Predicting The Survival Rate of Lobsters

Shivam Tripathi

The dataset for this assignment comes from the following paper:

Wilkinson, E. B., Grabowski, J. H., Sherwood, G. D., and Yund, O. Y. (2015) Influence of predator identity on the strength of predator avoidance responses in lobsters. Journal of Experimental Marine Biology and Ecology, 465, 107–112.

The authors were interested in how a juvenile lobster's size was related to its vulnerability to predation. In total, 159 juvenile lobsters were collected from their natural habitat in the Gulf of Maine, USA, and the length of each lobster's carapace (upper shell) was measured to the nearest 3 mm. The lobsters were then tethered to the ocean floor for 24 hours. Any missing lobsters were assumed to have been consumed by a predator, while the surviving lobsters were released.

The dataset only contains 2 variables:
- `size`: the length of the lobster's carapace, to the nearest 3 mm.
- `survived`: this takes the value 1 if the lobster survived, and the value 0 if it did not.

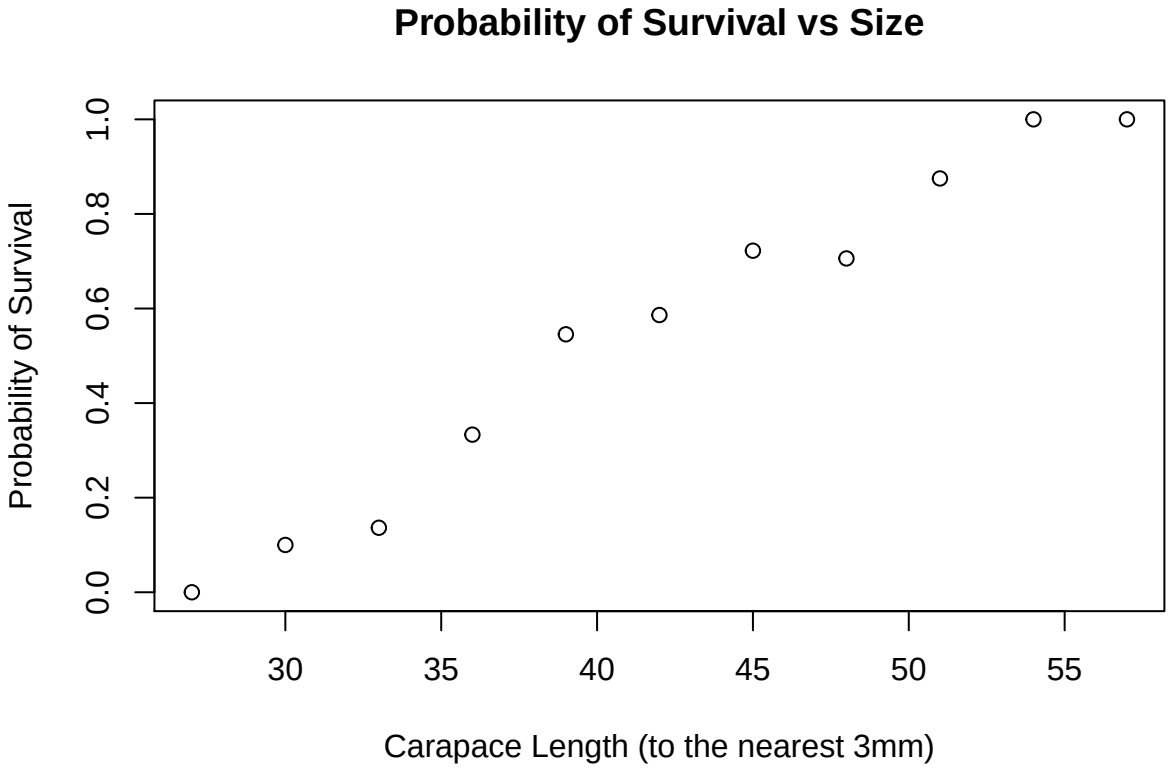
Using the values above, we'll group the data respective to size (for a more readable format), and create 3 new variables for our analysis:
- `y`: The number of lobsters of each size that survived.
- `n`: The total number of lobsters of each size.
- `p`: The proportion of lobsters of each size that survived.

```
lobster.df1 = read.csv("lobster.csv")
y = tapply(lobster.df1$survived, lobster.df1$size, sum)
n = tapply(lobster.df1$survived, lobster.df1$size, length)
p = y/n

lobster.df = data.frame(size = unique(sort(lobster.df1$size)), y, n, p)

lobster.df
```

```
##    size  y  n         p
## 27   27  0  5 0.0000000
## 30   30  1 10 0.1000000
## 33   33  3 22 0.1363636
## 36   36  7 21 0.3333333
## 39   39 12 22 0.5454545
## 42   42 17 29 0.5862069
## 45   45 13 18 0.7222222
## 48   48 12 17 0.7058824
## 51   51  7  8 0.8750000
## 54   54  6  6 1.0000000
## 57   57  1  1 1.0000000
```

Now we can examine the effect of a lobster's size on its vulnerability to predation through a basic scatter plot.

```
plot(lobster.df$p~lobster.df$size, main="Probability of Survival vs Size", xlab ="Carapace Length (to t
```



There seems to be an extremely strong positive correlation between size and proportion survived (p). It appears that the larger lobsters are less vulnerable to predation.

