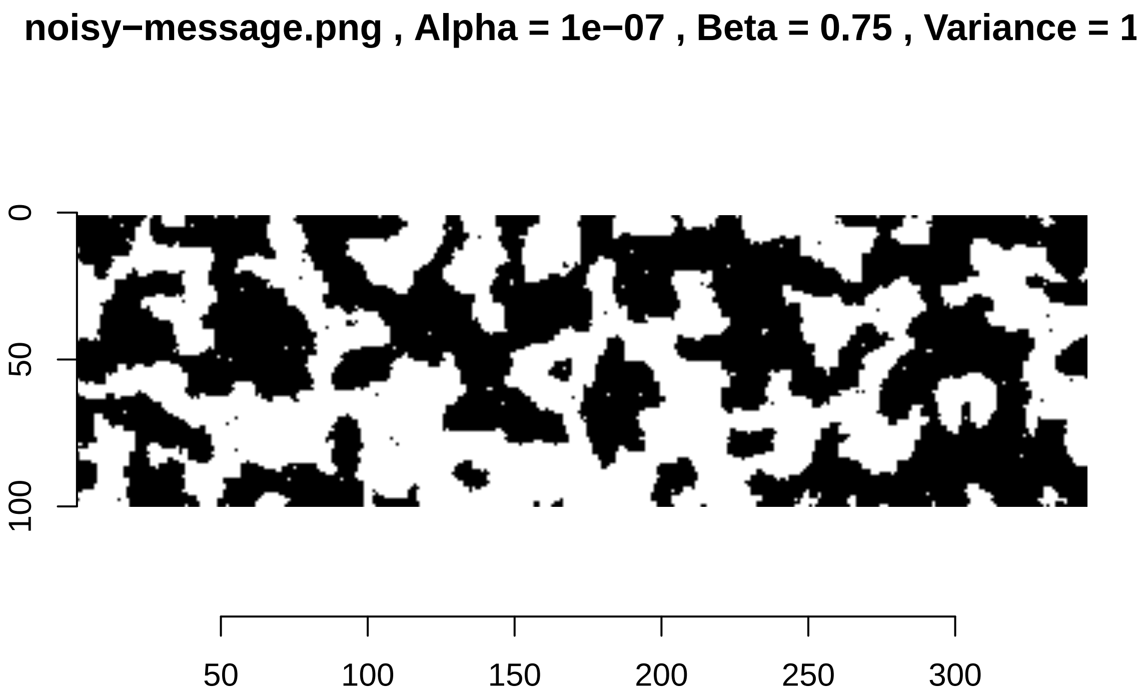
CS6190: Probabilistic Modeling Homework 3 MCMC

Gopal Menon

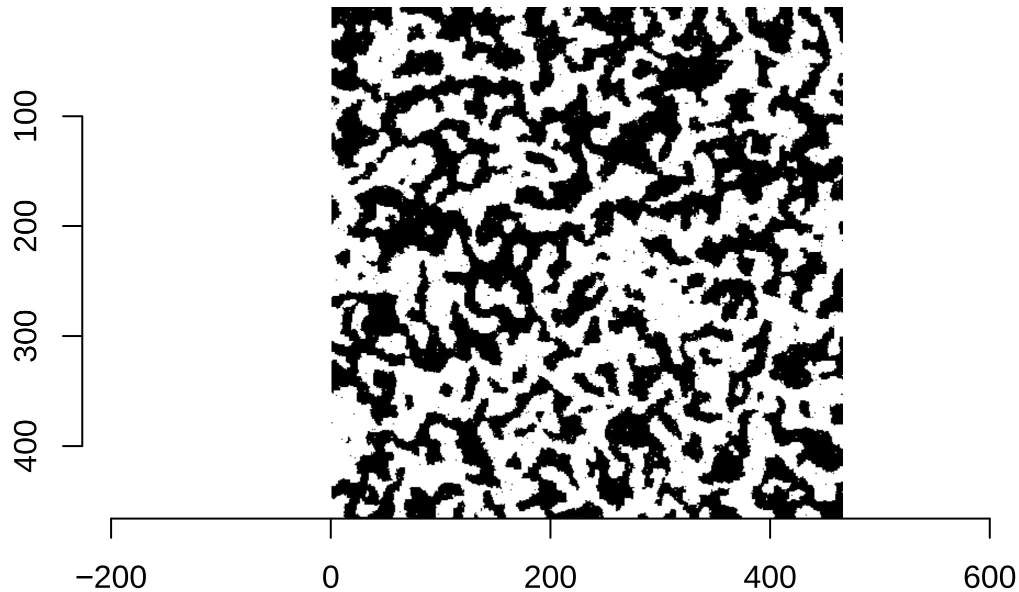
16 April, 2018

1(a) Gibbs Sampling of noisy image with Ising prior only.

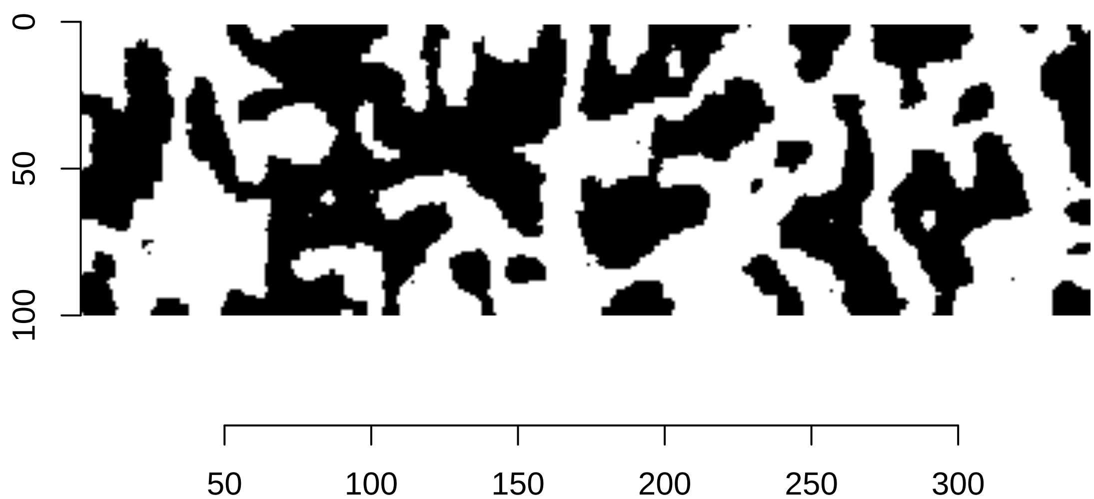
The sampled images are shown for various values of α and β. The variance values shown should be disregarded as that part of the energy function is not used. α values needed to be close to zero or zero for the image to be visible.



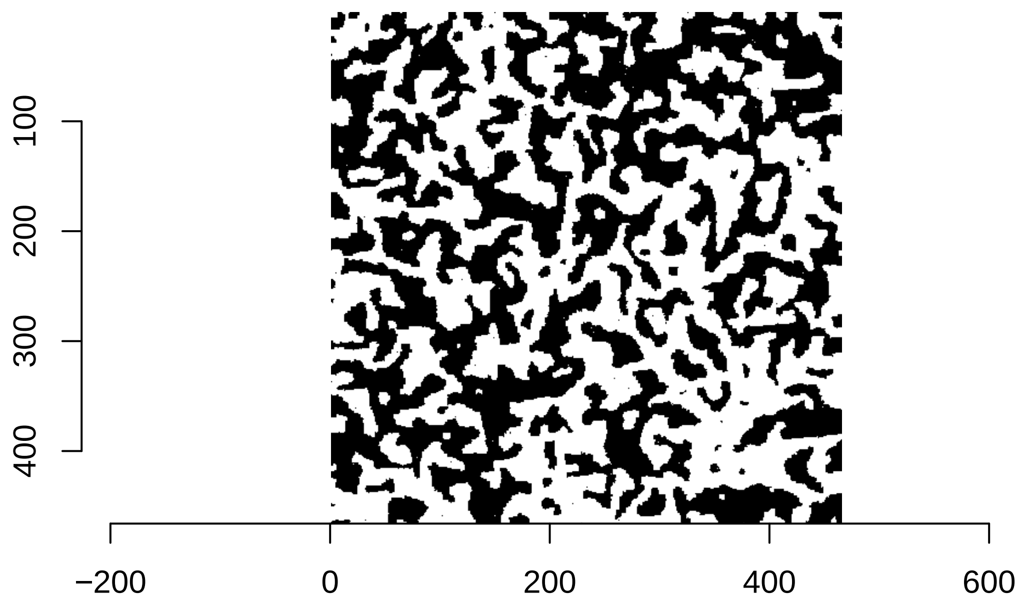
noisy−yinyang.png , Alpha = 1e−07 , Beta = 0.75 , Variance = 1



