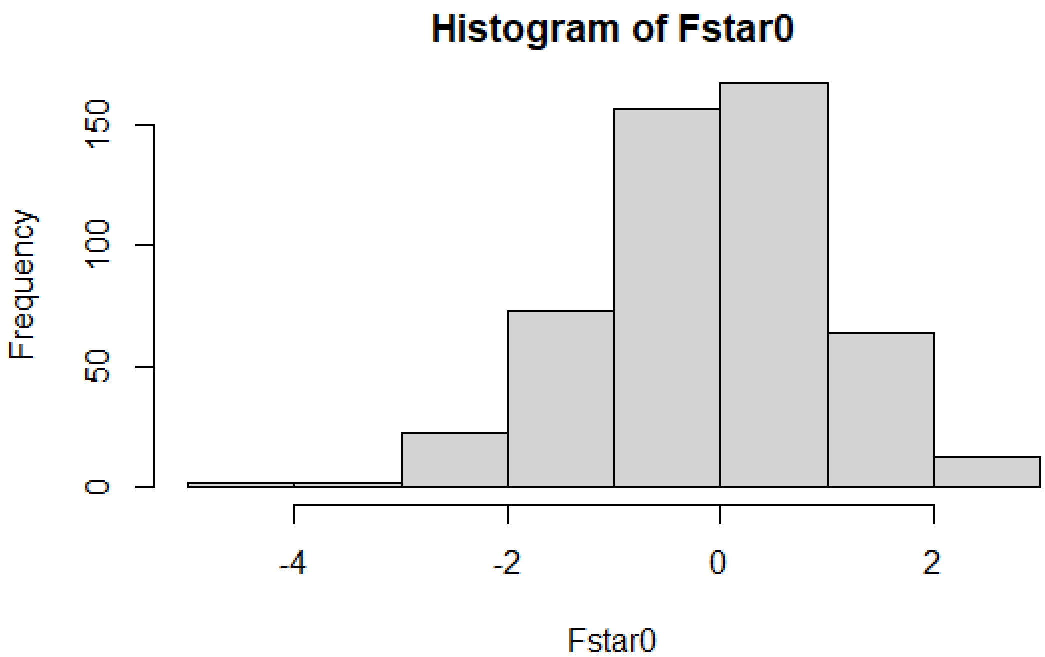
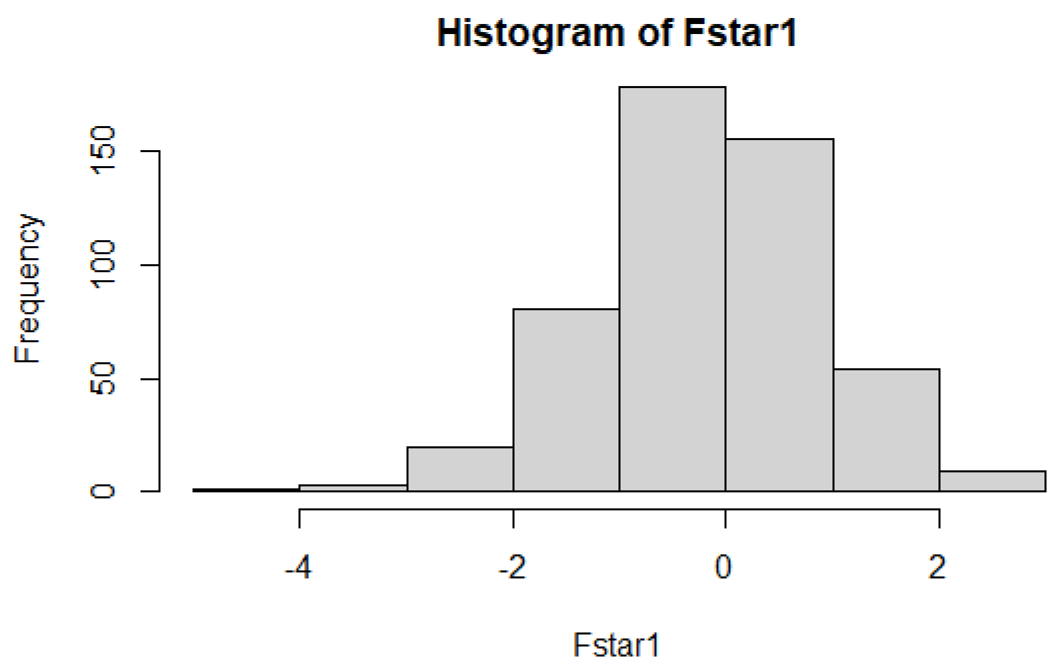
