CS3-mid-p5

5-1. (i)

5-2. (i), (iii)

5-3. (A) (i) **(B)** (iii) **(C)** (iv) **(D)** (ii)

5-4.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

```
C:\Users\shugo\Miniconda3\envs\cs3-2022\lib\site-packages\statsmodels\compat\pandas.py:61: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.
  from pandas import Int64Index as NumericIndex
```

In [2]:
```
csv_in = 'mid-p5.csv'
df = pd.read_csv(csv_in, sep=',', skiprows=0, header=0)
print(df.shape)
print(df.info())
display(df.head())
```

```
(100, 2)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   x       100 non-null    float64
 1   y       100 non-null    float64
dtypes: float64(2)
memory usage: 1.7 KB
None
```

	x	y
0	3.89	15.94
1	1.19	3.52
2	4.12	20.55
3	4.83	30.98
4	4.86	21.34

In [3]:
```
X = df[ ['x'] ]
y = df[ 'y' ]
print('X:', X.shape)
display(X.head())
print('y:', y.shape)
print(y.head())
```

```
X: (100, 1)
```

	x
0	3.89

	x
1	1.19
2	4.12
3	4.83

```
y: (100,)
0    15.94
1     3.52
2    20.55
3    30.98
4    21.34
Name: y, dtype: float64
```

In [4]:

```
X_c = sm.add_constant(X)
display(X.head())
display(X_c.head())
```

	x
0	3.89
1	1.19
2	4.12
3	4.83
4	4.86

	const	x
0	1.0	3.89
1	1.0	1.19
2	1.0	4.12
3	1.0	4.83
4	1.0	4.86

In [5]:

```
model = sm.OLS(y, X_c)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.613
Model:                            OLS   Adj. R-squared:                  0.609
Method:                 Least Squares   F-statistic:                     155.4
Date:                Fri, 03 Jun 2022   Prob (F-statistic):           6.22e-22
Time:                        17:43:03   Log-Likelihood:                -320.47
No. Observations:                 100   AIC:                             644.9
Df Residuals:                      98   BIC:                             650.2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.8024      1.118      1.613      0.110      -0.416       4.021
```

```
x              4.9406      0.396     12.465      0.000       4.154       5.727
==============================================================================
Omnibus:                        0.419   Durbin-Watson:                   2.058
Prob(Omnibus):                  0.811   Jarque-Bera (JB):                0.104
Skew:                           0.008   Prob(JB):                        0.949
Kurtosis:                       3.157   Cond. No.                         5.72
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly s
pecified.
```

In [6]:

```
print('R2:', results.rsquared)
print('Adj R2:', results.rsquared_adj)
```

```
R2: 0.6132358565364777
Adj R2: 0.6092892836439928
```

In [7]:

```
print(results.params)
```

```
const    1.802399
x        4.940600
dtype: float64
```

5-3. a0 1.802

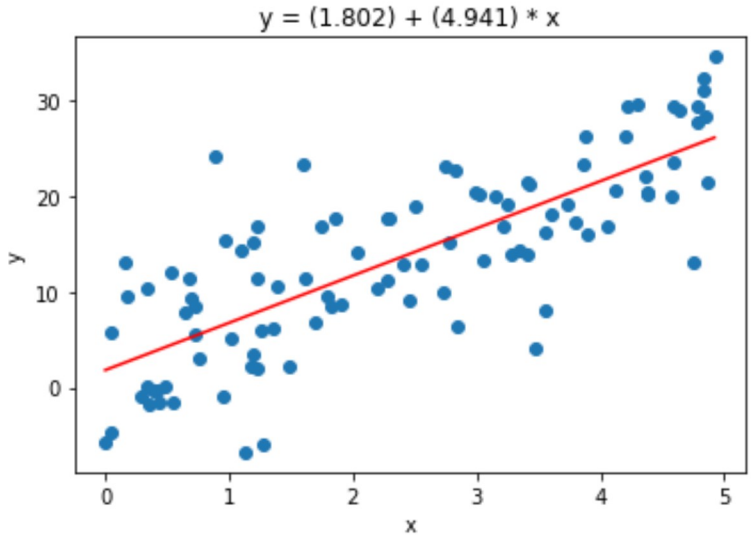
5-3. a1 4.941

In [8]:

```
a0 = results.params['const']
a1 = results.params['x']
x_min = X['x'].min()
x_max = X['x'].max()
x_min_max = np.array([x_min, x_max])
y_min_max = a0 + a1 * x_min_max
```

In [9]:

```
plt.title('y = ({:.3f}) + ({:.3f}) * x'.format(a0, a1))
plt.scatter(X['x'], y)
plt.plot(x_min_max, y_min_max, c='red')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



In [10]:

```
x_test = np.array([ 4.12 ])
y_test = a0 + a1 * x_test
print(y_test)
```

```
[22.15767035]
```

5-4. (1) 22.158

In [11]:

```
print(df[ df['x']==4.12 ])
print(y_test - 20.55)
```

```
      x      y
2  4.12  20.55
[1.60767035]
```

5-4. (2) 1.608