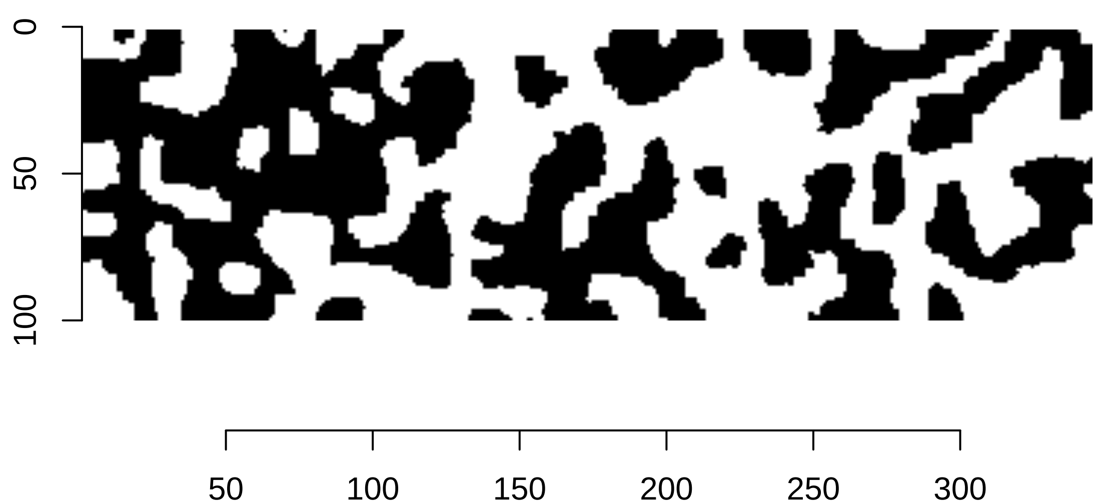
noisy−message.png , Alpha = 1e−07 , Beta = 1 , Variance = 1

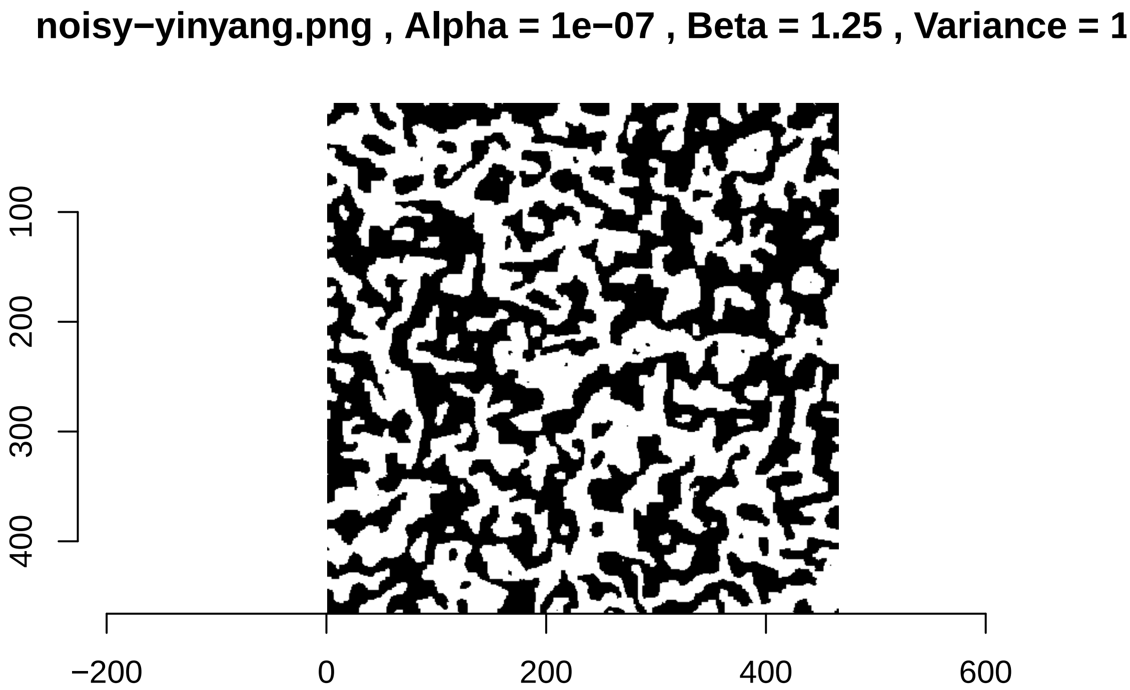


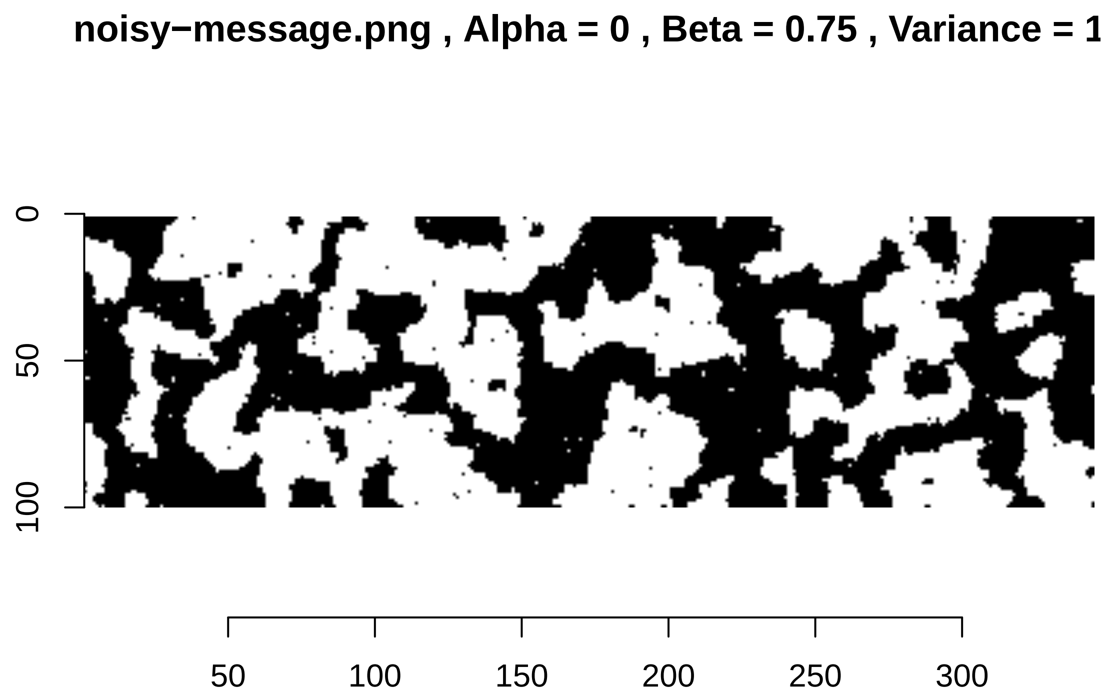
noisy−yinyang.png , Alpha = 1e−07 , Beta = 1 , Variance = 1



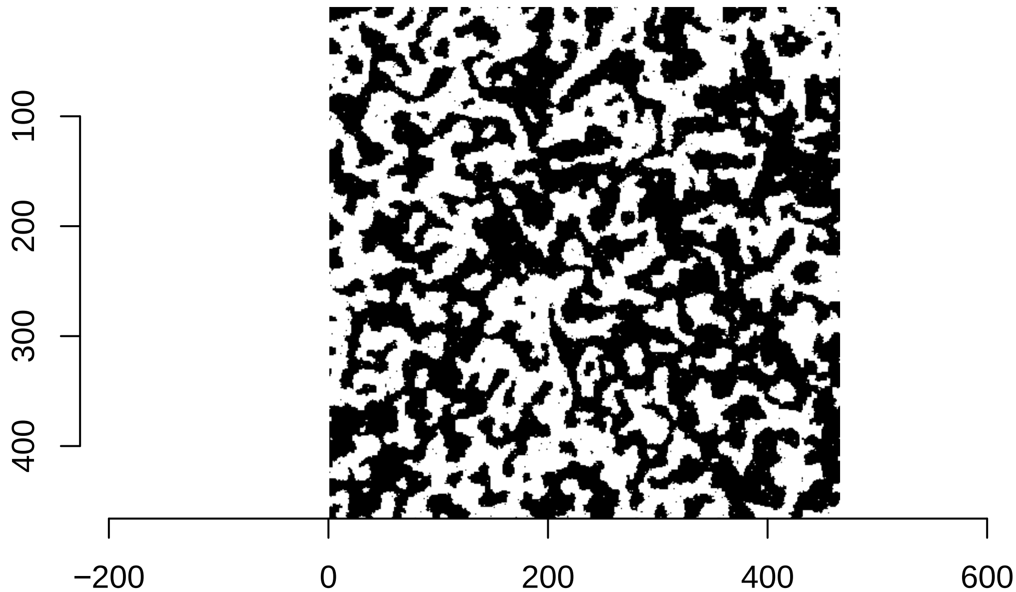
noisy−message.png , Alpha = 1e−07 , Beta = 1.25 , Variance = 1



