

Chapter 5

Simple linear regression

Exercises

Contents

5 Simple linear regression

Exercises	1
5.1 Plastic film folding machine	3
5.2 Linear regression life time model	6
5.3 Yield of chemical process	11
5.4 Plastic material	14
5.5 Water pollution	17
5.6 Membrane pressure drop	20
5.7 Membrane pressure drop (matrix form)	33
5.8 Independence and correlation	38

5.1 Plastic film folding machine

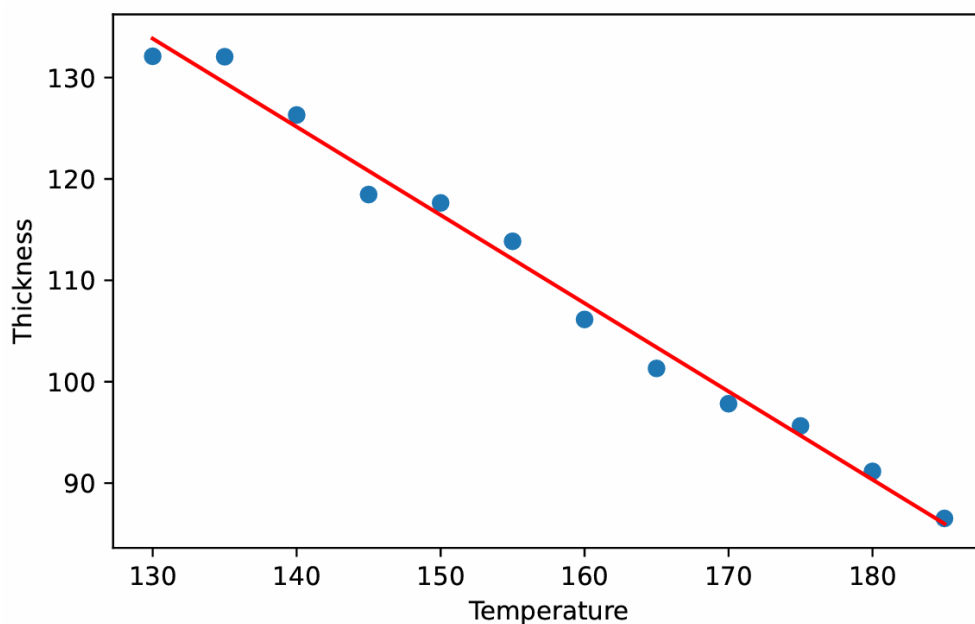
|||| Exercise 5.1 Plastic film folding machine

On a machine that folds plastic film the temperature may be varied in the range of 130–185 °C. For obtaining, if possible, a model for the influence of temperature on the folding thickness, $n = 12$ related sets of values of temperature and the fold thickness were measured.

The following regression model was fitted to the data:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

The data and the estimated model are illustrated in the following figure:



Exam-style questions

a) Determine by looking at the figure which of the following sets of estimates for the parameters in the regression model is correct:

- 1) $\hat{\beta}_0 = 0, \hat{\beta}_1 = -0.9, \hat{\sigma} = 36$
- 2) $\hat{\beta}_0 = 0, \hat{\beta}_1 = 0.9, \hat{\sigma} = 3.6$
- 3) $\hat{\beta}_0 = 252, \hat{\beta}_1 = -0.9, \hat{\sigma} = 3.6$

$$4) \hat{\beta}_0 = -252, \hat{\beta}_1 = -0.9, \hat{\sigma} = 36$$

$$5) \hat{\beta}_0 = 252, \hat{\beta}_1 = -0.9, \hat{\sigma} = 36$$

|||| Solution

Intercept ($\hat{\beta}_0$):

The plot does not show the intersection with the y -axis, but it is clear from the plot that the intersection should be a relatively large and positive number. The only possible intercept ($\hat{\beta}_0$) among the ones given in the answers is 252.

Slope ($\hat{\beta}_1$):

The slope estimate of -0.9 in the two options with $\hat{\beta}_0 = 252$ looks reasonable.

Standard deviation of errors ($s_e = \hat{\sigma}$):

We need to decide on whether the estimated standard deviation of the error $s_e = \hat{\sigma}$ is 3.6 or 36. From the figure it is clear that the points do not have an average vertical distance to the line of the size 36, so 3.6 must be the correct number and hence the correct answer is:

$$3) \hat{\beta}_0 = 252, \hat{\beta}_1 = -0.9, \hat{\sigma} = 3.6$$

- b) The proportion of the variation in fold thickness that is explained by the regression model is given by the coefficient of determination, R^2 .

Which of the following statements is correct?

- 1) The proportion of explained variation is 50% and the correlation is 0.98
- 2) The proportion of explained variation is 0% and the correlation is -0.98
- 3) The proportion of explained variation is 96% and the correlation is -1
- 4) The proportion of explained variation is 96% and the correlation is 0.98
- 5) The proportion of explained variation is 96% and the correlation is -0.98

|||| **Solution**

The points lie close to the fitted line, so the regression model explains most of the variation in fold thickness, i.e. R^2 must be close to 1. This rules out answers 1 and 2 (50% and 0%).

In simple linear regression the coefficient of determination equals the squared correlation, $R^2 = r^2$, and the sign of r is the same as the sign of the slope. Since the slope is clearly negative, the correlation must be negative, which rules out answers 1 and 4.

This leaves answers 3 and 5. A correlation of exactly -1 would require all observations to lie exactly on the line, which they do not. Moreover, $r = -1$ would give $R^2 = 1 = 100\%$, not 96%, so answer 3 is also internally inconsistent. Answer 5 is consistent: $(-0.98)^2 \approx 0.96$. So the correct answer is:

- 5) The proportion of explained variation is 96% and the correlation is -0.98

5.2 Linear regression life time model

|||| Exercise 5.2 Linear regression life time model

A company manufactures an electronic device to be used in a very wide temperature range. The company knows that increased temperature shortens the life time of the device, and a study is therefore performed in which the life time is determined as a function of temperature. The following data is found:

Temperature in Celsius (t)	10	20	30	40	50	60	70	80	90
Life time in hours (y)	420	365	285	220	176	117	69	34	5

In this exercise you should use Python for computations, data visualization etc.

