upsetplot Documentation

Release 0.6.0

Joel Nothman

Sep 02, 2021

Contents

This is another Python implementation of UpSet plots by Lex et al. [Lex2014]. UpSet plots are used to visualise set overlaps; like Venn diagrams but more readable. Documentation is at https://upsetplot.readthedocs.io.

This `upsetplot` library tries to provide a simple interface backed by an extensible, object-oriented design.

There are many ways to represent the categorisation of data, as covered in our Data Format Guide.

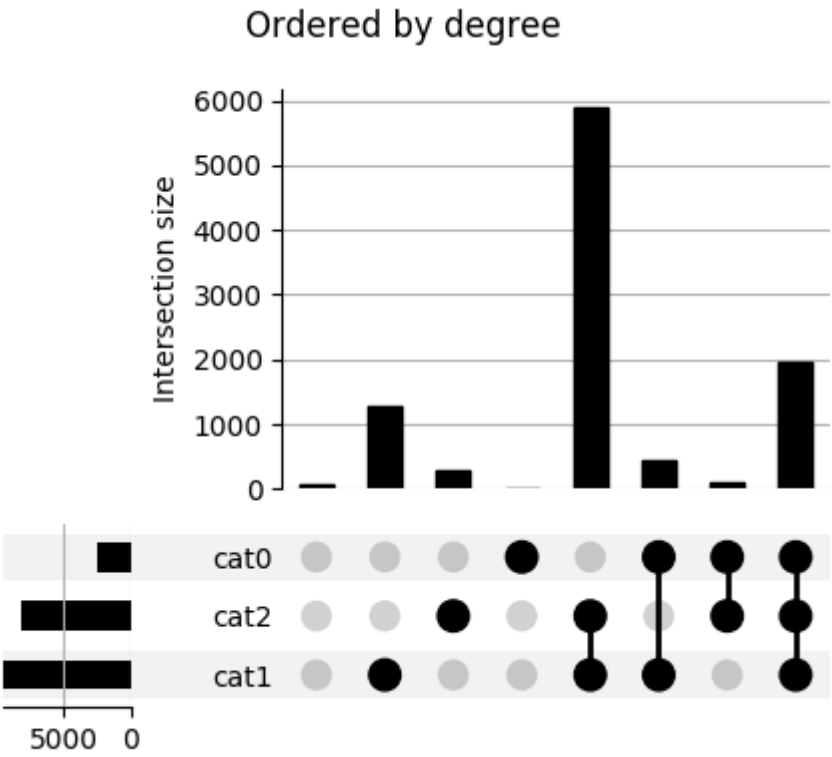
Our internal input format uses a `pandas.Series` containing counts corresponding to subset sizes, where each subset is an intersection of named categories. The index of the Series indicates which rows pertain to which categories, by having multiple boolean indices, like `example` in the following:

```
>>> from upsetplot import generate_counts
>>> example = generate_counts()
>>> example
cat0   cat1   cat2
False  False  False      56
              True      283
       True   False    1279
              True     5882
True   False  False      24
              True       90
       True   False     429
              True     1957
Name: value, dtype: int64
```

Then:

```
>>> from upsetplot import plot
>>> plot(example)
>>> from matplotlib import pyplot
>>> pyplot.show()
```

makes:

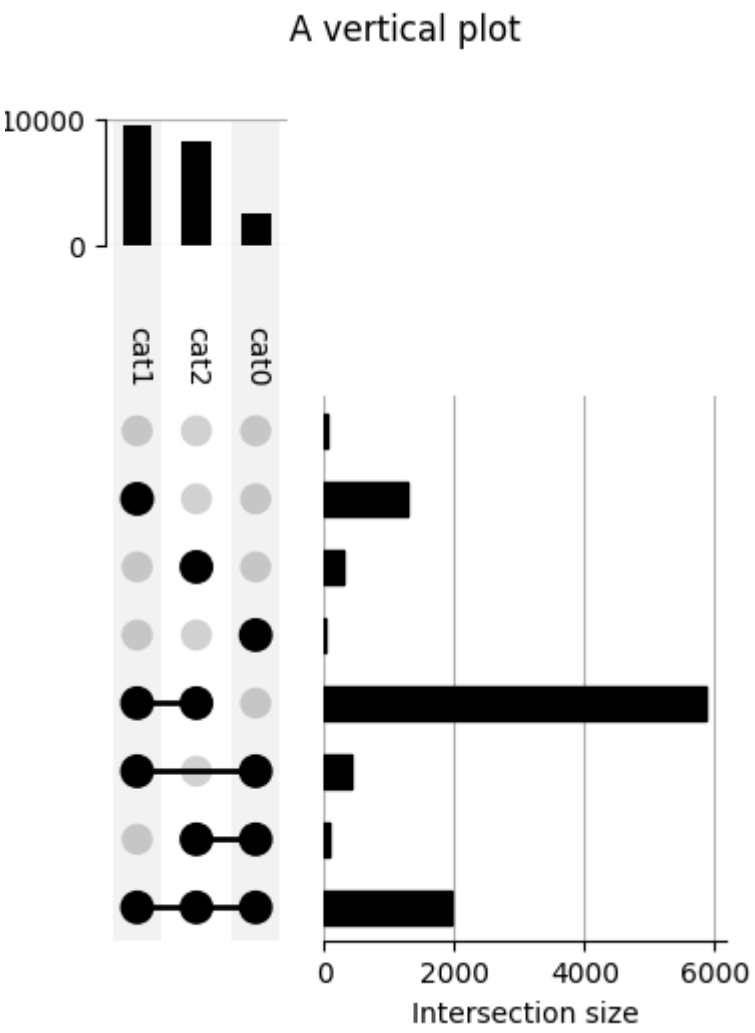


This plot shows the cardinality of every category combination seen in our data. The leftmost column counts items absent from any category. The next three columns count items only in `cat1`, `cat2` and `cat3` respectively, with following columns showing cardinalities for items in each combination of exactly two named sets. The rightmost column counts items in all three sets.

CHAPTER 1

Rotation

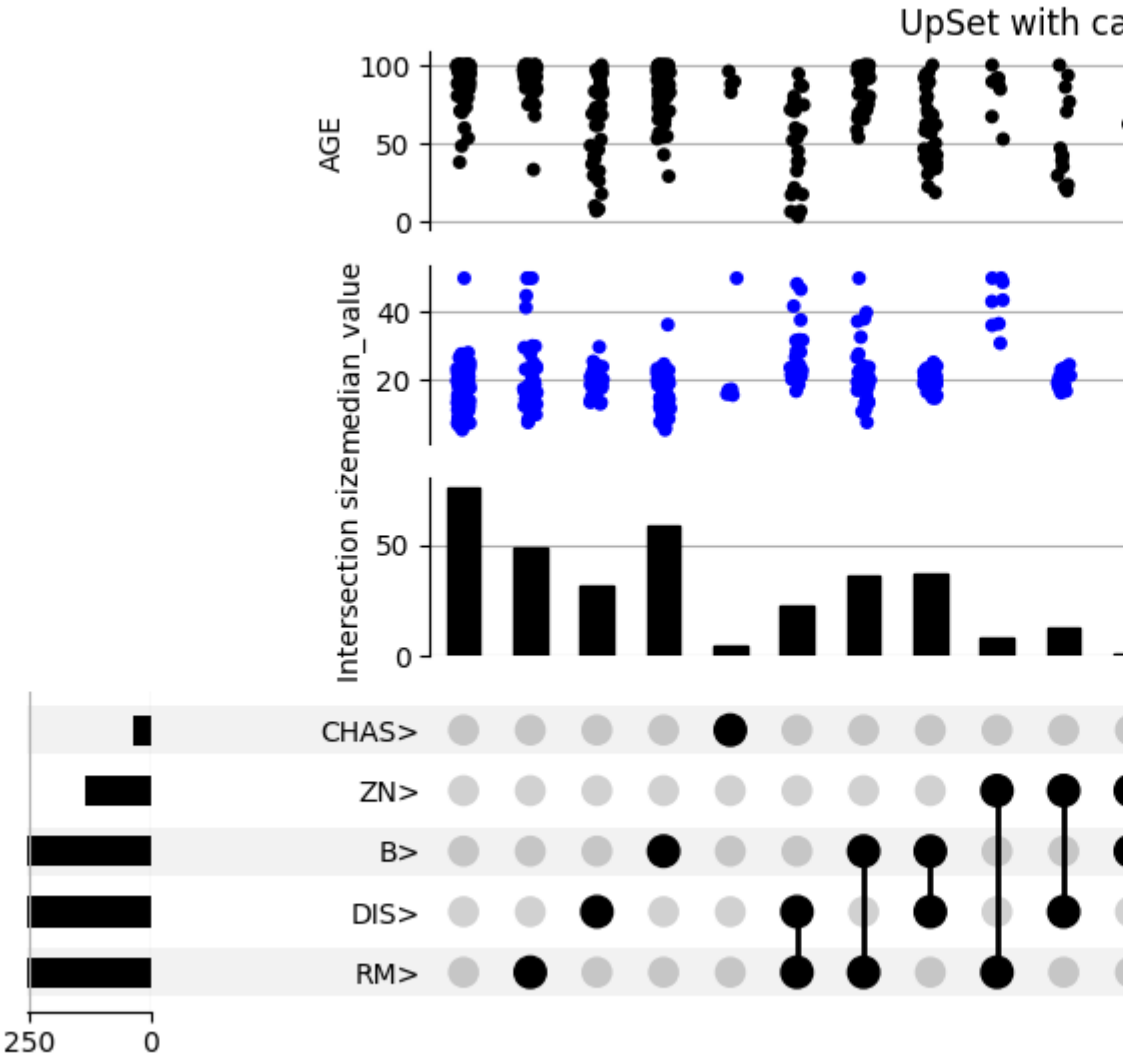
We call the above plot style “horizontal” because the category intersections are presented from left to right. Vertical plots are also supported!


A vertical plot
10000
0
cat1
cat2
cat0
0
2000
4000
6000
Intersection size

CHAPTER 2

Distributions

Providing a DataFrame rather than a Series as input allows us to expressively plot the distribution of variables in each subset.


UpSet with ca
AGE
100
50
0
Intersection sizemedian_value
40
20
50
0
CHAS>
ZN>
B>
DIS>
RM>
250
0

CHAPTER 3

Loading datasets

While the dataset above is randomly generated, you can prepare your own dataset for input to upsetplot. A helpful tool is *from_memberships*, which allows us to reconstruct the example above by indicating each data point's category membership:

```
>>> from upsetplot import from_memberships
>>> example = from_memberships(
...     [[],
...      ['cat2'],
...      ['cat1'],
...      ['cat1', 'cat2'],
...      ['cat0'],
...      ['cat0', 'cat2'],
...      ['cat0', 'cat1'],
...      ['cat0', 'cat1', 'cat2'],
...      ],
...      data=[56, 283, 1279, 5882, 24, 90, 429, 1957]
... )
>>> example
cat0   cat1   cat2
False  False  False      56
               True      283
       True   False    1279
               True     5882
True   False  False      24
               True       90
       True   False     429
               True     1957
dtype: int64
```

See also *from_contents*, another way to describe categorised data, and *from_indicators* which allows each category to be indicated by a column in the data frame (or a function of the column's data such as whether it is a missing value).

3.1 Installation

To install the library, you can use `pip`:

```
$ pip install upsetplot
```

Installation requires:

- pandas
- matplotlib >= 2.0
- seaborn to use *`UpSet.add_catplot`*

It should then be possible to:

```
>>> import upsetplot
```

in Python.

3.2 Why an alternative to py-upset?

Probably for petty reasons. It appeared py-upset was not being maintained. Its input format was undocumented, inefficient and, IMO, inappropriate. It did not facilitate showing plots of each subset's distribution as in Lex et al's work introducing UpSet plots. Nor did it include the horizontal bar plots illustrated there. It did not support Python 2. I decided it would be easier to construct a cleaner version than to fix it.

3.3 References

3.3.1 Examples

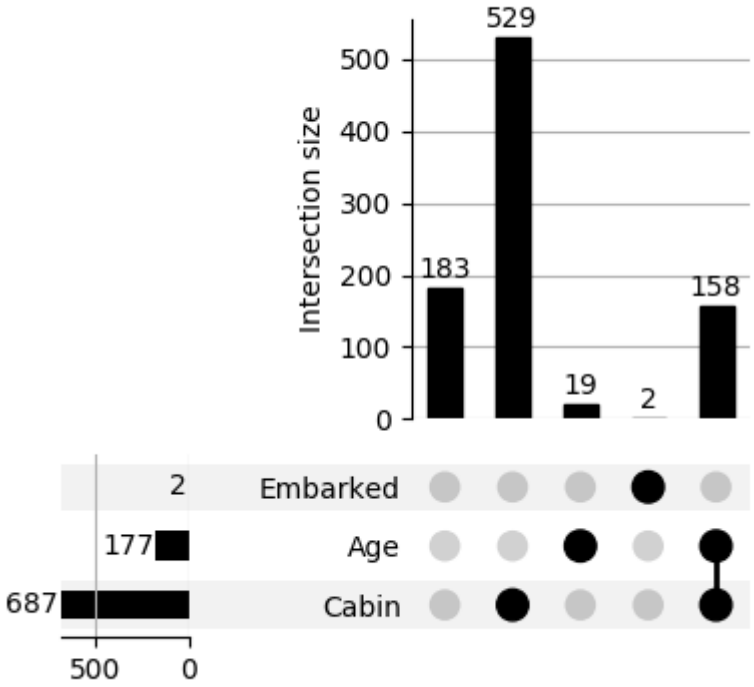
Introductory examples for upsetplot.

Note: Click *here* to download the full example code

Plot the distribution of missing values

UpSet plots are often used to show which variables are missing together.

Passing a callable `indicators=pd.isna` to `from_indicators()` is an easy way to categorise a record by the variables that are missing in it.




```
from matplotlib import pyplot as plt
import pandas as pd
from upsetplot import plot, from_indicators

TITANIC_URL = 'https://raw.githubusercontent.com/datasciencedojo/datasets/master/
↪titanic.csv'  # noqa
data = pd.read_csv(TITANIC_URL)

plot(from_indicators(indicators=pd.isna, data=data), show_counts=True)
plt.show()
```

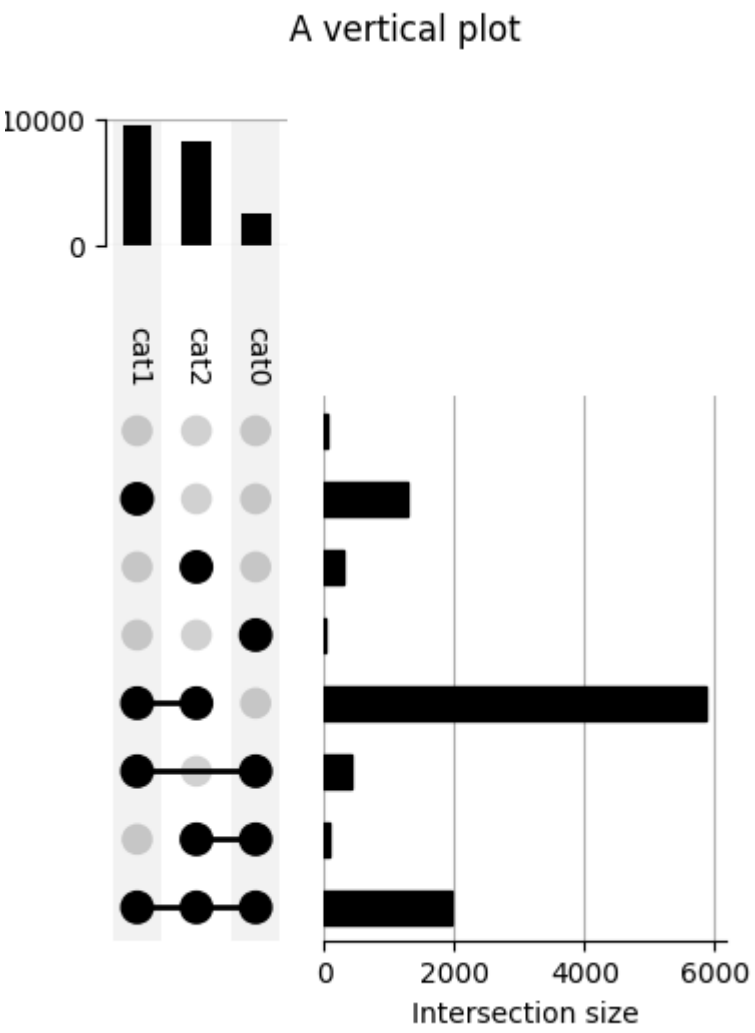
Total running time of the script: (0 minutes 0.383 seconds)

Note: Click *here* to download the full example code

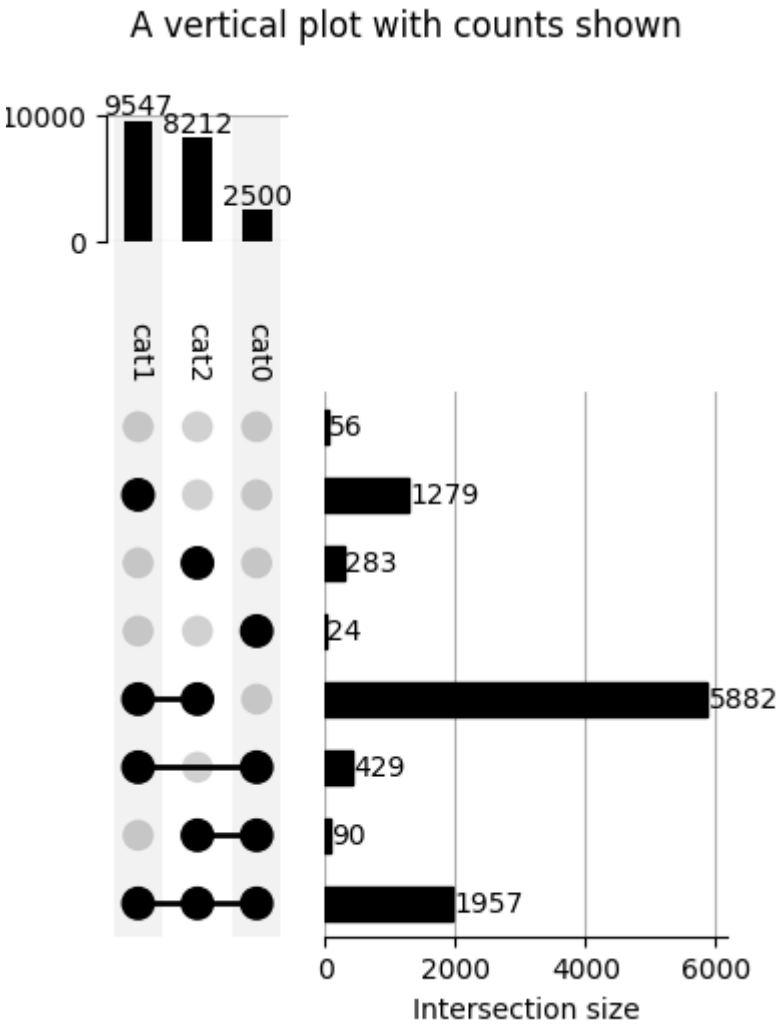
Vertical orientation

This illustrates the effect of orientation='vertical'.

