GRADUATE STUDENT STAT 840 A2

Vsevolod Ladtchenko 20895137

Problem 2

The value of θ is

```
theta = (qt(0.75, 3) - qt(0.25, 3)) / 1.34
theta
```

```
## [1] 1.14163
```

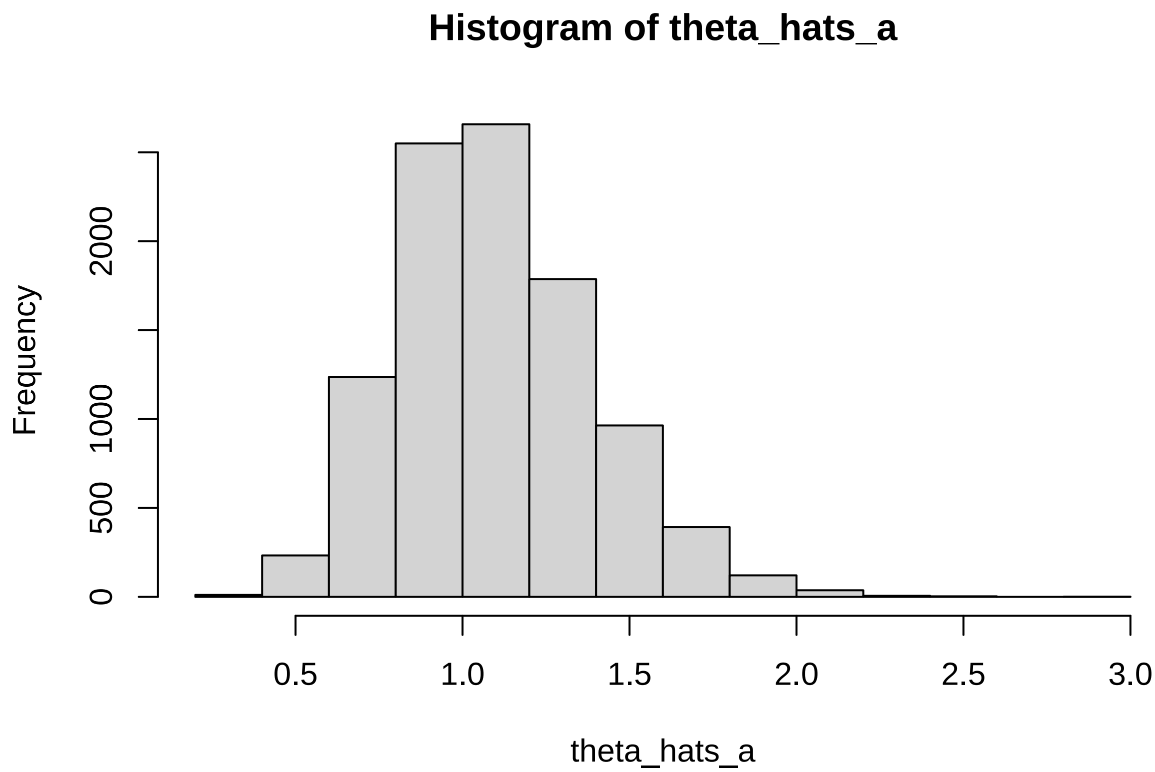
a)

```
n = 25
N = 10000

ans_a = data.frame(matrix(ncol = 2, nrow = N))
theta_hats_a = rep(NA,N)

for (j in 1:N)
{
  x = rt(n, 3)
  qs = quantile(x, probs = c(0.25, 0.75), names=FALSE)
  theta_hat = (qs[2] - qs[1]) / 1.34

  jack = rep(NA, n)
  for (i in 1:n)
  {
    qs = quantile(x[-i], probs = c(0.25, 0.75), names=FALSE)
    jack[i] = (qs[2] - qs[1]) / 1.34
  }
  theta_bar = mean(jack)
  #theta_jack = n*theta_hat - (n-1)*theta_bar
  se_jack = sqrt(((n-1) / n) * sum((jack - theta_bar)^2)) # notes 3 p 20
  ans_a[j,] = c(theta_hat + qnorm(0.025)*se_jack, theta_hat + qnorm(0.975)*se_jack)
  theta_hats_a[j] = theta_hat
}
# check if theta_hat dist is approx normal, since using normal CI
hist(theta_hats_a)
```




```r
# percent of intervals containing real theta
sum((theta > ans_a[,1]) & (theta < ans_a[,2])) / N
```

```
## [1] 0.8541
```

```r
# average interval length
mean(ans_a[,2] - ans_a[,1])
```

```
## [1] 1.25996
```

