

twinlab.Emulator.sample

Emulator.sample(*df*, *num_samples*, *params*=<*twinlab.params.SampleParams* object>, *wait*=True, *verbose*=False)

Draw samples from a trained emulator that exists on the twinLab cloud.

A secondary functionality of the emulator is to draw sample predictions from the trained emulator. Rather than quantifying the uncertainty in the predictions, this method draws samples from the emulator. The collection of samples can be used to explore the distribution of the emulator predictions. Each sample is a possible prediction of the emulator, and therefore a prediction of a possible new observation from the data-generation process. The covariance in the emulator predictions can therefore be explored, which is particularly useful for functional Gaussian Processes.

If the output of the multi-indexed DataFrame needs to be manipulated then we provide the convenience functions:

- `tl.get_sample`: Isolate an individual sample into a new `pandas.DataFrame`
- `tl.join_samples`: Join together multiple sets of samples into a single `pandas.DataFrame`

Parameters:

- **df** ([pandas.DataFrame](#)) – The `X` values for which to draw samples.
- **num_samples** ([int](#)) – Number of samples to draw for each row of the evaluation data.
- **params** ([SampleParams](#), optional) – A *SampleParams* object with sampling parameters.
- **wait** ([bool](#), optional) – If `True` wait for the job to complete, otherwise return the process ID and exit.
- **verbose** ([bool](#), optional) – Display detailed information about the operation while running.

Returns:

By default a multi-indexed DataFrame containing the `y` samples drawn from the emulator. Instead, if `wait=False` the process ID is returned. The results can then be retrieved later using `Emulator.get_process(<process_id>)`. Process IDs associated with an emulator can be found using `Emulator.list_processes()`.

Example

```
emulator = tl.Emulator("quickstart")
df = pd.DataFrame({'x': [0.1, 0.2, 0.3, 0.4]})
emulator.sample(df, 3)
```

	y		
	0	1	2
0	0.490081	1.336099	0.608441
1	0.829179	1.038671	0.807405
2	0.805102	0.773975	0.984713
3	0.605568	0.416630	0.713652

< [Previous](#)
[twinlab.Emulator.predict](#)

[twinlab.Emulator.plot](#) > [Next](#)