Project

Maduka Balasooriya

Question: Implement a Gibbs sampler to generate a bivariate normal chain (Xt, Yt) with zero means, unit standard deviations, and correlation 0.9. Plot the generated sample after discarding a suitable burn-in sample. Fit a simple linear regression model Y = (Beta0) + (Beta1)X to the sample and check the residuals of the model for normality and constant variance.

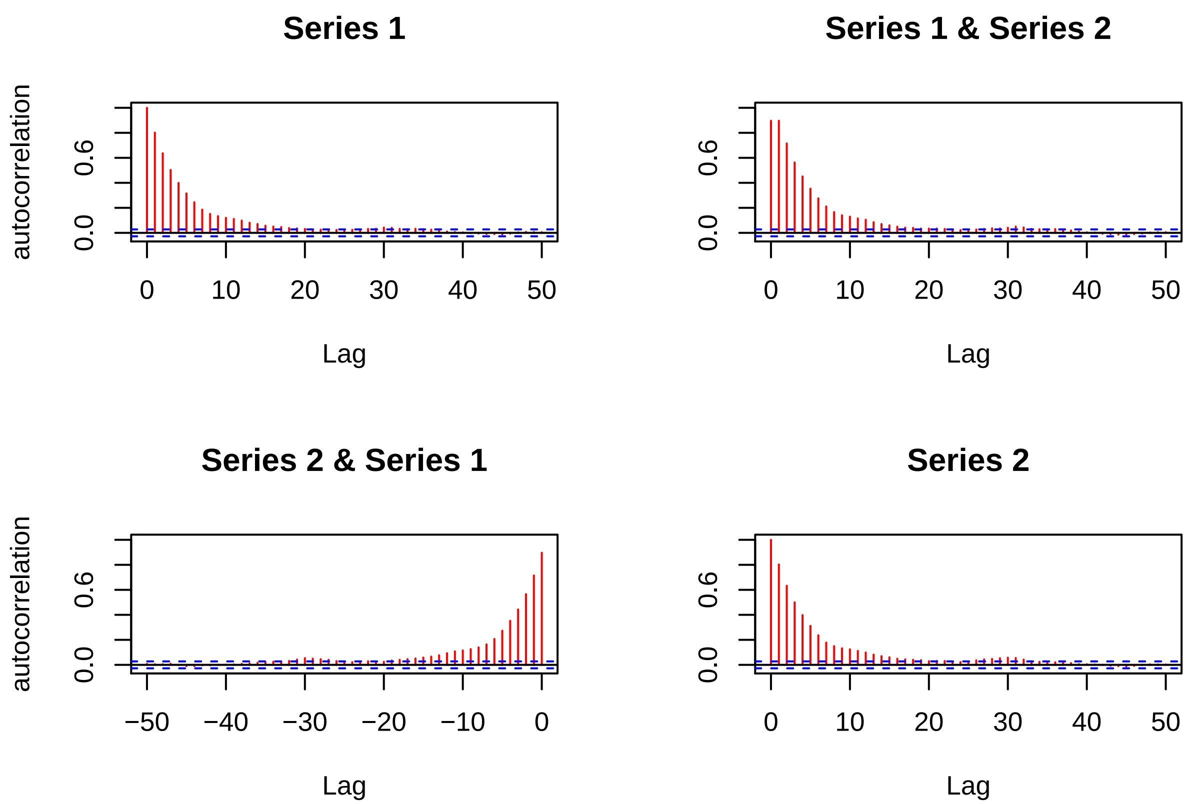
Plot the autocorrelation to check whether the generated values are independent

```
library(VGAM) # install VGAM package
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
#initialize constants and parameters
set.seed(1234)
N <- 5000 #length of chain
X <- matrix(0, N, 2) #the chain, a bivariate sample
rho <- 0.9 #correlation
mu_x <- 0
mu_y <- 0
sigma_x <- 1
sigma_y<- 1
s_x <- sqrt(1-rho^2)*sigma_x
s_y <- sqrt(1-rho^2)*sigma_y
###### generate the chain #####
X[1, ] <- c(mu_x, mu_y) #initialize
for (i in 2:N) {
  y <- X[i-1, 2]
  m_x <- mu_x + rho * (y - mu_y) * sigma_x/sigma_y
  X[i, 1] <- rnorm(1, m_x, s_x)
  x <- X[i, 1]
  m_y <- mu_y + rho * (x - mu_x) * sigma_y/sigma_x
  X[i, 2] <- rnorm(1, m_y, s_y)
}
x<-X[1:N, ]
acf(x, col = "red", lag.max = 50, ylab = "autocorrelation")
```




```
#It can be seen that the highly correlated values in the first part
#of the chain. This tells us that the chain has not converged until
#1000th (20th lag).Therfore first 1000 observations are the burning sample.
#The last 4000 values can now be treated as the generated values
#from the bivariate normal distribution.
```

take the sample after discarding burn in sample

```
burn <- 1000 #burn-in length
b <- burn + 1
w <- X[b:N, ]
```

```
#sample correlation is very close to population correlation(=0.9)# compare sample statistics to paramet
colMeans(w)
```

```
## [1] 0.04638344 0.03484338
```

