

---

# dp-learn Documentation

*Release latest*

Sep 28, 2021



---

## Contents

---

<b>1</b>	<b>Quickstart</b>	<b>1</b>
<b>2</b>	<b>User Guide</b>	<b>3</b>
<b>3</b>	<b>Downloads</b>	<b>5</b>



### 1.1 Installation

```
$ pip install dplearn
```

Or, you can download the source and

```
$ python setup.py install
```

Add sudo in the beginning if you met problem.



## 2.1 Overview of dplearn

### 2.1.1 Why Should I Use This?

This is a Python package for data analysis which contains some very useful tools and functions. The package is continually updated.

### 2.1.2 Modules

- **tools** Useful functions helping you to speed up your programming.
- **math** Functions that related to math calculation.
- **extra** Some other functions that are very useful in specific situations.

### 2.1.3 Usage

#### Use As A Python Module

```
from dplearn.tools import clean
data = clean(data)
print(data)
```

### 2.1.4 Documentation

Check out the latest `dplearn` documentation at [Read the Docs](#)

## 2.1.5 Contributing

Please send pull requests, very much appreciated.

1. Fork the [repository](#) on GitHub.
2. Make a branch off of master and commit your changes to it.
3. Create a Pull Request with your contribution

## 2.2 Tutorial

### 2.2.1 Tutorial on module dplearn.quant

#### wrapKLine

This is a function of wrapping K-Line dataframe into longer-duration one.

```
import dplearn.data_sample as ds
from dplearn.quant import wrapKLine

data = ds.kLine()
open_c = "open"
close_c = "close"
high_c = "high"
low_c = "low"
vol_c = "vol"
ts_c = "time"
ts_format = "%Y-%m-%d %H:%M:%S"
wrap = "1h"

df_new = wrapKLine(data, open_c, close_c, high_c, low_c, vol_c, ts_c, ts_format, wrap)

print(df_new.head())
```

### 2.2.2 Tutorial on module dplearn.tools

#### continue\_check

```
from dplearn.tools import continue_check
check_value = continue_check("Do you want to continue this procedure? [Y/n]: ")

if check_value in ['n', 'N']:
    break
else:
    pass
```



## CHAPTER 3

---

Downloads

---