Computing for Mathematics: Handout 8

This handout contains a summary of the topics covered and an activity to carry out prior or during your lab session.

At the end of the handout is a specific coursework like exercise.

For further practice you can do the exercises available at the statistics chapter of Python for Mathematics.

1 Summary

The purpose of this handout is to cover statistics which corresponds to the probability chapter of Python for Mathematics.

The topics covered are:

- Calculating measures of central tendency and spread
- Calculating bivariate coefficients
- Fitting a line of best fit
- Using the Normal distribution

2 Activity

We will be tackling the problem from the tutorial of the statistics chapter of Python for Mathematics.

Anna is investigating the relationship between exercise and resting heart rate. She takes a random sample of 19 people in her year group and records for each person

- their resting heart rate, h beats per minute. - the number of minutes, m, spent exercising each week.

A table with the data is available in the statistics chapter of Python for Mathematics where you can also see a scatter plot.

1. For all collected values of h and m obtain:
    - The mean
    - The median
    - The quartiles
    - The standard deviation
    - The variation
    - The maximum
    - The minimum
2. Obtain the Pearson Coefficient of correlation for the variables h and m.
3. Obtain the line of best fit for variables x and y as defined by:

$$x = \ln(m) \qquad y = \ln(h)$$

4. Using the above obtain a relationship between m and h of the form:

$$h = cm^k$$

There are instructions for how to do all of this is in the probability chapter of Python for Mathematics.

1. Create the variables `h` and `m` which have values the data for h and m respectively.
2. Import the `statistics` library and use it to obtain the mean, media, quartiles, standard deviation and variation of both h and m.
3. Use the `min` and `max` tools to compute the minimum and maximum of both h and m.
4. Use the `statistics` library to compute the Pearson Coefficient of correlation between h and m.
5. Create `x` which has value $x = \ln(m)$.
6. Create `y` which has value $y = \ln h$.
7. Use the `statistics` library to compute the slope and intercept for a linear regression line between y and x.
8. Use `sympy` to obtain the required final expression for h as a function of m.

3 Coursework like exercise

1. Create a variable `distribution` that has value a Normal distribution with mean 10 and standard deviation .5:
2. Create a variable `probability_of_less_than_eight` which has value the probability of selecting a random variable from the distribution which has value less than eight.
3. Output the value of a variable from the distribution which has probability of being less than, .9.

4 Summary examples

Calculate the sample standard deviation of $1, 4, 2, 3, 1.5, 7$:

```
import statistics as st
data = (1, 4, 2, 3, 1.5, 7)
st.stdev(data)
```

Other tools exist in the statistics library to compute measures of spread and tendency: `mean`, `median`, `pstdev`, `stdev`, `pvariance`, `variance`, `quantiles`.

Calculate the maximum and minimum of $1, 2, 3$:

```
data = (1, 2, 3)
max(data), min(data)
```

Obtain the covariance between $1, 2, 3$ and $3, 2, 1$:

```
import statistics as st
x = (1, 2, 3)
y = (3, 2, 1)
st.covariance(x, y)
```

Obtain the Pearson correlation coefficient between $1, 2, 3$ and $3, 2, 1$:

```
import statistics as st
x = (1, 2, 3)
y = (3, 2, 1)
st.correlation(x, y)
```

Fit a line of best fit between $1, 2.5, 3$ and $2.9, 2, 1$:

```
import statistics as st
x = (1, 2.5, 3)
y = (2.9, 2, 1)
st.linear_regression(x, y)
```

Create an instance of the normal distribution with mean 5 and standard deviation 1:

```
import statistics as st
st.NormalDist(mu=5, sigma=1)
```