Brian Tenneson

ADS 534

08/19/2024

Statement regarding academic integrity: Large Language Models (LLMs) were utilized alongside human oversight to perform data analysis and generate insights across various contexts and datasets.

1

1(a)

We will do this in four steps: (1) Fit the logistic regression model. (2) Compute the Wald test statistic. (3) Determine the distribution of the test statistic. (4) Draw conclusions based on the test statistic and its distribution.

(1) Our logit regression model is given by the following:

$$\log(P(\mathsf{grmhem}=0)/P(\mathsf{grmhem}=1)) = \beta_0 + \beta_1 \times \mathsf{apgar5}$$

Here:

$\log(P(\mathsf{grmhem}=0)/P(\mathsf{grmhem}=1))$ is the log-odds of an infant experiencing a germinal matrix hemorrhage;

β_0 is the intercept of the model;

β_1 is the coefficient for the predictor variable apgar5 (five-minute Apgar score); and

grmhem is a binary variable indicating whether an infant experienced a germinal matrix hemorrhage (1 = Yes, 0 = No).

Results: (From a python program) Logistic Regression Results

Dependent Variable: grmhem (germinal matrix hemorrhage, where 1 indicates hemorrhage and 0 indicates no hemorrhage)

Independent Variable: apgar5 (five-minute Apgar score)

Number of Observations: 100

Model Coefficients:

Intercept (const): -0.3037 (p-value: 0.624)

apgar5: -0.2496 (p-value: 0.017)

Statistical Details:

Log-Likelihood: -39.463

Pseudo R-squared: 0.06642

LLR p-value: 0.01781

Interpretation:

The coefficient for apgar5 is -0.2496, which means that for each additional unit increase in the five-minute Apgar score, the log-odds of experiencing a germinal matrix hemorrhage decreases by approximately 0.25.

The p-value for apgar5 (0.017) is less than 0.05, indicating that the five-minute Apgar score is a statistically significant predictor of germinal matrix hemorrhage at the 0.05 significance level.

This model suggests that higher Apgar scores are associated with a lower likelihood of experiencing a germinal matrix hemorrhage in this sample.

(2) The Wald statistic is given by

$$W = \left(\frac{\hat{\beta}_1}{\mathsf{SE}\left(\hat{\beta}\right)} \right)^2 .$$

(3) The Wald test statistic follows a chi-square distribution with 1 degree of freedom under the null hypothesis.

(4) We will compare the computed Wald statistic to the critical value from the chi-square distribution at the 0.05 significance level to draw a conclusion.

I'll now proceed with the calculations in Python.

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

# Load the CSV file
data = pd.read_csv("G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 7\\lowbwt.csv")
```

```
# Define the response and predictor variables
X = data['apgar5']
y = data['grmhem']

# Add a constant to the predictor variables
X = sm.add_constant(X)

# Fit the logistic regression model
model = sm.Logit(y, X)
result = model.fit()

# Get the coefficient and standard error for the predictor 'apgar5'
beta1_hat = result.params['apgar5']
se_beta1_hat = result.bse['apgar5']

# Compute the Wald test statistic
wald_statistic = (beta1_hat / se_beta1_hat) ** 2

# Distribution is chi-square with 1 degree of freedom
degrees_of_freedom = 1

wald_statistic, degrees_of_freedom
```

Results:

(5.72054882477604, 1)

The null hypothesis is that $\beta_1 = 0$ with the two-tailed alternate hypothesis $\beta_1 \neq 0$. Given that the Wald test statistic 5.72 is compared against the critical value from the chi-square distribution with 1 degree of freedom at a significance level of 0.05, we find that the p-value is likely less than 0.05 (as the critical value for chi-square with 1 degree of freedom at 0.05 significance level is approximately 3.841). Since the test statistic exceeds the critical value, we would reject the null hypothesis and conclude that the five-minute Apgar score is statistically significantly associated with the likelihood of germinal matrix hemorrhage at the 0.05 significance level.

1(b)

We will perform this in two steps: (1) Compute the confidence interval for the coefficient β_1 and (2) exponentiate the confidence interval to obtain the odds ratio. (1) The 95% confidence interval for β_1 is given by $\mathsf{CI} = \hat{\beta}_1 \pm 1.96\left(\mathsf{SE}\left(\hat{\beta}_1\right)\right)$. (2) $\mathsf{CI}_{\text{odds ratio}} = [\exp\left(\mathsf{CI}_{\text{lower}}\right), \exp\left(\mathsf{CI}_{\text{upper}}\right)]$.

