SARDAR PATEL UNIVERSITY

DEPARTMENT OF STATISTICS

Project Report On

"Analysis On GLM Result"

PRESENTED BY

Kunal Chatur Salunkhe

[M.Sc. Applied Statistics] (Roll no. 10)

For Data File & Code File Click Here

Contest

- Introduction & Motivation
- Objective

- Import Result of (Applied Statistics)
  - Find the summary statistics
  - Check the null values
  - Line Plot for students Performance
  - Multiple Bar Plot for Marks and Anxiety Levels of Students

- Import the Data (M.Sc. Statistics)
  - Find the summary statistics
  - Check the null values
  - Line Plot for students Performance
  - Multiple Bar Plot for Marks and Anxiety Levels of Students

- Import the GLM Result Applied & Pure Statistics
  - Find the summary statistics
  - Line Plot for students Performance
  - Histogram
  - Shapiro-Wilk test for check the normality
  - K-S test for check the normality
  - Find the correlation between Marks & Time
  - Apply the Linear Regression
    - Calculate the Predicted Y
    - Calculate the Residual
    - Calculate the RMSE & MSE
  - Fit a multiple regression model (OLS)
  - Perform the t-test for "Marks_pure_stat" & "Marks_applied_stat"
  - Perform the t-test for Gender (x3)
  - Apply Multinomial Logit Model
  - Calculate the Odds ratio of Gender (x3)

- Conclusion

Introduction

In the realm of statistical analysis, the Generalized Linear Model (GLM) stands as a cornerstone methodology, serving as a powerful tool for modeling a wide array of data types and patterns. It is a versatile statistical framework that finds applications in a multitude of disciplines, from epidemiology to finance, and from ecology to social sciences. In this report, we delve into a comprehensive study of GLM results, focusing on their application and relevance within the context of both Applied Statistics and Pure Statistics programs.

Our investigation revolves around a set of key variables: "Marks," "Course," "Time," "Anxiety Level," and "Gender." These variables are selected with the intention of uncovering the intricate relationships that exist between them and their collective impact on academic achievements. The "Marks" variable represents student performance, "Course" provides insights into the program type, "Time" delves into the time allocation of students for their studies, "Anxiety Level" sheds light on the psychological aspects, and "Gender" captures the demographic diversity within the study cohort.

Our project leverages the power of **Python** to drive innovation and deliver impactful results.

Motivation

The motivation behind this study is to bridge the gap between theory and practical application in the field of statistics, particularly within the academic setting of Applied and Pure Statistics programs. We recognize that student's performance, represented by the variable "Marks," is influenced by various factors, such as the "Course" they are enrolled in, the "Time" allocated for studying, their "Anxiety Level," and even "Gender" differences. Understanding the relationship between these variables is crucial in designing effective educational strategies, identifying potential challenges, and ultimately enhancing the learning experience for students.

By using Generalized Linear Models, we aim to provide a comprehensive analysis of the interplay between these variables, enabling us to draw meaningful conclusions and recommendations for program improvement. This study is not only valuable to educators but also to students, administrators, and policymakers who seek to optimize the educational environment, ensure fairness, and promote better learning outcomes.

OBJECTIVES

- Evaluate and compare the results and outcomes of the Applied Statistics and Pure Statistics programs, providing insights into their effectiveness, strengths, and areas for improvement.

- **Variables Affecting Course Success :** Aims to investigate and analyze the factors, such as "Time allocated for study", "Gender" and "Exam-related Anxiety Levels," to determine their potential influence on student's outcomes in the GLM result.

- **Gender-Based Academic Performance Analysis :** We'll investigate and analyze potential gender-based differences in academic performance.

- The main objective of this analysis is to investigate the factors influencing student performance in the GLM course. This study aims to provide valuable insights into the disparities in student grades and offer actionable recommendations for enhancing the learning experience. The ultimate goal is to ensure equal opportunities for success among students in both the Applied Statistics and Pure Statistics programs.

1. Applied Statistics

Import libraries

```
import pandas as pd
import numpy as np
import statsmodels.api as sm

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
 from sklearn.linear_model import LinearRegression
 from scipy import stats
```

Import the Data (M.Sc. Applied Statistics)

- GLM Result of Applied Statistics

```
applied = pd.read_excel("D:/M.SC 3sem 2023/GLM/Applied GLM.xlsx")
 applied
```

	Roll Number	Name	Gender	Course	Marks in Total (GLM)	Study Time(in Min)	Anxiety Level(1-10)
0	1	Adinath Nathu Pangarkar.	Male	Applied Statistic	8.75	60	2
1	2	Aishwarya Hindurao Shejwal.	Female	Applied Statistic	6.75	360	6
2	3	Akshay Changdev Pawar.	Male	Applied Statistic	8.50	120	8
3	4	Anisa Ashapak Mulani.	Female	Applied Statistic	8.25	240	7
4	5	Apekshit Brahmadev Gaikwad.	Male	Applied Statistic	3.50	300	8
5	6	Ashutosh Pratap Lotake.	Male	Applied Statistic	5.00	60	5
6	7	Gaurav Nale.	Male	Applied Statistic	3.50	90	8
7	8	Hiralal Suresh Mali.	Male	Applied Statistic	11.25	120	5
8	9	Joel George.	Male	Applied Statistic	7.25	60	6
9	10	Kunal Salunkhe.	Male	Applied Statistic	7.00	360	7
10	11	Manali Laxman Nalawade.	Female	Applied Statistic	14.75	400	4
11	12	Nisha Rajendra Lad.	Female	Applied Statistic	10.00	315	4
12	13	Om Prasanna Patukale.	Male	Applied Statistic	7.25	60	2
13	14	Omkar Dinesh Patil.	Male	Applied Statistic	7.75	40	1
14	15	Pawan Pravin Patil.	Male	Applied Statistic	5.00	60	6
15	16	Pranita Popat Mane.	Female	Applied Statistic	6.50	60	5
16	17	Rushikesh Bhaskar Sonawane	Male	Applied Statistic	8.25	40	5
17	18	Sachin Subash Derle.	Male	Applied Statistic	8.50	60	9
18	19	Sumit Ratnakar Motagi.	Male	Applied Statistic	7.50	30	1
19	20	Vikas Bajrang Nangare.	Male	Applied Statistic	6.00	360	7

