MATH 5900 – ADVANCED DATA ANALYTICS

ASSIGNMENT 1

TAIWO JEGEDE – E00755963

APPLIED DATA SCIENCE

SOFTWARE USED: R, Python and SAS

QUESTION 1

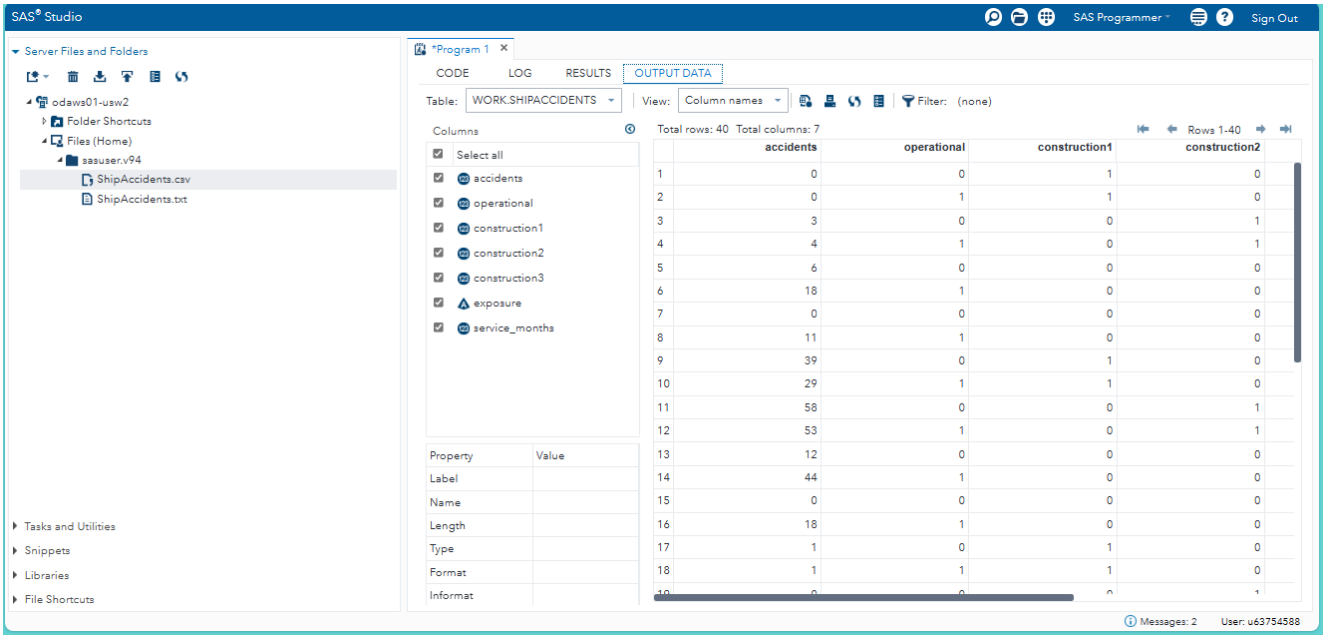
i. READING THE DATA INTO CSV AND TEXT USING SAS

Reading the data into SAS:

READING CSV

```
/* Importing a CSV File using SAS */

PROC IMPORT DATAFILE='/home/u63754588/sasuser.v94/ShipAccidents.csv'

OUT=ShipAccidents

DBMS=csv;

GETNAMES=yes;

RUN;
```

