Linear regression and correlation

The ASTA team

Contents

1 The regression problem

1.1 We want to predict

- We will study the dataset `trees`, which is on the course website:

```
import pandas as pd

trees = pd.read_csv("https://asta.math.aau.dk/datasets?file=trees.txt", sep='\t')
trees.head()
```

```
##    Girth  Height  Volume
## 0    8.3      70    10.3
## 1    8.6      65    10.3
## 2    8.8      63    10.2
## 3   10.5      72    16.4
## 4   10.7      81    18.8
```

- In this experiment we have measurements of 3 variables for 31 randomly chosen trees:
- `Girth` numeric. Tree diameter in inches.
- `Height` numeric. Height in ft.
- `Volume` numeric. Volume of timber in cubic ft.
- We want to predict the tree volume, if we measure the tree height and/or the tree girth (diameter).
- This type of problem is called **regression**.

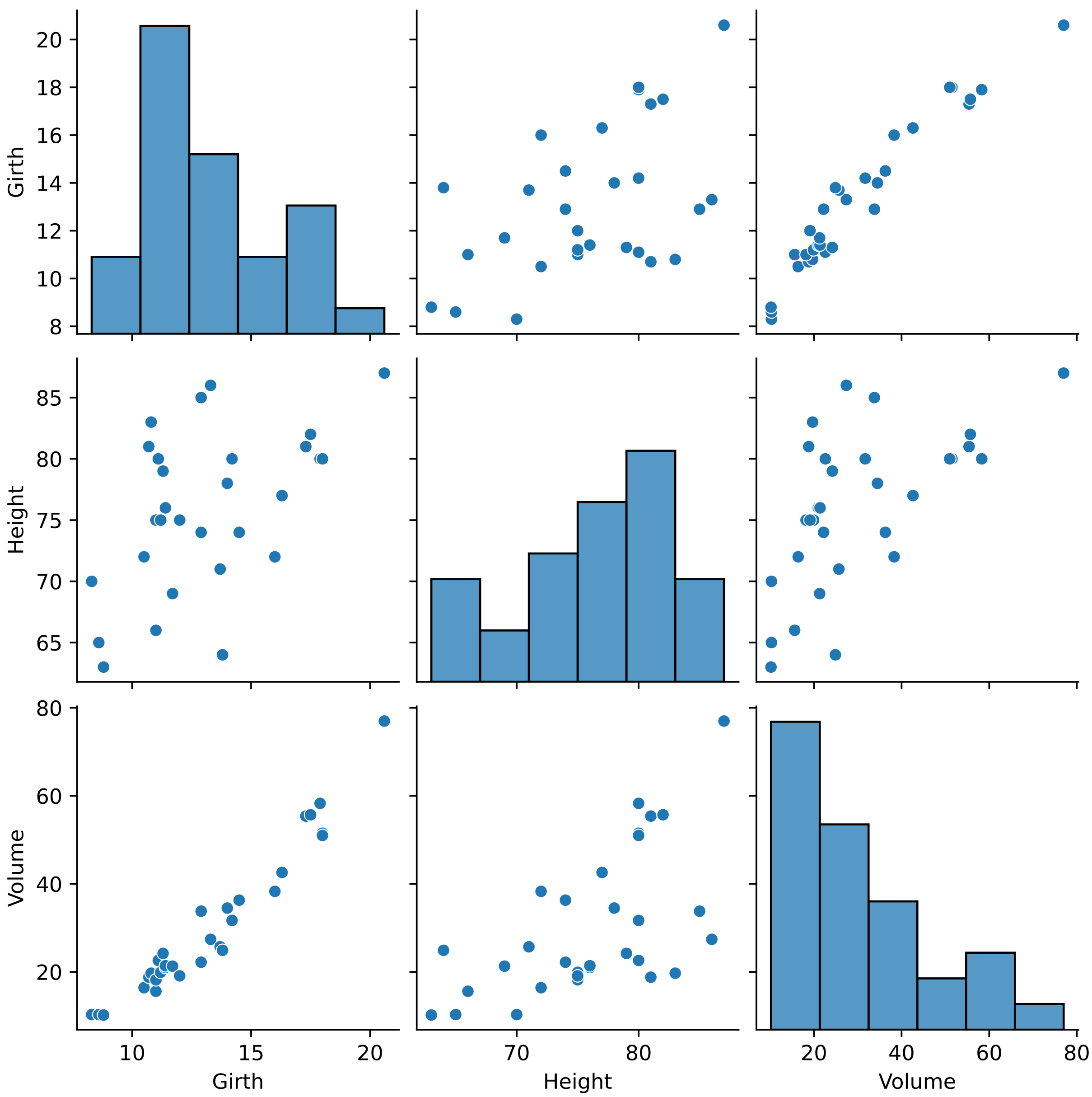
- Relevant terminology:
  - We measure a quantitative **response** y, e.g. `Volume`.
  - In connection with the response value y we also measure one (later we will consider several) potential **explanatory** variable x. Another name for the explanatory variable is **predictor**.

