

レポート用紙

講義名 : 数値解析 1	年月日 : 2024 年 7 月 15 日(月)
学籍番号 : 99999999	氏名 : 幸谷 智紀

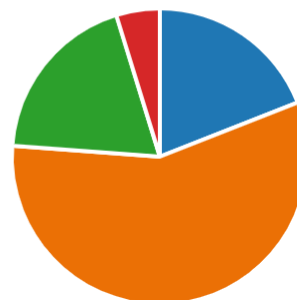
本日の課題 P.20 問題 3.2

感想

2. 課題の難易度はどうでしたか？

[詳細](#)

● 難しかった	4
● 少し難しかった	12
● ちょうど良かった	4
● 簡単だった	1
● とても簡単だった	0



解答例：

```
# student_grade.py: 生徒の成績統計処理 (問題 7.3)
import numpy as np # NumPy
import pandas as pd # Pandas
import matplotlib.pyplot as plt # Matplotlib
import japanize_matplotlib # Matplotlib 日本語対応
import os # os.path.dirname 関数使用

# 本スクリプト実行フォルダ名取得
pathname = os.path.dirname(__file__)

# Excel ファイル名: スクリプトと同じフォルダにある前提
filename = 'student_grade.xlsx'
fullpath_filename = pathname + '¥¥' + filename

# Excel シート名
sheetname = '学籍番号・氏名・タイトルなし'
# 教科名
subjects = ['国語', '英語', '数学', '理科', '社会']

# Excel ファイル読み込み確認
```

レポート用紙

```
try:
    pd.ExcelFile(fullpath_filename)
except:
    print(filename, 'が開けませんでした。')

# Excel ファイル読み込み時のみ動作
with pd.ExcelFile(fullpath_filename) as xls:
    sheet = pd.read_excel(xls, sheetname)
    # print(sheet)

# 科目ごとの平均点, 標準偏差, 中央値, 最高点, 最低点,
# -----
#     平均点, 標準偏差, 最高点, 最低点
# 国語 : 45.9,    30.5,    94,    2
# -----
print('     平均点, 標準偏差, 中央値, 最高点, 最低点')
for i in range(len(subjects)):
    data = sheet[subjects[i]]
    print(f'{subjects[i]:s}:                                {np.average(data):6.1f},
{np.median(data):8.1f},          {np.std(data):6.3g},          {np.amax(data):6d},
{np.amin(data):6d}')

# 科目ごとのトップ5抽出
student_no = sheet['学籍番号']
student_name = sheet['氏名']

# Figure オブジェクト(fig)と
# その上に Axes オブジェクト(ax[i])を
# 1行N列に生成し, サイズをN*5 x 5とする
no_subjects = len(subjects)

# -----
# 最高点グラフ
# -----
fig = plt.figure() # Figureのみ生成

# ソート用データ作成
ax = list[no_subjects]
```

レポート用紙

```
# 学籍番号, 氏名, 得点
for i in range(no_subjects):
    print('subject = ', subjects[i]) # 科目名表示
    subject_grade = sheet[subjects[i]] # 科目の得点
    #sort_data      =      [(student_no[index],      student_name[index],
subject_grade[index]) for index in range(len(student_no))]
    #sort_data.sort(key= lambda x: x[2], reverse = True) # 逆順得点でのソ
ーティング
    sort_data = [(student_name[index], subject_grade[index]) for index
in range(len(student_no))]
    sort_data.sort(key= lambda x: x[1], reverse = True) # 逆順得点でのソ
ーティング

    #print('sort_data  = ', sort_data)

    # トップ 5 抽出
    #x = [sort_data[index][0] + '¥n' + sort_data[index][1] for index in
range(5)] # 氏名
    #y = [sort_data[index][2] for index in range(5)] # 得点
    x = [sort_data[index][0] for index in range(5)] # 氏名
    y = [sort_data[index][1] for index in range(5)] # 得点
    print('x = ', x, ', y = ', y)

    ax = fig.add_subplot(1, no_subjects, i + 1)
    ax.set_title(subjects[i])
    if i == 0: ax.set_ylabel('得点') # 一番左のみ
    ax.set_ylim([0, 100]) # 最小値と最大値の設定
    ax.grid(axis = 'y') # y軸グリッドのみ描画
    ax.bar(x, y) # トップ 5 のみ棒グラフ
    ax.set_xticks(range(5), x, rotation='vertical')

# グラフ表示
plt.show()

# -----
# 最低点グラフ
# -----
fig = plt.figure() # Figureのみ生成
```

レポート用紙

```
# ソート用データ作成
ax = list[no_subjects]
# 学籍番号, 氏名, 得点
for i in range(no_subjects):
    print('subject = ', subjects[i]) # 科目名表示
    subject_grade = sheet[subjects[i]] # 科目の得点
    #sort_data      =      [(student_no[index],      student_name[index],
subject_grade[index]) for index in range(len(student_no))]
    #sort_data.sort(key= lambda x: x[2]) # 正順得点でのソート
    sort_data = [(student_name[index], subject_grade[index]) for index
in range(len(student_no))]
    sort_data.sort(key= lambda x: x[1]) # 正順得点でのソート

    #print('sort_data  = ', sort_data)

# トップ5抽出
#x = [sort_data[index][0] + '¥n' + sort_data[index][1] for index in
range(5)] # 氏名
#y = [sort_data[index][2] for index in range(5)] # 得点
x = [sort_data[index][0] for index in range(5)] # 氏名
y = [sort_data[index][1] for index in range(5)] # 得点
print('x = ', x, ', y = ', y)

ax = fig.add_subplot(1, no_subjects, i + 1)
ax.set_title(subjects[i])
if i == 0: ax.set_ylabel('得点') # 一番左のみ
ax.set_ylim([0, 100]) # 最小値と最大値の設定
ax.grid(axis = 'y') # y軸グリッドのみ描画
ax.bar(x, y) # トップ5のみ棒グラフ
ax.set_xticks(range(5), x, rotation='vertical')

# グラフ表示
plt.show()
```

実行結果 :

レポート用紙

