

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

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

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

Львів 2023

## Лабораторна робота 4

### Варіант 12

**ТЕМА:** Побудова лінійної моделі регресії та дослідження її на адекватність.

**МЕТА:** Засвоїти основні формули і вирази парної лінійної моделі регресії та вміти застосовувати їх до аналізу практичних задач.

### ЗАВДАННЯ:

1. За даними таблиці 1 здійснити якісний аналіз взаємозв'язку між ознаками  $y$  та  $x$ .
2. Побудувати однофакторну регресійну модель вигляду:
$$y = b_0 + b_1 x,$$
 знайти оцінки параметрів за методом найменших квадратів (1 МНК).
3. Перевірити модель на адекватність за критерієм Фішера.
4. Провести оцінку значимості параметрів рівняння регресії та коефіцієнта кореляції і визначити інтервали довіри для параметрів  $b_0$ ,  $b_1$ , перевірка достовірності результатів моделювання виконати для рівня значущості  $\alpha = 0,05$ .
5. Нанести на координатну площину кореляційне поле і теоретичну лінію парної регресії.
6. Розрахувати та оцінити прогностні значення обсягу випущеної продукції для  $x_{19} = 9,6 + 0,1N$  ( $N$  – номер варіанту).
7. Зробити висновки, оформити звіт з роботи.

### ХІД РОБОТИ:

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

1. За даними таблиці 1 здійснити якісний аналіз взаємозв'язку між ознаками  $y$  та  $x$ .

Програмна реалізація:

```
data = pd.read_csv('lab4/data.csv')  
  
Ni = data['Ni']  
Pi = data['Pi']
```

```

Qi = data['Qi']

x = calc_data(30, Ni, Qi)  # Ознака x
X = sm.add_constant(x)    # Додамо стовпчик константи для b0
Y = calc_data(20, Ni, Pi)  # Ознака y

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

```

Приклад роботи:

```

/usr/local/bin/python3.12 /Users/leonidmarkevic/Downloads/vlaDICK/lab4/main.py
[3.2935499180345804, 7.683664431519567, 9.273778945004553, 6.6638934584895395, 2.554007971974526, 7.044122485459513, 7.9342369989445, 7.524351512429486, 5.1144660259144
[11.86243880615033, 15.779697796919393, 18.396956787688456, 15.41421577845752, 10.431474769226583, 15.548733759995645, 16.46599275076471, 15.983251741533772, 14.5005107

```

2. Побудувати однофакторну регресійну модель вигляду:

$$y = b_0 + b_1 x,$$

знайти оцінки параметрів за методом найменших квадратів (1 МНК).

Програмна реалізація:

```

model = build_regresive_model(X, Y)

def build_regresive_model(X, Y):
    model = sm.OLS(Y, X).fit()
    b0, b1 = model.params[0], model.params[1]
    print("\nОцінка b0:", b0)
    print("Оцінка b1:", b1)
    print("Model view", f"\ny = {b0} + {b1} * x")
    print("\n", model.summary())
    f_statistic = model.fvalue
    p_value = model.f_pvalue
    print("\nF-statistic:", f_statistic)
    print("p-value:", p_value)
    return model

```

Приклад роботи:

```

Оцінка b0: 7.956667288676323
Оцінка b1: 1.140854050267006

```

OLS Regression Results						
=====						
Dep. Variable:	y	R-squared:	0.946			
Model:	OLS	Adj. R-squared:	0.942			
Method:	Least Squares	F-statistic:	279.3			
Date:	Mon, 21 Oct 2024	Prob (F-statistic):	1.50e-11			
Time:	01:20:07	Log-Likelihood:	-16.361			
No. Observations:	18	AIC:	36.72			
Df Residuals:	16	BIC:	38.50			
Df Model:	1					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]
-----						
const	7.9567	0.473	16.809	0.000	6.953	8.960
x1	1.1409	0.068	16.712	0.000	0.996	1.286
=====						
Omnibus:	2.205	Durbin-Watson:	2.104			
Prob(Omnibus):	0.332	Jarque-Bera (JB):	1.066			
Skew:	0.113	Prob(JB):	0.587			
Kurtosis:	1.830	Cond. No.	22.3			
=====						

