Readme3

要求对数据集进行清洗

运行

```
python 3_1.py
```

1.数据指定列名读入

假设有一份会员数据集（data.xlsx）。第一列代表会员的姓名，第二列是性别，第三列是年龄，第四列是体重，第五列是身高。

原始数据列名为数字，按照具体意义指定列名进行读入

需要低版本xlrd库（如1.2.0）支持xlsx读入

```
import pandas as pd
data = pd.read_excel('data.xlsx', 'Sheet1', names=['name', 'gender', 'age',
'weight', 'height'])
```

```
*************** original data ***************
         name  gender   age  weight  height
0      Emma??  female  18.0    50.0   161.0
1     larissa  female  16.0    50.0   170.0
2      Edith搭  female -20.0    46.0   172.0
3      Sophia    male  30.0    70.0     1.7
4       Joyce    male  25.0    52.0   182.0
5         NaN     NaN   NaN     NaN     NaN
6        May锘  female  40.0    70.0   178.0
7         lvy    male -15.0    50.0     1.4
8        Emma  female  18.0    50.0   161.0
9    Stella''    male  30.0    60.0   185.0
10    gloria    male  28.0     NaN   180.0
11       Amy  female  18.0    45.0   160.0
```

2. 数据的完整性检查

发现有一整行记录为空值

姓名为gloria的体重数据缺失

通过`print(data.isnull())`查看当前数据情况，可知data的第5行全为空值，第10行的weight为空值

```
      name  gender    age  weight  height
0    False   False  False   False   False
1    False   False  False   False   False
2    False   False  False   False   False
3    False   False  False   False   False
4    False   False  False   False   False
5     True    True   True    True    True
6    False   False  False   False   False
7    False   False  False   False   False
8    False   False  False   False   False
9    False   False  False   False   False
10   False   False  False    True   False
11   False   False  False   False   False
```

```
cleaned = data.dropna(how='all')    # 清除全为空的行
cleaned = cleaned.fillna({'weight': 80})    # 将weight列的NaN值改为80
```

结果如下图

```
*************** Cleaning step 1 ***************
          name  gender   age  weight  height
0      Emma??  female  18.0    50.0   161.0
1     larissa  female  16.0    50.0   170.0
2      Edith搭  female -20.0    46.0   172.0
3      Sophia    male  30.0    70.0     1.7
4       Joyce    male  25.0    52.0   182.0
6        May铖  female  40.0    70.0   178.0
7         lvy    male -15.0    50.0     1.4
8        Emma  female  18.0    50.0   161.0
9    Stella''    male  30.0    60.0   185.0
10    gloria     male  28.0    80.0   180.0
11       Amy  female  18.0    45.0   160.0
```

3. 数据的全面性检查

发现身高的度量单位不统一，有米的，也有厘米的

同一单位为厘米，检查数据中明显的米为单位的异常值（值小于2），换算为厘米

```
height = data[4]
height[height < 2] = height * 100
```

姓名的首字母大小写不统一，有大写，也有小写的

将所有姓名首字母大写

```
cleaned['name'] = cleaned['name'].str.title()
```

处理后结果如图

```
*************** Cleaning step 2 ***************
         name  gender   age  weight  height
0      Emma??  female  18.0    50.0   161.0
1     Larissa  female  16.0    50.0   170.0
2      Edith搭  female -20.0    46.0   172.0
3      Sophia    male  30.0    70.0   170.0
4       Joyce    male  25.0    52.0   182.0
6        May铖  female  40.0    70.0   178.0
7         Lvy    male -15.0    50.0   140.0
8        Emma  female  18.0    50.0   161.0
9    Stella''    male  30.0    60.0   185.0
10    Gloria     male  28.0    80.0   180.0
11       Amy  female  18.0    45.0   160.0
```

4. 数据的合法性检查

姓名字段存在非ASCII码字符、存在?号非法字符、出现空值

姓名中只包含英文字母A-Z，a-z，因此通过re模块删除name列的A-Z，a-z以外的所有字符

```
names = cleaned['name']
cleaned['name'] = [re.sub('[^A-Za-z]', '', name) for name in names]
```

