Aggregates in Pandas

Pandas' Groupby

In a pandas `DataFrame`, aggregate statistic functions can be applied across multiple rows by using a `groupby` function. In the example, the code takes all of the elements that are the same in `Name` and groups them, replacing the values in `Grade` with their mean. Instead of `mean()` any aggregate statistics function, like `median()` or `max()`, can be used. Note that to use the `groupby()` function, at least two columns must be supplied.

```
df = pd.DataFrame([
  ["Amy","Assignment 1",75],
  ["Amy","Assignment 2",35],
  ["Bob","Assignment 1",99],
  ["Bob","Assignment 2",35]
  ], columns=["Name", "Assignment",
"Grade"])

df.groupby('Name').Grade.mean()

# output of the groupby command
|Name | Grade|
| -  | - |
|Amy | 55|
|Bob |  67|
```

Pandas DataFrame Aggregate Function

Pandas' aggregate statistics functions can be used to calculate statistics on a column of a DataFrame. For example, `df.columnName.mean()` computes the mean of the column `columnName` of dataframe `df`. The code block shows how to calculate statistics on the column `columnName` of `df` using Pandas' aggregate statistics functions.

```
df.columnName.mean() # Average of all values in column
df.columnName.std() # Standard deviation of column
df.columnName.median() # Median value of column
df.columnName.max() # Maximum value in column
df.columnName.min() # Minimum value in column
df.columnName.count() # Number of values in column
df.columnName.nunique() # Number of unique values in column
df.columnName.unique() # List of unique values in column
```