**BONUS POINT**

It is a good way to replace null values with the mean of all values as any analysis you perform will be skewed and the results of the analysis weighted in an unpredictable manner.

FINDING THE NULL DATA

It’s essential to find null data in your dataset to avoid getting incorrect results from your analysis. The following code shows how you could obtain a listing of missing values without too much effort.

import pandas as pd

import numpy as np

s = pd.Series([1, 2, 3, np.NaN, 5, 6, None])

print s.isnull()

print

print s[s.isnull()]

Use the isnull() method to detect the missing values. The output shows True when the value is missing. By adding an index into the dataset, you obtain just the entries that are missing.

ENCODING MISSINGNESS

The three possibilities are to ignore the issue, fill in the missing items, or remove (drop) the missing entries from the dataset. Ignoring the problem could lead to all sorts of problems for your analysis, so it’s the option you use least often. The following example shows one technique for filling in missing data or dropping the errant entries from the dataset:

import pandas as pd

import numpy as np

s = pd.Series([1, 2, 3, np.NaN, 5, 6, None])

print s.fillna(int(s.mean()))

print

print s.dropna()

The two methods of interest are fillna(), which fills in the missing entries, and dropna(), which drops the missing entries. When using fillna(), you must provide a value to use for the missing data. This example uses the mean of all the values, but you could choose a number of other approaches.

IMPUTING MISSING DATA

import pandas as pd

import numpy as np

from sklearn.preprocessing import Imputer

s = pd.Series([1, 2, 3, np.NaN, 5, 6, None])

imp = Imputer(missing\_values=‘NaN’,

strategy=‘mean’, axis=0)

imp.fit([1, 2, 3, 4, 5, 6, 7])

x = pd.Series(imp.transform(s).tolist()[0])

print x

The strategy parameter defines how to replace the missing values:

* mean: Replaces the values by using the mean along the axis
* median: Replaces the values by using the medium along the axis
* most\_frequent: Replaces the values by using the most frequent value along the axis