準備

ここをクリックして cs3-mid-is.zip をダウンロードし、展開・解凍すると、cs3-mid フォルダが生成する。必要に応じて cs3-mid-p1-sample.ipynb を使用してよい。

「小数第N位まで」を答える場合は、第(N+1)位を四捨五入すること。たとえば「13.79584…」を「小数第2位まで」で答える場合は「13.80」となる。

問1. 以下の問いに答えよ。(各2点)

mid-p1.csv を読み込んで一連の解析を行うノートブックの `___(1)___` などの空欄を埋め、また問いに答えよ。なお、提出するipynb/htmlファイルには、以下に表示されていないプログラム行が含まれていても構わない。

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

mid-p1.csv の全データをデータフレーム df に読み込む (1)。

```
csv_in = 'mid-p1.csv'
___(1)___
```

(1)

df の行数(データ数)と列数 (2)、各列のデータ型と非欠損値数 (3)、df の最初の5行を表示 (4)。
また、データ数を (5) に答えよ。

```
print( ___(2)___ )
print( ___(3)___ )
display( ___(4)___ )
# num of data: (5)
```

(2)

(3)

(4)

(5)

d1 列の最初の5行を表示 (6)。また、取り出した d1 列 (1次元データ) のデータ型を (7) に答えよ。

```
print( ___(6)___ )
# data type: (7)
```

(6)

(7)

(未選択)

d1 列の最小値を表示 (8)。また、その数値を (9) に答えよ。

```
print( ___(8)___ )
# value: (9)
```

(8)

(9)

各行を、d2 列の値の降順にソートして、先頭5行を表示 (10)。また、この列の2番目に大きい値を (11) に答えよ。

```
display( ___(10)___ )
# value: (11)
```

(10)

(11)

d3 列に出現する各値とそれぞれの出現回数の一覧表示 (12)。また、値 H の出現回数を (13) に答えよ。

```
print( ___(12)___ )
# value: (13)
```

(12)

(13)

df から d4 列を削除したデータフレーム df2 を作成 (14)。 df2 のデータを、d3 列の値でまとめ、d1 列と d2 列の最大値を表示 (15)。 d3 列の値が D のデータの、d1 列の最大値を (16) に答えよ。

```
___(14)___
display( ___(15)___ )
# value: (16)
```

(14)

(15)

(16)

df の各行について、d1 列と d2 列の値の和を求め、それを新たな d_tot 列に格納 (17)。 d_tot 列の先頭行の値を (18) に答えよ。

```
___(17)___
# value: (18)
```

(17)

(18)

df の数値列だけを取り出したデータフレーム df3 を作成する (19)。 df3 の全列の箱ひげ図を作成 (20)。 x軸ラベルは、df3 の列名と同じにする。また、軸ラベルやタイトル、凡例などの装飾はつけずに、1行で答えること。

```
___(19)___
___(20)___
plt.show()
```

(19)

(20)

ipynbファイルのアップロード:

ファイルをアップロード 未提出

htmlファイルのアップロード:

ファイルをアップロード 未提出

必要に応じて cs3-mid-p2-sample.ipynb を使用してよい。
「小数第N位まで」を答える場合は、第(N+1)位を四捨五入すること。たとえば「13.79584...」を「小数第2位まで」で答える場合は「13.80」となる。

問2. 以下の問いに答えよ。(各2点)

mid-p2-1.csv と mid-p2-2.csv を読み込んで一連の処理を行うノートブックの `___(1)___` などの空欄を埋め、また問いに答えよ。なお、提出するipynb/htmlファイルには、以下に表示されていないプログラム行が含まれていても構わない。

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

(mid-p2-1.csv の全データを **データフレーム s1**、mid-p2-2.csv の全データを **データフレーム s2** に読み込んであるとする)

s1 の重複行をすべて表示 (1)。
重複を削除して、行番号を0からの連番に振り直し、変数 s1d に代入 (2)。 (元の行番号を格納する列は生成させない)
s1d の行数(データ数)を答えよ (3)。

```
display( ___(1)___ )
___(2)___
# num: (3)
```

(1)

(2)

(3)

s1d の各列の欠損値の数を表示 (4)。h1 列の欠損値の数を (5) に答えよ。 s1d の欠損値を1つでも含む行を表示 (6)。
s1d の欠損値を1つでも含む行を削除して、行番号を0からの連番に振り直し、変数 s1d2 に代入 (7)。
s1d2 の h4 列の中の値 f をすべて fff に置換する (8)。

```
print( ___(4)___ )   # num: (5)
display( ___(6)___ )
___(7)___
___(8)___
```

