Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут прикладної математики та фундаментальних наук
Кафедра прикладної математики

ЗВІТ
про виконання лабораторної роботи №5
з курсу «Математична Статистика»

Виконав:
Студент групи ПМ-33
Маркевич Леонід
Прийняв:
Янішевський В.С.

Львів 2023

Лабораторна робота 5
Варіант 12

ТЕМА: Побудова лінійної багатофакторної моделі та дослідження її на адекватність.

МЕТА: засвоїти основні формули і вирази багатофакторної регресії та вміти застосовувати їх до практичних задач. За даними спостережень визначити лінійну багатофакторну регресію, навести оцінки.

ЗАВДАННЯ:
1. За даними таблиці здійснити якісний аналіз залежності між факторами.
2. Побудувати регресійну модель вигляду
$$y = b_0 + b_1 x_1 + b_2 x_2.$$
3. Перевірити модель на адекватність за F – критерієм Фішера.
4. Провести оцінку значимості параметрів рівняння регресії b_1, b_2 та визначити інтервали довіри для них.
5. Знайти довірчий інтервал для окремого значення залежної змінної.
6. Навести тлумачення параметрів b_0, b_1, b_2.
7. Навести висновки і оформити звіт роботи.

ХІД РОБОТИ:

Реалізація виконана на мові програмування *Python* з використанням бібліотек *numpy, pandas, statsmodels, random, matplotlib, scipy*.

1. За даними таблиці здійснити якісний аналіз залежності між факторами.

Програмна реалізація:

```
data = pd.read_csv('lab5/data.csv')

y = calc_data(1.5, data['y'])
x1 = calc_data(2.5, data['x1'])
x2 = calc_data(10, data['x2'])

X = np.column_stack((np.ones(len(x1)), x1, x2))
B = find_params(y, X)
U = find_param_u(y, X, B)
depend_analysis(data)
```

```python
def calc_data(a, i):
    N = random.uniform(0, 1)
    data = i + a * (N - 1/2)
    return data

def find_params(y, X):
    B = np.linalg.inv(X.T @ X) @ (X.T @ y)
    print('B: ', B)
    return B

def find_param_u(y, X, B):
    y_hat = X @ B
    residuals = y - y_hat
    print('Залишки: ', residuals)
    return residuals

def depend_analysis(data):
    correlation_matrix = data.corr()
    print("\nКореляційна матриця:")
    print(correlation_matrix)
```

Приклад роботи:

```
y: 0     12.218862     x1: 0    4.344042     x2: 0    121.886858
1        16.118862     1        8.744042     1        171.886858
2        18.718862     2       10.344042     2        221.886858
3        15.718862     3        7.744042     3        171.886858
4        10.718862     4        3.644042     4        101.886858
5        15.818862     5        8.144042     5        181.886858
6        16.718862     6        8.944042     6        181.886858
7        16.218862     7        8.644042     7        191.886858
8        14.718862     8        6.244042     8        151.886858
9        15.118862     9        7.644042     9        171.886858
10       12.418862     10       4.544042     10       111.886858
11       18.318862     11       9.944042     11       201.886858
12       12.818862     12       5.644042     12       141.886858
13       14.618862     13       6.144042     13       161.886858
14       19.918862     14      11.044042     14       261.886858
15       20.518862     15      11.244042     15       291.886858
16       15.218862     16       8.044042     16       151.886858
17       16.118862     17       7.644042     17       191.886858
```

```
B:  [6.32602343 0.71711181 0.02158093]
Залишки:  0     0.147243
1     -0.187095
2      0.186480
3      0.130017
4     -0.419160
5     -0.272637
6      0.053674
7     -0.447002
8      0.637303
9     -0.398272
10     0.419630
11     0.504943
12    -0.616620
13     0.393205
14     0.021265
15    -0.169586
16    -0.153498
17     0.170110
```

```
Кореляційна матриця:
           y        x1        x2
y   1.000000  0.979313  0.962905
x1  0.979313  1.000000  0.925229
x2  0.962905  0.925229  1.000000
```

2. Побудувати регресійну модель вигляду

$$y=b_0+b_1\, x_1+b_2\, x_2.$$

Програмна реалізація:

```
model = build_regresive_model(x1, x2, y, B)

def build_regresive_model(x1, x2, y, B):
    X = sm.add_constant(pd.DataFrame({'x1': x1, 'x2': x2}))
    model = sm.OLS(y, X).fit()
    print('\n', model.summary())
    print("\nРегресійна модель:", f"\ny = {B[0]} + {B[1]} * x1 + {B[2]}
* x2")
    return model
```

Приклад роботи:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.979
Method:                 Least Squares   F-statistic:                     397.4
Date:                Thu, 09 Nov 2023   Prob (F-statistic):           1.02e-13
Time:                        11:56:38   Log-Likelihood:                -6.5144
No. Observations:                  18   AIC:                             19.03
Df Residuals:                      15   BIC:                             21.70
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          6.3260      0.352     17.995      0.000       5.577       7.075
x1             0.7171      0.108      6.631      0.000       0.487       0.948
x2             0.0216      0.005      4.262      0.001       0.011       0.032
==============================================================================
Omnibus:                        0.856   Durbin-Watson:                   2.952
Prob(Omnibus):                  0.652   Jarque-Bera (JB):                0.696
Skew:                           0.057   Prob(JB):                        0.706
Kurtosis:                       2.044   Cond. No.                         718.
==============================================================================
```

