

将数据转化成时间序列数据

- `import numpy as np`
- `import pandas as pd`
- `import matplotlib.pyplot as plt`
- `'''` 功能作用：将数据转化成时间序列数据
- 数据文件 `:data_timeseries.txt` 数据结构
- 包含4列的文本文件，其中第一列表示年，第二列表示月，第三列和第四列表示数据
- 时间序列数据分析常用于金融、信号处理、天气预测、轨道预测、地震预测或者任意需要处理时间数据的场合。在时间序列和顺序数据分析中构建的模型应该考虑数据的顺序，并提取相互之间的关系
- 步骤：
 - 1.加载输入文件
 - 2.提取时间的起始日期 和 结束日期
 - 其中结束日期需要处理,往后推一个月
 - 3.创建以月为间隔的日期变量 `M`
 - 4.然后把月为间隔的日期变量转化成时间序列,返回 `time_dataserries`
 - 提取 年-月 日期,其中涉及到`pd.series`,为数据制定索引`index`,索引就是 `M`
 - 5.使用`time_dataserries.plot` 画图即可
 - 为什么可以这样呢,其中`time_dataserries`就包括了变量和自变量,因为把索引当做自变量,所以时间月间隔就是X轴 `'''`
- `def convert_data_to_timeseries(input_file, column, verbose=False):`
 - `'''` 作用：定义一个函数读取文件,该文件将序列观察结果转换为时间序列数据
 - `:param input_file:` 输入的文件
 - `:param column:` 提取的列数
 - `:param verbose:` 是否打印输出
 - `:return:` `'''`
- `# 加载输入文件`
- `data = np.loadtxt(input_file, delimiter=',')`
- `# 提取 起始日期 和 终止日期`
- `start_date = str(int(data[0,0])) + '-' + str(int(data[0,1]))`
- `end_date = str(int(data[-1,0] + 1)) + '-' + str(int(data[-1,1] % 12 + 1))`
- `if verbose:`
 - `print ("\nStart date =", start_date)`
 - `print ("End date =", end_date)`
- `# 创建以月为间隔的变量`
- `dates = pd.date_range(start_date, end_date, freq='M')`
- `# 将给定的列转换为时间序列数据。可以用年和月访问这些数据（而不是索引）`
- `# 将日期转化成时间序列 data_timeseries = pd.Series(data[:,column], index=dates)`

- # Series 是一个类数组的数据结构，同时带有标签 (lable) 或者说索引 (index) ,不设置 index 就默认索引|0-(N-1)
- if verbose:
 - print ("\nTime series data:\n", data_timeseries[:10])
- #返回时间索引变量
- return data_timeseries
- if __name__ == '__main__':
 - # 输入数据文件
 - input_file = 'data_timeseries.txt'
 - # 加载输入数据
 - column_num = 2
 - data_timeseries = convert_data_to_timeseries(input_file, column_num)
 - # 画出数据序列数据
 - data_timeseries.plot()
 - plt.title('Input data')
 - plt.show()