Bayesian linear regression

Simon Grund

r Sys.Date()

# Bayesian Linear Model

### A statistical toolset for performing bayesian linear analysis on data

The purpose of this package is to provide an extensive toolset of functions for diverse functionality when analysing and intepretting data in a Bayesian statistical context.

A short outline of the functions contained in the packages:

* blm <- function(model, alpha, beta, data,...) Takes a prior distribution of model parameters and update it to a posterior distribution, based on observed data. This results in a fitted model.
* residuals.blm = function(object,...) Calulates the difference between predicted values and observed values for the response variable.
* deviance.blm = function(object,...) Calculates the sum of squared distances from the predicted response variables to the observed
* print.blm = function(x, ...) Shortly describes objects of class .blm
* summary.blm = function(object, ...) More extensive describtion of objects of class .blm
* plot.blm = function(object,...) plots a fitted model, based on observed and fitted variables.
* fitted.blm = function(object, ...) Calculates the fitted response variables.
* predict.blm = function(object, newData,...) predicted values of the response variable based on a fitted model and a new set of data
* coef.blm = function(object, ...) Extracts the coefficients from a fitted model
* confint.blm = function(object, parm=0, level=0.95,...) confidence interval for weights

|  |
| --- |
| ##Example of use Given a small dataset, D, beta = 1 and alpha = diag(1, nrow = 2, ncol = 2): |
| beta = 1.3 alpha = diag(1/a, nrow = 2, ncol = 2) w0 <- 0.2 w1 <- 3 |
| formula: y~x |
| x <- rnorm(10) y <- rnorm(10, mean = w0 + w1 \* x, sd = sqrt(1/beta)) D = data.frame(y=y, x=x |
| The fitted model is found by calling blm(y~x,alpha = alpha, beta = beta, data = D) |
| print(): |
| ## Bayesian Linear Model: ## y ~ x ## ## Call: ## blm(y ~ x, alpha = covariance, beta = b, data = D) ## ## Fitted coefficients (mean): ## NULL ## ## Covariance: ## (Intercept) x ## (Intercept) 0.008085149 0.001969170 ## x 0.001969170 0.008587165 |
| Plot(): |
| A more extensive view of the fitted model is provided calling summary. The summary demonstrates the coefficients resulting from calling coef(object,..) and the sum of squared distance from observed to estimated response variable y: |
| Summary(): |
| ## -------Summary Of Bayesian Linear Model------- ## ## Call: ## blm(y ~ x, alpha = covariance, beta = b, data = D) ## ## ----------------------------------------------------------------------- ## Sum of squared distance: ## 0 ## ----------------------------------------------------------------------- ## ## Coefficients (mean): ## estimate ## (Intercept) 0.1726752 ## x 2.8350273 ## ## ## Covariance Matrix, Sigma: ## (Intercept) x ## (Intercept) 0.008085149 0.001969170 ## x 0.001969170 0.008587165 ## ----------------------------------------------------------------------- ## Data[1:10,]: ## y x ## 1 -3.9573508 -1.29808300 ## 2 -3.6954332 -1.19642669 ## 3 -8.4066472 -2.44458813 ## 4 0.2439068 0.12609103 ## 5 0.1627386 -0.12723854 ## 6 -0.8281807 -0.33781365 ## 7 -0.6310241 -0.08535612 ## 8 -4.5627676 -1.46365070 ## 9 -3.6105888 -1.77278549 ## 10 0.3982502 -0.11226192 |
