STAT 52900 - Applied Decision Theory and Bayesian Statistics Project
Time series Forecasting for Agriculture Crop Production using Bayesian Analysis

By Priyanka Surapaneni
Guided By: Dr.Ben Boukai

Introduction

Problem Statement

How can we forecast agriculture crop production using historical data extracted from [Food and Agriculture Organization of United States]() using Bayesian Model

What's Interesting About Problem

The main factor is including priors in Bayesian model which is different compared to normal time series models or frequentist approaches

Motivation to Answer the Problem

As population increasing exponentially every year, forecast of crop production is valuable for economic planning and global food security.

Data Description

The Data contains world-wide agriculture data with a variety of fruits and vegetables cultivated at various locations.

- Area: Name of region.
- Item: Name of fruit or vegetable.
- Element: Area harvested, Yield and Production
- Units ha(hectares) for area, hg/ha(hectogramme(100 grammes)per hectare) for Yield, tonnes for crop production.
- Years are from 1961 to 2019.

Area	Item	Element	Unit	Y1961	Y1962	Y1963
United States of America	Almonds, with	Area harvested	ha	36138	37676	39578
United States of America	Almonds, with	Yield	hg/ha	16669	11558	13684
United States of America	Almonds, with	Production	tonnes	60237	43545	54159
United States of America	Apples	Area harvested	ha	184780	182390	182390
United States of America	Apples	Yield	hg/ha	139842	141510	143045
United States of America	Apples	Production	tonnes	2584000	2581000	2609000
United States of America	Apricots	Area harvested	ha	16070	15540	15580
United States of America	Apricots	Yield	hg/ha	107146	96730	115581
United States of America	Apricots	Production	tonnes	172183	150319	180075
United States of America	Artichokes	Area harvested	ha	3440	3237	3237
United States of America	Artichokes	Yield	hg/ha	67247	61656	67260
United States of America	Artichokes	Production	tonnes	23133	19958	21772
United States of America	Asparagus	Area harvested	ha	59751	59071	58747
United States of America	Asparagus	Yield	hg/ha	28027	28572	29000
United States of America	Asparagus	Production	tonnes	167465	168780	170368

Data Cleaning

Selected 2 regions, United States of America and Asia and selected apples from items for Bayesian Analysis

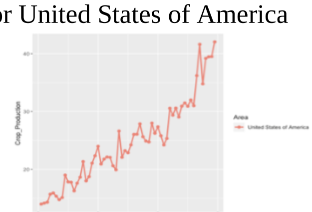
Area	Item	Element	Unit	Y1961	Y1962	Y1963
United States of America	Apples	Area harvested	ha	184780	182390	182390
United States of America	Apples	Yield	hg/ha	139842	141510	143045
United States of America	Apples	Production	tonnes	2584000	2581000	2609000

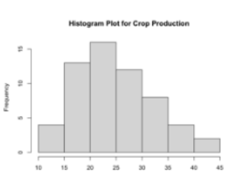
- Normalized the crop production by area harvested as the number are giant for computation
- Pivot the data to time series.

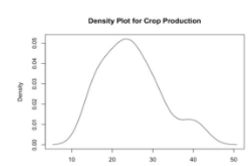
	Area	Year	Crop_Production
	<chr>	<dbl>	<dbl>
1	United States of America	1961	14.0
2	United States of America	1962	14.2
3	United States of America	1963	14.3
4	United States of America	1964	15.7
5	United States of America	1965	15.9
6	United States of America	1966	15.3
7	United States of America	1967	14.7
8	United States of America	1968	15.1
9	United States of America	1969	19.0
10	United States of America	1970	17.8

Visualization

For United States of America

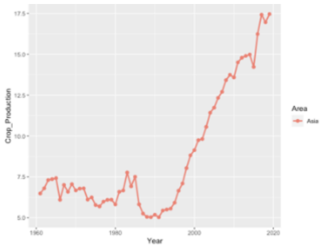


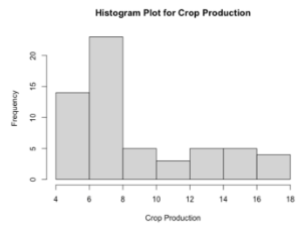


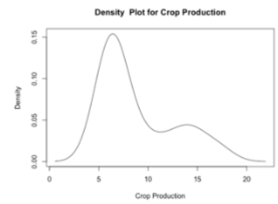


- The graphs shows the time series, histogram and density plots which resembles the normal distribution