The data can be stored in a Python DataFrame named `df` by copy-pasting the following code:

```
df = pd.DataFrame({
    't': [10, 20, 30, 40, 50, 60, 70, 80, 90],
    'y': [420, 365, 285, 220, 176, 117, 69, 34, 5]
})
```

a) Analyse the relationship between Temperature and Life time:

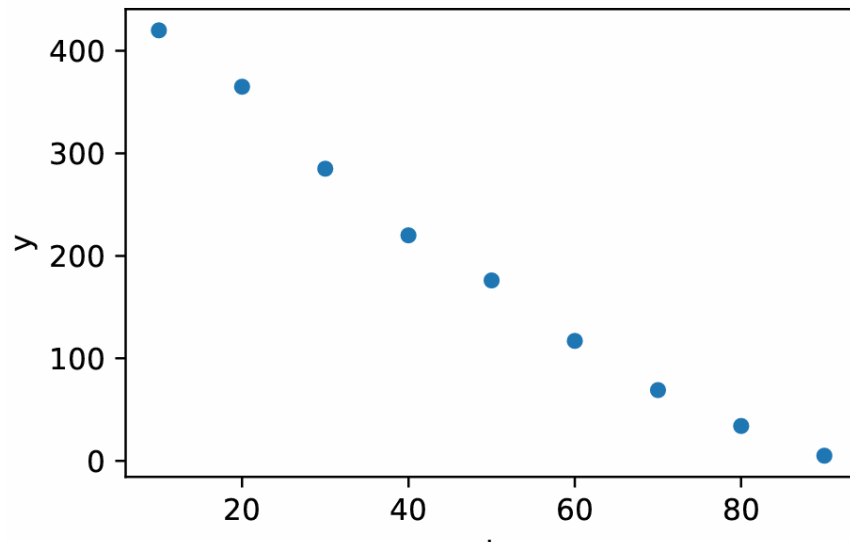
- Visualize the data
- Specify a relevant statistical model
- Estimate the model parameters
- Interpret the results

|||| Solution

First we visualize the data:

```
df = pd.DataFrame({
    't': [10, 20, 30, 40, 50, 60, 70, 80, 90],
    'y': [420, 365, 285, 220, 176, 117, 69, 34, 5]
})
```

```
df.plot.scatter('t', 'y')
plt.show()
```



From the plot we conclude that there seems to be a linear relationship between Temperature and Life time (with a negative slope). We therefore choose to model the data with a simple linear regression model:

$$Y_i = \beta_0 + \beta_1 t_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

We estimate the parameters (intercept and slope) using Theorem 5.4 (and 5.8 for standard errors):

```
ybar = df['y'].mean()
tbar = df['t'].mean()

# compute Sxx to use in calculations:
Sxx = np.sum((df['t']-tbar)**2)

# estimate parameters:
beta1_hat = np.sum((df['y']-ybar)*(df['t']-tbar))/Sxx
beta0_hat = ybar - beta1_hat*tbar

print([beta0_hat, beta1_hat])
[453.55555555555554, -5.3133333333333335]

# estimate sigma^2
sigma2_hat = np.sum((df['y'] - beta0_hat - beta1_hat*df['t']
                    )**2)/(9-2)
```

```
# estimate parameter standard errors:
var_beta1_hat = sigma2_hat/Sxx
var_beta0_hat = sigma2_hat/9 + tbar**2*sigma2_hat/Sxx

print([np.sqrt(var_beta1_hat), np.sqrt(var_beta0_hat)])
[0.2557818184006412, 14.393646942675979]
```

The model can also be estimated using `smf.ols`:

```
fit = smf.ols(formula= 'y~t', data=df).fit()
print(fit.summary(slim=True))
```

OLS Regression Results						
Dep. Variable:	y	R-squared:	0.984			
Model:	OLS	Adj. R-squared:	0.982			
No. Observations:	9	F-statistic:	431.5			
Covariance Type:	nonrobust	Prob (F-statistic):	1.51e-07			
	coef	std err	t	P> t	[0.025	0.975]
Intercept	453.5556	14.394	31.511	0.000	419.520	487.591
t	-5.3133	0.256	-20.773	0.000	-5.918	-4.709

The output given above states the parameter estimates in the column named `coef` and the standard errors for these estimates in column named `std err`.

Interpretation of the estimated model: Since the slope is -5.3 (and has unit hours per degree Celsius) we conclude that the average Life time of the electronic devices decreases by 5.3 hours for every one degree increase in temperature (Celsius). We can verify visually that this conclusion is in good agreement with the plot generated above.

- b) Compute a 95% confidence interval for the slope in the linear regression model. What do you conclude from this confidence interval?

|||| Solution

The confidence interval has already been computed in the regression table shown above and we can read off the 95% confidence interval for $\hat{\beta}_1$: $[-5.918; -4.709]$.

Conclusion: Since the 95% confidence interval does not include the value zero, we conclude that the slope is significantly different from zero at significance level $\alpha = 0.05$. Since the slope parameter describes the linear relationship between Life time and temperature, we also say that "there IS a significant (linear) relationship between

temperature and Life time" (if the slope parameter could be zero, then we say "there is NO significant relationship between temperature and Life time").

- c) Compute a 99% confidence interval for the slope in the linear regression model. What do you conclude from this confidence interval?

Use Python to compute quantiles and probabilities from relevant probability distribution(s).

|||| Solution

This confidence interval is not included in the regression table shown above, but we can compute it in the following way:

$$\begin{aligned} & \hat{\beta}_1 \pm t_{0.995, n-2} \cdot \hat{\sigma}_{\beta_1} \\ & -5.3133 \pm 3.499 \cdot 0.256 \\ & [-6.21; -4.42] \end{aligned}$$

where the value of the quantile $t_{0.995, n-2}$ has been found using Python (from t -distribution with $n - 2 = 9 - 2 = 7$ degrees of freedom):

```
print(stats.t.ppf(0.995, df=7))
3.499
```

Conclusion: Since the 99% confidence interval does not include the value zero, we conclude that the slope is significantly different from zero at significance level $\alpha = 0.01$.

- d) Compute the p -value for the slope.

Use Python to compute quantiles and probabilities from relevant probability distribution(s).

|||| **Solution**

To compute the p -value we first compute the test statistic, following equation (5-46) in the book:

$$t_{obs,\beta_1} = \frac{\hat{\beta}_1 - 0}{\hat{\sigma}_{\beta_1}} = \frac{-5.3133}{0.256} = -20.8$$

Notice that this value is also stated in the regression table shown above (column t).

Having this value, now we can compute our p -value:

$$p\text{-value} = P(T \leq -|t_{obs,\beta_1}|) + P(T \geq |t_{obs,\beta_1}|) = 2P(T \leq -|t_{obs,\beta_1}|) = 1.5 \cdot 10^{-7}$$

where the probability has been calculated in Python with:

```
print(2 * stats.t.cdf(-20.8, df=7))
1.5e-07
```

We can observe that the p -value is very small, which is consistent with our previous results, that indicated that the p -value should be smaller than 0.01.

This result is also consistent with the output shown in the regression table above: it states the p -value is 0.000, meaning that it rounds to 0.000, i.e. it is smaller than 0.0005.

If we would like to get the p -value(s) directly from Python we can use the following:

```
print(fit.pvalues)
Intercept      8.376549e-09
t              1.505039e-07
```

we see that the p -value stated for the slope is consistent with our own calculations.

5.3 Yield of chemical process

|||| Exercise 5.3 Yield of chemical process

The yield y of a chemical process is a random variable whose value is considered to be a linear function of the temperature x . The following data of corresponding values of x and y is found:

Temperature in °C (x)	0	25	50	75	100
Yield in grams (y)	14	38	54	76	95

The average and standard deviation of temperature and yield are

$$\bar{x} = 50, s_x = 39.52847, \bar{y} = 55.4, s_y = 31.66702.$$

In the exercise the usual linear regression model is used

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma_\varepsilon^2), \quad i = 1, \dots, 5$$

- a) Can a significant relationship between yield and temperature be documented on the usual significance level $\alpha = 0.05$?

|||| Solution

It could most easily be solved by running the regression in Python as:

```
df = pd.DataFrame({
    'x': [0, 25, 50, 75, 100],
    'y': [14, 38, 54, 76, 95]})
fit = smf.ols('y ~ x', data=df).fit()
print(fit.summary(slim=True))
```

```

                                OLS Regression Results
=====
Dep. Variable:                  y      R-squared:                0.997
Model:                        OLS      Adj. R-squared:           0.996
No. Observations:              5      F-statistic:              1071.
Covariance Type:               nonrobust  Prob (F-statistic):       6.27e-05
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept    15.4000      1.497     10.290     0.002     10.637     20.163
x              0.8000      0.024     32.733     0.000       0.722      0.878
=====
```

Alternatively one could use hand calculations and use the formula in Theorem 5.12 for the t -test of the null hypothesis: $H_0 : \beta_1 = 0$.

