AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

Big Data Analytics and Hadoop
24ASD511

A case study on
Food price analysis in India

Submitted By

Additi CB.PS.P2ASD24001
Sivapriya CB.PS.P2ASD24025

Dataset
Food prices India - Kaggle

Platform
Hive (on WSL)

Index

Abstract

Big Data Analytics is transforming how organizations extract insights from the massive data generated daily. Probability and Statistics (ProbStat) are crucial in this domain, offering the tools needed to analyze patterns and make informed predictions, vital for applications like market analysis and financial forecasting. This case study explores Indian food prices using Big Data techniques to reveal trends and insights. By applying advanced SQL methods such as window functions and grouping sets, we analyze commodity price changes across different regions and seasons. ProbStat's role in real-time analytics allows businesses to address uncertainties, leading to proactive strategy adjustments. The key outcomes include actionable insights for pricing strategy optimization. The study utilizes Apache Hive for processing and SQL for query execution, with visualizations highlighting the results. This work demonstrates the importance of Big Data Analytics and ProbStat in turning data into strategic assets, fostering innovation and efficiency in a data-driven world.

Introduction

This research focuses on the problem of typifying and forecasting the cash determinants within a specific segment of the economy – the Indian food marketplace. Price fluctuations, different commodities, and diverse geographical market areas of the country render these phenomena very complex and difficult to understand. An effective and complex supply chain management can be established only if adequate and relevant business intelligence is integrated. This study hopes to utilize the modern Big Data methodologies for in-depth analysis of the patterns of commodity prices and arm the relevant stakeholders with proper information.

The social relevance of this study is profound, as pricing affects the life of billions from farmers to consumers. Better understanding of price fluctuations can help stabilize the market and mitigate food waste while guaranteeing reasonable prices to be paid and profit earned. Reliable information on market pricing and tracking of volatile prices provided by this study helps narrow the resource gap and enhance the efficiency in the provision of basic food commodities. Improved endeavours will significantly increase food security and economic welfare, and therefore, this research is important for our society.

In order to gain these insights, the study uses tools like Apache Hive for data preprocessing and complex query execution and Matplotlib (Google CoLab) for visualizing the hive results. Commodity prices are analyzed using advanced SQL methods, like CTE's, window functions and grouping sets, to discover trends and correlations. The objective is to formulate insights that enable stakeholders to make decisions that improve pricing and market competitiveness. The study is designed in such a way that it helps companies to identify patterns and trends so they could better monitor and respond to changes, thus making the food market more stable and fair.

About the Tool

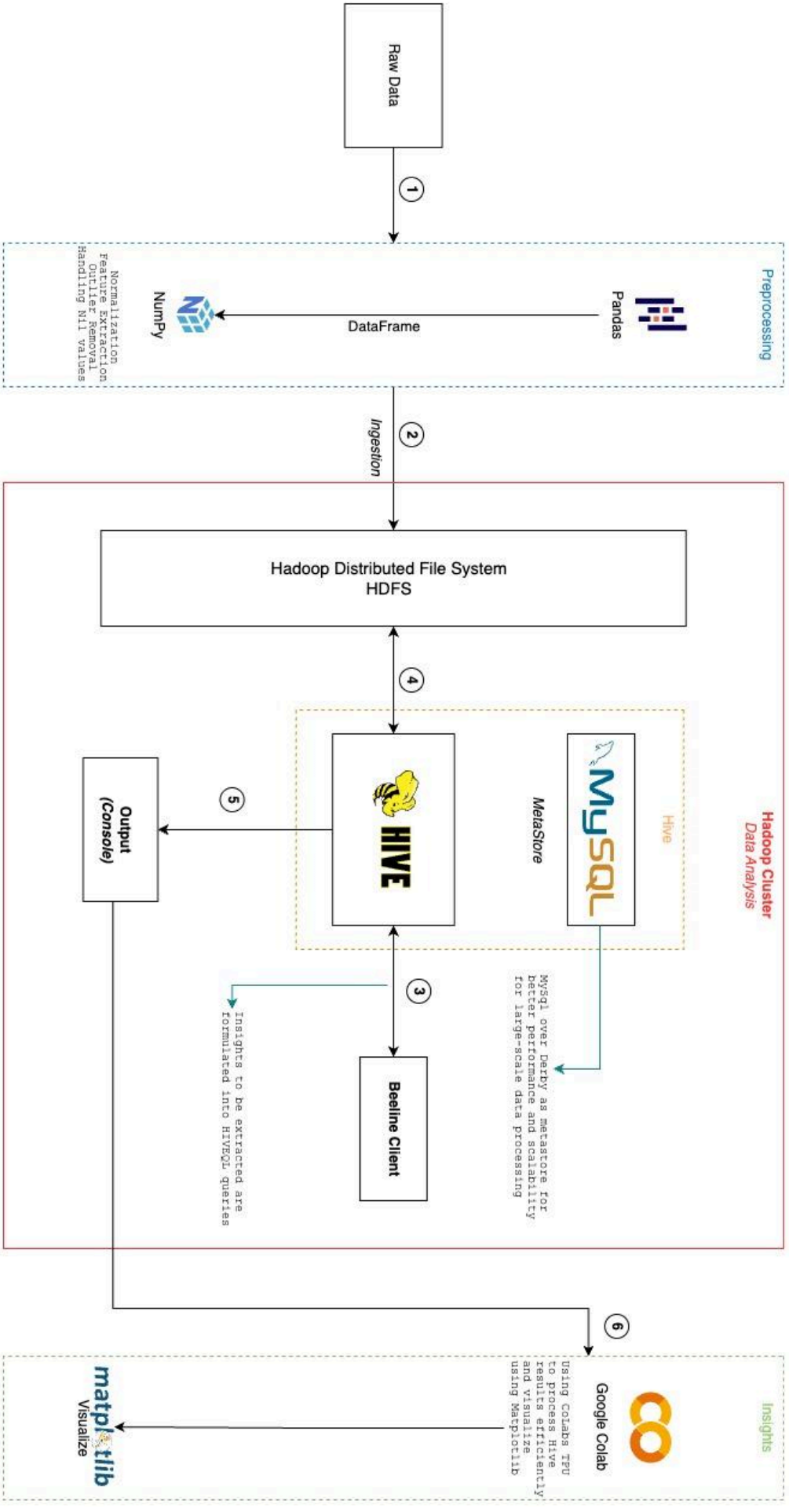
For this study, Apache Hive has been selected as the primary tool for data processing and analysis. The Apache Hive is a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale and facilitates reading, writing, and managing petabytes of data residing in distributed storage using SQL. Hive Metastore(HMS) provides a central repository of metadata that can easily be analyzed to make informed, data driven decisions, and therefore it is a critical component of many data lake architectures. Hive is built on top of Apache Hadoop and supports storage on S3, adls, gs etc though HDFS.

Hive's SQL-like query language, HiveQL, allows for complex data manipulations and aggregations, making it ideal for executing advanced SQL techniques like CTE's and window functions. These capabilities enable the extraction of meaningful insights from the dataset, such as identifying trends and correlations in commodity prices across different regions and seasons. Additionally, the use of Matplotlib complements Hive by transforming the query results into visualizations, providing a clear and intuitive representation of the data. This combination of tools ensures a comprehensive analysis, facilitating data-driven decision-making and enhancing the overall understanding of market dynamics.

Objective

The main purpose of this project is to analyze the factors that influence the fluctuations in food prices in the Indian market with the help of big data analysis. By using advanced SQL techniques in Apache Hive, this study aims to uncover trends, correlations and patterns in raw material prices in various aspects such as geography and seasonality. The knowledge gained can inform stakeholders about market conditions, optimize pricing strategies, and improve supply chain efficiency. Ultimately, the project aims to contribute to a more stable and fair food market by conferring implementable data-controlled knowledge.

Architecture Diagram



Pathway of Taking the Project

The completion of this project takes place in a stepwise manner, to convert raw data into insightful information. The process is a series of meticulous steps aimed to provide an accurate and relevant final analysis. Here is a summary of the major steps included:

- **Dataset Acquisition**
  A dataset containing Indian food price information was collected from Kaggle. This dataset acts as the basis for all later analysed data in terms of different commodities, geographical areas and time periods.

- **Preprocessing**
  Before diving into analysis, we meticulously prepare our raw datasets using a combination of Pandas and NumPy. This crucial first step involves normalizing values to comparable scales, extracting meaningful features that drive insights, identifying and removing statistical outliers that could skew results, and developing intelligent strategies for handling missing data points. This ensures our foundation is solid before moving to heavyweight processing.

- **Data Ingestion**
  Once cleaned, we migrate our datasets to Hadoop's distributed file system (HDFS). Unlike traditional storage solutions, HDFS spreads data across multiple nodes in the cluster, creating natural redundancy and enabling parallel processing capabilities. This architecture becomes particularly valuable as our dataset contains more than 160K records, providing both the reliability and performance needed for enterprise-scale analytics.

- **Data Analysis**
  We leverage Hive's SQL-like interface to perform complex analytical operations within the Hadoop ecosystem. Our Hive Metastore runs on a dedicated MySQL backend, dramatically improving query performance and metadata management compared to Derby alternatives. We craft specialized HiveQL queries that help uncover hidden patterns and relationships within the data, focusing particularly on temporal trends and categorical correlations.

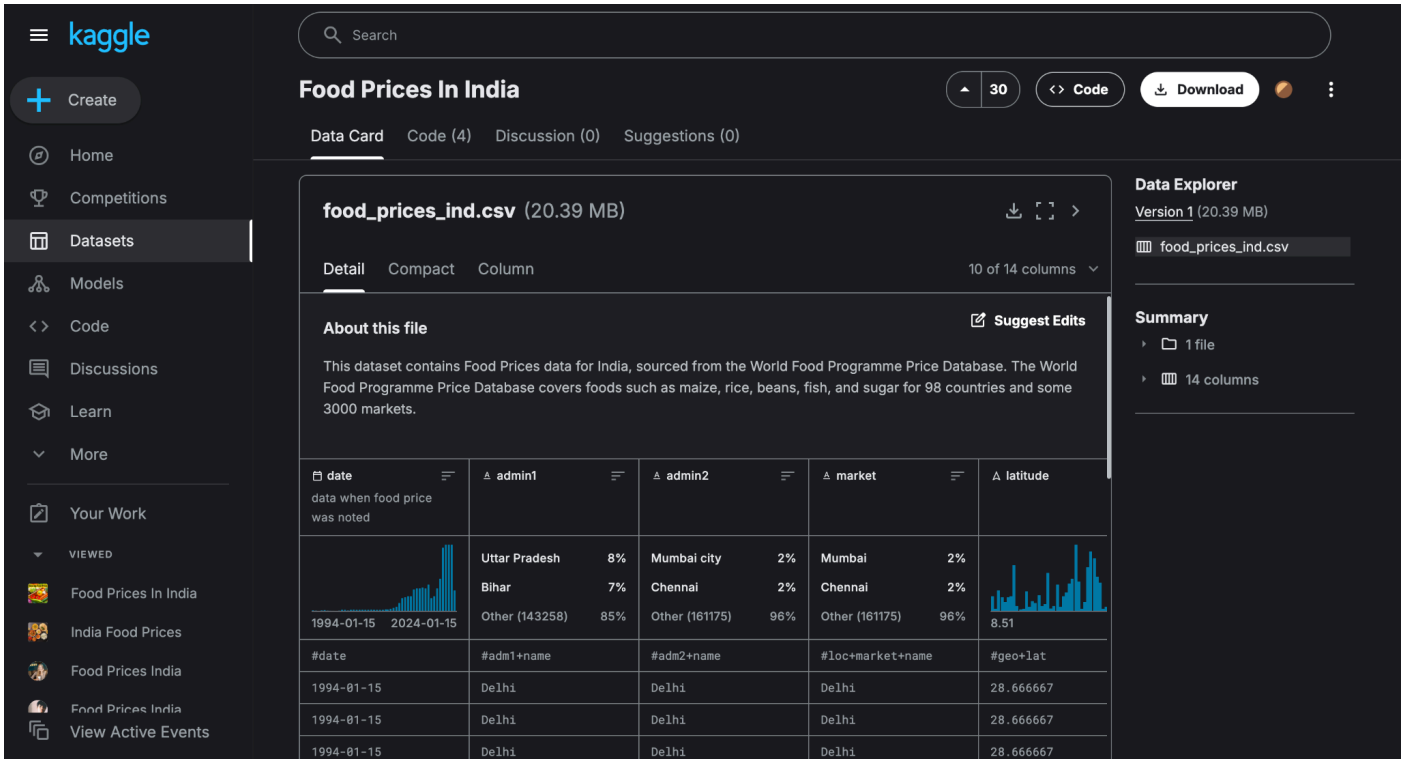
- **Visualize**
  The final analytical outputs are transferred to Google Colab notebooks where we harness Matplotlib's extensive visualization capabilities and Colab's Tensor Processing Units. These visual assets transform raw numbers into compelling narratives that drive strategic decision-making.

Dataset Description

The Indian Food Prices dataset is a comprehensive collection of food commodity price information, capturing over 1,60,000 records across multiple Indian states and cities, sourced from the World Food Programme Price Database. The World Food Programme Price Database covers foods such as maize, rice, beans, fish, and sugar for 98 countries and some 3000 markets.

