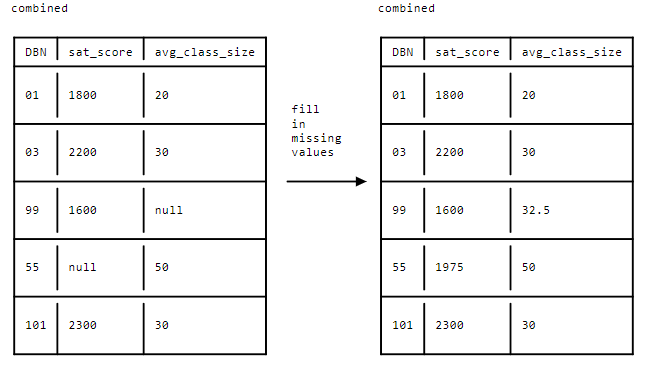
You may have noticed that the inner joins resulted in 116 fewer rows in sat\_results. This is because pandas couldn't find the DBN values that existed in sat\_results in the other data sets. While this is worth investigating, we're currently looking for high-level correlations, so we don't need to dive into which DBNs are missing.

You may also have noticed that we now have many columns with null (NaN) values. This is because we chose to do leftjoins, where some columns may not have had data. The data set also had some missing values to begin with. If we hadn't performed a left join, all of the rows with missing data would have been lost in the merge process, which wouldn't have left us with many high schools in our data set.

There are several ways to handle missing data, and we'll cover them in more detail later on. For now, we'll just fill in the missing values with the overall mean for the column, like so:



In the diagram above, the mean of the first column is (1800 + 1600 + 2200 + 2300) / 4, or 1975, and the mean of the second column is (20 + 30 + 30 + 50) / 4, or 32.5. We replace the missing values with the means of their respective columns, which allows us to proceed with analyses that can't handle missing values (like correlations).

We can fill in missing data in pandas using the [pandas.DataFrame.fillna()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.fillna.html" \t "_blank) method. This method will replace any missing values in a dataframe with the values we specify. We can compute the mean of every column using the [pandas.DataFrame.mean()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.mean.html" \t "_blank) method. If we pass the results of the df.mean() method into the df.fillna() method, pandas will fill in the missing values in each column with the mean of that column.

Here's an example of how we would accomplish this:



means = df.mean()

df = df.fillna(means)

Note that if a column consists entirely of null or NaN values, pandas won't be able to fill in the missing values when we use the df.fillna() method along with the df.mean() method, because there won't be a mean.

We should fill any NaN or null values that remain after the initial replacement with the value 0. We can do this by passing 0 into the df.fillna() method.

Instructions

* Calculate the means of all of the columns in combined using the [pandas.DataFrame.mean()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.mean.html" \t "_blank) method.
* Fill in any missing values in combined with the means of the respective columns using the [pandas.DataFrame.fillna()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.fillna.html" \t "_blank) method.
* Fill in any remaining missing values in combined with 0 using the df.fillna() method.
* Display the first few rows of combined to verify that the correct operations occurred.

# pass the mean of every column in the Nan values

combined = combined.fillna(combined.mean())

# where it is impossible to calculate the mean fillna with zero

combined = combined.fillna(0)

# print 5 first lines

print(combined.head(5))