The relevant test statistic and p -value can be read off in the Python output as 32.7 and 0.000063. So the answer is:

Yes, as the relevant test statistic and p -value are resp. 32.7 and $0.00006 < 0.05 = \alpha$.

- b) Give the 95% confidence interval of the expected yield at a temperature of $x_{\text{new}} = 80^\circ\text{C}$.

|||| Solution

We use the formula in Method 5.18 for the confidence limit of the line (the expected value of Y_i for a value x_{new}):

$$\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} \pm t_{1-\alpha/2} \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}},$$

and we have to compute $\hat{\beta}_0$, $\hat{\beta}_1$ and s_e either by hand OR in Python as above:

$$\hat{\beta}_0 = 15.4, \hat{\beta}_1 = 0.8, \hat{\sigma} = 1.932.$$

So the confidence interval becomes

$$(15.4 + 0.8 \cdot 80) \pm 3.182 \cdot 1.932 \sqrt{\frac{1}{5} + \frac{(80 - 50)^2}{6250}},$$

since

$$s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n-1} S_{xx} \Leftrightarrow$$

$$S_{xx} = (n-1)s_x^2 = 4 \cdot 39.528^2 = 6250.$$

Thus the answer is

$$79.40 \pm 3.61 = [75.79, 83.01].$$

In Python this could be done by:

```
print(fit.get_prediction(pd.DataFrame({'x': [80]})).summary_frame(alpha=0.05))
```

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
0	79.4	1.133255	75.793477	83.006523	72.27132	86.52868

c) What is the upper quartile of the residuals?

|||| **Solution**

The five residuals become: $-1.4, 2.6, -1.4, 0.6$ and -0.4 .

We use the basic definition of finding a quantile (from Definition 1.7) and the upper quartile is $q_{0.75}$ (see Definition 1.8). We set $n = 5, p = 0.75$, so

$$np = 3.75$$

So the upper quartile is the 4th observation in the ordered sequence:

$$-1.4, -1.4, -0.4, 0.6, 2.6.$$

The residuals can be shown using the `fit.resid` from the regression. They are:

```
Residuals :  
      1      2      3      4      5  
-1.4   2.6  -1.4   0.6  -0.4
```

So the answer is: 0.6.

5.4 Plastic material

|||| Exercise 5.4 Plastic material

In the manufacturing of a plastic material, it is believed that the cooling time has an influence on the impact strength. Therefore a study is carried out in which plastic material impact strength is determined for 4 different cooling times. The results of this experiment are shown in the following table:

Cooling times in seconds (x)	15	25	35	40
Impact strength in kJ/m^2 (y)	42.1	36.0	31.8	28.7

The following statistics may be used:

$$\bar{x} = 28.75, \bar{y} = 34.65, S_{xx} = 368.75.$$

- a) What is the 95% confidence interval for the slope of the regression model, expressing the impact strength as a linear function of the cooling time?

|||| Solution

The easiest way to get to the confidence interval is to use the standard error for the slope ($\hat{\sigma}_{\beta_1}$ or denoted with SE_{β_1}) given in the Python output:

```
x = [15, 25, 35, 40]
y = [42.1, 36.0, 31.8, 28.7]
df = pd.DataFrame({'x': x, 'y': y})
fit = smf.ols('y ~ x', data=df).fit()
print(fit.summary(slim=True))
```

```

                                OLS Regression Results
=====
Dep. Variable:                  y      R-squared:                0.994
Model:                        OLS      Adj. R-squared:           0.991
No. Observations:              4      F-statistic:              323.7
Covariance Type:               nonrobust  Prob (F-statistic):       0.00308
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept    49.6390      0.878     56.513     0.000     45.860     53.418
x            -0.5214      0.029    -17.991     0.003     -0.646     -0.397
=====
```

The standard error for the slope is $\hat{\sigma}_{\beta_1} = 0.029$ (also known as the sampling distribution standard deviation for $\hat{\beta}_1$). Finding the relevant t -quantile (with $\nu = 2$ degrees of freedom) (either of):

```
print(stats.t.ppf(0.025, 2), stats.t.ppf(0.975, 2))

-4.3026527299112765  4.302652729911275
```

$|t_{0.025}| = 4.303$, which using Method 5.15 gives

$$-0.521 \pm 4.303 \cdot 0.029,$$

giving

$$-0.521 \pm 0.125,$$

or, that we say with high confidence that the true parameter value is in the interval, i.e.

$$-0.646 \leq \beta_1 \leq -0.396.$$

- b) Can you conclude that there is a relation between the impact strength and the cooling time at significance level $\alpha = 5\%$?

|||| Solution

The relevant p -value can be read off directly from the summary output: 0.00308, and we can conclude: *Yes, as the relevant p -value is 0.00308, which is smaller than 0.05.* Again, 0.003 can be read off the summary, but use `fit.pvalues` for more specific p -values.

- c) For a similar plastic material the tabulated value for the linear relation between temperature and impact strength (i.e. the slope) is -0.30 . If the following hypothesis is tested (at level $\alpha = 0.05$)

$$H_0 : \beta_1 = -0.30$$

$$H_1 : \beta_1 \neq -0.30$$

with the usual t -test statistic for such a test, what is the range (for t) within which the hypothesis is accepted?

|||| Solution

The so-called critical values for the t -statistic with $\nu = 2$ degrees of freedom are found as (or at least the negative one of the two): $t_{0.025} = -4.303$ – in Python: `stats.t.ppf(0.025, 2)`. So the answer becomes:

$$[-4.303, 4.303].$$

5.5 Water pollution

|||| Exercise 5.5 Water pollution

In a study of pollution in a water stream, the concentration of pollution is measured at 5 different locations. The locations are at different distances to the pollution source. In the table below, these distances and the average pollution are given:

Distance to the pollution source (in km)	2	4	6	8	10
Average concentration	11.5	10.2	10.3	9.68	9.32

- a) What are the parameter estimates for the three unknown parameters in the usual linear regression model: 1) The intercept (β_0), 2) the slope (β_1) and 3) error standard deviation (σ)?

|||| Solution

The question is solved by considering the following Python output:

```
df = pd.DataFrame({
    'concentration': [11.5, 10.2, 10.3, 9.68, 9.32],
    'distance': [2, 4, 6, 8, 10]
})
fit = smf.ols('concentration ~ distance', data=df).fit()
print(fit.summary(slim=True))
```

```

                        OLS Regression Results
=====
Dep. Variable:          concentration    R-squared:                0.868
Model:                  OLS            Adj. R-squared:           0.823
No. Observations:        5             F-statistic:             19.66
Covariance Type:         nonrobust      Prob (F-statistic):       0.0213
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept      11.6640      0.365      31.955      0.000      10.502      12.826
distance       -0.2440      0.055      -4.434      0.021       -0.419      -0.069
=====
```

Given the knowledge of the Python output structure, the first two values can be read directly from the output. σ can be found from either the Mean Squared error of the residuals `np.sqrt(fit.mse_resid)` or the scale `np.sqrt(fit.scale)`.

