



pandas.crosstab()



Function `crosstab()` dalam Pandas membuat cross-tabulation table untuk menganalisis hubungan antara dua atau lebih categorical variables.

Misalkan

```
gender = pd.Series(['M', 'F', 'M', 'F', 'M', 'F', 'M'])
preference = pd.Series(['A', 'B', 'A', 'A', 'B', 'B', 'A'])
age_group = pd.Series(['Young', 'Old', 'Young', 'Old', 'Young', 'Old', 'Young'])
```

Contoh:

```
pd.crosstab(gender, preference)
```

Output:

	A	B
F	1	2
M	3	1



```
pd.crosstab(gender, preference, normalize='index')
```

Output:

	A	B
F	0.33	0.67
M	0.75	0.25



pandas.cut()



Function `cut()` dalam Pandas membagi continuous numerical data menjadi discrete bins atau categories berdasarkan range values.

Misalkan

```
ages = pd.Series([22, 25, 47, 52, 46, 56, 55, 60, 51, 45])
scores = pd.Series([65, 72, 88, 91, 45, 78, 93, 67, 82, 76])
```

Contoh:

```
pd.cut(ages, bins=3)
```

Output:

	0		
0	(21.962, 34.667]	5	(47.333, 60.0]
1	(21.962, 34.667]	6	(47.333, 60.0]
2	(34.667, 47.333]	7	(47.333, 60.0]
3	(47.333, 60.0]	8	(47.333, 60.0]
4	(34.667, 47.333]	9	(34.667, 47.333]





numpy.clip()



Method `clip()` dalam NumPy digunakan untuk membatasi nilai dalam array agar berada dalam range minimum dan maksimum yang ditentukan.

Misalkan

```
array_data = np.array([1, 5, 10, 15, 20, 25])
matrix_data = np.array([[2, 8, 12], [18, 22, 30]])
```

Contoh:

```
np.clip(array_data, 5, 20)
```

Output: "array([5, 5, 10, 15, 20, 20])"

```
np.clip(matrix_data, 10, 25)
```

Output: "array([[10, 10, 12],
[18, 22, 25]])"



numpy.nanmean()



Method `nanmean()` dalam NumPy menghitung rata-rata array dengan mengabaikan nilai NaN (Not a Number) dalam perhitungan.

Misalkan

```
data_with_nan = np.array([1, 2, np.nan, 4, 5, np.nan])
matrix_with_nan = np.array([[1, np.nan, 3], [4, 5, np.nan]])
```

Contoh:

```
np.nanmean(data_with_nan)
```

Output: "np.float64(3.0)"

```
np.nanmean(matrix_with_nan, axis=1)
```

Output: "array([2., 4.5])"





numpy.percentile()



Method `percentile()` dalam NumPy menghitung nilai pada persentil tertentu dari distribusi data untuk analisis statistik.

Misalkan

```
scores = np.array([65, 70, 75, 80, 85, 90, 95])
sales_data = np.array([100, 150, 200, 250, 300, 400, 500])
```

Contoh:

```
np.percentile(scores, 50)
```

Output: "np.float64(80.0)"

```
np.percentile(sales_data, [25, 75])
```

Output: "array([175., 350.])"



numpy.diff()



Method `diff()` dalam NumPy menghitung selisih antara elemen-elemen berturut-turut dalam array untuk analisis perubahan.

Misalkan

```
time_series = np.array([10, 12, 15, 13, 18, 20])
cumulative_data = np.array([5, 12, 20, 35, 50])
```

Contoh:

```
np.diff(time_series)
```

Output: "array([2, 3, -2, 5, 2])"

```
np.diff(cumulative_data, n=2)
```

Output: "array([1, 7, 0])"

