

統計検定「データサイエンスエキスパート」サンプル問題

2022年3月

問題1

次の表1に示すテーブル(テーブル名:「成績」)をもとに、データベース上でSQLを使って成績分析を行うタスクを考える。[1]~[4]の各問いに答えよ。

表1:「成績」

学生番号	氏名	性別	国語	英語	数学	理科	社会	合計
1	AAAAA	女	42	56	47	54	42	241
2	BBBBB	男	53	22	30	70	55	230
3	CCCCC	女	36	59	13	93	45	246
4	DDDDD	男	44	26	52	96	58	276
5	EEEEE	女	62	75	9	92	16	254
6	FFFFF	男	53	25	12	62	79	231
7	GGGGG	男	26	5	53	91	44	219
8	HHHHH	男	41	72	25	100	69	307
9	IIIII	女	60	52	24	44	62	242
10	JJJJJ	女	31	58	33	38	45	205

[1] 表1に示すテーブル「成績」から国語と英語の男女別平均点を得るためのSQL文として、次の①~⑤のうちから適切なものを一つ選べ。

- ① SELECT 性別, SUM(国語)/COUNT(国語) AS 国語平均点, SUM(英語)/COUNT(英語) AS 英語平均点 FROM 成績 ORDER BY 性別;
- ② SELECT 性別, AVG(国語) AS 国語平均点, AVG(英語) AS 英語平均点 FROM 成績 ORDER BY 性別;
- ③ SELECT 性別, AVG(国語) AS 国語平均点, AVG(英語) AS 英語平均点 FROM 成績 GROUP BY 性別;
- ④ SELECT 性別, AVG(国語) AS 国語平均点, AVG(英語) AS 英語平均点 FROM 成績 WHERE 性別 IN (SELECT 性別 FROM 成績 GROUP BY 性別);
- ⑤ SELECT 性別, SUM(国語)/COUNT(国語) AS 国語平均点, SUM(英語)/COUNT(英語) AS 英語平均点 FROM 成績 WHERE 性別 = '男性' OR 性別 = '女性';

[2] 性別ごとの5科目（国語，英語，数学，理科，社会）の平均値が表2のように求められた。このとき英語の平均点に性別で差があると言えるかについて，5%有意水準で両側検定を行う際の手順として，下の①～⑤の記述のうちから最も適切なものを一つ選べ。

表2：性別ごとの5科目平均値

性別	国語	英語	数学	理科	社会
男性	43.4	30.0	34.4	84.8	61.0
女性	46.2	60.0	25.2	64.2	42.0

- ① 等分散性を前提とするスチューデントの2標本t検定を用いればよいので，2標本両側t検定のp値を求めたところ0.03506を得た。このため，5%有意水準で帰無仮説「2群での平均値は等しい」を棄却し，英語平均点は性別で異なるという結論を得た。
- ② 女性の平均点が男性の平均点よりかなり高いことに注目し，等分散性を前提とするスチューデントの2標本t検定の片側p値を計算したところ0.01753を得た。このため，5%有意水準で帰無仮説「2群での平均値は等しい」は棄却し，英語平均点は性別で異なるという結論を得た。
- ③ 性別ごとの英語得点の分散に差がないことを5%有意水準でF検定を用いて検定したところ，p値が0.06793となったため，2群の分散に差があるとは言えない。そこで，2群は等分散であると仮定した上で，両側2標本スチューデントのt検定を用いてp値を求めたところ0.03506を得たため，5%有意水準で帰無仮説「2群での平均値は等しい」を棄却し，英語平均点は性別で異なるという結論を得た。
- ④ 5名の男女の受験者にそれぞれ対応があると考えて，データ間に対応のある2標本スチューデントの2標本t検定(自由度4)を用いてp値を求めたところ0.05966を得た。このため，5%有意水準で帰無仮説「2群での平均値は等しい」は棄却されないことから，英語平均点は性別で異なるとは言えないという結論とした。
- ⑤ 女性の平均点が男性の平均点よりかなり高いことから，Welchのt検定の片側p値である0.02625を用いることとした。このため，5%有意水準で帰無仮説「2群での平均値は等しい」を棄却し，英語平均点は性別で異なるという結論を得た。

[3] 次のPythonのコードを使って，合計得点に関する階級幅20点の度数分布表を計算することにした。

```
def funcx(vals):  
    x = {}  
    for i in range(25):  
        x[i*【ア】]=0  
    for v in vals:
```

```
x[int(v/【ア】)*【ア】] += 【イ】  
return(x)
```

```
total=[241, 230, 246, 276, 254, 231, 219, 307, 242, 205]  
print(funcx(total))
```

