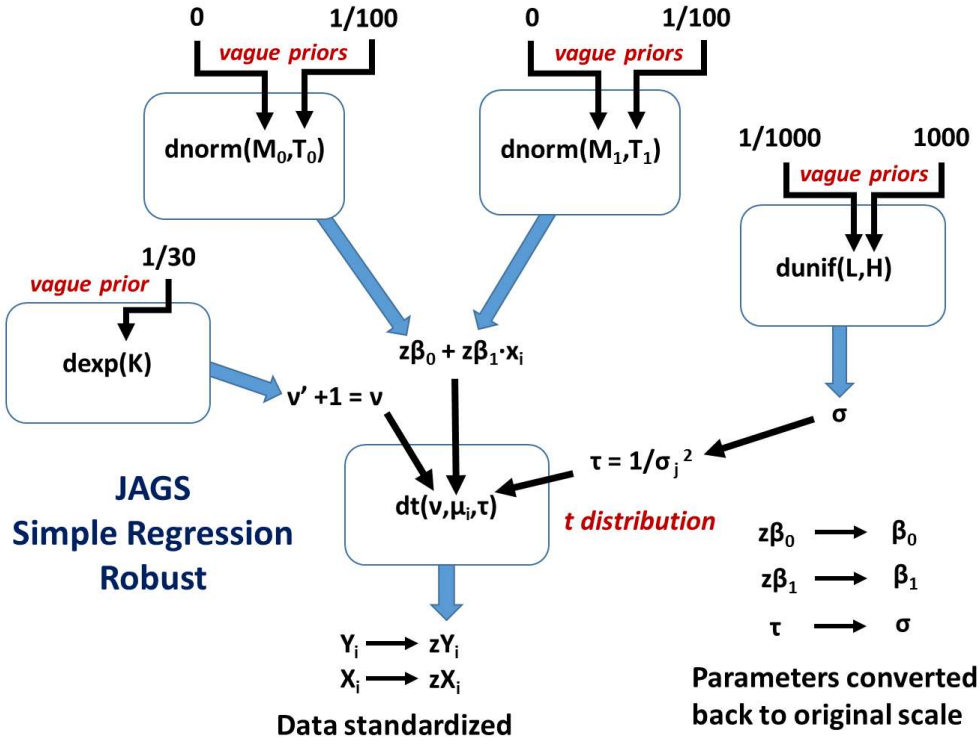
ORIGIN ≡ 0

Robust MCMC - Simple Regression

Simple Regression involves a model where the dependent variable y, with values indexed by *i*, are interpreted to be related to the independent variable x by means of the linear function $y_i = \beta_0 + \beta_1 x_i + \varepsilon$. The error term ε is commonly assumed belong to a Normal distribution. However in MCMC Robust regression using JAGS, the three parameter t-distribution is commonly utilized instead. JAGS Scaffolds are derived from the the text by J.K. Kruschke (K): *Doing Bayesian Data Analysis - A Tutorial with R, JAGS, and Stan*, available at https://sites.google.com/site/doingbayesiandataanalysis/.

As with other MCMC Robust models, variability in y_i are modeled with the t-distribution with parameters (ν,μ,τ). The behavior of μ is directly determined by the linear function with parameters β_0 and β_1 Priors are set with Normal distributions and vague initial values. Because the data y_i are not typically centered around zero, the β's are strongly correlated. This may cause problems with JAGS Gibbs samplers, so the usual procedure is to standardize y_i producing the new variable zy_i. After the MCMC analylsis is run, posterior parameters are then converted back to original scale. Priors for (ν,τ) are handled as before (see 030 MCMC). Because zy_i is standardized, values for all vague priors are set at standardized scales.



In K's 2nd level file containing function genMCMC() (in part shown below), note that the process of standardizing the data is handled within a data{} code segment, and conversion of standardized parameters back to scale of the original data is handled within the model{} segment.

