# Exploring Food Price Variability Across Cities using Principal Component Analysis (PCA) Approach

**By Meenakshi Sharad Sethi**

Principal Component Analysis (PCA) is a powerful statistical technique commonly used for dimensionality reduction and exploratory data analysis. In this assignment, we applied PCA to explore the trends in food prices across different cities in the year 1973. By transforming the original variables into a set of orthogonal principal components, PCA helps us identify the underlying patterns and relationships in the data.

The assignment involves conducting principal component analysis (PCA) on a food prices dataset to extract meaningful insights. Additionally, it explores the common goals and differences between PCA and Confirmatory Factor Analysis (CFA), as well as the distinction between Linear Discriminant Analysis (LDA) and PCA in dimensionality reduction. Finally, there's an opportunity for bonus points by leveraging PCA components for building a machine learning model.

## Loading necessary libraries

In [17]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')
pd.set_option("display.max_columns", 60)
pd.set_option('display.max_rows', 50)
pd.set_option('display.width', 1000)
```

In [18]:
```
#### Loading the dataset

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

Out[18]:

| | City | Bread | Hamburger | Butter | Apples | Tomatoes |
|---|---|---|---|---|---|---|
| **0** | Anchorage | 70.9 | 135.6 | 155.00 | 63.9 | 100.1 |
| **1** | Atlanta | 36.4 | 111.5 | 144.30 | 53.9 | 95.9 |
| **2** | Baltimore | 28.9 | 108.8 | 151.00 | 47.5 | 104.5 |
| **3** | Boston | 43.2 | 119.3 | 142.00 | 41.1 | 96.5 |
| **4** | Buffalo | 34.5 | 109.9 | 124.80 | 35.6 | 75.9 |
| **5** | Chicago | 37.1 | 107.5 | 145.40 | 65.1 | 94.2 |
| **6** | Cincinnati | 37.1 | 118.1 | 149.60 | 45.6 | 90.8 |
| **7** | Cleveland | 38.5 | 107.7 | 142.70 | 50.3 | 83.2 |
| **8** | Dallas | 35.5 | 116.8 | 142.50 | 62.4 | 90.7 |
| **9** | Detroit | 40.8 | 108.8 | 140.10 | 39.7 | 96.1 |

| | City | Bread | Hamburger | Butter | Apples | Tomatoes |
|---|---|---|---|---|---|---|
| **10** | Honolulu | 50.9 | 131.7 | 154.40 | 65.0 | 93.9 |
| **11** | Houston | 35.1 | 102.3 | 150.30 | 59.3 | 84.5 |
| **12** | Kansas City | 35.1 | 99.8 | 162.30 | 42.6 | 87.9 |
| **13** | Los Angeles | 36.9 | 96.2 | 140.40 | 54.7 | 79.3 |
| **14** | Milwaukee | 33.3 | 109.1 | 123.20 | 57.7 | 87.7 |
| **15** | Minneapolis | 32.5 | 116.7 | 135.10 | 48.0 | 89.1 |
| **16** | New York | 42.7 | 130.8 | 148.70 | 47.6 | 92.1 |
| **17** | Philadelphia | 42.9 | 126.9 | 153.80 | 51.9 | 101.5 |
| **18** | Pittsburgh | 36.9 | 115.4 | 138.90 | 43.8 | 91.9 |
| **19** | St. Louis | 36.9 | 109.8 | 140.00 | 46.7 | 79.0 |
| **20** | San Diego | 32.5 | 84.5 | 145.90 | 48.5 | 82.3 |
| **21** | San Francisco | 40.0 | 104.6 | 139.10 | 59.2 | 81.9 |
| **22** | Seattle | 32.2 | 105.4 | 136.80 | 54.0 | 88.6 |
| **23** | Washington | 31.8 | 116.7 | 154.81 | 57.6 | 86.6 |

In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   City       24 non-null     object 
 1   Bread      24 non-null     float64
 2   Hamburger  24 non-null     float64
 3   Butter     24 non-null     float64
 4   Apples     24 non-null     float64
 5   Tomatoes   24 non-null     float64
dtypes: float64(5), object(1)
memory usage: 1.2+ KB
```

In [20]:
```
# Statitical summary for numerical features
df.describe()
```

Out[20]:

| | Bread | Hamburger | Butter | Apples | Tomatoes |
|---|---|---|---|---|---|
| **count** | 24.000000 | 24.000000 | 24.000000 | 24.0000 | 24.000000 |
| **mean** | 38.441667 | 112.245833 | 144.212917 | 51.7375 | 89.758333 |
| **std** | 8.366544 | 11.633160 | 9.226894 | 8.3590 | 7.398879 |
| **min** | 28.900000 | 84.500000 | 123.200000 | 35.6000 | 75.900000 |
| **25%** | 34.200000 | 106.975000 | 139.775000 | 46.4250 | 84.175000 |
| **50%** | 36.900000 | 109.850000 | 143.500000 | 51.1000 | 89.900000 |
| **75%** | 40.200000 | 117.125000 | 150.475000 | 58.0750 | 94.625000 |
| **max** | 70.900000 | 135.600000 | 162.300000 | 65.1000 | 104.500000 |

In [21]:

