**Filling missing data (imputation) by group**

Many statistical and machine learning packages cannot determine the best action to take when missing data entries are encountered. Dealing with missing data is natural in pandas (both in using the default behavior and in defining a custom behavior). In Chapter 1, you practiced using the .dropna() method to drop missing values. Now, you will practice imputing missing values. You can use .groupby() and .transform() to fill missing data appropriately for each group.

Your job is to fill in missing 'age' values for passengers on the Titanic with the median age from their 'gender' and 'pclass'. To do this, you'll group by the 'sex' and 'pclass' columns and transform each group with a custom function to call .fillna() and *impute* the median value.

The DataFrame has been pre-loaded as titanic. Explore it in the IPython Shell by printing the output of titanic.tail(10). Notice in particular the NaNs in the 'age' column.

**INSTRUCTIONS**

* Group titanic by 'sex' and 'pclass'. Save the result as by\_sex\_class.
* Write a function called impute\_median() that fills missing values with the median of a series. This has been done for you.
* Call .transform() with impute\_median on the 'age' column of by\_sex\_class.
* Print the output of titanic.tail(10).

# Create a groupby object: by\_sex\_class

print(titanic.head())

by\_sex\_class = titanic.groupby(['sex', 'pclass'])

# Write a function that imputes median

def impute\_median(series):

return series.fillna(series.median())

# Impute age and assign to titanic['age']

titanic.age = by\_sex\_class['age'].transform(impute\_median)

# Print the output of titanic.tail(10)

print(titanic.tail(10))