Analysis of Economic Policy using Consumer Price Index

MA-541 FINAL PROJECT : GROUP 1

Team Members: Akanksha Wagh, Nihar Dugade, Varad Hattekar, Vidisha Parmar

Part 01: Exploratory Data Analysis

In [80]:
```
#Loading dataset
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_excel('Analysis.xlsx')
```

In [81]:
```
df.head()
```

Out[81]:

	Index	Expenditure category	Class	Subclass	Subclass div	Relative\nimportance\nDec.\n2023	Unadjusted percent change Jan.\n2023-\nJan.\n2024
0	1	Food	Food	Food at home	Cereals and bakery Products	13.555	2.6
1	1	Food at home	Food	Food at home	Cereals and bakery Products	8.167	1.2
2	1	Cereals and bakery products	Food	Food at home	Cereals and bakery Products	1.066	1.5
3	1	Cereals and cereal products	Food	Food at home	Cereals and bakery Products	0.314	-0.6
4	1	Flour and prepared flour mixes	Food	Food at home	Cereals and bakery Products	0.051	1.0

In [82]:
```
# Renaming columns for easier access
new_column_names = {
    'Index': 'index',
    'Expenditure category': 'expenditure_category',
    'Class': 'class',
```

```
    'Subclass': 'subclass',
    'Subclass div': 'subclass_division',
    'Relative\nimportance\nDec.\n2023': 'relative_importance_dec_2023',
    'Unadjusted percent change Jan.\n2023-\nJan.\n2024': 'unadjusted_pct_change_jan_20
    'Unadjusted percent change Dec.\n2023-\nJan.\n2024': 'unadjusted_pct_change_dec_20
    'Seasonally adjusted percent change Oct.\n2023-\nNov.\n2023': 'seasonally_adj_pct_
    'Seasonally adjusted percent change Nov.\n2023-\nDec.\n2023': 'seasonally_adj_pct_
    'Seasonally adjusted percent change Dec.\n2023-\nJan.\n2024': 'seasonally_adj_pct_
}
df.rename(columns=new_column_names, inplace=True)
```

In [83]:
```
df.head()
```

Out[83]:

	index	expenditure_category	class	subclass	subclass_division	relative_importance_dec_2023	unadj
0	1	Food	Food	Food at home	Cereals and bakery Products	13.555	
1	1	Food at home	Food	Food at home	Cereals and bakery Products	8.167	
2	1	Cereals and bakery products	Food	Food at home	Cereals and bakery Products	1.066	
3	1	Cereals and cereal products	Food	Food at home	Cereals and bakery Products	0.314	
4	1	Flour and prepared flour mixes	Food	Food at home	Cereals and bakery Products	0.051	

In [84]:
```
# Calculating summary statistics for numerical columns
summary_stats = df.describe()

# Display the summary statistics as a nicely formatted HTML table
# This can be useful for IPython display environments such as Jupyter Notebooks
from IPython.display import display, HTML

# Convert DataFrame to HTML; other styling can be added via CSS within the HTML string
summary_stats_html = summary_stats.to_html()

# Display the HTML table in Jupyter Notebook
display(HTML(summary_stats_html))
```

	index	relative_importance_dec_2023	unadjusted_pct_change_jan_2023_2024	unadjusted_pct_char
count	265.0	265.000000	265.000000	
mean	1.0	1.856894	1.076604	
std	0.0	7.341149	5.646397	
min	1.0	0.008000	-28.600000	
25%	1.0	0.122000	-1.100000	
50%	1.0	0.294000	1.400000	
75%	1.0	0.854000	4.200000	
max	1.0	79.790000	29.000000	

Data Description:

Expenditure Category (String): Category includes consumer goods and services in Food, Energy, Education, Transportation sector and more.

Relative Importance of December 2023: The relative importance metric in our dataset provides a weight for each category, indicating its significance in the average consumer's budget. This weighing helps to underscore the impact of price changes in more significant areas of spending. Understanding the relative importance of different categories or items in the CPI helps analysts and policymakers interpret the index and assess the impact of price changes on overall inflation and consumer purchasing power.

Class (String): All the items from expenditure category are divided into 3 major categories-Food, Energy and Other. Subclass and Subclass Division (String): The item is further divided into subcategories-Healthcare, Housing, Electricity, Transportation, Apparel, education and more.

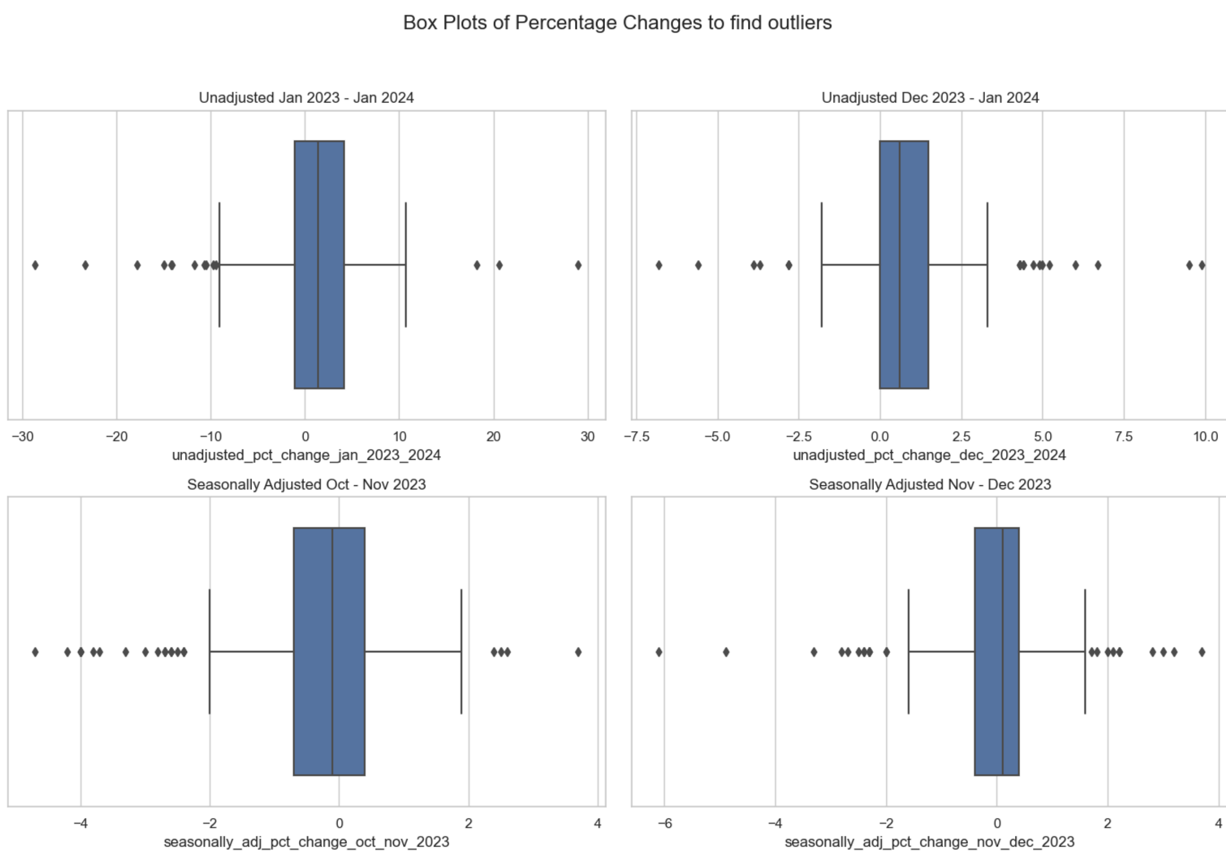
Unadjusted percent change (Float): This metric captures the raw change in prices over a specified period, reflecting the actual fluctuations in costs without adjustments for external influences. It represents the direct price movements without modifications for seasonal variations, changes in the composition of the consumer goods basket, or other potential distortions. The data for the unadjusted percentage change was collected from January 2023 to January 2024, providing a straightforward view of price dynamics during this interval.

Adjusted percent change (Float): This measure accounts for various factors that can influence price changes, including seasonal variations, shifts in product quality, substitution effects (where consumers switch between products in response to price changes), and other relevant variables. The seasonally adjusted percentage change offers a clearer representation of underlying price trends by filtering out or mitigating the impact of these elements. Data for the seasonally adjusted percentage change was meticulously compiled from October 2023 to January 2024, ensuring an accurate depiction of the economic landscape during this period.

In [85]:
```
# Handling missing values by filling with the mean of each column
for col in df.columns[6:]:
```

```
    df[col].fillna(df[col].mean(), inplace=True)
```

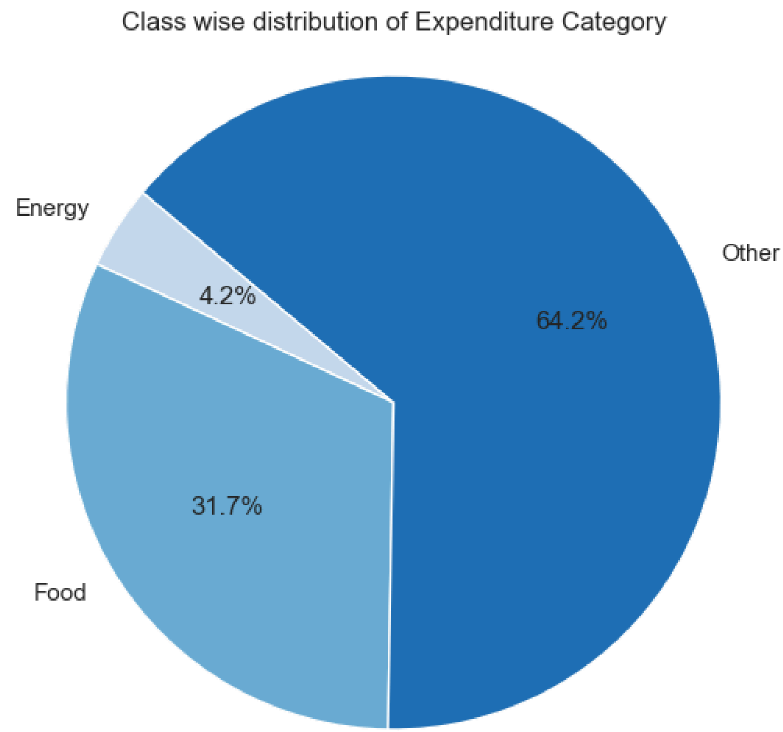
In [86]:
```
# Visualizing outliers using box plots
sns.set(style="whitegrid")
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(14, 10))
fig.suptitle('Box Plots of Percentage Changes to find outliers', fontsize=16)

sns.boxplot(ax=axes[0, 0], data=df, x='unadjusted_pct_change_jan_2023_2024')
axes[0, 0].set_title('Unadjusted Jan 2023 - Jan 2024')

sns.boxplot(ax=axes[0, 1], data=df, x='unadjusted_pct_change_dec_2023_2024')
axes[0, 1].set_title('Unadjusted Dec 2023 - Jan 2024')

sns.boxplot(ax=axes[1, 0], data=df, x='seasonally_adj_pct_change_oct_nov_2023')
axes[1, 0].set_title('Seasonally Adjusted Oct - Nov 2023')

sns.boxplot(ax=axes[1, 1], data=df, x='seasonally_adj_pct_change_nov_dec_2023')
axes[1, 1].set_title('Seasonally Adjusted Nov - Dec 2023')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])  # Adjust layout to make room for the title
plt.show()
```



These plots highlight the central tendency and variability within each period, with most data points clustering near the median, indicating stable changes over time. Notably, outliers are present in each period, suggesting occasional significant deviations from typical price changes. These outliers, especially prominent in the unadjusted data, underscore potential volatility in the market or effects of external shocks on pricing, which the seasonal adjustments aim to mitigate, providing a clearer view of underlying economic trends. We will not eliminate these outliers, as there are not many points which lie outside of the clusters, and these outliers are dependent on each other since they might belong to same consumer goods, commodities, or services.