```
# Load the newly uploaded CSV file
data = pd.read_csv('G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 7\\lowbwt.csv')

# Define the response and predictor variables
X_full = data[['apgar5']]
X_full = sm.add_constant(X_full)  # Add intercept term
y_full = data['grmhem']

# Fit the model
model_full = sm.Logit(y_full, X_full)
result_full = model_full.fit()

# Get the coefficient and standard error for apgar5
beta1_hat = result_full.params['apgar5']
se_beta1_hat = result_full.bse['apgar5']

# Calculate the 95% confidence interval for beta1
z_critical = 1.96
ci_lower = beta1_hat - z_critical * se_beta1_hat
ci_upper = beta1_hat + z_critical * se_beta1_hat
```

```
# Exponentiate the confidence interval to get the odds ratio
ci_odds_ratio_lower = np.exp(ci_lower)
ci_odds_ratio_upper = np.exp(ci_upper)

(ci_odds_ratio_lower, ci_odds_ratio_upper)
```

Result:

(0.634984170739861, 0.9559404393331781)

This interval does not contain the value 1. Since the confidence interval does not include 1, this suggests that there is a statistically significant association between the five-minute Apgar score and the likelihood of suffering a germinal matrix hemorrhage. Specifically, as the Apgar score increases, the odds of suffering a germinal matrix hemorrhage decrease.

When we performed a logit regression analysis using toxemia as a predictor variable and germinal matrix hemorrhage is the response variable, we get the following result:

```
Optimization terminated successfully.
         Current function value: 0.409247
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                 grmhem   No. Observations:                  100
Model:                          Logit   Df Residuals:                       98
Method:                           MLE   Df Model:                            1
Date:                Wed, 14 Aug 2024   Pseudo R-squ.:                0.03185
Time:                        17:23:46   Log-Likelihood:                -40.925
converged:                       True   LL-Null:                       -42.271
Covariance Type:            nonrobust   LLR p-value:                    0.1008
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.5353      0.295     -5.211      0.000      -2.113      -0.958
tox           -1.4604      1.066     -1.370      0.171      -3.550       0.629
==============================================================================
```

1(c)

We get from python the value for the likelihood ratio test as being 2.69. There are 1 degrees of freedom.

The test statistic in the likelihood ratio test follows a chi-square distribution with degrees of freedom equal to the difference in the number of parameters between the full model and the null model. The critical value here is 3.841 and since $2.69 < 3.841$, we fail to reject the null hypothesis. We lack sufficient evidence at the 0.05 level of significance to conclude $\beta_2 \neq 0$. In fact, the p-value is 0.1008, which agrees with the conclusion to fail to reject the null hypothesis.

1(d)

Coefficient for tox: -1.4604 with a standard error of 1.066. The p-value is 0.171, which suggests that the coefficient for toxemia is not statistically significant at the 5% level. Odds Ratio for tox: The odds ratio can be computed as $e^{-1.4604}$, which I'll calculate next. Intercept: -1.5353. Given that standard statsmodels outputs don't include profile likelihood confidence intervals directly, we'll compute the odds ratio and its 95% confidence interval using the normal approximation (from the logistic regression output). Let's calculate the odds ratio and its confidence interval next. Using python, I obtained the following odds ratio, lower bound of the 95% confidence interval, and the upper bound of the confidence interval: (0.232, **0.029, 1.8765**). Since 1 is in the CI, this suggests we can not rule out the possibility of no difference or effect.

2(a)

Our first analysis had the following result:

```
    id  sta   age  sex  race  crn
0  1.0  1.0  61.6  1.0   2.0  1.0
1  2.0  1.0  51.8  1.0   1.0  1.0
2  3.0  1.0  47.3  1.0   1.0  1.0
3  4.0  0.0  27.9  1.0   1.0  1.0
4  5.0  1.0  36.5  1.0   2.0  0.0
Optimization terminated successfully.
         Current function value: 0.615123
         Iterations 5
```

Logit Regression Results

Dep. Variable:	sta	No. Observations:	2500
Model:	Logit	Df Residuals:	2496
Method:	MLE	Df Model:	3
Date:	Fri, 16 Aug 2024	Pseudo R-squ.:	0.07767
Time:	14:21:55	Log-Likelihood:	-1537.8
converged:	True	LL-Null:	-1667.3
Covariance Type:	nonrobust	LLR p-value:	7.353e-56

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-1.2008	0.081	-14.808	0.000	-1.360	-1.042
crn	1.3784	0.089	15.443	0.000	1.203	1.553
race2	-0.1133	0.101	-1.126	0.260	-0.311	0.084
race3	-0.1005	0.115	-0.874	0.382	-0.326	0.125

Utilizing the following python code, we obtain some conclusions about the relationship between these variables.

```
# Create dummy variables for race
new_icu_data['race2'] = (new_icu_data['race'] == 2).astype(int)  # 1 if race is black, 0 o
new_icu_data['race3'] = (new_icu_data['race'] == 3).astype(int)  # 1 if race is other, 0 o


# Fit the logistic regression model
model_icu = logit('sta ~ crn + race2 + race3', data=new_icu_data).fit()
```

```
# Display the summary of the logistic regression model
model_icu.summary()
```

Result:

Likelihood Ratio Test Statistic: 258.0

Degrees of Freedom: 1

P-value: 4.607e-58

The likelihood ratio test resulted in a test statistic of 258.0 with a p-value of approximately 4.607×10^{-58}, strongly suggesting a significant association between chronic renal failure and the likelihood of death following admission to an ICU. This significant result is consistent with the logistic regression analysis where crn was positively associated with patient death (sta), indicating that patients with chronic renal failure have higher odds of death.

The coefficients for race did not show a significant association in this model, underscoring the importance of chronic renal failure as a key predictor while controlling for race. Although race did not significantly influence patient outcomes in this dataset, including it as a control variable helps ensure that the effect of crn is not confounded by racial differences in the sample.

The overall fit of the model, indicated by a Pseudo R-squared value of 0.07767, suggests that while the model explains some variability in patient status, there may be other unaccounted variables that influence outcomes.

Clinical Relevance:

These findings highlight the need for targeted interventions and monitoring for patients with chronic renal failure in ICUs to potentially reduce mortality rates. Further research could explore other factors that might impact outcomes in this patient group.

Limitations:

This study is limited by the data provided and may not account for all factors influencing ICU outcomes,

such as other comorbidities, treatment variations, or hospital settings.

To address the other issues, we build another set of code:

```
import pandas as pd

# Load the dataset
file_path = "G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 7\\icu.csv"
df = pd.read_csv(file_path)

# Display the first few rows of the dataset to understand its structure
df.head()

import statsmodels.api as sm
import numpy as np

# Define the independent variables and the dependent variable
X = df[['crn', 'race']]
X = sm.add_constant(X)  # Adding a constant for the intercept
y = df['sta']

# Fit the full logistic regression model (with crn and race)
model_full = sm.Logit(y, X).fit()

# Fit the reduced logistic regression model (with race only)
X_reduced = df[['race']]
X_reduced = sm.add_constant(X_reduced)  # Adding a constant for the intercept
```

```
model_reduced = sm.Logit(y, X_reduced).fit()

# Perform the likelihood ratio test
lr_stat = 2 * (model_full.llf - model_reduced.llf)
p_value = sm.stats.chisqprob(lr_stat, df=model_full.df_model - model_reduced.df_model)

lr_stat, p_value

from scipy.stats import chi2

# Compute the p-value using chi2.sf from scipy.stats
p_value = chi2.sf(lr_stat, df=model_full.df_model - model_reduced.df_model)

