

# Macroeconometrics - Assignment II

Englund, Kostecka, Kreuzbichler, Pfeiffenberger

July 27, 2022

## 1 Part I: Writing complex functions in R

### 1.1 Task 1: Functions to estimate regression models using the SSVS prior

Read the following introduction to functions in R (Advanced R by Hadley Wickham). In this task, you have to write a function that performs Bayesian inference in a regression model with a stochastic search variable selection prior based on the code discussed in class (see code SSVS.R). Use the economic growth dataset of Fernandez, Ley, and Steel (2001, J. Applied Econometrics) provided in the BMS package in R. To get this data, type `data(datafls)` after loading the BMS package.

- Write a function that takes the explanatory variables  $\mathbf{X}$  as well as the endogenous variable  $\mathbf{y}$  as input. In the growth dataset, the first column contains the endogenous variable whereas the remaining columns are the explanatory variables. Think about what additional inputs might be helpful! (Hint: you might want to vary *nsave* and *nburn*.) Also think carefully about the potential output of the function! (Hint: R functions can only return a single object, so use a list object.)
- Run the function using different values for *tau0* and *tau1*. What happens to the posterior inclusion probabilities (PIP.mean) if *tau0* is set equal to  $1e-15$ ? Describe this finding verbally and graphically!
- The variables in  $\mathbf{X}$  all feature a different scale. This causes problems since the simple implementation of the code sets *tau0* and *tau1* equal to fixed values that are independent of the scaling of the data. Try to standardize the data such that all columns of  $\mathbf{X}$  (and  $\mathbf{y}$ ) have mean zero and variance one.

#### 1.1.1 Writing a function

We create a function called *bayes* whose formals, i.e. arguments that the function takes as input, are variables *X*, *Y*, *nsave*, *nburn*, *tau0* and *tau1*. The variables *nsave*, *nburn*, *tau0* and *tau1* have a preset default value that can be overwritten, otherwise the function call can omit these variables and use the default values.

To import the growth dataset, we use the following commands:

```
> library(BMS)
> Y<-matrix(datafls$y)
```

```
> X<-as.matrix(datafls[2:42])
```

The function header (formals) looks like this:

```
> bayes <- function(X,Y,nsave=1000,nburn=1000, tau0=0.01, tau1=10) {
+
+ }
```

We then proceed with defining the dynamics of the function (body). The output is a matrix consisting of three columns; PIP.mean, A.mean, SIG.mean:

```
> bayes <- function(X,Y,nsave=1000,nburn=1000, tau0=0.01, tau1=10) {
+   #Start with actual estimation and Gibbs preliminaries
+   ntot <- nsave+nburn
+   #Prior prelims for SSVS
+   #tau0 <- 0.01
+   #tau1 <- 10
+
+   s0 <- 0.01
+   S0 <- 0.01
+   #Construct Y and X
+   N <- nrow(Y)
+   K <- ncol(X)
+
+   #Get OLS quantities
+   A.OLS <- solve(crossprod(X))%*%crossprod(X,Y)
+   SSE <- crossprod(Y-X%*%A.OLS)
+   SIG.OLS <- SSE/(N-K)
+   #In the next step, create storage matrices for Gibbs loop and initialize prior
+   gamma <- matrix(1,K,1) #indicators, start with full model
+   sigma2.draw <- as.numeric(SIG.OLS)
+   V.prior <- diag(as.numeric(gamma*tau1+(1-gamma)*tau0))
+
+   ALPHA.store <- matrix(NA,nsave,K)
+   SIGMA.store <- matrix(NA,nsave,1)
+   Gamma.store <- matrix(NA,nsave,K)
```