Data Visualization

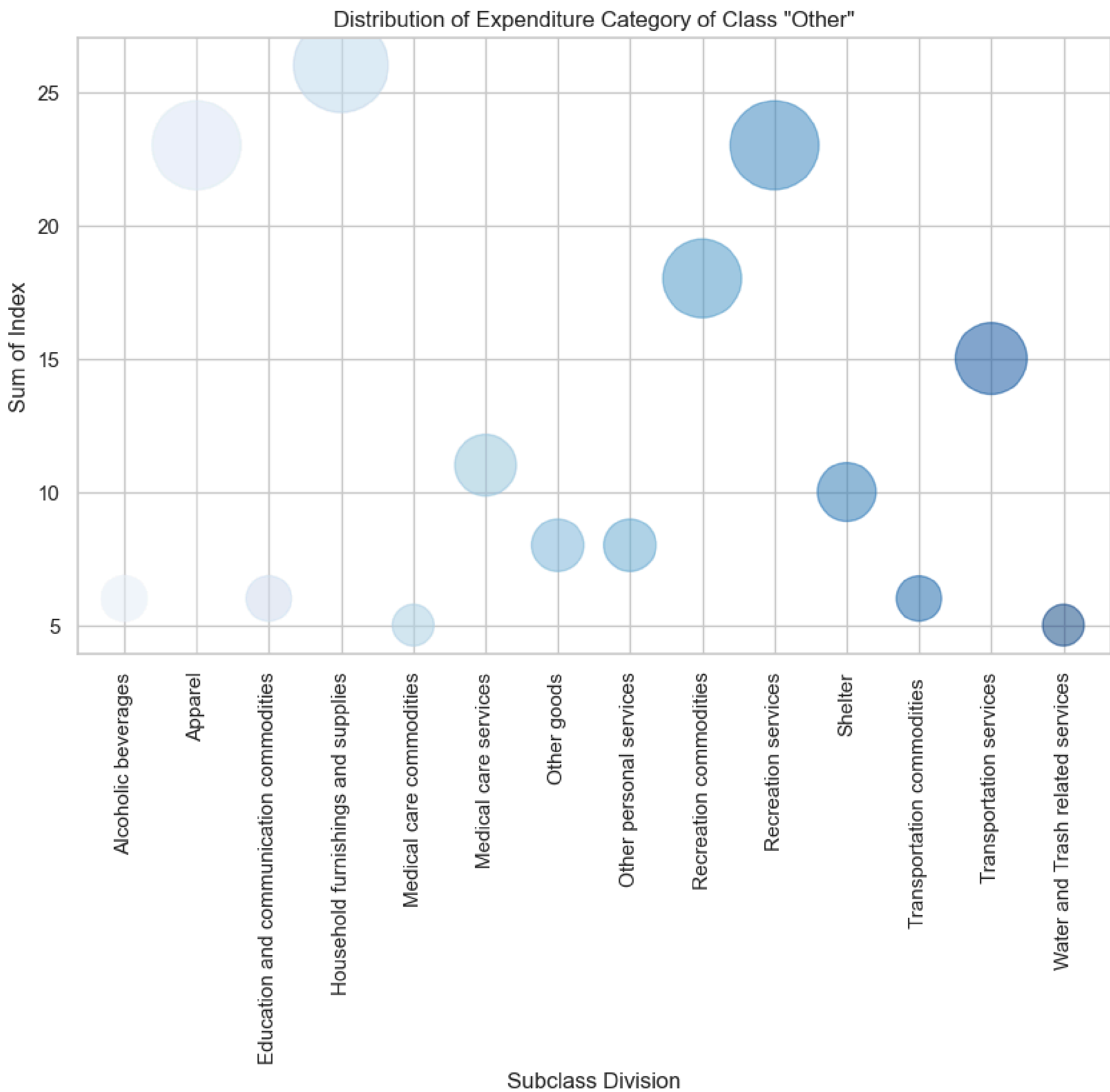
In [87]:
```
sum_index = df.groupby('class')['index'].sum()

# Define a color palette with shades of blue
colors = sns.color_palette('Blues', len(sum_index))

# Plotting the pie chart with specified colors
plt.figure(figsize=(10, 6))
plt.pie(sum_index, labels=sum_index.index, startangle=140, colors=colors, autopct='%1.
plt.title('Class wise distribution of Expenditure Category')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



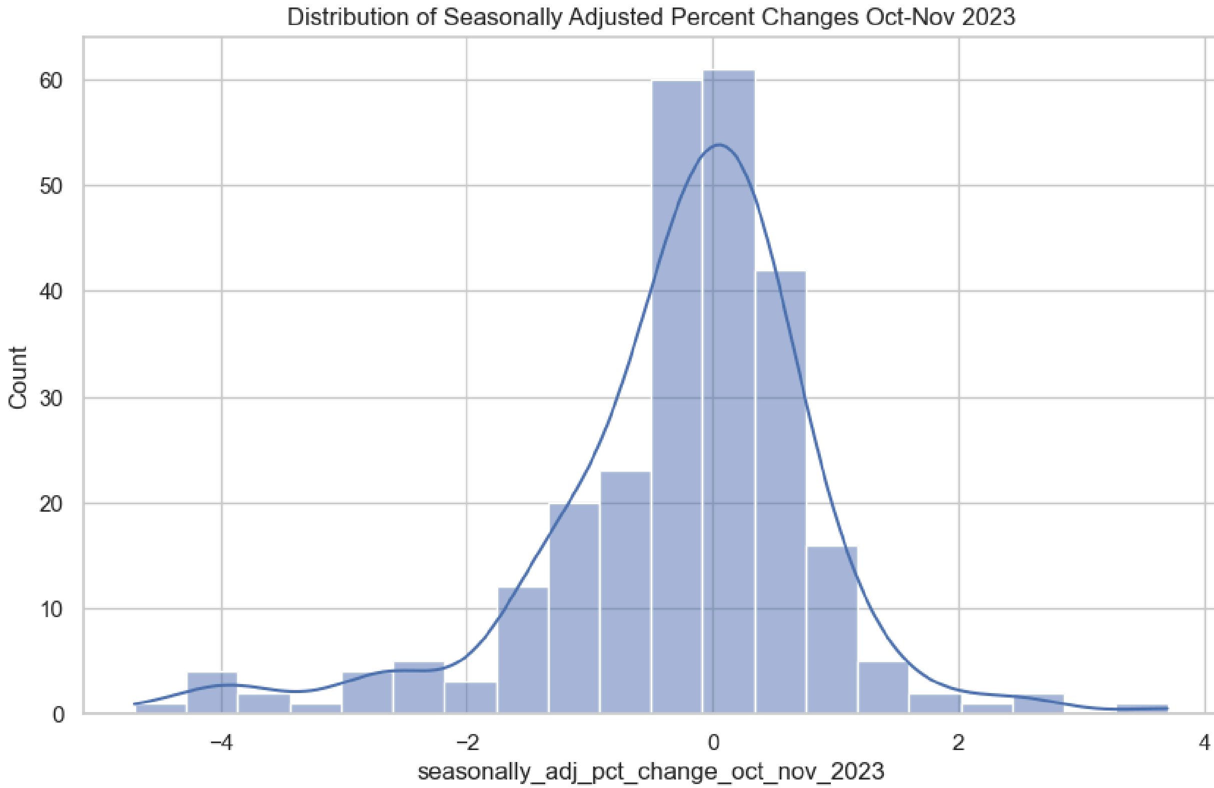
Since 'Other' category have maximum data points so following is distribution of sub-categories in 'Other- Expenditure Category.'

In [88]:
```
other_class = df[df['class'] == 'Other']  # Filter the DataFrame for 'Other' class
sum_index = other_class.groupby('subclass_division')['index'].sum()  # Group by 'subcl

# Define the sizes of the bubbles based on the sum of 'index'
sizes = sum_index.values

# Define a color palette with shades of blue
colors = sns.color_palette('Blues', len(sum_index))

# Plotting the bubble chart
plt.figure(figsize=(10, 6))
plt.scatter(sum_index.index, sum_index, s=sizes*100, c=colors, alpha=0.5)
plt.title('Distribution of Expenditure Category of Class "Other"')
plt.xlabel('Subclass Division')
plt.ylabel('Sum of Index')
```

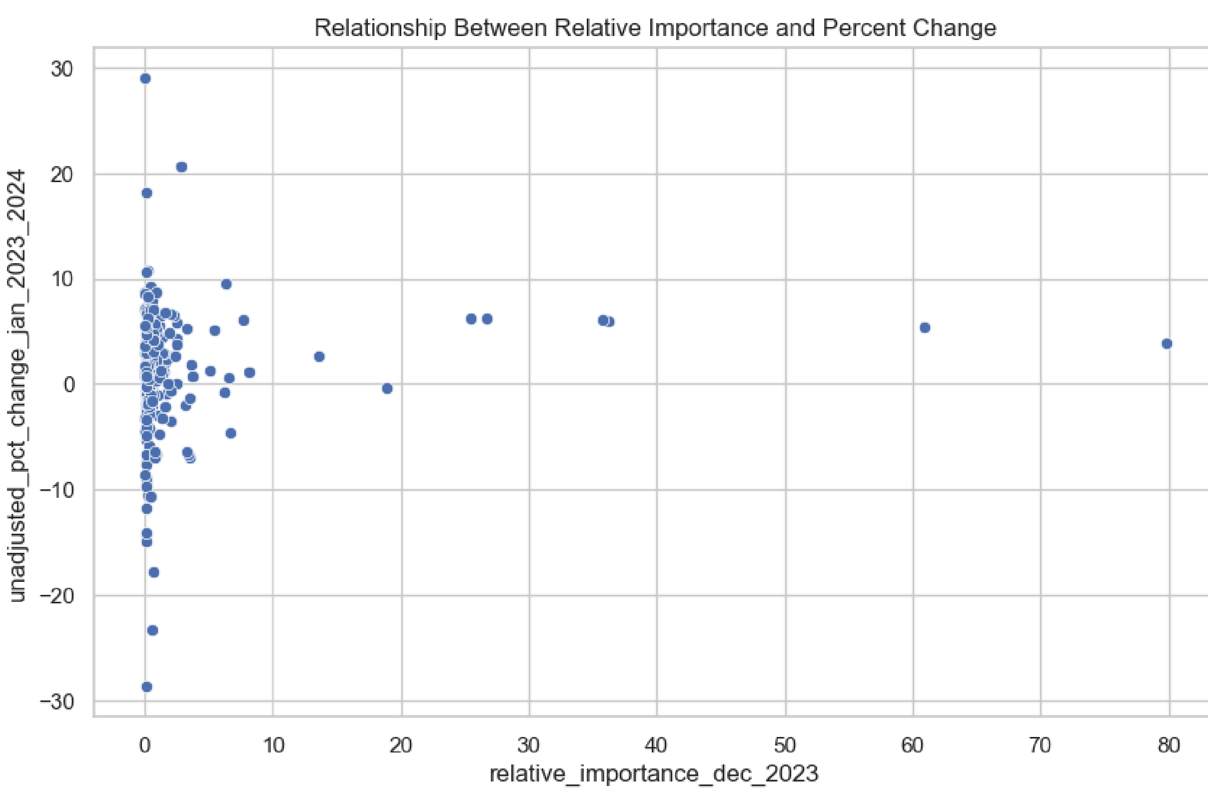
```
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



In [89]:
```
# Histograms of Seasonal Adjustments
plt.figure(figsize=(10, 6))
sns.histplot(df['seasonally_adj_pct_change_oct_nov_2023'], bins=20, kde=True)
plt.title('Distribution of Seasonally Adjusted Percent Changes Oct-Nov 2023')
plt.show()
```

