42111 Homework 4

Matthew DeSantis

2023-03-02

1

(a)

6.1.8 a: to find the mle, we maximize the log density function, for a Cauchy distribution, that means maximize

$$\sum_{i=1}^{n} \ln\left(\frac{1}{\pi}\cdot\frac{1}{1+(x_i-\theta)^2}\right) \to n\ln\left(\frac{1}{\pi}\right) + \sum_{i=1}^{n} \ln\left(\frac{1}{1+(x_i-\theta)^2}\right)$$

because $n\ln\left(\frac{1}{\pi}\right)$ has nothing to do with θ, we can drop it when maximizing, leaving us with

$$\sum_{i=1}^{n} \ln\left(\frac{1}{1+(x_i-\theta)^2}\right) \to -\sum_{i=1}^{n} \ln(1+(x_i-\theta))^2.$$

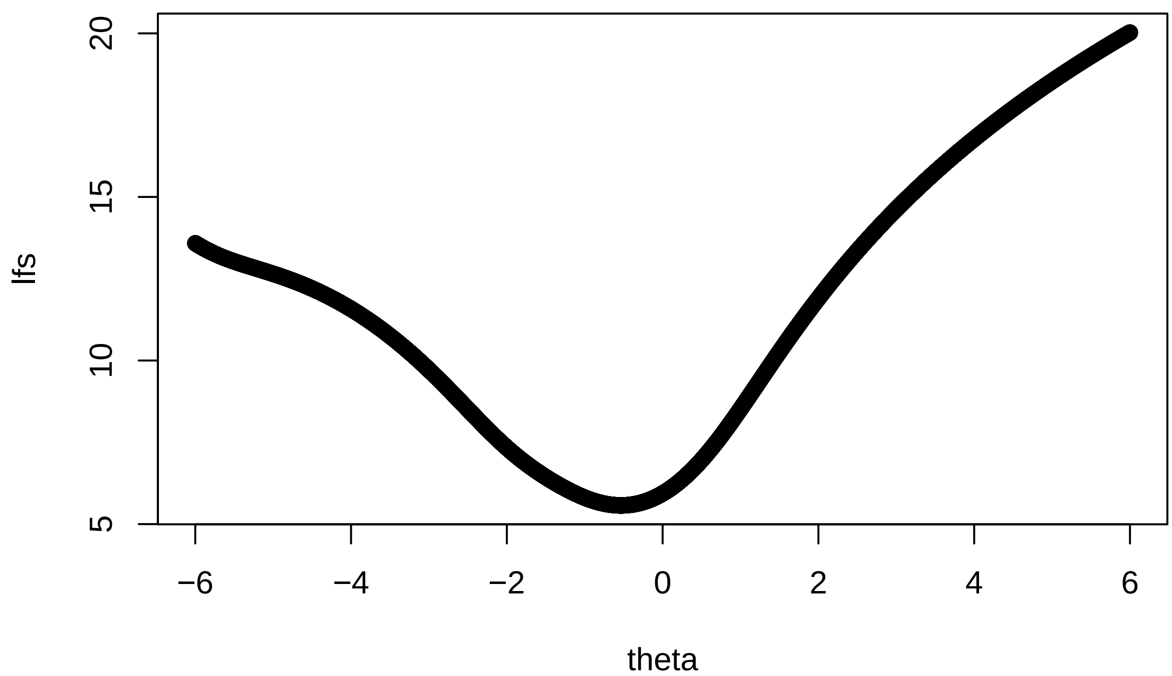
Maximizing this negative sum is clearly the same as minimizing it multiplied by -1, so minimize $\sum_{i=1}^{5} \ln\left(1+(x_i-\theta)^2\right)$

($n=5$ in this case.)

Figure 1: Work for 1a

(b)

```
x = c(-1.94, 0.59, -5.98, -0.08, -0.77)
theta=seq(-6,6,.001);lfs<-c()
for(th in theta){lfs=c(lfs,sum(log((x-th)^2+1)))}
plot(lfs~theta)
```




```
theta[which.min(lfs)]
```

