Chapter 5

Simple Linear regression (solutions to exercises)

Contents

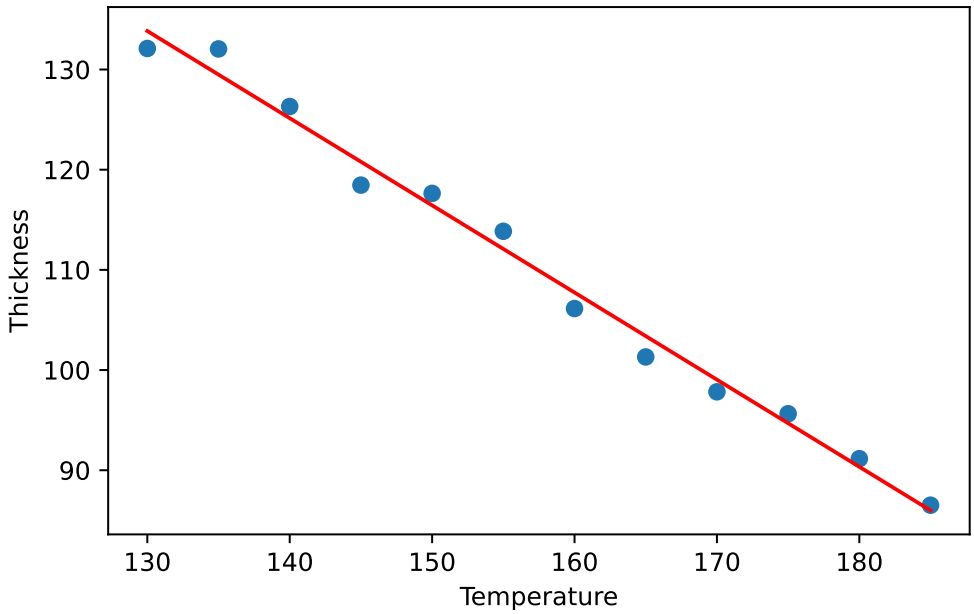
Import Python packages

```
# Import all needed python packages
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as stats
import statsmodels.formula.api as smf
import statsmodels.api as sm
```

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 set of values of temperature and the fold thickness were measured that is illustrated in the following figure:



a) Determine by looking at the figure, which of the following sets of estimates for the parameters in the usual 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

First of all, the only possible intercept ($\hat{\beta}_0$) among the ones given in the answers is 252. And then the slope estimate of -0.9 in these two options looks reasonable. We just 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 are NOT having an average vertical distance to the line in the size of 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) What is the only possible correct answer:

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 proportion of variation explained must be pretty high, so 0 can be ruled out. Answer 1 and 4 is also ruled out since the correlation clearly is negative. This also narrows the possibilities down to answer 3 and 5. And since the correlation is NOT exactly -1 (in which case the observations would be exactly on the line), 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 Celcius (t)	10	20	30	40	50	60	70	80	90
Life time in hours (y)	420	365	285	220	176	117	69	34	5

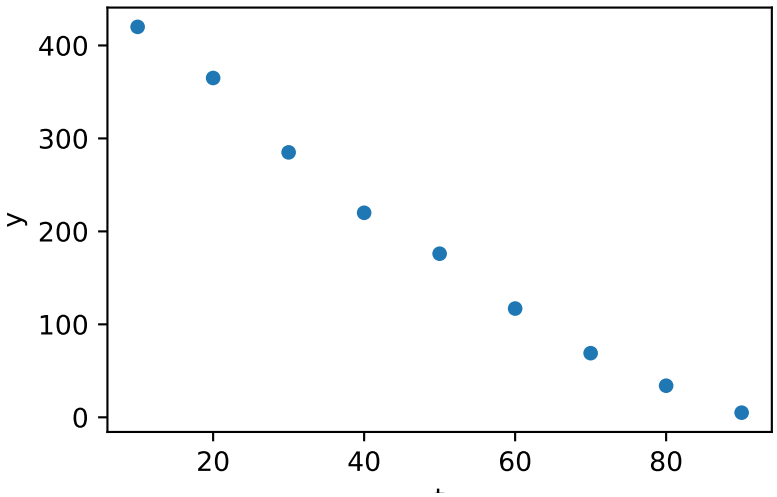
In this exercise you should use Python for computations, data visualization etc.