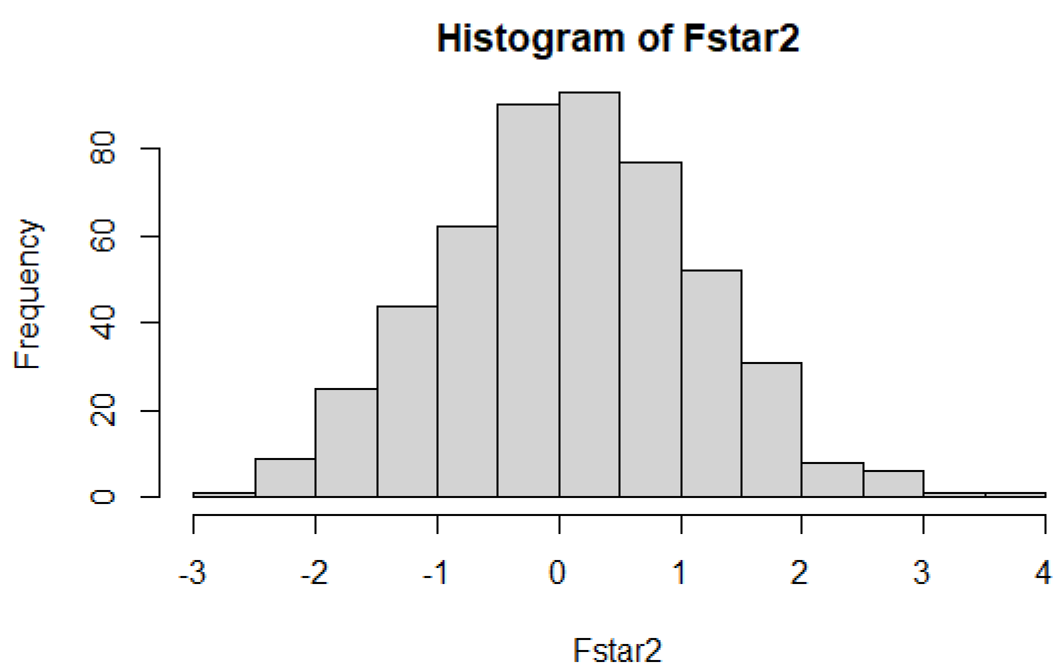
Keven Lin
04/05/2022

