Python for Data Mining Part 2

Data mining is the extraction of hidden patterns and insights from large datasets. It involves using statistical and machine learning techniques to discover trends, predict outcomes, and support decision-making across various industries by transforming raw data into valuable knowledge. Several programming languages and applications are in use for the statistical and machine learning process. One such programming language is Python. Python is a highly ranked language for data mining. It has extensive libraries such as Pandas, Numpy, Matplotlib, etc. for cleaning, analyzing, modelling, and visualizing data, as well as for presenting actionable insights from the data.

Python as a Programming Language

Python is a high-level general purpose programming language with many dominant areas of applications including data analysis, data mining, and artificial intelligence. Python Enhancement Proposal 8 (PEP 8) provides a guidance on how to write Python code to enhance readability and maintain consistency across projects https://peps.python.org/pep-0008/

Common modules and their usage

NumPy: Provides an n-dimensional array (ndarray) data structure and a wide range of mathematical operations for numerical computations.

matplotlib.pyplot: Offers high-quality data visualization capabilities for creating various types of plots and customizing their appearance.

pandas: Introduces Series and DataFrame data structures, which facilitate data manipulation operations and provide methods for data analysis and visualization.

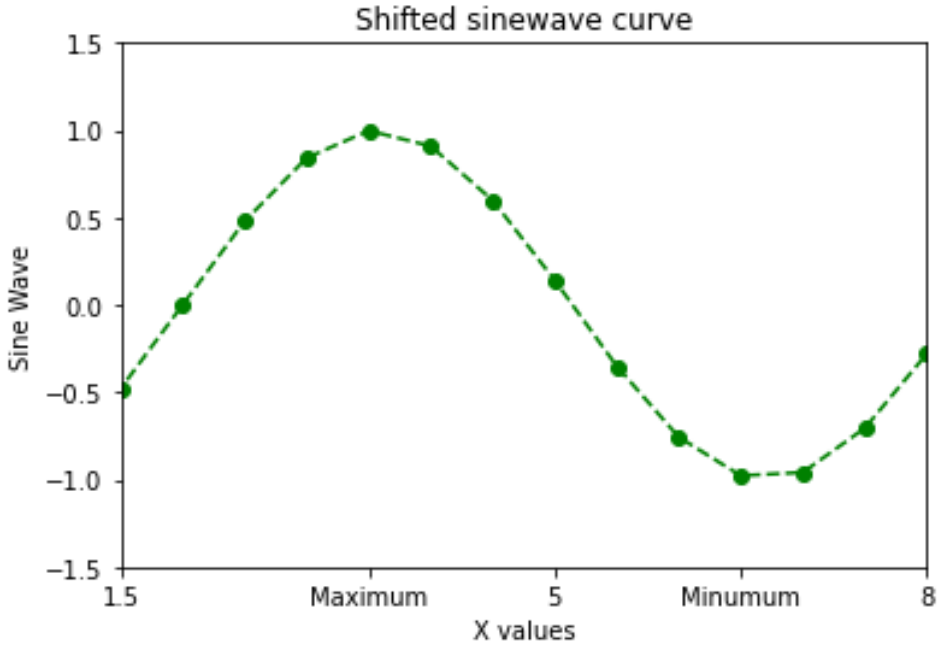
Pandas Data Structures

Series: Series: A 1-dimensional homogeneous array with an immutable size.
Data Frames: A general 2-dimensional labelled, size mutable tabular structure with potentially heterogeneously typed columns.
Panel: General 3-dimensional labelled size mutable array.

