שאלה 1

א. הוכיחו כי $OR = \dfrac{odds(E=1|D=1)}{odds(E=1|D=0)}$

$$OR = \frac{odds(D=1|E=1)}{odds(D=1|E=0)} = \frac{\left(\frac{P(D=1|E=1)}{P(D=0|E=1)}\right)}{\left(\frac{P(D=1|E=0)}{P(D=0|E=0)}\right)} =$$

$$= \frac{P(D=1|E=1) \cdot P(D=0|E=0)}{P(D=0|E=1) \cdot P(D=1|E=0)} \underset{\text{bayes}}{=}$$

$$= \frac{\left(\frac{P(E=1|D=1)\,P(D=1)}{P(E=1)}\right)\left(\frac{P(E=0|D=0)\,P(D=0)}{P(E=0)}\right)}{\left(\frac{P(E=1|D=0)\,P(D=0)}{P(E=1)}\right)\left(\frac{P(E=0|D=1)\,P(D=1)}{P(E=0)}\right)}$$

$$= \frac{P(E=1|D=1)P(D=1)P(E=0|D=0)P(D=0)P(E=1)P(E=0)}{P(E=1|D=0)P(D=0)P(E=0|D=1)P(D=1)P(E=1)P(E=0)}$$

$$= \frac{P(E=1|D=1)\,P(E=0|D=0)}{P(E=1|D=0)\,P(E=0|D=1)} = \frac{P(E=1|D=1)}{P(E=0|D=1)} \cdot \frac{P(E=0|D=0)}{P(E=1|D=0)}$$

$$= odds(E=1|D=1) \cdot \frac{1}{odds(E=1|D=0)} = \frac{odds(E=1|D=1)}{odds(E=1|D=0)} \blacksquare$$

ב. למצוא אומד ל- OR במודל Case Control:

לפי איך שלוקחים את הנתונים ב- Case Control יכולים לאמוד את ההסתברות לאנשים שנחשפו או לא נחשפו מכיוון שהם חולים או בריאים כפרופורציה:

לא להיחשף כשבריאים:
$$\hat{P}(E=0|D=0)=\frac{X_{00}}{x_0}$$

להיחשף כשבריאים:
$$\hat{P}(E=1|D=0)=\frac{X_{10}}{x_0}$$

לא להיחשף כשחולים:
$$\hat{P}(E=0|D=1)=\frac{X_{01}}{x_1}$$

להיחשף כשחולים:
$$\hat{P}(E=1|D=1)=\frac{X_{11}}{x_1}$$

ולכן מספיק אז נקבל:

$$\hat{OR}=\frac{\hat{odds}(E=1|D=1)}{\hat{odds}(E=1|D=0)}=\frac{\hat{P}(E=1|D=1)}{\hat{P}(E=0|D=1)}\cdot\frac{\hat{P}(E=0|D=0)}{\hat{P}(E=1|D=0)}$$

$$=\frac{\left(\frac{X_{11}}{x_1}\cdot\frac{X_{00}}{x_0}\right)}{\left(\frac{X_{01}}{x_1}\cdot\frac{X_{10}}{x_0}\right)}=\frac{X_{11}\cdot X_{00}}{X_{10}\cdot X_{01}}$$

שאלה 2

הראו והסבירו מה ההשפעה של עליה ביחידה אחת ב- X_1 במודלים הבאים:

1. $$\mathbb{E}[Y|X] = \pi(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{1,2} X_1 X_2)$$

$$\tilde{\pi} = \pi(\beta_0 + \beta_1(X_1+1) + \beta_2 X_2 + \beta_{1,2}(X_1+1)X_2)$$

$$\hat{\pi} = \pi(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{1,2} X_1 X_2)$$

$$\log\left(\frac{\frac{\tilde{\pi}}{1-\tilde{\pi}}}{\frac{\hat{\pi}}{1-\hat{\pi}}}\right) = \log\left(\frac{\tilde{\pi}}{1-\tilde{\pi}}\right) - \log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) =$$

$$= \beta_0 + \beta_1(X_1+1) + \beta_2 X_2 + \beta_{1,2}(X_1+1)X_2 - (\beta_0 + \beta_1 X_1 + \beta_2 X_2 +$$

$$+ \beta_{1,2} X_1 \cdot X_2) = \beta_1 + \beta_{1,2} X_2$$

$\Leftarrow$ הגדלת X_1 ביחידה אחת תגרור לשינוי ב- OR של $e^{\beta_1 + \beta_{1,2} X_2}$

2. $$\mathbb{E}(Y|X) = \pi(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2)$$

$$\log\left(\frac{\frac{\tilde{\pi}}{1-\tilde{\pi}}}{\frac{\hat{\pi}}{1-\hat{\pi}}}\right) = \beta_0 + \beta_1(X_1+1) + \beta_2 X_2 + \beta_3(X_1+1)^2 -$$

$$-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2) = \beta_1 + 2\beta_3 X_1 + \beta_3$$

$\Leftarrow$ הגדלת X_1 ביחידה אחת תגרום לשינוי ב- OR של $\beta_1 + \beta_3 + \beta_3 X_1$

שאלה 3

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.genmod.generalized_linear_model import GLM
import statsmodels.api as sm
import itertools
from scipy.stats import chi2
```

