Shark_Fatality_Analysis.R

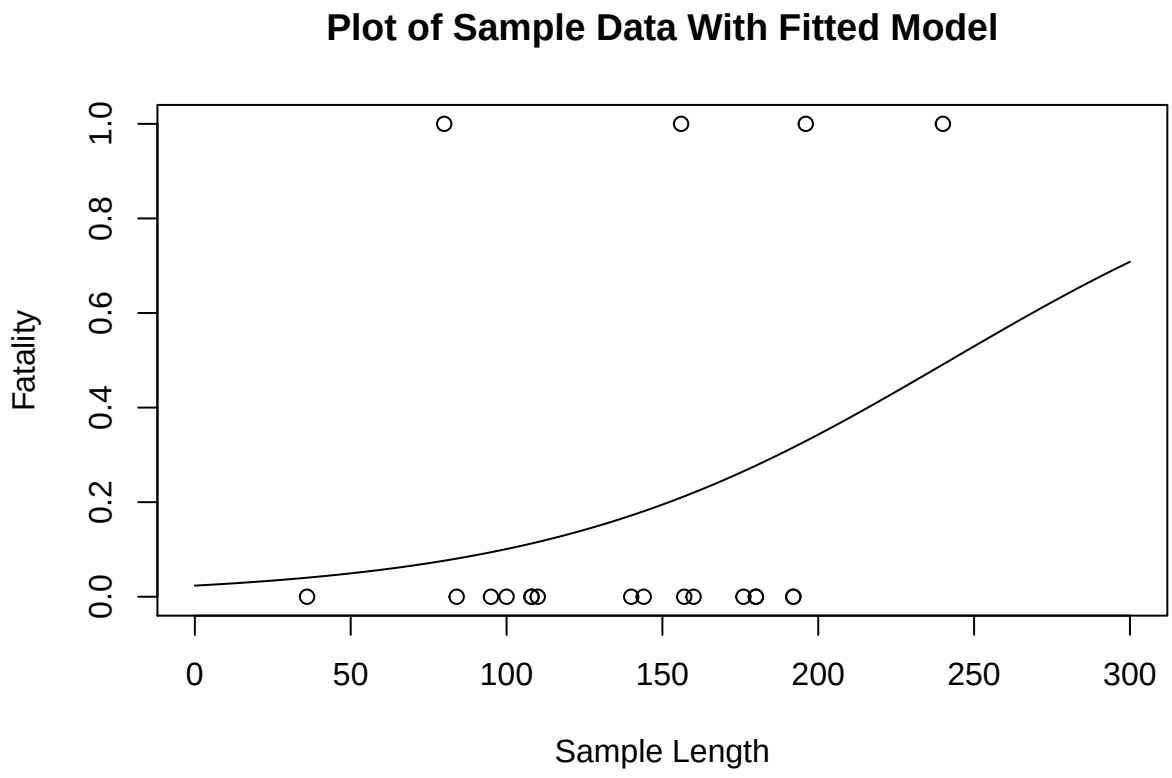
marke

Fri May 18 03:30:16 2018

```
# Shark Encounter Fatality Analysis
# Mark Erenberg
```

