

Take home 2

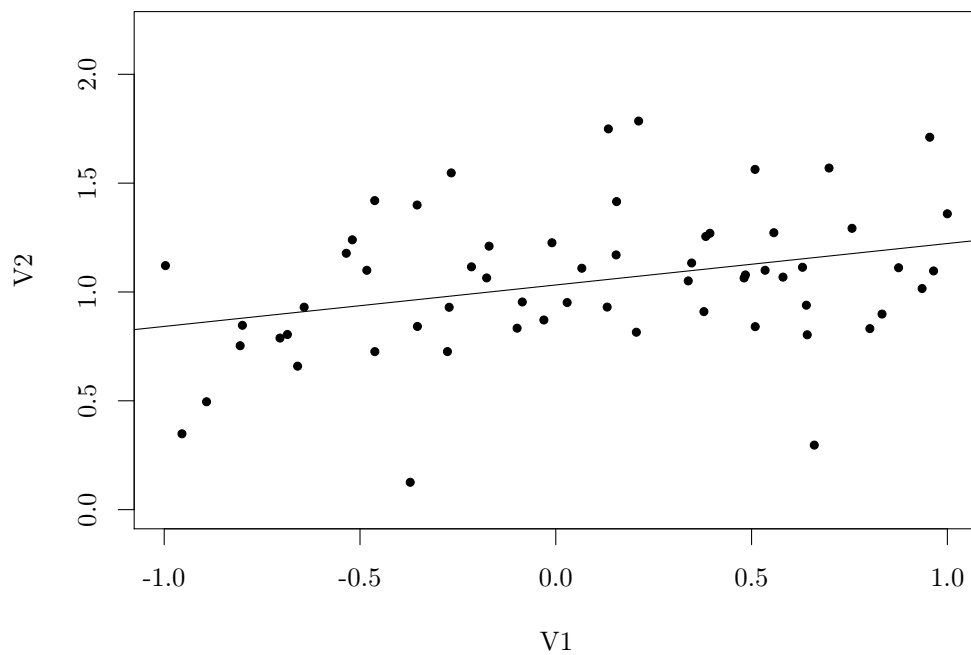
Pieter Luyten

13 december 2019

Question 1

(a)

The value for the intercept of the fit is 1.0321764 and for the slope of the fit is 0.1904323



Figuur 1: line fit through the data in Ex1.txt

(b)

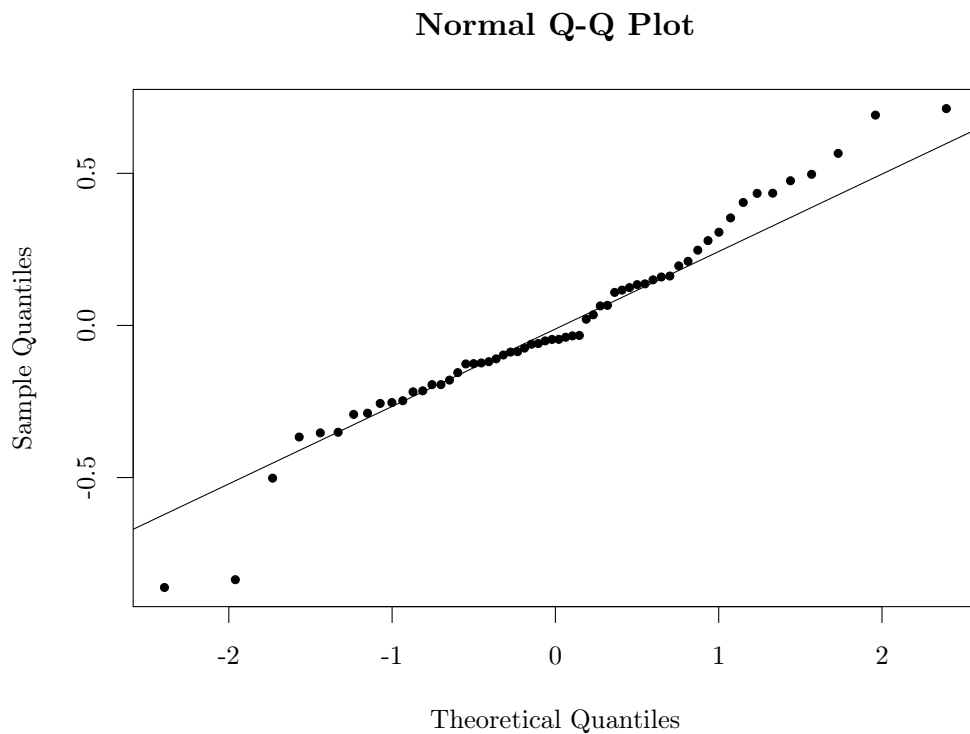
Using the result from section 7.4.1 in <REFERENCE cursus> we know that the random variable

$$T = \frac{\beta_1}{\sqrt{\frac{S^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}} \quad (1)$$

has a Student distribution with $n - 2$ degrees of freedom. The test value is 2.603. Using a student-t distribution with $60 - 2 = 58$ degrees of freedom we find a p-value of 0.0117. At the confidence level $\alpha = 0.01$, the null hypothesis that $\beta_1 = 0$ holds. The 99% confidence region for the test value is $[-2.663, 2.663]$.

(c)

A q-q plot is a plot where the quantiles of the assumed distribution are plotted against the quantiles from the sample. So in a sample of n points where we label the observation $x_i, i \in \{1, 2, \dots, n\}$ from lowest to highest, the i th point will be plotted at the $(Q(i/n), x_i)$. Here $Q(x)$ is the quantile function of the assumed distribution. This is a function such that $P(X < Q(p)) = p$. If the assumed distribution is a good model for the observed sample, the points will well fitted by a linear function.



Figuur 2: qq plot for the errors on the fit

(e)

We use the formula from section 7.5.4 in the book to find a confidence region for β . This region is of the form:

$$\left\{ \beta \left| \frac{(\hat{\beta} - \beta)^T (X^T X)(\hat{\beta} - \beta)}{p S^2} \leq F_{p, n-p}(1 - \alpha) \right. \right\} \quad (2)$$

Because the matrix $X^T X$ is positive definite, this is an ellips with center β and axes along the eigenvectors of $X^T X$. The upper bound for this expression is in this case: $F_{2, 58}(0.99) = 4.9910$. Filling in the values for $X, p = 2, n = 60$ and S^2 we get the following result (obtained using the cAS system sympy):

$$\{\beta | 5.2596\alpha^2 + 0.8222\alpha\beta - 0.4950\alpha + 1.6711\beta^2 - 0.6629\beta + 0.0711 \leq 4.9910\}$$

Question 2

(a)

The value for $\hat{\beta}_{OLS}$ obtained using ordinary least squares with the correlated errors is:

$$\hat{\beta}_{OLS} = (3.923161, 1.214355)^T \quad (3)$$

(b)

we know that the variance is the same for all ϵ_i and that they obey the recursive relation:

$$\epsilon_i = \rho\epsilon_{i-1} + \eta_i$$

with η_i standard normally distributed. Using $Var(\epsilon_i) = Var(\epsilon_j)$ for all $i, j \in \{1, 2, \dots, 80\}$ we find:

$$\begin{aligned} Var(\epsilon_i) &= \rho^2 Var(\epsilon_{i-1}) + Var(\eta_i) \\ \Leftrightarrow Var(\epsilon_i) &= \rho^2 Var(\epsilon_i) + 1 \\ \Leftrightarrow Var(\epsilon_i) &= \frac{1}{1 - \rho^2} \\ &= \frac{25}{9} \\ &= 2.777 \dots \end{aligned}$$

We conclude that the variance of ϵ is $\frac{25}{9}$.

(c)

To transform the errors to a basis where they are no longer correlated we will use the variance-covariance matrix V of the errors. The off-diagonal elements, so the covariances, are (suppose $i < j$):

$$\begin{aligned} Cov(\epsilon_i, \epsilon_j) &= E[(\epsilon_i - E(\epsilon_i))(\epsilon_j - E(\epsilon_j))] \\ &= E[\epsilon_i \epsilon_j] \\ &= E[\epsilon_i (\rho \epsilon_{j-1} + \eta_j)] \\ &\vdots \\ &= E[\epsilon_i (\rho^{j-i} \epsilon_i + \rho^{j-i-1} \eta_i + \rho^{j-i-2} \eta_{i-1} + \dots + \eta_j)] \\ &= E[\epsilon_i \rho^{i-j} \epsilon_i] \\ &= \rho^{i-j} Var(\epsilon) \\ &= \rho^{i-j} \frac{25}{9} \end{aligned}$$

So the matrix V has $\frac{25}{9}$ on the diagonal, $\frac{25}{9}\rho$ on the two sub-diagonals next to the diagonal, $\frac{25}{9}\rho^k$ on the k th sub-diagonal. the matrix V can be diagonalized with an orthogonal (or unitary) transformation U : $\Lambda = MVMT^T$. When we use the definition of the variance-covariance matrix (where ϵ denotes the column vector with the errors) we find:

$$\begin{aligned} V &= E((\epsilon - E(\epsilon))(\epsilon - E(\epsilon))^T) \\ \Leftrightarrow \Lambda &= UE(\epsilon\epsilon^T)U^T \\ \Leftrightarrow \Lambda &= E((U\epsilon)(U\epsilon)^T) \end{aligned}$$

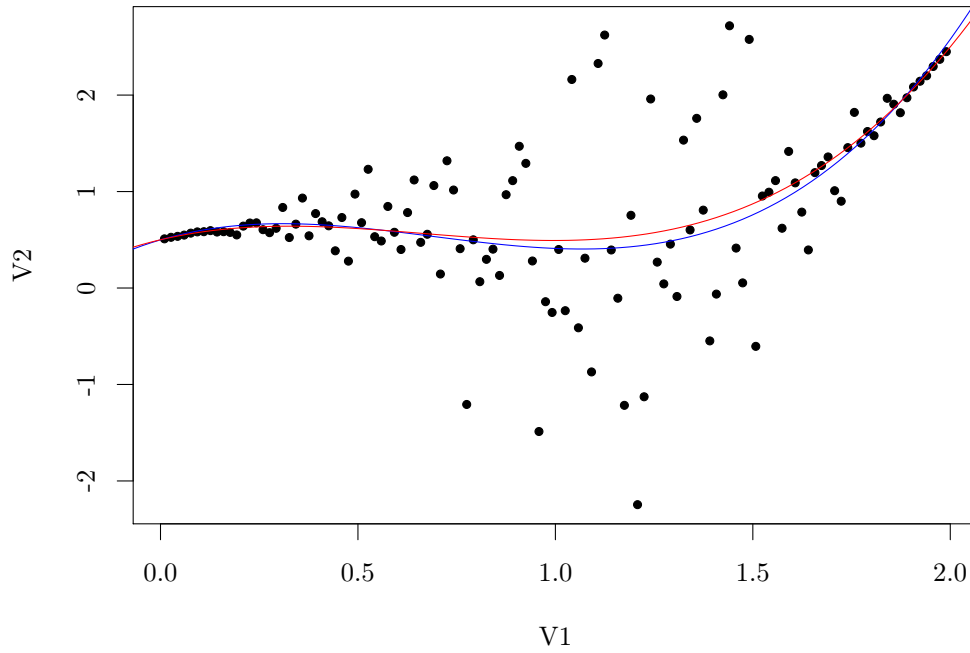


Figure 3: fits of a cubic function using ordinary least square (blue) and weighted least squares (red)

Where we used that $E(\epsilon = \mathbf{0})$. The vector $U\epsilon$ has variance-covariance matrix Λ so we conclude that it is a vector of uncorrelated stochastic variables.

The value for $\hat{\beta}_{OLS}$ that we find using these uncorrelated errors is exactly the same as with the uncorrelated errors:

$$\hat{\beta}_{OLS} = (3.923161, 1.214355)^T$$

This is expected because we used a unitary transformation to get uncorrelated errors and by filling this in the expression for the Sum of squares of the remainders we find:

$$\begin{aligned} RSS(\hat{\beta}_{Uncorrelated}) &= (UY - UX\beta)^T (UY - UX\beta) \\ &= (Y - X\beta)^T U^T U (Y - X\beta) \\ &= (Y - X\beta)^T I (Y - X\beta) \\ &= RSS(\hat{\beta}_{Correlated}) \end{aligned}$$

which means the RSS for the correlated and the uncorrelated errors will have the same optimal $\hat{\beta}$.

Question 3

(a)

The values for the fitted coefficients with ordinary least squares are: $[0.4994, 1.1985, -2.4959, 1.2082]$. In figure 3 the fit is plotted along with the data.

(b)

The coefficients fitted with the weighted least square algorithm are $[0.5003, 0.9669, -1.9655, 0.9910]$.

(c)

The coefficients that are calculated using the least squares method are closer to the real values than the ones calculated using the ordinary least squares. To really say something about the difference and how well they match the given real values of the function we need more information about the distribution of the parameters. In figure 3 the two fits are plotted together with the data to fit through.