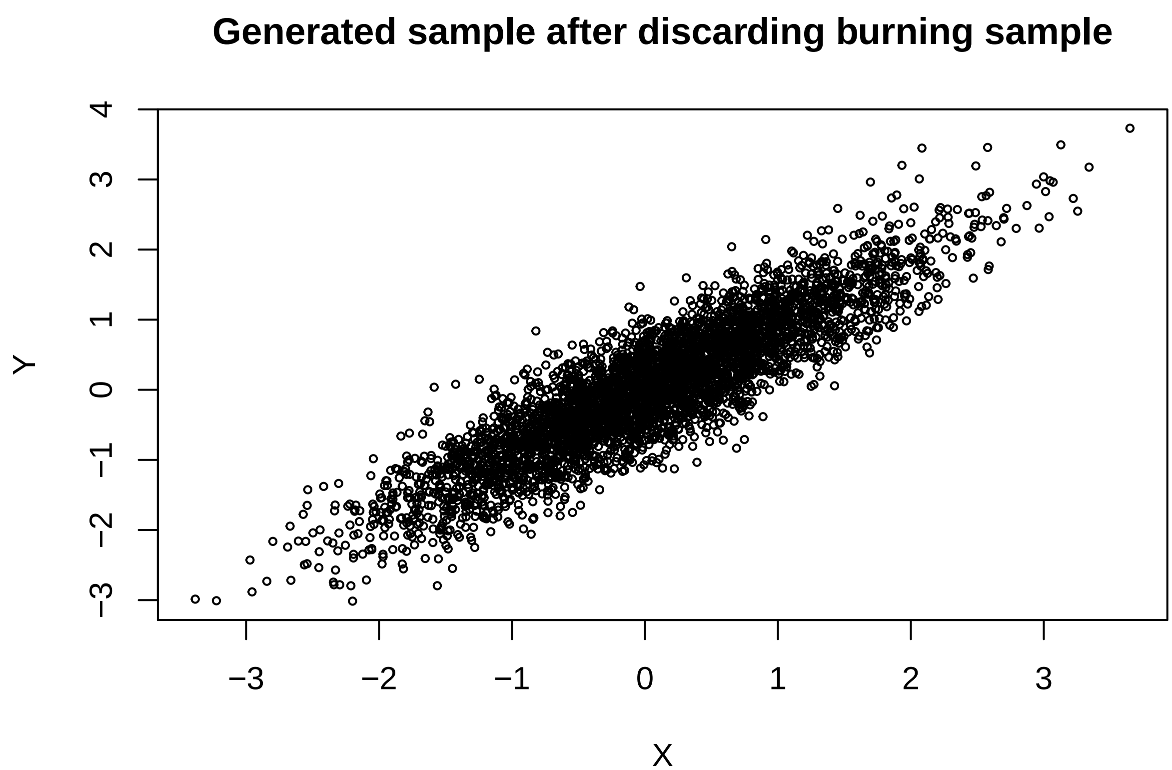
```
#sample means are very close to population mean =0
cov(w)
```

```
##            [,1]      [,2]
## [1,] 0.9344271 0.8436075
## [2,] 0.8436075 0.9513264
```

```r
#sample variances are very close to population variance(=1)
cor(w)
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.8947526
## [2,] 0.8947526 1.0000000
```

```r
par(mfcol = c(1,1))
plot(w, main="Generated sample after discarding burning sample", cex=.5, xlab=bquote(X),
     ylab=bquote(Y), ylim=range(w[,2]))
```

