实验项目四：

( 一）实验目的

了解Python基本编程语法，掌握Python进行数据载入、预处理、分析和可视化的方法。

（二）实验分析

## 1.在Python中导入数据

在Python中导入数据，CSV文件是由由逗号分割字段构成的数据记录型文件。我们可以方便地把EXCEL中的电子表格存储为CSV文件。

## 数据提取和筛选

我们希望只看到数据的部分行，部分行和部分列的情况时，应调用函数。

## 3.数据预处理

（1）数据转置

有的时候，我们希望交换数据的行和列，这称为“转置”。

（1）缺失值处理

由于各种各样的原因，原始数据中会有一些数据是缺失的，但我们希望以尽可能准确的方式将缺失的数据删除或补全回来。一般来说，我们可以利用缺失数据附近的数据来预测和推断缺失的值，这在数据上称为回归和拟合。而Pandas能够自动地帮我们完成这个操作。

（三）实验项目的实施

## 1.在Python中导入数据

读取CSV文件，CSV文件是由由逗号分割字段构成的数据记录型文件。我们可以方便地把EXCEL中的电子表格存储为CSV文件。例如，我们有一份CSV格式的数据（例子数据是英国近些年的降雨量统计数据，可以从以下网址找到：https://data.gov.uk/dataset/average-temperature-and-rainfall-england-and-wales/resource/3fea0f7b-5304-4f11-a809-159f4558e7da），也可扫描右侧二维码进行下载。从EXCEL中看到的数据如下图2-53所示：

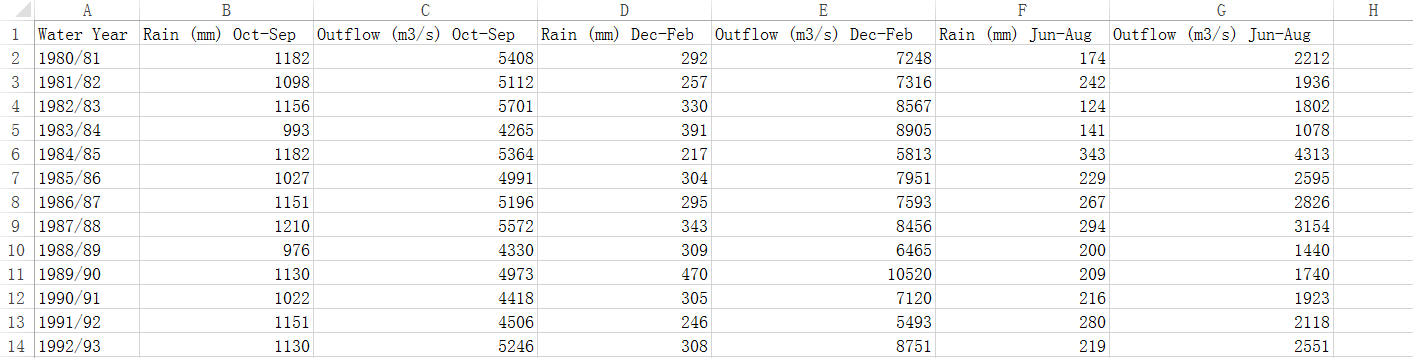


图2-53 读取CSV文件

如果这个文件被保存在以下位置：

D:\data\uk\_rain\_2014.csv

我们可以在Python中利用Pandas库将它导入：

>>>import pandas as pd

>>>df = pd.read\_csv('d:\\data\\uk\_rain\_2014.csv', header=0)

这里需要注意的是，因为windows下用于分割目录的“\”符号在Python中被用于转义符（转义符就是用来输入特殊符号的引导符号，例如\n是回车,\r是换行等），因此“\”本身在Python语言中需要通过“\\”来输入。

以上两行程序就将这个csv文件导入成pandas中的一种类型为Dataframe的对象中，并给这个对象起名为df。

## 2.数据提取和筛选

仍然针对上面导入的英国天气数据，由于数据有很多行，我们希望只看到数据的前5行：

>>> df.head(5)

由于列也很多（导致无法在屏幕宽度的一行里显示完而有了拆行），则我们可以仅显示前4行和前3列：

>>> df1= df.iloc[0:4, 0:3]

>>> df1

Water Year Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep

0 1980/81 1182 5408

1 1981/82 1098 5112

2 1982/83 1156 5701

3 1983/84 993 4265

这里，df对象的head文件用于选择df的前n行进行切片。而iloc用于根据指定的行和列的范围对df进行切片。0:4表示从第0行到第3行（即前4行，不含第4行），0:3表示从第0列到第2列（即前3列，不含第3列）。在计算机语言中，第一个索引常常是0而不是1，这一点与我们日常生活中的习惯不同，需要注意。