The dataset includes detailed market-level data with attributes such as date, state, city, market location, geographical coordinates, commodity category, specific commodity, pricing unit, and price variations. It provides a granular view of food pricing dynamics, encompassing wholesale and retail prices across different markets, enabling in-depth analysis of regional price fluctuations, seasonal trends, and economic patterns in the Indian food commodity landscape. The data spans multiple years, offering a robust tool for understanding agricultural market behaviors, price correlations, and potential economic insights.

Source: https://www.kaggle.com/datasets/abhinavshaw09/food-prices-in-india



Screenshot from kaggle

Dataset Attributes:

Field Name	Datatype	Description
date	STRING	The date on which the price was recorded.
state	STRING	The state in India where the price was observed.
city	STRING	The city within the state where the market is located.
market	STRING	The specific market or mandi where the commodity price was recorded.
latitude	FLOAT	The latitude coordinate of the market location.
longitude	FLOAT	The longitude coordinate of the market location.
category	STRING	The category of the commodity (e.g., grains, vegetables).
commodity	STRING	The specific commodity being priced (e.g., rice, wheat).
unit	STRING	The unit of measurement for the commodity price (e.g., kg, liter).
priceflag	STRING	An indicator of whether the price is estimated or actual.
currency	STRING	The currency in which the price is recorded. Mostly INR.
price	FLOAT	The recorded price of the commodity.
usdprice	FLOAT	The price converted to USD (if applicable).
Below columns were generated during preprocessing of the dataset		
year	INTEGER	The year in which the price was recorded.
month	INTEGER	The month in which the price was recorded.
season	STRING	The season during which the price was recorded.

Questions Overview

The analysis covers various dimensions such as geography, seasonality, and market conditions, providing a comprehensive understanding of the factors influencing food prices. We aim to uncover trends, correlations, and patterns that can inform strategic decision-making.

Questions:

1. What is the average price of each commodity across different states, and how does it vary by season?
2. Which city has the highest average price for each commodity, and how does it compare to the national average price for that commodity?
3. Identify the top 5 commodities with the highest price volatility across different markets and states.
4. Which state has the most significant price difference between the highest and lowest priced commodities?
5. Which markets have the highest average price for each category, and how does it compare to the overall average price for that category?
6. Show only the Top 10 commodities with the highest price increase over the past year.
7. If a market has high prices for vegetables and fruits, does it also have high prices for cereals and tubers?
8. Rank commodities within each state based on their average price, and identify the top 3 most expensive commodities per state.
9. How do food prices in metropolitan cities compare to other markets during different seasons of the year?
10. Analyse the evolution of commodity prices over time, revealing year-to-year price dynamics and identifying significant market trends.
11. Identify the top 5 markets with the highest average price for each commodity, considering only the latest year.
12. Analyze the impact of seasonality on price volatility to calculate the coefficient of variation for each commodity.
13. Identify the cheapest and costliest state for each commodity for a given year? Take the year 2024 for example.
14. Which cities experienced consistent food price inflation over the years?
15. Identify the top 10 regions in India where food prices are the most volatile, by calculating the average price variance across nearby markets (within a 100 km radius).

Why we Selected these Questions:

- To understand how prices fluctuate with seasons, helping stakeholders anticipate changes and plan inventory accordingly. This insight is crucial for managing supply and demand effectively.
- By analyzing price differences across states and cities, we can uncover regional disparities. This helps businesses tailor their strategies to local market conditions, ensuring competitive pricing.
- Examining how prices vary in different markets reveals the impact of local factors such as transportation costs and market demand. This knowledge allows for more precise pricing strategies.
- Focusing on individual commodities helps identify which items are most volatile or stable. This is vital for risk management and investment decisions.
- Analyzing price fluctuations over time uncovers patterns that can indicate market stability or instability. This insight is essential for long-term planning and forecasting.
- Understanding the relationships between different variables, such as location and price, helps identify underlying factors driving price changes. This can lead to more informed decision-making.

Data Loading (Preparing Hive for the analysis)

```
-- Main Table
CREATE TABLE food_prices (
    date DATE,
    state STRING,
    city STRING,
    market STRING,
    latitude DOUBLE,
    longitude DOUBLE,
    category STRING,
    commodity STRING,
    unit STRING,
    price DOUBLE,
    year INT,
    month INT,
    season STRING
) PARTITIONED BY (category STRING) CLUSTERED BY (state) INTO 10 BUCKETS STORED AS
PARQUET;

-- External table for data loading from hdfs
CREATE TABLE staging_food_prices (
    `date` DATE,
    state STRING,
    city STRING,
    market STRING,
    latitude DOUBLE,
    longitude DOUBLE,
    category STRING,
    commodity STRING,
    unit STRING,
    priceflag STRING,
    pricetype STRING,
    currency STRING,
    price DOUBLE,
    usdprice DOUBLE,
    year INT,
    month INT,
    season STRING
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

LOAD DATA INPATH '/data/preprocessed_india_food_prices.csv' INTO TABLE
staging_food_prices;

-- Skip the csv header while selecting
ALTER TABLE staging_food_prices
SET TBLPROPERTIES ("skip.header.line.count" = "1");

-- Enable dynamic partitioning
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;

-- Insert from staging to partitioned table
INSERT OVERWRITE TABLE food_prices PARTITION(category)
SELECT
```

```
  `date`,
  state,
  city,
  market,
  latitude,
  longitude,
  commodity,
  unit,
  price,
  year,
  month,
  season,
  category
FROM staging_food_prices;

SELECT COUNT(*) FROM food_prices;
```

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 2.3.2)
Driver: Hive JDBC (version 2.3.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.3.2 by Apache Hive
0: jdbc:hive2://localhost:10000> use case_study;
No rows affected (0.197 seconds)
0: jdbc:hive2://localhost:10000> CREATE TABLE food_prices ( `date` DATE, state STRING, city STRING, market STRING, latitude DOUBLE, longitude DOUBLE, commodity STRING, unit STRING, price DOUBLE, year INT,
 month INT, season STRING ) PARTITIONED BY (category STRING) CLUSTERED BY (state) INTO 10 BUCKETS STORED AS PARQUET;
No rows affected (0.411 seconds)
0: jdbc:hive2://localhost:10000> SET hive.exec.dynamic.partition=true;
No rows affected (0.037 seconds)
0: jdbc:hive2://localhost:10000> SET hive.exec.dynamic.partition.mode=nonstrict;
No rows affected (0.019 seconds)
0: jdbc:hive2://localhost:10000> CREATE TABLE staging_food_prices ( `date` DATE, state STRING, city STRING, market STRING, latitude DOUBLE, longitude DOUBLE, category STRING, commodity STRING, unit STRING
, priceflag STRING, pricetype STRING, currency STRING, price DOUBLE, usdprice DOUBLE, year INT, month INT, season STRING ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
No rows affected (0.607 seconds)
0: jdbc:hive2://localhost:10000> LOAD DATA INPATH '/data/preprocessed_india_food_prices.csv' INTO TABLE staging_food_prices;
No rows affected (1.526 seconds)
0: jdbc:hive2://localhost:10000> ALTER TABLE staging_food_prices SET TBLPROPERTIES ("skip.header.line.count"="1");
No rows affected (0.353 seconds)
0: jdbc:hive2://localhost:10000> INSERT OVERWRITE TABLE food_prices PARTITION(category) SELECT `date`, state, city, market, latitude, longitude, commodity, unit, price, year, month, season, category FROM
staging_food_prices;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
No rows affected (25.407 seconds)
0: jdbc:hive2://localhost:10000> Display all 560 possibilities? (y or n)
0: jdbc:hive2://localhost:10000> DROP TABLE staging_food_prices;
No rows affected (1.593 seconds)
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) FROM food_prices;
+---------+
|   _c0   |
+---------+
| 166096  |
+---------+
1 row selected (1.238 seconds)
0: jdbc:hive2://localhost:10000>
```

Data Analysis

Q1) What is the average price of each commodity across different states, and how does it vary by season?

```
SELECT
    commodity,
    state,
    season,
    AVG(price) AS avg_price
FROM
    food_prices
GROUP BY
    commodity, state, season
ORDER BY
    commodity, state, season;
```