(4)

(5)

(6)

(7)

(8)

(このあと、欠損値が含まれていたすべての列のデータ型を int型に変更する。
ある列のint型への変更は、`列 = 列.___(9)___` で実行できる。)

(9)

s1d2 の各行の右側に、s2 の対応する行を結合し、結果を変数 s3 に代入 (10)。このとき、s1d2 の h5 列の値が、s2 の alpha 列の値と一致する s2 の行を対応させるようにする。

```
___(10)___
```

(10)

s3 のデータフレームをCSVファイル mid-p2-out.csv に保存 (11)。なお、行番号をindex列に保存する必要はない。またencoding= など、その他のオプションの設定は不要。また、s3 の行数(データ数) と列数を (12) および (13) に答えよ。

```
___(11)___
# num: (12), (13)
```

(11)

(12)

(13)

ipynbファイルのアップロード:

ファイルをアップロード 未提出

htmlファイルのアップロード:

ファイルをアップロード 未提出

問3. 以下の問いに答えよ。(各2点)

mid-p3.csv を読み込んで一連の処理を行うノートブックの `___(1)___` などの空欄を埋め、また問いに答えよ。なお、提出するipynb/htmlファイルには、以下に表示されていないプログラム行が含まれていても構わない。

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

(mid-p3.csv の全データを **データフレーム ts** に読み込んであるとする)

ts の Date 列を、日付を扱うのに適したデータ型に変換して上書き (1)。

```
ts['Date'] = ___(1)___
```

(1)

ts の Date 列以外の各列について、オーバーラップしない1週間ごと (月曜から日曜まで) の平均値を求め、結果を変数 ts_w に格納 (2)(3)。
また、2022年04月25日～2022年05月01日の期間の X0 列の平均値 (小数第2位まで) を (4) に答えよ。

```
ts = ___(2)___
ts_w = ___(3)___
# value: (4)
```

(2)

(3)

(4)

以下の (5)～(8) はオプション問題です。

各行の曜日番号を求めて、ts に新たな dow 列を作って格納 (5)。
dow 列の値をもとに、各曜日の平均値を求め、結果を 変数 ts_wday_ave に代入 (6)。
ts_wday_ave の X0列を棒グラフに描画。ただし、x軸は wd を用いて 'Mon', 'Tue', ... とする (7)。また、火曜日の X1 列の平均値 (小数第2位まで) を (8) に答えよ。

```
___(5)___
___(6)___
wd = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
___(7)___
# value: (8)
```

(5)

(6)

(7)

(8)

ipynbファイルのアップロード:

ファイルをアップロード 未提出

htmlファイルのアップロード:

ファイルをアップロード 未提出

問4. 以下の問いに答えよ。(4-2-3のみ2点, 他は1点)

4-1. 次の文の空欄をもっとも適切な語で埋めよ。

KMeans法は [(1)] 型クラスタリングの方法の一つである。重心法と比べると、一般に計算量は [(2)]。また、一般に、複数回実行すると結果は毎回 [(3)]。

(1)

(2)

(未選択)

(3)

(未選択)

データの標準化を行ったあと、データの平均の値は [(4)]、データの母標準偏差の値は [(5)] となる。

(4)

(5)

4-2. 次のKMeans法のプログラムの一部について、以下の問いに答えよ。

必要なライブラリがimportされ、また各列が標準化されたデータがデータフレーム df に格納されているとする。
また、あらかじめエルボー法で、適切なクラスタ数が求まっているとする。なお、左端の番号列はプログラムの行番号であり、プログラムの一部ではない。

```
 1    n = 5
 2    rs = 33
 3    kmcls = KMeans(n_clusters=n, random_state=rs)
 4    kmout = kmcls.fit_predict(df)
 5    df['km_out'] = kmout
 6    for t in range(n):
 7        df1 = ___(6)___
 8        display(df1)
 9    plt.scatter(df['x1'], df['x2'], marker='o', c=df['km_out'])
10    plt.xlabel('x1')
11    plt.ylabel('x2')
12    plt.colorbar()
13    plt.show()
```

4-2-1. 指定しているクラスタ数を数値で答えよ。

4-2-2. クラスタリングを実行して、結果を変数に代入している行の行番号を答えよ。

(未選択)

4-2-3. 行番号7では、データフレーム df から、df の km_out 列の値が変数 t と一致するデータだけを取り出して、変数 df1 に代入している。空欄 (6) を埋めよ。

4-2-4. 散布図の点の色がどの値によって決まっているか答えよ。

(未選択)