Useful Snippets

A collection of useful code snippets. Here you can find code that allows you to get to get started on common tasks in Mesa.

Models with Discrete Time

If you have *Multiple* type agents and one of them has time attribute you can still build a model that is run by discrete time. In this example, each step of the model, and the agents have a time attribute that is equal to the discrete time to run its own step.

```
if self.model.schedule.time in self.discrete_time:
    self.model.space.move_agent(self, new_pos)
```

Using `numpy.random`

Sometimes you need to use `numpy`'s `random` library, for example to get a Poisson distribution.

```
class MyModel(Model):
    def __init__(self, ...):
        super().__init__()
        self.random = np.random.default_rng(seed)
```

And just use *numpy*'s random as usual, e.g. `self.random.poisson()`.

Using multi-process `batch_run` on Windows

You will have an issue with *batch_run* and *number_processes = None*. Your cell will show no progress, and in your terminal you will receive *AttributeError: Can't get attribute 'MoneyModel' on <module '__main__' (built-in)>*. One way to overcome this is to take your code outside of Jupyter and adjust the above code as follows.

```
from multiprocessing import freeze_support

params = {"width": 10, "height": 10, "N": range(10, 500, 10)}

if __name__ == '__main__':
    freeze_support()
    results = batch_run(
            MoneyModel,
            parameters=params,
            iterations=5,
            max_steps=100,
            number_processes=None,
            data_collection_period=1,
            display_progress=True,
    )
```

If you would still like to run your code in Jupyter you will need to adjust the cell as noted above. Then you can you can add the nbmultitask library or look at this stackoverflow.