Check the null values

```
14. applied.isnull().sum()
```

```
Roll Number              0
Name                     0
Gender                   0
Course                   0
Marks in Total (GLM)     0
Study Time(in Min)       0
Anxiety Level(1-10)      0
dtype: int64
```

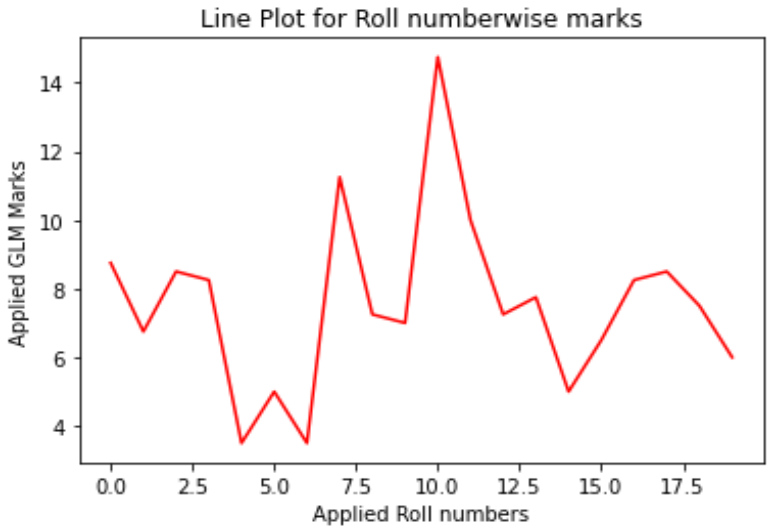
Find the summary statistics

```
15. applied.describe()
```

	Roll Number	Marks in Total (GLM)	Study Time(in Min)	Anxiety Level(1-10)
count	20.00000	20.000000	20.000000	20.000000
mean	10.50000	7.562500	159.750000	5.300000
std	5.91608	2.577579	135.999952	2.386365
min	1.00000	3.500000	30.000000	1.000000
25%	5.75000	6.375000	60.000000	4.000000
50%	10.50000	7.375000	75.000000	5.500000
75%	15.25000	8.500000	303.750000	7.000000
max	20.00000	14.750000	400.000000	9.000000

Line Plot for students Performance

```
16. plt.plot(applied['Marks in Total (GLM)'], color = "red")
17. plt.xlabel("Applied Roll numbers")
18. plt.ylabel("Applied GLM Marks")
19. plt.title("Line Plot for Roll number wise marks")
```



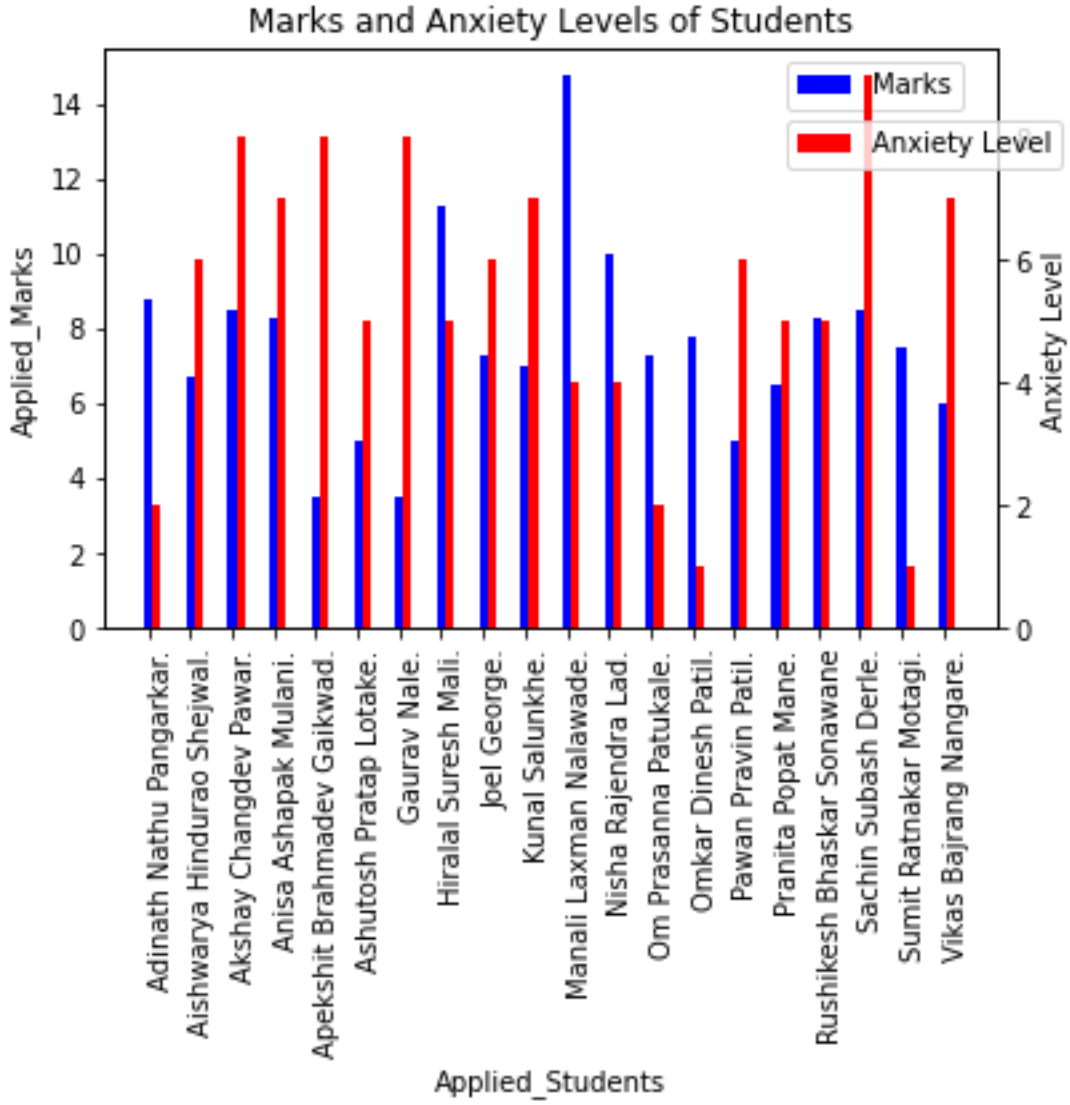
Student Performance:

- Each point on the line represents the marks obtained by an individual student. From those points, you can see how each student's marks vary. This plot can help identify outliers or students with exceptional performance.

Multiple Bar Plot for Marks and Anxiety Levels of Students

```
applied_marks = applied['Marks in Total (GLM)']
anxiety_levels = applied['Anxiety Level(1-10)']
students = applied['Name']

# Create a bar plot
width = 0.2  # Width of the bars
x = range(len(students))

fig, ax1 = plt.subplots()

#Here we create plot of applied_marks
ax1.bar(x, applied_marks, width, label='Marks', color='blue',
align='center')
ax1.set_xlabel('Applied_Students')
ax1.set_ylabel('Applied_Marks')
ax1.set_xticks(x)
ax1.set_xticklabels(students, rotation=90, ha='left')


ax2 = ax1.twinx()

# Plot anxiety levels
ax2.bar([i + width for i in x], anxiety_levels, width, label='Anxiety
Level', color='red', align='center')
ax2.set_ylabel('Anxiety Level')
```

```
ax1.legend(loc='upper left', bbox_to_anchor=(0.75, 1.0))
ax2.legend(loc='upper left', bbox_to_anchor=(0.75, 0.9))

plt.title('Marks and Anxiety Levels of Students')
plt.show()
```



From Multiple Bar Plot

- From this plot we can see that each student marks compare to their anxiety level.
- Also identify students with high marks and high or low anxiety levels, and vice versa.
    - For eg : "Manali Laxman Nalawade" has highest marks & moderate anxiety level.

Import the Data (M.Sc. Statistics)

- GLM Result of Pure Statistics

```
Data = pd.read_excel("D:/M.SC 3sem 2023/GLM/Pure GLM.xlsx")
Data
```

Here We'll extract variable from Data

```
Pure_stat = Data[['rollno','marks','course','x1','x2','X3']]
Pure_stat
```

	rollno	marks	course	x1	x2	X3
0	1	8.75	statistics	70	7	Female
1	2	9.00	statistics	150	3	Female
2	3	7.50	statistics	50	6	Female
3	4	3.75	statistics	45	7	Male
4	5	6.50	statistics	60	4	Male
5	6	9.25	statistics	100	4	Male
6	7	7.25	statistics	50	4	Male
7	8	12.25	statistics	60	6	Female
8	9	12.00	statistics	20	6	Female
9	10	6.00	statistics	120	5	Female
10	11	9.00	statistics	150	4	Female
11	12	4.50	statistics	60	4	Male
12	13	6.75	statistics	50	4	Male
13	14	3.75	statistics	60	4	Male
14	15	8.75	statistics	60	3	Male
15	16	11.50	statistics	120	6	Female
16	17	10.75	statistics	90	4	Female
17	18	5.00	statistics	60	4	Male
18	19	6.25	statistics	120	4	Female
19	20	6.50	statistics	55	5	Male
20	21	8.75	statistics	60	5	Male
21	22	5.25	statistics	20	3	Female

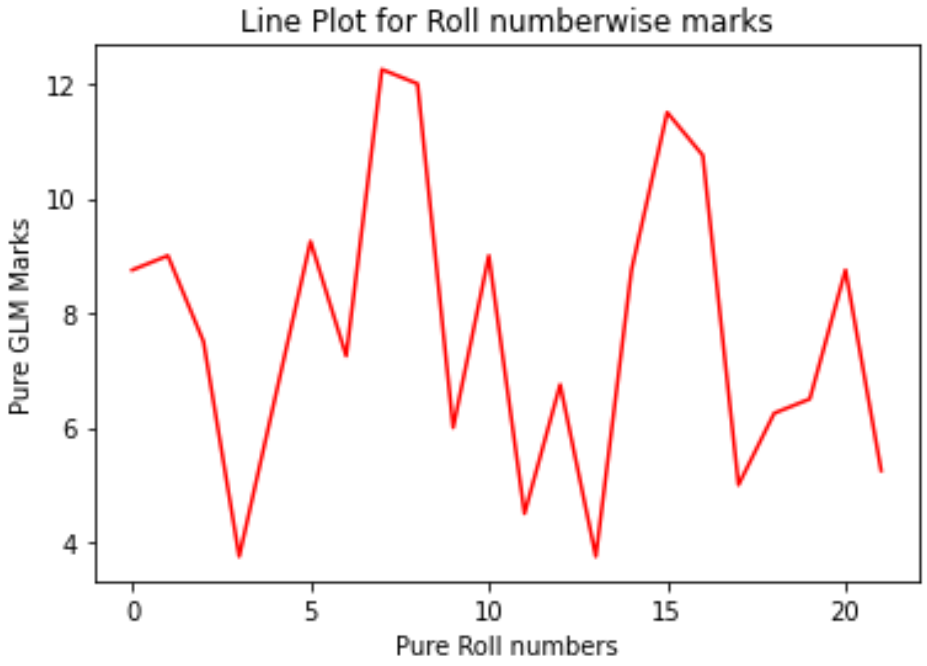
Find the summary statistics

```
Pure_stat.describe()
```