For Asia







- The graphs shows the time series, histogram and density plots which also resembles the normal distribution for Asia

Model

- I have consider Bayesian Auto regression model with p value(lag) = 2.
- Model equation : $Y_t = \alpha + \beta_1 Y_{\{t-1\}} + \beta_2 Y_{\{t-2\}} + \epsilon_t$
  Where, α is the constant term , $\beta_1\beta_2$ are coefficients of lag versions($Y_{\{t-1\}}, Y_{\{t-2\}}$) of target variable(Y_t) and $\epsilon_t \sim N(0, \sigma^2)$ (normally distributed error).
- Consider Matrix format $B = [\alpha\ ,\ \beta_1\ ,\ \beta_2]$ and $X_t = [1\ ,\ Y_{\{t-1\}}\ ,\ Y_{\{t-2\}}]$
- After submitting in our model equation, results as $Y_t = BX_t + \epsilon_t$, which resemble the linear regression matrix format.
- Likelihood function is given by $L(Y_t|B,\sigma^2) = (2\pi\sigma^2)^{\{\frac{N}{2}\}} \exp\left(-\frac{\{(Y_t - BX_t)^T(Y_t - BX_t)\}}{2\sigma^2}\right)$
- In this case, the optimal parameters can be found by taking the **derivative** of the **log** of Likelihood function and finding the values of B and σ^2 where the derivative equals zero.
- OLS estimator : $\hat{B} = (X_t{}'X_t)^{-1}(X_t{}'Y_t)$
- The optimal value for variance : $\sigma^2 = \frac{\epsilon'\epsilon}{N}$, where N is total number of rows in data.

Joint Posterior and Priors

- Joint posterior : $P(B, \sigma^2|Y_t) \propto L(Y_t|B,\sigma^2)\ p(B,\sigma^2)$
- Here our goal is to approximate the posterior distribution of coefficients : $\alpha, \beta_1, \beta_2, \sigma^2$
- The posterior mean and variance of the normal distribution conditional on B and σ^2 is taken from Time Series Analysis of Hamilton(1994) and in Bishop Pattern Recognition and Machine Learning in chapter 3 with different notation.
- M = $\left(\Sigma_0^{-1} + \frac{1}{\sigma^2} X_t'X_t\right)^{-1} (\Sigma_0^{-1} B_0 + \frac{1}{\sigma^2} X_t'Y_t) = \left(\Sigma_0^{-1} + \frac{1}{\sigma^2} X_t'X_t\right)^{-1} (\Sigma_0^{-1} B_0 + \frac{1}{\sigma^2} X_t'X_t B_{0ls})$
- Mean is the weighted average of prior mean and maximum likelihood estimator of B
- V = $\left(\Sigma_0^{-1} + \frac{1}{\sigma^2} X_t'X_t\right)^{-1}$
- Normal priors for B matrix coefficients with mean = 0 and variance = 1
- $\begin{pmatrix} \alpha_0 \\ \beta_1^0 \\ \beta_2^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} \Sigma\_\alpha & 0 & 0 \\ 0 & \Sigma\_\beta_1 & 0 \\ 0 & 0 & \Sigma\_\beta_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- For variance : Chosen inverse gamma prior (conjugate prior).
- $p(\sigma^2) \sim \Gamma^{-1}(\frac{T_0}{2}, \frac{D_0}{2})$, where $T_0 = 1$ and $D_0 = 0.1$, which are arbitrarily chosen.

Analysis of model

- For calculating marginal distributions, which is analytically complex but numerically possible by Markov Chain Monte Carlo (MCMC) using Gibbs Sampler.
- Models were implemented for United States of America and Asia independently.

- For each model, 15000 iterations were simulated and first 4000 are considered as burn down.
- To get a random variable from a Normal distribution with mean M and variance V we can sample a vector from a standard normal distribution and transform it using the equation below.
- To get a random variable for B from a Normal distribution (conditional posterior) with mean M and variance V we can sample a vector from a standard normal distribution and transform it using the equation below.
- $B' = M^* + \left[\bar{B} * V^{*\{\frac{1}{2}\}}\right]^T$
- Companion matrix (transformed version lagged coefficient matrix) as shown in below figures, is computed to check the drawn coefficient matrix B is stable/stationary for AR model
- Check for stability is, if the absolute values of the eigenvalues are less than 1(only need to check the largest eigenvalue is < |1|) , then model is dynamically stable and B samples can be drawn.

