STA 360/602: Assignment 4, Spring 2019

Kuei-Yueh Ko

Due at 10:00 AM on Monday, January 2019

Contents

Problem 1: 3.10 Change of variables

Part (a)

$$\theta \sim beta(a,b) \Rightarrow \frac{1}{\text{Beta}(a,b)}\theta^{a-1}(1-\theta)^{b-1}; \theta \in (0,1)$$

$$\psi = g(\theta) = \log\frac{\theta}{1-\theta} \Rightarrow \theta = h(\psi) = \frac{1}{1+e^{-\psi}}$$

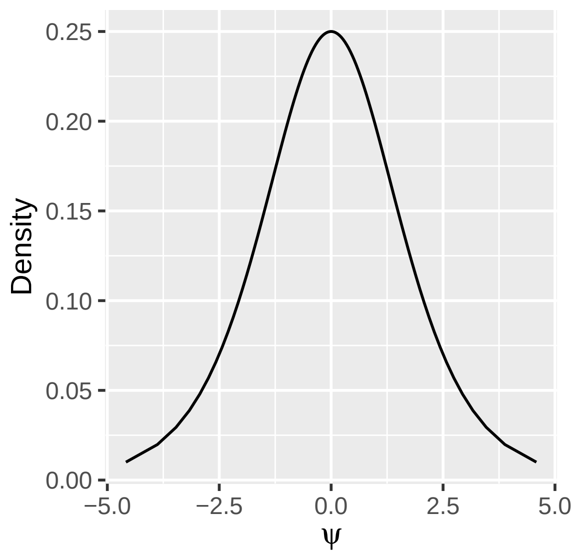
$$\theta \in (0,1) \Rightarrow \psi = \log\frac{\theta}{1-\theta} \in (-\inf, \inf)$$

$$p_\psi(\psi) = p_\theta(\theta) \times |\frac{dh}{d\psi}| = \frac{1}{\text{Beta}(a,b)}\left(\frac{1}{1+e^{-\psi}}\right)^{a-1}\left(\frac{e^{-\psi}}{1+e^{-\psi}}\right)^{b-1}\frac{e^{-\psi}}{(1+e^{-\psi})^2}$$

plot the $p_\psi(\psi)$ with parameter $a = 1, b = 1$

```
### set up function
myfun <- function(psi, a, b){
    x = 1 / (1 + exp(-psi))
    #y = x^(a - 1) * (1 - x)^(b - 1) * exp(-psi)
    y = dbeta(x, shape1 = a, shape2 = b) * exp(-psi) * x^2
    return(y)
} # end func

### parameters
a = 1; b = 1
th  = seq(0, 1, length.out = 100)
th  = th[ -c(1, length(th)) ]
psi = log(th / (1-th))
```

```r
### generate probability density
dat = data.frame(
    theta = th,
    psi   = psi,
    prob  = myfun(psi, a, b)
) # end df

### visualization
gp = ggplot(dat, aes(x = psi, y = prob)) +
    geom_line() +
    labs(x = expression(psi), y = "Density")
print(gp)
```



Part (b)

$$\theta \sim gamma(a,b) \Rightarrow \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta}; \text{support}: \theta \in (0, \inf)$$

$$\psi = g(\theta) = \log\theta \Rightarrow \theta = h(\psi) = e^{\psi}$$

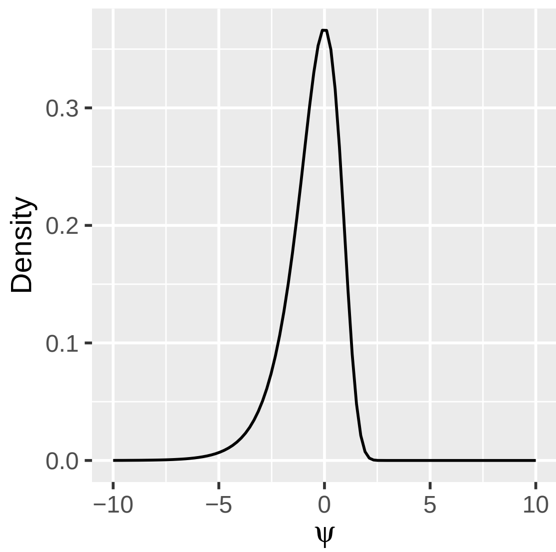
$$\theta \in (0, \inf) \Rightarrow \psi = \log\theta \in (-\inf, \inf)$$

