learn-co-curriculum / dsc-more-on-missing-data-lab Public

View license

0 stars 192 forks

Star Watch

Code Issues Pull requests Actions Projects Security Insights

solution

This branch is 4 commits ahead, 5 commits behind master.

Contribute

LoreDirick Merge pull request #1 from learn-co-curriculum/jeffs-bran... ... on Mar 25, 2020 5

View code

README.md

More on Missing Data - Lab

Introduction

In this lab, you'll continue to practice techniques for dealing with missing data. Moreover, you'll observe the impact on distributions of your data produced by various techniques for dealing with missing data.

Objectives

In this lab you will:

- Evaluate and execute the best strategy for dealing with missing, duplicate, and erroneous values for a given dataset
- Determine how the distribution of data is affected by imputing values

Load the data

To start, load the dataset `'titanic.csv'` using pandas.

```
# Your code here
import pandas as pd
df = pd.read_csv('titanic.csv')
df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
0	1.0	0.0	3	Braund, Mr. Owen Harris	male	22.0	1.0	0.
1	2.0	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1.0	0.
2	3.0	1.0	3	Heikkinen, Miss. Laina	female	26.0	0.0	0.
3	4.0	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1.0	0.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
4	5.0	0.0	3	Allen, Mr. William Henry	male	35.0	0.0	0.

Use the `.info()` method to quickly preview which features have missing data

```
# Your code here
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1391 entries, 0 to 1390
Data columns (total 12 columns):
PassengerId    1391 non-null float64
Survived       1391 non-null float64
Pclass         1391 non-null object
Name           1391 non-null object
Sex            1391 non-null object
Age            1209 non-null float64
SibSp          1391 non-null float64
Parch          1391 non-null float64
Ticket         1391 non-null object
Fare           1391 non-null float64
Cabin          602 non-null object
Embarked       1289 non-null object
dtypes: float64(6), object(6)
memory usage: 130.5+ KB
```

Observe previous measures of centrality

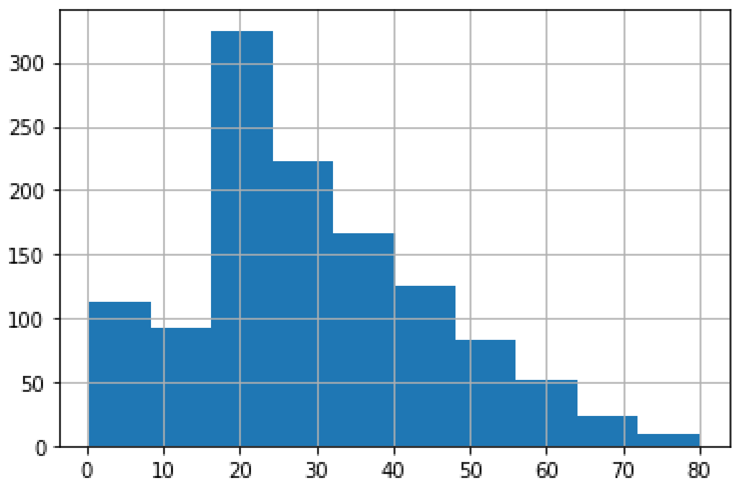
Let's look at the `'Age'` feature. Calculate the mean, median, and standard deviation of this feature. Then plot a histogram of the distribution.

```
# Your code here
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

print(df['Age'].apply(['mean', 'median', 'std']))
df['Age'].hist()
```

```
mean      29.731894
median    27.000000
std       16.070125
Name: Age, dtype: float64
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x11bdacd30>
```



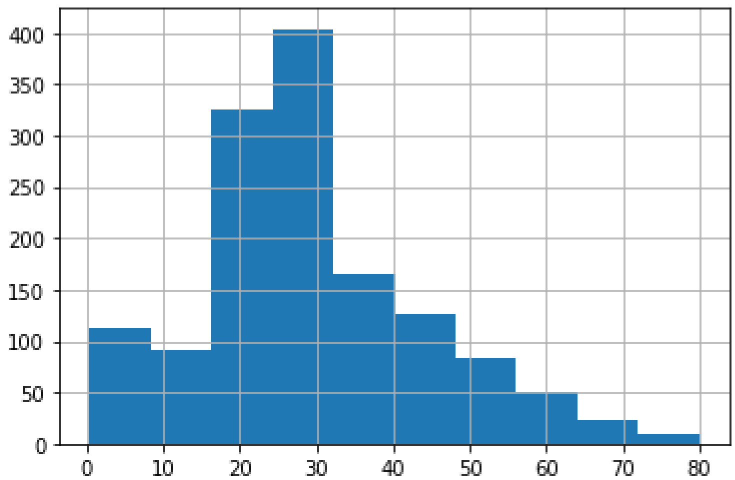
Impute missing values using the mean

Fill the missing `'Age'` values using the average age. (Don't overwrite the original data, as we will be comparing to other methods for dealing with the missing values.) Then recalculate the mean, median, and std and replot the histogram.