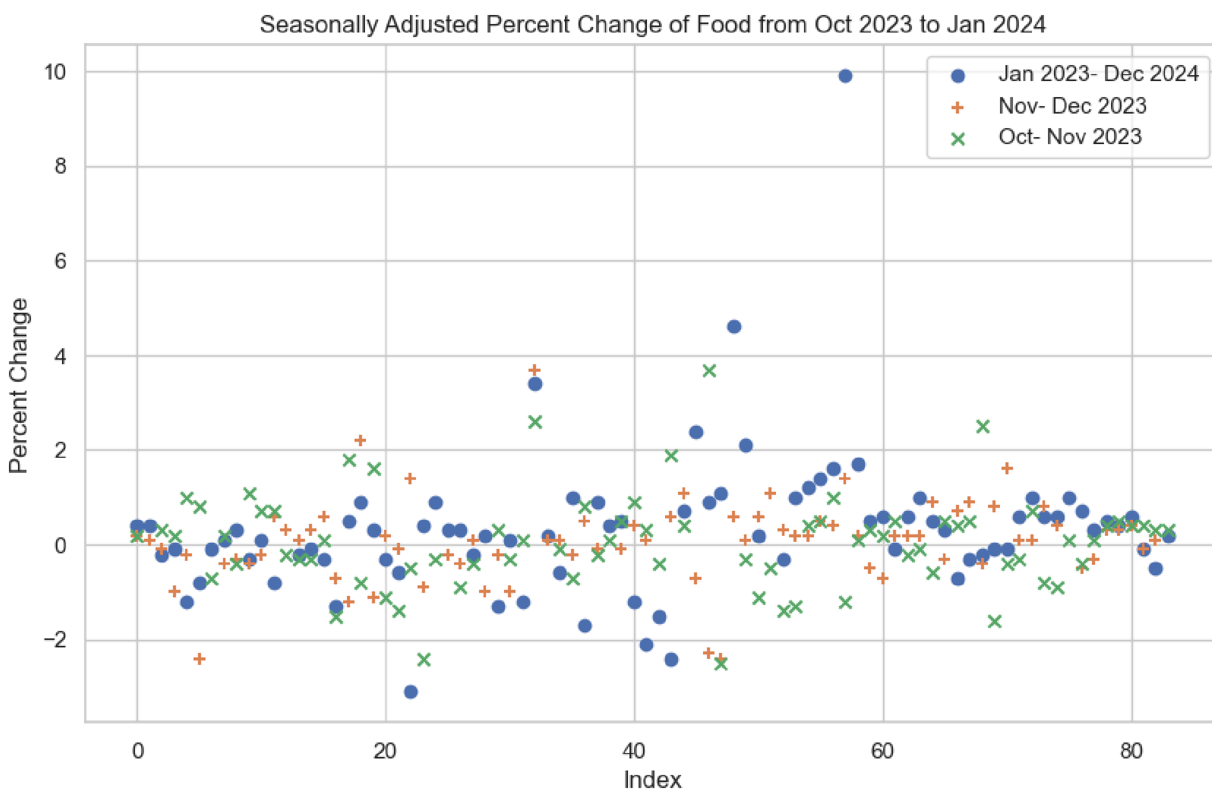


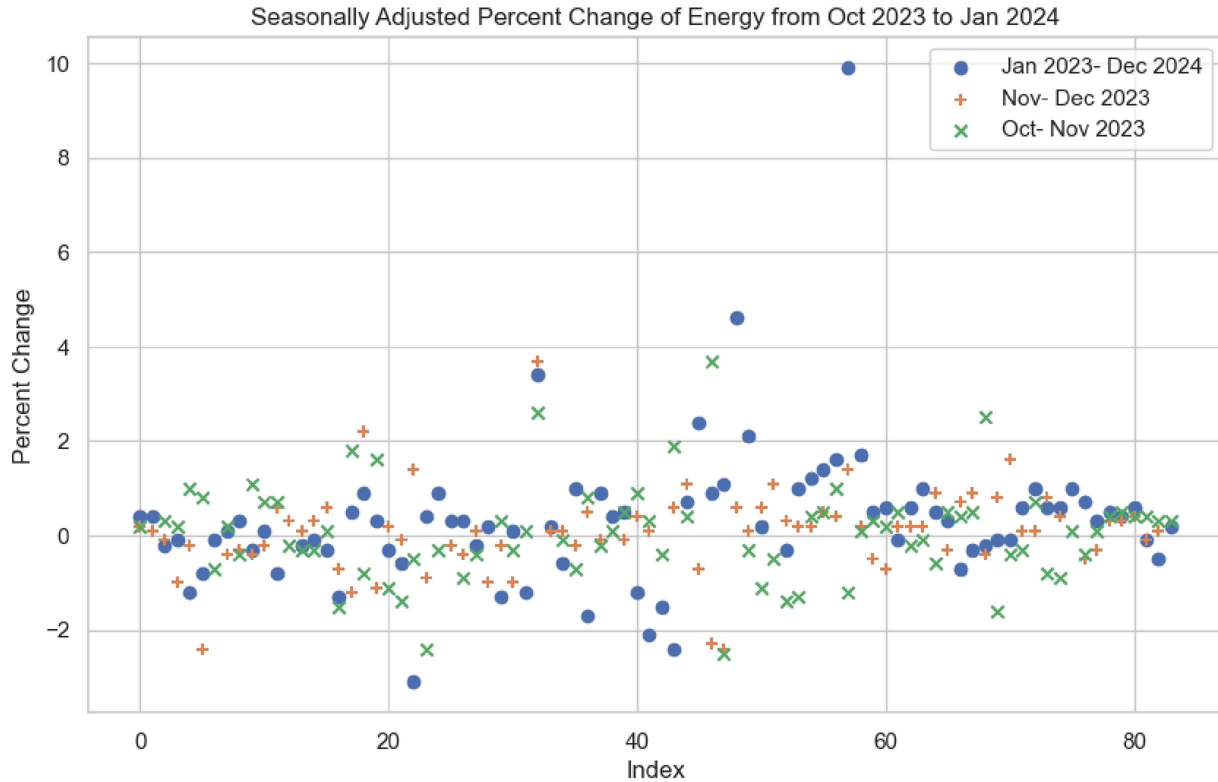
In [90]:
```
# Scatter Plots to Explore Relationships
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='relative_importance_dec_2023', y='unadjusted_pct_change_ja
plt.title('Relationship Between Relative Importance and Percent Change')
plt.show()
```



In [91]:
```
import matplotlib.pyplot as plt
```

```python
# Filter the DataFrame for 'Food' from the 'Subclass' column from Oct 2023 to Jan 2024
food_dec_jan = df[(df['class'] == 'Food') & (df['seasonally_adj_pct_change_dec_jan_202
food_nov_dec = df[(df['class'] == 'Food') & (df['seasonally_adj_pct_change_nov_dec_202
food_oct_nov = df[(df['class'] == 'Food') & (df['seasonally_adj_pct_change_oct_nov_202

# Plotting the scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(food_dec_jan.index, food_dec_jan['seasonally_adj_pct_change_dec_jan_2024']
plt.scatter(food_nov_dec.index, food_nov_dec['seasonally_adj_pct_change_nov_dec_2023']
plt.scatter(food_oct_nov.index, food_oct_nov['seasonally_adj_pct_change_oct_nov_2023']
plt.xlabel('Index')
plt.ylabel('Percent Change')
plt.title('Seasonally Adjusted Percent Change of Food from Oct 2023 to Jan 2024')
plt.legend()
plt.grid(True)
plt.show()

# Filter the DataFrame for 'Energy' from the 'Subclass' column from Oct 2023 to Jan 20
eng_dec_jan = df[(df['class'] == 'Energy') & (df['seasonally_adj_pct_change_dec_jan_20
eng_nov_dec = df[(df['class'] == 'Energy') & (df['seasonally_adj_pct_change_nov_dec_20
eng_oct_nov = df[(df['class'] == 'Energy') & (df['seasonally_adj_pct_change_oct_nov_20

# Plotting the scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(food_dec_jan.index, food_dec_jan['seasonally_adj_pct_change_dec_jan_2024']
plt.scatter(food_nov_dec.index, food_nov_dec['seasonally_adj_pct_change_nov_dec_2023']
plt.scatter(food_oct_nov.index, food_oct_nov['seasonally_adj_pct_change_oct_nov_2023']
plt.xlabel('Index')
plt.ylabel('Percent Change')
plt.title('Seasonally Adjusted Percent Change of Energy from Oct 2023 to Jan 2024')
plt.legend()
plt.grid(True)
plt.show()
```

