

2211cab01f88 52-28

Mini Project on Employees Office Data

Dataset : OfficeDataProject.csv

Source :<https://www.data.gov.in/resource/wholesale-price-index-base-year-2011-12-upto-may-2017>

Dataset Summary: WPI (Wholesale Price Index) 2011–2017

Dataset Overview

Name: WPI-2011-12-May2017.csv
Shape: 869 rows x 76 columns

Period Covered: April 2011 – May 2017 (74 months)
Purpose: Measures inflation trends in wholesale markets in India.

Columns

COMMODITIES

Name of commodity or group (e.g., All Commodities, Primary Articles, Wheat, Paddy, Jowar).

Includes aggregate groups (like "All Commodities") and detailed items (like "Paddy").

Weight

Relative importance in the overall WPI basket.

For example:

All Commodities = 100

Primary Articles = 22.6

Sub-items like Paddy, Wheat, Bajra have smaller weights.

Monthly Columns (Apr-11 ... May-17)

Index values for each month.

Base year = 2011-12 (WPI = 100 in that year).

Example:

Apr-11: 97.1 (slightly below base year)

May-17: 112.8 (price increase since 2011).

Dataset Highlights

Commodities Covered: 869 (very detailed breakdown).

Time Period: 6 years 2 months (April 2011 – May 2017).

