Preperation For The Final Test

xaero

Reviewing Material

信息技术学考导引试题详解

First Edition

浙江省杭州高级中學

目录

Sec 0.1 数据处理与应用

1. 答案：D。考查数据整理方法与目的。
2. 答案：C。
   A. 实验误差是测量值和真实值之间的偏差，不是虚假信息。
   B. 正确。
   C. 同一个信息对于不同的人价值可能不一样。
   D. 信息是信号、消息中所包含的含义，必须依附与数字、文字、图形、图像等载体。
3. 答案：B。
   A. 互联网上只有已数字化的信息，没有数字化当然查不到。
   B. 知识的获得是人利用自身已有的知识对信息进行加工，进而将新的信息纳入自己的知识结构的过程。检索到也只是看到，并不一定已内化成自己的知识。
   C. 天才也要记单词啊。
   D. 正确。
4. 答案：B。考查 pandas 数据处理 drop() 函数、groupby() 函数功能的理解。注释如下：

```
import pandas as pd                                  # 导入并使用 pd 作为别名
df = pd.read_csv("mnxk.csv", sep=",")                # 读取数据
df1 = df.drop("已选科目数", axis=1)                   # 删除“已选科目数”列
print(df.head())                                     # 打印 df 的前 5 行
print(df1.head())                                    # 打印 df1 的前 5 行
sc=df1.groupby("班级", as_index=False).count()       # 按“班级”分组
m = len(df)                                          # df 的行数
n = len(df1)                                         # df1 的行数
print(sc)                                            # 分组后的数据
```

注意 pandas 的很多操作处理后原始数据都不会改变。比如，第 3 行 drop() 函数删除了“已选科目数”列，参数“axis=1”指明了这是列而不是行。该函数调用后，产生了一个新的数据集合并赋值给对象 df1，而原始的数据集合 df 未曾变化。选项 B 就考查了 pandas 数据处理的这个特点：第 4 行打印的结果是原始数据的前 5 行，包含“已选科目数”这列数据，而第 5 行的输出的 5 行数据虽然大部分与前面相同，但不含“已选科目数”这列数据。选项 C 考查的是 df 和 df1 数据对象的行数是否相同，由于没有删除行，行数必然是一样的。选项 D，第 6 行的分组可以让相同班级的数据合并成一行数据，这个数据的每列数据是原先该列数据的非空单元格个个数（即 count() 函数的功能）。如，若原始数据如左侧所示，则执行第 6 行分组语句后的结果如右侧所示。在右侧数据中，“1 班”的“Name”值是 4，表示原始数据中 1 班“Name”列数据非空单元格个有 4 个；“1 班”的“物理”值是 2，表示原始数据中 1 班“物理”列数据非空单元格个有 2 个（相当于 1 班有两个 2 人选了物理）。

	班级	Name	物理	历史	技术	化学
0	1班	张三丰	1	1	1	
1	2班	郭靖	1		1	
2	1班	小龙女	1			1
3	2班	李秋水				
4	3班	杨过	1		1	1
5	1班	令狐冲				

	班级	Name	物理	历史	技术	化学
0	1班	4	2	1	2	2
1	2班	2	1	0	1	0
2	3班	2	2	1	1	2

```
6  3班  任我行   1   1       1
7  1班   黄蓉            1   1
```

5. 答案：D。
   A. 若化成十进制计算：$10H = 16D, 10B = 2D, 16D + 2D = 18D$。
   B. $1AH + 2AH = 44H$，注意十六进制下 $A + A$ 等于 14。
   C. 正确。
   D. 若化成十六进制计算：$10D + 10B = AH + 2H = CH$，即十六进制的值是 C。
6. 答案：B。
   A. 。
   B. 。
   C. 。
   D. 。
7. 答案：B。
   A. 。
   B. 。
   C. 。
   D. 。
8. 答案：D。
   A. 。
   B. 。
   C. 。
   D. 。
9. 答案：C。
   A. 。
   B. 。
   C. 。
   D. 。
10. 考查 pandas 数据处理与应用。
    (1) 考查数据处理的实际用途，帮助理解题目情景。
    (2) 考查 pandas 数据格式的识别。

```
def s_review(c):
    for r in range(df.shape[0]):              # 批阅 1 个单选题
        if df.at[r, qnum[c]] ==  ①  :
            tmp = 3
            df.at[r, score[c-2]] = tmp
            df.at[r, score[10]] += tmp        # 计算总分，存入"sum" 列
qnum = df.columns
sans = "BDCABDDBCB"                           # 本次作业的标准答案
score=["sc1","sc2","sc3","sc4","sc5","sc6","sc7","sc8","sc9","sc10","sum"]
for c in score:
    df[c] = 0
for c in range(2,12):                         # 逐题批阅
```

```
                    ②
print(df)
df.to_excel("客观题成绩.xlsx", index=False)  # 保存结果
```

(3) 解题过程:

(*a*) 从第 7 行主程序开始阅读，对 pandas 程序阅读，一定要直到变量保存的数据是什么？数据的结构是怎样的?

