数据科学编程homework10说明文档

数据科学与大数据技术
320180941011
胡叶龙
995587015@qq.com

Github无法上传大于25MB的文件，我仅将.py源代码和几个示例数据集放入此次提交的作业文件夹

原文件结构如下：

```
homework10
├── data
│   ├── preprocessing.py
│   ├── acc_time.py
│   ├── acc_time_contrast.html
│   ├── accelerometer
│   │   ├── anxiety
│   │   │   └── female
│   │   │       ├── all_the_rawdata.json
│   │   │       └── data_line_charts.html
│   │   ├── health
│   │   │   └── female
│   │   │       ├── all_the_rawdata.json
│   │   │       └── data_line_charts.html
│   │   ├── aa.py
│   │   └── ah.py
│   ├── device_motion
│   │   ├── anxiety
│   │   │   └── female
│   │   │       ├── all_the_rawdata.json
│   │   │       └── data_scatter3D_charts.html
│   │   ├── health
│   │   │   └── female
│   │   │       ├── all_the_rawdata.json
│   │   │       └── data_scatter3D_charts.html
│   │   ├── dma.py
│   │   └── dmh.py
│   └── gyroscope
│       ├── anxiety
│       │   └── female
│       │       ├── all_the_rawdata.json
│       │       └── data_scatter3D_charts.html
│       ├── health
│       │   └── female
│       │       ├── all_the_rawdata.json
│       │       └── data_scatter3D_charts.html
│       ├── ga.py
│       └── gh.py
└── documentation.pdf
```

（本文件夹中所有html文件为一些对数据的可视化处理，打开前需确保网络连接正常。）

其中对数据的处理思路是：将数据分为accelerometer、device_motion、gyroscope三大类，每类中分为anxiety组和health组。使用os.walk()遍历文件夹中的所有数据集，对每一份数据集以json.load()的方式加载数据，随后使用pyecharts库将数据处理为一些数据预处理中常用的可视化图表类型，对数据的正确性和可用性进行直观的判断。