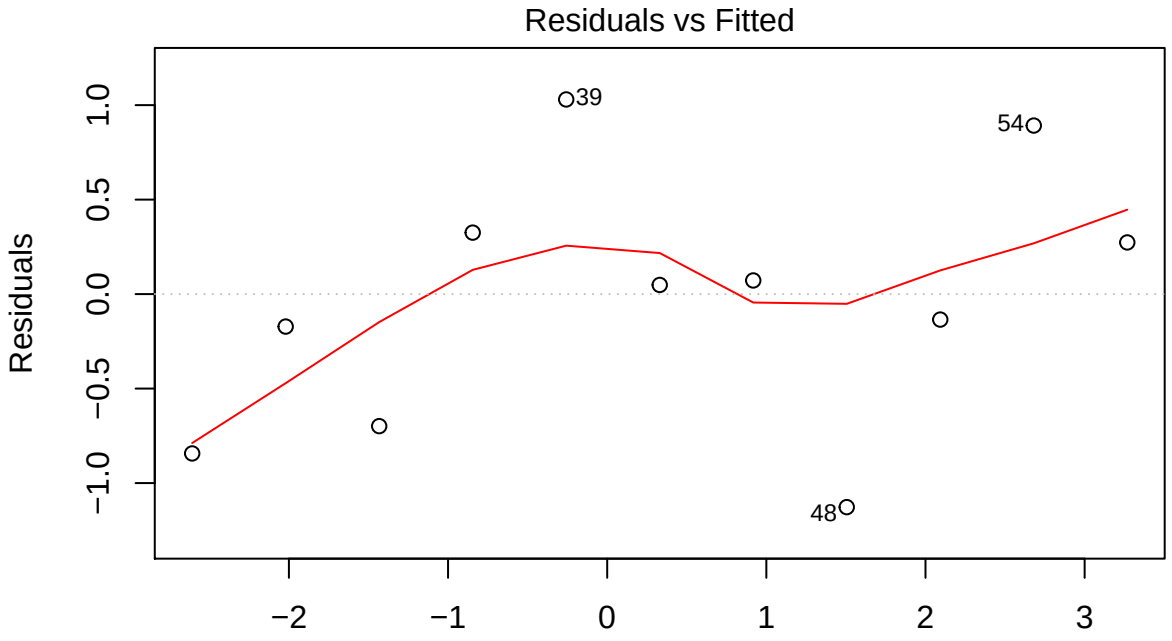
We will now fit a logistic regression model to the data, and conduct diagnostic checks to see whether it is suitable or not.

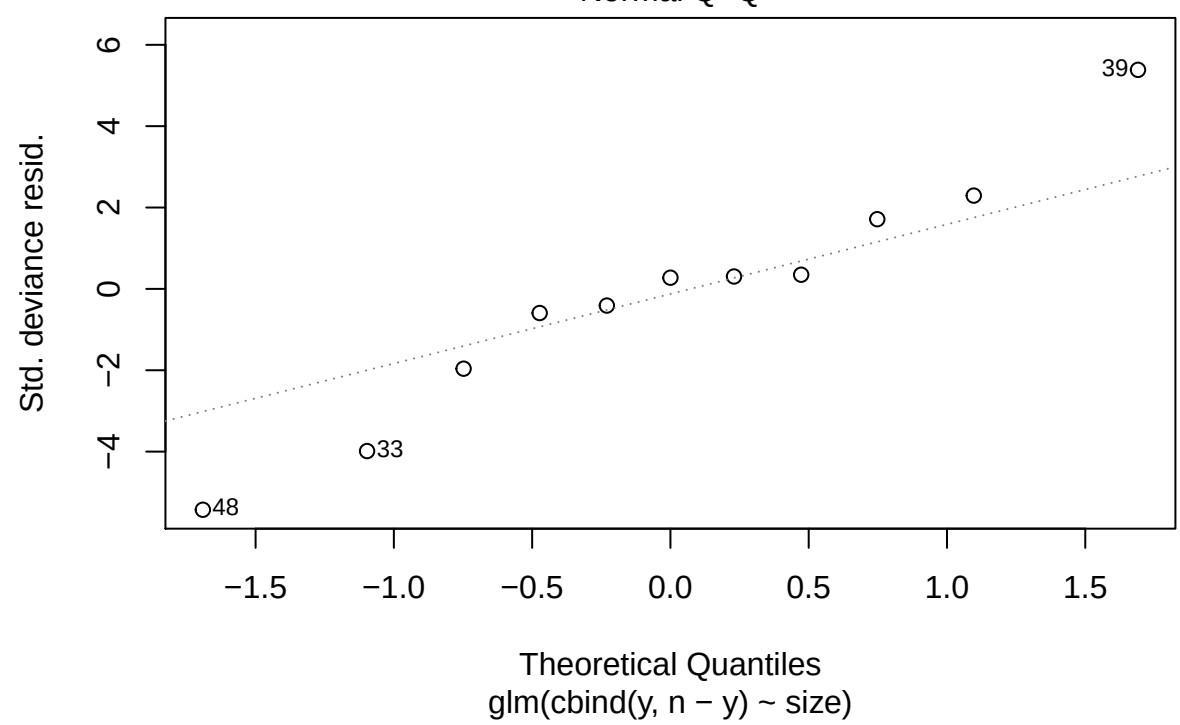
```
# Create Logistic Regression Model
lobster.glm = glm(formula = cbind(y, n - y) ~ size,family = binomial,data = lobster.df)

# Summary and Diagnostics
summary(lobster.glm)
```

