

**ANL252 (Online)**

**PYTHON FOR DATA ANALYTICS**

# **Tutor-Marked Assignment**

**July 2021 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **ISABEL LOH LI JUN** | **Z2070330** |
| Click to enter NAME. | Click to enter PI NO. |
| Click to enter NAME. | Click to enter PI NO. |
| Click to enter NAME. | Click to enter PI NO. |

**Tutorial Group: ­­­­­­­­­­­­ T 09**

**Instructor’s Name: Prof. Kumar**

**Submission Date: 13/09/2021**

**Question 1**

1. **(i)**

import pandas as pd

import numpy as np

import sqlalchemy

ship = pd.read\_csv *#load data set*

missing\_values = ["."] *#identify missing values*

ship = pd.read\_csv ("downloads/ship.csv", na\_values = missing\_values) *#Declare "." are missing values*

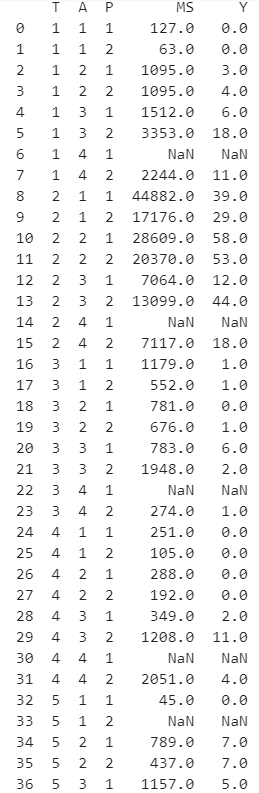
print (ship)

**Question 1**

1. **(i)**

*#Results for part (a) (i)*

*#Declare "." are missing values on dataframe Ship*





**Question 1**

1. **(ii)**

ship = ship.rename(columns={"T" : "types", "A" : "c\_years", "P" : "o\_periods", "MS" : "s\_months", "Y" : "incidents"})

*#rename ship types (T) to types*

*#rename construction years (A) to c\_years*

*#rename operation periods (P) to o\_periods*

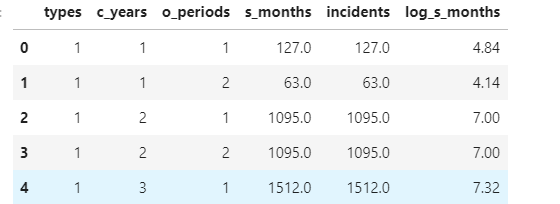
*#reaname aggregate months of service (MS) to s\_months*

*#rename number of incidents (Y) to incidents*

ship.head() *#first 5 rows of Dataframe Ship with updated column names*

*#Results for (a) (ii)*

*#first 5 rows of Dataframe Ship with updated column names*



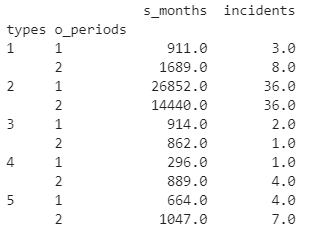
**Question 1**

1. **(iii)**

shipgroup = ship.groupby(['types', 'o\_periods'])[['s\_months', 'incidents']].mean().round()*#average service months and average number of incidents in every category based on the types and operation periods*

*#Results for (a) (iii)*

*#average service months and average number of incidents in every category based on the types and operation periods*



**Question 1**

1. **(iv)**

shipgroup.loc[1,1][0] *#type 1, period 1 [Check and test to see if all values in Avg service months, under type 1, period 1, will give me the Avg service months]*



shipgroup.loc[1,1][1] *#type 1, period 1 [Check and test to see if all values in Avg no. of incidents, under type 1, period 1, will give me the Average no. of incidents]*



ship\_smonths = lambda sg: shipgroup.loc[sg['types'], sg['o\_periods']][0] if pd.isnull(sg['s\_months']) else sg['s\_months'] *#This basically means that if a value in s\_months is null, it will replace the null value with the mean number calculated based on which type and period it belongs to.*

ship['s\_months'] = ship.apply(ship\_smonths, axis=1)

ship\_incidents = lambda sg: shipgroup.loc[sg['types'], sg['o\_periods']][1] if pd.isnull(sg['incidents']) else sg['incidents'] *#This basically means that if a value in incidents is null, it will replace the null value with the mean number calculated based on which type and period it belongs to.*