Missing Values: Small number of NaNs (1-4 per column); dataset is clean overall.

```
In [1]: import pandas as pd

df = pd.read_csv("WPI-2011-12-May2017.csv", encoding="latin1")

print("Shape:", df.shape)
print("Columns:", df.columns.tolist())
print("Missing values:", df.isnull().sum())
print("\nSample data:\n", df.head(10))

Shape: (869, 76)
Columns: ['COMMODITIES', 'Weight', 'Apr-11', 'May-11', 'Jun-11', 'Jul-11', 'Aug-11', 'Sep-11', 'Oct-11', 'Nov-11', 'Dec-11', 'Jan-12', 'Feb-12', 'Mar-12', 'Apr-12', 'May-12', 'Jun-12', 'Jul-12', 'Aug-12', 'Sep-12', 'Oct-12', 'Nov-12', 'Dec-12', 'Jan-13', 'Feb-13', 'Mar-13', 'Apr-13', 'May-13', 'Jun-13', 'Jul-13', 'Aug-13', 'Sep-13', 'Oct-13', 'Nov-13', 'Dec-13', 'Jan-14', 'Feb-14', 'Mar-14', 'Apr-14', 'May-14', 'Jun-14', 'Jul-14', 'Aug-14', 'Sep-14', 'Oct-14', 'Nov-14', 'Dec-14', 'Jan-15', 'Feb-15', 'Mar-15', 'Apr-15', 'May-15', 'Jun-15', 'Jul-15', 'Aug-15', 'Sep-15', 'Oct-15', 'Nov-15', 'Dec-15', 'Jan-16', 'Feb-16', 'Mar-16', 'Apr-16', 'May-16', 'Jun-16', 'Jul-16', 'Aug-16', 'Sep-16', 'Oct-16', 'Nov-16', 'Dec-16', 'Jan-17', 'Feb-17', 'Mar-17', 'Apr-17', 'May-17']
Missing values:
COMMODITIES    0
Weight          1
Apr-11          1
May-11          4
Jun-11          3
Jan-12          2
Feb-12          2
Mar-12          2
Apr-12          1
May-12          6
Length: 76, dtype: int64

Sample data:
COMMODITIES  Weight  Apr-11  May-11  Jun-11  Jul-11  \
0  All Commodities  100.0000  97.1   97.7   98.2   98.6
1  PRIMARY ARTICLES  22.6176  95.2   95.3   91.2   99.2
2  (A). FOOD ARTICLES  15.2555  94.1   95.0   91.5   99.1
3  a. FOOD GRAINS (CEREALS+PULSES)  3.68238  97.5   98.6   98.7   99.5
4      a1. CEREALS  2.82378  98.5   99.7   99.7   100.5
5      Paddy  1.42062  97.5   98.9   99.7   100.1
6      Wheat  1.02823  101.1  101.0  100.4  101.2
7      Jowar  0.06734  84.1   91.9   96.3  100.4
