
# Advanced Certificate in Machine Learning

## Lesson 6: Python for Data Analysis

# Logistics

- Blog Submissions – Pending
- Using Python for Data Analysis
- Create GitHub Account
- Start thinking about the second blog
- Start thinking about the project: Collect original and current data (highest value), try basic experimentation to finalize topic (do not leave it for later) – you know enough to begin!
- Data Collection Routines, Scraping, etc. cannot be done in a hurry.
- Declare area or areas (two max) of interest by next Saturday.

# Explore the Occupations dataset further: Using pandas operations & Visualization

# Merge

## Explore mpg datasets

Import the necessary libraries

```
import pandas as pd
import numpy as np
```

Import the first dataset cars1 and cars2.

Assign each to a to a variable called cars1 and cars2

```
cars1 = pd.read_csv("https://raw.githubusercontent.com/guipsamora/
pandas_exercises/master/05_Merge/Auto_MPG/cars1.csv")

cars2 = pd.read_csv("https://raw.githubusercontent.com/guipsamora/
pandas_exercises/master/05_Merge/Auto_MPG/cars2.csv")

print(cars1.head())
print(cars2.head())
```

It seems our first dataset has some unnamed blank columns, fix cars1

```
cars1 = cars1.loc[:, "mpg":"car"]
cars1.head()
```

`cars1 = cars1.loc[:, "mpg":"car”]: Purely label-location based indexer for selection by label`

What is the number of observations in each dataset?

```
print(cars1.shape)

print(cars2.shape)
```

Combine rows of cars1 and cars2 into a single DataFrame called cars

```
cars = cars1.append(cars2)

cars
```

Add the column owners to cars

```
nr_owners = np.random.randint(15000, high=73001, size=398, dtype='l')

nr_owners
```

Add the column owners to cars

```
cars['owners'] = nr_owners

cars.tail()
```

# Explore the mpg dataset further: Using pandas operations & Visualization

## Explore Housing Market Dataset

Introduction:

This time we will create our own dataset with fictional numbers to describe a house market. As we are going to create random data don't try to reason of the numbers.

Import the necessary libraries

```
import pandas as pd

import numpy as np
```

Create 3 differents Series, each of length 100, as follows:

The first a random number from 1 to 4
The second a random number from 1 to 3
The third a random number from 10,000 to 30,000

```
s1 = pd.Series(np.random.randint(1, high=5, size=100, dtype='l'))
s2 = pd.Series(np.random.randint(1, high=4, size=100, dtype='l'))
s3 = pd.Series(np.random.randint(10000, high=30001, size=100, dtype='l'))

print(s1, s2, s3)
```

Exercise:

1. Let's create a DataFrame by joining the Series by column
2. Change the name of the columns to bedrs, bathrs, price_sqr_meter
3. Create a one column DataFrame with the values of the 3 Series and assign it to 'bigcolumn'
4. Resolve any issues in the final DataFrame.

## Let's create a DataFrame by joining the Series by column

```
housemkt = pd.concat([s1, s2, s3], axis=1)
housemkt.head()
```

```
type(s1): Series
```

concat([s1, s2, s3], axis=1): pandas provides various facilities for easily combining together Series, DataFrame, and Panel objects with various kinds of set logic for the indexes and relational algebra functionality in the case of join / merge-type operations.

(Deprecated) Panel is a somewhat less-used, but still important container for 3-dimensional data. The term panel data is derived from econometrics and is partially responsible for the name pandas: pan(el)-da(ta)-s. The names for the 3 axes are intended to give some semantic meaning to describing operations involving panel data and, in particular, econometric analysis of panel data.