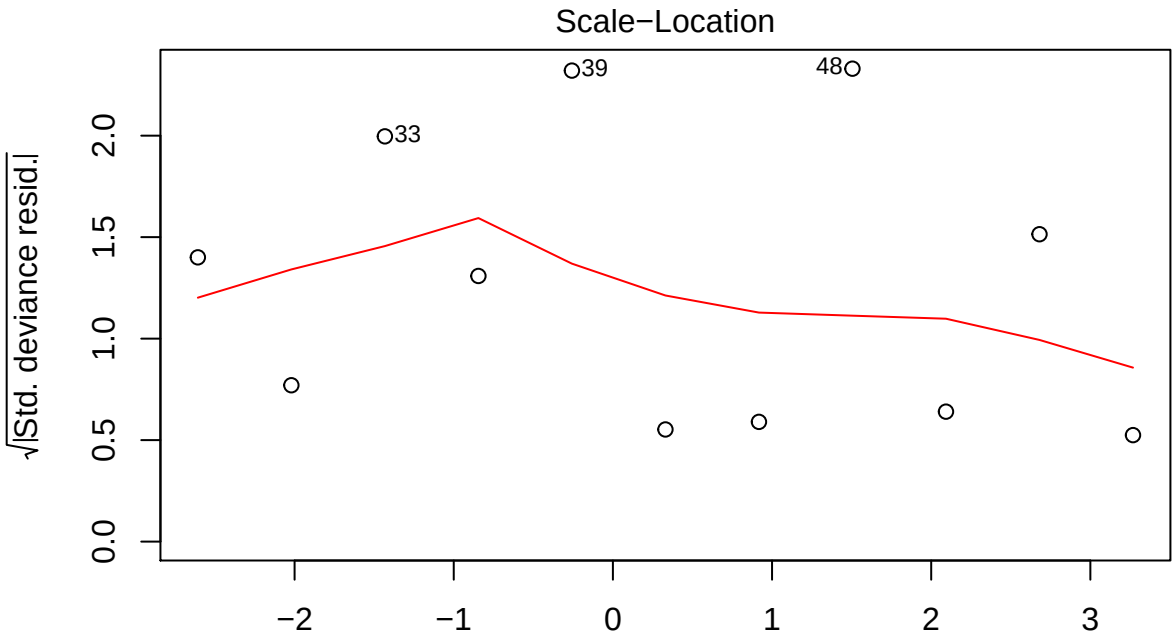
```
##
## Call:
## glm(formula = cbind(y, n - y) ~ size, family = binomial, data = lobster.df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.12729  -0.43534   0.04841   0.29938   1.02995
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.89597    1.38501  -5.701 1.19e-08 ***
## size         0.19586    0.03415   5.735 9.77e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 52.1054  on 10  degrees of freedom
## Residual deviance:  4.5623  on  9  degrees of freedom
```

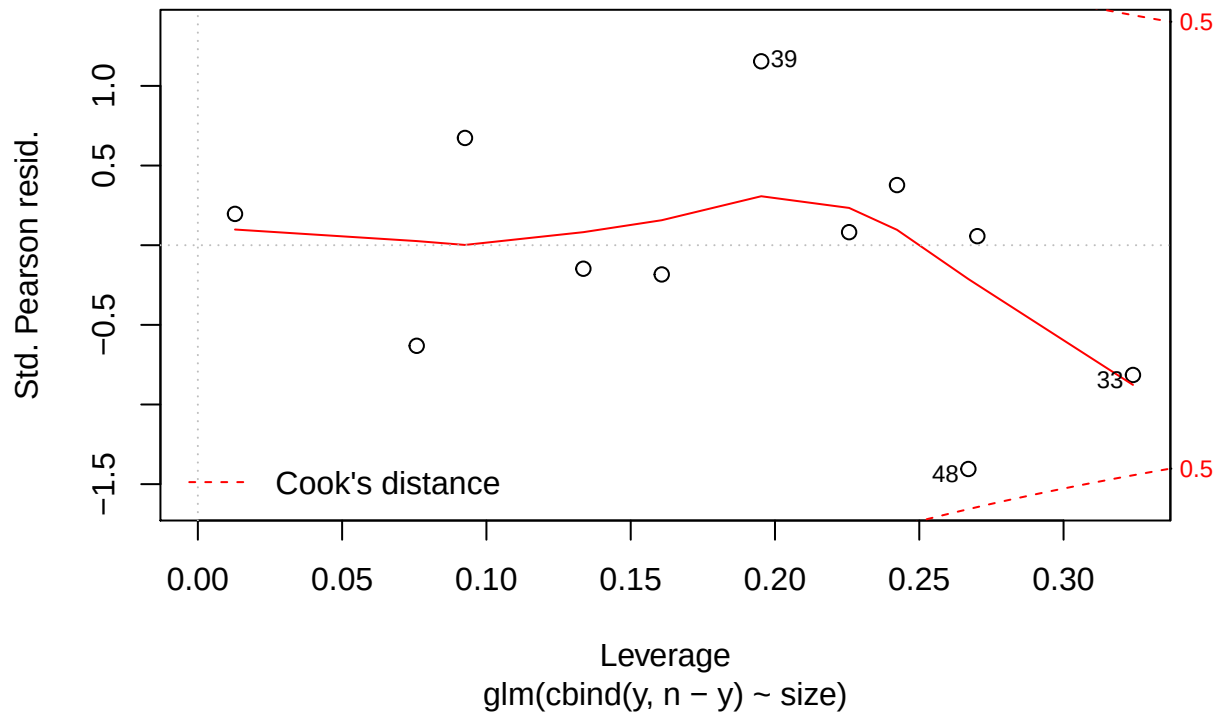
```
## AIC: 32.24
##
## Number of Fisher Scoring iterations: 4
```

```
plot(lobster.glm)
```