出力

```
{0: 0, 20: 0, 40: 0, 60: 0, 80: 0, 100: 0, 120: 0, 140: 0, 160: 0,  
180: 0, 200: 2, 220: 2, 240: 4, 260: 1, 280: 0, 300: 1, 320: 0, 340:  
0, 360: 0, 380: 0, 400: 0, 420: 0, 440: 0, 460: 0, 480: 0}
```

コード中の【ア】、【イ】に入る数値の組合せとして、次の①～⑤のうちから最も適切なもの一つ選べ。

- ① 【ア】 25 【イ】 25
- ② 【ア】 20 【イ】 25
- ③ 【ア】 1 【イ】 25
- ④ 【ア】 25 【イ】 20
- ⑤ 【ア】 20 【イ】 1

[4] この試験において、合計点で上位 20%の者を特別に選抜したい。選抜される者を識別するための境界となる点として、[4-1]の①～⑤のうちから最も適切なもの一つを選び、選抜される者の氏名と合計得点を出力する SQL 文として、[4-2]の①～⑤のうちから最も適切なもの一つ選べ。

[4-1]

- ① 215 点
- ② 235 点
- ③ 250 点
- ④ 260 点
- ⑤ 280 点

[4-2]

- ① SELECT 氏名, 合計 FROM 成績 ORDER BY 合計 DESC LIMIT 2;
- ② SELECT 氏名, 合計 FROM 成績 ORDER BY 合計 LIMIT 2;
- ③ SELECT 氏名, 合計 FROM 成績 WHERE 合計 IN (SELECT 合計 FROM 成績 ORDER

BY 合計 DESC) LIMIT 2;

④ SELECT 氏名, 合計 FROM 成績 ORDER BY 合計 WHERE 合計 > (SELECT
0.8*MAX(合計) FROM 成績) ;

⑤ SELECT 氏名, 合計 FROM 成績 GROUP BY 合計 ORDER BY 合計 DESC;

問題 2

1 カ月当たりの平均合計労働時間 x (時間) と身体的不調の有無 y (0:ない/1:ある)に関するデータを、ある事業所の従業員 100 名からアンケートフォームへ回答してもらうことにより収集した。このデータを分析することにより、合計労働時間 x と身体的不調 y との関係をモデル化したい。[1]～[4]の各問いに答えよ。

[1] x_i を従業員 i の労働時間、 y_i を従業員 i の身体的不調を表す 0 または 1 の二値変数とし、サイズ n のデータを $(x_1, y_1), \dots, (x_n, y_n)$ とする。この時、ロジスティック関数 $F(x) = \exp(x)/(1 + \exp(x))$ を用いて、 x を与えた時の Y の条件つき確率を

$$\Pr[Y = 1|x] = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)} = F(\alpha + \beta x), \quad \Pr[Y = 0|x] = 1 - F(\alpha + \beta x)$$

とモデル化し、最尤法によりパラメータ α, β を推定するロジスティック回帰を実行したい。パラメータを推定するための対数尤度関数 $l(\alpha, \beta)$ として、次の①～⑤のうちから適切なものを一つ選べ。

① $l(\alpha, \beta) = \sum_{i=1}^n \{y_i \log F(\alpha + \beta x_i) + (1 - y_i) \log(1 - F(\alpha + \beta x_i))\}$

② $l(\alpha, \beta) = \sum_{i=1}^n \{y_i F(\alpha + \beta x_i) + (1 - y_i)(1 - F(\alpha + \beta x_i))\}$

③ $l(\alpha, \beta) = \sum_{i=1}^n \{y_i \log F(\alpha + \beta x_i) + (1 - y_i)(1 - \log F(\alpha + \beta x_i))\}$

④ $l(\alpha, \beta) = \sum_{i=1}^n \{y_i \log(1 - F(\alpha + \beta x_i)) + (1 - y_i) \log F(\alpha + \beta x_i)\}$

⑤ $l(\alpha, \beta) = \sum_{i=1}^n \{y_i(1 - F(\alpha + \beta x_i)) + (1 - y_i)F(\alpha + \beta x_i)\}$

[2] 次の文章は、対数尤度関数 $l(\alpha, \beta)$ に対する最尤法に関する記述である。文章中の【ア】、【イ】、【ウ】に入る語句の組合せとして、下の①～⑤のうちから適切なものを一つ選べ。