## 3.数据预处理

（1）数据转置

为了方便演示，我们把刚刚的df对象中的数据进行切片，并放在一个新的对象中：

我们对df1进行转置：

>>> df1.T

这里用于的T方法，与我们在线性代数中学到的转置的表示方式完全一致。

（2）缺失值处理

为了学习如何补全缺失值，我们需要找到一些有缺失值的数据。简单起见，我们故意把刚才导入的数据“弄丢”一个点。例如，对于上面的df1对象，我们将其中第2行第1列的1156设置为缺失值：

>>> import numpy as np

>>> df1.iloc[2,1] = np.nan

在Python中，缺失值由numpy中的nan来表示。因此为了表示“缺失值”的概念，我们需要先引入numpy包（并起别名为np）。如果之前已经导入numpy，则这一句可以省略。从上面的输出可以看出，我们已经把df1的第2行第1列的值设置为NaN。

数据清洗就是处理缺失数据以及清除无意义的信息。

>>> df1. dropna()

然后，重新设置df1的第2行第1列的值设置为NaN，我们利用Pandas的补全缺失值功能进行补全：

>>> df1.iloc[2,1] = np.nan

>>> df1.interpolate()

1. 实验项目的结果分析

## 1.在Python中导入数据

为了验证我们确实导入了这个数据文件，我们可以把df的内容打印出来：

>>>print（df）

Water Year Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep Rain (mm) Dec-Feb \

0 1980/81 1182 5408 292

1 1981/82 1098 5112 257

2 1982/83 1156 5701 330

3 1983/84 993 4265 391

4 1984/85 1182 5364 217

5 1985/86 1027 4991 304

6 1986/87 1151 5196 295

7 1987/88 1210 5572 343

8 1988/89 976 4330 309

9 1989/90 1130 4973 470

10 1990/91 1022 4418 305

11 1991/92 1151 4506 246

1. 1992/93 1130 5246 308

## 2.数据提取和筛选

（1）显示前5行

Water Year Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep Rain (mm) Dec-Feb \

0 1980/81 1182 5408 292

1 1981/82 1098 5112 257

2 1982/83 1156 5701 330

3 1983/84 993 4265 391

4 1984/85 1182 5364 217

Outflow (m3/s) Dec-Feb Rain (mm) Jun-Aug Outflow (m3/s) Jun-Aug

0 7248 174 2212

1 7316 242 1936

2 8567 124 1802

3 8905 141 1078

4 5813 343 4313

（2）仅显示前4行和前3列

Water Year Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep

0 1980/81 1182.0 5408

1 1981/82 1098.0 5112

2 1982/83 NaN 5701

3 1983/84 993.0 4265

## 3.数据预处理

（1）数据转置

0 1 2 3

Water Year 1980/81 1981/82 1982/83 1983/84

Rain (mm) Oct-Sep 1182 1098 1156 993

Outflow (m3/s) Oct-Sep 5408 5112 5701 4265

（2）缺失值处理

设置df1的第2行第1列的值设置为NaN

Water Year Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep

0 1980/81 1182.0 5408

1 1981/82 1098.0 5112

2 1982/83 NaN 5701

3 1983/84 993.0 4265

删除无效行

Water Year Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep

0 1980/81 1182.0 5408

1 1981/82 1098.0 5112

3 1983/84 993.0 4265

我们利用Pandas的补全缺失值功能进行补全：

Water Year Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep

0 1980/81 1182.0 5408

1 1981/82 1098.0 5112

2 1982/83 1045.5 5701

3 1983/84 993.0 4265

可以看到，Dataframe对象的interpolate方法自动通过线性插值的方法，估算出被我们‘弄丢’的值为1045.5。

学生完成后依次向教师演示，教师根据程序结果以10分为满分进行打分。学生上交实验报告，实验报告满分10分，教师针对实验报告打分并给出最终成绩。

实验报告四

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 班级 |  | | 姓名 |  | 学号 |  |
| 日期 |  | | 教室 |  | 成绩 |  |
| 实验名称 | |  | | | | |
| 实验目的 | |  | | | | |
| 实  验  内  容 | | 完成题目要求,将代码与显示结果截图展示如下   1. 题目：在Python中导入指定数据，在Python中打印显示数组 2. 数据提取和筛选   将数据的前5行；前4行和前3列两组数组打印出来   1. 数据预处理   （1）数据转置，对数据的前4行和前3列，交换数据的行和列，将转置后的数据打印出来。  （2）缺失值处理，将其中第2行第1列的1156设置为缺失值，并完成删除无效行和补全缺失值功能进行补全操作，将列表打印如下 | | | | |
| 实验总结 | |  | | | | |