Change the name of the columns to bedrs, bathrs, price_sqr_meter

```
housemkt.rename(columns = {0: 'bedrs', 1: 'bathrs', 2: 'price_sqr_meter'},
inplace=True)

housemkt.head()
```

Create a one column DataFrame with the values of the 3 Series and assign it to 'bigcolumn'

```
# join concat the values
bigcolumn = pd.concat([s1, s2, s3], axis=0)

# it is still a Series, so we need to transform it to a DataFrame
bigcolumn = bigcolumn.to_frame()
print type(bigcolumn)

bigcolumn

to_frame: Convert Series to DataFrame
```

Ops it seems it is going only until index 99. Is it true?

```
# no the index are kept but the length of the DataFrame is 300
len(bigcolumn)
```

Reindex the DataFrame so it goes from 0 to 299

```
bigcolumn.reset_index(drop=True, inplace=True)

bigcolumn
```

reset_index: When we reset the index, the old index is added as a column, and a new sequential index is used.

inplace: Modify the DataFrame in place (do not create a new object)

drop: Do not try to insert index into dataframe columns. This resets the index to the default integer index.

# Explore the Housing Market dataset further: Using pandas operations & Visualization

## Explore US - Baby Names Dataset

Introduction:

We are going to use a subset of US Baby Names from Kaggle.
In the file it will be names from 2004 until 2014

Import the necessary libraries

```
import pandas as pd
```

Import the dataset from this address.

Assign it to a variable called baby_names.

```
baby_names = pd.read_csv('file:///Users/aurobindosarkar/Downloads/
babynames.csv')

baby_names.info()
```

Exercise:

1. See the first 10 entries
2. Delete the column 'Unnamed: 0' and 'Id'
3. Is there more male or female names in the dataset?
4. Group the dataset by name and assign to names
    1. Delete the Year column – Why?
    2. print the first 5 observations
    3. print the size of the dataset
    4. print dimensions of the dataset
    5. sort it from the biggest value to the smallest one
5. How many different names exist in the dataset?
6. Are the names in 6 already unique?
7. What is the name with most occurrences?
8. How many different names have the least occurrences?
9. What is the standard deviation of names?
10. Get a summary with the mean, min, max, std and quartiles.

See the first 10 entries

```
baby_names.head(10)
```

Delete the column 'Unnamed: 0' and 'Id'

```
# deletes Unnamed: 0
del baby_names['Unnamed: 0']

# deletes Unnamed: 0
del baby_names['Id']

baby_names.head()
```

Is there more male or female names in the dataset?

```
baby_names['Gender'].value_counts('F')
```

```
type(baby_names['Gender']): Series
```

```
baby_names['Gender'].value_counts('F'): Returns object containing counts of unique values. The resulting object will be in descending order so that the first element is the most frequently-occurring element. Excludes NA values by default.
```

## Group the dataset by name and assign to names

```
# you don't want to sum the Year column, so you delete it
del baby_names["Year"]

# group the data
names = baby_names.groupby("Name").sum()

# print the first 5 observations
names.head()

# print the size of the dataset
print(names.shape)

# sort it from the biggest value to the smallest one
names.sort_values("Count", ascending = 0).head()
```

## How many different names exist in the dataset?

```
# as we have already grouped by the name, all the names are unique already.
# get the length of names
len(names)
```

What is the name with most occurrences?

```
names.Count.idxmax()

# OR

# names[names.Count == names.Count.max()]
```

type(names): DataFrame

type(names.Count): Series

```
names.Count.idxmax(): Index label of the first occurrence of maximum of values.

# OR

# names[names.Count == names.Count.max()]: Below

Count: Return Series with number of non-NA/null observations over requested
axis.

Max: This method returns the maximum of the values in the object.
```

How many different names have the least occurrences?

```
len(names[names.Count == names.Count.min()])
```

How many different names have the least occurrences?

```
len(names[names.Count == names.Count.min()])
```

What is the standard deviation of names?

```
names.Count.std()
```

What is the standard deviation of names?

```
names.Count.std()
```

Get a summary with the mean, min, max, std and quartiles.

```
names.describe()
```

# Explore the US Baby Names dataset further: Using pandas operations & Visualization

# Exercise: Select a Dataset from the UCI site and explore it using pandas operations & Visualization

# Homework: Explore one dataset per day using pandas operations & Visualization – Form your methods & approach, Get fluent, Research on the net for Exploratory Data Analysis

# Hard Stop