性别字段存在空格

年龄字段存在负数

将age列中的负数改为其绝对值

```
age[age < 0] = np.abs(age)
```

处理后结果如图

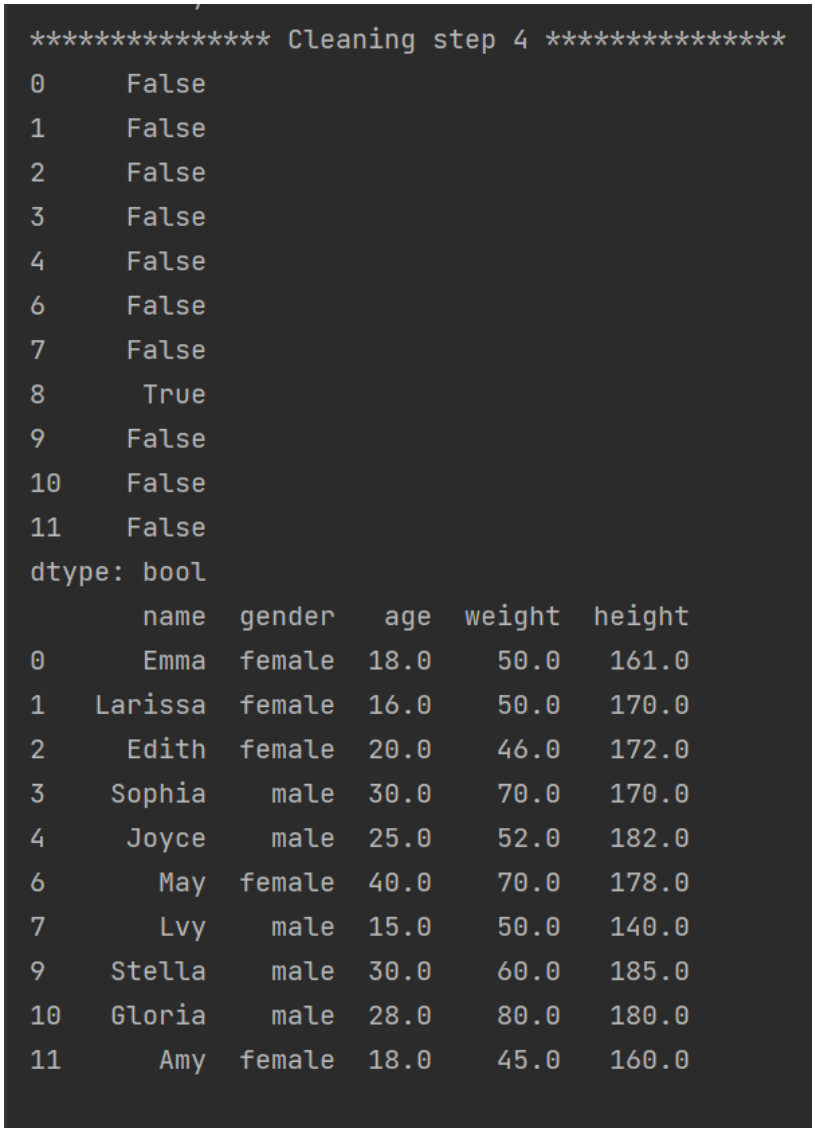
```
*************** Cleaning step 3 ***************
       name  gender   age  weight  height
0      Emma  female  18.0    50.0   161.0
1   Larissa  female  16.0    50.0   170.0
2     Edith  female  20.0    46.0   172.0
3    Sophia    male  30.0    70.0   170.0
4     Joyce    male  25.0    52.0   182.0
6       May  female  40.0    70.0   178.0
7       Lvy    male  15.0    50.0   140.0
8      Emma  female  18.0    50.0   161.0
9    Stella    male  30.0    60.0   185.0
10   Gloria    male  28.0    80.0   180.0
11      Amy  female  18.0    45.0   160.0
```

5. 数据的唯一性检查

姓名为Emma的记录存在重复

```
print(cleaned.duplicated())      # 检查数据中是否有冗余项
cleaned.drop_duplicates()
```

处理后结果如图，可以看出序号8的行有冗余项，处理后冗余项被删除



```
*************** Cleaning step 4 ***************
0     False
1     False
2     False
3     False
4     False
6     False
7     False
8      True
9     False
10    False
11    False
dtype: bool
       name  gender   age  weight  height
0      Emma  female  18.0    50.0   161.0
1   Larissa  female  16.0    50.0   170.0
2     Edith  female  20.0    46.0   172.0
3    Sophia    male  30.0    70.0   170.0
4     Joyce    male  25.0    52.0   182.0
6       May  female  40.0    70.0   178.0
7       Lvy    male  15.0    50.0   140.0
9    Stella    male  30.0    60.0   185.0
10   Gloria    male  28.0    80.0   180.0
11      Amy  female  18.0    45.0   160.0
```

6.存入xlsx文档

保存数据至新xlsx文档，且不保留列索引

```
# 存入xlsx文件
# filepath为文件路径
filepath = 'data_cleaned.xlsx'
cleaned.to_excel(filepath, index=False)
```

"chained" assignments warning

当尝试对dataframe中的数据进行直接更改时，会触发`A value is trying to be set on a copy of a slice from a DataFrame`报错，官方称这是由于 "chained" assignments时，代码中的变量并不是原dataframe的副本，而是指向原始dataframe的数据，因此变量改变时原dataframe也会被更改，这种情况下的结果可能是用户所不希望的，因此触发系统的warning。由于我希望直接覆盖原dataframe，因此我显式关闭了该warning

```
pd.options.mode.chained_assignment = None  # default='warn'
```