```
%pip install ucimlrepo
%pip install -U ydata-profiling
from ucimlrepo import fetch_ucirepo
import pandas as pd
import numpy as np
from ydata_profiling import ProfileReport
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
from scipy.stats import wilcoxon, shapiro
import matplotlib.pyplot as plt
import seaborn as sns
import math
from scipy.stats import chi2_contingency, f_oneway
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import VarianceThreshold
from statsmodels.api import MNLogit, add_constant
from scipy.stats import chi2_contingency, f_oneway
from statsmodels.discrete.discrete_model import MNLogit
```

```
Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.11/dist-packages (0.0.7)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.11/dist-packages (from ucimlrepo) (2025.7.9)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.9.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0.0->ucimlr
Requirement already satisfied: ydata-profiling in /usr/local/lib/python3.11/dist-packages (4.16.1)
Requirement already satisfied: scipy<1.16,>=1.4.1 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (1.15.3)
Requirement already satisfied: pandas!=1.4.0,<3.0,>1.1 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (2.2.2)
Requirement already satisfied: matplotlib<=3.10,>=3.5 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (3.10.0)
Requirement already satisfied: pydantic>=2 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (2.11.7)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (6.0.2)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (3.1.6)
Requirement already satisfied: visions<0.8.2,>=0.7.5 in /usr/local/lib/python3.11/dist-packages (from visions[type_image_path]<0.8.2,>
Requirement already satisfied: numpy<2.2,>=1.16.0 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (2.0.2)
Requirement already satisfied: htmlmin==0.1.12 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (0.1.12)
Requirement already satisfied: phik<0.13,>=0.11.1 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (0.12.4)
Requirement already satisfied: requests<3,>=2.24.0 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (2.32.3)
Requirement already satisfied: tqdm<5,>=4.48.2 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (4.67.1)
Requirement already satisfied: seaborn<0.14,>=0.10.1 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (0.13.2)
Requirement already satisfied: multimethod<2,>=1.4 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (1.12)
Requirement already satisfied: statsmodels<1,>=0.13.2 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (0.14.5)
Requirement already satisfied: typeguard<5,>=3 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (4.4.4)
Requirement already satisfied: imagehash==4.3.1 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (4.3.1)
Requirement already satisfied: wordcloud>=1.9.3 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (1.9.4)
Requirement already satisfied: dacite>=1.8 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (1.9.2)
Requirement already satisfied: numba<=0.61,>=0.56.0 in /usr/local/lib/python3.11/dist-packages (from ydata-profiling) (0.60.0)
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.11/dist-packages (from imagehash==4.3.1->ydata-profiling) (1.8.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (from imagehash==4.3.1->ydata-profiling) (11.2.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2<3.2,>=2.11.1->ydata-profiling)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5->ydata-profili
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5->ydata-profiling)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5->ydata-profil
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5->ydata-profil
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5->ydata-profilin
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5->ydata-profili
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5->ydata-pro
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.11/dist-packages (from numba<=0.61,>=0.56.0->ydata
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas!=1.4.0,<3.0,>1.1->ydata-profiling)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas!=1.4.0,<3.0,>1.1->ydata-profilin
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.11/dist-packages (from phik<0.13,>=0.11.1->ydata-profiling) (1
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic>=2->ydata-profiling) (
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic>=2->ydata-profiling) (2
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.11/dist-packages (from pydantic>=2->ydata-profiling
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic>=2->ydata-profiling)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0->ydata-pr
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0->ydata-profiling) (3.
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0->ydata-profilin
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0->ydata-profilin
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.11/dist-packages (from statsmodels<1,>=0.13.2->ydata-profiling)
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.11/dist-packages (from visions<0.8.2,>=0.7.5->visions[type_imag
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.11/dist-packages (from visions<0.8.2,>=0.7.5->visions[type_imag
Requirement already satisfied: puremagic in /usr/local/lib/python3.11/dist-packages (from visions<0.8.2,>=0.7.5->visions[type_image_pa
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib<=3.10,>=3.5-
```

```
obesity = fetch_ucirepo(id=544)  # Load the Obesity dataset by ID
df = pd.concat([obesity.data.features, obesity.data.targets], axis=1)

# Mappiing feature variables
column_rename_map = {
    'FAVC': 'FAVC_FrequentHighCaloricFood',
    'FCVC': 'FCVC_VegetableConsumptionFreq',
    'NCP': 'NCP_NumberOfMainMeals',
    'CAEC': 'CAEC_BetweenMealSnacking',
    'CH2O': 'CH2O_DailyWaterIntake',
    'SCC': 'SCC_CalorieMonitoring',
    'FAF': 'FAF_PhysicalActivityFreq',
    'TUE': 'TUE_ScreenTimeHours',
    'CALC': 'CALC_AlcoholConsumption',
    'MTRANS': 'MTRANS_TransportationMode',
    'NObeyesdad': 'ObesityLevel_NObeyesdad'
}

df=df.rename(columns=column_rename_map)
profile = ProfileReport(df, title="YData Profiling Report")
profile.to_notebook_iframe()
```

```
Summarize dataset: 100%                                 90/90 [00:23<00:00,  2.02it/s, Completed]

  0%|          | 0/17 [00:00<?, ?it/s]
 24%|██▎       | 4/17 [00:00<00:00, 33.88it/s]
 47%|████▋     | 8/17 [00:00<00:00, 30.97it/s]
 71%|███████   | 12/17 [00:00<00:00, 27.33it/s]
100%|██████████| 17/17 [00:00<00:00, 31.13it/s]
Generate report structure: 100%                               1/1 [00:05<00:00,  5.39s/it]

Render HTML: 100%                                 1/1 [00:02<00:00,  2.22s/it]
```

YData Profiling Report   Overview Variables Interactions Correlations Missing values Sample Duplicate rows

Duplicate rows

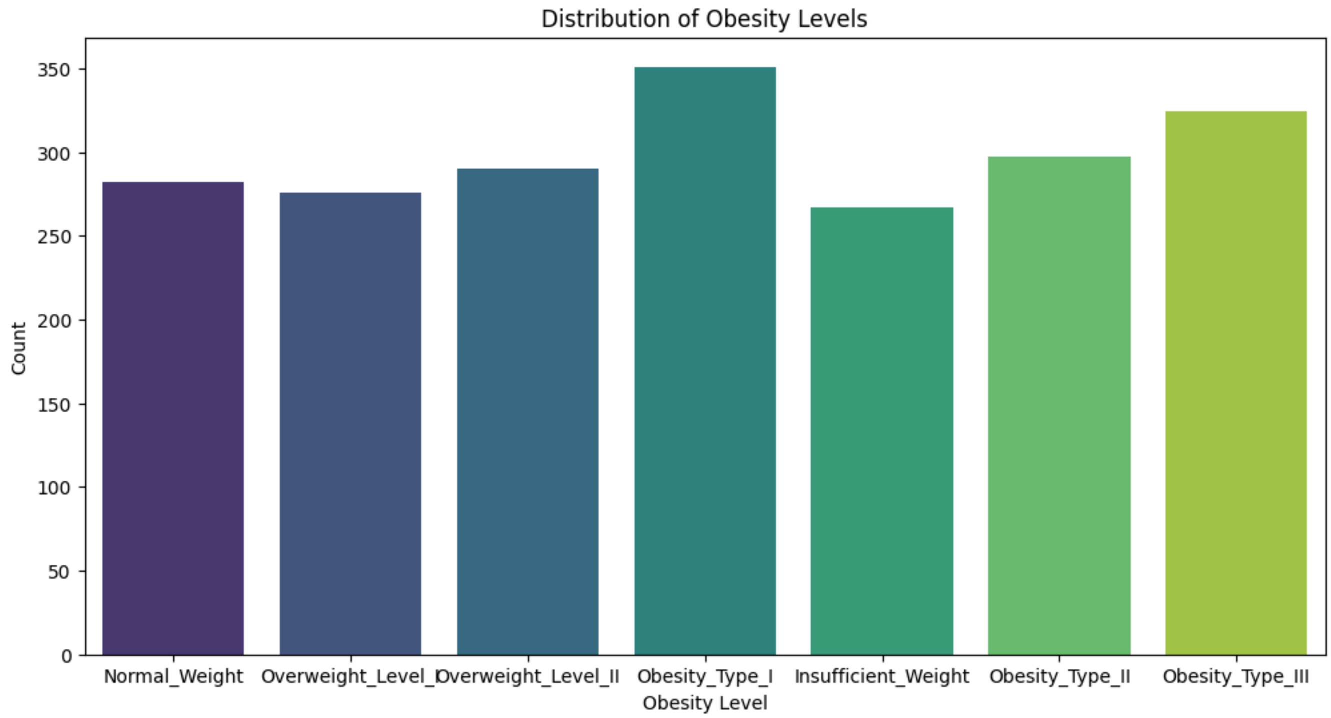
Most frequently occurring

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC_FrequentHighCaloricFood	FCVC_Vegetab
7	Male	21.0	1.62	70.0	no	yes	2.0
3	Female	21.0	1.52	42.0	no	yes	3.0
0	Female	16.0	1.66	58.0	no	no	2.0
2	Female	21.0	1.52	42.0	no	no	3.0
1	Female	18.0	1.62	55.0	yes	yes	2.0
4	Female	22.0	1.69	65.0	yes	yes	2.0
5	Female	25.0	1.57	55.0	no	yes	2.0
6	Male	18.0	1.72	53.0	yes	yes	2.0
8	Male	22.0	1.74	75.0	yes	yes	3.0

Report generated by YData.