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \cdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix}$$

matrix of coefficients

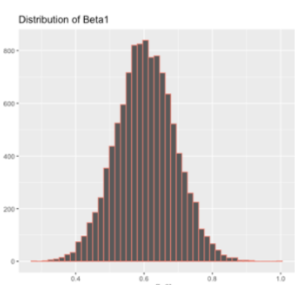
$$\begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_{n-1} & \beta_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

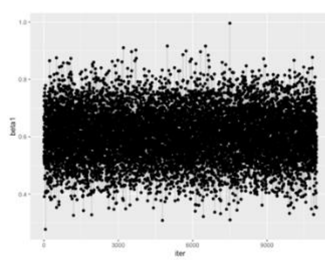
Companion form of matrix

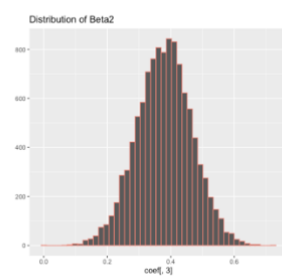
- To draw sigma from the Inverse Gamma distribution conditional on B. To sample a random variable from the inverse Gamma distribution with degrees of freedom T/2 and scale $D/2$. we can sample T variables from a standard normal distribution z0 ~ N(0,1) and then make the following adjustment
- $\sigma^2 = \frac{D}{z_0' z_0}$, σ^2 is now a draw from the correct Inverse Gamma distribution.
- Forecast equation for t+1 : $\hat{Y}_{\{t+1\}} = \alpha + \beta_1 \hat{Y}_t + \beta_2 \hat{Y}_{\{t-1\}} + \sigma_v^*$

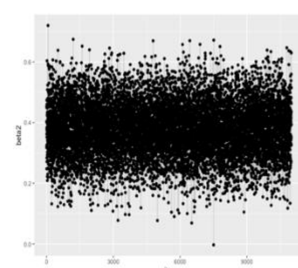
Results

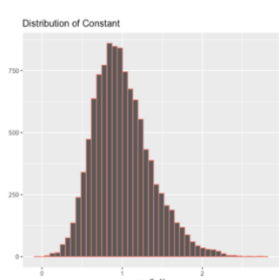
Histogram and trace plots of United States of America for $\alpha, \beta_1, \beta_2, \sigma^2$, which are significant.