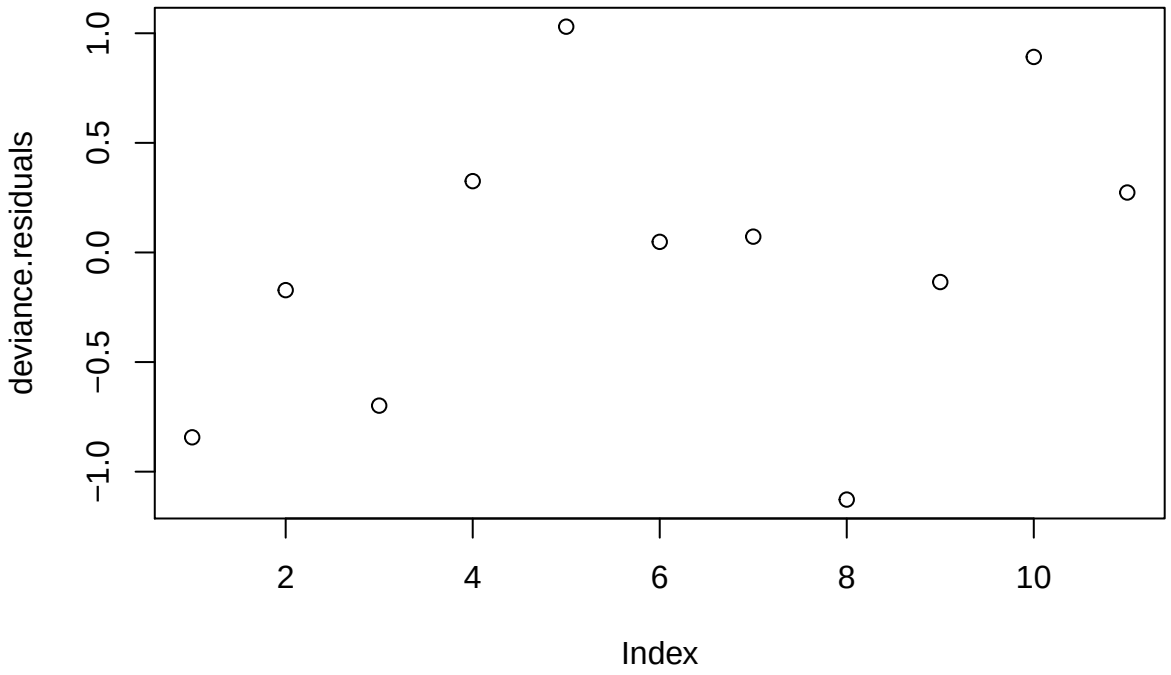




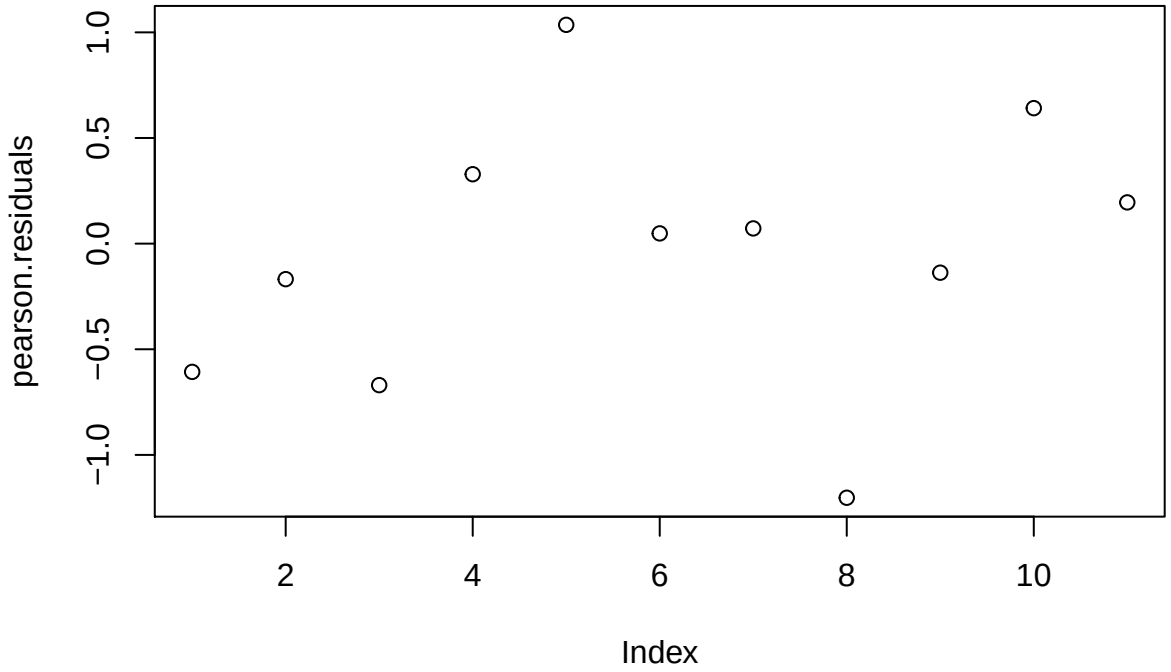




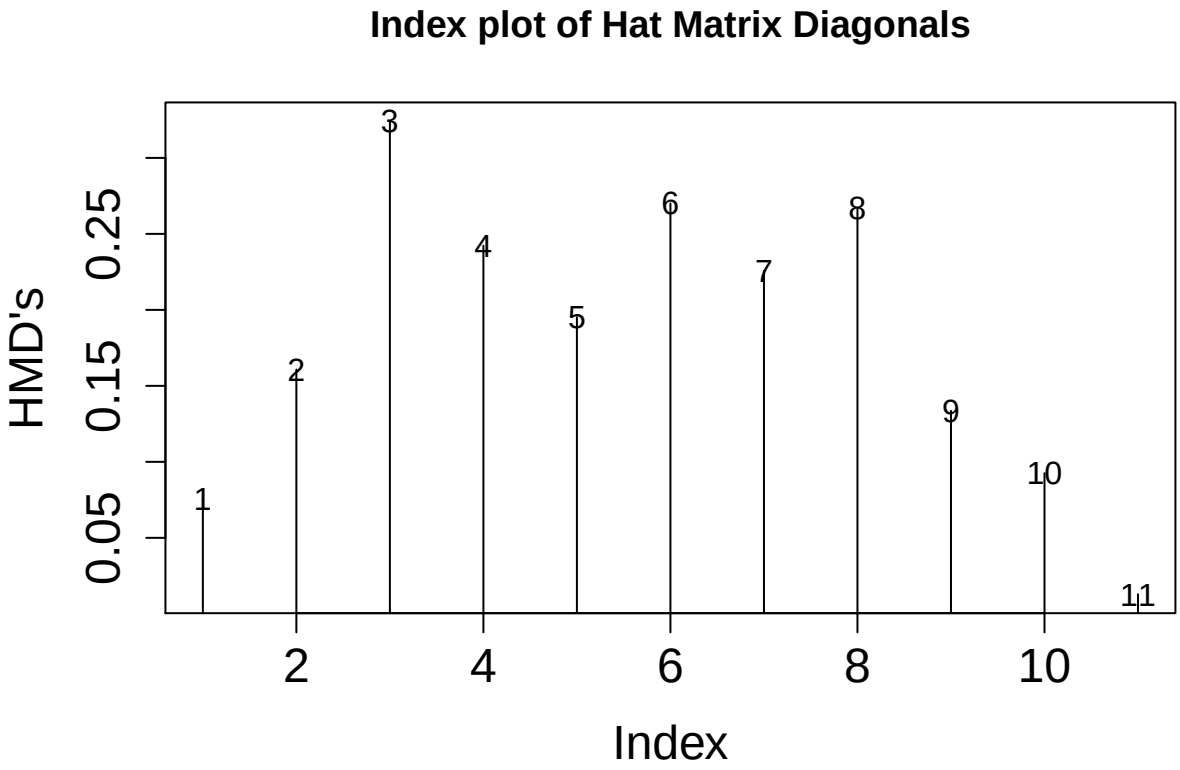

```
# Pearson and Deviance Residuals
pearson.residuals = residuals(lobster.glm, type = "pearson")
deviance.residuals = residuals(lobster.glm, type = "deviance")

plot(deviance.residuals)
```



