

Assignment 8

1. Returning again to the setting of Problem 2 from Assignment 7, repeat the problem using the estimates produced by

```
gam  
rpart  
gbm  
randomForest  
nnet.
```

Be sure to use appropriate simulation sample sizes and to include simulation standard errors in your report.

You should submit your assignment electronically using Icon. Submit your work as a single compressed tar file. If your work is in a directory `mywork` then you can create a compressed tar file with the command

```
tar czf mywork.tar.gz mywork
```

Solutions and Comments

1.
 - Please follow the coding standards.
 - If there are parameters you need to specify for an algorithm you should state what you used, and why, in your report.
 - For simulation studies you should include some indication of accuracy in plots and tables.
 - A simulation size of 1,000 or 2,000 for a simple problem such as this is too small. You should use at least 10,000 independent replicates unless there are compelling reasons not to.

One way to run the simulation for `gam` (this takes about 90 seconds on my laptop):

```
library(mgcv)

m <- function(x) 1 - sin(5 * x)^2 * exp(-5 * x)
n <- 50
x <- (1 : n) / (n + 1)

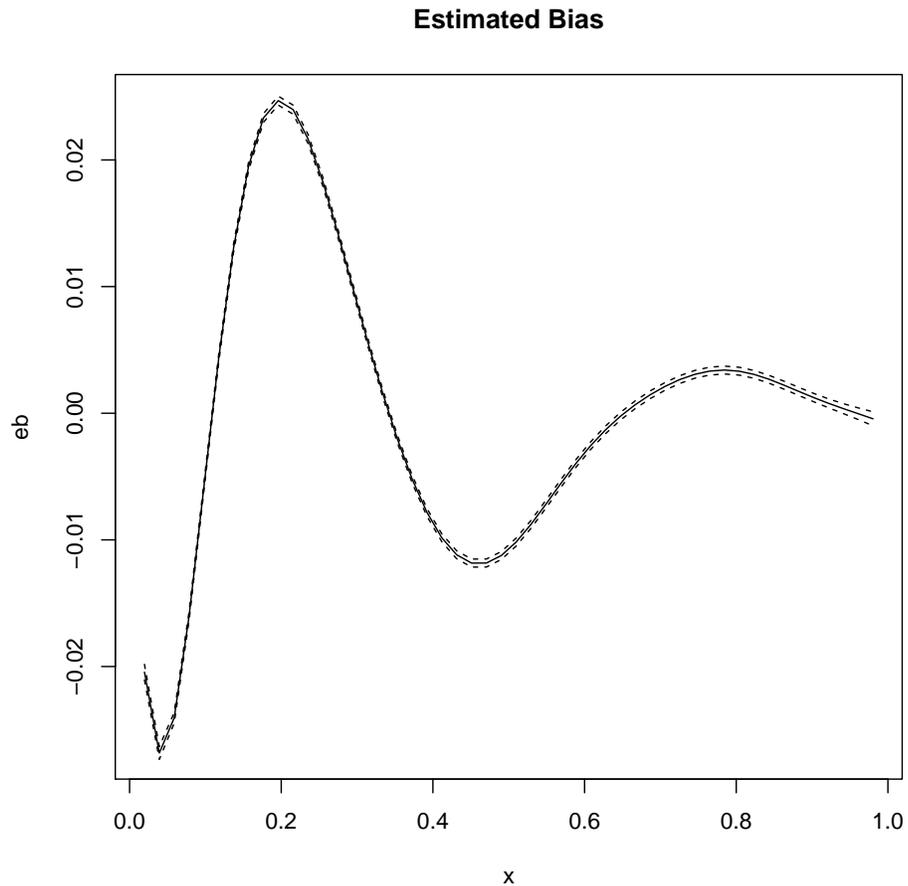
R <- 10000

sim <- function(fitfun, sigma = 0.1) {
  sapply(seq_len(R), function(i) {
    y <- rnorm(n, m(x), sigma)
    predict(fitfun(x, y))
  })
}

yhat <- sim(function(x, y) gam(y ~ s(x)))
```

A plot of the estimated bias, with dashed lines showing \pm one simulation standard error:

```
eb <- rowMeans(yhat) - m(x)
ese <- apply(yhat, 1, sd)
plot(x, eb, type = "l", main = "Estimated Bias")
lines(x, eb + ese / sqrt(R), lty = 2)
lines(x, eb - ese / sqrt(R), lty = 2)
```

