

Bayesian Linear Model

```
##Read in the data
df <- read.csv("gdpgrowth.csv")
library(mvtnorm)
y <- df$GR6096
n <- length(y)
X <- matrix(NA,ncol=2,nrow=n)
X[,1] <- rep(1,n)
X[,2] <-df$DEF60
```

D) Gaussian Model

```
Lambda <- diag(1,n)
K <- diag(0.001,2)

##Prior Param
d <- 1
eta <- var(y)
m <- matrix(c(0,0),nrow=2)

##Posterior Param
d_n <- n + d
K_n <- t(X)%*%Lambda%*%X +K
Sigma_n <- solve(K_n)
m_n <- Sigma_n%*%(t(X)%*%Lambda%*%y + K%*%m)
eta_n <- t(y)%*%Lambda%*%y+ t(m)%*%K%*%m - t(m_n)%*%K_n%*%m_n + eta
```

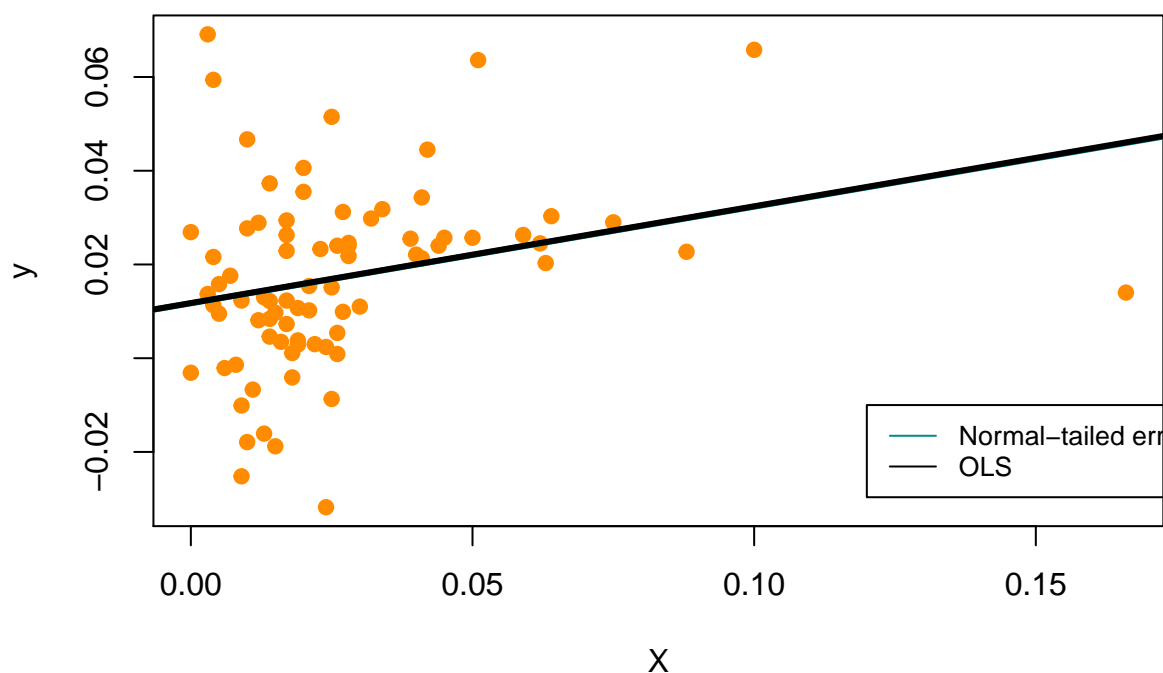
We initilized prior parameters and update posterior parameter.

```
##Sampling steps
S <- 10000
omega <-rgamma(S,d_n/2,eta_n/2)
Beta_1 <- matrix(NA,ncol=2,nrow=S)
for(i in 1:S){
  Beta_1[i,] <- rmvnorm(1,m_n,sigma= Sigma_n/omega[i])
}
```

Since we know the closed form distribution of the β and ω , we can draw them independently from their distributions. Here, β_1 is a matrix of 2 columns where each column represents β_0 and β_1 respectively and each row represents a draw from the posterior distributions.

```
final_idx <- seq(from=0.1*S,to=S,by=5)
plot(X[,2],y=y,col="darkorange",pch=19,main="Regression plot",xlab="X")
Beta_lm <- coef(lm(GR6096~DEF60,data=df))
# plot(par_mat[final_idx,1])
abline(a=mean(Beta_1[final_idx,1]),b=mean(Beta_1[final_idx,2]),col="cyan4",lwd=3)
abline(a=Beta_lm[1],b=Beta_lm[2],lwd=3)
legend(0.12,-0.01, legend=c("Normal-tailed error","OLS"),
      col=c("cyan4","black"), lty=1, cex=0.8)
```