```
# histogram of target variable-Obesity level

print(df['ObesityLevel_NObeyesdad'].value_counts())
plt.figure(figsize=(12, 6))
sns.countplot(x='ObesityLevel_NObeyesdad', data=df, palette='viridis')
plt.xlabel('Obesity Level')
plt.ylabel('Count')
plt.title('Distribution of Obesity Levels')
plt.show()
```

```
ObesityLevel_NObeyesdad
Obesity_Type_I         351
Obesity_Type_III       324
Obesity_Type_II        297
Overweight_Level_II    290
Normal_Weight          282
Overweight_Level_I     276
Insufficient_Weight    267
Name: count, dtype: int64
/tmp/ipython-input-157-2726866138.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.countplot(x='ObesityLevel_NObeyesdad', data=df, palette='viridis')
```




```
# Find the duplicate
duplicates = df[df.duplicated(keep=False)]

# Display the duplicates
#print(f" Total duplicate rows: {len(duplicates)}")
#print(duplicates)
#print(duplicates.count())
#duplicates.to_csv('duplicates_output.csv', index=False) # output as csv
#cleaned_df=df.drop_duplicates(keep='first')
#df=cleaned_df
#print(df)
#print(df.info())

df_str = df.apply(lambda x: x.str.strip().str.lower() if x.dtype == 'object' else x)
duplicates = df_str[df_str.duplicated(keep=False)]
print(f"True duplicates after cleaning: {len(duplicates)}")


#duplicates = df[df.duplicated(keep=False)]
#print(f"Total duplicate rows (all copies): {len(duplicates)}")

# Step 2: Remove duplicates (keep first occurrence)
cleaned_df = df.drop_duplicates(keep='first')
print(f"Rows after deduplication: {len(cleaned_df)}")

# Step 3: Overwrite original DataFrame (optional)
df = cleaned_df

# Verify
print(df.info())
```

```
True duplicates after cleaning: 33
Rows after deduplication: 2087
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 2087 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Gender                          2087 non-null   object 
 1   Age                             2087 non-null   float64
 2   Height                          2087 non-null   float64
 3   Weight                          2087 non-null   float64
 4   family_history_with_overweight  2087 non-null   object 
 5   FAVC_FrequentHighCaloricFood    2087 non-null   object 
 6   FCVC_VegetableConsumptionFreq   2087 non-null   float64
 7   NCP_NumberOfMainMeals           2087 non-null   float64
 8   CAEC_BetweenMealSnacking        2087 non-null   object 
 9   SMOKE                           2087 non-null   object 
 10  CH2O_DailyWaterIntake           2087 non-null   float64
 11  SCC_CalorieMonitoring           2087 non-null   object 
 12  FAF_PhysicalActivityFreq        2087 non-null   float64
 13  TUE_ScreenTimeHours             2087 non-null   float64
 14  CALC_AlcoholConsumption         2087 non-null   object 
 15  MTRANS_TransportationMode       2087 non-null   object 
 16  ObesityLevel_NObeyesdad         2087 non-null   object 
dtypes: float64(8), object(9)
memory usage: 293.5+ KB
None
```

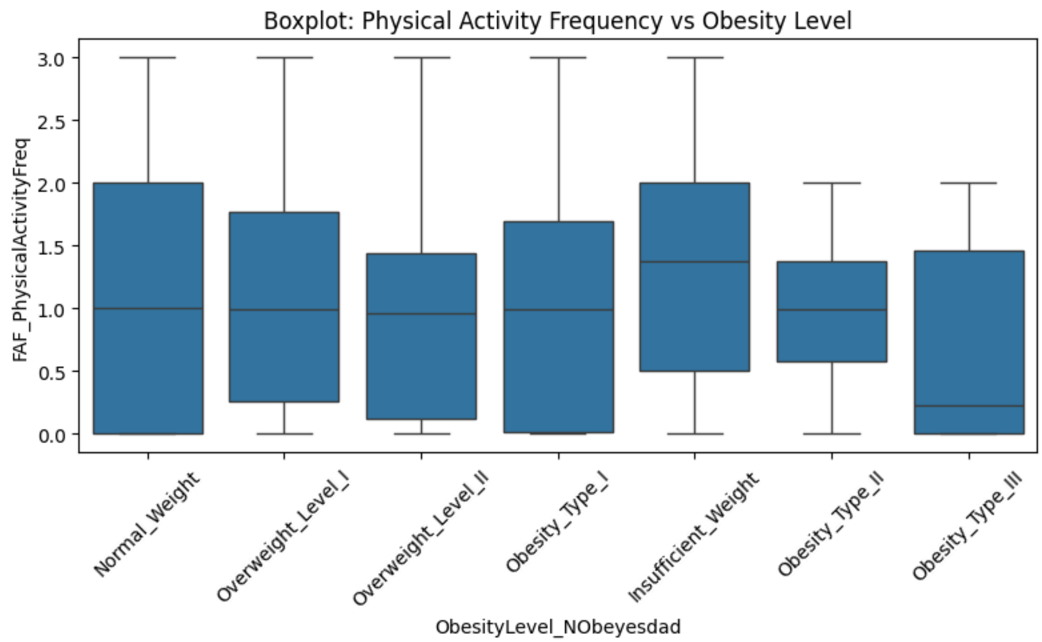
Double-click (or enter) to edit

```python
# Q1: How do dietary habits and physical activity influence obesity levels

# Step 1: Data Preparation & Encoding

df['CAEC_BetweenMealSnacking'] = df['CAEC_BetweenMealSnacking'].str.strip().str.lower()
df_encoded = df.copy()

# One-hot encode nominal categorical variables

one_hot_cols = ['CAEC_BetweenMealSnacking', 'MTRANS_TransportationMode', 'Gender']
df_encoded = pd.get_dummies(df_encoded, columns=one_hot_cols, drop_first=True,dtype=int)


