# chords_bias

*Jonathan Rosenblatt*

*13/08/2014*

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

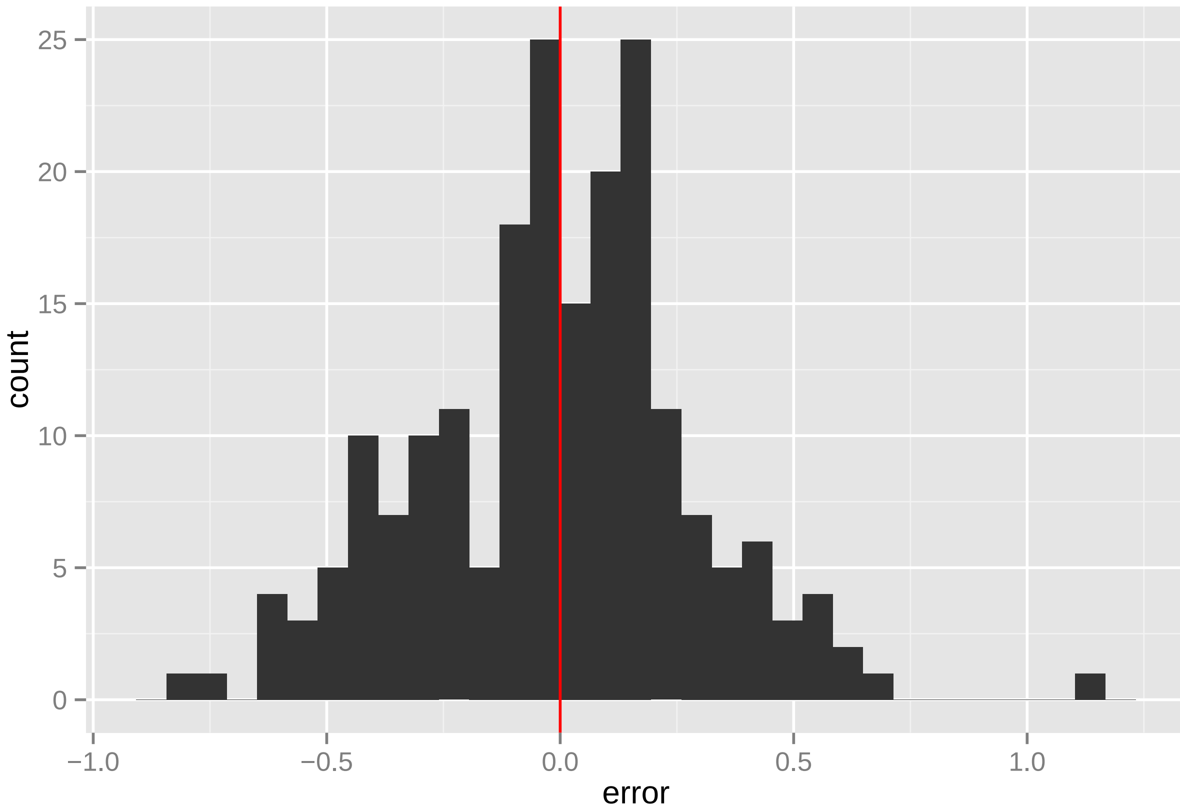
```
getError <- function (beta, theta, sample.length) {
    dk <- c(2,10)
    true.Nks <- rep(0,max(dk)); true.Nks[dk] <- 1000

    true.log.bks <- rep(-Inf, max(dk))
    true.log.bks[dk] <- log(beta)+theta*log(dk)

    rds.simulated.object <- makeRdsSample(
      N.k =true.Nks ,
      b.k = exp(true.log.bks),
      sample.length = sample.length)
    rds.simulated.object$estimates <- estimate.b.k(rds.simulated.object)
    theta- getTheta(rds.simulated.object)$theta
}
# Testing
beta <- 5e-6
theta <- 0.1
sample.length <- 800L
getError(beta, theta, sample.length )
```

```
## [1] -0.226
```