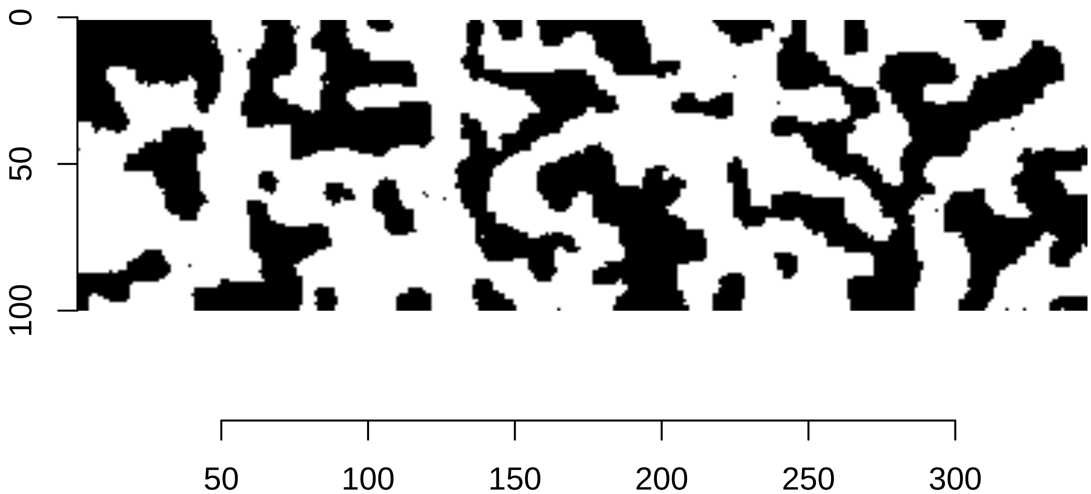

noisy−yinyang.png , Alpha = 1e−07 , Beta = 1.25 , Variance = 1
100
200
300
400
−200
0
200
400
600

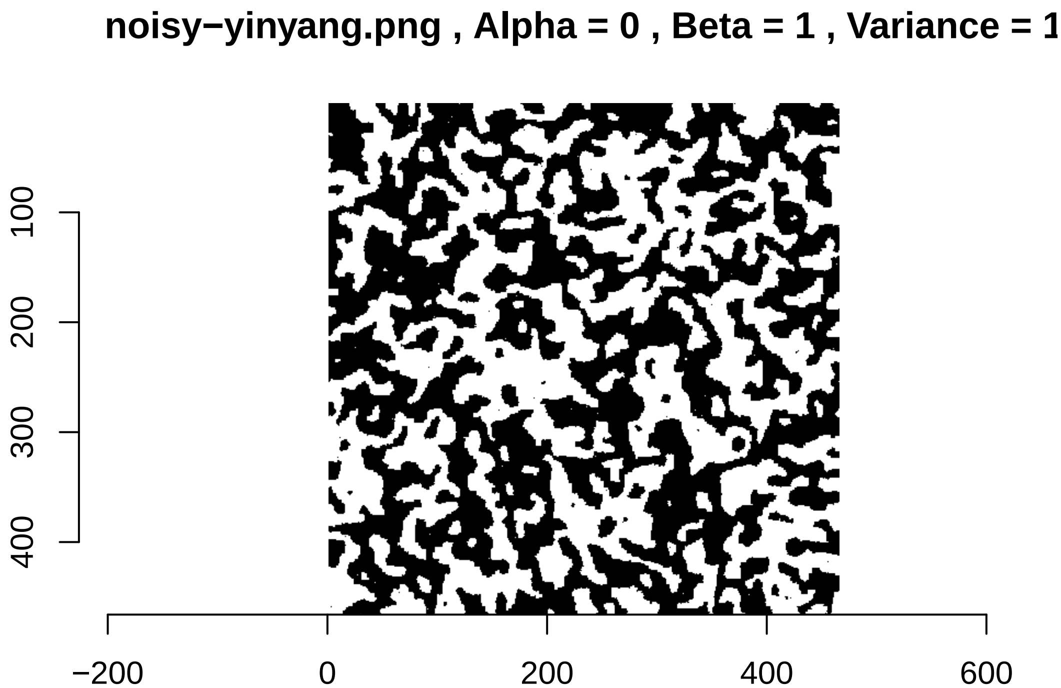

noisy−message.png , Alpha = 0 , Beta = 0.75 , Variance = 1
0
50
100
50
100
150
200
250
300

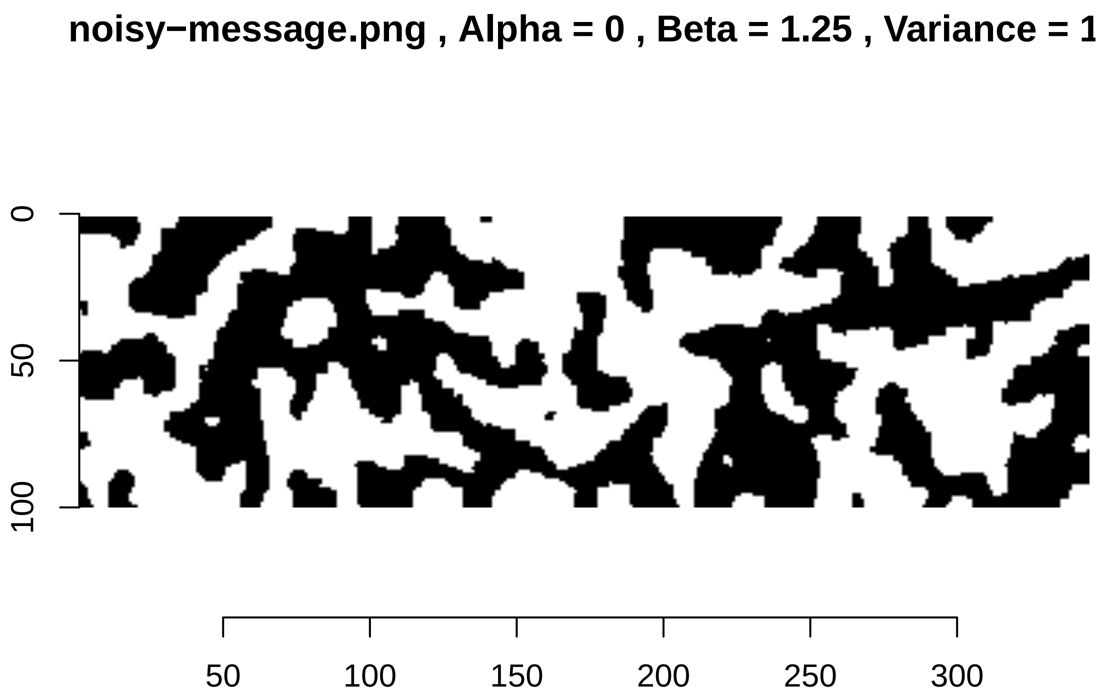
noisy−yinyang.png , Alpha = 0 , Beta = 0.75 , Variance = 1

