

Useful R-commands: part 1

Use `help(command)` for more information.

- Estimate a linear regression model of variable y on variable x from the data frame `mydata`:
`lm(y ~ x, data = mydata)`
The above fits a model with intercept, that is $E(y) = \beta_0 + \beta_1 \cdot x$.
If you wish to fit a model without intercept (i.e. you wish to impose $\beta_0 = 0$): then write
`lm(y ~ x - 1, data = mydata)`
- By saving the result of `lm`, we can access further information:
`model <- lm(y ~ x, data = mydata)`
`summary(model)`
- Plot the data and add the fitted line:
`plot(mydata$y ~ mydata$x)`
`abline(model)`
- The design matrix \mathbf{X} is returned when using the option `x = TRUE`
`model <- lm(y ~ x, data = mydata, x = TRUE)`
`model$x`
- The residuals can be obtained and plotted by
`e <- model$residuals`
`hist(e)`
`qqnorm(e)`
`qqline(e)`
- the matrix $(\mathbf{X}^T \mathbf{X})^{-1}$ is returned when using
`summary(model)$cov.unscaled`
and the estimated residual standard deviation by
`summary(model)$sigma`
- The parameter estimates and their confidence intervals are obtained with
`model$coefficients`
`confint(model)`
- `predict(model)` produces predicted values, obtained by evaluating the regression function in the x -values in the data frame that was used to estimate the model, i.e. `mydata`.
- We can also evaluate the regression function in some other x -values specified in the data frame `newdata`. It has to include the x -variables used in the model, using the same names, e.g.
`newdata <- data.frame(x = c(30))`
`predict(model, newdata)`
- `predict(model, newdata, interval = "interval")` uses the optional parameter `interval` which can be of type `interval = "confidence"`, or `interval = "prediction"` to compute 95 % confidence intervals and 95 % prediction intervals respectively.
- `predict(model, newdata, interval = "interval", level = level)` uses the optional parameter `level` to specify the probability level (e.g. `level = 0.90`).
- Percentiles from Student's t distribution can be obtained using the `qt` function, type `help(qt)` to learn more. For example the percentile leaving a probability area p at its left hand side using a Student's t distribution with df degrees of freedom is given by `qt(p, df)`.