

# 课后作业：Python机器学习环境的安装和配置

## 【作业提交】

将分类结果保存到文本文档进行提交，同时提交源代码。

1. 测试结果命名为: ex01-结果图01-你的学号-你的姓名.png, ex01-结果图02-你的学号-你的姓名.png
2. 源代码命名为: ex01-01-你的学号-你的姓名.py, ex01-02-你的学号-你的姓名.py, ex01-03-你的学号-你的姓名.py

*结果文件，要求每小题标注题号，两题之间要求空一行*

要求完成以下几个作业：

- 利用pandas库和字典完成以下习题（ex01-01）
- 利用matplotlib库绘制如下图（1）使用代码生成一个表达式为： $y = 2x^2 - 3x^4 + 6x - 3$ 的曲线图。（ex01-02）
- 利用matplotlib库绘制如下图（2）生成直方图（ex01-03）

## 一. 利用pandas库和字典完成以下习题

1. 生成如下表格

```
1 # 使用import关键字引入pandas库，为了简便使用缩写“pd”来表示pandas库。
2 import pandas as pd
3
4 # 使用字典数据类型创建一个数据表，并用pandas库的DataFrame数据结构进行显示
5 data = {"姓名":["拿破仑","哥德巴赫","秦始皇","凯撒","耶稣"],
6         "国家":["地球联盟","黑暗祭祀","地球联盟","黑暗祭祀","地球联盟"],
7         "学号":["2019001","2019002","2019003","2019004","2019005"],
8         "语文":[33,28,32,30,99],
9         "数学":[98,97,94,92,100],
10        "英语":[33,28,32,30,97],
11        "科学":[98,97,94,92,98],
12        "历史":[94,57,96,98,91]
13     }
14 data_frame = pd.DataFrame(data)
15 display(data_frame)
```

```
1 .dataframe tbody tr th {
2     vertical-align: top;
3 }
4
5 .dataframe thead th {
6     text-align: right;
7 }
```

	姓名	国家	学号	语文	数学	英语	科学	历史
0	拿破仑	地球联盟	2019001	33	98	33	98	94
1	哥德巴赫	黑暗祭祀	2019002	28	97	28	97	57
2	秦始皇	地球联盟	2019003	32	94	32	94	96
3	凯撒	黑暗祭祀	2019004	30	92	30	92	98
4	耶稣	地球联盟	2019005	99	100	97	98	91

2. 通过对表格的操作，获得以下几个结果

(1) 获取国家不等于“地球联盟”的数据

```
1 # 使用“不等于 !=”操作符排除字段中包含特定值的数据
2 display(data_frame[data_frame.国家 != "地球联盟"])
```

```
1 .dataframe tbody tr th {
2     vertical-align: top;
3 }
4
5 .dataframe thead th {
6     text-align: right;
7 }
```

	姓名	国家	学号	语文	数学	英语	科学	历史
1	哥德巴赫	黑暗祭祀	2019002	28	97	28	97	57
3	凯撒	黑暗祭祀	2019004	30	92	30	92	98

(2) 获取学号等于2019003的数据

```
1 display(data_frame[data_frame.学号 == "2019003"])
```

```
1 .dataframe tbody tr th {
2     vertical-align: top;
3 }
4
5 .dataframe thead th {
6     text-align: right;
7 }
```

	姓名	国家	学号	语文	数学	英语	科学	历史
2	秦始皇	地球联盟	2019003	32	94	32	94	96

(3) 获取数学成绩大于95的同学的数据

```
1 display(data_frame[data_frame.数学 > 95])
```

```

1 | .dataframe tbody tr th {
2 |     vertical-align: top;
3 | }
4 |
5 | .dataframe thead th {
6 |     text-align: right;
7 | }

```

	姓名	国家	学号	语文	数学	英语	科学	历史
0	拿破仑	地球联盟	2019001	33	98	33	98	94
1	哥德巴赫	黑暗祭祀	2019002	28	97	28	97	57
4	耶稣	地球联盟	2019005	99	100	97	98	91

(4) 获取历史成绩大于60，同时属于“黑暗祭祀”的同学的信息

```

1 | display(data_frame[(data_frame.历史 > 60) & (data_frame.国家 == "黑暗祭祀")])

```

```

1 | .dataframe tbody tr th {
2 |     vertical-align: top;
3 | }
4 |
5 | .dataframe thead th {
6 |     text-align: right;
7 | }

```

	姓名	国家	学号	语文	数学	英语	科学	历史
3	凯撒	黑暗祭祀	2019004	30	92	30	92	98

```

1 | (data_frame.历史 > 60) & (data_frame.国家 == "黑暗祭祀")

