

# Homework #4 STA 360

Jerry Xin STA 360: Homework 4

Due Friday September 17th, 5 PM EDT

```
library(plyr)
library(ggplot2)
library(dplyr)
library(xtable)
library(reshape)
library(tidyverse)
```

1. (10 points, 5 points each) Hoff, 3.10 (Change of variables).

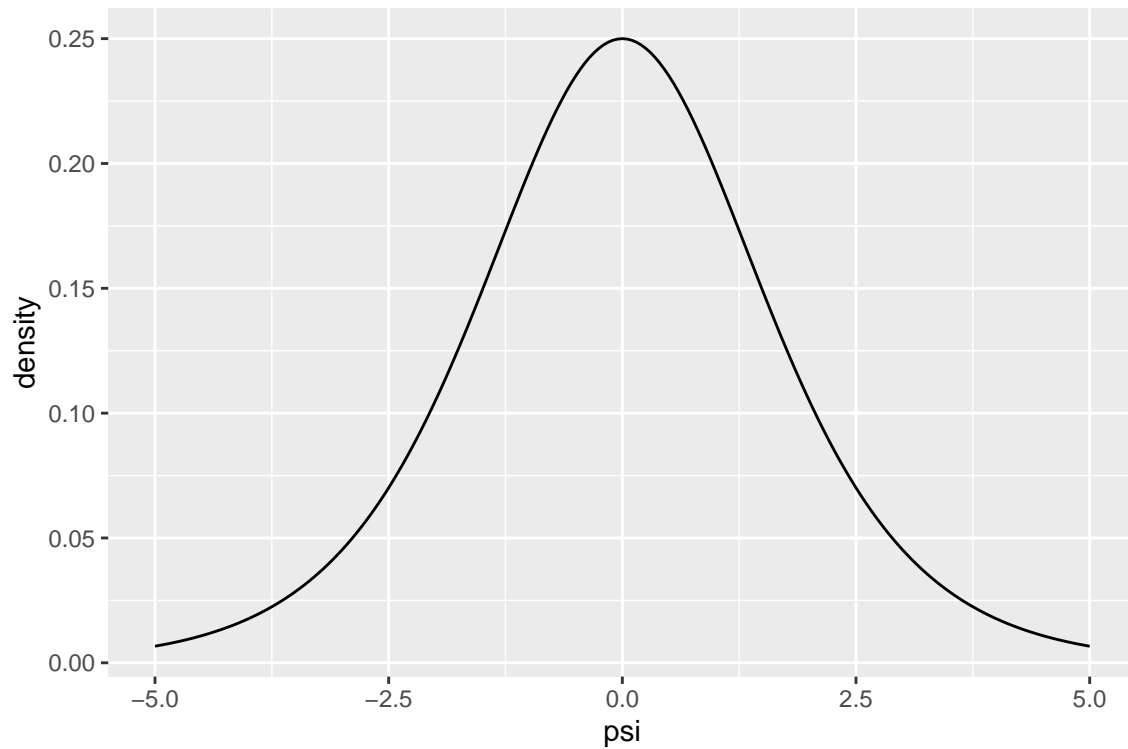
(a)

Mathematical calculations are on separate sheet of paper (handwritten).

This is the Plot for the case  $a = b = 1$ :

```
h = function(psi)
{
  exp(psi) / (exp(psi) + 1)
}
dh = function(psi)
{
  exp(psi) / (exp(psi) + 1)^2
}
dpsi = function(psi, a, b)
{
  (gamma(a + b) / (gamma(a) * gamma(b))) * h(psi)^(a - 1) * (1 - h(psi))^(
    (b - 1) * dh(psi)
  )
}
```

```
psi = seq(-5, 5, by = 0.05)
density = dpsi(psi, 1, 1)
qplot(psi, density, geom = 'line')
```