```
0: jdbc:hive2://localhost:10000> SELECT commodity, state, season, AVG(price) AS avg_price FROM food_prices GROUP BY commodity, state, season ORDER BY commodity, state, season;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+--------------------+----------------------+----------+--------------------+
|      commodity     |        state         |  season  |      avg_price     |
+--------------------+----------------------+----------+--------------------+
| Chickpeas          | Bihar                | Autumn   | 43.699803921568616 |
| Chickpeas          | Bihar                | Monsoon  | 40.633695652173905 |
| Chickpeas          | Bihar                | Summer   | 41.48404255319149  |
| Chickpeas          | Bihar                | Winter   | 41.48348837209302  |
| Chickpeas          | Delhi                | Autumn   | 54.031136363636364 |
| Chickpeas          | Delhi                | Monsoon  | 58.1788            |
| Chickpeas          | Delhi                | Summer   | 42.90411764705883  |
| Chickpeas          | Delhi                | Winter   | 46.83926829268295  |
| Chickpeas          | Maharashtra          | Autumn   | 52.02208333333331  |
| Chickpeas          | Maharashtra          | Monsoon  | 48.896274509803933 |
| Chickpeas          | Maharashtra          | Summer   | 46.523333333333326 |
| Chickpeas          | Maharashtra          | Winter   | 55.50717948717948  |
| Chickpeas          | Tamil Nadu           | Autumn   | 50.016000000000005 |
| Chickpeas          | Tamil Nadu           | Monsoon  | 44.98367346938776  |
| Chickpeas          | Tamil Nadu           | Summer   | 47.099999999999994 |
| Chickpeas          | Tamil Nadu           | Winter   | 44.425106382978726 |
| Ghee (vanaspati)   | Andaman And Nicobar  | Autumn   | 162.618125         |
| Ghee (vanaspati)   | Andaman And Nicobar  | Monsoon  | 154.72625          |
| Ghee (vanaspati)   | Andaman And Nicobar  | Summer   | 157.32176470588234 |
| Ghee (vanaspati)   | Andaman And Nicobar  | Winter   | 167.016            |
| Ghee (vanaspati)   | Andhra Pradesh       | Autumn   | 103.20545454545454 |
| Ghee (vanaspati)   | Andhra Pradesh       | Monsoon  | 103.65545454545453 |
| Ghee (vanaspati)   | Andhra Pradesh       | Summer   | 106.55314285714286 |
| Ghee (vanaspati)   | Andhra Pradesh       | Winter   | 109.11545454545454 |
| Ghee (vanaspati)   | Assam                | Autumn   | 107.40863636363635 |
| Ghee (vanaspati)   | Assam                | Monsoon  | 105.86571428571429 |
| Ghee (vanaspati)   | Assam                | Summer   | 98.86409090909092  |
| Ghee (vanaspati)   | Assam                | Winter   | 98.39421052631579  |
| Ghee (vanaspati)   | Bihar                | Autumn   | 120.54626262626265 |
| Ghee (vanaspati)   | Bihar                | Monsoon  | 123.38545454545459 |
| Ghee (vanaspati)   | Bihar                | Summer   | 126.0041221374046  |
| Ghee (vanaspati)   | Bihar                | Winter   | 121.78205882352941 |
| Ghee (vanaspati)   | Chandigarh           | Autumn   | 98.1464705882353   |
| Ghee (vanaspati)   | Chandigarh           | Monsoon  | 109.78333333333333 |
| Ghee (vanaspati)   | Chandigarh           | Summer   | 102.5526086956522  |
| Ghee (vanaspati)   | Chandigarh           | Winter   | 101.34235294117647 |
| Ghee (vanaspati)   | Chhattisgarh         | Autumn   | 116.69718750000001 |
| Ghee (vanaspati)   | Chhattisgarh         | Monsoon  | 129.2775           |
| Ghee (vanaspati)   | Chhattisgarh         | Summer   | 133.7174074074074  |
| Ghee (vanaspati)   | Chhattisgarh         | Winter   | 122.94760000000001 |
| Ghee (vanaspati)   | Delhi                | Autumn   | 111.61571428571429 |
| Ghee (vanaspati)   | Delhi                | Monsoon  | 115.699            |
| Ghee (vanaspati)   | Delhi                | Summer   | 117.37857142857142 |
| Ghee (vanaspati)   | Delhi                | Winter   | 105.21416666666669 |
| Ghee (vanaspati)   | Goa                  | Autumn   | 125.914375         |
| Ghee (vanaspati)   | Goa                  | Monsoon  | 142.6423076923077  |
| Ghee (vanaspati)   | Goa                  | Summer   | 132.63277777777776 |
| Ghee (vanaspati)   | Goa                  | Winter   | 131.3790909090909  |
| Ghee (vanaspati)   | Gujarat              | Autumn   | 110.88227272727272 |
| Ghee (vanaspati)   | Gujarat              | Monsoon  | 106.2840277777778  |
| Ghee (vanaspati)   | Gujarat              | Summer   | 110.80283582089552 |
| Ghee (vanaspati)   | Gujarat              | Winter   | 110.01666666666668 |
| Ghee (vanaspati)   | Haryana              | Autumn   | 106.0271186440678  |
| Ghee (vanaspati)   | Haryana              | Monsoon  | 101.3058241758242  |
| Ghee (vanaspati)   | Haryana              | Summer   | 104.74516129032257 |
| Ghee (vanaspati)   | Haryana              | Winter   | 103.01914634146338 |
| Ghee (vanaspati)   | Himachal Pradesh     | Autumn   | 122.51979591836736 |
| Ghee (vanaspati)   | Himachal Pradesh     | Monsoon  | 122.96983050847462 |
```

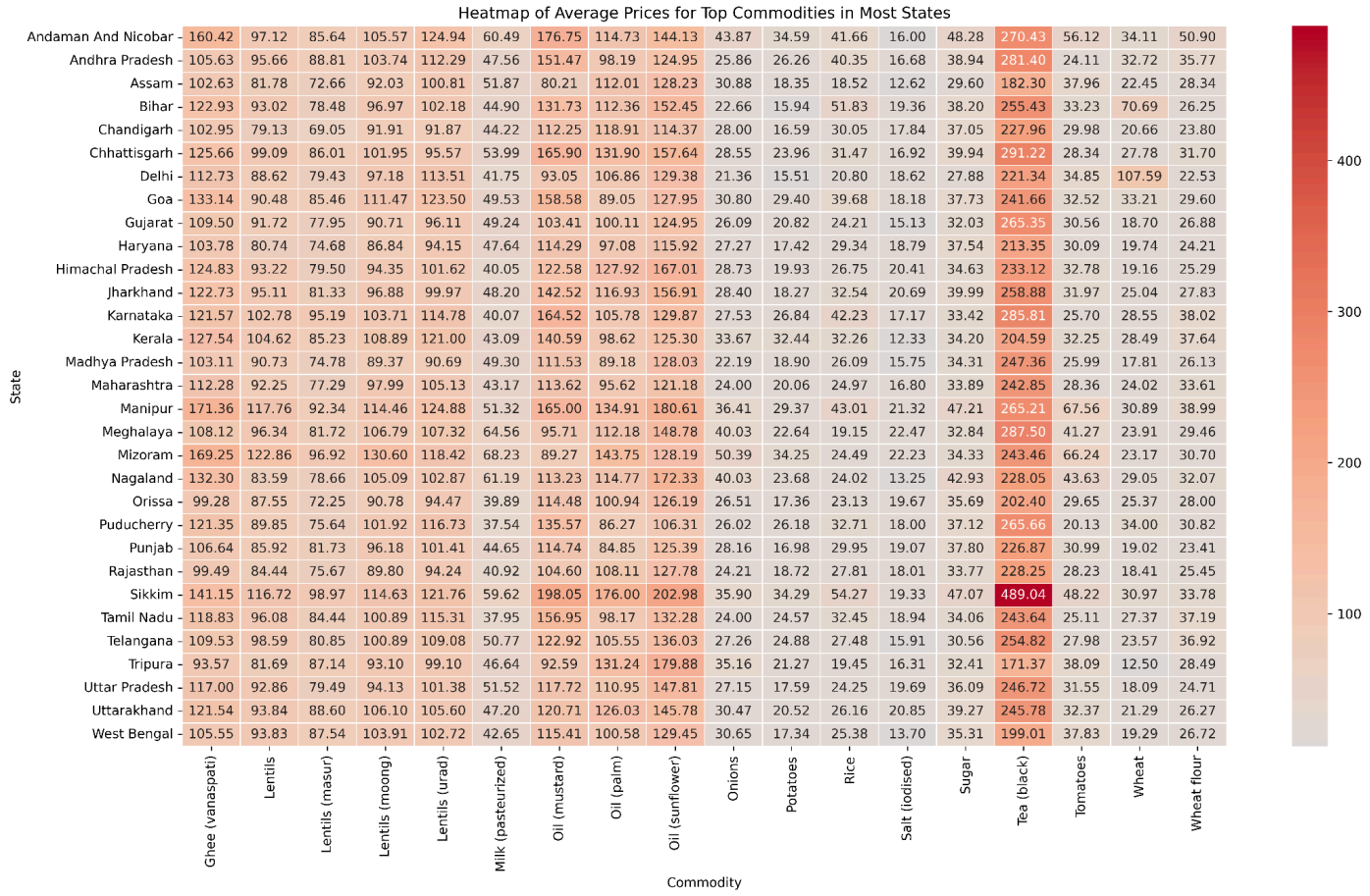
Relevance

Understanding average commodity prices across states and seasons reveals crucial regional and temporal price patterns. This helps identify which states consistently experience higher food costs and when prices typically peak or dip throughout the year. Such data helps both

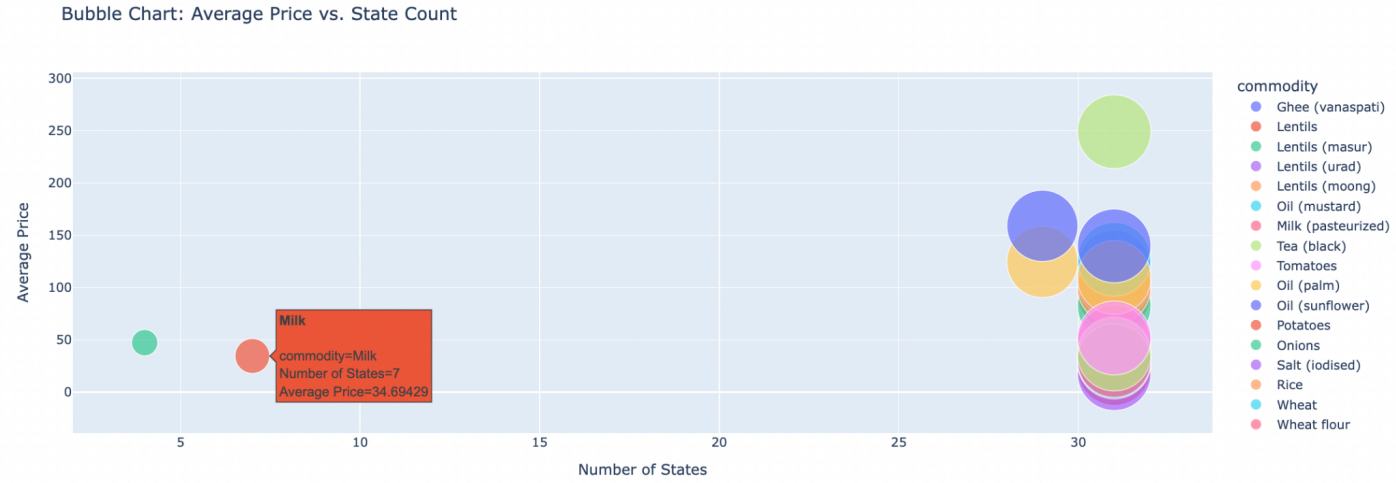
consumers and policymakers manage their budgets and determine where and when to deploy food security measures.

Visualization

Heatmap showing the average prices of the commodities that are present in most states.



The bubble chart shows the relationship between the number of states a commodity is present in and its average price. Larger bubbles indicate commodities that are available in more states, while the y-axis represents their average price.



Q2) Which city has the highest average price for each commodity, and how does it compare to the national average price for that commodity?