```
data = pd.read_csv('mine_fracture.csv')
Y = np.array(data['y'])
X = [np.array(data[f'x{i}']) for i in range(1, 5)]
X1, X2, X3, X4 = X
X_combined = [X1, X2, X3, X4]
```

Poison regression model:

$$\forall i \in [4]/\{0\} : E(Y_i|X_i) = e^{X_i^T \cdot \beta_0}$$
$$\implies E(Y|X) = e^{X^T \cdot \beta_0}$$

where

$$\beta_0$$

is the paramater that the GLM model calculates

```
X_combined = np.column_stack((X1, X2, X3, X4))
X_combined = sm.add_constant(X_combined)  # Add intercept

# Fit the Poisson regression model
model_A = GLM(Y, X_combined, family=sm.families.Poisson()).fit()
print(len(model_A.params), model_A.params)
print(model_A.summary())
```

```
-67.06383637505789
5 [-3.59308958e+00 -1.40658814e-03  6.23457605e-02 -2.08034186e-03
 -3.08134926e-02]
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                   44
Model:                            GLM   Df Residuals:                       39
Model Family:                 Poisson   Df Model:                            4
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -67.064
Date:                Thu, 26 Dec 2024   Deviance:                       37.856
Time:                        12:05:05   Pearson chi2:                     35.9
No. Iterations:                     5   Pseudo R-squ. (CS):             0.5699
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -3.5931      1.026     -3.503      0.000      -5.603      -1.583
x1            -0.0014      0.001     -1.683      0.092      -0.003       0.000
x2             0.0623      0.012      5.074      0.000       0.038       0.086
x3            -0.0021      0.005     -0.411      0.681      -0.012       0.008
x4            -0.0308      0.016     -1.894      0.058      -0.063       0.001
==============================================================================
```

```
beta_3 = model_A.params[2]
change_factor = np.exp(beta_3)
print(f"Multiplicative change in the expectation of Y for a one-unit increase in X3: {change_factor}")
```

```
Multiplicative change in the expectation of Y for a one-unit increase in X3: 1.0643302844937554
```

```
X_dict = {"X1": X1,"X2": X2,"X3": X3,"X4": X4}

assert len(Y) == len(X1) == len(X2) == len(X3) == len(X4), "Mismatch in length between Y and X variables"

pairs = list(itertools.combinations(X_dict.items(), 2))

best_aic = float('inf')
best_pair = None
best_model = None

for pair in pairs:
```

```python
    (name1, X1_array), (name2, X2_array) = pair
    X_pair = np.column_stack((X1_array, X2_array))
    X_pair = sm.add_constant(X_pair)
    model = sm.GLM(Y, X_pair, family=sm.families.Poisson()).fit()
    aic_score = model.aic
    if aic_score < best_aic:
        best_aic = aic_score
        best_pair = (name1, name2)
        best_model = model
print("Best model for D:")
print(f"Best pair of variables: {best_pair}")
print(f"Best AIC score: {best_aic}")
print(best_model.summary())
print(len(best_model.params))
```

```
Best model for D:
Best pair of variables: ('X2', 'X4')
Best AIC score: 143.8977622958081
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                   44
Model:                            GLM   Df Residuals:                       41
Model Family:                 Poisson   Df Model:                            2
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -68.949
Date:                Thu, 26 Dec 2024   Deviance:                       41.626
Time:                        12:02:07   Pearson chi2:                     40.7
No. Iterations:                     5   Pseudo R-squ. (CS):             0.5315
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -3.5894      0.945     -3.799      0.000      -5.441      -1.738
x1             0.0587      0.012      5.027      0.000       0.036       0.082
x2            -0.0380      0.015     -2.460      0.014      -0.068      -0.008
==============================================================================
3
```

```
params_model1 = len(model_A.params)
params_model2 = len(best_model.params)
print(f"Number of parameters in model1: {params_model1}")
print(f"Number of parameters in model2: {params_model2}")
```

```
Number of parameters in model1: 5
Number of parameters in model2: 3
```

```
log_likelihood_model1 = best_model.llf  # Log-likelihood for simpler model
log_likelihood_model2 = model_A.llf  # Log-likelihood for more complex model
print(f"Log-likelihood for model1: {log_likelihood_model1}")
print(f"Log-likelihood for model2: {log_likelihood_model2}")
lr_statistic = -2 * (log_likelihood_model1 - log_likelihood_model2)

df_diff = params_model1 - params_model2

p_value = chi2.sf(lr_statistic, df_diff)

print(f"Likelihood Ratio Test Statistic: {lr_statistic}")
print(f"Degrees of Freedom: {df_diff}")
print(f"P-value: {p_value}")

if p_value < 0.05:
    print("Model A is better.")
else:
    print("Reduced model performs as well as Model A, therefore, model D is good")
```

```
Log-likelihood for model1: -68.94888114790405
Log-likelihood for model2: -67.06383637505789
Likelihood Ratio Test Statistic: 3.770089545692315
Degrees of Freedom: 2
P-value: 0.15182226176248714
Reduced model performs as well as Model A, therefore, model D is good
```

```
print('Checking which model is better with BIC')
print(f"BIC for model from A: {model_A.bic}")
print(f"BIC for model from choosing two X: {best_model.bic}")

if model_A.bic < best_model.bic:
```

```
if model_A.bic < best_model.bic:
    print("model A is better (lower BIC).")
else:
    print("Reduced model is better (lower BIC).")
```

```
Checking which model is better with BIC
BIC for model from A: -109.72737135082559
BIC for model from choosing two X: -113.52566107296977
Reduced model is better (lower BIC).
```