# Binary encoding (yes/no, True/False)

binary_cols = df_encoded.columns[df_encoded.isin(['yes', 'no', 'Yes', 'No', 'True', 'False']).any()]
for col in binary_cols:
    df_encoded[col] = df_encoded[col].map({'yes': 1, 'no': 0, 'Yes': 1, 'No': 0, 'True': 1, 'False': 0})


# Round NCP up if needed (or leave as is)

df_encoded['NCP_NumberOfMainMeals'] = df_encoded['NCP_NumberOfMainMeals'].apply(math.ceil)
df_encoded['FCVC_VegetableConsumptionFreq'] = df_encoded['FCVC_VegetableConsumptionFreq'].apply(math.ceil)

print([col for col in df_encoded.columns if 'CAEC_BetweenMealSnacking' in col])

# Encode the target

label_encoder = LabelEncoder()
df_encoded['ObesityLevel_NObeyesdad_encoded'] = label_encoder.fit_transform(df_encoded['ObesityLevel_NObeyesdad'])

# Step 2: Define Feature Matrix & Target


X = df_encoded.drop(columns=['ObesityLevel_NObeyesdad', 'ObesityLevel_NObeyesdad_encoded'])
y = df_encoded['ObesityLevel_NObeyesdad_encoded']

# Optional: Remove low-variance features

selector = VarianceThreshold(threshold=0.01)

X = pd.DataFrame(selector.fit_transform(X),
                 columns=X.columns[selector.get_support()],
                 index=X.index)

# Step 3: Bivariate Analysis (Diet & Physical Activity vs Obesity Level)
```

```
# Combine X and y for analysis

df_bi = X.copy()
df_bi['Target'] = y

# Continuous features → ANOVA

print("\n ANOVA: Continuous vs Obesity Level")
cont_features = df_bi.select_dtypes(include='number').drop(columns='Target').columns
for col in cont_features:
    groups = [df_bi[df_bi['Target'] == cls][col].dropna().values for cls in sorted(df_bi['Target'].unique())]

    # Skip if any group is empty
    if any(len(group) == 0 for group in groups):
        print(f"{col}: Skipped (one or more groups are empty)")
        continue

    stat, p = f_oneway(*groups)
    print(f"{col}: p = {p:.6e}")  # use scientific notation for small p-values

# Categorical features → Chi-square

print("\n  Chi-square: Categorical vs Obesity Level")
cat_features = df_encoded.select_dtypes(include='uint8').columns
for col in cat_features:
    contingency = pd.crosstab(df_encoded[col], df_encoded['ObesityLevel_NObeyesdad'])
    chi2, p, _, _ = chi2_contingency(contingency)
    print(f"{col}: p = {p:.4e}")

# Boxplot example

plt.figure(figsize=(8, 5))
sns.boxplot(data=df_encoded, x='ObesityLevel_NObeyesdad', y='FAF_PhysicalActivityFreq')
plt.title("Boxplot: Physical Activity Frequency vs Obesity Level")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Step 4: Feature Importance (Random Forest)

#rf_model = RandomForestClassifier(random_state=42)
#rf_model.fit(X, y)

#importances = pd.Series(rf_model.feature_importances_, index=X.columns)
#importances_sorted = importances.sort_values(ascending=False)

#Plot top 10
#plt.figure(figsize=(10, 6))
#importances_sorted.head(10).plot(kind='barh', color='teal')
#plt.title("Top 10 Important Features (Random Forest)")
#plt.gca().invert_yaxis()
#plt.tight_layout()
#plt.show()


# Step 5: Multinomial Logistic Regression (focused on diet + physical activity)

# Pick relevant features

features_of_interest = [
    'FAVC_FrequentHighCaloricFood',
    'FCVC_VegetableConsumptionFreq',
    'NCP_NumberOfMainMeals',
    'CAEC_BetweenMealSnacking_always',
    'CAEC_BetweenMealSnacking_frequently',
    'FAF_PhysicalActivityFreq'
]
features_available = [f for f in features_of_interest if f  in X.columns]

X_subset = X[features_available]

# Add constant
X_subset_const = add_constant(X_subset)