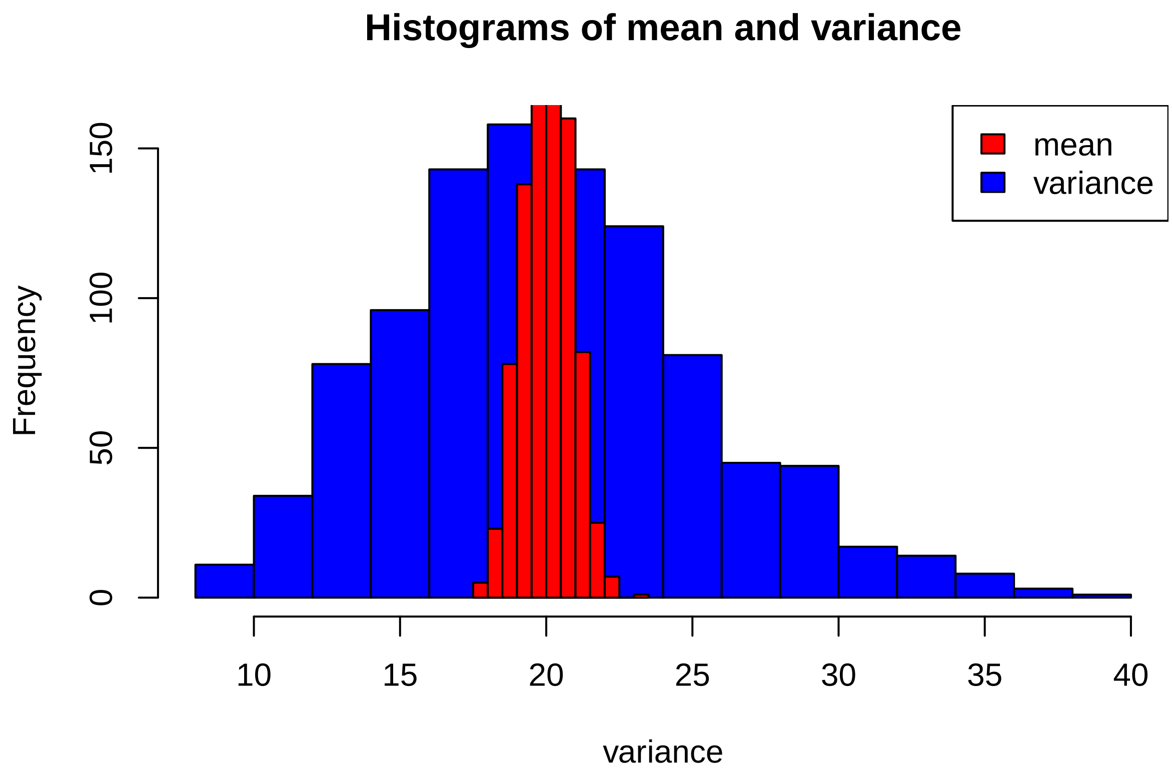
```
## [1] -0.534
```

2

```
comp2 = function(k = 1000, n = 30, l = 20, hist = TRUE){
mean = c()
variance = c()
intervals = matrix(ncol = 2, nrow = 0)
intervals2 = matrix(ncol = 2, nrow = 0)
for (x in 1:k) {
  ran = rpois(n,l)

  upper = mean(ran) + 1.96*sd(ran)/sqrt(n)
  lower = mean(ran) - 1.96*sd(ran)/sqrt(n)

  upper2 = mean(ran) + 1.96*sqrt(mean(ran))/sqrt(n)
  lower2 = mean(ran) - 1.96*sqrt(mean(ran))/sqrt(n)

  intervals = rbind(intervals, c(lower, upper))
  intervals2 = rbind(intervals2, c(lower2, upper2))

  mean = c(mean, mean(ran))
  variance = c(variance, var(ran))
}
msemean = (mean(mean)-l)^2+var(mean)
msevar = (mean(variance)-l)^2+var(variance)

if (hist){
  hist(variance, col = 'blue', breaks = 15, main = 'Histograms of mean and variance')
  hist(mean, col = 'red', add = TRUE, breaks = 15)
  legend('topright', c('mean','variance'), fill = c('red', 'blue'))
}

return (list("msemean" = msemean, "msevar" = msevar, "mean_data" = mean, "var_data" = variance, "interva
}
```

(a)

```
set.seed(488102)
a = comp2()
```



Although both histograms have roughly the same center, the histogram for variance has a drastically greater spread.

(b)

```
mean(a$mean_data)
```

```
## [1] 20.0388
```

```
var(a$mean_data)
```

```
## [1] 0.6429153
```

```
(mean(a$mean_data)-20)^2+var(a$mean_data)
```

```
## [1] 0.6444207
```

(c)

```
(mean(a$var_data)-20)^2+var(a$var_data)
```

```
## [1] 27.98503
```

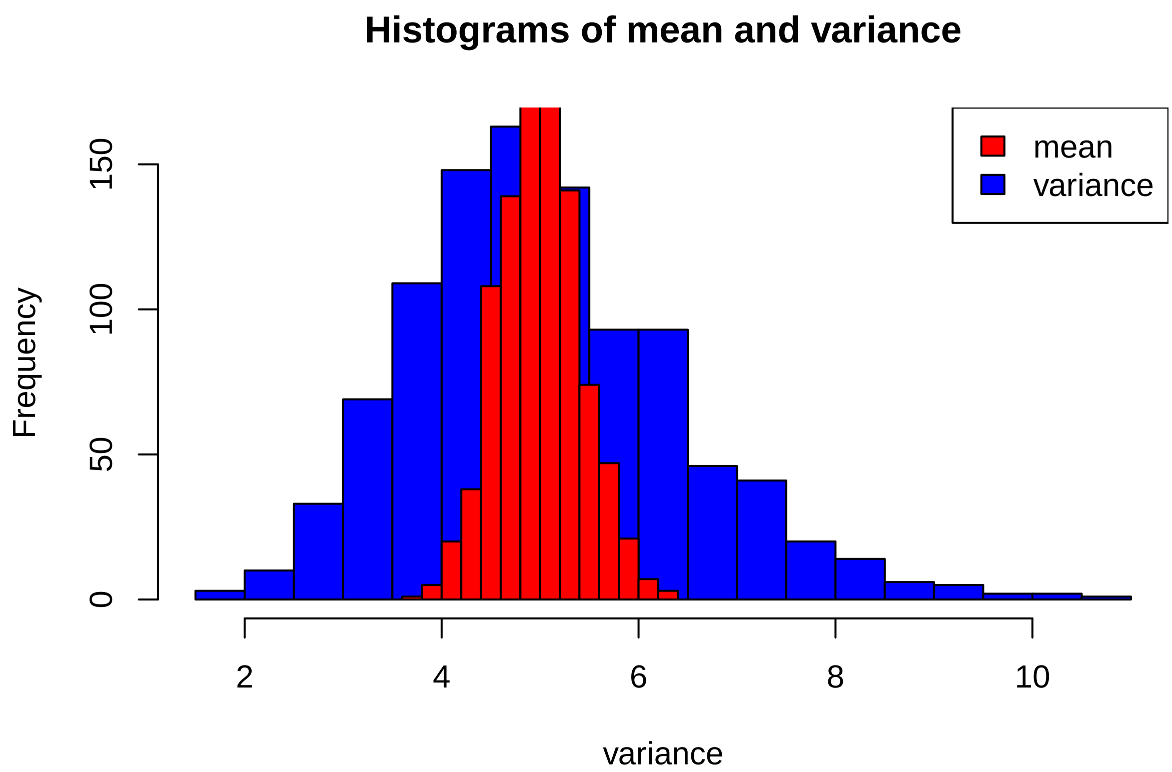
It is much higher than the MSE for mean.