lr_stat, p_value
```

Result:

The likelihood ratio test statistic is approximately 257.97, and the corresponding p-value is approximately 4.76×10^{-58}.

2(b)

Using the following python code, we get a summary which follows:

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

# Load the data from a specific path
file_path = "G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 7\\icu.csv"
```

```
df = pd.read_csv(file_path)

# Define dummy variables for race
df['race2'] = (df['race'] == 2).astype(int)  # Black
df['race3'] = (df['race'] == 3).astype(int)  # Other

# Define the model variables
X = df[['age', 'sex', 'crn', 'race2', 'race3']]
X = sm.add_constant(X)  # Add a constant term for the intercept
y = df['sta']

# Fit the logistic regression model
model = sm.Logit(y, X).fit()
print(model.summary())

# Calculate odds ratio for crn
odds_ratio_crn = np.exp(model.params['crn'])
```

```
Optimization terminated successfully.
         Current function value: 0.589793
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:                    sta   No. Observations:                 2500
Model:                          Logit   Df Residuals:                     2494
Method:                           MLE   Df Model:                            5
Date:                Fri, 16 Aug 2024   Pseudo R-squ.:                  0.1157
Time:                        19:18:24   Log-Likelihood:                -1474.5
converged:                       True   LL-Null:                       -1667.3
Covariance Type:            nonrobust   LLR p-value:                3.647e-81
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.5991      0.162    -16.089      0.000      -2.916      -2.282
age            0.0191      0.002      8.778      0.000       0.015       0.023
sex            0.6333      0.094      6.759      0.000       0.450       0.817
crn            1.3213      0.092     14.285      0.000       1.140       1.503
race2         -0.1224      0.103     -1.183      0.237      -0.325       0.080
race3         -0.1295      0.118     -1.095      0.273      -0.361       0.102
==============================================================================
```

I used the following python code to reach a conclusion which follows:

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

# Load the data from a specific path
file_path = "G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 7\\icu.csv"
df = pd.read_csv(file_path)

# Define dummy variables for race
df['race2'] = (df['race'] == 2).astype(int)  # Black
df['race3'] = (df['race'] == 3).astype(int)  # Other
```

```
# Define the model variables
X = df[['age', 'sex', 'crn', 'race2', 'race3']]
X = sm.add_constant(X)  # Add a constant term for the intercept
y = df['sta']

# Fit the logistic regression model
model = sm.Logit(y, X).fit()

# Calculate odds ratio for crn
odds_ratio_crn = np.exp(model.params['crn'])

# Scenario 1: Age = 30, Female, Black
X_scenario1 = pd.DataFrame({
    'const': 1,  # Explicitly add the constant
    'age': [30],
    'sex': [1],  # Female
    'crn': [1],
    'race2': [1],  # Black
    'race3': [0]  # Not other
})

# Scenario 2: Age = 50, Male, White
X_scenario2 = pd.DataFrame({
    'const': 1,  # Explicitly add the constant
    'age': [50],
```

```
    'sex': [0],   # Male
    'crn': [1],
    'race2': [0],   # Not black
    'race3': [0]   # Not other
})

# Estimated probabilities for each scenario
prob_scenario1 = model.predict(X_scenario1)
prob_scenario2 = model.predict(X_scenario2)