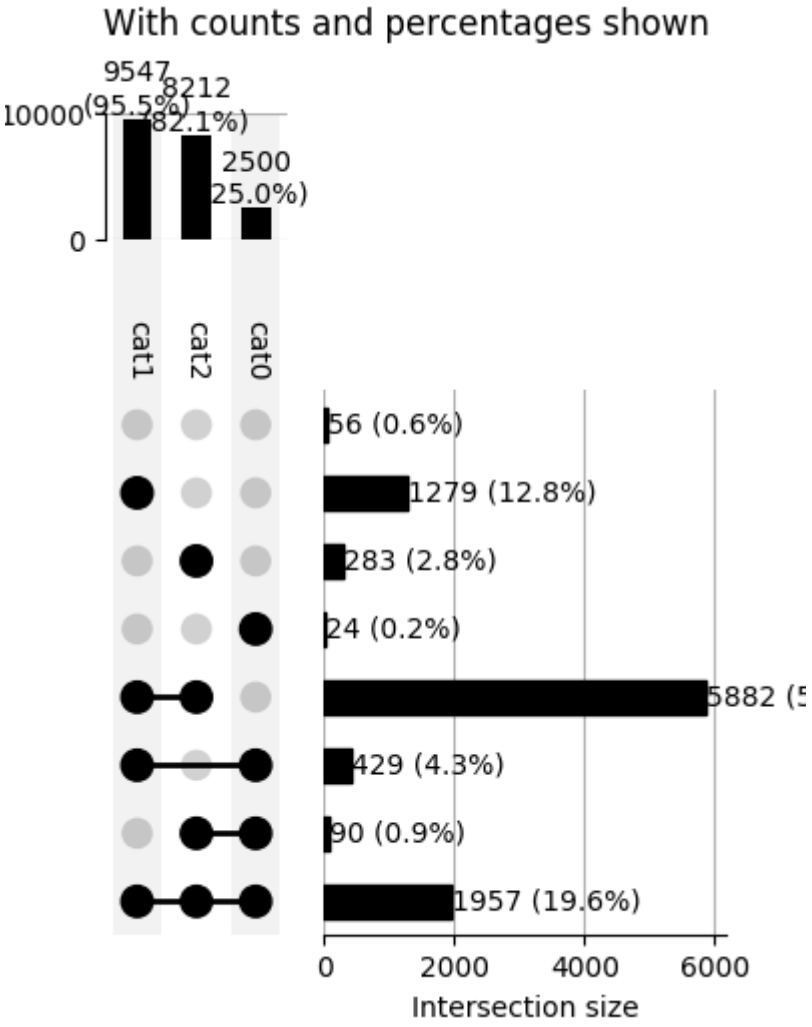
```
from matplotlib import pyplot as plt
from upsetplot import generate_counts, plot

example = generate_counts()
plot(example, orientation='vertical')
plt.suptitle('A vertical plot')
plt.show()
```




```
plot(example, orientation='vertical', show_counts='%d')
plt.suptitle('A vertical plot with counts shown')
plt.show()
```




```
plot(example, orientation='vertical', show_counts='%d', show_percentages=True)
plt.suptitle('With counts and percentages shown')
plt.show()
```



Total running time of the script: (0 minutes 0.819 seconds)

Note: Click *here* to download the full example code

Plotting with generated data

This example illustrates basic plotting functionality using generated data.

```
from matplotlib import pyplot as plt
from upsetplot import generate_counts, plot

example = generate_counts()
print(example)
```

Out:

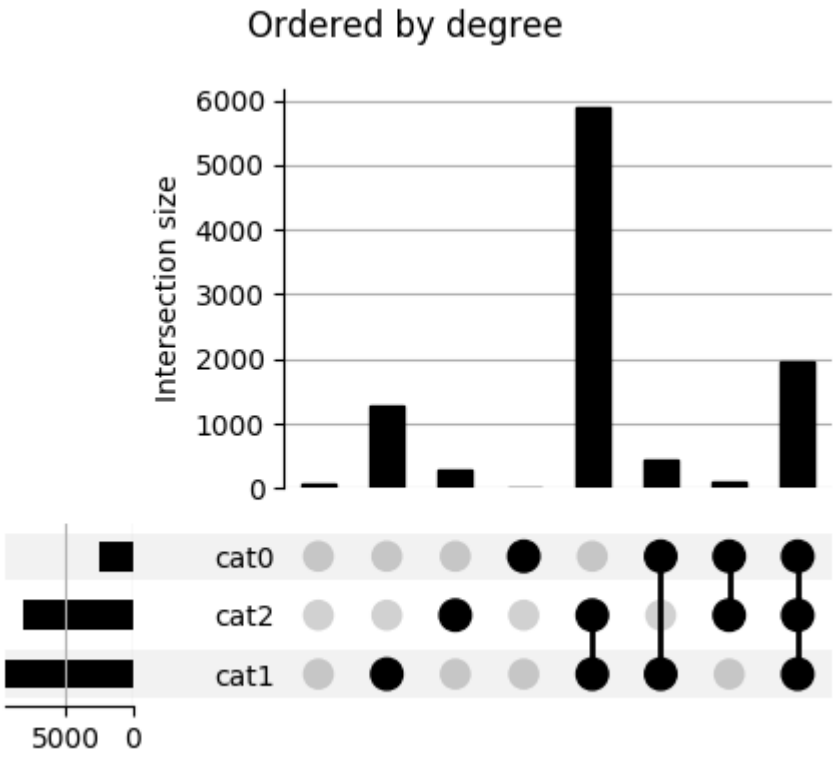
```
cat0   cat1   cat2
False  False  False      56
```

(continues on next page)

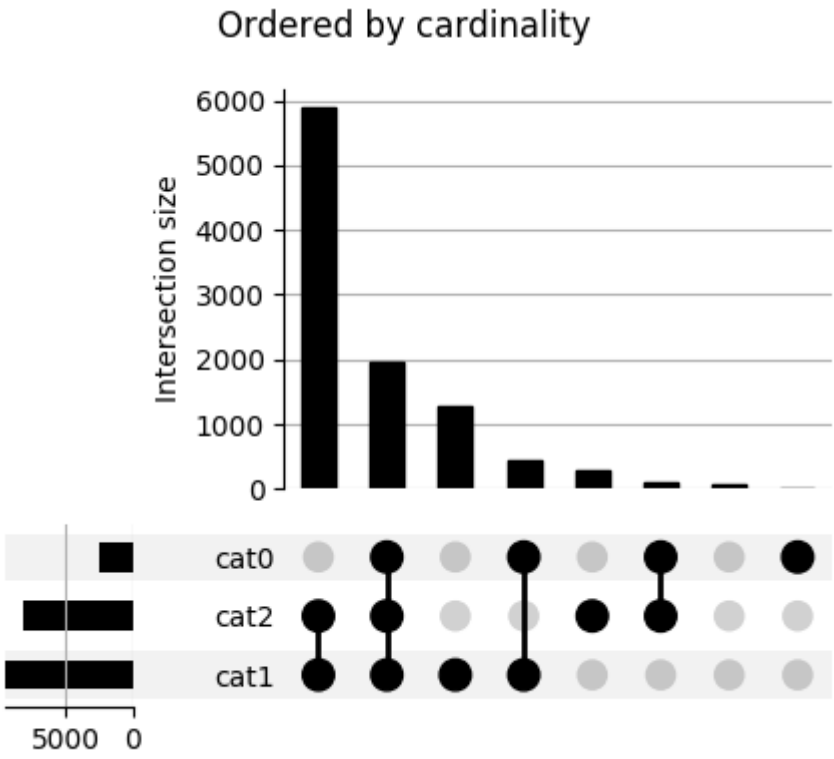
(continued from previous page)

```
                True      283
       True     False    1279
                True     5882
True   False    False      24
                True       90
       True     False     429
                True     1957
Name: value, dtype: int64
```

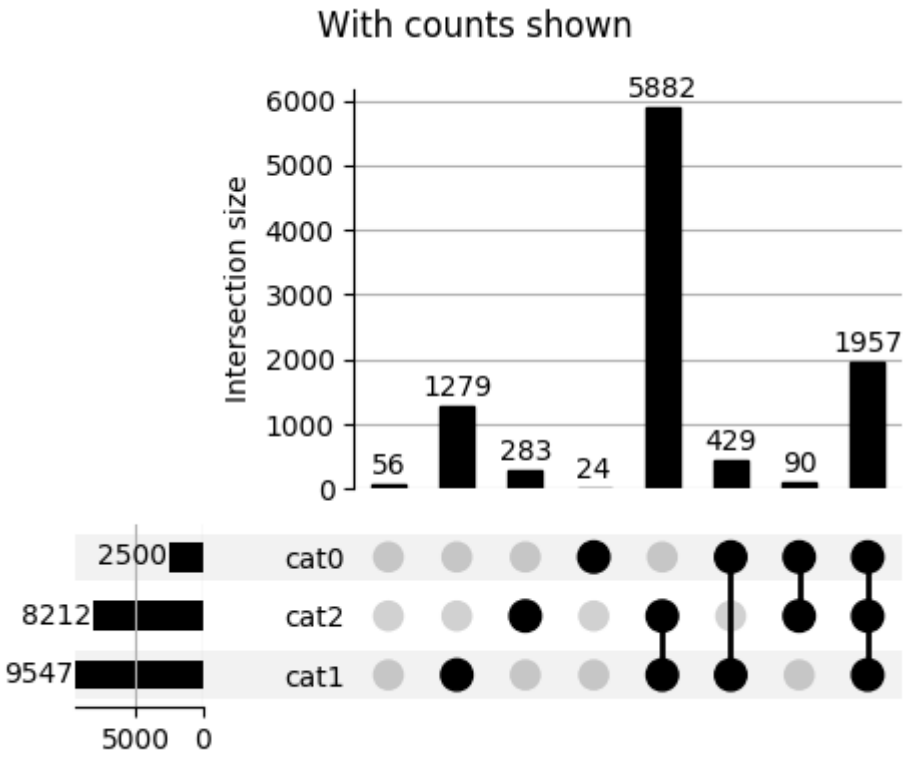
```
plot(example)
plt.suptitle('Ordered by degree')
plt.show()
```



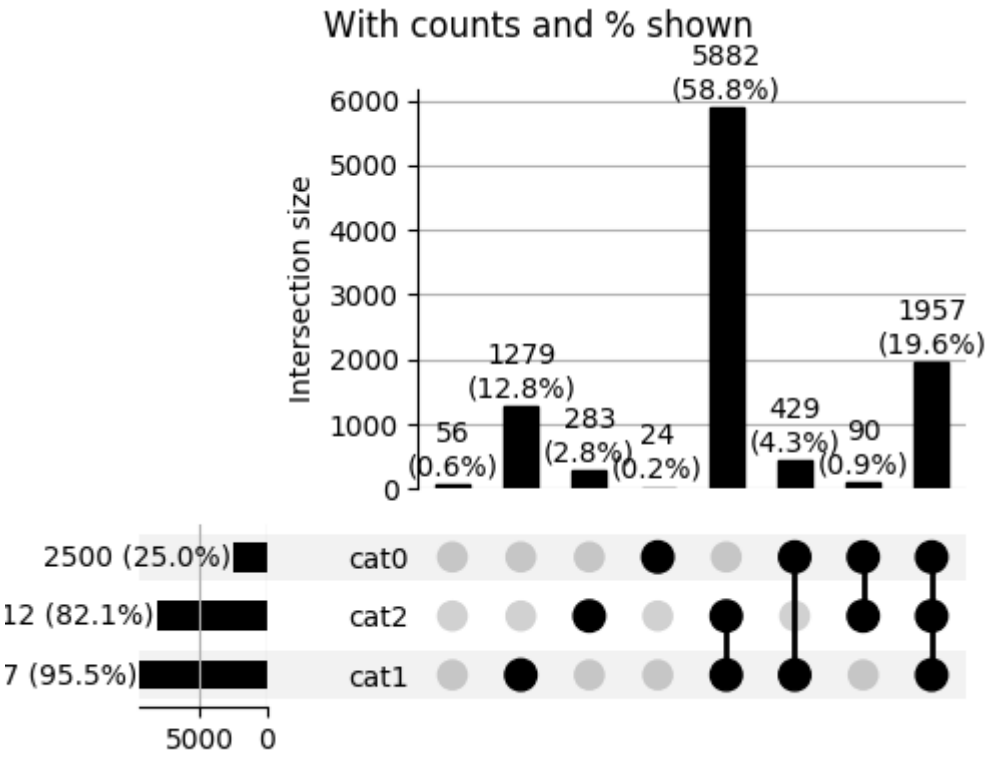

```
plot(example, sort_by='cardinality')
plt.suptitle('Ordered by cardinality')
plt.show()
```




```
plot(example, show_counts='%d')
plt.suptitle('With counts shown')
plt.show()
```




```
plot(example, show_counts='%d', show_percentages=True)
plt.suptitle('With counts and % shown')
plt.show()
```



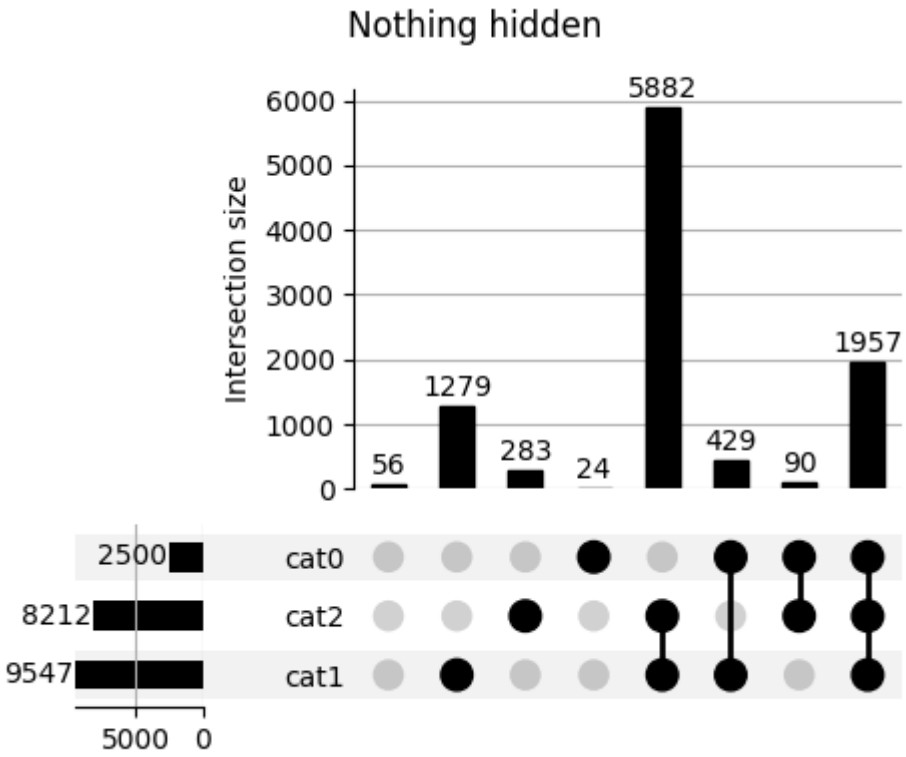
Total running time of the script: (0 minutes 1.080 seconds)

Note: Click *here* to download the full example code

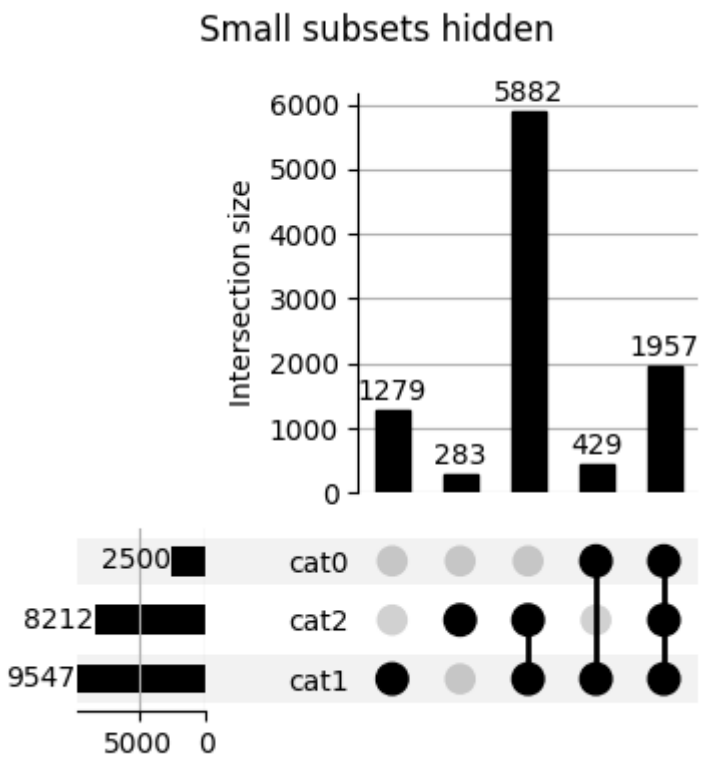
Hiding subsets based on size or degree

This illustrates the use of `min_subset_size`, `max_subset_size`, `min_degree` or `max_degree`.

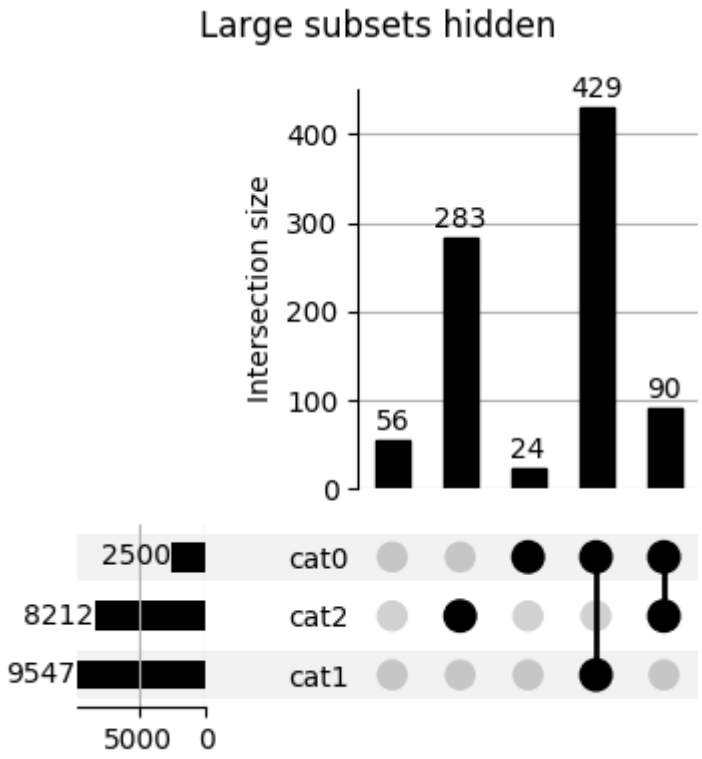
```
from matplotlib import pyplot as plt
from upsetplot import generate_counts, plot

example = generate_counts()

plot(example, show_counts=True)
plt.suptitle('Nothing hidden')
plt.show()
```



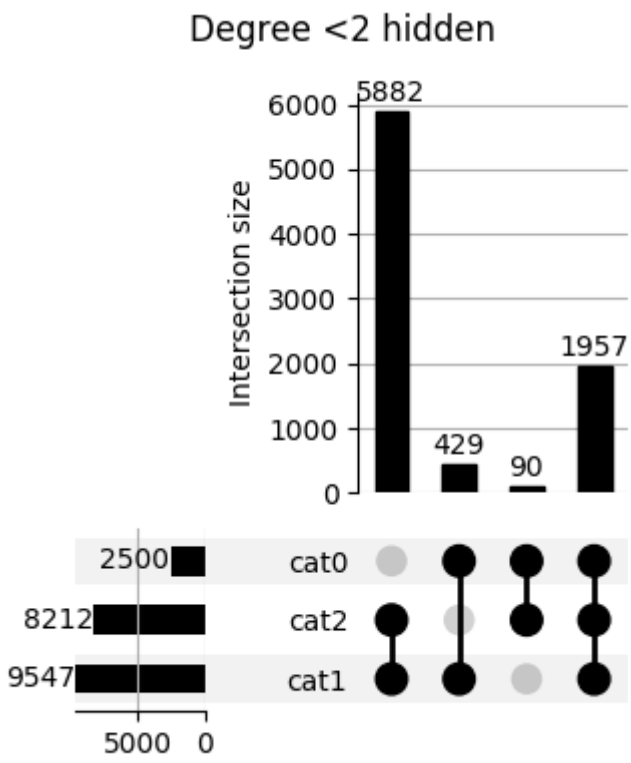

```
plot(example, show_counts=True, min_subset_size=100)
plt.suptitle('Small subsets hidden')
plt.show()
```



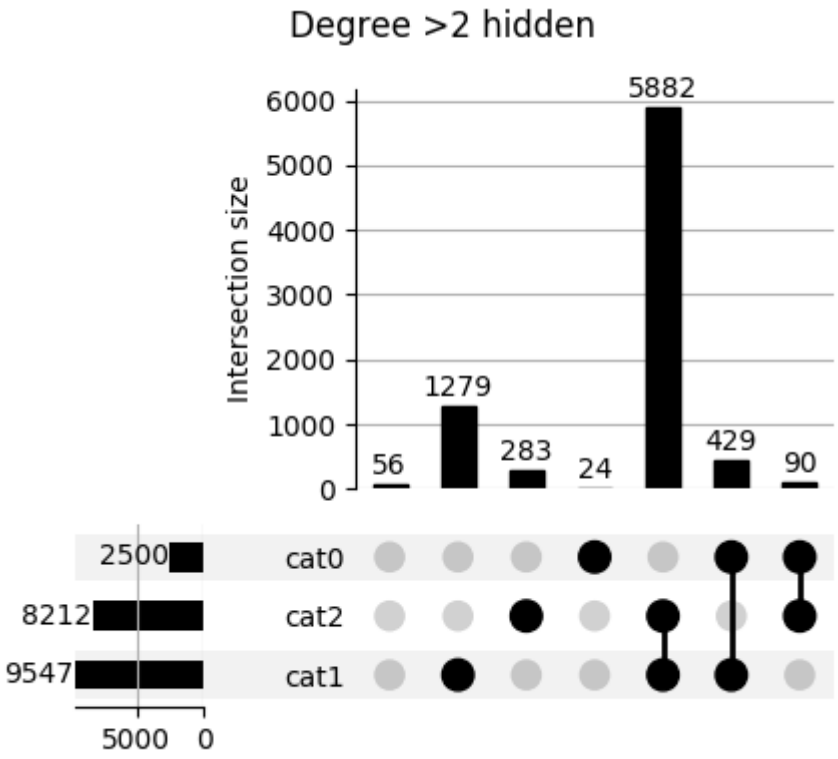

```
plot(example, show_counts=True, max_subset_size=500)
plt.suptitle('Large subsets hidden')
plt.show()
```




```
plot(example, show_counts=True, min_degree=2)
plt.suptitle('Degree <2 hidden')
plt.show()
```




```
plot(example, show_counts=True, max_degree=2)
plt.suptitle('Degree >2 hidden')
plt.show()
```