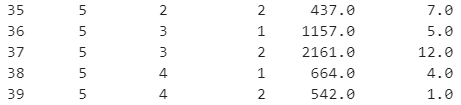
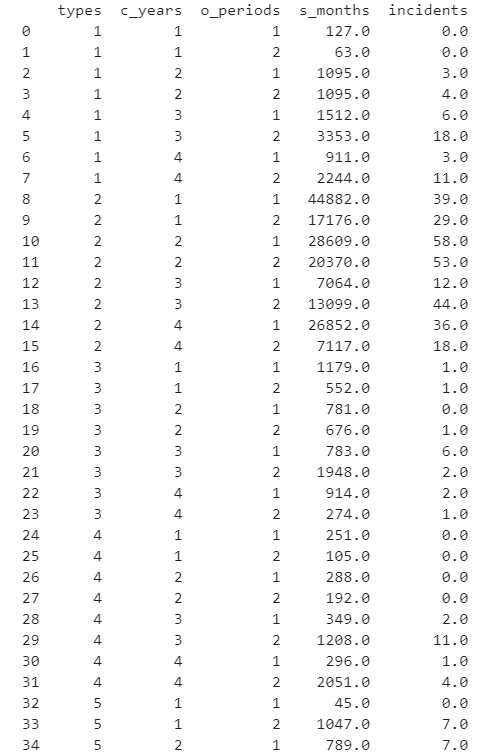
ship['incidents'] = ship.apply(ship\_incidents, axis=1)

print (ship) *#dataframe ship with missing values being replaced.*

**Question 1**

1. **(iv)**

*#Results for (a) (iv)*

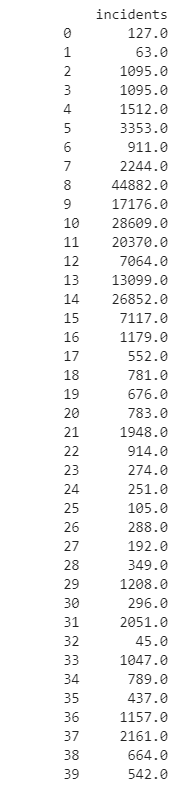
*#dataframe ship with missing values being replaced.* ****

**Question 1**

1. **(v)**

Y = pd.DataFrame(ship['incidents'])

*#Results for dataframe Y*



**Question 1**

1. **(i)**

for col in ['types', 'c\_years', 'o\_periods']:

ship[col] = ship[col].astype('category')

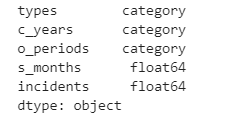
*#convert data types in columns types, c\_years & o\_periods to categorical*

datatypes = ship.dtypes

print(datatypes)

*#check if columns type, c\_years, o\_periods are categorical*

*#Results for (b) (i)*



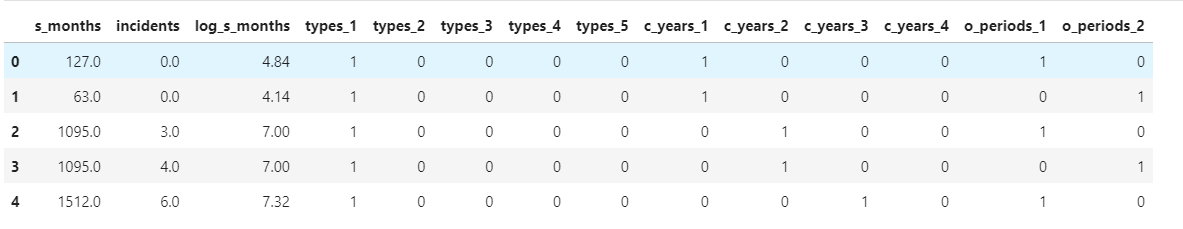
**Question 1**

**(b) (ii)**

X = ship.copy() *# Make a deep copy*

X = pd.get\_dummies(X,columns = ['types', 'c\_years','o\_periods'])