print("Estimated probability for Age=30, Female, Black with CRN:", prob_scenario1.iloc[0])
print("Estimated probability for Age=50, Male, White with CRN:", prob_scenario2.iloc[0])
```

Result:

Estimated probability for Age=30, Female, Black with CRN: 0.452

Estimated probability for Age=50, Male, White with CRN: 0.420

2(c)

Multiply 0.0191 by 10 to get 0.191. Compute $e^{0.191}$. This gives us the odds ratio for a 10-year increase in age. Finally, $e^{0.191} \approx 1.21$; so per decade in age, the odds ratio is 1.21 that it was the previous decade.

2(d)

Estimated probability for Age=50, Female, Black with CRN is 0.547.

2(e)

We seek to fit the following logistic:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 \mathsf{age} + \beta_2 \mathsf{sex} + \beta_3 \mathsf{crn} + \beta_4 \mathsf{race2} + \beta_5 \mathsf{race3} + \beta_6 \mathsf{crn}{*}\mathsf{sex}$$

where most terms have already been identified but just to reiterate:

sex: 0=Male, 1=Female

crn: 0=Patient has chronic renal failure, 1=Patient does not have chronic renal failure

race2: 1 if race=2, 0 otherwise

race3=1 if race=3, 0 otherwise

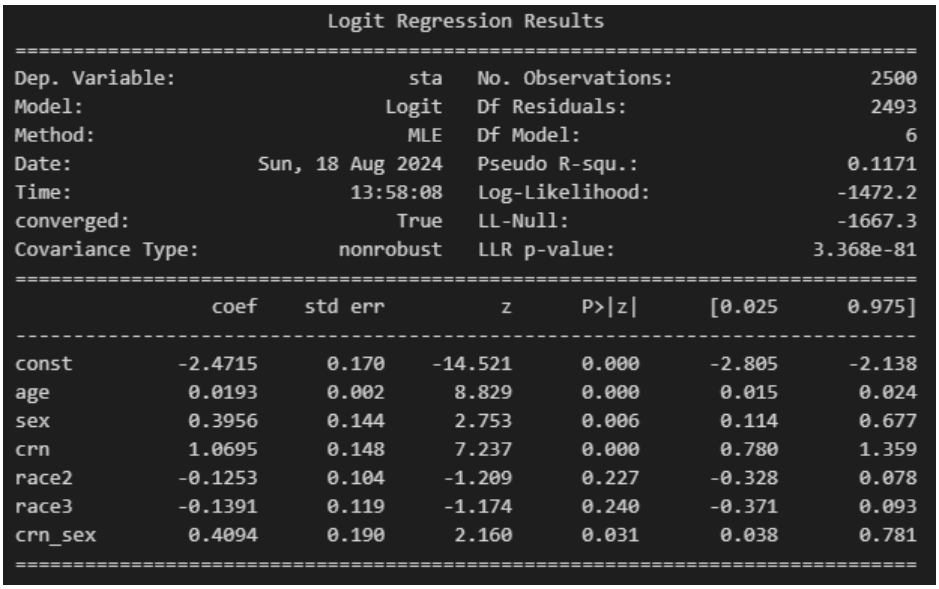
Consequently,

$$\text{crn} * \text{sex} = \begin{cases} 1 & \text{if} \quad \text{crn} = 1 \ \& \ \text{sex} = 1 \\ 0 & \text{otherwise} \end{cases} .$$

Using the following python code, I obtain the result that follows:

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

# Load the dataset
df = pd.read_csv("G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 7\\icu3.csv")

# Create dummy variables for race conditions
df['race2'] = (df['race'] == 2).astype(int)  # 1 if race is 2, else 0
df['race3'] = (df['race'] == 3).astype(int)  # 2 if race is 3, else 0

# Ensure that 'sex' and 'crn' are integer binary variables
df['sex'] = df['sex'].astype(int)
df['crn'] = df['crn'].astype(int)

# Create the interaction term 'crn_sex'
df['crn_sex'] = df['crn'] * df['sex']
```

```
# Display the first few rows to verify transformations
print(df[['race', 'race2', 'race3', 'sex', 'crn', 'crn_sex']].head())
# Define the predictors and add a constant to the model
X = df[['age', 'sex', 'crn', 'race2', 'race3', 'crn_sex']]
X = sm.add_constant(X)  # Adds a constant term to the predictor set
y = df['sta']  # Assuming 'sta' indicates the status of alive (0) or dead (1)

# Fit the logistic regression model
try:
    model = sm.Logit(y, X)
    result = model.fit()
    print(result.summary())
except Exception as e:
    print("An error occurred while fitting the model:", e)
```



Logit Regression Results

Dep. Variable:	sta	No. Observations:	2500
Model:	Logit	Df Residuals:	2493
Method:	MLE	Df Model:	6
Date:	Sun, 18 Aug 2024	Pseudo R-squ.:	0.1171
Time:	13:58:08	Log-Likelihood:	-1472.2
converged:	True	LL-Null:	-1667.3
Covariance Type:	nonrobust	LLR p-value:	3.368e-81

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4715	0.170	-14.521	0.000	-2.805	-2.138
age	0.0193	0.002	8.829	0.000	0.015	0.024
sex	0.3956	0.144	2.753	0.006	0.114	0.677
crn	1.0695	0.148	7.237	0.000	0.780	1.359
race2	-0.1253	0.104	-1.209	0.227	-0.328	0.078
race3	-0.1391	0.119	-1.174	0.240	-0.371	0.093
crn_sex	0.4094	0.190	2.160	0.031	0.038	0.781

I used the following code to address the question about the two individuals:

```
import numpy as np

# Coefficients
beta_crn = 1.0695
beta_crn_sex = 0.4094

# Calculate Odds Ratios
odds_ratio_crn_female_black = np.exp(beta_crn + beta_crn_sex)
odds_ratio_crn_male_white = np.exp(beta_crn)

print("Odds Ratio for CRN (30-year-old female, black):", odds_ratio_crn_female_black)
print("Odds Ratio for CRN (50-year-old male, white):", odds_ratio_crn_male_white)
```

