计算机 22A4 2022015232 孙春辉
2025 年 3 月 19 日 星期三 第四次作业

Report 4

Part 1: Course exercises

1.1:NAN
1.2:堆叠合并
1.3:哑变量
1.4:异常值
1.5:pivot（）函数

2.1:TRUE
2.2:TRUE
2.3:FALSE
2.4:FALSE
2.5:FALSE

3.1:B
3.2:A
3.3:C
3.4:D
3.5:A

4.1: 异常值是指数据集中的个别值明显偏离它所属数据集的其余值，这些数值是不合理的或错误的。

4.2:堆叠合并、主键合并、根据索引合并、合并重叠数据，总共四种。
5.1:

```
import pandas as pd
import numpy as np
group_a = pd.DataFrame({'A': [2,3,5,2,3],
                        'B': ['5',np.nan,'2','3','6'],
                        'C': [8,7,50,8,2],
                        'key': [3,4,5,2,5]})
group_b = pd.DataFrame({'A': [3,3,3],
                        'B': [4,4,4],
                        'C': [5,5,5]})
print(group_a)
print(group_b)
```

5.2:

```
group_a  = group_a.combine_first(group_b)
group_a
```

5.3:

```
group_a.rename(columns={'key':'D'})
```

Part 2:

Download the files from the Moodle and complete the following Pandas practices and screenshot your solutions and results.

1. 读取 2004 年的数据，并展示查看数据的格式.

```
import numpy as np
import pandas as pd
df = pd.read_csv('./by_year/2014.csv')
print(df.info())
```

```
RangeIndex: 36 entries, 0 to 35
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  1 non-null      object
 1   Unnamed: 1  34 non-null     object
 2   Unnamed: 2  35 non-null     object
 3   Unnamed: 3  34 non-null     object
 4   Unnamed: 4  34 non-null     object
 5   Unnamed: 5  22 non-null     object
dtypes: object(6)
memory usage: 1.8+ KB
None
```

2. 对第一年的数据进行预处理：删除第一行，选取真实列名所在的第一行，更改列名，删除多余的行.

```
df = df.drop(0)


# 更改前的结果 ：
          Unnamed: 0 Unnamed: 1 Unnamed: 2 Unnamed: 3 Unnamed: 4 Unnamed: 5
0                NaN        NaN      流行性感冒        NaN        NaN        NaN
1                NaN         地区         发病数         死亡数         发病率         死亡率
2                NaN         全国       49496          15     3.8077     0.0012
# 更改后的结果：
          Unnamed: 0 Unnamed: 1 Unnamed: 2 Unnamed: 3 Unnamed: 4 Unnamed: 5
1                NaN         地区         发病数         死亡数         发病率         死亡率
2                NaN         全国       49496          15     3.8077     0.0012

df.columns = df.iloc[0]
#更改 df 列名称
1                NaN    地区     发病数   死亡数       发病率      死亡率
1                NaN    地区     发病数   死亡数       发病率      死亡率
2                NaN    全国   49496    15    3.8077   0.0012
3                NaN   北京市       8     0    0.0540      NaN
4                NaN   天津市      13     0    0.1399      NaN

df = df.drop(df.columns[0],axis=1) # 删除多余的列
```

```
1     地区     发病数   死亡数      发病率      死亡率
2     全国   49496    15    3.8077   0.0012
3    北京市       8     0    0.0540      NaN
4    天津市      13     0    0.1399      NaN
5    河北省    1923     0    2.8283      NaN
6    山西省      57     0    0.1720      NaN
```

3. 第二步删除行后，index 被打乱：重新设置被打乱的 index，添加年份变量.

```
df = df.reset_index(drop=True)
# 重新设置索引
# 更改列名称 添加年份变量
df['年份'] = 2023
```

```
 1     地区     发病数   死亡数      发病率      死亡率     年份
 0     全国   49496    15    3.8077   0.0012   2023
 1    北京市       8     0    0.0540      NaN   2023
 2    天津市      13     0    0.1399      NaN   2023
 3    河北省    1923     0    2.8283      NaN   2023
```

4. 批量读取连接数据：自定义一个函数对后面所有年份进行批量预处理（删除第 0 列，删除前朗行后后一行，重塑年份变量）.

```
def modify_file(path,year) :
    df = pd.read_csv(path)
    df = df.drop(0)  # 删除 列名
    df.columns = df.iloc[0]
    df = df.drop(1)  # 删除第一行

    df = df.drop(df.columns[0], axis=1)  # 删除多余的列
    df = df.reset_index(drop=True)
    df['年份'] = year
```

5. 定义函数用于批量读取及凭借数据，调用第四步中定义的函数进行批量处理：建立空列表用于存放数据，通过循环遍历读取文件，读取数据，一次累加年份，进行预处理及其重塑变量等.

```
list_path = []# 用于存放文件的地址
```

```
year = 2004
for i in list_path :
    modify_file(i,year)
    year += 1
```

6. 重塑列名，链接数据，并填充缺失值.

```
other_data.columns = col_name.append(pd.Series("年份")) # 重塑数据的列名
flu_data = pd.concat([dat0, other_data]) # 连接数据，命名为 flu_data
flu_data.fillna(0, inplace = True)       # 使用 0 填充缺失值
```

7. 检查数据：对数据进行计数，替换文字中的空格，删除多余类型数据，替换有歧义数据，再次检查.

```
    flu_data["地区"].value_counts() # 对数据进行计数
flu_data["地区"] = flu_data["地区"].apply(lambda x: x.replace(" ", "")) # 替换文字中的空格
flu_data = flu_data.loc[flu_data["地区"] != "建设兵团"]                # 删除地区为建设
flu_data.loc[flu_data['地区'] =='黑龙江','地区']='黑龙江省'             # 将黑龙江替换为黑龙江省

flu_data["地区"].value_counts()
```

8. 人口数据的清洗与重塑：修改列名，删除多余的行，统一格式.

```
people = pd.read_csv("people_2.csv",encoding="gbk") # 读取人口数据，命名为 people
people.columns = people.iloc[2]                                    # 用第 2 行作为列名
people.drop([0,1,2,len(people)-1,len(people)-2], axis = 0, inplace = True) # 删除多余的行
people.reset_index(inplace=True, drop=True)                        # 删除多余的行
## 统一地区名的格式
people.loc[people['地区'] == '内蒙古自治区','地区']='内蒙古'
people.loc[people['地区'] == '广西壮族自治区','地区']='广西'
people.loc[people['地区'] == '西藏自治区','地区']='西藏'
people.loc[people['地区'] == '宁夏回族自治区','地区']='宁夏'
people.loc[people['地区'] == '新疆维吾尔自治区','地区']='新疆'
peo_name = list(people.columns)
# 获取 people 的变量名
```

```
peo_name.remove("地区")
# 去除地区变量，得到年份数据
change_people = pd.melt(people, id_vars=["地区"], value_vars=peo_name, \
                        var_name="年份", value_name="总人口数")
# 通过 melt 重塑数据
change_people["年份"] = change_people["年份"].apply(lambda x: re.findall("\d+", x)[0]) # 去除年份的“年”字
change_people["年份"] = change_people["年份"].astype(np.int)
```

9. 拼接数据：填充数据，转换数据类型，最终查看数据并保存输出.

```
result = pd.merge(flu_data, change_people, on=['年份', '地区'])
change_list = ['发病率','死亡率','总人口数','发病数','死亡数']
result[change_list] = result[change_list].apply(pd.to_numeric)
result.head()
```

Ref：Pandas 数据预处理与数据重塑案例（https://zhuanlan.zhihu.com/p/44677396）

问题：所给出的数据文件 CSV 2004.csv、2005.csv、2006.cs、2008.csv、2016.csv 文件的编码方式是 GB2312 但是其中 2009.csv、2010.csv、2011.csv、2013.csv、2014.csv 文件的编码方式未知。查找到了是 GBK 文件