","Tree Map"],"hovertemplate":"labels=%
{label}<br>transactions=%{value}<br>parent=%{parent}<br>id=%
{id}<br>items=%{customdata[0]}<br>color=%{color}<extra></
extra>","customdata":[["bean",2],["beer",2],["bread",5],["butter",5],
["coffe",3],["milk",2],["rice",2],
["(?)",3.5714285714285716]],"values":[2,2,5,5,3,2,2,21],"domain":{"y":
[0,1],"x":[0,1]},"labels":
["bean","beer","bread","butter","coffe","milk","rice","Tree
Map"],"name":"","branchvalues":"total","type":"treemap","marker":
{"colors":
[2,2,5,5,3,2,2,3.5714285714285716],"coloraxis":"coloraxis"}}],"config"
:{"plotlyServerURL":"https://plot.ly"},"layout":{"coloraxis":
{"colorbar":{"title":{"text":"color"}},"colorscale":
[[0,"rgb(247,251,255)"],[0.125,"rgb(222,235,247)"],
[0.25,"rgb(198,219,239)"],[0.375,"rgb(158,202,225)"],
[0.5,"rgb(107,174,214)"],[0.625,"rgb(66,146,198)"],
[0.75,"rgb(33,113,181)"],[0.875,"rgb(8,81,156)"],
```

```
[1,"rgb(8,48,107)"]]},"legend":{"tracegroupgap":0},"margin":
{"t":60},"template":{"data":{"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"scattermapbox"
:[{"type":"scattermapbox","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"pie":
[{"automargin":true,"type":"pie"}],"carpet":[{"aaxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"baxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"type":"carpet"}],"ba
r":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"type":"bar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"barpolar":
[{"type":"barpolar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"scatter3d":
[{"line":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d","marker":
{"colorbar":{"outlinewidth":0,"ticks":""}}}],"contour":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"histogram2d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"scatterpolar":
[{"type":"scatterpolar","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"histogram":
[{"type":"histogram","marker":{"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"histogram2dcontour
":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
```

```
[1,"#f0f921"]],"type":"histogram2dcontour"}],"parcoords":[{"line":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"scatterpolargl":
[{"type":"scatterpolargl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"scattercarpet":
[{"type":"scattercarpet","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"choropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"scatterternary":
[{"type":"scatterternary","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"scatter":[{"fillpattern":
{"solidity":0.2,"fillmode":"overlay","size":10},"type":"scatter"}],"ta
ble":[{"cells":{"fill":{"color":"#EBF0F8"},"line":
{"color":"white"}},"header":{"fill":{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}],"scattergeo":
[{"type":"scattergeo","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"surface":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"scattergl":
[{"type":"scattergl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}]},"layout":{"ternary":{"aaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"baxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"caxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"autotypenumbers":"strict","shapedefaults":{"line":
{"color":"#2a3f5f"}},"annotationdefaults":
{"arrowwidth":1,"arrowcolor":"#2a3f5f","arrowhead":0},"coloraxis":
{"colorbar":{"outlinewidth":0,"ticks":""}},"title":{"x":5.0e-
2},"hoverlabel":{"align":"left"},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
```

```
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","scene":{"zaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"xaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"yaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
},"font":{"color":"#2a3f5f"},"xaxis":{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"polar":{"angularaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"radialaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"plot_bgcolor":"#E5ECF6","geo":
{"subunitcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","sho
wland":true,"showlakes":true,"bgcolor":"white"},"yaxis":
{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"]}}}}
```


Generating association rules

To generate association rules, we need to define:

Minimum support – should be provided as a percentage of the overall items from the dataset

Minumum combinations – the minimum amount of items in the transaction

Maximum combinations – the minimum amount of items in the transaction

Note: the higher the value of the maximum combinations the longer the calculation will take.

```
# the item shoud appear at least at 5% of transactions
min_support = 10/100

# start from transactions containing at least 2 items
min_combination = 2

# up to maximum items per transaction
max_combination = max(items_per_transaction)

rule_indices, rule_supports = eclat.fit(min_support=min_support,
```

```
min_combination=min_combination,

max_combination=max_combination,
                                                    separator=' & ',
                                                    verbose=True)

Combination 2 by 2

21it [00:00, 138.35it/s]

Combination 3 by 3

35it [00:00, 223.09it/s]

Combination 4 by 4

35it [00:00, 159.73it/s]

import pandas as pd
result = pd.DataFrame(rule_supports.items(),columns=['Item',
'Support'])
result.sort_values(by=['Support'], ascending=False)
```

	Item	Support
1	bread & butter	0.4
3	bread & coffe	0.3
6	butter & coffe	0.3
11	bread & butter & coffe	0.3
4	bread & milk	0.2
7	butter & milk	0.2
12	bread & butter & milk	0.2
0	bean & rice	0.1
13	bread & beer & milk	0.1
17	bread & butter & beer & milk	0.1
16	butter & coffe & milk	0.1
15	butter & beer & milk	0.1
14	bread & coffe & milk	0.1
9	coffe & milk	0.1
10	bread & butter & beer	0.1
8	beer & milk	0.1
5	butter & beer	0.1
2	bread & beer	0.1
18	bread & butter & coffe & milk	0.1

```
# the item shoud appear at least at 5% of transactions
min_support = 20/100

# start from transactions containing at least 2 items
min_combination = 2

# up to maximum items per transaction
```

```
max_combination = max(items_per_transaction)

rule_indices, rule_supports = eclat.fit(min_support=min_support,

min_combination=min_combination,

max_combination=max_combination,
                                                    separator=' & ',
                                                    verbose=True)
```

```
Combination 2 by 2

21it [00:00, 171.53it/s]

Combination 3 by 3

35it [00:00, 213.12it/s]

Combination 4 by 4

35it [00:00, 171.02it/s]
```

```
import pandas as pd
result = pd.DataFrame(rule_supports.items(),columns=['Item',
'Support'])
result.sort_values(by=['Support'], ascending=False)
```

	Item	Support
0	bread & butter	0.4
1	bread & coffe	0.3
3	butter & coffe	0.3
5	bread & butter & coffe	0.3
2	bread & milk	0.2
4	butter & milk	0.2
6	bread & butter & milk	0.2