Exercise Set 10-1

1) To begin, let's set some parameters and simulate data we'll use in parts (a-c).

```
n <- 20
true.mean <- 2
known.sd <- 1
prior.mean <- 0
prior.sd <- 1

set.seed(8675309)
z <- rnorm(n,true.mean,known.sd)
```

a) The mean of my sample is 2.15, fairly close to the true expectation of 2. Here are functions to compute the posterior mean and variance using equations 10.4 and 10.5:

```
post.conj.norm.norm <- function(z, known.sd, prior.mean,
prior.sd){
  xbar <- mean(z)
  post.expec <- (prior.mean / prior.sd^2 + xbar*length(z) /
        known.sd^2)/(1 /   prior.sd^2 + length(z) / known.sd^2)
  post.var <- 1 / (1 /   prior.sd^2 + length(z) / known.sd^2)
  list("posterior.expectation" = post.expec,
"posterior.variance" = post.var)
}
```

With my simulated data, using it gives

```
> post.conj.norm.norm(z, known.sd, prior.mean, prior.sd)
$posterior.expectation
[1] 2.046966

$posterior.variance
[1] 0.04761905
```

The posterior standard deviation is the square root of the variance, approximately 0.218 here.

b) After installing and loading the `MCMCpack` package, use

```
mn.mod <- MCnormalnormal(z, sigma2 = 1, mu0 = prior.mean, tau20
= prior.sd^2, mc = 10000)
```

Calling `summary(mn.mod)` reveals results extremely similar to part (a).

c) Using the `reject.samp.norm()` function from problem 2(d) gives, with my simulated data,

```
> rsamps <- reject.samp.norm(z, known.sd, prior.mean, prior.sd)
> mean(rsamps)
[1] 2.04764
> sd(rsamps)
[1] 0.2194684
```

These results are extremely similar to those obtained in parts (a) and (b). The small differences are due to stochasticity inherent in MCMC and rejection sampling, which are Monte Carlo procedures.

2) a) The ratio is equal to the likelihood, $f_D(d|\theta)f_\theta(\theta)/f_\theta(\theta) = f_D(d|\theta) = L(\theta)$.

b) If c times the unscaled posterior is equal to the prior, then

$$c\mathcal{L}(\theta)f_\theta(\theta) = f_\theta(\theta),$$

which implies $c = 1/L(\theta)$. Similarly, if c times the unscaled posterior is less than the prior, then $c < 1/L(\theta)$. If we set $c = 1/\max\big(L(\theta)\big)$, then $cL(\theta)f_\theta(\theta) = f_\theta(\theta)$ for the value(s) of θ that maximize the likelihood, and $cL(\theta)f_\theta(\theta) < f_\theta(\theta)$ for all other values of θ.

c) The only step that changes from the algorithm given in the text is the second one. Instead of computing m as the likelihood, we can compute m as the likelihood divided by the maximum possible value of the likelihood, which guarantees that m is between 0 and 1 but also that it can, in least in principle, be as large as 1. This usually means that a much larger proportion of the samples can be accepted, which increases the efficiency of the algorithm.

d) The below two functions are one way (of many) to do it. Rather than multiplying the densities associated with each datum in the input vector `z`, we take the logs of the densities, sum them, and then exponentiate the result. This is useful because `R` (and other programs) can lose precision when forced to perform arithmetic with extremely small numbers—using the logs is a trick to get around the numerical instability that can result.

```
#Get 1 sample under rejection sampling from normal with known sd.
#z is a vector of data.
get.1.samp.norm <- function(z, known.sd = 1, prior.mn = 0, prior.sd =
1){
  accepted <- FALSE
  max.like <- exp(sum(log(dnorm(z, mean = mean(z), sd = known.sd))))
  while(accepted == FALSE){
    cand <- rnorm(1, prior.mn, prior.sd)
    like <- exp(sum(log(dnorm(z, mean = cand, sd = known.sd))))
    crit <- like / max.like
    xunif <- runif(1,0,1)
    if(xunif <= crit){accepted <- TRUE}
  }
  cand
}
```

```
#Wrapper for get.1.samp.norm() that gets rejection sample from
posterior of desired size.
reject.samp.norm <- function(z, known.sd = 1, prior.mn = 0, prior.sd =
1, nsamps = 10000){
  samps <- numeric(nsamps)
  for(i in seq_along(samps)){
    samps[i] <- get.1.samp.norm(z, known.sd, prior.mn, prior.sd)
  }
  samps
}
```