```
Регресійна модель:
y = 6.326023427021678 + 0.7171118144838624 * x1 + 0.021580926355914715 * x2
```

3. Перевірити модель на адекватність за F – критерієм Фішера.

Програмна реалізація:

```
check_Fisher(calculate_F_value(X, y, B, U))

def calculate_F_value(X, Y, B, U):
    n = len(Y)
    m = len(B)

    Y_hat = X @ B
    SSR = ((Y_hat - Y.mean()) ** 2).sum()
    SSE = (U ** 2).sum()
    MSR = SSR / (m - 1)
    MSE = SSE / (n - m)
    F_value = MSR / MSE

    print('\nF-value: ', F_value)
    return F_value

def check_Fisher(F_value):
    alpha = 0.05
```

```
    if F_value < alpha:
        print("Модель є адекватною за F-критерієм Фішера")
    else:
        print("Модель не є адекватною за F-критерієм Фішера")
```

Приклад роботи:

```
F-value:  397.425295244778
Модель не є адекватною за F-критерієм Фішера
```

4. Провести оцінку значимості параметрів рівняння регресії b_1, b_2 та визначити інтервали довіри для них.

Програмна реалізація:

```
evaluate_significance(y, X, B, U)

def evaluate_significance(y, X, B, U):
    degrees_of_freedom = len(y) - X.shape[1]
    mse = np.sum(U**2) / degrees_of_freedom
    var_B = mse * np.linalg.inv(X.T @ X)

    se_B = np.sqrt(np.diag(var_B))
    print('Стандартні помилки: ', se_B)

    t_stats = B / se_B
    print('t-статистика: ', t_stats)

     p_values = [2 * (1 - stats.t.cdf(np.abs(t), degrees_of_freedom))
for t in t_stats]
    print('p-значення: ', p_values)

     confidence_interval = [stats.t.interval(0.95, degrees_of_freedom,
loc=b, scale=se) for b, se in zip(B, se_B)]
    print('Інтервали довіри: ', confidence_interval)
```

Приклад роботи:

```
Стандартні помилки:  [0.35153634 0.10814681 0.00506409]
t-статистика:  [17.99536124  6.63091047  4.26155914]
p-значення:  [1.4491297051222318e-11, 7.999216658172159e-06, 0.0006826404978952372]
Інтервали довіри:  [(5.576741454857739, 7.075305399185387), (0.48660234540367553, 0.9476212835640004), (0.010787070363601037, 0.03237478234825655)]
```

5. Знайти довірчий інтервал для окремого значення залежної змінної.

Програмна реалізація:

```python
def predict_confidence_interval(x_new, X, y, B, U, confidence=0.95):  1 usage
    y_hat = x_new @ B
    degrees_of_freedom = len(y) - len(B)
    mse = np.sum(U**2) / degrees_of_freedom
    se = np.sqrt(mse * (1 + x_new @ np.linalg.inv(X.T @ X) @ x_new.T))
    t_value = stats.t.ppf((1 + confidence) / 2, degrees_of_freedom)
    confidence_interval = (y_hat - t_value * se, y_hat + t_value * se)

    return confidence_interval
```

Приклад роботи:

```
Довірчий інтервал для прогнозованого значення y:  (31.56574423486063, 54.09233964892202)
```

6. Навести тлумачення параметрів b_0, b_1, b_2.

У контексті лінійної регресії, параметри b_0, b_1, b_2 мають наступне тлумачення:

b_0 - це інтерсепт або вільний член регресійного рівняння. Це значення y, коли всі незалежні змінні x_1, x_2, ... дорівнюють нулю. Іншими словами, це очікуване значення залежної змінної, якщо всі незалежні змінні відсутні або не мають впливу.

b_1 - це коефіцієнт нахилу для незалежної змінної x_1. Він показує, на скільки зміниться очікуване значення залежної змінної y, якщо x_1 збільшиться на одиницю, при умові, що всі інші незалежні змінні залишаються незмінними. Це можна розглядати як "вагу" або "вплив" змінної x_1 на y.

b_2 - це коефіцієнт нахилу для незалежної змінної x_2. Аналогічно до b_1, він показує зміну в очікуваному значенні y при збільшенні x_2 на одиницю, при незмінних інших змінних.

ВИСНОВОК:

У цьому дослідженні розглянув створення та оцінку адекватності лінійної моделі, яка враховує кілька факторів. Методом візуального аналізу та оцінки кореляції між залежною змінною та факторами сконструював лінійну регресію за допомогою методу найменших квадратів.

Перевірив адекватність моделі, використовуючи F-критерій Фішера, та оцінив значущість регресійних коефіцієнтів b1 і b2, встановивши для них інтервали довіри. Крім того, було визначено довірчий інтервал для окремих прогнозованих значень залежної змінної і проаналізовано значення параметрів моделі.

Це дослідження надало цінні відомості про взаємодію між факторами та залежною змінною, що зробило дану модель корисною для подальшого аналізу та прогнозування.