

4 Bivariate Statistics

4.1 Introduction

4.2 Correlation Coefficients

The null hypothesis of no correlation is rejected if the calculated t is greater than the critical t ($n-2$ degrees of freedom, $\alpha=0.05$). This test is also only valid if both variables are Gaussian distributed.

$$r_{xy} = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}$$

The correlation coefficient is well below the level required for a strong linear correlation.

According to this test, we can reject the null hypothesis, that is, that the correlation coefficient is significant, if the calculated t is larger than the critical t ($n-2$ degrees of freedom, $\alpha=0.05$).

This result indeed indicates that we can reject the null hypothesis, and therefore the correlation coefficient is significant.

```
print(np.mean(rhos1000[:,1]))  
0.9331641766158479
```

This value is similar to our first result of $r=0.9332$, but we now have confidence in the validity of the result.

4.3 Classical Linear Regression Analysis

Classical linear regression offers another way of describing the relationship between the two variables x and y .

Statistically testing the significance of the linear model provides some insights into the accuracy of these predictions.

Classical linear regression assumes that y responds to x and that the entire dispersion in the data set is contained within the y -value (Fig. 4.4).

The dependent variable y (also known as response variable or regressand) contains errors because its magnitude cannot be determined accurately.

Both the data and the fitted line can be plotted on the same graph (Fig. 4.5):

Instead of using the equation for the regression line, we can also use

`poly1d()` to calculate the y-values:

4.4 Analyzing Residuals

```
n_obs,e = np.histogram(res,bins=6)
v = np.diff(e[0:2])*0.5+e[0:-1]

n_exp = stats.norm.pdf(v,np.mean(res),np.std(res,ddof=1))

n_exp = n_exp/np.sum(n_exp)
n_exp = np.sum(n_obs)*n_exp

chisq,p = stats.chisquare(n_obs,n_exp,ddof=0,axis=0)
print(chisq)
print(p)

6.341920672629337
0.27435548829477924
```

The result means that we cannot reject the null hypothesis without another cause at a 5% significance level. The p -value of 0.2744 (or ~28%, which is much greater than the significance level) means that the chances of observing either the same result or a more extreme result from similar experiments in which the null hypothesis is true would be 2,744 in 10,000. Alternatively, we can use `normaltest()` to perform the χ^2 -test,

```
chisq,p = stats.normaltest(res)
print(chisq)
print(p)
```

which yields

```
1.4387222292581452
0.4870633342166629
```

Again, the result means that we cannot reject the null hypothesis without another cause at a 5% significance level. The p -value of 0.4970 (or ~50%, which is much greater than the significance level) means that the chances of observing either the same result or a more extreme result from similar experiments in which the null hypothesis is true would be 4,960 in 10,000. However, the quality of the result is not very good because the sample size of 30 measurements is very small.

4.5 Bootstrap Estimates of Regression Coefficients

```
ind = np.linspace(0, 29, 30, dtype=int)
```

4.6 Jackknife Estimates of Regression Coefficients

The **median** of the i subsamples provides an improved estimate of the regression coefficients.

4.7 Cross-Validation

A third method for testing the quality of the result of a regression analysis involves *cross-validation*. In the *leave-one-out* cross-validation method used here, the regression line is **calculated** using $n-1$ data points.

```
j_meters = meters  
j_age = age
```

The standard deviation of the deviations of the data points from the predicted line **is calculated using**

4.8 Reduced Major Axis Regression

4.9 Curvilinear Regression

4.10 Nonlinear and Weighted Regression

In such cases, **SciPy** provides various tools for fitting nonlinear models to the data.

The function `curve_fit()` returns a vector `popt` of coefficient estimates for the nonlinear regression of the responses in `data2` to the **regressors** in `data1` using the model specified by `func()`.

Recommended Reading

Figure Captions

Fig. 4.8 Curvilinear regression from measurements of barium content. The plot shows the original data points (circles), the regression line for a **second-degree** polynomial (solid line), and the prediction error (dashed lines) of the regression.