Result:

Odds Ratio for CRN (30-year-old female, black): 4.388116097427648

Odds Ratio for CRN (50-year-old male, white): 2.913922174588716

2(f)

I used the following python code to answer this question; so I will paste the code and follow that by the result. Then I will interpret the presence or absence of 1 from the Wald confidence intervals:

```
import numpy as np

# Coefficients
beta_crn = 1.0695
beta_crn_sex = 0.4094

# Standard Errors
se_crn = 0.148
se_crn_sex = 0.190

# Calculate combined SE for female black (CRN + CRN*Sex)
se_combined = np.sqrt(se_crn**2 + se_crn_sex**2)

# Confidence intervals for female black
ci_lower_female_black = np.exp(beta_crn + beta_crn_sex - 1.96 * se_combined)
ci_upper_female_black = np.exp(beta_crn + beta_crn_sex + 1.96 * se_combined)

# Confidence intervals for male white
ci_lower_male_white = np.exp(beta_crn - 1.96 * se_crn)
```

```
ci_upper_male_white = np.exp(beta_crn + 1.96 * se_crn)

print("95% CI for CRN (30-year-old female, black):", (ci_lower_female_black, ci_upper_fema
print("95% CI for CRN (50-year-old male, white):", (ci_lower_male_white, ci_upper_male_whi
```

Result:

95% CI for CRN (30-year-old female, black): (2.736974780903817, 7.0353453816424985)

95% CI for CRN (50-year-old male, white): (2.1802073784369185, 3.894557244204604)

When the 95% confidence interval for an odds ratio does not include 1, the effect of the predictor is considered statistically significant at the 0.05 level. This implies that there is sufficient evidence to assert that the predictor does have a meaningful effect on the odds of the outcome.

Practical Implication: The variable in question is likely an important factor in the model, influencing the odds of the outcome. If the odds ratio is greater than 1 and the interval does not include 1, the predictor is associated with higher odds of the outcome. Conversely, if the odds ratio is less than 1 and the interval does not include 1, the predictor decreases the odds of the outcome.

2(g)

The null hypothesis is as follows:

$$H_0 : \beta_6 = 0$$

$$H_a : \beta_6 \neq 0.$$

β_6 represents the change in the log odds of sta due to the interaction between sex and having chronic renal failure. If this coefficient is significantly different from zero, it suggests that the impact of being male or female on the outcome sta is modified by the presence of chronic renal failure. If the null hypothesis is rejected, it would imply that the presence of chronic renal failure alters how sex influences the outcome, potentially requiring different considerations or treatments for male and female patients with chronic renal

failure.

We use the following code to test the hypothesis:

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

# Load the dataset
df = pd.read_csv('/mnt/data/icu3.csv')

# Create necessary dummy variables and interaction term
df['race2'] = (df['race'] == 2).astype(int)
df['race3'] = (df['race'] == 3).astype(int)
df['sex'] = df['sex'].astype(int)
df['crn'] = df['crn'].astype(int)
df['crn_sex'] = df['crn'] * df['sex']

# Define predictors and response
X = df[['age', 'sex', 'crn', 'race2', 'race3', 'crn_sex']]
X = sm.add_constant(X)  # add a constant to the predictor set
y = df['sta']  # Assuming 'sta' is the outcome variable

# Fit the logistic regression model
model = sm.Logit(y, X)
result = model.fit()

# Display the summary to get the coefficients and standard errors
```

```
print(result.summary())