```
# Display unique values for all columns in the dataset
cols = df.columns

# for each column
for col in cols:
    print(col)

    # get a list of unique values
    unique = df[col].unique()
    print(unique, '\n===================================\n\n')
```

```
City
['Anchorage' 'Atlanta' 'Baltimore' 'Boston' 'Buffalo' 'Chicago'
 'Cincinnati' 'Cleveland' 'Dallas' 'Detroit' 'Honolulu' 'Houston'
 'Kansas City' 'Los Angeles' 'Milwaukee' 'Minneapolis' 'New York'
 'Philadelphia' 'Pittsburgh' 'St. Louis' 'San Diego' 'San Francisco'
 'Seattle' 'Washington']
===================================


Bread
[70.9 36.4 28.9 43.2 34.5 37.1 38.5 35.5 40.8 50.9 35.1 36.9 33.3 32.5
 42.7 42.9 40.  32.2 31.8]
===================================


Hamburger
[135.6 111.5 108.8 119.3 109.9 107.5 118.1 107.7 116.8 131.7 102.3  99.8
  96.2 109.1 116.7 130.8 126.9 115.4 109.8  84.5 104.6 105.4]
===================================


Butter
[155.   144.3  151.   142.   124.8  145.4  149.6  142.7  142.5  140.1
 154.4  150.3  162.3  140.4  123.2  135.1  148.7  153.8  138.9  140.
 145.9  139.1  136.8  154.81]
===================================


Apples
[63.9 53.9 47.5 41.1 35.6 65.1 45.6 50.3 62.4 39.7 65.  59.3 42.6 54.7
 57.7 48.  47.6 51.9 43.8 46.7 48.5 59.2 54.  57.6]
===================================


Tomatoes
[100.1  95.9 104.5  96.5  75.9  94.2  90.8  83.2  90.7  96.1  93.9  84.5
  87.9  79.3  87.7  89.1  92.1 101.5  91.9  79.   82.3  81.9  88.6  86.6]
===================================
```

Based on the statistical summary and unique values displayed for each column in the dataset, we observe the following:

- **City**: This column contains the names of different cities and is non-numerical. Since PCA works with numerical data, this column will not be included in the analysis.
- **Bread, Hamburger, Butter, Apples, Tomatoes**: These columns contain numerical data representing the prices of respective food items in different cities. They are suitable for PCA analysis.

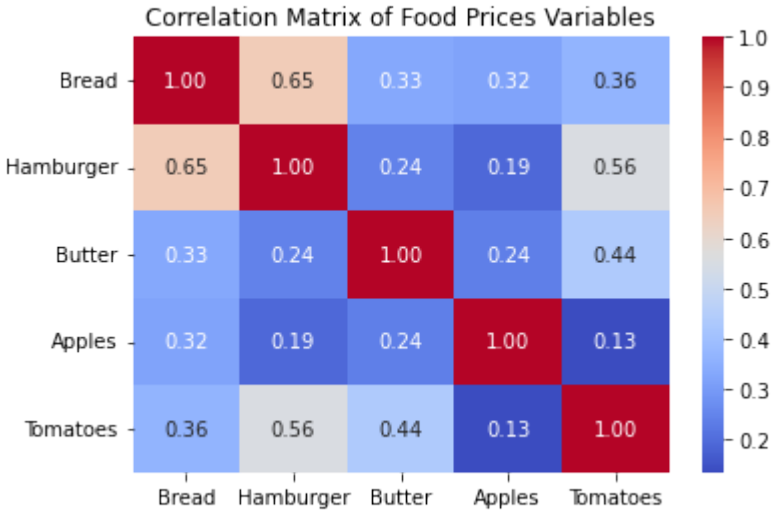
To proceed with PCA, we should exclude the "City" column and standardize the numerical variables before applying PCA.

## Principal Component Analysis.

In [22]:
```
# Exclude the "City" column
numerical_columns = df.drop(columns=['City'])
```

In [23]:
```
# Standardize the numerical variables
scaler = StandardScaler()
numerical_columns_scaled = scaler.fit_transform(numerical_columns)

