homework3

PSTAT 115, Spring 2021

Due on May 23, 2021 at 11:59 pm

Problem 1. Rejection Sampling the Beta distribution. (15 pts)

Assume we did not have access to the rbeta function for sampling from a Beta, but we were able to evaluate the density, dbeta. This is a very common setting in Bayesian statistics, since we can always evaluate the (proportional) posterior density $p(\theta \mid y) \propto p(y \mid \theta)p(\theta)$ but we don't have immediate access to a method for sampling from this distribution.

1. Let p(x) be a Beta(3, 9) density, $q_1(x)$ a Uniform(0, 1) density, and $q_2(x)$ a Normal($\mu = 0.25, \sigma = 0.15$) density.

```r
p <- function(x) { dbeta(x, 3, 9) }
q_1 <- function(x) { dunif(x, 0, 1) }
q_2 <- function(x) { dnorm(x, 0.25, 0.15) }
```

1. Use rejection sampling to sample from p(x) by proposing samples from $q_1(x)$. To do so, first find $M_1 = \max_x p(x)/q_1(x)$ using the optimize function and set lower=0, upper=1, and maximum = TRUE (since we are maximizing not minimizing, the default). M will be the value in the objective argument returned by optimize (maximum tells us where the maximum occurs, but not what height it achieves). Propose 10000 samples and keep only the accepted samples.

```r
density_ratio <- function(x) { p(x) / q_1(x) }
M_1 <- optimize(density_ratio, lower = 0, upper = 1, maximum = TRUE)$objective
M_1
```

```
## [1] 3.321889
```

```r
n <- 10000
q_1_samp <- runif(n, 0, 1)
accept <- runif(n) < density_ratio(q_1_samp) / M_1
q_1_samp2 <- q_1_samp[accept]
```

1. Use rejection sampling to sample from p(x) by proposing samples from $q_2(x)$. To do this you need to find $M_2 = \max_x p(x)/q_2(x)$ as above. Propose 10000 samples and keep only the accepted samples.

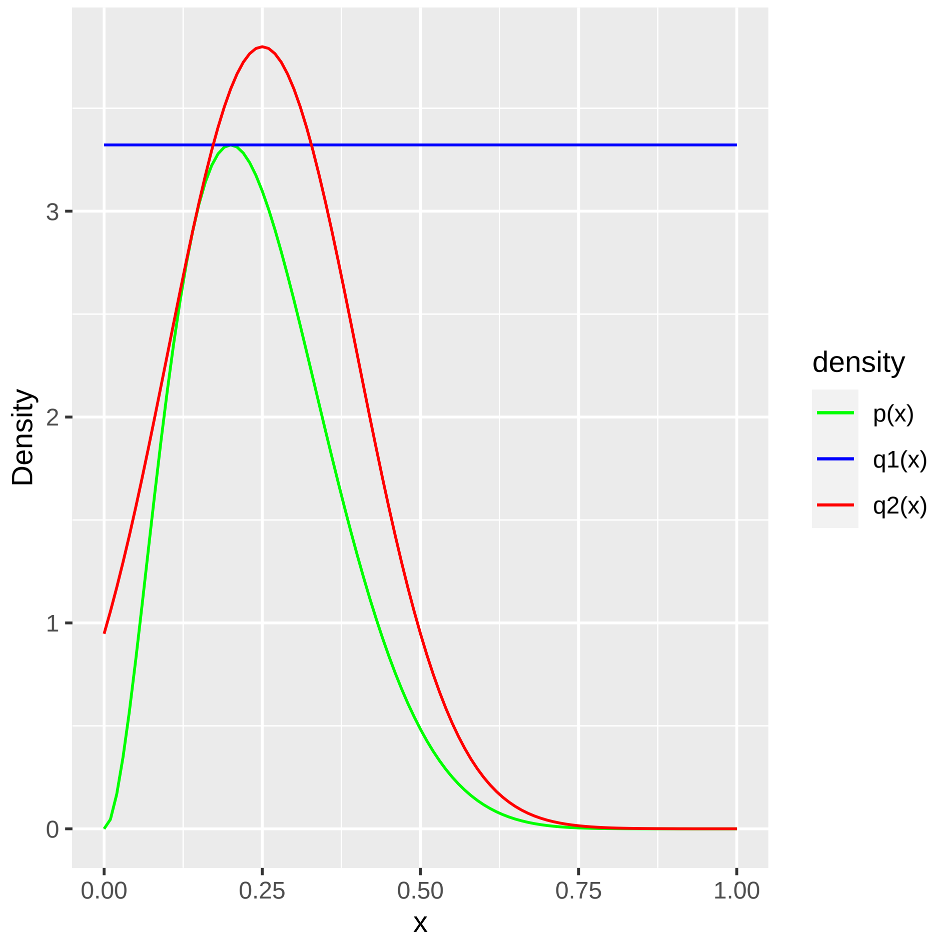
```r
density_ratio_2 <- function(x) { p(x) / q_2(x) }
M_2 <- optimize(density_ratio_2, lower = 0, upper = 1, maximum = TRUE)$objective
M_2
```

```
## [1] 1.428353
```

```r
q_2_samp <- rnorm(n, 0.25, 0.15)
accept2 <- runif(n) < density_ratio_2(q_2_samp) / M_2
q_2_samp2 <- q_2_samp[accept2]
```