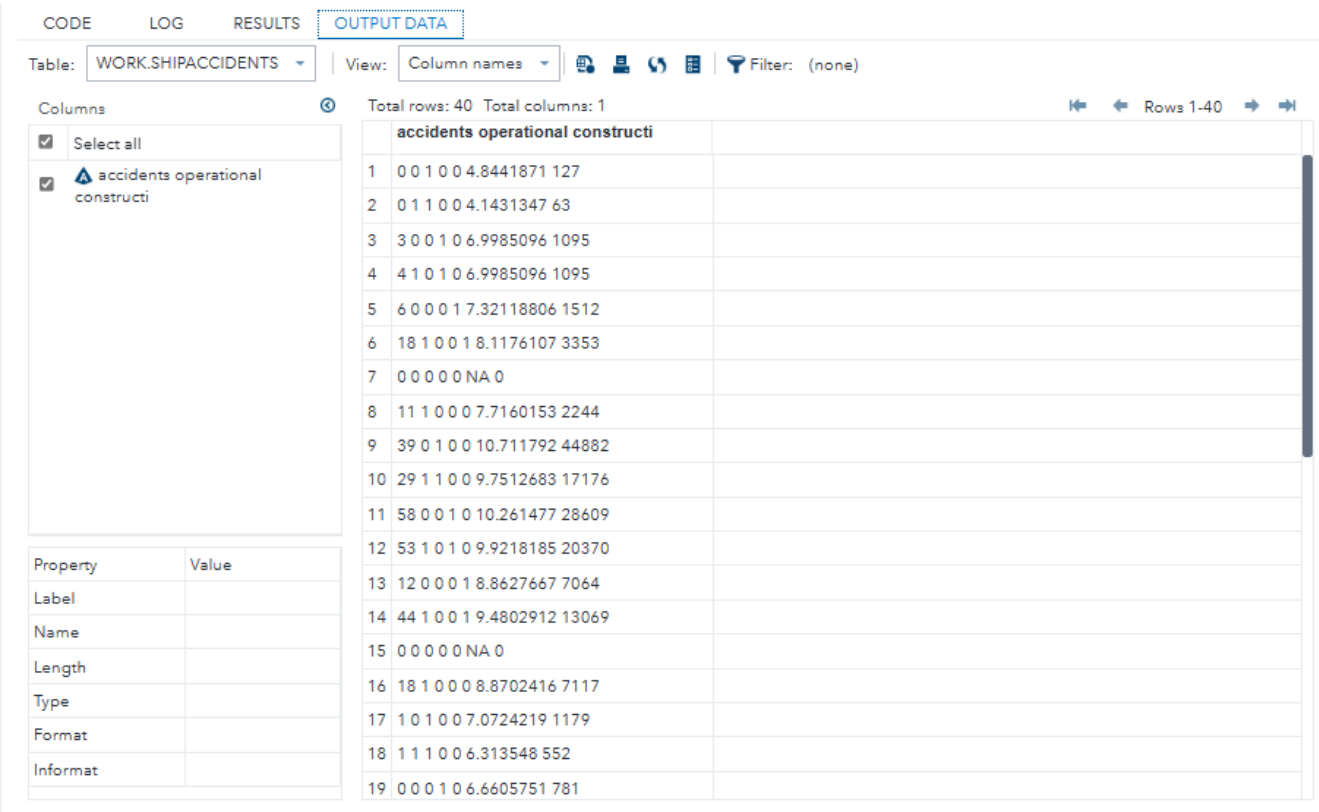
Output:



READING TEXT

```
/* Importing a TXT File using SAS */

PROC IMPORT DATAFILE='/home/u63754588/sasuser.v94/ShipAccidents.txt'

OUT=ShipAccidents

DBMS=tab replace;

GETNAMES=yes;

RUN;
```

Output:



ii. READING THE DATA INTO CSV AND TEXT USING R

1. i. Reading the Data into R consists of using two different types of formats provided. A text file and a Spreadsheet file (CSV, also known as Comma Separated Values).
   To read the file into R using the CSV file, the following code is writing and ran:

READING CSV

```
#Reading CSV file on R
file_location <- "C:/Users/Taiwo Jegede/Downloads/ShipAccidents.csv"
shipping <- read.csv(file_location)
View(shipping)
```

Output

Reading CSV.R* | shipping

Filter

	accidents	operational	construction1	construction2	construction3	exposure	service_months
1	0	0	1	0	0	4.844187	127
2	0	1	1	0	0	4.143135	63
3	3	0	0	1	0	6.998510	1095
4	4	1	0	1	0	6.998510	1095
5	6	0	0	0	1	7.321188	1512
6	18	1	0	0	1	8.117611	3353
7	0	0	0	0	0	NA	0
8	11	1	0	0	0	7.716015	2244
9	39	0	1	0	0	10.711792	44882
10	29	1	1	0	0	9.751268	17176
11	58	0	0	1	0	10.261477	28609
12	53	1	0	1	0	9.921819	20370
13	12	0	0	0	1	8.862767	7064
14	44	1	0	0	1	9.480291	13069
15	0	0	0	0	0	NA	0
16	18	1	0	0	0	8.870242	7117
17	1	0	1	0	0	7.072422	1179
18	1	1	1	0	0	6.313548	552
19	0	0	0	1	0	6.660575	781
20	1	1	0	1	0	6.516193	676
21	6	0	0	0	1	6.663133	783
22	2	1	0	0	1	7.574559	1948

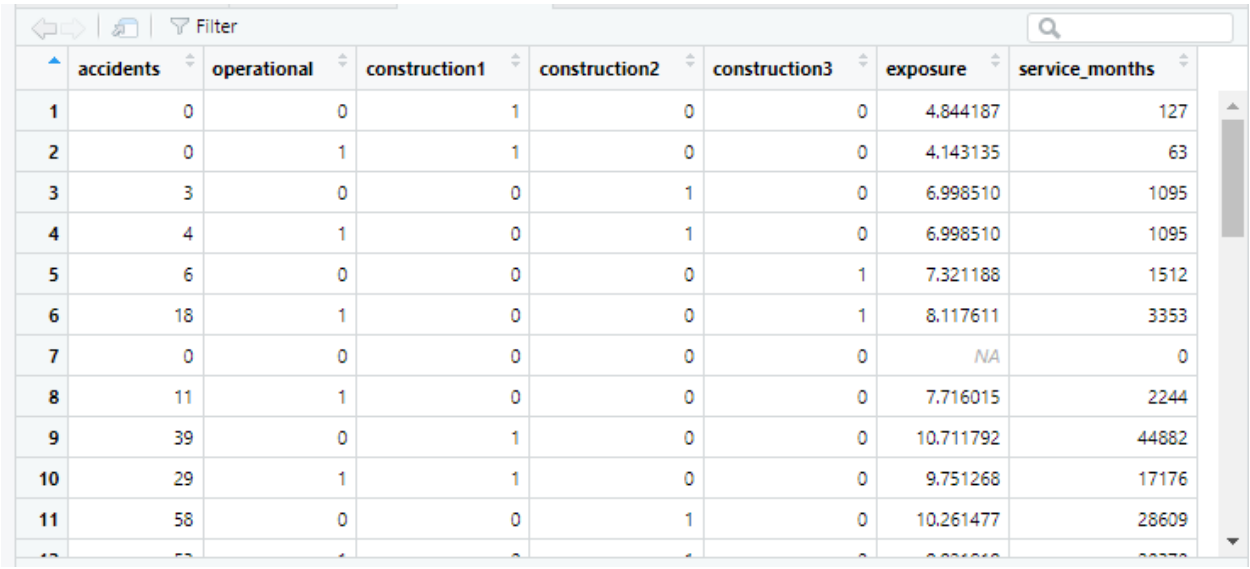
Showing 1 to 22 of 40 entries, 7 total columns

