Estimation under mis-specification: Cauchy example

We consider estimation of the location parameter θ in the Cauchy location family

$$f_X(x;\theta) = \frac{1}{\pi}\frac{1}{1+(x-\theta)^2} \qquad x,\theta \in \mathbb{R}.$$

To generate appropriate data, we set the true value of θ_0 to be 5:

```
set.seed(101)
n<-8
theta0<-5
x<-rcauchy(n)+theta0
x<-round(x,2)
print(x)

+ [1] 7.36 5.14 3.71 3.15 6.00 6.38 1.34 6.73
```

The log-likelihood takes the form

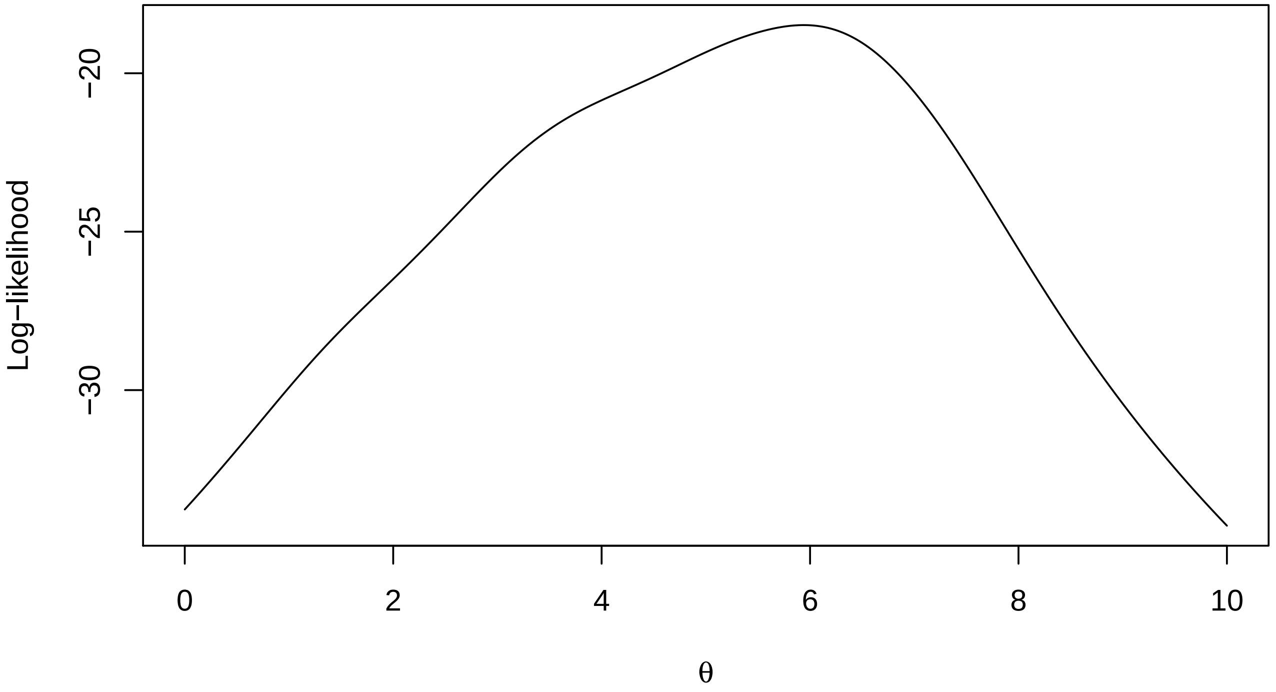
$$\ell_n(\theta) = -\log\pi - \sum_{i=1}^{n}\log(1+(x_i-\theta)^2)$$

which cannot be maximized analytically. The corresponding score function is

$$\dot{\ell}_n(\theta) = 2\sum_{i=1}^{n}\frac{(x_i-\theta)}{(1+(x_i-\theta)^2)} \tag{1}$$

which has no analytic solution if we equate to zero. We may plot the log-likelihood on a fine grid of values: computation is performed efficiently using `outer` and `apply`.

```
thvec<-seq(0,10,length=10001)
log.pdf.func<-function(x,th){
        return(-log(pi)-log(1+(x-th)^2))
}
log.pdf.mat<-outer(thvec,x,log.pdf.func)
log.like.vec<-apply(log.pdf.mat,1,sum)
par(mar=c(4,4,1,1))
plot(thvec,log.like.vec,type="l",ylab='Log-likelihood',xlab=expression(theta))
```