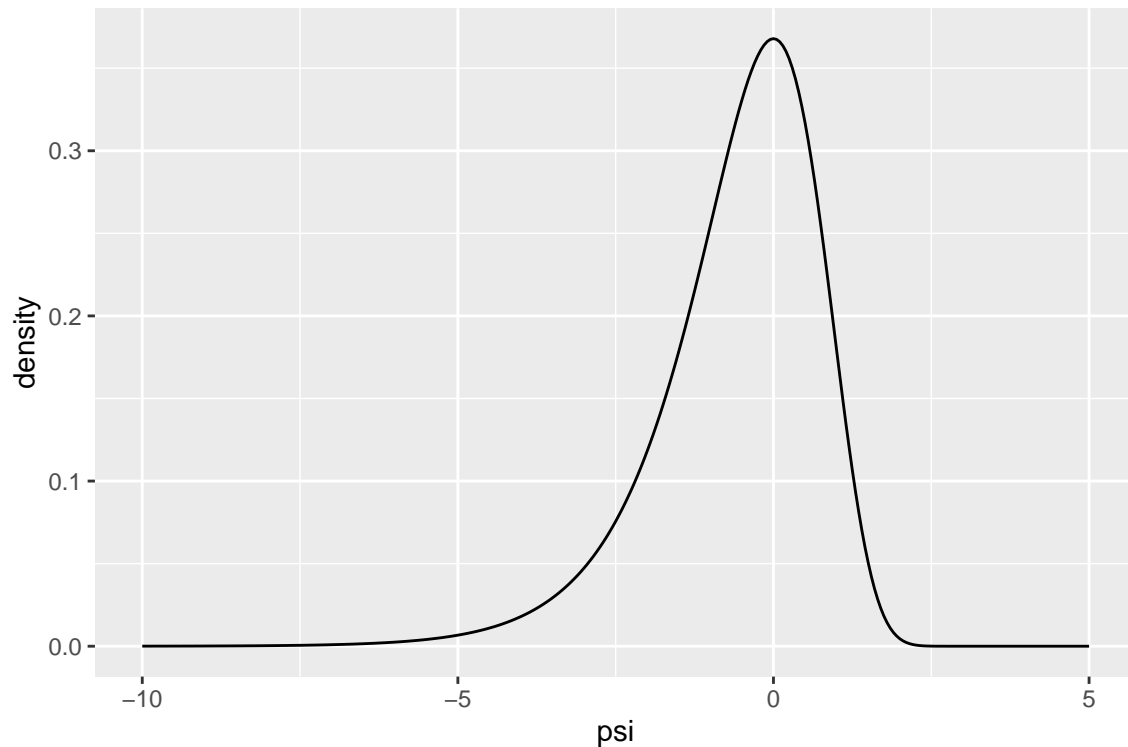
(b)

Mathematical calculations are on separate sheet of paper (handwritten).

This is the Plot for the case  $a = b = 1$ :

```
dpsi = function(psi, a, b) {
  (b^a) / (gamma(a)) * exp(a * psi - b * exp(psi))
}
```

```
psi = seq(-10, 5, by = 0.05)
density = dpsi(psi, 1, 1)
qplot(psi, density, geom = 'line')
```