from: `Jags-Ymet-Xmet-Mrobust.R`

```
# THE DATA.
 y = data[,yName]
 x = data[,xName]
  # Specify the data in a list, for later shipment to JAGS:
 dataList = list(
  x = x ,
  y = y
 )
```

```
 # THE MODEL.
modelString = "
# Standardize the data:
data {
 Ntotal <- length(y)
 xm <- mean(x)
 ym <- mean(y)
 xsd <- sd(x)
 ysd <- sd(y)
 for ( i in 1:length(y) ) {
  zx[i] <- ( x[i] - xm ) / xsd
  zy[i] <- ( y[i] - ym ) / ysd
 }
}
```

```
# Specify the model for standardized data:
model {
 for ( i in 1:Ntotal ) {
  zy[i] ~ dt( zbeta0 + zbeta1 * zx[i] ,
1/zsigma^2 , nu )
 }
 # Priors vague on standardized scale:
 zbeta0 ~ dnorm( 0 , 1/(10)^2 )
 zbeta1 ~ dnorm( 0 , 1/(10)^2 )
 zsigma ~ dunif( 1.0E-3 , 1.0E+3 )
 nu ~ dexp(1/30.0)
 # Transform to original scale:
 beta1 <- zbeta1 * ysd / xsd
 beta0 <- zbeta0 * ysd  + ym - zbeta1 * xm *
ysd / xsd
 sigma <- zsigma * ysd
}
" # close quote for modelString
```

Standard Simple Linear Regression Results:

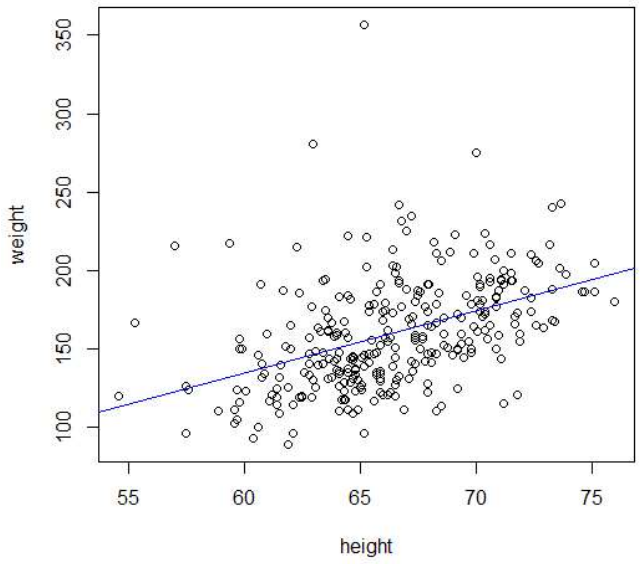
```
> summary(LM)
Call:
lm(formula = weight ~ height, data = myData)

Residuals:
    Min      1Q  Median      3Q     Max
-63.95 -21.17  -5.26  16.24 201.94

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -104.7832    31.5056  -3.326 0.000992 ***
height         3.9822     0.4737   8.406 1.77e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 31.59 on 298 degrees of freedom
Multiple R-squared:  0.1917,      Adjusted R-squared:  0.189
F-statistic: 70.66 on 1 and 298 DF,  p-value: 1.769e-15
```

```
#Run standard linear regression:
LM=lm(weight~height,data=myData)
summary(LM)
confint(LM, level=0.95)
Yhat=fitted(LM)
e=residuals(LM)
X=myData$height
Y=myData$weight
RESULTS=data.frame(X,Y,Yhat,e)
#PLOTTING REGRESSION LINE & POINTS:
X=myData$height
Y=myData$weight
sd(Y)
plot(X,Y,xlab='height',ylab='weight')
abline(LM,col="blue")
```

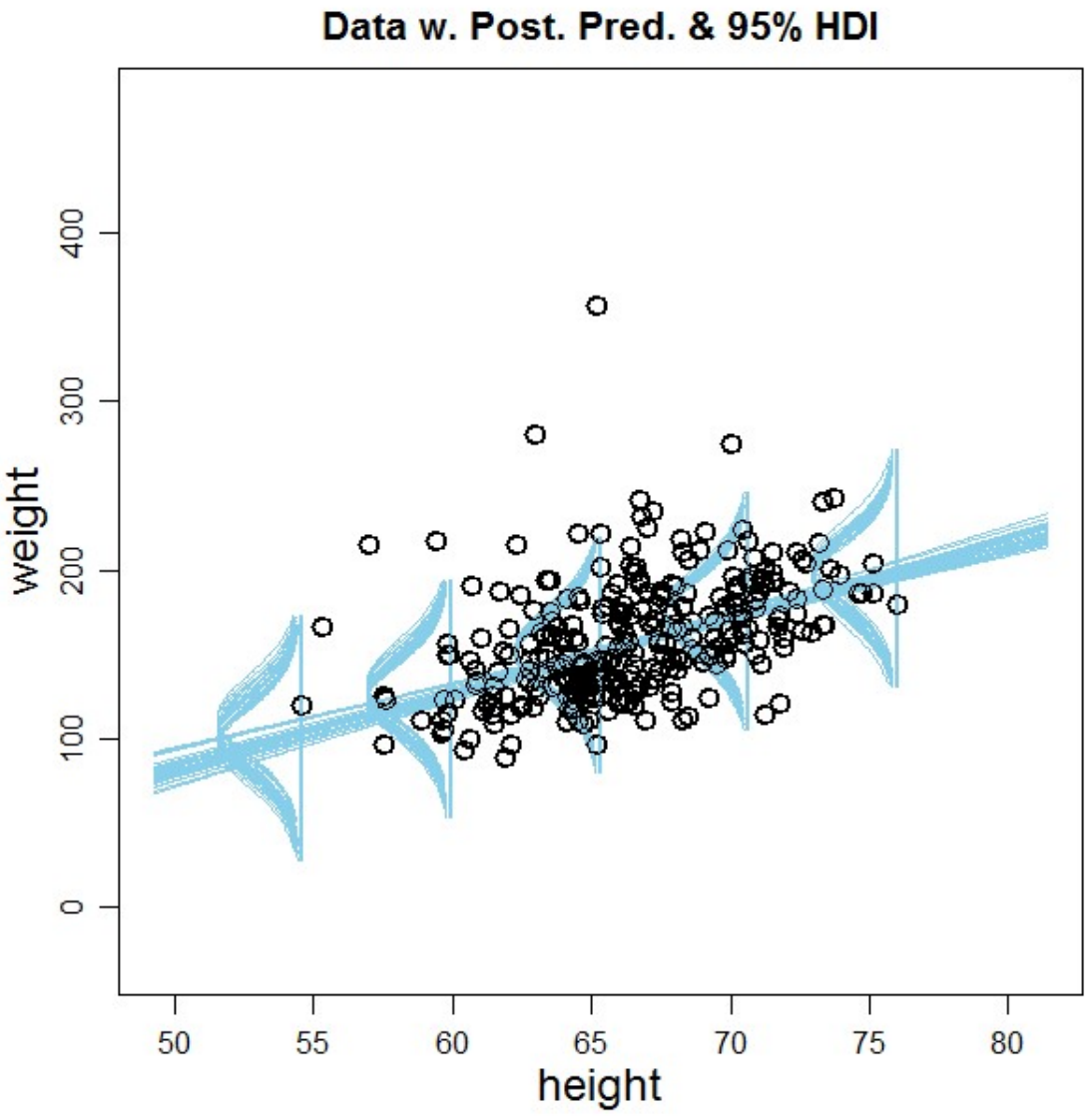



```
> confint(LM,level=0.95)
                  2.5 %     97.5 %
(Intercept) -166.784813 -42.781609
height         3.049948   4.914503

> sd(Y)
[1] 35.07212
```

MCMC Results:

```
> show(summaryInfo)
                  Mean        Median          Mode     ESS HDImass
beta0     -139.775801 -139.9595264 -138.8772426 16358.9    0.95
beta1        4.458403    4.4597229    4.4412420 16651.3    0.95
sigma       24.009506   23.9536256   23.8904383  6294.0    0.95
nu           5.411979    5.1014283    4.6649534  4317.9    0.95
log10(nu)    0.715502    0.7076918    0.7041195  4999.0    0.95
                HDIlow     HDIhigh CompVal PcntGtCompVal ROPElow ROPEhigh
beta0     -193.4713221 -85.7968016      NA            NA      NA       NA
beta1        3.6295620   5.2409550       0           100    -0.5      0.5
sigma       20.8806035  27.2448702      NA            NA      NA       NA
nu           2.9115509   8.6545470      NA            NA      NA       NA
log10(nu)    0.4904864   0.9610638      NA            NA      NA       NA
          PcntLtROPE PcntInROPE PcntGtROPE
beta0             NA         NA         NA
beta1              0          0        100
sigma             NA         NA         NA
nu                NA         NA         NA
log10(nu)         NA         NA         NA
```



