**Filtering using NaNs**

In certain scenarios, it may be necessary to remove rows and columns with missing data from a DataFrame. The .dropna()method is used to perform this action. You'll now practice using this method on a dataset obtained from [**Vanderbilt University**](http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.html), which consists of data from passengers on the Titanic.

The DataFrame has been pre-loaded for you as titanic. Explore it in the IPython Shell and you will note that there are many NaNs. You will focus specifically on the 'age' and 'cabin' columns in this exercise. Your job is to use .dropna() to remove rows where *any* of these two columns contains missing data and rows where *all* of these two columns contain missing data.

You'll also use the .shape attribute, which returns the number of rows and columns in a tuple from a DataFrame, or the number of rows from a Series, to see the effect of dropping missing values from a DataFrame.

Finally, you'll use the thresh= keyword argument to drop columns from the full dataset that have more than 1000 missing values.

**INSTRUCTIONS**

* Select the 'age' and 'cabin' columns of titanic and create a new DataFrame df.
* Print the shape of df. This has been done for you.
* Drop rows in df with how='any' and print the shape.
* Drop rows in df with how='all' and print the shape.
* Drop columns from the titanic DataFrame that have more than 1000 missing values by specifying the thresh and axiskeyword arguments. Print the output of .info() from this.

# Select the 'age' and 'cabin' columns: df

df = titanic[['age', 'cabin']]

# Print the shape of df

print(df.shape)

# Drop rows in df with how='any' and print the shape

print(df.dropna(how='any').shape)

# Drop rows in df with how='all' and print the shape

print(df.dropna(how='all').shape)

# Call .dropna() with thresh=1000 and axis='columns' and print the output of .info() from titanic

print(titanic.dropna(thresh=1000, axis='columns').info())