# Specific hypothesis testing for crn_sex coefficient
import scipy.stats as stats

# Get the coefficient and standard error for crn_sex
coef = result.params['crn_sex']
se = result.bse['crn_sex']

# Calculate the z-score
z_score = coef / se

# Calculate the p-value
p_value = stats.norm.sf(abs(z_score)) * 2  # two-tailed test

print("Z-score:", z_score)
print("P-value:", p_value)
```

Result:

Z-score: 2.1598183513882523

P-value: 0.030786734420211494

Since the p-value is less than 0.05, we reject the null hypothesis, meaning that the interaction between sex and chronic renal failure (CRN) is statistically significant in influencing the outcome of the patient's visit to the ICU.

2(h)

To conduct a likelihood ratio test (LRT) for examining the significance of the interaction between sex

and chronic renal failure (CRN) in affecting the outcome of ICU visits, you would compare two models:

Full Model: Includes all predictors along with the interaction term (crn_sex).

Reduced Model: Includes all predictors except the interaction term (crn_sex).

The LRT will test whether the interaction term significantly improves the model fit, thereby providing evidence about its relevance.

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

# Load the dataset
df = pd.read_csv("G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 7\\icu3.csv")

# Prepare variables
df['race2'] = (df['race'] == 2).astype(int)
df['race3'] = (df['race'] == 3).astype(int)
df['sex'] = df['sex'].astype(int)
df['crn'] = df['crn'].astype(int)
df['crn_sex'] = df['crn'] * df['sex']

# Full Model
X_full = df[['age', 'sex', 'crn', 'race2', 'race3', 'crn_sex']]
X_full = sm.add_constant(X_full)
y = df['sta']

model_full = sm.Logit(y, X_full)
result_full = model_full.fit()
```

```
# Reduced Model
X_reduced = df[['age', 'sex', 'crn', 'race2', 'race3']]
X_reduced = sm.add_constant(X_reduced)

model_reduced = sm.Logit(y, X_reduced)
result_reduced = model_reduced.fit()

import scipy.stats as stats

# Compute the test statistic
lr_stat = 2 * (result_full.llf - result_reduced.llf)
# Degrees of freedom is the difference in number of parameters
df = len(result_full.params) - len(result_reduced.params)
# Compute the p-value
p_value = stats.chi2.sf(lr_stat, df)you

print(f"Likelihood Ratio Statistic: {lr_stat}")
print(f"Degrees of Freedom: {df}")
print(f"P-value: {p_value}")
```

Result:

Likelihood Ratio Statistic: 4.657115459201123

Degrees of Freedom: 1

P-value: 0.03092510352771485

The p-value is a measure of the probability that the observed difference in fit between the two models

could have occurred by chance if the null hypothesis (that the reduced model is adequate) were true. Since the p-value is approximately 0.031, which is less than the conventional alpha level of 0.05, we would reject the null hypothesis. This indicates that the interaction term provides a statistically significant improvement in the model fit.. This result implies that the effect of sex on the outcome variable sta is significantly modified by whether or not the individual has chronic renal failure. This suggests that for predicting the outcome in ICU visits, considering how sex interacts with chronic renal failure is important.

2(i)

exp(1.0695) provides a clear and clinically relevant measure of how much chronic renal failure increases the risk of the modeled outcome, making it a valuable piece of information for both clinical decision-making and research into the effects of chronic conditions on patient outcomes. The odds ratio exp(1.0695) provides a quantifiable measure of risk increase due to chronic renal failure, translating a complex statistical concept into actionable medical insights. By conveying how much a condition like CRN elevates the risk of a serious outcome, it underscores the importance of targeted medical intervention and informs a wide range of clinical and policy-related decisions.

2(j)

To calculate the estimated probability of death for an ICU patient with the characteristics age = 50, sex = female, chronic renal failure (CRN) = yes, and race = black using the logistic regression model and coefficients provided, we'll use the formula derived from the logistic model:

$$log(\frac{\pi}{1-\pi}) = -2.4715 + 0.0193(50) + 0.3956(1) + 1.0695(1) - 0.1253(1) + \beta_5(0) + 0.4094(1).$$

Now we have to solve for π. In my contribution to the discussion, I worked out how to solve such an equation for π:

$$\pi = \frac{1}{1 + \exp\left(-\left(-2.4715 + 0.0193(50) + 0.3956(1) + 1.0695(1) - 0.1253(1) + \beta_5(0) + 0.4094(1)\right)\right)}$$

so,

$$\pi = \frac{1}{1 + \exp\left(2.4715 - 0.0193(50) - 0.3956 - 1.0695 + 0.1253 - 0.4094\right)}$$

$$\pi \approx 0.5603789141607256.$$

Therefore, the patient has roughly a coin flip chance of dying, highlighting the extreme danger of chronic renal failure. I would assume that such patients are already put on dialysis and even with intervention, the probability of death is still 0.56.