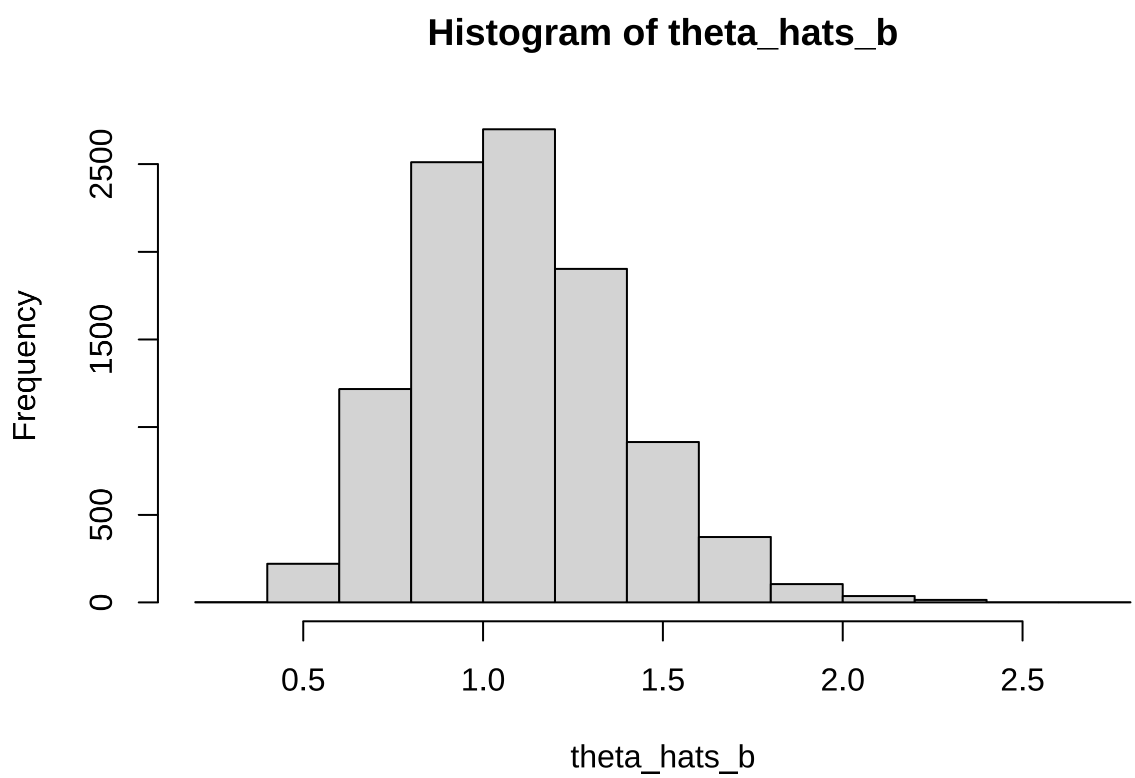
b)

```r
ans_b = data.frame(matrix(ncol = 2, nrow = N))
theta_hats_b = rep(NA,N)

for (j in 1:N)
{
  x = rt(n, 3)
  qs = quantile(x, probs = c(0.25, 0.75), names=FALSE)
  theta_hat = (qs[2] - qs[1]) / 1.34

  B = 1000
  boot = rep(NA, B)
  for (i in 1:B)
  {
    qs = quantile(sample(x, n, replace=TRUE), probs = c(0.25, 0.75), names=FALSE)
    boot[i] = (qs[2] - qs[1]) / 1.34
```

```
  }
  se_boot = sqrt(sum((boot - theta_hat)^2) / (B-1))
  ans_b[j,] = c(theta_hat + qnorm(0.025)*se_boot, theta_hat + qnorm(0.975)*se_boot)
  theta_hats_b[j] = theta_hat
}
# check if theta_hat is approx normal, since using normal CI
hist(theta_hats_b)
```




```
# percent of intervals containing real theta
sum((theta > ans_b[,1]) & (theta < ans_b[,2])) / N
```

```
## [1] 0.9668
```

```
# average interval length
mean(ans_b[,2] - ans_b[,1])
```

```
## [1] 1.407512
```

c)

```
ans_c = data.frame(matrix(ncol = 2, nrow = N))

for (j in 1:N)
{
  x = rt(n, 3)
  qs = quantile(x, probs = c(0.25, 0.75), names=FALSE)
  theta_hat = (qs[2] - qs[1]) / 1.34
```

```
  B = 1000
  boot = rep(NA, B)
  for (i in 1:B)
  {
    qs = quantile(sample(x, n, replace=TRUE), probs = c(0.25, 0.75), names=FALSE)
    boot[i] = (qs[2] - qs[1]) / 1.34
  }
  q_boot = boot - theta_hat
  ans_c[j,] = c(theta_hat - quantile(q_boot, 0.975), theta_hat - quantile(q_boot, 0.025))
}
# percent of intervals containing real theta
sum((theta > ans_c[,1]) & (theta < ans_c[,2])) / N
```

```
## [1] 0.8424
```

```
# average interval length
mean(ans_c[,2] - ans_c[,1])
```

```
## [1] 1.307315
```

d)

Normal interval with SE from jackknife appears to have a fairly normal distribution, but with some skew, so the approximation is debatable. 85% of the intervals contain the true value of the parameter. Since the CI was for 95%, we note the disparity here. The average interval length is 1.24.

Normal interval with SE from nonparametric bootstrap also has a somewhat normal, but skewed distribution. This time, 96% of the intervals contain the true value of the parameter. The average interval length is 1.41 which is larger than for jackknife, but is consistent with the promise of the 95% confidence interval.

Nonparametric pivotal bootstrap does not require approximate normality, so we forgo a histogram of the estimates. 84% of the intervals contain the true value of the parameter. The average interval length is 1.31 which is larger than jackknife, yet does not contain the true value more often. It is smaller than nonparametric bootstrap but gives a drastic drop in true value containment.

Based on these results, nonparametric bootstrap from part (b) appears to be the best candidate. Since we used 95% quantiles, this method is the only one that yields close to 95% containment of the true value.