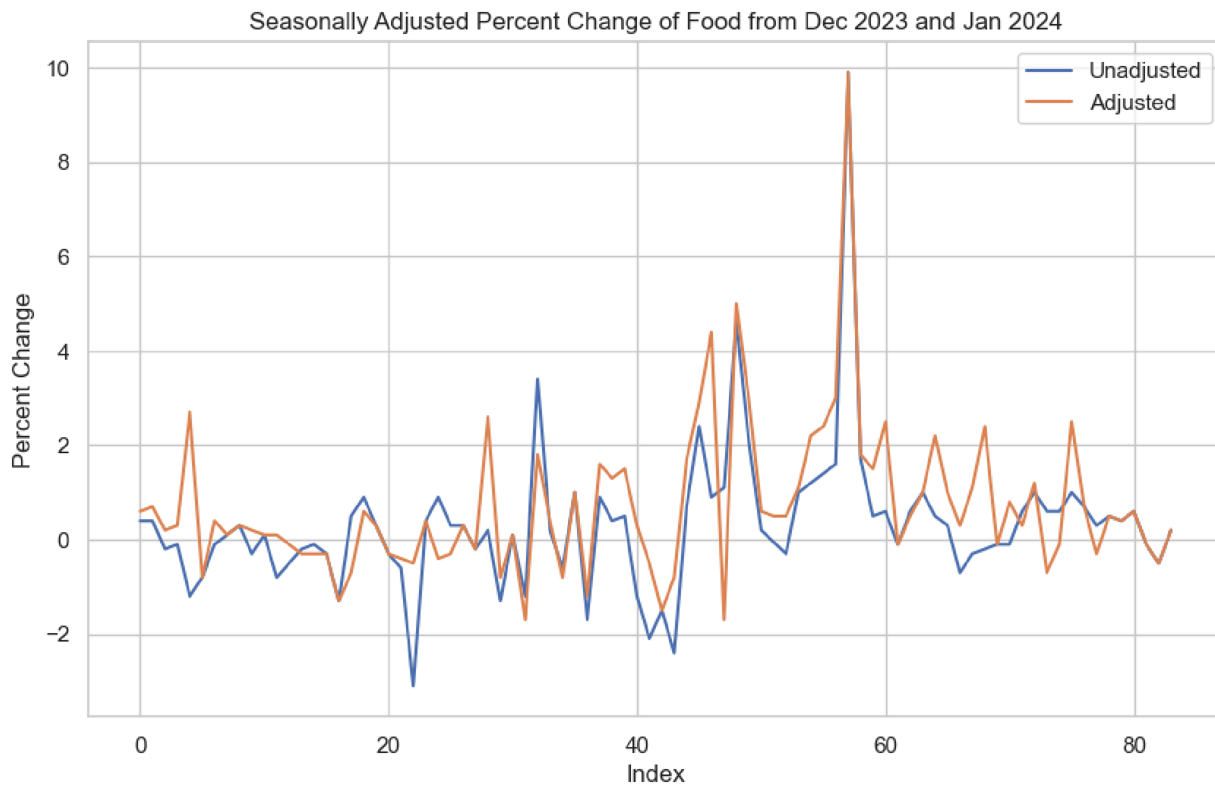


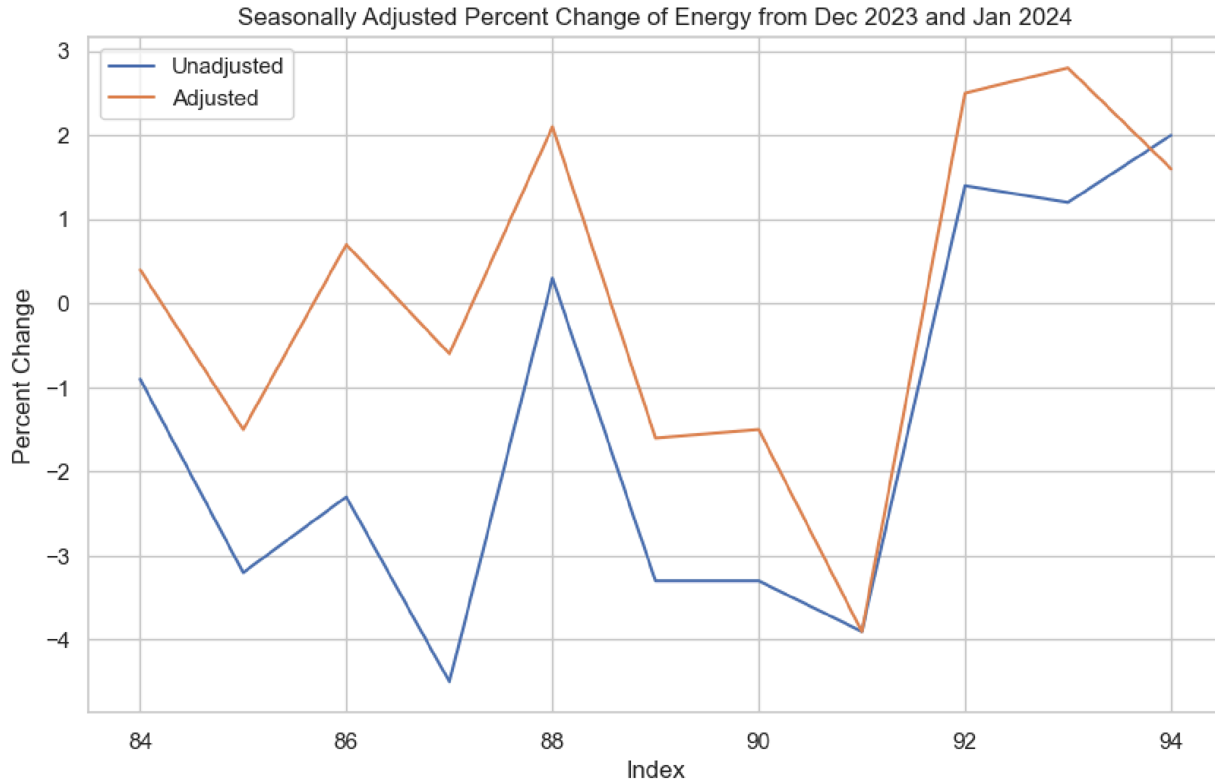


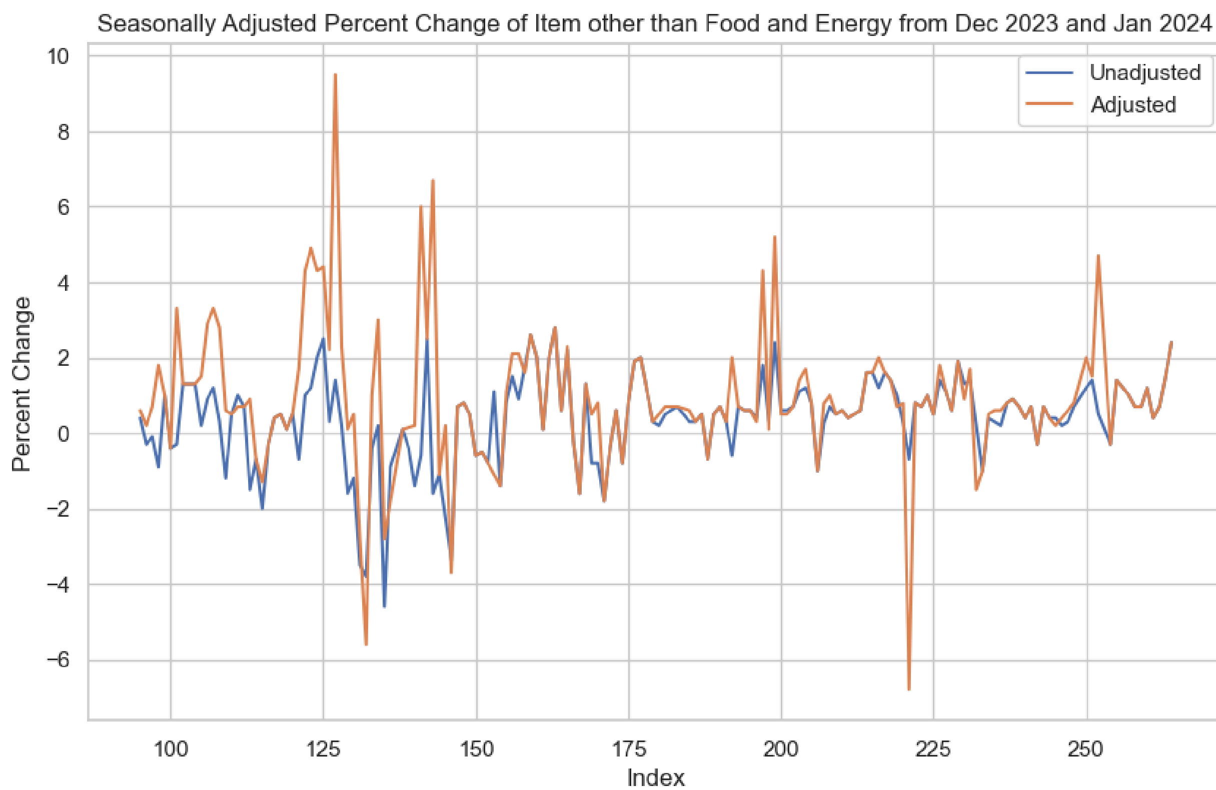
In [92]:

```
import matplotlib.pyplot as plt

# Filter the DataFrame for 'Food' from the 'Subclass' column for years 2023 and 2024
food_unadjusted = df[(df['class'] == 'Food') & (df['seasonally_adj_pct_change_dec_jan_
food_adjusted = df[(df['class'] == 'Food') & (df['unadjusted_pct_change_dec_2023_2024'

# Plotting the line graph
plt.figure(figsize=(10, 6))
plt.plot(food_unadjusted.index, food_unadjusted['seasonally_adj_pct_change_dec_jan_202
plt.plot(food_adjusted.index, food_adjusted['unadjusted_pct_change_dec_2023_2024'], la
plt.xlabel('Index')
plt.ylabel('Percent Change')
plt.title('Seasonally Adjusted Percent Change of Food from Dec 2023 and Jan 2024')
plt.legend()
plt.grid(True)
plt.show()


# Filter the DataFrame for 'Energy' from the 'Subclass' column for years 2023 and 2024
eng_unadjusted = df[(df['class'] == 'Energy') & (df['seasonally_adj_pct_change_dec_jar
eng_adjusted = df[(df['class'] == 'Energy') & (df['unadjusted_pct_change_dec_2023_2024

# Plotting the line graph
plt.figure(figsize=(10, 6))
plt.plot(eng_unadjusted.index, eng_unadjusted['seasonally_adj_pct_change_dec_jan_2024'
plt.plot(eng_adjusted.index, eng_adjusted['unadjusted_pct_change_dec_2023_2024'], labe
plt.xlabel('Index')
plt.ylabel('Percent Change')
plt.title('Seasonally Adjusted Percent Change of Energy from Dec 2023 and Jan 2024')
plt.legend()
plt.grid(True)
plt.show()
```

```
# Filter the DataFrame for 'Other' from the 'Subclass' column for years 2023 and 2024
other_unadjusted = df[(df['class'] == 'Other') & (df['seasonally_adj_pct_change_dec_ja
other_adjusted = df[(df['class'] == 'Other') & (df['unadjusted_pct_change_dec_2023_202

# Plotting the line graph
plt.figure(figsize=(10, 6))
plt.plot(other_unadjusted.index, other_unadjusted['seasonally_adj_pct_change_dec_jan_2
plt.plot(other_adjusted.index, other_adjusted['unadjusted_pct_change_dec_2023_2024'],
plt.xlabel('Index')
plt.ylabel('Percent Change')
plt.title('Seasonally Adjusted Percent Change of Item other than Food and Energy from
plt.legend()
plt.grid(True)
plt.show()
```







Prediction Model Building

SVM

In [93]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error

# Sample data setup (assuming 'df' is your DataFrame and already loaded)
X = df[['relative_importance_dec_2023', 'seasonally_adj_pct_change_oct_nov_2023',
        'seasonally_adj_pct_change_nov_dec_2023', 'seasonally_adj_pct_change_dec_jan_2
y = df['unadjusted_pct_change_dec_2023_2024']

# Splitting data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

# Scaling features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# SVM Regression
svr = SVR(kernel='linear')
svr.fit(X_train_scaled, y_train)
y_pred = svr.predict(X_test_scaled)

# Predicting and evaluating the model
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)

# Print coefficients
print("Coefficients:", svr.coef_)
```

```
Mean Squared Error: 0.8507026565374077
Coefficients: [[-0.00312975 -0.03036429  0.01446231  1.31011718]]
```

In [94]:

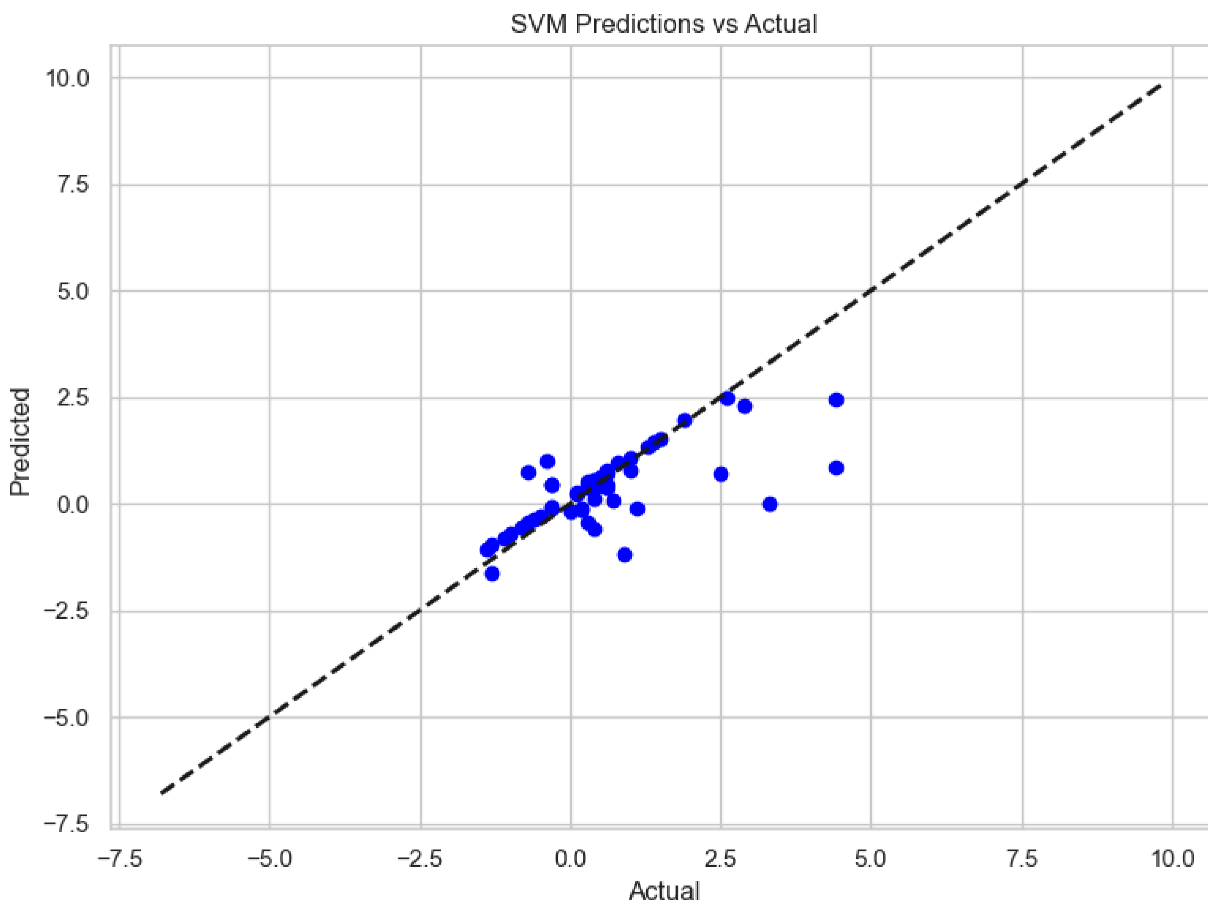
```
# Creating a DataFrame for comparison
results_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})

# Calculate absolute difference
results_df['Difference'] = abs(results_df['Actual'] - results_df['Predicted'])

# Sort by the most accurate (smallest difference)
sorted_results = results_df.sort_values(by='Difference')

# Select the top 10 most accurate predictions
top_accurate_predictions = sorted_results.head(10)

# Display the results in a better-formatted table using pandas
print(top_accurate_predictions.to_markdown(index=False))
```

```
|   Actual |   Predicted |   Difference |
|---------:|------------:|-------------:|
|      1.5 |    1.50585  |   0.00584885 |
|      1.4 |    1.44189  |   0.0418855  |
|      0.4 |    0.356648 |   0.043352   |
|      1.3 |    1.34627  |   0.0462651  |
|      1.9 |    1.9641   |   0.064097   |
|      1   |    1.06524  |   0.0652369  |
|      0.3 |    0.387048 |   0.0870482  |
|      0.6 |    0.713837 |   0.113837   |
|      0.5 |    0.61567  |   0.11567    |
|      2.6 |    2.48254  |   0.117461   |
```

In [95]:

```
# Plotting predictions vs actual values
plt.figure(figsize=(8,6)) # Adjust the figure size to fit on the screen
```