8      Bajra  0.08627  97.7  100.6   95.3  101.4
9      Maize  0.18927  98.2  100.8   99.5   99.1

Aug-11  Sep-11  Oct-11  Nov-11  ...  Aug-16  Sep-16  Oct-16  Nov-16  \
0  99.2  100.1  100.5  100.8  ...  111.2  111.6  111.5  111.9
1  99.8  101.6  101.9  101.3  ...  131.4  130.6  129.1  128.8
2  101.8  103.1  104.2  102.5  ...  142.6  141.9  141.9  142.0
3  100.1  101.9  100.8  99.9  ...  153.6  153.0  155.2  158.2
4  100.8  100.7  99.8  98.9  ...  142.2  142.6  142.7  145.5
5  101.3  101.3  101.5  100.8  ...  144.6  144.7  145.1  145.0
6  99.6  98.9  98.0  97.6  ...  139.1  140.0  140.8  147.9
7  102.3  102.9  107.9  98.2  ...  121.4  124.0  121.9  123.2
8  102.3  101.4  94.8  94.9  ...  159.3  155.1  148.1  151.0
9  99.8  100.8  94.2  93.2  ...  135.3  135.8  136.6  137.7

Dec-11  Jan-12  Feb-12  Mar-12  Apr-12  May-12
0  111.7  112.6  113.0  113.2  113.2  113.8
1  126.8  126.7  127.0  127.1  128.4  126.5
2  137.6  136.0  136.6  137.6  139.4  137.7
3  157.5  153.0  150.2  146.8  146.6  144.8
4  146.7  146.3  147.0  145.2  144.5  142.2
5  144.2  143.8  146.4  147.7  147.6  147.9
6  136.8  137.1  149.3  142.7  141.0  137.5
7  122.4  132.3  136.0  132.1  131.6  130.6
8  154.7  151.9  151.3  152.6  153.8  153.5
9  134.0  134.6  135.6  135.2  135.5  134.2

[10 rows x 76 columns]
```

```
In [14]: # Make dataset into long format
df_long = df.melt(id_vars=["COMMODITIES", "Weight"],
                  var_name="Date", value_name="Index")

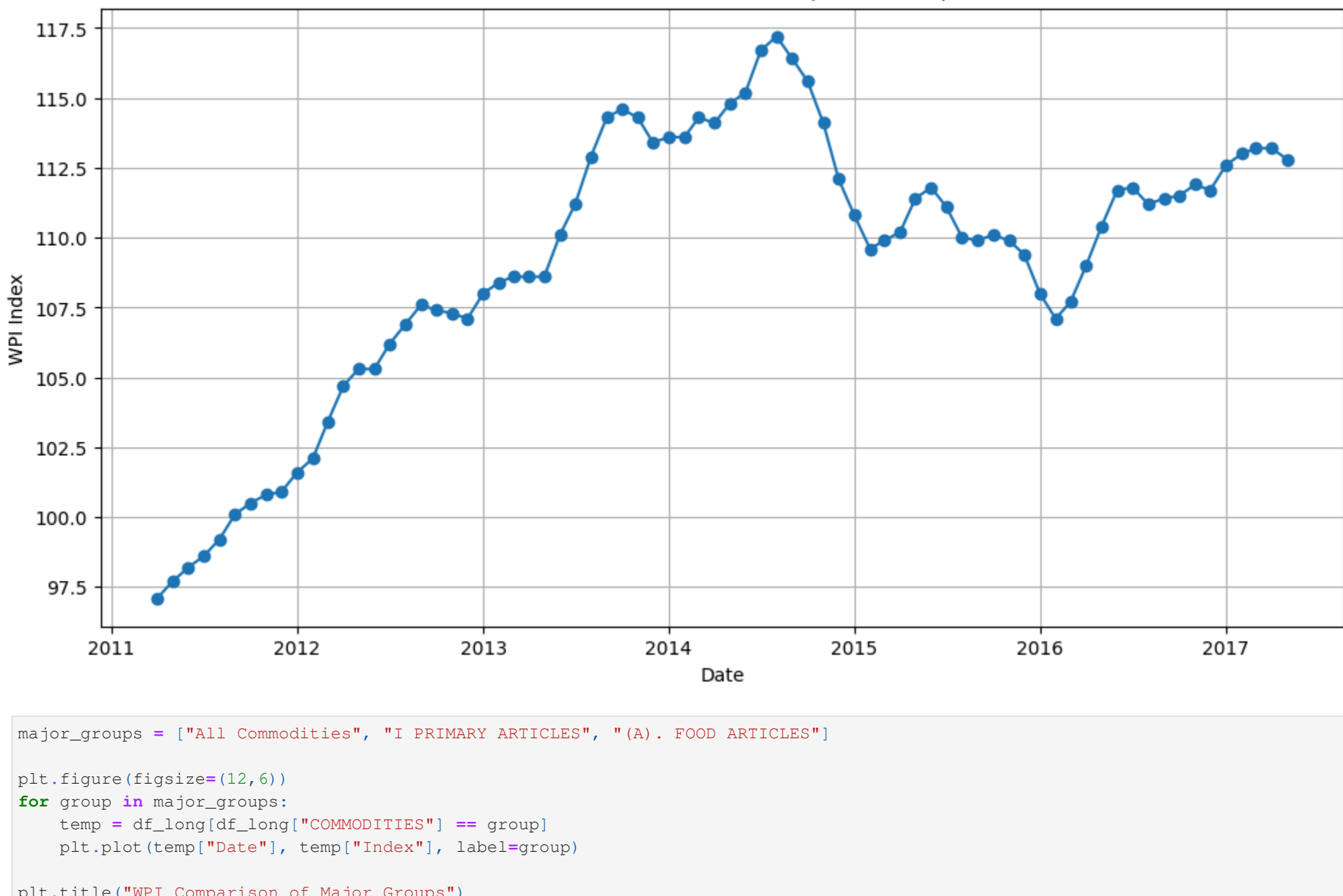
# Convert Date column into datetime format
df_long["Date"] = pd.to_datetime(df_long["Date"], format="%b-%y")

print(df_long.head())

COMMODITIES  Weight  Date  Index
0  All Commodities  100.0000  2011-04-01  97.1
1  PRIMARY ARTICLES  22.6176  2011-04-01  95.2
2  (A). FOOD ARTICLES  15.2555  2011-04-01  94.1
3  a. FOOD GRAINS (CEREALS+PULSES)  3.68238  2011-04-01  97.5
