MATH 5900 – ADVANCED DATA ANALYTICS

ASSIGNMENT 2

TAIWO JEGEDE – E00755963

APPLIED DATA SCIENCE

SOFTWARE USED: R

QUESTION 1

An appropriate regression model (i)

i. An appropriate normal linear regression model we could use would be:

$$Accidents=\beta_0+\beta_1 \times Exposure+\beta_2 \times Construction1+\beta_3 \times Construction2+\beta_4 \times Construction3+\beta_5 \times Operational + \beta_6 \times service\_months +\varepsilon$$

Where:

Accidents: Accidents is the number of health-related accidents reported over one-month periods.

Exposure: Exposure is the continuous score of exposure to aggressive contact.

Construction1, Construction2 and Construction3 are indicators for different eras of ship construction.

Operational: Operational is an indicator of whether the ship had been operational for more than 10 years.

Service months:

β_0 is the intercept term

β_1 to β_6 are the coefficients associated with each respective variable.

ε is the error term.

Code:

```
# Fit the linear regression model

model <- lm(accidents ~ exposure + construction1 + construction2 + construction3 + operational + service_months, data = ShipData)

# Display the summary of the model

summary(model)
```

Output:

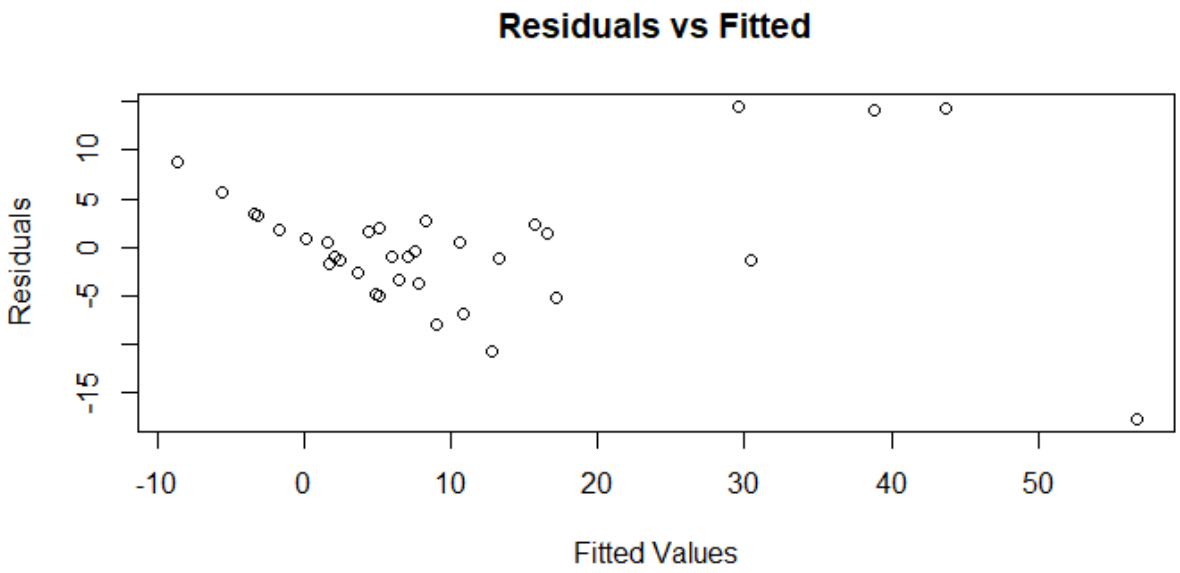
```
Call:
lm(formula = accidents ~ exposure + construction1 + construction2 +
    construction3 + operational + service_months, data = ShipData)

Residuals:
     Min       1Q   Median       3Q      Max
-17.7207  -3.1917  -0.7128   2.1686  14.4716

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.115e+01  1.108e+01  -1.909  0.06696 .
exposure        2.936e+00  1.570e+00   1.871  0.07229 .
construction1   1.193e+00  4.912e+00   0.243  0.81001
construction2   5.883e+00  4.270e+00   1.378  0.17958
construction3   5.200e+00  4.254e+00   1.222  0.23221
operational     4.474e+00  2.765e+00   1.618  0.11730
service_months  1.008e-03  2.739e-04   3.679  0.00103 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.278 on 27 degrees of freedom
  (6 observations deleted due to missingness)
Multiple R-squared:  0.825,     Adjusted R-squared:  0.7861
F-statistic: 21.21 on 6 and 27 DF,  p-value: 4.769e-09
```

Listing the assumptions of the model

i. Linearity
ii. Independence
iii. Constant variance
iv. Normality

The assumption of the models

i. Linearity: The relationship between the predictors and the response variable is assumed to be linear.