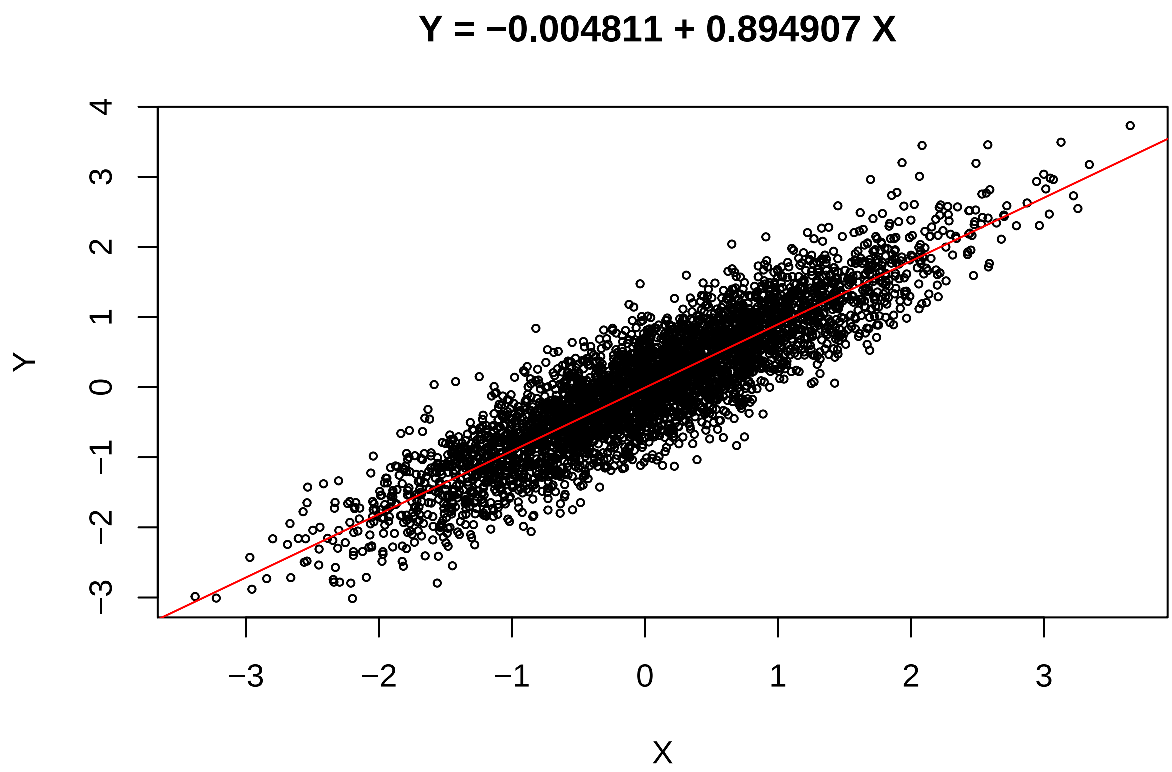


Fit linear regression model

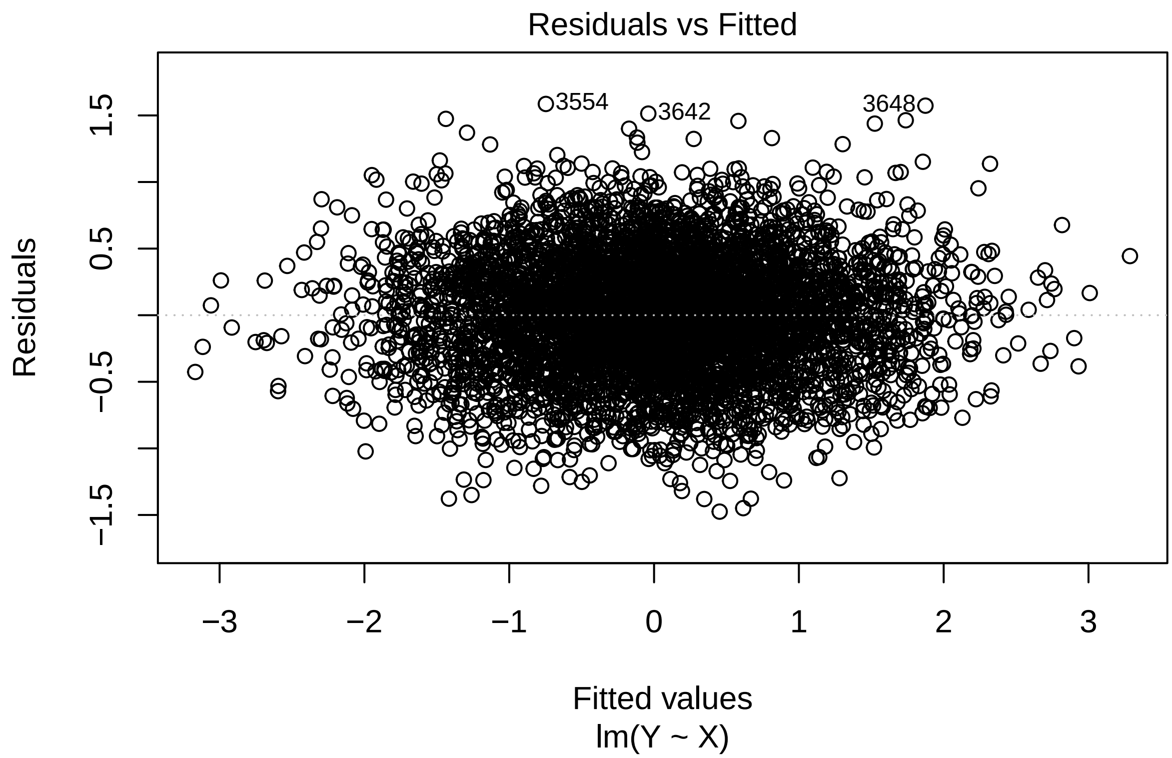
```r
Y<- X[, 2]
X<-X[, 1]
fit <- lm(Y~ X)
print(fit)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Coefficients:
## (Intercept)            X
##   -0.006763     0.902666
```

