

Data Cleaning

dropna()

Remove missing values from a dataset

fillna()

Fill missing values with a specified value

astype()

Convert column data type

nan_to_num()

Replace NaN with numeric value

reshape()

Reshape arrays without changing data

unique()

Get all unique values in a column



dropna()



Remove missing values from a dataset

Used to remove rows or columns with missing values (NaNs) in a DataFrame.



Example:

```
import pandas as pd
df =
pd.read_csv('data.csv')
df_clean = df.dropna()
```

	1	2	1.3	
1	30	133	200	
2	30	188	234	
4	30	239	238	
5	29	238	237	
0	38	238	238	
4	30	274	235	
6	50	295	239	
7	38	238	230	
18	42	238	280	
17	38	235	246	
14	28	233	286	
15	38	288	280	
10	36	238	257	
10	65	353	250	

fillna()



Purpose

- Fill missing values in a dataset

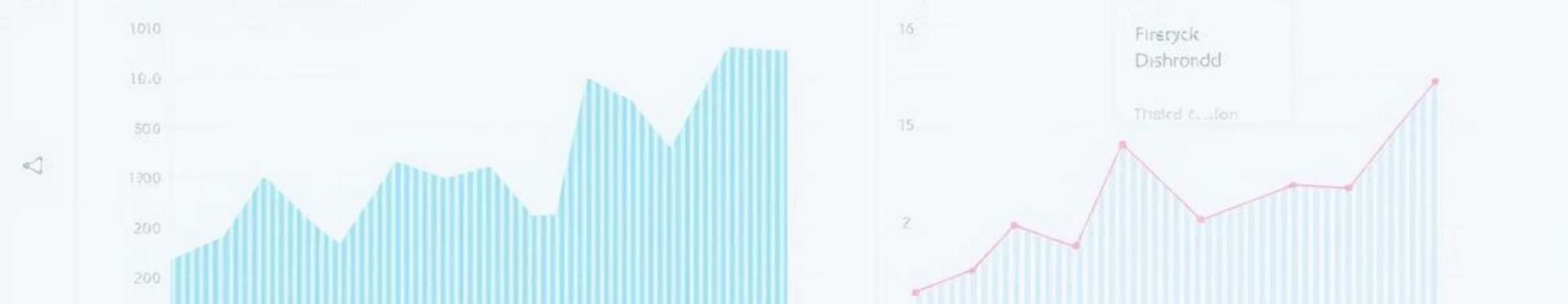
Replaces missing values with a specified constant or method (like forward fill or backward fill).



Example

```
df['column'].fillna(0, inplace=True)
```

	Tatr Oill	Pay Ont	Tatr Oill	Tatr Oill	Patr Oill	Percinll Fecestatione	Peta Fecel
	Pollrance	Ecoppmante	Repercnate	Pollncase	Pollenate	Payment	Pollcr
1	31270	13,757	17720	13,7570	13,670	16,650	13
1	33700	11,420	12529	234420	35,300	14,000	12
1	34570	13,750	13573	25,1319	14,999	25,730	12
1	42550	13,404	13544	261550	15,790	11,654	12
1	35600	12,954	13744	223550	19,200	13,700	12
1	33800	22,790	15504	266570	19,820	13,844	1
1	35590	12,990	18083	227550	16,090	12,740	7
1	63170	13,740	14645	211252	16,355	11,597	3
1	33850	12,400	15564	225590	4,1150	13,487	3
1	33555	14,200	16574	166752	3,5550	13,747	2
1	34590	41,700	15148	286754	3,5545	13,847	3
1	31590	13,900	16574	245316	5,387	15,249	3
1	35590	13,630	12594	125538	9,1370	16,644	2
1	25560	11,790	13972	145508	3,389	16,740	3
1	33062	15,990	13676	215560	34090	13,400	3
1	35770	15,990	12376	165390	41399	13,540	6
1	31650	12,700	18677	286440	35367	13,543	2
1	33380	15,797	15778	275108	93772	15,500	3
1	35120	15,800	14573	225550	43377	16,770	9
1	35760	13,500	18647	224550	83357	12,508	1
1	32350	12,450	19664	225858	45940	15,007	2
1	61140	13,950	15564	288990	35666	15,770	3
1	33590	11,750	15994	266559	45767	15,750	2
1	26750	13,280	16074	186550	93200	11,609	2
1	23140	13,780	15594	288550	84550	11,800	1
1	23780	11,282	16807	26,525	11,000	13,500	2
1	23500	13,420	15000	227360	12,070	12,320	2
1	35160	17,730	15979	23,024	13,500	15,477	2
1	31500	22,660	29907	235377	18,300	13,370	3
1	33900	50,450	26677	233900	18,400	13,799	7
1	53500	45,720	26674	26,550	15,540	13,394	2
1	41790	32,920	15703	25,224	11,040	13,780	2
1	41350	33,790	16679	255530	13,157	13,799	2
1	35729	40,460	15077	286490	13,172	18,742	3
1	61252	62,010	17573	56,559	13,180	14,839	3



astype()

Convert data types of columns

Used to change the data type of a column (e.g., float to int, int, object to datetime).