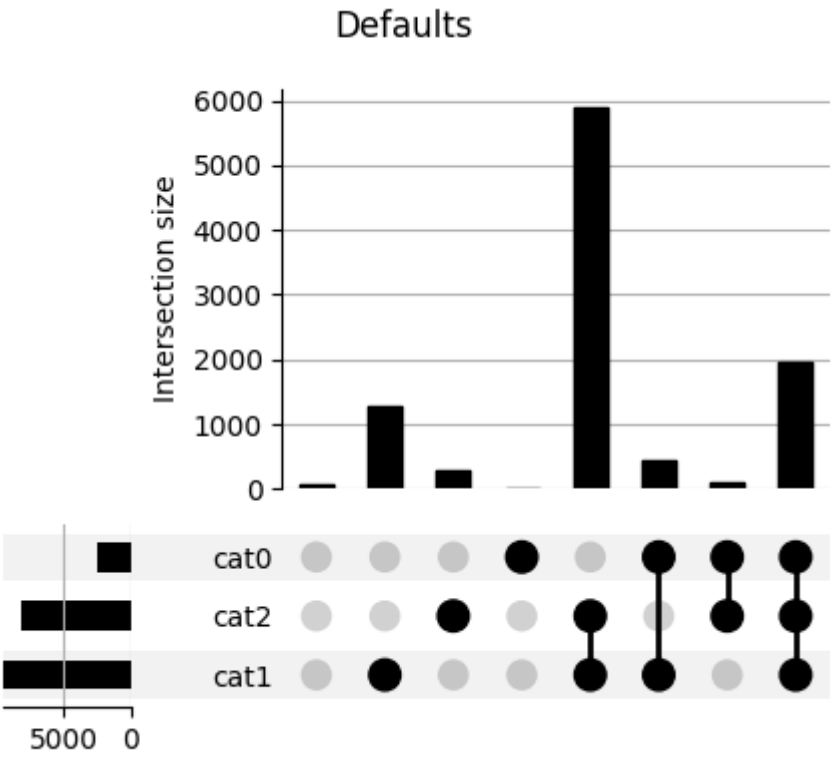
Total running time of the script: (0 minutes 1.327 seconds)

Note: Click *here* to download the full example code

Customising element size and figure size

This example illustrates controlling sizing within an UpSet plot.

```
from matplotlib import pyplot as plt
from upsetplot import generate_counts, plot

example = generate_counts()
print(example)

plot(example)
plt.suptitle('Defaults')
plt.show()
```

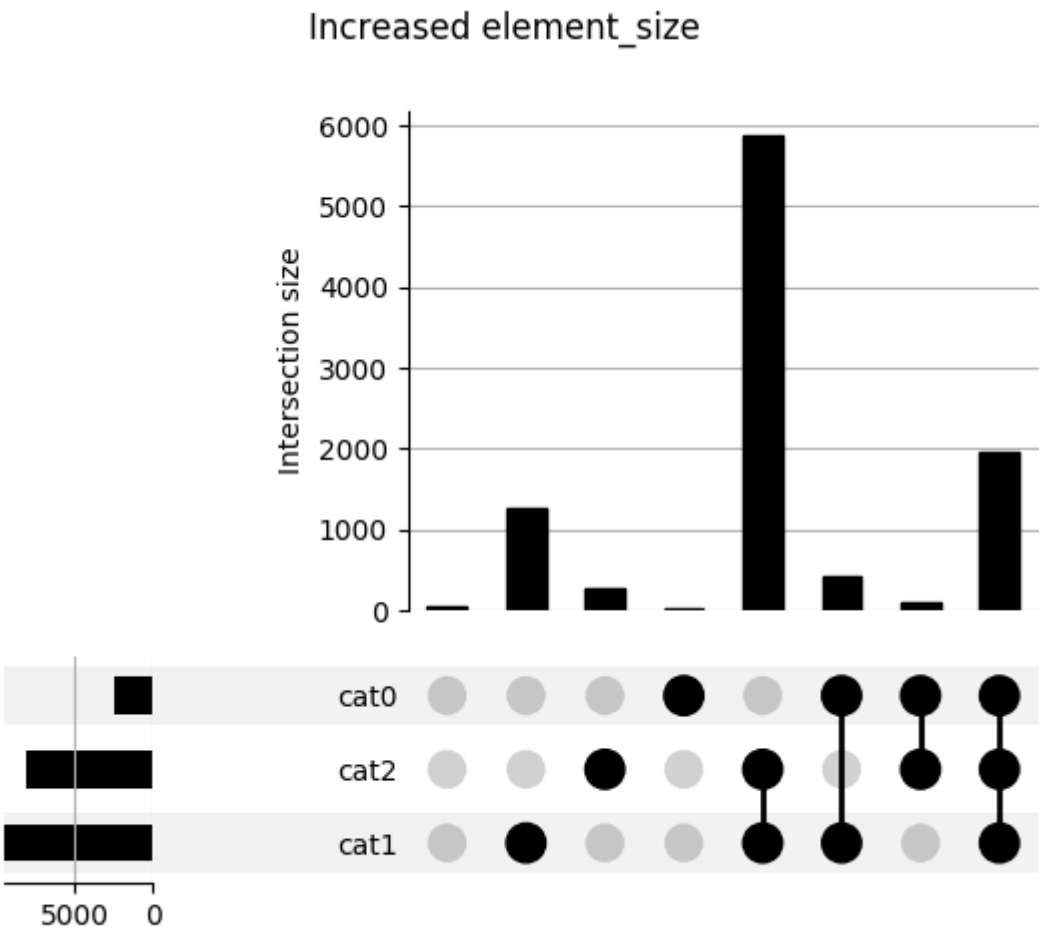


Out:

```
cat0   cat1   cat2
False  False  False      56
              True      283
       True   False    1279
              True     5882
True   False  False      24
              True       90
       True   False     429
              True     1957
Name: value, dtype: int64
```

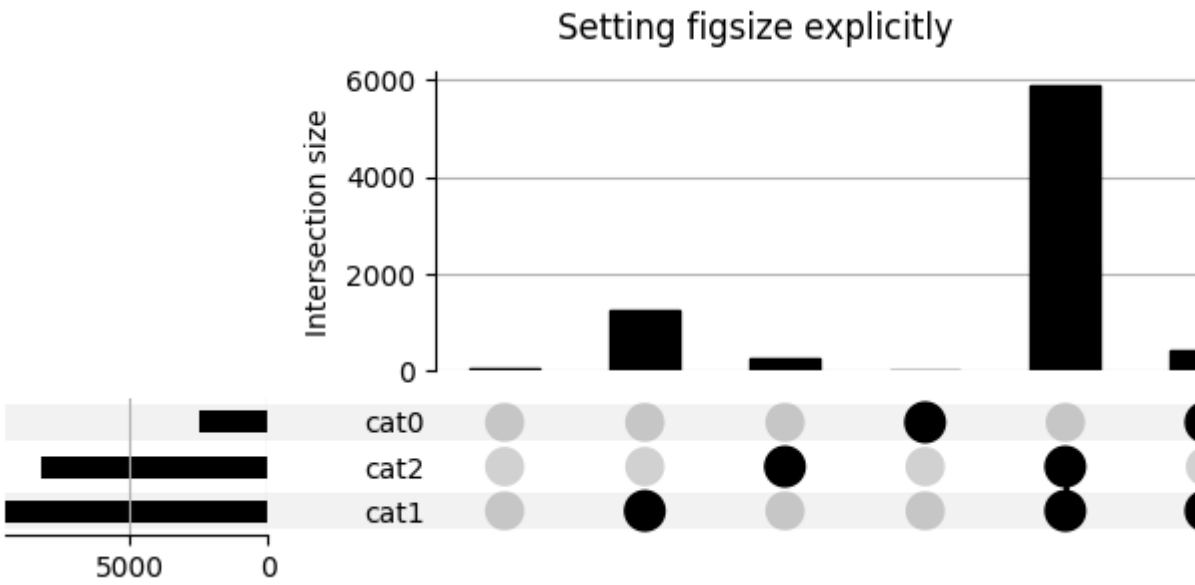
upsetplot uses a grid of square "elements" to display. Controlling the size of these elements affects all components of the plot.

```
plot(example, element_size=40)
plt.suptitle('Increased element_size')
plt.show()
```



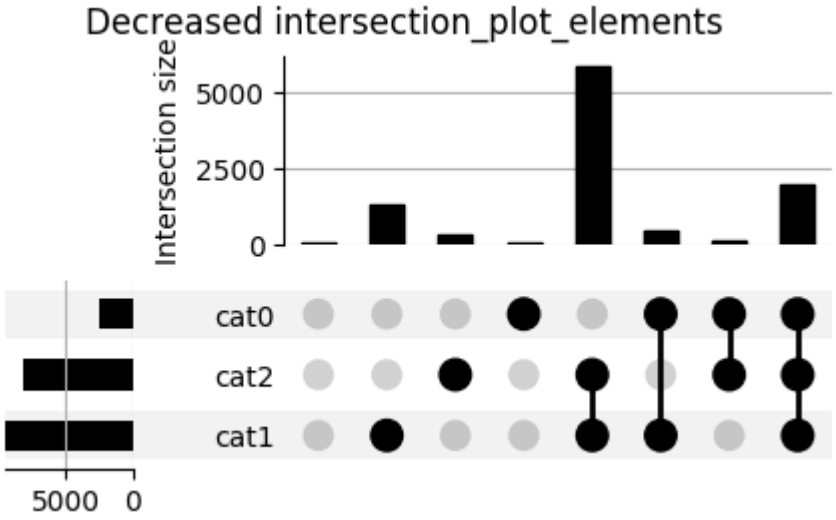
When setting `figsize` explicitly, you then need to pass the figure to `plot`, and use `element_size=None` for optimal sizing.

```
fig = plt.figure(figsize=(10, 3))
plot(example, fig=fig, element_size=None)
plt.suptitle('Setting figsize explicitly')
plt.show()
```

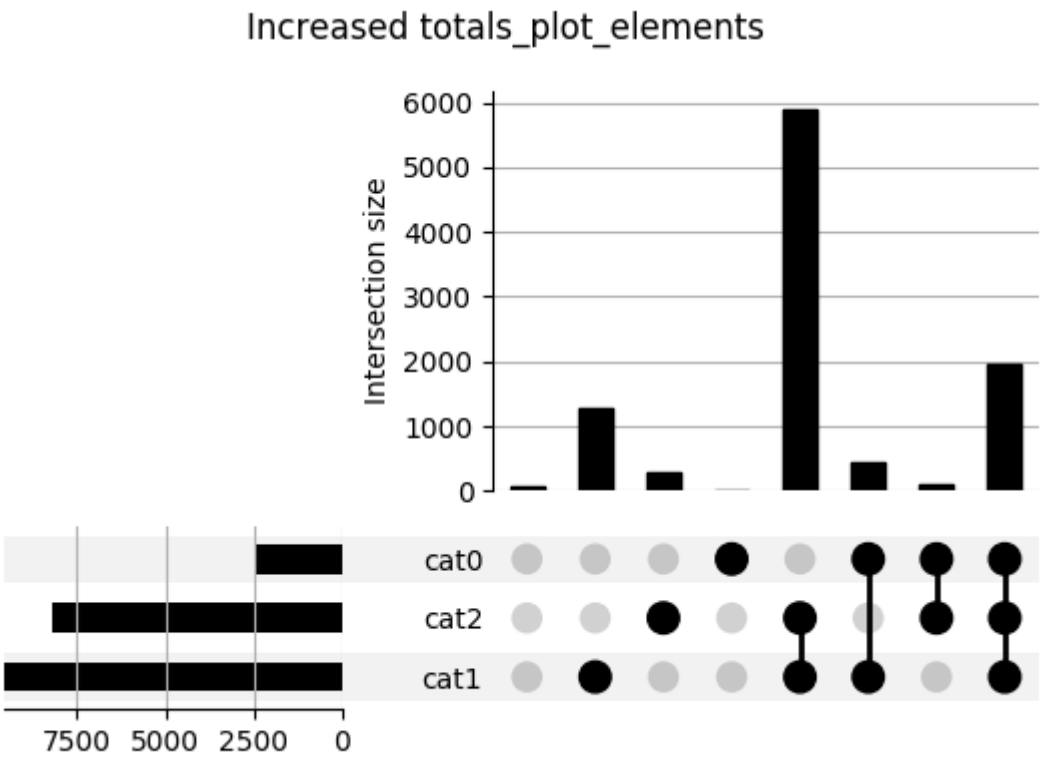


Components in the plot can be resized by indicating how many elements they should equate to.

```
plot(example, intersection_plot_elements=3)
plt.suptitle('Decreased intersection_plot_elements')
plt.show()
```




```
plot(example, totals_plot_elements=5)
plt.suptitle('Increased totals_plot_elements')
plt.show()
```

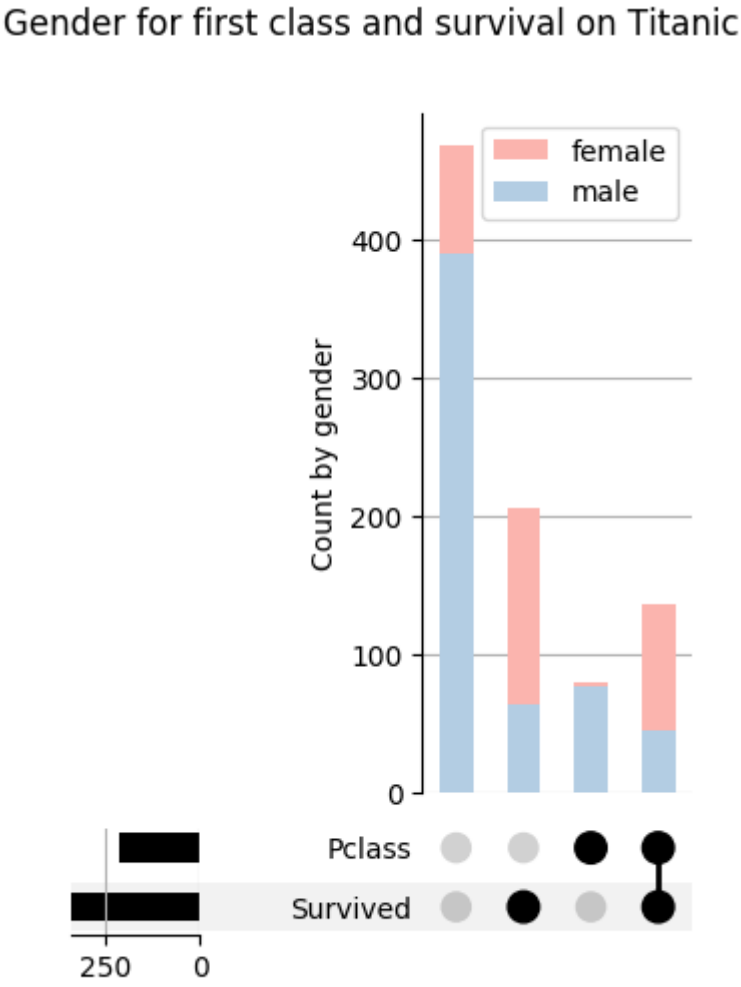


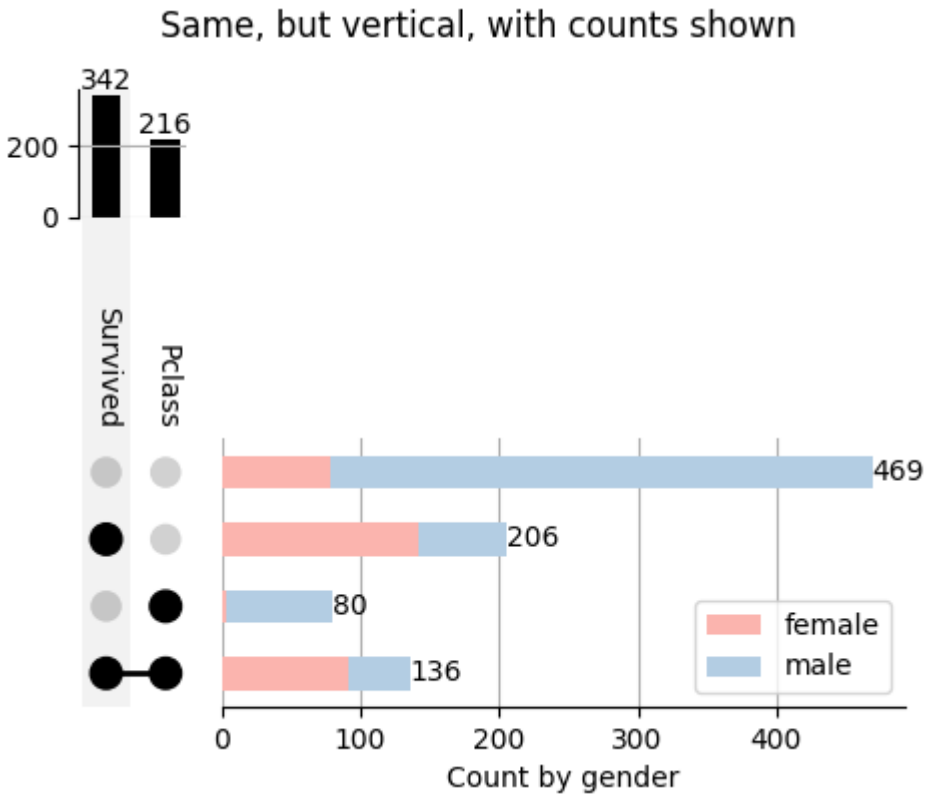
Total running time of the script: (0 minutes 1.323 seconds)

Note: Click *here* to download the full example code

Plotting discrete variables as stacked bar charts

Currently, a somewhat contrived example of `add_stacked_bars`.


Gender for first class and survival on Titanic
female
male
Count by gender
400
300
200
100
0
Pclass
Survived
250
0

- 


```
import pandas as pd
from upsetplot import UpSet
from matplotlib import pyplot as plt
from matplotlib import cm

TITANIC_URL = 'https://raw.githubusercontent.com/datasciencedojo/datasets/master/
↪titanic.csv'  # noqa
df = pd.read_csv(TITANIC_URL)
# Show UpSet on survival and first classs
df = df.set_index(df.Survived == 1).set_index(df.Pclass == 1, append=True)

upset = UpSet(df,
              intersection_plot_elements=0)  # disable the default bar chart
upset.add_stacked_bars(by="Sex", colors=cm.Pastel1,
                       title="Count by gender", elements=10)
upset.plot()
plt.suptitle("Gender for first class and survival on Titanic")
plt.show()


upset = UpSet(df, show_counts=True, orientation="vertical",
              intersection_plot_elements=0)
upset.add_stacked_bars(by="Sex", colors=cm.Pastel1,
                       title="Count by gender", elements=10)
upset.plot()
plt.suptitle("Same, but vertical, with counts shown")
plt.show()
```

Total running time of the script: (0 minutes 0.621 seconds)

Note: Click *here* to download the full example code

Changing Plot Colors

This example illustrates use of matplotlib and upsetplot color settings, aside from matplotlib style sheets, which can control colors as well as grid lines, fonts and tick display.