1. Plot the p(x), $M_1 q_1(x)$ and $M_2 q_2(x)$ all on the same plot and verify visually that the scaled proposal densities "envelope" the target, p(x). Set the xlimits of the plot from 0 to 1. Use different color lines for the various densities so are clearly distinguishable. (5 pts)

```r
M1_q1 <- function(x) { M_1 * q_1(x) }
M2_q2 <- function(x) { M_2 * q_2(x) }

ggplot(aes(x=x), data = data.frame(x=0)) +
  stat_function(fun = p, aes(colour = "p(x)")) +
  stat_function(fun = M1_q1, aes(colour = "q1(x)")) +
  stat_function(fun = M2_q2, aes(colour = "q2(x)")) +
  xlim(c(0,1))+scale_y_continuous(name = "Density") +
  scale_colour_manual(name="density",values = c("p(x)" = "green",
                                                "q1(x)" = "blue",
                                                "q2(x)" = "red"))
```



1. Which rejection sampler had the higher rejection rate? Why does this make sense given the plot from the previous part? This means when proposing 10000 samples from each proposal, the Monte Carlo error of our approximation will be higher when proposing from ______ (choose q_1 or q_2). (5 pts)

From the plot above, we can see that the rejection sampler, q1(x), would have the higher rejection rate. This makes sense because in comparison to the rejection sampler, q2(x), q1(x) "envelopes" the target p(x) better, since the area of rejection within q2(x) is smaller than the area of rejection in q1(x). This means when proposing 10000 samples from each proposal, the Monte Carlo error from our approximation will be higher when proposing from q1 .

1. Report the variance of Beta(3, 9) distribution by computing the variance of the beta samples. How does this compare to the theoretical variance (refer to the probability cheatsheet). (5 pts)

```r
alpha <- 3
beta <- 9
beta_var <- (alpha * beta) / ((alpha + beta)^2 * (alpha + beta + 1))
beta_var
```

```
## [1] 0.01442308
```

```
var(q_1_samp)
```

```
## [1] 0.08282204
```

```
var(q_2_samp)
```

```
## [1] 0.02251753
```

Overall, we can see that the variance of Beta(3, 9) distribution is smaller than the theoretical variance.

Problem 2. Frequentist Coverage of The Bayesian Posterior Interval. (35 pts)

Suppose that $y_1, .., y_n$ is an IID sample from a $Normal(\mu, 1)$. We wish to estimate μ.

2a. For Bayesian inference, we will assume the prior distribution $\mu \sim Normal(0, \frac{1}{\kappa_0})$ for all parts below. Remember, from lecture that we can interpret κ_0 as the pseudo-number of prior observations with sample mean $\mu_0 = 0$. State the posterior distribution of μ given $y_1, .., y_n$. Report the lower and upper bounds of the 95% quantile-based posterior credible interval for μ, using the fact that for a normal distribution with standard eviation σ, approximately 95% of the mass is between $\pm 1.96\sigma$. (5 pts)

In this scenario we are in a conjugate model. This means that the posterior distribution is also the normal distribution. This is due to the fact that we know that our sampling distribution and prior distribution are both normally distributed as well.