```

```

1 | 0    False
2 | 1    False
3 | 2    False
4 | 3     True
5 | 4    False
6 | dtype: bool

```

## 二. 利用matplotlib库绘制如下图

1. 使用代码生成一个表达式为： $y = 2x^2 - 3x^4 + 6x - 3$ 的曲线图。

自变量x的取值范围为：(-20, 20, 10)

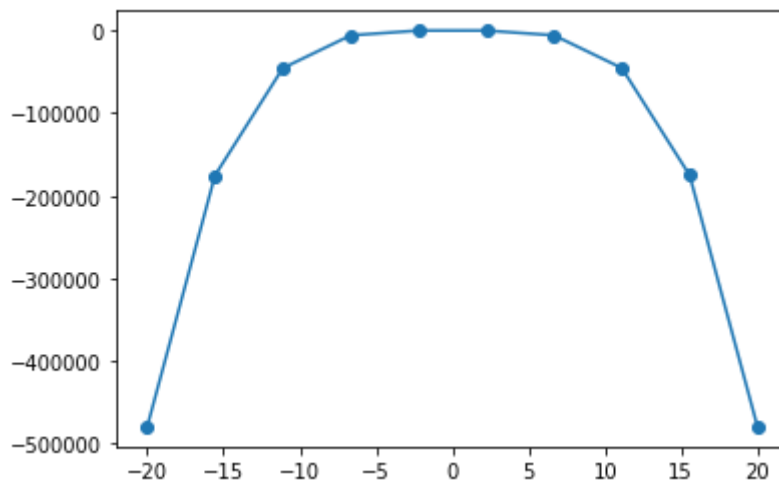
对于取值范围有两种理解：

- np.arange(-20, 20, 10) 起点-20, 终点20, 步长为10
- np.linspace(-20, 20, 10) 起点-20, 终点20, 生成10个元素的等差数列

```

1 # 通过inline指令, 实现在Jupyter中的实时绘图功能
2 %matplotlib inline
3
4 # 1. 引入库
5 # 使用import关键字引入matplotlib库, 为了简便使用缩写 "plt"来表示matplotlib库。
6 import matplotlib.pyplot as plt
7 import numpy as np
8
9 # 2. 指定坐标轴 (x,y值)
10 # 使用linspace()函数生成一个-20到20, 元素个数为10的等差数列。
11 # 令数列中的值为 x, 并根据表达式计算对应的 y值。
12 x = np.linspace(-20, 20, 10)
13 y = 2*x**2 - 3*x**4 + 6*x - 3
14
15 # 3. 把x,y的关系画到坐标轴上 plot(横坐标, 纵坐标, 标记, 颜色。。。)
16 #使用plot()函数绘制出曲线图
17 plt.plot(x, y, marker = "o")
18 plt.show()
19

```



## 2. 生成直方图

直方图应包含6个序列, 每个序列3种类型的特征, 取值如下:

```

1 x_data = ('China', 'America', 'Japan', 'Germany', 'France', 'Italy')
2 y1_data = (120, 325, 310, 235, 227, 256)
3 y2_data = (225, 312, 314, 221, 253, 341)
4 y3_data = (232, 332, 222, 241, 190, 299)

```

```

1 # 通过inline指令, 实现在Jupyter中的实时绘图功能
2 %matplotlib inline
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 plt.figure(1)
7 x_index = np.arange(6) #柱的索引
8 x_data = ('China', 'America', 'Japan', 'Germany', 'France', 'Italy')
9 y1_data = (120, 325, 310, 235, 227, 256)
10 y2_data = (225, 312, 314, 221, 253, 341)

```

```

11 y3_data = (232, 332, 222, 241, 190, 299)
12 bar_width = 0.2 #定义一个数字代表每个独立柱的宽度
13
14 # 使用 bar()函数定义柱状图的各个参数，依次包括：左偏移、高度、柱宽、透明度、颜色、图例
15 # 关于左偏移，不用关心每根柱的中心不中心，因为只要把刻度线设置在柱的中间就可以了
16 rects1 = plt.bar(x_index, y1_data, width=bar_width,alpha=0.4,
17 color='b',label='legend1')
18 rects2 = plt.bar(x_index + bar_width, y2_data,
19 width=bar_width,alpha=0.5,color='r',label='legend2')
20 # 使用 xticks() 函数设置x轴的刻度线
21 plt.xticks(x_index + bar_width/2, x_data)
22 plt.legend() #显示图例
23 plt.show()

```