Problem 1: Bootstrap

a. Fit a regression model with **`medv`** as your response and **`crim`** and **`age`** as your predictors. Obtain a bootstrapped confidence interval of $\hat{\beta}_{\text{crim}}$. Plot your bootstrapped distribution of $\tilde{F}^{(b)}$.







b. Compare this with the confidence intervals obtained assuming normality using analytical formulas (you can use the `confint()` function). How do they compare your results from (f) compare?

```
                  2.5 %      97.5 %
(Intercept) 27.8933870 31.70794700
crim        -0.4004172 -0.22321431
age         -0.1166275 -0.06247902
[1] "beta0"
    2.5%
27.41891
   97.5%
31.76669
[1] "beta1"
      2.5%
-0.4270035
     97.5%
-0.2307949
[1] "beta2"
      2.5%
-0.1147564
       97.5%
-0.06089034
```

The values are very close together with only a tiny variation between the non bootstrap and bootstrap values. Because bootstrap is a reasonable range for predictors that requires no distribution assumption and can trust the validity of this. Thus because our analytical model is very close to our bootstrap model, we can conclude that the normality or CLT holds.

c. Based on this data set, provide an estimate $\hat{\mu}_{med}$ for the median value of `medv`.

```
c.

```{r}
median(Boston$medv,na.rm=TRUE)
```

 [1] 21.2
```

d. We would like to estimate the standard error of $\hat{\mu}_{med}$. Since there is no simple formula for computing the standard error of the median, bootstrap the standard error. Copy/paste your code and report your standard error here.

````r
```{r}
B = 2000
medianBoot = rep(NA, 2000)
for(b in 1:B) {
  index = sample(1:n, n, replace=TRUE)
  bootstrap = Boston[index, ]

  ## obtain median of horsepower
  medianBoot[b] = median(bootstrap$medv, na.rm = TRUE)
}

sqrt(sum((medianBoot-mean(medianBoot))^2)/(B-1))
```
````

```
[1] 0.3819209
```

e. Using bootstrap, provide a 95% confidence interval for the median of `medv`. Plot your bootstrapped distribution of $\tilde{F}^{(b)} = \frac{\hat{\mu}^{(b)}_{\text{med}} - \hat{\mu}_{\text{med}}}{se(\hat{\mu}^{(b)}_{\text{med}})}$.

f. Based on this data set, provide an estimate $\hat{\mu}_{0.1}$, the 10th percentile of `medv`.

g. Use bootstrap to estimate the standard error of $\hat{\mu}_{0.1}$. Comment on your findings.

Problem 2: Email Spam

We will use a well-known dataset to practice classification. You can find it here: `https://archive.ics.uci.edu/ml/datasets/Spambase`. Read the attribute information and download the dataset onto your computer. To load this data into R, use the follow code:

```
spam = read.csv('.../spambase.data',header=FALSE)
```

The last column of the `spam` data set, called `V58`, denotes whether the e-mail was considered spam (1) or not (0).

a. What proportion of emails are classified as spam and what proportion of emails are non-spam?

```
```{r}
table(spam$V58)
1813 / 2788
```
```

```

   0    1 
2788 1813 
[1] 0.6502869
```

b. Carefully split the data into training and testing sets. Check to see that the proportions of spam vs. non-spam in your training and testing sets are similar to what you observed in part (a). Report those proportions here.

```
b.

```{r}
library(caret)
index = createDataPartition(spam$V58, p = 0.60)
train = spam[as.numeric(index[[1]]),]
test = spam[as.numeric(-index[[1]]),]

table(train$V58)
1105 / 1656
table(test$V58)
708 / 1132
```
```

```

   0    1 
1670 1091 
[1] 0.6672705

   0    1 
1118  722 
[1] 0.6254417
```

Train

Test

c. Fit a logistic regression model here and apply it to the test set. Use the `predict()` function to predict the probability that an email in our data set will be spam or not. Print the first ten predicted probabilities here.

c.

```{r}
model = glm(v58 ~., data = train, family = binomial)
probabilities = model %>% predict(test, type = "response")
head(probabilities, 10)
# Model accuracy
#mean(predicted.classes == test.data$diabetes)
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
        3         6        10        11        14        15        16        17
1.0000000 0.8409085 0.9135889 0.8874578 0.8461742 0.9993670 0.9862302 0.6666816
       21        25
0.2198876 0.1567057
```

P of getting No Spam

d. We can convert these probabilities into labels. If the predicted probability is greater than 0.5, then we predict the email is spam ($\hat{Y}_i = 1$), otherwise it is not spam ($\hat{Y}_i = 0$). Create a confusion matrix based on your results. What's the overall misclassification rate? Break this down and report the false negative rate and false positive rate.

```
d.