To find the mle, we may use different approaches

- **Newton's Method:** Newton's method requires the second derivative of the log-likelihood

$$\ddot{\ell}_n(\theta) = 2\sum_{i=1}^{n} \frac{(x_i-\theta)^2-1}{(1+(x_i-\theta)^2)^2} \tag{2}$$

  and then the recursion

$$\widehat{\theta}^{(t+1)} = \widehat{\theta}^{(t)} - \frac{\dot{\ell}_n(\widehat{\theta}^{(t)})}{\ddot{\ell}_n(\widehat{\theta}^{(t)})} \quad t = 0, 1, 2\ldots$$

  However, this method can be sensitive to starting values: starting from the sample mean, the recursion diverges as can be observed in the following.

```
nsteps<-10
theta.newton<-l0<-l1<-l2<-rep(0,nsteps)
theta.newton[1]<-mean(x)
print(c(0,theta.newton[1]))
+ [1] 0.00000 4.97625
for(j in 2:nsteps){
        th<-theta.newton[j-1]
        l1[j-1]<-2*sum((x-th)/(1+(x-th)^2))
        l2[j-1]<-2*sum(((x-th)^2-1)/((1+(x-th)^2))^2)
        theta.newton[j]<-theta.newton[j-1]-l1[j-1]/l2[j-1]
        print(c(j-1,theta.newton[j]))
}
+ [1] 1.000000 7.511358
+ [1] 2.000000 3.733407
+ [1] 3.00000 4.86724
+ [1] 4.00000 9.92511
+ [1]  5.00000 14.47505
+ [1]  6.00000 23.55046
+ [1]  7.00000 41.86728
+ [1]  8.00000 78.61892
+ [1]   9.0000 152.1894
```

- **Fisher scoring:** Fisher scoring requires the Fisher information, computed here as the expected value of the second derivative of the log-likelihood. We have

$$-\ddot{\ell}(\theta) = -2\frac{(x-\theta)^2-1}{(1+(x-\theta)^2)^2}$$

  To compute the expected value (under correct specification) we have to evaluate

$$-\frac{2}{\pi}\int_{-\infty}^{\infty} \frac{(x-\theta)^2-1}{(1+(x-\theta)^2)^3}\,dx \equiv \frac{2}{\pi}\int_{-\infty}^{\infty} \frac{1-x^2}{(1+x^2)^3}\,dx$$

  which can be achieved using integration by parts. The integral can be re-expressed using partial fractions

$$\frac{2}{\pi}\int_{-\infty}^{\infty} \left[-\frac{1}{(1+x^2)^2} + \frac{2}{(1+x^2)^3}\right]\,dx.$$

  If

$$I_r = \int_{-\infty}^{\infty} \frac{1}{(1+x^2)^r}\,dx$$

  we can easily derive the recursion

$$I_{r+1} = \frac{2r-1}{2r} I_r$$

  and with $I_1 = \pi$, we have $I_2 = \pi/2$, $I_3 = 3\pi/8$, and hence we have the Fisher information

$$\mathcal{I}_\theta(\theta) = \frac{2}{\pi}(-I_2 + 2I_3) = \frac{1}{2}.$$

Hence we have the recursion

$$\widehat{\theta}^{(t+1)} = \widehat{\theta}^{(t)} + \frac{\dot{\ell}_n(\widehat{\theta}^{(t)})}{n\mathcal{I}_{\widehat{\theta}^{(t)}}(\widehat{\theta}^{(t)})} = \widehat{\theta}^{(t)} + \frac{2}{n}\dot{\ell}_n(\widehat{\theta}^{(t)}) \quad t = 0, 1, 2\ldots$$

```
nsteps<-15
theta.scoring<-l0<-l1<-l2<-rep(0,nsteps)
theta.scoring[1]<-mean(x)
print(c(0,theta.scoring[1]))
+ [1] 0.00000 4.97625
for(j in 2:nsteps){
        th<-theta.scoring[j-1]
        l1[j-1]<-2*sum((x-th)/(1+(x-th)^2))
        theta.scoring[j]<-theta.scoring[j-1]+2*l1[j-1]/n
        print(c(j-1,theta.scoring[j]))
}
+ [1] 1.000000 5.354053
+ [1] 2.000000 5.639546
+ [1] 3.000000 5.817052
+ [1] 4.000000 5.899933
+ [1] 5.000000 5.928379
+ [1] 6.000000 5.936529
+ [1] 7.000000 5.938712
+ [1] 8.000000 5.939285
+ [1] 9.000000 5.939435
+ [1] 10.000000  5.939474
+ [1] 11.000000  5.939484
+ [1] 12.000000  5.939487
+ [1] 13.000000  5.939487
+ [1] 14.000000  5.939488
```