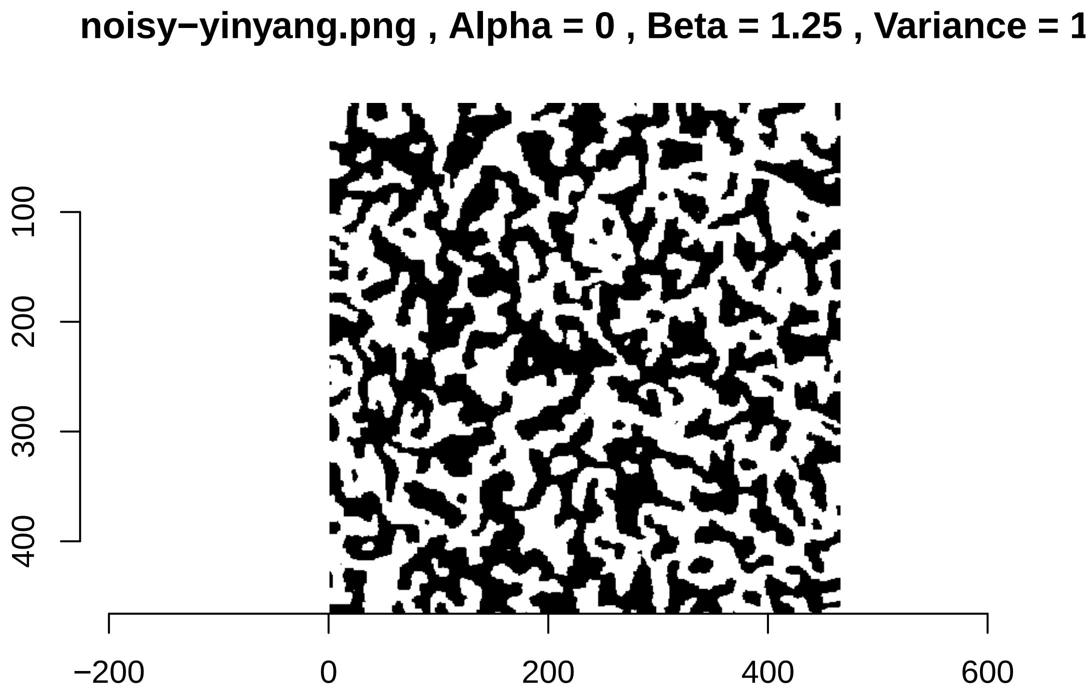


noisy−message.png , Alpha = 0 , Beta = 1 , Variance = 1

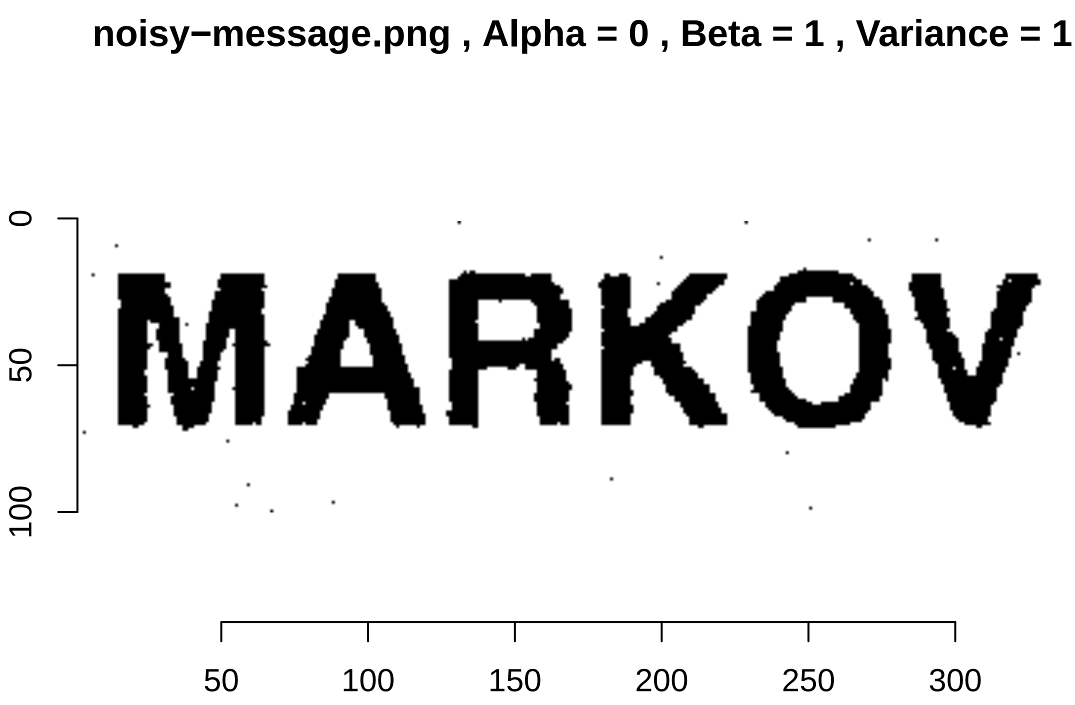


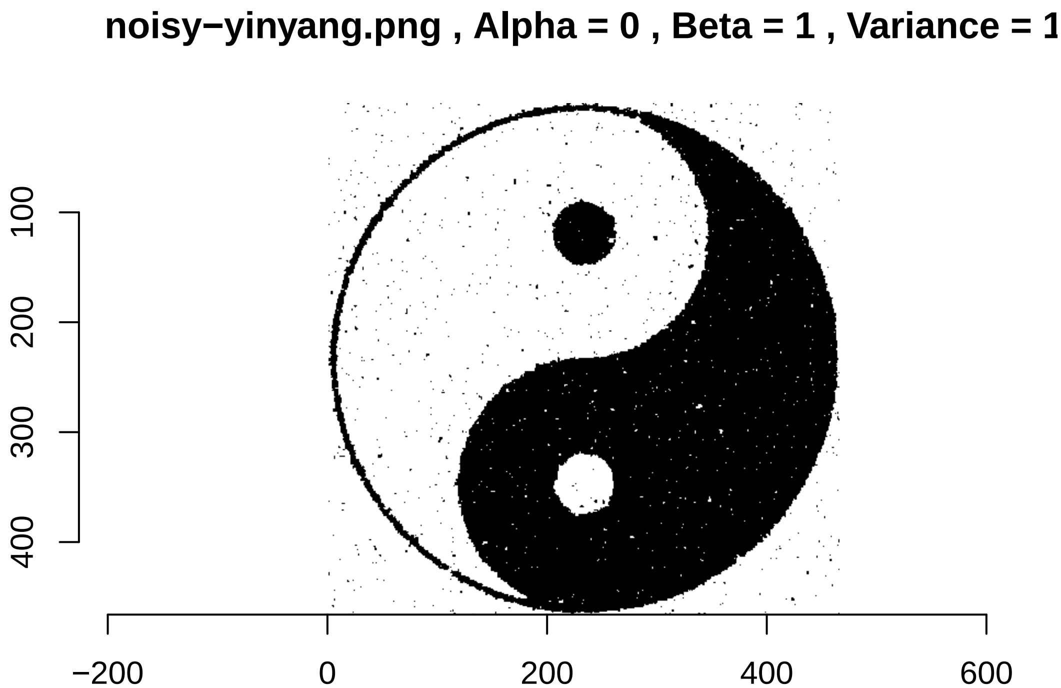

noisy−yinyang.png , Alpha = 0 , Beta = 1 , Variance = 1
100
200
300
400
−200
0
200
400
600


noisy−message.png , Alpha = 0 , Beta = 1.25 , Variance = 1
0
50
100
50
100
150
200
250
300



1(b) Here are the posterior images for 10 burn in and 20 denoising iterations. The parameter values I used were $\alpha = 0$, $\beta = 1$ and $\sigma^2 = 1$. The images converged really fast in the first few iterations. The function generate_gifs in the R code can be called to generate gif images that show the image transition. I did not try include the gif images in the output as it did not seem to be platform independent.





1(c) For estimating σ^2, I started with a wide guess range and then did a maximum likelihood computation at each iteration. Final estimates for σ^2:

```
## [1] "Standard deviation estimate for noisy message: 1.188"
## [1] "Standard deviation estimate for noisy yinyang: 1.58503521060864"
```

2. Say you are given data (X, Y), with $X \in \mathbb{R}^d$ and $Y \in \{0, 1\}$. The goal is to train a classifier that will predict an unknown class label $\tilde{y}$ from a new data point $\tilde{x}$. Consider the following model:

$$Y \sim Ber\left(\frac{1}{1+e^{-X^T\beta}}\right),$$
$$\beta \sim N(0, \sigma^2 I).$$

This is a **Bayesian logistic regression** model. Your goal is to derive and implement a Hamiltonian Monte Carlo sampler for doing Bayesian inference on β.