	rollno	marks	x1	x2
count	22.000000	22.000000	22.000000	22.000000
mean	11.500000	7.681818	74.090909	4.636364
std	6.493587	2.542709	37.149517	1.216766
min	1.000000	3.750000	20.000000	3.000000
25%	6.250000	6.062500	51.250000	4.000000
50%	11.500000	7.375000	60.000000	4.000000
75%	16.750000	9.000000	97.500000	5.750000
max	22.000000	12.250000	150.000000	7.000000

Line Plot for students Performance

```
plt.plot(Pure_stat['marks'], color = "red")
plt.xlabel("Pure Roll numbers")
plt.ylabel("Pure GLM Marks")
plt.title("Line Plot for Roll numberwise marks")
```



Student Performance:

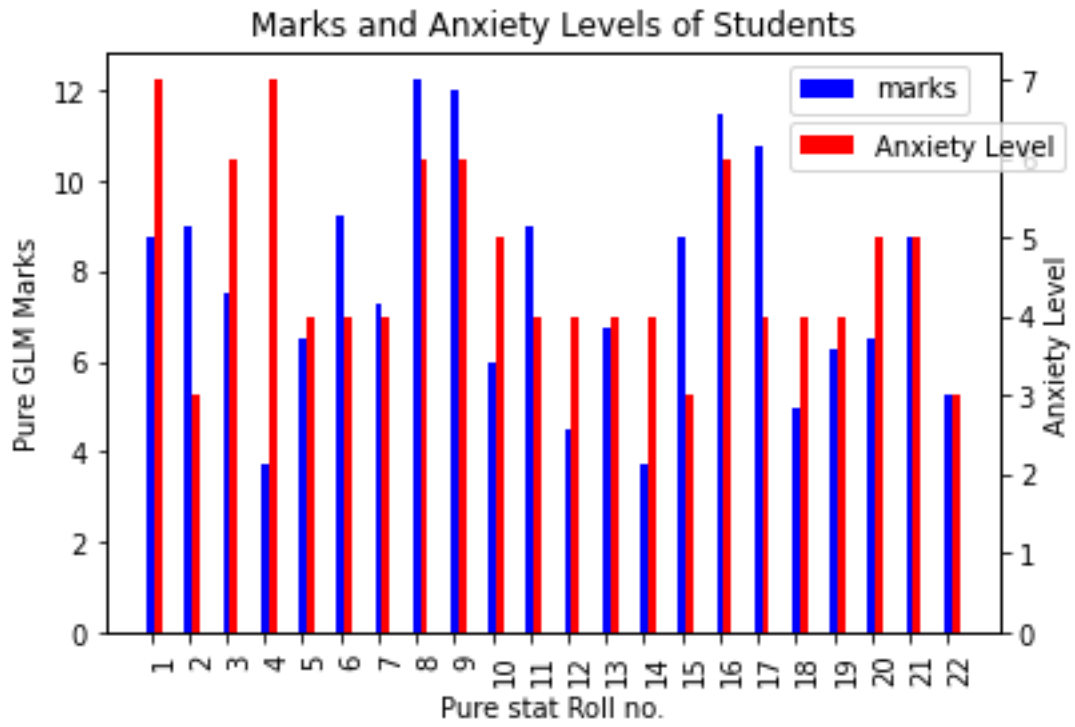
- Each point on the line represents the marks obtained by an individual student. From those points, you can see how each student's marks vary. This plot can help identify outliers or students with exceptional performance.

Multiple Bar Plot for Marks and Anxiety Levels of Students

```
Pure_marks = Pure_stat['marks']
Pure_anxiety_levels = Pure_stat['x2']
students = Pure_stat['rollno']

# Create a bar plot
width = 0.2  # Width of the bars
x = range(len(students))

fig, ax1 = plt.subplots()

#Here we create plot of applied_marks
ax1.bar(x, Pure_marks, width, label='marks', color='blue', align='center')
ax1.set_xlabel('Pure stat Roll no.')
ax1.set_ylabel('Pure GLM Marks')
ax1.set_xticks(x)
ax1.set_xticklabels(students, rotation=90, ha='left')


ax2 = ax1.twinx()

# Plot anxiety levels
ax2.bar([i + width for i in x], Pure_anxiety_levels, width, label='Anxiety
Level', color='red', align='center')
ax2.set_ylabel('Anxiety Level')

ax1.legend(loc='upper left', bbox_to_anchor=(0.75, 1.0))
ax2.legend(loc='upper left', bbox_to_anchor=(0.75, 0.9))

plt.title('Marks and Anxiety Levels of Students')
plt.show()
```

From Multiple Bar Plot

- From this plot we can see that each student marks compare to their anxiety level.
- Also identify students with high marks and high or low anxiety levels, and vice versa.
    - For eg : Roll no. 4 has low marks & high anxiety level



Import the GLM Result Applied & Pure Statistics

- Independent variable : course , x1 = "time", x2 = "Anxiety Level", x3 = "Gender"
  - course : Pure Statistics = 1 & Applied Statistics = 0
  - x3 = Gender : Male = 1 & Female = 0
- Dependent variable : Marks

```
##Male = 1 & # Female = 0
GLM_result = pd.read_excel("D:/M.SC 3sem 2023/GLM/GLM result.xlsx")
GLM_result
```

	Marks	course	x1	x2	x3
0	8.75	1	70	7	0
1	9.00	1	150	3	0
2	7.50	1	50	6	0
3	3.75	1	45	7	1
4	6.50	1	60	4	1
5	9.25	1	100	4	1
6	7.25	1	50	4	1
7	12.25	1	60	6	0
8	12.00	1	20	6	0
9	6.00	1	120	5	0
10	9.00	1	150	4	0
11	4.50	1	60	4	1
12	6.75	1	50	4	1

24	8.50	0	120	8	1
25	8.25	0	240	7	0
26	3.50	0	300	8	1
27	5.00	0	60	5	1
28	3.50	0	90	8	1
29	11.25	0	120	5	1
30	7.25	0	60	6	1
31	7.00	0	360	7	1
32	14.75	0	400	4	0
33	10.00	0	315	4	0
34	7.25	0	60	2	1
35	7.75	0	40	1	1
36	5.00	0	60	6	1
37	6.50	0	60	5	0
38	8.25	0	40	5	1
39	8.50	0	60	9	1
40	7.50	0	30	1	1
41	6.00	0	360	7	1

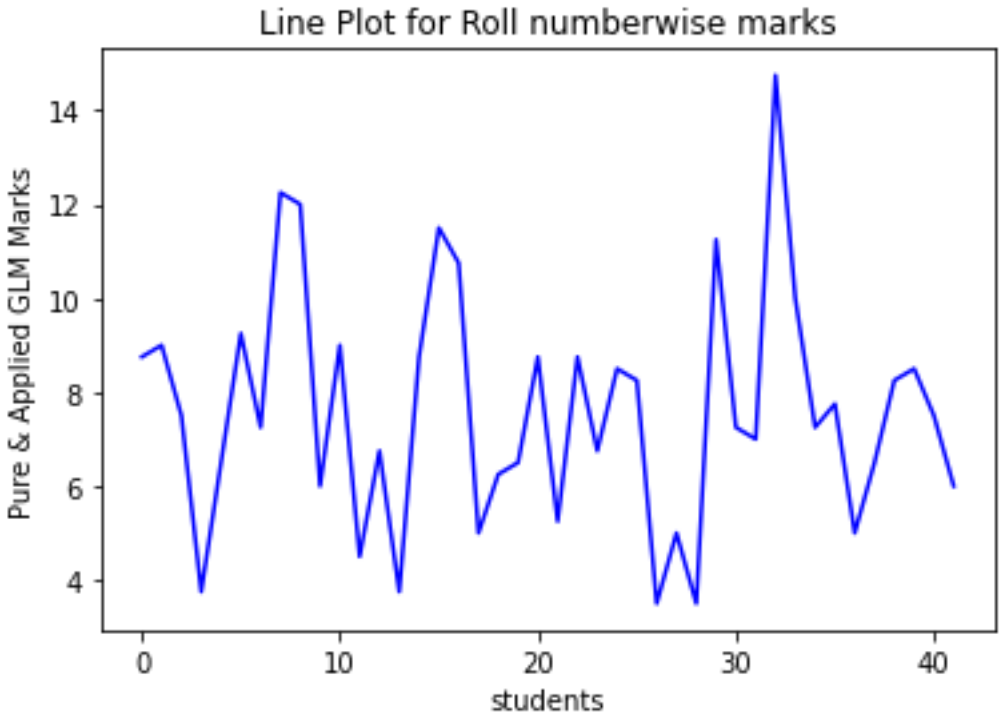
Find the summary statistics

```
GLM_result.describe()
```

	Marks	course	x1	x2	x3
count	42.000000	42.000000	42.000000	42.000000	42.000000
mean	7.625000	0.523810	114.880952	4.952381	0.619048
std	2.528647	0.505487	105.607927	1.873465	0.491507
min	3.500000	0.000000	20.000000	1.000000	0.000000
25%	6.062500	0.000000	60.000000	4.000000	0.000000
50%	7.375000	1.000000	60.000000	5.000000	1.000000
75%	8.750000	1.000000	120.000000	6.000000	1.000000
max	14.750000	1.000000	400.000000	9.000000	1.000000

Line Plot for students Performance

```
plt.plot(GLM_result['Marks'], color = "blue")
plt.xlabel("students")
plt.ylabel("Pure & Applied GLM Marks")
plt.title("Line Plot for Roll numberwise marks")
```