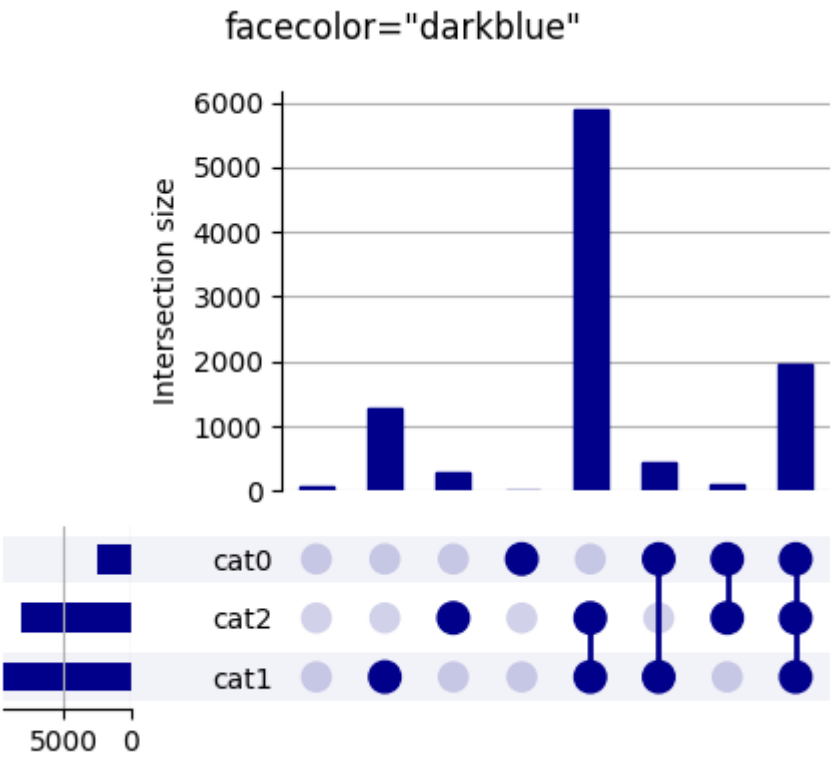
Upsetplot provides some color settings:

- `facecolor`: sets the color for intersection size bars, and for active matrix dots. Defaults to white on a dark background, otherwise black.
- `other_dots_color`: sets the color for other (inactive) dots. Specify as a color, or a float specifying opacity relative to facecolor.
- `shading_color`: sets the color odd rows. Specify as a color, or a float specifying opacity relative to facecolor.

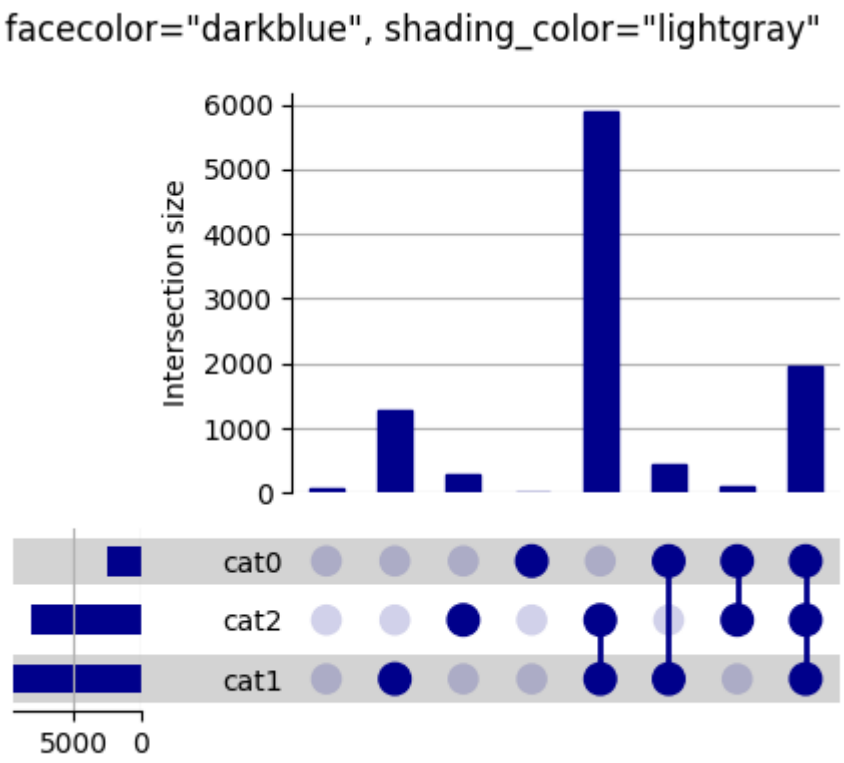
For an introduction to matplotlib theming see:

- Tutorial
- Reference

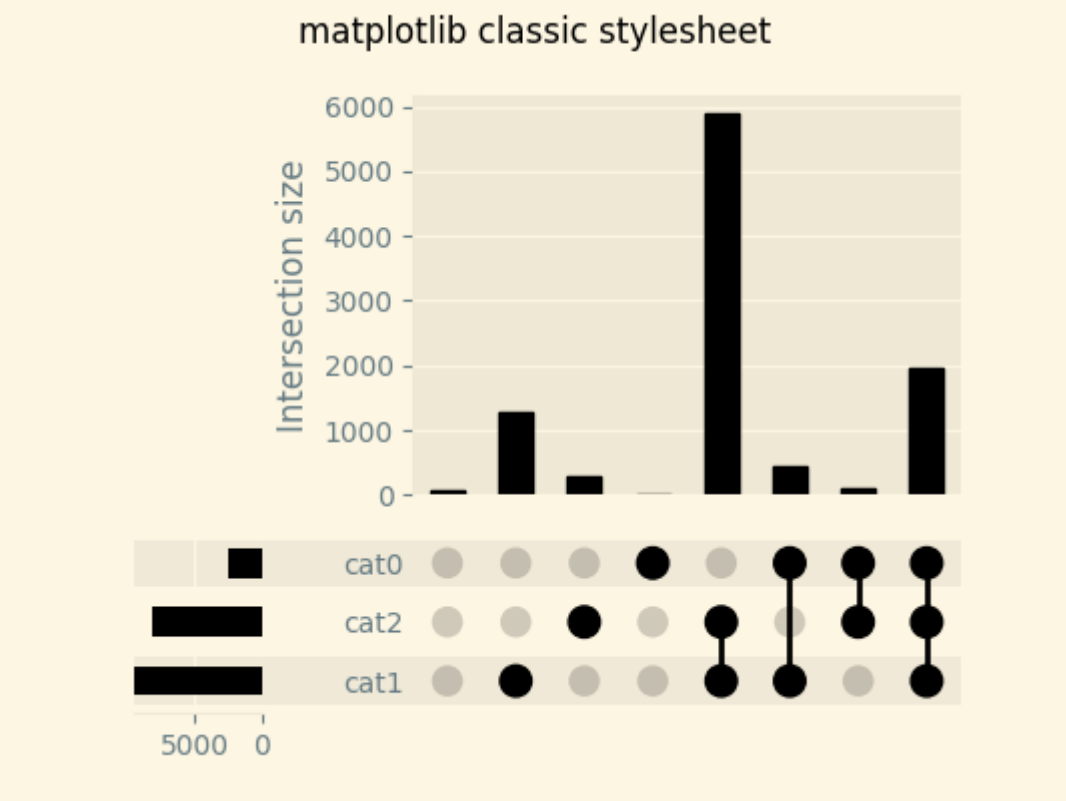
```
from matplotlib import pyplot as plt
from upsetplot import generate_counts, plot

example = generate_counts()

plot(example, facecolor="darkblue")
plt.suptitle('facecolor="darkblue"')
plt.show()
```



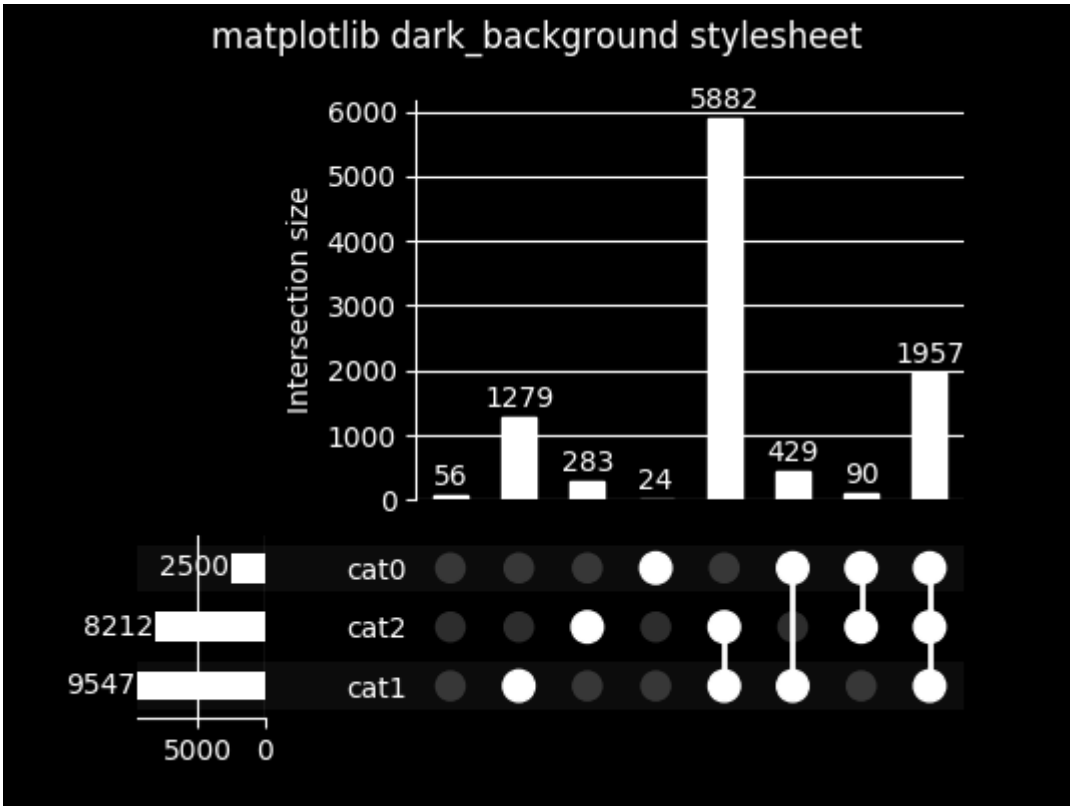

```
plot(example, facecolor="darkblue", shading_color="lightgray")
plt.suptitle('facecolor="darkblue", shading_color="lightgray"')
plt.show()
```



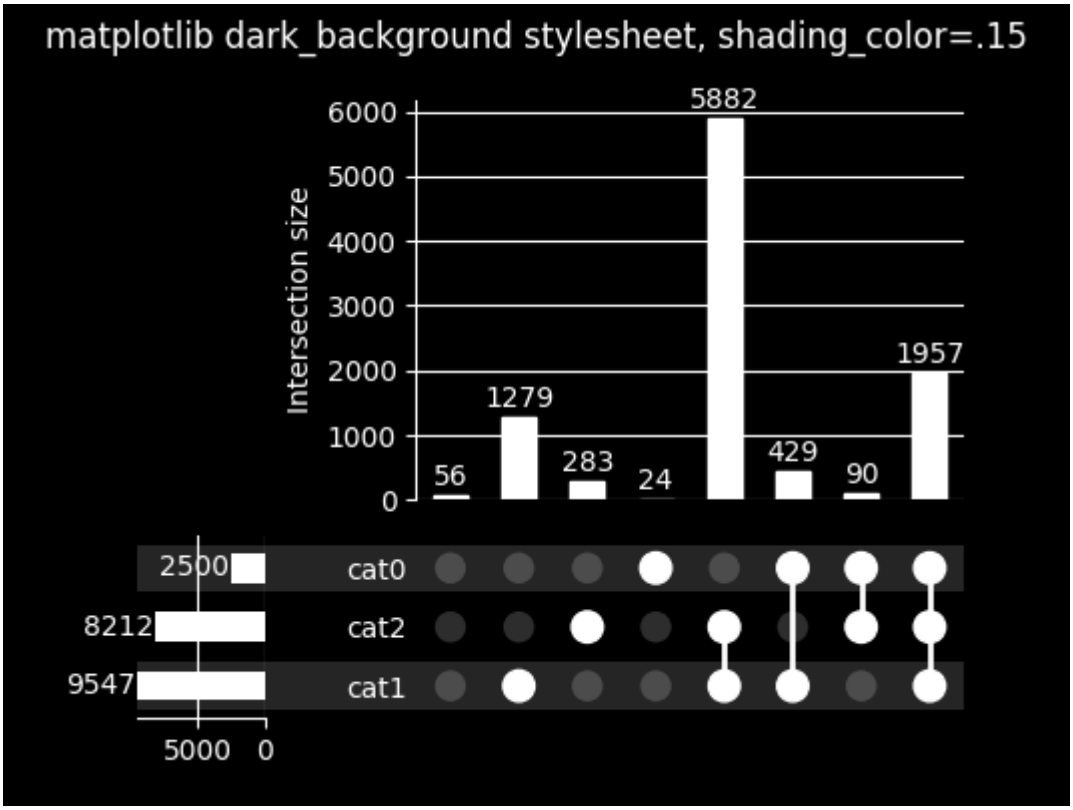

```
with plt.style.context('Solarize_Light2'):
    plot(example)
    plt.suptitle('matplotlib classic stylesheet')
    plt.show()
```



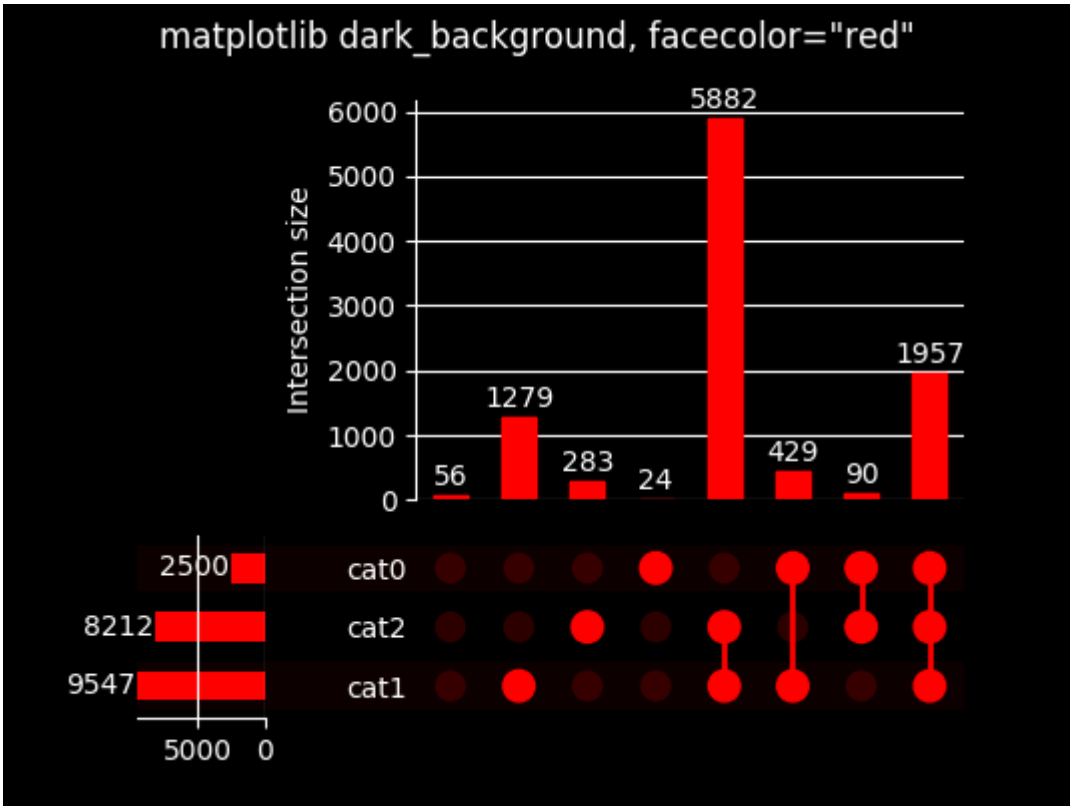

```
with plt.style.context('dark_background'):
    plot(example, show_counts=True)
    plt.suptitle('matplotlib dark_background stylesheet')
    plt.show()
```



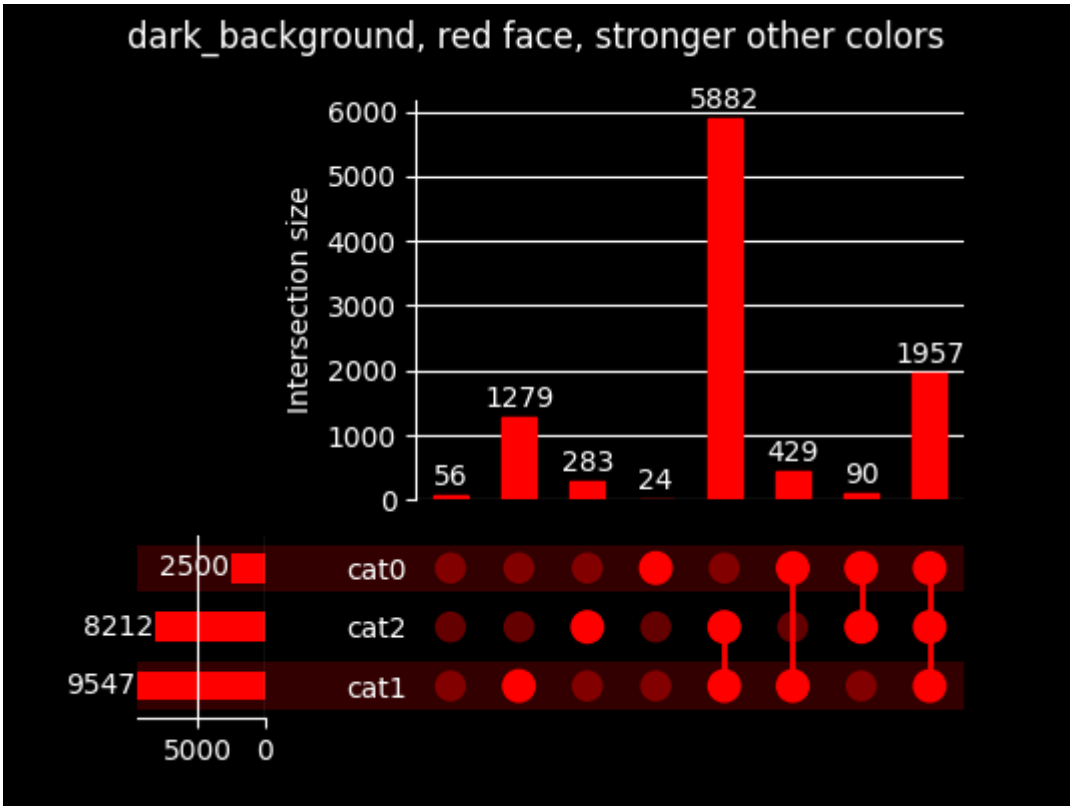

```
with plt.style.context('dark_background'):
    plot(example, show_counts=True, shading_color=.15)
    plt.suptitle('matplotlib dark_background stylesheet, shading_color=.15')
    plt.show()
```




```
with plt.style.context('dark_background'):
    plot(example, show_counts=True, facecolor="red")
    plt.suptitle('matplotlib dark_background, facecolor="red"')
    plt.show()
```




```
with plt.style.context('dark_background'):
    plot(example, show_counts=True, facecolor="red", other_dots_color=.4,
         shading_color=.2)
    plt.suptitle('dark_background, red face, stronger other colors')
    plt.show()
```



Total running time of the script: (0 minutes 1.860 seconds)

Note: Click *here* to download the full example code

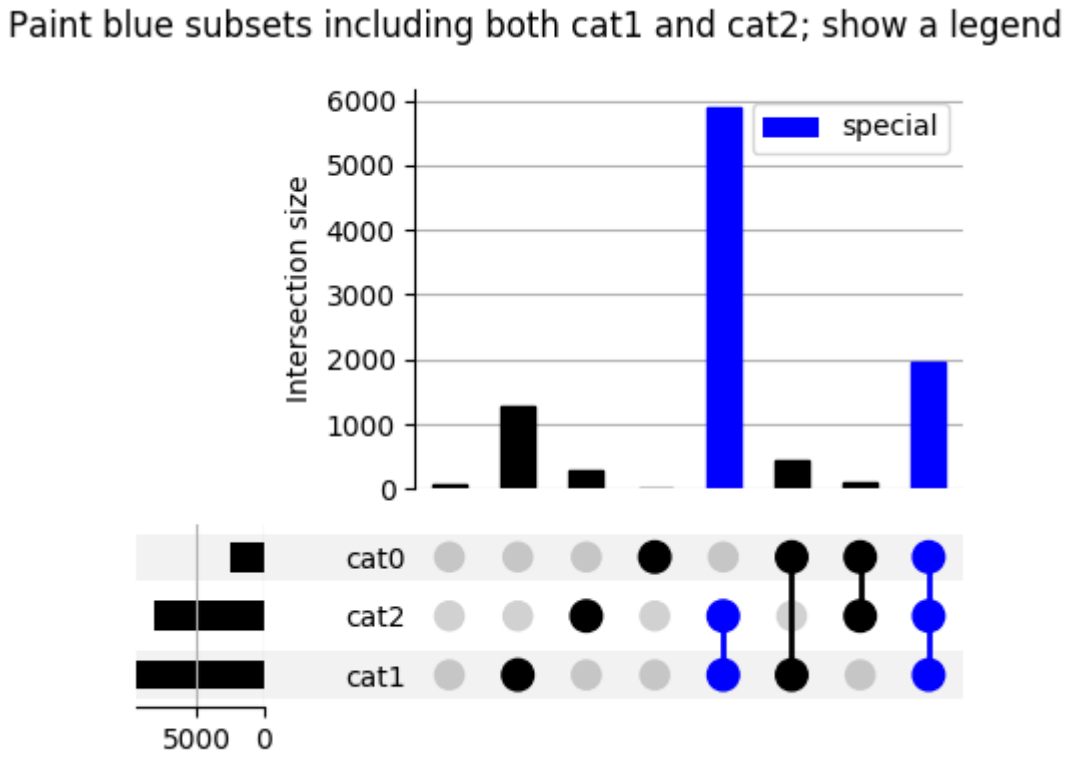
Highlighting selected subsets

Demonstrates use of the *style_subsets* method to mark some subsets as different.

```
from matplotlib import pyplot as plt
from upsetplot import generate_counts, UpSet

example = generate_counts()
```