(a) Write down the formula for the unormalized posterior of $\beta|Y$, i.e.,

$$\begin{aligned} p(\beta|y; x, \sigma) &\propto \prod_{i=1}^{n} p(y_i|\beta; x_i) p(\beta; \sigma) \\ &= \prod_{i=1}^{n} \left(\left[\frac{1}{1+e^{-x_i^T\beta}}\right]^{y_i} \left[1 - \frac{1}{1+e^{-x_i^T\beta}}\right]^{1-y_i} \right) \frac{1}{\sqrt{(2\pi)^k |\sigma^2 I|}} e^{\left(-\frac{1}{2}\beta^T(\sigma^2 I)^{-1}\beta\right)} \end{aligned}$$

where, $\beta \in \mathbb{R}^k$, and $k = d + 1$

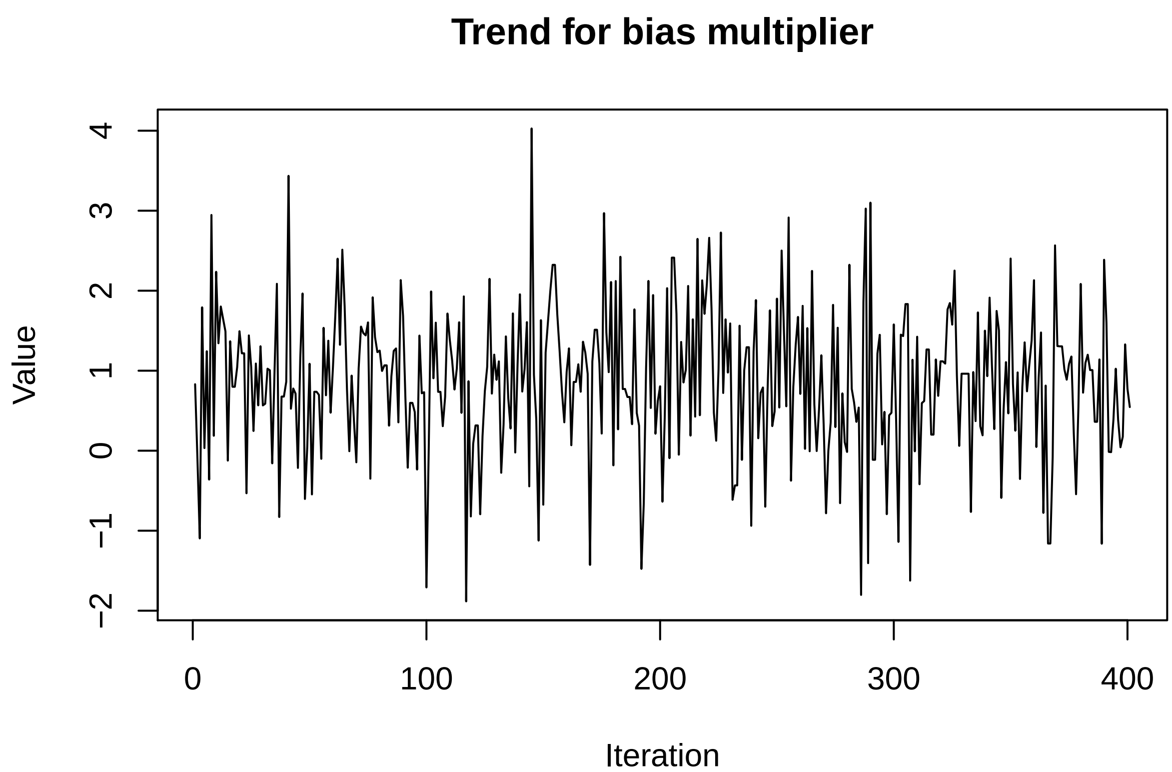
(b) Show that this posterior is proportional to $exp(-U(\beta))$, where

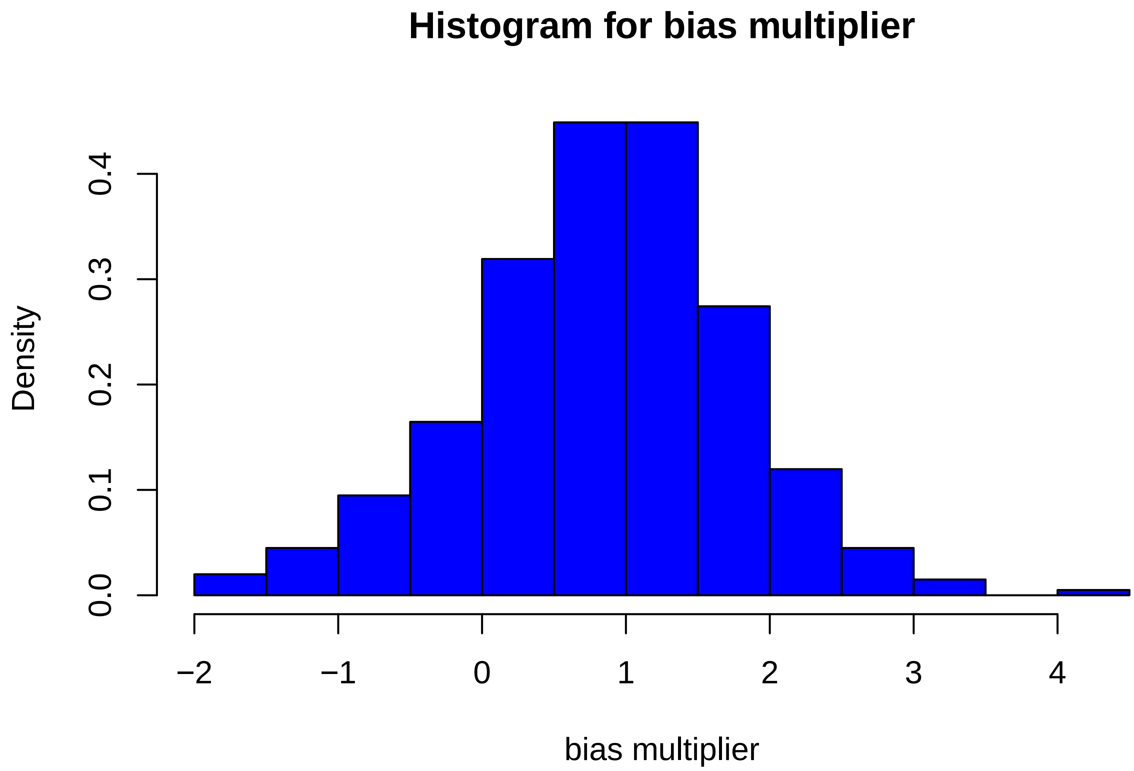
$$U(\beta) = \sum_{i=1}^{n} (1 - y_i) x_i^T \beta + \log(1 + e^{-x_i^T\beta}) + \frac{1}{2\sigma^2}\|\beta\|^2$$

