About

Behavpy is the default object in ethoscopy, a way of storing your metadata and data in a single structure, whilst adding methods to help you manipulate and analyse your data.

Metadata is crucial for proper statistical analysis of the experimental data. In the context of the Ethoscope, the data is a long time series of recorded variables, such as position and velocity for each individual. It is easier and neater to always keep the data and metadata together. As such, behavpy class is a child class of Pandas dataframe, the widely used data table package in python, but enhanced by the addition of a metadata as a class variable. Both the metadata and data must be linked by a common unique identifier for each individual (the 'id' column), which is automatically done if you've loaded with Ethoscopy (If not see the bottom of the page for the requirements for creating a behavpy object from alternative data sources).

The behavpy class has variety of methods designed to augment, filter, and inspect your data which we will go into later. However, if you're not already familiar with Pandas, take some time to look through their guide to get an understanding of its many uses.

Initialising behavpy

To create a behavpy object you need matching metadata and data. To match they both need to have an id column that has linked/same id's in each. Don't worry, if you downloaded/formatted the data using the built in ethoscopy functions shown previously they'll be in there already.

```python
import ethoscopy as etho


# You can load you data using the previously mentioned functions
meta = etho.link_meta_index(meta_loc, remote, local)
data = etho.load_ethoscope(meta, reference_hour = 9.0, FUN = partial(etho.sleep_annotation,
time_window_length = 60, min_time_immobile = 300))


# Or you can load then from a saved object of your choice into a pandas df,
# mine is a pickle file
meta = pd.read_pickle('users\folder\experiment1_meta.pkl')
data = pd.read_pickle('users\folder\experiment1_data.pkl')


# to initialise a behavpy object, just call the class with the data and meta as arguments. Have check = True to
ensure the ids match between metadata and data.
# As of version 1.3.0 you can choose the colour palette for the plotly plots - see
https://plotly.com/python/discrete-color for the choices
# The default is 'Safe' (the one used before), but this example we'll use 'Vivid'
df = etho.behavpy(data = data, meta = meta, colour = 'Vivid', check = True)
```

Using behavpy with non-ethoscope data

The behavpy class is made to work with the ethoscope system, utilising the data structure it records to create the analytics pipeline you'll see next. However, you can still use it on non-ethoscope data if you follow the same structure.

Data sources:
You will need the metadata file as discussed prior, however you will need to manually create a column called id that contains a unique id per specimen in the experiment.
Additionally, you will need a data source where each row is a log of a time per specimen. Each row must have the following

- **id** column, that references back to the unique id in the metadata
- **t** column, the time (in seconds) the row is describing, i.e. 0, 60, 120, 180, 240
- **moving** column, a boolean column (true/false) of whether the specimen is moving

The above columns are necessary for all the methods to work, but feel free to add other columns with extra information per timestamp. Both these data sources must be converted to pandas DataFrames, which can then be used to create a behavpy class object as shown above.