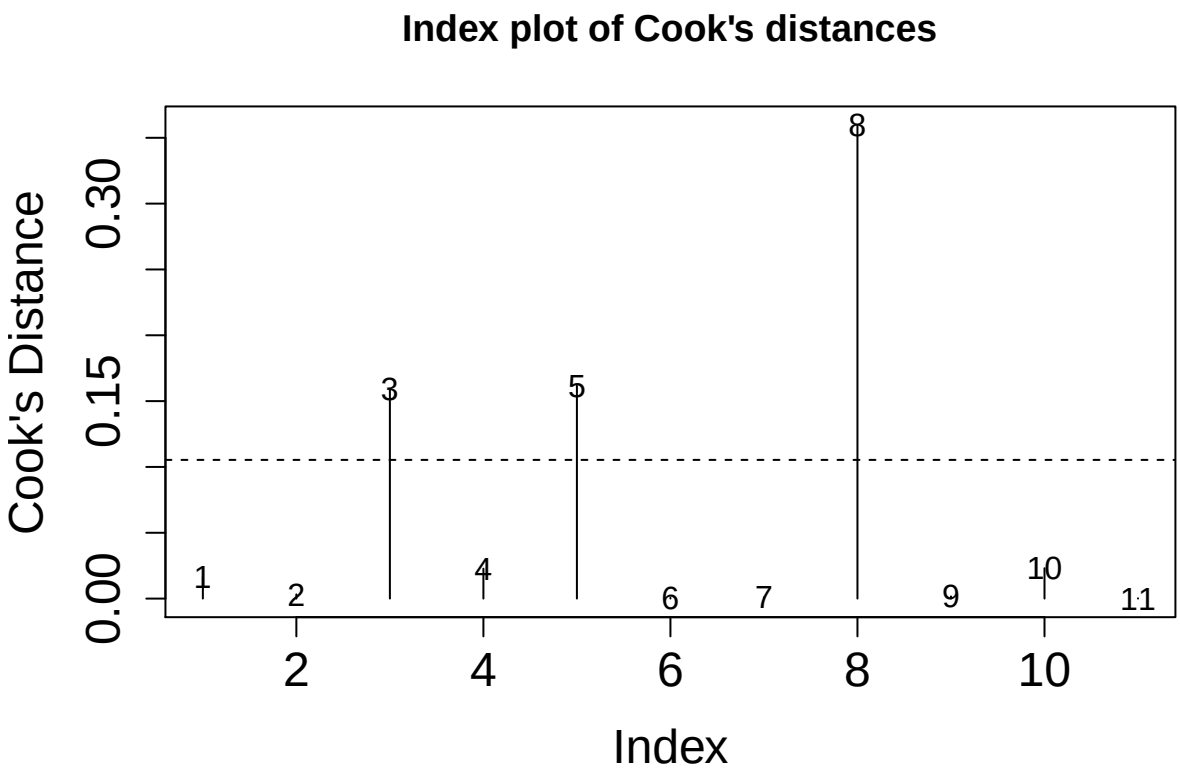

```
plot(pearson.residuals)
```




```
# Hat Matrix Diagonals
hmd = hatvalues(lobster.glm)
plot(hmd, ylab="HMD's", type = "h", cex = 1.5, cex.axis = 1.5,
cex.lab = 1.5, main = "Index plot of Hat Matrix Diagonals")
text(hmd); abline(h = 3 * 2 / 11, lty = 2)
```




```
# Cook's Distance Plot
df = 2
CD = cooks.distance(lobster.glm)
bigcook = qchisq(0.1, df) / df
plot(CD, ylab = "Cook's Distance", type = "h", cex = 1.5, cex.axis = 1.5, cex.lab = 1.5,
     main = "Index plot of Cook's distances")
text(CD); abline(h = bigcook, lty = 2)
```



There are no outliers on the Hat Matrix Diagonals plot, also both the deviance and pearson residuals look fine. However, we have reason for concern that the distribution of our residual deviance may not be a chi-square distribution, because some of the groups have a small number of lobsters. There are also 3 outliers in the Cook's Distance plot, which raises another issue. But those points do not seem to be causing any more problems according to the HMD, and Deviance/Pearson Residual plots. Therefore it's fair to say that the Cook's Distance outliers are not worth deleting. Also, there are only 11 groups of lobsters, which means that we might wipe out a significant portion of our data if we decide to delete observations, so it shouldn't be done unless absolutely necessary.

Theoretical Model:

$$log(Odds_i) = \beta_0 + \beta_1 * Size_i + \epsilon_i$$

Fitted Model:

$$log(Odds) = -7.89597 + 0.19586 * Size$$

```
# Confidence Interval
100 * (exp(confint(lobster.glm)) - 1)
```

```
## Waiting for profiling to be done...
```

```
##                  2.5 %    97.5 %
## (Intercept) -99.99797 -99.52279
## size         14.22485  30.68074
```

From our model, we recieved evidence that the size of a lobster has a significant effect on its odds of surviving predation. We predict that for every mm increase in size, the odds of survival increase between 14.2 and 30.7 percent.

The authors of this study fitted a (standard) linear regression, using size as the explanatory variable and proportion survived as the response variable. They assumed that, for each size group, the proportion that survived came from a normal distribution, with constant variance across all groups.

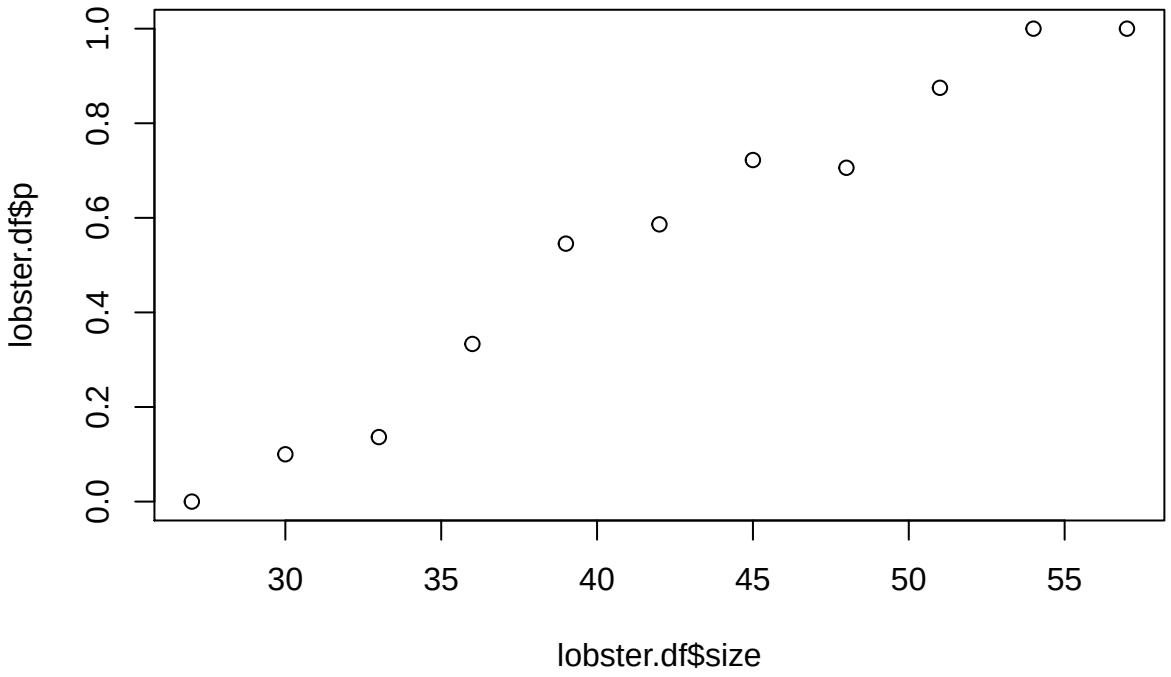
We will fit a standard linear regression to this dataset, and compare our model to the ones the authors of this study used.

```
# Fitting the Linear Model
lobster.lmfit = lm(p~size,data = lobster.df)

# Summary
summary(lobster.lmfit)
```