(d)

To derive the distribution of $\hat{\beta}_{WLS}$ we start from the distribution for the errors, which is known:

$$\epsilon \sim N_n(0, W)$$

Where W is a diagonal matrix with at position W_{ii} the variance of ϵ_i . This is the same matrix W used in the formula to calculate the estimate for β with the weighted least squares method. This formula is:

$$\hat{\beta}_{WLS} = (X^T W^{-1} X)^{-1} X^T W^{-1} Y \quad (4)$$

here Y is the vector with samples used to calculate the least squares estimator. X is the matrix as defined on page 193 in the course notes, this matrix is deterministic. Y can be rewritten as follows:

$$Y = X\beta + \epsilon$$

substituting this into equation 4 yields:

$$\begin{aligned} \hat{\beta}_{WLS} &= (X^T W^{-1} X)^{-1} X^T W^{-1} (X\beta + \epsilon) \\ &= (X^T W^{-1} X)^{-1} (X^T W^{-1} X) \beta + (X^T W^{-1} X)^{-1} X^T W^{-1} \epsilon \\ &= \beta + X^{-1} W X^{-T} X^T W^{-1} \epsilon \\ &= \beta + X^{-1} \epsilon \end{aligned}$$

We see this is a linear transformation of a multivariate normal distribution with mean $\mathbf{0}$ and as variance-covariance matrix $\Sigma_\epsilon = W$. Again using the results from section 6.2, page 172 in the course notes we find that the distribution of $\hat{\beta}_{WLS}$ is a multivariate normal distribution with mean β and variance-covariance matrix $X^{-1} W X^{-T} = (X^T W^{-1} X)^{-1}$. We conclude that:

$$\hat{\beta}_{WLS} \sim N_p(\beta, (X^T W^{-1} X)^{-1})$$

Question 4

(a)

To generate a sample from a multi-normal distribution we work analogues to the derivation in section 6.2 in the course notes, but backwards. We are given the variance-covariance matrix Σ_X of a multivariate normal model. This matrix must be positive definite, so the square root of this matrix exists. Let A be a matrix such that $AA^T = \Sigma_X$. This matrix can be found by diagonalizing Σ_X : $\Sigma_X = U\Lambda U^T$ and then taking $A = U\sqrt{\Lambda}$. We can do this because all eigenvalues of Σ_X are positive. The result from section 6.2 now tells us that $X = AY + \mu$ where Y is a vector of standard normal distributions. This observation can be used to draw a sample from X by drawing n independent samples from a standard normal distribution, filling them in in a vector y and the calculating the sample from X as $x = Ay + \mu$. This was implemented in R

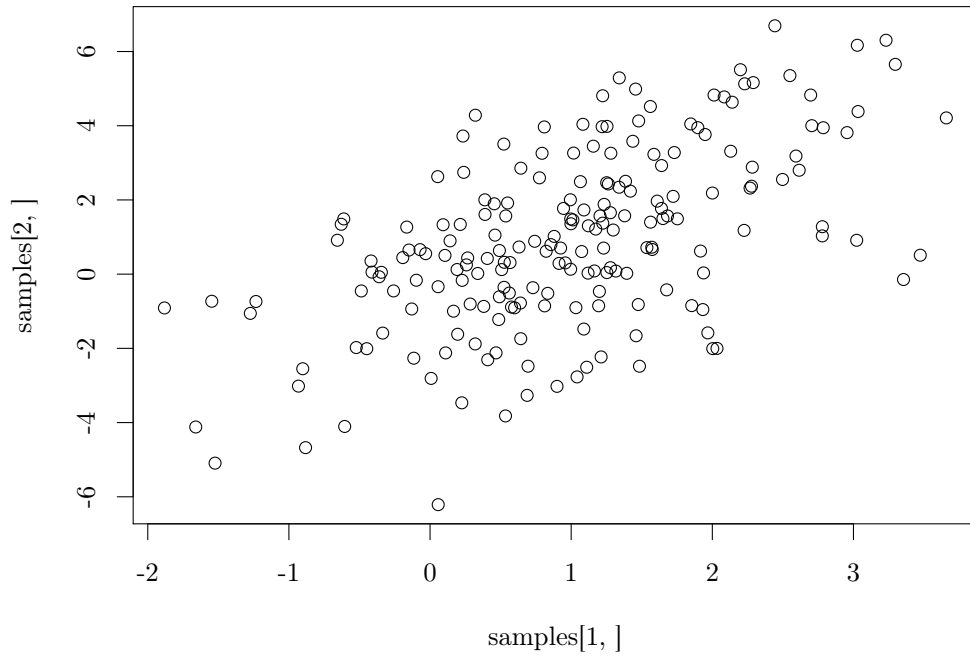


Figure 4: Projection on the xy -plane of the sample from the multivariate normal distribution

Acknowledgements

Rune Buckinx and Michaël Maex for helping to fix stupid mistakes in stupid R code
Seppe for stupid discussions about how well the questions are asked
Thibaut for organising a group crying session
Robbe for genuinely good input about trying to solve stupid questions