READING TEXT

#Reading Text file on R
file_location <- "C:/Users/Taiwo Jegede/Downloads/ShipAccidents.txt"
shipping <- read.table(file_location, header = TRUE)
View(shipping)

```
1  #Reading Text file on R
2  file_location <- "C:/Users/Taiwo Jegede/Downloads/ShipAccidents.txt"
3  shipping <- read.table(file_location, header = TRUE)
4  View(shipping)
```

Output



Filter

	accidents	operational	construction1	construction2	construction3	exposure	service_months
1	0	0	1	0	0	4.844187	127
2	0	1	1	0	0	4.143135	63
3	3	0	0	1	0	6.998510	1095
4	4	1	0	1	0	6.998510	1095
5	6	0	0	0	1	7.321188	1512
6	18	1	0	0	1	8.117611	3353
7	0	0	0	0	0	NA	0
8	11	1	0	0	0	7.716015	2244
9	39	0	1	0	0	10.711792	44882
10	29	1	1	0	0	9.751268	17176
11	58	0	0	1	0	10.261477	28609

iii. READING THE DATA INTO CSV AND TEXT USING PYTHON

READING CSV

```
import pandas as pd

#Reading a CSV File

shipping = pd.read_csv("C:/Users/Taiwo Jegede/Downloads/ShipAccidents.csv")
```

READING TEXT

```
import pandas as pd

#Reading a TXT File

shipping = pd.read_csv("C:/Users/Taiwo Jegede/Downloads/ShipAccidents.txt", delimiter= " ")

print(shipping)
```

Output

In [3]:
```
import pandas as pd
#Reading a TXT File
shipping = pd.read_csv("C:/Users/Taiwo Jegede/Downloads/ShipAccidents.txt", delimiter= " ")
print(shipping)
```

```
    accidents  operational  construction1  construction2  construction3  \
0           0            0              1              0              0
1           0            1              1              0              0
2           3            0              0              1              0
3           4            1              0              1              0
4           6            0              0              0              1
5          18            1              0              0              1
6           0            0              0              0              0
7          11            1              0              0              0
8          39            0              1              0              0
9          29            1              1              0              0
10         58            0              0              1              0
11         53            1              0              1              0
12         12            0              0              0              1
13         44            1              0              0              1
14          0            0              0              0              0
15         18            1              0              0              0
16          1            0              1              0              0
17          1            1              1              0              0
```

iv. MISSING VALUES

DEALING WITH MISSING VALUES ON SAS

Checking for missing values in SAS

We use the code below to get the total frequency of each variables in the table.

proc freq data=ShipAccidents;

tables _all_;

run;

accidents	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	14	35.00	14	35.00
1	5	12.50	19	47.50
2	2	5.00	21	52.50
3	1	2.50	22	55.00
4	2	5.00	24	60.00
5	1	2.50	25	62.50
6	2	5.00	27	67.50
7	2	5.00	29	72.50
11	2	5.00	31	77.50
12	2	5.00	33	82.50
18	2	5.00	35	87.50
29	1	2.50	36	90.00
39	1	2.50	37	92.50
44	1	2.50	38	95.00
53	1	2.50	39	97.50
58	1	2.50	40	100.00

operational	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	20	50.00	20	50.00

Output:

After further observation, we notice the exposure column has missing values as shown below

exposure	Frequency	Percent	Cumulative Frequency	Cumulative Percent
5.5254529	1	2.50	8	20.00
5.6131281	1	2.50	9	22.50
5.6629605	1	2.50	10	25.00
5.8550719	1	2.50	11	27.50
6.0799332	1	2.50	12	30.00
6.295266	1	2.50	13	32.50
6.313548	1	2.50	14	35.00
6.5161931	1	2.50	15	37.50
6.6605751	1	2.50	16	40.00
6.6631327	1	2.50	17	42.50
6.6707663	1	2.50	18	45.00
6.9985096	2	5.00	20	50.00
7.0535857	1	2.50	21	52.50
7.0724219	1	2.50	22	55.00
7.0967214	1	2.50	23	57.50
7.32118806	1	2.50	24	60.00
7.5745585	1	2.50	25	62.50
7.6260828	1	2.50	26	65.00
7.6783264	1	2.50	27	67.50
7.7160153	1	2.50	28	70.00
8.1176107	1	2.50	29	72.50
8.8627667	1	2.50	30	75.00
8.8702416	1	2.50	31	77.50
9.4802912	1	2.50	32	80.00
9.7512683	1	2.50	33	82.50
9.9218185	1	2.50	34	85.00
NA	6	15.00	40	100.00