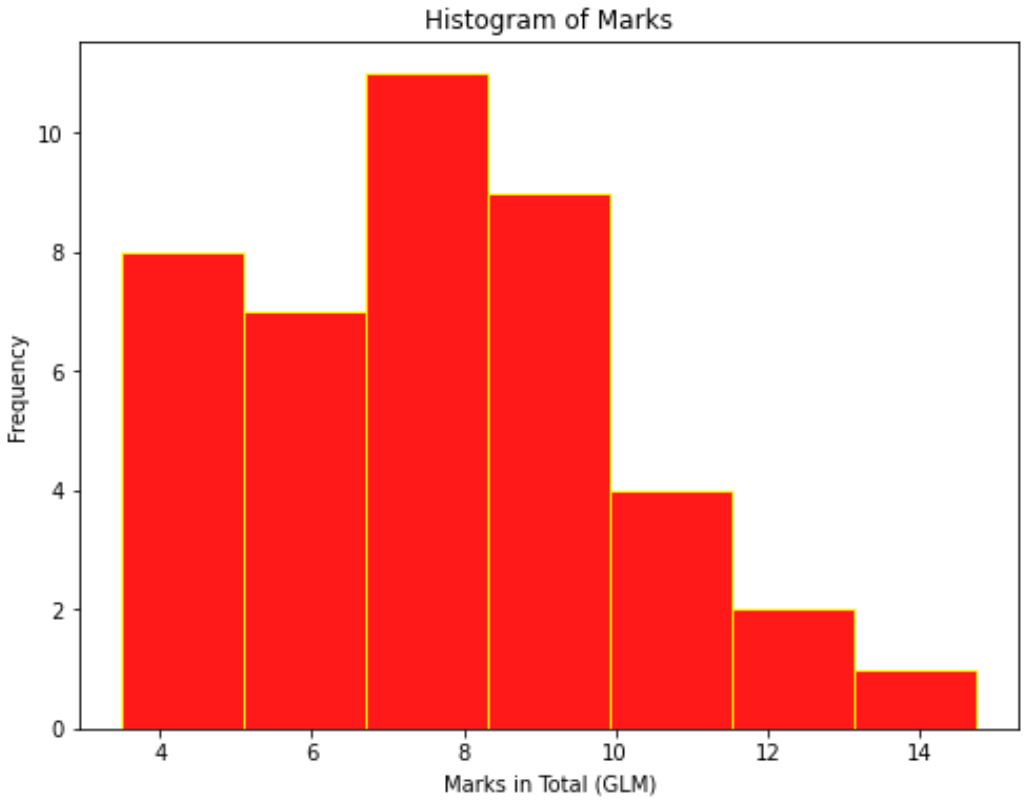


Student Performance:

- Each point on the line represents the marks obtained by an individual student. From those points, you can see how each student's marks vary. This plot can help identify outliers or students with exceptional performance.

Histogram

```
Both_marks = GLM_result['Marks']

plt.figure(figsize=(8, 6))
plt.subplot(1, 1, 1)
plt.hist(Both_marks, bins=7, edgecolor='yellow', alpha=0.9, color='red')
plt.xlabel('Marks in Total (GLM)')
plt.ylabel('Frequency')
plt.title('Histogram of Marks')

plt.show()
```



Plot the Histogram using Sturge Rule :

- n = 42
- Range (R) = 11.25
- k = 1+3.222 log(n)
  - 1+3.222 log(42) = 6.230109
- Class Interval = R/k = 1.8057
- So we take bins = 7 (approx from k)

From the Histogram :

- From the histogram, We can say that our dependent (Marks) variable show the normal (bell shaped) pattern.

We'll perform the Shapiro-Wilk test for check the normality

```
from scipy import stats
```

```
## perform the Shapiro-Wilk test
statistic, p_value = stats.shapiro(GLM_result['Marks'])
print("P_value :",p_value)

### set the significance level (alpha)
alpha = 0.05

### check the p-value against the significance level
if p_value > alpha:
    print("Sample appears to be normally distributed (fail to reject H0)")
else:
    print("Sample does not appear to be normally distributed (reject H0)")
```

```
P_value : 0.276019424200058
Sample appears to be normally distributed (fail to reject H0)
```

Find the correlation between Marks & Time

```
### Find the correlation

from scipy.stats import pearsonr

# Calculate Pearson correlation
correlation_coefficient, p_value =
pearsonr(GLM_result['Marks'],GLM_result['x1'])

print("Pearson Correlation Coefficient:", correlation_coefficient)
print("p-value:", p_value)
```

```
Pearson Correlation Coefficient: 0.1658278994289421
p-value: 0.2939281374382552
```

Result & Interpretation :

- The correlation coefficient = 0.1658278994289421, the correlation is relatively weak & value is close to zero we can say that there is a positive relationship.
- From the P-value = 0.2939281374382552 is greater than the commonly used significance level of 0.05, So we accept the null hypothesis.

```
x = GLM_result[['course', 'x1', 'x2', 'x3']]
y = GLM_result['Marks']
```

Split the data

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,random_state = 0,
train_size = 33, test_size = 9)
```

We apply the Linear Regression

```
## We apply the Linear Regression
lr = LinearRegression(fit_intercept = True)
lreg = lr.fit(x_train,y_train)

print("R-Square :",lreg.score(x_train,y_train))
```

```
R-Square : 0.2330715354441164
```

Calculate the Predicted Y

```
## calculate the Y_Predicted

y_pred = lr.predict(x_test)
y_pred
```

```
array([ 8.03535144,  8.03535144,  8.10437784,  6.4656565 ,  8.94
352228,
       10.13445337,  7.79405281,  6.4656565 , 10.89198106])
```

Calculate the Residual

```
### calculate the residual
resid = y_test - y_pred
resid
```

```
30   -0.785351
36   -3.035351
27   -3.104378
4     0.034344
10    0.056478
25   -1.884453
28   -4.294053
11   -1.965656
```

```
37   -4.391981
Name: Marks, dtype: float64
```

We'll calculate the RMSE & MSE

```
## RMSE
RMSE = np.sqrt(np.mean(resid*resid))
print("RMSE :",RMSE)

## Calculate the MSE
MSE = np.mean((y_pred - y_test)**2)
print("MSE :",MSE)
```

```
RMSE : 2.67944901801981
MSE : 7.179447040167324
```

Fit a multiple regression model (OLS)

```
import statsmodels.api as sm

### Fit a multiple regression model

x2 = sm.add_constant(x)  ## add a constant (intercept) term

model = sm.OLS(y, x2).fit()

print(model.summary())