### Shape of distribution of theta:

```
nsims <- 200
error <- replicate(nsims, getError(beta, theta, sample.length ))
qplot(error, geom="histogram")+geom_vline(aes(intercept=0), col='red')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



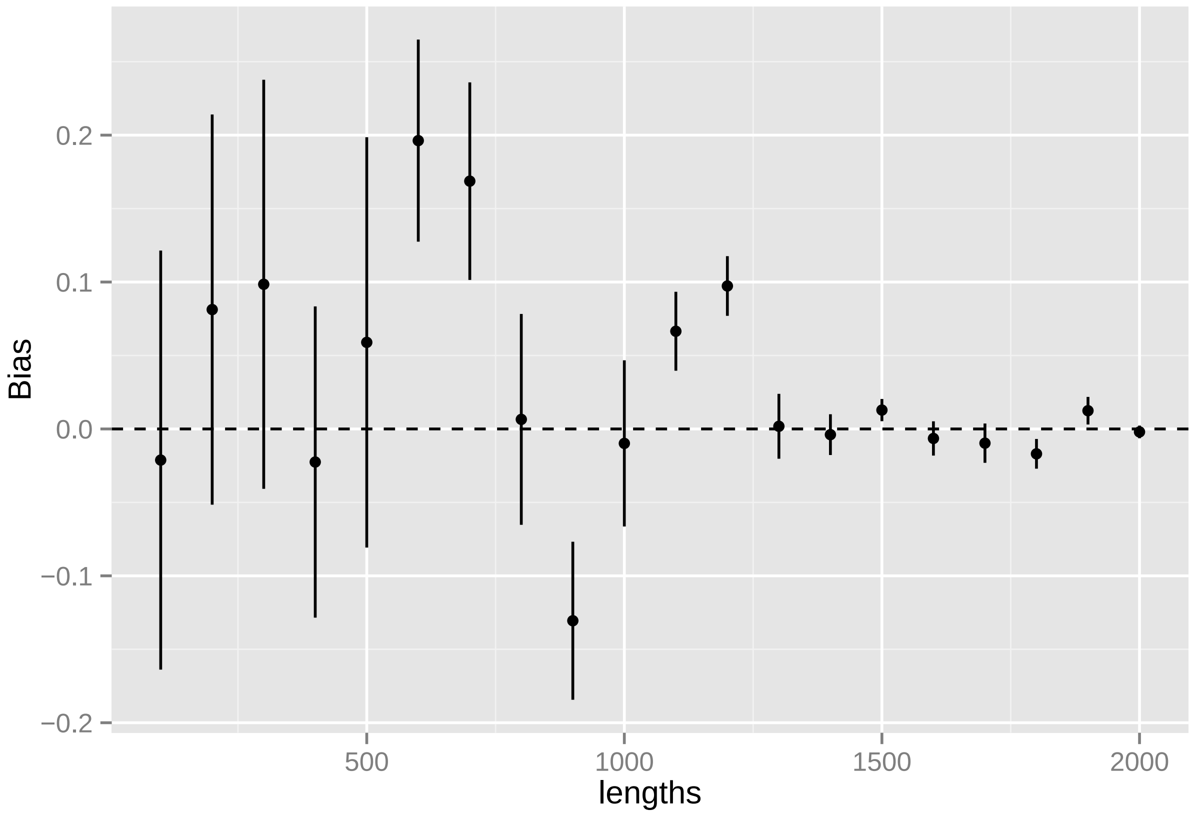
Distribution seems fairly symmetric. We can thus concentrate on the firts two moments of the errors to study the performance of the estimator.

Replicate to compute bias:

```
getBias <- function(nsims, beta, theta, sample.length){
  error <- replicate(nsims, getError(beta, theta, sample.length ))
  c(error=mean(error), sd=sd(error))
}
```

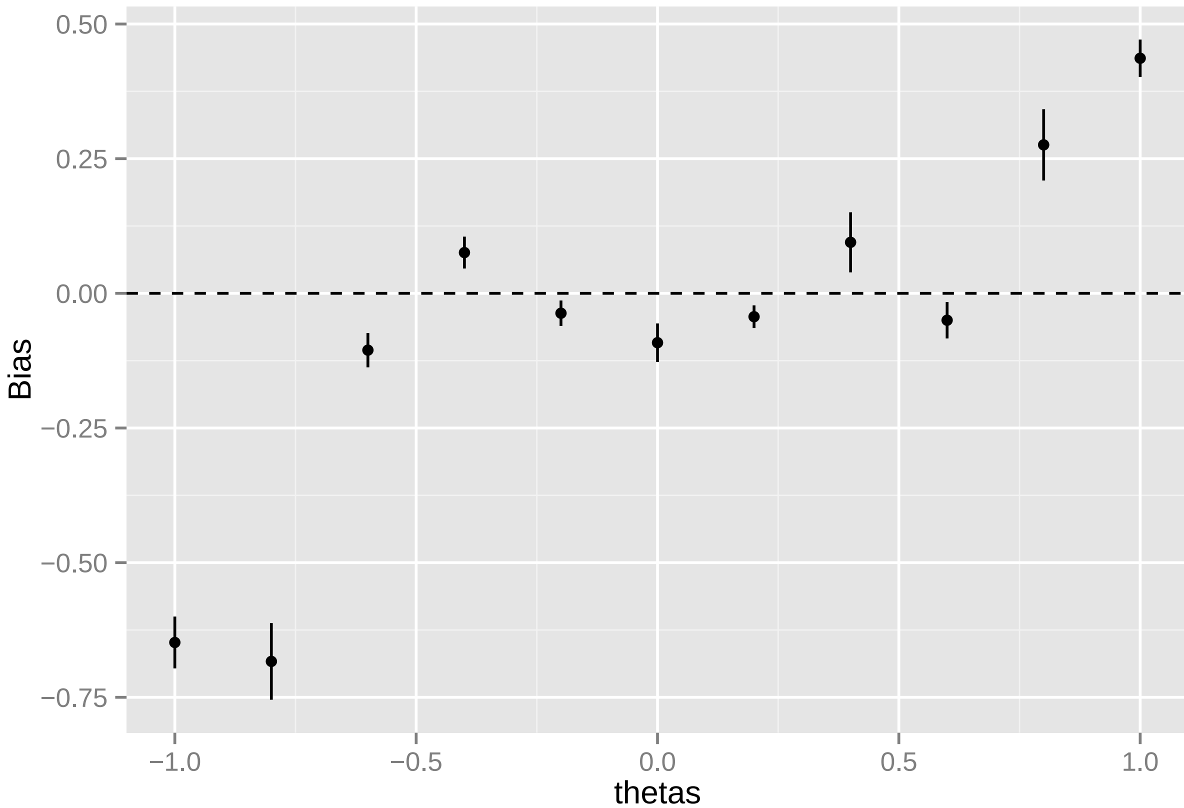
### Bias as function of sample lenth:

```
nsims <- 10
lengths <- seq(1e2, 2e3, length.out = 20) %>% floor
biases <- mapply(getBias, nsims, beta, theta, lengths)
std.err <- biases['sd',]
error <- biases['error',]
qplot(x= lengths, y= error)+
  geom_segment(aes(x = lengths, y=error-2*std.err/nsims,
                   xend=lengths, yend=error+2*std.err/nsims))+
  geom_hline(aes(itercept=0),lty=2)+ ylab('Bias')
```




**Bias as a function of theta:**

```
sample.length <- 800
thetas <- seq(-1,1, length.out = 11)
biases.2 <- mapply(getBias, nsims, beta, thetas, sample.length)

std.err <- biases.2['sd',]
error <- biases.2['error',]
qplot(x= thetas, y= error)+
  geom_segment(aes(x = thetas, y=error-std.err/nsims,
                   xend=thetas, yend=error+std.err/nsims))+
    geom_hline(aes(itercept=0),lty=2)+ ylab('Bias')
```



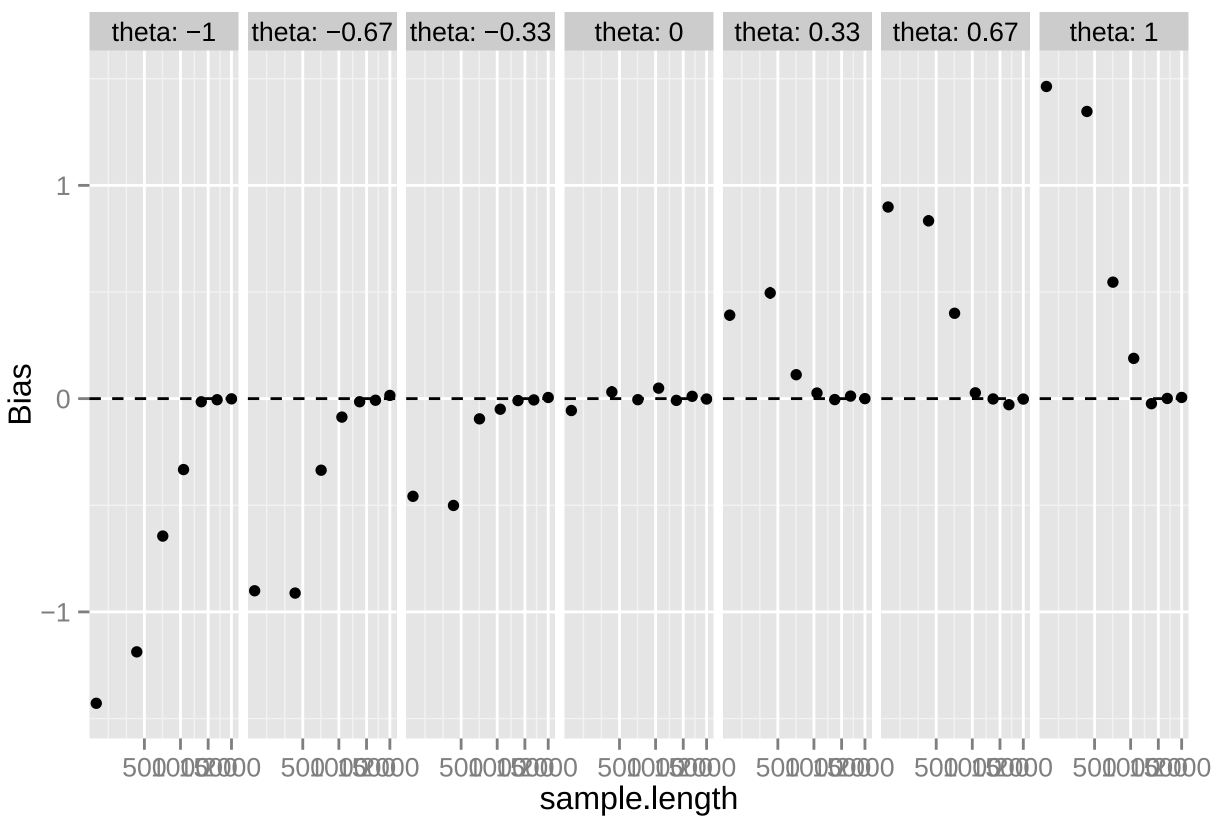

### As a function of theta and sample length:

```
nsims <- 50
lengths <- seq(1e2, 2e3, length.out = 7) %>% floor
thetas <- seq(-1,1, length.out = 7)

design <- expand.grid(nsims=nsims, beta=beta, theta=thetas, sample.length=lengths)
getBias.wrap <- function(x) do.call(getBias,as.list(x))

# Serial version
## biases.3 <- apply(design, 1, getBias.wrap)

# Paralle version:
cl <- makeCluster(detectCores())
clusterEvalQ(cl, {
  library(chords)
})
```

```
## [[1]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
##
## [[2]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
##
```

```
## [[3]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
##
## [[4]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
```

```
clusterExport(cl, c('getError','getBias'))
biases.3 <- parApply(cl, design, 1, getBias.wrap)
stopCluster(cl)