4  a1. CEREALS  2.82378  2011-04-01  98.5
```

```
In [16]: # All items = df_long[df_long["COMMODITIES"] == "All Commodities"]

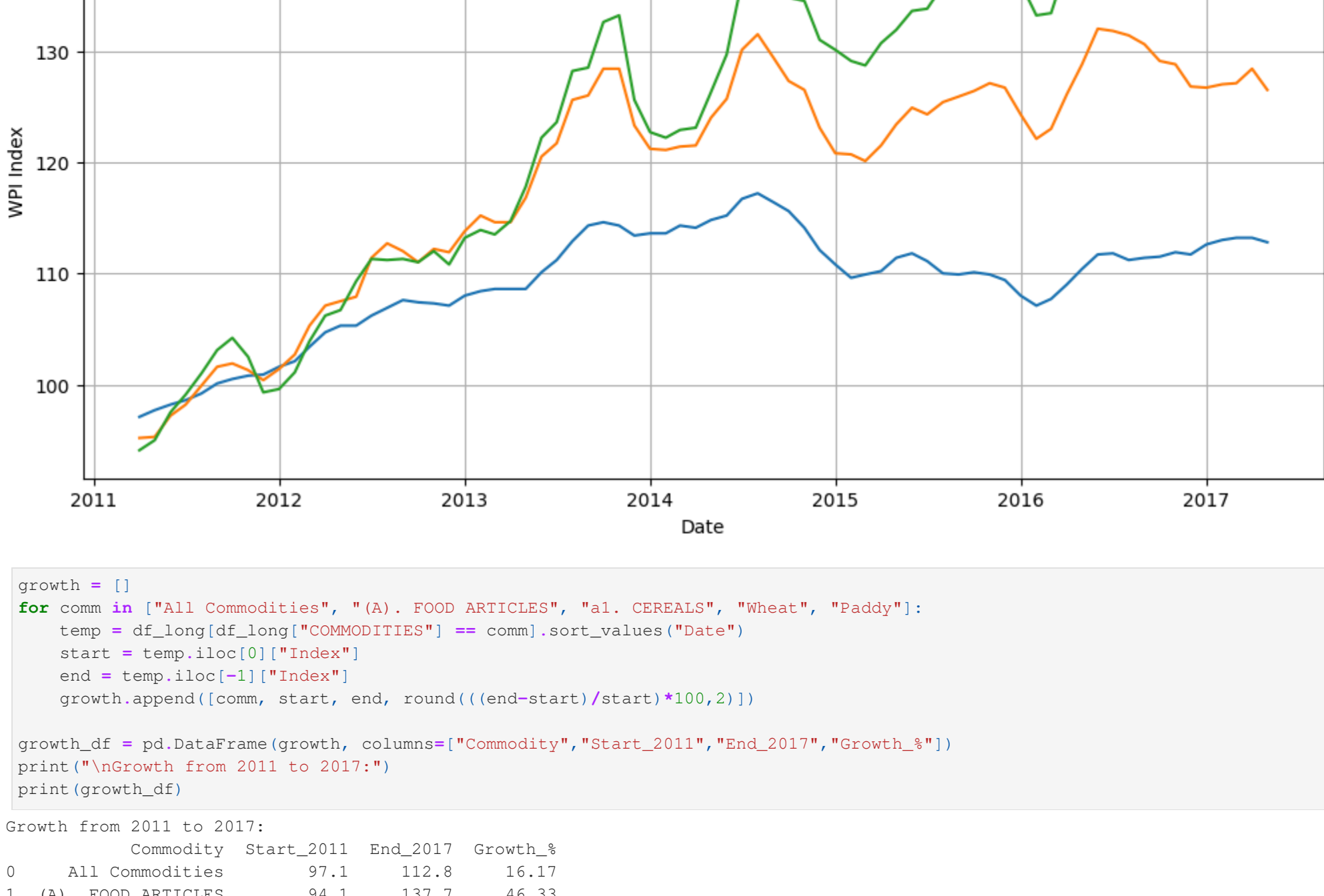
plt.figure(figsize=(12,6))
plt.plot(all_comm["Date"], all_comm["Index"], markers="o")
plt.title("WPI Trend: All Commodities (2011-2017)")
plt.xlabel("Date")
plt.ylabel("WPI Index")
plt.legend()
plt.grid(True)
plt.show()
```



```
In [17]: # Major groups = ["All Commodities", "I PRIMARY ARTICLES", "(A). FOOD ARTICLES"]

plt.figure(figsize=(12,6))
for group in major_groups:
    temp = df_long[df_long["COMMODITIES"] == group]
    plt.plot(temp["Date"], temp["Index"], label=group)

plt.title("WPI Comparison of Major Groups")
plt.xlabel("Date")
plt.ylabel("WPI Index")
plt.legend()
plt.grid(True)
plt.show()
```



```
In [18]: # Growth = []
for comm in ["All Commodities", "(A). FOOD ARTICLES", "a1. CEREALS", "Wheat", "Paddy"]:
    temp = df_long[df_long["COMMODITIES"] == comm].sort_values("Date")
    start = temp.iloc[0]["Index"]
    end = temp.iloc[-1]["Index"]
    growth.append((comm, start, end, round((end-start)/(start*100,2)))

growth_df = pd.DataFrame(growth, columns=["Commodity", "Start_2011", "End_2017", "Growth_%"])
print("Growth from 2011 to 2017:")
print(growth_df)

Commodity  Start_2011  End_2017  Growth_%
0  All Commodities      97.1    112.8      16.17
1  (A). FOOD ARTICLES     94.1    137.7      46.33
2      a1. CEREALS       98.5    143.2      45.38
3      Wheat            101.1    137.5      36.00
4      Paddy             97.5    147.9      51.69
```

```
In [19]: # Plot
plt.figure(figsize=(10,6))
plt.bar(growth_df["Commodity"], growth_df["Growth_%"])
plt.title("Growth % of Selected Commodities (2011-2017)")
plt.xlabel("Commodity")
plt.ylabel("Growth_%")
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