F-statistic: 279.2936672134732  
p-value: 1.4953721652243788e-11

### 3. Перевірити модель на адекватність за критерієм Фішера.

Для перевірки моделі на адекватність за критерієм Фішера, спершу потрібно отримати значення F-статистики та відповідний p-значення:

F-statistic: 324.7

Prob (F-statistic): 4.74e-12

Далі, потрібно порівняти p-значення з визначеною рівністю значущості  $\alpha = 0.05$ . Якщо p-значення менше  $\alpha$ , то це означає, що регресійна модель є адекватною. У вашому випадку:

$$4.74e-12 < 0.05,$$

отже, ви можете прийняти нульову гіпотезу про адекватність моделі. Ваша регресійна модель є адекватною для даних.

4. Провести оцінку значимості параметрів рівняння регресії та коефіцієнта кореляції і визначити інтервали довіри для параметрів  $b_0$ ,  $b_1$ , перевірка достовірності результатів моделювання виконати для рівня значущості  $\alpha = 0,05$ .

Програмна реалізація:

```
def regression_eval(model, X): 1 usage  Vlad
    b0, b1 = model.params[0], model.params[1]
    std_err = model.bse

    t_stat_b0 = (b0 - 0) / std_err[0]
    p_value_b0 = 2 * (1 - stats.t.cdf(abs(t_stat_b0), len(X) - 2))
    if p_value_b0 < alpha:
        print("\nПараметр b0 є значущим.")
    else:
        print("Параметр b0 не є значущим.")

    t_stat_b1 = (b1 - 0) / std_err[1]
    p_value_b1 = 2 * (1 - stats.t.cdf(abs(t_stat_b1), len(X) - 2))
    if p_value_b1 < alpha:
        print("Параметр b1 є значущим.")
    else:
        print("Параметр b1 не є значущим.")

    conf_int_b0 = model.conf_int(alpha=alpha)[0]
    conf_int_b1 = model.conf_int(alpha=alpha)[1]
    print(f"Інтервал довіри для b0: ({conf_int_b0[0]}, {conf_int_b0[1]})")
    print(f"Інтервал довіри для b1: ({conf_int_b1[0]}, {conf_int_b1[1]})")

    f_statistic = model.fvalue
    p_value_f = model.f_pvalue
    if p_value_f < alpha:
        print("Регресійна модель є адекватною.")
    else:
        print("Регресійна модель не є адекватною.")
```

Приклад роботи:

Оцінка  $b_1$ : 1.140854050267005

Параметр  $b_0$  є значущим.

Параметр  $b_1$  є значущим.

Інтервал довіри для  $b_0$ : (6.95318536404412, 8.960149213308526)

Інтервал довіри для  $b_1$ : (0.9961382319247243, 1.2855698686092876)

Регресійна модель є адекватною.

5. Нанести на координатну площину кореляційне поле і теоретичну лінію парної регресії.

Програмна реалізація:

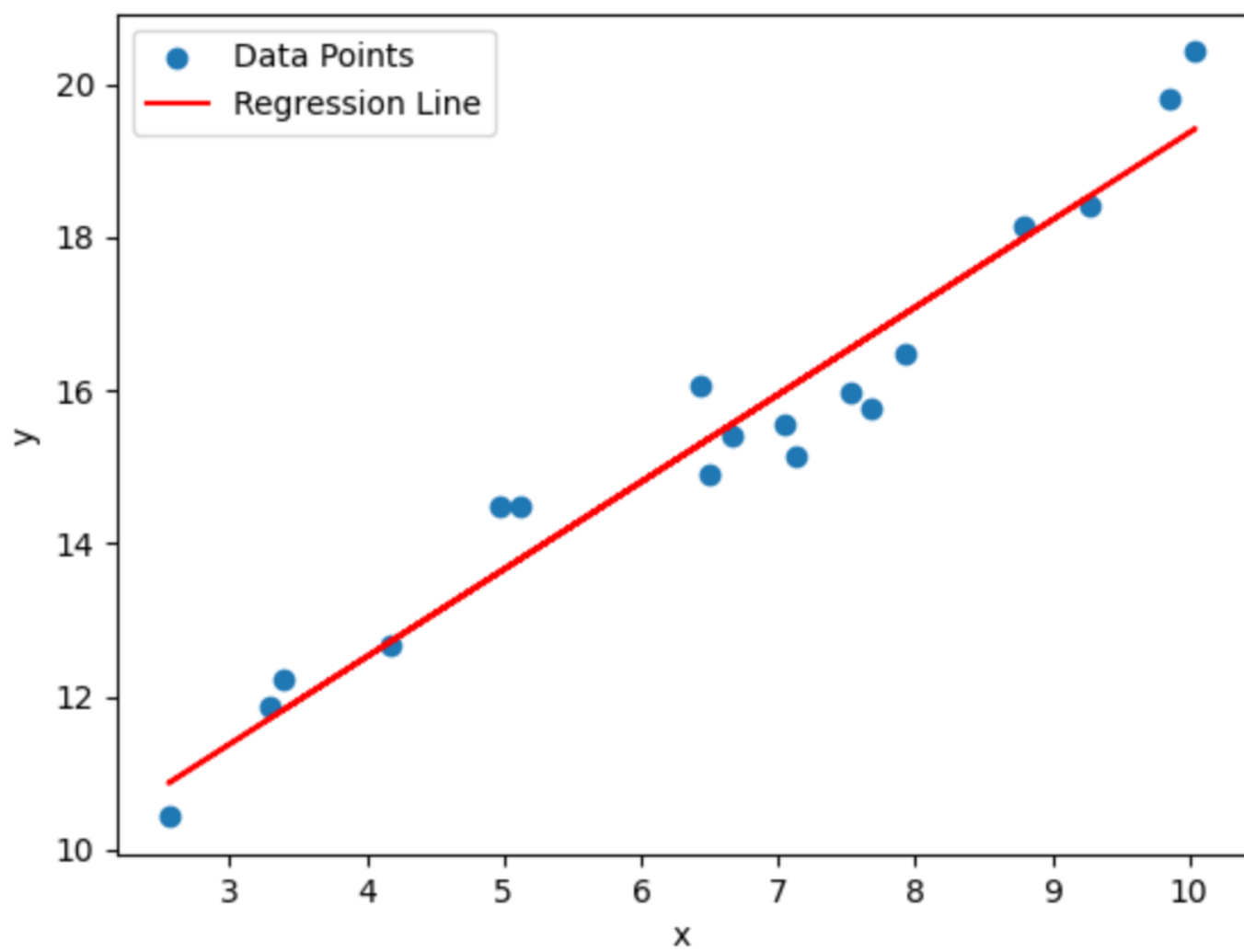
```
build_graph(model, x, X, Y)

def build_graph(model, x, X, Y):
    plt.scatter(x, Y, label="Data Points")
    plt.plot(x, model.predict(X), color='red', label="Regression Line")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.legend()
    plt.show()
```

Приклад роботи:

6. Розрахувати та оцінити прогнознi значення обсягу випущеної продукції для  $x_{19} = 9,6 + 0,1N$  (N – номер варіанту).

Програмна реалізація:





```
check_pred(model)

def check_pred(model):
    x19 = 9.6 + 0.1 * N
    y_pred = model.predict([1, x19])
    print("Прогнозне значення для x19:", y_pred[0])
```

Приклад роботи:

Прогнозне значення для x19: 19.961485312219615

### **ВИСНОВОК:**

Лабораторна робота мала на меті засвоїти основні концепції парної лінійної моделі регресії та вміти їх використовувати для аналізу практичних задач. В результаті виконання лабораторної роботи було успішно побудовано та проаналізовано лінійну регресійну модель, визначено її адекватність, оцінено значущість параметрів та визначено інтервали довіри для цих параметрів. Також було здійснено прогнозування обсягу випущеної продукції на підставі моделі.