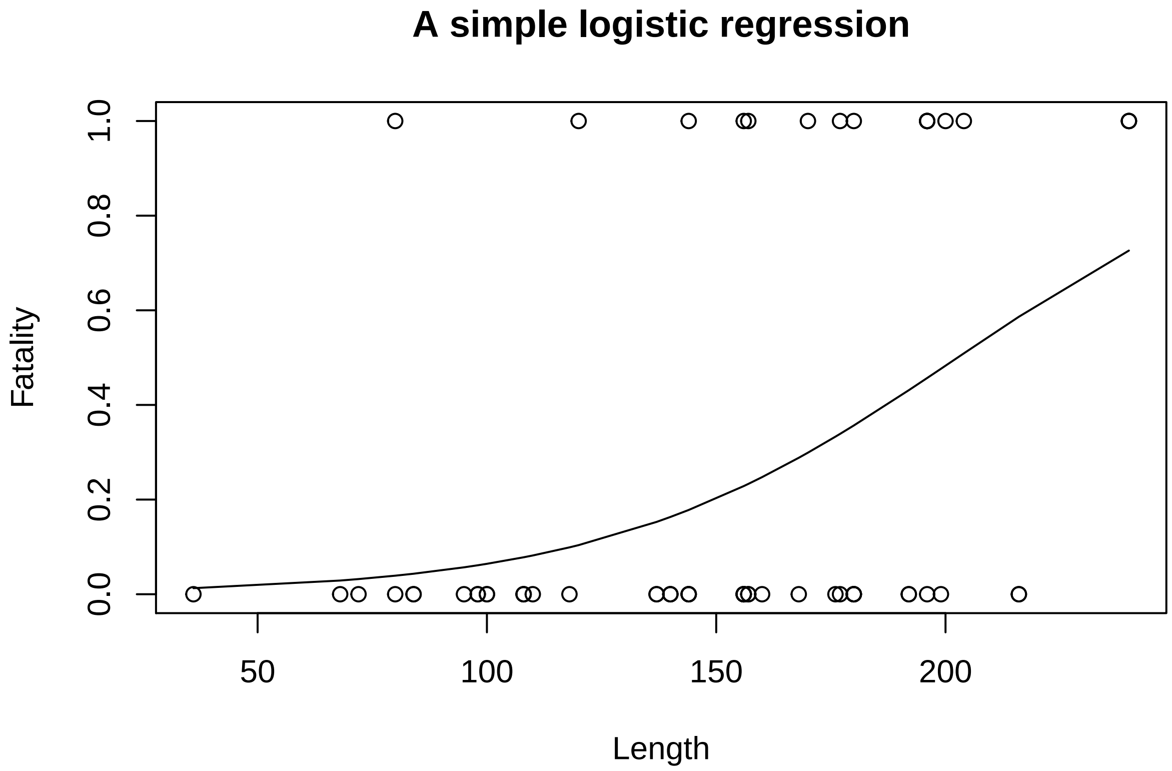
Loading up the Data, and libraries

```
library(rmarkdown)
library(shiny)
library(knitr)
setwd('C:/Users/Abhijeet/Desktop/341 Assignments')
shark=read.csv('sharks.csv')
shark=shark[with(shark, order(Length)), ]
original_shark=shark
##Given Fns Here:##
createStrataMechanism <- function (pop, grp ) {

  method <- function(sampSize) {
    sam = list()
    for (h in 1:length(sampSize)) sam[[h]] = sample(pop[grp == h], sampSize[h])
    sam
  }
  return(method)
}
```

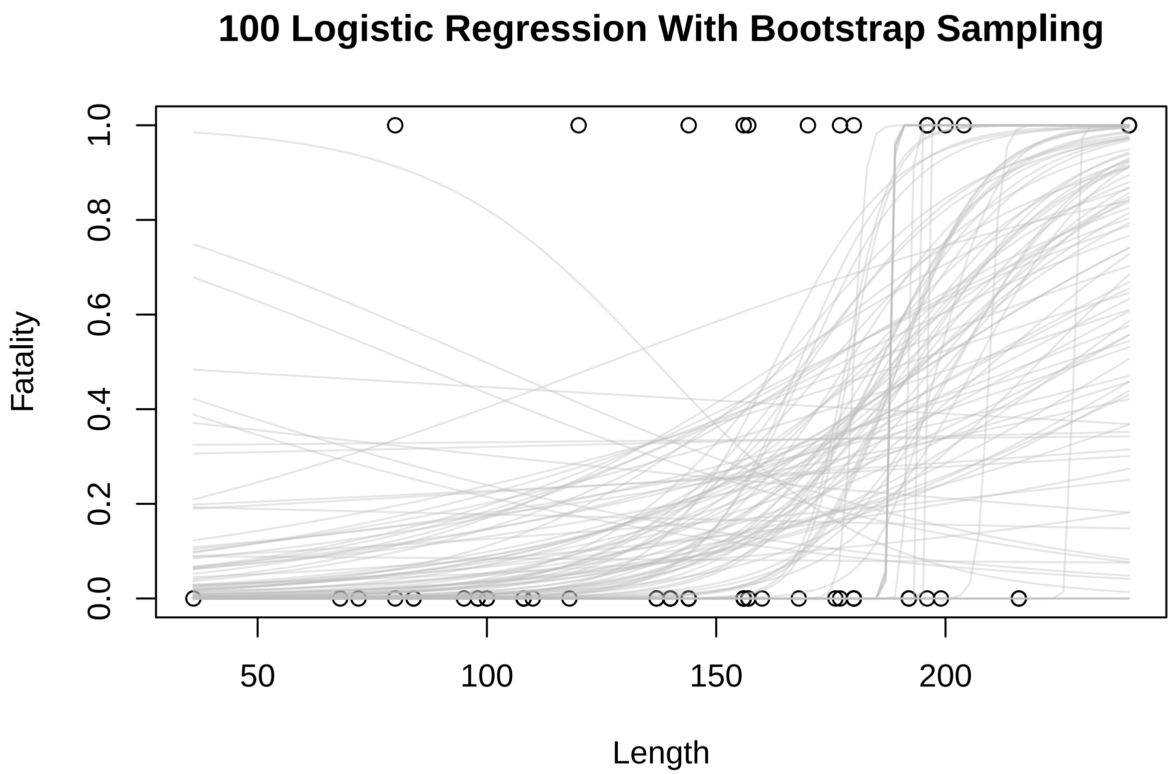
A simple logistic model based off the data:

```
logit=glm(Fatality ~ Length, data=shark, family=binomial())
plot(shark[,'Length'],shark[,'Fatality'],xlab='Length',ylab='Fatality',main='A simple logistic regressi
predicted=predict(logit,shark,type='response')
lines(shark[,'Length'],predicted)
```



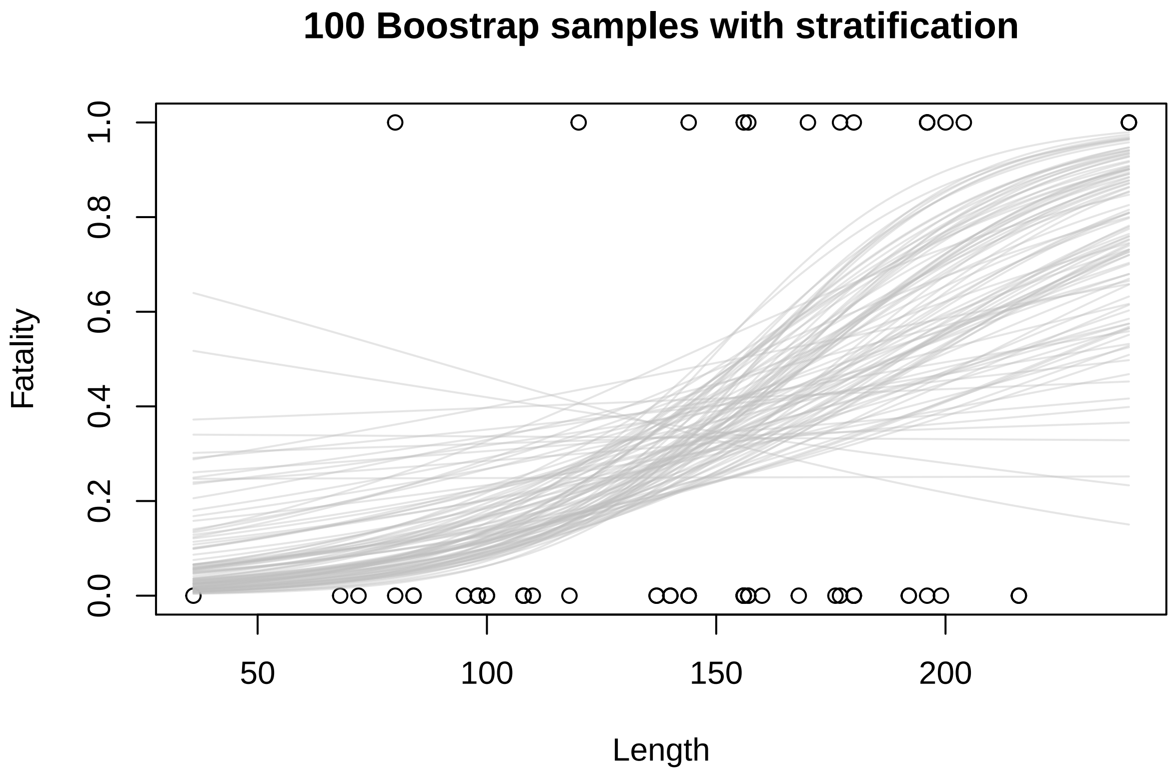