#### multiple correlation coefficient (R-squared)
multiple_corr = model.rsquared

print("multiple Correlation (Mark vs. course, x1= time, x2= anxiety, and x3=
gender):", multiple_corr)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Marks   R-squared:                       0.229
Model:                            OLS   Adj. R-squared:                  0.145
Method:                 Least Squares   F-statistic:                     2.740
Date:                Mon, 09 Oct 2023   Prob (F-statistic):             0.0430
Time:                        00:01:02   Log-Likelihood:                -92.603
No. Observations:                  42   AIC:                             195.2
Df Residuals:                      37   BIC:                             203.9
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         10.2435      1.503      6.815      0.000       7.198      13.289
course        -0.4806      0.866     -0.555      0.582      -2.235       1.274
x1             0.0015      0.004      0.363      0.719      -0.007       0.010
x2            -0.2213      0.204     -1.087      0.284      -0.634       0.191
x3            -2.3373      0.835     -2.799      0.008      -4.029      -0.646
==============================================================================
Omnibus:                        1.981   Durbin-Watson:                   1.988
Prob(Omnibus):                  0.371   Jarque-Bera (JB):                1.512
Skew:                           0.274   Prob(JB):                        0.470
Kurtosis:                       2.249   Cond. No.                         733.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
multiple Correlation (Mark vs. course, x1= time, x2= anxiety, and x3= gender): 0.2285492303875335
```

Result & Interpretation :

- 1) R-squared: 0.229
  - R-squared: 0.229 that is 22% of the variance in dependent variable is explained by the model.
- 2) Adj. R-squared: 0.145
  - Adj. R-squared: 0.145 i.e. Explained about 15% of the variability.
- 3) AIC = 195.2
  - An AIC value of 195 suggests that the model has relatively good fit to the data while considering the complexity.
- 4) BIC = 203.9
  - A BIC value of 203.9 suggests that the model is relatively good in terms of fitting the data.
- 5) Kurtosis = 2.249
  - The Kurtosis value is less than 3 but greater then 0, we conclude that the distribution heavier-tailed than normal distribution, but not extremely so.
- 6) Durbin-Watson = 1.988

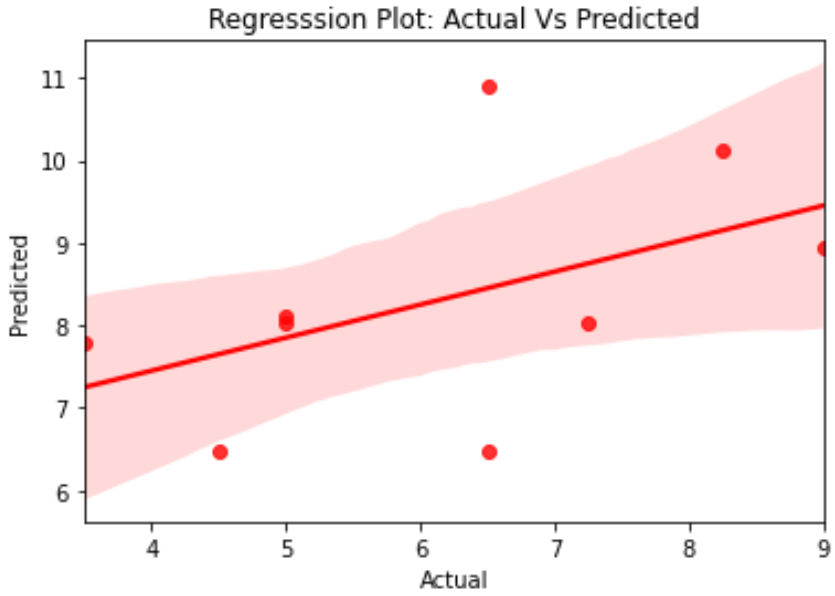
    - it's close to the value of 2, which indicates that the residuals are independent and there is no systematic pattern of correlation between consecutive residuals.
- 7) Prob(JB) = 0.470
    - Prob(JB) value of 0.470 greater than 0.05(alpha).So we conclude that, we accept the null hypothesis (H0), suggesting that there is no strong evidence to indicate that the residuals significantly from the normal distribution.
- 8) Cond. No. = 733
    - from Condition number, there is a high degree of multicollinearity among the independent variables.

Interpretation : P - value

- course, x1= time, x2 = Anxiety Level those P-values are greater than 0.05, So we accept H0. i.e. there is no significance difference.
- x3 = Gender ; P-value is less than 0.05, So we reject H0. i.e. there is significance difference.

Regression Plot: Actual Vs Predicted

```
sns.regplot(x = y_test, y = y_pred, color = "red")
plt.ylabel("Predicted")
plt.xlabel("Actual")

plt.title("Regression Plot: Actual Vs Predicted")
plt.show()
```



Perform the t-test for "Marks_pure_stat" & "Marks_applied_stat"

```
marks_pure_stat = GLM_result[GLM_result['course'] == 1]['Marks']  ### Pure
Statistics = 1

marks_applied_stat = GLM_result[GLM_result['course'] == 0]['Marks']  ###
Applied Statistics = 0

t_statistic, p_value = stats.ttest_ind(marks_pure_stat, marks_applied_stat )
print('P_value :',p_value)

### Here we set the significance level
alpha = 0.05

#### We'll check the p-value against the significance level
if p_value < alpha:
    print("There is a significant relationship between 'marks_pure_stat' and
'marks_applied_stat'.")
else:
    print("There is no significant relationship between 'marks_pure_stat'
and 'marks_applied_stat'.")
```

```
P_value : 0.8808150695031178
There is no significant relationship between 'marks_pure_stat' a
nd 'marks_applied_stat'.
```

Result & Interpretation :

- 5% level of significance (alpha) = 0.05
- P_value : 0.8808150695031178

- Here p-value is (0.8808150695031178) greater than the alpha (p-value > alpha), So we conclude that, we accept the null hypothesis (H0) at 5% level of significance.
    - So we can say that, there is no significant relationship between 'Marks_pure_stat' and 'Marks_applied_stat'.

Perform the t-test for Gender (x3)

```
marks_male = GLM_result[GLM_result['x3'] == 1]['Marks']  ### Male = 1

marks_female = GLM_result[GLM_result['x3'] == 0]['Marks']  ### female = 0

t_statistic, p_value = stats.ttest_ind(marks_male, marks_female )
print('P_value :',p_value)
```