a) Analyse the relationship between Temperature and Life time (including data visualization and estimation of parameters in a relevant statistical model).

|||| 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. We estimate the parameters (intercept and slope) using Theorem 5.4:

```
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]
```

Since the slope is -5.3 (and has unit hours pr degree Celcius) we conclude that the average Life time of the electronic devices decrease by 5.3 hours for every one degree increase in temperature (Celcius).

b) Compute a 95% confidence interval for the slope in the linear regression model.

|||| Solution

First we estimate the standard errors on the intercept and slope using Theorem 5.8 (and equations 5-43 and 5-44). To do this we first need to compute all the residuals (ϵ_i)

```
# add residuals to dataframe:
df['residual'] = df['y'] - (beta0_hat + beta1_hat*df['t'])

# estimate standard deviation of residuals:
sigma_hat = np.sqrt(np.sum((df['residual'])**2/(9-2)))

# estimate standard error of parameters:
sigma_beta0_hat = sigma_hat*np.sqrt(1/9 + Tbar**2/Sxx)
sigma_beta1_hat = sigma_hat*np.sqrt(1/Sxx)

print([sigma_hat, sigma_beta0_hat, sigma_beta1_hat])

[19.81277445856666, 14.393646942675977, 0.2557818184006412]
```

Then we use the formula for the confidence interval for β_1 in Method 5.15:

$$\hat{\beta}_1 \pm t_{1-\alpha/2} \cdot \hat{\sigma}_{\beta_1}$$

```
# we need the following quantile from a t-distribution with n-2 = 9-2 =
7 degrees of freedom:
t0975 = stats.t.ppf(0.975, df=7)

CI_beta1_lower = beta1_hat - t0975 * sigma_beta1_hat
CI_beta1_upper = beta1_hat + t0975 * sigma_beta1_hat

print([CI_beta1_lower, CI_beta1_upper])

[-5.918161224091002, -4.708505442575665]
```

c) Can a relation between temperature and life time be documented on level 5%?

|||| Solution

Since the confidence interval does not include 0, we conclude that the slope parameter β_1 is statistically different from zero at significance level 0.05. Since the slope paratemer 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 no NO significant relationship between temperature and Life time").

To back up the analysis we also compute the corresponding p-value:

```
tobs = beta1_hat / sigma_beta1_hat

pvalue = 2*stats.t.cdf(tobs, df=7)

print(pvalue)

1.505038769216071e-07
```

We see that the p-values is (much) smaller than 0.05, so the relationship is ignificant.

d) Estimate the linear regression model using Pythons inbuilt function `fit = smf.ols(...).fit()`. Compare the output table with your own results (print the output table in Python using "`print(fit.summary(slim=True))`". Can you find the parameter estimates, their standard errors, confidence intervals and the corresponding test statistics and p-values? Make sure you understand what these p-values represent).

|||| Solution

```
fit = smf.ols('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
==============================================================================
```

In the bottom part of the output table we find a row for each parameter. The "Intercept" row is for $\hat{\beta}_0$ (the intercept) and the "t" row is for $\hat{\beta}_1$ (the slope with respect to temperature). For each parameter there is a parameter estimate in the column called "coef" (short for *coefficients*). The standard errors ($\hat{\sigma}_{\beta_0}$ and $\hat{\sigma}_{\beta_1}$) are found in the column "std err". The next two columns ("t" and "P>|t|") state the test statistic and p-values for the hypothesis test: testing if the parameter is equal to zero. Finally, the 95% confidence intervals for each parameter are also given in the output table.