The posterior distribution is $N(\mu_0, \tau_0^2)$. In our conjugate model the parameters $\mu_0 = \frac{\frac{1}{\tau_0^2}\mu_0 + \frac{n}{\sigma^2}\bar{y}}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}}$, and $\tau_0^2 = \frac{1}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}}$. In order to calculate the 95% interval, we will need to plug in values to get τ_0^2 in terms of κ_0 and n.

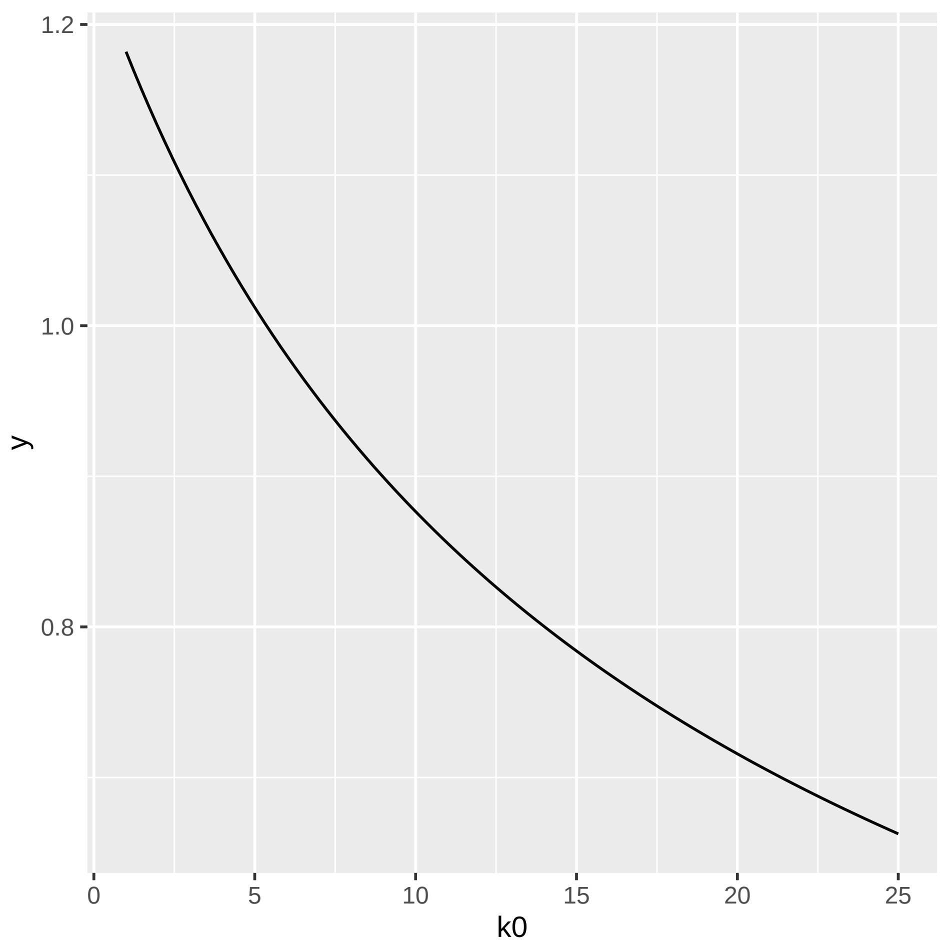
We know that our sampling variance is 1 and the prior variance is $\frac{1}{\kappa_0}$. We also know that our prior mean is 0, thus we can plug in these values to get our interval.

The upper and lower bounds of the 95 quantile-based posterior credible interval for μ are …

Upper $= +1.96\sqrt{\frac{1}{n+\kappa_0}}$ Lower $= -1.96\sqrt{\frac{1}{n+\kappa_0}}$

2b. Plot the length of the posterior credible interval as a function of κ_0, for $\kappa_0 = 1, 2, ..., 25$ assuming $n = 10$. Report how this prior parameter effects the length of the posterior interval and why this makes intuitive sense. (10 pts)