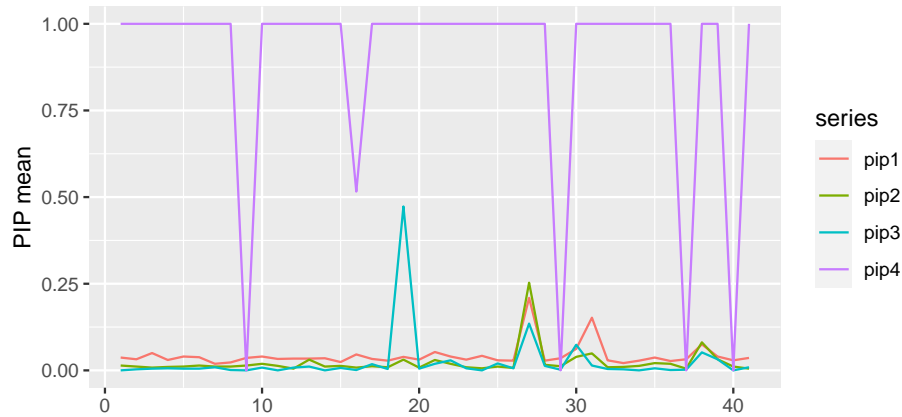
```

+
+   for (irep in 1:ntot){
+     #Draw ALPHA given rest from multivariate normal
+     V.post <- solve(crossprod(X)*1/sigma2.draw+diag(1/diag(V.prior)))
+     A.post <- V.post%*(crossprod(X,Y)*1/sigma2.draw)
+     A.draw <- A.post+t(chol(V.post))%*rnorm(K)
+
+     #Draw indicators conditional on ALPHA
+     for (jj in 1:K){
+       p0 <- dnorm(A.draw[[jj]],0,sqrt(tau0))
+       p1 <- dnorm(A.draw[[jj]],0,sqrt(tau1))
+       p11 <- p1/(p0+p1)
+
+       if (p11>runif(1)) gamma[[jj]] <- 1 else gamma[[jj]] <- 0
+     }
+     #Construct prior VC matrix conditional on gamma
+     V.prior <- diag(as.numeric(gamma*tau1+(1-gamma)*tau0))
+
+     #Simulate sigma2 from inverse Gamma
+     S.post <- crossprod(Y-X%*A.draw)/2+S0
+     s.post <- S0+N/2
+     sigma2.draw <- 1/rgamma(1,s.post,S.post)
+
+     if (irep>nburn){
+       ALPHA.store[irep-nburn,] <- A.draw
+       SIGMA.store[irep-nburn,] <- sigma2.draw
+       Gamma.store[irep-nburn,] <- gamma
+     }
+     #print(irep)
+   }
+
+   #Calculate posterior inclusion probabilities
+   PIP.mean <- apply(Gamma.store,2,mean)
+   A.mean <- apply(ALPHA.store,2,mean)
+   SIG.mean <- apply(SIGMA.store,2,mean)
+
+   return(cbind(PIP.mean,A.mean,SIG.mean))
+ }

```

### 1.1.2 Testing different values for tau0 and tau1

For smaller tau0, the posterior inclusion probabilities mean (PIP.mean) is gradually smaller. For tau0 set to 1e-15, the PIP.mean alternates between 0 and 1. The following chart confirms the above:



### 1.1.3 Standardize data

To standardize our input data, we use the `scale()` function. We can verify that the mean is 0 and standard deviation 1 as we would expect like this:

```
> X_norm <- scale(X)
> Y_norm <- scale(Y)
> mean(X_norm)
```

```
[1] 9.792022e-18
```

```
> mean(Y_norm)
```

```
[1] 7.437326e-17
```

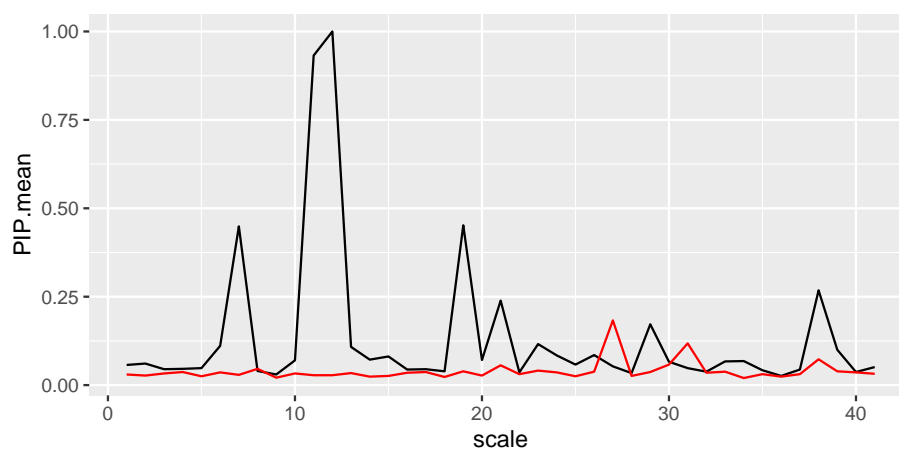
```
> sd(X_norm)
```

```
[1] 0.9931995
```

```
> sd(Y_norm)
```

```
[1] 1
```

Plotting the difference in PIP.mean for the standardized and non-standardized dataset:



## 2 Part II: Model uncertainty in economic growth regressions

### 2.1 Task 1: Bayesian Normal Linear Regression

Read the paper by Fernandez, Ley and Steel (2001, J. Applied Econometrics) as well as Chapter 11 in Gary Koop's textbook. Consider the data for the paper by Fernandez, Ley and Steel (2001, J. Applied Econometrics), available from the BMS package

- Reproduce the results in Table 11.1 (in Koop) using the BMS package in R.
- Use your custom function for the SSVS prior to reproduce Table 11.1.
- How do results differ? To what extent is this related to the specific choices of  $\tau_0$  and  $\tau_1$ ?

**2.1.1 Reproducing Data****2.1.2 Custom Function for the SSVS Prior****2.1.3 Analyzing Results**