Notice that the p-values in the table are rounded to zero, but they are not exactly zero(!). It is therefore necessary to print the p-values separately in Python:

```
print(fit.pvalues)

Intercept    8.376549e-09
t            1.505039e-07
dtype: float64
```

We can also print the estimated variance of the residuals ($\hat{\sigma}^2$)::

```
print(fit.scale)

392.54603174603125
```

and if we want the standard deviation ($\hat{\sigma}$):

```
print(np.sqrt(fit.scale))
```

```
19.812774458566658
```

These values all agree with our own computations.

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

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

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
==============================================================================


/home/pydni/.local/lib/python3.10/site-packages/statsmodels/stats/stattools.py:74: Valu
  warn("omni_normtest is not valid with less than 8 observations; %i "
```

Alternatively one could use hand calculations and use the formula in Theorem **??** 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 $x_{\text{new}} = 80$ °C.

|||| Solution

We use the formula in Equation (**??**) 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 og -0.4.

We use the basic definition of finding a quantile (from Definition **??**) and the upper quartile is $q_{0.75}$ (see Definition **??**). 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
==============================================================================


/home/pydni/.local/lib/python3.10/site-packages/statsmodels/stats/stattools.py:74: Valu
  warn("omni_normtest is not valid with less than 8 observations; %i "
```

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 Theorem **??** 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 of 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 is 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 polution

Exercise 5.5 Water polution

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
==============================================================================


/home/pydni/.local/lib/python3.10/site-packages/statsmodels/stats/stattools.py:74: Valu
  warn("omni_normtest is not valid with less than 8 observations; %i "
```

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 $SE_{\hat{\sigma}} = \hat{\sigma} = 0.348$.

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, that 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

Copy this into Python to avoid typing in the data:

```
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]
})
```

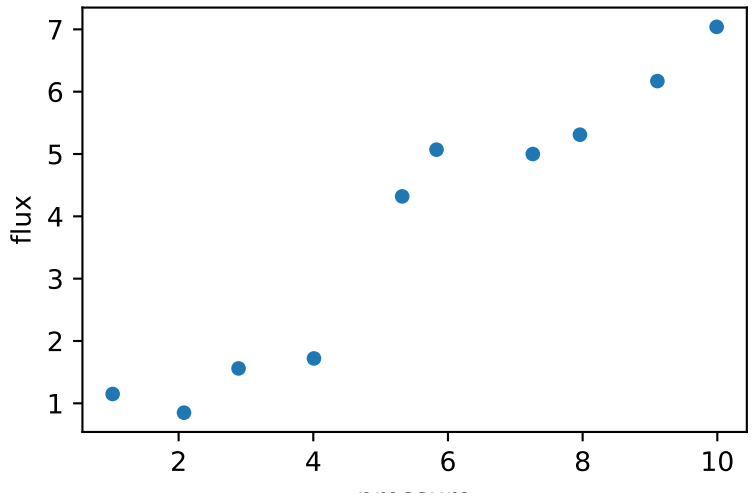
a) Estimate the empirical correlation between pressure and flux. Does the result match your expectations (from inspecting the data visually)?

|||| Solution

We plot the data and calculate the correlation:

```
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]
})

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




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

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

from the output we see that the correlation 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).

A linear regression model is estimated for the data and 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?

|||| 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:

```
# we need the following quantile from a t-distribution with n-2 = 10-2
= 8 degrees of freedom:
t095 = stats.t.ppf(0.95, df=8)
CI_beta1_lower = 0.7225 - t095 * 0.071
CI_beta1_upper = 0.7225 + t095 * 0.071
print([CI_beta1_lower, CI_beta1_upper])

[0.5904720893358782, 0.8545279106641218]
```

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

c) How large a part of the flux-variation ($\sum_{i=1}^{10}(y_i - \bar{y})^2$) is NOT explained by pressure differences?

|||| Solution

The R-squared (equal to the squared correlation: $r^2 = 0.964^2 = 0.929$) expresses the explained variation. This means that $1 - 0.929 = 0.071$ expresses the unexplained variation by the model. So the answer is 7.1% of the flux-variation is NOT explained by the pressure differences in the dataset.