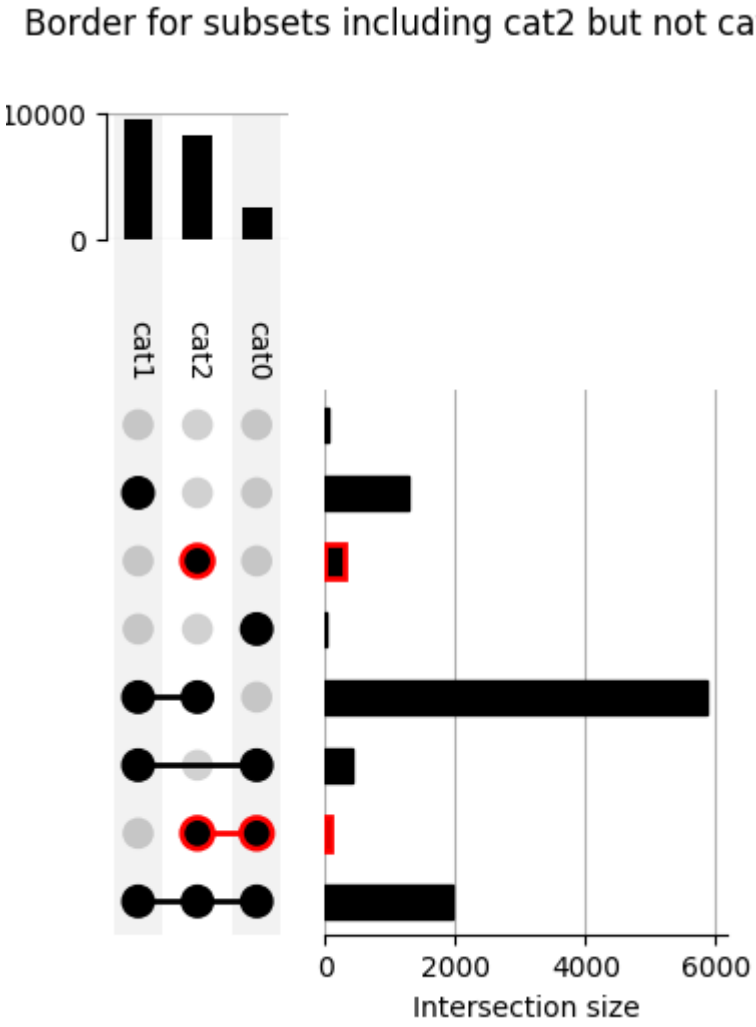
Subsets can be styled by the categories present in them, and a legend can be optionally generated.

```
upset = UpSet(example)
upset.style_subsets(present=["cat1", "cat2"],
                    facecolor="blue",
                    label="special")
upset.plot()
plt.suptitle("Paint blue subsets including both cat1 and cat2; show a legend")
plt.show()
```



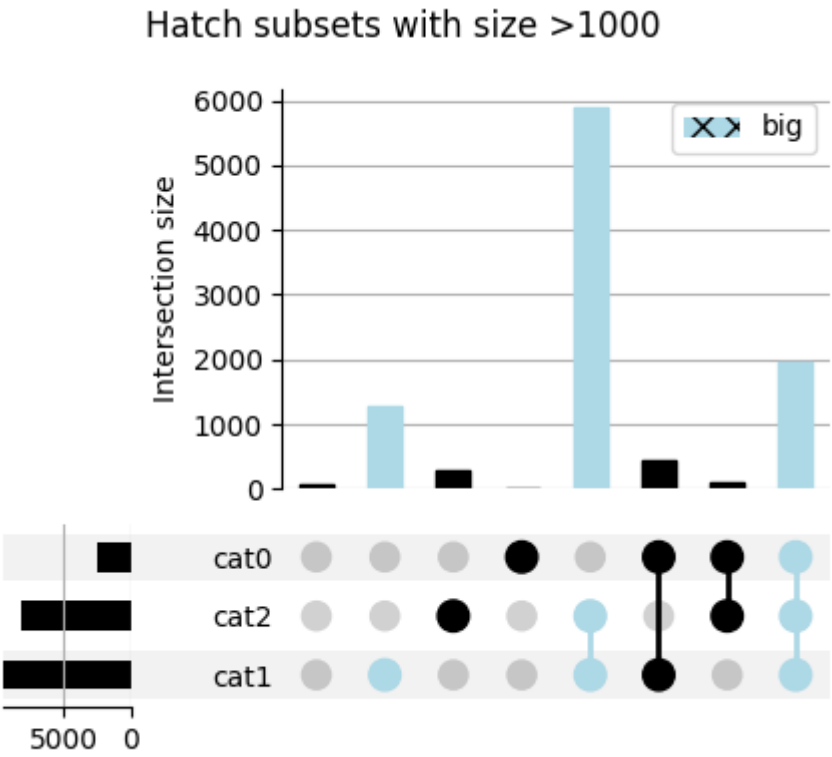
. . . or styling can be applied by the categories absent in a subset.

```
upset = UpSet(example, orientation="vertical")
upset.style_subsets(present="cat2", absent="cat1", edgecolor="red",
                    linewidth=2)
upset.plot()
plt.suptitle("Border for subsets including cat2 but not cat1")
plt.show()
```



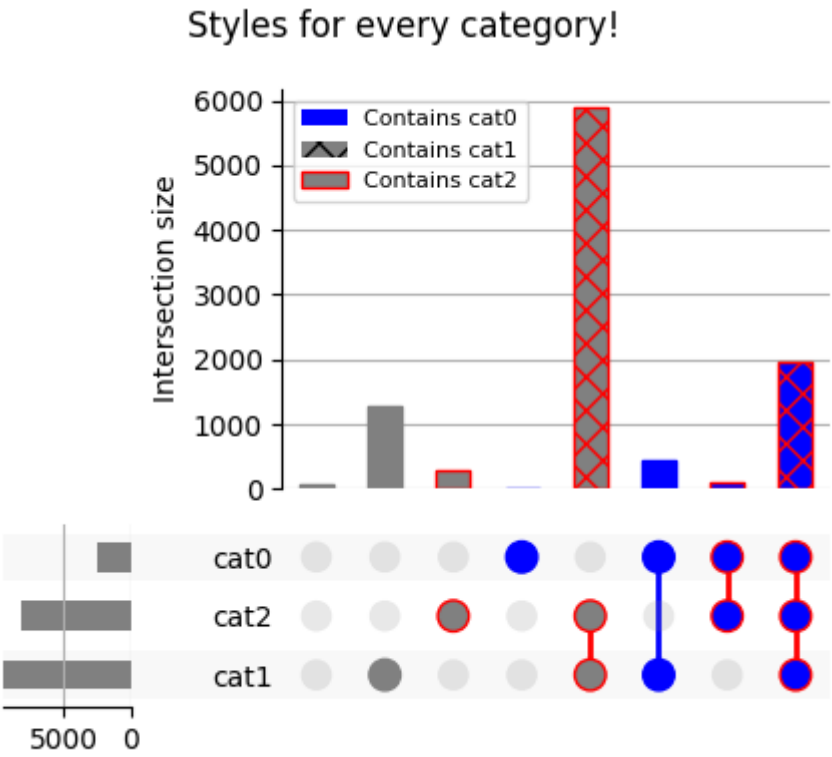
. . . or their size or degree.

```
upset = UpSet(example)
upset.style_subsets(min_subset_size=1000,
                    facecolor="lightblue", hatch="xx",
                    label="big")
upset.plot()
plt.suptitle("Hatch subsets with size >1000")
plt.show()
```



Multiple stylings can be applied with different criteria in the same plot.

```
upset = UpSet(example, facecolor="gray")
upset.style_subsets(present="cat0", label="Contains cat0", facecolor="blue")
upset.style_subsets(present="cat1", label="Contains cat1", hatch="xx")
upset.style_subsets(present="cat2", label="Contains cat2", edgecolor="red")

# reduce legend size:
params = {'legend.fontsize': 8}
with plt.rc_context(params):
    upset.plot()
plt.suptitle("Styles for every category!")
plt.show()
```



Total running time of the script: (0 minutes 1.117 seconds)

Note: Click *here* to download the full example code

Above-average features in Boston

Explore above-average neighborhood characteristics in the Boston dataset.

Here we take some features correlated with house price, and look at the distribution of median house price when each of these features is above average.

The most correlated features are:

ZN proportion of residential land zoned for lots over 25,000 sq.ft.

CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

RM average number of rooms per dwelling

DIS weighted distances to five Boston employment centres

B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

This kind of dataset analysis may not be a practical use of UpSet, but helps to illustrate the `UpSet.add_catplot()` feature.

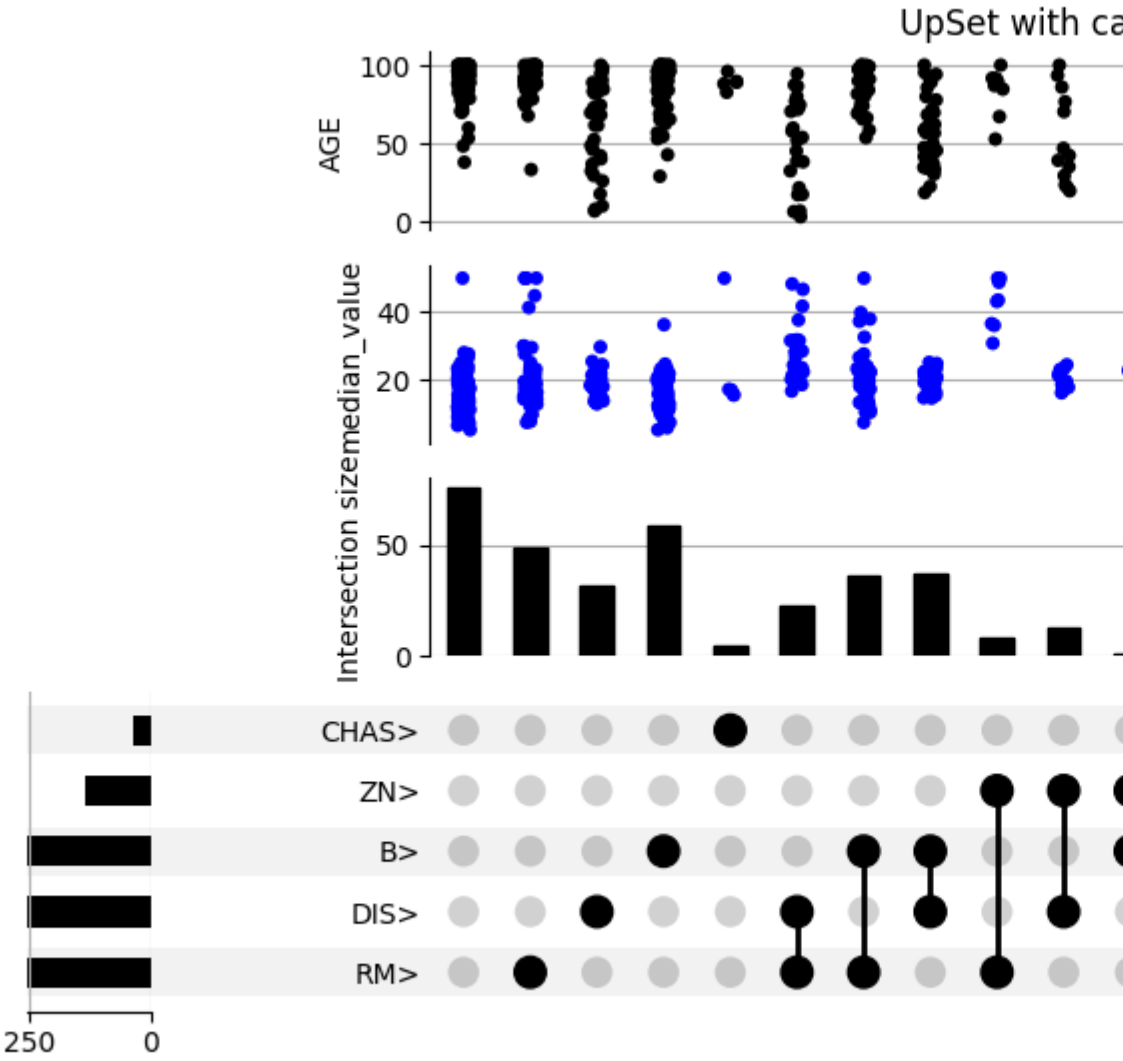
```
import pandas as pd
from sklearn.datasets import load_boston
from matplotlib import pyplot as plt
from upsetplot import UpSet
```

(continues on next page)

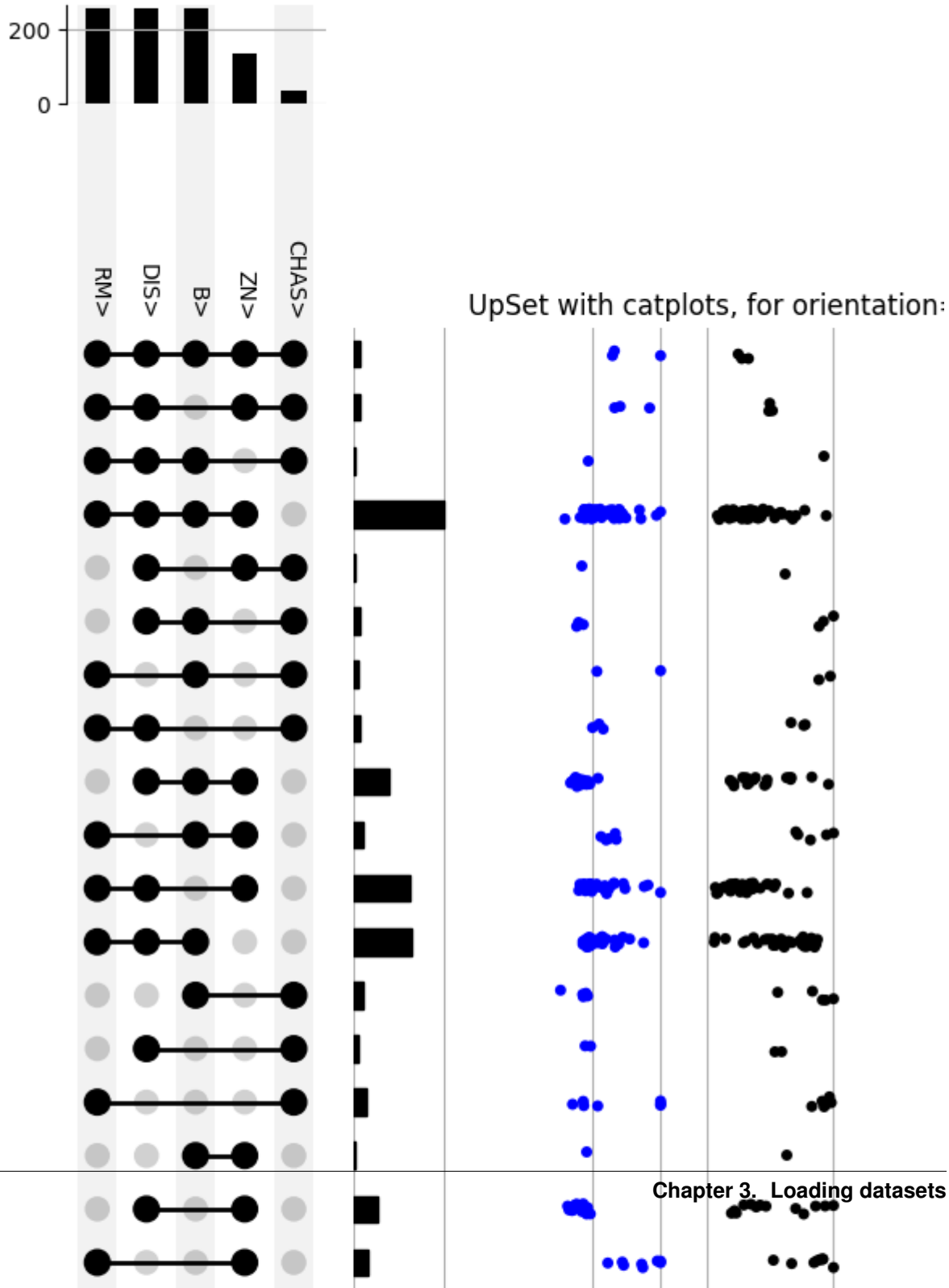
(continued from previous page)

```
# Load the dataset into a DataFrame
boston = load_boston()
boston_df = pd.DataFrame(boston.data, columns=boston.feature_names)

# Get five features most correlated with median house value
correls = boston_df.corrwith(pd.Series(boston.target),
                             method='spearman').sort_values()
top_features = correls.index[-5:]

# Get a binary indicator of whether each top feature is above average
boston_above_avg = boston_df > boston_df.median(axis=0)
boston_above_avg = boston_above_avg[top_features]
boston_above_avg = boston_above_avg.rename(columns=lambda x: x + '>')

# Make this indicator mask an index of boston_df
boston_df = pd.concat([boston_df, boston_above_avg],
                      axis=1)
boston_df = boston_df.set_index(list(boston_above_avg.columns))

# Also give us access to the target (median house value)
boston_df = boston_df.assign(median_value=boston.target)
```

```
# UpSet plot it!
upset = UpSet(boston_df, subset_size='count', intersection_plot_elements=3)
upset.add_catplot(value='median_value', kind='strip', color='blue')
upset.add_catplot(value='AGE', kind='strip', color='black')
upset.plot()
plt.title("UpSet with catplots, for orientation='horizontal'")
plt.show()
```




```
# And again in vertical orientation

upset = UpSet(boston_df, subset_size='count', intersection_plot_elements=3,
              orientation='vertical')
upset.add_catplot(value='median_value', kind='strip', color='blue')
upset.add_catplot(value='AGE', kind='strip', color='black')
upset.plot()
plt.title("UpSet with catplots, for orientation='vertical'")
plt.show()
```


200
0
RM>
DIS>
B>
ZN>
CHAS>
UpSet with catplots, for orientation=

Total running time of the script: (0 minutes 3.087 seconds)

3.3.2 Data Format Guide

UpSetPlot fundamentally is about visualizing datapoints (or data aggregates) that are each assigned to one or more categories. Curiously, there are many ways to represent categories as data structures. Object 1 belongs to categories `A` and `B` and object 2 belongs to category `B` only, this information can be represented by:

- listing the memberships for each object, i.e. `[["A", "B"], # object 1 ["B"]] # object 2`
- listing the contents of each category, i.e. `{"A": [1], "B": [1, 2]}`
- using a boolean-valued indicator matrix (perhaps columns in a larger DataFrame), i.e. `# A B [[ True, True ], # object 1 [ False, True ]] # object 2`

Moreover, UpSetPlot aims to handle both of the following cases:

- where only aggregates (e.g. counts) of the values in each category subset are given; and
- there are data points with several attributes in each category subset, where these attributes can be visualized as well as aggregates.

This guide reviews the internal data format and alternative representations, but we recommend using the helper functions `` `from_memberships <api.html#upsetplot.from_memberships>`__ ``, `` `from_contents <api.html#upsetplot.from_contents>`__ `` or `` `from_indicators <api.html#upsetplot.from_indicators>`__ `` depending on how it's most convenient to express your data.

Internal data format

UpSetPlot internally works with data based on Pandas data structres: a Series when all you care about is counts, or a DataFrame when you're interested in visualising additional properties of the data, such as with the `UpSet.add_catplot` method.

UpSetPlot expects the Series or DataFrame to have a MultiIndex as input, with this index being an indicator matrix. Specifically, each category is a level in the `pandas.MultiIndex` with boolean values.

Note: This internal data format may change in a future version since it is not efficient. Using the `from_*` methods will provide more stable compatibility with future releases.