Exercise Set 10-2

1) a) Obtain the least-squares estimates of the intercept and slope—3 and ½, respectively—with

```
y <- anscombe$y1
x <- anscombe$x1
reg.ml <- lm(y~x)
summary(reg.ml)
```

b) Code to fit the model with all 9 possible priors is:

```
reg11 <- MCMCregress(y ~ x, b0 = c(0,0), B0 = 0.0001)
reg12 <- MCMCregress(y ~ x, b0 = c(0,0), B0 = 1)
reg13 <- MCMCregress(y ~ x, b0 = c(0,0), B0 = 100)
reg21 <- MCMCregress(y ~ x, b0 = c(3,0), B0 = 0.0001)
reg22 <- MCMCregress(y ~ x, b0 = c(3,0), B0 = 1)
reg23 <- MCMCregress(y ~ x, b0 = c(3,0), B0 = 100)
reg31 <- MCMCregress(y ~ x, b0 = c(10,-5), B0 = 0.0001)
reg32 <- MCMCregress(y ~ x, b0 = c(10,-5), B0 = 1)
reg33 <- MCMCregress(y ~ x, b0 = c(10,-5), B0 = 100)
```

When prior precision is low—meaning that prior variance is high—then the prior means are not especially important; these three choices lead to similar conclusions. If the prior precision is higher, then the prior means matter much more, and estimates are generally close to the prior means.

2) a) Squared-error loss implies that the expected loss given a particular choice of θ_0 is $\int_{-\infty}^{\infty}(\theta - \theta_0)^2 f_\theta(\theta)d\theta$. Expanding the square term inside the integral gives $\int_{-\infty}^{\infty}(\theta^2 - 2\theta\theta_0 + {\theta_0}^2)f_\theta(\theta)d\theta$. Splitting this into three integrals by distributing the $f_\theta(\theta)d\theta$ term and then applying the definition of expectation and the fact that density functions integrate to 1 gives

$$\int_{-\infty}^{\infty} \theta^2 f_{\theta|D}(\theta)d\theta - 2\theta_0 \int_{-\infty}^{\infty} \theta f_{\theta|D}(\theta)d\theta + {\theta_0}^2 \int_{-\infty}^{\infty} f_{\theta|D}(\theta)d\theta = \mathrm{E}(\theta^2|D) - 2\theta_0\mathrm{E}(\theta|D) + {\theta_0}^2$$

To minimize the loss, $\mathrm{E}(\theta^2|D) - 2\theta_0\mathrm{E}(\theta|D) + {\theta_0}^2$, we take the derivative with respect to θ_0, giving

$$-2\mathrm{E}(\theta|D) + 2\theta_0.$$

Setting this derivative equal to 0 gives

$$\theta_0 = \mathrm{E}(\theta|D),$$

which is the value of θ_0 that minimizes the expectation of the loss function, and thus the Bayes estimator.

b) Under 0-1 loss, if we choose θ_i, then the loss is 0 if $\theta_i = \theta$ and the loss is 1 if $\theta_i \neq \theta$. The expected loss if we choose θ_i, given the data, is thus

$$1 * \mathrm{P}(\theta_i \neq \theta|D = d) + 0 * \mathrm{P}(\theta_i = \theta|D = d) = 1 - \mathrm{P}(\theta_i = \theta|D = d).$$

Minimizing the expected loss is equivalent to maximizing the posterior probability $\mathrm{P}(\theta_i = \theta|D = d)$. By definition, the θ_i with the highest posterior probability is the posterior mode, and so the posterior mode minimizes the expected loss.

c) Under absolute-error loss, the expected loss is $\int_{-\infty}^{\infty} |\theta - \theta_0| f_{\theta|D}(\theta) d\theta$. To minimize the expected loss, we want to set

$$\frac{\partial}{\partial\theta_0} \int_{-\infty}^{\infty} |\theta - \theta_0| f_{\theta|D}(\theta) d\theta = 0.$$

It is hard to make progress in this form because the function $|\theta - \theta_0|$ has an undifferentiable point at $\theta = \theta_0$. We will respond by breaking the integral into two pieces,

$$\frac{\partial}{\partial\theta_0}\left[\int_{-\infty}^{\theta_0} (\theta_0 - \theta) f_{\theta|D}(\theta) d\theta + \int_{\theta_0}^{\infty} (\theta - \theta_0) f_{\theta|D}(\theta) d\theta\right].$$

The Leibniz integral rule holds that we can change the order of integration and differentiation, using

$$\frac{\partial}{\partial t}\left(\int_{a(t)}^{b(t)} g(x,t) dx\right) = \int_{a(t)}^{b(t)} \frac{\partial g(x,t)}{\partial t} dx + g(b(t),t)b'(t) - g(a(t),t)a'(t).$$

Replacing t with θ_0, $b(t)$ with $b(\theta_0) = \theta_0$, $a(t)$ with $a(\theta_0) = -\infty$, x with θ, and $g(x,t)$ with $g(\theta,\theta_0) = (\theta_0 - \theta) f_{\theta|D}(\theta)$ gives

$$\frac{\partial}{\partial\theta_0}\int_{-\infty}^{\theta_0}(\theta_0-\theta)f_{\theta|D}(\theta)d\theta$$

$$= \int_{-\infty}^{\theta_0} f_{\theta|D}(\theta)d\theta + (\theta_0-\theta_0)f_{\theta|D}(\theta_0) * 1 - \lim_{\theta\to-\infty}(\theta_0-\theta)f_{\theta|D}(\theta) * 0$$

$$= \int_{-\infty}^{\theta_0} f_{\theta|D}(\theta)d\theta = \mathrm{P}(\theta < \theta_0|D).$$

