BDA - Assignment 3

Anonymous

Contents

```
library(markmyassignment)
exercise_path = 'https://github.com/avehtari/BDA_course_Aalto/blob/master/exercises/tests/ex3.yml'
set_assignment(exercise_path)
```

```
## Assignment set:
## ex3: Bayesian Data Analysis: Assignment 3
## The assignment contain the following (6) tasks:
## - mu_point_est
## - mu_interval
## - mu_pred_interval
## - mu_pred_point_est
## - posterior_odds_ratio_point_est
## - posterior_odds_ratio_interval
```

```
library(aaltobda)
data('windshieldy1')
head(windshieldy1)
```

```
## [1] 13.357 14.928 14.896 15.297 14.820 12.067
```

```
windshieldy_test = c(13.357, 14.928, 14.896, 14.820)
```

- Posterior distribution:

$$t_{n-1}(\mu, \frac{s^2}{n})$$

$$t_8(14.61, \frac{2.17}{9})$$

- Prior distribution:

$$p(\mu, \frac{1}{s^2})$$

$$p(14.5, \frac{1}{2.17})$$

- Likelihood:

$$p(\mu)$$

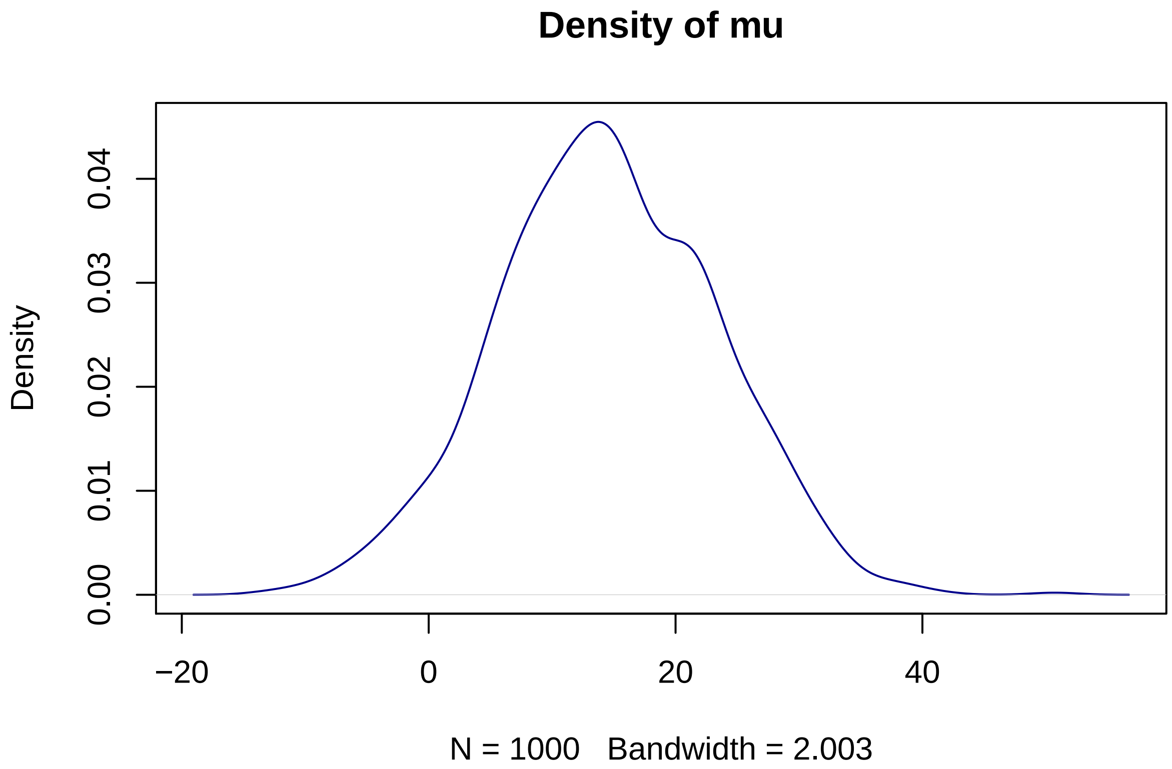
$$p(14.5)$$

Exercise 1 a

We assume here that the observations from sample y1 follow a normal distribution with an unknown standard deviation sigma and we wish to obtain information about the unknown average hardness mu.

```r
mu_point_est = function(data){
  n = length(data)
  sample_var = var(data)
  sample_std = sqrt(sample_var)
  y_hat = mean(data)

  # Density by simulation
  n_samples = 1000
  sigma_rand = n-1*sample_var / rchisq(n_samples, n-1)
  posterior_median = y_hat + sqrt(sigma_rand/n)*rnorm(length(sigma_rand))
  y_new = rnorm(n_samples, posterior_median, sigma_rand)
  mu_sim_mean = mean(y_new, na.rm=TRUE)
  cat('The   from data is:',y_hat,' and by simulation the   is:', mu_sim_mean, '\n')

