计算机学院 数据科学概论 课程实验报告

实验题目：python 数据统计挖掘与应用		学号：202200130041
日期：2023/3/23	班级： 22 级数据班	姓名： 左景萱
Email：zuojingxuan1130@mail.sdu.edu.cn		

实验目的：python 数据统计挖掘与应用编程练习

熟练运用 numpy 和 pandas，本实验首先安装配置好 Anaconda 以及 Python 环境，通过代码编写，从而使同学们熟练掌握 pandas 和 numpy 库的基本操作，并学会将其运用到实践中。

实验软件和硬件环境：

1. 操作系统：Windows10；
2. Anaconda 版本：5.3.0；
3. Python 版本：3.6.8；
4. pandas 版本：版本号不低于 1.2.0

实验步骤与内容：

1. 通过简单的实例了解了 numpy 的一些基本操作，包括随机数的生成以及一些基本的矩阵运算手段和常用函数。
2. 通过简单的实例了解了 pandas 的一些基本操作，包括文件的读入与写出以及基本的数据操作方法。
3. 完成实验要求的第一部分内容

```
# 题目一:
# 计算六项能力的加总
df["Sum"] = df["HP"] + df["Attack"] + df["Defense"] + df["Sp. Atk"] + df["Sp. Def"] + df["Speed"]

# 比较加总值和Total值是否相等，返回布尔值
df["Equal"] = df["Sum"] == df["Total"]

# 查看是否有不相等的数据行，如果有则打印出来
if not df["Equal"].all():
    print(df[df["Equal"] == False])
else:
    print("所有数据行都相等")
```

首先将六项能力值加总，再将总和与”Total”进行比较，如果有不一样的就返回 0。再对于返回值进行遍历，如果都是 1，那就是都一样，如果有 0 存在，那就不一样。

4. 完成实验要求的第二部分内容的 a

```python
# 去除重复记录，保留第一条
df_2 = df.drop_duplicates(["#"], keep="first")

# 求第一属性的种类数量和前三多数量对应的种类
type1_count = df_2["Type 1"].value_counts()
print("第一属性的种类数量为:"+str(type1_count.count()))
print("前三多数量对应的种类:")
for i in range(3):
    print("属性：{:<12}    数量：{}".format(type1_count.index[i],type1_count.iloc[i]))
```

首先用 drop_duplicates 方法去除重复记录，重复记录只保留第一条。再用 value_counts 方法将属性按照出现次数进行排序。最后将属性及对应出现次数进行循环输出。

5. 完成实验要求的第二部分内容的 b

```python
# 获取对应的属性值
df_2_sub=df_2[['Type 1','Type 2']]
# 进行清洗和属性的组合操作
type_combination=df_2_sub.dropna().drop_duplicates(['Type 1','Type 2'])
type_combination=type_combination.apply(tuple,axis=1)
for i,j in enumerate(type_combination):
    print("第{0}种组合为: {1}".format(i+1,j))
```

首先获取对应的需要的数值的列，然后用 dropna 方法去除 nan 的值，再去除重复的属性组合，最后将属性组合按行打包成元组，再循环输出。

6. 完成实验要求的第二部分内容的 c

```python
# 求尚未出现过的属性组合
# 将Type 1和Type 2中所有出现过的属性放到一个set中
type_set = set(df_2["Type 1"]) | set(df_2["Type 2"].dropna())
type_all_combination = set()
# 将互异的属性值进行两两组合
for t1 in type_set:
    for t2 in type_set:
        # 第一属性和第二属性不同
        if t1 != t2:
            type_all_combination.add(tuple([t1,t2]))
#   取出互异的元素
type_not_appear = type_all_combination.difference(type_combination)
for i,j in enumerate(type_not_appear):
    print("第{0}种组合为: {1}".format(i+1,j))

```

首先将 Type1 和 Type2 中所有出现过的属性放到一个 set 中，再将其中所有互异的元素进行组合形成集合 type_set，再将之与已有组合之间的差集作为 type_not_appear

集合，循环进行输出。

7. 完成实验要求第三部分内容的 a

```python
# 取出物攻，超过120的替换为high，不足50的替换为low，否则设为mid
def helper_func(x):
    if x>120:
        return "high"
    elif x<50:
        return "low"
    else:
        return "mid"

df_3=df['Attack'].apply(helper_func)
df_3
```

利用 apply 方法进行值的替换作为 df_3

8. 完成实验要求第三部分的内容 b

```python
# 取出第一属性，用replace替换所有字母为大写
type_1_upper = df["Type 1"].str.upper()
dictionary=dict(zip(df["Type 1"],type_1_upper))
type1=df["Type 1"]
type1=type1.replace(dictionary)
type1
```

首先取出第一属性,并将之全部转化为大写。再使用 dict 封装原 type1 和大写的 type1，最后再使用 replace 方法进行替换。

9. 完成实验要求第三部分的内容 c

```python
# 求每个妖怪六项能力的离差，即所有能力中偏离中位数(median（）)最大的值，添加到df并从大到小排序
# 按照索引获得中位数
stats = ["HP", "Attack", "Defense", "Sp. Atk", "Sp. Def", "Speed"]
median = df[stats].median()
# 按照行获得中位数的偏移量
deviation = df[stats].apply(lambda x: abs(x - median), axis=1)
# 按照行获得中位数偏移量的最大值
max_deviation = deviation.max(axis=1)
df["Max Deviation"] = max_deviation
df = df.sort_values(by="Max Deviation", ascending=False)
df
```

首先按照行获得每一个对象的能力中位数，再按照行获取每一个数离中位数的偏移量，再取出偏移量的最大值拼接回原 dataframe。

结论分析与体会：

Pandas 处理数据比较高效，但是在使用过程中要注意 Nan 值的处理，防止使用常用函数的时候返回 Nan 值。

使用 Pandas 进行数据处理的时候可以使用 set，tuple 等方法进行数据的分析与展示，可以更加清晰。

就实验过程中遇到的问题及解决处理方法，自拟 1－3 道问答题：

1. 在使用 replace 方法的时候应该如何对于某一个列整体进行替换？
   答：可以直接令 dataframe 中的需要被替换的列等于被替换掉的列，也可以将被替换的列和替换的列传入字典再进行替换。