Use `Series` as input

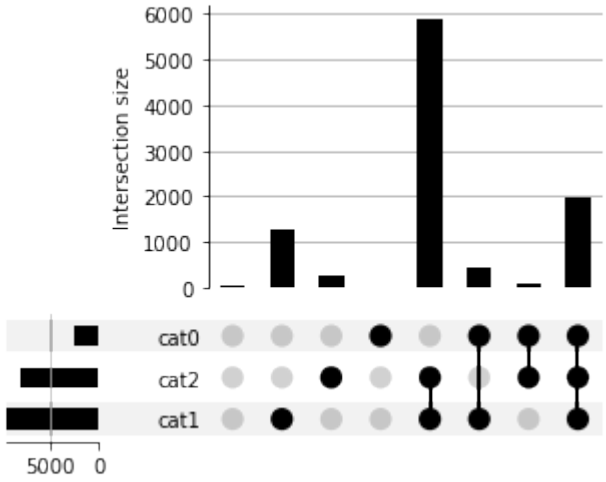
Below is a minimal example using `Series` as input:

[1]:
```
from upsetplot import generate_counts
example_counts = generate_counts()
example_counts
```

[1]:
```
cat0   cat1   cat2
False  False  False      56
              True      283
       True   False    1279
              True     5882
True   False  False      24
              True       90
       True   False     429
              True     1957
Name: value, dtype: int64
```

This is a `pandas.Series` with 3-level Multi-index. Each level is a `Set`: `cat0`, `cat1`, and `cat2`. Each row is a unique subset with boolean values in indices indicating memberships of each row. The value in each row indicates the number of observations in each subset. `upsetplot` will simply plot these numbers when supplied with a `Series`:

[2]:
```
from upsetplot import UpSet
plt = UpSet(example_counts).plot()
```



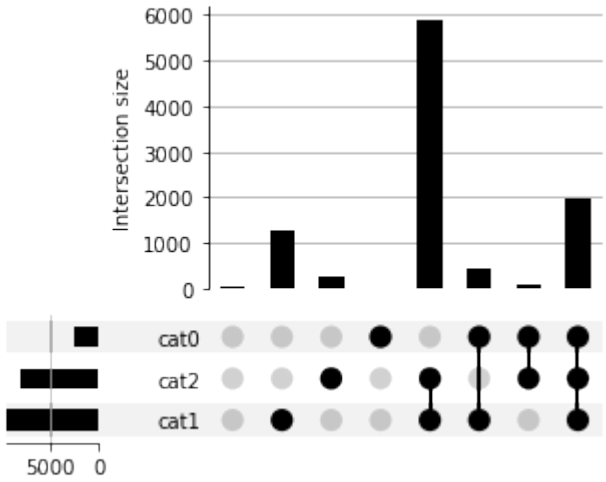
Alternatively, we can supply a `Series` with each observation in a row:

[3]:
```
from upsetplot import generate_samples
example_values = generate_samples().value
example_values
```

[3]:
```
cat0   cat1   cat2
False  True   True    1.652317
              True    1.510447
       False  True    1.584646
              True    1.279395
       True   True    2.338243
                        ...
              True    1.701618
              True    1.577837
True   True   True    1.757554
False  True   True    1.407799
True   True   True    1.709067
Name: value, Length: 10000, dtype: float64
```

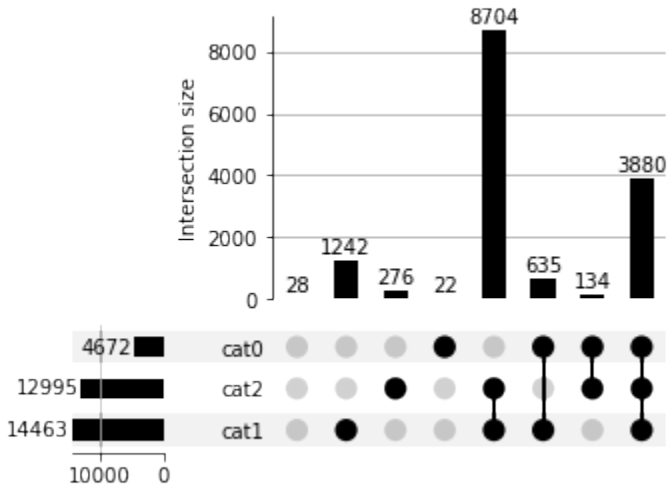
In this case, we can use `subset_size='count'` to have `upsetplot` count the number of observations in each unique subset and plot them:

[4]:
```
from upsetplot import UpSet
plt = UpSet(example_values, subset_size='count').plot()
```



Or, we can weight each subset's size by the series value:

[5]:
```
from upsetplot import UpSet
plt = UpSet(example_values, subset_size='sum', show_counts=True).plot()
```



Use `DataFrame` as input:

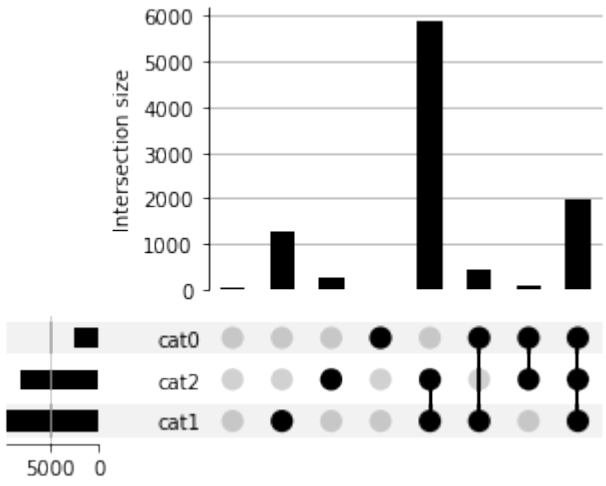
A `DataFrame` can also be used as input to carry additional information.

[6]:
```
from upsetplot import generate_samples
example_samples_df = generate_samples()
example_samples_df.head()
```

[6]:
```
                  index     value
cat0  cat1  cat2
False True  True      0  1.652317
            True      1  1.510447
      False True      2  1.584646
            True      3  1.279395
      True  True      4  2.338243
```

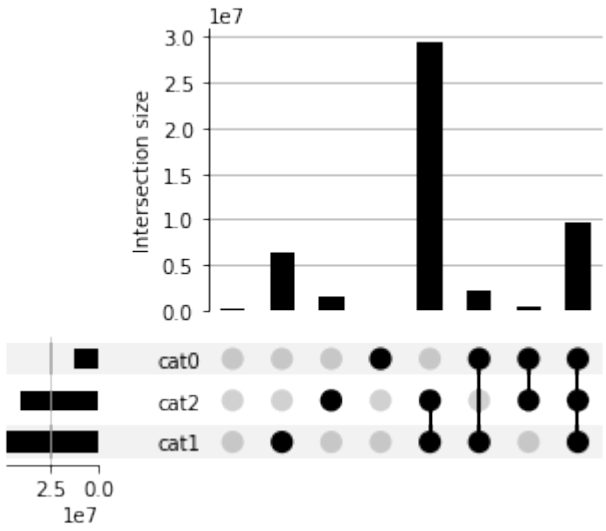
In this data frame, each observation has two variables: `index` and `value`. If we simply want to count the number of observations in each unique subset, we can use `subset_size='count'`:

[7]:
```
from upsetplot import UpSet
plt = UpSet(example_samples_df, subset_size='count').plot()
```



If for some reason, we want to plot the sum of a variable in each subset (eg. `index`), we can use `sum_over='index'`. This will make `upsetplot` to take sum of a given variable in each unique subset and plot that number:

[8]:
```
from upsetplot import UpSet
plt = UpSet(example_samples_df, sum_over='index', subset_size='sum').plot()
```



Convert Data to UpSet-compatible format

We can convert data from common formats to be compatible with `upsetplot`.

Suppose we have three categories (the data is not scientifically true!):

[9]:
```
mammals = ['Cat', 'Dog', 'Horse', 'Sheep', 'Pig', 'Cattle', 'Rhinoceros', 'Moose']
herbivores = ['Horse', 'Sheep', 'Cattle', 'Moose', 'Rhinoceros']
domesticated = ['Dog', 'Chicken', 'Horse', 'Sheep', 'Pig', 'Cattle', 'Duck']
(mammals, herbivores, domesticated)
```

[9]:
```
(['Cat', 'Dog', 'Horse', 'Sheep', 'Pig', 'Cattle', 'Rhinoceros', 'Moose'],
 ['Horse', 'Sheep', 'Cattle', 'Moose', 'Rhinoceros'],
 ['Dog', 'Chicken', 'Horse', 'Sheep', 'Pig', 'Cattle', 'Duck'])
```

Since this format lists the entries in each category, we can use `from_contents` to construct a data frame ready for plotting.

`from_contents` takes a dictionary as input. The input dictionary should have categories names as key and a list or set of category members as values:

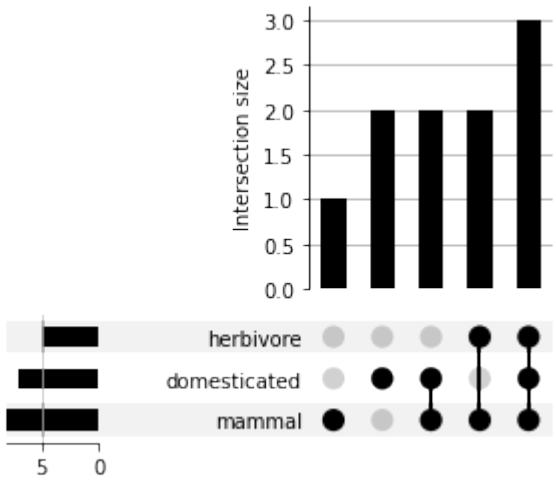
[10]:
```
from upsetplot import from_contents
animals = from_contents({'mammal': mammals, 'herbivore': herbivores, 'domesticated':␣
↪domesticated})
animals
```

[10]:

mammal	herbivore	domesticated	id
True	False	False	Cat
		True	Dog
	True	True	Horse
		True	Sheep
	False	True	Pig
	True	True	Cattle
		False	Rhinoceros
		False	Moose
False	False	True	Chicken
		True	Duck

Now we can plot:

[11]:
```
from upsetplot import UpSet
plt = UpSet(animals, subset_size='count').plot()
```



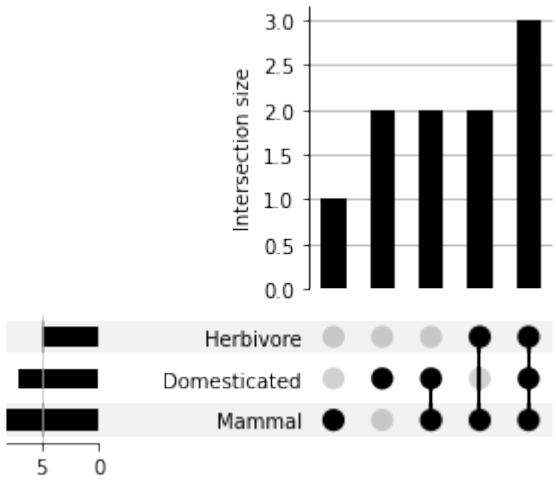
Alternatively, our input data may have been structured by species, allowing us to use `from_memberships`:

[12]:
```
from upsetplot import from_memberships

animal_memberships = {
    "Cat": "Mammal",
    "Dog": "Mammal,Domesticated",
    "Horse": "Mammal,Herbivore,Domesticated",
    "Sheep": "Mammal,Herbivore,Domesticated",
    "Pig": "Mammal,Domesticated",
    "Cattle": "Mammal,Herbivore,Domesticated",
    "Rhinoceros": "Mammal,Herbivore",
    "Moose": "Mammal,Herbivore",
    "Chicken": "Domesticated",
    "Duck": "Domesticated",
}

# Turn this into a list of lists:
animal_membership_lists = [categories.split(",") for categories in animal_memberships.
↪values()]

animals = from_memberships(animal_membership_lists)
animals
```

[12]:
```
Domesticated  Herbivore  Mammal
False         False      True      1
True          False      True      1
              True       True      1
                         True      1
              False      True      1
              True       True      1
False         True       True      1
                         True      1
True          False      False     1
                         False     1
Name: ones, dtype: int64
```

This should produce the same plot:

[13]:
```
from upsetplot import UpSet
plt = UpSet(animals, subset_size='count').plot()
```



When category membership is indicated in DataFrame columns

Let's take a look at a `movies` dataset like that used in the original publication by Alexander Lex et al..

[14]:
```
import pandas as pd

movies = pd.read_csv("https://raw.githubusercontent.com/peetck/IMDB-Top1000-Movies/
↪master/IMDB-Movie-Data.csv")
movies.head()
```

[14]:
```
   Rank                    Title                     Genre  \
0     1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
1     2               Prometheus  Adventure,Mystery,Sci-Fi
2     3                    Split           Horror,Thriller
3     4                     Sing   Animation,Comedy,Family
4     5            Suicide Squad  Action,Adventure,Fantasy

                                     Description              Director  \
0  A group of intergalactic criminals are forced ...           James Gunn
1  Following clues to the origin of mankind, a te...          Ridley Scott
2  Three girls are kidnapped by a man with a diag...    M. Night Shyamalan
3  In a city of humanoid animals, a hustling thea...  Christophe Lourdelet
4  A secret government agency recruits some of th...            David Ayer

                                        Actors  Year  Runtime (Minutes)  \
0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...  2014                121
1  Noomi Rapace, Logan Marshall-Green, Michael Fa...  2012                124
2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar...  2016                117
3  Matthew McConaughey,Reese Witherspoon, Seth Ma...  2016                108
4  Will Smith, Jared Leto, Margot Robbie, Viola D...  2016                123

   Rating   Votes  Revenue (Millions)  Metascore
0     8.1  757074              333.13         76
1     7.0  485820              126.46         65
2     7.3  157606              138.12         62
3     7.2   60545              270.32         59
4     6.2  393727              325.02         40
```

Here Genre category membership is represented with a comma-separated Genre column.

`from_memberships` is our best option:

[15]:
```
movies_by_genre = from_memberships(movies.Genre.str.split(','), data=movies)
movies_by_genre
```

[15]:
```
                                                                                ␣
↪                                                                  Rank  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True       False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False       1
False  True       False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False       2
       False      False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False       3
                  True       False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False       4
True   True       False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False       5
```

(continues on next page)

(continued from previous page)

```
...                                                                              ␣
↪                                                                         ...
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True     False    False  False False    False False     996
                                            False False False  False   False   True  ␣
↪False False   False    False    False  False False    False False     997
                                                  True  False  False   False   False ␣
↪True  False   False    True     False  False False    False False     998
       True      False     False     True    False False False  False   False   False ␣
↪False False   False    False    False  False False    False False     999
       False     False     False     True    False False True   True    False   False ␣
↪False False   False    False    False  False False    False False    1000

                                                                                 ␣
↪                                                                                ␣
↪ Title  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False   False False False  False   False   False ␣
↪False False   False    False    True    False False    False False    Guardians of the␣
↪Galaxy
False  True      False     False     False   False False False  False   False   False ␣
↪False False   True     False    True    False False    False False              ␣
↪Prometheus
       False     False     False     False   False False False  False   False   True  ␣
↪False False   False    False    False  False True     False False               ␣
↪ Split
                 True      False     True    False False True   False   False   False ␣
↪False False   False    False    False  False False    False False               ␣
↪  Sing
True   True      False     False     False   False False False  True    False   False ␣
↪False False   False    False    False  False False    False False            ␣
↪Suicide Squad
...                                                                              ␣
↪                                                                                ␣
↪   ...
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True     False    False  False False    False False      Secret in␣
↪Their Eyes
                                            False False False  False   False   True  ␣
↪False False   False    False    False  False False    False False          Hostel:␣
↪Part II
                                                  True  False  False   False   False ␣
↪True  False   False    True     False  False False    False False     Step Up 2: The␣
↪Streets
       True      False     False     True    False False False  False   False   False ␣
↪False False   False    False    False  False False    False False             ␣
↪Search Party
       False     False     False     True    False False True   True    False   False ␣
↪False False   False    False    False  False False    False False               ␣
↪Nine Lives

                                                                                 ␣
↪                                                                                ␣
↪  Genre  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
```

(continues on next page)

(continued from previous page)

```
True   True      False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False     Action,
↪Adventure,Sci-Fi
False  True      False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False    Adventure,
↪Mystery,Sci-Fi
       False     False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False            Horror,
↪Thriller
                 True      False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False     Animation,
↪Comedy,Family
True   True      False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False    Action,Adventure,
↪Fantasy
...                                                                                  ␣
↪                                                                                    ␣
↪    ...
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True    False   False  False False    False False        Crime,Drama,
↪Mystery
                                           False False False  False   False   True  ␣
↪False False   False   False   False  False False    False False                    ␣
↪ Horror
                                                 True  False  False   False   False ␣
↪True  False   False   True    False  False False    False False        Drama,Music,
↪Romance
       True      False     False     True   False False False  False   False   False ␣
↪False False   False   False   False  False False    False False                   ␣
↪Adventure,Comedy
       False     False     False     True   False False True   True    False   False ␣
↪False False   False   False   False  False False    False False      Comedy,Family,
↪Fantasy


                                                                                     ␣
↪                                                                                    ␣
↪                     Description  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False    A group of␣
↪intergalactic criminals are forced ...
False  True      False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False    Following clues␣
↪to the origin of mankind, a te...
       False     False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False    Three girls are␣
↪kidnapped by a man with a diag...
                 True      False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False    In a city of␣
↪humanoid animals, a hustling thea...
True   True      False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False    A secret␣
↪government agency recruits some of th...
...                                                                                  ␣
↪                                                                                    ␣
↪                               ...
```

(continues on next page)

(continued from previous page)

```
False  False     False     False     False  True  True  False  False   False   False␣
↪False False   True    False   False  False False    False False    A tight-knit␣
↪team of rising investigators, alo...
                                              False False False  False   False   True  ␣
↪False False   False   False   False  False False    False False    Three American␣
↪college students studying abroa...
                                                    True  False  False   False   False␣
↪True  False   False   True    False  False False    False False    Romantic sparks␣
↪occur between two dance studen...
      True      False     False     True   False False False  False   False   False ␣
↪False False   False   False   False  False False    False False    A pair of␣
↪friends embark on a mission to reuni...
      False     False     False     True   False False True   True    False   False ␣
↪False False   False   False   False  False False    False False    A stuffy␣
↪businessman finds himself trapped ins...


↪                                                                                  ␣
↪Director  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False             James␣
↪Gunn
False  True      False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False           Ridley␣
↪Scott
      False     False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False      M. Night␣
↪Shyamalan
                True      False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False    Christophe␣
↪Lourdelet
True   True      False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False             David␣
↪Ayer
...                                                                                ␣
↪                                                                                  ␣
↪...
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True    False   False  False False    False False              Billy␣
↪Ray
                                              False False False  False   False   True  ␣
↪False False   False   False   False  False False    False False                Eli␣
↪Roth
                                                    True  False  False   False   False␣
↪True  False   False   True    False  False False    False False             Jon M.␣
↪Chu
      True      False     False     True   False False False  False   False   False ␣
↪False False   False   False   False  False False    False False          Scot␣
↪Armstrong
      False     False     False     True   False False True   True    False   False ␣
↪False False   False   False   False  False False    False False        Barry␣
↪Sonnenfeld


↪                                                                                  ␣
↪                         Actors  \
```

(continues on next page)

(continued from previous page)

```
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False    Chris Pratt, Vin␣
↪Diesel, Bradley Cooper, Zoe S...
False  True      False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False    Noomi Rapace,␣
↪Logan Marshall-Green, Michael Fa...
       False     False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False    James McAvoy,␣
↪Anya Taylor-Joy, Haley Lu Richar...
                 True      False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False    Matthew␣
↪McConaughey,Reese Witherspoon, Seth Ma...
True   True      False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False    Will Smith,␣
↪Jared Leto, Margot Robbie, Viola D...
...
↪                                                                                  ␣
↪                                                                                  ␣
↪                                ...
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True    False   False  False False    False False    Chiwetel Ejiofor,
↪ Nicole Kidman, Julia Roberts...
                                           False False False  False   False   True  ␣
↪False False   False   False   False  False False    False False    Lauren German,␣
↪Heather Matarazzo, Bijou Philli...
                                                 True  False  False   False   False ␣
↪True  False   False   True    False  False False    False False    Robert Hoffman,␣
↪Briana Evigan, Cassie Ventura,...
       True      False     False     True   False False False  False   False   False ␣
↪False False   False   False   False  False False    False False    Adam Pally, T.J.␣
↪Miller, Thomas Middleditch,Sh...
       False     False     False     True   False False True   True    False   False ␣
↪False False   False   False   False  False False    False False    Kevin Spacey,␣
↪Jennifer Garner, Robbie Amell,Ch...


↪                                                                        Year  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False    2014
False  True      False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False    2012
       False     False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False    2016
                 True      False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False    2016
True   True      False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False    2016
...
↪                                                                             ...  ␣
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True    False   False  False False    False False    2015
                                           False False False  False   False   True  ␣
↪False False   False   False   False  False False    False False    2007
                                                 True  False  False   False   False ␣
↪True  False   False   True    False  False False    False False    2008
```