Both Newton's method and Fisher scoring will converge from a slightly different starting value:

```
nsteps<-15
theta.newton<-theta.scoring<-l0<-l1<-l2<-rep(0,nsteps)
theta.newton[1]<-theta.scoring[1]<-5.0
print(c(0,theta.newton[1]),theta.scoring[1])
+ [1] 0 5
for(j in 2:nsteps){
        th<-theta.newton[j-1]
        l1[j-1]<-2*sum((x-th)/(1+(x-th)^2))
        l2[j-1]<-2*sum(((x-th)^2-1)/((1+(x-th)^2))^2)
        theta.newton[j]<-theta.newton[j-1]-l1[j-1]/l2[j-1]
        th<-theta.scoring[j-1]
        l1[j-1]<-2*sum((x-th)/(1+(x-th)^2))
        theta.scoring[j]<-theta.scoring[j-1]+2*l1[j-1]/n
        print(c(j-1,theta.newton[j],theta.scoring[j]))
}
+ [1] 1.000000 7.282847 5.374087
+ [1] 2.000000 5.136358 5.653149
+ [1] 3.000000 6.594274 5.824345
+ [1] 4.000000 5.978296 5.902714
+ [1] 5.000000 5.940454 5.929212
+ [1] 6.000000 5.939488 5.936755
+ [1] 7.000000 5.939488 5.938771
+ [1] 8.000000 5.939488 5.939301
+ [1] 9.000000 5.939488 5.939439
+ [1] 10.000000  5.939488  5.939475
+ [1] 11.000000  5.939488  5.939484
+ [1] 12.000000  5.939488  5.939487
+ [1] 13.000000  5.939488  5.939487
+ [1] 14.000000  5.939488  5.939488
```

- **EM algorithm:** The EM algorithm uses the 'scale mixture' representation of the Cauchy distribution: if

$$\begin{aligned} X|Z = z &\sim Normal(\theta, 1/z) \\ Z &\sim Gamma(1/2, 1/2) \end{aligned}$$

then marginally

$$\begin{aligned} f_X(x;\theta) &= \int_0^\infty \left(\frac{z}{2\pi}\right)^{1/2} \exp\left\{-\frac{z}{2}(x-\theta)^2\right\} \frac{(1/2)^{1/2}}{\Gamma(1/2)} z^{1/2} \exp\{-z/2\}\, dz \\ &= \frac{1}{2\pi}\int_0^\infty z \exp\left\{-\frac{z}{2}[1+(x-\theta)^2]\right\}\, dz \\ &= \frac{1}{\pi}\frac{1}{1+(x-\theta)^2} \end{aligned}$$

as required, recalling that $\Gamma(1/2) = \sqrt{\pi}$. From the joint structure, we can deduce the conditional model

$$f_{Z|X}(z|x;\theta) \propto z \exp\left\{-\frac{z}{2}[1+(x-\theta)^2]\right\} \qquad z > 0$$

so therefore $Z|X = x \sim Gamma(2, ((1+(x-\theta)^2)/2)$, and

$$\mathbb{E}_{Z|X}[Z|X = x;\theta] = \frac{1}{1+(x-\theta)^2}.$$

To compute the EM update, we need to take the expectation of the complete data log likelihood with respect to this conditional distribution. Up to an additive constant, the complete data log-likelihood takes the form

$$-\log z - \frac{z}{2} - \frac{z}{2}(x-\theta)^2$$

and with the M-step in mind, we may work with only the terms that depend on θ, and define the $Q_i(.|.)$ function for datum i by

$$Q_i(\theta|\theta^*) = \mathbb{E}_{Z_i|X_i}\left[-Z_i(X_i-\theta)^2|X_i = x_i;\theta^*\right] = -\frac{(x_i-\theta)^2}{1+(x_i-\theta^*)^2}$$

and the entire $Q(.|.)$ function as

$$Q(\theta|\theta^*) = \sum_{i=1}^n Q_i(\theta|\theta^*) = -\sum_{i=1}^n \frac{(x_i-\theta)^2}{1+(x_i-\theta^*)^2} = -\sum_{i=1}^n w(x_i;\theta^*)(x_i-\theta)^2$$

say. We seek to maximize this wrt θ for fixed θ^*, and by elementary calculus (or sums of squares decomposition methods) we note that the maximizing value is

$$\widehat{\theta} = \frac{\sum\limits_{i=1}^n w(x_i;\theta^*)x_i}{\sum\limits_{i=1}^n w(x_i;\theta^*)}$$

This results yields the EM recursion. Note that the score equation in equation (1) gives an indication that such a recursion will result as

$$\dot{\ell}_n(\theta) = \sum_{i=1}^n \frac{(x_i-\theta)}{(1+(x_i-\theta)^2)} = \sum_{i=1}^n w(x_i,\theta)(x_i-\theta) = 0$$

is the score equation that would result from a weighted least squares estimation of θ, which we may then solve using *iteratively reweighted* least squares.

Convergence is a little slower for this algorithm:

```
nsteps<-35
th.EM<-rep(0,nsteps)
th.EM[1]<-mean(x)
print(c(0,th.EM[1]))

+ [1] 0.00000 4.97625

for(j in 2:nsteps){
        wv<-1/(1+(x-th.EM[j-1])^2)
        th.EM[j]<-sum(wv*x)/sum(wv)
        print(c(j-1,th.EM[j]))
}

+ [1] 1.000000 5.238714
+ [1] 2.000000 5.445281
+ [1] 3.000000 5.601301
+ [1] 4.000000 5.715487
+ [1] 5.000000 5.795414
+ [1] 6.000000 5.848897
+ [1] 7.000000 5.883418
+ [1] 8.000000 5.905142
+ [1] 9.000000 5.918587
+ [1] 10.00000  5.92682
+ [1] 11.00000  5.93183
+ [1] 12.000000  5.934865
+ [1] 13.0000  5.9367
+ [1] 14.000000  5.937807
+ [1] 15.000000  5.938475
+ [1] 16.000000  5.938878
+ [1] 17.00000  5.93912
+ [1] 18.000000  5.939266
+ [1] 19.000000  5.939354
+ [1] 20.000000  5.939407
+ [1] 21.000000  5.939439
+ [1] 22.000000  5.939459
+ [1] 23.00000  5.93947
+ [1] 24.000000  5.939477
+ [1] 25.000000  5.939481
+ [1] 26.000000  5.939484
+ [1] 27.000000  5.939485
+ [1] 28.000000  5.939486
+ [1] 29.000000  5.939487
+ [1] 30.000000  5.939487
+ [1] 31.000000  5.939487
+ [1] 32.000000  5.939487
+ [1] 33.000000  5.939488
+ [1] 34.000000  5.939488
```

Correct specification: Under correct specification, ML theory suggests that the distribution of the ML estimator will be asymptotically normal, with variance given by the inverse Fisher information, that is, here

$$\sqrt{n}(\widehat{\theta}_n - \theta_0) \xrightarrow{d} Normal(0, 2)$$

We can test this in the following simulation with $n = 20$; we will use Fisher scoring for estimation, but only allow the recursive calculations to run for 1000 steps; if convergence – according to the criterion

$$|\widehat{\theta}^{(t+1)} - \widehat{\theta}^{(t)}| < 1\mathrm{e}^{-6}$$

– has not been achieved by 1000 steps, we instead use direct maximization.

In this simulation we inspect 5000 data sets of size 20, and plot the distribution of the standardized estimator

$$\frac{\sqrt{n}(\widehat{\theta}_n - \theta_0)}{\sqrt{2}}$$

which should be distributed as $Normal(0,1)$ under the asymptotic result.

```
N<-5000    #Number of replicates
n<-20      #Number of data points
ml.est<-iter.count<-flag<-rep(0,N)
set.seed(12378)
for(i in 1:N){
        x<-rcauchy(n)+theta0
    theta.scoring<-mean(x)
    tol<-1e-6; eps<-1
    while(eps > tol & iter.count[i] <= 1000){
       iter.count[i]<-iter.count[i]+1
       th<-theta.scoring
       l1<-2*sum((x-th)/(1+(x-th)^2))
       theta.scoring<-theta.scoring+2*l1/n
       eps<-abs(theta.scoring-th)
    }
    if(eps > tol){  #If Fisher scoring does not converge, use direct maximization
        flag[i]<-1
        thvec<-seq(-10,10,length=10001)
        log.pdf.mat<-outer(thvec,x,log.pdf.func)
        log.like.vec<-apply(log.pdf.mat,1,sum)
        theta.scoring<-thvec[which.max(log.like.vec)]
    }
    ml.est[i]<-theta.scoring
}
print(paste('Number of Fisher scoring failures is',sum(flag),'out of',N))

+ [1] "Number of Fisher scoring failures is 74 out of 5000"

print(summary(iter.count[iter.count <= 1000]))

+    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
+    3.00    8.00   12.00   23.59   19.00  982.00
```

The Fisher scoring algorithm fails relatively rarely, only 74 times out of 5000. The median number of recurrence steps before diagnosed convergence is 12.

The summary statistics from the standardized estimates indicate that the asymptotic result is quite supported even at $n = 20$, although the tail behaviour of the sampling distribution is heavier than Normal.

```
th.std<-sqrt(n)*(ml.est-theta0)/sqrt(2)
mean(th.std);var(th.std)                                          #Moments

+ [1] 0.06347704
+ [1] 31.91893

pvec<-c(0.01,0.025,0.25,0.5,0.975,0.99)
qmat<-rbind(quantile(th.std,prob=pvec),qnorm(pvec))               #Quantiles
row.names(qmat)<-c('Empirical','Asymptotic')
qmat

+                   1%      2.5%        25%        50%    97.5%      99%
+ Empirical  -2.581506 -2.183927 -0.6435583 0.02655388 2.115986 2.552065
+ Asymptotic -2.326348 -1.959964 -0.6744898 0.00000000 1.959964 2.326348
```

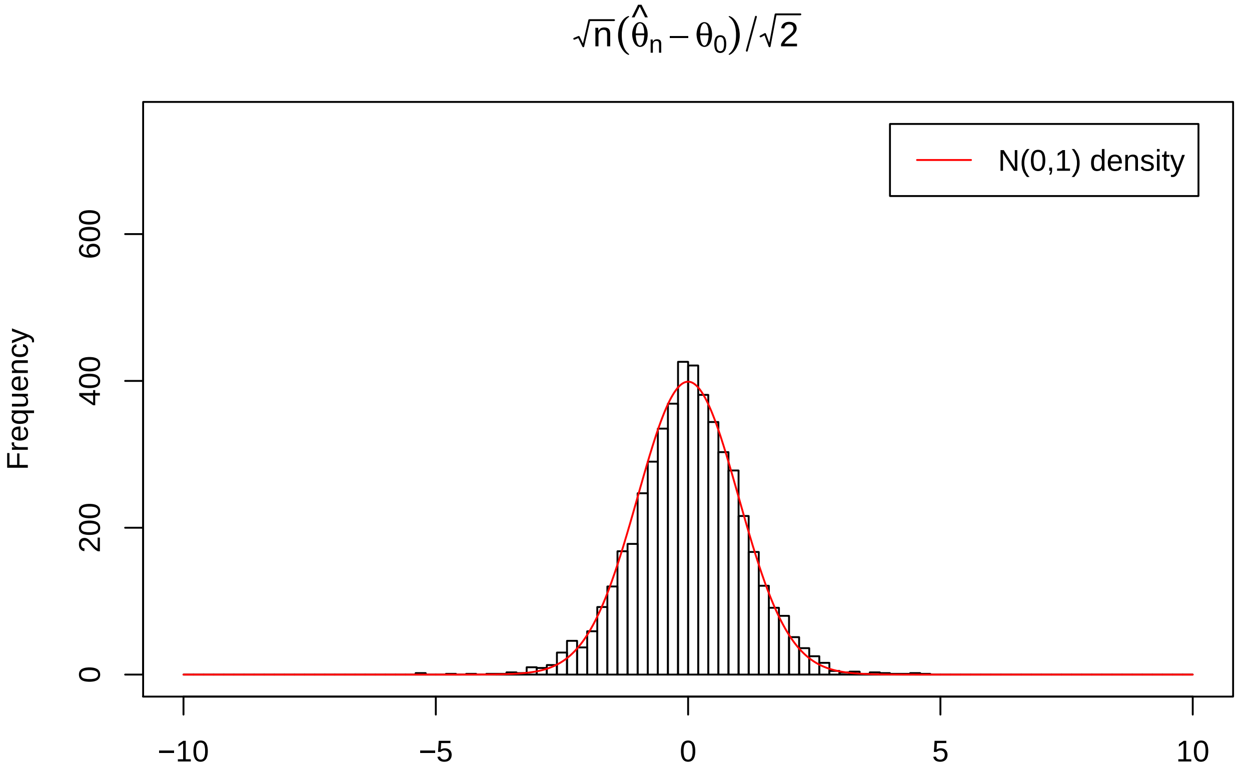


Figure 1: Distribution of standardized estimator under correct specification

```
par(mar=c(2,4,4,2))
hist(th.std[abs(th.std)<10],breaks=seq(-10,10,by=0.2),ylim=range(0,750),
main=expression(sqrt(n)(hat(theta)[n]-theta[0])/sqrt(2)));box()
tvec<-seq(-10,10,length=1001);dvec<-dnorm(tvec);lines(tvec,dvec*N*0.2,col='red')
legend(4,750,c('N(0,1) density'),col='red',lty=1)
```

Incorrect specification: We may also study performance for a misspecified model; suppose that the data are in fact generated from a $Normal(5,1)$ density. In this case, we proceed with Fisher Scoring **as if** the correct model was being fitted. However, the asymptotic distribution now takes the form

$$\sqrt{n}(\widehat{\theta}_n - \theta_0) \xrightarrow{d} Normal(0, \{\mathcal{I}_{f_0}(\theta_0)\}^{-1}\mathcal{J}_{f_0}(\theta_0)\{\mathcal{I}_{f_0}(\theta_0)\}^{-1})$$

where

$$\mathcal{I}_{f_0}(\theta) = \mathbb{E}_{f_0}\left[-\frac{d^2}{d\theta^2}\{\log f_X(X;\theta)\}\right] \quad \mathcal{J}_{f_0}(\theta) = \mathbb{E}_{f_0}\left[\left(\frac{d}{d\theta}\{\log f_X(X;\theta)\}\right)^2\right]$$

We first need to compute the “true” θ_0, that is, the value of θ that minimizes the KL-divergence between the $Cauchy(\theta)$ and the $Normal(5,1)$. We have

$$KL(N(5,1), Cauchy(\theta)) = \text{constant} + \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \log(1+(x-\theta)^2)\exp\left\{-\frac{1}{2}(x-5)^2\right\} : dx$$

Elementary calculations reveal that this function attains its minimum value when $\theta = 5$, which defines θ_0.

```
N<-5000     #Number of replicates
n<-20       #Number of data points
ml.est<-iter.count<-flag<-rep(0,N)
set.seed(12378)
for(i in 1:N){
    x<-rnorm(n)+theta0
    theta.scoring<-mean(x)
    tol<-1e-6; eps<-1
    while(eps > tol & iter.count[i] <= 1000){
       iter.count[i]<-iter.count[i]+1
       th<-theta.scoring
       l1<-2*sum((x-th)/(1+(x-th)^2))
       theta.scoring<-theta.scoring+2*l1/n
       eps<-abs(theta.scoring-th)
    }
    if(eps > tol){  #If Fisher scoring does not converge, use direct maximization
        flag[i]<-1
        thvec<-seq(-10,10,length=10001)
        log.pdf.mat<-outer(thvec,x,log.pdf.func)
        log.like.vec<-apply(log.pdf.mat,1,sum)
        theta.scoring<-thvec[which.max(log.like.vec)]
    }
    ml.est[i]<-theta.scoring
}
print(paste('Number of Fisher scoring failures is',sum(flag),'out of',N))

+ [1] "Number of Fisher scoring failures is 388 out of 5000"

print(summary(iter.count[iter.count <= 1000]))

+    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
+    3.00    9.00   15.00   31.53   28.00  898.00
```

```
th.std<-sqrt(n)*(ml.est-theta0)/sqrt(2)
mean(th.std);var(th.std)                                              #Moments

+ [1] -0.009503763
+ [1] 0.6683596

pvec<-c(0.01,0.025,0.25,0.5,0.975,0.99)
qmat<-rbind(quantile(th.std,prob=pvec),qnorm(pvec))                  #Quantiles
row.names(qmat)<-c('Empirical','Asymptotic')
qmat

+                   1%      2.5%         25%         50%    97.5%      99%
+ Empirical  -1.988524 -1.608132 -0.5524074 0.005867677 1.583577 1.901539
+ Asymptotic -2.326348 -1.959964 -0.6744898 0.000000000 1.959964 2.326348
```

```
par(mar=c(2,4,4,2))
hist(th.std[abs(th.std)<10],breaks=seq(-10,10,by=0.2),ylim=range(0,750),
main=expression(sqrt(n)(hat(theta)[n]-theta[0])/sqrt(2)));box()
tvec<-seq(-10,10,length=1001);dvec<-dnorm(tvec);lines(tvec,dvec*N*0.2,col='red')
legend(4,750,c('N(0,1) density'),col='red',lty=1)
```

This plot indicates that the asymptotic distribution is in fact correctly centered at zero, but the asymptotic variance computed **assuming correct specification** not accurate. To adjust the variance, we must use the empirical variance estimates based on the estimates

$$\widehat{I}_n(\widehat{\theta}_n) = -\frac{1}{n}\sum_{i=1}^n \frac{d^2}{d\theta^2}\{\log f_X(x_i;\theta)\}_{\theta=\widehat{\theta}_n} = -\frac{2}{n}\sum_{i=1}^n \frac{(x_i-\widehat{\theta}_n)^2-1}{(1+(x_i-\widehat{\theta}_n)^2)^2}$$

and

$$\widehat{J}_n(\widehat{\theta}_n) = \frac{1}{n}\sum_{i=1}^n \left(\frac{d}{d\theta}\{\log f_X(x_i;\theta)\}_{\theta=\widehat{\theta}_n}\right)^2 = \frac{4}{n}\sum_{i=1}^n \frac{(x_i-\widehat{\theta}_n)^2}{(1+(x_i-\widehat{\theta}_n)^2)^2}$$

We need to compute these quantities for each replication.