Interpretation: From the plot, we can see that there is no linearity between the residuals and the predicted values because there is not a sign of randomness, which means transformations of the existing variable.

ii. Independence: The residuals should be independent of each other.

```
20  # Calculate residuals
21  residuals <- residuals(model)
22
23  # Calculate Durbin-Watson statistic
24  dw_statistic <- sum(diff(residuals)^2) / sum(residuals^2)
25
26  # Print Durbin-Watson statistic
27  print(dw_statistic)
```

20:1 (Top Level) R Script

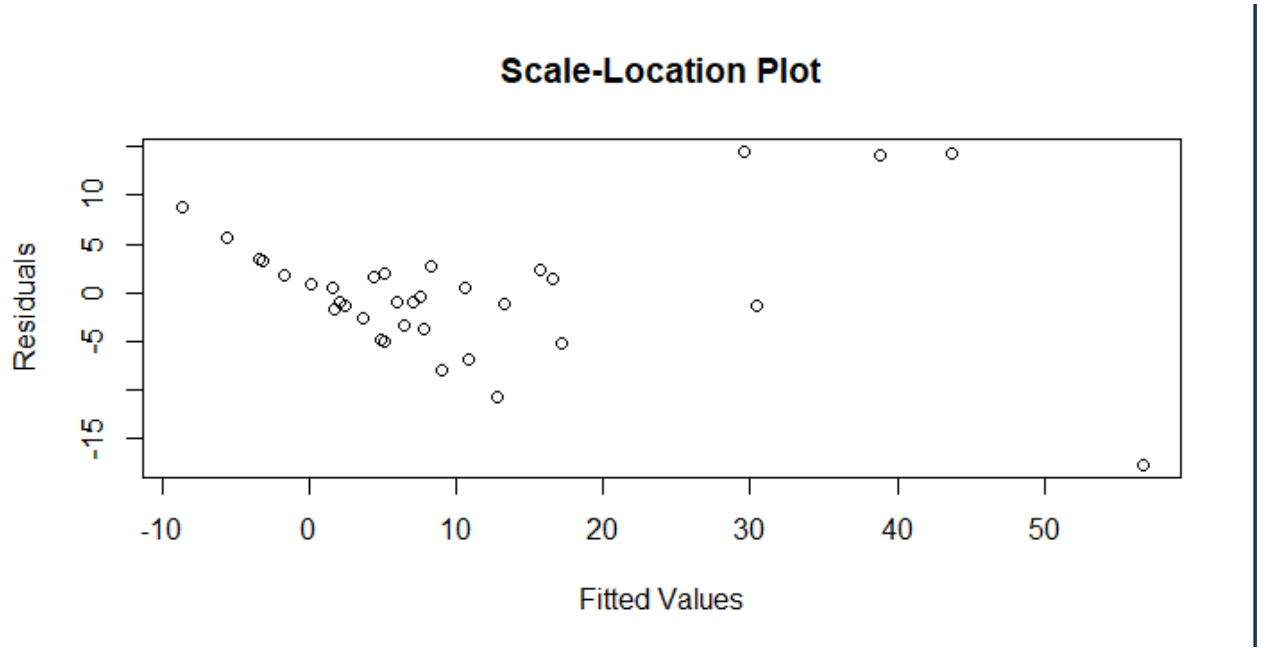
Console Terminal Background Jobs

R 4.3.2 · C:/Users/student75/Desktop/Assignements and Project/MATH5900/Assignment 2/

```
> plot(model$fitted.values, model$residuals, xlab = "Fitted Values", ylab =
"Residuals", main = "Residuals vs Fitted")
> # Calculate residuals
> residuals <- residuals(model)
>
> # Calculate Durbin-Watson statistic
> dw_statistic <- sum(diff(residuals)^2) / sum(residuals^2)
>
> # Print Durbin-Watson statistic
> print(dw_statistic)
[1] 1.858368
>
```

Interpretation: There seems to be no autocorrelation because the statistic is less than 2, it suggests no autocorrelation.

iii. Constant variance: The variance of the residuals should be constant across all levels of the predictors.



