You can now move on to our second question, and explore which domains were submitted most often. We'll want to make a separate script, called domains.py, for this.

Instructions

Here are the steps:

* Make a file called domains.py, using the file browser, or the command line.
* Add in the code to read the file hn\_stories.csv, and add column names.
* You can think of each domain name as a "word". A domain will look like scala-lang.org, or blog.iweb.com.
  + You can use the [value\_counts](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.value_counts.html) method in pandas to count the number of occurrences of each value in a column.
* Print the 100 most submitted domains.

By default, Pandas only prints 10 rows of a Dataframe or Series. There is a pandas option to make it print more rows (see [this](http://stackoverflow.com/questions/19124601/is-there-a-way-to-pretty-print-the-entire-pandas-series-dataframe) thread on Stackoverflow), but there are bugs with it and Series. Instead, just loop through the series and print the index value, and the total. Here's some sample code:

for name, row in domains.items():

print("{0}: {1}".format(name, row))

The above code assumes that the results of running value\_counts is assigned to domains.

You can extend this analysis and make it a bit more robust by removing subdomains. For example, blog.iweb.com and iweb.com would be separate domains at the moment, but they are the same. By removing the subdomain, you can turn blog.iweb.com into iweb.com. You can remove the subdomain using the [Series.apply](http://pandas.pydata.org/pandas-docs/stable/generated/panda.Series.apply.html) and [Dataframe.apply](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.apply.html) methods in pandas.