How to calculate summary statistics functions

# Pandas dataframe.aggregate()

Dataframe.aggregate() function is used to apply some aggregation across one or more columns. Aggregate using callable, string, dict or list of string/callables.

The most frequently used aggregations are:

* **sum:** Return the sum of the values for the requested axis
* **min:** Return the minimum of the values for the requested axis
* **max:** Return the maximum of the values for the requested axis

### Syntax

*DataFrame.aggregate(func, axis=0, \*args,)*

**Return Type:** Returns Aggregated DataFrame.

#import pandas

import pandas as pd

#read train xls file

df=pd.read\_csv("tested.csv")

#print first 10 rows

df.head(10)

*output*

| **PassengerId** | **Survived** | **Pclass** | **Name** | **Sex** | **Age** | **SibSp** | **Parch** | **Ticket** | **Fare** | **Cabin** | **Embarked** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 892 | 0 | 3 | Kelly, Mr. James | male | 34.5 | 0 | 0 | 330911 | 7.8292 | NaN | Q |
| 1 | 893 | 1 | 3 | Wilkes, Mrs. James (Ellen Needs) | female | 47.0 | 1 | 0 | 363272 | 7.0000 | NaN | S |
| 2 | 894 | 0 | 2 | Myles, Mr. Thomas Francis | male | 62.0 | 0 | 0 | 240276 | 9.6875 | NaN | Q |
| 3 | 895 | 0 | 3 | Wirz, Mr. Albert | male | 27.0 | 0 | 0 | 315154 | 8.6625 | NaN | S |
| 4 | 896 | 1 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | female | 22.0 | 1 | 1 | 3101298 | 12.2875 | NaN | S |
| 5 | 897 | 0 | 3 | Svensson, Mr. Johan Cervin | male | 14.0 | 0 | 0 | 7538 | 9.2250 | NaN | S |
| 6 | 898 | 1 | 3 | Connolly, Miss. Kate | female | 30.0 | 0 | 0 | 330972 | 7.6292 | NaN | Q |
| 7 | 899 | 0 | 2 | Caldwell, Mr. Albert Francis | male | 26.0 | 1 | 1 | 248738 | 29.0000 | NaN | S |
| 8 | 900 | 1 | 3 | Abrahim, Mrs. Joseph (Sophie Halaut Easu) | female | 18.0 | 0 | 0 | 2657 | 7.2292 | NaN | C |
| 9 | 901 | 0 | 3 | Davies, Mr. John Samuel | male | 21.0 | 2 | 0 | A/4 48871 | 24.1500 | NaN | S |

Use df.select\_datatypes() function

With use of this function we can do sum , min, max values from row but here it takes only numerical data

#For each column which are having numeric values, minimum and sum of all values has been found.

print('We can Aggregate data across all numeric columns using built-in functions such as sum and min')

df.select\_dtypes(include='number').aggregate(['sum','min'])

output

| **PassengerId** | **Survived** | **Pclass** | **Age** | **SibSp** | **Parch** | **Fare** |
| --- | --- | --- | --- | --- | --- | --- |
| sum | 460009 | 152 | 947 | 10050.50 | 187 | 164 | 14856.5376 |
| min | 892 | 0 | 1 | 0.17 | 0 | 0 | 0.0000 |

### Aggregating Specific Columns

In Pandas, we can also apply different aggregation functions across different columns. For that, we need to pass a dictionary with key containing the column names and values containing the list of aggregation functions for any specific column.

syntax

df.aggregate({"Number":['sum', 'min'],

"Age":['max', 'min'],

"Weight":['min', 'sum'],

"Salary":['sum']})

Example

df.columns

df.aggregate({'PassengerId':['max'],

             'Ticket':['max'],

             'Fare':['min','max']})# it calculate these functions on each column

Output

| **PassengerId** | **Ticket** | **Fare** |
| --- | --- | --- |
| max | 1309.0 | W.E.P. 5734 | 512.3292 |
| min | NaN | NaN | 0.0000 |

According to this output each column have calculate but where we don’t find min value it shows null

The aggregations that work with .agg() include:

Mean - df.agg('mean')

Median - df.agg('median')

Mode - df.agg('mode')

Sum - df.agg('sum')

Count - df.agg('count')

Max - df.agg('max')

Min - df.agg('min')

Standard Deviation - df.agg('std')

Variance - df.agg('var')

Skewness - df.agg('skew')

Kurtosis - df.agg('kurt')

df.aggregate({'Valuation':['min','max','sum','mean','median']})

output

| **Valuation** |
| --- |
| min | 2.10000 |
| max | 140.00000 |
| sum | 1955.21000 |
| mean | 7.82084 |
| median | 4.02500 |

df=pd.read\_csv("startups.csv")

df

df.dtypes#it shows all column data types

df.shape #it display only number of rows and columns

df.isnull().sum() #it display how many null values available

df['Company'].isin(['Telegram','Canva'])#criteria matches return true value else display false

df['Industry'].notna()#if null value have display false

output

0 True

1 True

2 True

3 True

4 True

...

245 True

246 False

247 True

248 True

249 True

Name: Industry, Length: 250, dtype: bool

df.agg(

    {

        'Valuation':['median','skew']

    }

)

Output

| **Valuation** |
| --- |
| median | 4.025000 |
| skew | 6.554372 |