In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
# Library to help with statistical analysis
import scipy.stats as stats
#rom statsmode_s.rormula.ap1 imbort Ols
# -or n-wav ANOVA
from statsmodels.stats.anova import _get_covariance, anova_lm # For n-
```

In [2]:

```
# uncomment the below code cell to mount your google drive
# from google.colab import drive
data = pd.read_csv('Downloads/commontest2.csv')
# copying data to another variable to avoid any changes to original da
df = data.copy()
```

In [3]:

```
df.head()
```

Out[3]:

	diet	preweight	weight6weeks	age
0	B	60	60.0	45
1	B	103	103.0	38
2	A	58	54.2	31
3	A	60	54.0	18
4	A	64	63.3	35

In [4]:

```
df.shape
```

Out[4]:

```
(78, 4)
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 78 entries, 0 to 77
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   diet          78 non-null     object 
 1   preweight     78 non-null     int64  
 2   weight6weeks  78 non-null     float64
 3   age           78 non-null     int64  
dtypes: float64(1), int64(2), object(1)
memory usage: 2.6+ KB
```

In [7]:
```
# Calculate the weight loss for each participant
df['weight_loss'] = df['preweight'] - df['weight6weeks']

# Define age categories using a helper function
def categorize_age(age):
    if 18 <= age < 25:
        return "18-25"
    elif 25 <= age < 40:
        return "25-40"
    else:
        return "40+"

# Apply the categorization function to the 'age' column
df['age_category'] = df['age'].apply(categorize_age)

# Display the first few rows of the DataFrame
print(df.head())
```

```
  diet  preweight  weight6weeks  age  weight_loss age_category
0    B         60          60.0   45          0.0          40+
1    B        103         103.0   38          0.0        25-40
2    A         58          54.2   31          3.8        25-40
3    A         60          54.0   18          6.0        18-25
4    A         64          63.3   35          0.7        25-40
```

In [8]:
```
df.age_category.value_counts()
```

Out[8]:
```
25-40    47
40+      19
18-25    12
Name: age_category, dtype: int64
```

In [10]:
```
# create separate variables to store the weightlosses with respect to
weightloss_Elderly = df[df['age_category']=='40+']['weight_loss']
weightloss_Middle_aged = df [df[ 'age_category']=='25-40' ]['weight_lo
weightloss_Young = df [df ['age_category']=='18-25']['weight_loss']
```

In [12]:
```
# import the required function
from scipy.stats import f_oneway
# find the p-value
test_stat, p_value = f_oneway (weightloss_Elderly, weightloss_Middle_a
print('The p-value is ', p_value)
```

```
The p-value is  0.05544168556142372
```

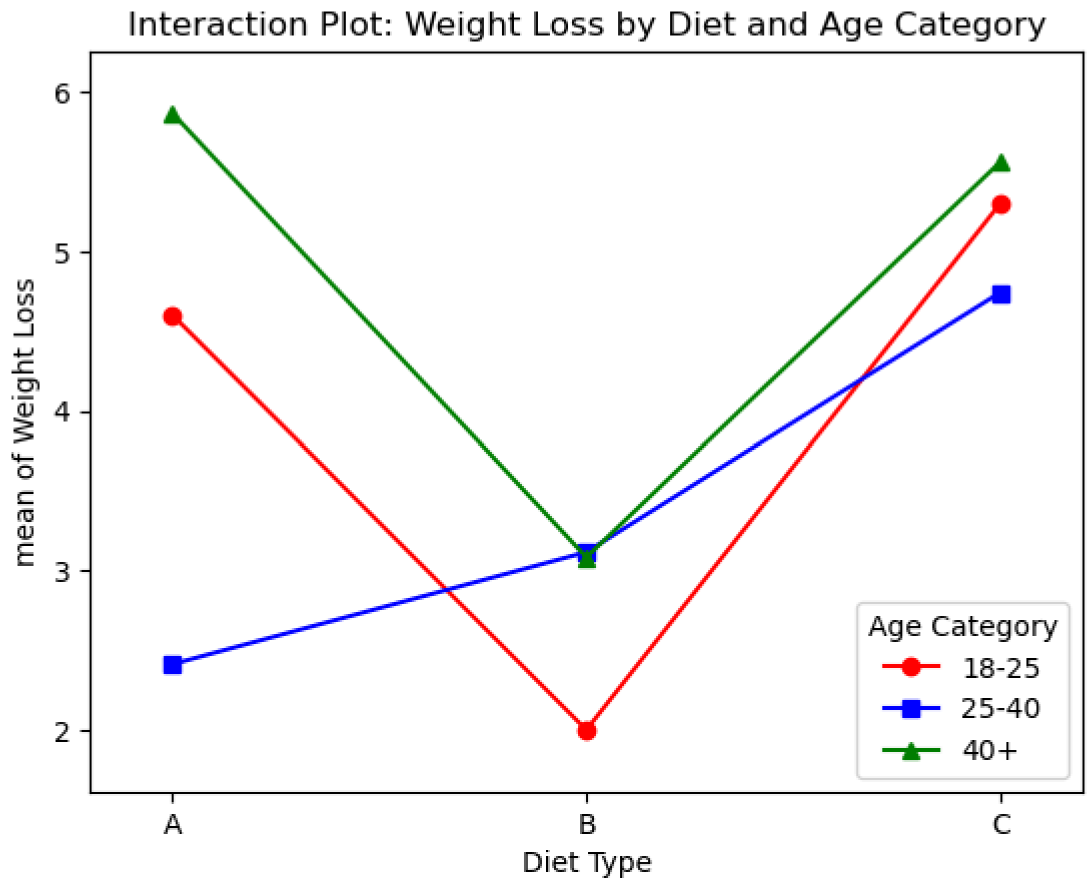
In [13]:

```
from statsmodels.graphics.factorplots import interaction_plot
import matplotlib.pyplot as plt

# Ensure the necessary columns are available in the DataFrame
# 'diet', 'age_category', and 'weight_loss'

# Create the interaction plot
fig = interaction_plot(
    x=np.array(df['diet']),                  # Independent variable 1
    trace=np.array(df['age_category']),      # Independent variable 2 (g
    response=np.array(df['weight_loss']), # Dependent variable
    colors=['red', 'blue', 'green'],         # Colors for different age
    markers=['o', 's', '^'],                 # Markers for the plot
    xlabel='Diet Type',                      # X-axis label
    ylabel='Weight Loss',                    # Y-axis label
    legendtitle='Age Category'               # Legend title
)

# Display the plot
plt.title('Interaction Plot: Weight Loss by Diet and Age Category')
plt.show()
```



In [20]:

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

# Define the formula for the ANOVA
formula = 'weight_loss ~ C(diet) + C(age_category) + C(diet):C(age_cat

# Fit the model
model = ols(formula, df).fit()

# Perform ANOVA
aov_table = anova_lm(model)

# Display the ANOVA table
print(aov_table)
```

```
                          df      sum_sq    mean_sq         F    PR
(>F)
C(diet)                  2.0   71.093689  35.546845  6.399140  0.00
2822
C(age_category)          2.0   17.498000   8.749000  1.574994  0.21
4359
C(diet):C(age_category)  4.0   29.390330   7.347582  1.322711  0.27
0226
Residual                69.0  383.290930   5.554941       NaN
NaN
```

In [ ]: