日照职业技术学院
RIZHAO POLYTECHNIC

数据采集与治理

项目五 井下环境监测数据处理

主讲：赵娜

项目五 井下环境监测数据处理

进行数据分析的数据源，经常出现缺失值和异常值。在对这些数据进行分析之前，需要首先处理这些缺失值和异常值。

本项目处理数据的缺失值和异常值。通过案例展示，用户可以掌握使用 python、numpy 和 pandas 处理异常值和缺失值的方法。

知识目标和技能目标

【知识目标】

- 了解数据缺失值、异常值的概念
- 了解插值的概念

【技能目标】

- 能够进行缺失值、异常值处理
- 能够自己开发异常值处理程序
- 能够使用第三方包进行异常值处理

情境描述

超稳安防公司是一家专业从事煤矿安全监控、预防的公司。最近公司承接了某煤井的井下环境监控工作，需要通过采集井下温度、井下湿度、以及气体的涌出量，评估、监控井下的环境及状况。

由于井下环境复杂，采集到的数据中存在较为严重的噪声，而且在数据采集和传输过程中，经常会产生异常值，甚至发生无法采集到数据的情况。为此，上位机需要对原始数据进行处理。

超稳公司将数据处理的业务交给了欢喜科技公司。

小刘经过调研与分析认为，数据表中的缺失数据、显然不正常的数据和噪声，是由于采集手段、设备故障、传输过程中的电磁干扰等问题引起的。可以使用插值技术对缺失值和异常值进行处理，使得数据符合规律。可以使用滤波技术对原始数据进行处理，达到平滑数据，去除噪音的目的。

大纲

1 任务分析

2 井下温度缺失值和异常值处理

3 使用numpy处理其他指标

4 使用pandas进行处理

任务分析

数据表的前几行

采集时间点	温度（?C）	相对湿度	瓦斯(m?/min)	一氧化碳(m?/min)
1	30.22	69	2.9	3.6
2	37.68		2.86	3.64
3	29.32	66		1.66
4	37.44	68	1.18	6.49
5	29.46	75	3.81	4.78
6	30.12	77	1.93	4.2

数据表的第9行-15行

9	34.38	62	3.36	4.39
10	30.79	75	2.4	5.79
11	25.17	69	3.34	6.28
12	20.5	80	2.95	6.22
13	37.21		999.99	4.75
14	31.88	94	2.41	1.49
15	39.94	77	1.98	1.08

数据表中的“温度（?C）”应该是“温度（°C）”，“(m?/min)”应该是(m³/min)。在这里出现无法识别的符号“?”，应该是数据采集系统文件存储格式不支持特殊字符。

在该表中，存在一些缺失的数据，例如第2行第3列的数据。与此同时，表格中还存在一些异常值，例如第13行第4列的数据。进一步观察该表格可以确定，所有值为999.99的数据均为异常值。

井下温度缺失值和异常值处理

加载numpy包

加载numpy包

```
import numpy as np
```

读取温度值并绘制温度曲线

使用loadtxt( )函数读取数据文件的第2列。

```
temperature = np.loadtxt('ug_detect.csv',\
                         delimiter=',',\
                         skiprows=1, \
                         usecols=(1),\
                         unpack = False)
```

读取温度值并绘制温度曲线

运行代码，出现错误，即“ValueError: could not convert string to float:”。原因在于，numpy数组默认存储的数据类型是浮点数，因此需要将使用loadtxt( )函数从数据表中读取的字符串数据转换为浮点数。

```
Traceback (most recent call last):
  File "C:\Users\xuegw\Desktop\code\xx.py", line 6, in <module>
    unpack = False)
  File "C:\Program Files (x86)\Python36-32\lib\site-packages\numpy\lib\npyio.py"
, line 1092, in loadtxt
    for x in read_data(_loadtxt_chunksize):
  File "C:\Program Files (x86)\Python36-32\lib\site-packages\numpy\lib\npyio.py"
, line 1019, in read_data
    items = [conv(val) for (conv, val) in zip(converters, vals)]
  File "C:\Program Files (x86)\Python36-32\lib\site-packages\numpy\lib\npyio.py"
, line 1019, in <listcomp>
    items = [conv(val) for (conv, val) in zip(converters, vals)]
  File "C:\Program Files (x86)\Python36-32\lib\site-packages\numpy\lib\npyio.py"
, line 738, in floatconv
    return float(x)
ValueError: could not convert string to float:
```