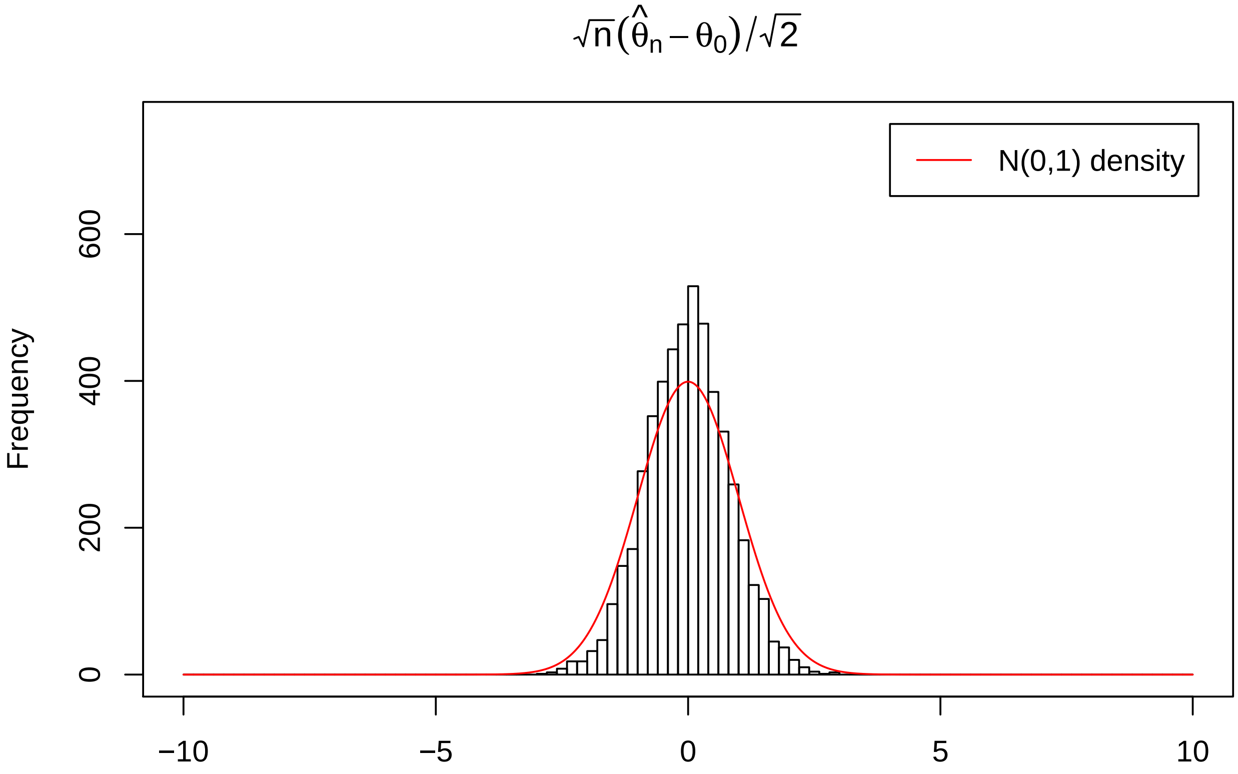


Figure 2: Distribution of standardized estimator under incorrect specification

```r
N<-5000    #Number of replicates
n<-20      #Number of data points
ml.est<-var.est<-iter.count<-flag<-rep(0,N)
set.seed(12378)
for(i in 1:N){
    x<-rnorm(n)+theta0
    theta.scoring<-mean(x)
    tol<-1e-6; eps<-1
    while(eps > tol & iter.count[i] <= 1000){
       iter.count[i]<-iter.count[i]+1
       th<-theta.scoring
       l1<-2*sum((x-th)/(1+(x-th)^2))
       theta.scoring<-theta.scoring+2*l1/n
       eps<-abs(theta.scoring-th)
    }
    if(eps > tol){  #If Fisher scoring does not converge, use direct maximization
        flag[i]<-1
        thvec<-seq(-10,10,length=10001)
        log.pdf.mat<-outer(thvec,x,log.pdf.func)
        log.like.vec<-apply(log.pdf.mat,1,sum)
        theta.scoring<-thvec[which.max(log.like.vec)]
    }
    Jhat<- 4*mean((x-theta.scoring)^2/(1+(x-theta.scoring)^2)^2)
    Ihat<--2*mean(((x-theta.scoring)^2-1)/((1+(x-theta.scoring)^2))^2)

    ml.est[i]<-theta.scoring
    var.est[i]<-Jhat/Ihat^2
}
```