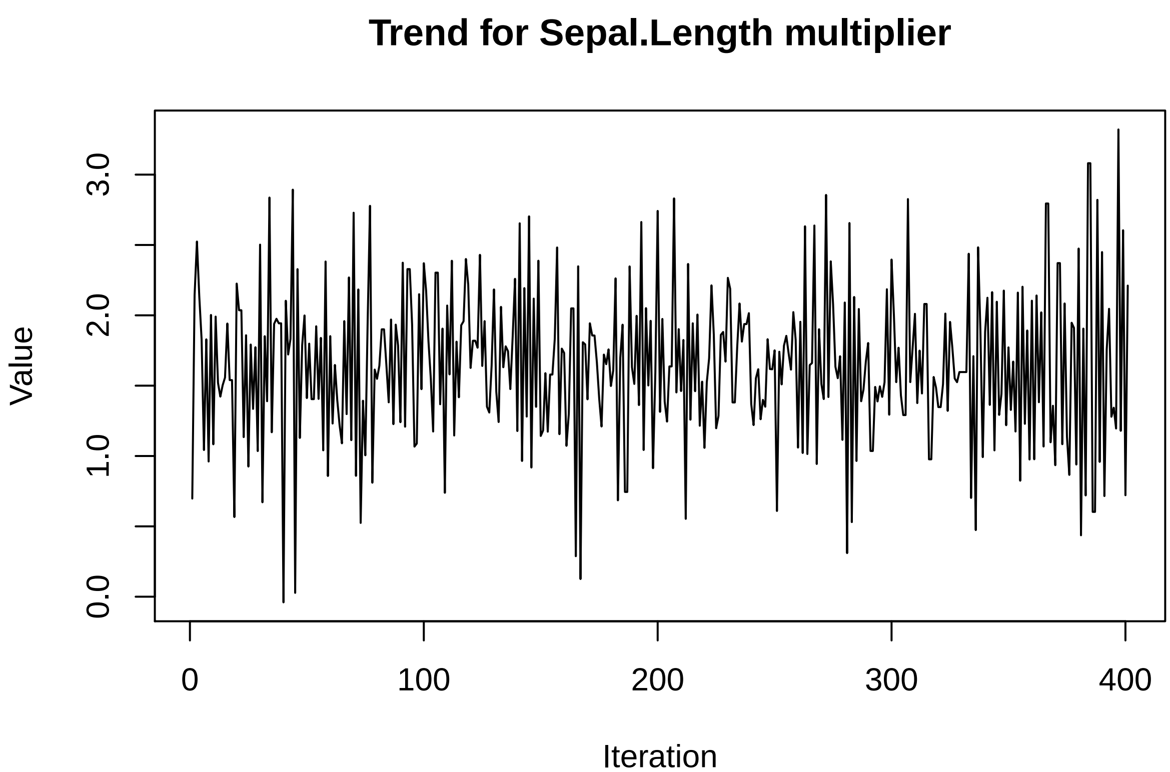
$$
\begin{aligned}
p(\beta|y;x,\sigma) &\propto \prod_{i=1}^{n}\left(\left[\frac{1}{1+e^{-x_i^T\beta}}\right]^{y_i}\left[1-\frac{1}{1+e^{-x_i^T\beta}}\right]^{1-y_i}\right)\frac{1}{\sqrt{(2\pi)^k|\sigma^2 I|}}e^{\left(-\frac{1}{2}\beta^T\left(\sigma^2 I\right)^{-1}\beta\right)} \\
&= \prod_{i=1}^{n}\left(\left[\frac{1}{1+e^{-x_i^T\beta}}\right]^{y_i}\left[\frac{e^{-x_i^T\beta}}{1+e^{-x_i^T\beta}}\right]^{1-y_i}\right)\frac{1}{\sqrt{(2\pi)^k|\sigma^2 I|}}e^{\left(-\frac{\|\beta\|^2}{2\sigma^2}\right)} \\
\log\left(p(\beta|y;x,\sigma)\right) &\propto -\sum_{i=1}^{n} y_i\log\left(1+e^{-x_i^T\beta}\right)+\sum_{i=1}^{n}(1-y_i).-x_i^T\beta-\sum_{i=1}^{n}(1-y_i)\log\left(1+e^{-x_i^T\beta}\right)-\frac{1}{2\sigma^2}\|\beta\|^2 \\
&= -\sum_{i=1}^{n} y_i\log\left(1+e^{-x_i^T\beta}\right)-\sum_{i=1}^{n}(1-y_i)x_i^T\beta-\sum_{i=1}^{n}\log\left(1+e^{-x_i^T\beta}\right)+\sum_{i=1}^{n} y_i\log\left(1+e^{-x_i^T\beta}\right)-\frac{\|\beta\|^2}{2\sigma^2} \\
&= -\sum_{i=1}^{n}(1-y_i)x_i^T\beta-\sum_{i=1}^{n}\log\left(1+e^{-x_i^T\beta}\right)-\frac{1}{2\sigma^2}\|\beta\|^2 \\
\Rightarrow p(\beta|y;x,\sigma) &\propto exp\left(-U(\beta)\right), \text{where} \\
U(\beta) &= \sum_{i=1}^{n}(1-y_i)x_i^T\beta+\sum_{i=1}^{n}\log\left(1+e^{-x_i^T\beta}\right)+\frac{1}{2\sigma^2}\|\beta\|^2
\end{aligned}
$$

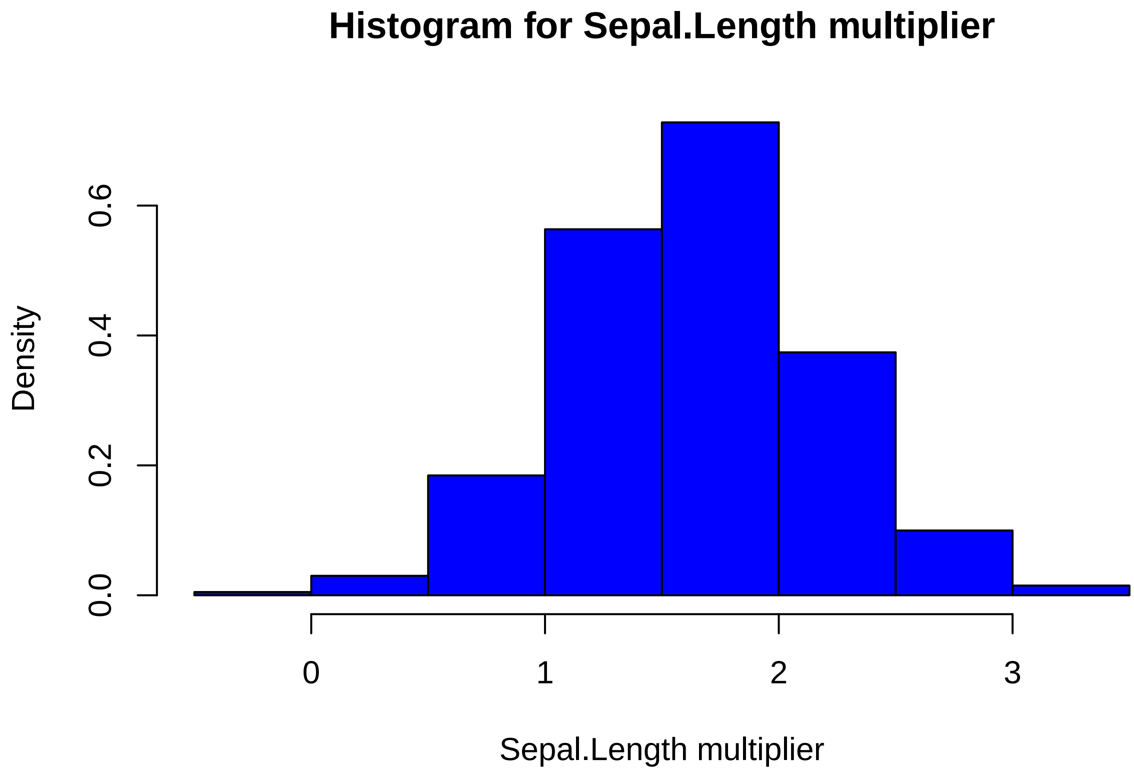
2(d) The Monte Carlo estimate of posterior predictive probability of the label was computed by first doing a dot product of each β with the test data and then averaging it out for each data point. A value of 0.5 and above was classified as Versicolor.

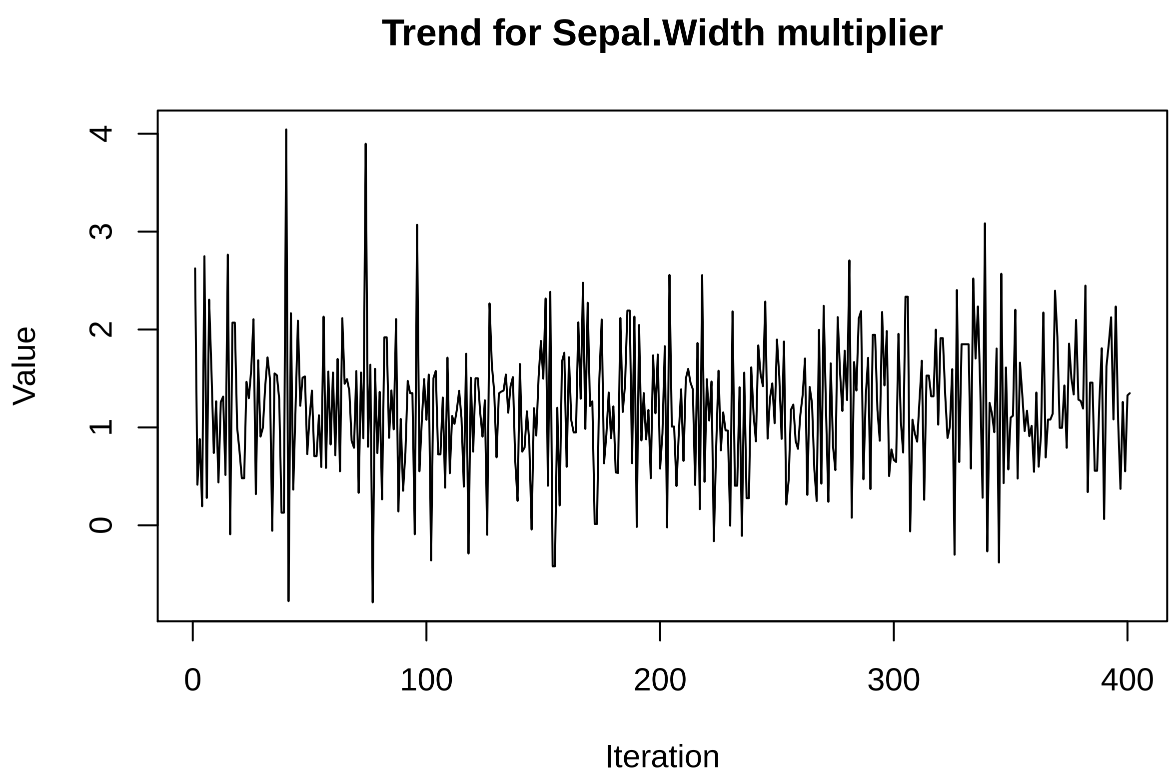
2(e) Trace plots of the β sequence and histograms are shown below.