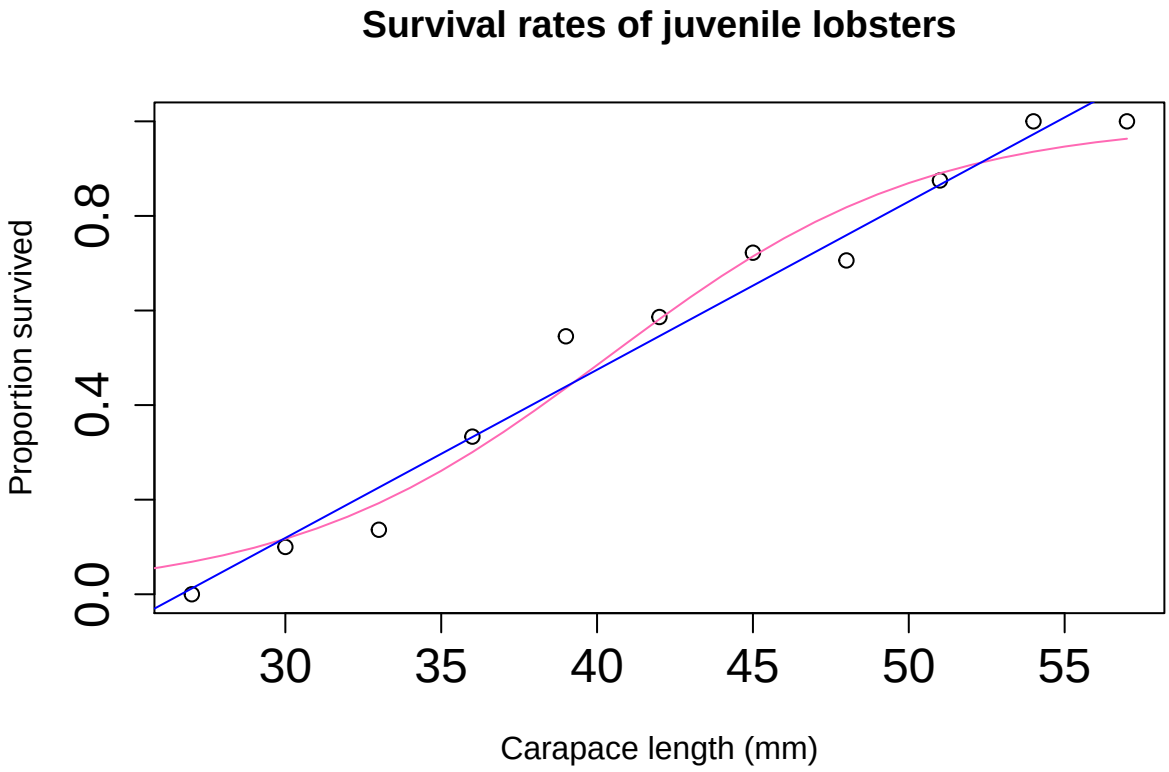
```
##
## Call:
## lm(formula = p ~ size, data = lobster.df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.089376 -0.036212  0.000887  0.033829  0.106301
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.948038   0.086867  -10.91 1.72e-06 ***
## size         0.035569   0.002017   17.63 2.75e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06348 on 9 degrees of freedom
## Multiple R-squared:  0.9719, Adjusted R-squared:  0.9687
## F-statistic: 310.8 on 1 and 9 DF,  p-value: 2.752e-08
```

Below, we plot the relationship between carapace length and odds of survival, with the linear and logistic models both overlaid.

```
plot(lobster.df$p~lobster.df$size)
```




```
sequence.size = seq(0, max(lobster.df$size))
pred.df = data.frame(size = sequence.size)
fitted.values = predict(lobster.glm, pred.df)
fitted.probabilities = exp(fitted.values) / (1 + exp(fitted.values))

plot(p ~ size, cex.axis = 1.5, xlab = "Carapace length (mm)",
     ylab = "Proportion survived", main = "Survival rates of juvenile lobsters", data=lobster.df)
lines(fitted.probabilities ~ sequence.size, col = "hotpink")
abline(a = lobster.lmfit$coefficients[1], b = lobster.lmfit$coefficients[2], col = "blue")
```



Our logistic regression model is better at covering higher variance at the extreme probabilities of survival (for very large and very small lobsters), as well as lower variance towards the middle (for average sized lobsters).