X.head()

*#Results for (b) (ii)*

**Question 1**

**(b) (iii)**

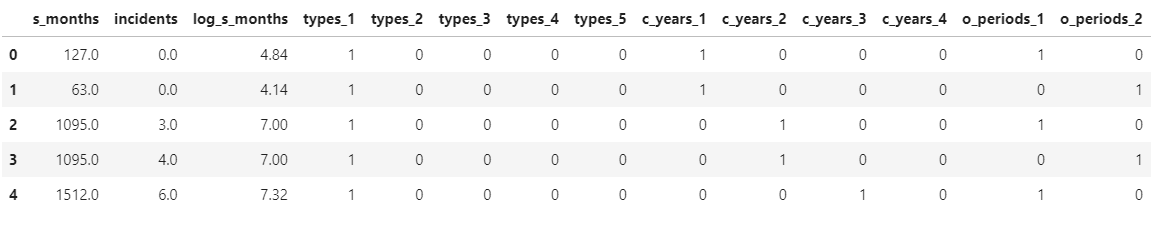
log\_s\_months = np.log(ship['s\_months'])*#apply log transformation*

log\_s\_months = round(log\_s\_months, 2) *#round off values to nearest 2 decimal places*

ship['log\_s\_months'] = log\_s\_months

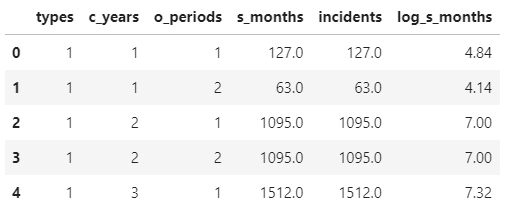
X['log\_s\_months'] = log\_s\_months *#log s\_months to dataframes ship & X*

X.head()*#First 5 rows of Dataframe X with newly added column log\_s\_months*

*#Results for (b) (iii)*

ship.head()*#First 5 rows of Dataframe ship with newly added log\_s\_months column*

*#Results for (b) (iii)*

**

**Question 1**

As the dataset provided is relatively small, separating the data for Training and Testing purposes, would result in a smaller dataset. According to SearchEnterpriseAI, The Predictive Model is used to predict future events through analysing past data (Carew, 2020). Hence if we were to use a small dataset, this leads to several issues, due to usage of the small data set, such as overfitting of the data. Therefore, we shall use the entire dataset for training purpose instead.

**Question 1**

ship.to\_csv('ship\_prepared.csv')*#Dataframe ship to .csv file*

engine = sqlalchemy.create\_engine('sqlite:///ship.db', echo = True)

sqlite\_connection = engine.connect()

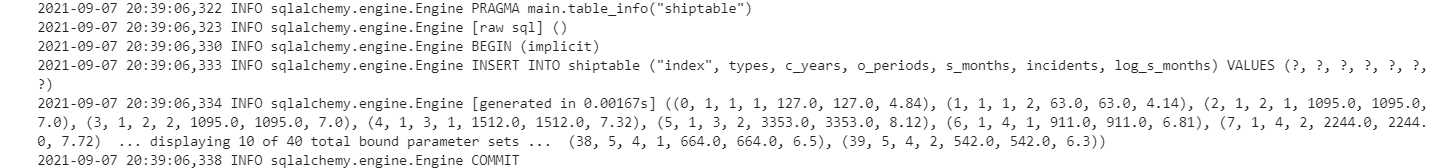
ship.to\_sql('shiptable' , engine, if\_exists = 'append')