  plot(density(y_new),col='darkblue',main='Density of mu')

}
mu_point_est(windshieldy1)
```

```
## The   from data is: 14.61122  and by simulation the   is: 14.43491
```



```
mu_interval = function(data, prob){
  n = length(data)
  sample_var = var(data)
  sample_std = sqrt(sample_var)
  y_hat = mean(data)

  error = qt(0.975, n-1)*sample_std/sqrt(n)
  lower =  y_hat - error
  upper = y_hat + error
  cat('The 95% central interval for mu is: [',lower,',',upper,']')
}

mu_interval(windshieldy1)
```

```
## The 95% central interval for mu is: [ 13.47808 , 15.74436 ]
```

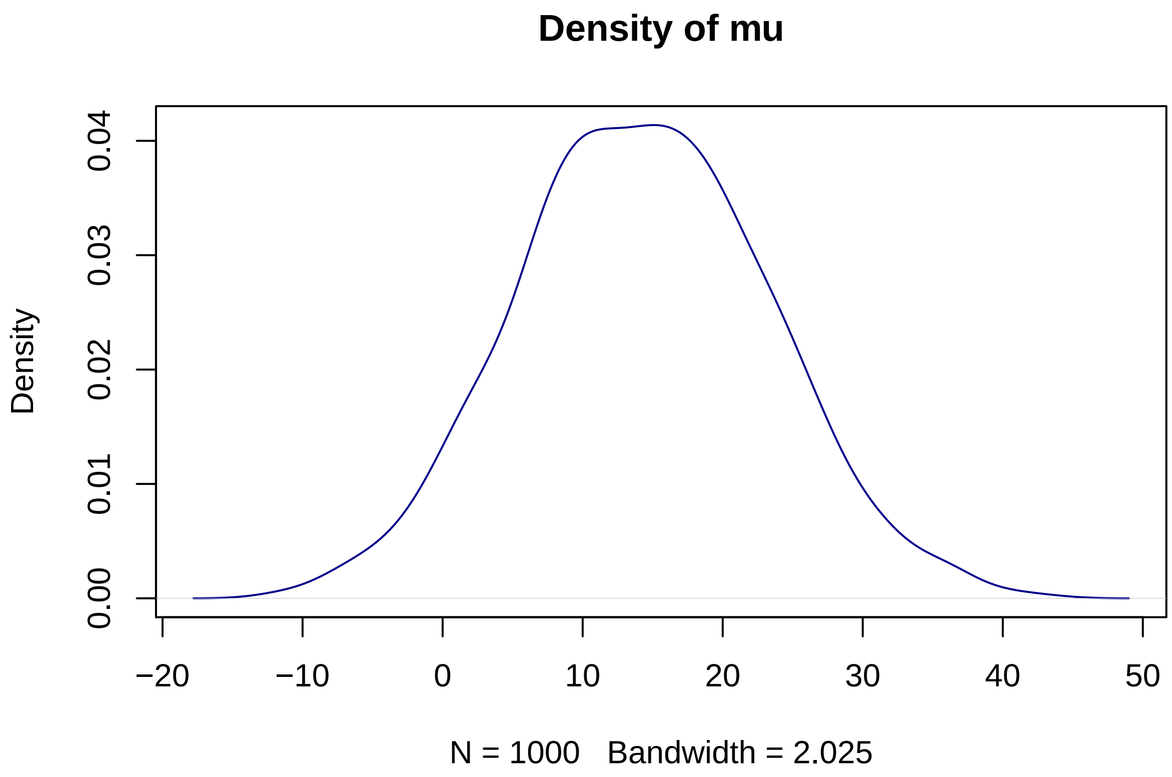
Exercise 1 b

The posterior predictive distribution for a future observation mu_hat is a t-distribution with location y_hat scale (1+1/n)^(1/2) and n-1 degrees of freedom. In other words:

$$p(\bar{\mu}|y) = t_{n-1}(\bar{y}, (1+1/n)s^2)$$

```r
mu_pred_point_est = function(data){
  n = length(data)
  sample_var = var(data)
  sample_std = sqrt(sample_var)
  y_hat = mean(data)

  # Density by simulation
  n_samples = 1000
  sigma_rand = n-1*sample_var / rchisq(n_samples, n-1)
  posterior_median = y_hat + sqrt(sigma_rand/n)*rnorm(length(sigma_rand))

  y_new = rnorm(n_samples, posterior_median, sigma_rand)
  mu_sim_mean = mean(y_new, na.rm=TRUE)
  cat('The   from data is:',y_hat,' and by simulation the   is:', mu_sim_mean, '\n')

  plot(density(y_new),col='darkblue', main='Density of mu')
}
mu_pred_point_est(windshieldy1)
```

```
## The   from data is: 14.61122  and by simulation the   is: 14.31835
```