The linear model seems to fit the data, the line of best fit is incorrect for our situation, because it will return probabilities greater than 1 for lobsters of size greater than 55 mm, and below 0 for lobsters below size 25 mm which is not possible. The probability of survival will get closer to 1 as the size of the lobsters increase, and closer to 0 as the size of the lobsters decrease.

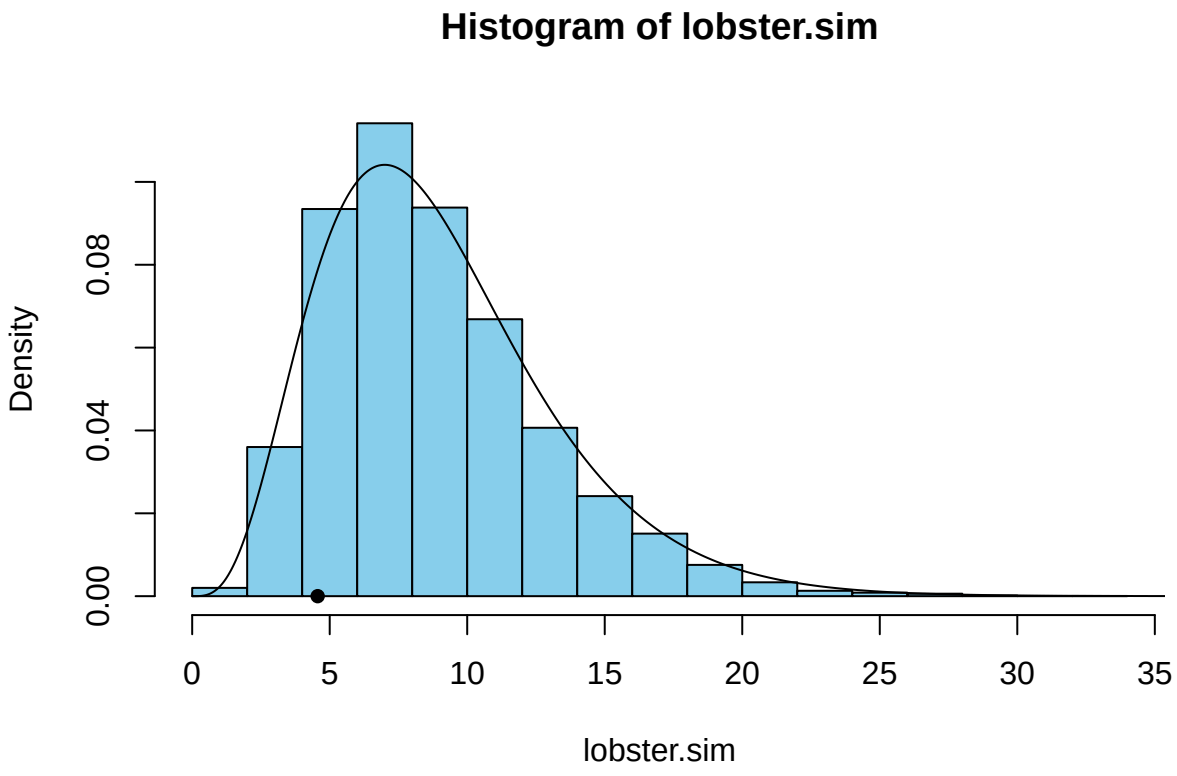
Therefore, it would be best to keep the logistic regression model over the linear regression model, because it is more realistic for this situation.

However, we still have one major issue: the residual deviance is closely approximated by a chi-square distribution when the number of groups is not too large and there is a sizeable number of observations for each group. This condition actually does not hold for our data, because a couple observations have a very small number of trials.

One way to establish the distribution of the residual deviance is to simulate data under the null hypothesis. Let's see if we can calculate the residual deviances using a chi-square distribution

```
set.seed(946706583)

# Simulation size.
n.sims = 10000

# Probabilities of successes under the fitted model.
fitted.ps <- predict(lobster.glm, type = "response")

# Number of observations in the original data set.
n.obs <- nrow(lobster.df)

# Vector to fill with deviances.
lobster.sim = numeric(n.sims)
```

```r
for (i in 1:n.sims){

  # Generating new response data.
  y.sim = rbinom(n.obs, size = lobster.df$n, prob = fitted.ps)

  # Fitting model to simulated data.
  sim.fit = glm(cbind(y.sim, lobster.df$n - y.sim) ~ lobster.df$size, family = "binomial")

  # Saving deviance from this model.
  lobster.sim[i] = sim.fit$deviance
}

# Generating Histogram and Lines
hist(lobster.sim, freq = FALSE, col = "skyblue")
xx = seq(0, max(lobster.sim) + 5, length.out = 1000)
yy <- dchisq(xx, df = lobster.glm$df.residual)
lines(xx, yy)
points(lobster.glm$deviance, 0, pch = 16)
```




```r
# Obtaining a p-value based on the simulated deviances.
mean(lobster.sim >= lobster.glm$deviance)
```

```
## [1] 0.8833
```

The deviances strongly resemble a chi-squared distribution with 9 degrees of freedom. This is also reflected from our very high p-value of 0.8833. Thus, we cannot reject the null hypothesis, which says that the residual deviances can be approximated by a chi-squared distribution.

Therefore, we may approximate our model's residual deviance with a chi-squared distribution.