Example:

```
df['column'] = df['column'].astype(int)
```

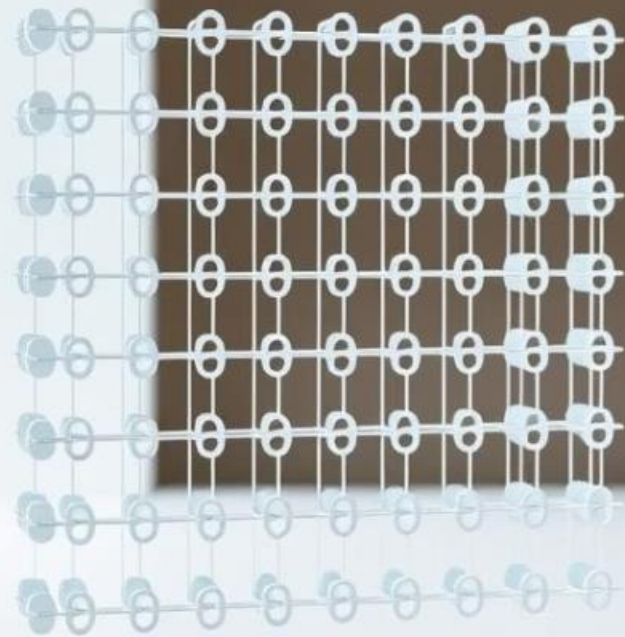
nan_to_num()

Replace NaN with numeric values (NumPy)

Useful for numerical computations where NaN values need to be replaced with 0 or other values.

Example:

```
import numpy as np
data = np.array([1, 2, np.nan])
clean_data = np.nan_to_num(data, nan=0)
```



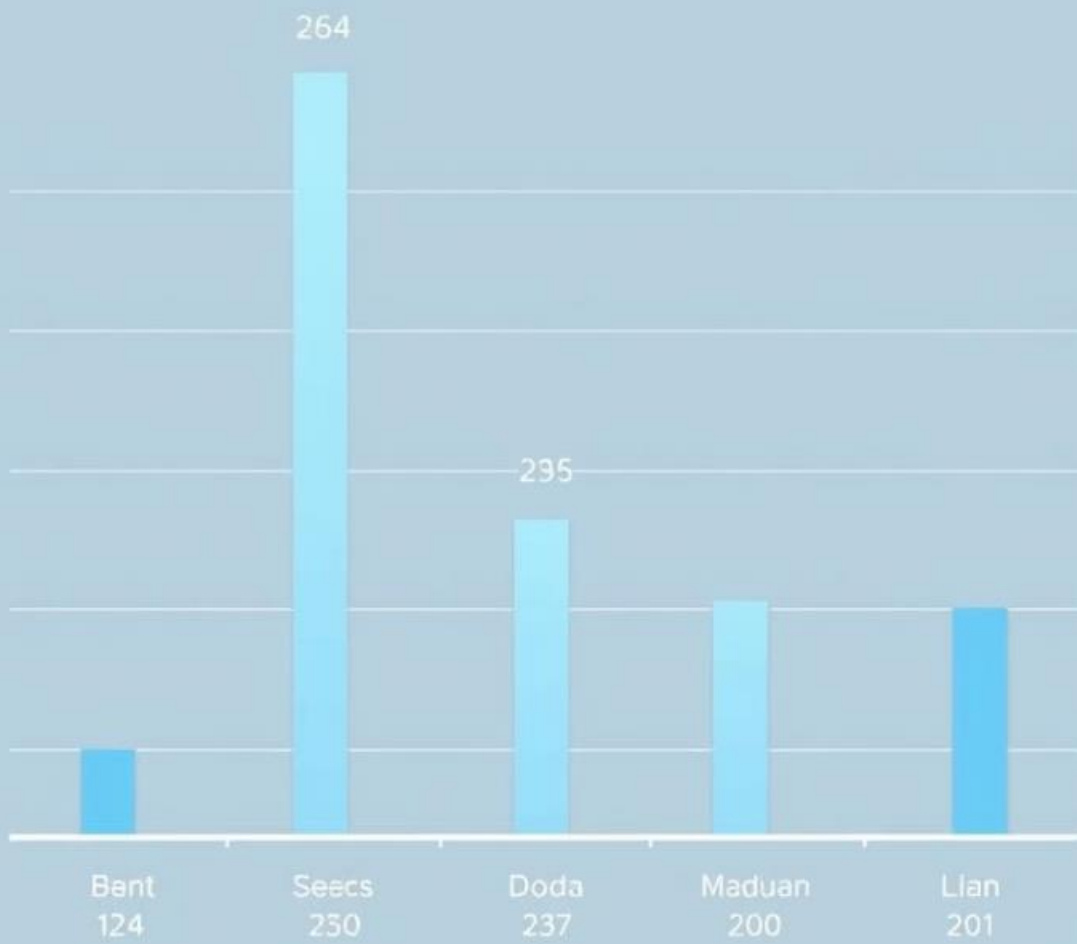
reshape()

1 Reshape arrays (NumPy)

Used to change the shape of arrays without modifying their data.

2 Example:

```
•array = np.array([[1, 2], [3, 4]])  
reshaped = array.reshape((4,))
```



unique()

Get unique values from a column

Returns unique values from a column or array, useful for understanding understanding category spreads.

Example:

```
unique_values = df['column'].unique()
```


Thank You!