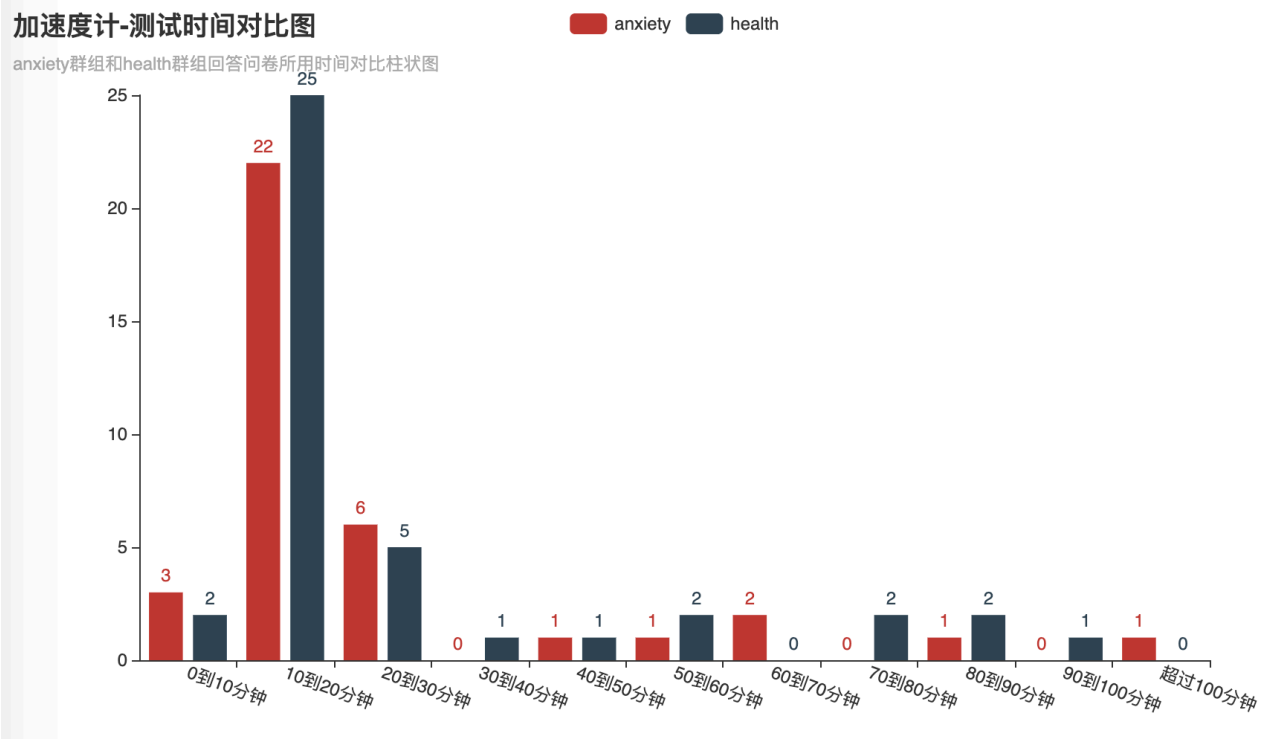
acc_time.py

以accelerometer加速度计的传回数据比较anxiety组和health组的答题时间，数据采样频率为5Hz，则将数据总量除5再除60 即可得到当前样本的答题时间，保留小数点后两位。

```
        if path[-5:] == ".json":
            with open(path, 'r', encoding = 'utf-8') as f:
                data = json.load(f)
                counts_aa = len(data)
                time_aa = round(counts_aa / 5 / 60, 2) #sample frequency 5Hz
```

加入if path[-5:] == “.json”判断条件是由于macOS会在文件夹中增加一个.DS_Store的隐藏文件，加入这个条件避免使用os.walk()时导致json.load()无法解析而出错。

采用的策略是将anxiety和health分别的答题时间记为两个list，每十分钟为一组，采用if_elif_else结构判断答题时间，随后得到完整的anxiety组和health组的分组数据列表，使用pyecharts库进行可视化，形成一份柱状图，渲染出acc_time_contrast.html文件。可以粗略地看出health组更多人能够在合理的时间里答完问卷（老师的说法是问卷设计时长是十多分钟）。