frame.3 <- data.frame(design, t(biases.3))
frame.3 <- mutate(frame.3, theta=round(theta,2))
qplot(data = frame.3,x=sample.length, y=error)+
  facet_grid(facets=.~theta, labeller = label_both)+
  geom_segment(aes(x=sample.length, y=error+2*sd/nsims,
                   xend=sample.length, yend=error-2*sd/nsims))+
  scale_x_sqrt()+
  geom_hline(aes(intercept=0), lty=2)+ ylab('Bias')
```



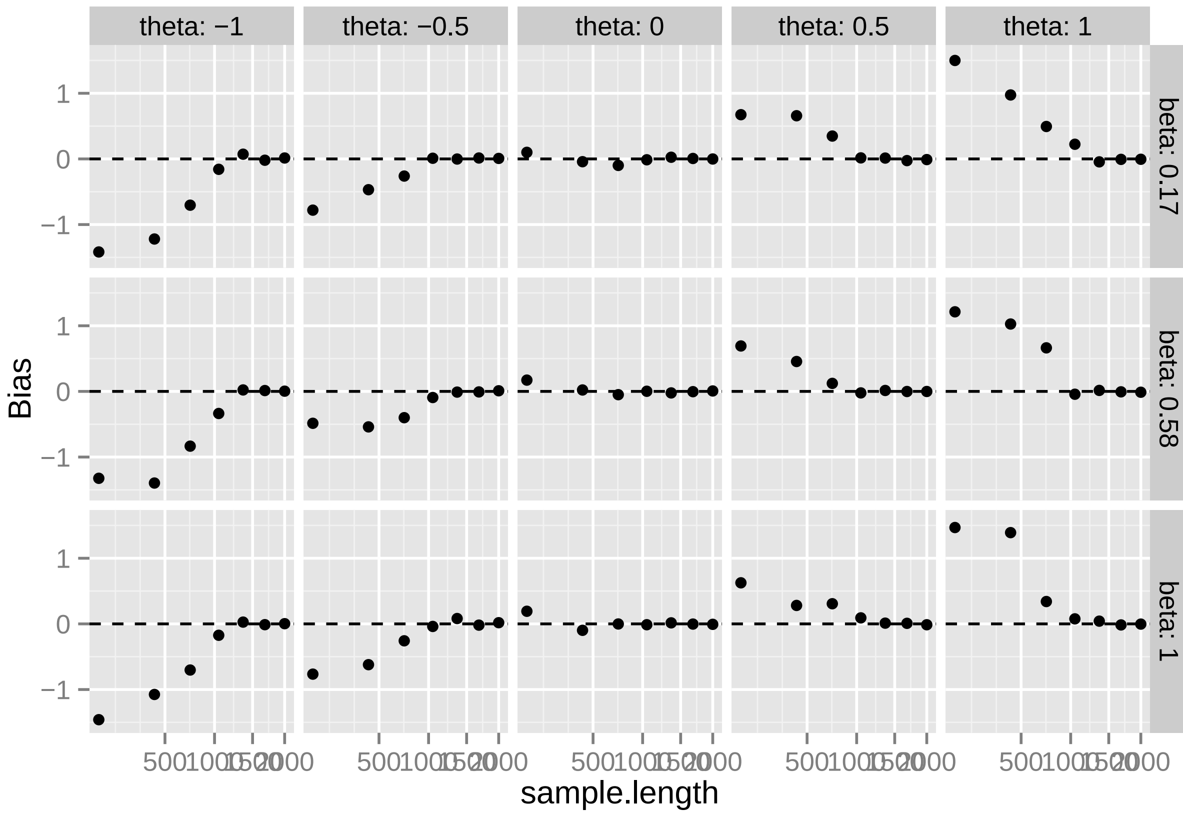

Estimator is biased but clearly consistent. Convergence seems to be at $\sqrt{n}$.

### As a function of theta, beta and sample length:

```r
nsims <- 20
lengths <- seq(1e2, 2e3, length.out = 7) %>% floor
thetas <- round(seq(-1,1, length.out = 5),2)
betas <- round(seq(1/6,1,length.out = 3),2)

design.2 <- expand.grid(nsims=nsims, beta=betas, theta=thetas, sample.length=lengths)
getBias.wrap <- function(x) do.call(getBias,as.list(x))

# Paralle version:
cl <- makeCluster(detectCores())
clusterEvalQ(cl, library(chords))
```

```
## [[1]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
##
## [[2]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
##
## [[3]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
##
## [[4]]
## [1] "chords"    "methods"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "base"
```

```r
clusterExport(cl, c('getError','getBias'))
## Careful: long run:
biases.4 <- parApply(cl, design.2, 1, getBias.wrap)
stopCluster(cl)

frame.4 <- data.frame(design.2, t(biases.4))
frame.4 <- mutate(frame.4, theta=round(theta,2))
qplot(data = frame.4, x=sample.length, y=error)+
  facet_grid(facets=beta~theta, labeller = label_both)+
  geom_segment(aes(x=sample.length, y=error+2*sd/nsims,
                   xend=sample.length, yend=error-2*sd/nsims))+
  scale_x_sqrt()+
  geom_hline(aes(intercept=0), lty=2)+ ylab('Bias')
```




Seems beta has no effect of the magnitude of the bias.