```
In [110]: subset = df_long[df_long["COMMODITIES"].isin(["Wheat", "Paddy", "Jowar", "Bajra"])]

plt.figure(figsize=(10,6))
subset.groupby(column="Index", by="COMMODITIES")
plt.title("Price Distribution (2011-2017)")
plt.xlabel("Date")
plt.ylabel("WPI Index")
plt.grid(True)
plt.show()

<Figure size 1000x600 with 6 Axes>
```

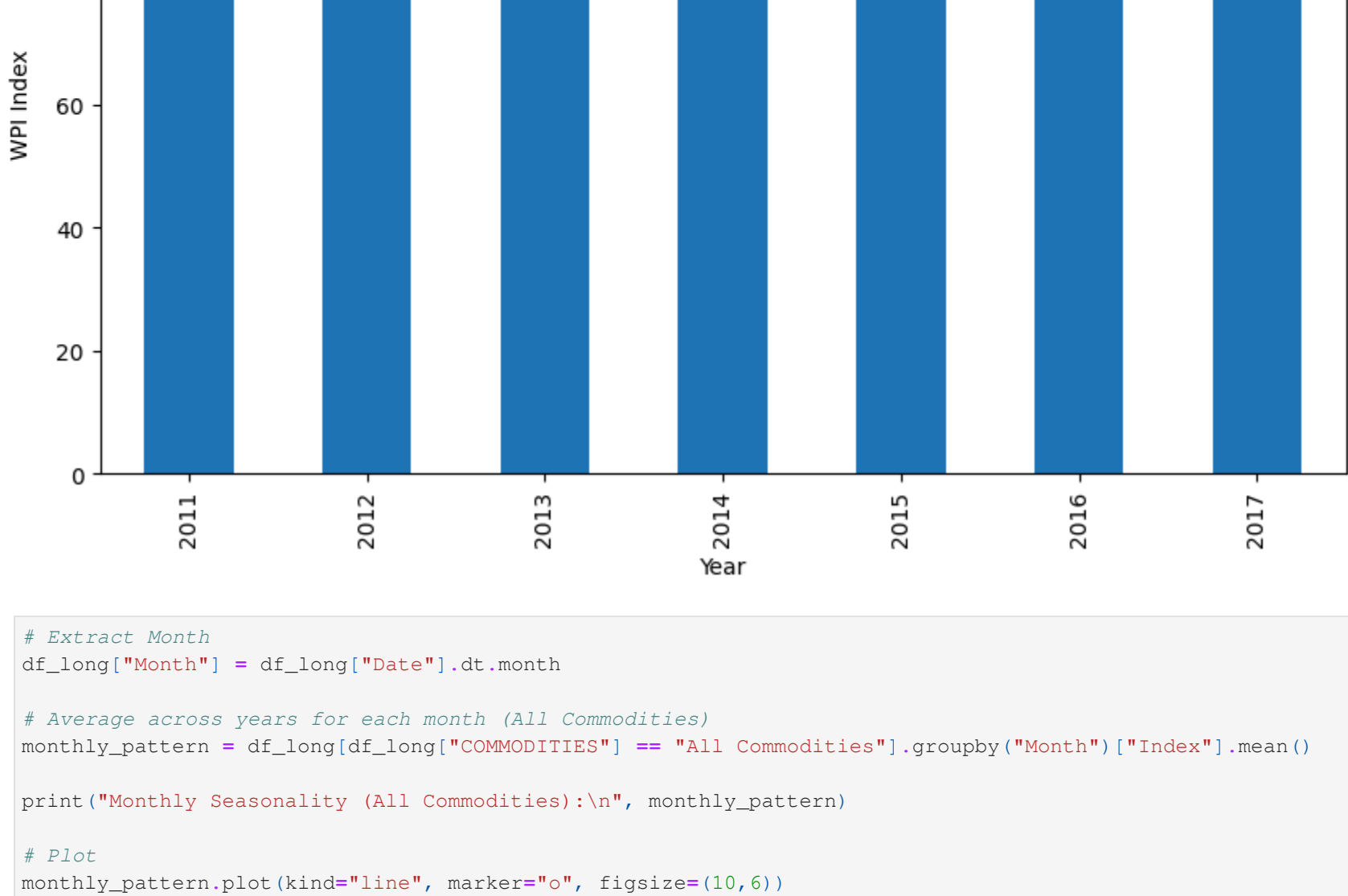


```
In [111]: # Extract Year from Date
df_long["Year"] = df_long["Date"].dt.year

# Average WPI per year for All Commodities
yearly_avg = df_long[df_long["COMMODITIES"] == "All Commodities"].groupby("Year")["Index"].mean()
print("Yearly Average WPI (All Commodities):\n", yearly_avg)

# Plot
yearly_avg.plot(kind="bar", figsize=(10,6))
plt.title("Yearly Average WPI (All Commodities)")
plt.xlabel("Year")
plt.ylabel("WPI Index")
plt.grid(True)
plt.show()