# Convert the standardized numerical variables to a DataFrame
numerical_df_scaled = pd.DataFrame(numerical_columns_scaled, columns=numerical_colum
```

In [24]:
```
# Calculate the correlation matrix
correlation_matrix = df[['Bread', 'Hamburger', 'Butter', 'Apples', 'Tomatoes']].corr

# Check the correlation between variables
correlation_matrix = numerical_df_scaled.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title('Correlation Matrix of Food Prices Variables')
plt.show()
```




## Understanding the Correlation Matrix for PCA

The correlation matrix above shows the relationship between the prices of different food items (bread, hamburger, butter, apples, and tomatoes) in 1973 for various cities. Here's a breakdown of what the values in the matrix represent:

- **Positive correlation (values between 0 and 1)**: When one food item's price increases, the other item's price also tends to increase (stronger correlation with values closer to 1, weaker correlation with values closer to 0).
- **Negative correlation (values between -1 and 0)**: When one food item's price increases, the other item's price tends to decrease (stronger correlation with values closer to -1, weaker correlation with values closer to 0).
- **Zero correlation (value = 0)**: No relationship between the prices of the two food items.

For example, the value in the cell at the intersection of the "Bread" row and "Hamburger" column is 0.65, which indicates a positive correlation between the prices of bread and hamburger.

**How Correlation Matrix Relates to PCA**

The correlation matrix provided captures the linear relationships between the food prices. PCA leverages this information to identify a new set of uncorrelated variables (principal components) that explain most of the variance in the original data (food prices).

Each principal component represents a new direction in the data space that captures the maximum remaining variance. The correlation matrix helps PCA avoid redundant information by focusing on these principal components, which summarize the price movements across different food items. By analyzing the data using PCA, we can identify the underlying factors influencing food price variations across US cities in 1973. These principal components might represent factors like:

- Overall food price level: A component capturing the general trend of food prices being high or low across cities.
- Price disparity between staples and produce: A component that reflects the difference in price movements between staple foods (bread, hamburger) and produce (apples, tomatoes).

## PCA Analysis

In [27]:

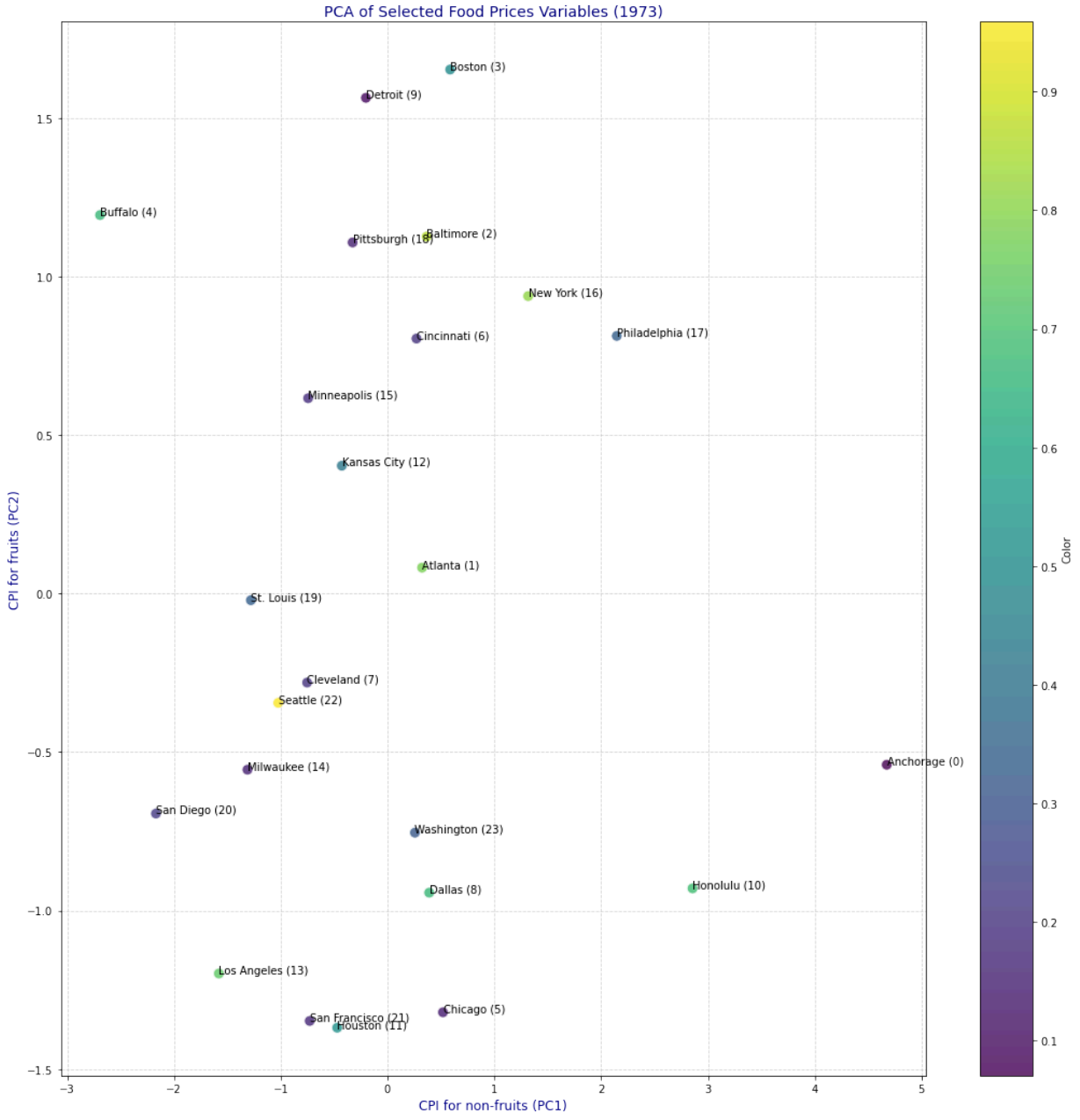
```
# Create PCA model
pca = PCA(n_components=2)
principal_components = pca.fit_transform(numerical_df_scaled)