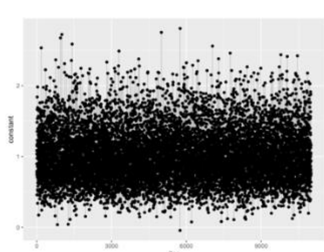


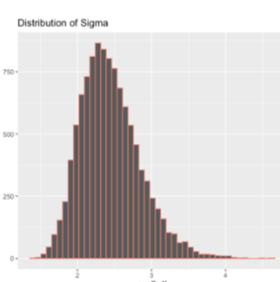


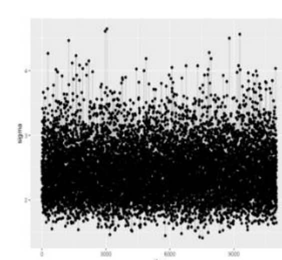




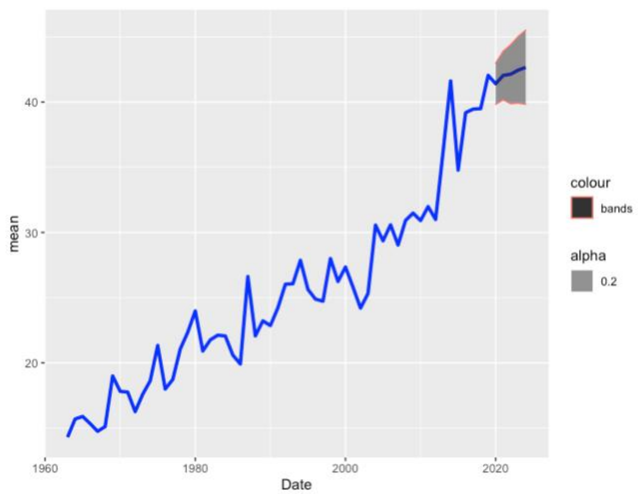






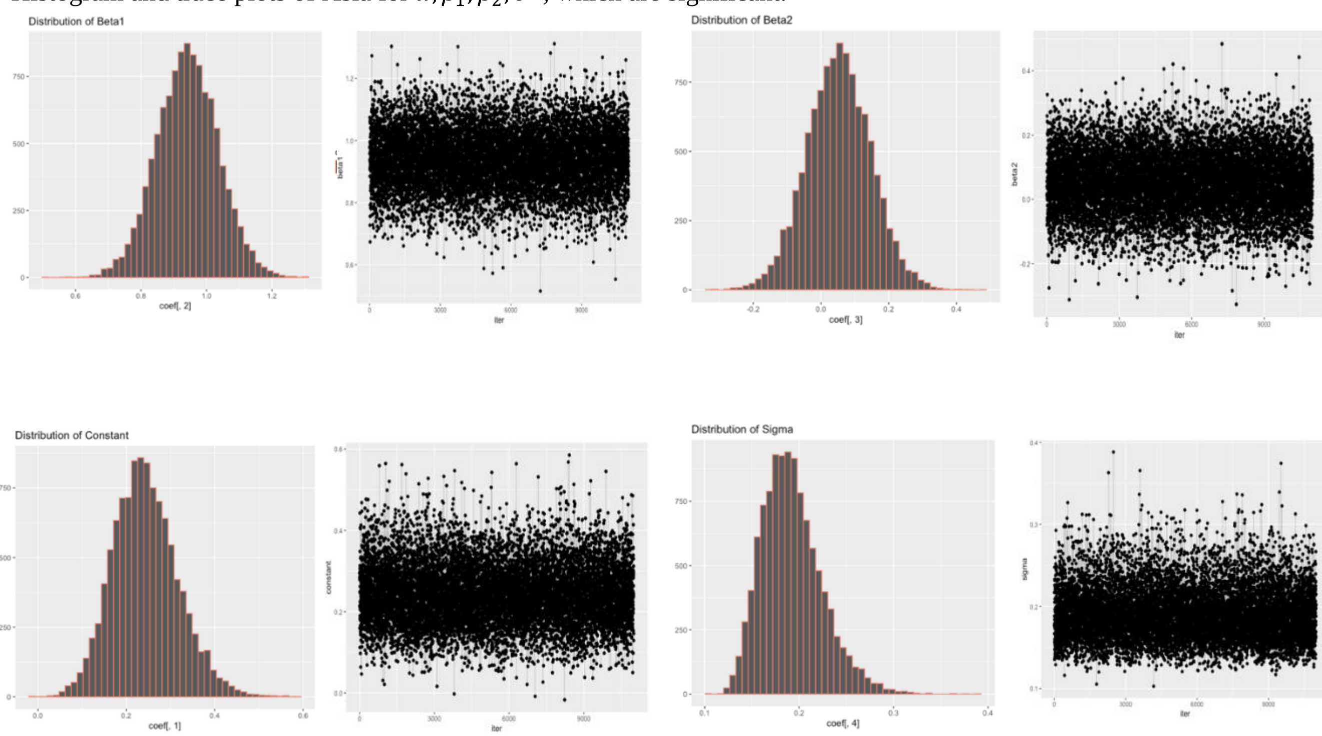


Forecast for next 5 years for United States of America

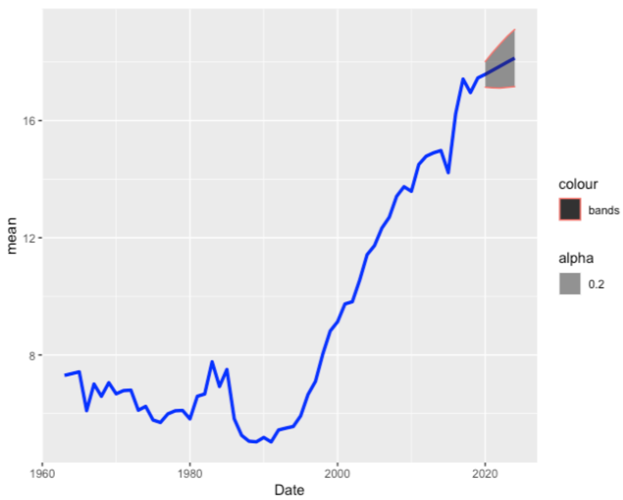



```
        Date    lower     mean    upper
58 2020-01-01 39.82700 41.40071 42.98024
59 2021-01-01 40.16865 42.04307 43.89678
60 2022-01-01 39.87215 42.13550 44.39467
61 2023-01-01 39.92362 42.44964 45.02719
62 2024-01-01 39.82801 42.65633 45.51170
> const <- mean(coef[,1])
> beta1 <- mean(coef[,2])
> beta2 <- mean(coef[,3])
> sigma <- mean(coef[,4])
> const
[1] 1.004706
> beta1
[1] 0.6046927
> beta2
[1] 0.3796223
> sigma
[1] 2.444173
```

