Lab Exercise: Retail Assets Analysis

Overview

The objective of this lab is to give you some more experience in using a Jupyter notebook to:

- Load in a csv file into a pandas DataFrame
- Display some data contained in the DataFrame
- Create some filters to derive more selective information from the DataFrame
- Plot a bar chart of some filtered data
- Save the bar chart to a file

One of the objects of this exercise is to try to read, and understand, an existing example in order to write something similar. With that in mind, while doing this exercise you could refer to the "03 - Pandas First Steps.ipynb" notebook, as it has some similar code to what's required here.

Create a new notebook

Import the libraries

- pandas
- matplotlib.pyplot

Load csv file

Load the csv file into a DataFrame from:

https://neueda.conygre.com/pydata/Retail_Asset_List.csv

Set the 'Asset ID' column as the index. Familiarize yourself with the data it contains.

Display the first 5 and last 5 rows of data

Describe the DataFrame.

Transpose the Describe method.

Print selected information

Print the number of distinct Brands, Locations, Provinces, Asset Types contained in the DataFrame?

What are the names of the distinct Brands and Asset Types?

(Hint: look at the unique() and nunique() functions for a DataFrame)

Using Filters

How many stores in China?

How many Factories have more than 500 workers?

List all Asset Types, Cities and Brands for Assets in United States, Vietnam and China.

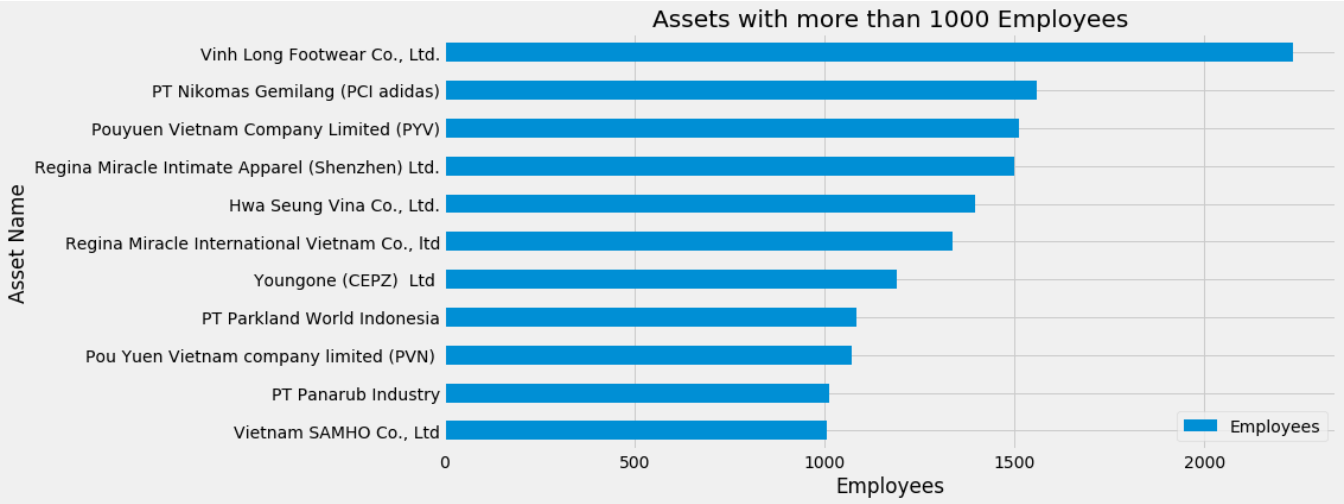
hint: look at the pandas 'isin()' function e.g.

```
df['Location'].isin(['Vietnam', 'China'])
```

Produce a Simple Bar Chart

Produce a bar chart similar to the one below showing the names of all the assets that have more than 1,000 employees

Save the plot as a png file.



Optional – Merging DataFrames

This section of the Lab is optional, as it uses some more advanced pandas functions.

Load into a new DataFrame the csv file:

https://neueda.conygre.com/pydata/Retail_Asset_Sales.csv

Familiarize yourself with the data it contains.

Note how there are a number of rows for each Asset ID in this DataFrame

We will now ‘merge’ or ‘join’ the Retail_Asset_List DataFrame into this new DataFrame. Each row in the resulting DataFrame will now have BOTH the Financial Sales information AND the corresponding Asset information. This is similar to a SQL join.

```
df_sales = df_sales.merge( df_assets, on='Asset ID',
                           right_index=True )
```

Check the help for the merge function to understand the above line.

Save the resulting DataFrame to a csv file with the to_csv() function of a DataFrame.