1.2 Initial graphics

- Any analysis starts with relevant graphics.

```
import seaborn as sns
import matplotlib.pyplot as plt

p = sns.pairplot(trees)
```



- For each combination of the variables we plot the (x, y) values.

- It looks like `Girth` is a good predictor for `Volume`.
- If we only are interested in the association between two (and not three or more) variables we use the usual `lmplot` function.

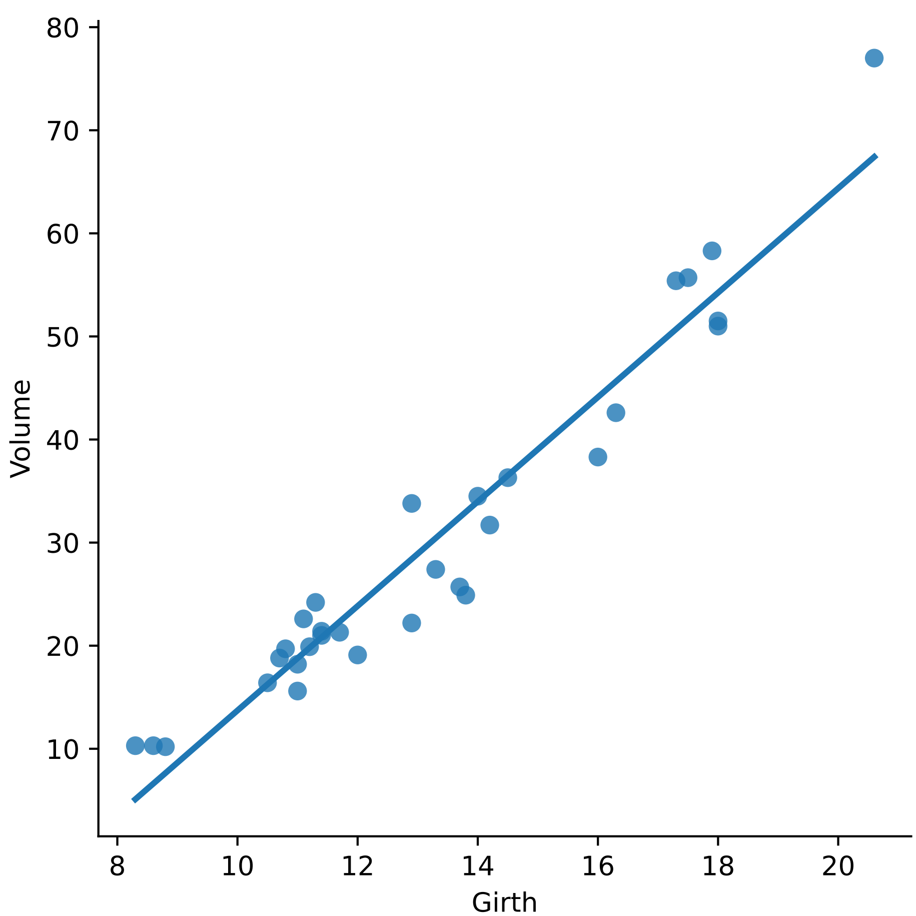
1.3 Simple linear regression

- We choose to use `x=Girth` as predictor for `y=Volume`. When we only use one predictor we are doing **simple regression**.
- The simplest **model** to describe an association between **response** y and a **predictor** x is **simple linear regression**.
- I.e. ideally we see the picture

$$y(x) = \alpha + \beta x$$

  where
  - α is called the `Intercept` - the line's intercept with the y-axis, corresponding to the response for $x = 0$.
  - β is called `Slope` - the line's slope, corresponding to the change in response, when we increase the predictor by one unit.

```
p = sns.lmplot(x='Girth', y='Volume', data=trees, ci=None)
```



1.4 Model for linear regression

- Assume we have a sample with joint measurements (x, y) of predictor and response.