```
### Here we set the significance level
alpha = 0.05

#### We'll check the p-value against the significance level
if p_value < alpha:
    print("There is a significant relationship between 'marks_male' and
'marks_female'.")
else:
    print("There is no significant relationship between 'marks_male' and
'marks_female'.")
```

```
P_value : 0.003419197114342527
There is a significant relationship between 'marks_male' and
'marks_female'.
```

Result & Interpretation :

- 5% level of significance (alpha) = 0.05
- P_value : 0.003419197114342527

- Here p-value is (0.003419197114342527) smaller than the alpha (p-value < alpha), So we conclude that, we reject the null hypothesis (H0) at 5% level of significance.
    - So we can say that, there is a significant relationship between 'Marks_male' and 'Marks_male'.

Import the GLM Result Applied & Pure Statistics

- dependent = Marks
    - Pass_students (greater than or equal to 8 marks) = 1
    - Fail_students (less than 8 marks) = 0

```
GLM_result_Bin = pd.read_csv("D:/M.SC 3sem 2023/GLM/Binary Result GLM.csv")
GLM_result_Bin
```

	Marks	course	x1	x2	x3
0	1	1	70	7	0
1	1	1	150	3	0
2	0	1	50	6	0
3	0	1	45	7	1
4	0	1	60	4	1
5	1	1	100	4	1
6	0	1	50	4	1
7	1	1	60	6	0
8	1	1	20	6	0
9	0	1	120	5	0
10	1	1	150	4	0
11	0	1	60	4	1
12	0	1	50	4	1
13	0	1	60	4	1
14	1	1	60	3	1
15	1	1	120	6	0
16	1	1	90	4	0
17	0	1	60	4	1
37	0	0	60	5	0
38	1	0	40	5	1
39	1	0	60	9	1
40	0	0	30	1	1
41	0	0	360	7	1

We apply Multinomial Logit Model

```
X = GLM_result_Bin[['course', 'x1', 'x2', 'x3']]

X = sm.add_constant(X)
## independent variable
Y = GLM_result_Bin['Marks']

# Fit the binary cumulative logit model
```

```
model = sm.MNLogit(Y, X)
result = model.fit()

# Get the summary of the model
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.631069
         Iterations 5
                          MNLogit Regression Results
==============================================================================
Dep. Variable:                  Marks   No. Observations:                   42
Model:                        MNLogit   Df Residuals:                       37
Method:                           MLE   Df Model:                            4
Date:                Mon, 09 Oct 2023   Pseudo R-squ.:                 0.07591
Time:                        00:01:04   Log-Likelihood:                -26.505
converged:                       True   LL-Null:                       -28.682
Covariance Type:            nonrobust   LLR p-value:                    0.3602
==============================================================================
   Marks=1       coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2617      1.397      0.187      0.851      -2.477       3.000
course        -0.1266      0.804     -0.158      0.875      -1.702       1.449
x1            -0.0007      0.004     -0.173      0.863      -0.008       0.007
x2             0.0870      0.187      0.466      0.641      -0.279       0.453
x3            -1.3907      0.762     -1.825      0.068      -2.885       0.103
==============================================================================
```

Interpretation :

- Here all P-values greater than 0.05, So we conclude that, we accept the null hypothesis (H0) at 5% level of significance. So we can say that, there is no significant difference.

We calculate the Odds ratio of Gender (x3)

```
odds_ratios = np.exp(-1.3907)
odds_ratios
print("odds_ratios :", odds_ratios)
```

```
odds_ratios : 0.24890101292763933
```

Result & Interpretation :

- odds ratio = 0.24890101292763933
    - odds ratio is 0.24890101292763933 (i.e. 25%), it indicates that for every one-unit increase in the x3 (Gender), it means that the odds of the dependent variable (Marks) happening decrease by approximately 75%

Conclusion

Applied Statistics Result :

1) From Multiple Bar Plot

- From this plot we can see that each student marks compare to their anxiety level.
- Also identify students with high marks and high or low anxiety levels, and vice versa.
    - For eg : "Manali Laxman Nalawade" has highest marks & moderate anxiety level.

Pure Statistics Result :

2) From Multiple Bar Plot

- From this plot we can see that each student marks compare to their anxiety level.
- Also identify students with high marks and high or low anxiety levels, and vice versa.
    - For eg : Roll no. 4 has low marks & high anxiety level

GLM Result Applied & Pure Statistics :

3) From Line Plot Student Performance:

- Each point on the line represents the marks obtained by an individual student. From those points, you can see how each student's marks vary. This plot can help identify outliers or students with exceptional performance.
- So we can say that Sr. no. 30 to 35 show a highest marks points.

4) From the Histogram :

From the histogram, We can say that our dependent (Marks) variable show the normal (bell shaped) pattern

5) Frm Shapiro-Wilk test for check the normality

Sample appears to be normally distributed (fail to reject H0)

6) From Fit a multiple regression model (OLS) :

- 22% of the variance in dependent variable is explained by the model.
- model has relatively good fit to the data while considering the complexity.
- Distribution heavier-tailed than normal distribution, but not extremely so.
- the residuals are independent and there is no systematic pattern of correlation between consecutive residuals.
- There is a high degree of multicollinearity among the independent variables.

7) From Perform the t-test for "Marks_pure_stat" & "Marks_applied_stat"

We can say that, there is no significant relationship between 'Marks_pure_stat' and 'Marks_applied_stat'.

8) From Perform the t-test for Gender (x3) :

We can say that, there is a significant relationship between 'Marks_male' and 'Marks_male'.

9) From Multinomial Logit Model :

From P-values, There is no significant difference.

10) From Odds ratio of Gender (x3) :

It indicates that for every one-unit increase in the x3 (Gender), it means that the odds of the dependent variable (Marks) happening decrease by approximately 75%

Thank You..!!