y= y.loc[X_subset_const.index]

# Fit multinomial logistic regression
model = MNLogit(y, X_subset_const)
```

```
result = model.fit(disp=0)

# Show results
print("\n Multinomial Logistic Regression Summary:")
print(result.summary())
```

```
/tmp/ipython-input-147-184815850.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-
  df['CAEC_BetweenMealSnacking'] = df['CAEC_BetweenMealSnacking'].str.strip().str.lower()
['CAEC_BetweenMealSnacking_frequently', 'CAEC_BetweenMealSnacking_no', 'CAEC_BetweenMealSnacking_sometimes']

 ANOVA: Continuous vs Obesity Level
Age: p = 3.246862e-86
Weight: p = 0.000000e+00
family_history_with_overweight: p = 1.468826e-154
FAVC_FrequentHighCaloricFood: p = 6.035625e-50
FCVC_VegetableConsumptionFreq: p = 1.690906e-88
NCP_NumberOfMainMeals: p = 2.187292e-24
SMOKE: p = 1.615012e-05
CH2O_DailyWaterIntake: p = 4.297247e-17
SCC_CalorieMonitoring: p = 5.345447e-26
FAF_PhysicalActivityFreq: p = 1.155420e-20
TUE_ScreenTimeHours: p = 1.772380e-08
CAEC_BetweenMealSnacking_frequently: p = 2.960702e-118
CAEC_BetweenMealSnacking_no: p = 8.389095e-15
CAEC_BetweenMealSnacking_sometimes: p = 2.837371e-126
MTRANS_TransportationMode_Public_Transportation: p = 8.737212e-31
MTRANS_TransportationMode_Walking: p = 1.896946e-19
Gender_Male: p = 6.852093e-167

  Chi-square: Categorical vs Obesity Level
```




```
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607: ConvergenceWarning: Maximum Likelihood optimization failed to c
  warnings.warn("Maximum Likelihood optimization failed to "

 Multinomial Logistic Regression Summary:
                          MNLogit Regression Results
==========================================================================================
Dep. Variable:     ObesityLevel_NObeyesdad_encoded   No. Observations:                 2087
Model:                                     MNLogit   Df Residuals:                     2051
Method:                                        MLE   Df Model:                           30
Date:                             Sat, 12 Jul 2025   Pseudo R-squ.:                  0.1747
Time:                                     22:22:32   Log-Likelihood:                -3344.6
converged:                                   False   LL-Null:                       -4052.5
Covariance Type:                         nonrobust   LLR p-value:                 3.419e-279
=========================================================================================================
  ObesityLevel_NObeyesdad_encoded=1       coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
const                                   7.3957      0.716     10.332      0.000       5.993       8.799
FAVC_FrequentHighCaloricFood           -0.8893      0.230     -3.865      0.000      -1.340      -0.438
FCVC_VegetableConsumptionFreq          -1.4831      0.180     -8.219      0.000      -1.837      -1.129
NCP_NumberOfMainMeals                  -0.9286      0.136     -6.823      0.000      -1.195      -0.662
CAEC_BetweenMealSnacking_frequently    -0.7405      0.201     -3.683      0.000      -1.135      -0.346
FAF_PhysicalActivityFreq                0.1047      0.106      0.989      0.323      -0.103       0.312
---------------------------------------------------------------------------------------------------------
  ObesityLevel_NObeyesdad_encoded=2       coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