We can also check for the constant variance using the formal method

Code:

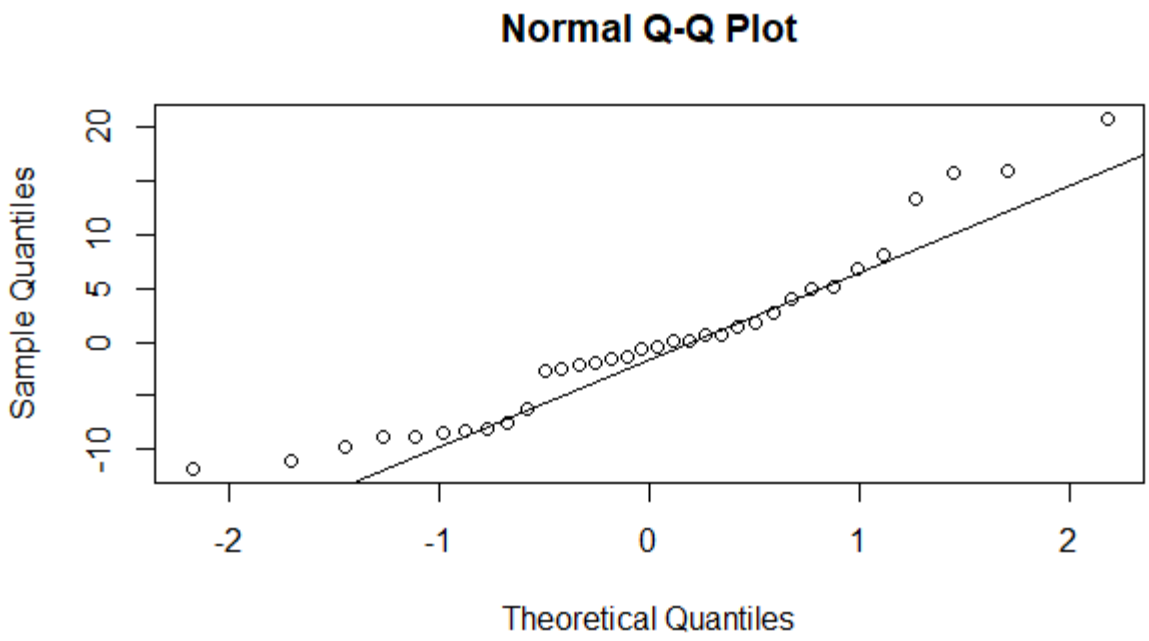
```
ncvTest(model)
```

Output:

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 34.30046, Df = 1, p = 4.7227e-09
```

From our above result, we can see that the the p value is less than 0.05 which makes it not a good fit

iv. Normality: The residuals should follow a normal distribution.



The above plot does not follow a normal distribution therefore the normality assumption is not met.

Also, using the Anderson-Darling test, we can check for Normality by using the code

```
ad.test(model$residuals)
```

Output:

```
        Anderson-Darling normality test

data:  model$residuals
A = 0.91897, p-value = 0.01721
```

From the results from our Anderson-Darling test, we can see that the p-value is below 0.05, which means the normality assumption is not met.

Normality assumptions and what happens when we ignore it.

Ignoring the normality assumption violations could lead to inefficient parameter estimations, which would reduce the estimates' precision. And even in certain circumstances, it may result in inaccurate inferences on the statistical significance of predictors. Deviations from normalcy in residuals may point to problems with the model and call for changes or evaluation of other models.

QUESTION 2

Purpose of using transformation (i)

Whenever we use transformations to accommodate departures from normality, what we do is modify the data to make it more normally distributed. So, whenever we need to meet the assumptions of the statistical model, particularly assumptions of a normally distributed residual, we use Transformation.

We select the transformation to use based on the visual inspection of our data, and sometimes through statistical techniques like Box- Cox transformation.

Transforming and justifying number of monthly accidents (ii)

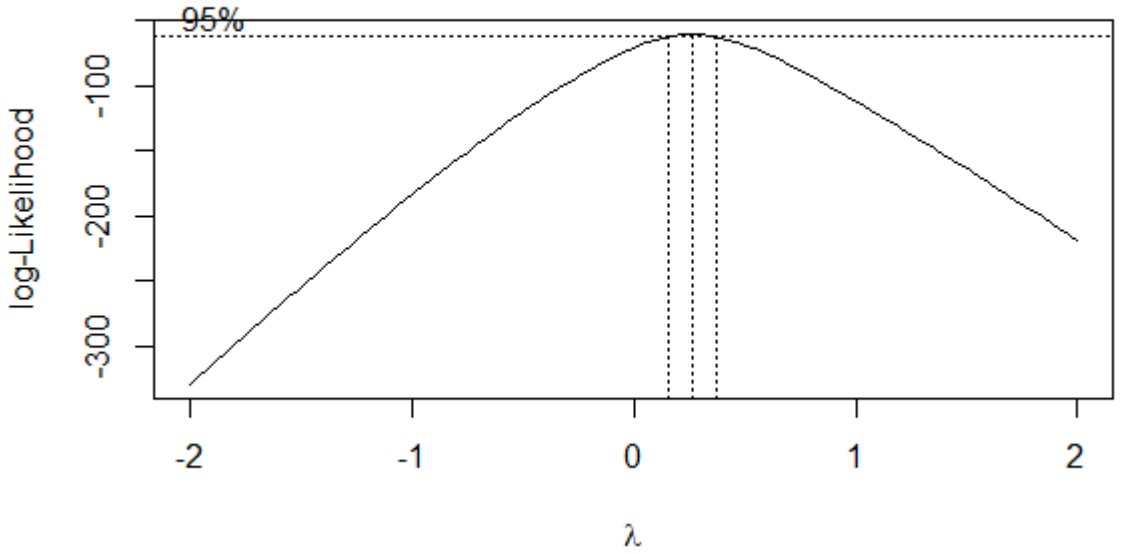
In this particular scenario, I am going to apply a box cox transformation to my model.

Code:
```
library(MASS)
bct <- boxcox(model)
```

But after running the code, an error occurred, my response variable needs to be positive. So to correct that I added 0.01 to my response variable and performed that transformation again.

Code:
```
# Add 0.01 to the response variable to make it positive
ShipData$accidents <- ShipData$accidents + 0.01
bct <- boxcox(model)
```

Output:



From our output, we can see that the Lambda (λ) value is more close to 0 than it Is to 1 and what this suggest is that there needs to be a transformation. And according to the considerations when dealing with Box Cox transformation, the best choice of transformation is that we take the Logarithm of the response variable.

So, using the below code:

```
# Transform the outcome variable
ShipData$log_accidents <- log(ShipData$accidents)
```

Then, we re-fit the linear model into our model using the code below

Refitting the transformed linear model (iii)

