Линейная регрессия. Задача 2

Ильичёв А.С., 693

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

1. Теоретическое введение.

Имеем задачу $X_i = \beta_1 + i\beta_2 + \varepsilon_0 + \cdots + \varepsilon_i$, $i = 0, 1, \ldots, n$, ε_i независимы и распределены по $N(0, \sigma^2)$. Эта задача сводится к линейной модели следующим образом:

$$X_0 = \beta_1 + \varepsilon_0$$

$$X_i - X_{i-1} = \beta_2 + \varepsilon_i, \;\; i = 1, \ldots, n$$

В этом случае целевая переменная (наблюдение) Y и матрица весов Z примут вид:

$$Y = \begin{pmatrix} X_0 \\ X_1 - X_0 \\ \vdots \\ X_n - X_{n-1} \end{pmatrix}, \;\; Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix}$$

(всего $n + 1$ строк).

2. Считаем данные и проведем их обработку.

```
df = pd.read_csv('Regression.csv', names=['x'])
df.head()
```

	x
0	63.5725
1	72.9531
2	83.6808
3	96.2717
4	103.2173

Создадим вектор Y, как описано выше.

```
Y = np.zeros(len(df))
Y = np.append(df.x.iloc[0],
              [df.x.iloc[i] - df.x.iloc[i - 1]
               for i in range(1, len(df))])
```

```
Y[0]
```

```
63.5725
```

```
np.mean(Y[1:])
```

```
9.967341441441443
```

Создадим матрицу Z. Здесь $m = n + 1$, где n - из условия.

```
m = len(df)
Z = np.concatenate(([[1, 0]],
                    np.column_stack((np.zeros(m - 1),
                                     np.ones(m - 1)))), axis=0)
```

3. Найдем оценки наименьших квадратов для β_1 и β_2, а также несмещенную оценку для σ^2.

Имеем $(\hat{\beta}_1, \hat{\beta}_2)^T = \hat{\theta} = (Z^T Z)^{-1} Z^T Y, \quad \hat{\sigma}^2 = \frac{1}{m-k} \|Y - Z\hat{\theta}\|$, где m = размер наблюдения, k - количество столбцов в Z.

```
k = Z.shape[1]
```

```
t = np.dot((np.linalg.inv(Z.T @ Z) @ Z.T), Y)
```

```
t
```

```
array([63.5725    ,  9.96734144])
```

```
sigma2 = np.sum((Y - Z @ t) ** 2) / (m - k)
sigma2
```

```
4.222797059623577
```

В теоретической задаче 8.2 были получены аналитические формулы для оцениваемых параметров:

$$\begin{pmatrix}\hat\beta_1\\ \hat\beta_2\end{pmatrix} = \begin{pmatrix} X_0 \\ \frac{1}{n}(X_n - X_0)\end{pmatrix}, \quad \hat\sigma^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n}(X_i - X_{i-1})^2 - \frac{1}{n}(X_n - X_0)^2\right).$$

Подставим в них наши значения.

```
n = m - 1
b1 = Y[0]
b1
```

```
63.5725
```

```
b2 = (df.x.iloc[n] - Y[0]) / n
b2
```

```
9.967341441441441
```

```
sigma2_theor = (np.sum(Y[1:] ** 2) - n * b2 ** 2) / (n - 1)
sigma2_theor
```

```
4.222797059623573
```

Очевидно, значения совпадают с полученными первым способом (в последнем разряде уже влияют ошибки округления).

4. Найдем оценку дисперсии отсчета времени.

По условию $\varepsilon_i = \varepsilon_i^t\beta_2$, откуда $D\varepsilon_i^t = \frac{D\varepsilon_i}{\beta_2^2}$. Такое же соотношение по теореме о наследовании сходимости верно для оценок: $\hat\sigma_t^2 = \frac{\hat\sigma^2}{\hat\beta_2^2}$

```
sigma2_t = sigma2 / b2 ** 2
sigma2_t
```

```
0.04250514862126407
```

5. Вывод

Оценка по методу наименьших квадратов в нашем случае совпадает с выборочным средним (ОМП). Можно сделать вывод, что данная задача хорошо описывается линейной

регрессионной моделью. Оценки, полученные численным расчетом через матрицы и аналитическим выводом, совпадают.