So the correct answer is: $\hat{\beta}_0 = 11.7$, $\hat{\beta}_1 = -0.244$ and $\hat{\sigma} = 0.348$.

- b) How large a part of the variation in concentration can be explained by the distance?

|||| Solution

The amount of variation in the model output (Y) explained by the variable input (x) can be found from the squared correlation, which can be read off directly from the output as “R-squared”. So the correct answer is: $R^2 = 86.8\%$ (it is actually an estimate of the variation in concentration which can be explained by distance, since it is what we found with the particular data at hand. If the sample was taken again, then this value would vary. We should actually calculate a confidence interval for R^2 to understand how accurate this estimate is!).

- c) What is a 95%-confidence interval for the expected pollution concentration 7 km from the pollution source?

|||| Solution

The wanted number is estimated by the point on the line (using $x_{\text{new}} = 7$)

$$-0.244 \cdot 7 + 11.664 = 9.96,$$

and the confidence interval is given by

$$9.96 \pm t_{0.025}(3) \cdot \hat{\sigma} \sqrt{\frac{1}{5} + \frac{(7-6)^2}{S_{xx}}},$$

where $S_{xx} = 4^2 + 2^2 + 0^2 + 2^2 + 4^2 = 40$ and $t_{0.025}(3) = 3.182$ (in Python: `stats.t.ppf(0.975, 3)`). We have that

$$3.182 \cdot 0.348 \sqrt{\frac{1}{5} + \frac{1}{40}} = 0.525,$$

where s_x is:

```
print(df['distance'].std(ddof=1))
```

```
3.1622776601683795
```

and thus

$$S_{xx} = (n - 1) \cdot s_x^2 = 4 \cdot 3.162^2 = 40.$$

This could also have been found by

```
print(fit.get_prediction(pd.DataFrame({'distance': [7]})).summary_frame(alpha
    =0.05))
```

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
0	9.956	0.165082	9.430636	10.481364	8.730151	11.181849

So the correct answer is:

$$9.96 \pm 0.525 = [9.43, 10.5].$$

5.6 Membrane pressure drop

|||| Exercise 5.6 Membrane pressure drop

When purifying drinking water you can use a so-called membrane filtration. In an experiment one wishes to examine the relationship between the pressure drop across a membrane and the flux (flow per area) through the membrane. We observe the following 10 related values of pressure (x) and flux (y):

	1	2	3	4	5	6	7	8	9	10
Pressure (x)	1.02	2.08	2.89	4.01	5.32	5.83	7.26	7.96	9.11	9.99
Flux (y)	1.15	0.85	1.56	1.72	4.32	5.07	5.00	5.31	6.17	7.04

The data can be stored in a Python DataFrame named `df` by copy-pasting the following code:

```
df = pd.DataFrame({
    'pressure':
        [1.02, 2.08, 2.89, 4.01, 5.32, 5.83, 7.26, 7.96, 9.11, 9.99],
    'flux':
        [1.15, 0.85, 1.56, 1.72, 4.32, 5.07, 5.00, 5.31, 6.17, 7.04]
})
```

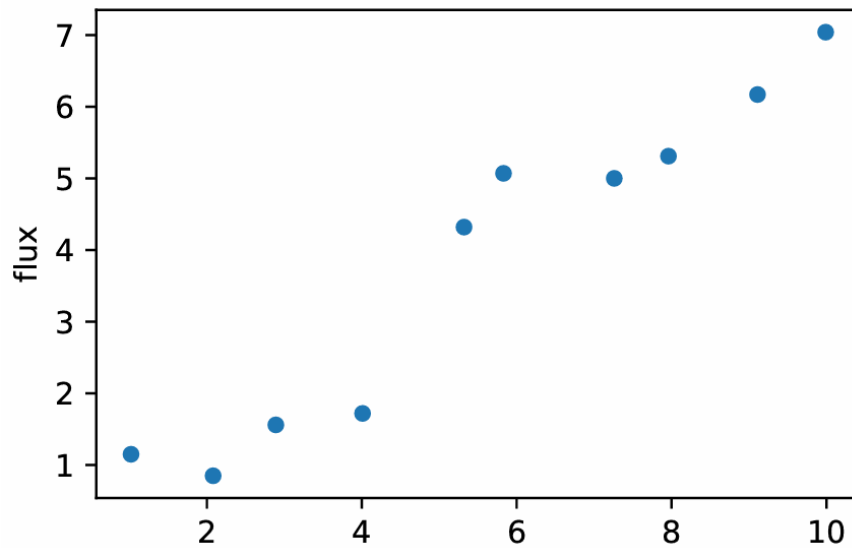
- Visualize the data and estimate the sample correlation coefficient between pressure and flux. Use Python for both tasks.

Does the resulting correlation match your expectations from inspecting the data visually?

|||| Solution

We plot the data and calculate the sample correlation coefficient:

```
df.plot.scatter('pressure', 'flux')
plt.show()
```



```
print(np.corrcoef(df['flux'], df['pressure']))
```

```
[[1.          0.96381844]
 [0.96381844  1.          ]]
```

From the output (the off-diagonal elements) we see that the sample correlation coefficient between pressure and flux is:

$$\hat{\rho} = 0.964$$

We see that there is a positive correlation, which is also confirmed by a visual inspection of the scatter plot (the flux seems to increase linearly with increasing pressure).

The following regression model is fitted to the data:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2),$$

where x_i is the pressure and Y_i is the flux. The resulting output table is shown here:

```
fit = smf.ols('flux ~ pressure', data=df).fit()
print(fit.summary(slim=True))
```

```

=====
                        OLS Regression Results
=====
Dep. Variable:          flux    R-squared:                0.929
Model:                OLS    Adj. R-squared:            0.920
No. Observations:      10    F-statistic:             104.6
Covariance Type:      nonrobust    Prob (F-statistic):    7.18e-06
=====
                        coef    std err          t      P>|t|      [0.025    0.975]
-----
Intercept            -0.1886     0.442     -0.427     0.681    -1.207     0.830
pressure              0.7225     0.071     10.227     0.000     0.560     0.885
=====

```

- b) What is a 90% confidence interval for the slope β_1 ? Compare this 90% confidence interval to the 95% confidence interval shown in the regression table.

Use Python to compute quantiles and probabilities from relevant probability distribution(s).

|||| Solution

For the slope we get the following information from the table:

$$\hat{\beta}_1 = 0.7225$$

$$\hat{\sigma}_{\beta_1} = 0.071$$

We also see that the 95% confidence interval for β_1 is [0.560; 0.885].