```
WITH national_avg AS (
    SELECT
        commodity,
        AVG(price) AS national_avg_price
    FROM
        food_prices
    GROUP BY
        commodity
)
SELECT
    a.city,
    a.commodity,
    AVG(a.price) AS city_avg_price,
    b.national_avg_price
FROM
    food_prices a
JOIN
    national_avg b ON a.commodity = b.commodity
GROUP BY
    a.city, a.commodity, b.national_avg_price
ORDER BY
    a.commodity, city_avg_price DESC;
```

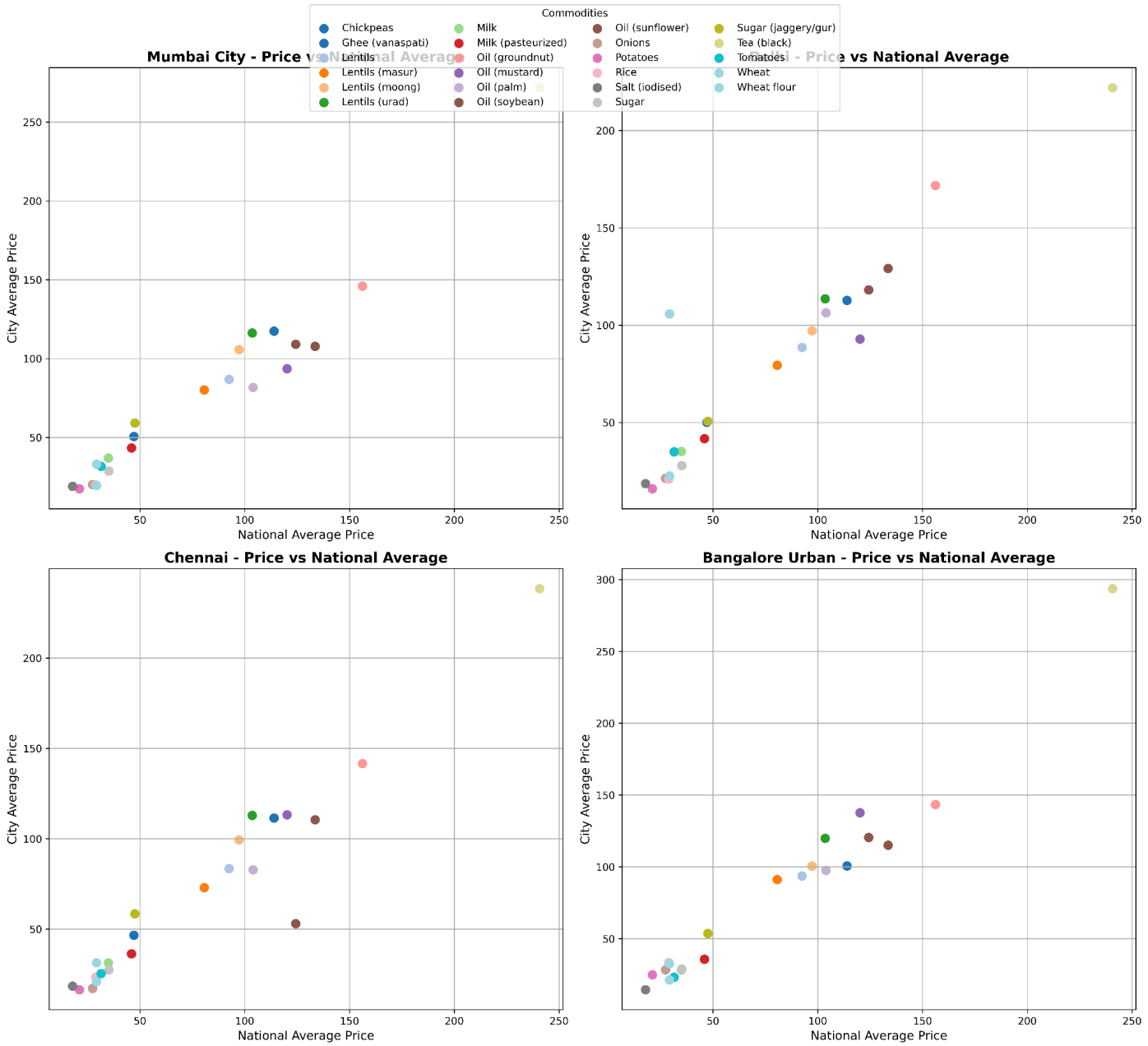
```
0: jdbc:hive2://localhost:10000> WITH national_avg AS ( SELECT commodity, AVG(price) AS national_avg_price FROM food_prices GROUP BY commodity ) SELECT a.city, a.commodity, AVG(a.price) AS city_avg_price,
 b.national_avg_price FROM food_prices a JOIN national_avg b ON a.commodity = b.commodity GROUP BY a.city, a.commodity, b.national_avg_price ORDER BY a.commodity, city_avg_price DESC;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20250322133646_3ef2da41-abed-4cf3-834e-07e925b4277c.log
2025-03-22 13:36:54     Starting to launch local task to process map join;      maximum memory = 477626368
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
2025-03-22 13:36:58     Dump the side-table for tag: 0 with group count: 23 into file: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_13-36-46_827_5331832392289659293-1/-local-10007/H
ashTable-Stage-2/MapJoin-mapfile00--.hashtable
2025-03-22 13:36:58     Uploaded 1 File to: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_13-36-46_827_5331832392289659293-1/-local-10007/HashTable-Stage-2/MapJoin-mapfile00--.hashta
ble (3693832 bytes)
2025-03-22 13:36:58     End of local task; Time Taken: 3.512 sec.
+--------------------+--------------------+--------------------+----------------------+
|       a.city       |     a.commodity    |   city_avg_price   | b.national_avg_price |
+--------------------+--------------------+--------------------+----------------------+
| Mumbai City        | Chickpeas          | 50.608500000000014 | 47.07922077922078    |
| Delhi              | Chickpeas          | 50.07631944444445  | 47.07922077922078    |
| Chennai            | Chickpeas          | 46.56043956043957  | 47.07922077922078    |
| Patna              | Chickpeas          | 41.8790374331551   | 47.07922077922078    |
| Mysore             | Ghee (vanaspati)   | 207.25678571428574 | 113.96860593289009   |
| Kohima             | Ghee (vanaspati)   | 178.84714285714287 | 113.96860593289009   |
| Wayanad            | Ghee (vanaspati)   | 170.6677777777778  | 113.96860593289009   |
| West Imphal        | Ghee (vanaspati)   | 170.1375           | 113.96860593289009   |
| Aizawl             | Ghee (vanaspati)   | 168.62             | 113.96860593289009   |
| Valsad             | Ghee (vanaspati)   | 161.21666666666667 | 113.96860593289009   |
| Andaman Islands    | Ghee (vanaspati)   | 159.01962962962966 | 113.96860593289009   |
| Saharanpur         | Ghee (vanaspati)   | 158.8345           | 113.96860593289009   |
| Thrissur           | Ghee (vanaspati)   | 156.50222222222223 | 113.96860593289009   |
| Palakkad           | Ghee (vanaspati)   | 153.61862068965513 | 113.96860593289009   |
| Jhabua             | Ghee (vanaspati)   | 153.26769230769233 | 113.96860593289009   |
| Vadodara           | Ghee (vanaspati)   | 152.6717391304348  | 113.96860593289009   |
| Munger             | Ghee (vanaspati)   | 152.35071428571428 | 113.96860593289009   |
| Purba Singhbhum    | Ghee (vanaspati)   | 151.935652173913   | 113.96860593289009   |
| Saran              | Ghee (vanaspati)   | 151.21384615384613 | 113.96860593289009   |
| Araria             | Ghee (vanaspati)   | 151.04272727272726 | 113.96860593289009   |
| The Dangs          | Ghee (vanaspati)   | 150.6221052631579  | 113.96860593289009   |
| Jaintia Hills      | Ghee (vanaspati)   | 149.8245           | 113.96860593289009   |
| Katihar            | Ghee (vanaspati)   | 149.78833333333336 | 113.96860593289009   |
| Meerut             | Ghee (vanaspati)   | 149.20033333333336 | 113.96860593289009   |
| Raigarh            | Ghee (vanaspati)   | 149.08789473684212 | 113.96860593289009   |
| Bijapur            | Ghee (vanaspati)   | 149.0694117647059  | 113.96860593289009   |
| Baleshwar          | Ghee (vanaspati)   | 148.85111111111115 | 113.96860593289009   |
| Lohardaga          | Ghee (vanaspati)   | 148.47466666666665 | 113.96860593289009   |
| Cuddalore          | Ghee (vanaspati)   | 148.4345833333333  | 113.96860593289009   |
| Mirzapur           | Ghee (vanaspati)   | 148.16611111111112 | 113.96860593289009   |
| Solan              | Ghee (vanaspati)   | 146.00150000000002 | 113.96860593289009   |
| Chamba             | Ghee (vanaspati)   | 145.72812499999998 | 113.96860593289009   |
| Darrang            | Ghee (vanaspati)   | 145.24166666666667 | 113.96860593289009   |
| Aligarh            | Ghee (vanaspati)   | 144.42736842105262 | 113.96860593289009   |
| Kangra             | Ghee (vanaspati)   | 144.40166666666667 | 113.96860593289009   |
| Vellore            | Ghee (vanaspati)   | 144.35307692307694 | 113.96860593289009   |
| Hoshangabad        | Ghee (vanaspati)   | 144.0776923076923  | 113.96860593289009   |
| Bareilly           | Ghee (vanaspati)   | 143.87478260869565 | 113.96860593289009   |
| Naini Tal          | Ghee (vanaspati)   | 143.73089523809524 | 113.96860593289009   |
| Una                | Ghee (vanaspati)   | 143.34411764705882 | 113.96860593289009   |
| Madhubani          | Ghee (vanaspati)   | 143.32500000000002 | 113.96860593289009   |
| Haridwar           | Ghee (vanaspati)   | 142.33318181818183 | 113.96860593289009   |
```

Relevance

Identifying cities with the highest costs for each product highlights urban areas with the most severe cost-of-living constraints. Comparing these prices to national averages reveals how much more inhabitants in these areas spend for necessities, which is useful for understanding urban poverty and designing targeted subsidies or market interventions.

Visualization

The visualization shows a price comparison using a scatter plot for 4 major Indian cities (Mumbai City, Delhi, Chennai and Bengaluru Urban). Results from the above query were processed in Python to draw the relevant visualization.



Q3) Identify the top 5 commodities with the highest price volatility across different markets and states.

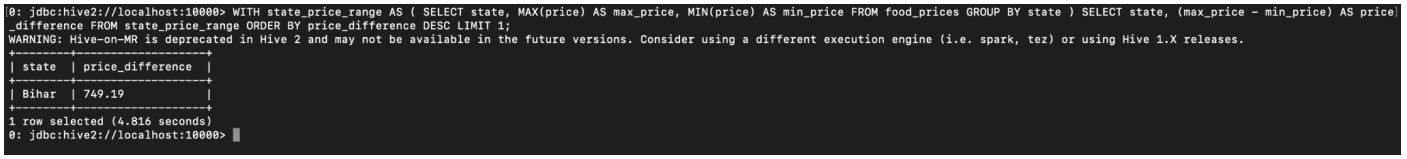
```
SELECT
    commodity,
    STDDEV(price) AS price_volatility
FROM
    food_prices
GROUP BY
    commodity
ORDER BY
    price_volatility DESC
LIMIT 5;
```

Relevance

Price volatility has a direct impact on food security and household budgets. By finding the top five commodities with the highest volatility, we can determine which foods cause the most financial worry for households. This information is crucial for government agencies focused on price stabilisation and organisations working on food security programs.

Q4) Which state has the most significant price difference between the highest and lowest priced commodities?

```
WITH state_price_range AS (
    SELECT
        state,
        MAX(price) AS max_price,
        MIN(price) AS min_price
    FROM
        food_prices
    GROUP BY
        state
)
SELECT
    state,
    (max_price - min_price) AS price_difference
FROM
    state_price_range
ORDER BY
    price_difference DESC
LIMIT 1;
```



Relevance

The state with the largest price difference between commodities likely experiences significant economic disparity or logistical challenges. This analysis highlights where certain foods might be luxury items while others remain affordable, pointing to possible supply chain inefficiencies or regional production specialization that affects accessibility of diverse foods.

Q5) Which markets have the highest average price for each category, and how does it compare to the overall average price for that category?

```
WITH category_avg AS (
    SELECT
        category,
        AVG(price) AS overall_avg_price
    FROM
        food_prices
    GROUP BY
        category
)
SELECT
    a.market,
    a.category,
    AVG(a.price) AS market_avg_price,
    b.overall_avg_price
FROM
    food_prices a
JOIN
    category_avg b ON a.category = b.category
GROUP BY
    a.market, a.category, b.overall_avg_price
ORDER BY
    a.category, market_avg_price DESC;
```

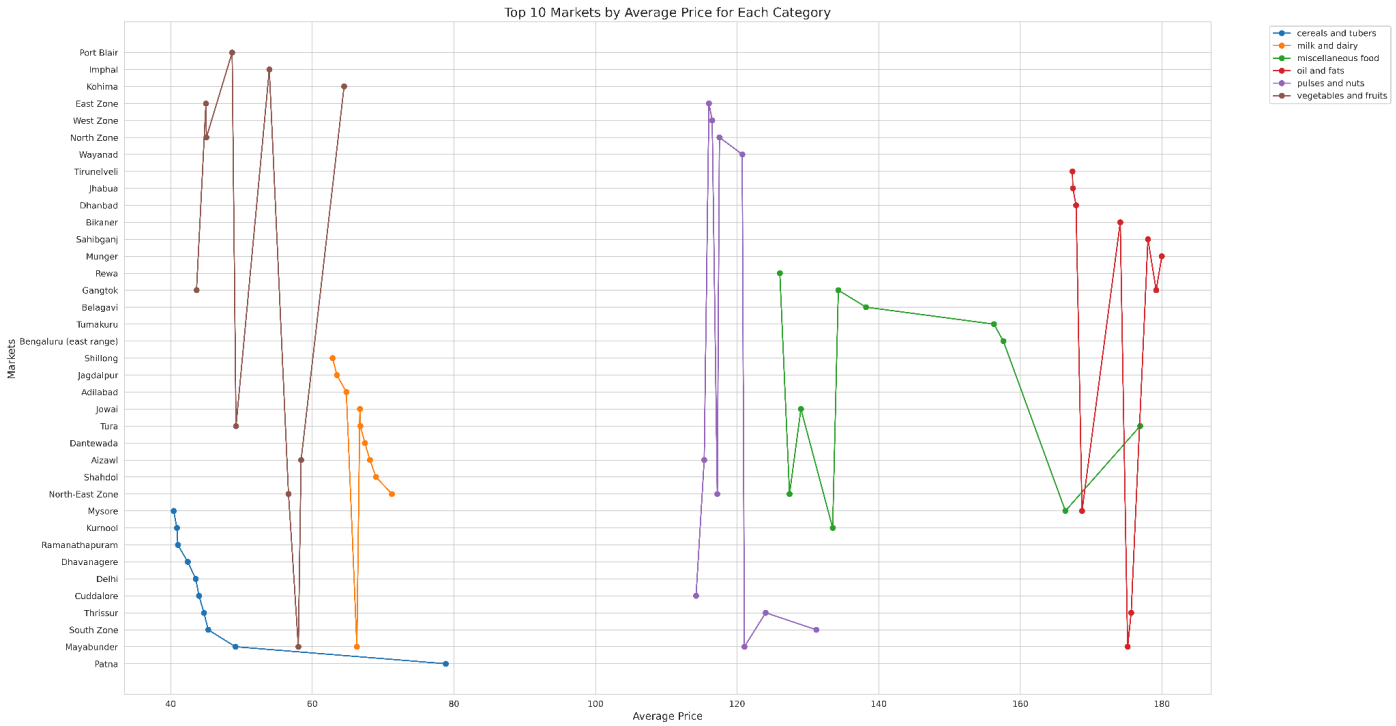
```
0: jdbc:hive2://localhost:10000> WITH category_avg AS ( SELECT category, AVG(price) AS overall_avg_price FROM food_prices GROUP BY category ) SELECT a.market, a.category, AVG(a.price) AS market_avg_price,
 b.overall_avg_price FROM food_prices a JOIN category_avg b ON a.category = b.category GROUP BY a.market, a.category, b.overall_avg_price ORDER BY a.category, market_avg_price DESC;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20250322135656_ebad5668-dc35-44ab-8c69-93d8147ab51d.log
2025-03-22 13:57:05     Starting to launch local task to process map join;      maximum memory = 477626368
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
2025-03-22 13:57:07     Dump the side-table for tag: 0 with group count: 6 into file: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_13-56-56_721_6876095130143795652-1/-local-10007/HashTable-Stage-2/MapJoin-mapfile20--.hashtable
2025-03-22 13:57:08     Uploaded 1 File to: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_13-56-56_721_6876095130143795652-1/-local-10007/HashTable-Stage-2/MapJoin-mapfile20--.hashtable (3620529 bytes)
2025-03-22 13:57:08     End of local task; Time Taken: 3.145 sec.
+-------------------------+-------------------------+--------------------+----------------------+
|        a.market         |       a.category        |  market_avg_price  | b.overall_avg_price  |
+-------------------------+-------------------------+--------------------+----------------------+
| Patna                   | cereals and tubers      | 78.85346652267819  | 27.28121837270347    |
| Mayabunder              | cereals and tubers      | 49.1423157894737   | 27.28121837270347    |
| South Zone              | cereals and tubers      | 45.298260869565226 | 27.28121837270347    |
| Thrissur                | cereals and tubers      | 44.699907407407416 | 27.28121837270347    |
| Cuddalore               | cereals and tubers      | 43.995940594059405 | 27.28121837270347    |
| Delhi                   | cereals and tubers      | 43.5377419354838B6 | 27.28121837270347    |
| Dhavanagere             | cereals and tubers      | 42.4068085106383   | 27.28121837270347    |
| Ramanathapuram          | cereals and tubers      | 41.007183098591554 | 27.28121837270347    |
| Kurnool                 | cereals and tubers      | 40.87576923076924  | 27.28121837270347    |
| Mysore                  | cereals and tubers      | 40.42229357798165  | 27.28121837270347    |
| Tirupathi               | cereals and tubers      | 39.64473214285715  | 27.28121837270347    |
| Tirunelveli             | cereals and tubers      | 39.583333333333336 | 27.28121837270347    |
| Adilabad                | cereals and tubers      | 39.48296296296297  | 27.28121837270347    |
| Vellore                 | cereals and tubers      | 39.244736842105276 | 27.28121837270347    |
| Maharashtra_Pune        | cereals and tubers      | 39.21092783505154  | 27.28121837270347    |
| Palakkad                | cereals and tubers      | 38.969067796610176 | 27.28121837270347    |
| Wayanad                 | cereals and tubers      | 38.57773584905661  | 27.28121837270347    |
| Selamba                 | cereals and tubers      | 38.254125          | 27.28121837270347    |
| Mangalore               | cereals and tubers      | 38.00768000000001  | 27.28121837270347    |
| Port Blair              | cereals and tubers      | 37.62918421052631  | 27.28121837270347    |
| Gangtok                 | cereals and tubers      | 37.47848837209303  | 27.28121837270347    |
| Coimbatore              | cereals and tubers      | 37.40037037037036  | 27.28121837270347    |
| Bellary                 | cereals and tubers      | 37.211914893617035 | 27.28121837270347    |
| Dharmapuri              | cereals and tubers      | 37.114186046511634 | 27.28121837270347    |
| Kalaburagi              | cereals and tubers      | 37.06673469387756  | 27.28121837270347    |
| Tumakuru                | cereals and tubers      | 37.05448979591838  | 27.28121837270347    |
| Mangaon-raigad          | cereals and tubers      | 36.960243902439025 | 27.28121837270347    |
| North-East Zone         | cereals and tubers      | 36.770454545454544 | 27.28121837270347    |
| T.Puram                 | cereals and tubers      | 36.407037037037704 | 27.28121837270347    |
| Imphal                  | cereals and tubers      | 36.33415384615385  | 27.28121837270347    |
| Dantewada               | cereals and tubers      | 36.229367088607584 | 27.28121837270347    |
| Vijayapur               | cereals and tubers      | 35.99598039215686  | 27.28121837270347    |
| Shivamogga              | cereals and tubers      | 35.73020408163264  | 27.28121837270347    |
| Kozhikode               | cereals and tubers      | 35.69007843137254  | 27.28121837270347    |
| Belagavi                | cereals and tubers      | 35.49911764705881  | 27.28121837270347    |
| Ajmer                   | cereals and tubers      | 35.46107692307692  | 27.28121837270347    |
| Bengaluru (east range)  | cereals and tubers      | 35.4158762886598   | 27.28121837270347    |
| Kohima                  | cereals and tubers      | 35.40709677419355  | 27.28121837270347    |
| Bikaner                 | cereals and tubers      | 34.8436170212766   | 27.28121837270347    |
| Karimnagar              | cereals and tubers      | 34.78304347826087  | 27.28121837270347    |
| Warangal                | cereals and tubers      | 34.450101010101015 | 27.28121837270347    |
| Tura                    | cereals and tubers      | 33.677592592592596 | 27.28121837270347    |
```

Relevance

Markets with the highest average prices for food categories often indicate areas where residents face greater food insecurity risks. Comparing these to overall category averages reveals the magnitude of the cost burden in these locations. This information helps identify communities that might need targeted support through public distribution systems or improved market access.

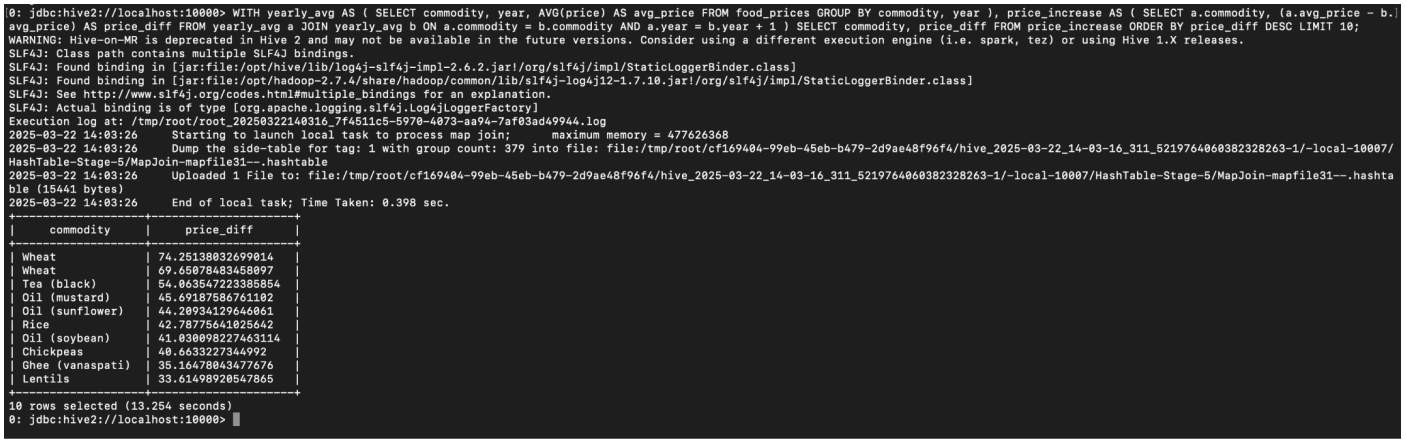
Visualization

This line chart visualizes the top 10 markets for each product category, displaying their average prices across different regions of India.



Q6) Show only the Top 10 commodities with the highest price increase over the past year.

```
WITH yearly_avg AS (
    SELECT
        commodity,
        year,
        AVG(price) AS avg_price
    FROM
        food_prices
    GROUP BY
        commodity, year
),
price_increase AS (
    SELECT
        a.commodity,
        (a.avg_price - b.avg_price) AS price_diff
    FROM
        yearly_avg a
    JOIN
        yearly_avg b ON a.commodity = b.commodity AND a.year = b.year + 1
)
SELECT
    commodity,
    price_diff
FROM
    price_increase
ORDER BY
    price_diff DESC
LIMIT 10;
```



```
0: jdbc:hive2://localhost:10000> WITH yearly_avg AS ( SELECT commodity, year, AVG(price) AS avg_price FROM food_prices GROUP BY commodity, year ), price_increase AS ( SELECT a.commodity, (a.avg_price - b.
avg_price) AS price_diff FROM yearly_avg a JOIN yearly_avg b ON a.commodity = b.commodity AND a.year = b.year + 1 ) SELECT commodity, price_diff FROM price_increase ORDER BY price_diff DESC LIMIT 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20250322140316_7f4511c5-5970-4073-aa94-7af03ad49944.log
2025-03-22 14:03:26     Starting to launch local task to process map join;      maximum memory = 477626368
2025-03-22 14:03:26     Dump the side-table for tag: 1 with group count: 379 into file: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_14-03-16_311_5219764060382328263-1/-local-10007/
HashTable-Stage-5/MapJoin-mapfile31--.hashtable
2025-03-22 14:03:26     Uploaded 1 File to: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_14-03-16_311_5219764060382328263-1/-local-10007/HashTable-Stage-5/MapJoin-mapfile31--.hashta
ble (15441 bytes)
2025-03-22 14:03:26     End of local task; Time Taken: 0.398 sec.
+-------------------+---------------------+
|     commodity     |     price_diff      |
+-------------------+---------------------+
| Wheat             | 74.25138032699014   |
| Wheat             | 69.65078483458097   |
| Tea (black)       | 54.063547223385854  |
| Oil (mustard)     | 45.69187586761102   |
| Oil (sunflower)   | 44.20934129646061   |
| Rice              | 42.78775641025642   |
| Oil (soybean)     | 41.030098227463114  |
| Chickpeas         | 40.6633227344992    |
| Ghee (vanaspati)  | 35.16478043477676   |
| Lentils           | 33.61498920547865   |
+-------------------+---------------------+
10 rows selected (13.254 seconds)
0: jdbc:hive2://localhost:10000>
```

Relevance

Tracking commodities with the highest price increases spotlights the emerging affordability crisis. These rapid price jumps can signal supply shortages, distribution problems, or changing consumption patterns. Policymakers and market regulators need this information to monitor inflation triggers and implement timely interventions before staple foods become unaffordable.

Q7) If a market has high prices for vegetables and fruits, does it also have high prices for cereals and tubers?

```
WITH category_avg AS (
    SELECT market, category, AVG(price) AS avg_price
    FROM food_prices
    WHERE category IN ('vegetables and fruits', 'cereals and tubers')
    GROUP BY market, category
)
SELECT market,
    MAX(CASE WHEN category IN ('vegetables and fruits') THEN avg_price ELSE 0 END) AS
veg_fruit_price,
    MAX(CASE WHEN category IN ('cereals and tubers') THEN avg_price ELSE 0 END) AS
cereal_tuber_price
FROM category_avg
GROUP BY market
HAVING veg_fruit_price > 0 AND cereal_tuber_price > 0
ORDER BY veg_fruit_price DESC, cereal_tuber_price DESC;
```

```
0: jdbc:hive2://localhost:10000> WITH category_avg AS ( SELECT market, category, AVG(price) AS avg_price FROM food_prices WHERE category IN ('vegetables and fruits', 'cereals and tubers') GROUP BY market,
 category ) SELECT market, MAX(CASE WHEN category IN ('vegetables and fruits') THEN avg_price ELSE 0 END) AS veg_fruit_price, MAX(CASE WHEN category IN ('cereals and tubers') THEN avg_price ELSE 0 END) AS
 cereal_tuber_price FROM category_avg GROUP BY market HAVING veg_fruit_price > 0 AND cereal_tuber_price > 0 ORDER BY veg_fruit_price DESC, cereal_tuber_price DESC;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+-------------------------+--------------------+--------------------+
|         market          |   veg_fruit_price  | cereal_tuber_price |
+-------------------------+--------------------+--------------------+
| Kohima                  | 64.48636363636363  | 35.40709677419355  |
| Aizawl                  | 58.417396449704135 | 28.43846153846154  |
| Mayabunder              | 58.02531914893619  | 49.1423157894737   |
| North-East Zone         | 56.64              | 36.77045454545454  |
| Imphal                  | 53.93150000000001  | 36.33415384615385  |
| Tura                    | 49.233061224489795 | 33.677592592592596 |
| Port Blair              | 48.67621359223301  | 37.62918421052631  |
| North Zone              | 45.054545454545455 | 29.44818181818182  |
| East Zone               | 44.975             | 31.53285714285714  |
| Gangtok                 | 43.652549019607854 | 37.47848837209303  |
| Jamshedpur              | 41.79409090909092  | 30.178571428571434 |
| Jowai                   | 40.976212121212136 | 28.96262295081967  |
| West Zone               | 39.7516666666666665 | 33.02818181818182  |
| Dimapur                 | 39.68563451776651  | 25.991705882352942 |
| Suryapet                | 38.17545454545455  | 32.83356521739131  |
| Shillong                | 38.12506097560974  | 20.09372093023255  |
| Dharamshala             | 37.85422222222222  | 24.693582089552244 |
| Raiganj                 | 37.414423076923086 | 25.460348837209303 |
| Thrissur                | 37.080192307692315 | 44.699907407407416 |
| Agartala                | 36.83878504672898  | 21.493638253638253 |
| South Zone              | 36.606363636363646 | 45.298260869565226 |
| Ramanathapuram          | 36.4565306122449   | 41.007183098591554 |
| Mangaon-raigad          | 35.889767441860464 | 36.960243902439025 |
| Sahibganj               | 35.837619047619036 | 24.256428571428577 |
| Malda                   | 35.507254901960785 | 24.998437499999998 |
| Guwahati                | 35.401388888888896 | 20.078104906937387 |
| Wayanad                 | 35.2248888888888884 | 38.57773584905661  |
| Tiruvanantapuram        | 35.1824193548387   | 18.789526184538644 |
| Purulia                 | 34.857391304347836 | 22.717619047619046 |
| Haldwani                | 34.75887096774194  | 27.97550847457628  |
| Gorakhpur               | 34.559622641509435 | 23.165789473684207 |
| Allahabad               | 34.15063492063492  | 27.689669421487594 |
| Warangal                | 34.06282608695653  | 34.450101010101015 |
| Bikaner                 | 33.69960000000001  | 34.8436170212766   |
| Darbhanga               | 33.6665306122449   | 24.579157894736845 |
| T.Puram                 | 33.66082089552238  | 36.40703703703704  |
| Balasore                | 33.621551724137944 | 27.286039603960393 |
| Kharagpur               | 33.59275862068966  | 30.005315315315308 |
| Gumla                   | 33.58441176470588  | 28.73022988505747  |
| Saharsa                 | 33.52680851063829  | 28.79076923076922  |
| Haridwar                | 33.48571428571429  | 26.48780701754387  |
| Baripada                | 33.480624999999996 | 26.589000000000002 |
| Siliguri                | 33.31627659574467  | 24.27              |
| Ernakulam               | 33.30257575757577  | 33.43887434554975  |
| Dantewada               | 33.19790697674418  | 36.22936708860758  |
| Solan                   | 32.98714285714287  | 28.09524390243903  |
| Bokaro                  | 32.9448717948718   | 25.78241379310345  |
| Jeypore                 | 32.62423728813561  | 28.00767441860466  |
| Dhanbad                 | 32.354615384615386 | 28.25348837209302  |
| Surat                   | 32.343999999999994 | 27.936395348837205 |
| Palakkad                | 32.29086956521739  | 38.969067796610176 |
| Panaji                  | 31.753025641025648 | 32.79130919220057  |
| Araria                  | 31.746000000000002 | 27.904303797468362 |
| Berhampur               | 31.62954545454545  | 28.776581196581194 |
| Kolkata                 | 31.548120805369134 | 18.408392857142854 |
| Rampurhat               | 31.480377358490568 | 30.56127906976746  |
```

Relevance

The correlation between prices across different food groups reveals important market dynamics. If a market shows consistently high prices across categories, this suggests underlying structural issues like transportation bottlenecks or higher local operating costs, rather than commodity-specific problems. This helps distinguish between localized supply chain issues and broader regional economic challenges.

Q8) Rank commodities within each state based on their average price, and identify the top 3 most expensive commodities per state.