| The model can be used to predict target variables from new datasets, using the original posterior as a prior. A new dataset is introduced: |
| x = runif(100,-10,10) y = rnorm(n = 200, m = 0, 5) newD = data.frame(y=y, x=x) |
| ## Predicted (mean) value Sigma ## 1 0.138329790 0.7736340 ## 2 0.061455951 0.7735050 ## 3 0.151561862 0.7740520 ## 4 0.224373571 0.7784328 ## 5 0.047580077 0.7738998 ## 6 0.159507403 0.7743590 ## 7 0.221053905 0.7781564 ## 8 0.172555670 0.7749539 ## 9 0.068643479 0.7733507 ## 10 0.007171886 0.7757779 ## 11 0.086553000 0.7731156 ## 12 0.037551977 0.7742647 ## 13 -0.007849616 0.7767525 ## 14 0.130541813 0.7734422 ## 15 0.154543885 0.7741623 ## 16 0.231348303 0.7790372 ## 17 -0.019103430 0.7775808 ## 18 0.036633988 0.7743015 ## 19 0.220497558 0.7781108 ## 20 0.127734281 0.7733830 ## 21 -0.016604362 0.7773896 ## 22 0.220452865 0.7781072 ## 23 0.199325145 0.7765285 ## 24 0.253465197 0.7811674 ## 25 0.024203749 0.7748541 ## 26 0.033322767 0.7744386 ## 27 0.208609973 0.7771858 ## 28 0.291706365 0.7856170 ## 29 -0.014859656 0.7772586 ## 30 -0.023782106 0.7779500 ## 31 0.249246213 0.7807360 ## 32 -0.073486105 0.7827689 ## 33 -0.016574832 0.7773874 ## 34 0.009743792 0.7756260 ## 35 0.163091372 0.7745111 ## 36 0.245147563 0.7803282 ## 37 0.264707325 0.7823747 ## 38 0.030356204 0.7745677 ## 39 -0.075835458 0.7830373 ## 40 -0.047633197 0.7800577 ## 41 0.171630669 0.7749080 ## 42 0.162571014 0.7744885 ## 43 0.047841728 0.7738912 ## 44 0.005785426 0.7758615 ## 45 0.204215091 0.7768675 ## 46 -0.056741993 0.7809623 ## 47 0.026866898 0.7747270 ## 48 -0.089742625 0.7847011 ## 49 0.142935137 0.7737663 ## 50 0.045775881 0.7739605 ## 51 0.191855220 0.7760412 ## 52 0.021974149 0.7749640 ## 53 -0.091412074 0.7849094 ## 54 0.216826496 0.7778152 ## 55 0.157642939 0.7742832 ## 56 0.231417718 0.7790433 ## 57 0.085217661 0.7731258 ## 58 0.136835453 0.7735941 ## 59 0.092810460 0.7730837 ## 60 0.117663226 0.7732136 ## 61 0.260993849 0.7819666 ## 62 -0.007848036 0.7767524 ## 63 -0.081922685 0.7837497 ## 64 0.239876904 0.7798201 ## 65 0.098192905 0.7730770 ## 66 -0.087126665 0.7843783 ## 67 0.275109465 0.7835666 ## 68 -0.053003008 0.7805843 ## 69 0.239916656 0.7798239 ## 70 0.078536807 0.7731945 ## 71 -0.046915103 0.7799887 ## 72 0.258758551 0.7817254 ## 73 0.072082414 0.7732891 ## 74 0.164861067 0.7745894 ## 75 0.033994913 0.7744102 ## 76 -0.045351137 0.7798397 ## 77 0.213677689 0.7775687 ## 78 0.026982029 0.7747216 ## 79 0.061711699 0.7734989 ## 80 0.267851085 0.7827273 ## 81 0.053523049 0.7737150 ## 82 0.039320495 0.7741955 ## 83 0.280735356 0.7842411 ## 84 0.119181370 0.7732348 ## 85 0.146787038 0.7738878 ## 86 0.038415129 0.7742307 ## 87 0.013479357 0.7754133 ## 88 0.273426456 0.7833689 ## 89 -0.061052375 0.7814096 ## 90 0.012058523 0.7754931 ## 91 0.087900369 0.7731066 ## 92 -0.003866810 0.7764795 ## 93 0.072411478 0.7732836 ## 94 0.150853429 0.7740267 ## 95 0.233761578 0.7792538 ## 96 0.134717306 0.7735400 ## 97 0.043605258 0.7740364 ## 98 0.137636748 0.7736153 ## 