Yearly Average WPI (All Commodities):
Year
2011    99.233333
2012   105.408333
2013   111.893333
2014   114.808333
2015   110.341667
2016   110.283333
2017   112.860000
Name: Index, dtype: float64
```

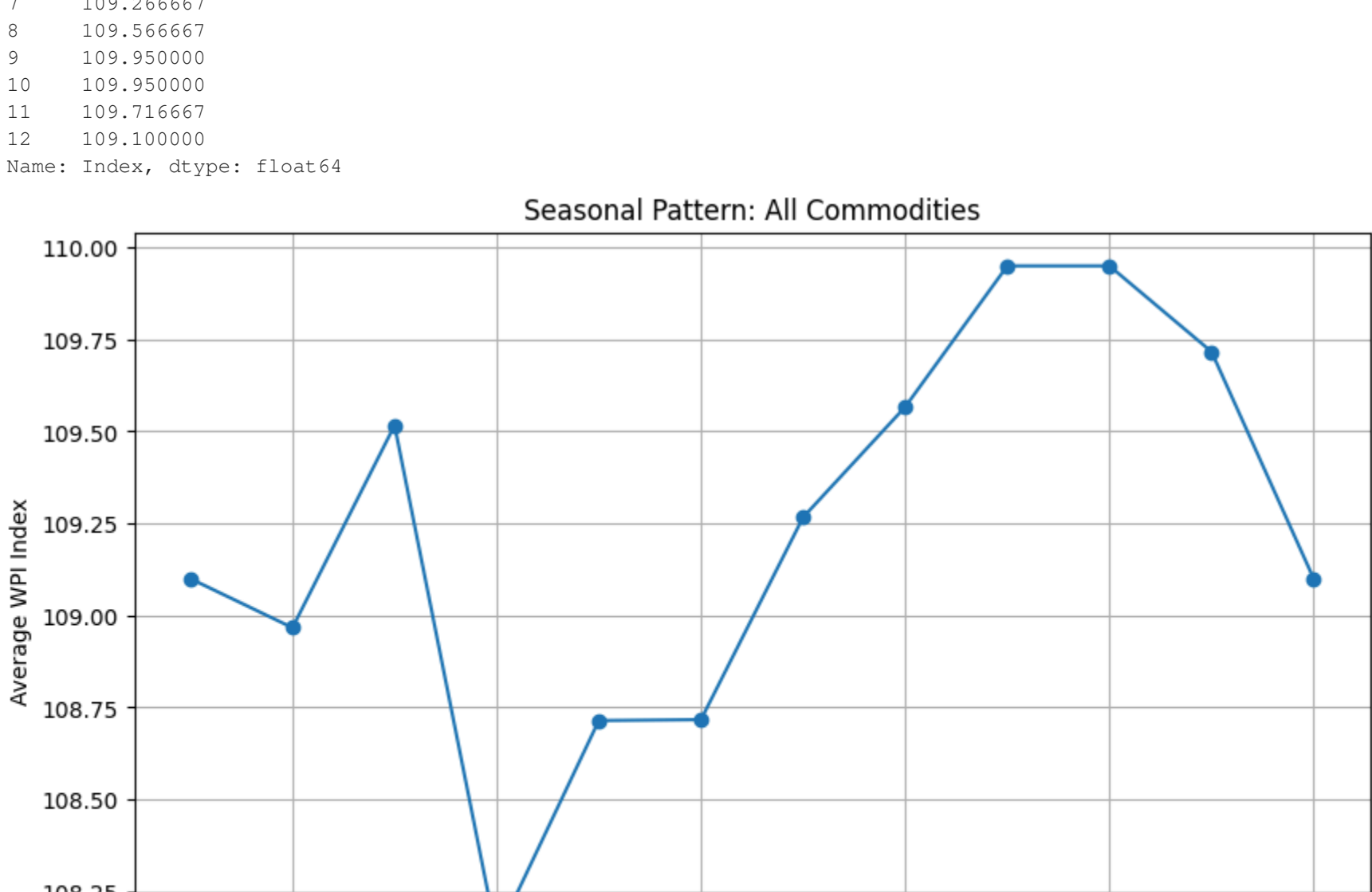


```
In [112]: # Extract Month
df_long["Month"] = df_long["Date"].dt.month

# Average across years for each month (All Commodities)
monthly_pattern = df_long[df_long["COMMODITIES"] == "All Commodities"].groupby("Month")["Index"].mean()
print("Monthly Seasonality (All Commodities):\n", monthly_pattern)

# Plot
monthly_pattern.plot(kind="line", marker="o", figsize=(10,6))
plt.title("Seasonal Pattern: All Commodities")
plt.xlabel("Month (1-Jan, ... 12-Dec)")
plt.ylabel("Average WPI Index")
plt.grid(True)
plt.show()