(d)

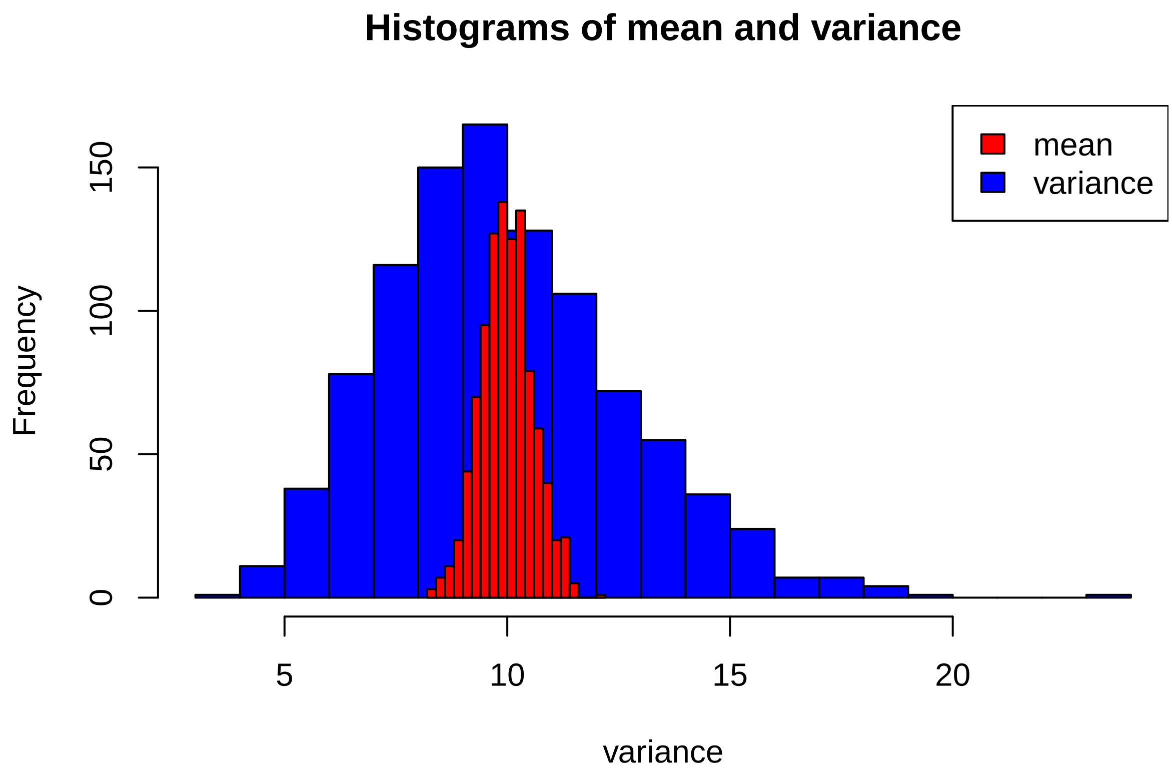
Sample mean appears to be a much better estimator, because it is unbiased and has a much smaller MSE

(e)

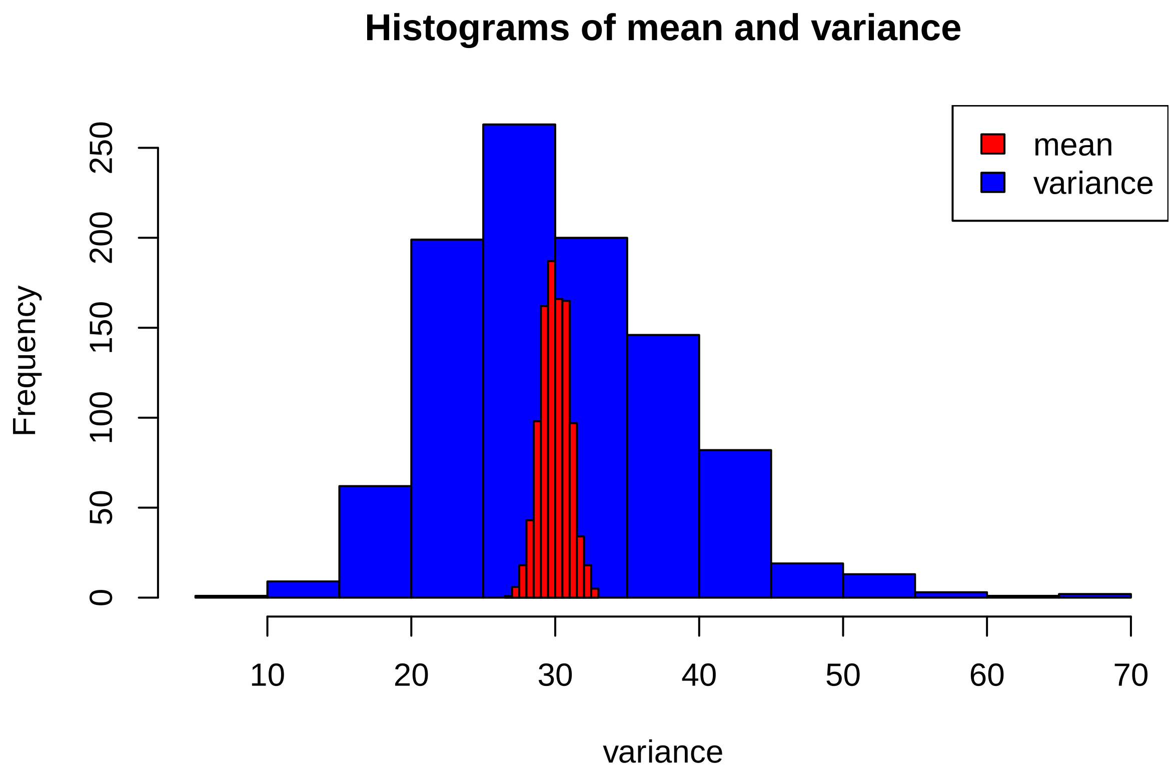
```
for (x in c(5,10,30,50,100)){
  e = comp2(l = x)
  print(c(e$msemean, e$msevar))
}
```



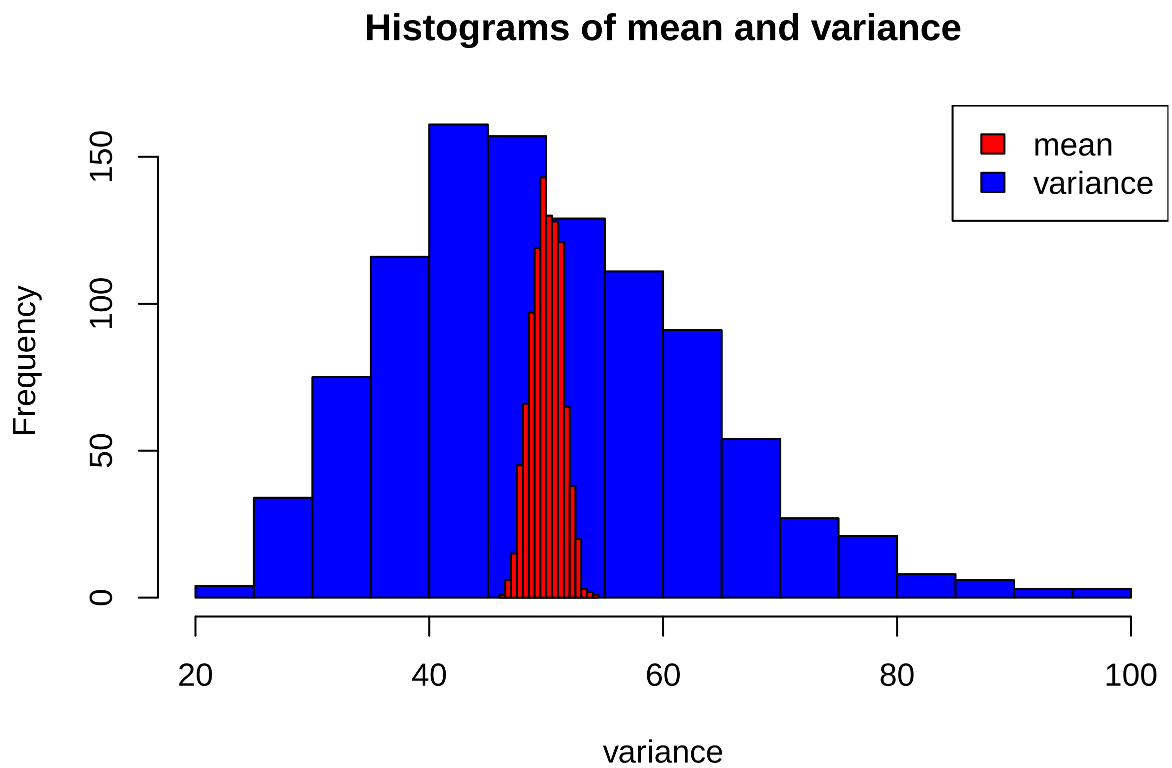

```
## [1] 0.1695517 1.8975406
```



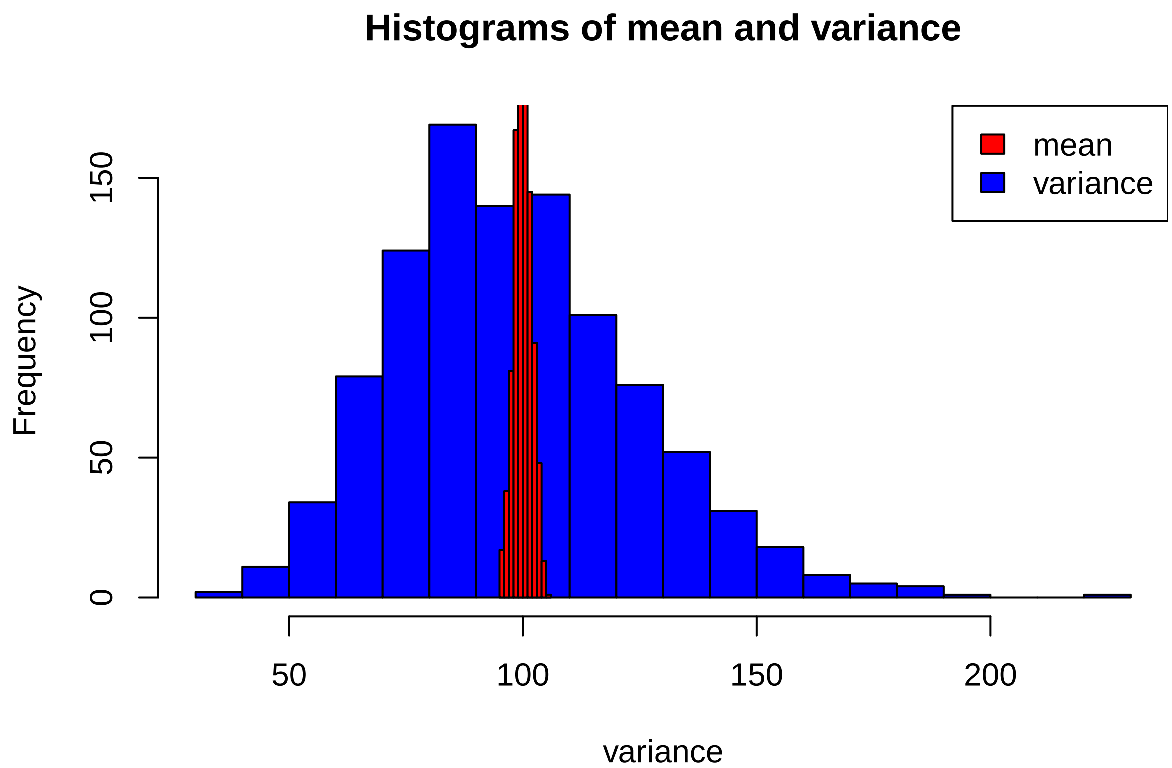

```
## [1] 0.3385251 7.1255870
```




```
## [1]  1.005859 63.455466
```




```
## [1]   1.706891 165.480061
```




```
## [1]   3.457964 686.212415
```

While sample mean is always a better estimator, it appears that the two grow closer in precision when lambda gets smaller.

3

(a)

```r
head(a$intervals, 5)
```

```
##          [,1]     [,2]
## [1,] 17.76022 21.17311
## [2,] 18.93906 21.52761
## [3,] 17.70563 20.82770
## [4,] 19.45882 22.07452
## [5,] 19.52399 23.20935
```

```r
sum(a$intervals[,1]<20 & a$intervals[,2]>20)
```

```
## [1] 950
```

Exactly 95% of them! Wow!

(b)

```r
head(a$intervals2, 5)
```

```
##          [,1]     [,2]
## [1,] 17.88782 21.04552
## [2,] 18.62369 21.84297
## [3,] 17.69595 20.83739
## [4,] 19.13595 22.39738
## [5,] 19.71256 23.02077
```

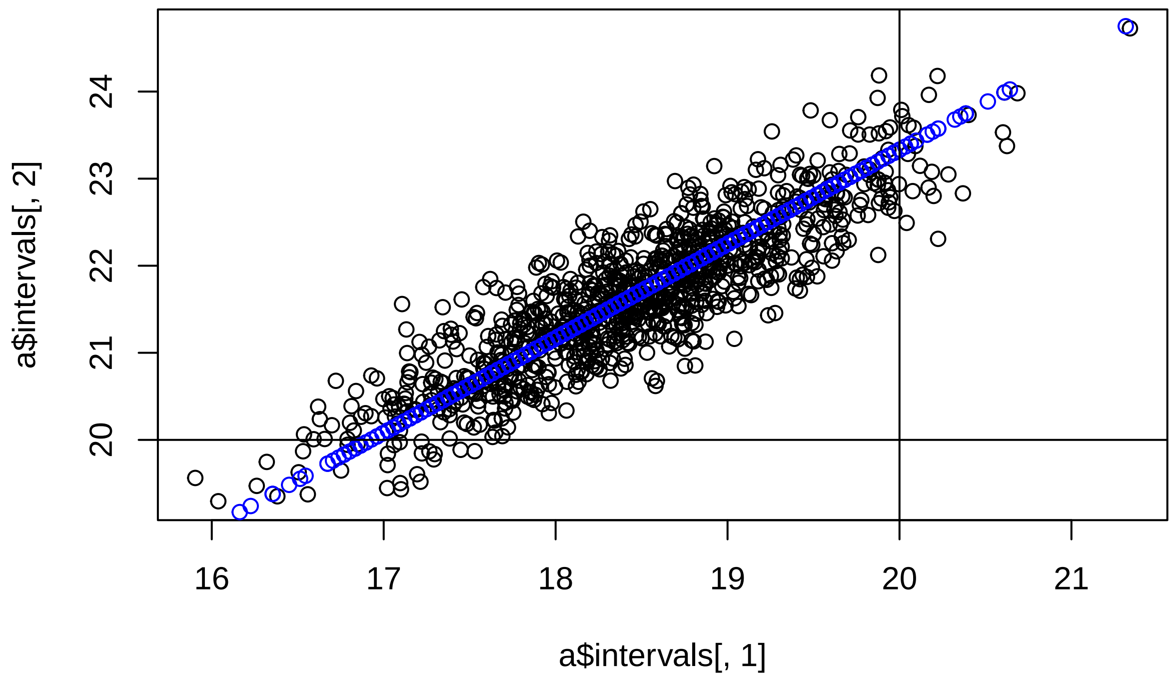
```r
sum(a$intervals2[,1]<20 & a$intervals2[,2]>20)
```

```
## [1] 954
```

Slightly more than 95% of them.

(c)

```r
plot(a$intervals[,1], a$intervals[,2], col = 'black')
points(a$intervals2[,1], a$intervals2[,2], col = 'blue')
abline(h = 20)
abline(v = 20)
```



The top left quadrant of the graph is the section which contains intervals which contain the true value.

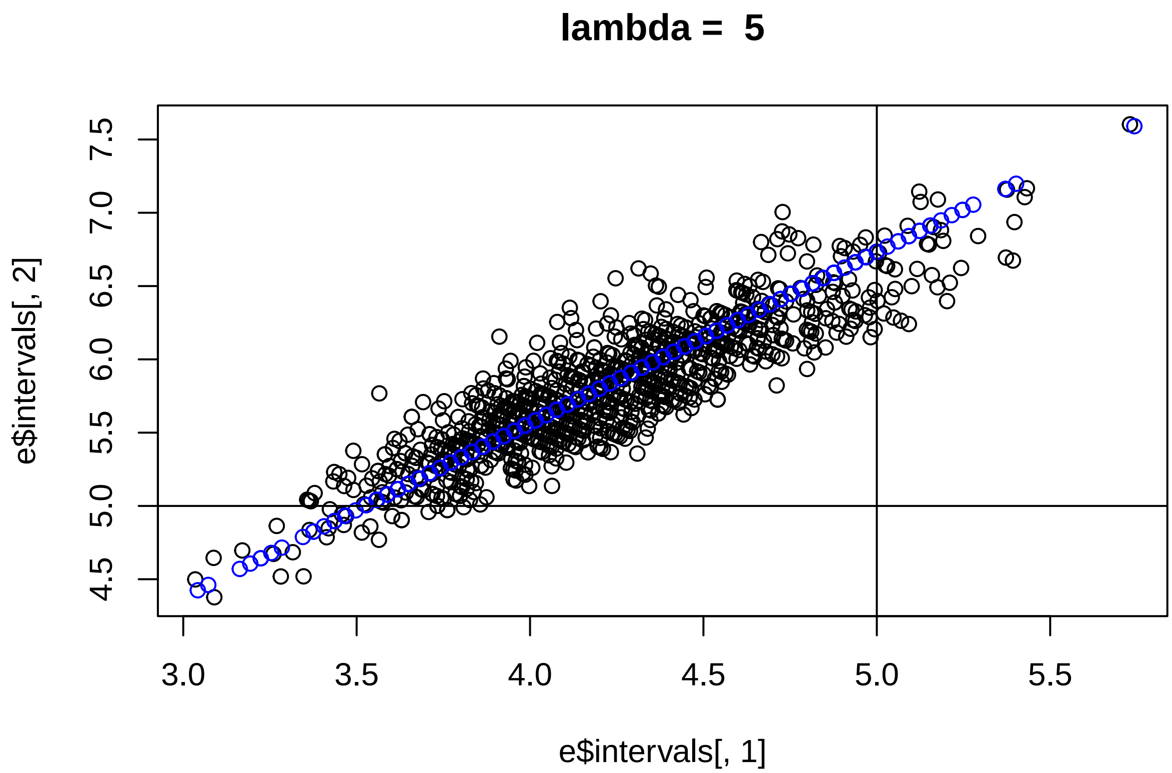
(d)

based on the graph, I would choose the confidence intervals from part (b), because they have a much smaller spread.

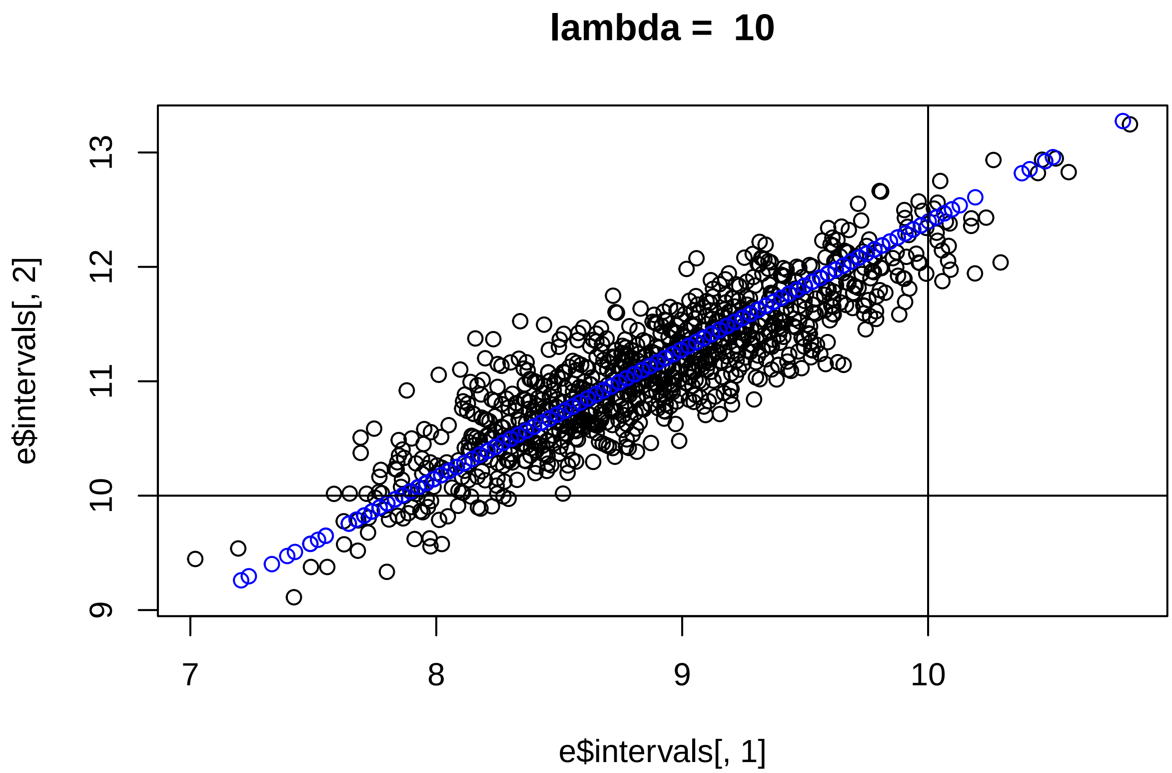
(e)

```
for (x in c(5,10,30,50,100)){
  e = comp2(l = x, hist = FALSE)
  print(c((sum(e$intervals[,1]<x & e$intervals[,2]>x)), sum(e$intervals2[,1]<x & e$intervals2[,2]>x)))
  plot(e$intervals[,1], e$intervals[,2], col = 'black', main = paste("lambda = ", as.character(x)))
  points(e$intervals2[,1], e$intervals2[,2], col = 'blue')
  abline(h = x)
  abline(v = x)
}
```

