THE UNIVERSITY OF SYDNEY

STAT3023 Statiscal Inference

Lab Week 9

Tutor: Wen Dai

SID: 470408326

School of Mathematics and Statistics

The University of Sydney

Semester 2, 2021

1. Obtain the two interval estimates based on the observation of X = 8. Call them `simple` and `bayes`.

Solution

```
X = 8
simple = c(X - 1, X + 1)
bayes = c(2, 2 * exp(2/X))/(exp(2/X) - 1)
simple
```

```
[1] 7 9
```

```
bayes
```

```
[1] 7.041623 9.041623
```

2. We shall now write functions that compute these two intervals, so we can approximate their risk functions by simulation. Note that the procedure for constructing the interval is different (in *both* cases) if the Poisson observation is zero (which is certainly possible!); in both cases for an observation of zero the interval is (0, 2). Write two functions `simple()` and `bayes()`. They should both be of the same conditional form:

```
simple=function(X){
  if (X==0) {
    out=...
  } else {
    ...
    out=...
    }
  out
}
```

Once you have written them, test them out by executing both `simple(8)` and `bayes(8)`

Solution

```
simple = function(x) {
    if (x == 0) {
        out = c(0, 2)
    } else {
        out = c(x - 1, x + 1)
    }
    out
}

bayes = function(x) {
    if (x == 0) {
        out = c(0, 2)
    } else {
        out = c(2, 2 * exp(2/x))/(exp(2/x) - 1)
    }
    out
}
```

3. Feel free to edit this part according to what you think the main idea was

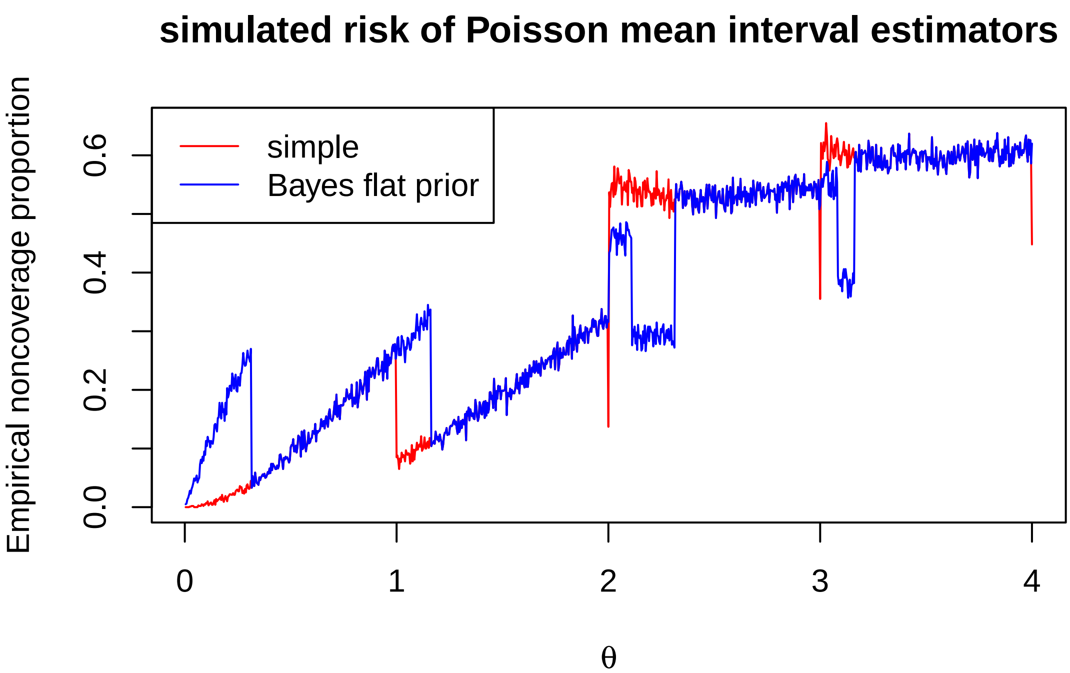
```r
th = (1:1000)/250   # specify 1000 thetas between 0 and 4
L = length(th)
B = 1000   # number of random drawings, number of intervals
noncoverage.simple = noncoverage.bayes = 0

# we fix the theta value
for (i in 1:L) {
    s.mat = matrix(0, B, 2)
    b.mat = matrix(0, B, 2)

    # we fix the row in our matrix.
    for (j in 1:B) {
        X = rpois(1, th[i])   # draw one random number from pois(th[i])
        s.mat[j, ] = simple(X)   # constructing intervals
        b.mat[j, ] = bayes(X)
    }

    # count the number of intervals not containing theta
    noncoverage.simple[i] = sum(th[i] < s.mat[, 1]) + sum(th[i] > s.mat[, 2])
    noncoverage.bayes[i] = sum(th[i] < b.mat[, 1]) + sum(th[i] > b.mat[, 2])
}

plot(th, noncoverage.simple/B, type = "l", col = "red", main = "simulated risk of Poisson mean interval estimators",
    xlab = expression(theta), ylab = "Empirical noncoverage proportion")
lines(th, noncoverage.bayes/B, type = "l", col = "blue")
legend("topleft", legend = c("simple", "Bayes flat prior"), col = c("red", "blue"), lty = c(1, 1))
```



4. In general as θ increases, the non-coverage rate increases. This means the risks of both methods will increase

For some θ ranges, the simple interval method is better, whereas for others, the Bayes interval method is better. So we compare the two methods over several intervals:

- near 0 and just above 1, the simple interval estimation performs better than the bayes interval estimation
- near 2, just above 2 and near 3 (just above 3) the simple interval method performs worse
- just below 4, there are only a few θ values for which the rate of non-coverage interval is lower than that of Bayes method. so we are not convinced that the simple one performs better near 4
- In the other ranges, the risks of the two estimators are very similar

So no one method is uniformly better than the other one.