STAT 578 (Spring 2020) HW1 Solution

1.

(a) posterior distribution of p_1: Beta(426, 76);

posterior distribution of p_2: Beta(10, 2)

(b) For p_1:

- posterior mean: $\frac{426}{426+76} = 0.849$
- posterior median: 0.849

```r
qbeta(0.5, 426, 76)
```

```
## [1] 0.8490687
```

- posterior mode: $\frac{426-1}{426+76-2} = 0.85$

For p_2:

- posterior mean: $\frac{10}{10+2} = 0.833$
- posterior median: 0.852

```r
qbeta(0.5, 10, 2)
```

```
## [1] 0.8520366
```

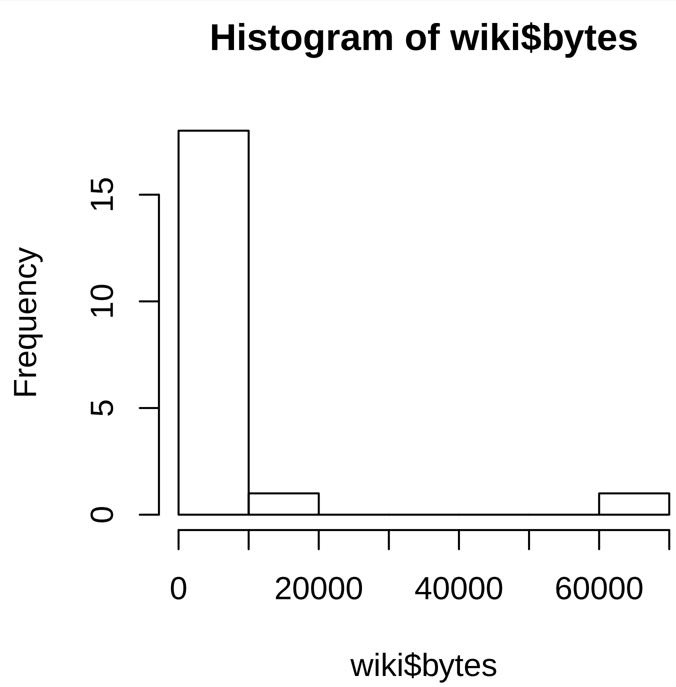
- posterior mode: $\frac{10-1}{10+2-2} = 0.9$

Movie 1 has a higher posterior mean, Movie 2 has higher posterior median and posterior mode.

2.

(a) (i)

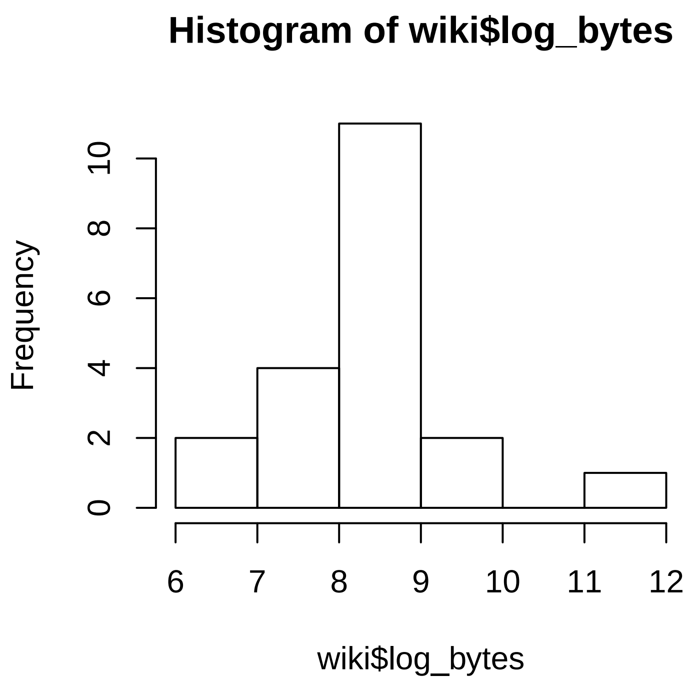
```r
wiki = read.table("~/UIUC/STAT578_20Spring/HW1/randomwikipedia.txt")
hist(wiki$bytes)
```



The distribution of article length is skewed right. Most of the articles have size less than 20000 bytes, while the largest article is over 60000 bytes.

(ii)

```r
wiki$log_bytes = log(wiki$bytes)
hist(wiki$log_bytes)
```



The right skewness is alleviated and the distribution is closer to normal.

(iii) The log scale is better if we want to assume sampling distribution to be normal distribution which is symmetric.

(b)

- sample mean: 8.331
- sample variance: 0.960

```r
(ybar <- mean(wiki$log_bytes))
```

```
## [1] 8.331264
```

```r
(s.2 <- var(wiki$log_bytes))
```

```
## [1] 0.9600394
```

(c) (i)

- Posterior Mean: 8.331
- Posterior Variance: 0.048
- Posterior Precision: 20.832

```r
n <- nrow(wiki)

# Posterior mean
(mun <- ybar)
```

```
## [1] 8.331264
```

```r
# Posterior variance
(tau.2.n <- s.2/n)
```

