

# “How can I keep certain columns in Pandas using examples?”

Authored by  
**stats writer**

July 2, 2024

## RECOMMENDED CITATION

stats writer (2024). “How can I keep certain columns in Pandas using examples?”. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165646>

Pandas is a popular data manipulation library in Python that allows users to work with tabular data. One common task in data analysis is selecting and keeping certain columns from a dataset. This can be easily achieved using the "loc" function in Pandas. For example, if we have a dataframe with columns named "Name", "Age", "Gender", and "Salary", and we want to keep only the "Name" and "Salary" columns, we can use the following code:

```
df.loc]
```

This will select all rows and only the specified columns, resulting in a new dataframe with only the "Name" and "Salary" columns. This functionality is useful for tasks such as data cleaning, feature selection, and creating new datasets for analysis. By using the "loc" function, users can effectively keep certain columns in Pandas and manipulate their data in a more efficient and organized manner.

## Keep Certain Columns in Pandas (With Examples)

**You can use the following methods to only keep certain columns in a pandas DataFrame:**

### Method 1: Specify Columns to Keep

```
#only keep columns 'col1' and 'col2'  
df]
```

### Method 2: Specify Columns to Drop

```
#drop columns 'col3' and 'col4'  
df)]]
```

**The following examples show how to use each method**

with the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'points': ,
'assists': ,
'rebounds': })

#view DataFrame
df

team points assists rebounds
0 A 11 5 11
1 A 7 7 8
2 A 8 7 10
3 B 10 9 6
4 B 13 12 6
5 B 13 9 5
```

Method 1: Specify Columns to Keep

The following code shows how to define a new DataFrame that only keeps the "team" and "points" columns:

```
#create new DataFrame and only keep 'team' and  
'points' columns
```

```
df2 = df]
```

```
#view new DataFrame
```

```
df2
```

```
team points
```

```
0 A 11
```

```
1 A 7
```

```
2 A 8
```

```
3 B 10
```

```
4 B 13
```

```
5 B 13
```

**Notice that the resulting DataFrame only keeps the two columns that we specified.**

**Method 2: Specify Columns to Drop**

**The following code shows how to define a new DataFrame that drops the "assists" and "rebounds" columns from the original DataFrame:**

```
#create new DataFrame and that drops 'assists' and  
'rebounds'
```

```
df2 = df)]]
```

```
#view new DataFrame
```

```
df2
```

```
team points
```

```
0 A 11
```

```
1 A 7
```

```
2 A 8
```

```
3 B 10
```

```
4 B 13
```

```
5 B 13
```

Notice that the resulting DataFrame drops the "assists" and "rebounds" columns from the original DataFrame and keeps the remaining columns.

Additional Resources

The following tutorials explain how to perform other common operations in pandas: