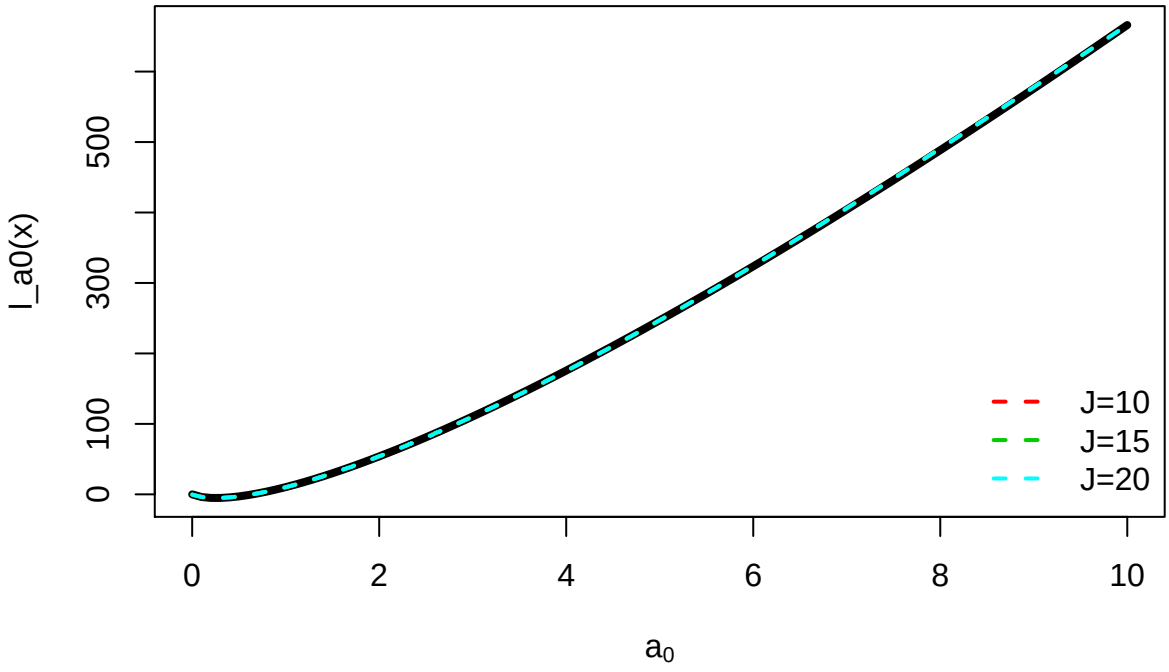
test_curve_fitting.r

luiz

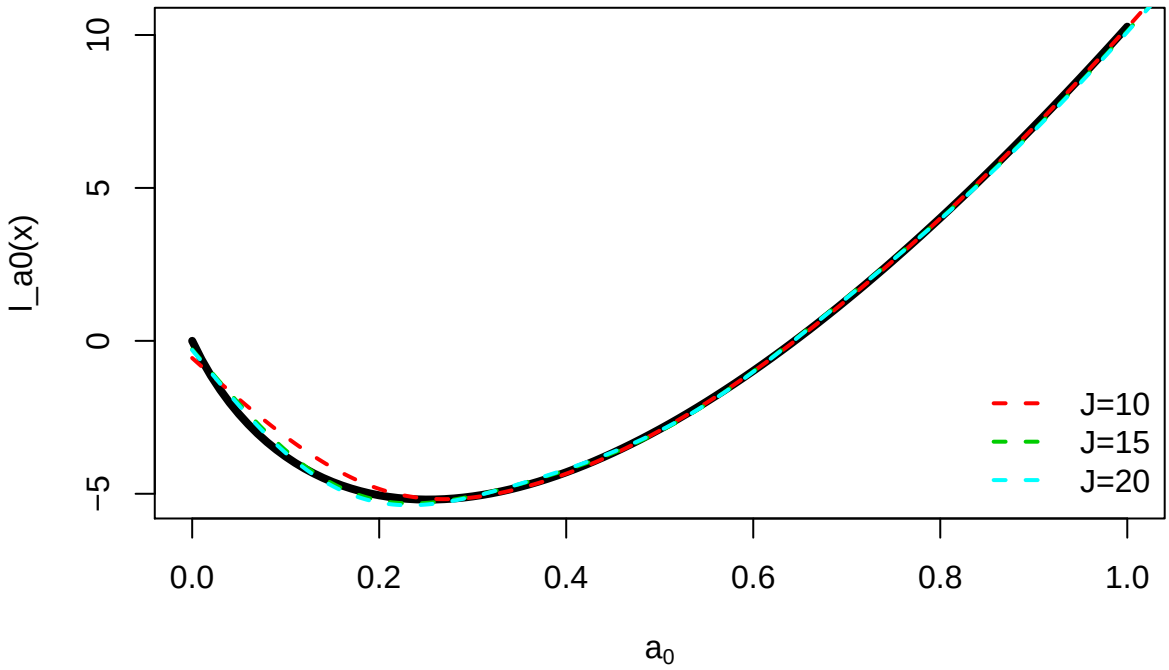
2020-02-27

```r
source("../power_priors_aux.r")
source("../data_Gaussian.r")

gs.data <- list(
  N0 = N_0,
  y0 = y_0,
  mu0 = mu_0,
  kappa0 = kappa_0,
  alpha0 = alpha_0,
  beta0 = beta_0,
  a_0 = 1
)

###
get_l_a0_gaussian <- function(y0, n0, alpha0, beta0, m0, k0, a_0){
  nstar <- a_0 * n0
  ybar <- mean(y0)
  s <- mean( (y0-ybar)^2 )
  kappa_n <- k0 + nstar
  alpha_n <- alpha0 + nstar/2
  beta_n <- beta0 + .5 * (nstar * s +  (k0 * nstar * (ybar - m0)^2 )/kappa_n )
  ans <- lgamma(alpha_n)-lgamma(alpha0)
  ans <- ans + alpha0 * log(beta0) - alpha_n * log(beta_n)
  ans <- ans + .5 *( log(k0) - log(kappa_n) )-nstar/2 * log(2*pi)
  return(ans)
}

#############
l_a0 <- function(x) {
  get_l_a0_gaussian(
    y0 = gs.data$y0,
    n0 = gs.data$N0,
    alpha0 = gs.data$alpha0,
    beta0 = gs.data$beta0,
    m0 = gs.data$mu0,
    k0 = gs.data$kappa0,
    a_0 = x
  )
}
l_a0 <- Vectorize(l_a0)
#########
#########

d10 <- read.csv("../../data/Gaussian_logCA0_adaptive_J=10.csv")
d15 <- read.csv("../../data/Gaussian_logCA0_adaptive_J=15.csv")
d20 <- read.csv("../../data/Gaussian_logCA0_adaptive_J=20.csv")
```

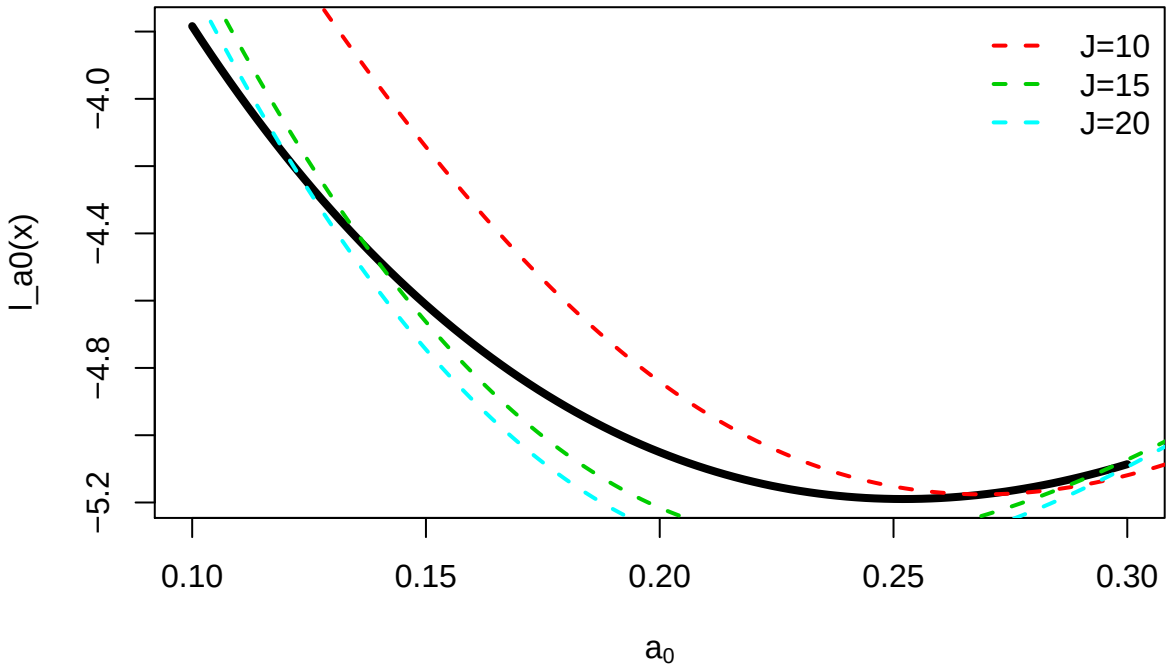
```r
gam10 <- mgcv::gam(lc_a0 ~ s(a0), data = d10)
gam15 <- mgcv::gam(lc_a0 ~ s(a0), data = d15)
gam20 <- mgcv::gam(lc_a0 ~ s(a0), data = d20)

maxA <- 10
K <- 20000
pred_a0s <- seq(0, maxA, length.out = K)

pred10 <- predict(gam10, newdata = data.frame(a0 = pred_a0s))
pred15 <- predict(gam15, newdata = data.frame(a0 = pred_a0s))
pred20 <- predict(gam20, newdata = data.frame(a0 = pred_a0s))

curve(l_a0, 0, 10, lwd = 4, xlab = expression(a[0]))
lines(pred_a0s, pred10, col = 2, lwd = 2, lty = 2)
lines(pred_a0s, pred15, col = 3, lwd = 2, lty = 2)
lines(pred_a0s, pred20, col = 5, lwd = 2, lty = 2)
legend(x = "bottomright",
       legend = c("J=10", "J=15", "J=20"), col = c(2, 3, 5),
       bty = 'n', lty = 2, lwd = 2)
```




```r
curve(l_a0, 0, 1, lwd = 4, xlab = expression(a[0]))
lines(pred_a0s, pred10, col = 2, lwd = 2, lty = 2)
lines(pred_a0s, pred15, col = 3, lwd = 2, lty = 2)
lines(pred_a0s, pred20, col = 5, lwd = 2, lty = 2)
legend(x = "bottomright",
       legend = c("J=10", "J=15", "J=20"), col = c(2, 3, 5),
       bty = 'n', lty = 2, lwd = 2)
```

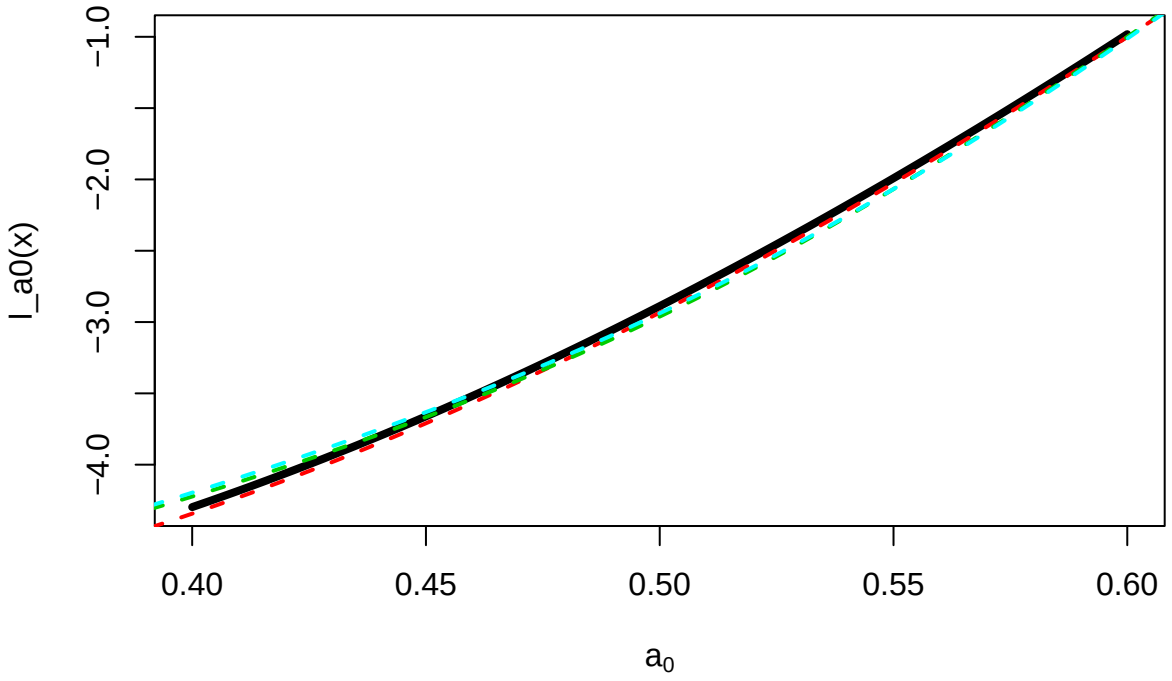



```
curve(l_a0, 0.1, .3, lwd = 4, xlab = expression(a[0]))
lines(pred_a0s, pred10, col = 2, lwd = 2, lty = 2)
lines(pred_a0s, pred15, col = 3, lwd = 2, lty = 2)
lines(pred_a0s, pred20, col = 5, lwd = 2, lty = 2)
legend(x = "topright",
       legend = c("J=10", "J=15", "J=20"), col = c(2, 3, 5),
       bty = 'n', lty = 2, lwd = 2)
```

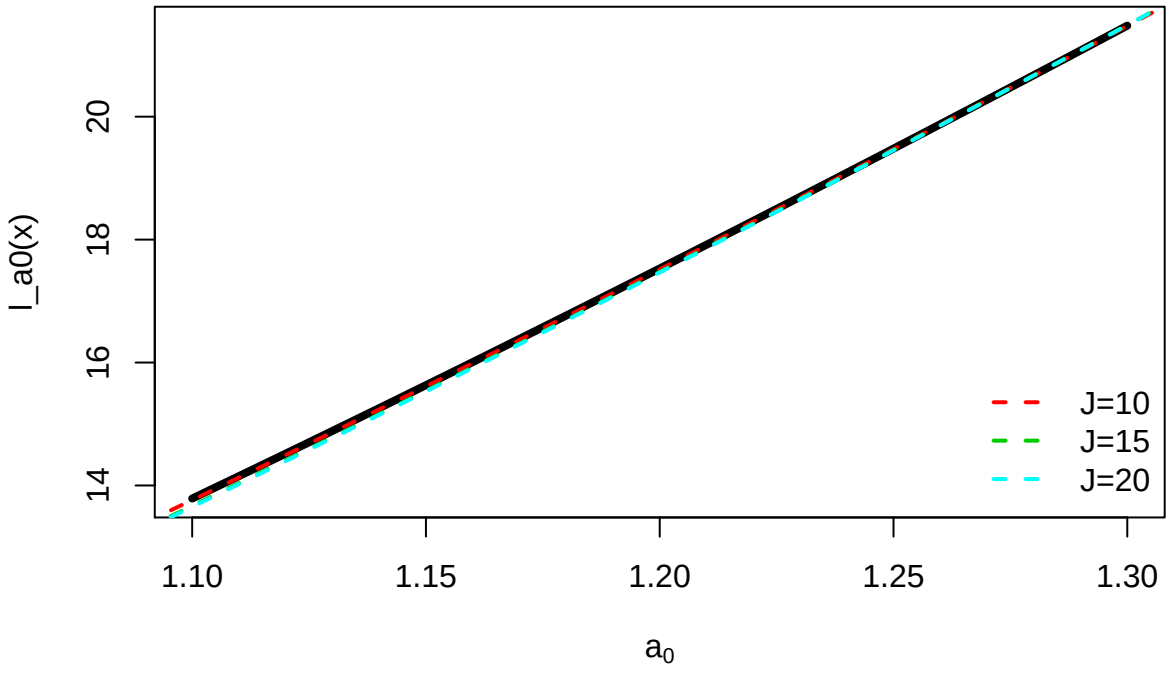



```
curve(l_a0, .4, .6, lwd = 4, xlab = expression(a[0]))
lines(pred_a0s, pred10, col = 2, lwd = 2, lty = 2)
lines(pred_a0s, pred15, col = 3, lwd = 2, lty = 2)
```

```r
lines(pred_a0s, pred20, col = 5, lwd = 2, lty = 2)
```




```r
curve(l_a0, 1.1, 1.3, lwd = 4, xlab = expression(a[0]))
lines(pred_a0s, pred10, col = 2, lwd = 2, lty = 2)
lines(pred_a0s, pred15, col = 3, lwd = 2, lty = 2)
lines(pred_a0s, pred20, col = 5, lwd = 2, lty = 2)
legend(x = "bottomright",
       legend = c("J=10", "J=15", "J=20"), col = c(2, 3, 5),
       bty = 'n', lty = 2, lwd = 2)
```



```r
true.ls <- l_a0(pred_a0s)

rmse <- function(x, y){
  sqrt(mean( (x-y)^2 ))
}

scaled_rmse <- function(x, y){
  sqrt(mean( (x-y)^2 / x ))
}

rmse(x = true.ls, y = pred10)
```

```
## [1] 1.147063
```

```r
rmse(x = true.ls, y = pred15)
```

```
## [1] 1.18154
```

```r
rmse(x = true.ls, y = pred20)
```

```
## [1] 1.186388
```

```r
scaled_rmse(x = true.ls[-1], y = pred10[-1])
```

```
## [1] 0.01504535
```

```r
scaled_rmse(x = true.ls[-1], y = pred15[-1])
```

```
## [1] 0.04982541
```

```r
scaled_rmse(x = true.ls[-1], y = pred20[-1])
```

```
## [1] 0.05030795
```