$$p_{\psi}(\psi) = p_{\theta}(\theta) \times |\frac{dh}{d\psi}| = \frac{b^a}{\Gamma(a)}e^{\psi(a-1)}e^{-be^{\psi}} \times e^{\psi} = \frac{b^a}{\Gamma(a)}\exp\{a\psi - be^{\psi}\}$$

plot the $p_{\psi}(\psi)$ with parameter $a = 1, b = 1$

```r
### set up function
myfun <- function(psi, a, b){
    x = exp(psi)
```

```
    y = dgamma(x, shape = a, scale = b) * x
    return(y)
} # end func

### parameters
a = 1; b = 1
psi  = seq(-10, 10, length.out = 100)

### generate probability density
dat = data.frame(
    psi   = psi,
    prob  = myfun(psi, a, b)
) # end df

### visualization
gp = ggplot(dat, aes(x = psi, y = prob)) +
    geom_line() +
    labs(x = expression(psi), y = "Density")
print(gp)
```



Problem 2: Lab 04 Task 4 and Task 5

set up function to find the posterior

```
findParam = function(prior, data){
    ### initialization
    c = prior$c
    m = prior$m
    a = prior$a
    b = prior$b
    n = length(data)
```

```r
    ### calculate posterior and return the list of parameters
    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)
}
```

set the data, prior and get the poster from data and prior

```r
### input data
x = c(18, 40, 15, 17, 20, 44, 38)
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))

### set prior and find posterior
prior = data.frame(m = 0, c = 1, a = 0.5, b = 50)
postS = findParam(prior, x)
postC = findParam(prior, y)
```

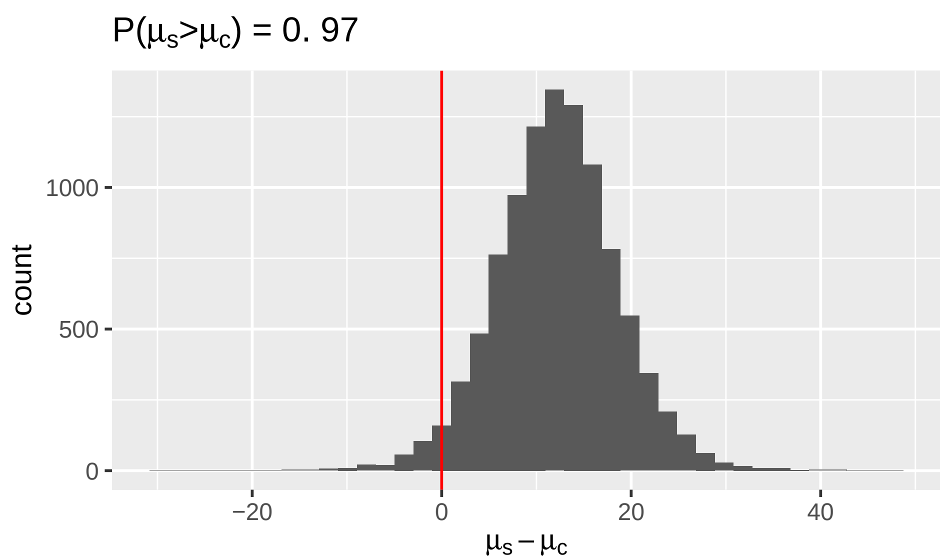
Task 04

Simulate μ and λ from the posterior. Then, count the incidents of $\mu_s > \mu_c$. The probability of $\mu_s > \mu_c$ can be approximated from the frequency of such incidents.

$$P(\mu_s > \mu_c | x_{1:n_s}, y_{1:n_c}) \approx \frac{1}{N} \sum_{i=1}^{N} ⊮(\mu_s^{(i)} > \mu_c^{(i)})$$

```r
### initialization
set.seed(123)
sim = 1e+4

### simulating lambda (1/sigma2) and mu (mean)
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)

### arrange the results
```

```
simDF = data.frame(mu_diff = mus - muc)

### visualize
gp = ggplot(simDF, aes(x = mu_diff)) +
    geom_histogram(bins = 40) +
    geom_vline(xintercept = 0, color = "red") +
    labs(title = TeX(paste("P($\\mu_s > \\mu_c$) =", mean(simDF$mu_diff > 0))),
         x = TeX("$\\mu_s - \\mu_c$"))
print(gp)
```