```
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('SVM Predictions vs Actual')
plt.tight_layout()  # Adjust layout to prevent overlapping labels
plt.show()
```



Linear Regression

In [96]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler

# Sample data setup (assuming 'df' is your DataFrame and already loaded)
X = df[['relative_importance_dec_2023', 'seasonally_adj_pct_change_oct_nov_2023',
        'seasonally_adj_pct_change_nov_dec_2023', 'seasonally_adj_pct_change_dec_jan_2
y = df['unadjusted_pct_change_dec_2023_2024']

# Splitting data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=


# Scaling features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Linear Regression
```

```
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
y_pred_lr = lr_model.predict(X_test_scaled)

# Predicting and evaluating the model
mse_lr = mean_squared_error(y_test, y_pred_lr)
print("Mean Squared Error:", mse_lr)

# Print coefficients
print("Coefficients:", lr_model.coef_)
```

```
Mean Squared Error: 0.9760381772638129
Coefficients: [-0.00961613 -0.11453442  0.12458962  0.82329613]
```

```
C:\Users\akank\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does n
ot have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

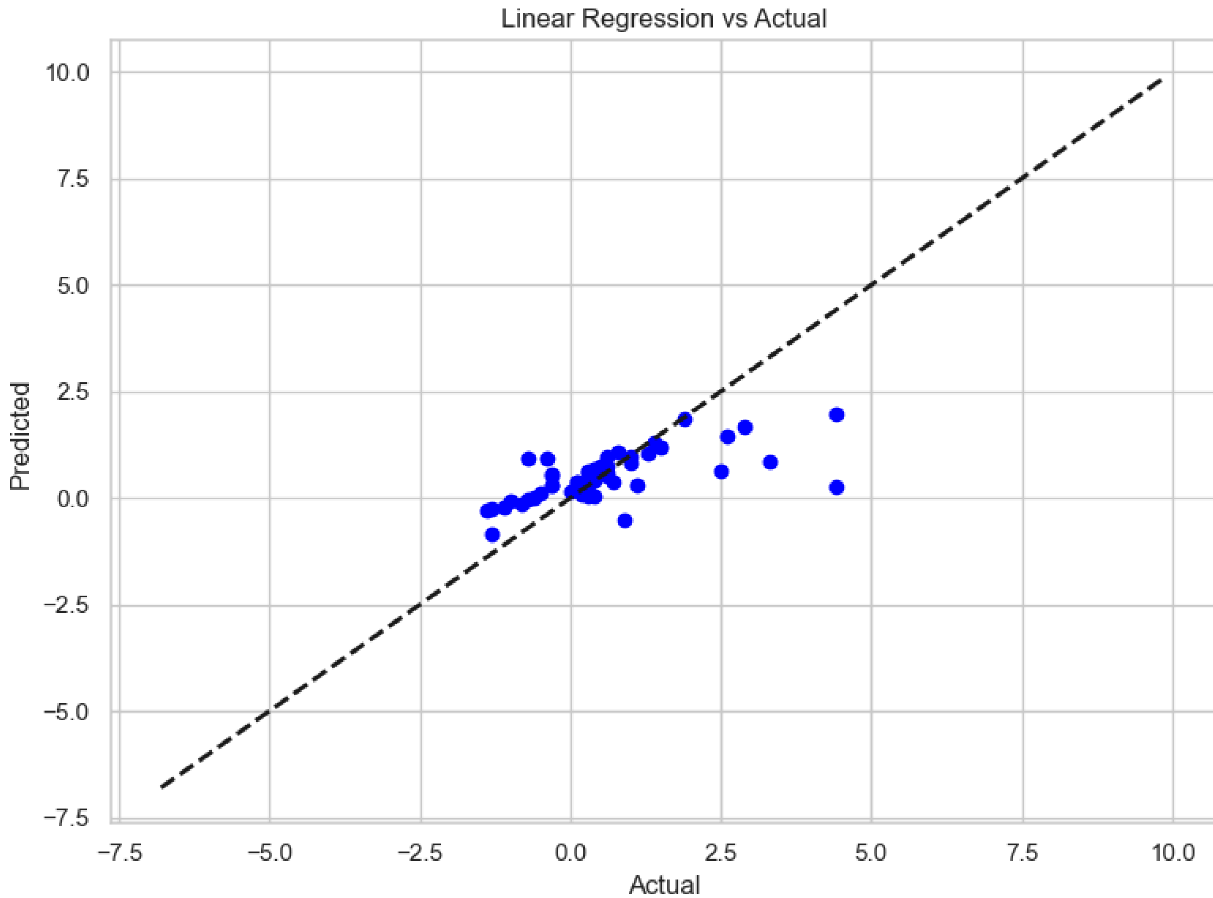
In [97]:

```
# Creating a DataFrame for comparison
results_df_lr = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred_lr})

# Calculate absolute difference
results_df_lr['Difference'] = abs(results_df_lr['Actual'] - results_df_lr['Predicted']

# Sort by the most accurate (smallest difference)
sorted_results = results_df_lr.sort_values(by='Difference')

# Select the top 10 most accurate predictions
top_accurate_predictions = sorted_results.head(10)

# Display the results in a better-formatted table using pandas
print(top_accurate_predictions.to_markdown(index=False))
```

Actual	Predicted	Difference
0.6	0.604547	0.00454731
0.3	0.287188	0.0128118
0.4	0.424995	0.0249954
1.9	1.87384	0.0261613
0.2	0.227374	0.0273738
1	0.953059	0.0469413
0.6	0.5188	0.0812004
1.4	1.29284	0.107156
0.2	0.0824373	0.117563
0.4	0.533906	0.133906

In [98]:

```
# Plotting predictions vs actual values
plt.figure(figsize=(8,6)) # Adjust the figure size to fit on the screen
plt.scatter(y_test, y_pred_lr, color='blue')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Linear Regression vs Actual')
plt.tight_layout()  # Adjust layout to prevent overlapping labels
plt.show()
```



Lasso Regression

In [99]:

```
import numpy as np
from sklearn.linear_model import Lasso, LassoCV
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error

# Sample data setup (assuming 'df' is your DataFrame and already loaded)
X = df[['relative_importance_dec_2023', 'seasonally_adj_pct_change_oct_nov_2023',
        'seasonally_adj_pct_change_nov_dec_2023', 'seasonally_adj_pct_change_dec_jan_2
y = df['unadjusted_pct_change_dec_2023_2024']

# Splitting data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

# Scaling features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Lasso Regression with Cross-Validation
# Using LassoCV to find the best alpha (regularization strength)
lasso_cv = LassoCV(cv=5, random_state=0, alphas=np.logspace(-6, 6, 13))
lasso_cv.fit(X_train_scaled, y_train)

print("Optimal alpha:", lasso_cv.alpha_)

# Using the best alpha to fit the final Lasso model
lasso = Lasso(alpha=lasso_cv.alpha_)
```

```
lasso.fit(X_train_scaled, y_train)
y_pred_lasso = lasso.predict(X_test_scaled)

# Predicting and evaluating the model
mse_lasso = mean_squared_error(y_test, y_pred_lasso)
print("Mean Squared Error:", mse_lasso)

# Print coefficients
print("Coefficients:", lasso.coef_)
```

```
Optimal alpha: 0.001
Mean Squared Error: 0.9138474344654427
Coefficients: [-0.07638062 -0.12783274  0.13058446  1.20255951]
```

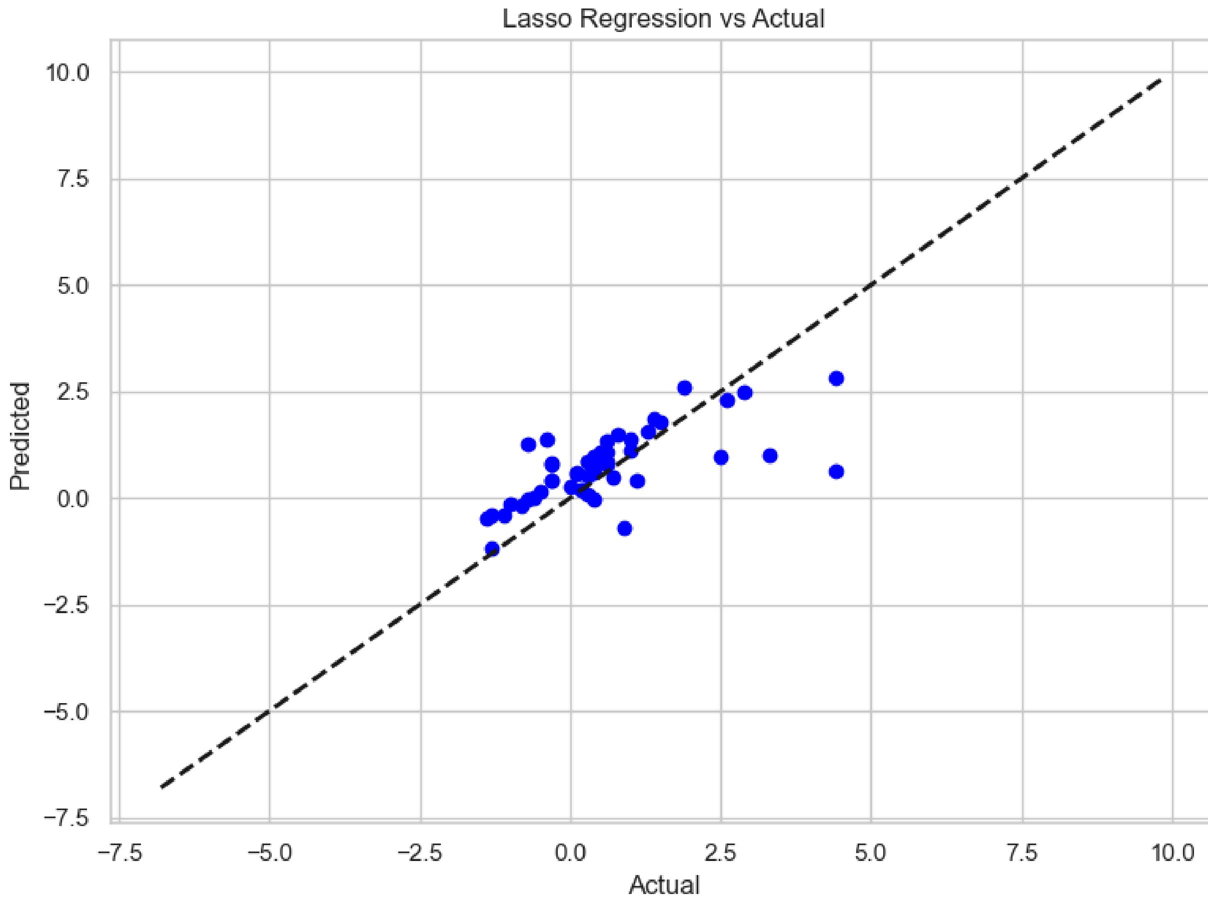
In [100…

```
# Creating a DataFrame for comparison
results_df_lasso = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred_lasso})

# Calculate absolute difference
results_df_lasso['Difference'] = abs(results_df_lasso['Actual'] - results_df_lasso['Pr

# Sort by the most accurate (smallest difference)
sorted_results = results_df_lasso.sort_values(by='Difference')

# Select the top 10 most accurate predictions
top_accurate_predictions = sorted_results.head(10)

# Display the results in a better-formatted table using pandas
print(top_accurate_predictions.to_markdown(index=False))
```

Actual	Predicted	Difference
0.2	0.182062	0.0179381
0.2	0.175925	0.024075
1	1.11694	0.116944
-1.3	-1.17597	0.124033
0.6	0.769942	0.169942
0.4	0.593684	0.193684
0.7	0.501837	0.198163
0.3	0.527034	0.227034
0.3	0.0625711	0.237429
0.6	0.841461	0.241461

In [101…

```
# Plotting predictions vs actual values
plt.figure(figsize=(8,6)) # Adjust the figure size to fit on the screen
plt.scatter(y_test, y_pred_lasso, color='blue')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Lasso Regression vs Actual')
plt.tight_layout()  # Adjust layout to prevent overlapping labels
plt.show()
```



Part 02: Statistical Testing

Hypothesis Testing

ANOVA for Expenditure Category: Food

Null Hypothesis (H0): There is no significant difference in the seasonally adjusted percent changes between expenditure categories or time periods.

Alternative Hypothesis (H1): There is a significant difference in the seasonally adjusted percent changes between expenditure categories or time periods.

In [102…

```
# Food sector ANOVA test

import pandas as pd
from scipy.stats import f_oneway

# Create a DataFrame from the provided data
data_food = {
    "Expenditure_category": ["Food", "Food at home", "Food away from home(1)",
                             "Limited service meals and snacks(1)(2)", "Full service m
                             "Other food at home", "Meats, poultry, fish, and eggs",
                             "Other foods", "Meats, poultry, and fish", "Fruits and ve
                             "Fresh fruits and vegetables", "Cereals and bakery produc
                             "Meats", "Nonalcoholic beverages and beverage materials",
                             "Dairy and related products", "Juices and nonalcoholic dr
                             "Fresh fruits", "Other miscellaneous foods(2)", "Fresh ve
```