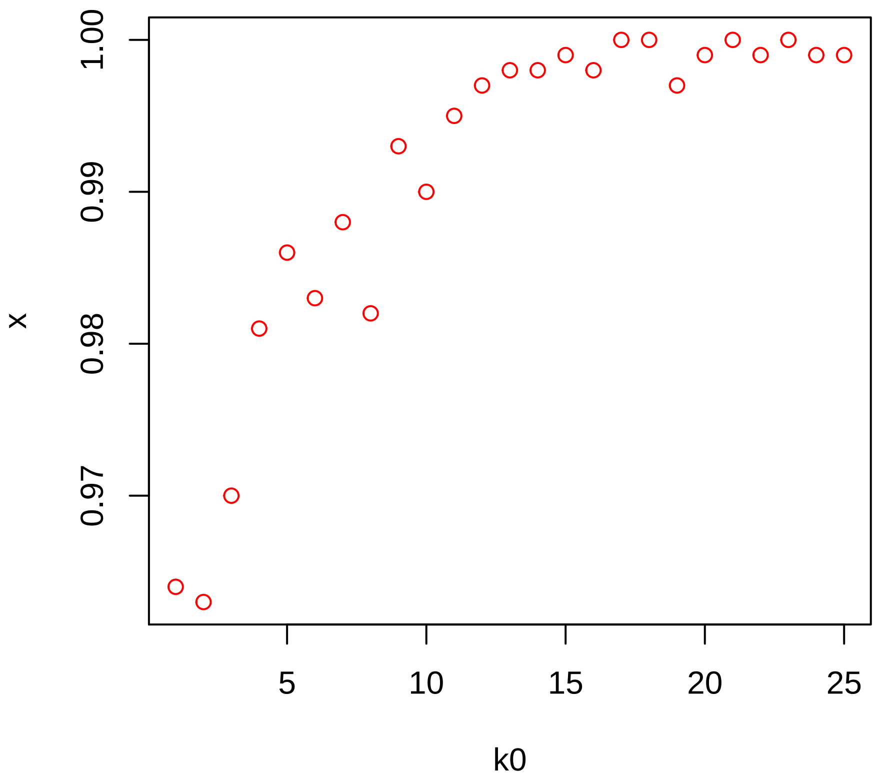
```
# YOUR CODE HERE
n<-10
k0 <- seq(1,25)

len_func <- function(k0){
  2*1.96*sqrt(1/(n+k0))
}

plot1 <- ggplot(data.frame(k0), aes(k0)) +
  stat_function(fun = len_func, geom='line')

plot1
```



From the plot we can conclude that as we get more samples and k0 increases, the variance for the estimate of mu decreases as well. I think this suggests that as we get more samples and kappa increases, the variance/length decreases, so our confidence in our estimate increases.

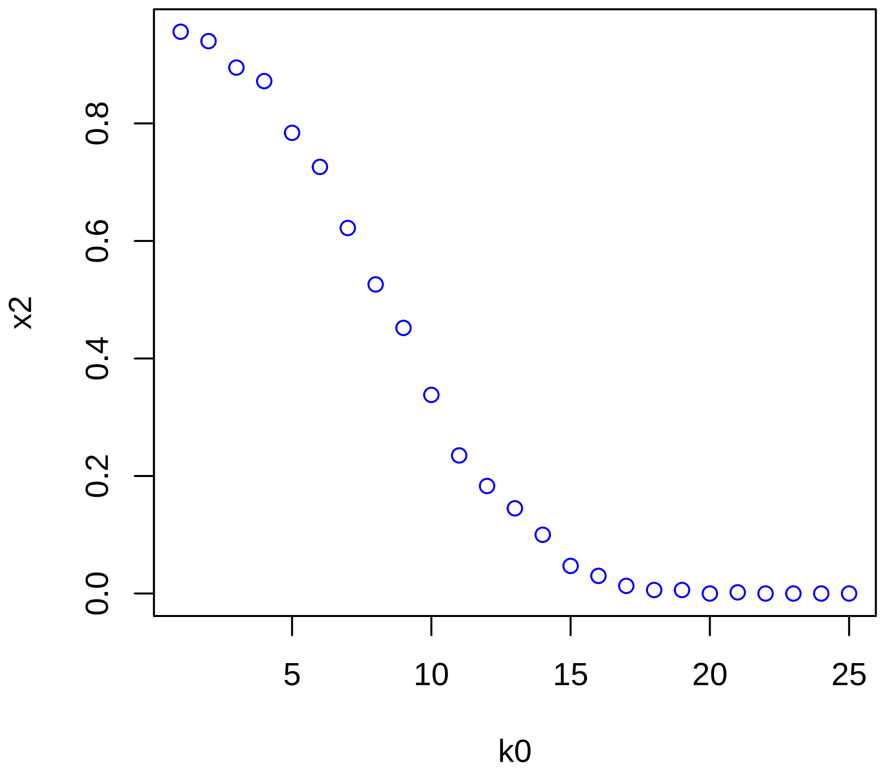
2c. Now we will evaluate the *frequentist coverage* of the posterior credible interval on simulated data. Generate 1000 data sets where the true value of $\mu = 0$ and $n = 10$. For each dataset, compute the posterior 95% interval endpoints (from the previous part) and see if it the interval covers the true value of $\mu = 0$. Compute the frequentist coverage as the fraction of these 1000 posterior 95% credible intervals that contain $\mu = 0$. Do this for each value of $\kappa_0 = 1, 2, ..., 25$. Plot the coverage as a function of κ_0. (5 pts)

```r
# YOUR CODE HERE
mu <- 0
generations <- 1000
dataset <- matrix(0,generations,length(k0))
x <- rep(0,length(k0))

for (k in k0) {
    for (data in seq(generations)) {
        y <- rnorm(n,mu,1)
        posterior_mu <- (mean(y)*n/(k+n))
        cred_interval <- qnorm(c(0.025,0.975),posterior_mu,sqrt(1/(k+n)))
        if(between(mu,cred_interval[1],cred_interval[2])==TRUE){
            dataset[data,k] <- 1
        }
    }
    x[k] <- sum(dataset[,k])/generations
}

plot(k0,x,col= "red")
```