We now estimate the 90% confidence interval for β_1 using Method 5.15:

$$\hat{\beta}_1 \pm t_{0.95, n-2} \cdot \hat{\sigma}_{\beta_1}$$

$$0.7225 \pm 1.860 \cdot 0.071$$

$$[0.590; 0.855]$$

where the value of the quantile $t_{0.95, n-2}$ has been found using Python (from a t -distribution with $n - 2 = 10 - 2 = 8$ degrees of freedom):

```
print(stats.t.ppf(0.95, df=8))
1.860
```

We see that the 90% confidence interval is a bit narrower than the 95% confidence interval (as we expected).

c) Let

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

be the predicted (or *fitted*) values of flux corresponding to pressure x_i (values on the regression line).

Now compute the following measures of variability of the flux (use Python for computations).

1. The sample variance of the flux.
2. The variability **explained** by the regression line, i.e.:

$$\frac{1}{n-1} \sum_{i=1}^{10} (\hat{y}_i - \bar{y})^2$$

Explain in words what the *variability explained by the regression line* expresses.

3. The variability **NOT explained** by the regression line, i.e.:

$$\frac{1}{n-1} \sum_{i=1}^{10} (y_i - \hat{y}_i)^2$$

Explain in words what the *variability NOT explained by the regression line* expresses.

What is the proportion of the flux-variation that is explained by the model, and what is the proportion that is NOT explained? How does this result compare to the R-squared value printed in the regression table and to the sample correlation coefficient calculated above?

|||| Solution

1.

The **sample variance** of the flux is

$$s_y^2 = \frac{1}{n-1} \sum_{i=1}^{10} (y_i - \bar{y})^2,$$

and it expresses how much the observed flux values vary around their mean $\bar{y}$. The sample variance is computed using Python:

```
print(df['flux'].var(ddof=1))
5.198
```

2.

The variability explained by the regression line is

$$\frac{1}{n-1} \sum_{i=1}^{10} (\hat{y}_i - \bar{y})^2$$

To calculate this directly we first need to compute the **fitted values**

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

i.e., the points on the regression line at each of the observed pressures x_i .

We insert the estimated values $\hat{\beta}_0$ and $\hat{\beta}_1$ in the equation for the line and calculate $\hat{y}_i$ for each x_i :

```
yhat = -0.1886 + 0.7225 * df['pressure']

print(yhat)
[0.55, 1.31, 1.90, 2.71, 3.66, 4.02, 5.06, 5.56, 6.39, 7.03]
```

(here the output from Python is displayed slightly differently in order to save space - but you can compare the values to the output in your own Python code).

Note that the fitted values are also stored in the fitted model, so we can get them directly from `fit.fittedvalues`:

```
print(fit.fittedvalues)
[0.55, 1.31, 1.90, 2.71, 3.65, 4.02, 5.06, 5.56, 6.39, 7.03]
```

(again the output from Python is displayed slightly differently in order to save space).

We now compute **the variability explained by the regression line**:

```
n = 10
ybar = df['flux'].mean()

explained_var = np.sum((yhat - ybar)**2) / (n-1)

print(explained_var)
4.829
```

or equivalently:

```
print(fit.fittedvalues.var(ddof=1))
4.829
```

This expresses how much the points on the regression line vary around the mean $\bar{y}$, i.e. the part of the variation in flux that is due to the differences in pressure.

We can compare the variability explained (4.829, in units of (flow per area)²) to the total variability given by the sample variance which was 5.198 (in units of (flow per area)²). We see that the total variability in the data is a bit larger than the variability explained by the model.

3.

The variability NOT explained by the regression line (the *residual variation*) is

$$\frac{1}{n-1} \sum_{i=1}^{10} (y_i - \hat{y}_i)^2$$

It expresses how much the observations vary **around the regression line**, i.e. the part of the variation in flux which is left when we have accounted for the pressure (it is the *sample variance of the residuals*: $e_i = y_i - \hat{y}_i$).

We now compute **the variability NOT explained by the regression line**:

```
residual_var = np.sum((df['flux'] - yhat)**2) / (n-1)

print(residual_var)
0.369
```

Note that the residuals are also stored in the fitted model, so we can get them directly from `fit.resid` and calculate the variability as:

```
print(fit.resid.var(ddof=1))
0.369
```

We can compare the variability NOT explained (0.369, in units of (flow per area)²) to the total variability given by the sample variance which was 5.198 (in units of (flow per area)²). We see that the variability NOT explained is quite small compared to the total variability.

4. Question regarding proportion explained/NOT explained

We see that the sample variance is split into the two parts:

$$s_y^2 = 5.198 = 4.829 + 0.369$$

The proportion of the variation in the outcome (Y) explained by the regression line is equal to the *coefficient of determination* r^2 (see also Definition 5.25):

$$r^2 = \frac{4.829}{5.198} = 0.929 \quad \text{or } 92.9\%$$

and the proportion NOT explained is

$$1 - r^2 = \frac{0.369}{5.198} = 0.071 \quad \text{or } 7.1\%$$

So 92.9% of the flux-variation is explained by the pressure differences, and 7.1% of the flux-variation is NOT explained by the pressure differences in the dataset.

The value $r^2 = 0.929$ is the R-squared in the regression output above, and it is equal to the squared sample correlation coefficient: $\hat{\rho}^2 = 0.964^2 = 0.929$.

- d) Can you (at significance level $\alpha = 0.05$) reject the hypothesis that the line passes through (0,0)?

|||| Solution

The hypothesis is the same as stating the intercept is zero:

$$H_0 : \beta_0 = 0 \quad \text{vs.} \quad H_1 : \beta_0 \neq 0$$

The p -value for this hypothesis test is given in the output table and is: 0.681. This value is larger than 0.05. Therefore the answer to the question is: *No, since the relevant p -value is 0.68, which is larger than α .*

The same conclusion could be made by inspecting the 95% confidence interval for β_0 , since this interval includes the value zero.

- e) What is the expected (mean) flux at pressure 5.0 and what is the 95% confidence interval for this expected flux?

|||| Solution

At $x_{\text{new}} = 5.0$ the expected flux is estimated by the point on the line:

$$\hat{y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} = -0.1886 + 0.7225 \cdot 5.0 = 3.4239.$$

The 95% confidence interval for the expected flux is found using Method 5.18 (*confidence interval for the line*):

$$\hat{y}_{\text{new}} \pm t_{0.975, n-2} \cdot \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}},$$

where $n = 10$. To use this formula we need four quantities: the t -quantile $t_{0.975, n-2}$, the estimated error standard deviation $\hat{\sigma}$, the mean pressure $\bar{x}$ and S_{xx} . We find them one at a time.