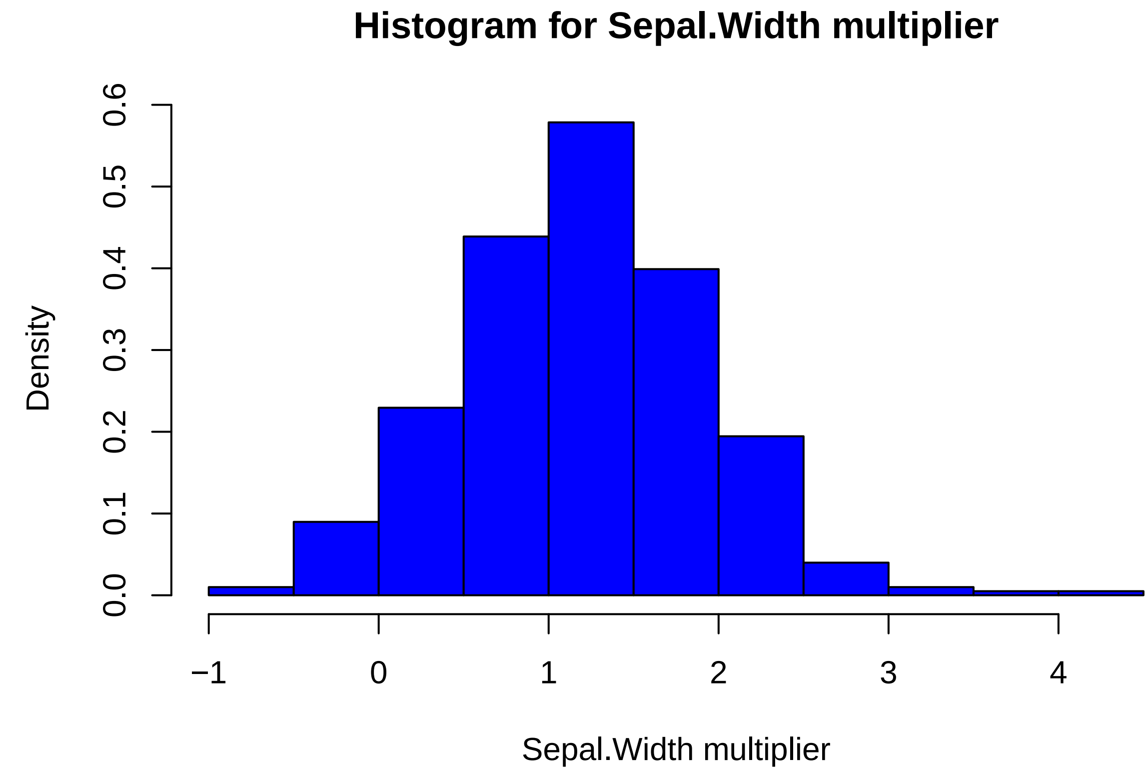


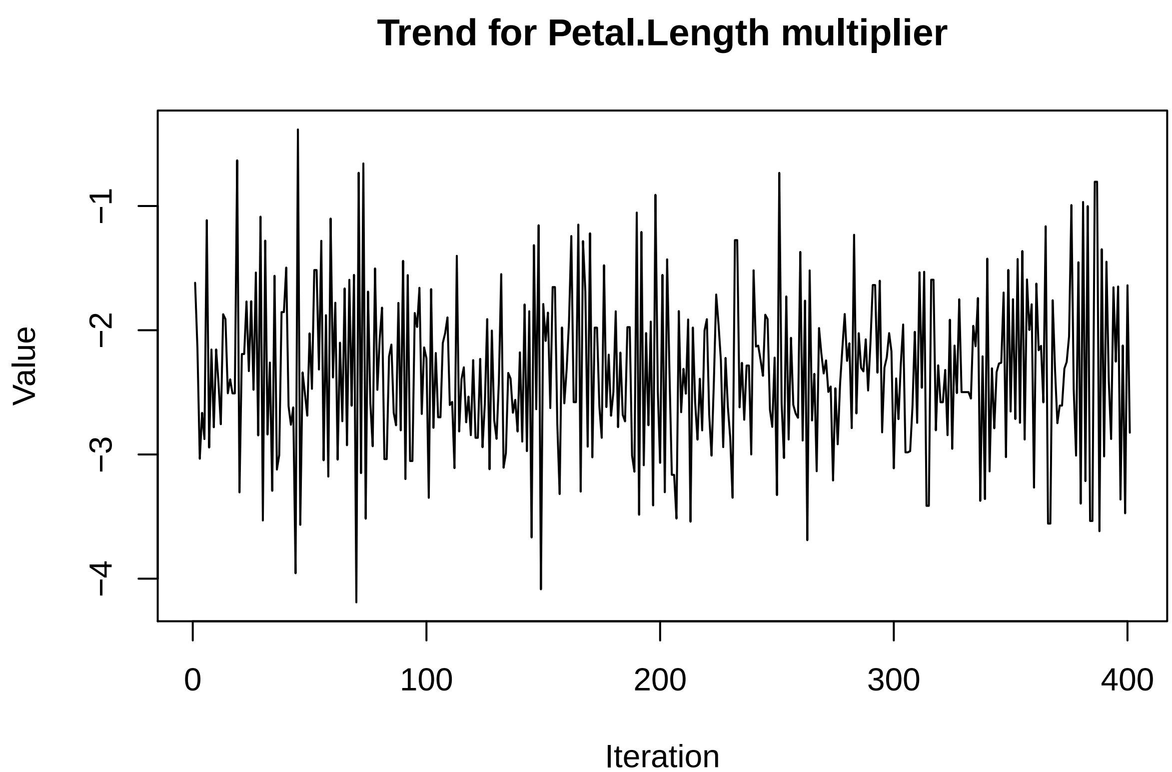

Histogram for bias multiplier
Density
0.0
0.1
0.2
0.3
0.4
−2
−1
0
1
2
3
4
bias multiplier