const                                   5.7824      0.767      7.536      0.000       4.279       7.286
FAVC_FrequentHighCaloricFood            1.3978      0.370      3.777      0.000       0.672       2.123
FCVC_VegetableConsumptionFreq          -1.0391      0.186     -5.594      0.000      -1.403      -0.675
```

```
NCP_NumberOfMainMeals                  -1.1329      0.141     -8.061      0.000      -1.408      -0.857
CAEC_BetweenMealSnacking_frequently    -3.8970      0.445     -8.766      0.000      -4.768      -3.026
FAF_PhysicalActivityFreq               -0.2361      0.111     -2.134      0.033      -0.453      -0.019
-------------------------------------------------------------------------------------------------------
  ObesityLevel_NObeyesdad_encoded=3       coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------
const                                  -0.4343      0.892     -0.487      0.626      -2.183       1.314
FAVC_FrequentHighCaloricFood            1.7047      0.434      3.928      0.000       0.854       2.555
FCVC_VegetableConsumptionFreq           0.5914      0.213      2.775      0.006       0.174       1.009
NCP_NumberOfMainMeals                  -0.5161      0.156     -3.310      0.001      -0.822      -0.210
CAEC_BetweenMealSnacking_frequently    -5.4470      1.013     -5.377      0.000      -7.433      -3.461
FAF_PhysicalActivityFreq               -0.4785      0.118     -4.055      0.000      -0.710      -0.247
-------------------------------------------------------------------------------------------------------
  ObesityLevel_NObeyesdad_encoded=4       coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------
const                                 -67.1456   1.29e+04     -0.005      0.996   -2.53e+04    2.52e+04
FAVC_FrequentHighCaloricFood            3.8118      1.028      3.707      0.000       1.796       5.827
FCVC_VegetableConsumptionFreq          21.9115   4291.071      0.005      0.996   -8388.434    8432.257
NCP_NumberOfMainMeals                   0.0200      0.181      0.110      0.912      -0.335       0.375
CAEC_BetweenMealSnacking_frequently    -5.6453      1.019     -5.541      0.000      -7.642      -3.648
FAF_PhysicalActivityFreq               -1.2090      0.131     -9.252      0.000      -1.465      -0.953
-------------------------------------------------------------------------------------------------------
  ObesityLevel_NObeyesdad_encoded=5       coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------
const                                   4.9053      0.755      6.497      0.000       3.426       6.385
FAVC_FrequentHighCaloricFood            0.4535      0.297      1.529      0.126      -0.128       1.035
FCVC_VegetableConsumptionFreq          -0.7843      0.189     -4.141      0.000      -1.155      -0.413
NCP_NumberOfMainMeals                  -0.8290      0.144     -5.740      0.000      -1.112      -0.546
CAEC_BetweenMealSnacking_frequently    -2.7252      0.311     -8.757      0.000      -3.335      -2.115
FAF_PhysicalActivityFreq               -0.1918      0.113     -1.702      0.089      -0.413       0.029
-------------------------------------------------------------------------------------------------------
  ObesityLevel_NObeyesdad_encoded=6       coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------
const                                   6.8002      0.732      9.292      0.000       5.366       8.235
FAVC_FrequentHighCaloricFood           -0.9586      0.242     -3.957      0.000      -1.433      -0.484