Monthly Seasonality (All Commodities):
Month
1    109.100000
2   108.966667
3   109.516667
4   108.128571
5   108.714286
6   108.746667
7   109.266667
8   109.566667
9   109.950000
10  109.950000
11  109.716667
12  109.100000
Name: Index, dtype: float64
```



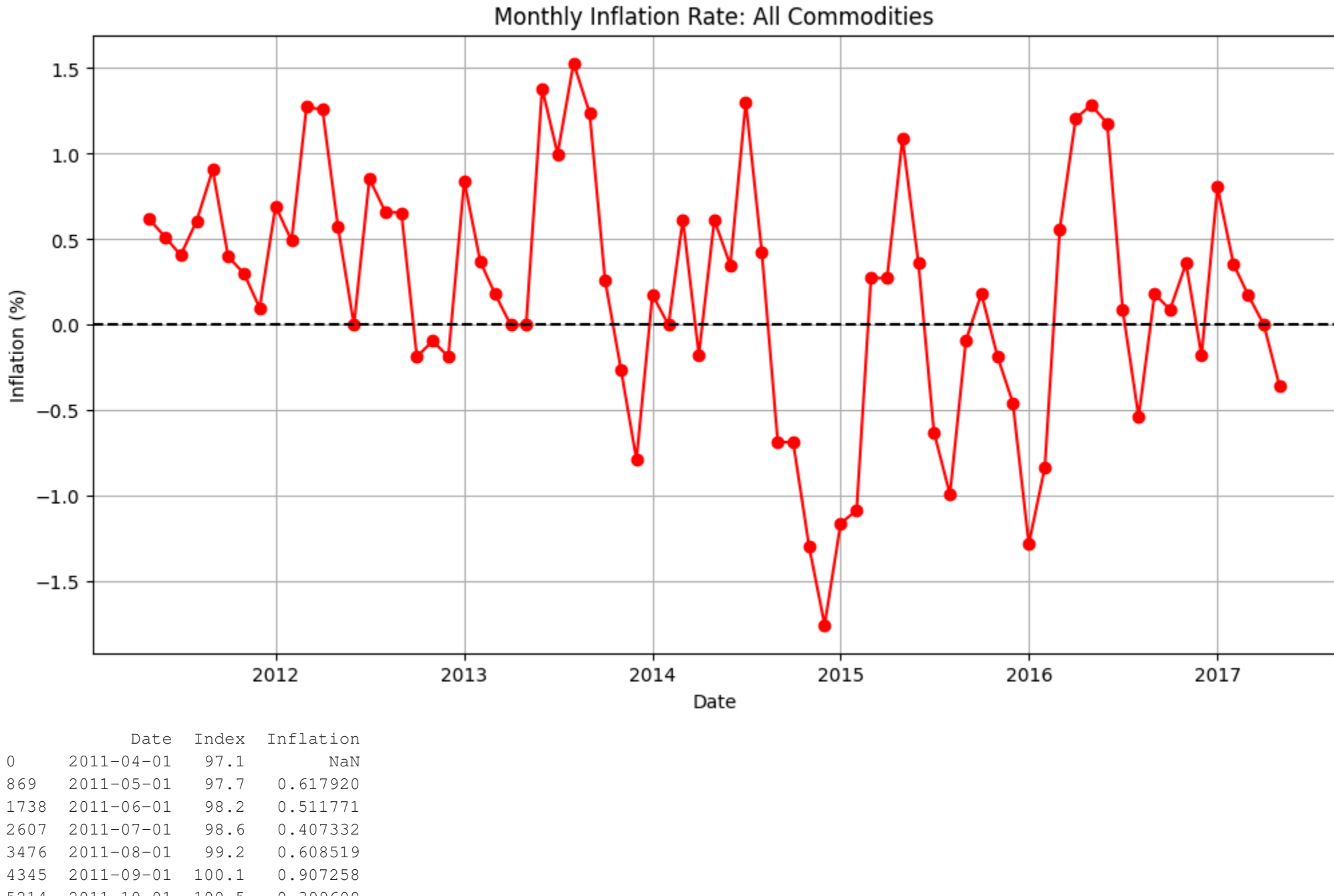
```
In [113]: # Month-to-month percentage change (All Commodities)
all_comm["Inflation"] = all_comm["Index"].pct_change() * 100

plt.figure(figsize=(12,6))
plt.plot(all_comm["Date"], all_comm["Inflation"], color="red", markers="o")
plt.title("Monthly Inflation Rate: All Commodities")
plt.xlabel("Date")
plt.ylabel("Inflation (%)")
plt.axhline(0, color="black", linestyle="--")
plt.grid(True)
plt.show()