读取温度值并绘制温度曲线

设置loadtxt( )函数的参数dtype。

```
temperature_str = np.loadtxt('ug_detect.csv',\
                             dtype = bytes, \
                             delimiter=',',\
                             skiprows=1,\
                             usecols=(1),\
                             unpack = False)
print("读取出的数组是temperature_str: \n", \
      temperature_str)
```

读取温度值并绘制温度曲线

运行代码，输出打印数据。

```
读取出的数组是temperature_str:
 [b'30.22' b'37.68' b'29.32' b'37.44' b'29.46' b'30.12' b'26.3' b''
 b'34.38' b'30.79' b'25.17' b'20.5' b'37.21' b'31.88' b'39.94' b'33.65'
 b'27.21' b'27.57' b'31.59' b'' b'34.88' b'29.65' b'26.05' b'33.5'
 b'34.71' b'999.99' b'37.06' b'28.57' b'28' b'25.67' b'20.59' b'39.84'
 b'29.22' b'33.19' b'38.7' b'' b'21.56' b'38.06' b'37.95' b'35.8' b'29.75'
 b'32.69' b'33.94' b'36.31']
```

数组元素的值是“b’ x”这样的形式，这意味着该元素是字符串，字符串的前缀“b”表明该字符串是以bytes格式存储的。之后，需要将bytes格式存储的字符串，存储为Python 3.x支持的格式，例如utf-8、unicode、gb2312等

读取温度值并绘制温度曲线

创建一个长度为len(temperature_str)的数组temperature，使用for循环结构，将temperature_str中的字符串数据，转换为浮点型数据，存储在temperature中。在转换过程中，首先需要判断转换的字符串是不是“空”字符串，即其值是否为“b""”。

```
temperature = np.ndarray( len(temperature_str) )
for index in range(0, len(temperature_str)) :
    item = temperature_str[index]
    if item != b"":
        item = item.decode( 'gb2312' )
        item = float( item )
    else:
        item = None
    temperature[index] = item
```

读取温度值并绘制温度曲线

运行代码，输出打印数据。

```
温度是：
 [ 30.22  37.68  29.32  37.44  29.46  30.12  26.3     nan  34.38  30.79
  25.17  20.5   37.21  31.88  39.94  33.65  27.21  27.57  31.59    nan
  34.88  29.65  26.05  33.5   34.71 999.99  37.06  28.57  28.     25.67
  20.59  39.84  29.22  33.19  38.7     nan  21.56  38.06  37.95  35.8
  29.75  32.69  33.94  36.31]
```

读取温度值并绘制温度曲线

井下温度应该是在50℃以下。从数据文件读出的数据中，存在999.99，将该值作为异常值处理。将异常值用None对象来代替。

```
for index in range(0, len(temperature)) :
    item = temperature[index]
    if item >= 500.0:
        item = None
    temperature[index] = item

print("温度是: \n", temperature)
```

读取温度值并绘制温度曲线

运行代码，输出打印数据。

```
温度是:
 [30.22 37.68 29.32 37.44 29.46 30.12 26.3    nan 34.38 30.79 25.17 20.5
 37.21 31.88 39.94 33.65 27.21 27.57 31.59   nan 34.88 29.65 26.05 33.5
 34.71   nan 37.06 28.57 28.    25.67 20.59 39.84 29.22 33.19 38.7    nan
 21.56 38.06 37.95 35.8  29.75 32.69 33.94 36.31]
```