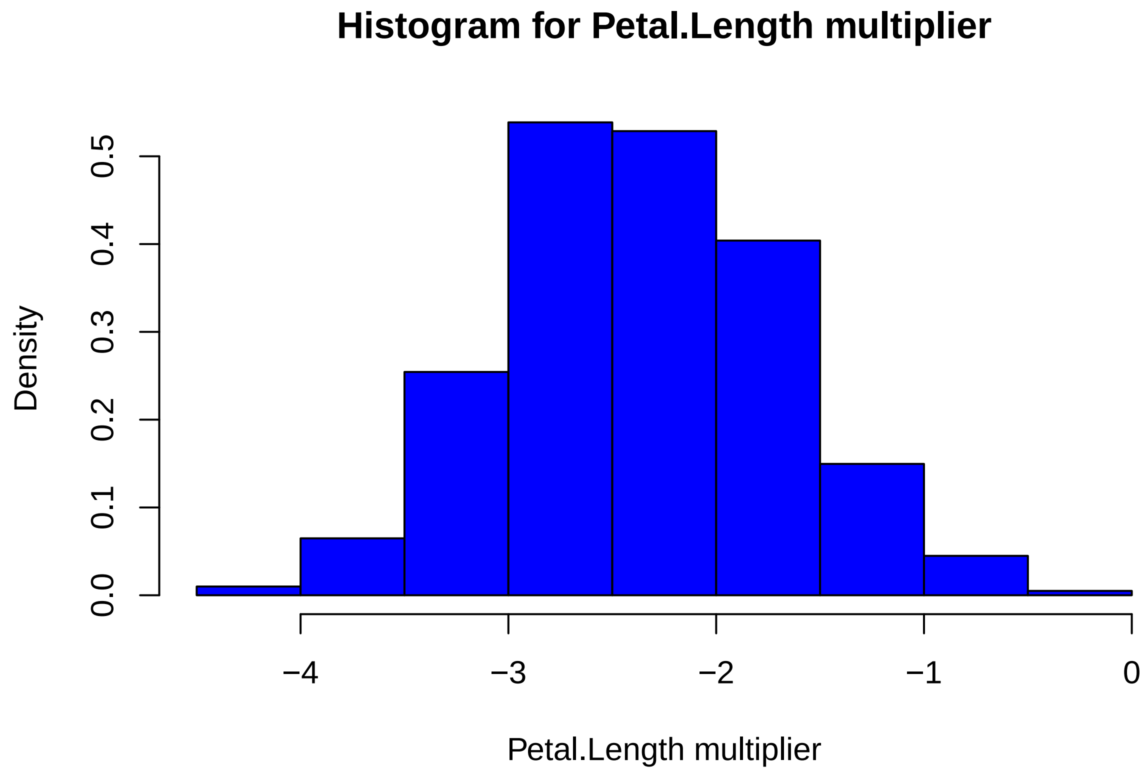

Trend for Sepal.Length multiplier
Value
0.0
1.0
2.0
3.0
0
100
200
300
400
Iteration

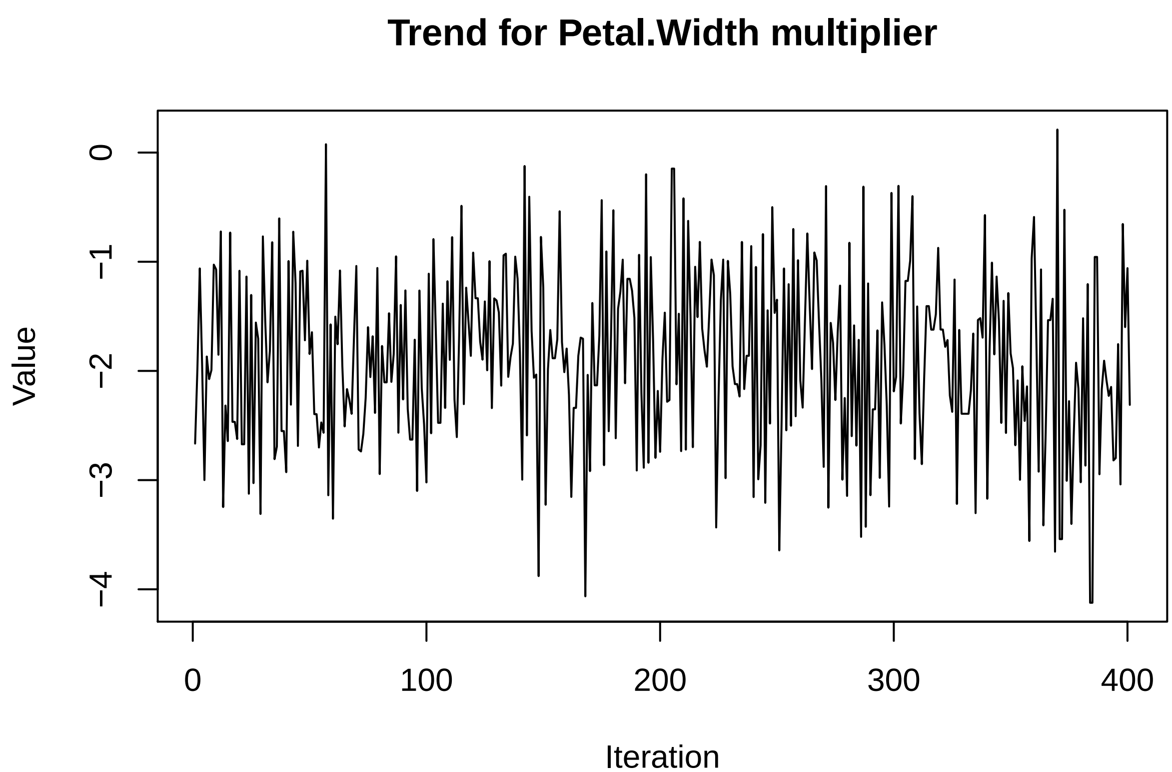

Histogram for Sepal.Length multiplier
Density
0.0
0.2
0.4
0.6
0
1
2
3
Sepal.Length multiplier

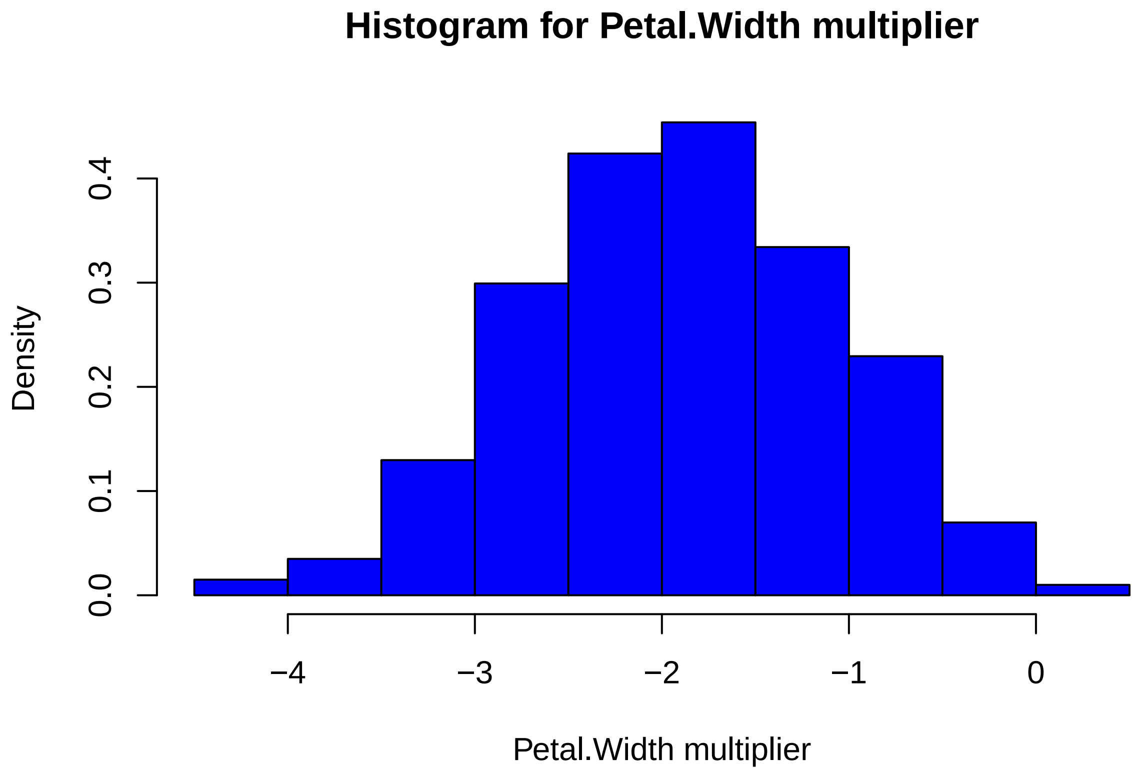

Trend for Sepal.Width multiplier
Value
0
1
2
3
4
0
100
200
300
400
Iteration


Histogram for Sepal.Width multiplier
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−1
0
1
2
3
4
Sepal.Width multiplier


Trend for Petal.Length multiplier
Value
−4
−3
−2
−1
0
100
200
300
400
Iteration


Histogram for Petal.Length multiplier
Density
0.0
0.1
0.2
0.3
0.4
0.5
−4
−3
−2
−1
0
Petal.Length multiplier


Trend for Petal.Width multiplier
Value
0
−1
−2
−3
−4
0
100
200
300
400
Iteration



2(f) A variance σ^2 value of 1 was used, with a step size ϵ of 0.04 and number of steps L of 40.

```
## [1] "The zero-one loss was 3"
```