99 0.069704360 0.7733309 ## 100 -0.073665931 0.7827893 ## 101 0.138329790 0.7736340 ## 102 0.061455951 0.7735050 ## 103 0.151561862 0.7740520 ## 104 0.224373571 0.7784328 ## 105 0.047580077 0.7738998 ## 106 0.159507403 0.7743590 ## 107 0.221053905 0.7781564 ## 108 0.172555670 0.7749539 ## 109 0.068643479 0.7733507 ## 110 0.007171886 0.7757779 ## 111 0.086553000 0.7731156 ## 112 0.037551977 0.7742647 ## 113 -0.007849616 0.7767525 ## 114 0.130541813 0.7734422 ## 115 0.154543885 0.7741623 ## 116 0.231348303 0.7790372 ## 117 -0.019103430 0.7775808 ## 118 0.036633988 0.7743015 ## 119 0.220497558 0.7781108 ## 120 0.127734281 0.7733830 ## 121 -0.016604362 0.7773896 ## 122 0.220452865 0.7781072 ## 123 0.199325145 0.7765285 ## 124 0.253465197 0.7811674 ## 125 0.024203749 0.7748541 ## 126 0.033322767 0.7744386 ## 127 0.208609973 0.7771858 ## 128 0.291706365 0.7856170 ## 129 -0.014859656 0.7772586 ## 130 -0.023782106 0.7779500 ## 131 0.249246213 0.7807360 ## 132 -0.073486105 0.7827689 ## 133 -0.016574832 0.7773874 ## 134 0.009743792 0.7756260 ## 135 0.163091372 0.7745111 ## 136 0.245147563 0.7803282 ## 137 0.264707325 0.7823747 ## 138 0.030356204 0.7745677 ## 139 -0.075835458 0.7830373 ## 140 -0.047633197 0.7800577 ## 141 0.171630669 0.7749080 ## 142 0.162571014 0.7744885 ## 143 0.047841728 0.7738912 ## 144 0.005785426 0.7758615 ## 145 0.204215091 0.7768675 ## 146 -0.056741993 0.7809623 ## 147 0.026866898 0.7747270 ## 148 -0.089742625 0.7847011 ## 149 0.142935137 0.7737663 ## 150 0.045775881 0.7739605 ## 151 0.191855220 0.7760412 ## 152 0.021974149 0.7749640 ## 153 -0.091412074 0.7849094 ## 154 0.216826496 0.7778152 ## 155 0.157642939 0.7742832 ## 156 0.231417718 0.7790433 ## 157 0.085217661 0.7731258 ## 158 0.136835453 0.7735941 ## 159 0.092810460 0.7730837 ## 160 0.117663226 0.7732136 ## 161 0.260993849 0.7819666 ## 162 -0.007848036 0.7767524 ## 163 -0.081922685 0.7837497 ## 164 0.239876904 0.7798201 ## 165 0.098192905 0.7730770 ## 166 -0.087126665 0.7843783 ## 167 0.275109465 0.7835666 ## 168 -0.053003008 0.7805843 ## 169 0.239916656 0.7798239 ## 170 0.078536807 0.7731945 ## 171 -0.046915103 0.7799887 ## 172 0.258758551 0.7817254 ## 173 0.072082414 0.7732891 ## 174 0.164861067 0.7745894 ## 175 0.033994913 0.7744102 ## 176 -0.045351137 0.7798397 ## 177 0.213677689 0.7775687 ## 178 0.026982029 0.7747216 ## 179 0.061711699 0.7734989 ## 180 0.267851085 0.7827273 ## 181 0.053523049 0.7737150 ## 182 0.039320495 0.7741955 ## 183 0.280735356 0.7842411 ## 184 0.119181370 0.7732348 ## 185 0.146787038 0.7738878 ## 186 0.038415129 0.7742307 ## 187 0.013479357 0.7754133 ## 188 0.273426456 0.7833689 ## 189 -0.061052375 0.7814096 ## 190 0.012058523 0.7754931 ## 191 0.087900369 0.7731066 ## 192 -0.003866810 0.7764795 ## 193 0.072411478 0.7732836 ## 194 0.150853429 0.7740267 ## 195 0.233761578 0.7792538 ## 196 0.134717306 0.7735400 ## 197 0.043605258 0.7740364 ## 198 0.137636748 0.7736153 ## 199 0.069704360 0.7733309 ## 200 -0.073665931 0.7827893 |
| This allows for continuous optimization of the model as more data (under the same formula) is introduced. |

To install the package: install.packages("blm") library(blm)