```
WITH ranked_prices AS (
  SELECT
    state,
    commodity,
    AVG(price) AS avg_price,
    RANK() OVER (PARTITION BY state ORDER BY AVG(price) DESC) AS price_rank
  FROM food_prices
  GROUP BY state, commodity
)
SELECT
  state,
  commodity,
  avg_price,
  price_rank
FROM ranked_prices
WHERE price_rank <= 3
ORDER BY state, price_rank;
```

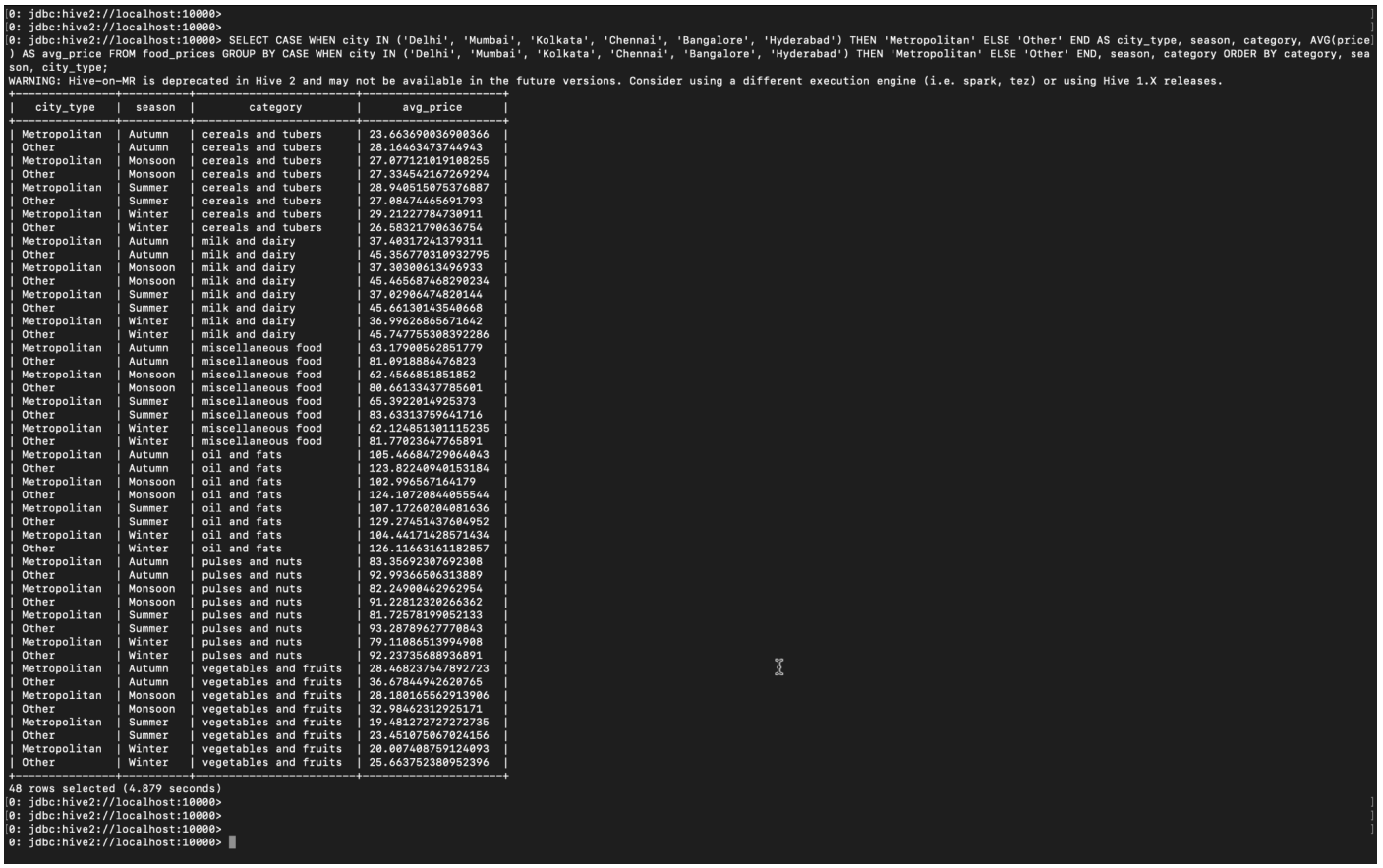
```
0: jdbc:hive2://localhost:10000> WITH ranked_prices AS ( SELECT state, commodity, AVG(price) AS avg_price, RANK() OVER (PARTITION BY state ORDER BY AVG(price) DESC) AS price_rank FROM food_prices GROUP BY
 state, commodity ) SELECT state, commodity, avg_price, price_rank FROM ranked_prices WHERE price_rank <= 3 ORDER BY state, price_rank;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+---------------------+------------------+--------------------+-------------+
|        state        |     commodity    |      avg_price     | price_rank  |
+---------------------+------------------+--------------------+-------------+
| Andaman And Nicobar | Tea (black)      | 271.0917924528302  | 1           |
| Andaman And Nicobar | Oil (mustard)    | 175.07275          | 2           |
| Andaman And Nicobar | Ghee (vanaspati) | 159.01962962962966 | 3           |
| Andhra Pradesh      | Tea (black)      | 281.17457627118637 | 1           |
| Andhra Pradesh      | Oil (mustard)    | 151.1039106145251  | 2           |
| Andhra Pradesh      | Oil (groundnut)  | 142.51570707070712 | 3           |
| Assam               | Oil (groundnut)  | 194.61777777777777 | 1           |
| Assam               | Tea (black)      | 181.74469565217393 | 2           |
| Assam               | Oil (sunflower)  | 128.10233766233767 | 3           |
| Bihar               | Tea (black)      | 255.38183448275876 | 1           |
| Bihar               | Oil (groundnut)  | 188.43056874766635 | 2           |
| Bihar               | Oil (sunflower)  | 152.55872017353573 | 3           |
| Chandigarh          | Tea (black)      | 227.92811965811964 | 1           |
| Chandigarh          | Oil (groundnut)  | 139.12076923076924 | 2           |
| Chandigarh          | Oil (palm)       | 118.8790566037736  | 3           |
| Chhattisgarh        | Tea (black)      | 290.05765765765767 | 1           |
| Chhattisgarh        | Oil (groundnut)  | 179.65989583333328 | 2           |
| Chhattisgarh        | Oil (mustard)    | 165.59371681415925 | 3           |
| Delhi               | Tea (black)      | 221.97128205128203 | 1           |
| Delhi               | Oil (groundnut)  | 171.78246376811592 | 2           |
| Delhi               | Oil (sunflower)  | 129.1298969072165  | 3           |
| Goa                 | Tea (black)      | 241.61477477477482 | 1           |
| Goa                 | Oil (groundnut)  | 178.14101123595506 | 2           |
| Goa                 | Oil (mustard)    | 158.98766233766233 | 3           |
| Gujarat             | Tea (black)      | 265.50549242424233 | 1           |
| Gujarat             | Oil (groundnut)  | 148.2668120805369  | 2           |
| Gujarat             | Oil (soybean)    | 126.17003861003855 | 3           |
| Haryana             | Tea (black)      | 213.41817973478261 | 1           |
| Haryana             | Oil (groundnut)  | 150.06591511936344 | 2           |
| Haryana             | Oil (sunflower)  | 115.97388888888885 | 3           |
| Himachal Pradesh    | Tea (black)      | 233.44239263803686 | 1           |
| Himachal Pradesh    | Oil (groundnut)  | 167.80349206349206 | 2           |
| Himachal Pradesh    | Oil (sunflower)  | 166.56114942528734 | 3           |
| Jharkhand           | Tea (black)      | 258.7871578947369  | 1           |
| Jharkhand           | Oil (groundnut)  | 177.88599999999994 | 2           |
| Jharkhand           | Oil (sunflower)  | 157.0774537037037  | 3           |
| Karnataka           | Tea (black)      | 286.65858880778586 | 1           |
| Karnataka           | Oil (mustard)    | 164.5722434367541  | 2           |
| Karnataka           | Oil (groundnut)  | 156.9955649717515  | 3           |
| Kerala              | Tea (black)      | 204.70535087719293 | 1           |
| Kerala              | Oil (groundnut)  | 166.3139837398374  | 2           |
| Kerala              | Oil (mustard)    | 140.5805893909625  | 3           |
| Madhya Pradesh      | Tea (black)      | 247.42497326203215 | 1           |
| Madhya Pradesh      | Oil (groundnut)  | 150.5496641791044  | 2           |
| Madhya Pradesh      | Oil (sunflower)  | 128.1180962800875  | 3           |
| Maharashtra         | Tea (black)      | 242.90563157894738 | 1           |
| Maharashtra         | Oil (groundnut)  | 154.1169019607842  | 2           |
| Maharashtra         | Oil (sunflower)  | 121.15905707196026 | 3           |
| Manipur             | Tea (black)      | 263.995            | 1           |
| Manipur             | Oil (soybean)    | 179.18789473684208 | 2           |
| Manipur             | Oil (sunflower)  | 178.14888888888888 | 3           |
| Meghalaya           | Tea (black)      | 287.58265193370164 | 1           |
| Meghalaya           | Oil (sunflower)  | 148.70181818181814 | 2           |
| Meghalaya           | Oil (soybean)    | 128.4872641509434  | 3           |
| Mizoram             | Tea (black)      | 268.6525454545455  | 1           |
| Mizoram             | Oil (groundnut)  | 171.77599999999998 | 2           |
| Mizoram             | Ghee (vanaspati) | 168.62             | 3           |
```

Relevance

Ranking commodities by price within each state creates a clear picture of local food affordability. Identifying the top 3 most expensive items per state reveals which essential foods might be out of reach for lower-income households in those regions.

Q9) How do food prices in metropolitan cities compare to other markets during different seasons of the year?

```
SELECT
  CASE
    WHEN city IN ('Delhi', 'Mumbai', 'Kolkata', 'Chennai', 'Bangalore', 'Hyderabad')
THEN 'Metropolitan'
    ELSE 'Other'
  END AS city_type,
  season,
  category,
  AVG(price) AS avg_price
FROM food_prices
GROUP BY
  CASE
    WHEN city IN ('Delhi', 'Mumbai', 'Kolkata', 'Chennai', 'Bangalore', 'Hyderabad')
THEN 'Metropolitan'
    ELSE 'Other'
  END,
  season,
  category
ORDER BY
  category,
  season,
  city_type;
```

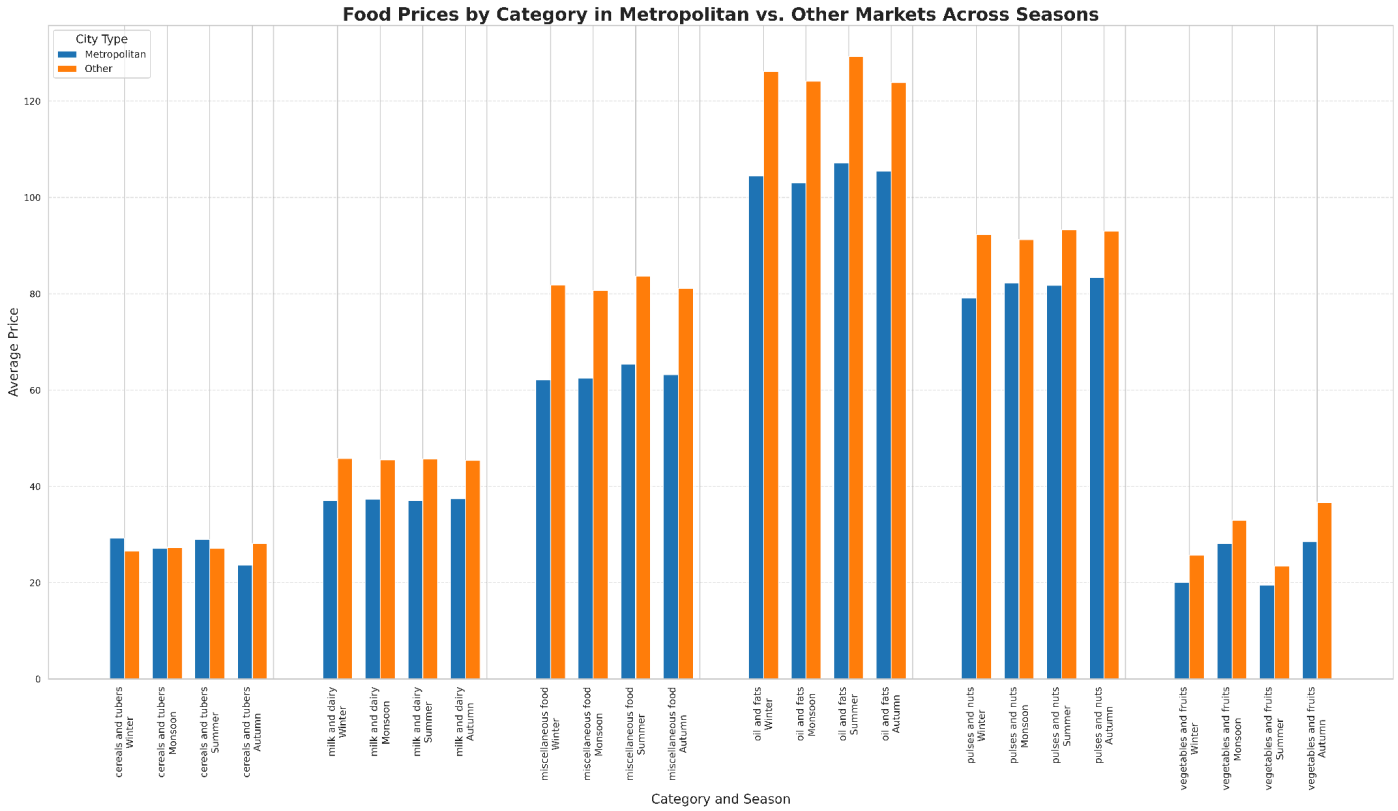