上述の対数尤度関数 $l(\alpha, \beta)$ を最大化

$$(\hat{\alpha}, \hat{\beta}) = \arg \max_{\alpha, \beta} l(\alpha, \beta)$$

することによりパラメータ α, β を決定する方法を最尤法とよぶ。一般に、最尤法によりデータから求められる適切なパラメータ α と β は、次の【ア】の解として求められる。

$$\frac{\partial l}{\partial \alpha} = 0, \quad \frac{\partial l}{\partial \beta} = 0$$

数値的にこの問題を解くためには、ニュートン・ラフソン法として得られる次の漸化式

$$\begin{bmatrix} \alpha_{k+1} \\ \beta_{k+1} \end{bmatrix} = \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} - \begin{bmatrix} \frac{\partial^2 l}{\partial \alpha^2} & \frac{\partial^2 l}{\partial \alpha \partial \beta} \\ \frac{\partial^2 l}{\partial \alpha \partial \beta} & \frac{\partial^2 l}{\partial \beta^2} \end{bmatrix}_{\alpha=\alpha_k, \beta=\beta_k}^{-1} \begin{bmatrix} \frac{\partial l}{\partial \alpha} \\ \frac{\partial l}{\partial \beta} \end{bmatrix}_{\alpha=\alpha_k, \beta=\beta_k}$$

を利用して、適当な初期値 (α_0, β_0) から、漸化式を繰り返し計算することで、 (α_k, β_k) の収束値として解 $(\hat{\alpha}, \hat{\beta})$ を近似的に求めることができる。ここで、行列

$$\begin{bmatrix} \frac{\partial^2 l}{\partial \alpha^2} & \frac{\partial^2 l}{\partial \alpha \partial \beta} \\ \frac{\partial^2 l}{\partial \alpha \partial \beta} & \frac{\partial^2 l}{\partial \beta^2} \end{bmatrix}$$

は対数尤度関数の【イ】であり、ベクトル

$$\begin{bmatrix} \frac{\partial l}{\partial \alpha} \\ \frac{\partial l}{\partial \beta} \end{bmatrix}$$

を【ウ】と呼ぶ。

- | | | |
|----------------|-----------|-----------|
| ① 【ア】 尤度方程式 | 【イ】 ヤコビ行列 | 【ウ】 スコア関数 |
| ② 【ア】 正規方程式 | 【イ】 ヘッセ行列 | 【ウ】 特性関数 |
| ③ 【ア】 ニュートン方程式 | 【イ】 ヤコビ行列 | 【ウ】 汎関数 |
| ④ 【ア】 オイラー方程式 | 【イ】 回転行列 | 【ウ】 強度関数 |
| ⑤ 【ア】 尤度方程式 | 【イ】 ヘッセ行列 | 【ウ】 スコア関数 |

[3] ロジスティック回帰の最尤法をニュートン・ラフソン法により実行するために必要となる次の方程式の【ア】、【イ】、【ウ】に入る項として、下の①～⑤のうちから適切なもの一つ選べ。

$$\frac{\partial l}{\partial \alpha} = \sum_{i=1}^n \left\{ y_i \frac{\text{【ア】}}{1 + \exp(\alpha + \beta x_i)} - (1 - y_i) \frac{\text{【ウ】}}{1 + \exp(\alpha + \beta x_i)} \right\}$$

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^n \left\{ y_i \frac{\text{【イ】}}{1 + \exp(\alpha + \beta x_i)} - (1 - y_i) \frac{x_i \exp(\alpha + \beta x_i)}{1 + \exp(\alpha + \beta x_i)} \right\}$$

$$\frac{\partial^2 l}{\partial \alpha^2} = - \sum_{i=1}^n \frac{\text{【ウ】}}{(1 + \exp(\alpha + \beta x_i))^2}$$

$$\frac{\partial^2 l}{\partial \alpha \partial \beta} = - \sum_{i=1}^n \frac{x_i \exp(\alpha + \beta x_i)}{(1 + \exp(\alpha + \beta x_i))^2}$$

$$\frac{\partial^2 l}{\partial \beta^2} = - \sum_{i=1}^n \frac{x_i^2 \exp(\alpha + \beta x_i)}{(1 + \exp(\alpha + \beta x_i))^2}$$