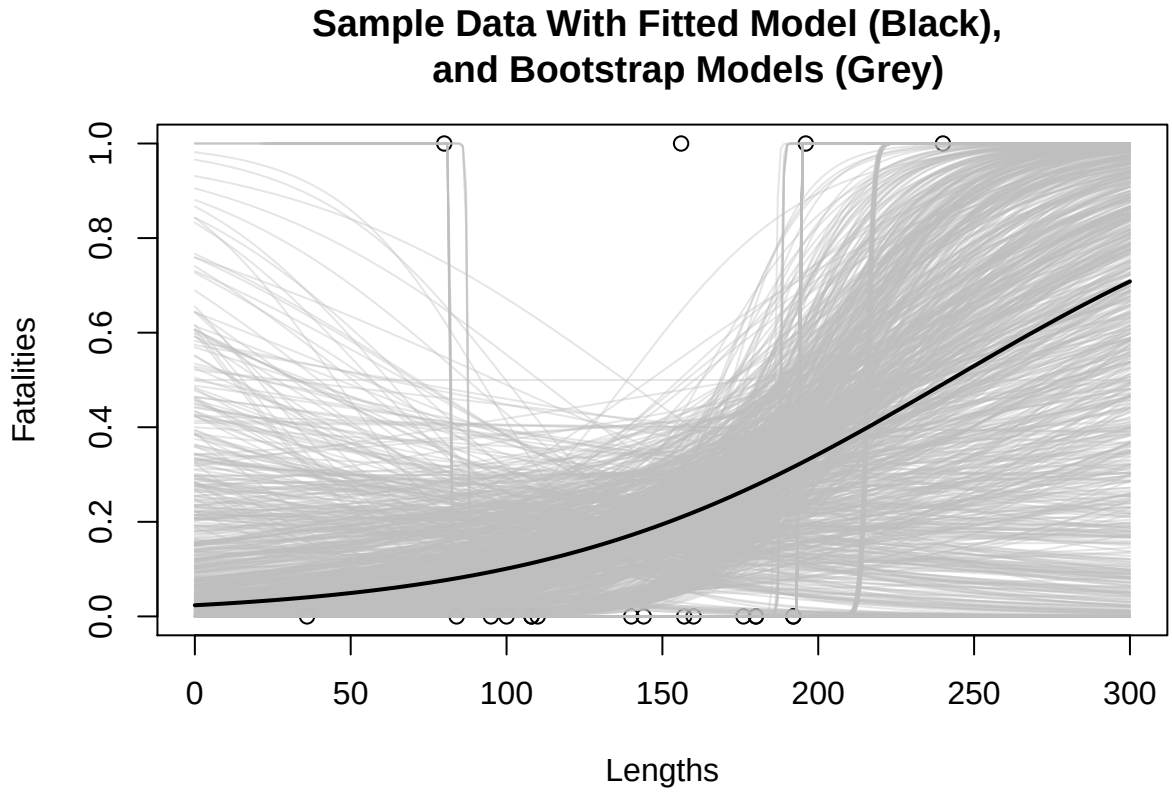
```
## The purpose of this analysis is to build a model that can confidently
## represent the likelihood of a fatal shark encounter in the years 1999 through
## to 2014 in the USA. The model is not intended to be a prediction model, but
## is rather intended to look through the historical data set and capture the
## variation of the response variable. To assess the confidence of this
## representation, bootstrap and parametric bootstrap samples were calculated
## with replacement, and 90% confidence intervals were constructed using these
## samples.


## The data set that was used contains information about documented
## great white shark encounters in Australia and the US. The attributes used
## in this analysis are Length, representing shark length, and Fatality,
## representing whether or not the documented encounter resulted in a fatality.
## The respone variable in this data set is Fatality, a binary variable taking
## on values 1 if an encounter was fatal, and 0 otherwise.
## The explanatory variable in this data set is Length, a continuous variable
## taking on real numbers that correspond to the length of the shark involved
## in the encounter.


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

## Since the repsonse variable is binary, it was determined that a logistic
## regression model could accurately regress for the probability of the repsonse
## outcome. Using the shark data and the variables Length and Fatality, the
## function glm was thus used to fit logistic regression models.

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

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]))
```