```{r}
predictValue = factor(ifelse(probabilities > 0.5, 1, 0))
testValue = factor(test$V58)
confusionMatrix(predictValue, testValue)
```

Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 1060   83
         1   61 636

               Accuracy : 0.9217
                 95% CI : (0.9085, 0.9336)
    No Information Rate : 0.6092
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8347

 Mcnemar's Test P-Value : 0.08012

            Sensitivity : 0.9456
            Specificity : 0.8846
         Pos Pred Value : 0.9274
         Neg Pred Value : 0.9125
             Prevalence : 0.6092
         Detection Rate : 0.5761
   Detection Prevalence : 0.6212
      Balanced Accuracy : 0.9151

       'Positive' Class : 0
```

```
       'Positive' Class : 0

[1] 0.07826087
```

Missclassification

False Negative: 7.26%
False Positive: 8.75%

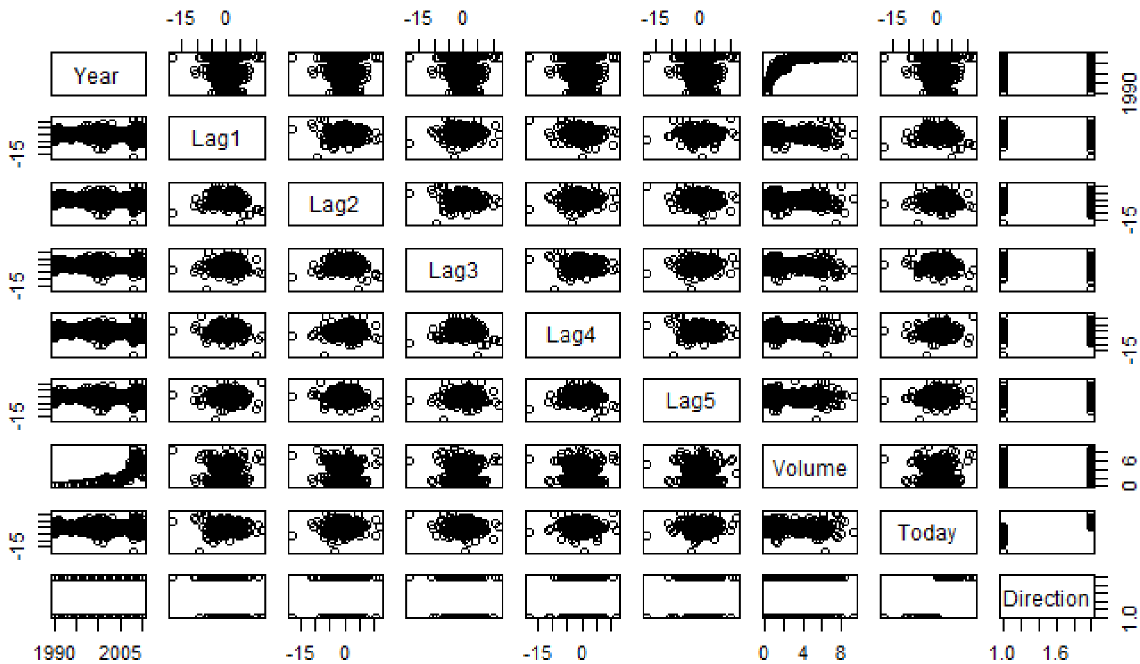
e. What type of mistake do we think is more critical here: reporting a meaningful email as spam or a spam email as meaningful? How can we adjust our classifier to accommodate this?

Reporting something meaningful as spam is a major problem as that could lead the user to miss a legitimate email. We can fix this by raising the probability needed for an email to be classified as spam to say 80 percent.

Problem 3: Weekly Data Set

This question should be answered using the `Weekly` data set, which is part of the `ISLR2` package. This data is similar in nature to the `Smarket` data we saw in class, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

a. Produce some numerical and graphical summaries of the `Weekly` data. Do there appear to be any patterns?