2d. Repeat the 1c but now generate data assuming the true $\mu = 1$. (5 pts)

```
# YOUR CODE HERE
mu2 <- 1
generations <- 1000
dataset <- matrix(0,generations,length(k0))
x2 <- rep(0,length(k0))

for (k in k0) {
    for (data in seq(generations)) {
        y <- rnorm(n,mu2,1)
        posterior_mu2 <- (mean(y)*n/(k+n))
        cred_int <- qnorm(c(0.025,0.975),posterior_mu2,sqrt(1/(k+n)))
        if(between(mu2,cred_int[1],cred_int[2])==TRUE){
            dataset[data,k] <- 1
        }
    }
    x2[k] <- sum(dataset[,k])/generations
}
#plot
plot(k0, x2, col = "blue")
```



2e. Explain the differences between the coverage plots when the true $\mu = 0$ and the true $\mu = 1$. For what values of κ_0 do you see closer to nominal coverage (i.e. 95%)? For what values does your posterior interval tend to overcover (the interval covers the true value more than 95% of the time)? Undercover (the interval covers the true value less than 95% of the time)? Why does this make sense? (10 pts)

When $\mu = 0$, we see that in the coverage plots, the posterior tends to have higher coverage over k. When the values of k are greater than 2 and equal to 1. The posterior also seems to undercover when k is greater than 2. Furthermore, it seems that as k increases towards 25, the intervals become smaller. Because of this, we can be much more certain of our prior belief that $\mu_0 = 0$.

Problem 3. Bayesian inference for the normal distribution in Stan. (50pts)

Create a new Stan file by selecting "Stan file" in the Rstudio menu. Save it as `IQ_model.stan`. We will make some basic modifications to the template example in the default Stan file for this problem. Consider the IQ example used from class. Scoring on IQ tests is designed to yield a N(100, 15) distribution for the general population. We observe IQ scores for a sample of n individuals from a particular town, $y_1, \dots y_n \sim N(\mu, \sigma^2)$. Our goal is to estimate the population mean in the town. Assume the $p(\mu, \sigma) = p(\mu \mid \sigma)p(\sigma)$, where $p(\mu \mid \sigma)$ is $N(\mu_0, \sigma/\sqrt{\kappa_0})$ and $p(\sigma)$ is Gamma(a, b). Before you administer the IQ test you believe the town is no different than the rest of the population, so you assume a prior mean for μ of $\mu_0 = 100$, but you aren't to sure about this a priori and so you set $\kappa_0 = 1$ (the effective number of pseudo-observations). Similarly, a priori you assume σ has a mean of 15 (to match the intended standard deviation of the IQ test) and so you decide on setting $a = 15$ and $b = 1$ (remember, the mean of a Gamma is a/b). Assume the following IQ scores are observed:

```r
y <- c(70, 85, 111, 111, 115, 120, 123)
n <- length(y)
```