Task 05

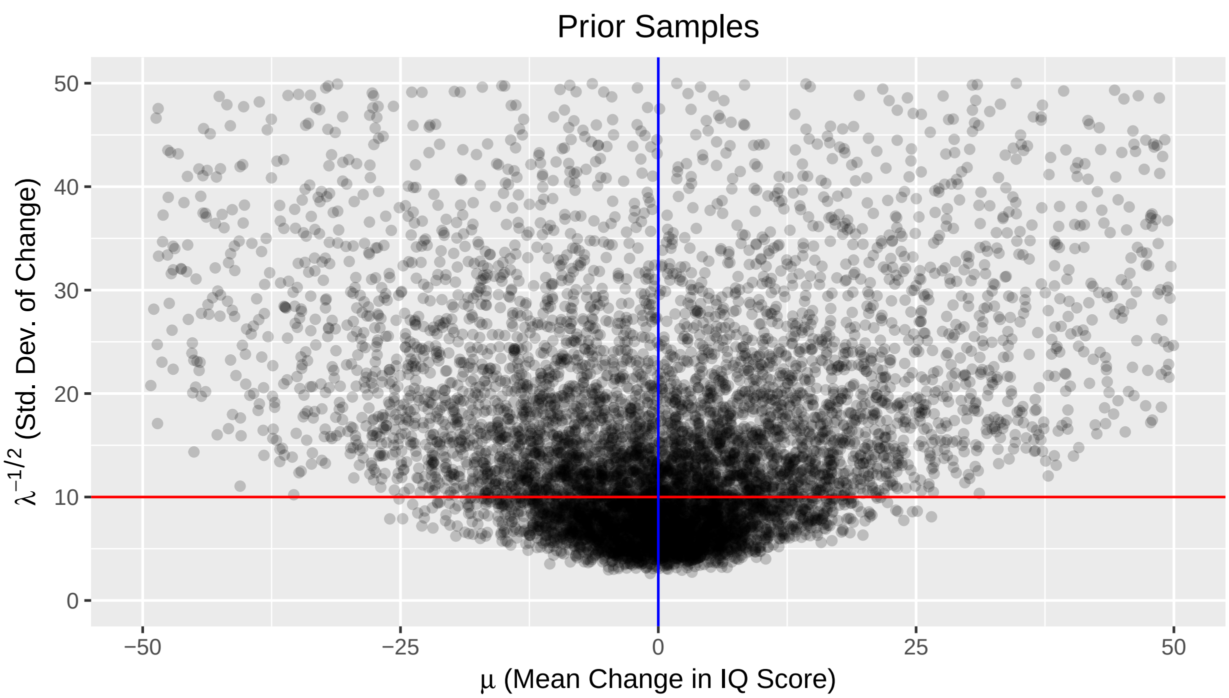
Does the parameters of prior conforms with our prior beliefs?

From the simulate, we could observe

- the distribution of mu is symmetric around zero (blue line). That is, we do not know whether students will improve or not.
- the standard deviation of the changes is first set to be around 10 (red line).

```
### initialization
set.seed(123)
sim = 1e+4

### simulation
lambda = rgamma(sim, shape = prior$a, rate = prior$b)
mu = sapply(sqrt(1/(prior$c*lambda)),rnorm, n = 1, mean = prior$m)

### arrange the data
simDF = data.frame(lambda = lambda, mu = mu)
simDF$lambda = simDF$lambda^{-0.5}

## visualize the simulated paramters from the prior
```

```
gp = ggplot(data = simDF, aes(x = mu, y = lambda)) +
    geom_point(alpha = 0.2) + ylim(c(0, 50)) + xlim(c(-50, 50)) +
    geom_hline(yintercept = sqrt(prior$b/prior$a), color = "red") +
    geom_vline(xintercept = 0,                     color = "blue") +
    labs(title = "Prior Samples",
         x = expression(paste(mu, " (Mean Change in IQ Score)")),
         y = expression(paste(lambda^{-1/2}, " (Std. Dev. of Change)"))) +
    theme(plot.title = element_text(hjust = 0.5))
print(gp)
```