绘制温度曲线

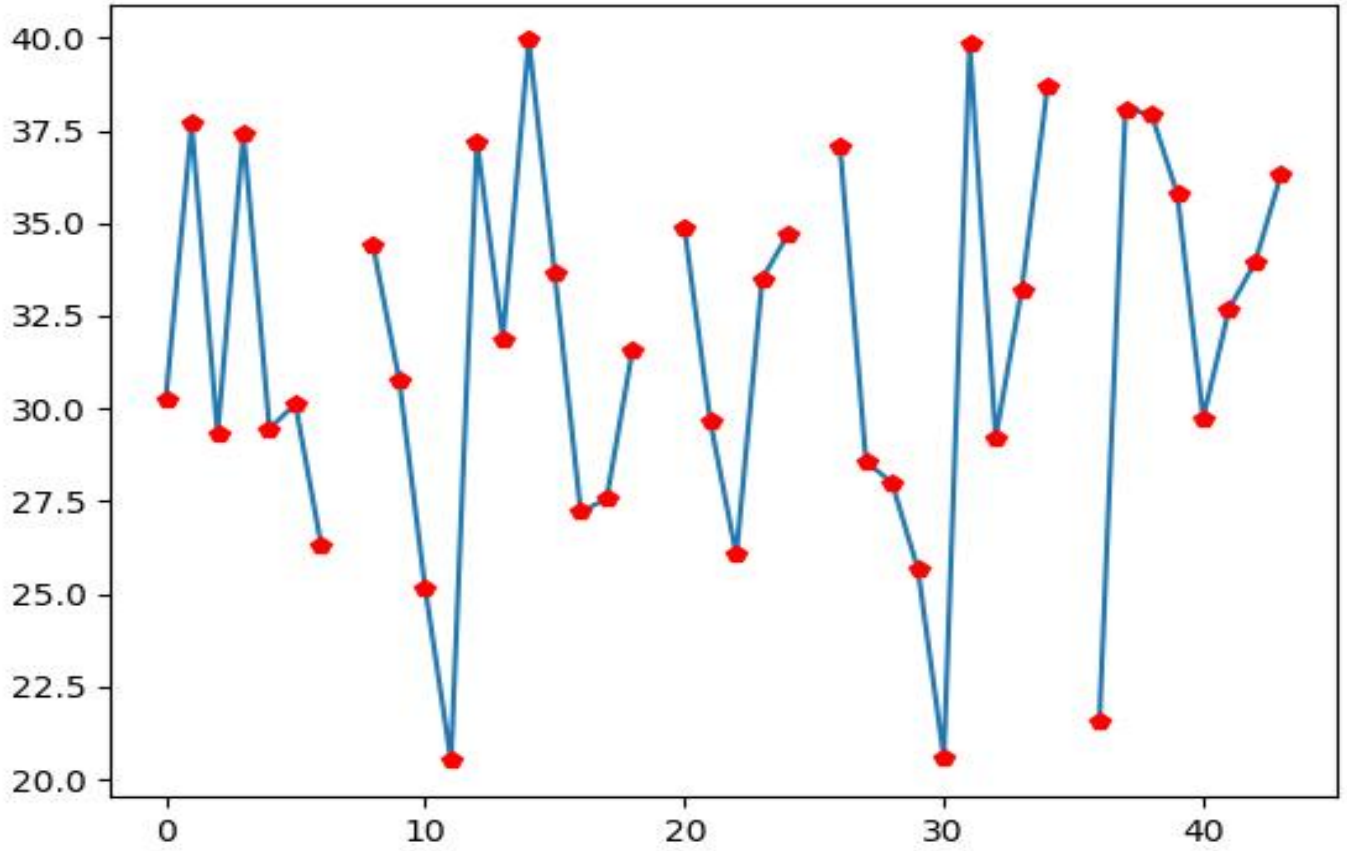
为了能够直观的检测数据表中的缺失数据，绘制数据图。

```
import matplotlib.pyplot as plt

t = np.arange( len( temperature  ))
plt.plot(t,temperature)
plt.plot(t,temperature,'pr')
plt.show()
```

绘制温度曲线

运行代码，绘制数据。数据曲线上存在部分中断的部分。中断的部分即由缺失值或nan值引起。



读取温度值绘制温度曲线

运行代码，输出打印数据。

```
温度是:
 [30.22 37.68 29.32 37.44 29.46 30.12 26.3    nan 34.38 30.79 25.17 20.5
 37.21 31.88 39.94 33.65 27.21 27.57 31.59   nan 34.88 29.65 26.05 33.5
 34.71   nan 37.06 28.57 28.    25.67 20.59 39.84 29.22 33.19 38.7    nan
 21.56 38.06 37.95 35.8  29.75 32.69 33.94 36.31]
```

处理缺失值

对于numpy数组中的nan值，主要是通过数学方法，找到合理的值进行代替。这是通过插值等数值计算技术实现的。

```
def bisec(dataArray):
    for index in range(0, len(dataArray)) :
        if np.isnan ( dataArray[index]):
            dataArray[index] = 0.5 * \
                               ( dataArray[index - 1] + \
                                 dataArray[index + 1] )
```

