

## Responses

What are your thoughts?

There are currently no responses for this story.  
Be the first to respond.

PROGRAMMING, PYTHON

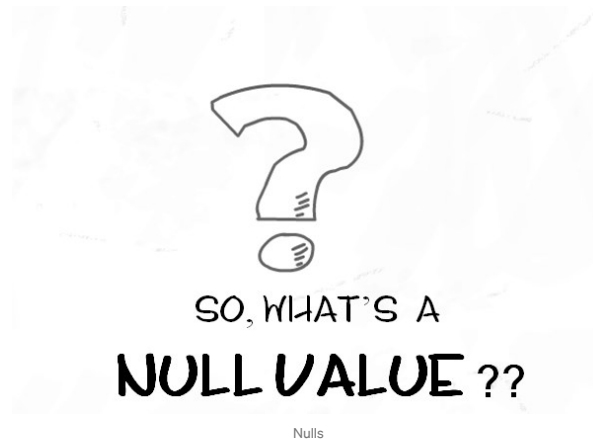
# Handle Missing Data in Pyspark



Vivek Chaudhary [Follow](#)  
Jul 12, 2020 · 3 min read



The objective of this article is to understand various ways to handle missing or null values present in the dataset. A null means an unknown or missing or irrelevant value, but with machine learning or a data science aspect, it becomes essential to deal with nulls efficiently, the reason being an ML engineer can't afford to get short on the dataset.



Nulls

Let's check out various ways to handle missing data or Nulls in Spark Dataframe.

## Pyspark connection and Application creation

```
import pyspark
from pyspark.sql import SparkSession
spark= SparkSession.builder.appName('NULL_Handling').getOrCreate()
print('NULL_Handling')
```

## 2. Import Dataset

```
null_df=spark.read.csv(r'D:\python_coding\pyspark_tutorial\Nulls.csv',header=True,inferSchema=True)
null_df.show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek| 397.0|
| e2| null|  null|
| e3| null|545.68|
| e4| Jack| 448.0|
+---+-----+-----+
```

Dataset

## 3. Dropping Null values

```
#na func to drop rows with null values
#rows having atleast a null value is dropped

null_df.na.drop().show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek|397.0|
| e4| Jack|448.0|
+---+-----+-----+
```

drop nulls

```
#rows having nulls greater than 2 are dropped

null_df.na.drop(thresh=2).show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek| 397.0|
| e3| null|545.68|
| e4| Jack| 448.0|
+---+-----+-----+
```

drop nulls

#### 4. Drop Nulls with 'HOW' argument

```
#drop rows having nulls using how parameter
#records having atleast a null will be dropped

null_df.na.drop(how='any').show()
```

```
+---+-----+-----+
| Id| Name|Sales|
+---+-----+-----+
| e1|Vivek|397.0|
| e4| Jack|448.0|
+---+-----+-----+
```

any

```
#record having all nulls will be dropped

null_df.na.drop(how='all').show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek| 397.0|
| e2| null| null|
| e3| null|545.68|
| e4| Jack| 448.0|
+---+-----+-----+
```

all

#### 5. Drop Nulls basis of a column

```
#dropping null values on basis of a column
null_df.na.drop(subset=['Sales']).show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek| 397.0|
| e3| null|545.68|
| e4| Jack| 448.0|
+---+-----+-----+
```

subset

```
#records having both Name and Sales as Nulls are dropped

null_df.na.drop(how='all',subset=['Name','Sales']).show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek| 397.0|
| e3| null|545.68|
| e4| Jack| 448.0|
+---+-----+-----+
```

```
#records having both Name and Sales as Nulls are dropped

null_df.na.drop(how='any',subset=['Name','Sales']).show()
```

```
+---+-----+-----+
| Id| Name|Sales|
+---+-----+-----+
| e1|Vivek|397.0|
+---+-----+-----+
```

```
| e4| Jack|448.0|
+---+-----+-----+
```

subset any

## 6. Fill the Nulls

```
#filling null values into dataset
#spark automatically detects if a column is string or numeric
null_df.na.fill('NA').show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek| 397.0|
| e2|  NA|  null|
| e3|  NA|545.68|
| e4| Jack| 448.0|
+---+-----+-----+
```