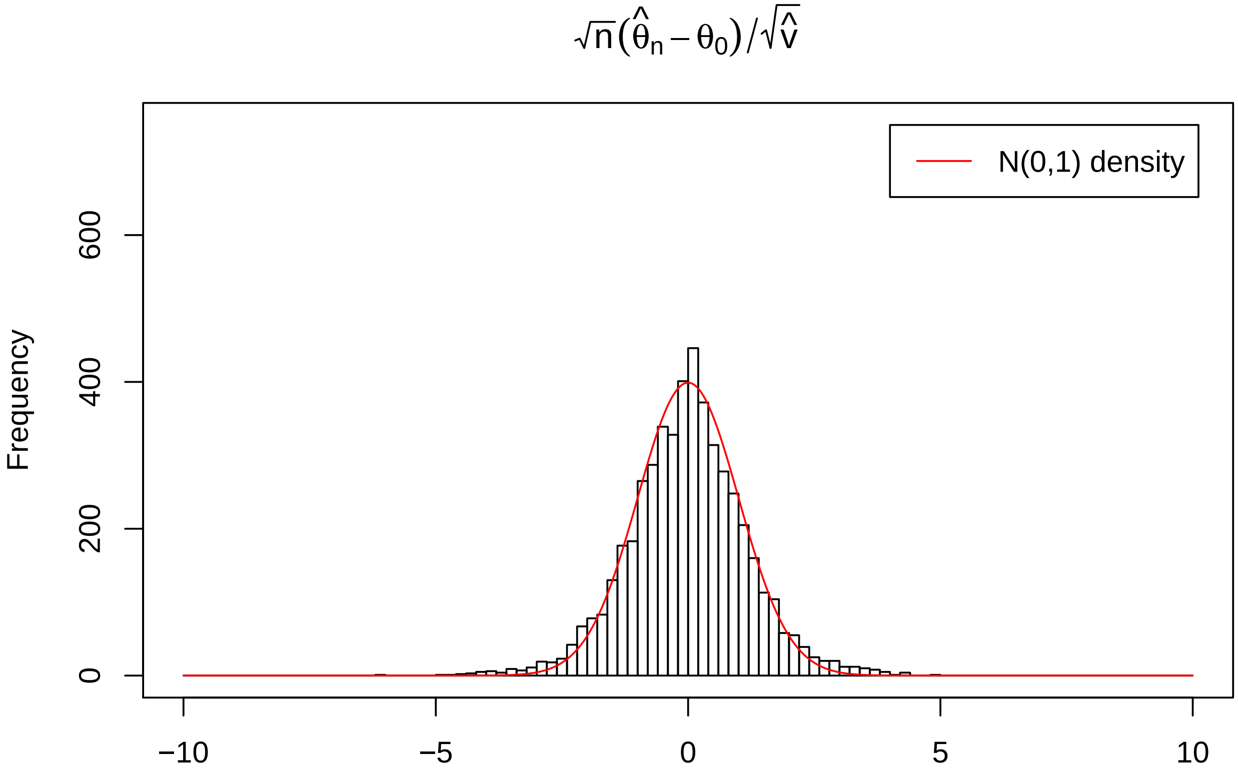


Figure 3: Distribution of standardized estimator under incorrect specification using corrected variance estimate

```
th.std<-sqrt(n)*(ml.est-theta0)/sqrt(var.est)
mean(th.std);var(th.std)                                            #Moments

+ [1] -0.01128697
+ [1] 1.354052

pvec<-c(0.01,0.025,0.25,0.5,0.975,0.99)
qmat<-rbind(quantile(th.std,prob=pvec),qnorm(pvec))                 #Quantiles
row.names(qmat)<-c('Empirical','Asymptotic')
qmat

+                    1%      2.5%        25%         50%    97.5%      99%
+ Empirical  -2.996245 -2.335196 -0.7083672 0.007724191 2.350263 3.030239
+ Asymptotic -2.326348 -1.959964 -0.6744898 0.000000000 1.959964 2.326348
```

The asymptotic approximation is now better.

```
par(mar=c(2,4,4,2))
hist(th.std[abs(th.std)<10],breaks=seq(-10,10,by=0.2),ylim=range(0,750),
main=expression(sqrt(n)(hat(theta)[n]-theta[0])/sqrt(hat(v))));box()
tvec<-seq(-10,10,length=1001);dvec<-dnorm(tvec);lines(tvec,dvec*N*0.2,col='red')
legend(4,750,c('N(0,1) density'),col='red',lty=1)
```