*#Database name: ship.db*

*#engine is the connection object*

*#export df ship to db ship, echo true is to see all the output that comes from our database connection*

*#records of Dataframe ship are inserted to the database, if table exists, replace the data*

*#Results for (d)*

**Question 2**

1. Scikit-learn module:

* Scikit-learn is built upon SciPy, Numpy, Pandas and matplotlib libraries
* It is easy to use for building any kind of machine models
* The installation of this library is through using ‘pip install scikit learn’ command in command prompt
* Python has to be installed in the system before scikit learn can be installed in library

Poisson Regression Estimator:

* Poisson Regression Estimator is based on a generalized linear model with a Poisson regression.
* It consists of:
* alpha parameter: float, determines the regularization strength
* fit\_intercept parameter: Boolean, constant should be
* max\_iter parameter: integer, highest number of iterations
* tol parameter: float, benchmark given to stop the solver.
* warm\_start parameter Boolean uses True/False. If it is set to ‘True’, it reuses the solution provided previously.
* verbose parameter: integer, number of verbosity

Fit () function:

* Fit takes 2 parameters, E.g. X and y parameter
* It is used to train data so that better accuracy can be achieved
* X parameter: Array-like of shape using a number of samples and features
* y parameter: Array-like of shape using a number of samples given only
* The syntax can be written as “fix (X, y)”

Predict () function:

* It predicts labels of the data values
* It takes only 1 parameter, E.g. y
* The syntax can be written as “predict (X)”

1. *#b - 2) Fit Poisson Regression (using website)*

from sklearn import linear\_model

PRM = linear\_model.PoissonRegressor()

X = X.to\_numpy()

Y = Y.to\_numpy()

PRM.fit(X, Y)

from collections import defaultdict *#Store data*

datatable = defaultdict(float)

datatable['β0'] = PRM.intercept\_

i=1

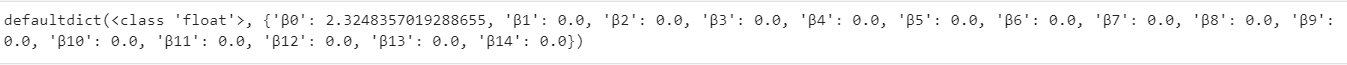
for value in PRM.coef\_:

datatable['β'+str(i)]= value

i+=1

print(datatable)

*#Result for 2(b)*



1. *#deviance formula = sgn (Y-EY) X sqrt (2 x Y x log(Y/EY) - (Y-EY)*

EY = PRM.predict(X)

D = np.sign(Y-(np.mean(EY))\*np.sqrt(2\*Y\*np.log(Y/(np.mean(EY)))-(Y-(np.mean(EY)))))

EY = PRM.predict(X)

D = np.sign(Y-(np.mean(EY))\*np.sqrt(2\*Y\*np.log(Y/(np.mean(EY)))-(Y-(np.mean(EY)))))

for Y in Y:

if(Y==0):

D.append((np.sign(Y-(np.mean(EY))\*np.sqrt(-(Y-(np.mean(EY))))))) #if Y = 0, then log[y/exp(E(Y))] = 0

else:

D.append((np.sign(Y-(np.mean(EY))\*np.sqrt(2\*Y\*np.log(Y/(np.mean(EY)))-(Y-(np.mean(EY)))))))

print("Deviance is ", np.mean(D))

**References**

Carew, J. M. (2020, December 2). *What is predictive modeling?* SearchEnterpriseAI. Retrieved September 9, 2021, from <https://searchenterpriseai.techtarget.com/definition/predictive-modeling>.

scikit-learn developers. (2017). *scikit learn*. scikit. <https://scikit-learn.org/stable/>.

scikit-learn developers. (2017). *Sklearn.linear\_model.poissonregressor¶*. scikit learn. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PoissonRegressor.html#sklearn.linear_model.PoissonRegressor>.