In the dataset, there are multiple parameters that one can look as such as: Age of the Diver, was he or she going scuba? Diving? etc. To explore whether some of the other paramets could have an impact on this, we can use stratified sampling, or Bootstrap sampling. Below is a chart of Bootstrap sampling being done to the original data, the sample sizes are of size 12, and there are a total of 100 samples.

```
set.seed(12345)
options(warn=-1)
sam_indice <- Map(function(i){unlist(sample(seq(1,65), size=12, replace = FALSE))}, 1:100)
samp_shark <- Map(function(j){shark[j,]}, sam_indice)
plot(shark[,'Length'], shark[,'Fatality'],xlab='Length',ylab='Fatality',main='100 Logistic Regression Wi
invisible(Map(function(k){
  fat <- k$Fatality
  length <- k$Length
  model<-glm(fat~length, family = binomial(), control = list(maxit = 50))
  curve(predict(model, data.frame(length=x),type="resp"),add=TRUE,    col = adjustcolor("grey",.4))},
  samp_shark))
```



The graph looks a bit like a jumbled mess, so let's do Boostrap sampling with stratification on whether the person was doing scuba diving.

```
plot(shark[,'Length'],shark[,'Fatality'],xlab='Length',ylab='Fatality',main='100 Boostrap samples with s
strata_3 = numeric(nrow(shark))
strata_3[shark$Scuba == 0] = 1
strata_3[shark$Scuba == 1] = 2
shark_strata_3=createStrataMechanism(1:65, strata_3)
invisible(Map(function(k){
  indice <- unlist(shark_strata_3(c(6,6)))
  fat <- shark[indice,"Fatality"]
  length <- shark[indice, "Length"]
  model <- glm(fat ~ length,data=shark, family = binomial())
  curve(predict(model, data.frame(length=x), type="resp"),
        add=TRUE, col=adjustcolor("grey",.4))}, 1:100))
```



Most of the curves on this graph go from 0 to 1, therefore we can say that when comparing the shark lengths (once scuba is treated as a fixed parameter). As the shark length increases, the likelihood of it being a fatal encounter also increases.