Year and volume has an exponential relationship.

b. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Weekly)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500.4

Number of Fisher Scoring iterations: 4
```

The only predictor that is statistically significant is Lag2.

c. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
Confusion Matrix and Statistics

          Reference
Prediction Down  Up
      Down   54  48
      Up    430 557

               Accuracy : 0.5611
                 95% CI : (0.531, 0.5908)
    No Information Rate : 0.5556
    P-Value [Acc > NIR] : 0.369

                  Kappa : 0.035

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.11157
            Specificity : 0.92066
         Pos Pred Value : 0.52941
         Neg Pred Value : 0.56434
             Prevalence : 0.44444
         Detection Rate : 0.04959
   Detection Prevalence : 0.09366
      Balanced Accuracy : 0.51612

       'Positive' Class : Down

[1] 0.4389348
```

We can see that our model has an accuracy of 44%.

d. Now fit the logistic regression model using a training data period from 1990 to 2008, with `Lag2` as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
Confusion Matrix and Statistics

          Reference
Prediction Down Up
      Down    0  2
      Up     43 59

               Accuracy : 0.5673
                 95% CI : (0.4665, 0.6641)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.6921

                  Kappa : -0.0382

 Mcnemar's Test P-Value : 2.479e-09

            Sensitivity : 0.00000
            Specificity : 0.96721
         Pos Pred Value : 0.00000
         Neg Pred Value : 0.57843
             Prevalence : 0.41346
         Detection Rate : 0.00000
   Detection Prevalence : 0.01923
      Balanced Accuracy : 0.48361

       'Positive' Class : Down

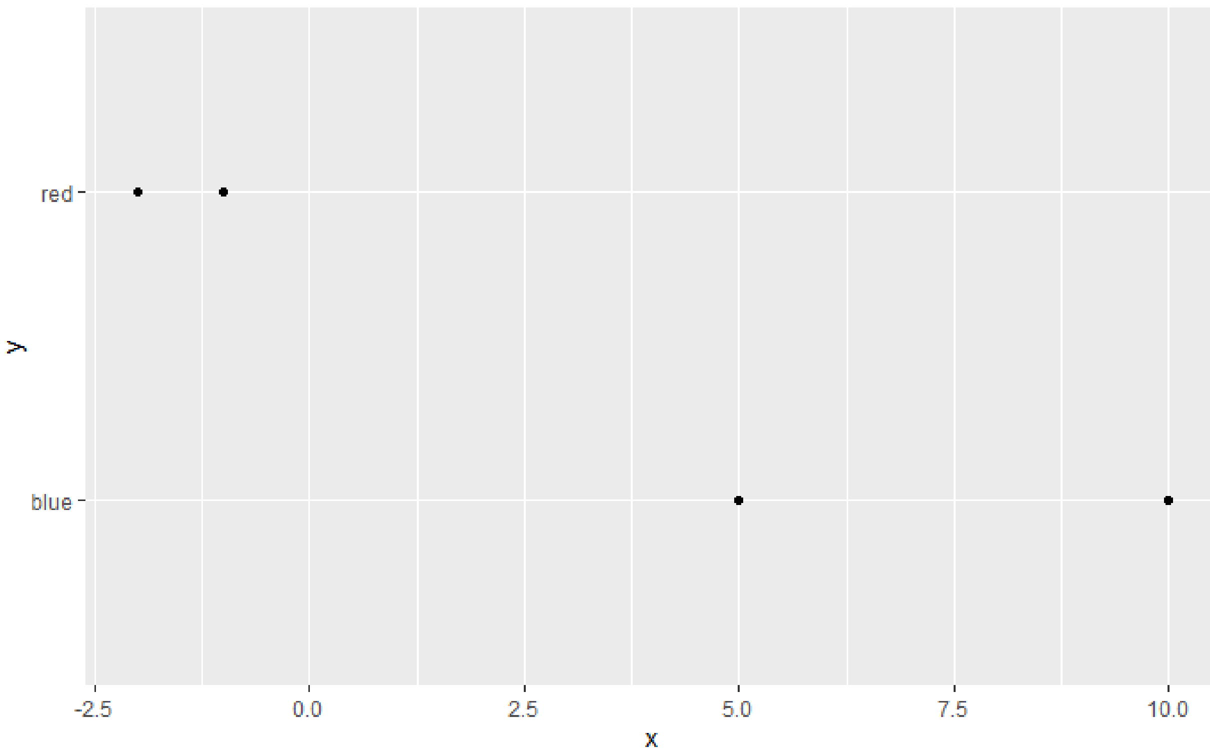
[1] 0.4326923
```

Problem 4: Limitation of Logistic Regression

Consider the dataset:

x	y
-2	red
5	blue
-1	red
10	blue
5	blue

a. Plot the data in R **in a single plot by group** (red vs. blue). What do you observe? Are the two groups well-separated?



b. Fit a logistic regression model on the data. What happens? Report any error message here.

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Call:
glm(formula = y ~ x, family = binomial, data = df)

Deviance Residuals:
         1           2           3           4           5
 2.110e-08  -1.164e-05   1.334e-05  -2.110e-08  -1.164e-05

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    15.38   54271.11   0.000        1
x              -7.76   13792.23  -0.001        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6.7301e+00  on 4  degrees of freedom
Residual deviance: 4.4882e-10  on 3  degrees of freedom
AIC: 4

Number of Fisher Scoring iterations: 23
```