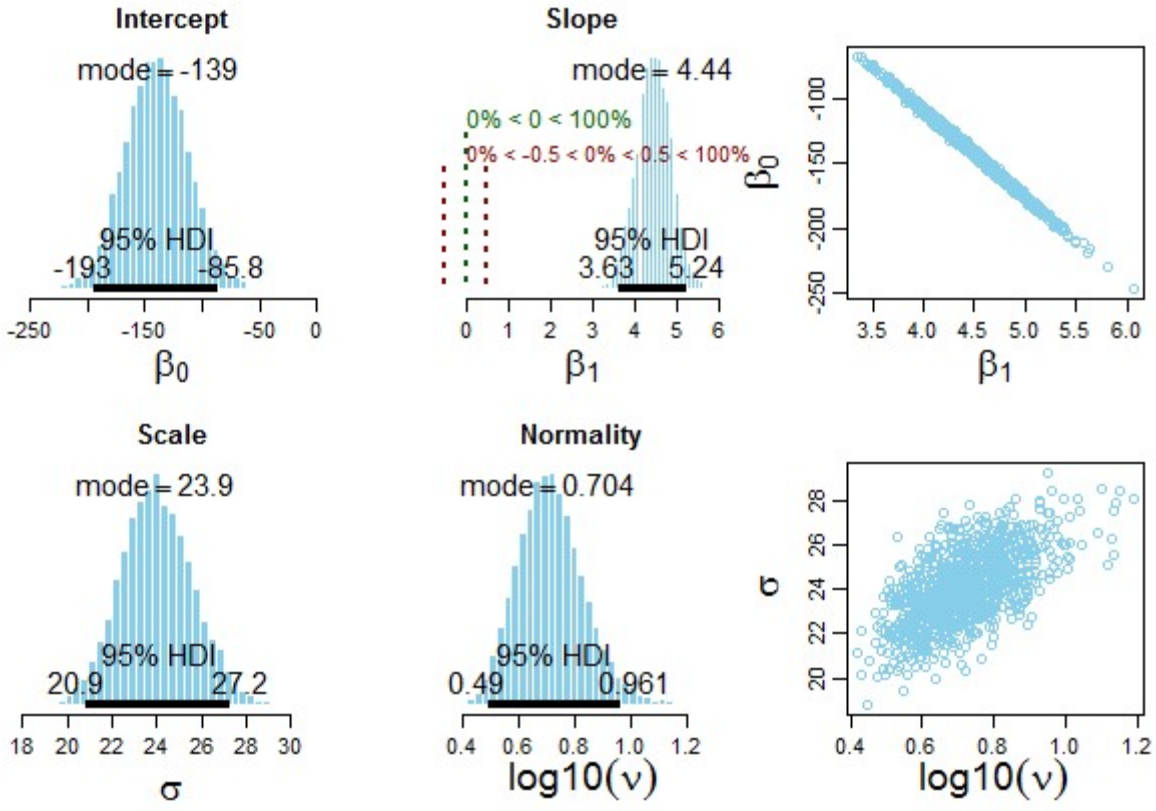
Summary of Findings:

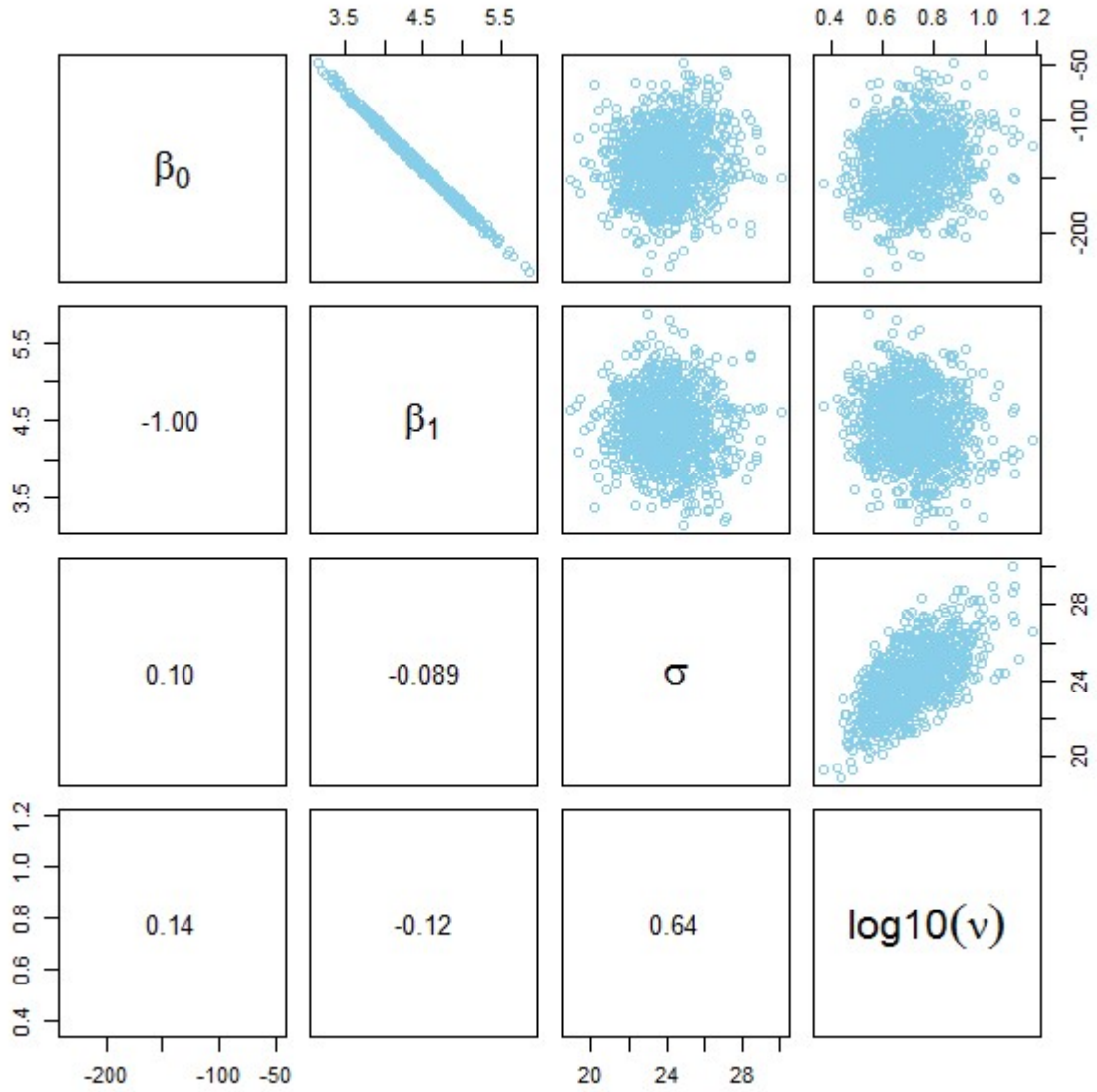
Parameter:	Simple Regression: point estimate	95% CI	MCMC using JAGS: mode	95% HDI
β_0	-104.78888832	[-166.78 - -42.78]	-139	[-193 - -85.8]
β_1	3.9822	[3.05 - 4.91]	4.44	[3.63 - 5.24]
σ	35.07212		23.9	[20.9 - 27.2]

As before, values of the Normality parameter $\nu > 30$ ($\log_{10}(\nu) > 1.4771$ indicates Normal distribution behavior. Therefore, observed posterior distribution of ν (mode = 0.704) indicates a role to be played by the t-distribution in accounting for outlier data points. As a result, MCMC derived σ (mode = 23.9) is less than the point estimate determined by the standard deviation of the sample. The JAGS Bayesian modes for regression parameters are more-or-less simiar to the point standard (frequentist) estimates, although slope is more nearly similar than intercept. This makes sense because the center of the data distribution is far from height = 0, so intercept beyond range of the data might be expected to be highly variable.

The Post Prediction Plot, above, shows the original data points (N=300) along with a sample of regression prediction lines representing frequently encountered joint β_0 and β_1. The humped distributions at values of x_i, are predictions of weight (y_i) given values of height (x_i) derived from sampling β_0 and β_1 along the MCMC chains.

In the pairwise plots, strong correlation between regression parameters β_0 and β_1 is seen, and is expected from a linear model. Correlations between σ and ν is seen and also expected for t-distributions models.




β0
β1
σ
log10(ν)
-1.00
0.10
-0.089
0.14
-0.12
0.64
3.5
4.5
5.5
0.4
0.6
0.8
1.0
1.2
-50
-100
-200
20
24
28