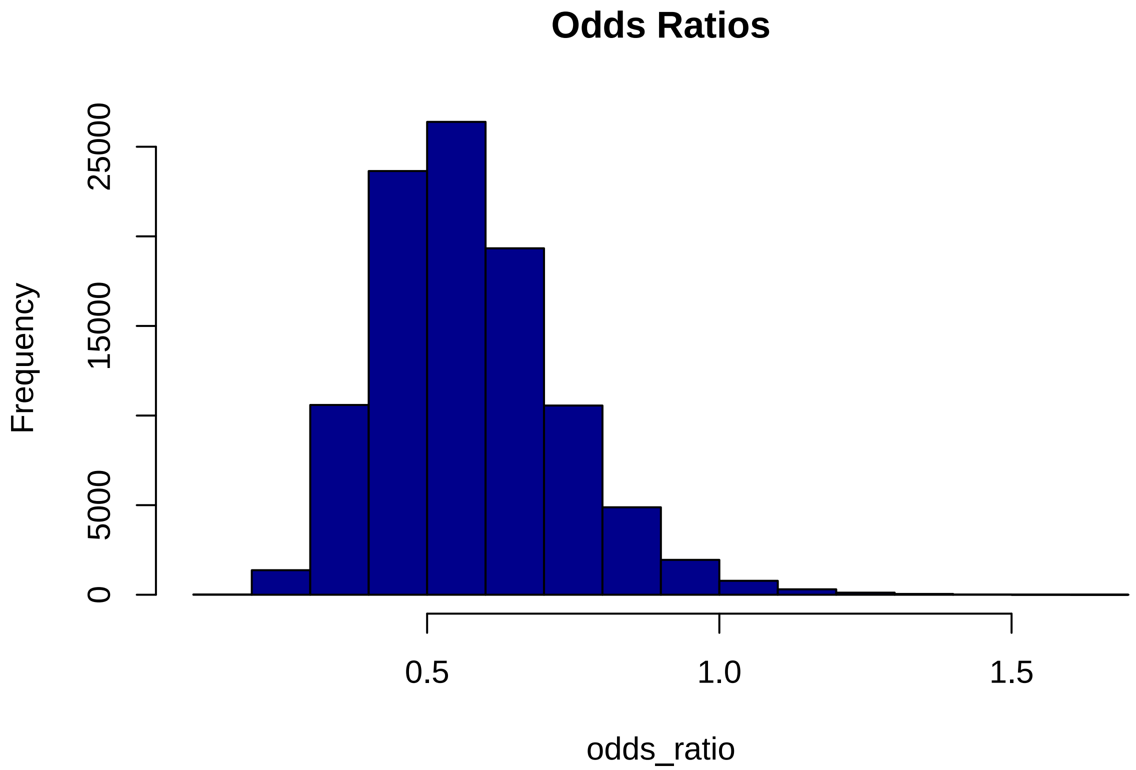
```r
mu_pred_interval = function(data, prob){
  n = length(data)
  sample_std = sqrt(var(data))
  y_hat = mean(data)
  scale = sqrt((1+1/n))*sample_std
  probs = c((1-prob)/2, 1-(1-prob)/2)
  interval = qt(probs, n-1)
  interval_shifted = y_hat + interval * scale
  cat('The 95% central interval for mu is: [',interval_shifted[1],',',interval_shifted[2],']')
}

mu_pred_interval(windshieldy1, 0.95)
```

```
## The 95% central interval for mu is: [ 11.02792 , 18.19453 ]
```

