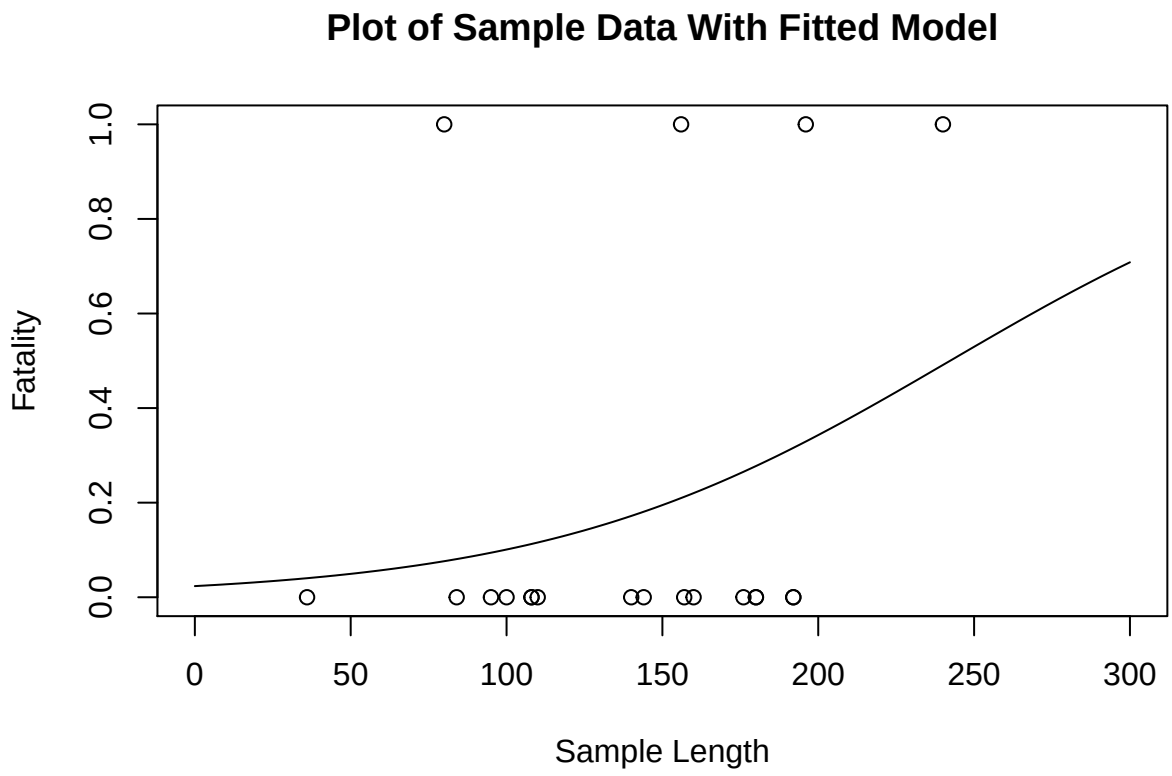
Logistic_Regression_Analysis.R

marke

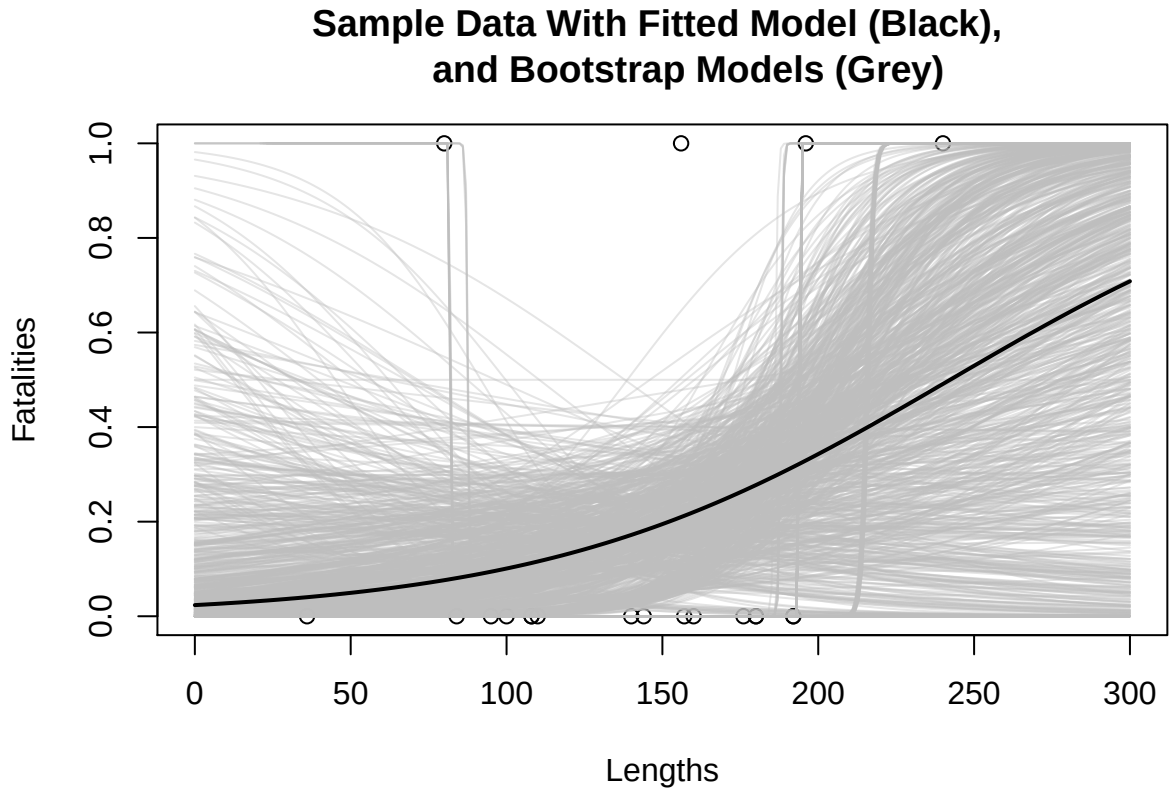
Mon May 14 13:53:25 2018

```
# Shark Attack Fatality Analysis
# Mark Erenberg
```

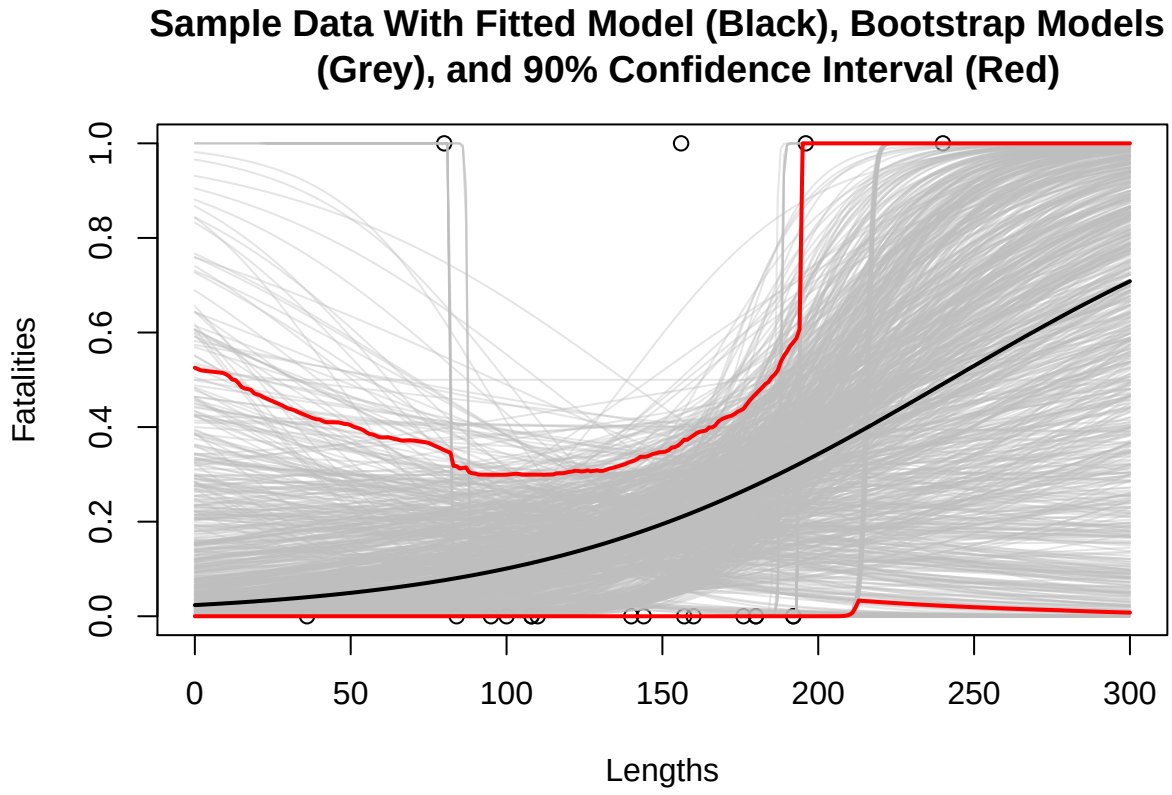
```
shark <- read.csv("C:/Users/marke/Desktop/Github/Shark Attack Analysis Repository/sharks.csv",
                  header=T)
Length <- shark$Length
Fatality <- shark$Fatality

logistic.fn <- function(z) {exp(z)/(1+exp(z))}