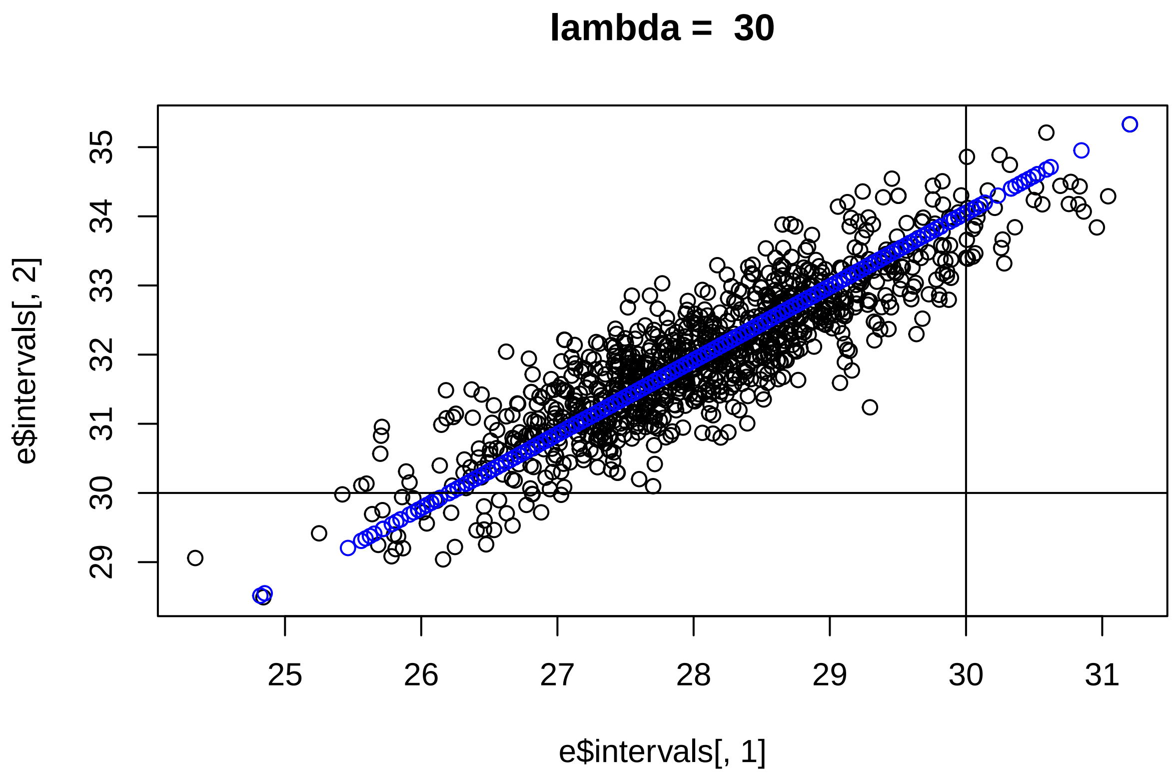
```
## [1] 938 957
```


lambda = 5
e$intervals[, 2]
e$intervals[, 1]
4.5
5.0
5.5
6.0
6.5
7.0
7.5
3.0
3.5
4.0
4.5
5.0
5.5