The t -quantile: we need the 0.975 quantile of a t -distribution with $n - 2 = 10 - 2 = 8$ degrees of freedom:

```
print(stats.t.ppf(0.975, df=8))
2.306
```

The error standard deviation: the estimated error variance $\hat{\sigma}^2$ is stored in the fitted model as `fit.scale`, so $\hat{\sigma}$ is found by taking the square root:

```
sigma_hat = np.sqrt(fit.scale)
print(sigma_hat)
0.6446
```

The mean pressure:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

```
xbar = df['pressure'].mean()
print(xbar)
5.547
```

S_{xx} can be calculated from the sample variance of x :

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 = (n - 1) \cdot s_x^2$$

```
Sxx = (10-1) * df['pressure'].var(ddof=1)
print(Sxx)
83.26
```

Putting it all together: inserting the values in the formula gives

$$\begin{aligned} & 3.4239 \pm 2.306 \cdot 0.6446 \sqrt{\frac{1}{10} + \frac{(5.0 - 5.547)^2}{83.26}} \\ & 3.4239 \pm 2.306 \cdot 0.6446 \cdot 0.3219 \\ & 3.4239 \pm 0.478 \\ & [2.945; 3.902]. \end{aligned}$$

The 95% confidence interval for the expected flux is [2.945; 3.902].

The 95% confidence interval can also be computed using Python's inbuilt functions:

```
new_data = pd.DataFrame({'pressure': [5.0]})
print(fit.get_prediction(new_data).summary_frame(alpha=0.05))
```

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
0	3.423806	0.207473	2.945372	3.902239	1.862243	4.985368

where we see our calculations match the output for “mean_ci_lower” and “mean_ci_upper”.

- f) At pressure 5.0, what is the 95% prediction interval? What is the interpretation of this prediction interval?

|||| Solution

As in the previous question the point on the line at $x_{\text{new}} = 5.0$ is

$$\hat{y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} = -0.1886 + 0.7225 \cdot 5.0 = 3.4239.$$

The 95% prediction interval for a new observation is found using Method 5.18 (*prediction interval*):

$$\hat{y}_{\text{new}} \pm t_{0.975, n-2} \cdot \hat{\sigma} \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}},$$

where $n = 10$. The only difference from the confidence interval for the line is the extra 1 under the square root, which accounts for the variation of a single new observation around the line.

We need the same four quantities as in the previous question, which we have already found:

- the t -quantile: $t_{0.975, 8} = 2.306$,
- the error standard deviation: $\hat{\sigma} = 0.6446$,
- the mean pressure: $\bar{x} = 5.547$,
- $S_{xx} = (n - 1) \cdot s_x^2 = 83.26$.

Putting it all together: inserting the values in the formula gives

$$\begin{aligned} & 3.4239 \pm 2.306 \cdot 0.6446 \sqrt{1 + \frac{1}{10} + \frac{(5.0 - 5.547)^2}{83.26}} \\ & 3.4239 \pm 2.306 \cdot 0.6446 \cdot 1.0505 \\ & 3.4239 \pm 1.562 \\ & [1.862; 4.985]. \end{aligned}$$

The 95% prediction interval at pressure 5.0 is [1.862; 4.985]. As expected, it is much wider than the 95% confidence interval for the expected flux, [2.945; 3.902].

The 95% prediction interval can also be computed using Python’s inbuilt functions:

```
new_data = pd.DataFrame({'pressure': [5.0]})
print(fit.get_prediction(new_data).summary_frame(alpha=0.05))
```

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
0	3.423806	0.207473	2.945372	3.902239	1.862243	4.985368

where we see our calculations match the output for “obs_ci_lower” and “obs_ci_upper”.

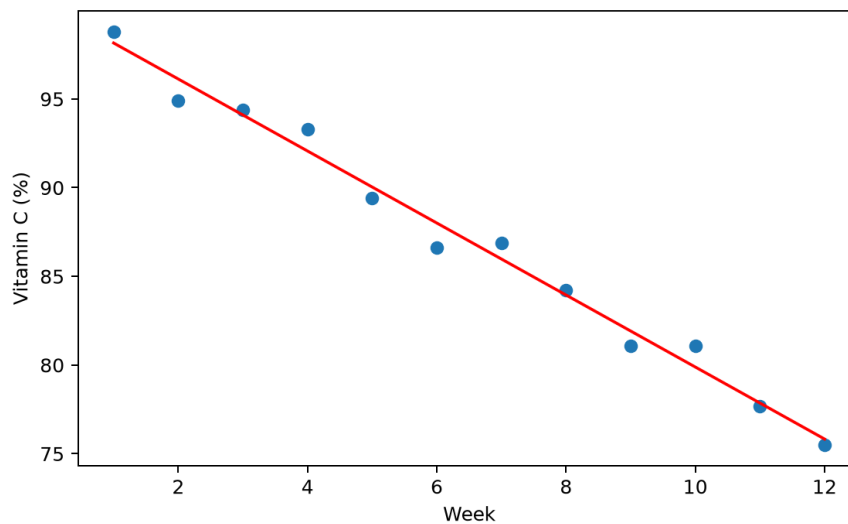
The interpretation of the prediction interval is that we expect 95% of new observations at pressure $x = 5.0$ to fall within the range given by the prediction interval.

Exam-style questions

A food company examines how the vitamin C content in bottled orange juice decreases during storage. In each of the weeks $1, 2, \dots, 12$ after bottling, one bottle is opened and the remaining vitamin C content is measured (in percent of the content at bottling). The following regression model is fitted to the data:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2),$$

where x_i is the storage time (in weeks) and Y_i is the vitamin C content (in %). The data and the estimated regression line are shown in the following figure:



The regression table from Python is:

```
fit = smf.ols('vitaminC ~ week', data=df).fit()
print(fit.summary(slim=True))
```

OLS Regression Results						
=====						
Dep. Variable:	vitaminC		R-squared:	0.985		
Model:	OLS		Adj. R-squared:	0.983		
No. Observations:	12		F-statistic:	647.2		
Covariance Type:	nonrobust		Prob (F-statistic):	2.02e-10		
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	100.1848	0.587	170.614	0.000	98.876	101.493
week	-2.0297	0.080	-25.440	0.000	-2.207	-1.852
=====						

Furthermore, the following values have been printed:

```
print(df['week'].mean())
6.5
print(df['week'].var(ddof=1))
13.0
print(np.sqrt(fit.scale))
0.9541
print(stats.t.ppf(0.95, df=10))
1.812
print(stats.t.ppf(0.975, df=10))
2.228
```

g) What is the 95% confidence interval for the expected vitamin C content after 10 weeks of storage?

- 1) [79.27; 80.50]
- 2) [77.59; 82.19]
- 3) [79.01; 80.76]
- 4) [77.73; 82.04]
- 5) [79.18; 80.60]

|||| Solution

The expected vitamin C content after $x_{\text{new}} = 10$ weeks is estimated by the point on the line:

$$\hat{y}_{\text{new}} = 100.1848 - 2.0297 \cdot 10 = 79.888.$$

The 95% confidence interval for the line is found using Method 5.18:

$$\hat{y}_{\text{new}} \pm t_{0.975, n-2} \cdot \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}}.$$

From the printed values we have $n = 12$, $t_{0.975,10} = 2.228$, $\hat{\sigma} = 0.9541$, $\bar{x} = 6.5$ and the sample variance of the weeks $s_x^2 = 13.0$, so

$$S_{xx} = (n - 1) \cdot s_x^2 = 11 \cdot 13.0 = 143.$$

Inserting the values gives

$$\begin{aligned} & 79.888 \pm 2.228 \cdot 0.9541 \sqrt{\frac{1}{12} + \frac{(10 - 6.5)^2}{143}} \\ & 79.888 \pm 2.228 \cdot 0.9541 \cdot 0.4111 \\ & 79.888 \pm 0.874 \\ & [79.01; 80.76]. \end{aligned}$$

So the correct answer is 3).

h) Which of the following statements is correct?