实验项目五：

( 一）实验目的

了解Python基本编程语法，掌握Python进行数据分析和可视化的方法。

（二）实验分析

## 1.数据统计分析

对数据进行分析，获得数据的统计特征。

## 2.Python中的数据可视化：matplotlib

在数据分析中，可视化是一种有力的工具，好的数据可视化方式能够让我们方便直观地洞察数据的分布和趋势，也易于得出相应的分析结论。

（三）实验项目的实施

## 1.数据统计分析

对数据进行分析的第一步往往是对数据进行统计，获得数据的统计特征。在Pandas，达到以上目的最便捷的方法，就是采用describe方法。具体示例如下：

>>> df.describe()

Rain (mm) Oct-Sep Outflow (m3/s) Oct-Sep Rain (mm) Dec-Feb \

count 33.000000 33.000000 33.000000

mean 1129.000000 5019.181818 325.363636

std 101.900074 658.587762 69.995008

min 856.000000 3479.000000 206.000000

25% 1053.000000 4506.000000 268.000000

50% 1139.000000 5112.000000 309.000000

75% 1182.000000 5497.000000 360.000000

max 1387.000000 6391.000000 484.000000

Outflow (m3/s) Dec-Feb Rain (mm) Jun-Aug Outflow (m3/s) Jun-Aug

count 33.000000 33.000000 33.000000

mean 7926.545455 237.484848 2439.757576

std 1692.800049 66.167931 1025.914106

min 4578.000000 103.000000 1078.000000

25% 6690.000000 193.000000 1797.000000

50% 7630.000000 229.000000 2142.000000

75% 8905.000000 280.000000 2959.000000

max 11486.000000 379.000000 5261.000000

可以看出，describe方法方便地计算中了每一列数据的个数（因为可能有缺失值的存在，因此每一列数据的个数量可能不相同）、平均值、标准差、最小值、最大值，以及25%、50%、75%分位点的数值。可以通过这些基础的统计指标对数据整体的分布和概况有所了解。当然，Pandas还提供了其它丰富的统计分析功能，可参阅pandas官方文档（<http://pandas.pydata.org/）和互联网上的相关教程>。

## 2.Python中的数据可视化：matplotlib

我们的要求构造两个序列X和Y：

>>> X = [[(float)(value[0:4])] for value in df.iloc[:, 0]]

>>> Y = [value for value in df.iloc[:, 1]]

>>> X

[[1980.0], [1981.0], [1982.0], [1983.0], [1984.0], [1985.0], [1986.0], [1987.0], [1988.0], [1989.0], [1990.0], [1991.0], [1992.0], [1993.0], [1994.0], [1995.0], [1996.0], [1997.0], [1998.0], [1999.0], [2000.0], [2001.0], [2002.0], [2003.0], [2004.0], [2005.0], [2006.0], [2007.0], [2017.0], [2009.0], [2010.0], [2011.0], [2012.0]]

>>> Y

[1182, 1098, 1156, 993, 1182, 1027, 1151, 1210, 976, 1130, 1022, 1151, 1130, 1162, 1110, 856, 1047, 1169, 1268, 1204, 1239, 1185, 1021, 1165, 1095, 1046, 1387, 1225, 1139, 1103, 1053, 1285, 1090]

在数据分析中，可视化是一种有力的工具，好的数据可视化方式能够让我们方便直观地洞察数据的分布和趋势，也易于得出相应的分析结论。在Python中，我们可以通过matplotlib库来进行数据的可视化。

我们可以上述分析中的点描画成图，方便我们直观地进行分析。在上面程序代码的基础上,完成可视化绘图:

>>> import matplotlib.pyplot as plt

>>> plt.scatter(X,Y,color='blue')

<matplotlib.collections.PathCollection object at 0x000000001E4292E8>

>>> plt.show()

实验报告五

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 班级 |  | | 姓名 |  | 学号 |  |
| 日期 |  | | 教室 |  | 成绩 |  |
| 实验名称 | |  | | | | |
| 实验目的 | |  | | | | |
| 实  验  内  容 | | 完成题目要求,将代码与显示结果截图展示如下   1. 题目：调用Pandas的describe方法,对指定数据进行分析获得的统计特征。      1. 可视化绘图   我们的要求构造两个序列X和Y,采用可视化工具matplotlib库来进行数据的可视化,完成绘图如下: | | | | |
| 实验总结 | |  | | | | |