# Using the shark data and the variables Length and Fatality, the function glm
# was used to fit logistic regression models.

set.seed(341)
sam.Shark = sample(65,20)
coef1 <- glm(Fatality ~ Length, data=shark, subset=sam.Shark,
             family=binomial(), control=list(maxit=30))$coef

# The data from the sample was then plotted and the fitted logistic model was
# overlayed.

xseq = seq(0,300, length.out=301)

plot(shark$Length[sam.Shark],shark$Fatality[sam.Shark],
     xlab = "Sample Length",
     ylab = "Fatality",
     main = "Plot of Sample Data With Fitted Model",
     xlim=range(0,300))
lines(xseq, logistic.fn(coef1[1]+ xseq*coef1[2]))
```



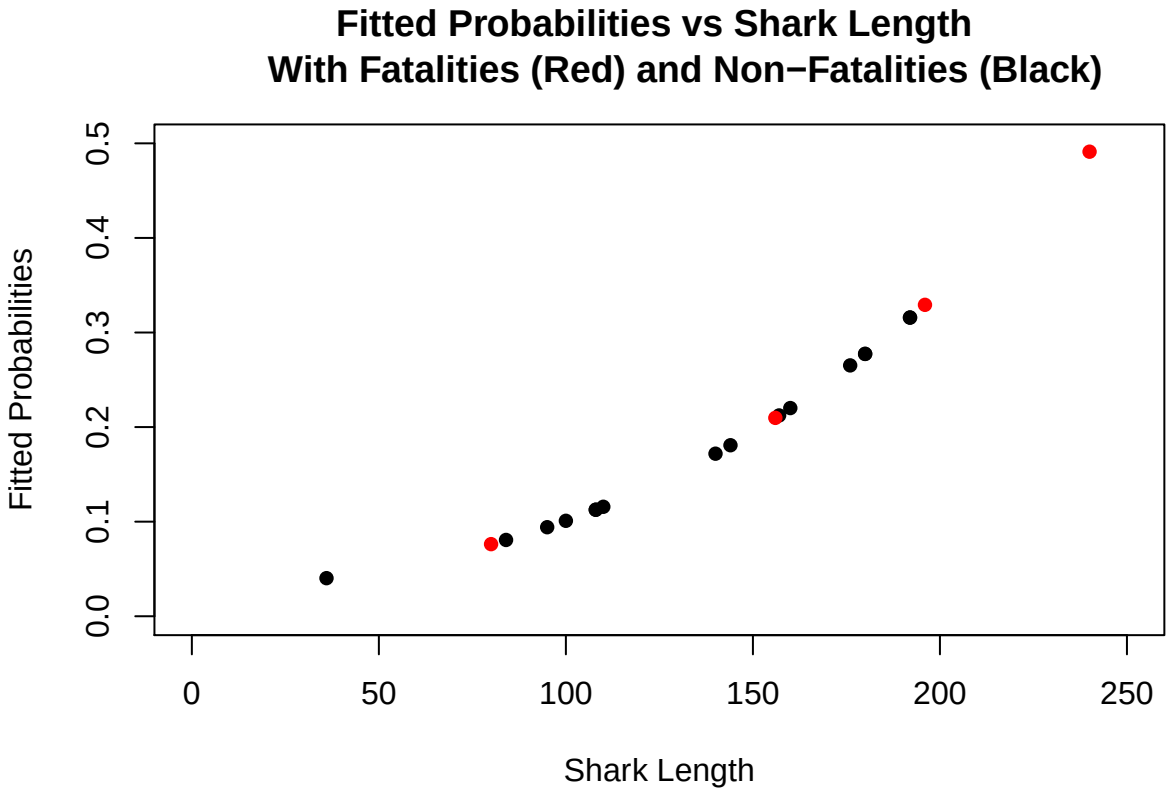

```
# 1000 bootstrap samples were then generated by sampling from the previous sample
# with replacement. The simple logistic regression model was then fitted to all
# these samples, and the data was then plotted with all the bootstrap logistic
# lines and the fitted logstic line from the original model overlayed.

B = 1000
x = shark$Length[sam.Shark]
y = shark$Fatality[sam.Shark]
n = 20

options(warn=-1)
beta.boot = t(sapply(1:B, FUN =function(b)
  glm(y~x, subset=sample(n,n, replace=TRUE),
      family=binomial(), control=list(maxit=30))$coef))
options(warn=0)

plot(x,y, xlab = "Lengths",
     ylab = "Fatalities",
     main = "Sample Data With Fitted Model (Black),
     and Bootstrap Models (Grey)",
     xlim=range(0,300))
for (i in 1:B) lines(xseq, logistic.fn(beta.boot[i,1] + xseq*beta.boot[i,2]),
                     col=adjustcolor("grey",alpha=0.4))
lines(xseq, logistic.fn(coef1[1]+ xseq*coef1[2]),lwd=2)
```




```
# A 90% confidence interval was then created from the bootstrap samples, and
# the sample data was again plotted and overlayed with the original fitted
# logistic line and the 90% confidence interval.

boot.ci=matrix(0, nrow=length(xseq), 2)

for (i in 1:length(xseq)) {
  y.hat =apply(beta.boot, 1, function(z,a) {sum(z*a)}, a=c(1, xseq[i]))
  y.hat=logistic.fn(y.hat)
  y.hat[is.nan(y.hat)] <- 1
  boot.ci[i,] =quantile(y.hat, prob=c(.05, .95))
  }

plot(x,y, xlab = "Lengths",
     ylab = "Fatalities",
     main = "Sample Data With Fitted Model (Black), Bootstrap Models
     (Grey), and 90% Confidence Interval (Red)",
     xlim=range(0,300))
for (i in 1:B) lines(xseq, logistic.fn(beta.boot[i,1] + xseq*beta.boot[i,2]),
                     col=adjustcolor("grey",alpha=0.4))
lines(xseq, logistic.fn(coef1[1]+ xseq*coef1[2]), lwd=2)
lines(xseq,boot.ci[,1],col="red", lwd=2)
lines(xseq,boot.ci[,2],col="red", lwd=2)
```

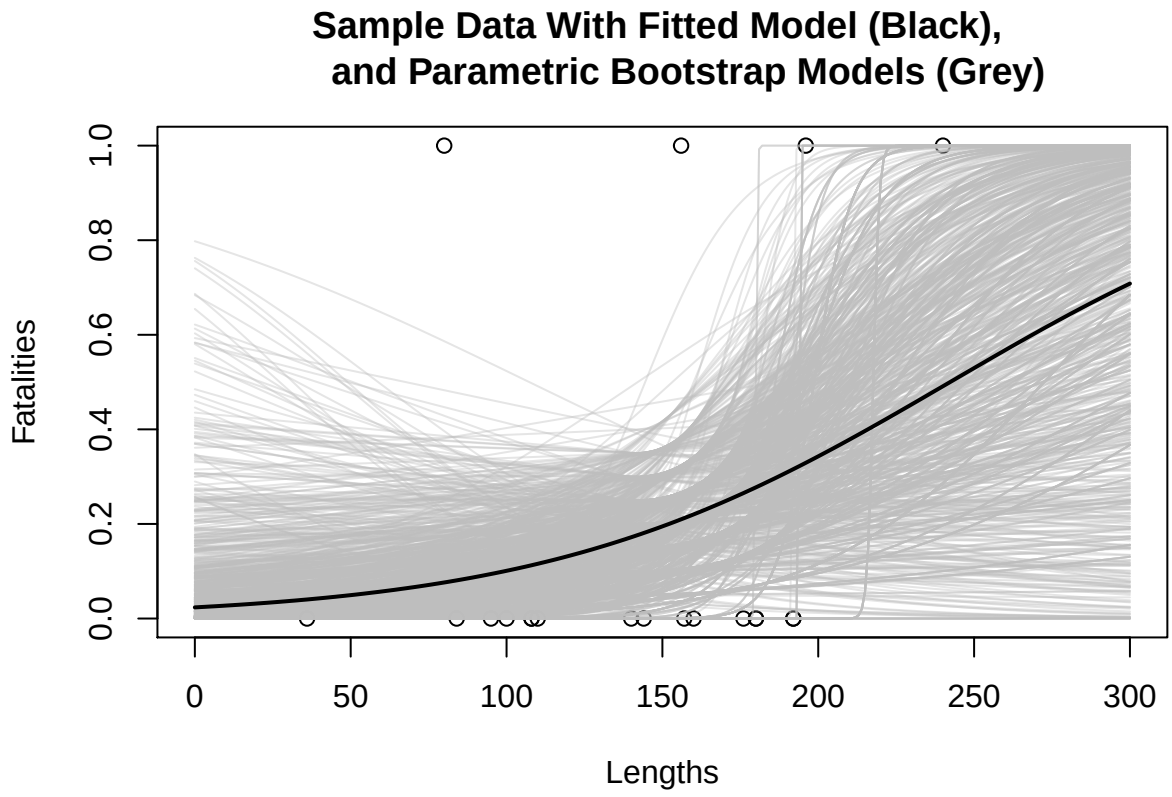



```
# The fitted probabilties form the inital model were then plotted versus shark
# length.

phat <- function(x){logistic.fn(coef1[1]+ x*coef1[2])}

fatal <- c()
nonfatal <- c()

for(i in 1:length(x)){
  if(y[i]==1){
    fatal <- c(fatal, rownames(shark[sam.Shark,])[i])

  }
  else{nonfatal <- c(nonfatal, rownames(shark[sam.Shark,])[i])}
}

pfatal <- phat(shark[sam.Shark,][fatal,]$Length)
pnonfatal <- phat(shark[sam.Shark,][nonfatal,]$Length)


plot(shark[sam.Shark,][nonfatal,]$Length, pnonfatal, xlab = "Shark Length",
     ylab = "Fitted Probabilities",
     main = "Fitted Probabilities vs Shark Length
     With Fatalities (Red) and Non-Fatalities (Black)",
     col="black", pch=16, ylim=range(0,0.5), xlim=range(0,250))
points(shark[sam.Shark,][fatal,]$Length,pfatal,col="red",pch=16)
```



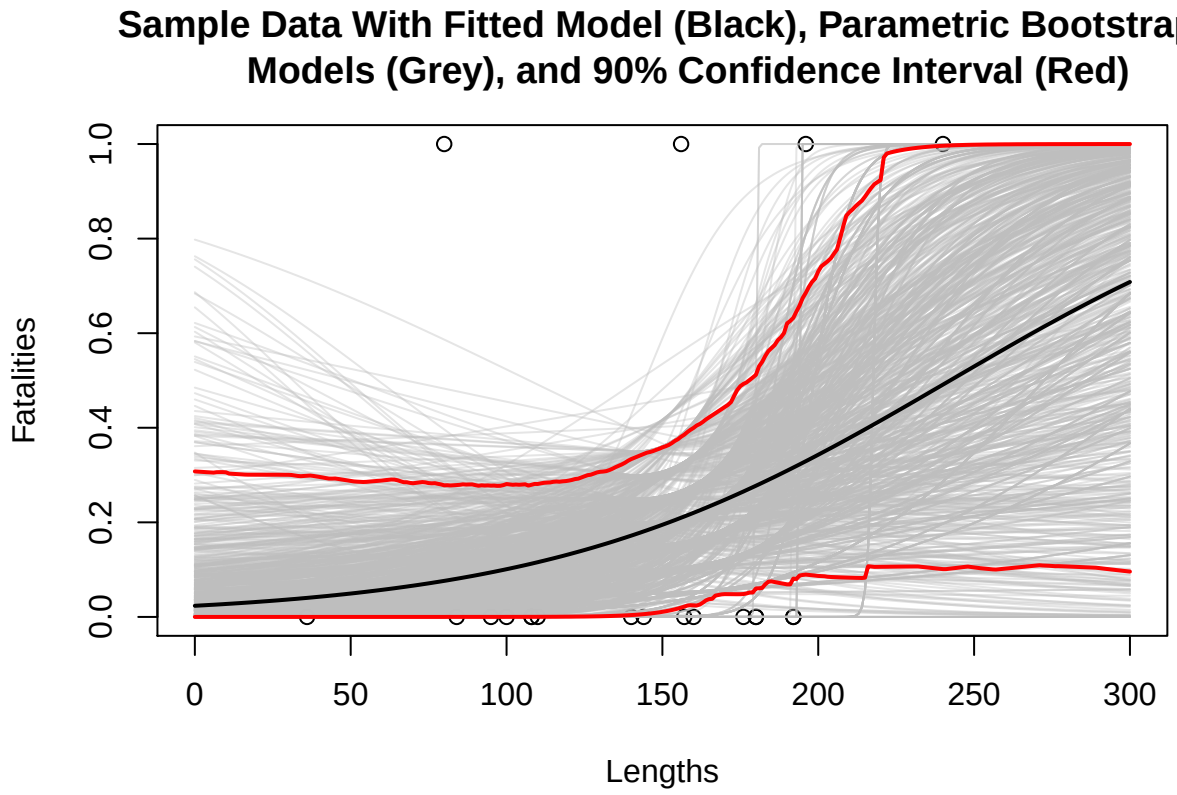

```
# Another 1000 bootstrap samples were then generated by sampling from the model.
# ie. each sample consisted of {(x1, y*1), . . . , (xn, y*n )} where y*i was
# generated from the distribution of yi conditional on the model fit.
# The simple logistic regression model was again fitted to these parametric
# bootstrap samples.

fit1 <- glm(y~x,family=binomial(),control=list(maxit=30))
fit2 <- glm(y~I(x-mean(x)),family=binomial(),control=list(maxit=30))

set.seed(341)
par.boot.sam =Map(function(b)
  { y=rbinom(20, size=1, prob=phat(x))
    data.frame( x = x, y= y ) } , 1:B)

options(warn=-1)
par.boot.coef =Map(function(sam)
  glm(y~x, data=sam, family=binomial(), control=list(maxit=30))$coef,
  par.boot.sam)
options(warn=0)

plot(x,y, xlab = "Lengths",
     ylab = "Fatalities",
     main = "Sample Data With Fitted Model (Black),
     and Parametric Bootstrap Models (Grey)",
     xlim=range(0,300))
for (i in 1:B) lines(xseq, logistic.fn(unlist(par.boot.coef[i])[1] +
```

```
                                        xseq*unlist(par.boot.coef[i])[2]),
                    col=adjustcolor("grey",alpha=0.4))
lines(xseq, logistic.fn(coef1[1]+ xseq*coef1[2]),lwd=2)
```




```
# From these parametric bootstrap samples, another 90% confidence interval
# was created for the original logistic regression line, and the sample data
# was again plotted with the regression lines and the confidence interval.

alphas <- unlist(Map(function(i){unlist(par.boot.coef[i])[1]},1:B))
betas <- unlist(Map(function(i){unlist(par.boot.coef[i])[2]},1:B))

par.boot.coef2 <- data.frame(Intercept=alphas,x=betas)

boot.ci=matrix(0, nrow=length(xseq), 2)

for (i in 1:length(xseq)) {
  y.hat =apply(par.boot.coef2, 1, function(z,a) {sum(z*a)}, a=c(1, xseq[i]))
  y.hat=logistic.fn(y.hat)
  y.hat[is.nan(y.hat)] <- 1
  boot.ci[i,] =quantile(y.hat, prob=c(.05, .95))
}

plot(x,y, xlab = "Lengths",
     ylab = "Fatalities",
     main = "Sample Data With Fitted Model (Black), Parametric Bootstrap
     Models (Grey), and 90% Confidence Interval (Red)",
```

```
     xlim=range(0,300))
for (i in 1:B) lines(xseq, logistic.fn(unlist(par.boot.coef[i])[1] +
                                          xseq*unlist(par.boot.coef[i])[2]),
                     col=adjustcolor("grey",alpha=0.4))
lines(xseq, logistic.fn(coef1[1]+ xseq*coef1[2]),lwd=2)

lines(xseq,boot.ci[,1],col="red", lwd=2)
lines(xseq,boot.ci[,2],col="red", lwd=2)
```




```
cor( c( 1 , 1 ), c( 2 , 3 ) )
```

```
## Warning in cor(c(1, 1), c(2, 3)): the standard deviation is zero
```

```
## [1] NA
```

```
options(warn=-1)
cor( c( 1 , 1 ), c( 2 , 3 ) )
```

```
## [1] NA
```

```
options(warn=0)
```