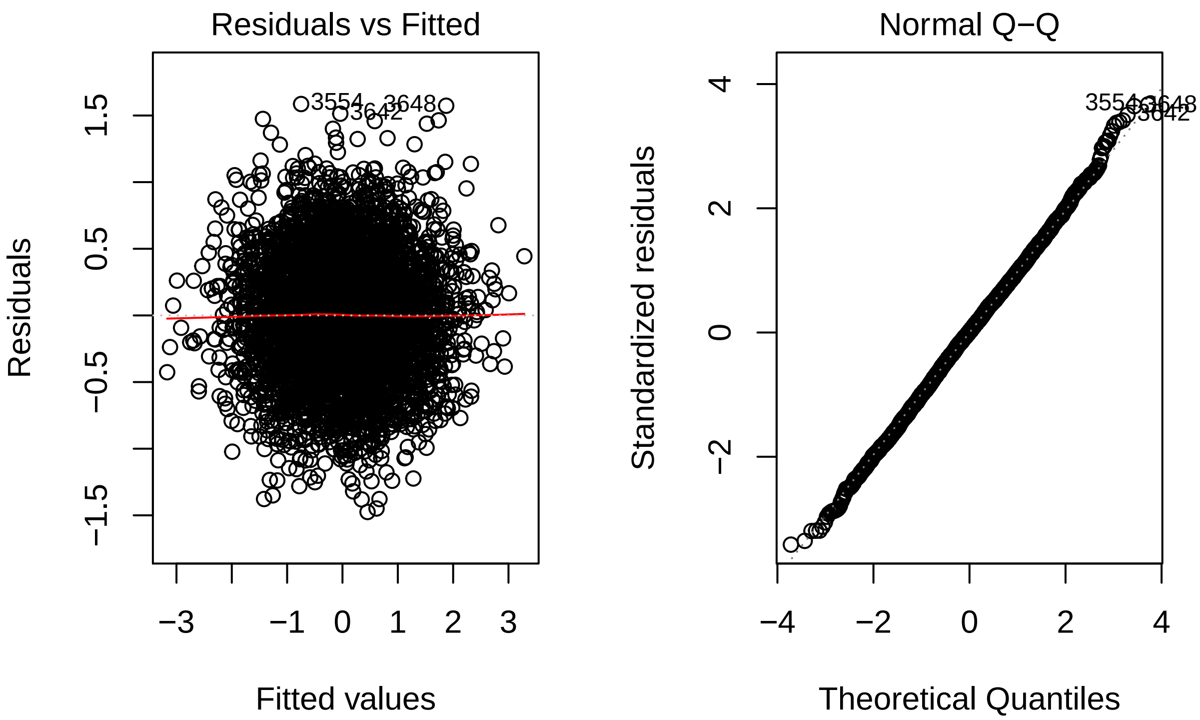
```
#The fitted regression line is:
#Y = -0.004811+0.894907 X


plot(w, main="Y = -0.004811 + 0.894907 X ", cex=.5, xlab=bquote(X),
     ylab=bquote(Y), ylim=range(w[,2]))
abline(fit,col="red")#regression line
```




```
par(mfcol = c(1,1))  # back to one figure panel
plot(fit, which = 1, add.smooth = FALSE)# residual plot
```




```r
#a plot of residuals vs fits (1) and a QQ plot to check
#for normality and constant variance of residuals
par(mfcol = c(1,2))
plot(fit, which = 1:2, add.smooth=TRUE)
```



According to Normal QQ plot It s clear that resuduals follow normal distribution. According to the residual plot, it is clear that residuals have constant variance