Regression plot



C) Heavy tailed error model

```
##Initial Values
B <- 10000
cur_beta <- c(0,0)
cur_omega <- 1
cur_lambda <- rep(1,n)
num_par <- length(cur_beta)+length(cur_lambda) +1
par_mat <- matrix(NA,nrow=B,ncol=num_par)
```

Again, we initailzed objects to store our parameters.

```
##Prior Param
Lambda <- diag(cur_lambda,n)
K <- diag(0.001,2)
h <- 0.01
d <- 1
eta <- var(y)
m <- matrix(c(0,0),nrow=2)

##Posterior Param
d_n <- n + d
K_n <- t(X)%*%Lambda%*%X +K
eta_n <- t(y)%*%Lambda%*%y+ t(m)%*%K%*%m - t(m_n)%*%K_n%*%m_n + eta
m_n <- Sigma_n%*%(t(X)%*%Lambda%*%y + K%*%m)
Sigma_n <- solve(K_n)
```

Here, we initialized the prior and update posterior parameters.

Gibb Sampling Step

```
for(i in 1:B){
  ##Update all the parameters for full conditional of beta
  Lambda <- diag(cur_lambda,n)
  K_n <- t(X)%*%Lambda%*%X +K
  Sigma_n <- solve(K_n)
  m_n <- Sigma_n%*%(t(X)%*%Lambda%*%y + K%*%m)

  cur_beta <-rmvnorm(1,mean=m_n,sigma= Sigma_n/cur_omega)

  ##Update all the parameters for full conditional of omega

  eta_n <- t(y)%*%Lambda%*%y+ t(m)%*%K%*%m - t(m_n)%*%K_n%*%m_n + eta
  cur_omega <- rgamma(1,d_n/2,eta_n/2)

  ##Update all the parameters for full conditional of lambda
  mu <- X%*%t(cur_beta)
  cur_lambda <- rgamma(n,rep((h+1)/2,n),(cur_omega*(y-mu)^2+h)/2)

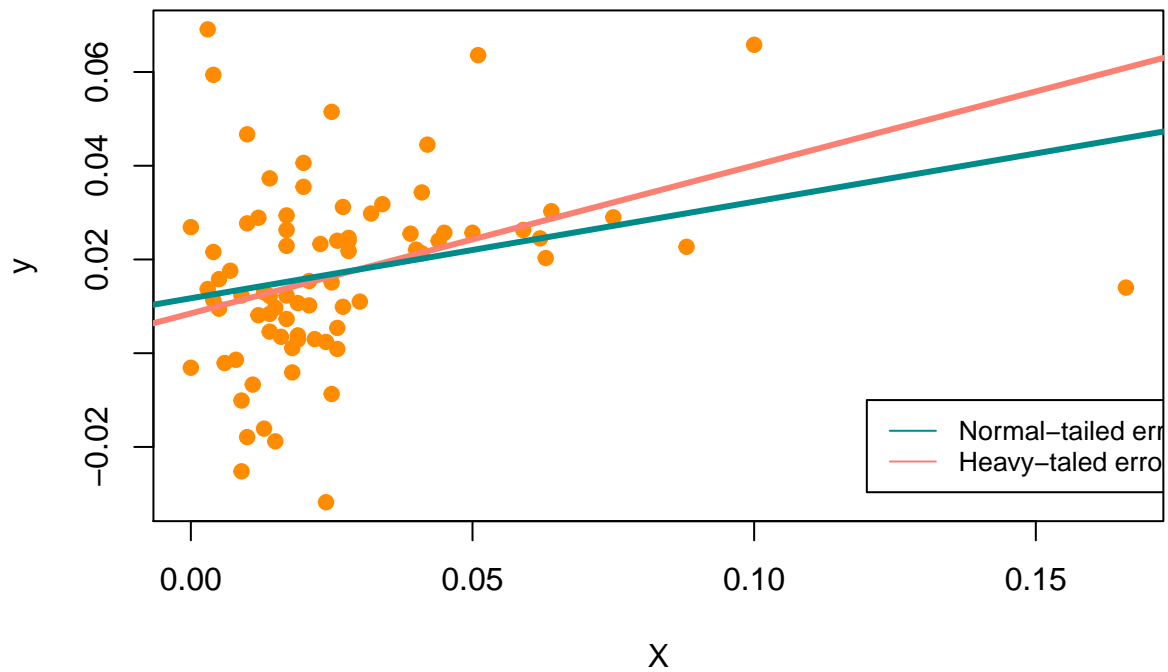
  par_mat[i,] <- c(cur_beta,cur_omega,cur_lambda)
}

Beta_2 <- par_mat[,1:2]

final_idx <- seq(from=0.1*B,to=B,by=5)
plot(X[,2],y=y,col="darkorange",pch=19,main="Regression plot",xlab="X")

# plot(par_mat[final_idx,1])
abline(a=mean(Beta_2[final_idx,1]),b=mean(Beta_2[final_idx,2]),col="salmon",lwd=3)
abline(a=mean(Beta_1[final_idx,1]),b=mean(Beta_1[final_idx,2]),col="cyan4",lwd=3)
legend(0.12,-0.01, legend=c("Normal-tailed error","Heavy-tailed error"),
      col=c("cyan4","salmon"), lty=1, cex=0.8)
```