- Ideally the model states that

$$y(x) = \alpha + \beta x,$$

but due to random variation there are deviations from the line.
- What we observe can then be described by

$$y = \alpha + \beta x + \varepsilon,$$

where ε is a **random error**, which causes deviations from the line.
- We will continue under the following **fundamental assumption**:
  - The errors ε are normally distributed with mean zero and standard deviation $\sigma_{y|x}$.
- We call $\sigma_{y|x}$ the **conditional standard deviation** given x, since it describes the variation in y around the regression line, when we know x.

1.5 Least squares

- In summary, we have a model with 3 parameters:
  - (α, β) which determine the line
  - $\sigma_{y|x}$ which is the standard deviation of the deviations from the line.
- How are these estimated, when we have a sample $(x_1, y_1) \dots (x_n, y_n)$ of (x, y) values??
- To do this we focus on the errors

$$\varepsilon_i = y_i - \alpha - \beta x_i$$

which should be as close to 0 as possible in order to fit the data best possible.
- We will choose the line, which minimizes the sum of squares of the errors:

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2.$$

- If we set the partial derivatives to zero we obtain two linear equations for the unknowns (α, β), where the solution (a, b) is given by:

$$b = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \quad \text{and} \quad a = \bar{y} - b\bar{x}$$

1.6 The prediction equation and residuals

- The equation for the estimates $(\hat{\alpha}, \hat{\beta}) = (a, b)$,

$$\hat{y} = a + bx$$

is called **the prediction equation**, since it can be used to predict y for any value of x.
- Note: The prediction equation is determined by the current sample. I.e. there is an uncertainty attached to it. A new sample would without any doubt give a different prediction equation.
- Our best estimate of the errors is

$$e_i = y_i - \hat{y} = y_i - a - bx_i,$$

i.e. the vertical deviations from the prediction line.
- These quantities are called **residuals**.
- We have that
  - The prediction line passes through the point $(\bar{x}, \bar{y})$.
  - The sum of the residuals is zero.

1.7 Estimation of conditional standard deviation

- To estimate $\sigma_{y|x}$ we need **Sum of Squared Errors**

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

  which is the squared distance between the model and data.

- We then estimate $\sigma_{y|x}$ by the quantity

$$s_{y|x} = \sqrt{\frac{SSE}{n-2}}$$

- Instead of n we divide SSE with **the degrees of freedom** $df = n - 2$. Theory shows, that this is reasonable.
- The degrees of freedom df are determined as the sample size minus the number of parameters in the regression equation.
- In the current setup we have 2 parameters: (α, β).

1.8 Example in R

```
import numpy as np
import statsmodels.formula.api as smf

model = smf.ols('Volume ~ Girth', data=trees).fit()
model.summary(slim = True) # text output
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                 Volume   R-squared:                       0.935
## Model:                            OLS   Adj. R-squared:                  0.933
## No. Observations:                  31   F-statistic:                     419.4
## Covariance Type:            nonrobust   Prob (F-statistic):           8.64e-19
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept    -36.9435      3.365    -10.978      0.000     -43.826     -30.061
## Girth          5.0659      0.247     20.478      0.000       4.560       5.572
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

```
[model.resid.min(), model.resid.max(), model.resid.median()]
```

```
## [np.float64(-8.065359510665438), np.float64(9.586816813996933), np.float64(0.15196668471900665)]
```

```
np.sqrt(model.mse_resid)
```

```
## np.float64(4.2519875217291965)
```

- The estimated residuals vary from -8.065 to 9.578 with median 0.152.
- The estimate of `Intercept` is $a = -36.9435$

- The estimate of slope of `Girth` is $b = 5.0659$
- The estimate of the conditional standard deviation (also called residual standard error) is $s_{y|x} = 4.252$ with $31 - 2 = 29$ degrees of freedom.

1.9 Test for independence

- We consider the regression model

$$y = \alpha + \beta x + \varepsilon$$

  where we use a sample to obtain estimates (a, b) of (α, β), an estimate $s_{y|x}$ of $\sigma_{y|x}$ and the degrees of freedom $df = n - 2$.
- We are going to test

$$H_0 : \beta = 0 \quad \text{against} \quad H_a : \beta \neq 0$$

- The null hypothesis specifies, that y **doesn't** depend linearly on x.
- In other words the question is: Is the value of b far away from zero?
- It can be shown that b has standard error

$$se_b = \frac{s_{y|x}}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

  with df degrees of freedom.
- So, we want to use the test statistic

$$t_{\text{obs}} = \frac{b}{se_b}$$

  which has to be evaluated in a t-distribution with df degrees of freedom.

1.10 Example

- Recall the summary of our example:

```
coef_table = model.summary2().tables[1] # data output
coef_table
```

```
##                 Coef.  Std.Err.          t         P>|t|     [0.025     0.975]
## Intercept -36.943459  3.365145 -10.978267  7.621449e-12 -43.825953 -30.060965
## Girth       5.065856  0.247377  20.478288  8.644334e-19   4.559914   5.571799
```

```
np.sqrt(model.mse_resid)
```

```
## np.float64(4.2519875217291965)
```

```
model.df_resid
```

```
## np.float64(29.0)
```

- As we noted previously $b = 5.0659$ and $s_{y|x} = 4.252$ with $df = 29$ degrees of freedom.
- In the second column(`Std. Error`) of the `Coefficients` table we find $se_b = 0.2474$.
- The observed t-score (test statistic) is then

$$t_{\text{obs}} = \frac{b}{se_b} = \frac{5.0659}{0.2474} = 20.48$$

  which also can be found in the third column (`t value`).
- The corresponding p-value is found in the usual way by using the t-distribution with 29 degrees of freedom.
- In the fourth column(`Pr(>|t|)`) we see that the p-value is less than 2×10^{-16}. This is no surprise since the t-score was way above 3.

1.11 Confidence interval for slope

- When we have both the standard error and the reference distribution, we can construct a confidence interval in the usual way:

$$b \pm t_{crit} se_b,$$

  where the t-score is determined by the confidence level.

- The t-score can be found using `t.ppf` in **Python**: In our example we have 29 degrees of freedom and with a confidence level of 95% we get $t_{crit} =$ `from scipy import stats; stats.t.ppf(0.975, df=29)`$= 2.045$.

- If you are lazy (like most statisticians are):

```
model.conf_int(alpha = 0.05)
```

```
##                   0          1
## Intercept -43.825953 -30.060965
## Girth       4.559914   5.571799
```

- i.e. $(4.56, 5.57)$ is a 95% confidence interval for the slope of `Girth`.

1.12 Correlation

- The estimated slope b in a linear regression doesn't say anything about the strength of association between y and x.
- `Girth` was measured in inches, but if we rather measured it in kilometers the slope is much larger: An increase of 1km in `Girth` yield an enormous increase in `Volume`.
- Let s_y and s_x denote the sample standard deviation of y and x, respectively.
- The corresponding t-scores

$$y_t = \frac{y}{s_y} \quad \text{and} \quad x_t = \frac{x}{s_x}$$

  are independent of the chosen measurement scale.
- The corresponding prediction equation is then

$$\hat{y}_t = \frac{a}{s_y} + \frac{s_x}{s_y} b x_t$$

- i.e. **the standardized regression coefficient** (slope) is

$$r = \frac{s_x}{s_y} b$$

  which also is called **the correlation** between y and x.

- It can be shown that:
  - $-1 \leq r \leq 1$
  - The absolute value of r measures the (linear) strength of dependence between y and x.
  - When $r = 1$ all the points are on the prediction line, which has positive slope.
  - When $r = -1$ all the points are on the prediction line, which has negative slope.
- To calculate the correlation:

```
trees.corr()
```

```
##            Girth   Height    Volume
## Girth   1.000000  0.51928  0.967119
## Height  0.519280  1.00000  0.598250
## Volume  0.967119  0.59825  1.000000
```

- There is a strong positive correlation between `Volume` and `Girth` (r=0.967).
- Note, it only works when all columns are numeric. Otherwise the relevant numeric columns should be extracted like this:

```
trees[['Height', 'Girth', 'Volume']].corr()
```

```
##           Height     Girth    Volume
## Height  1.00000  0.519280  0.598250
## Girth   0.51928  1.000000  0.967119
## Volume  0.59825  0.967119  1.000000
```

which produces the same output as above.

- Alternatively, one can calculate the correlation between two variables of interest like:

```
trees['Height'].corr(trees['Volume'])
```

```
## np.float64(0.5982496519917821)
```

2 R-squared: Reduction in prediction error

2.1 R-squared: Reduction in prediction error

- We want to compare two different models used to predict the response y.
- Model 1: We do not use the knowledge of x, and use $\bar{y}$ to predict any y-measurement. The corresponding prediction error is defined as

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

  and is called the **Total Sum of Squares**.
- Model 2: We use the prediction equation $\hat{y} = a + bx$ to predict y_i. The corresponding prediction error is then the Sum of Squared Errors

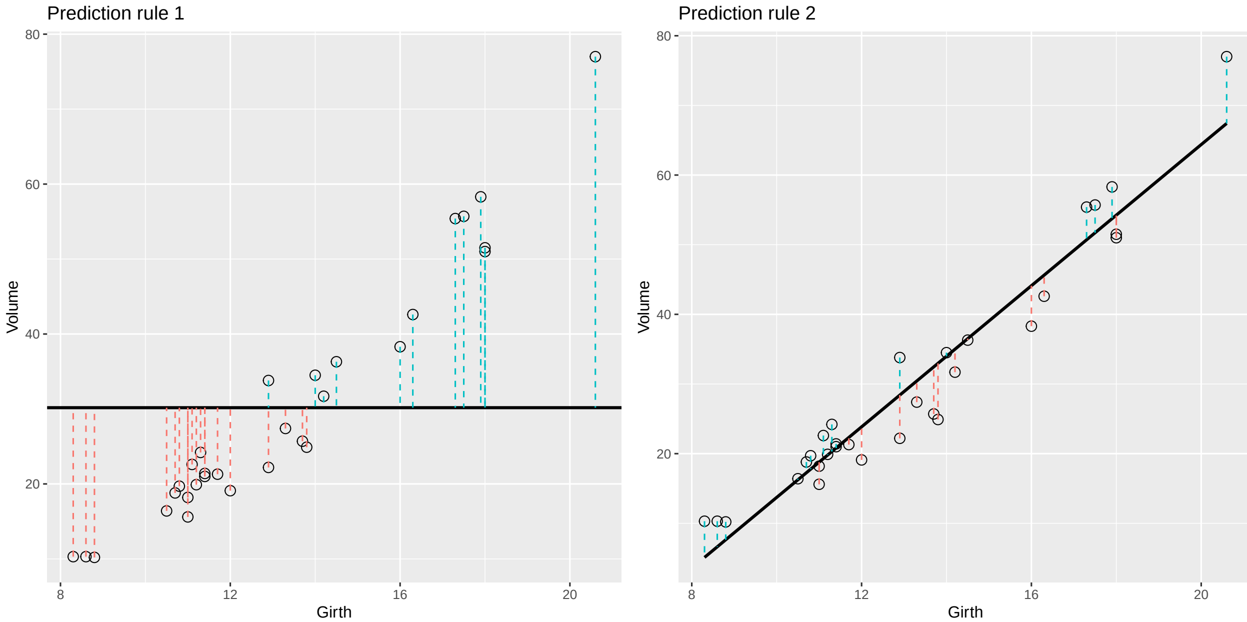
$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

- We then define

$$r^2 = \frac{TSS - SSE}{TSS}$$

  which can be interpreted as the relative reduction in the prediction error, when we include x as explanatory variable.
- This is also called the **fraction of explained variation**, **coefficient of determination** or simply **r-squared**.
- For example if $r^2 = 0.65$, the interpretation is that x explains about 65% of the variation in y, whereas the rest is due to other sources of random variation.

2.2 Graphical illustration of sums of squares



- Note the data points are the same in both plots. Only the prediction rule changes.
- The error of using Rule 1 is the total sum of squares $E_1 = TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$.
- The error of using Rule 2 is the residual sum of squares (sum of squared errors) $E_2 = SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.

2.3 r^2: Reduction in prediction error

- For the simple linear regression we have that

$$r^2 = \frac{TSS - SSE}{TSS}$$

is equal to the square of the correlation between y and x, so it makes sense to denote it r^2.

- At the top of the output below we can read off the value R-squared $= r^2 = 0.9353 = 93.53\%$, which is a large fraction of explained variation.

```
model.summary(slim = True)
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                                 OLS Regression Results
## ==============================================================================
## Dep. Variable:                 Volume   R-squared:                       0.935
## Model:                            OLS   Adj. R-squared:                  0.933
## No. Observations:                  31   F-statistic:                     419.4
## Covariance Type:            nonrobust   Prob (F-statistic):           8.64e-19
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept    -36.9435      3.365    -10.978      0.000     -43.826     -30.061
## Girth          5.0659      0.247     20.478      0.000       4.560       5.572
## ==============================================================================
##
## Notes:
```

```
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```