2. Lab component (25 points total) Please refer to lab 4 and complete tasks 4—5.

### (a) (10) Task 4 (Finish for Homework)

```
set.seed(123)
# input data
# spurters
x = c(18, 40, 15, 17, 20, 44, 38)
# control group
y = c(-4, 0, -19, 24, 19, 10, 5, 10,
      29, 13, -9, -8, 20, -1, 12, 21,
      -7, 14, 13, 20, 11, 16, 15, 27,
      23, 36, -33, 34, 13, 11, -19, 21,
      6, 25, 30, 22, -28, 15, 26, -1, -2,
      43, 23, 22, 25, 16, 10, 29)
# store data in data frame
iqData = data.frame(Treatment = c(rep("Spurters", length(x)),
                                   rep("Controls", length(y))),
                    Gain = c(x, y))

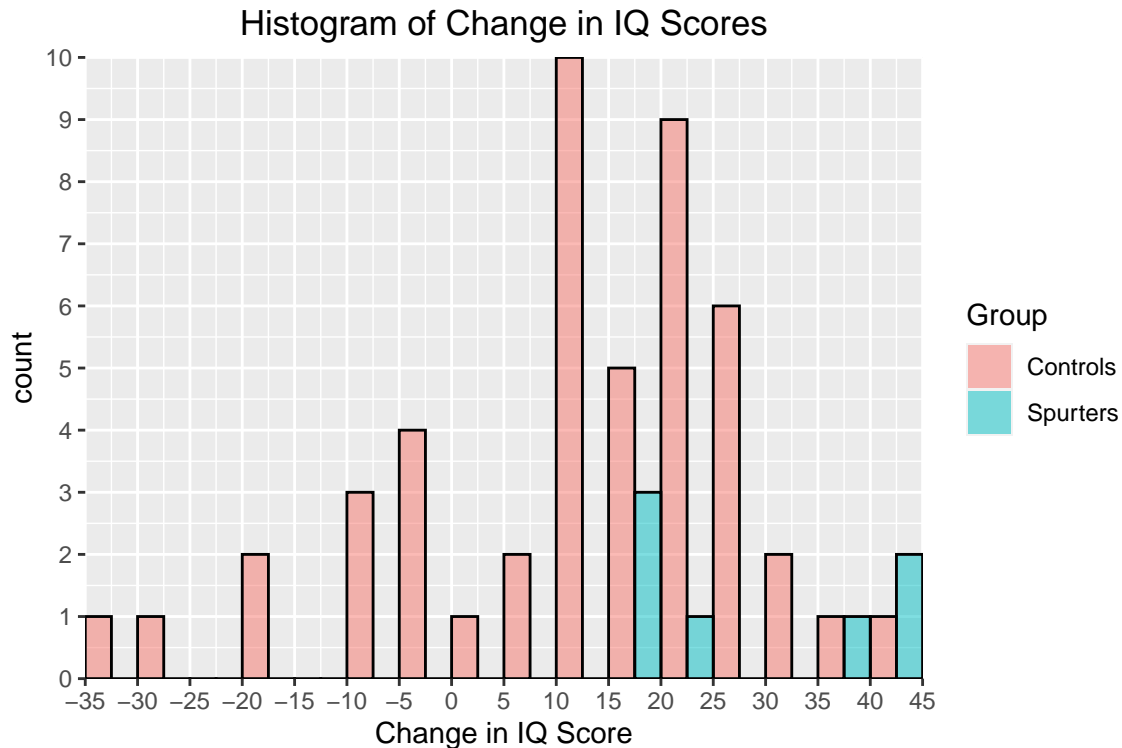
xLimits = seq(min(iqData$Gain) - (min(iqData$Gain) %% 5),
              max(iqData$Gain) + (max(iqData$Gain) %% 5),
              by = 5)

ggplot(data = iqData, aes(x = Gain, fill = Treatment, colour = I("black"))) +
  geom_histogram(position = "dodge", alpha = 0.5,
                 breaks = xLimits, closed = "left") +
  scale_x_continuous(breaks = xLimits,
                     expand = c(0,0)) +
```

```

scale_y_continuous(expand = c(0,0),
breaks = seq(0, 10, by = 1))+
ggtitle("Histogram of Change in IQ Scores") +
labs(x = "Change in IQ Score", fill = "Group") +
theme(plot.title = element_text(hjust = 0.5))

```



```

prior = data.frame(m = 0, c = 1, a = 0.5, b = 50)
findParam = function(prior, data){
  postParam = NULL
  c = prior$c
  m = prior$m
  a = prior$a
  b = prior$b
  n = length(data)
  postParam = data.frame(m = (c*m + n*mean(data))/(c + n),
                          c = c + n,
                          a = a + n/2,
                          b = b + 0.5*(sum((data - mean(data))^2)) +
                            (n*c*(mean(data) - m)^2)/(2*(c+n)))
  return(postParam)
}
postS = findParam(prior, x)
postC = findParam(prior, y)

```

% latex table generated in R 4.0.2 by xtable 1.8-4 package % Fri Sep 17 02:19:18 2021

```

# sampling from two posteriors
set.seed(123)
# Number of posterior simulations
sim = 1000

```

	m	c	a	b
prior	0.00	1.00	0.50	50.00
Spurters Posterior	24.00	8.00	4.00	855.00
Controls Posterior	11.80	49.00	24.50	6343.98