Exerices 2 a

```
# p0 = rbeta(100000,alpha0,beta0), where alpha0 = y0+1 and beta0 = n0-y0+1
# p1 = rbeta(100000,alpha0,beta0),  where alpha1 = y1+1 and beta1 = n1-y1+1

set.seed(4711)
y0 = 39
n0 = 674
y1 = 22
n1 = 680
beta1 = 680
alpha0 = y0+1; beta0 = n0-alpha0
alpha1 = y1+1; beta1 = n1-alpha1
p0 = rbeta(100000,alpha0,beta0)
p1 = rbeta(100000,alpha1,beta1)

posterior_odds_ratio_point_est = function(p0, p1){
  odds_ratio = (p1/(1 - p1))/(p0/(1 - p0))
  posterior_odd_ratio_point_est = mean(odds_ratio)
  cat('The posterior odds ratio point estimate is:', posterior_odd_ratio_point_est, '\n')
  hist(odds_ratio, breaks = 20, main='Odds Ratios', col='darkblue')
}

posterior_odds_ratio_point_est(p0,p1)
```

```
## The posterior odds ratio point estimate is: 0.5709178
```



```
posterior_odds_ratio_interval = function(p0, p1, prob){
  odds_ratio = (p1/(1 - p1))/(p0/(1 - p0))
  lower_q = quantile(odds_ratio, probs = 1-prob-(1-prob)/2)
  upper_q = quantile(odds_ratio, probs = prob+(1-prob)/2)
  cat('The 95% central interval for mu is: [',lower_q,',',upper_q,']')
}
posterior_odds_ratio_interval(p0,p1,0.95)
```

```
## The 95% central interval for mu is: [ 0.3210165 , 0.9249218 ]
```

Exerices 2 b

```
library(knitr)
library(kableExtra)

df = matrix(c(1, 1, 1, 1, 0.570,'[0.320,0.923]',
             6, 94, 3, 97, 0.549, '[0.318,0.872]',
             12, 188, 6, 194, 0.541, '[0.324,0.839]',
             30, 470, 15, 485, 0.526, '[0.340,0.770]'), ncol = 6, byrow = TRUE)
colnames(df) = c('alpha0', 'beta0', 'alpha1', 'beta1', 'Point estimate', '95% interval')
df = as.table(df)
```

```
kable(df, 'latex', booktabs = T, caption = 'Sensitivity analysis', linesep = "") %>%
  kable_styling(latex_options = c('striped', 'hold_position', 'scale_down'))
```

Table 1: Sensitivity analysis

	alpha0	beta0	alpha1	beta1	Point estimate	95% interval
A	1	1	1	1	0.57	[0.320,0.923]
B	6	94	3	97	0.549	[0.318,0.872]
C	12	188	6	194	0.541	[0.324,0.839]
D	30	470	15	485	0.526	[0.340,0.770]

By computing different combinations of alpha0;beta0 and alpha1;beta1 with the same simulation algorithm as above (posterior_odds_ratio_point_est and posterior_odds_ratio_interval), the point estimate decreases as we increase the parameters of the prior distributions (alpha0;beta0 and alpha1;beta1). Furthermore, the 95% interval is noticeably narrower with larger prior parameters. It's interesting, however, that the lower bound of the 95% interval does not change as much as the upper bound. To summarize our findings, the choice of prior distribution has a noticeable effect on the results. The uniform prior is arguably the simplest to use and it gives a rough estimate of the results. But since we assumed that the outcomes are independent and binomially distributed, using a Beta prior might be more rational.

Exercise 3 a

- Windshieldy1 posterior:

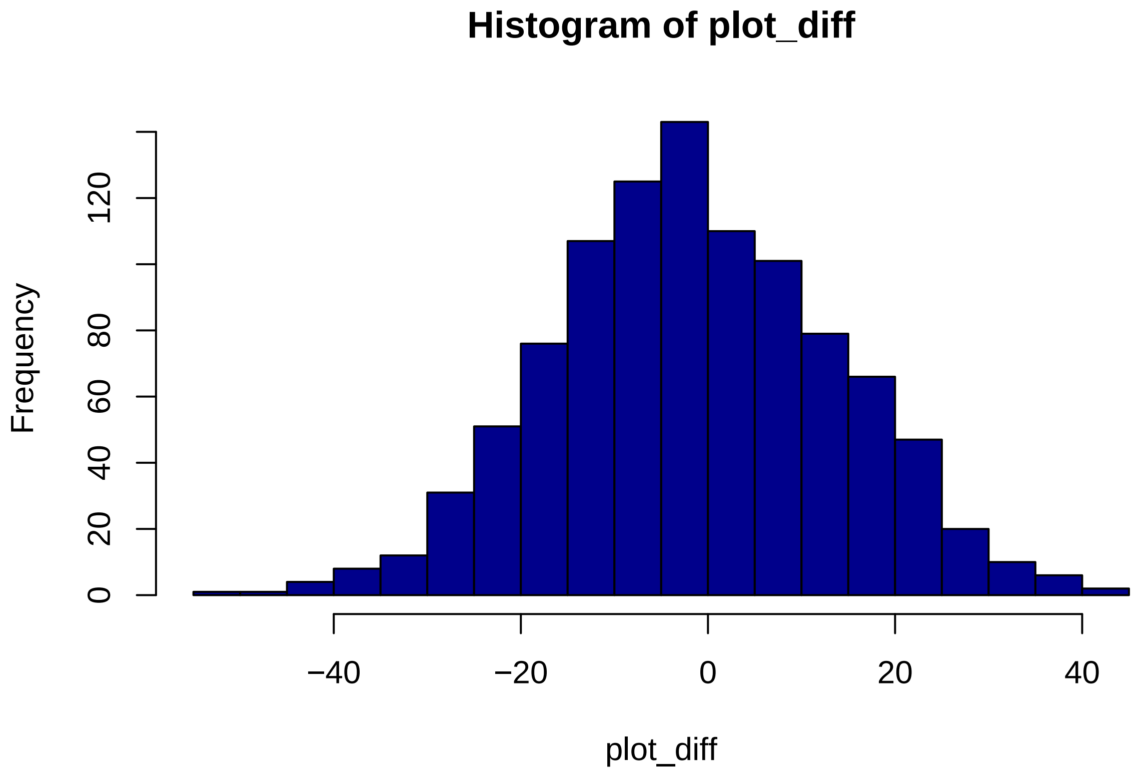
$$t_8(14.61, \frac{2.17}{9})$$

- Windshieldy2 posterior:

$$t_{14}(15.82), \frac{0.761}{13}$$

```
data(windshieldy1)
data(windshieldy2)

difference_means = function(data1, data2){
  n1 = length(data1)
  sample_var1 = var(data1)
  sample_std1 = sqrt(sample_var1)
  y_hat1 = mean(data1)
  # Density by simulation
  n_samples = 1000
  sigma_rand1 = n1-1*sample_var1 / rchisq(n_samples, n1-1)
  posterior_median1 = y_hat1 + sqrt(sigma_rand1/n1)*rnorm(length(sigma_rand1))
  y_new1 = rnorm(n_samples, posterior_median1, sigma_rand1)
  mu_sim_mean1 = mean(y_new1, na.rm=TRUE)
  cat('The  1 from data is:',y_hat1,' and by simulation the  1 is:', mu_sim_mean1, '\n')
  n2 = length(data2)
  sample_var2 = var(data2)
  sample_std2 = sqrt(sample_var2)
  y_hat2 = mean(data2)
  # Density by simulation
  n_samples = 1000
  sigma_rand2 = n2-1*sample_var2 / rchisq(n_samples, n2-1)
  posterior_median2 = y_hat2 + sqrt(sigma_rand2/n2)*rnorm(length(sigma_rand2))
  y_new2 = rnorm(n_samples, posterior_median2, sigma_rand2)
  mu_sim_mean2 = mean(y_new2, na.rm=TRUE)
  cat('The  2 from data is:',y_hat2,' and by simulation the  2 is:', mu_sim_mean2, '\n')

  diff1 = y_hat1-y_hat2
  diff2 = mu_sim_mean1-mu_sim_mean2
  cat('The  d from data is:',diff1,'and by simulation the  d is:',diff2,'\n')

  plot_diff = y_new1-y_new2
  hist(plot_diff, breaks = 20, col='darkblue')

  # By using the Welch-Satterthwaite approximation t-quantile t0.025 = 2.179
  # with approximately 11.88571 degrees of freedom. This is not an
  # integer, but that is no problem because the t-distribution is defined also
  # in cases when its degree of freedom is not an integer.

  alpha1 = sample_var1/n1
  alpha2 = sample_var2/n2
  df = ((alpha1+alpha2)^2) / (alpha1^2/(n1-1)+alpha2^2/(n2-1))

  up = y_hat1-y_hat2 + 2.179*sqrt(sample_var1/n1+sample_var2/n2)
  lo = y_hat1-y_hat2 - 2.179*sqrt(sample_var1/n1+sample_var2/n2)
  cat('The 95% central interval is: [',lo,',',up,']')
}

difference_means(windshieldy1, windshieldy2)
```

```
## The  1 from data is: 14.61122  and by simulation the  1 is: 14.79766
## The  2 from data is: 15.82108  and by simulation the  2 is: 16.31254
## The  d from data is: -1.209855 and by simulation the  d is: -1.514871
```



```
## The 95% central interval is: [ -2.403411 , -0.01629889 ]
```

Exercise 3 b

With a mean of 14.61 for windshieldy1 data, and assuming a true population standard deviation of 1.47, I can conclude that windshieldy1 data has different mean score to the windshieldy2 data with mean 15.82. This is illustrated by their absolute difference of 1.209855