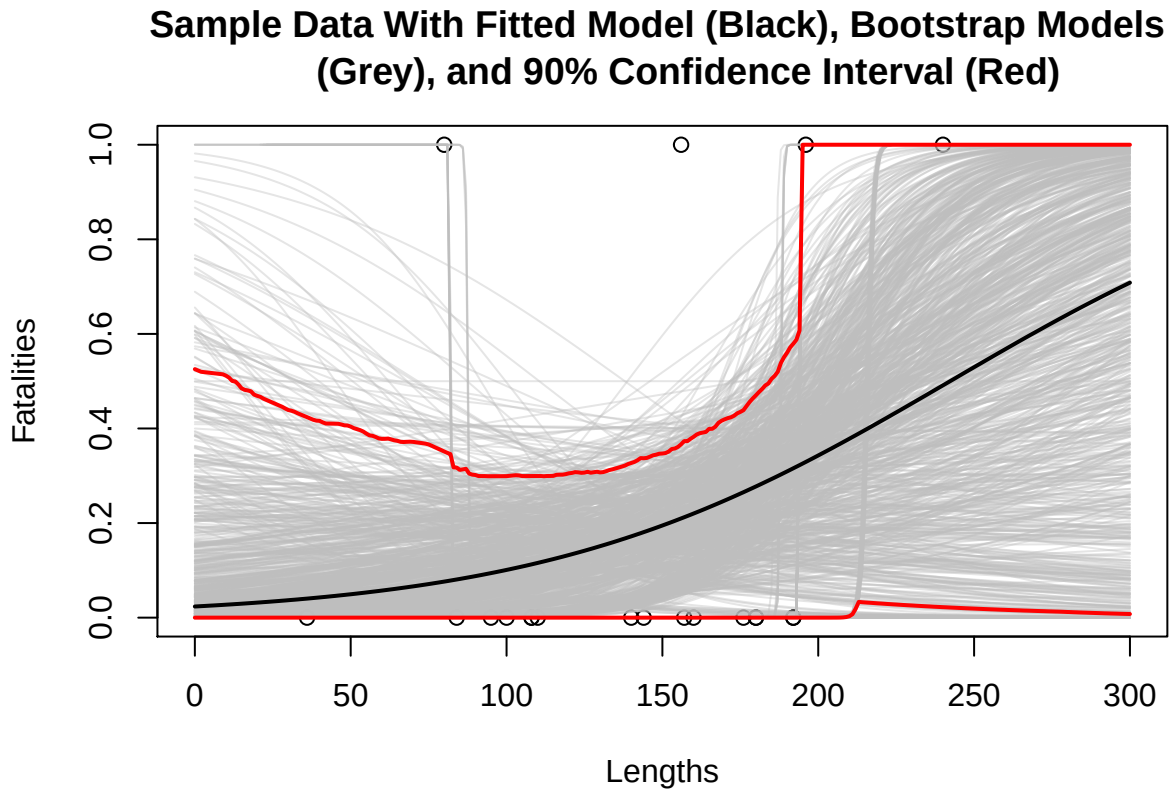

```
## 1000 bootstrap samples were then generated by sampling from the original sample
## with replacement. The simple logistic regression model was then fitted to all
## these samples, and the data was plotted with all the bootstrap logistic
## lines and the fitted logistic line from the original model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



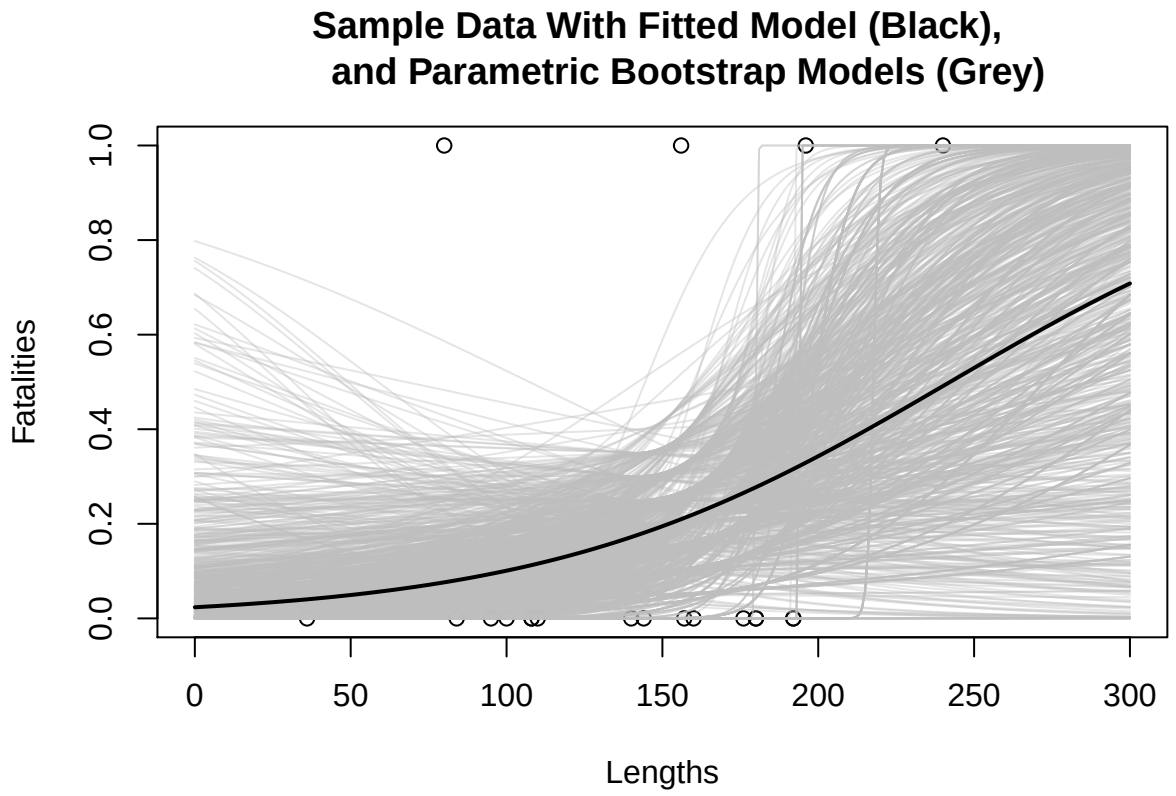