- 1) Since $R^2 = 0.985$, 98.5% of the observations are expected to lie exactly on the regression line.
- 2) The p -value for the slope (0.000) is the probability that the vitamin C content does not change with storage time.
- 3) The 95% confidence interval for the expected (mean) vitamin C content is narrowest in week 1 and week 12, since these are the endpoints of the data.
- 4) If the experiment were repeated with many more bottles, the 95% prediction interval for a single new bottle after 10 weeks would still not shrink towards zero width.
- 5) The model can safely be used to predict the vitamin C content after 60 weeks of storage.

|||| Solution

The correct answer is 4). The prediction interval includes the variation of a single new observation around the line (the extra 1 under the square root in Method 5.18). When n increases, the terms $\frac{1}{n}$ and $\frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}$ go towards zero, but the interval will still be approximately $\hat{y}_{\text{new}} \pm 1.96 \cdot \sigma$, so it does not shrink towards zero width.

The other statements are wrong:

- 1) R^2 is the proportion of the variation in Y explained by the regression line. It does not say anything about how many observations lie exactly on the line (in the figure, none of them do).
- 2) The p -value is the probability of observing a test statistic at least as extreme as the one observed, *given* that $H_0 : \beta_1 = 0$ is true. It is not the probability that H_0 is true.
- 3) The confidence interval for the line is narrowest at $x_{\text{new}} = \bar{x} = 6.5$, where the term $(x_{\text{new}} - \bar{x})^2$ is zero, and widest at the endpoints of the data.
- 5) 60 weeks is far outside the range of the data (1–12 weeks), and the model would predict $100.1848 - 2.0297 \cdot 60 = -21.6\%$, which is impossible. We cannot assume that the linear relationship holds outside the observed range.

5.7 Membrane pressure drop (matrix form)

|||| Exercise 5.7 Membrane pressure drop (matrix form)

This exercise uses the same data and regression model as in Exercise 5.6 above. We observe the following 10 related values of pressure (x) and flux (y):

	1	2	3	4	5	6	7	8	9	10
Pressure (x)	1.02	2.08	2.89	4.01	5.32	5.83	7.26	7.96	9.11	9.99
Flux (y)	1.15	0.85	1.56	1.72	4.32	5.07	5.00	5.31	6.17	7.04

The data can be stored in a Python DataFrame named `df` by copy-pasting the following code:

```
df = pd.DataFrame({
    'pressure':
        [1.02, 2.08, 2.89, 4.01, 5.32, 5.83, 7.26, 7.96, 9.11, 9.99],
    'flux':
        [1.15, 0.85, 1.56, 1.72, 4.32, 5.07, 5.00, 5.31, 6.17, 7.04]
})
```

The following regression model is fitted to the data:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2),$$

where x_i is the pressure and Y_i is the flux. The regression model can also be written in matrix form:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}),$$

where $\mathbf{X}$ is the design matrix, $\mathbf{Y}$ is the vector of outcome values (flux) and $\boldsymbol{\beta} = (\beta_0, \beta_1)^T$ is the vector of parameters. The resulting output table is shown here:

```
fit = smf.ols('flux ~ pressure', data=df).fit()
print(fit.summary(slim=True))
```

OLS Regression Results						
=====						
Dep. Variable:	flux	R-squared:	0.929			
Model:	OLS	Adj. R-squared:	0.920			
No. Observations:	10	F-statistic:	104.6			
Covariance Type:	nonrobust	Prob (F-statistic):	7.18e-06			
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	-0.1886	0.442	-0.427	0.681	-1.207	0.830
pressure	0.7225	0.071	10.227	0.000	0.560	0.885
=====						

- a) Write down (by hand) the design matrix X and the vector of outcome values Y corresponding to the linear regression model given above. How many rows and columns do they have, and what does each column of X contain?

See also Example 5.24.

|||| Solution

The design matrix X has one row for each observation. The first column is a column of ones (it belongs to the intercept β_0) and the second column contains the pressures x_i (it belongs to the slope β_1). The vector Y contains the observed fluxes:

$$X = \begin{pmatrix} 1 & 1.02 \\ 1 & 2.08 \\ \vdots & \vdots \\ 1 & 9.99 \end{pmatrix}, \quad Y = \begin{pmatrix} 1.15 \\ 0.85 \\ \vdots \\ 7.04 \end{pmatrix}.$$

With these, the model $Y = X\beta + \varepsilon$ says that each row is $Y_i = 1 \cdot \beta_0 + x_i \cdot \beta_1 + \varepsilon_i$, which is exactly the simple linear regression model. So X is a 10×2 matrix and Y is a vector of length 10.

- b) Reproduce the parameter estimates and standard errors from the output table above, but using matrix-vector calculations in Python.

Start by implementing the design matrix X and the vector Y from question a) in Python, and print them to check that they match your sketch.

You will need some matrix notation in Python:

- A column of n ones: `np.ones(n)`
- Combine two columns `x1` and `x2` into a matrix:
`np.column_stack((x1, x2))`
- A column from the data, e.g. the fluxes: `df['flux']`
- Matrix multiplication (XY): `np.dot(X, Y)` or `X@Y`
- Matrix transpose (X^T): `X.T`
- Matrix inverse (X^{-1}): `np.linalg.inv(X)`

See also Theorem 5.23 and Example 5.24.

|||| **Solution**

From Theorem 5.23 the parameter estimates are

$$\hat{\beta} = (X^T X)^{-1} X^T Y,$$

and the covariance matrix of the estimates is

$$V[\hat{\beta}] = \sigma^2 (X^T X)^{-1}, \quad \text{where} \quad \hat{\sigma}^2 = \frac{RSS}{n-2}.$$

We compute these one at a time.

The design matrix and the outcome vector: first we implement X and Y from question a) in Python. The design matrix is made from a column of $n = 10$ ones and the pressures:

```
n = 10
X = np.column_stack((np.ones(n), df['pressure']))
print(X)

[[1.    1.02]
 [1.    2.08]
 [1.    2.89]
 [1.    4.01]
 [1.    5.32]
 [1.    5.83]
 [1.    7.26]
 [1.    7.96]
 [1.    9.11]
 [1.    9.99]]
```

and the vector of outcome values is simply the flux column:

```
Y = df['flux']
print(Y)

0    1.15
1    0.85
2    1.56
3    1.72
4    4.32
5    5.07
6    5.00
7    5.31
```

```
8      6.17
9      7.04
Name: flux, dtype: float64
```

The parameter estimates, $\hat{\beta}$:

We implement the expression:

$$\hat{\beta} = (X^T X)^{-1} X^T Y,$$

in Python:

```
beta_hat = np.linalg.inv(XtX) @ X.T @ Y
print(beta_hat)

[-0.188574  0.722476]
```

So $\hat{\beta}_0 = -0.1886$ and $\hat{\beta}_1 = 0.7225$, which are the same as in the coef column of the table above.