```
# Fit the linear model with transformed response variable
lm_model <- lm(log_accidents ~ exposure + construction1 + construction2 +
construction3 + operational + service_months, data = ShipData)
```

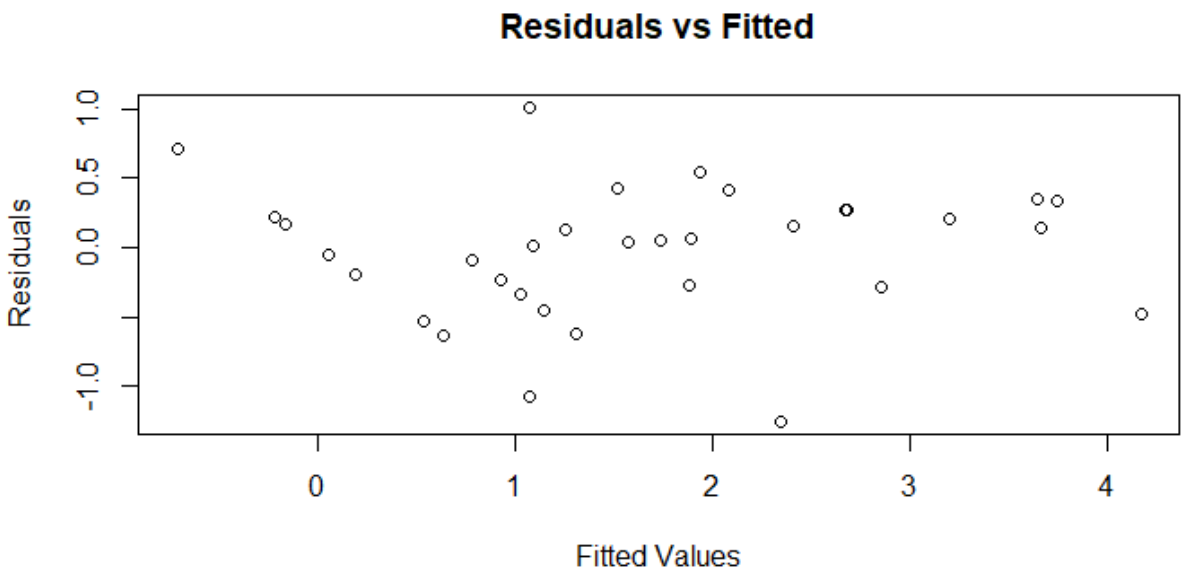
To evaluate the model assumptions, we check for linearity, independence of variance, constant variance, and normality.

Linearity

Code:

```
plot(lm_model$fitted.values, lm_model$residuals, xlab = "Fitted Values", ylab =
"Residuals", main = "Residuals vs Fitted")
summary(lm_model)
```

Output:

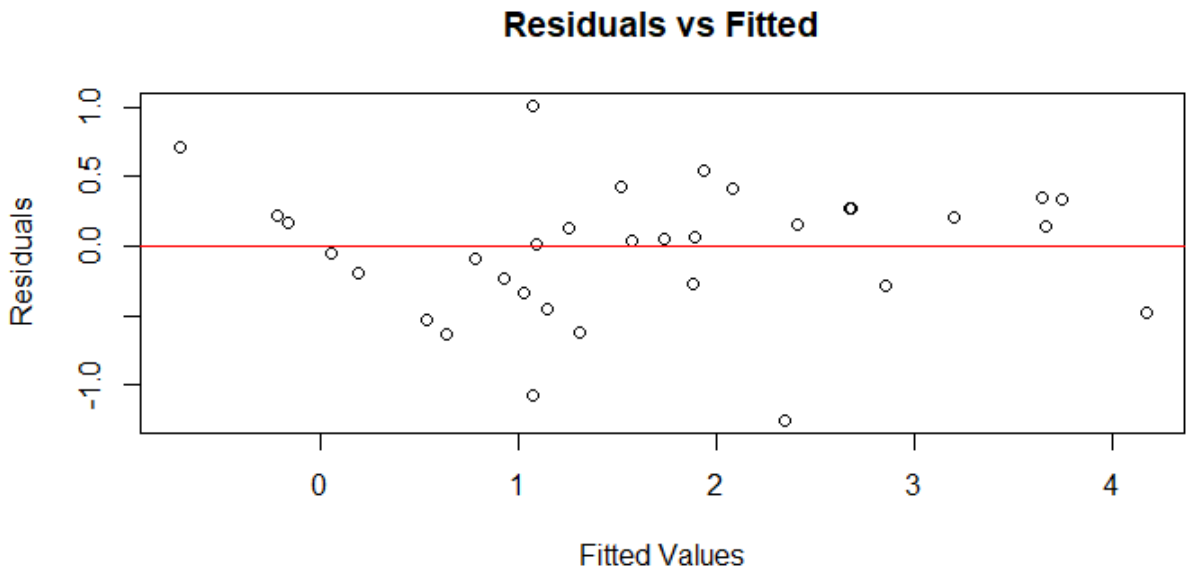


From our plot, we can see that there is a bit of randomness here as opposed to the one we had in Question 1, and this shows linearity from the origin 0.

Constant Variance

Code:

```
# Visualize constant variance assumption

plot(lm_model$fitted.values, lm_model$residuals, xlab = "Fitted values", ylab = "Residuals", main
= "Constant variance Check", pch = 19)

abline(h = 0, col = "red")
```



Using the formal method:

```
ncvTest(lm_model)
```

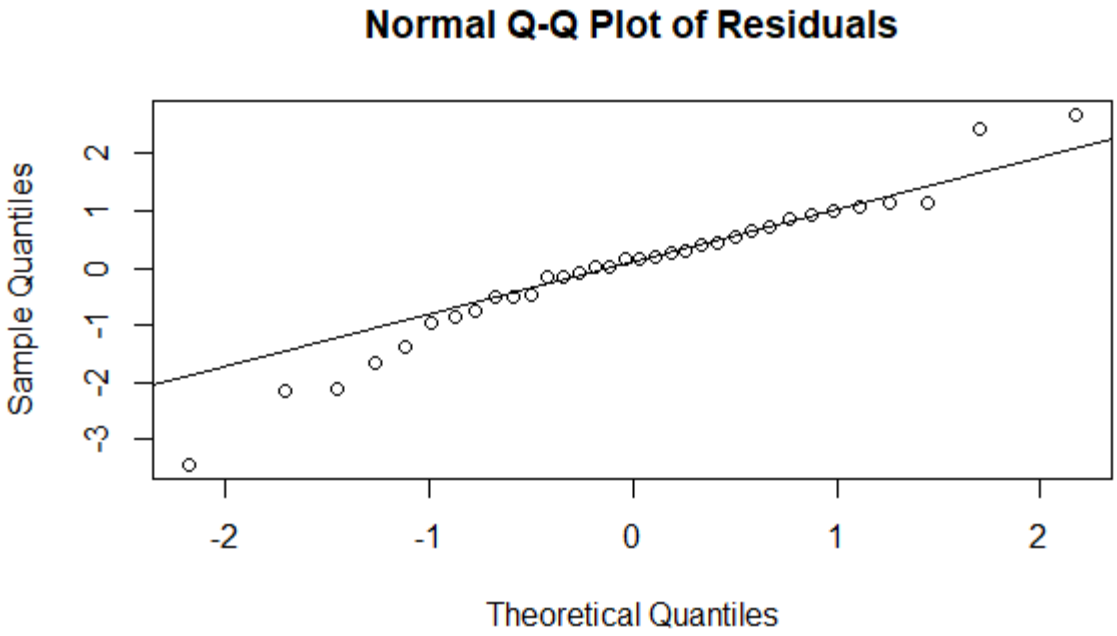
```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 3.073677, Df = 1, p = 0.079569
```

From the result of the code, we can see that the p-value is greater than 0.05, which means that there isn't a reason to reject null hypothesis of constant variance.

Normality

Code:

```
# Check normal probability plot

qqnorm(lm_model$residuals, main = "Normal Q-Q Plot of Residuals")

qqline(lm_model$residuals)
```



Interpretation: From the above visualization we can see that the plots fall into the line at a certain consistency rate, and they are actually normally distributed,so we can say that the normality assumption was met.

We can also perform the Shapiro-Wilk test, to check for normality

Code:
```
# Perform Shapiro-Wilk test for normality

shapiro.test(lm_model$residuals)
```

Output:
```
        Shapiro-Wilk normality test

data:  lm_model$residuals
W = 0.9606, p-value = 0.2531
```

From the result, the p-value is greater than 0.05, so it suggest that there isn't a great depature form normality.

QUESTION 3

Description of the data to be analyzed (i)

The type of data being analyzed as the response (accidents) is categorical or binary data. This means that we will have just 0s and 1s, and the response variable indicates whether there are any monthly accidents ("yes" or "no"). This type of data is commonly referred to as binary or dichotomous, where each observation falls into one of two categories.

Descriptive statistics (ii)

Creating a Binary response variable and the Descriptive Statistics:

Code:

```
# CREATE A BINARY RESPONSE VARIABLE #
binary_acc = ifelse(ShipData$accidents > 0, 1, 0)
summary(binary_acc)
```

Output:

```
> summary(binary_acc)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    1.00    0.65    1.00    1.00
```

```
# Descriptive statistics of the binary response variable
table(binary_acc, ShipData$accidents)
table(ShipData$exposure, binary_acc)
table(ShipData$construction1, binary_acc)
table(ShipData$construction2, binary_acc)
table(ShipData$construction3, binary_acc)
```