(continues on next page)

(continued from previous page)

```
       True      False     False     True   False False False  False   False   False ␣
↪False False   False   False   False  False False    False False    2014
       False     False     False     True   False False True   True    False   False ␣
↪False False   False   False   False  False False    False False    2016


                                                                                    ␣
↪                                                                 Runtime␣
↪(Minutes)  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False                121
False  True      False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False                124
       False     False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False                117
                 True      False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False                108
True   True      False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False                123
...                                                                                 ␣
↪                                                                                  ...
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True    False   False  False False    False False                111
                                           False False False  False   False   True  ␣
↪False False   False   False   False  False False    False False                 94
                                         True  False  False   False   False ␣
↪True  False   False   True    False  False False    False False                 98
       True      False     False     True   False False False  False   False   False ␣
↪False False   False   False   False  False False    False False                 93
       False     False     False     True   False False True   True    False   False ␣
↪False False   False   False   False  False False    False False                 87


                                                                                    ␣
↪                                                                 Rating  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False ␣
↪False False   False   False   True   False False    False False       8.1
False  True      False     False     False  False False False  False   False   False ␣
↪False False   True    False   True   False False    False False       7.0
       False     False     False     False  False False False  False   False   True  ␣
↪False False   False   False   False  False True     False False       7.3
                 True      False     True   False False True   False   False   False ␣
↪False False   False   False   False  False False    False False       7.2
True   True      False     False     False  False False False  True    False   False ␣
↪False False   False   False   False  False False    False False       6.2
...                                                                                 ␣
↪                                                                                  ...
False  False     False     False     False  True  True  False  False   False   False ␣
↪False False   True    False   False  False False    False False       6.2
                                           False False False  False   False   True  ␣
↪False False   False   False   False  False False    False False       5.5
                                         True  False  False   False   False ␣
↪True  False   False   True    False  False False    False False       6.2
       True      False     False     True   False False False  False   False   False ␣
↪False False   False   False   False  False False    False False       5.6
```

(continues on next page)

(continued from previous page)

```
       False     False     False     True   False False True   True    False   False␣
↪False False   False   False   False  False False    False False       5.3

␣
↪                                                                  Votes  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False␣
↪False False   False   False   True   False False    False False    757074
False  True      False     False     False  False False False  False   False   False␣
↪False False   True    False   True   False False    False False    485820
       False     False     False     False  False False False  False   False   True␣
↪False False   False   False   False  False True     False False    157606
                 True      False     True   False False True   False   False   False␣
↪False False   False   False   False  False False    False False     60545
True   True      False     False     False  False False False  True    False   False␣
↪False False   False   False   False  False False    False False    393727
...
↪                                                                    ...
False  False     False     False     False  True  True  False  False   False   False␣
↪False False   True    False   False  False False    False False     27585
                                           False False False  False   False   True␣
↪False False   False   False   False  False False    False False     73152
                                         True  False  False   False   False␣
↪True  False   False   True    False  False False    False False     70699
       True      False     False     True   False False False  False   False   False␣
↪False False   False   False   False  False False    False False      4881
       False     False     False     True   False False True   True    False   False␣
↪False False   False   False   False  False False    False False     12435

␣
↪                                                                 Revenue␣
↪(Millions)  \
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False␣
↪False False   False   False   True   False False    False False                333.
↪13
False  True      False     False     False  False False False  False   False   False␣
↪False False   True    False   True   False False    False False                126.
↪46
       False     False     False     False  False False False  False   False   True␣
↪False False   False   False   False  False True     False False                138.
↪12
                 True      False     True   False False True   False   False   False␣
↪False False   False   False   False  False False    False False                270.
↪32
True   True      False     False     False  False False False  True    False   False␣
↪False False   False   False   False  False False    False False                325.
↪02
...
↪                                                                                ␣
↪...
False  False     False     False     False  True  True  False  False   False   False␣
↪False False   True    False   False  False False    False False                  0.
↪00
                                           False False False  False   False   True␣
↪False False   False   False   False  False False    False False        17.
↪54
```

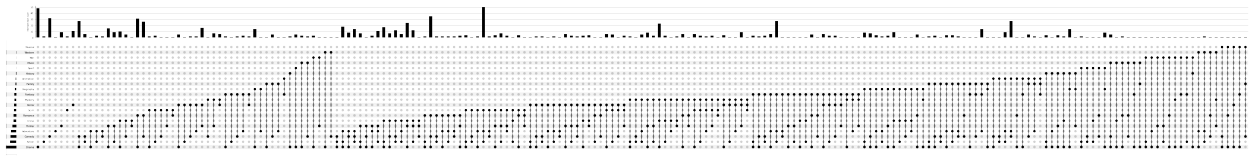
(continues on next page)

(continued from previous page)

```
                                                    True  False  False   False   False␣
↪True  False   False   True    False  False False    False False               58.
↪01
       True      False     False     True   False False False  False   False   False␣
↪False False   False   False   False  False False    False False                0.
↪00
       False     False     False     True   False False True   True    False   False␣
↪False False   False   False   False  False False    False False               19.
↪64


↪                                                                                  ␣
                                                                Metascore
Action Adventure Animation Biography Comedy Crime Drama Family Fantasy History Horror␣
↪Music Musical Mystery Romance Sci-Fi Sport Thriller War   Western
True   True      False     False     False  False False False  False   False   False␣
↪False False   False   False   True   False False    False False          76
False  True      False     False     False  False False False  False   False   False␣
↪False False   True    False   True   False False    False False          65
       False     False     False     False  False False False  False   False   True ␣
↪False False   False   False   False  False True     False False          62
                 True      False     True   False False True   False   False   False␣
↪False False   False   False   False  False False    False False          59
True   True      False     False     False  False False False  True    False   False␣
↪False False   False   False   False  False False    False False          40
...                                                                                ␣
↪                                                                        ...
False  False     False     False     False  True  True  False  False   False   False␣
↪False False   True    False   False  False False    False False          45
                                         False False False  False   False   True ␣
↪False False   False   False   False  False False    False False          46
                                                    True  False  False   False   False␣
↪True  False   False   True    False  False False    False False          50
       True      False     False     True   False False False  False   False   False␣
↪False False   False   False   False  False False    False False          22
       False     False     False     True   False False True   True    False   False␣
↪False False   False   False   False  False False    False False          11

[1000 rows x 12 columns]
```

[16]:
```
UpSet(movies_by_genre)
```

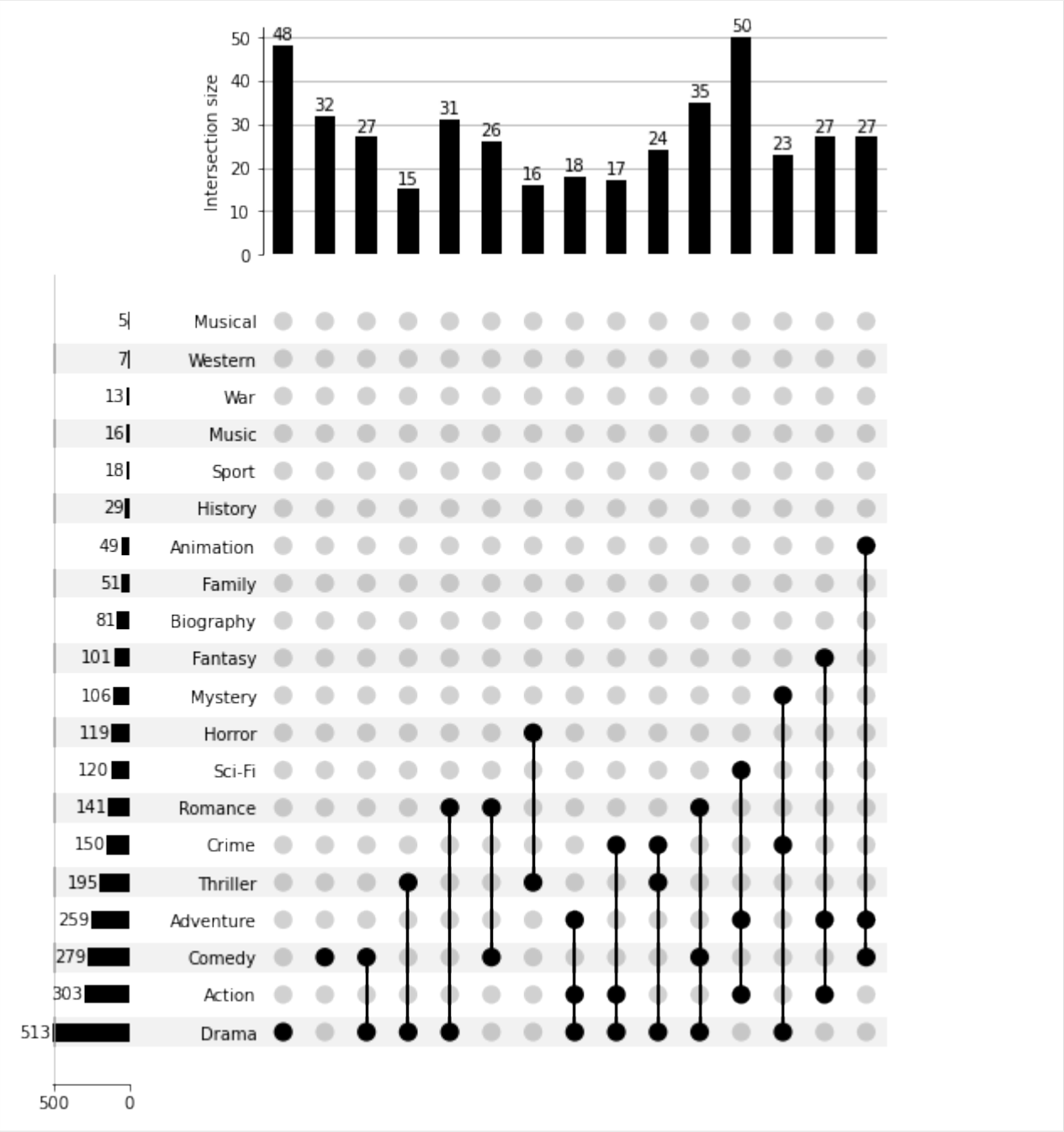
[16]:
```
<upsetplot.plotting.UpSet at 0x7faa985332e8>
```



Given the size of this plot, we limit ourselves to frequent genres:

[17]:
```
UpSet(movies_by_genre, min_subset_size=15, show_counts=True).plot()
```

[17]:
```
{'matrix': <matplotlib.axes._subplots.AxesSubplot at 0x7faaa87e8ef0>,
 'shading': <matplotlib.axes._subplots.AxesSubplot at 0x7faad876a7b8>,
 'totals': <matplotlib.axes._subplots.AxesSubplot at 0x7faac8b93978>,
 'intersections': <matplotlib.axes._subplots.AxesSubplot at 0x7faaf845f978>}
```



If the genres were instead presented as a series of boolean columns, we could use `from_indicators`.

[18]:
```
genre_indicators = pd.DataFrame([{cat: True
                                  for cat in cats}
                                 for cats in movies.Genre.str.split(',').values]).
↪fillna(False)
genre_indicators
```

[18]:
```
     Action  Adventure  Sci-Fi  Mystery  Horror  Thriller  Animation  Comedy  \
0      True       True    True    False   False     False      False   False
1     False       True    True     True   False     False      False   False
```

(continues on next page)

(continued from previous page)

```
2     False      False   False    False    True      True      False   False
3     False      False   False    False   False     False       True    True
4      True       True   False    False   False     False      False   False
..      ...        ...     ...      ...     ...       ...        ...     ...
995   False      False   False     True   False     False      False   False
996   False      False   False    False    True     False      False   False
997   False      False   False    False   False     False      False   False
998   False       True   False    False   False     False      False    True
999   False      False   False    False   False     False      False    True

     Family  Fantasy  Drama  Music  Biography  Romance  History  Crime  \
0     False    False  False  False      False    False    False  False
1     False    False  False  False      False    False    False  False
2     False    False  False  False      False    False    False  False
3      True    False  False  False      False    False    False  False
4     False     True  False  False      False    False    False  False
..      ...      ...    ...    ...        ...      ...      ...    ...
995   False    False   True  False      False    False    False   True
996   False    False  False  False      False    False    False  False
997   False    False   True   True      False     True    False  False
998   False    False  False  False      False    False    False  False
999    True     True  False  False      False    False    False  False

     Western    War  Musical  Sport
0      False  False    False  False
1      False  False    False  False
2      False  False    False  False
3      False  False    False  False
4      False  False    False  False
..       ...    ...      ...    ...
995    False  False    False  False
996    False  False    False  False
997    False  False    False  False
998    False  False    False  False
999    False  False    False  False

[1000 rows x 20 columns]
```

[19]:
```
from upsetplot import from_indicators
# this produces the same result as from_memberships above
movies_by_genre = from_indicators(genre_indicators, data=movies)
```

These columns could also be part of the original matrix. For this case `from_indicators` allows the `indicators` to be specified as a list of column names, or as a function of the data frame.

[20]:
```
movies_with_indicators = pd.concat([movies, genre_indicators], axis=1)
movies_with_indicators
```

[20]:
```
     Rank                    Title                       Genre  \
0       1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
1       2               Prometheus  Adventure,Mystery,Sci-Fi
2       3                    Split           Horror,Thriller
3       4                     Sing   Animation,Comedy,Family
4       5             Suicide Squad  Action,Adventure,Fantasy
..    ...                      ...                       ...
995   996     Secret in Their Eyes       Crime,Drama,Mystery
996   997          Hostel: Part II                    Horror
```

(continues on next page)

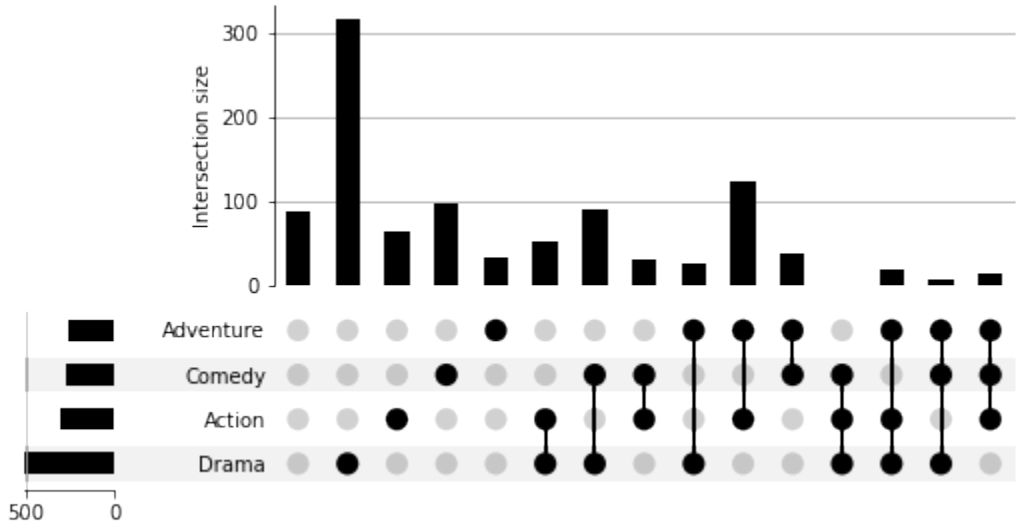
(continued from previous page)

```
997   998   Step Up 2: The Streets       Drama,Music,Romance
998   999             Search Party          Adventure,Comedy
999  1000               Nine Lives     Comedy,Family,Fantasy

                                          Description              Director  \
0    A group of intergalactic criminals are forced ...           James Gunn
1    Following clues to the origin of mankind, a te...          Ridley Scott
2    Three girls are kidnapped by a man with a diag...    M. Night Shyamalan
3    In a city of humanoid animals, a hustling thea...  Christophe Lourdelet
4    A secret government agency recruits some of th...            David Ayer
..                                                 ...                   ...
995  A tight-knit team of rising investigators, alo...             Billy Ray
996  Three American college students studying abroa...              Eli Roth
997  Romantic sparks occur between two dance studen...           Jon M. Chu
998  A pair of friends embark on a mission to reuni...        Scot Armstrong
999  A stuffy businessman finds himself trapped ins...      Barry Sonnenfeld

                                               Actors  Year  \
0    Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...  2014
1    Noomi Rapace, Logan Marshall-Green, Michael Fa...  2012
2    James McAvoy, Anya Taylor-Joy, Haley Lu Richar...  2016
3    Matthew McConaughey,Reese Witherspoon, Seth Ma...  2016
4    Will Smith, Jared Leto, Margot Robbie, Viola D...  2016
..                                                 ...   ...
995  Chiwetel Ejiofor, Nicole Kidman, Julia Roberts...  2015
996  Lauren German, Heather Matarazzo, Bijou Philli...  2007
997  Robert Hoffman, Briana Evigan, Cassie Ventura,...  2008
998  Adam Pally, T.J. Miller, Thomas Middleditch,Sh...  2014
999  Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...  2016

     Runtime (Minutes)  Rating   Votes  ...  Drama  Music  Biography  Romance  \
0                  121     8.1  757074  ...  False  False      False    False
1                  124     7.0  485820  ...  False  False      False    False
2                  117     7.3  157606  ...  False  False      False    False
3                  108     7.2   60545  ...  False  False      False    False
4                  123     6.2  393727  ...  False  False      False    False
..                 ...     ...     ...  ...    ...    ...        ...      ...
995                111     6.2   27585  ...   True  False      False    False
996                 94     5.5   73152  ...  False  False      False    False
997                 98     6.2   70699  ...   True   True      False     True
998                 93     5.6    4881  ...  False  False      False    False
999                 87     5.3   12435  ...  False  False      False    False

     History  Crime  Western    War  Musical  Sport
0      False  False    False  False    False  False
1      False  False    False  False    False  False
2      False  False    False  False    False  False
3      False  False    False  False    False  False
4      False  False    False  False    False  False
..       ...    ...      ...    ...      ...    ...
995    False   True    False  False    False  False
996    False  False    False  False    False  False
997    False  False    False  False    False  False
998    False  False    False  False    False  False
999    False  False    False  False    False  False

[1000 rows x 32 columns]
```

We can now specify some or all category column names instead of passing a separate indicator matrix:

[21]:
```
UpSet(from_indicators(["Drama", "Action", "Comedy", "Adventure"],
                      data=movies_with_indicators))
```

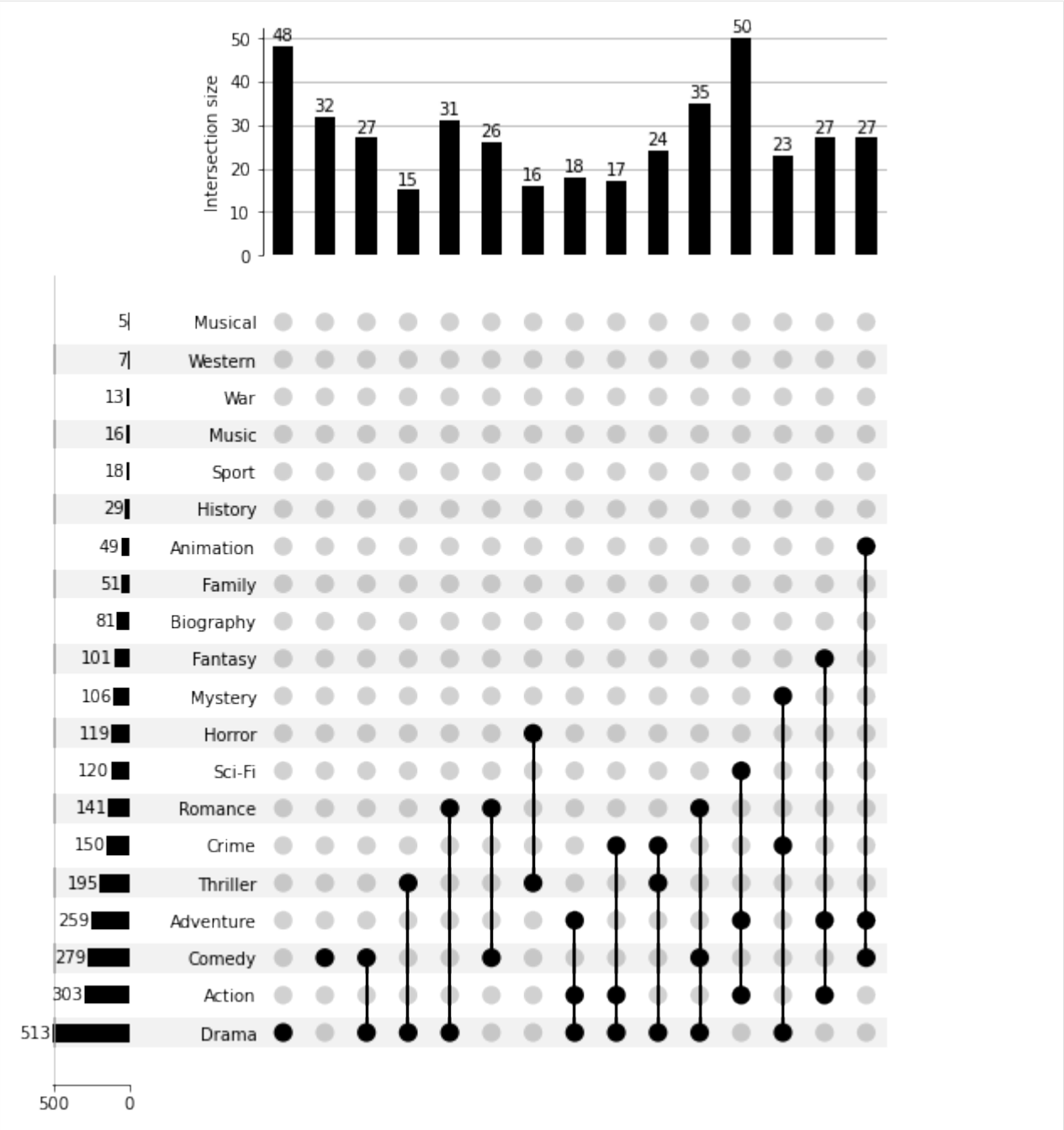
[21]:
```
<upsetplot.plotting.UpSet at 0x7faae8a30a20>
```