print(all_comm[["Date", "Index", "Inflation"]].head(15))
```

```
C:\Users\J100\AppData\Local\Temp\ipykernel_12492\104019253.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of 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-copy
all_comm["Inflation"] = all_comm["Index"].pct_change() * 100
```

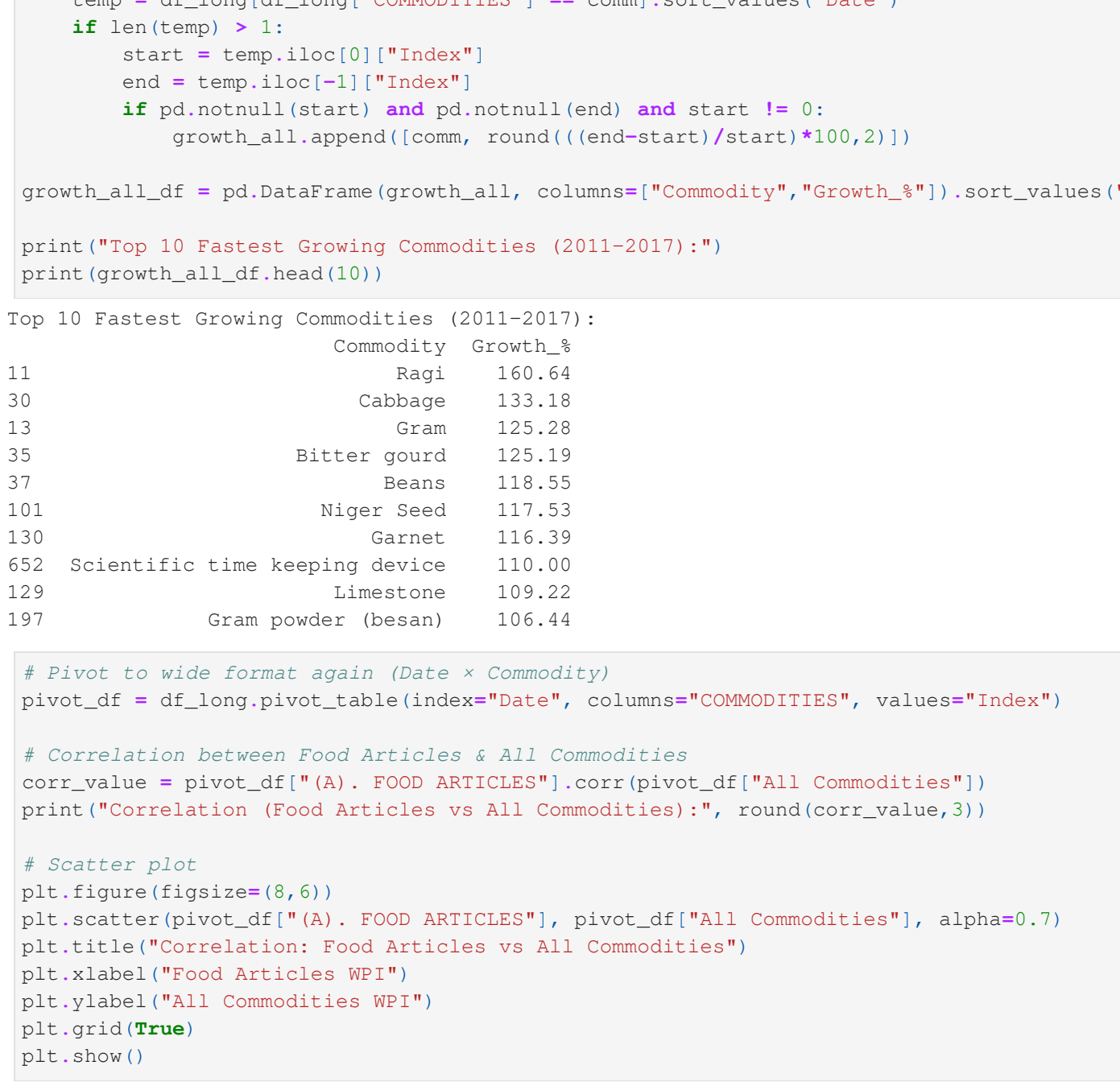


```
In [114]: # Pivot to wide format again (Date = Commodity)
growth_df = df_long.pivot_table(index="Date", columns="COMMODITIES", values="Index")

# Correlation between Food Articles & All Commodities
corr_value = pivot_df["(A). FOOD ARTICLES"].corr(pivot_df["All Commodities"])
print("Correlation (Food Articles vs All Commodities):", round(corr_value,3))

# Scatter plot
plt.figure(figsize=(8,6))
plt.scatter(pivot_df["(A). FOOD ARTICLES"], pivot_df["All Commodities"], alpha=0.7)
plt.title("Correlation: Food Articles vs All Commodities")
plt.xlabel("Food Articles WPI")
plt.ylabel("All Commodities WPI")
plt.grid(True)
plt.show()

Correlation (Food Articles vs All Commodities): 0.821
```



```
In [116]: import seaborn as sns

major = ["All Commodities", "I PRIMARY ARTICLES", "(A). FOOD ARTICLES", "a1. CEREALS", "Wheat", "Paddy"]
corr_matrix = pivot_df[major].corr()

plt.figure(figsize=(8,6))
sns.heatmap(corr_matrix, annot=True, cmap="magma", cbar=True, fmt=".2f")
plt.title("Correlation Heatmap (Major Commodities)")
plt.show()
```