```
    "Relative_importance_Dec_2023": [13.555, 8.167, 5.388, 2.523, 2.474, 2.193, 1.722,
                                     1.603, 1.410, 1.070, 1.066, 1.033, 1.027, 0.752,
                                     0.730, 0.575, 0.572, 0.495],
    "Seasonally_adjusted_percent_change_Oct_2023_Nov_2023": [0.2, 0.0, 0.4, 0.4, 0.5,
                                                             -0.3, 0.3, 0.1, 0.5, 0.3,
                                                             0.2, 0.0, 0.5, 0.9, 0.1,
    "Seasonally_adjusted_percent_change_Nov_2023_Dec_2023": [0.2, 0.1, 0.3, 0.4, 0.3,
                                                             0.1, 0.1, 0.0, -0.1, -0.1
                                                             -0.4, 0.1, 0.5, 0.4, -0.3
    "Seasonally_adjusted_percent_change_Dec_2023_Jan_2024": [0.4, 0.4, 0.5, 0.6, 0.4,
                                                             0.6, -0.2, 0.4, 0.5, -0.2
                                                             0.1, 0.2, 1.4, -1.2, 0.3,
}

df_food = pd.DataFrame(data_food)

# Perform ANOVA test
f_statistic_food, p_value_food = f_oneway(
    df_food["Seasonally_adjusted_percent_change_Oct_2023_Nov_2023"],
    df_food["Seasonally_adjusted_percent_change_Nov_2023_Dec_2023"],
    df_food["Seasonally_adjusted_percent_change_Dec_2023_Jan_2024"]
)

# Print results
print("ANOVA Results for Food Expenditure Categories:")
print("F-statistic:", f_statistic_food)
print("p-value:", p_value_food)
```

```
ANOVA Results for Food Expenditure Categories:
F-statistic: 2.3444376760831487
p-value: 0.10508436022226833
```

With a p-value of approximately 0.105, at a significance level of 0.05, we do not have enough evidence to reject the null hypothesis. This suggests that there is no significant difference in the seasonally adjusted percent changes between the time periods (October 2023-November 2023, November 2023-December 2023, and December 2023-January 2024) for the food expenditure categories.

In other words, based on the data provided, we cannot conclude that the percentage changes in expenditure categories within the food sector significantly differ across the specified time periods.

ANOVA for Expenditure Category: Energy

Null Hypothesis (H0): There is no significant difference in the percentage changes between expenditure categories or time periods.

Alternative Hypothesis (H1): There is a significant difference in the percentage changes between expenditure categories or time periods.

In [103…

```
# Energy sector ANOVA test

import pandas as pd
from scipy.stats import f_oneway
```

```
# Create a DataFrame from the provided data
data = {
    "Expenditure_category": ["Energy", "Energy commodities", "Fuel oil and other fuels
                             "Propane, kerosene, and firewood", "Motor fuel", "Gasoli
                             "Gasoline, unleaded regular", "Gasoline, unleaded midgra
                             "Gasoline, unleaded premium", "Other motor fuels", "Ener
                             "Electricity", "Utility (piped) gas service"],
    "Relative_importance_Dec_2023": [6.655, 3.539, 0.167, 0.084, 0.083, 3.372, 3.261,
    "Seasonally_adjusted_percent_change_Oct_2023_Nov_2023": [-0.2, -0.7, -2.5, -3.3, -
    "Seasonally_adjusted_percent_change_Nov_2023_Dec_2023": [-0.9, -3.2, -2.3, -4.5, 0
    "Seasonally_adjusted_percent_change_Dec_2023_Jan_2024": [-0.9, -3.2, -2.3, -4.5, 0
}

df = pd.DataFrame(data)

# Drop rows with missing values
df.dropna(inplace=True)

# Perform ANOVA test
f_statistic, p_value = f_oneway(

    df["Seasonally_adjusted_percent_change_Oct_2023_Nov_2023"],
    df["Seasonally_adjusted_percent_change_Nov_2023_Dec_2023"],
    df["Seasonally_adjusted_percent_change_Dec_2023_Jan_2024"]
)

# Print results
print("ANOVA Results for Energy Expenditure Categories:")
print("F-statistic:", f_statistic)
print("p-value:", p_value)
```

```
ANOVA Results for Energy Expenditure Categories:
F-statistic: 0.03464661654135346
p-value: 0.9659852972590984
```

With a p-value of approximately 0.966, we fail to reject the null hypothesis. This suggests that there is no significant difference in the seasonally adjusted percentage changes between the time periods (October 2023-November 2023, November 2023-December 2023, and December 2023-January 2024) for the expenditure categories within the energy sector.

In other words, based on the data provided, we do not have sufficient evidence to conclude that the percentage changes in expenditure categories significantly differ across the specified time periods.

CHI Square test of independance for energy sector

Null Hypothesis (H0): There is no significant association between the expenditure categories and the months.

Alternative Hypothesis (H1): There is a significant association between the expenditure categories and the months.

In [104…

```
import numpy as np
from scipy.stats import chi2_contingency

# Define the observed frequencies (hypothetical in this case)
```

```
observed = np.array([
    [1000, 984, 975],
    [984, 977, 944],
    [974, 949, 928],
    [963, 925, 885],
    [973, 921, 954],
    [960, 921, 888],
    [960, 921, 888],
    [959, 920, 887],
    [960, 920, 887],
    [962, 922, 889],
    [958, 901, 865],
    [1010, 1013, 1027],
    [1010, 1016, 1028],
    [1012, 999, 1019]
])

# Perform chi-square test
chi2, p, dof, expected = chi2_contingency(observed)

# Print the results
print("Chi-square statistic:", chi2)
print("p-value:", p)
print("Degrees of freedom:", dof)
print("Expected frequencies:")
print(expected)
```

```
Chi-square statistic: 11.941417141295489
p-value: 0.9914976682255342
Degrees of freedom: 26
Expected frequencies:
[[1011.3870573   982.12076028  965.49218243]
 [ 992.92984165  964.19763724  947.8725211 ]
 [ 974.47262601  946.27451421  930.25285978]
 [ 947.81220341  920.38555872  904.80223787]
 [ 973.44722514  945.27878515  929.27398971]
 [ 946.44500225  919.05791998  903.49707778]
 [ 946.44500225  919.05791998  903.49707778]
 [ 945.41960138  918.06219092  902.5182077 ]
 [ 945.76140167  918.3941006   902.84449773]
 [ 947.81220341  920.38555872  904.80223787]
 [ 931.06398921  904.12198412  888.81402667]
 [1042.49088366 1012.32454169  995.18457465]
 [1043.85808482 1013.65218043  996.48973475]
 [1035.65487787 1005.68634797  988.65877416]]
```

In [105…

```
# Data provided
data = {
    "Food": [0.2, 0.2, 0.4],
    "Food at home": [0.0, 0.1, 0.4],
    "Food away from home(1)": [0.4, 0.3, 0.5],
    "Limited service meals and snacks(1)(2)": [0.4, 0.4, 0.6],
    "Full service meals and snacks(1)(2)": [0.5, 0.3, 0.4],
    "Other food at home": [-0.2, 0.2, 0.6],
    "Meats, poultry, fish, and eggs": [-0.2, 0.3, 0.0],
    "Other foods": [-0.3, 0.1, 0.6],
    "Meats, poultry, and fish": [-0.3, 0.1, -0.2],
    "Fruits and vegetables": [0.1, 0.0, 0.4],
    "Fresh fruits and vegetables": [0.5, -0.1, 0.5],
    "Cereals and bakery products": [0.3, -0.1, -0.2],
```

```
    "Meats": [-0.3, 0.3, -0.1],
    "Nonalcoholic beverages and beverage materials": [0.4, 0.2, 1.2],
    "Bakery products(1)": [0.2, -0.4, 0.1],
    "Dairy and related products": [0.0, 0.1, 0.2],
    "Juices and nonalcoholic drinks(2)": [0.5, 0.5, 1.4],
    "Fresh fruits": [0.9, 0.4, -1.2],
    "Other miscellaneous foods(2)": [0.1, -0.3, 0.3],
    "Fresh vegetables": [0.0, -0.7, 2.4],
    "Beef and veal": [0.1, 0.6, -0.3],
    "Nonfrozen noncarbonated juices and drinks(2)": [0.1, 0.2, 1.7],
    "Snacks": [-0.9, 0.4, 0.6],
    "Processed fruits and vegetables(2)": [-1.1, 0.6, 0.2],
    "Pork": [-1.1, 0.2, -0.3],
    "Carbonated drinks": [1.0, 0.4, 1.6],
    "Spices, seasonings, condiments, sauces": [0.1, 0.0, 1.0],
    "Cereals and cereal products": [0.2, -1.0, -0.1],
    "Poultry(1)": [-0.9, -0.4, 0.3],
    "Beverage materials including coffee and tea(2)": [0.3, -0.5, 0.5]
}

# Calculate observed frequencies
observed_frequencies = {category: len(changes) for category, changes in data.items()}

print("Observed Frequencies:")
for category, frequency in observed_frequencies.items():
    print(f"{category}: {frequency}")
```

```
Observed Frequencies:
Food: 3
Food at home: 3
Food away from home(1): 3
Limited service meals and snacks(1)(2): 3
Full service meals and snacks(1)(2): 3
Other food at home: 3
Meats, poultry, fish, and eggs: 3
Other foods: 3
Meats, poultry, and fish: 3
Fruits and vegetables: 3
Fresh fruits and vegetables: 3
Cereals and bakery products: 3
Meats: 3
Nonalcoholic beverages and beverage materials: 3
Bakery products(1): 3
Dairy and related products: 3
Juices and nonalcoholic drinks(2): 3
Fresh fruits: 3
Other miscellaneous foods(2): 3
Fresh vegetables: 3
Beef and veal: 3
Nonfrozen noncarbonated juices and drinks(2): 3
Snacks: 3
Processed fruits and vegetables(2): 3
Pork: 3
Carbonated drinks: 3
Spices, seasonings, condiments, sauces: 3
Cereals and cereal products: 3
Poultry(1): 3
Beverage materials including coffee and tea(2): 3
```

CHI Square test of independance for food sector

In [106…

```
import numpy as np
from scipy.stats import chi2_contingency

# Define the observed frequencies as an array
observed_array = np.array(list(observed_frequencies.values()))

# Define expected frequencies as uniform distribution
n = observed_array.sum()
expected_frequency = n / len(observed_frequencies)

# Calculate expected frequencies
expected_array = np.full_like(observed_array, expected_frequency)

# Perform chi-square test
chi2, p_value = chi2_contingency(observed_array.reshape(1, -1), correction=False)[:2]

print("\nResults of Chi-square Test:")
print(f"Chi-square statistic: {chi2}")
print(f"P-value: {p_value}")

# Interpret the results
alpha = 0.05
if p_value < alpha:
    print("Reject null hypothesis: There is a significant relationship between expendi
else:
    print("Fail to reject null hypothesis: There is no significant relationship betwee
```

```
Results of Chi-square Test:
Chi-square statistic: 0.0
P-value: 1.0
Fail to reject null hypothesis: There is no significant relationship between expendit
ure categories.
```

We have a high p-value (greater than 0.05), which suggests that we fail to reject the null hypothesis. Therefore, we do not have sufficient evidence to conclude that there is a significant association between the expenditure categories and the months.

There are no significant conclusion that we can draw from Chi-square test so, we decided to implement t-test.

T-test

In [107…

```
from scipy.stats import ttest_rel

# Data from the table
data = {
    "Oct-Nov": [0.2, 0.0, 0.4, 0.4, 0.5, -0.2, -0.3, -0.3, 0.1, 0.3, 0.5, 0.0, 0.2, 0.
    "Nov-Dec": [0.2, 0.1, 0.3, 0.4, 0.3, 0.2, 0.1, 0.1, 0.0, -0.1, -0.1, -0.4, 0.5, 0.
    "Dec-Jan": [0.4, 0.4, 0.5, 0.6, 0.4, 0.6, 0.0, -0.2, 0.4, -0.2, -0.2, 0.1, 1.2, 1.
}

# Perform paired t-test for each pair of columns
for col1, col2 in [("Oct-Nov", "Nov-Dec"), ("Nov-Dec", "Dec-Jan"), ("Oct-Nov", "Dec-Ja
    t_statistic, p_value = ttest_rel(data[col1], data[col2])
    print(f"Paired t-test between {col1} and {col2}:")
```

```
    print(f"    t-statistic: {t_statistic}")
    print(f"    p-value: {p_value}")
    print()
```