3a. Make a scatter plot of the posterior distribution of the median, μ, and the precision, $1/\sigma^2$. Put μ on the x-axis and $1/\sigma^2$ on the y-axis. What is the posterior relationship between μ and $1/\sigma^2$? Why does this make sense? *Hint:* review the lecture notes. (10pts)

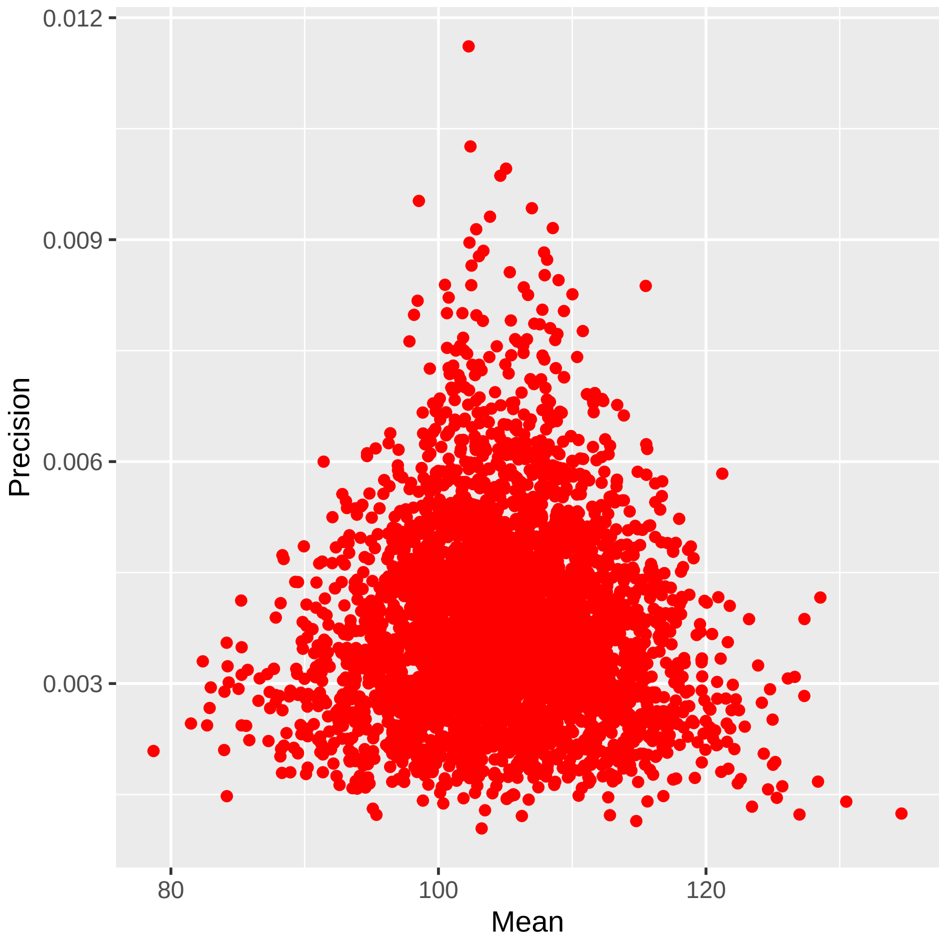
```r
library(rstan)
y <- c(70, 85, 111, 111, 115, 120, 123)
n <- length(y)

# YOUR CODE HERE
stan_model <- stan_model(file = "IQ_model.stan")
```

```
## Trying to compile a simple C file
```

```
## Running /opt/microsoft/ropen/4.0.2/lib64/R/bin/R CMD SHLIB foo.c
## gcc -std=gnu99 -I/opt/microsoft/ropen/4.0.2/lib64/R/include -DNDEBUG   -I"/opt/microsoft/ropen/4.0.2
## In file included from /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Core:88:0,
##                  from /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
##                  from /opt/microsoft/ropen/4.0.2/lib64/R/library/StanHeaders/include/stan/math/prim/
##                  from <command-line>:0:
## /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: err
##  namespace Eigen {
##  ^~~~~~~~~
## /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:17: er
##  namespace Eigen {
##                  ^
## In file included from /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Dense:1:0,
##                  from /opt/microsoft/ropen/4.0.2/lib64/R/library/StanHeaders/include/stan/math/prim/
##                  from <command-line>:0:
## /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Core:96:10: fatal error: complex:
##  #include <complex>
##           ^~~~~~~~~
## compilation terminated.
## /opt/microsoft/ropen/4.0.2/lib64/R/etc/Makeconf:159: recipe for target 'foo.o' failed
## make: *** [foo.o] Error 1
```

```r
a <- 15
b <- 1
k0 <- 1
mu0 <- 100
```

```r
stan_fit <- sampling(stan_model,data=list(N=n,y=y,mu0=mu0,k0=k0,a=a,b=b),refresh=0)
samples <- extract(stan_fit)
samples_mu <- samples$mu
samples_sigma <- samples$sigma
tibble(Mean = samples_mu,Precision=1/samples_sigma^2) %>%
    ggplot()+geom_point(aes(x=Mean,y=Precision), color = 'red')
```