DEALING WITH MISSING VALUES ON R

#To get the total number of Missing Values (NA)

print(sum(is.na(shipping)))

shipping <- na.omit(shipping)

```
> #To get the total number of Missing Values (NA)
> print(sum(is.na(shipping)))
[1] 6
> shipping <- na.omit(shipping)
> print(sum(is.na(shipping)))
[1] 0
>
```

DEALING WITH MISSING VALUES ON PYTHON

missing_values = shipping.isnull().sum().sum()

print("Total number of missing values:", missing_values)

Output:

```
In [12]: missing_values = shipping.isnull().sum().sum()
         print("Total number of missing values:", missing_values)

         Total number of missing values: 6
```

v. DESCRIPTIVE STATISTICS

USING R

To get the descriptive statistics of the Data we have just imported on R, we use "summary", as written below.

summary(shipping)

Output

```
> summary(shipping)
   accidents       operational      construction1     construction2
 Min.   : 0.00   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.: 1.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median : 4.00   Median :1.0000   Median :0.0000   Median :0.0000
 Mean   :10.47   Mean   :0.5588   Mean   :0.2647   Mean   :0.2941
 3rd Qu.:11.75   3rd Qu.:1.0000   3rd Qu.:0.7500   3rd Qu.:1.0000
 Max.   :58.00   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
 construction3       exposure       service_months
 Min.   :0.0000   Min.   : 3.807   Min.   :   45
 1st Qu.:0.0000   1st Qu.: 5.911   1st Qu.:  371
 Median :0.0000   Median : 6.999   Median : 1095
 Mean   :0.2941   Mean   : 7.049   Mean   : 4810
 3rd Qu.:1.0000   3rd Qu.: 7.707   3rd Qu.: 2223
 Max.   :1.0000   Max.   :10.712   Max.   :44882
```

USING PYTHON

Using Python, we simply use "describe", as shown below

```
In [17]: shipping.describe()
```

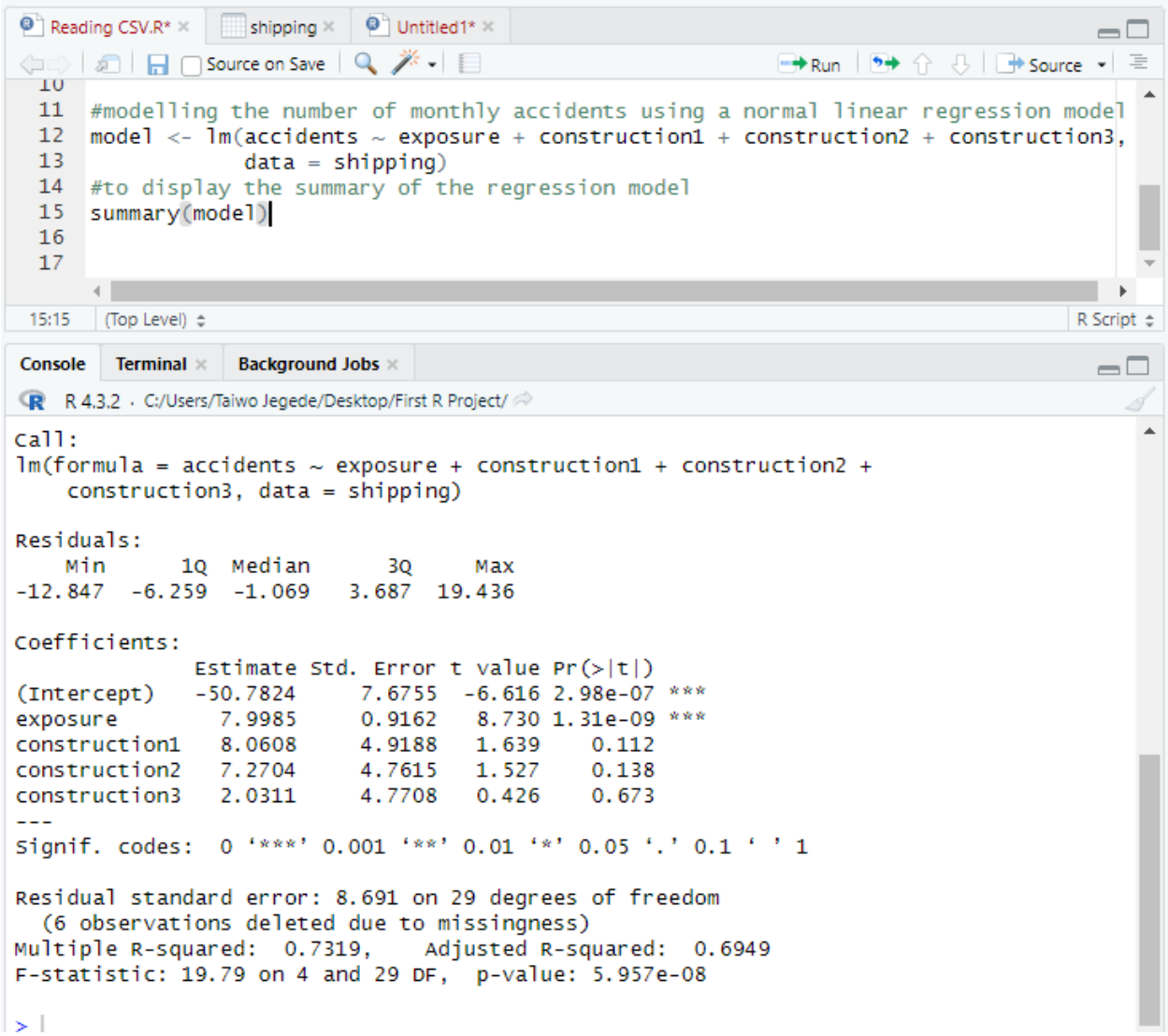
Out[17]:

	accidents	operational	construction1	construction2	construction3	exposure	service_months
count	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000
mean	10.470588	0.558824	0.264706	0.294118	0.294118	7.049255	4810.117647
std	15.734993	0.503995	0.447811	0.462497	0.462497	1.721094	9643.386309
min	0.000000	0.000000	0.000000	0.000000	0.000000	3.806662	45.000000
25%	1.000000	0.000000	0.000000	0.000000	0.000000	5.911287	371.000000
50%	4.000000	1.000000	0.000000	0.000000	0.000000	6.998510	1095.000000
75%	11.750000	1.000000	0.750000	1.000000	1.000000	7.706593	2223.250000
max	58.000000	1.000000	1.000000	1.000000	1.000000	10.711792	44882.000000

QUESTION 2

i. An appropriate normal regression model is given by:

#modelling the number of monthly accidents using a normal linear regression model
model <- lm(accidents ~ exposure + construction1 + construction2 + construction3, data = shipping)
#to display the summary of the regression model
summary(model)



II. ASUMPTIONS MADE

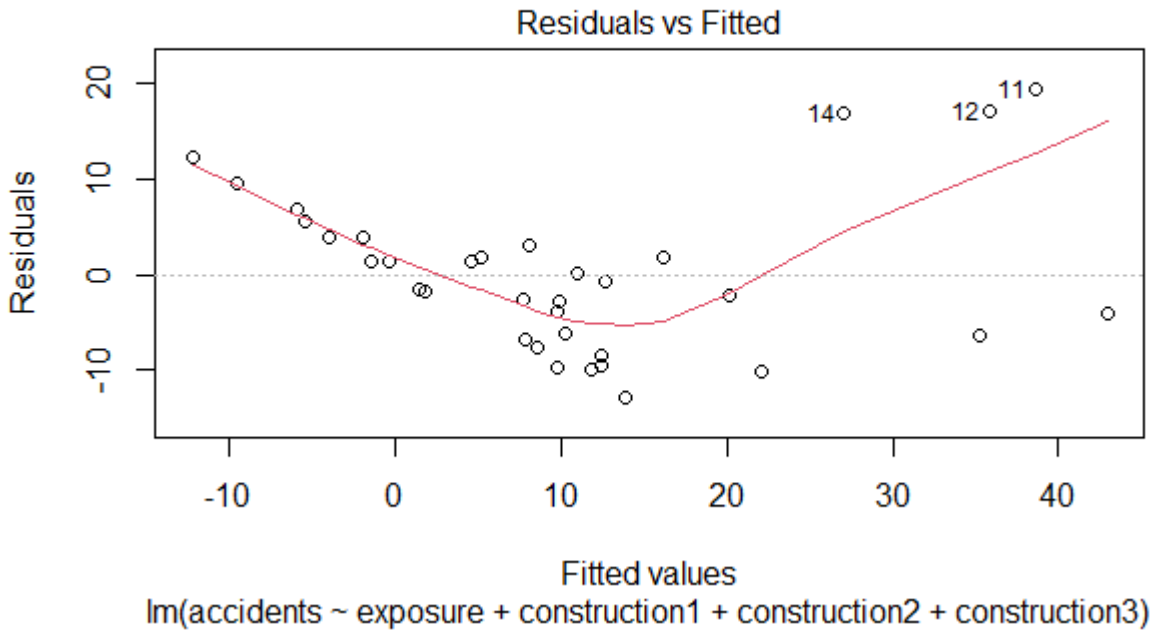
List of Assumptions made

i. Linearity
ii. Independence of Errors
iii. Normality of Residuals
iv. Homogeneity of Variance