Or we can use `pd.select_dtypes` to extract out all boolean columns:

[22]:
```
UpSet(from_indicators(lambda df: df.select_dtypes(bool),
                      data=movies_with_indicators),
      min_subset_size=15, show_counts=True)
```

[22]:
```
<upsetplot.plotting.UpSet at 0x7faab010a5c0>
```



3.3.3 API Reference

Plotting

`upsetplot.plot`(*data*, *fig=None*, ***kwargs*)

Make an UpSet plot of data on fig

Parameters

data [pandas.Series or pandas.DataFrame] Values for each set to plot. Should have multi-index

where each level is binary, corresponding to set membership. If a DataFrame, `sum_over` must be a string or False.

fig [matplotlib.figure.Figure, optional] Defaults to a new figure.

kwargs Other arguments for *UpSet*

Returns

subplots [dict of matplotlib.axes.Axes] Keys are 'matrix', 'intersections', 'totals', 'shading'

class upsetplot.**UpSet**(*data*, *orientation='horizontal'*, *sort_by='degree'*, *sort_categories_by='cardinality'*, *subset_size='auto'*, *sum_over=None*, *min_subset_size=None*, *max_subset_size=None*, *min_degree=None*, *max_degree=None*, *facecolor='auto'*, *other_dots_color=0.18*, *shading_color=0.05*, *with_lines=True*, *element_size=32*, *intersection_plot_elements=6*, *totals_plot_elements=2*, *show_counts=''*, *show_percentages=False*)

Manage the data and drawing for a basic UpSet plot

Primary public method is *plot()*.

Parameters

data [pandas.Series or pandas.DataFrame] Elements associated with categories (a DataFrame), or the size of each subset of categories (a Series). Should have MultiIndex where each level is binary, corresponding to category membership. If a DataFrame, `sum_over` must be a string or False.

orientation [{'horizontal' (default), 'vertical'}] If horizontal, intersections are listed from left to right.

sort_by [{'cardinality', 'degree', None}] If 'cardinality', subset are listed from largest to smallest. If 'degree', they are listed in order of the number of categories intersected. If None, the order they appear in the data input is used.

Changed in version 0.5: Setting None was added.

sort_categories_by [{'cardinality', None}] Whether to sort the categories by total cardinality, or leave them in the provided order.

New in version 0.3.

subset_size [{'auto', 'count', 'sum'}] Configures how to calculate the size of a subset. Choices are:

'auto' (default) If `data` is a DataFrame, count the number of rows in each group, unless `sum_over` is specified. If `data` is a Series with at most one row for each group, use the value of the Series. If `data` is a Series with more than one row per group, raise a ValueError.

'count' Count the number of rows in each group.

'sum' Sum the value of the `data` Series, or the DataFrame field specified by `sum_over`.

sum_over [str or None] If `subset_size='sum'` or `'auto'`, then the intersection size is the sum of the specified field in the `data` DataFrame. If a Series, only None is supported and its value is summed.

min_subset_size [int, optional] Minimum size of a subset to be shown in the plot. All subsets with a size smaller than this threshold will be omitted from plotting. Size may be a sum of values, see `subset_size`.

New in version 0.5.

max_subset_size [int, optional] Maximum size of a subset to be shown in the plot. All subsets with a size greater than this threshold will be omitted from plotting.

New in version 0.5.

min_degree [int, optional] Minimum degree of a subset to be shown in the plot.

New in version 0.5.

max_degree [int, optional] Maximum degree of a subset to be shown in the plot.

New in version 0.5.

facecolor ['auto' or matplotlib color or float] Color for bar charts and active dots. Defaults to black if axes.facecolor is a light color, otherwise white.

Changed in version 0.6: Before 0.6, the default was 'black'

other_dots_color [matplotlib color or float] Color for shading of inactive dots, or opacity (between 0 and 1) applied to facecolor.

New in version 0.6.

shading_color [matplotlib color or float] Color for shading of odd rows in matrix and totals, or opacity (between 0 and 1) applied to facecolor.

New in version 0.6.

with_lines [bool] Whether to show lines joining dots in the matrix, to mark multiple categories being intersected.

element_size [float or None] Side length in pt. If None, size is estimated to fit figure

intersection_plot_elements [int] The intersections plot should be large enough to fit this many matrix elements. Set to 0 to disable intersection size bars.

Changed in version 0.4: Setting to 0 is handled.

totals_plot_elements [int] The totals plot should be large enough to fit this many matrix elements.

show_counts [bool or str, default=False] Whether to label the intersection size bars with the cardinality of the intersection. When a string, this formats the number. For example, '%d' is equivalent to True.

show_percentages [bool, default=False] Whether to label the intersection size bars with the percentage of the intersection relative to the total dataset. This may be applied with or without show_counts.

New in version 0.4.

Methods

add_catplot(self, kind[, value, elements])	Add a seaborn catplot over subsets when *plot()* is called.
add_stacked_bars(self, by[, sum_over, ...])	Add a stacked bar chart over subsets when *plot()* is called.
make_grid(self[, fig])	Get a SubplotSpec for each Axes, accounting for label text width
plot(self[, fig])	Draw all parts of the plot onto fig or a new figure
plot_intersections(self, ax)	Plot bars indicating intersection size

Continued on next page

Table 1 – continued from previous page

plot_matrix(self, ax)	Plot the matrix of intersection indicators onto ax
plot_totals(self, ax)	Plot bars indicating total set size
style_subsets(self[, present, absent, ...])	Updates the style of selected subsets' bars and matrix dots

plot_shading

add_catplot (*self*, *kind*, *value=None*, *elements=3*, ***kw*)

Add a seaborn catplot over subsets when *plot()* is called.

Parameters

kind [str] One of {"point", "bar", "strip", "swarm", "box", "violin", "boxen"}

value [str, optional] Column name for the value to plot (i.e. y if orientation='horizontal'), required if `data` is a DataFrame.

elements [int, default=3] Size of the axes counted in number of matrix elements.

****kw** [dict] Additional keywords to pass to `seaborn.catplot()`.

Our implementation automatically determines 'ax', 'data', 'x', 'y' and 'orient', so these are prohibited keys in `kw`.

Returns

None

add_stacked_bars (*self*, *by*, *sum_over=None*, *colors=None*, *elements=3*, *title=None*)

Add a stacked bar chart over subsets when *plot()* is called.

Used to plot categorical variable distributions within each subset.

New in version 0.6.

Parameters

by [str] Column name within the dataframe for color coding the stacked bars, containing discrete or categorical values.

sum_over [str, optional] Ordinarily the bars will chart the size of each group. sum_over may specify a column which will be summed to determine the size of each bar.

colors [Mapping, list-like, str or callable, optional] The facecolors to use for bars corresponding to each discrete label, specified as one of:

Mapping Maps from label to matplotlib-compatible color specification.

list-like A list of matplotlib colors to apply to labels in order.

str The name of a matplotlib colormap name.

callable When called with the number of labels, this should return a list-like of that many colors. Matplotlib colormaps satisfy this callable API.

None Uses the matplotlib default colormap.

elements [int, default=3] Size of the axes counted in number of matrix elements.

title [str, optional] The axis title labelling bar length.

Returns

None

make_grid(*self*, *fig=None*)
Get a SubplotSpec for each Axes, accounting for label text width

plot(*self*, *fig=None*)
Draw all parts of the plot onto fig or a new figure

Parameters

fig [matplotlib.figure.Figure, optional] Defaults to a new figure.

Returns

subplots [dict of matplotlib.axes.Axes] Keys are 'matrix', 'intersections', 'totals', 'shading'

plot_intersections(*self*, *ax*)
Plot bars indicating intersection size

plot_matrix(*self*, *ax*)
Plot the matrix of intersection indicators onto ax

plot_totals(*self*, *ax*)
Plot bars indicating total set size

style_subsets(*self*, *present=None*, *absent=None*, *min_subset_size=None*, *max_subset_size=None*, *min_degree=None*, *max_degree=None*, *facecolor=None*, *edgecolor=None*, *hatch=None*, *linewidth=None*, *linestyle=None*, *label=None*)
Updates the style of selected subsets' bars and matrix dots

Parameters are either used to select subsets, or to style them with attributes of `matplotlib.patches.Patch`, apart from label, which adds a legend entry.

Parameters

present [str or list of str, optional] Category or categories that must be present in subsets for styling.

absent [str or list of str, optional] Category or categories that must not be present in subsets for styling.

min_subset_size [int, optional] Minimum size of a subset to be styled.

max_subset_size [int, optional] Maximum size of a subset to be styled.

min_degree [int, optional] Minimum degree of a subset to be styled.

max_degree [int, optional] Maximum degree of a subset to be styled.

facecolor [str or matplotlib color, optional] Override the default UpSet facecolor for selected subsets.

edgecolor [str or matplotlib color, optional] Set the edgecolor for bars, dots, and the line between dots.

hatch [str, optional] Set the hatch. This will apply to intersection size bars, but not to matrix dots.

linewidth [int, optional] Line width in points for edges.

linestyle [str, optional] Line style for edges.

label [str, optional] If provided, a legend will be added

Dataset loading and generation

upsetplot.**from_contents**(*contents*, *data=None*, *id_column='id'*)
Build data from category listings

Parameters

contents [Mapping (or iterable over pairs) of strings to sets] Keys are category names, values are sets of identifiers (int or string).

data [DataFrame, optional] If provided, this should be indexed by the identifiers used in `Python Documentation contents`.

id_column [str, default='id'] The column name to use for the identifiers in the output.

Returns

DataFrame `data` is returned with its index indicating category membership, including a column named according to id_column. If data is not given, the order of rows is not assured.

Notes

The order of categories in the output DataFrame is determined from `Python Documentation contents`, which may have non-deterministic iteration order.

Examples

```
>>> from upsetplot import from_contents
>>> contents = {'cat1': ['a', 'b', 'c'],
...             'cat2': ['b', 'd'],
...             'cat3': ['e']}
>>> from_contents(contents)
                  id
cat1  cat2  cat3
True  False False  a
      True  False  b
      False False  c
False True  False  d
      False True   e
>>> import pandas as pd
>>> contents = {'cat1': [0, 1, 2],
...             'cat2': [1, 3],
...             'cat3': [4]}
>>> data = pd.DataFrame({'favourite': ['green', 'red', 'red',
...                                    'yellow', 'blue']})
>>> from_contents(contents, data=data)
                   id favourite
cat1  cat2  cat3
True  False False   0     green
      True  False   1       red
      False False   2       red
False True  False   3    yellow
      False True    4      blue
```

upsetplot.**from_indicators**(*indicators*, *data=None*)
Load category membership indicated by a boolean indicator matrix

This loader also supports the case where the indicator columns can be derived from `data`.

New in version 0.6.

Parameters

indicators [DataFrame-like of booleans, Sequence of str, or callable] Specifies the category indicators (boolean mask arrays) within `data`, i.e. which records in `data` belong to which categories.

If a list of strings, these should be column names found in `data` whose values are boolean mask arrays.

If a DataFrame, its columns should correspond to categories, and its index should be a subset of those in `data`, values should be True where a data record is in that category, and False or NA otherwise.

If callable, it will be applied to `data` after the latter is converted to a Series or DataFrame.

data [Series-like or DataFrame-like, optional] If given, the index of category membership is attached to this data. It must have the same length as `indicators`. If not given, the series will contain the value 1.

Returns

DataFrame or Series `data` is returned with its index indicating category membership. It will be a Series if `data` is a Series or 1d numeric array or None.

Notes

Categories with indicators that are all False will be removed.

Examples

```
>>> import pandas as pd
>>> from upsetplot import from_indicators
```

Just indicators >>> indicators = {“cat1”: [True, False, True, False], … “cat2”: [False, True, False, False], … “cat3”: [True, True, False, False]} >>> from_indicators(indicators) cat1 cat2 cat3 True False True 1.0 False True True 1.0 True False False 1.0 False False False 1.0 Name: ones, dtype: float64

Where indicators are included within data, specifying columns by name >>> data = pd.DataFrame({“value”: [5, 4, 6, 4], **indicators}) >>> from_indicators([“cat1”, “cat3”], data=data)

value cat1 cat2 cat3

cat1 cat3 True True 5 True False True False True 4 False True True True False 6 True False False False False 4 False False False

Making indicators out of all boolean columns >>> from_indicators(lambda data: data.select_dtypes(bool), data=data)

value cat1 cat2 cat3

cat1 cat2 cat3 True False True 5 True False True False True True 4 False True True True False False 6 True False False False False False 4 False False False

Using a dataset with missing data, we can use missingness as an indicator >>> data = pd.DataFrame({“val1”: [pd.NA, .7, pd.NA, .9], … “val2”: [“male”, pd.NA, “female”, “female”], … “val3”: [pd.NA, pd.NA, 23000, 78000]}) >>> from_indicators(pd.isna, data=data)

val1 val2 val3

val1 val2 val3 True False True <NA> male <NA> False True True 0.7 <NA> <NA> True False False <NA> female 23000 False False False 0.9 female 78000

upsetplot.**from_memberships**(*memberships*, *data=None*)

Load data where each sample has a collection of category names

The output should be suitable for passing to *UpSet* or *plot*.

Parameters

memberships [sequence of collections of strings] Each element corresponds to a data point, indicating the sets it is a member of. Each category is named by a string.

data [Series-like or DataFrame-like, optional] If given, the index of category memberships is attached to this data. It must have the same length as `memberships`. If not given, the series will contain the value 1.

Returns

DataFrame or Series `data` is returned with its index indicating category membership. It will be a Series if `data` is a Series or 1d numeric array. The index will have levels ordered by category names.

Examples

```
>>> from upsetplot import from_memberships
>>> from_memberships([
...     ['cat1', 'cat3'],
...     ['cat2', 'cat3'],
...     ['cat1'],
...     []
... ])
cat1   cat2   cat3
True   False  True     1
False  True   True     1
True   False  False    1
False  False  False    1
Name: ones, dtype: ...
>>> # now with data:
>>> import numpy as np
>>> from_memberships([
...     ['cat1', 'cat3'],
...     ['cat2', 'cat3'],
...     ['cat1'],
...     []
... ], data=np.arange(12).reshape(4, 3))
                  0   1   2
cat1  cat2  cat3
True  False True   0   1   2
False True  True   3   4   5
True  False False  6   7   8
False False False  9  10  11
```

upsetplot.**generate_counts**(*seed=0*, *n_samples=10000*, *n_categories=3*)

Generate artificial counts corresponding to set intersections

Parameters

seed [int] A seed for randomisation

n_samples [int] Number of samples to generate statistics over

n_categories [int] Number of categories (named "cat0", "cat1", ...) to generate

Returns

Series Counts indexed by boolean indicator mask for each category.

See also:

generate_samples Generates a DataFrame of samples that these counts are derived from.

upsetplot.**generate_samples**(*seed=0*, *n_samples=10000*, *n_categories=3*)

Generate artificial samples assigned to set intersections

Parameters

seed [int] A seed for randomisation

n_samples [int] Number of samples to generate

n_categories [int] Number of categories (named "cat0", "cat1", ...) to generate

Returns

DataFrame Field 'value' is a weight or score for each element. Field 'index' is a unique id for each element. Index includes a boolean indicator mask for each category.

Note: Further fields may be added in future versions.

See also:

generate_counts Generates the counts for each subset of categories corresponding to these samples.

3.3.4 Changelog

What's new in version 0.6

- Added *add_stacked_bars*, similar to *add_catplot* but to add stacked bar charts to show discrete variable distributions within each subset. (#137)
- Improved ability to control colors, and added a new example of same. Parameters `other_dots_color` and `shading_color` were added. `facecolor` will now default to white if `matplotlib.rcParams['axes.facecolor']` is dark. (#138)
- Added *style_subsets* to colour intersection size bars and matrix dots in the plot according to a specified query. (#152)
- Added *from_indicators* to allow yet another data input format. This allows category membership to be easily derived from a DataFrame, such as when plotting missing values in the columns of a DataFrame. (#143)

What's new in version 0.5

- Support using input intersection order with `sort_by=None` (#133 with thanks to Brandon B).
- Add parameters for filtering by subset size (with thanks to Sichong Peng) and degree. (#134)
- Fixed an issue where tick labels were not given enough space and overlapped category totals. (#132)
- Fixed an issue where our implementation of `sort_by='degree'` apparently gave incorrect results for some inputs and versions of Pandas. (#134)

What's new in version 0.4.4

- Fixed a regresion which caused the first column to be hidden (#125)

What's new in version 0.4.3

- Fixed issue with the order of catplots being reversed for vertical plots (#122 with thanks to Enrique Fernandez-Blanco)
- Fixed issue with the x limits of vertical plots (#121).

What's new in version 0.4.2

- Fixed large x-axis plot margins with high number of unique intersections (#106 with thanks to Yidi Huang)

What's new in version 0.4.1

- Fixed the calculation of percentage which was broken in 0.4.0. (#101)

What's new in version 0.4

- Added option to display both the absolute frequency and the percentage of the total for each intersection and category. (#89 with thanks to Carlos Melus and Aaron Rosenfeld)
- Improved efficiency where there are many categories, but valid combinations are sparse, if `sort_by='degree'`. (#82)
- Permit truthy (not necessarily bool) values in index. (#74 with thanks to @ZaxR)
- `intersection_plot_elements` can now be set to 0 to hide the intersection size plot when *`add_catplot`* is used. (#80)

What's new in version 0.3

- Added *`from_contents`* to provide an alternative, intuitive way of specifying category membership of elements.
- To improve code legibility and intuitiveness, `sum_over=False` was deprecated and a `subset_size` parameter was added. It will have better default handling of DataFrames after a short deprecation period.
- `generate_data` has been replaced with *`generate_counts`* and *`generate_samples`*.
- Fixed the display of the "intersection size" label on plots, which had been missing.
- Trying to improve nomenclature, upsetplot now avoids "set" to refer to the top-level sets, which are now to be known as "categories". This matches the intuition that categories are named, logical groupings, as opposed to "subsets". To this end:
  - *`generate_counts`* (formerly `generate_data`) now names its categories "cat1", "cat2" etc. rather than "set1", "set2", etc.
  - the `sort_sets_by` parameter has been renamed to `sort_categories_by` and will be removed in version 0.4.

What's new in version 0.2.1

- Return a Series (not a DataFrame) from *from_memberships* if data is 1-dimensional.

What's new in version 0.2

- Added *from_memberships* to allow a more convenient data input format.
- *plot* and *UpSet* now accept a pandas.DataFrame as input, if the sum_over parameter is also given.
- Added an *add_catplot* method to *UpSet* which adds Seaborn plots of set intersection data to show more than just set size or total.
- Shading of subset matrix is continued through to totals.
- Added a show_counts option to show counts at the ends of bar plots. (#5)
- Defined _repr_html_ so that an *UpSet* object will render in Jupyter notebooks. (#36)
- Fix a bug where an error was raised if an input set was empty.

Bibliography

[Lex2014] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, Hanspeter Pfister, *UpSet: Visualization of Intersecting Sets*, IEEE Transactions on Visualization and Computer Graphics (InfoVis '14), vol. 20, no. 12, pp. 1983–1992, 2014. doi: doi.org/10.1109/TVCG.2014.2346248

Index