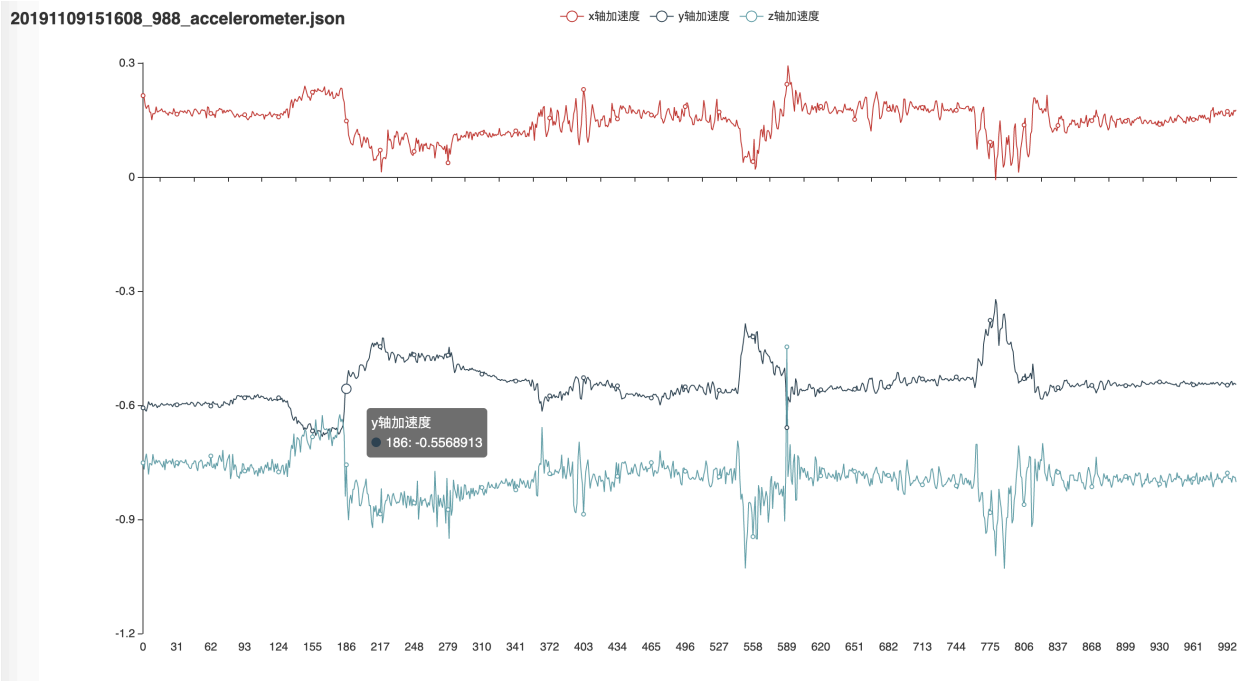


（更多详细数据可以点击acc_time_contrast.html动态查看）

aa.py、ah.py、dma.py、dmh.py、ga.py、gh.py

分别代表accelerometer/anxiety、accelerometer/health、device_motion/anxiety、device_motion/health、gyroscope/anxiety、gyroscope/health。

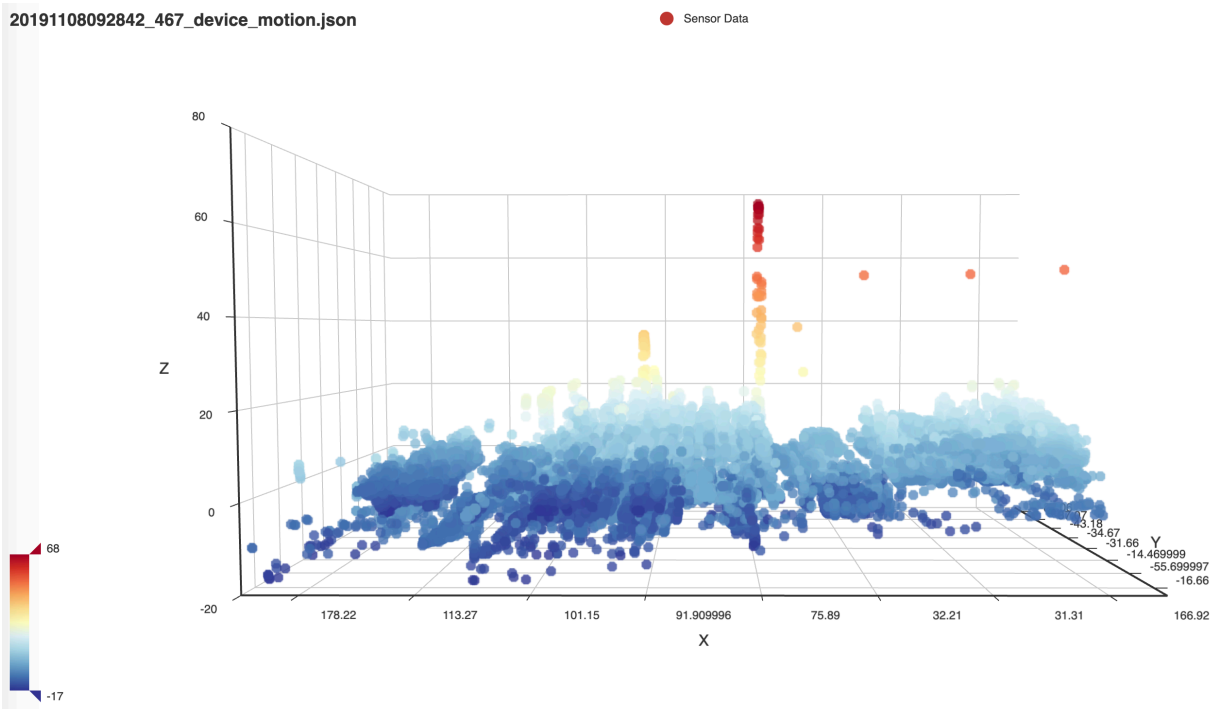
对于accelerometer的数据，将他们做成折线图，因为我觉得对于加速度计来说，不同朝向的波动都是比较重要的数据，折线图中能够较好的表现这种波动，通过分析波动的剧烈程度能够看出当时那个采样点是否是一个合理的数据，如果波动过于剧烈，或是某一维度的变化趋势与其他的维度都不相关，也许可以作为该采样点附近或者是整个数据文件是否都是一些脏数据的判断依据。