III. ASSUMPTIONS (DEFINING THEM)

i. Linearity: The relationship between the predictor variables and the response variable is linear. The chart below shows the relationship between the residuals and the fitted values

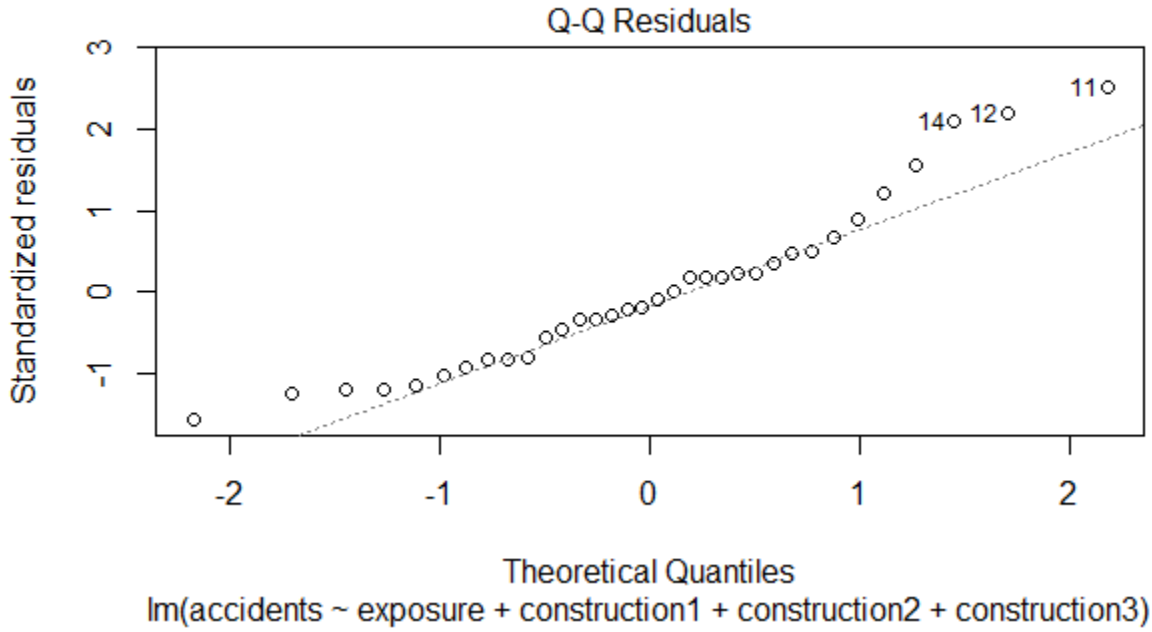
```
#modelling the number of monthly accidents using a normal linear regression model
model <- lm(accidents ~ exposure + construction1 + construction2 + construction3,
        data = shipping)
#to display the summary of the regression model
summary(model)

plot(model, which=1)
```



ii. Independence of Errors: The residuals (errors) are independent of each other.
Test: Durbin-Watson test for autocorrelation in residuals.
durbinWatsonTest(model)

iii. Normality of Residuals: Here, the residuals are normally distributed

Test: plot(model, which = 2)



IV. HOMOGENEITY OF VARIANCE

Homogeneity of Variance: Here, homogeneity of variance, means that the spread of the residuals (the differences between the observed and predicted values) is approximately constant across all levels of the predictors.

When we violate homogeneity of variance, it can lead to inefficient parameter estimates and affect the precision of hypothesis tests.

QUESTION 3

I. Purpose of using transformations to accommodate departures from constant variance

Transformations are used to address violations of the assumption of constant variance (homoscedasticity) in linear regression models. The purpose is to stabilize the variance of the residuals across different levels of the predictors. This is important because violating the homoscedasticity assumption can lead to biased parameter estimates and incorrect inferences.

Selection of Transformations

The most popular ways of selecting transformations includes the following:

- Visual inspection: this involves using pictorial representation of the values and diagnosing them. For example, in Question 2, we used plot to identify linearity between values
- Statistical Analysis: Here, we use “summary” to get the descriptive statistics of our data. For example, using Breusch-Pagan test to formally assess Homogeneity of Variance.

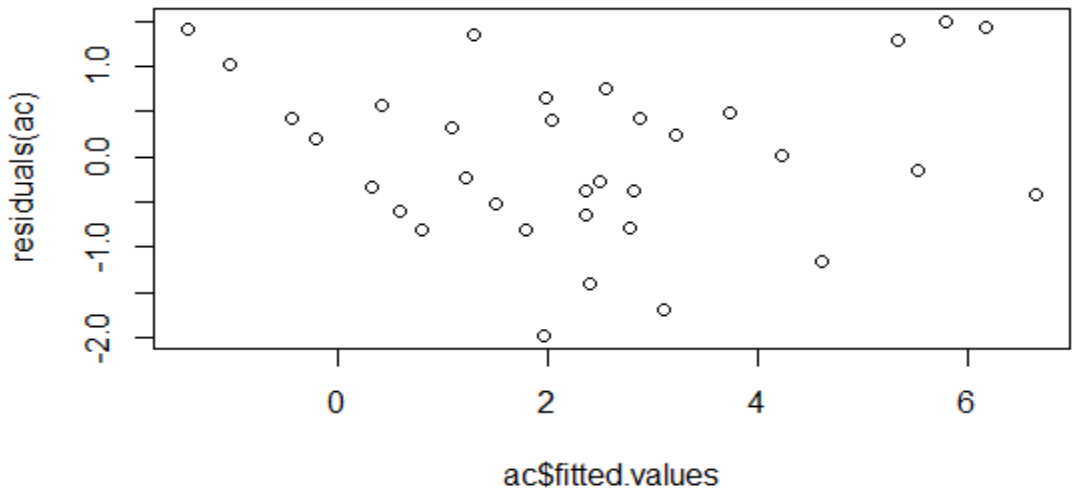
- Transformation Types: Examples include logarithmic (log), square root, and Box-Cox transformations. We want to find a transformation that stabilizes the variance and makes the residuals approximately constant