处理缺失值

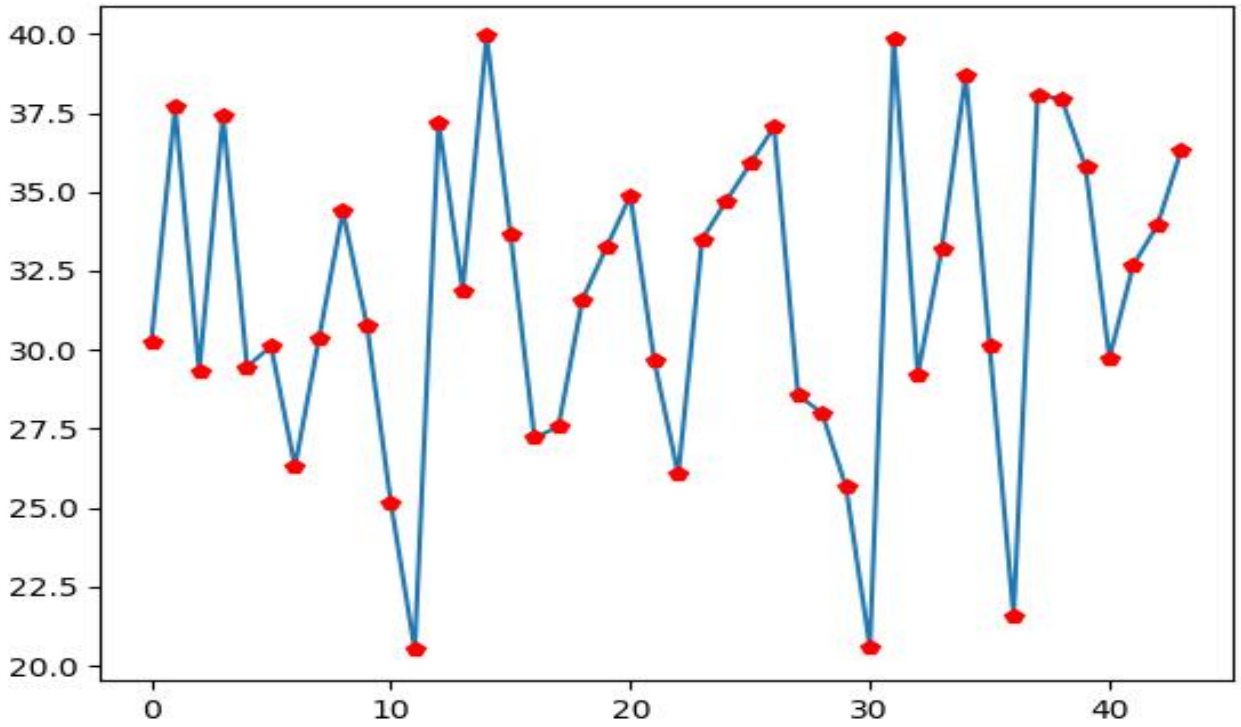
调用该函数，完成对数据表中nan值的处理，

```
温度是：
 [30.22  37.68  29.32  37.44  29.46  30.12  26.3   30.34  34.38  30.79
 25.17  20.5   37.21  31.88  39.94  33.65  27.21  27.57  31.59  33.235
 34.88  29.65  26.05  33.5   34.71  35.885 37.06  28.57  28.    25.67
 20.59  39.84  29.22  33.19  38.7   30.13  21.56  38.06  37.95  35.8
 29.75  32.69  33.94  36.31 ]
```

数据可视化


```
t = np.arange( len( temperature  ))
plt.plot(t,temperature)
plt.plot(t,temperature,'pr')
plt.show()
```

数据可视化



将数据写入数据文件


```
np.savetxt('../data/ug_temperature.csv',\
           temperature,\
           delimiter=',',\
           fmt='%.2f')
```

使用numpy处理其他指标

引入numpy包

需要使用numpy包中的函数和数据类型，需要使用matplotlib.pyplot中的绘图函数。引入相应的包。

```
import numpy as np
import matplotlib.pyplot as plt
```