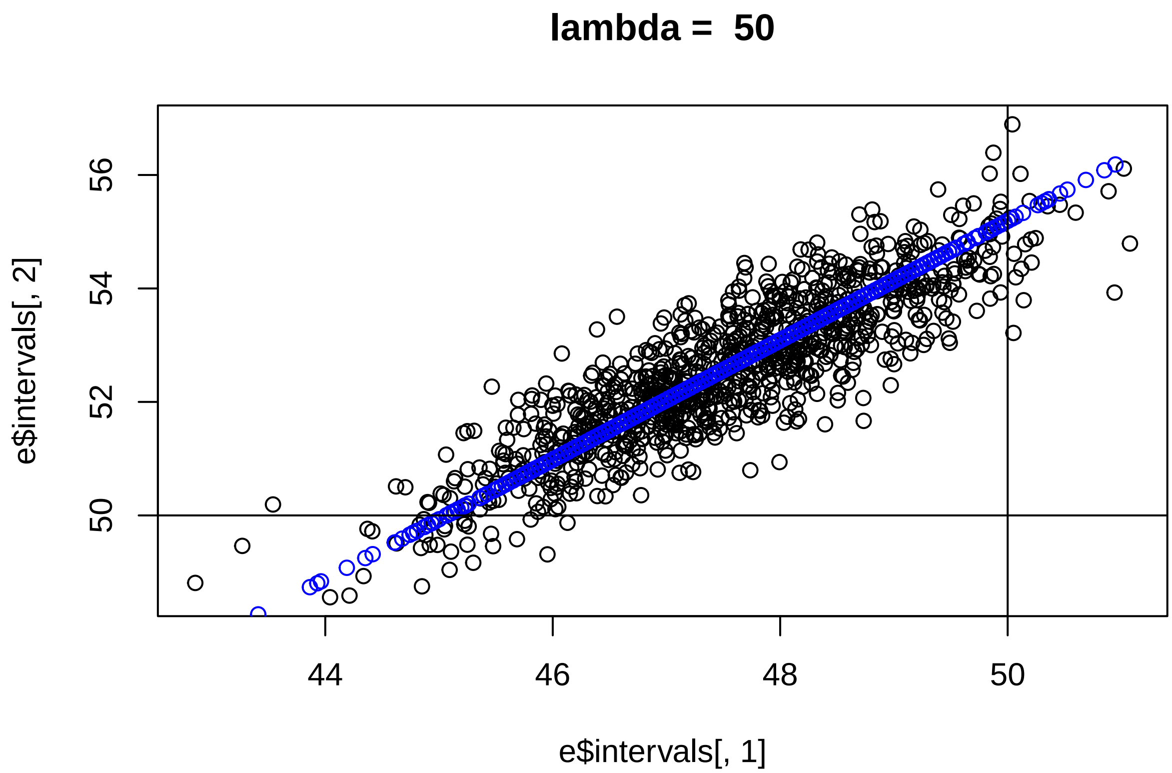
```
## [1] 938 949
```



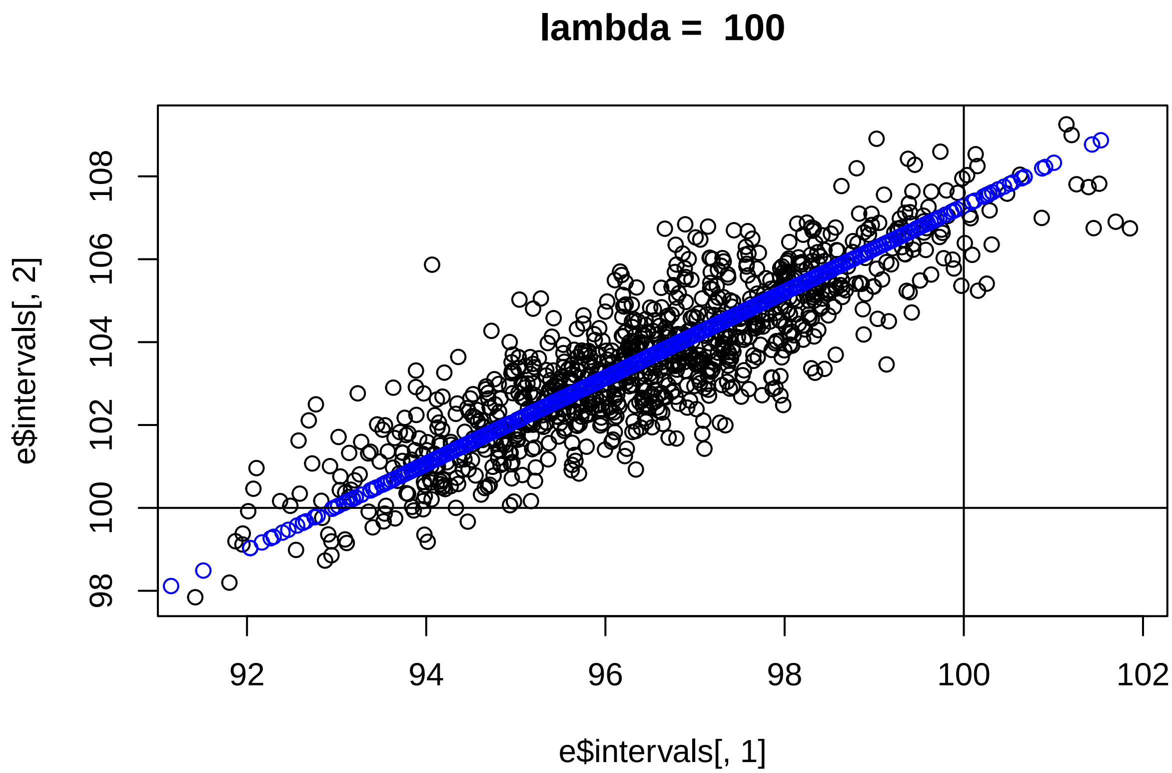

```
## [1] 933 944
```


lambda = 30
e$intervals[, 2]
e$intervals[, 1]
25
26
27
28
29
30
31
29
30
31
32
33
34
35

```
## [1] 950 962
```




```
## [1] 953 958
```



The result does not change with different values of lambda. Using the mean to estimate the standard deviation always results in better intervals.