II. Investigating the Transformation

After using the below code:

```
transformed_accidents = sqrt(shipping$accidents)

summary(transformed_accidents)

ac = lm(transformed_accidents~exposure+construction1+construction2+construction3, data=shipping)

plot(ac$fitted.values, residuals(ac))
```

Output:



From our image above, we can see the randomness that exist in the data, so this is a viable transformation of the response variable.

III. SELECTING AND JUSTIFICATION

The number of monthly accidents was represented by the response variable, where i used a square root transformation (sqrt) in order to choose and justify a variance-stabilizing transformation for the Ship Accidents data. The following are the transformed variable's descriptive statistics:

Output:

```
> transformed_accidents = sqrt(shipping$accidents)
> summary(transformed_accidents)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   2.000   2.388   3.427   7.616
```

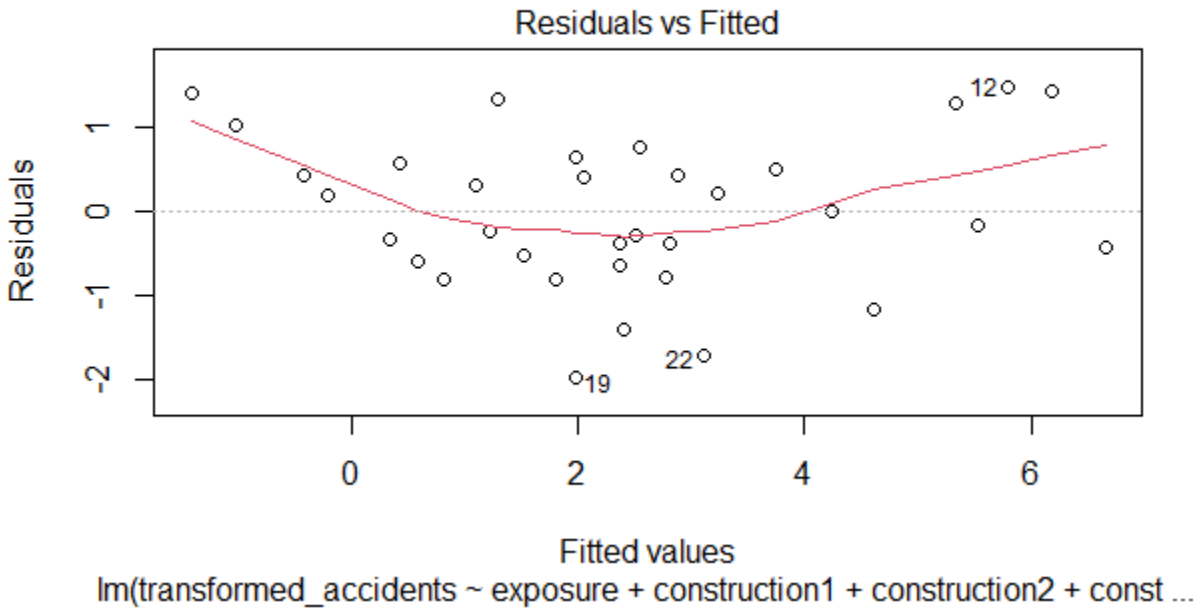
Justification:

The converted variable has a range from 0 to 7.616 and a mean of roughly 2.388. This suggests that the transformed variable has a moderate spread and is, on average, centered around 2.

IV. REFITTING THE LINEAR MODEL

Evaluating the model assumptions

Linearity: Plotting the Residuals vs Fitted Plots, we get an output as seen below



The residuals are evenly spread with no discernible pattern, and this suggest that the assumption of homoscedasticity is met and the linear assumption is appropriate.

QUESTION 4

i. Purpose of using Weighted Least Squares

Weighted Least Squares is a regression technique used to address Homogenety of Variance in the data. We use WLS in order to give different weights to observations based on the variance of their residuals, which allows the model to give less influence to observations with higher variance, making the estimation more efficient.