创建输入数据格式转换函数

创建一个函数inMixData2float( )，该函数将包含空白数据且元素数据类型为字符串的输入数组，转换为元素类型为浮点数的数组。

```
def inMixData2float(org_array, new_array):
    for index in range(0, len(org_array)) :
        item = org_array[index]
        if item != b'':
            item = item.decode( 'gb2312' )
            item = float( item )
        else:
            item = None
        new_array[index] = item
```

创建异常值转换函数

创建一个函数defectsCop（），该函数将数组中的异常值转换为空值None。

```
def defectsCop(data_array, threshold):
    for index in range(0, len(data_array)) :
        item = data_array[index]
        if item >= float(threshold):
            item = None
        data_array[index] = item
```

创建异常值转换函数

创建一个函数bisec（），该函数将数组中的空值nan用相邻两个元素的平均值代替。

```
def bisec(dataArray):
    for index in range(0, len(dataArray)) :
        if np.isnan ( dataArray[index]):
            if index == 0:
                dataArray[index] = \
                                   dataArray[index + 1]
            elif index == len(dataArray):
                dataArray[index] = \
                                   dataArray[index - 1]
            else:
                dataArray[index] = \
                                   0.5 * \
                                   ( dataArray[index - 1] +\
                                     dataArray[index + 1] )
```

读取表格数据

使用loadtxt( )函数读取数据文件的第3、4、5列，分别赋值给变量humidity_str、gas_str和co_str。这三列分别是井下的湿度、瓦斯浓度和一氧化碳浓度。

```
(humidity_str,
 gas_str,
 co_str) = np.loadtxt('ug_detect.csv', \
                      dtype = bytes,\
                      delimiter=',',\
                      skiprows=1,\
                      usecols=(2,3,4),\
                      unpack=True)
```

创建新数组并将数组元素处理为可处理的值


```
humidity = np.ndarray( len( humidity_str ) )
gas = np.ndarray( len( gas_str ) )
co = np.ndarray( len( co_str ) )
inMixData2float(humidity_str,humidity)
defectsCop( humidity, 200 )
inMixData2float(gas_str,gas)
defectsCop( gas, 100 )
inMixData2float(co_str,co)
defectsCop( co, 100 )

print("井下的湿度是: \n", humidity)
print("井下的瓦斯气体浓度是: \n", gas)
print("井下的一氧化碳浓度是: \n", co)
```

创建新数组并将数组元素处理为可处理的值

运行代码，输出打印数据。。

```
井下的湿度是：
 [69. nan 66. 68. 75. 77. 83. 66. 62. 75. 69. 80. nan 94. 77. 67. 79. nan
 94. 93. 92. 92. 80. 79. 72. nan 99. 64. 66. 60. 67. nan 74. 61. 62. 62.
 93. 67. 68. 96. 80. 69. 96. 78.]
井下的瓦斯气体浓度是：
 [2.9  2.86  nan 1.18 3.81 1.93 2.07 1.46 3.36 2.4  3.34 2.95  nan 2.41
 1.98 2.03 1.41 3.39 3.27  nan 1.58 2.69 2.61 1.26 1.82 3.77 1.94 3.93
 1.08 2.33 3.88 3.4  1.09 3.22 3.61  nan 3.63 3.21 1.12 1.52 1.3  3.26
 3.42 3.69]
井下的一氧化碳浓度是：
 [3.6  3.64 1.66 6.49 4.78 4.2  3.67  nan 4.39 5.79 6.28 6.22 4.75 1.49
 1.08  nan 2.5  1.7  5.02 4.69 3.01 5.16 2.8  6.81 2.49  nan 3.85  nan
 2.4  3.84 5.19 3.77  nan 1.74 6.52 6.83 4.43 2.41 4.56 5.54 2.23 1.43
 1.4  5.34]
```

创建数组中的空元素


```
bisec(humidity)
bisec(gas)
bisec(co)
print("井下的湿度是：\n", humidity)
print("井下的瓦斯气体浓度是：\n", gas)
print("井下的一氧化碳浓度是：\n", co)
```