(*b*) 第 7 行 qnum 保存了数据对象 df 的所有列名称，即 `["name", "snum", "ans1", "ans2", ..., "ans10"]`。第 10 行的循环，结合第 9 行 score 列表中的数据可以知道第 11 行在数据对象 df 中新增了很多数据列, 列名称分别是“sc1”、“sc2”、“sc3”、……、“sum”，每列的值都是 0。这也是 pandas 的特点，数据列直接可以参与算术运算、关系运算和赋值操作，每种操作都可以将该列的所有行都进行相应处理。

(*c*) 第 ② 空处所进行的循环是逐题批阅。原始表格数据中一题就是一列数据，列序号是 2 ~ 12，刚好能对上这里的循环范围。因此，第 12 行的循环变量 *c* 相当于列序号——不过，pandas 需要的是列名称，这就需要 qnum 中对应的列名称来引用原始数据了。这里需要调用第一行的 s_review() 函数。

(*d*) 阅读 s_review() 函数。第 2 行 df.shape 可以返回数据对象 df 的维度“形状”：行数 (df.shape[0]) 和列数 (df.shape[1]), 因此 *r* 就是行索引号。由 `df.at[r, qnum[c]]`操作可知 qnum[c] 必然是列名称，结合前面的分析可以知道 *c* 必然是列序号。由于列表 qnum 中索引 2 号的列名称才是第一题名称“ans1”，因此 *c* 期望的值也应该从 2 开始。那么第 ② 空的函数调用就好办了：函数名已知的，参数作用也推知了，所以答案应该是 `s_revieww(c)`，就让 *c* 的值从 2 开始传递、调用函数。另外从 s_review() 函数的结构上看，它有 return 语句返回值，所以这空也无需考虑赋值——直接调用即可。

(*e*) 再回到第 3 行程序， `df.at[r, qnum[c]]`取得了 *c* 列每个人填写的答案，它们需要与标准答案做比较，标准答案保存再 sans 字符串中，它的索引号是从 0 开始的，所以第 ① 空的答案是 `sans[c-2]`。

(*f*) 第 5 行的程序是将该行（第 *r* 行）对应的得分列赋值为 tmp 分分值（如“ans1”列对应的分值是“sc1”列）。第 6 行的程序是将该分值累加到它的总分中去（即“sum”列，它的值是 10 个选择题的得分累加而来）。

11. 考查文本数据处理、分词、字符串统计与字典的应用。

```
import jieba                                          # 导入 jieba 模块
import pandas as pd
text = open("news.txt", encoding="utf-8").read() # 打开文本文件
words = jieba.lcut(text, cut_all=False)           # 分词
counts = {}
for name in words:
    if len(name) != 1 and not ("a" < name[0] < "z") and not ("0" <
       name[0] < "9"):
        if name in counts:
            counts[name] += 1                         # 词语已出现过
        else:
            counts[name] = 1                          # 词语第一次出现
```

```
# 字典转化为 DataFrame 格式存储
df = pd.DataFrame(list(counts.items()), columns=["词", "次数"])
df = df.sort_values("次数", ascending=False)      # 按“次数”降序排序
print(df)
```

(1) jieba 是目前常用的分词模块，它是一个基于词典分词的模块。模块导入后，程序再第 3 行通过 python 的内置 open() 函数打开了文本文件，read() 函数可以读取文件中的所有数据。第 4 行调用了 jieba 的 lcut() 函数进行分词，函数名中的“l”表示分词结果数据是一个列表（即 list，这里了解即可，无需记忆），函数的 cut_all 参数设定为 False 表示是精准分词，不会分隔“词中词”，当该参数设定为 True 时表示全模式分词，会分隔所有词。如“中华人民共和国”，False 模式下结果是一个词 `["中华人民共和国"]`，True 模式下会有多个词 `["中华", "人民", "共和国", "中华人民共和国"]`。对于 words 列表中的每个词，第 7 行程序过滤掉了单字、字母开头的、数字开头的字符串，因此答案选 C。