Histogram and trace plots of Asia for $\alpha, \beta_1, \beta_2, \sigma^2$, which are significant.



Forecast for next 5 years for Asia




```
           Date    lower     mean    upper
58 2020-01-01 17.13777 17.57766 18.01450
59 2021-01-01 17.11599 17.71662 18.32644
60 2022-01-01 17.10725 17.85631 18.60907
61 2023-01-01 17.13304 17.99785 18.88011
62 2024-01-01 17.15792 18.13298 19.11018
> const <- mean(coef[,1])
> beta1 <- mean(coef[,2])
> beta2 <- mean(coef[,3])
> sigma <- mean(coef[,4])
> const
[1] 0.243364
> beta1
[1] 0.9418801
> beta2
[1] 0.05266765
> sigma
[1] 0.1922409
```

Validation of Model

- I have remove the last 5 years from data and train the model on remaining years and forecasted for the removed 5 years for validation
- Comparison table for United States of America

Years	Original Crop Production	Forecast Crop Production(with interval – lower , mean , upper)
2015	34.8	38.96049, 40.56323, 42.12828
2016	39.2	39.25492, 41.19394, 43.13202
2017	39.5	38.92396, 41.31705, 43.73401
2018	39.5	38.92769, 41.62066, 44.32378
2019	42.0	38.83851, 41.82587, 44.82580

- Comparison table for Asia

Years	Original Crop Production	Forecast Crop Production(with interval – lower , mean , upper)
2015	14.2	14.70510, 15.09226, 15.48614
2016	16.2	14.66096, 15.20922, 15.76060
2017	17.4	14.63063, 15.31606, 15.99959
2018	17.0	14.60752, 15.42147, 16.21781
2019	17.5	14.61474, 15.53005, 16.44135

Conclusion

By above tables we can conclude that, Bayesian Auto Regression Model is forecasting well without overfitting on train data. The main factor which eliminates the overfitting is including priors in the model which is ultimately the Bayesian approach.

Future Work

- P value for AR model is arbitrary chosen and there are formal tests such as the AIC (Akaike information criterion) and BIC (Bayesian information criterion), we can use to choose the best number of lags.

- All the priors are arbitrary chosen and have to do robustness tests by changing our initial priors and seeing if it changes the posterior significantly by visualization or fit a normal linear regression and determine the coefficients and take those as initial priors for Bayesian approach.
- Other more accurate and complex models like BVAR (Bayesian Vector Auto Regression)
- Model can be implemented on other regions and crops