From what I can see from the plot, it seems that when there is high precision (high $1/\sigma^2$), there is is a smaller posterior variability in μ. On the other hand, when there is low precision (high $1/\sigma^2$), there is is a larger posterior variability in μ. This seems to make sense since σ^2 captures information about the spread of the data.

3b. You are interested in whether the mean IQ in the town is greater than the mean IQ in the overall population. Use Stan to find the posterior probability that μ is greater than 100. (20pts)

```r
#library(rstan)
y <- c(70, 85, 111, 111, 115, 120, 123)
n <- length(y)

mean(samples_mu>100)
```

```
## [1] 0.78275
```

3c. You notice that two of the seven scores are significantly lower than the other five. You think that the normal distribution may not be the most appropriate model, in particular because you believe some people in this town are likely have extreme low and extreme high scores. One solution to this is to use a model that is more robust to these kinds of outliers. The Student's t distribution and the Laplace distribution are two so called "heavy-tailed distribution" which have higher probabilities of outliers (i.e. observations further from the mean). Heavy-tailed distributions are useful in modeling because they are more robust to outliers. Fit the model assuming now that the IQ scores in the town have a Laplace distribution, that is $y_1, \ldots, y_n \sim Laplace(\mu, \sigma)$. Create a copy of the previous stan file, and name it "IQ_laplace_model.stan". *Hint:* In the Stan file you can replace `normal` with `double_exponential` in the model section, another name for the Laplce distribution. Like the normal distribution it has two arguments, μ and σ. Keep the same prior distribution, $p(\mu, \sigma)$ as used in the normal model. Under the Laplace model, what is the posterior probability that the median IQ in the town is greater than 100? How does this compare to the probability under the normal model? Why does this make sense? (20pts)

```r
laplace_stan_model <- stan_model("IQ_laplace_model.stan")
```

```
## Trying to compile a simple C file
```

```
## Running /opt/microsoft/ropen/4.0.2/lib64/R/bin/R CMD SHLIB foo.c
## gcc -std=gnu99 -I/opt/microsoft/ropen/4.0.2/lib64/R/include -DNDEBUG   -I"/opt/microsoft/ropen/4.0.2/
## In file included from /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Core:88:0,
##                  from /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
##                  from /opt/microsoft/ropen/4.0.2/lib64/R/library/StanHeaders/include/stan/math/prim/m
##                  from <command-line>:0:
## /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: erro
##  namespace Eigen {
##  ^~~~~~~~~
## /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:17: err
##  namespace Eigen {
##                  ^
## In file included from /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Dense:1:0,
##                  from /opt/microsoft/ropen/4.0.2/lib64/R/library/StanHeaders/include/stan/math/prim/m
##                  from <command-line>:0:
## /opt/microsoft/ropen/4.0.2/lib64/R/library/RcppEigen/include/Eigen/Core:96:10: fatal error: complex: 
##  #include <complex>
##           ^~~~~~~~~
## compilation terminated.
## /opt/microsoft/ropen/4.0.2/lib64/R/etc/Makeconf:159: recipe for target 'foo.o' failed
## make: *** [foo.o] Error 1
```

```r
laplace_stan_fit <- sampling(laplace_stan_model,data=list(N=n,y=y,mu0=mu0,k0=k0,a=a,b=b),refresh=0)
samples_laplace <- extract(laplace_stan_fit)
mean(samples_laplace$mu>100)
```

```
## [1] 0.9455
```

Under the Laplace model,the posterior probability that the median IQ in the town is greater than 100 is 0.9315. Under the normal model it was 0.77225. This makes sense because we are using a model that is more robust to the certain kinds of outliers in our problem, therefore our posterior probability is greater.