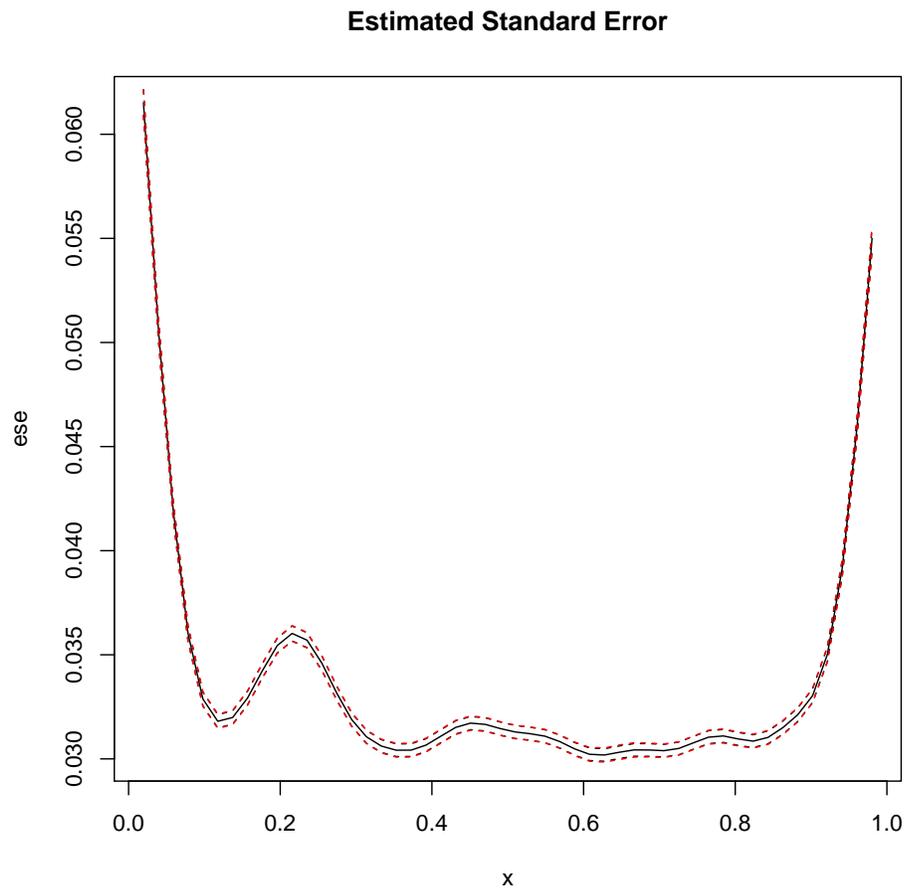


Assuming the estimates are approximately normal we can use the gamma (scaled χ^2) distribution of the sample variance and the delta method to obtain approximate simulation standard errors on the standard errors of the function estimates. If we do not use normality then we can estimate the variance of the sample variance using the variance of the squared deviations from the sample mean.

```
plot(x, ese, type = "l", main = "Estimated Standard Error")
```

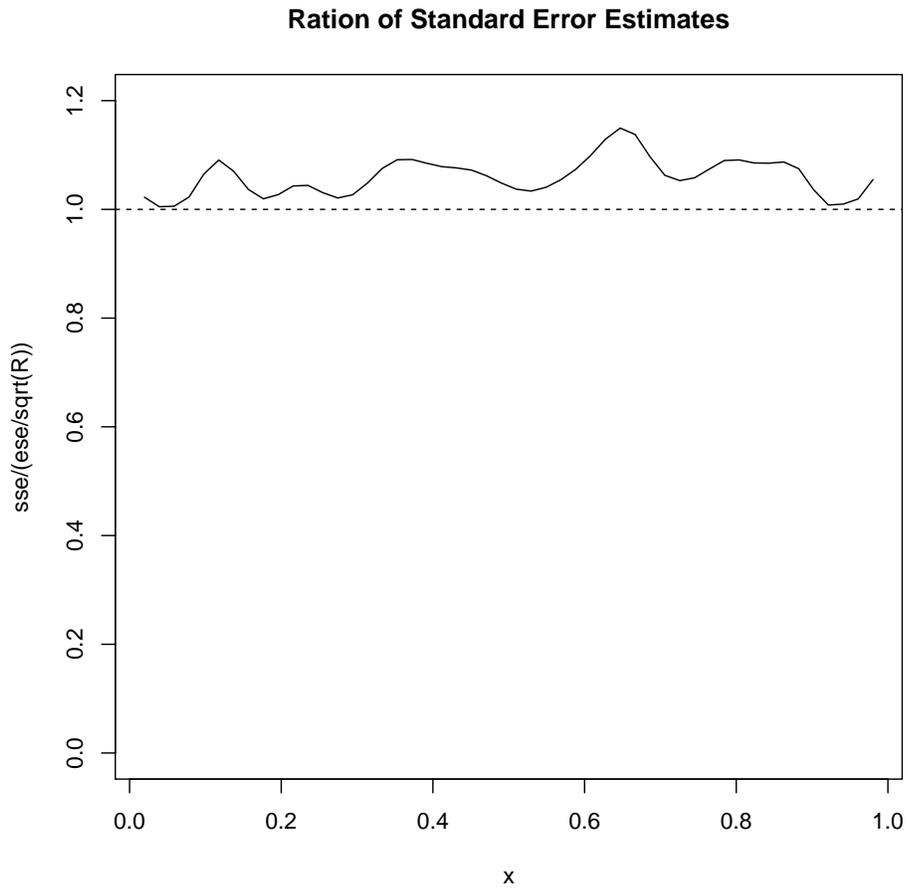
```
## Chi-square approximation with delta method
lines(x, ese * (1 + sqrt(1 / R)), lty = 2)
lines(x, ese * (1 - sqrt(1 / R)), lty = 2)
```

```
## SE of squared deviations with delta method
sse <- apply(sweep(yhat, 1, rowMeans(yhat)) ^ 2, 1, sd) / ese / sqrt(2 * R)
lines(x, ese + sse, lty = 2, col = "red")
lines(x, ese - sse, lty = 2, col = "red")
```



The estimates assuming normality are slightly smaller:

```
plot(x, sse / (ese / sqrt(R)), type = "l", ylim = c(0, 1.2),
     main = "Ration of Standard Error Estimates")
abline(h = 1, lty = 2)
```



The estimated average mean square error and the simulation standard error of the estimate:

```
> ase <- colMeans(sweep(yhat, 1, m(x)) ^ 2)
> amse <- mean(ase)
> sea <- sd(ase) / sqrt(R)
> sea / amse
[1] 0.005583625
```