

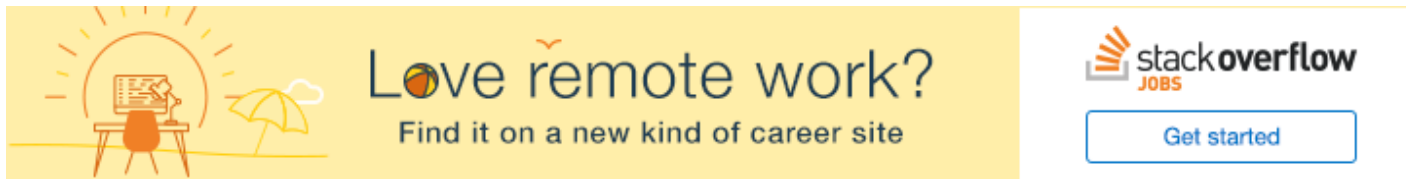
## Announcing Stack Overflow Documentation

We started with Q&A. Technical documentation is next, and we need your help.

Whether you're a beginner or an experienced developer, you *can* contribute.

[Sign up and start helping →](#)[Learn more about Documentation →](#)

## pyspark: Take average of a column after using filter function



I am using the following code to get the average age of people whose salary is greater than some threshold.

```
dataframe.filter(df['salary']>100000).agg({"avg":"age"})
```

the column age is numeric (float) but still I am getting this error.

```
py4j.protocol.Py4JJavaError: An error occurred while calling o86.agg.  
: scala.MatchError: age (of class java.lang.String)
```

Do you know any other way to obtain the avg etc. without using `groupBy` function and `sql` queries

[python](#) [apache-spark](#) [apache-spark-sql](#) [pyspark](#) [pyspark-sql](#)

edited Sep 23 '15 at 16:07

asked Sep 13 '15 at 14:06



zero323

70.7k 16 87 151



iota

669 6 19

## 1 Answer

---

Aggregation function should be a value and a column name a key:

```
dataframe.filter(df['salary']>100000).agg({"age": "avg"})
```

Alternatively you can use `pyspark.sql.functions` :

```
from pyspark.sql.functions import col, avg
dataframe.filter(df['salary']>100000).agg(avg(col("age")))
```

answered Sep 13 '15 at 14:52



zero323

70.7k 16 87 151