pandas.get_dummies

pandas.**get_dummies**(*data*, *prefix=None*, *prefix_sep='_'*, *dummy_na=False*, *columns=None*, *sparse=False*, *drop_first=False*)

Convert categorical variable into dummy/indicator variables

Parameters:

data *: array-like, Series, or DataFrame*

prefix *: string, list of strings, or dict of strings, default None*

String to append DataFrame column names Pass a list with length equal to the number of columns when calling get_dummies on a DataFrame. Alternativly, *prefix* can be a dictionary mapping column names to prefixes.

prefix_sep *: string, default '_'*

If appending prefix, separator/delimiter to use. Or pass a list or dictionary as with *prefix.*

dummy_na *: bool, default False*

Add a column to indicate NaNs, if False NaNs are ignored.

columns *: list-like, default None*

Column names in the DataFrame to be encoded. If *columns* is None then all the columns with *object* or *category* dtype will be converted.

sparse *: bool, default False*

Whether the dummy columns should be sparse or not. Returns SparseDataFrame if *data* is a Series or if all columns are included. Otherwise returns a DataFrame with some SparseBlocks.

New in version 0.16.1.

drop_first *: bool, default False*

Whether to get k-1 dummies out of n categorical levels by removing the first level.

New in version 0.18.0.

Returns

——-

dummies *: DataFrame or SparseDataFrame*

See also: `Series.str.get_dummies`

Examples

```
>>> import pandas as pd
>>> s = pd.Series(list('abca'))
```

```
>>> pd.get_dummies(s)
   a  b  c
0  1  0  0
1  0  1  0
2  0  0  1
3  1  0  0
```

```
>>> s1 = ['a', 'b', np.nan]
```

```
>>> pd.get_dummies(s1)
   a  b
0  1  0
1  0  1
2  0  0
```

```
>>> pd.get_dummies(s1, dummy_na=True)
   a  b  NaN
0  1  0    0
1  0  1    0
2  0  0    1
```

```
>>> df = pd.DataFrame({'A': ['a', 'b', 'a'], 'B': ['b', 'a', 'c'],
                       'C': [1, 2, 3]})
```

```
>>> pd.get_dummies(df, prefix=['col1', 'col2'])
   C  col1_a  col1_b  col2_a  col2_b  col2_c
```

```
0  1       1       0       0       1       0
1  2       0       1       1       0       0
2  3       1       0       0       0       1
```

```
>>> pd.get_dummies(pd.Series(list('abcaa')))
   a  b  c
0  1  0  0
1  0  1  0
2  0  0  1
3  1  0  0
4  1  0  0
```

```
>>> pd.get_dummies(pd.Series(list('abcaa')), drop_first=True)
   b  c
0  0  0
1  1  0
2  0  1
3  0  0
4  0  0
```