The error standard deviation, $\hat{\sigma}$:

To compute standard errors on $\hat{\beta}_0$ and $\hat{\beta}_1$, we must first estimate $\hat{\sigma}$. We have:

$$\hat{\sigma}^2 = \frac{RSS}{n-2} \quad \text{so:} \quad \hat{\sigma} = \sqrt{\frac{RSS}{n-2}} :$$

where RSS is the residual sum of squares: $RSS = \sum_i e_i^2$.

The residuals are $e = Y - X\hat{\beta}$, which we compute in Python:

```
e = Y - X @ beta_hat
```

Now we can compute $\hat{\sigma}$:

```
sigma_hat = np.sqrt(np.sum(e**2) / (n - 2))
print(sigma_hat)

0.6446
```

So $\hat{\sigma} = 0.6446$ which is the same value as we found in Exercise 5.6 using `np.sqrt(fit.scale)`.

The covariance matrix of the estimates:

Now we can compute the covariance matrix of the estimates:

$$V[\hat{\beta}] = \hat{\sigma}^2 (X^T X)^{-1}$$

We insert $\hat{\sigma}^2$ in place of σ^2 :

```
V_beta = sigma_hat**2 * np.linalg.inv(XtX)
print(V_beta)

[[ 0.19510937 -0.027683   ]
 [-0.027683   0.00499063]]
```

The diagonal elements are the variances of the estimates, $V[\hat{\beta}_0] = 0.1951$ and $V[\hat{\beta}_1] = 0.00499$, and the off-diagonal elements are the covariance $\text{Cov}[\hat{\beta}_0, \hat{\beta}_1] = -0.0277$.

The standard errors: these are the square roots of the variances, i.e. of the diagonal elements:

```
se_beta = np.sqrt(np.diag(V_beta))
print(se_beta)

[0.44171187  0.07064436]
```

So $\hat{\sigma}_{\beta_0} = 0.442$ and $\hat{\sigma}_{\beta_1} = 0.071$, which are the same as in the std err column of the table above.

5.8 Independence and correlation

|||| Exercise 5.8 Independence and correlation

Consider the layout of independent variable in Example 5.11,

a) Show that $S_{xx} = \frac{n \cdot (n+1)}{12 \cdot (n-1)}$.

Hint: you can use the following relations

$$\sum_{i=1}^n i = \frac{n(n+1)}{2},$$

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}.$$

|||| Solution

$\bar{x}$ becomes

$$\begin{aligned}\bar{x} &= \frac{1}{n} \sum_{i=1}^n \frac{i-1}{n-1} = \frac{1}{n(n-1)} \sum_{i=1}^n (i-1) \\ &= \frac{1}{n(n-1)} \left(\frac{n(n+1)}{2} - n \right) = \frac{1}{2},\end{aligned}$$

and S_{xx} becomes

$$\begin{aligned}S_{xx} &= \sum_{i=1}^n \left(\frac{i-1}{n-1} - \frac{1}{2} \right)^2 \\ &= -\frac{n}{4} + \frac{1}{(n-1)^2} \sum_{i=1}^n (i^2 + 1 - 2i) \\ &= -\frac{n}{4} + \frac{1}{(n-1)^2} \left(\frac{n(n+1)(2n+1)}{6} - 6n^2 \right) \\ &= \frac{n}{(n-1)^2} \left(\frac{4n^2 + 6n + 2 - 12n - 3(n-1)^2}{12} \right) \\ &= \frac{n}{(n-1)^2} \left(\frac{n^2 - 1}{12} \right) = \frac{n(n+1)}{12(n-1)}.\end{aligned}$$

b) Show that the asymptotic correlation between $\hat{\beta}_0$ and $\hat{\beta}_1$ is

$$\lim_{n \rightarrow \infty} \rho_n(\hat{\beta}_0, \hat{\beta}_1) = -\frac{\sqrt{3}}{2}.$$

|||| Solution

The correlation between $\hat{\beta}_0$ and $\hat{\beta}_1$ is

$$\begin{aligned} \rho_n(\hat{\beta}_0, \hat{\beta}_1) &= \frac{\text{Cov}(\hat{\beta}_0, \hat{\beta}_1)}{\sqrt{V(\hat{\beta}_0) V(\hat{\beta}_1)}} \\ &= -\frac{\sigma^2 \bar{x} / S_{xx}}{\sqrt{\sigma^4 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right) \frac{1}{S_{xx}}}} \\ &= -\frac{\bar{x} / S_{xx}}{\frac{1}{S_{xx}} \sqrt{\left(\frac{S_{xx}}{n} + \bar{x}^2 \right)}} \\ &= -\frac{\bar{x}}{\sqrt{\frac{S_{xx}}{n} + \bar{x}^2}}. \end{aligned}$$

Notice that the correlation is not a function of the variance (σ^2), but only a function of the independent variables. Now insert the values of $\bar{x}$ and S_{xx}

$$\begin{aligned} \rho_n(\hat{\beta}_0, \hat{\beta}_1) &= -\frac{1}{2\sqrt{\frac{n+1}{12(n-1)} + \frac{1}{4}}} = -\frac{1}{2\sqrt{\frac{n+1+3(n-1)}{12(n-1)}}} \\ &= -\frac{1}{2\sqrt{\frac{2n-1}{6(n-1)}}} = -\frac{\sqrt{6(n-1)}}{2\sqrt{2n-1}} \\ &= -\frac{1}{2}\sqrt{\frac{6(n-1)}{2(n-1/2)}} = -\frac{\sqrt{3}}{2}\sqrt{\frac{n-1}{n-1/2}}, \end{aligned}$$

which converges to $-\frac{\sqrt{3}}{2}$ for $n \rightarrow \infty$.

Consider a layout of the independent variable where $n = 2k$ and $x_i = 0$ for $i \leq k$ and $x_i = 1$ for $k < i \leq n$.

c) Find S_{xx} for the new layout of x .

|||| Solution

$$\bar{x} = \frac{1}{2},$$

and

$$\begin{aligned} S_{xx}^{\text{new}} &= \sum_{i=1}^k \left(0 - \frac{1}{2}\right)^2 + \sum_{i=k+1}^{2k} \left(1 - \frac{1}{2}\right)^2 \\ &= \frac{k}{4} + \frac{k}{4} = \frac{k}{2} = \frac{n}{4}. \end{aligned}$$

d) Compare S_{xx} for the two layouts of x .

|||| Solution

$$\frac{S_{xx}}{S_{xx}^{\text{new}}} = \frac{n(n+1)}{12(n-1)} \frac{4}{n} = \frac{(n+1)}{3(n-1)} < 1; \quad \text{for } n > 2$$

which implies that $S_{xx}^{\text{new}} > S_{xx}$ for all $n > 2$.

e) What is the consequence for the parameter variance in the two layouts?

|||| Solution

The larger S_{xx} for the new layout implies that the parameter variance is smaller for the new layout (given that data comes from the same model).

f) Discuss pros and cons for the two layouts.

|||| **Solution**

The smaller parameter variance for the new layout would suggest that we should use this layout. However, we would not be able to check that data is in fact generated by a linear model. Consider e.g. data generated by the model

$$y_i = \beta_0 + \beta_1 x_i^2 + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2),$$

if we only look at $x_i = 0$ or $x_i = 1$ we will not be able to detect that the relationship is in fact non-linear.