Fill NA

```
#fill integer value column
null_df.na.fill(0).show()
```

```
+---+-----+-----+
| Id| Name| Sales|
+---+-----+-----+
| e1|Vivek| 397.0|
| e2| null|  0.0|
| e3| null|545.68|
| e4| Jack| 448.0|
+---+-----+-----+
```

Fill 0

## 7. Filling Nulls on basis of column

```
#filling on basis of column name

null_df.na.fill('Name Missing',subset=['Name']).show()
```

```
+---+-----+-----+
| Id|      Name| Sales|
+---+-----+-----+
| e1|      Vivek| 397.0|
| e2|Name Missing|  null|
| e3|Name Missing|545.68|
| e4|      Jack| 448.0|
+---+-----+-----+
```

Name Missing

```
#filling multiple column values basis of datatypes

null_df.na.fill({'Name': 'Missing Name', 'Sales': 0}).show()
```

```
+---+-----+-----+
| Id|      Name| Sales|
+---+-----+-----+
| e1|      Vivek| 397.0|
| e2|Missing Name|  0.0|
| e3|Missing Name|545.68|
| e4|      Jack| 448.0|
+---+-----+-----+
```

## 8. Filling null columns with another column value

```
#fill null values in Name column with Id value

from pyspark.sql.functions import when

name_fill_df=null_df.select('ID','Name',
                             when( null_df.Name.isNull(),
                                null_df.Id).otherwise(null_df.Name).alias('Name_Filled'),'Sales')

name_fill_df.show()
```

```
+---+-----+-----+
| ID| Name|Name_Filled|
+---+-----+-----+
| e1|Vivek|      Vivek|
| e2| null|      e2|
| e3| null|      e3|
| e4| Jack|      Jack|
+---+-----+-----+
```

## 9. Filling nulls with mean or average

```
#filling numeric column values with the mean or average value of
that particular column

from pyspark.sql.functions import mean
mean_val=null_df.select(mean(null_df.Sales)).collect()

print(type(mean_val)) #mean_val is a list row object

print('mean value of Sales', mean_val[0][0])
mean_sales=mean_val[0][0]

#now using men_sales value to fill the nulls in sales column
null_df.na.fill(mean_sales,subset=['Sales']).show()
```

```
mean value of Sales 463.55999999999995
+---+-----+
| Id| Name|      Sales|
+---+-----+
| e1|Vivek|      397.0|
| e2| null|463.55999999999995|
| e3| null|      545.68|
| e4| Jack|      448.0|
+---+-----+
```

## Summary:

- Drop null values
- Drop nulls with argument How
- Drop nulls with argument subset
- Fill the null values
- Fill the null column with another column value or with an average value

Hurray, here we have discussed several ways to deal with null values in a Spark data frame.

**Towards AI**  
The Best of Tech,  
Science, and  
Engineering.

Follow

👏 130



### Sign up for Towards AI Newsletter

By Towards AI

Towards AI publishes the best of tech, science, and engineering. Subscribe to receive our updates right in your inbox. Interested in working with us? Please contact us → <https://towardsai.net/contact> [Take a look](#)

✉ Get this newsletter

Emails will be sent to [vladmaksyk@gmail.com](mailto:vladmaksyk@gmail.com).  
[Not you?](#)

Apache Spark

Big Data Analytics

Python

Programming

Pyspark

👏 130 claps



WRITTEN BY

**Vivek Chaudhary**

Data Engineer by profession, Blogger, Python Trainer and DS/ML enthusiast. LinkedIn: [linkedin.com/in/vivek-chaudhary-5378a954](https://www.linkedin.com/in/vivek-chaudhary-5378a954). Twitter: @vc90

Follow



**Towards AI**

Towards AI is the world's leading multidisciplinary science publication. Towards AI publishes the best of tech, science, and engineering. Read by thought-leaders and decision-makers around the world.

Follow

## More From Medium

More from Towards AI **Building a Product Recommendation Engine on Google Cloud's Platform**

anuragbisht in Towards AI  
Dec 29, 2020 · 8 min read

👏 138

More from Towards AI **MIT's Free Online Course to Learn Julia — The Rising Star** More from Towards AI **Time Series Analysis with Python**

Frederik Bussler in Towards AI  
Dec 18, 2020 · 3 min read

👏 613

Amit Chauhan in Towards AI  
Dec 30, 2020 · 4 min read

👏 114

### Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. [Learn more](#)

### Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox. [Explore](#)

### Share your thinking.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. [Write on Medium](#)

---

[About](#)[Help](#)[Legal](#)