三轴的传感器数据分别为三条折线，横坐标为采样点次序。上例中可观察出x轴与z轴的数据相关性，佐证数据的正确性。

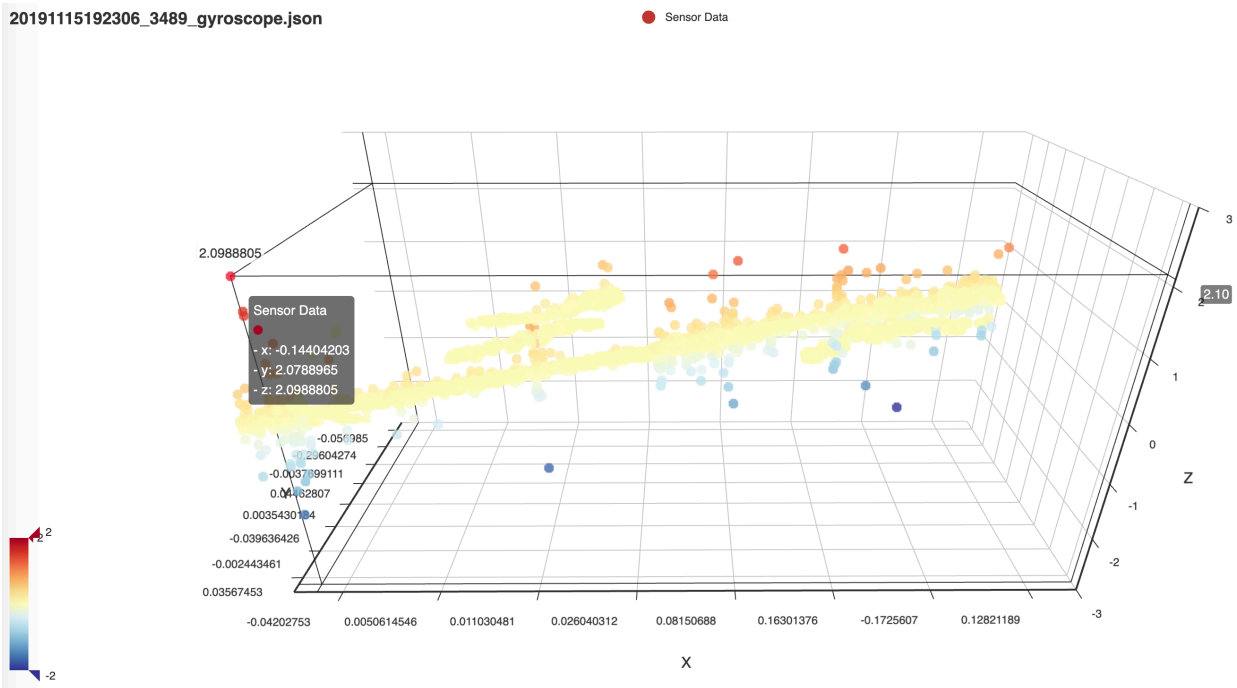
对于device_motion和gyroscope来说，做成了三维的散点图。

对于device_motion，三维散点图可以立体地展示样本所处的空间位置（老师没细说device_motion的记录是什么，我按照名字推测是设备位置运动），若是位置记录过于散乱的部分，可以判断是脏数据，或是不在问卷期间的采样数据。或者又出现一些数据采样点显现出二维线性关系的，明显数据出现了错误，不可采用。



上例是一例device_motion/health中得到的数据，可以观察到数据采样位置相对集中，可以明显观察到进入（或离开）以及回答问卷时，设备的位置情况，证实此例可用作研究数据。