```
> table(binary_acc, ShipData$accidents)

binary_acc  0  1  2  3  4  5  6  7 11 12 18 29 39 44 53 58
         0 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
         1  0  5  2  1  2  1  2  2  2  2  2  1  1  1  1  1
```

```
> table(ShipData$exposure, binary_acc)
           binary_acc
             0 1
  3.8066625  1 0
  4.1431347  1 0
  4.6539604  1 0
  4.8441871  1 0
  5.2574954  1 0
  5.5254529  1 0
  5.6131281  0 1
  5.6629605  1 0
  5.8550719  0 1
  6.0799332  0 1
  6.295266   0 1
  6.313548   0 1
  6.5161931  0 1
  6.6605751  1 0
  6.6631327  0 1
  6.6707663  0 1
  6.9985096  0 2
  7.0535857  0 1
  7.0724219  0 1
  7.0967214  0 1
  7.32118806 0 1
  7.5745585  0 1
  7.6260828  0 1
  7.6783264  0 1
  7.7160153  0 1
  8.1176107  0 1
  8.8627667  0 1
  8.8702416  0 1
  9.4802912  0 1
  9.7512683  0 1
  9.9218185  0 1
  10.261477  0 1
```

```
table(ShipData$construction1, binary_acc)
 binary_acc
   0  1
0  8 22
1  6  4
table(ShipData$construction2, binary_acc)
 binary_acc
   0  1
0 11 19
1  3  7
table(ShipData$construction3, binary_acc)
 binary_acc
   0  1
0 14 16
1  0 10
```

Proposing an appropriate model for the binary response (iii)

Logistic regression is a suitable model for the binary response and it's given by

Code:

```
logist_model <- glm(binary_acc ~ exposure + construction1 + construction2 + construction3, family = binomial, data = ShipData)
```

Among the assumptions made by logistic regression are:

i. The relationship is appropriated
ii. Predictors needs to be appropriate (no collinearity)
iii. Good fit.

Evaluating the fit of the model using two approaches (iv)

Evaluating the fit using the Hosmer-Lemeshow test and the ROC Curve.

Hosmer-Lemeshow test:

Code:
```
install.packages("ResourceSelection")
library(ResourceSelection)
hoslem.test(binary_acc,logist_model$fitted.values)
```

Output:

```
> library(ResourceSelection)
> hoslem.test(binary_acc,logist_model$fitted.values)

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  binary_acc, logist_model$fitted.values
X-squared = 3.2354, df = 8, p-value = 0.9187
```

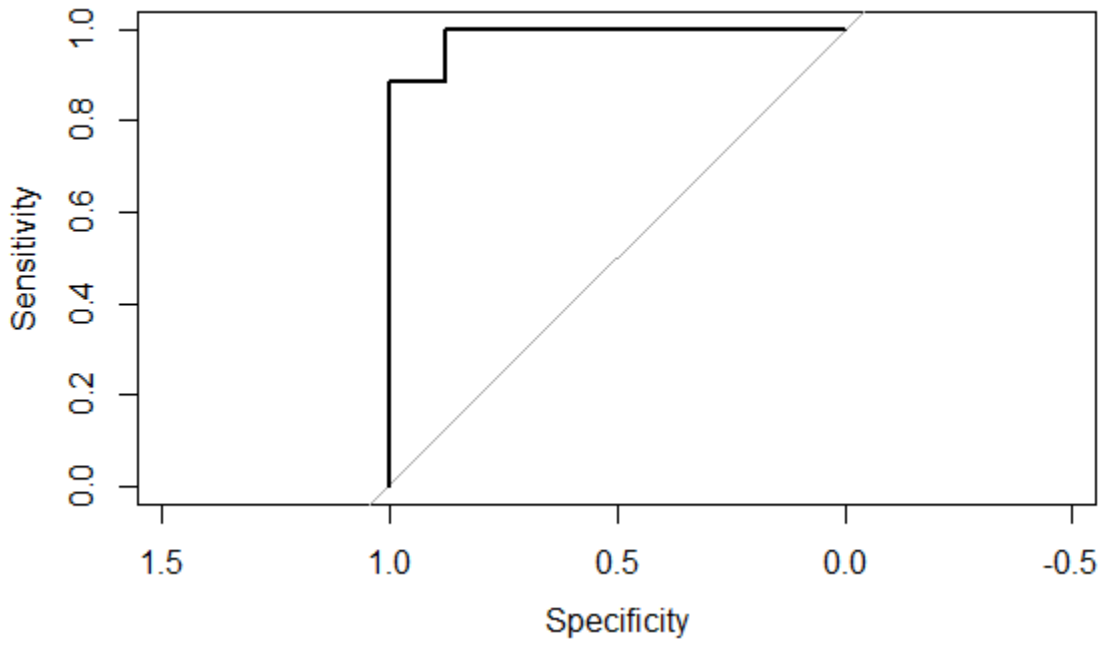
Interpretation:

With a p-value of 0.9187 being greater than our usual significance level of 0.05, we fail to reject the null hypothesis, because it suggest that we don't have evidence to say that the model does not fit the data well, based on the Hosmer and Lemeshow goodness-of-fit test I performed. Therefore, the logistic regression model appears to have a good fit to the data.

ROC Curve:

Code:
```
install.packages("pROC")
library(pROC)
roc(binary_acc~logit_model$fitted.values,plot=TRUE)
```

Output:




```
Call:
roc.formula(formula = binary_acc ~ logist_model$fitted.values,     plot = TRUE)

Data: logist_model$fitted.values in 8 controls (binary_acc 0) < 26 cases (binary_acc 1).
Area under the curve: 0.9856
```

Interpretation:

Looking at the plot, the ROC Curve gears towards 1 (the top left corner), which indicates a better classifier.

From the output I got, the Logistic regression model has a very good discriminatory power as we can see the AUC value is 0.9856. This simply means that the model does a great job in differentiating the observations with and without monthly accident based on the predicted probabilities.

Significance of parameter estimates (v)

Parameter estimates in logistic regression is the change in log odds of the outcome associated with a one-unit change in the predictor variable, holding other variables constant.

We use p-values to check for parameter estimates, so if a p-value is significant, we can say that the predictor variable has a statistically significant effect on the log odds of the outcome.

Benefits of using Logistic Regression Model (vi)

i. Logistic regression is specifically designed for binary response variables, because it provides estimates of probabilities and odds ratios.

ii. When it models the relationship between predictors and the probability of the outcome, it does not assume linearity or normality

PICKING A DATASET

After exploring the links shared, I finally got a dataset titled “NYPD Shooting Incident Data (Historic)” from https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic. This dataset contains 21 columns and a total of 27,313observations. It contains missing values and also contain Latitudinal and Longitudinal data. I have also attached it to the dropbox as a CSV file.

The columns contains:

1. INCIDENT_KEY: A unique identifier for each incident.
2. OCCUR_DATE: The date when the incident occurred in MM/DD/YYYY format.
3. OCCUR_TIME: The time when the incident occurred in HH:MM:SS format.
4. BORO: The borough (district) where the incident occurred.
5. LOC_OF_OCCUR_DESC: Description of the location of the incident.
6. PRECINCT: The precinct associated with the incident.
7. JURISDICTION_CODE: The jurisdiction code associated with the incident.
8. LOC_CLASSFCTN_DESC: Description of the classification of the location where the incident occurred.
9. LOCATION_DESC: A description of the location where the incident occurred.
10. STATISTICAL_MURDER_FLAG: A binary indicator (TRUE/FALSE) whether the incident is classified as a murder.
11. PERP_AGE_GROUP: Age group of the perpetrator.
12. PERP_SEX: Gender of the perpetrator.
13. PERP_RACE: Race of the perpetrator.
14. VIC_AGE_GROUP: Age group of the victim.
15. VIC_SEX: Gender of the victim.
16. VIC_RACE: Race of the victim.
17. X_COORD_CD: X coordinate of the location where the incident occurred.
18. Y_COORD_CD: Y coordinate of the location where the incident occurred.
19. Latitude: Latitude of the location where the incident occurred.
20. Longitude: Longitude of the location where the incident occurred.
21. Lon_Lat: Combined field indicating the longitude and latitude of the location where the incident occurred.