# Identify explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_
print("Eigenvalues (Explained Variance Ratio)", explained_variance_ratio)

# Get the PCA components
pca_components = pca.components_

# Create a DataFrame to display the PCA components
pca_components_df = pd.DataFrame(pca_components, columns=numerical_columns.columns)

# Display the PCA components
print("PCA Components:")
print(pca_components_df)

# Set up colors for scatter plot
colors = np.random.rand(len(principal_components))
```

```
Eigenvalues (Explained Variance Ratio) [0.48787109 0.18591823]
PCA Components:
      Bread  Hamburger    Butter    Apples  Tomatoes
0  0.509927   0.520072  0.397311  0.290942  0.476442
1 -0.056496   0.277616 -0.099401 -0.876753  0.375714
```

In [28]:

```
# Visualize the PCA results with city names and indices
plt.figure(figsize=(18, 18))
plt.scatter(
    principal_components[:, 0],
    principal_components[:, 1],
    c=colors,
    alpha=0.8,
    edgecolors='w',  # Add white edges for better visibility
```

```
    cmap='viridis',  # Use Viridis colormap for better color contrast
    s=100  # Increase marker size for better visibility
)
for i, city in enumerate(df['City']):
    plt.text(principal_components[i, 0], principal_components[i, 1], f"{city} ({i})"
plt.title('PCA of Selected Food Prices Variables (1973)', fontsize=14, color='navy')
plt.xlabel('CPI for non-fruits (PC1)', fontsize=12, color='darkblue')  # Updated x-a
plt.ylabel('CPI for fruits (PC2)', fontsize=12, color='darkblue')  # Updated y-axis
plt.xticks(fontsize=10, color='black')
plt.yticks(fontsize=10, color='black')
plt.grid(True, linestyle='--', alpha=0.5)  # Add grid lines with dashed style
plt.colorbar(label='Color')
plt.show()
```




- **PCA Analysis:** Using the PCA algorithm with two principal components, we obtained insightful results regarding the variability in food prices. The explained variance ratio revealed that the first principal component (PCA 1) captured approximately 48.79% of the variance, while the second principal component (PCA 2) explained around 18.59% of the variance. Together, these two components accounted for approximately 67.38% of the total variance in the data.

- **Interpretation of Results:**

- Analyzing the PCA components provided valuable insights into the underlying structure of the data and the contributions of the original variables.

- In PCA 1:
  - We observed positive correlations for variables such as 'Bread', 'Hamburger', 'Butter', and 'Tomatoes', suggesting that these food items share similar price trends across different cities.
  - The relatively lower positive correlation of 'Apples' indicates a slight deviation from the overall trend.
- In PCA 2:
  - We observed a positive correlation between 'Hamburger' and 'Tomatoes', indicating a similar price trend for these food items.
  - However, 'Apples' exhibited a strong negative correlation in PCA 2, suggesting a contrasting price trend compared to 'Hamburger' and 'Tomatoes'.

--- End ---