- | | | | |
|---|------------------------------------|-----------|------------------------------------|
| ① | 【ア】 1 | 【イ】 x_i | 【ウ】 $x_i \exp(\alpha + \beta x_i)$ |
| ② | 【ア】 1 | 【イ】 x_i | 【ウ】 $\exp(\alpha + \beta x_i)$ |
| ③ | 【ア】 x_i | 【イ】 1 | 【ウ】 x_i^2 |
| ④ | 【ア】 $x_i \exp(\alpha + \beta x_i)$ | 【イ】 x_i | 【ウ】 1 |
| ⑤ | 【ア】 x_i^2 | 【イ】 1 | 【ウ】 $\exp(\alpha + \beta x_i)$ |

[4] ある事業所から集めた、1 カ月当たりの平均合計労働時間 x (時間)と身体的不調 y (0:ない/1:ある)とに関する 100 人分のデータを `sampledata.csv` と名前を付けて UTF-8-BOM 形式のファイルに格納した。1 列目に平均労働時間, 2 列目に身体的不調の有無を 0 または 1 で表記したデータをコンマ区切り形式で作成した。このデータを使ってロジスティック回帰のパラメータ α, β を計算するためのコンピュータプログラムを Python により下のコード 1 として実装した。コード 1 の【ア】、【イ】、【ウ】に入る関数として、次の①～⑤のうちから適切なものを 1 つ選べ。

- | | | | |
|---|----------------------|----------------------|----------------------|
| ① | 【ア】 <code>laa</code> | 【イ】 <code>lab</code> | 【ウ】 <code>lbb</code> |
| ② | 【ア】 <code>lab</code> | 【イ】 <code>lab</code> | 【ウ】 <code>lab</code> |
| ③ | 【ア】 <code>laa</code> | 【イ】 <code>lbb</code> | 【ウ】 <code>lab</code> |
| ④ | 【ア】 <code>la</code> | 【イ】 <code>lb</code> | 【ウ】 <code>lab</code> |
| ⑤ | 【ア】 <code>lb</code> | 【イ】 <code>la</code> | 【ウ】 <code>lab</code> |

ただし、 n, x, y, a, b を引数とする関数 `la, lb, laa, lab, lbb` はデータ $(x_1, y_1), \dots, (x_n, y_n)$ を配列 `x, y` に設定したときに、パラメータ $\alpha=a, \beta=b$ として、それぞれ、対数尤度関数 $l(\alpha, \beta)$

の 1 階偏微分 $\frac{\partial l}{\partial \alpha}, \frac{\partial l}{\partial \beta}$ の値と 2 階偏微分 $\frac{\partial^2 l}{\partial \alpha^2}, \frac{\partial^2 l}{\partial \alpha \partial \beta}, \frac{\partial^2 l}{\partial \beta^2}$ の値を求める関数であるとする。

コード 1

```
import math
# functions
def nextalpha(n, x, y, a, b):
    delta = laa(n, x, y, a, b) * lbb(n, x, y, a, b) - 【ア】 (n, x, y, a, b) ** 2 # 行列式
    a -= (lbb(n, x, y, a, b) * la(n, x, y, a, b) - 【イ】 (n, x, y, a, b) * lb(n, x, y, a, b)) / delta
    return(a)

def nextbeta(n, x, y, a, b):
    delta = laa(n, x, y, a, b) * lbb(n, x, y, a, b) - 【ア】 (n, x, y, a, b) ** 2 # 行列式
```

```

    b -= (laa(n,x,y,a,b)*lb(n,x,y,a,b) - 【ウ】(n,x,y,a,b)*la(n,x,y,a,b))/delta
    return(b)

# init
alpha0 = 1.0
beta0 = 0.1

# data
n = 0
x = []
y = []
f = open("sampledata.csv","r",encoding="utf_8_sig")
for line in f:
    dd = line.strip()
    data = dd.split(',')
    x += [float(data[0])]
    y += [int(data[1])]
    n += 1

# calculation
alpha = alpha0 # init
beta = beta0 # init
nalpaha = nextalpha(n,x,y,alpha,beta)
nbeta = nextbeta(n,x,y,alpha,beta)
err = math.sqrt((alpha-nalpaha)**2+(beta-nbeta)**2)
while(err > 1e-8):
    alpha = nalpaha
    beta = nbeta
    nalpaha = nextalpha(n,x,y,alpha,beta)
    nbeta = nextbeta(n,x,y,alpha,beta)
    err = math.sqrt((alpha-nalpaha)**2+(beta-nbeta)**2)
    print("alpha: %f, beta: %f, error = %20.20f" % (alpha,beta,err))
print("result = alpha: %f, beta: %f" % (alpha,beta))

```