Examples of plotting graphs with Matplotlib.pyplot

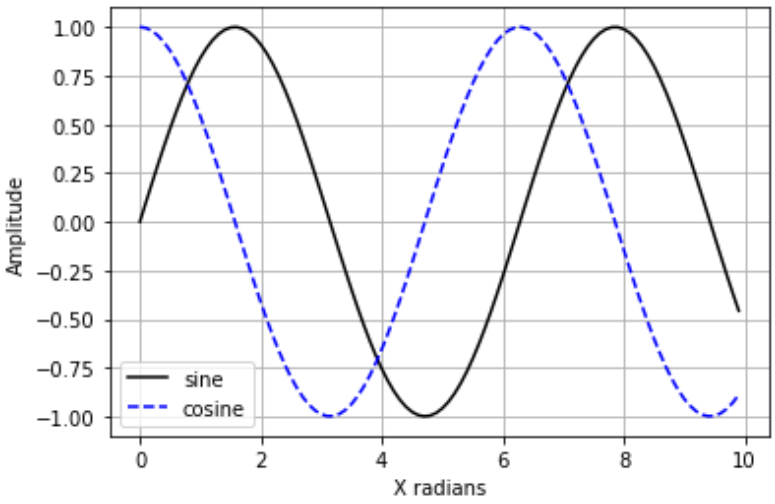
#Example 1: Shifted Sinewave curve (One plot, axes tick labels)

```
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
x = np.arange(0,10,0.5)
y = np.sin(x-2)
ax.plot(x,y, 'go--')
print( ax.get_xlim() ) # (-0.47500000000000003, 9.975)
ax.set_xlim(1.5,8)
ax.set_xticks([1.5,3.5,5,6.5,8])
ax.set_xticklabels([1.5, "Maximum", "5", "Minimum", 8])
ax.set_ylim(-1.5,1.5)
ax.set_xlabel("X values")
ax.set_ylabel("Sine Wave")
ax.set_title("Shifted sinewave curve")
```



#Example 2: Sine and Cosine Curve (Two plots with legend)

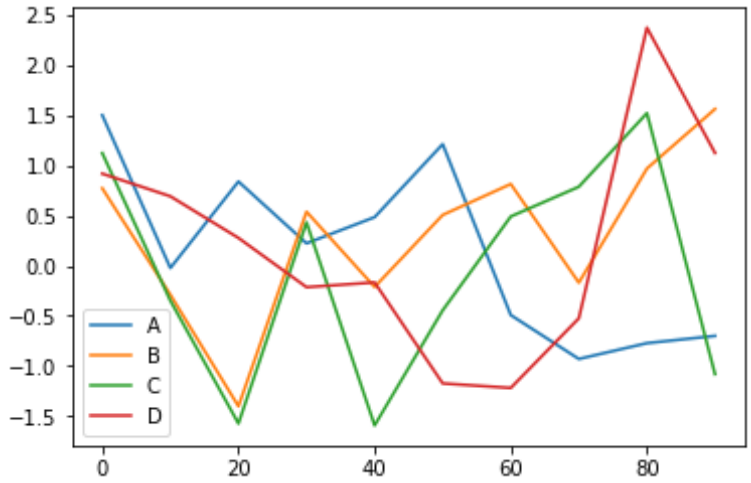
```
fig = plt.figure();ax = fig.add_subplot(1,1,1)
x = np.arange(0,10,0.1) ; y1 = np.sin(x); y2 = np.cos(x)
ax.plot(x,y1, 'k', label="sine")
ax.plot(x,y2, 'b--', label="cosine")
ax.grid()
ax.legend()
ax.set_xlabel("X radians"); ax.set_ylabel("Amplitude")
```



Examples of plotting graphs with pandas

#Example 3: Line plot (Four plots with legend)

```
df = pd.DataFrame(np.random.randn(10,4),columns=['A', 'B', 'C', 'D'],
        index=np.arange(0,100,10))
#Generate 10x4 random matrix (4 columns each consists of 10 row points)
#and label them
df.plot()
```



#Example 4: Bar plot (Two plots using subplot, transparent colour)

```
s = pd.Series(np.random.rand(10),index=list('ABCDEFGHIJ'))
#Series
fig = plt.figure()
ax1 = fig.add_subplot(2,1,1)
ax2 = fig.add_subplot(2,1,2)
s.plot.bar(ax=ax1, color='k', alpha=0.6) # pass axes and any desired options,
#e.g., alpha is the transparency of the color
s.plot.barh(ax=ax2, color='k', alpha=0.2)
```