```
## A 90% confidence interval was then created from the bootstrap samples, and
## the original sample data was again plotted and overlayed with the original
## fitted logistic line and the 90% confidence interval.

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)
```



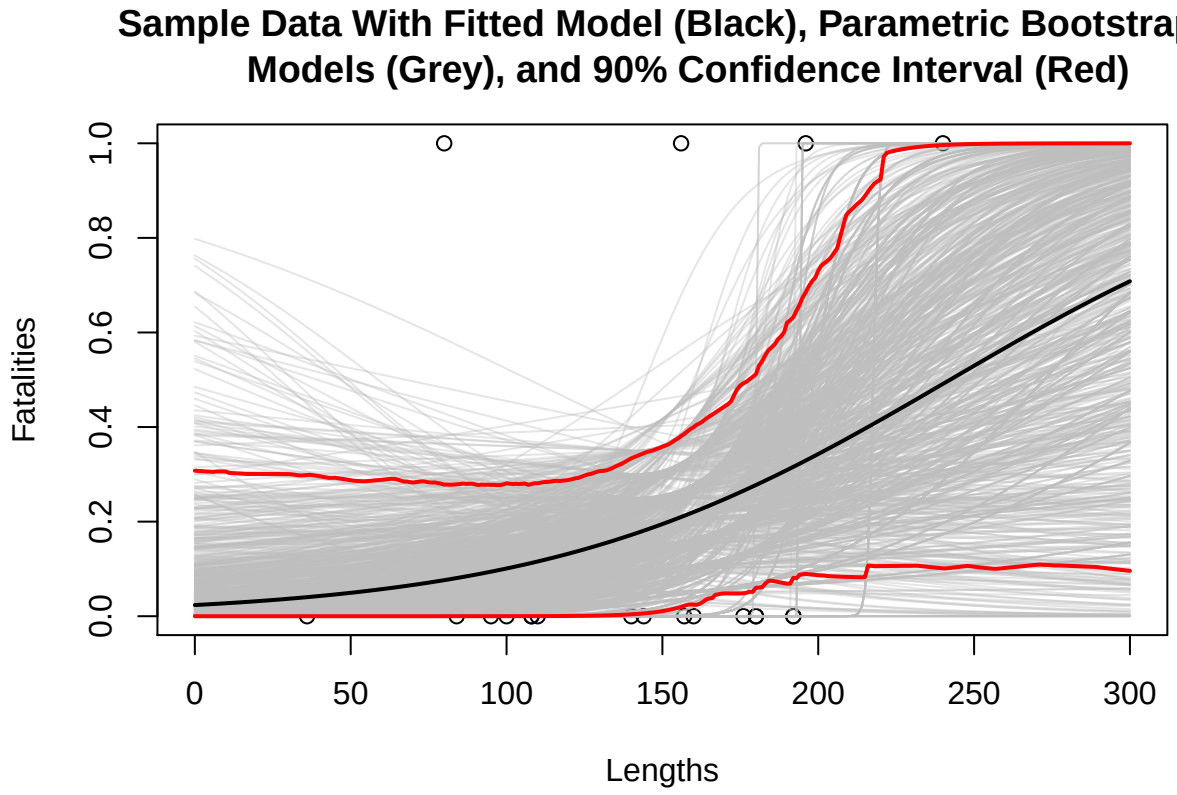

```
## Another 1000 bootstrap samples were then generated by sampling from the model.
## ie. each sample consisted of a pair of observations generated from the
## distribution of the response, conditional on the model fit.
## The simple logistic regression model was again fitted to these parametric
## bootstrap samples.

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

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 original
## sample data was again plotted with the regression lines and the confidence
## interval overlayed.

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)
```