(2) 本小题考查 jieba 分词的规则特点，因为它是用现有的词典进行分词的，因此想要添加一个新词时,只需在分词前添加该词再进行分词即可。具体可以通过 `jieba.add_word("公益活动")`来添加该词。

(3) 第 8 行程序先判定某个单词 name 是否在字典 counts 的键名中出现过，如果出现过，则直接根据该键名取出其键值，然后加 1 后仍然存放在该键名上。else 分支就是该键名第一次出现，该键值初始为 1，答案是 `counts[name] = 1`。

12. 考查 pandas 数据处理与应用，matplotlib 数据可视化。给出三行数据示例如下：

```
109, 2007-02-20 00:07:10, 121.443100, 31.273000,  0, 45, 0
109, 2007-02-20 00:08:06, 121.447600, 31.272000,  6, 22, 1
109, 2007-02-20 00:09:07, 121.452500, 31.271000, 46, 67, 1
```

```
import pandas as pd
import matplotlib.pyplot as plt                # 导入 matplotlib 模块
plt.rcParams["font.sans-serif"] = ["KaiTi"] # 图表中中文以楷体显示
df = pd.read_csv("Taxi_105.txt", sep=",")
df.columns = ["xh", "sk", "jd", "wd", "sd", "jj", "zkzt"]
df = _____
```

(1) 第 3 行程序设置字体以便显示中文，这行代码了解即可。第 4 行程序读取了 csv 文件并转成 DataFrame 数据对象保存再 df 中。由于原始数据中没有标题行，程序的第 5 行指定了数据各列的标题。从处理结果上看，“速度”、“夹角”列都可以删除，因此删除这两列数据都可以。drop() 函数删除列的语法是 `df.drop("jj", axis=1)`，其中 axis 参数为 1 指明了删除的是列。

```
def pickup(rid):
    # 计算每次上客的停车时长，代码略
    return t, id1    # 返回停车时长、上客停车时的数据行索引
# 计算该出租车当日载客次数
ty = []; px = []; py = []; count = 0
n = len(df)
for row in range(n-1):
    if df.at[row, "zkzt"]==0 and df.at[row+1,"zkzt"]==1:     # 上客
        pt, id = pickup(row)
        ty.append(pt)
```

```
17                ①
18          px.append(df.at[id, "jd"])
19          py.append(df.at[id, "wd"])
20  print("该出租车当日载客次数为:", count)
21  # 上客平均时长四舍五入取整
22  print("上客平均时长、最大及最小时长(单位:秒):",          ②          ,
        max(ty), min(ty))
```

(2) 第 12 行程序中 n 取得了 df 数据对象的行数，第 13 行循环了 $n-1$ 次，可以确定 row 就是数据对象 df 的行索引值。第 14 行的判定是说当前行载客状态是空，但是下一行的载客状态是有客则表示当前正在上客。由 pickup() 函数的注释和 return 语句可知，第 15 行中的变量 pt 保存了函数的第一个返回值停车时长，id 保存了第二个值上客时的数据行索引。因此下面几行程序也容易理解：列表变量 ty 保存了各次上客前停车时长，列表 px 保存了各次上车时的经度值，py 保存了各次上车时的纬度值。第 ① 空似乎不需要填什么程序。不过从循环结束后的第 20 行程序上看，这里输出了 count 的值，为载客次数，再考查它的初值是零，因此需要在循环内进行次数统计，于是第 ① 空答案是 `count += 1`。第 ② 空是输出上客平均时长，上车次数是 count，这就需要算出上车总时长（接到各个客人时需要空车停留、闲逛的总时长）。这里已经将各次上车时间保存在列表 ty 中，求和只需调用 sum() 函数即可，四舍五入保留整数可以使用 round() 函数，于是答案是 `round(suum(ty)/count)`。注意很多同学的答案写成了 mean(ty)。在 python 内置函数中，只有求和 sum() 函数，求最值 max()/min() 函数，没有求平均值 mean() 函数。在 pandas 中，数据框对象有 mean() 方法，但它的格式是 df.mean()，而且会求出 df 每一列的平均值，结果不是一个整数而是一个 Series。这里不能用该函数。

```
tx = range(count)
plt.figure(1)
plt.title("当日上客时长对比图(单位:秒)")
          ①              # 绘制图表分析当日搭载乘客的上客时长
plt.figure(2)
plt.title("当日上客地点分布图")
          ②              # 绘制图表分析当日搭载乘客的上客地点分布
plt.show()
```

(3) round(sum(ty) / count)