```
Paired t-test between Oct-Nov and Nov-Dec:
    t-statistic: 0.17376545644737687
    p-value: 0.8636387772685213

Paired t-test between Nov-Dec and Dec-Jan:
    t-statistic: -2.986252946432471
    p-value: 0.006807960815979522

Paired t-test between Oct-Nov and Dec-Jan:
    t-statistic: -1.8335370709939431
    p-value: 0.08028850272371114
```

Between October-November and November-December: The t-statistic is low (0.1738), indicating that the difference between these two periods is not significantly different from zero. The p-value (0.8636) is high, suggesting that there is no statistically significant difference between October-November and November-December.

Between November-December and December-January: The t-statistic is larger in absolute value (-2.9863), indicating a stronger evidence against the null hypothesis. The p-value (0.0068) is less than the typical significance level of 0.05, suggesting that there is a statistically significant difference between these two periods. Between October-November and December-January:

The t-statistic is moderate (-1.8335), suggesting some evidence against the null hypothesis. The p-value (0.0803) is higher than 0.05, but it's close, indicating that there might be a borderline significant difference between October-November and December-January, but it's not as strong as the difference between November-December and December-January.

Correlation Analysis

In [108…

```
import pandas as pd

# Data
data = {
    'Energy commodities': [-3.8, -0.7, -3.2],
    'Fuel oil and other fuels': [-1.0, -2.5, -2.3],
    'Fuel oil': [-1.1, -3.3, -4.5],
    'Propane, kerosene, and firewood': [-0.1, -0.4, 0.3],
    'Motor fuel': [-4.0, -0.6, -3.3],
    'Gasoline (all types)': [-4.0, -0.6, -3.3],
    'Gasoline, unleaded regular': [-4.1, -0.6, -3.4],
    'Gasoline, unleaded midgrade': [-3.9, -0.5, -2.7],
    'Gasoline, unleaded premium': [-3.7, -0.3, -2.6],
    'Other motor fuels': [-4.2, -6.1, -3.9],
    'Energy services': [1.0, 0.3, 1.4],
    'Electricity': [1.0, 0.6, 1.2],
    'Utility (piped) gas service': [1.2, -0.6, 2.0],
    'Food at home': [0.0, 0.1, 0.4],
    'Food away from home': [0.4, 0.3, 0.5],
    'Limited service meals and snacks': [0.4, 0.4, 0.6],
    'Full service meals and snacks': [0.5, 0.3, 0.4],
```

```
    'Other food at home': [-0.2, 0.2, 0.6],
    'Meats, poultry, fish, and eggs': [-0.2, 0.3, 0.0],
    'Other foods': [-0.3, 0.1, 0.6],
    'Meats, poultry, and fish': [-0.3, 0.1, -0.2],
    'Fruits and vegetables': [0.1, 0.0, 0.4],
    'Fresh fruits and vegetables': [0.5, -0.1, 0.5]
}

# Create DataFrame
df = pd.DataFrame(data)

# Calculate correlation matrix
correlation_matrix = df.corr()

print(correlation_matrix)
```

```
                                  Energy commodities  \
Energy commodities                          1.000000
Fuel oil and other fuels                   -0.738053
Fuel oil                                   -0.344489
Propane, kerosene, and firewood            -0.704285
Motor fuel                                  0.999919
Gasoline (all types)                        0.999919
Gasoline, unleaded regular                  0.999978
Gasoline, unleaded midgrade                 0.985261
Gasoline, unleaded premium                  0.990342
Other motor fuels                          -0.952470
Energy services                            -0.852048
Electricity                                -0.869324
Utility (piped) gas service                -0.883002
Food at home                               -0.097391
Food away from home                        -0.760257
Limited service meals and snacks           -0.333590
Full service meals and snacks              -0.942718
Other food at home                          0.182462
Meats, poultry, fish, and eggs              0.974761
Other foods                                 0.119144
Meats, poultry, and fish                    0.998256
Fruits and vegetables                      -0.550258
Fresh fruits and vegetables                -0.983213

                                  Fuel oil and other fuels  Fuel oil  \
Energy commodities                               -0.738053 -0.344489
Fuel oil and other fuels                          1.000000  0.887693
Fuel oil                                          0.887693  1.000000
Propane, kerosene, and firewood                   0.040789 -0.423845
Motor fuel                                       -0.746572 -0.356396
Gasoline (all types)                             -0.746572 -0.356396
Gasoline, unleaded regular                       -0.742515 -0.350711
Gasoline, unleaded midgrade                      -0.842596 -0.500000
Gasoline, unleaded premium                       -0.824474 -0.471318
Other motor fuels                                 0.497425  0.042129
Energy services                                   0.275653 -0.197902
Electricity                                       0.308122 -0.164517
Utility (piped) gas service                       0.334999 -0.136455
Food at home                                     -0.599655 -0.900778
Food away from home                               0.122782 -0.347960
Limited service meals and snacks                 -0.389885 -0.770097
Full service meals and snacks                     0.920864  0.637927
Other food at home                               -0.798082 -0.985887
Meats, poultry, fish, and eggs                   -0.870062 -0.545380
Other foods                                      -0.757871 -0.973147
Meats, poultry, and fish                         -0.776603 -0.399314
Fruits and vegetables                            -0.157287 -0.594328
Fresh fruits and vegetables                       0.602549  0.167412

                                  Propane, kerosene, and firewood  Motor fuel  \
Energy commodities                                      -0.704285    0.999919
Fuel oil and other fuels                                 0.040789   -0.746572
Fuel oil                                                -0.423845   -0.356396
Propane, kerosene, and firewood                          1.000000   -0.695203
Motor fuel                                              -0.695203    1.000000
Gasoline (all types)                                    -0.695203    1.000000
Gasoline, unleaded regular                              -0.699559    0.999982
Gasoline, unleaded midgrade                             -0.572466    0.987356
Gasoline, unleaded premium                              -0.599059    0.992025
```

```
Other motor fuels                                           0.887074   -0.948520
Energy services                                             0.971701   -0.845324
Electricity                                                 0.963123   -0.862971
Utility (piped) gas service                                 0.955098   -0.876964
Food at home                                                0.775133   -0.084730
Food away from home                                         0.996616   -0.751936
Limited service meals and snacks                            0.904194   -0.321578
Full service meals and snacks                               0.427121   -0.946883
Other food at home                                          0.569495    0.194946
Meats, poultry, fish, and eggs                             -0.528020    0.977520
Other foods                                                 0.620949    0.131756
Meats, poultry, and fish                                   -0.661143    0.998926
Fruits and vegetables                                       0.980316   -0.539598
Fresh fruits and vegetables                                 0.821995   -0.980814

                                  Gasoline (all types)  \
Energy commodities                            0.999919
Fuel oil and other fuels                     -0.746572
Fuel oil                                     -0.356396
Propane, kerosene, and firewood              -0.695203
Motor fuel                                    1.000000
Gasoline (all types)                          1.000000
Gasoline, unleaded regular                    0.999982
Gasoline, unleaded midgrade                   0.987356
Gasoline, unleaded premium                    0.992025
Other motor fuels                            -0.948520
Energy services                              -0.845324
Electricity                                  -0.862971
Utility (piped) gas service                  -0.876964
Food at home                                 -0.084730
Food away from home                          -0.751936
Limited service meals and snacks             -0.321578
Full service meals and snacks                -0.946883
Other food at home                            0.194946
Meats, poultry, fish, and eggs                0.977520
Other foods                                   0.131756
Meats, poultry, and fish                      0.998926
Fruits and vegetables                        -0.539598
Fresh fruits and vegetables                  -0.980814

                                  Gasoline, unleaded regular  \
Energy commodities                                  0.999978
Fuel oil and other fuels                           -0.742515
Fuel oil                                           -0.350711
Propane, kerosene, and firewood                    -0.699559
Motor fuel                                          0.999982
Gasoline (all types)                                0.999982
Gasoline, unleaded regular                          1.000000
Gasoline, unleaded midgrade                         0.986374
Gasoline, unleaded premium                          0.991241
Other motor fuels                                  -0.950427
Energy services                                    -0.848555
Electricity                                        -0.866025
Utility (piped) gas service                        -0.879868
Food at home                                       -0.090784
Food away from home                                -0.755929
Limited service meals and snacks                   -0.327327
Full service meals and snacks                      -0.944911
Other food at home                                  0.188982
Meats, poultry, fish, and eggs                      0.976221
```

```
Other foods                                        0.125730
Meats, poultry, and fish                           0.998625
Fruits and vegetables                             -0.544705
Fresh fruits and vegetables                       -0.981981

                                  Gasoline, unleaded midgrade  \
Energy commodities                                  0.985261
Fuel oil and other fuels                           -0.842596
Fuel oil                                           -0.500000
Propane, kerosene, and firewood                    -0.572466
Motor fuel                                          0.987356
Gasoline (all types)                                0.987356
Gasoline, unleaded regular                          0.986374
Gasoline, unleaded midgrade                         1.000000
Gasoline, unleaded premium                          0.999462
Other motor fuels                                  -0.886321
Energy services                                    -0.749946
Electricity                                        -0.771966
Utility (piped) gas service                        -0.789697
Food at home                                        0.074291
Food away from home                                -0.637927
Limited service meals and snacks                   -0.167412
Full service meals and snacks                      -0.985887
Other food at home                                  0.347960
Meats, poultry, fish, and eggs                      0.998583
Other foods                                         0.287228
Meats, poultry, and fish                            0.993641
Fruits and vegetables                              -0.399314
Fresh fruits and vegetables                        -0.937509

                                  Gasoline, unleaded premium  \
Energy commodities                                 0.990342
Fuel oil and other fuels                          -0.824474
Fuel oil                                          -0.471318
Propane, kerosene, and firewood                   -0.599059
Motor fuel                                         0.992025
Gasoline (all types)                               0.992025
Gasoline, unleaded regular                         0.991241
Gasoline, unleaded midgrade                        0.999462
Gasoline, unleaded premium                         1.000000
Other motor fuels                                 -0.901036
Energy services                                   -0.771245
Electricity                                       -0.792406
Utility (piped) gas service                       -0.809400
Food at home                                       0.041533
Food away from home                               -0.662849
Limited service meals and snacks                  -0.199667
Full service meals and snacks                     -0.979864
Other food at home                                 0.317015
Meats, poultry, fish, and eggs                     0.996299
Other foods                                        0.255648
Meats, poultry, and fish                           0.996800
Fruits and vegetables                             -0.429178
Fresh fruits and vegetables                       -0.948421

                                  Other motor fuels  ...  Food at home  \
Energy commodities                        -0.952470  ...     -0.097391
Fuel oil and other fuels                   0.497425  ...     -0.599655
Fuel oil                                   0.042129  ...     -0.900778
Propane, kerosene, and firewood            0.887074  ...      0.775133
```

```
Motor fuel                          -0.948520  ...     -0.084730
Gasoline (all types)                -0.948520  ...     -0.084730
Gasoline, unleaded regular          -0.950427  ...     -0.090784
Gasoline, unleaded midgrade         -0.886321  ...      0.074291
Gasoline, unleaded premium          -0.901036  ...      0.041533
Other motor fuels                    1.000000  ...      0.395946
Energy services                      0.971014  ...      0.603957
Electricity                          0.978568  ...      0.576557
Utility (piped) gas service          0.984018  ...      0.553134
Food at home                         0.395946  ...      1.000000
Food away from home                  0.922018  ...      0.720577
Limited service meals and snacks     0.604917  ...      0.970725
Full service meals and snacks        0.796288  ...     -0.240192
Other food at home                   0.125730  ...      0.960769
Meats, poultry, fish, and eggs      -0.860421  ...      0.127257
Other foods                          0.188982  ...      0.976554
Meats, poultry, and fish            -0.932823  ...     -0.038462
Fruits and vegetables                0.778471  ...      0.884615
Fresh fruits and vegetables          0.992065  ...      0.277350

                                  Food away from home  \
Energy commodities                          -0.760257
Fuel oil and other fuels                     0.122782
Fuel oil                                    -0.347960
Propane, kerosene, and firewood              0.996616
Motor fuel                                  -0.751936
Gasoline (all types)                        -0.751936
Gasoline, unleaded regular                  -0.755929
Gasoline, unleaded midgrade                 -0.637927
Gasoline, unleaded premium                  -0.662849
Other motor fuels                            0.922018
Energy services                              0.987829
Electricity                                  0.981981
Utility (piped) gas service                  0.976221
Food at home                                 0.720577
Food away from home                          1.000000
Limited service meals and snacks             0.866025
Full service meals and snacks                0.500000
Other food at home                           0.500000
Meats, poultry, fish, and eggs              -0.596040
Other foods                                  0.554416
Meats, poultry, and fish                    -0.720577
Fruits and vegetables                        0.960769
Fresh fruits and vegetables                  0.866025