Table 1: Parameters

```
# initialize vectors to store samples
mus = NULL
lambdas = NULL
muc = NULL
lambdac = NULL

# Following formula from the NormalGamma with
# the update paramaters accounted accounted for below

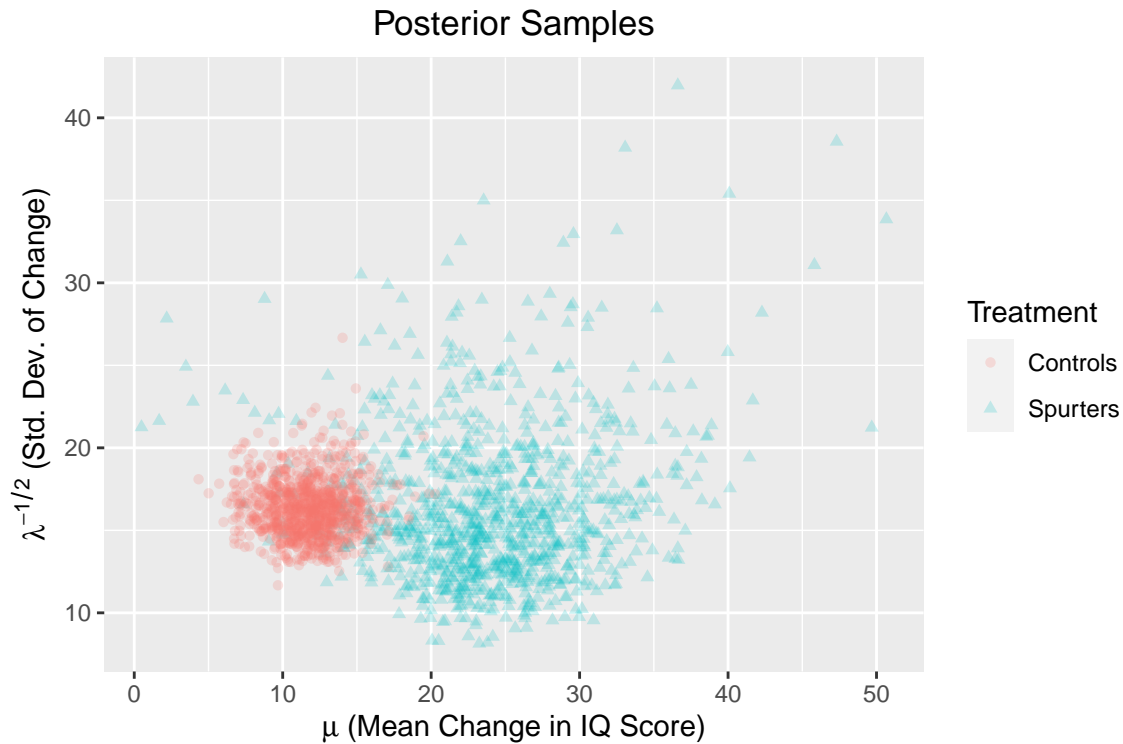
lambdas = rgamma(sim, shape = postS$a, rate = postS$b)
lambdac = rgamma(sim, shape = postC$a, rate = postC$b)

mus = sapply(sqrt(1/(postS$c*lambdas)),rnorm, n = 1, mean = postS$m)
muc = sapply(sqrt(1/(postC$c*lambdac)),rnorm, n = 1, mean = postC$m)

# Store simulations
simDF = data.frame(lambda = c(lambdas, lambdac),
                    mu = c(mus, muc),
                    Treatment = rep(c("Spurters", "Controls"),
                                     each = sim))

simDF$lambda = simDF$lambda^{-0.5}

# Plot the simulations
ggplot(data = simDF, aes(x = mu, y = lambda,
                        colour = Treatment, shape = Treatment)) +
  geom_point(alpha = 0.2) +
  labs(x = expression(paste(mu, " (Mean Change in IQ Score)")),
       y = expression(paste(lambda^{-1/2}, " (Std. Dev. of Change)"))) +
  ggtitle("Posterior Samples")+
  theme(plot.title = element_text(hjust = 0.5))
```



Now, we can answer our original question: “What is the posterior probability that  $\mu_S > \mu_C$ ?”

The easiest way to do this is to take a bunch of samples from each of the posteriors, and see what fraction of times we have  $\mu_S > \mu_C$ . This is an example of a Monte Carlo approximation (much more to come on this in the future).

To do this, we draw  $N = 10^6$  samples from each posterior:

```
set.seed(123)
mp <- 24
cp <- 8
ap <- 4
bp <- 855

ms <- 11.8
cs <- 49
as <- 24.5
bs <- 6344.0

#Sampling from posterior
S <- 1000000

# Number of posterior simulations
sim = 1000000

# initialize vectors to store samples
mu_s = NULL
lambda_s = NULL
mu_c = NULL
lambda_c = NULL
```

```

# Following formula from the NormalGamma with
# the update parameters accounted for below

lambda_c = rgamma(S, as, bs)
lambda_s = rgamma(S, ap, bp)

mu_c = rnorm(S, ms, sd = sqrt(1/(cs*lambda_c)))
mu_s = rnorm(S, mp, sd = sqrt(1/(cp*lambda_s)))

# Store simulations
dataF = data.frame(lambda = c(lambda_s, lambda_c),
                    mu = c(mu_s, mu_c),
                    Treatment = rep(c("Spurters", "Controls"),
                                    each = sim))

dataF$lambda = dataF$lambda^{-0.5}
mean(mu_s > mu_c)

```

```
## [1] 0.970683
```

Interpret the posterior probability that you compute above.

After sampling  $10^6$  times, we get 0.970683. 0.970683 means that 97.0683% of people who were told they were spurters (special group) had a higher change in IQ than the change in IQ of the control group. This also means that the probability that the spurters (special group) have a higher change in IQ than the change in IQ of the control group is 97.0683%. We expect that 97.0683% of spurters (special group) to have a larger mean than the control.

## (b) (15) Task 5 (Finish for Homework)

Let's return back to the prior assumptions. There are a few ways that you can check that the prior conforms with our prior beliefs. Let's go back and check these. Draw some samples from the prior and look at them—this is probably the best general strategy. See Figure pygmalion-prior. It's also a good idea to look at sample hypothetical datasets  $X_{1:n}$  drawn using these sampled parameter values.

Please replicate a plot similar to Figure pygmalion-prior and report your findings.

```

set.seed(123)
sim = 1000

mus = NULL
lambdas = NULL
muc = NULL
lambdac = NULL

lambdas = rgamma(sim, shape = prior$a, rate = prior$b)

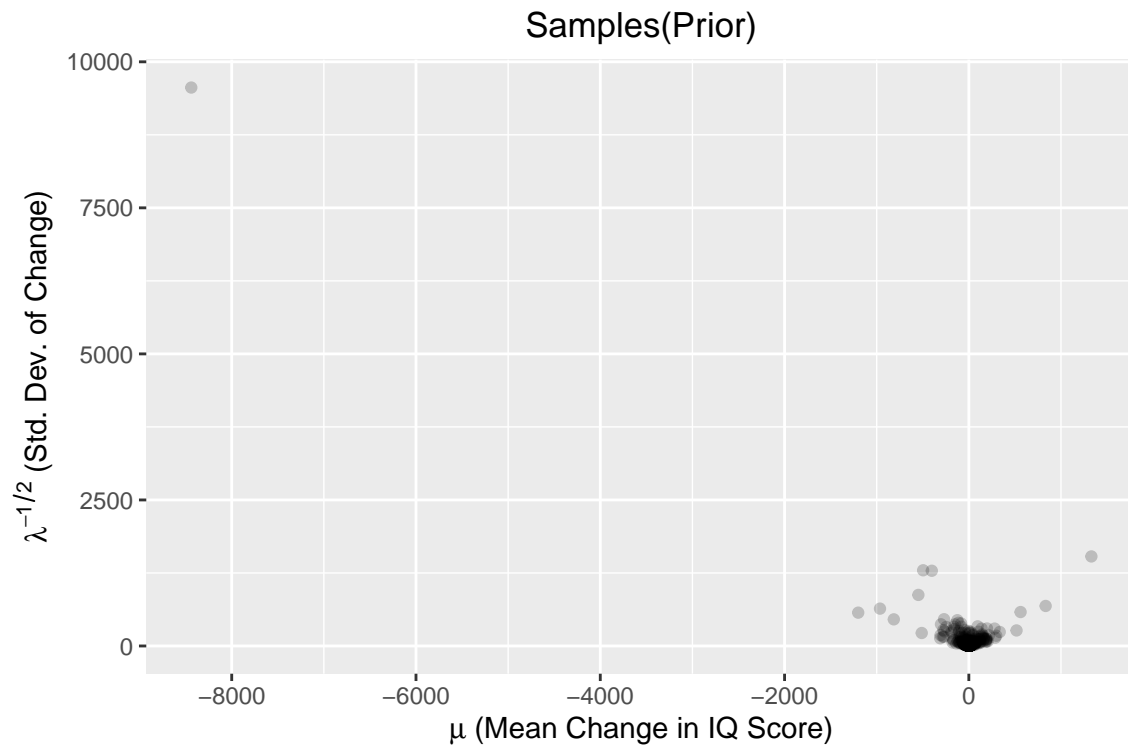
mus = sapply(sqrt(1/(prior$c*lambdas)), rnorm, n = 1, mean = prior$m)

simDF = data.frame(lambda = lambdas,
                    mu = mus)

simDF$lambda = simDF$lambda^{-0.5}

```

```
ggplot(data = simDF, aes(x = mu, y = lambda)) +
  geom_point(alpha = 0.2) +
  labs(x = expression(paste(mu, " (Mean Change in IQ Score)")),
       y = expression(paste(lambda^{-1/2}, " (Std. Dev. of Change)"))) +
  ggtitle("Samples(Prior)") +
  theme(plot.title = element_text(hjust = 0.5))
```



Graphically the fact that the samples are centered at 0 is fine because it is reasonable to assume that there is no change in the mean. However, because there seems to be a statistically significant amount of outliers, this means our prior assumption is not reasonable.