

データサイエンス レポート課題

25G1026 大菅 海兎

選択した分析手法

回帰分析

例題

2024 年発売の PlayStation5 用フルプライスタイトル (8000 円から 10000 円) の Metacritic による評価と、大手中古販売会社のオンラインショップ価格を 100 分の 1 にし、四捨五入してまとめた。下記データについて中古販売価格と評価の回帰直線を求めよ。

表 1 運動前後での脈拍の測定結果

	A	B	C	D	E	F	G	H
販売価格	18	39	34	43	44	46	48	15
Meta スコア	82	87	94	92	86	94	81	79

解答

説明変数 x を中古販売価格、目的変数 y を Meta スコアとして単回帰分析を行う。回帰分析は、最小二乗法から求められる標本回帰直線を利用する。この直線は、式 1 のように表される。

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \quad (1)$$

まずは、各変数の平均値を求める。 x の平均値 \bar{x} は式 2 のように求められる。また、 y の平均値 \bar{y} は式 3 のように求められる。

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{287}{8} = 35.875 \quad (2)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i = \frac{695}{8} = 86.875 \quad (3)$$

次に、回帰係数 $\hat{\beta}_1$ を式 4 のように求める。

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{268.8750}{1134.8750} = 0.2369 \quad (4)$$

最後に、切片 $\hat{\beta}_0$ を式 5 のように求める.

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 86.875 - 0.2369 \times 35.875 = 78.4064 \quad (5)$$

よって、求める標本回帰直線は式 6 のようになる.

$$\hat{y} = 78.4064 + 0.2369x \quad (6)$$

式 6 から、中古販売価格が高いほど Meta スコアも高くなる傾向があるといえる.

ソースコード 1 プログラム例

```
1 # 1. データの準備
2     x = np.array([18, 39, 34, 43, 44, 46, 48, 15]) # 価格
3     y = np.array([82, 87, 94, 92, 86, 94, 81, 79]) # 評価
4     titles = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
5
6 # 2. 基本統計量の計算
7     n = len(x)
8     x_mean = np.mean(x)
9     y_mean = np.mean(y)
10
11 # 3. 偏差、偏差の二乗、偏差の積の計算 (途中経過用)
12     x_diff = x - x_mean # の偏差 x (x - x_bar)
13     y_diff = y - y_mean # の偏差 y (y - y_bar)
14     x_diff_sq = x_diff**2 # の偏差平方 x (x - x_bar)^2
15     xy_diff = x_diff * y_diff # 偏差の積 (x - x_bar)(y - y_bar)
16
17 # 4. 合計値の計算
18     sum_x_diff_sq = np.sum(x_diff_sq) # の偏差平方和 x 分母 ( )
19     sum_xy_diff = np.sum(xy_diff) # 偏差共分散分子 ( )
20
21 # 5. 回帰係数の算出
22     beta_1_hat = sum_xy_diff / sum_x_diff_sq
23     beta_0_hat = y_mean - (beta_1_hat * x_mean)
24     y_hat = beta_1_hat * x + beta_0_hat
25 # 6. 結果の表示
26     print(f"価格の平均: {x_mean:.4f}")
27     print(f"評価の平均: {y_mean:.4f}")
28     print(f"傾き: {beta_1_hat:.4f}")
29     print(f"切片: {beta_0_hat:.4f}")
30     print(f"最終的な回帰式: {y_hat} = {beta_0_hat:.4f} + {beta_1_hat:.4f} * x")
```