Decisions by the Researcher:

The method by which the researcher wishes to weight the data must be determined; at times, the square root of the variance, the inverse of the variance, or the inverse of the variance square are used.

ii. Using descriptive statistics to investigate possible weights:

Code: # Fit a linear regression model to obtain residuals

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

residuals <- residuals(model)

Investigate possible weights based on the variance of residuals

weights <- 1 / sd(residuals)^2

print(weights)

Output:

```
> print(weights)
[1] 0.01506551
```

iii. Selecting and justifying the appropriate weights

Code: # Fit a linear regression model and specify weights

weights <- 1 / residuals^2

weighted_model <- lm(accidents ~ exposure + construction1 + construction2 + construction3, data = shipping, weights = weights)

Display the summary of the weighted regression model

summary(weighted_model)

Output:

```
> # Display the summary of the weighted regression model
> summary(weighted_model)

Call:
lm(formula = accidents ~ exposure + construction1 + construction2 +
    construction3, data = shipping, weights = weights)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-1.1482 -0.8270 -0.1833  0.8663  1.4791

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -41.5197     3.3571 -12.368 4.34e-13 ***
exposure        6.8064     0.4351  15.642 1.13e-15 ***
construction1   5.0099     1.3418   3.734 0.000820 ***
construction2   4.8551     1.1931   4.069 0.000331 ***
construction3   1.5727     0.5180   3.036 0.005023 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9393 on 29 degrees of freedom
Multiple R-squared:  0.9279,    Adjusted R-squared:  0.918
F-statistic: 93.34 on 4 and 29 DF,  p-value: 3.962e-16
```

Justification:

After I successfully fitted the weighted linear regression model using the inverse of the squared residuals as weights. We see from the output that the coefficient estimates provide insights into the relationships between the predictor variables (exposure, construction1, construction2, construction3) and the response variable (number of monthly accidents).

The significant p-values for the coefficients suggest that each predictor variable is likely contributing significantly to explaining the variability in the number of monthly accidents.

iv. Refitting the model

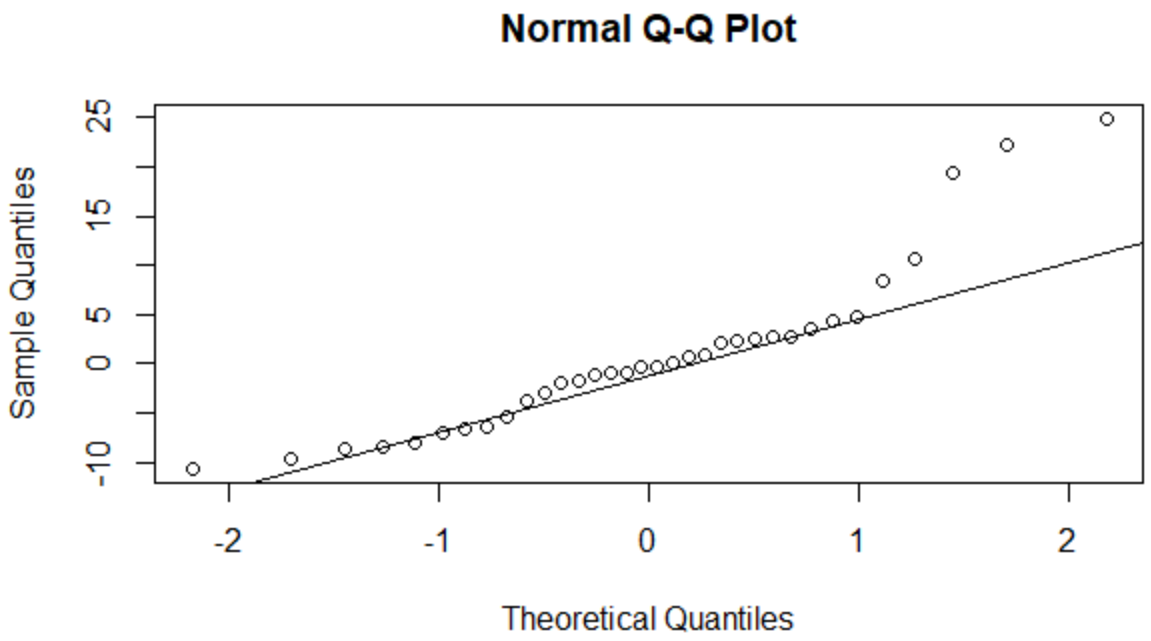
After refitting the model, using the code below:

weighted_model <- lm(accidents ~ exposure + construction1 + construction2 + construction3, data = shipping, weights = weights)

Display the summary of the weighted regression model

summary(weighted_model)

Output:



From the above Q-Q Plot, we can see that the normal assumption was not met because the plot is positively skewed. Other attempts were also used, but they were very scattered and did not meet the assumptions also.

QUESTION 5. FINDING THE DATASET

I selected this dataset from kaggle.com. And it includes 45,315 international football match results from the first-ever match played in 1872 until 2023. The matches include FIFA World Cup, FIFA Wild Cup, and standard friendlies. These are men's full international matches only.

The dataset contains three different tables, namely results, shootouts and goalscorers. This dataset contains categorical and continuous variables.

Dataset Source : [International football results from 1872 to 2023 (kaggle.com)]