对于gyroscope，三维散点图可以发现偏离严重的数据采样点，或是数据样本的数据分布不正常的情况，辅助判断一份数据集的可用性。而实际情况陀螺仪的数据能够体现的意义应该结合别的数据使用（陀螺仪的数据能够表现出什么意义我没有想通）。下面是gyroscope/anxiety中随机选取的一例，可以分辨出部分偏离较为严重的采样点。



(文件夹中每一份数据都进行了上述的各式处理，可以查看py源代码，以及对可视化渲染的html文件进行动态查看)

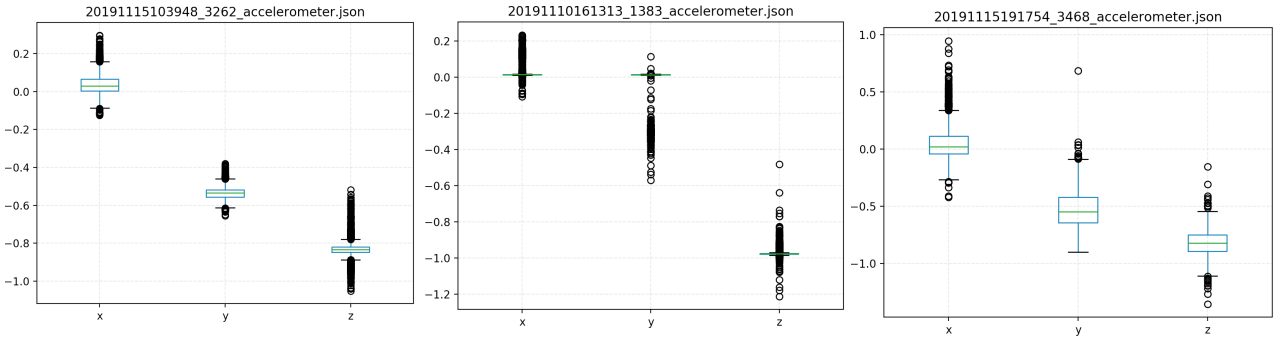
数据预处理

至此对数据的初步了解已经完成，然后就对数据进行合适的预处理即可。preprocessing.py完成了基本的数据预处理，以/accelerometer/anxiety/female/中的数据为例。

```python
import os
import pandas as pd
import matplotlib.pyplot as plt

dir = "/Users/huyelong/Desktop/homework10/data/accelerometer/anxiety/female/"
for root, dirs, files in os.walk(dir):
    for file in files:
        path = os.path.join(root,file)
        #print(path) <-checkpoint
        if path[-5:] == ".json":
            with open(path, 'r', encoding = 'utf-8') as f:
                data = pd.read_json(f)
                print(data.isnull())   #查看data是否有缺失值（空值），False无缺失值
                data.dropna()   #删除带空值的行
                print(data.describe())
                data.plot.box(title = file)
                plt.grid(linestyle = "--", alpha = 0.3)
                plt.show()
```

缺失值处理：isnull()方法判断数据中哪里是缺失值，然后用dropna()删除那一行数据。对于这次给出的数据，是完整度比较高的数据，一般无需太多缺失值处理操作。
通过describe()可以快速查看数据集的基本信息，随后使用箱线图发现异常离群点。



箱线图挺离谱的，应该根据原始数据自己调节合适上下四分位点，但是无法获得更多原数据的有用信息。到这个时候，应该删去异常点和极端异常点，但这种操作应该结合原始数据的说明来进行，我也不知道原始数据要多大范围什么程度的数据才计为合理范围。要删的话加点判断语句即可，非常简单。

数据集成与数据规约对于这样一份规范的原始数据来说不必做。

总结

1.数据集中的数据都是相当规范的，能大大减轻数据预处理的工作量。

2.借助可视化工具可以更好地了解数据情况，决定数据预处理所需的策略。

3.数据预处理中最重要的还应当是与数据提供者咨询讨论数据的具体情况，或是阅读数据来源网站给出的说明，不然容易出现剔除异常值后的效果更差的情况。