Problem 6

```
{R}
install.packages("faraway")
library(faraway)
data(teengamb)
write.csv(teengamb, "C:/Users/dkkdk/Documents/grad/230/HW2/teengamb.csv", row.names=FALSE)
```

Part a

```
import pandas as pd
df = pd.read_csv('teengamb.csv')
print(df.head())
```

```
   sex  status  income  verbal  gamble
0    1      51     2.0       8     0.0
1    1      28     2.5       8     0.0
2    1      37     2.0       6     0.0
3    1      28     7.0       4     7.3
4    1      65     2.0       8    19.6
```

```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model = smf.ols(formula="gamble ~ sex + status + income + verbal", data=df).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 gamble   R-squared:                       0.527
Model:                            OLS   Adj. R-squared:                  0.482
Method:                 Least Squares   F-statistic:                     11.69
Date:                Sat, 15 Feb 2025   Prob (F-statistic):           1.81e-06
Time:                        16:10:34   Log-Likelihood:                -210.78
No. Observations:                  47   AIC:                             431.6
Df Residuals:                      42   BIC:                             440.8
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     22.5557     17.197      1.312      0.197     -12.149      57.260
sex          -22.1183      8.211     -2.694      0.010     -38.689      -5.548
status         0.0522      0.281      0.186      0.853      -0.515       0.620
income         4.9620      1.025      4.839      0.000       2.893       7.031
verbal        -2.9595      2.172     -1.362      0.180      -7.343       1.424
==============================================================================
Omnibus:                       31.143   Durbin-Watson:                   2.214
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              101.046
Skew:                           1.604   Prob(JB):                     1.14e-22
Kurtosis:                       9.427   Cond. No.                         264.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Part b

```
r_squared = model.rsquared
print(f"R-squared: {r_squared:.4f}")
adj_r_squared = model.rsquared_adj
print(f"Adjusted R-squared: {adj_r_squared:.4f}")
```

```
R-squared: 0.5267
Adjusted R-squared: 0.4816
```

Part c

```
# compute residuals
df["residuals"] = model.resid
# find max residual
max_residual_idx = df["residuals"].idxmax()
max_residual_value = df.loc[max_residual_idx, "residuals"]
print(f"max residual: {max_residual_value:.4f}")
```

```
max residual: 94.2522
```

Part d

```
residual_mean = df["residuals"].mean()
residual_median = df["residuals"].median()
print(f"mean of residuals: {residual_mean:.4f}")
print(f"median of residuals: {residual_median:.4f}")
```

```
mean of residuals: 0.0000
median of residuals: -1.4514
```

Part e

```
import numpy as np
cor_residuals_fitted = np.corrcoef(df["residuals"], model.fittedvalues)[0, 1]
print(f"Correlation between residuals and fitted values: {cor_residuals_fitted:.4f}")
```

```
Correlation between residuals and fitted values: 0.0000
```

Part f

```
cor_residuals_income = np.corrcoef(df["residuals"], df["income"])[0, 1]
print(f"Correlation between residuals and income: {cor_residuals_income:.4f}")
```

```
Correlation between residuals and income: 0.0000
```

Part g

```
# Define new observation
new_data = pd.DataFrame({
        "sex": [0],
        "status": [60],
        "income": [7],
        "verbal": [3]
})

# Get prediction results
pred_results = model.get_prediction(new_data)
# Extract confidence interval and prediction interval
pred_summary = pred_results.summary_frame(alpha=0.05)
# Print results
print(f"Predicted gambling expenditure: {pred_summary['mean'][0]:.4f}")
print(f"95% Confidence Interval: ({pred_summary['mean_ci_lower'][0]:.4f}, {pred_summary['mean_ci_upper'][0]:.4f})")
print(f"95% Prediction Interval: ({pred_summary['obs_ci_lower'][0]:.4f}, {pred_summary['obs_ci_upper'][0]:.4f})")
```

```
Predicted gambling expenditure: 51.5451
95% Confidence Interval: (29.4109, 73.6792)
95% Prediction Interval: (0.6851, 102.4050)
```

Part h

```
sex_coefficient = model.params["sex"]
print(f"Difference in predicted gambling expenditure (Male - Female): {sex_coefficient:.4f}")
```

```
Difference in predicted gambling expenditure (Male - Female): -22.1183
```

- This means that, holding all other predictors constant, a male is predicted to spend 10.31 pounds less on gambling compared to a female.
- A negative coefficient suggests that males tend to gamble less than females