References

- http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf

Code Dump

```
# Time Series Forecasting for Agriculture Crop production By Bayesian Analysis Project
# Authors :- Priyanka Surapaneni

# Setting the working directory
setwd("~/Downloads/Production_Crops_E_All_Data")

# Import required libraries
library(dplyr)
library(tidyverse)
library(tidyr)
library(matrixStats)
library(ggplot2)
library(reshape2)

# Read the data
data = read.csv("Production_Crops_E_All_Data_NOFLAG.csv")

# function which will take data and region
df_clean <- function(data,region) {
  # Filter with specified region and Apples
  data <- data %>% filter(Item == "Apples")
  vc <- c(region)
  data <- data[data$Area %in% vc,]

  # Remove unnecessary columns
  data <- select(data,-c("Area.Code", "Item.Code", "Element.Code","Unit" ))

  # Removing Y character in year columns
  df <- data
  colnames(df) <- substring(colnames(df), 2)
  colnames(df)[1:3] <- c("Area","Item","Element")

  # Normalizing production by area harvested
  df <- select(df,-c("Area", "Item", "Element"))
  df <- df[3,]/df[1,]
  df$Area <- data$Area[1]

  # Pivot the data into time series
  df_pivot <- df %>% pivot_longer(colnames(df[,-60]), names_to = "Year", values_to = "Crop_Pro
duction")
  df_pivot$Year <- as.numeric(df_pivot$Year)
  return (df_pivot)
}

#Data cleaning for USA and Asia
df_USA = df_clean(data,"United States of America")
```

```r
df_A = df_clean(data,"Asia")

# Plotting
# Line plot
#Change to df_A for Asia plots instead of df_USA
ggplot(df_USA, aes(x = Year, y = Crop_Production)) +
  geom_point(aes(color = Area), size = 2)  +geom_line(aes(color = Area), size = 1)
theme_minimal()+labs(title = "Apples")

# Histogram plot
hist(df_USA$Crop_Production,
     main="Histogram Plot for Crop Production",
     xlab="Crop Production",
)

# Density Plot
plot(density(df_USA$Crop_Production), main="Density  Plot for Crop Production",
     xlab="Crop Production")

# Bayesian Auto Regression Model
AR_Model <- function(df_pivot){
  # Reading Y
  Y.df <- select(df_pivot,-c("Area"))
  names <- c('Date', 'Crop_Production')
  Y <- data.frame(Y.df[,2])

  #lag value
  p = 2
  T1 = nrow(Y)

  # Coefficient Matrix
  regression_matrix  <- function(data,p,constant){
    nrow <- as.numeric(dim(data)[1])
    nvar <- as.numeric(dim(data)[2])

    Y1 <- as.matrix(data, ncol = nvar)
    X <- embed(Y1, p+1)
    X <- X[,(nvar+1):ncol(X)]
    if(constant == TRUE){
      X <-cbind(rep(1,(nrow-p)),X)
    }
    Y = matrix(Y1[(p+1):nrow(Y1),])
    nvar2 = ncol(X)
    return = list(Y=Y,X=X,nvar2=nvar2,nrow=nrow)
  }

  # Companion Matrix of Coefficient Matrix for checking stability
  ar_companion_matrix <- function(beta){
    #check if beta is a matrix
    if (is.matrix(beta) == FALSE){
      stop('error: beta needs to be a matrix')
    }
    # dont include constant
    k = nrow(beta) - 1
    FF <- matrix(0, nrow = k, ncol = k)

    #insert identity matrix
    FF[2:k, 1:(k-1)] <- diag(1, nrow = k-1, ncol = k-1)

    temp <- t(beta[2:(k+1), 1:1])
    #state space companion form
```

```r
  #Insert coeffcients along top row
  FF[1:1,1:k] <- temp
  return(FF)
}
results = list()
results <- regression_matrix(Y, p, TRUE)
X <- results$X
Y <- results$Y
nrow <- results$nrow
nvar <- results$nvar
# Initialise Priors
B <- c(rep(0, nvar))
B <- as.matrix(B, nrow = 1, ncol = nvar)
B0 = B
sigma0 <- diag(1,nvar)
T0 = 1 # prior degrees of freedom
D0 = 0.1 # prior scale (theta0)
# initial value for variance
sigma2 = 1

reps = 15000
burn = 4000
# here horizon is no of years to predict plus lag = 5 years + 2 lag
horizon = 7
out = matrix(0, nrow = reps, ncol = nvar + 1)
colnames(out) <- c("constant", "beta1","beta2", "sigma")
out1 <- matrix(0, nrow = reps, ncol = horizon)

gibbs_sampler <- function(X,Y,B0,sigma0,sigma2,theta0,D0,reps,out,out1){
  for(i in 1:reps){
    if (i %% 1000 == 0){
      print(sprintf("Interation: %d", i))