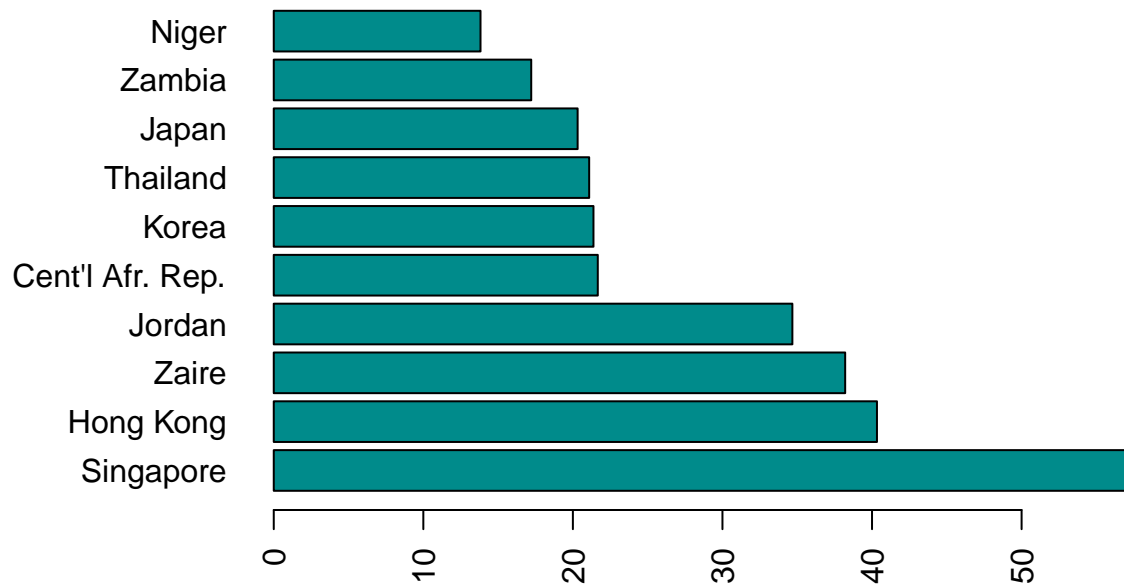
Regression plot



```
pos_mean_param <- apply(par_mat,2,mean)
pos_var <- (1/pos_mean_param)[4:num_par]
var_idx <-order(pos_var,decreasing=T)
```

```
par(las=2) # make label text perpendicular to axis
par(mar=c(5,8,4,2)) # increase y-axis margin.
top_10_idx <- var_idx[1:10]
barplot(pos_var[top_10_idx], main="Top 10 Posterior Variance", horiz=TRUE,
        names.arg=factor(df[top_10_idx,"COUNTRY"]), col="cyan4")
```

Top 10 Posterior Variance



```
plot(X[,2],y=y,col="cyan4",pch=19,main="Top 10 Potential Outliers Based on Variance",xlab="X")
points(x=X[top_10_idx,2],y=y[top_10_idx],cex=2.5, col=heat.colors(10),pch=16)
text(X[top_10_idx,2],y[top_10_idx],labels= df[top_10_idx,"COUNTRY"],cex = 0.7)
```

Top 10 Potential Outliers Based on Variance