FCVC_VegetableConsumptionFreq          -0.8669      0.190     -4.570      0.000      -1.239      -0.495
NCP_NumberOfMainMeals                  -0.9283      0.143     -6.513      0.000      -1.208      -0.649
CAEC_BetweenMealSnacking_frequently    -2.8261      0.302     -9.368      0.000      -3.417      -2.235
FAF_PhysicalActivityFreq               -0.3562      0.113     -3.149      0.002      -0.578      -0.134
=======================================================================================================
```

```python
# Q2: Does family history of overweight influence the relationship between lifestyle choices (diet, exercise) and obesity


# 1. Data Encoding & Preprocessing

df_encoded = df.copy()

# Encode yes/no or True/False columns

binary_cols = df_encoded.columns[df_encoded.isin(['yes', 'no', 'Yes', 'No', 'True', 'False']).any()]
for col in binary_cols:
    df_encoded[col] = df_encoded[col].map({'yes': 1, 'no': 0, 'Yes': 1, 'No': 0, 'True': 1, 'False': 0})

# Round NCP up

df_encoded['NCP_NumberOfMainMeals'] = df_encoded['NCP_NumberOfMainMeals'].apply(math.ceil)

# One-hot encode nominal variables

one_hot_cols = ['CAEC_BetweenMealSnacking', 'MTRANS_TransportationMode', 'Gender']
df_encoded = pd.get_dummies(df_encoded, columns=one_hot_cols, drop_first=True)

# Encode target

label_encoder = LabelEncoder()
df_encoded['ObesityLevel_NObeyesdad_encoded'] = label_encoder.fit_transform(df_encoded['ObesityLevel_NObeyesdad'])
#print(df_encoded)


# 2. Create Interaction Terms

# Check FamilyHistory (family overweight) moderates effects of diet/activity
# Create main variables and their interactions
df_encoded['FAVC_FamilyHistory'] = df_encoded['FAVC_FrequentHighCaloricFood'] * df_encoded['family_history_with_overweight']
df_encoded['FAF_FamilyHistory'] = df_encoded['FAF_PhysicalActivityFreq'] * df_encoded['family_history_with_overweight']



# 3. Multinomial Logistic Regression: Interaction Effects
```

```
X_cols = ['FAVC_FrequentHighCaloricFood',   'FAF_PhysicalActivityFreq',  'family_history_with_overweight',
          'FAVC_FamilyHistory', 'FAF_FamilyHistory']
X = df_encoded[X_cols]
y = df_encoded['ObesityLevel_NObeyesdad_encoded']

# Add constant
X_const = add_constant(X)

# Fit model
model = MNLogit(y, X_const)
result = model.fit(disp=0)
print("Multinomial Logistic Regression with interaction terms:")
print(result.summary())

# 4. Subgroup Analysis by Family History

print("\n Subgroup Analysis: With vs Without Family History\n")

for group, label in zip([1, 0], ['With FH', 'Without FH']):
    df_group = df_encoded[df_encoded['family_history_with_overweight'] == group]
    X_group = df_group[['FAVC_FrequentHighCaloricFood', 'FCVC_VegetableConsumptionFreq', 'NCP_NumberOfMainMeals', 'FAF_PhysicalActivityFreq'
    y_group = df_group['ObesityLevel_NObeyesdad_encoded']
    X_group_const = add_constant(X_group)

    model_group = MNLogit(y_group, X_group_const)
    result_group = model_group.fit(disp=0)
    print(f"Subgroup: {label}")
    print(result_group.summary())
    print("\n" + "="*80 + "\n")


# Visualization of Interaction

sns.lmplot(data=df_encoded, x='FAF_PhysicalActivityFreq', y='NCP_NumberOfMainMeals', hue='family_history_with_overweight',
           palette='Set1', fit_reg=False)
plt.title("Physical Activity vs Meals by Family History")
plt.xlabel("Physical Activity (FAF)")
plt.ylabel("Number of Meals (NCP)")
plt.show()
```