                                  Limited service meals and snacks  \
Energy commodities                                       -0.333590
Fuel oil and other fuels                                 -0.389885
Fuel oil                                                 -0.770097
Propane, kerosene, and firewood                           0.904194
Motor fuel                                               -0.321578
Gasoline (all types)                                     -0.321578
Gasoline, unleaded regular                               -0.327327
Gasoline, unleaded midgrade                              -0.167412
Gasoline, unleaded premium                               -0.199667
Other motor fuels                                         0.604917
Energy services                                           0.777714
Electricity                                               0.755929
Utility (piped) gas service                               0.737043
Food at home                                              0.970725
```

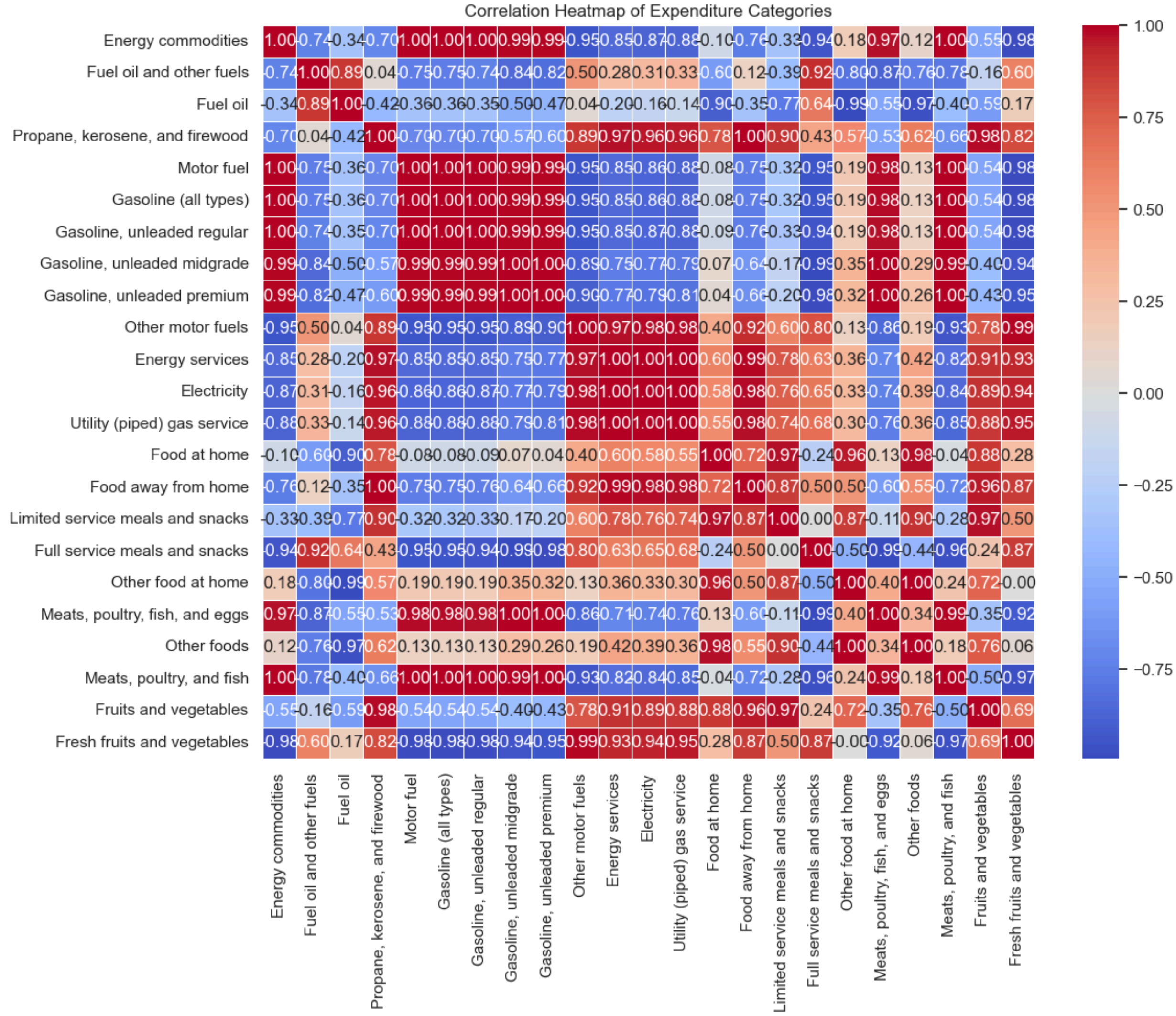
```
Food away from home                                0.866025
Limited service meals and snacks                   1.000000
Full service meals and snacks                      0.000000
Other food at home                                 0.866025
Meats, poultry, fish, and eggs                    -0.114708
Other foods                                        0.896258
Meats, poultry, and fish                          -0.277350
Fruits and vegetables                              0.970725
Fresh fruits and vegetables                        0.500000

                                  Full service meals and snacks  \
Energy commodities                                    -0.942718
Fuel oil and other fuels                               0.920864
Fuel oil                                               0.637927
Propane, kerosene, and firewood                        0.427121
Motor fuel                                            -0.946883
Gasoline (all types)                                  -0.946883
Gasoline, unleaded regular                            -0.944911
Gasoline, unleaded midgrade                           -0.985887
Gasoline, unleaded premium                            -0.979864
Other motor fuels                                      0.796288
Energy services                                        0.628619
Electricity                                            0.654654
Utility (piped) gas service                            0.675845
Food at home                                          -0.240192
Food away from home                                    0.500000
Limited service meals and snacks                       0.000000
Full service meals and snacks                          1.000000
Other food at home                                    -0.500000
Meats, poultry, fish, and eggs                        -0.993399
Other foods                                           -0.443533
Meats, poultry, and fish                              -0.960769
Fruits and vegetables                                  0.240192
Fresh fruits and vegetables                            0.866025

                                  Other food at home  \
Energy commodities                      1.824616e-01
Fuel oil and other fuels               -7.980819e-01
Fuel oil                               -9.858870e-01
Propane, kerosene, and firewood         5.694948e-01
Motor fuel                              1.949465e-01
Gasoline (all types)                    1.949465e-01
Gasoline, unleaded regular              1.889822e-01
Gasoline, unleaded midgrade             3.479601e-01
Gasoline, unleaded premium              3.170147e-01
Other motor fuels                       1.257297e-01
Energy services                         3.592106e-01
Electricity                             3.273268e-01
Utility (piped) gas service             3.003757e-01
Food at home                            9.607689e-01
Food away from home                     5.000000e-01
Limited service meals and snacks        8.660254e-01
Full service meals and snacks          -5.000000e-01
Other food at home                      1.000000e+00
Meats, poultry, fish, and eggs          3.973597e-01
Other foods                             9.979487e-01
Meats, poultry, and fish                2.401922e-01
Fruits and vegetables                   7.205767e-01
Fresh fruits and vegetables            -1.001543e-16
```

```
                                   Meats, poultry, fish, and eggs  Other foods  \
Energy commodities                                       0.974761     0.119144
Fuel oil and other fuels                                -0.870062    -0.757871
Fuel oil                                                -0.545380    -0.973147
Propane, kerosene, and firewood                         -0.528020     0.620949
Motor fuel                                               0.977520     0.131756
Gasoline (all types)                                     0.977520     0.131756
Gasoline, unleaded regular                               0.976221     0.125730
Gasoline, unleaded midgrade                              0.998583     0.287228
Gasoline, unleaded premium                               0.996299     0.255648
Other motor fuels                                       -0.860421     0.188982
Energy services                                         -0.713679     0.418219
Electricity                                             -0.737043     0.387147
Utility (piped) gas service                             -0.755929     0.360822
Food at home                                             0.127257     0.976554
Food away from home                                     -0.596040     0.554416
Limited service meals and snacks                        -0.114708     0.896258
Full service meals and snacks                           -0.993399    -0.443533
Other food at home                                       0.397360     0.997949
Meats, poultry, fish, and eggs                           1.000000     0.337797
Other foods                                              0.337797     1.000000
Meats, poultry, and fish                                 0.986241     0.177555
Fruits and vegetables                                   -0.349957     0.763487
Fresh fruits and vegetables                             -0.917663     0.064018

                                   Meats, poultry, and fish  \
Energy commodities                                 0.998256
Fuel oil and other fuels                          -0.776603
Fuel oil                                          -0.399314
Propane, kerosene, and firewood                   -0.661143
Motor fuel                                         0.998926
Gasoline (all types)                               0.998926
Gasoline, unleaded regular                         0.998625
Gasoline, unleaded midgrade                        0.993641
Gasoline, unleaded premium                         0.996800
Other motor fuels                                 -0.932823
Energy services                                   -0.819656
Electricity                                       -0.838628
Utility (piped) gas service                       -0.853750
Food at home                                      -0.038462
Food away from home                               -0.720577
Limited service meals and snacks                  -0.277350
Full service meals and snacks                     -0.960769
Other food at home                                 0.240192
Meats, poultry, fish, and eggs                     0.986241
Other foods                                        0.177555
Meats, poultry, and fish                           1.000000
Fruits and vegetables                             -0.500000
Fresh fruits and vegetables                       -0.970725

                                   Fruits and vegetables  \
Energy commodities                             -0.550258
Fuel oil and other fuels                       -0.157287
Fuel oil                                       -0.594328
Propane, kerosene, and firewood                 0.980316
Motor fuel                                     -0.539598
Gasoline (all types)                           -0.539598
Gasoline, unleaded regular                     -0.544705
Gasoline, unleaded midgrade                    -0.399314
Gasoline, unleaded premium                     -0.429178
```

```
Other motor fuels                           0.778471
Energy services                             0.905936
Electricity                                 0.891042
Utility (piped) gas service                 0.877800
Food at home                                0.884615
Food away from home                         0.960769
Limited service meals and snacks            0.970725
Full service meals and snacks               0.240192
Other food at home                          0.720577
Meats, poultry, fish, and eggs             -0.349957
Other foods                                 0.763487
Meats, poultry, and fish                   -0.500000
Fruits and vegetables                       1.000000
Fresh fruits and vegetables                 0.693375

                                  Fresh fruits and vegetables
Energy commodities                               -9.832130e-01
Fuel oil and other fuels                          6.025490e-01
Fuel oil                                          1.674124e-01
Propane, kerosene, and firewood                   8.219949e-01
Motor fuel                                       -9.808139e-01
Gasoline (all types)                             -9.808139e-01
Gasoline, unleaded regular                       -9.819805e-01
Gasoline, unleaded midgrade                      -9.375093e-01
Gasoline, unleaded premium                       -9.484206e-01
Other motor fuels                                 9.920645e-01
Energy services                                   9.332565e-01
Electricity                                       9.449112e-01
Utility (piped) gas service                       9.538210e-01
Food at home                                      2.773501e-01
Food away from home                               8.660254e-01
Limited service meals and snacks                  5.000000e-01
Full service meals and snacks                     8.660254e-01
Other food at home                               -1.001543e-16
Meats, poultry, fish, and eggs                   -9.176629e-01
Other foods                                       6.401844e-02
Meats, poultry, and fish                         -9.707253e-01
Fruits and vegetables                             6.933752e-01
Fresh fruits and vegetables                       1.000000e+00

[23 rows x 23 columns]
```

In [109…

```python
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation matrix
correlation_matrix = df.corr()

# Create heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Heatmap of Expenditure Categories')
plt.xticks(rotation=90)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```



Key Observations:

1. Energy Sector Correlations: The heatmap shows a very highly positive correlation among different types of motor fuels and gasoline categories (regular, midgrade, premium). This suggests that price fluctuations in these subcategories are tightly linked, likely due to common underlying factors such as global oil prices, taxation, and refining costs. For instance, correlations close to 1.00 between regular and premium gasoline types indicate that any change in the price of one is almost perfectly mirrored by the other.
2. Negative Correlations Between Energy and Food: Notably, there are significant negative correlations between motor fuels and food categories like fresh fruits and vegetables. This pattern suggests that increases in fuel prices might lead to higher transportation costs, which could negatively affect the prices of perishable goods, making them more expensive and possibly reducing their consumption.
3. Utility Services: Electricity and piped gas services exhibit strong positive correlations with each other but show negative correlations with motor fuels. This indicates divergent pricing factors affecting these services compared to those affecting fuel prices, possibly due to different regulatory and market dynamics.