```
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000> SELECT CASE WHEN city IN ('Delhi', 'Mumbai', 'Kolkata', 'Chennai', 'Bangalore', 'Hyderabad') THEN 'Metropolitan' ELSE 'Other' END AS city_type, season, category, AVG(price
) AS avg_price FROM food_prices GROUP BY CASE WHEN city IN ('Delhi', 'Mumbai', 'Kolkata', 'Chennai', 'Bangalore', 'Hyderabad') THEN 'Metropolitan' ELSE 'Other' END, season, category ORDER BY category, sea
son, city_type;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+--------------+---------+----------------------+--------------------+
|   city_type  |  season |       category       |      avg_price     |
+--------------+---------+----------------------+--------------------+
| Metropolitan | Autumn  | cereals and tubers   | 23.663690036900366 |
| Other        | Autumn  | cereals and tubers   | 28.164634737444943 |
| Metropolitan | Monsoon | cereals and tubers   | 27.077121019108255 |
| Other        | Monsoon | cereals and tubers   | 27.334542167269294 |
| Metropolitan | Summer  | cereals and tubers   | 28.940515075376887 |
| Other        | Summer  | cereals and tubers   | 27.08474465691793  |
| Metropolitan | Winter  | cereals and tubers   | 29.21227784730911  |
| Other        | Winter  | cereals and tubers   | 26.583217906367 54 |
| Metropolitan | Autumn  | milk and dairy       | 37.40317241379311  |
| Other        | Autumn  | milk and dairy       | 45.356770310932795 |
| Metropolitan | Monsoon | milk and dairy       | 37.30300613496933  |
| Other        | Monsoon | milk and dairy       | 45.465687468290234 |
| Metropolitan | Summer  | milk and dairy       | 37.02906474820144  |
| Other        | Summer  | milk and dairy       | 45.66130143540668  |
| Metropolitan | Winter  | milk and dairy       | 36.99626865671642  |
| Other        | Winter  | milk and dairy       | 45.7477553083922 86 |
| Metropolitan | Autumn  | miscellaneous food   | 63.17900562851779  |
| Other        | Autumn  | miscellaneous food   | 81.0918886476823   |
| Metropolitan | Monsoon | miscellaneous food   | 62.4566851851852   |
| Other        | Monsoon | miscellaneous food   | 80.66133437785601  |
| Metropolitan | Summer  | miscellaneous food   | 65.3922014925373   |
| Other        | Summer  | miscellaneous food   | 83.633137596417 16 |
| Metropolitan | Winter  | miscellaneous food   | 62.124851301115235 |
| Other        | Winter  | miscellaneous food   | 81.770236477658 91 |
| Metropolitan | Autumn  | oil and fats         | 105.46684729064043 |
| Other        | Autumn  | oil and fats         | 123.82240940153184 |
| Metropolitan | Monsoon | oil and fats         | 102.996567164179   |
| Other        | Monsoon | oil and fats         | 124.10720844055544 |
| Metropolitan | Summer  | oil and fats         | 107.17260204081636 |
| Other        | Summer  | oil and fats         | 129.27451437604952 |
| Metropolitan | Winter  | oil and fats         | 104.44171428571434 |
| Other        | Winter  | oil and fats         | 126.11663161182857 |
| Metropolitan | Autumn  | pulses and nuts      | 83.35692307692308  |
| Other        | Autumn  | pulses and nuts      | 92.99366506313889  |
| Metropolitan | Monsoon | pulses and nuts      | 82.24900462962954  |
| Other        | Monsoon | pulses and nuts      | 91.22812320266362  |
| Metropolitan | Summer  | pulses and nuts      | 81.72578199052133  |
| Other        | Summer  | pulses and nuts      | 93.2878962770843   |
| Metropolitan | Winter  | pulses and nuts      | 79.11086513994908  |
| Other        | Winter  | pulses and nuts      | 92.23735688936891  |
| Metropolitan | Autumn  | vegetables and fruits | 28.468237547892723 |
| Other        | Autumn  | vegetables and fruits | 36.67844942620765  |
| Metropolitan | Monsoon | vegetables and fruits | 28.180165562913906 |
| Other        | Monsoon | vegetables and fruits | 32.98462312925171  |
| Metropolitan | Summer  | vegetables and fruits | 19.481272727272735 |
| Other        | Summer  | vegetables and fruits | 23.451075067024156 |
| Metropolitan | Winter  | vegetables and fruits | 20.007408759124093 |
| Other        | Winter  | vegetables and fruits | 25.663752380952396 |
+--------------+---------+----------------------+--------------------+
48 rows selected (4.879 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Relevance

Comparing metropolitan and non-metropolitan food prices across seasons reveals urban-rural divides and seasonal vulnerabilities. This analysis helps understand whether urban consumers consistently pay premiums regardless of season, or if there are specific times when the urban-rural price gap widens or narrows, which impacts migration patterns and urban poverty levels.

Visualization



Q10) Analyse the evolution of commodity prices over time, revealing year-to-year price dynamics and identifying significant market trends.

```
SELECT
    commodity,
    year,
    AVG(price) AS avg_price,
    LAG(AVG(price), 1) OVER (PARTITION BY commodity ORDER BY year) AS
prev_year_price,
    ((AVG(price) - LAG(AVG(price), 1) OVER (PARTITION BY commodity ORDER BY year)) /
LAG(AVG(price), 1) OVER (PARTITION BY commodity ORDER BY year)) * 100 AS pct_change
FROM
    food_prices
GROUP BY
    commodity, year
ORDER BY
    commodity, year;
```

```
0: jdbc:hive2://localhost:10000> SELECT commodity, year, AVG(price) AS avg_price, LAG(AVG(price), 1) OVER (PARTITION BY commodity ORDER BY year) AS prev_year_price, ((AVG(price) - LAG(AVG(price), 1) OVER
(PARTITION BY commodity ORDER BY year)) / LAG(AVG(price), 1) OVER (PARTITION BY commodity ORDER BY year)) * 100 AS pct_change FROM food_prices GROUP BY commodity, year ORDER BY commodity, year;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+----------------------+-------+---------------------+---------------------+---------------------+
|      commodity       | year  |      avg_price      |   prev_year_price   |      pct_change     |
+----------------------+-------+---------------------+---------------------+---------------------+
| Chickpeas            | 2000  | 20.351785714285715  | NULL                | NULL                |
| Chickpeas            | 2001  | 25.371153846153845  | 20.351785714285715  | 24.663035481671958  |
| Chickpeas            | 2002  | 23.17241379310345   | 25.371153846153845  | -8.66629900391273   |
| Chickpeas            | 2003  | 22.318648648648647  | 23.17241379310345   | -3.6844031531531596 |
| Chickpeas            | 2004  | 22.288918918918917  | 22.318648648648647  | -0.13320577870887634 |
| Chickpeas            | 2005  | 24.29259259259259   | 22.288918918918917  | 8.989550731296116   |
| Chickpeas            | 2006  | 32.574411764705886  | 24.29259259259259   | 34.09195268288748   |
| Chickpeas            | 2007  | 33.986216216216214  | 32.574411764705886  | 4.334090394964574   |
| Chickpeas            | 2008  | 35.53484848484849   | 33.986216216216214  | 4.556648079857031   |
| Chickpeas            | 2009  | 34.349000000000004  | 35.53484848484849   | -3.337142369846075  |
| Chickpeas            | 2010  | 34.37694444444444   | 34.349000000000004  | 0.08135446285026217 |
| Chickpeas            | 2011  | 43.132580645161305  | 34.37694444444444   | 25.469500975767605  |
| Chickpeas            | 2012  | 61.17241379310346   | 43.132580645161305  | 41.824145177749514  |
| Chickpeas            | 2013  | 55.34124999999999   | 61.17241379310346   | -9.53234216459981   |
| Chickpeas            | 2014  | 49.296388888888885  | 55.34124999999999   | -10.922885028999353 |
| Chickpeas            | 2015  | 61.40432432432433   | 49.296388888888885  | 24.56150583915184   |
| Chickpeas            | 2016  | 102.06764705882352  | 61.40432432432433   | 66.22224604202847   |
| Chickpeas            | 2017  | 86.30264705882352   | 102.06764705882352  | -15.445638705587413 |
| Chickpeas            | 2018  | 66.83058823529413   | 86.30264705882352   | -22.562527902831672 |
| Chickpeas            | 2019  | 69.66894736842104   | 66.83058823529413   | 4.247095840506067   |
| Chickpeas            | 2020  | 73.71807692307692   | 69.66894736842104   | 5.811957418049397   |
| Ghee (vanaspati)     | 2015  | 73.74183266932273   | NULL                | NULL                |
| Ghee (vanaspati)     | 2016  | 75.17497164461251   | 73.74183266932273   | 1.9436545134189185  |
| Ghee (vanaspati)     | 2017  | 78.717572815534     | 75.17497164461251   | 4.712474236330983   |
| Ghee (vanaspati)     | 2018  | 80.92605042016805   | 78.717572815534     | 2.805712665447397   |
| Ghee (vanaspati)     | 2019  | 81.54829341317367   | 80.92605042016805   | 0.768903202089674   |
| Ghee (vanaspati)     | 2020  | 94.35707717569792   | 81.54829341317367   | 15.706991803773382  |
| Ghee (vanaspati)     | 2021  | 129.52185761047468  | 94.35707717569792   | 37.267772049888805  |
| Ghee (vanaspati)     | 2022  | 150.76835451977402  | 129.52185761047468  | 16.403792611743004  |
| Ghee (vanaspati)     | 2023  | 133.78655172413784  | 150.76835451977402  | -11.263506091663908 |
| Ghee (vanaspati)     | 2024  | 127.25              | 133.78655172413784  | -4.885806263708723  |
| Lentils              | 2013  | 51.81151898734176   | NULL                | NULL                |
| Lentils              | 2014  | 47.10025125628143   | 51.81151898734176   | -9.09308938078298   |
| Lentils              | 2015  | 58.396993736952     | 47.10025125628143   | 23.98446330828012   |
| Lentils              | 2016  | 92.01198294243065   | 58.396993736952     | 57.56287619341614   |
| Lentils              | 2017  | 85.8033264033264    | 92.01198294243065   | -6.747660837815916  |
| Lentils              | 2018  | 64.62271428571425   | 85.8033264033264    | -24.68407108693051?  |
| Lentils              | 2019  | 81.58068728522336   | 64.62271428571425   | 26.24150308076956   |
| Lentils              | 2020  | 97.82982935153578   | 81.58068728522336   | 19.91787836930323   |
| Lentils              | 2021  | 105.51795180722891  | 97.82982935153578   | 7.8586689833293075  |
| Lentils              | 2022  | 106.36427891156455  | 105.51795180722891  | 0.8020693065401806  |
| Lentils              | 2023  | 120.50394495412827  | 106.36427891156455  | 13.29362280951483   |
| Lentils              | 2024  | 149.992             | 120.50394495412827  | 24.470613851767933  |
| Lentils (masur)      | 2012  | 52.6404961832060 96 | NULL                | NULL                |
| Lentils (masur)      | 2013  | 57.222027397260256  | 52.6404961832060 96 | 8.703434705685403   |
| Lentils (masur)      | 2014  | 66.76578048780485   | 57.222027397260256  | 16.67846024449238   |
| Lentils (masur)      | 2015  | 82.39393406593409   | 66.76578048780485   | 23.40743036919012   |
| Lentils (masur)      | 2016  | 82.54581740976641   | 82.39393406593409   | 0.1843380165714415  |
| Lentils (masur)      | 2017  | 68.66752066115703   | 82.54581740976641   | -16.812840655167314 |
| Lentils (masur)      | 2018  | 60.76174568965517   | 68.66752066115703   | -11.513121335067941 |
| Lentils (masur)      | 2019  | 63.17878962536023   | 60.76174568965517   | 3.977904038587501   |
| Lentils (masur)      | 2020  | 75.57837795275589   | 63.17878962536023   | 19.62618847388999   |
| Lentils (masur)      | 2021  | 88.16298056155507   | 75.57837795275589   | 16.65106204934146   |
| Lentils (masur)      | 2022  | 95.99739299610903   | 88.16298056155507   | 8.886283545148517   |
| Lentils (masur)      | 2023  | 91.77799783549781   | 95.99739299610903   | -4.395322652962289  |
| Lentils (masur)      | 2024  | 93.795              | 91.77799783549781   | 2.197696846816649   |
| Lentils (moong)      | 2015  | 102.10896551724146  | NULL                | NULL                |
```

Relevance

Tracking year-to-year price dynamics exposes longer-term food security threats or improvements. By analyzing how prices evolve over time, we can separate random fluctuations from genuine trends, helping forecast future prices and identify which commodities are becoming increasingly unaffordable or more accessible over time.