The reason the warning popped up is because y is a factor not a continuous variable.

c. To understand why this happen, we need to understand conceptually what is happening with our logistic regression model. In our setup Y is binary variable that is either red ($Y = 1$) or blue ($Y = 0$). Our model is estimating:

$$P(Y_i = \text{red}|x_i) = \frac{1}{1+\exp(\beta_0+\beta_1 x_i)} \quad \text{and} \quad P(Y_i = \text{blue}|x_i) = \frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}$$

for all $i = 1, 2, 3, 4, 5$. What value(s) of β_0 and β_1 would maximize the likelihood (and therefore be the estimates we would get from fitting this model)? Recall that our likelihood looks like:

$$l(\beta_0, \beta_1, X) = P(Y_1 = \text{red}|\beta_0, \beta_1, x_1) \times P(Y_2 = \text{blue}|\beta_0, \beta_1, x_2) \times \ldots \times P(Y_5 = \text{blue}|\beta_0, \beta_1, x_5).$$

Hint: What is $P(Y_i = \text{blue}|x_i > 4)$? Now what is the $P(Y_2 = \text{blue}|x_2 = 5)$? What values of β_0 and β_1 will get us close to this probability?

```
Call:
glm(formula = y ~ x, family = binomial, data = red)

Deviance Residuals:
[1]  0  0

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.357e+01  1.777e+05       0        1
x           -1.828e-14  1.124e+05       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 0.0000e+00  on 1  degrees of freedom
Residual deviance: 2.3305e-10  on 0  degrees of freedom
AIC: 4

Number of Fisher Scoring iterations: 22
```

Beta0hat = -0.24

Beta1hat = `-1.828e-14`

```
Call:
glm(formula = y ~ x, family = binomial, data = blue)

Deviance Residuals:
         1           2           3
-1.079e-05  -1.079e-05  -1.079e-05

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.357e+01  1.376e+05       0        1
x           -7.461e-16  1.946e+04       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 0.0000e+00  on 2  degrees of freedom
Residual deviance: 3.4957e-10  on 1  degrees of freedom
AIC: 4

Number of Fisher Scoring iterations: 22
```

Beta0hat = -2.357e+01

Beta1hat = -7.461e-16

d. Putting all this together, explain one limitation of the logistic regression model.

Logistic regression models do not handle factors very well. Instead, it prefers to work with continuous variables.