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:

$$H_0 : \beta_0 = 0$$

The p-values 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.

e) What is the predicted average flux at pressure 5.0 and what is the 95% confidence interval for this predicted average flux?

Solution

At $x = 5.0$ the predicted flux is: $-0.1886 + 0.7225 \cdot 5.0 = 3.4239$.

The 95% confidence interval for this average flux is calculated using Method 5.18 (for *confidence intervals for the line*):

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

# compute Sxx to use in calculations:
Sxx = np.sum((df['pressure']-xbar)**2)

# estimate sigma:
df['residual'] = df['flux'] - (-0.1886 + 0.7225*df['pressure'])
sigma_hat = np.sqrt(np.sum((df['residual'])**2/(10-2)))
print(sigma_hat)

0.644606542602249


# estimate CI:
t0975 = stats.t.ppf(0.975, df=8)
upper = 3.4239 + t0975 * sigma_hat * np.sqrt(1/10 + (5.0-xbar)**2/Sxx)
lower = 3.4239 - t0975 * sigma_hat * np.sqrt(1/10 + (5.0-xbar)**2/Sxx)

print([lower, upper])

[2.945466659533394, 3.9023333404666065]
```

alternatively we can use Pythons 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 with the output for "mean_ci_lower" and "mean_ci_upper".

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

|||| Solution

The 95% prediction interval at pressure equal to 5.0 can be calculated using Method 5.18 (for *prediction intervals*):

```
# estimate prediction interval:
t0975 = stats.t.ppf(0.975, df=8)
upper = 3.4239 + t0975 * sigma_hat * np.sqrt(1 + 1/10 +
(5.0-xbar)**2/Sxx)
lower = 3.4239 - t0975 * sigma_hat * np.sqrt(1 + 1/10 +
(5.0-xbar)**2/Sxx)

print([lower, upper])

[1.862337415254085, 4.985462584745916]
```

alternatively we can use Pythons 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 with the output for "obs_ci_lower" and "obs_ci_upper".

The interpretation of the prediction interval is that we expect 95% of potential new observations at this exact pressure (at $x = 5$) to fall within the range given by the prediction interval.

5.7 Membrane pressure drop (matrix form)

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

This exercise uses the data presented in Exercise 6 above.

A linear regression model is estimated for the data and the resulting output table is shown here:

```
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]
})

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) Construct the Design Matrix X and the vector of outcome values (dependent variable) Y corresponding to the linear regression model given above.
You may want to make a matrix from vectors ($X = [x_1^T; x_2^T]$): `np.column_stack((x1,x2))`
See also Example 5.24.

|||| Solution

```
n = 10

# make the design matrix for the model:
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]]
```

```
# make the y-vector for the model:
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
```

b) Reproduce the parameter estimates and standard errors from the output table above, but using matrix vector calculations.
You will need some matrix notation in Python:
   - 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 Example 5.24.

|||| Solution

```
# Beta calculation
beta = np.linalg.inv(X.T @ X) @ X.T @ Y
print(beta)

[-0.18857437  0.722476  ]
```

These are the parameters estimates. Compare with the table above.

```
# Error term and standard error calculation
e = Y - X @ beta
s = np.sqrt(np.sum(e**2) / (n - 2))
print(s)

0.6446065267476389


Vbeta = s**2 * np.linalg.inv(X.T @ X)
se_beta = np.sqrt(np.diag(Vbeta))
print(se_beta)

[0.44171187 0.07064436]
```

These are the estimated standard errors of the parameters (and "s" is $\hat{\sigma}$). Compare with the table above.

5.8 Independence and correlation

|||| Exercise 5.8 Independence and correlation

Consider the layout of independent variable in Example **??**,

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

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

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

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

|||| Solution

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

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

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 $n \to \infty$.

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

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

which imply 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 imply that the parameter variance is smaller for the new layout (given that data comes from the same model).

f) Discuss pro's 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.