```
# Your code here
age_na_mean = df['Age'].fillna(value=df['Age'].mean())
print(age_na_mean.apply(['mean', 'median', 'std']))
age_na_mean.hist()
```

```
mean      29.731894
median    29.731894
std       14.981155
Name: Age, dtype: float64
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x11ec35a90>
```



Commentary

Note that the standard deviation dropped, the median was slightly raised and the distribution has a larger mass near the center.

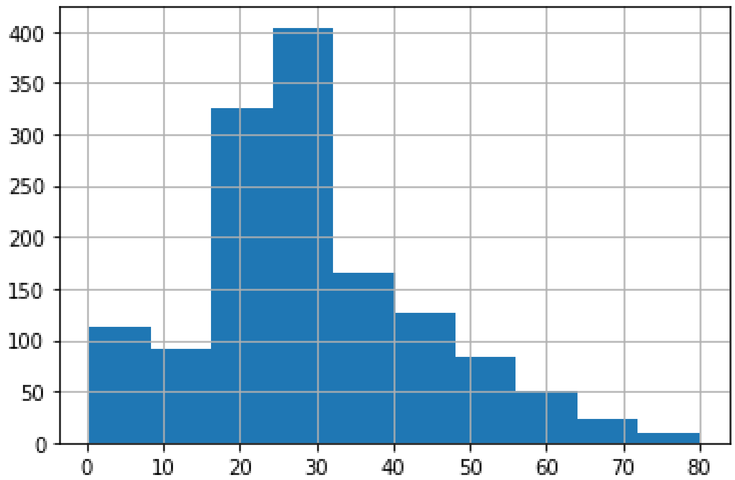
Impute missing values using the median

Fill the missing `'Age'` values, this time using the median age. (Again, don't overwrite the original data, as we will be comparing to other methods for dealing with the missing values.) Then recalculate the mean, median, and std and replot the histogram.

```
# Your code here
age_na_median = df['Age'].fillna(value=df['Age'].median())
print(age_na_median.apply(['mean', 'median', 'std']))
age_na_median.hist()
```

```
mean      29.374450
median    27.000000
std       15.009476
Name: Age, dtype: float64
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x11edc73c8>
```



Commentary

Imputing the median has similar effectiveness to imputing the mean. The variance is reduced, while the mean is slightly lowered. You can once again see that there is a larger mass of data near the center of the distribution.

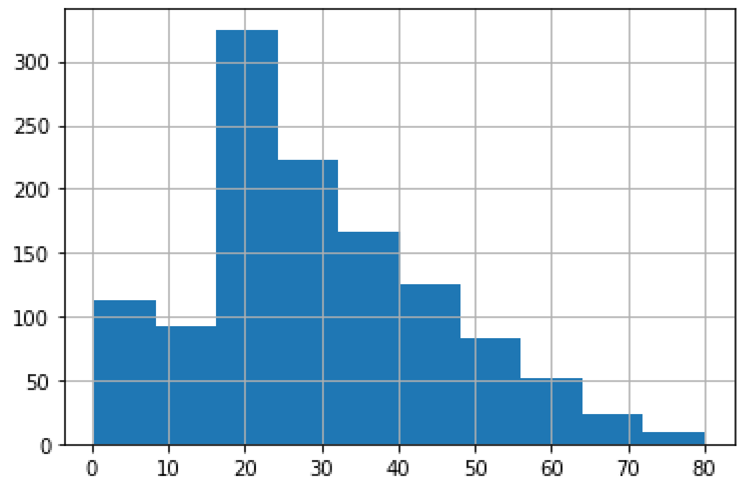
Dropping rows

Finally, let's observe the impact on the distribution if we were to simply drop all of the rows that are missing an age value. Then, calculate the mean, median and standard deviation of the ages along with a histogram, as before.

```
# Your code here
age_na_dropped = df[~df['Age'].isnull()]['Age']
print(age_na_dropped.apply(['mean', 'median', 'std']))
age_na_dropped.hist()
```

```
mean      29.731894
median    27.000000
std       16.070125
Name: Age, dtype: float64
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x11eebe0f0>
```



Commentary

Dropping missing values leaves the distribution and associated measures of centrality unchanged, but at the cost of throwing away data.

Summary

In this lab, you briefly practiced some common techniques for dealing with missing data. Moreover, you observed the impact that these methods had on the distribution of the feature itself. When you begin to tune models on your data, these considerations will be an essential process of developing robust and accurate models.

Releases

No releases published

Packages

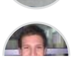
No packages published

Contributors 4

 **LoreDirick** Lore Dirick

 **mathymitchell**

 **sumedh10** Sumedh Panchadhar

 **fpolchow** Forest Polchow

fpolchow Forest Polchow

Languages

- **Jupyter Notebook** 100.0%