处理数组中的空元素

运行代码，输出打印数据。。

```
井下的湿度是：
 [69.   67.5 66.   68.   75.   77.   83.   66.   62.   75.   69.   80.   87.   94.
 77.   67.   79.   86.5 94.   93.   92.   92.   80.   79.   72.   85.5 99.   64.
 66.   60.   67.   70.5 74.   61.   62.   62.   93.   67.   68.   96.   80.   69.
 96.   78. ]
井下的瓦斯气体浓度是：
 [2.9   2.86  2.02  1.18  3.81  1.93  2.07  1.46  3.36  2.4   3.34  2.95
 2.68  2.41  1.98  2.03  1.41  3.39  3.27  2.425 1.58  2.69  2.61  1.26
 1.82  3.77  1.94  3.93  1.08  2.33  3.88  3.4   1.09  3.22  3.61  3.62
 3.63  3.21  1.12  1.52  1.3   3.26  3.42  3.69 ]
井下的一氧化碳浓度是：
 [3.6   3.64  1.66  6.49  4.78  4.2   3.67  4.03  4.39  5.79  6.28  6.22
 4.75  1.49  1.08  1.79  2.5   1.7   5.02  4.69  3.01  5.16  2.8   6.81
 2.49  3.17  3.85  3.125 2.4   3.84  5.19  3.77  2.755 1.74  6.52  6.83
 4.43  2.41  4.56  5.54  2.23  1.43  1.4   5.34 ]
```

将数据写入数据文件


```
print("保存处理后的湿度数据文件。")
np.savetxt('ug_humidity.csv', \
           humidity,
           delimiter = ',', \
           fmt = '%.2f')
print("保存处理后的瓦斯浓度数据文件。")
np.savetxt('ug_gas.csv', \
           gas,
           delimiter = ',', \
           fmt = '%.2f')
print("保存处理后的一氧化碳浓度数据文件。")
np.savetxt('ug_co.csv', \
           co,
           delimiter = ',', \
           fmt = '%.2f')
```

使用pandas进行处理

引入包

需要使用pandas、matplotlib.pyplot和scipy.interpolate包中的函数和数据类型，因此引入这三个包。

```
import pandas as pd
import matplotlib.pyplot as plt
import scipy.interpolate as itp
```

创建异常值转换函数

创建一个函数defectsCop（），用来寻找数据集当中的异常值。

```
def defectsCop(data_series, threshold):
    for index in range(0, len(data_series)):
        item = data_series[index]
        if item >= float(threshold):
            item = None
```

将空白值替换为其相邻元素拉格朗日插值的结果

创建一个函数seriesltp（）

```
def seriesItp(data_series):
    for index in range(0,len(data_series)):
        item = data_series[index]
        if pd.isnull(data[index]):
         x_list = [index-1,index+1]
         y_list = [data[index-1],data[index+1]]
         lagrange_poly =itp.lagrange(x_list,y_list)
         data_series[index] = lagrange_poly(index)
```

读取数据文件并创建Series对象

使用pandas的read_csv( )函数读取数据文件，并赋值给变量humidity_data、gas_data和co_data。

```
ug_data = pd.read_csv('ug_detect.csv',\
                      header = 0, \
                      encoding='gb2312')
temperature_data = ug_data[u'温度（?C）']
humidity_data = ug_data[u'相对湿度']
gas_data = ug_data[u'瓦斯(m?/min)']
co_data = ug_data[u'一氧化碳(m?/min)']
```

进行数据处理


```
defectsCop(temperature_data, 60)
defectsCop(humidity_data, 200)
defectsCop(gas_data, 100)
defectsCop(co_data, 100)
seriesItp(temperature_data)
seriesItp(humidity_data)
seriesItp(gas_data)
seriesItp(co_data)
```

绘制处理后数据的图像


```
ug_data = pd.read_csv(
    "ug_detect.csv",
    header=0,
    encoding='gb2312')
gas_data_org = ug_data[u'瓦斯(m?/min)']
defetcsCop(gas_data_100, 100)
t = range(len(gas_data_org))
plt.plot(t, gas_data_org)
plt.plot(t, gas_data, 'pr')
plt.show()
```

将数据写入数据文件


```
all_data = pd.DataFrame(\
    {"温度":temperature_data,\
     "相对湿度":humidity_data,\
     "瓦斯浓度":gas_data, \
     "一氧化碳浓度":co_data})
all_data.to_csv('all_data_pandas.csv',\
                        index = False, \
                        encoding='gb2312')
```

日照职业技术学院
RIZHAO POLYTECHNIC

感谢观看

主讲：赵娜