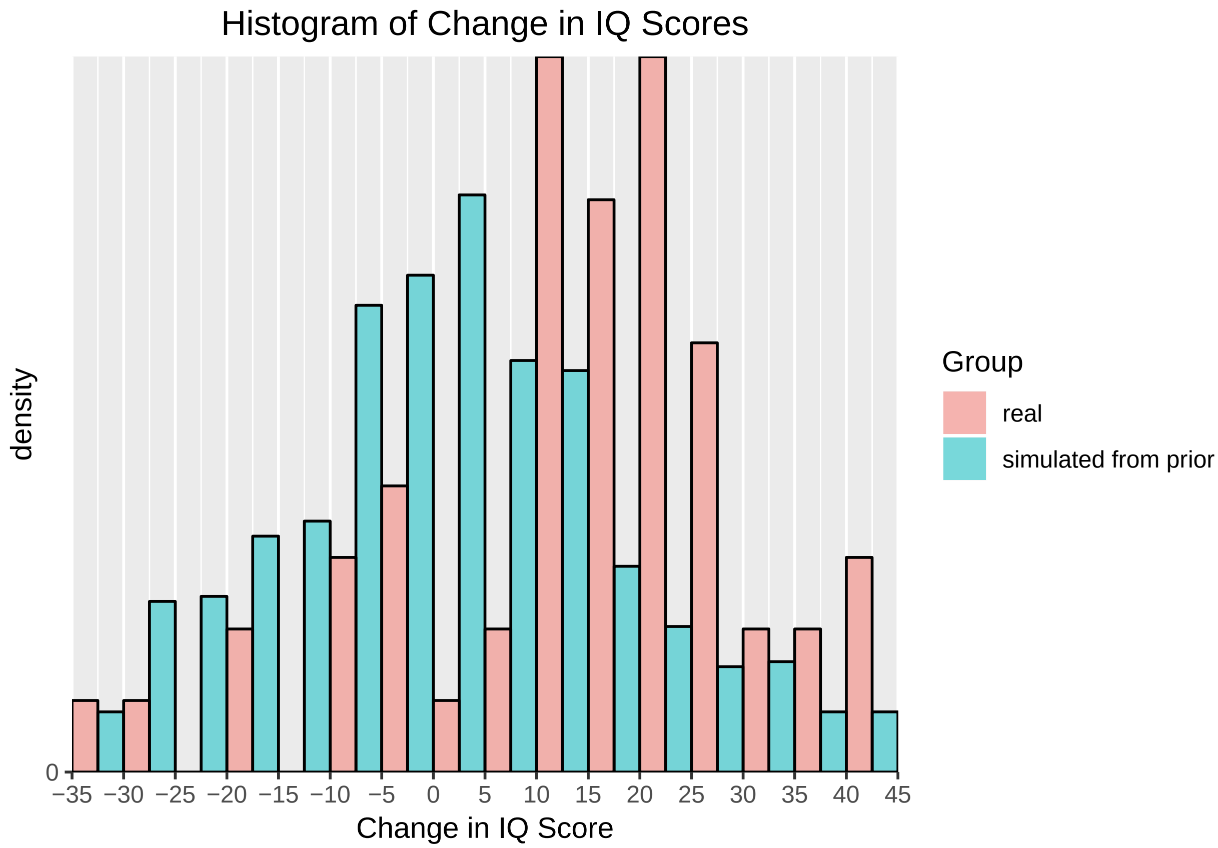


Hypothetical datasets drawn using sampled parameters values simulated from prior

Simulate data from prior distribution and compare the simulated data and the observed data. From the histogram, the distribution of simulated data from prior looks similar to the distribution of data. Since they are similar, we could use the plot to show that the prior parameters are suitable for calculating the posterior parameters given the real data.

```
### data
x = c(18, 40, 15, 17, 20, 44, 38)
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)

### simualted parameters from prior
set.seed(123)
sim    = 1e+3 #length(x) + length(y)
lambda = rgamma(sim, shape = prior$a, rate = prior$b)
mu     = sapply(sqrt(1/(prior$c*lambda)),rnorm, n = 1, mean = prior$m)
```

```r
### simualted data from prior
z = mapply(
    function(a_mu, a_sig){rnorm(1, mean = a_mu, sd = a_sig)},
    mu,
    1 / sqrt(lambda),
    SIMPLIFY = TRUE)

### store all data in data frame
iqData2 = data.frame(
    Treatment = c(rep("real",  length(x)),
                  rep("real",  length(y)),
                  rep("simulated from prior", length(z))),
    Gain = c(x, y, z))

### set up breaks of historgram with binwidth = 5
xLimits = seq(min(iqData$Gain) - (min(iqData$Gain) %% 5),
              max(iqData$Gain) + (max(iqData$Gain) %% 5),
              by = 5)

### visualization
gp = ggplot(data = iqData2, aes(
        x = Gain, y=stat(density), #..density..,
        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))+
    labs(title = "Histogram of Change in IQ Scores",
         x = "Change in IQ Score", fill = "Group") +
    theme(plot.title = element_text(hjust = 0.5))
print(gp)
```


Histogram of Change in IQ Scores
density
0
−35
−30
−25
−20
−15
−10
−5
0
5
10
15
20
25
30
35
40
45
Change in IQ Score
Group
real
simulated from prior