Similarly, replacing t with θ_0, $b(t)$ with $b(\theta_0) = \infty$, $a(t)$ with $a(\theta_0) = \theta_0$, x with θ, and $g(x,t)$ with $g(\theta,\theta_0) = (\theta-\theta_0)f_{\theta|D}(\theta)$ gives

$$\frac{\partial}{\partial\theta_0}\int_{\theta_0}^{\infty}(\theta-\theta_0)f_{\theta|D}(\theta)d\theta$$

$$= -\int_{\theta_0}^{\infty} f_{\theta|D}(\theta)d\theta + \lim_{\theta\to\infty}(\theta-\theta_0)f_{\theta|D}(\theta) * 0 - (\theta_0-\theta_0)f_{\theta|D}(\theta_0) * 1$$

$$= -\int_{\theta_0}^{\infty} f_{\theta|D}(\theta)d\theta = -\mathrm{P}(\theta > \theta_0|D)$$

Summing these two derivatives, the derivative of the expected absolute error loss with respect to θ_0 is

$$\frac{\partial}{\partial\theta_0}\int_{-\infty}^{\infty}|\theta-\theta_0|f_\theta(\theta)d\theta = \mathrm{P}(\theta < \theta_0|D) - \mathrm{P}(\theta > \theta_0|D).$$

Setting the derivative equal to zero gives

$$\mathrm{P}(\theta < \theta_0|D) = \mathrm{P}(\theta > \theta_0|D),$$

which, by definition of the median, is satisfied when θ_0 is the median of the posterior distribution.

Exercise Set 10-3

1) Assuming you have already fit the models in the previous exercise set, problem 1, you can get the quantile interval by looking at the 2.5th and 97.5th quantiles of the slope parameter in the summary output. For these models, the quantile and highest-posterior-density intervals are similar. When the prior precision is very low (i.e. its variance is very high), the credible interval largely agrees with the frequentist confidence intervals you have already calculated. If the precision is higher, then the credible interval is pulled toward the prior mean.

Exercise Set 10-4

1) a) Here is the code:

```
reg0 <- MCMCregress(y~1, b0 = 0, B0 = 1/100, c0 = 0.001, d0 =
0.001, marginal.likelihood = "Laplace")
reg1 <- MCMCregress(y~x, b0 = 0, B0 = 1/100, c0 = 0.001, d0 =
0.001, marginal.likelihood = "Laplace")
summary(reg0)
summary(reg1)
summary(BayesFactor(reg1, reg0))
```

Some relevant outputs are:
Intercept under H_1: posterior mean of 2.97, 95% credible interval [0.48, 5.52]
Slope under H_1: posterior mean of 0.50, 95% credible interval [0.23, 0.77]
Bayes factor B_{10}: 3.26. By Kass & Raftery's scale, this is positive evidence for H_1 over H_0.

b) Here is the code:

```
reg01 <- MCMCregress(y~1, b0 = 0, B0 = 1/16, c0 = 0.001, d0 =
0.001, marginal.likelihood = "Laplace")
reg11 <- MCMCregress(y~x, b0 = 0, B0 = 1/16, c0 = 0.001, d0 =
0.001, marginal.likelihood = "Laplace")
summary(reg01)
summary(reg11)
summary(BayesFactor(reg11, reg01))
```

Some relevant outputs are:
Intercept under H_1: posterior mean of 2.76, 95% credible interval [0.33, 5.13]
Slope under H_1: posterior mean of 0.52, 95% credible interval [0.27, 0.78]
Bayes factor B_{10}: 26.9. By Kass & Raftery's scale, this is strong evidence for H_1 over H_0.

c) Here is the code:

```
reg02 <- MCMCregress(y~1, b0 = 0, B0 = 1/10000, c0 = 0.001, d0 =
0.001, marginal.likelihood = "Laplace")
reg12 <- MCMCregress(y~x, b0 = 0, B0 = 1/10000, c0 = 0.001, d0 =
0.001, marginal.likelihood = "Laplace")
summary(reg02)
summary(reg12)
summary(BayesFactor(reg12, reg02))
```

Some relevant outputs are:
Intercept under H_1: posterior mean of 3.01, 95% credible interval [0.51, 5.61]
Slope under H_1: posterior mean of 0.50, 95% credible interval [0.22, 0.76]
Bayes factor B_{10}: 0.26. By Kass & Raftery's scale, this is positive evidence for H_0 over H_1—notice that we have switched from having the data support H_1 to having them support H_0.

d) In this case, changing the prior precision/variance had only a small effect on the point estimates and credible intervals obtained. In contrast, the Bayes factors changed in consequential ways. With intermediate prior variance, we had relatively weak but "positive" support for the model with the slope included. After decreasing the prior variance, that support became much stronger. But increasing the prior variance causes it to reverse, and the intercept-only model becomes supported over the model with the slope included. One has to be careful about prior specification when working with Bayes factors.