    }
    M = solve(solve(sigma0) + as.numeric(1/sigma2) * t(X) %*% X) %*%
      (solve(sigma0) %*% B0 + as.numeric(1/sigma2) * t(X) %*% Y)

    V = solve(solve(sigma0) + as.numeric(1/sigma2) * t(X) %*% X)

    chck = -1
    while(chck < 0){   # check for stability

      B <- M + t(rnorm(p+1) %*% chol(V))

      # Check : not stationary for 3 lags
      b = ar_companion_matrix(B)
      ee <- max(sapply(eigen(b)$values,abs))
      if( ee<=1){
        chck=1
      }
    }
    # compute residuals
    resids <- Y- X%*%B
    T2 = T0 + T1
    D1 = D0 + t(resids) %*% resids

    # keeps samples after burn period
    out[i,] <- t(matrix(c(t(B),sigma2)))
```

```
#draw from Inverse Gamma
      z0 = rnorm(T1,1)
      z0z0 = t(z0) %*% z0
      sigma2 = D1/z0z0

      # keeps samples after burn period
      out[i,] <- t(matrix(c(t(B),sigma2)))

      # compute 2 year forecasts
      yhat = rep(0,horizon)
      end = as.numeric(length(Y))
      yhat[1:2] = Y[(end-1):end,]
      cfactor = sqrt(sigma2)
      X_mat = c(1,rep(0,p))
      for(m in (p+1):horizon){
        for (lag in 1:p){
          #create X matrix with p lags
          X_mat[(lag+1)] = yhat[m-lag]
        }
        # Use X matrix to forecast yhat
        yhat[m] = X_mat %*% B + rnorm(1) * cfactor
      }

      out1[i,] <- yhat
    }
    return = list(out,out1)
  }
  results1 <- gibbs_sampler(X,Y,B0,sigma0,sigma2,T0,D0,reps,out,out1)
  # burn first 4000
  coef <- results1[[1]][(burn+1):reps,]
  forecasts <- results1[[2]][(burn+1):reps,]
  return = list(coef,forecasts,Y)
}

# Pass df_A for asia results instead of df_USA
results = AR_Model(df_USA)
coef = results[[1]]
forecasts = results[[2]]
Y =  results[[3]]

# Creating dataframe for plotting
# Modify accordingly for Asia
res_df <-as.data.frame(matrix(nrow=reps-burn,ncol=5))
colnames(res_df)<-c("constant","beta1","beta2","sigma","iter")
res_df$constant <- coef[,1]
res_df$beta1 <- coef[,2]
res_df$beta2 <- coef[,3]
res_df$sigma <- coef[,4]
res_df$iter <- 1:(reps-burn)

#Trace plots
ggplot(res_df,aes(x = iter, y = beta1))+geom_point()+ geom_line(alpha = 0.2)+labs(y ='beta1')
ggplot(res_df,aes(x = iter, y = beta2))+geom_point()+ geom_line(alpha = 0.2)+labs(y ='beta2')
ggplot(res_df,aes(x = iter, y = sigma))+geom_point()+ geom_line(alpha = 0.2)+labs(y ='sigma')
ggplot(res_df,aes(x = iter, y = constant))+geom_point()+ geom_line(alpha = 0.2)+labs(y ='const
ant')

#checking the means
const <- mean(coef[,1])
beta1 <- mean(coef[,2])
beta2 <- mean(coef[,3])
```

```r
sigma <- mean(coef[,4])

# Histogram plots
qplot(coef[,1], geom = "histogram", bins = 45, main = 'Distribution of Constant',
      colour="#FF9999")
qplot(coef[,2], geom = "histogram", bins = 45,main = 'Distribution of Beta1',
      colour="#FF9999")
qplot(coef[,3], geom = "histogram", bins = 45,main = 'Distribution of Beta2',
      colour="#FF9999")
qplot(coef[,4], geom = "histogram", bins = 45,main = 'Distribution of Sigma',
      colour="#FF9999")


#quantiles for all data points, makes plotting easier
post_means <- colMeans(coef)
forecasts_m <- as.matrix(colMeans(forecasts))
#Creating error bands/credible intervals around our forecasts
error_bands <- colQuantiles(forecasts,prob = c(0.16,0.84))
Y_temp = cbind(Y,Y)
error_bands <- rbind(Y_temp, error_bands[3:dim(error_bands)[1],])
all <- as.matrix(c(Y[1:(length(Y)-2)],forecasts_m))
forecasts.mat <- cbind.data.frame(error_bands[,1],all, error_bands[,2])
names(forecasts.mat) <- c('lower', 'mean', 'upper')
# create date vector for plotting
Date <- seq(as.Date('1963/1/1'), by = 'year', length.out = dim(forecasts.mat)[1])
data.plot <- cbind.data.frame(Date, forecasts.mat)
data_subset <- data.plot[1:62,]
data_fore <- data.plot[58:62,]
ggplot(data_subset, aes(x = Date, y = mean)) + geom_line(colour = 'blue', lwd = 1.2) +
  geom_ribbon(data = data_fore,aes(ymin = lower, ymax = upper , colour = "bands", alpha = 0.2)
)
```