1. 3(a)

```
data <- c(1.2, 2.4, 1.3, 1.3, 0.0, 1.0,1.8, 0.8, 4.6, 1.4)
> length(data)
[1] 10
> k <- 10
> s2 <- (1/k)*sum(((data) - mean(data))^2)
> tausq.hat <- s2 - 1
> b.hat <- 1 / (1 + tausq.hat)
> js.theta <- (1 - (k-2) / (sum( data^2)) )  * data;js.theta
 [1] 0.9511664 1.9023328 1.0304303 1.0304303 0.0000000 0.7926387 1.4267496
 [8] 0.6341109 3.6461379 1.1096941
```

2. 3(b)

```
> js.prime.theta <- mean(data) + (1 - (k-3)/(sum( (data - mean(data))^2)))*(data
 [1] 1.3953584 1.9784371 1.4439483 1.4439483 0.8122797 1.2981786 1.6868978
 [8] 1.2009988 3.0474148 1.4925382
```

3. 3 (c)

```
> center <- b.hat*mean(data) + (1 - b.hat)*data[9]
> lower <- -1.96*sqrt(1 - b.hat)
> upper <- 1.96*sqrt(1 - b.hat)
> center + lower
[1] 1.371965
> center + upper
[1] 3.392077
```

4. 3 (d)

```
> b.hat.m <- (k-3)/(k-1) * b.hat
> center.m <- b.hat.m * mean(data) + (1- b.hat.m)*data[9]
> v.hat.m <- (1 - ((k-1)/k)*b.hat.m) + (2/(k-3))*(b.hat.m^2)*(data[9] - mean(data
> lower <- -1.96*sqrt(v.hat.m)
```

```
> upper <- 1.96*sqrt(v.hat.m)
> center+lower
[1] 0.1164012
> center+upper
[1] 4.647641
```

5. 4(c) simulation:

   I run a 10000 step simulation where theta is generated anew at each step and I base the observations Y on theta. I also estimate a and b from the prior distribution at each step, since I am constructing an empirical confidence interval. Then I calculate the posterior distribution with my data, $\hat{a}$ and $\hat{b}$. This estimated distribution is then used to find the desired quantiles. After running my simulation, I find that the actual coverage probability is less than 90%.

```
theta.true <- c(rep(0,m))
k <- 5
a<-3;b<-3
m <- 10000
count <- c(rep(0,m))
data <-matrix( c(rep(0,k*m)), nrow = m)
post.data<-matrix( c(rep(0,5*5000)), nrow = m)
a.hat <- c(rep(0,m))
b.hat <- c(rep(0,m))
lower <- c(rep(0,m))
upper <- c(rep(0,m))

for (i in 1:m){

    theta.true[i] <- rgamma(1,shape = 3, scale = 3)

    data[i,] <-rpois(5,theta.true[i])

    a.hat[i] <- (mean(data[i,]))^2 / (sum ( (data - mean(data[i,]))^2 ) -  mean(

    b.hat[i] <- mean(data[i,]) / a.hat[i]


    lower[i] <- qgamma(0.05, shape = a.hat[i] + sum(data[i,]), scale = b.hat[i]

    upper[i] <- qgamma(0.95, shape = a.hat[i] + sum(data[i,]), scale = b.hat[i]
```

```
    if ((lower[i] < theta.true[i]) & (theta.true[i] < upper[i])) {count[i] = 1}

}

p <- sum(count)/m;p  #estimate
se <- sd(count)/sqrt(m); se #standard error

> p <- sum(count)/m;p  #estimate
[1] 0.898
> se <- sd(count)/sqrt(m); se #standard error
[1] 0.003026634
```