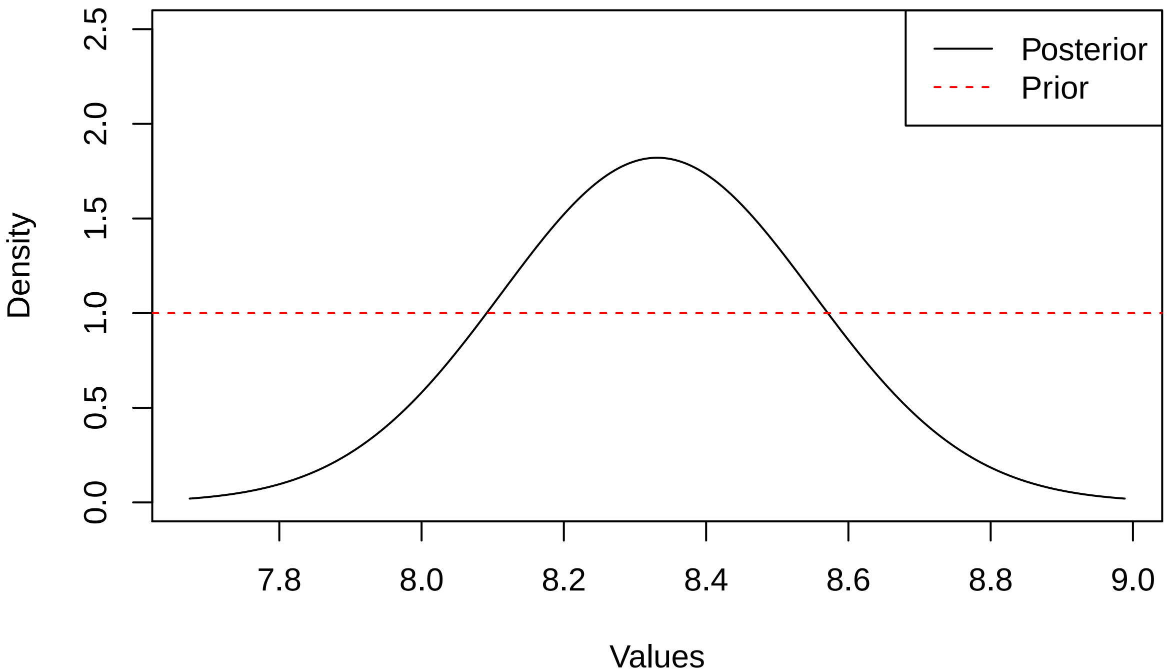
```
## [1] 0.04800197
```

```r
# Posterior precision
n/s.2
```

```
## [1] 20.83248
```

(ii)

```r
xlim = mun + c(-3,3) * sqrt(tau.2.n)
plot(NULL, NULL, xlim=xlim, ylim=c(0,2.5), ylab="Density", xlab='Values')
# posterior
curve(dnorm(x,mun,sqrt(tau.2.n)), add=T, n=1000)
# prior
abline(h=1, lty=2, col=2)
legend('topright', legend=c('Posterior', 'Prior'), lty=1:2, col=1:2)
```



(iii) 95% central posterior interval for μ: (7.902, 8.761)

```r
mun + c(-1.96, 1.96) * sqrt(tau.2.n)
```

```
## [1] 7.901841 8.760687
```

(d) (i) Use simulation to approximate the posterior distribution of μ. 1000 samples were generated.

- Posterior Mean: 8.335
- Posterior Variance: 0.055
- Posterior Precision: 18.223

```r
set.seed(578)
post.sigma.2.sim <- (n-1) * s.2 / rchisq(1000, n-1)
post.mu.sim <- rnorm(1000, ybar, sqrt(post.sigma.2.sim / n))
```

```r
# Posterior mean
mean(post.mu.sim)
```

```
## [1] 8.334605
```

```r
# Posterior variance
var(post.mu.sim)
```

```
## [1] 0.05487582
```

```r
# Posterior precision
1/var(post.mu.sim)
```

```
## [1] 18.22296
```

(ii) Approximated 95% central posterior interval for μ from simulation: (7.873, 8.795)

```r
quantile(post.mu.sim, c(0.05/2, 1-0.05/2))
```

```
##     2.5%    97.5%
## 7.873011 8.795443
```

(iii) Approximated 95% central posterior interval for σ^2 from simulation: (0.534, 2.116)

```r
quantile(post.sigma.2.sim, c(0.05/2, 1-0.05/2))
```

```
##     2.5%    97.5%
## 0.534442 2.115506
```

(e) (i) Approximated 95% central posterior predictive interval: (506, 33820)

```r
set.seed(578)
post.sigma.2.sim <- (n-1) * s.2 / rchisq(1e6, n-1)
post.mu.sim <- rnorm(1e6, ybar, sqrt(post.sigma.2.sim / n))
post.pred.sim <- rnorm(1e6, post.mu.sim, sqrt(post.sigma.2.sim))
exp(quantile(post.pred.sim, c(0.05/2, 1-0.05/2)))
```

```
##       2.5%      97.5%
##   505.9665 33819.8278
```

(ii) Approximated posterior predictive probability: 0.007

```r
mean(post.pred.sim > max(wiki$log_bytes))
```

```
## [1] 0.006796
```

(iii) Approximated posterior predictive probability: 0.114

```r
set.seed(578)
max_log_bytes = max(wiki$log_bytes)
cnt = 0
for (i in 1:1e6) {
  if (max(rnorm(20, post.mu.sim[i], sqrt(post.sigma.2.sim[i]))) > max_log_bytes) {
    cnt = cnt + 1
  }
}
cnt / 1e6
```

```
## [1] 0.113609
```