Q11) Identify the top 5 markets with the highest average price for each commodity, considering only the latest year.

```
WITH latest_year AS (
    SELECT MAX(year) AS max_year FROM food_prices
),
ranked_data AS (
    SELECT
        fp.year, fp.state, fp.market, fp.commodity,
        AVG(fp.price) AS avg_price,
        RANK() OVER (PARTITION BY fp.commodity ORDER BY AVG(fp.price) DESC) AS
market_rank
    FROM food_prices fp
    JOIN latest_year ly ON fp.year = ly.max_year
    GROUP BY fp.year, fp.state, fp.market, fp.commodity
)
SELECT
    year, state, market, commodity, avg_price, market_rank
FROM ranked_data WHERE market_rank <= 5
ORDER BY commodity, market_rank, state, year;
```

```
0: jdbc:hive2://localhost:10000> WITH latest_year AS ( SELECT MAX(year) AS max_year FROM food_prices ), ranked_data AS ( SELECT fp.year, fp.state, fp.market, fp.commodity, AVG(fp.price) AS avg_price, RANK
() OVER (PARTITION BY fp.commodity ORDER BY AVG(fp.price) DESC) AS market_rank FROM food_prices fp JOIN latest_year ly ON fp.year = ly.max_year GROUP BY fp.year, fp.state, fp.market, fp.commodity ) SELECT
 year, state, market, commodity, avg_price, market_rank FROM ranked_data WHERE market_rank <= 5 ORDER BY commodity, market_rank, state, year;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20250322155742_b86d1402-20f6-40a6-a7d1-90b282c296e6.log
2025-03-22 15:57:50     Starting to launch local task to process map join;      maximum memory = 477626368
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
2025-03-22 15:57:53     Dump the side-table for tag: 0 with group count: 31 into file: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_15-57-42_406_3662607117817411882-1/-local-10008/H
ashTable-Stage-2/MapJoin-mapfile70--.hashtable
2025-03-22 15:57:53     Uploaded 1 File to: file:/tmp/root/cf169404-99eb-45eb-b479-2d9ae48f96f4/hive_2025-03-22_15-57-42_406_3662607117817411882-1/-local-10008/HashTable-Stage-2/MapJoin-mapfile70--.hashta
ble (7338399 bytes)
2025-03-22 15:57:53     End of local task; Time Taken: 3.391 sec.
+-------+--------------+-----------------+---------------------+-----------+--------------+
| year  |    state     |      market     |      commodity      | avg_price | market_rank  |
+-------+--------------+-----------------+---------------------+-----------+--------------+
| 2024  | Assam        | North-East Zone | Ghee (vanaspati)    | 143.88    | 1            |
| 2024  | Rajasthan    | North Zone      | Ghee (vanaspati)    | 127.82    | 2            |
| 2024  | Karnataka    | South Zone      | Ghee (vanaspati)    | 125.15    | 3            |
| 2024  | Maharashtra  | West Zone       | Ghee (vanaspati)    | 122.68    | 4            |
| 2024  | Orissa       | East Zone       | Ghee (vanaspati)    | 116.72    | 5            |
| 2024  | Karnataka    | South Zone      | Lentils             | 165.57    | 1            |
| 2024  | Maharashtra  | West Zone       | Lentils             | 148.04    | 2            |
| 2024  | Orissa       | East Zone       | Lentils             | 147.64    | 3            |
| 2024  | Rajasthan    | North Zone      | Lentils             | 147.53    | 4            |
| 2024  | Assam        | North-East Zone | Lentils             | 141.18    | 5            |
| 2024  | Karnataka    | South Zone      | Lentils (masur)     | 101.14    | 1            |
| 2024  | Assam        | North-East Zone | Lentils (masur)     | 95.89     | 2            |
| 2024  | Maharashtra  | West Zone       | Lentils (masur)     | 91.54     | 3            |
| 2024  | Orissa       | East Zone       | Lentils (masur)     | 86.61     | 4            |
| 2024  | Karnataka    | South Zone      | Lentils (moong)     | 124.26    | 1            |
| 2024  | Assam        | North-East Zone | Lentils (moong)     | 118.29    | 2            |
| 2024  | Orissa       | East Zone       | Lentils (moong)     | 115.17    | 3            |
| 2024  | Maharashtra  | West Zone       | Lentils (moong)     | 112.64    | 4            |
| 2024  | Rajasthan    | North Zone      | Lentils (moong)     | 112.64    | 4            |
| 2024  | Karnataka    | South Zone      | Lentils (urad)      | 140.15    | 1            |
| 2024  | Orissa       | East Zone       | Lentils (urad)      | 118.47    | 2            |
| 2024  | Maharashtra  | West Zone       | Lentils (urad)      | 116.7     | 3            |
| 2024  | Assam        | North-East Zone | Milk (pasteurized)  | 71.83     | 1            |
| 2024  | Rajasthan    | North Zone      | Milk (pasteurized)  | 60.3      | 2            |
| 2024  | Maharashtra  | West Zone       | Milk (pasteurized)  | 57.1      | 3            |
| 2024  | Orissa       | East Zone       | Milk (pasteurized)  | 55.15     | 4            |
| 2024  | Karnataka    | South Zone      | Milk (pasteurized)  | 51.8      | 5            |
| 2024  | Rajasthan    | North Zone      | Oil (groundnut)     | 192.87    | 1            |
| 2024  | Karnataka    | South Zone      | Oil (groundnut)     | 192.75    | 2            |
| 2024  | Assam        | North-East Zone | Oil (groundnut)     | 192.01    | 3            |
| 2024  | Maharashtra  | West Zone       | Oil (groundnut)     | 189.81    | 4            |
| 2024  | Orissa       | East Zone       | Oil (groundnut)     | 185.04    | 5            |
| 2024  | Assam        | North-East Zone | Oil (mustard)       | 149.64    | 1            |
| 2024  | Orissa       | East Zone       | Oil (mustard)       | 133.0     | 2            |
| 2024  | Assam        | North-East Zone | Oil (palm)          | 118.25    | 1            |
| 2024  | Rajasthan    | North Zone      | Oil (palm)          | 109.58    | 2            |
| 2024  | Maharashtra  | West Zone       | Oil (palm)          | 105.68    | 3            |
| 2024  | Orissa       | East Zone       | Oil (palm)          | 99.95     | 4            |
| 2024  | Karnataka    | South Zone      | Oil (palm)          | 93.78     | 5            |
| 2024  | Karnataka    | South Zone      | Oil (soybean)       | 137.5     | 1            |
| 2024  | Assam        | North-East Zone | Oil (soybean)       | 136.11    | 2            |
```

Relevance

Identifying markets with the highest prices for each commodity in the latest year reveals current affordability hotspots. This time-sensitive information shows where consumers are currently facing the greatest pressure, helping target immediate relief efforts and understand whether high-price areas are persistent or shifting over time.

Q12) Analyze the impact of seasonality on price volatility to calculate the coefficient of variation for each commodity.

```
SELECT
    commodity,
    season,
    AVG(price) AS avg_price,
    STDDEV(price) AS stddev_price,
    (STDDEV(price) / AVG(price)) * 100 AS coeff_variation
FROM
    food_prices
GROUP BY
    commodity, season
ORDER BY
    commodity, season;
```

```
0: jdbc:hive2://localhost:10000> SELECT commodity, season, AVG(price) AS avg_price, STDDEV(price) AS stddev_price, (STDDEV(price) / AVG(price)) * 100 AS coeff_variation FROM food_prices GROUP BY commodity
, season ORDER BY commodity, season;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+----------------------+-----------+---------------------+---------------------+---------------------+
|      commodity       |  season   |      avg_price      |    stddev_price     |   coeff_variation   |
+----------------------+-----------+---------------------+---------------------+---------------------+
| Chickpeas            | Autumn    | 49.754468085106375  | 28.042488075829223  | 56.361748311451706  |
| Chickpeas            | Monsoon   | 46.909532163742696  | 24.290053021854867  | 51.78063370375953   |
| Chickpeas            | Summer    | 44.472987804878066  | 21.59098310496481   | 48.54853287504237   |
| Chickpeas            | Winter    | 46.80564705882355   | 21.80289842831527   | 46.58176907780854   |
| Ghee (vanaspati)     | Autumn    | 113.95461485079807  | 33.5624008777614    | 29.45242798784849   |
| Ghee (vanaspati)     | Monsoon   | 112.47162805662812  | 36.7900434723959115 | 32.710510293202624  |
| Ghee (vanaspati)     | Summer    | 116.17740543735212  | 37.04024238547987   | 31.882483729121986  |
| Ghee (vanaspati)     | Winter    | 113.03012820512828  | 33.152888511413764  | 29.331019116644324  |
| Lentils              | Autumn    | 94.2528065241845    | 25.48180066602516   | 27.035588228862693  |
| Lentils              | Monsoon   | 91.7355100864553    | 26.11772144431787   | 28.4706777339587    |
| Lentils              | Summer    | 92.26643200000005   | 26.130279431393582  | 28.320461586065846  |
| Lentils              | Winter    | 91.65946149392009   | 24.6674296250659    | 26.912038564291723  |
| Lentils (masur)      | Autumn    | 81.19141884816754   | 17.29638372843036   | 21.303216489879993  |
| Lentils (masur)      | Monsoon   | 80.35036958817312   | 17.042119003749384  | 21.209758077152436  |
| Lentils (masur)      | Summer    | 80.6702481751827    | 17.318351176742798  | 21.468077226109976  |
| Lentils (masur)      | Winter    | 80.50281035393547   | 17.05440769081658   | 21.184860026421244  |
| Lentils (moong)      | Autumn    | 96.67383948676      | 15.765002920921772  | 16.307413614271248  |
| Lentils (moong)      | Monsoon   | 95.2164227642277    | 16.363077653067208  | 17.185142203446365  |
| Lentils (moong)      | Summer    | 99.73570005858242   | 15.755389300069254  | 15.797141134834272  |
| Lentils (moong)      | Winter    | 97.42632936507931   | 14.969788232995317  | 15.365238873877725  |
| Lentils (urad)       | Autumn    | 104.39674490426195  | 21.442549252237168  | 20.539480681989907  |
| Lentils (urad)       | Monsoon   | 101.62102439024412  | 22.435857972785777  | 22.07796871504443   |
| Lentils (urad)       | Summer    | 104.41640468227428  | 21.496758096178052  | 20.58752948024779   |
| Lentils (urad)       | Winter    | 103.64351145038167  | 19.822549978899403  | 19.125702807724085  |
| Milk                 | Autumn    | 35.532849090909915  | 8.138834438673918   | 22.905104771939568  |
| Milk                 | Monsoon   | 35.38723958333334   | 8.531219577778227   | 24.108180457783594  |
| Milk                 | Summer    | 35.00424581005587   | 8.684899298658884   | 24.810988203505083  |
| Milk                 | Winter    | 33.722040816326526  | 8.717280452533918   | 25.850394108749934  |
| Milk (pasteurized)   | Autumn    | 45.650066225165546  | 9.719827247305814   | 21.29203317989396   |
| Milk (pasteurized)   | Monsoon   | 45.77698764160661   | 9.795812985569446   | 21.398989951593173  |
| Milk (pasteurized)   | Summer    | 46.00653658536583   | 10.529777465273124  | 22.887568260512204  |
| Milk (pasteurized)   | Winter    | 46.365887265135676  | 9.885501176113857   | 21.320634110990373  |
| Oil (groundnut)      | Autumn    | 153.94389819156058  | 33.56972864832424   | 21.806469137576105  |
| Oil (groundnut)      | Monsoon   | 153.7168546195652   | 33.981839502736364  | 22.106775204864963  |
| Oil (groundnut)      | Summer    | 160.67317934782636  | 34.8125618912947    | 21.666691374751622  |
| Oil (groundnut)      | Winter    | 156.36430965147474  | 33.17056533763519   | 21.213642302114906  |
| Oil (mustard)        | Autumn    | 119.11686447302903  | 48.22867922773225   | 40.48854604176614   |
| Oil (mustard)        | Monsoon   | 117.6244642180993   | 47.353781860787755  | 40.258446383215286  |
| Oil (mustard)        | Summer    | 121.93750907770507  | 49.360276223762014  | 40.47997748773678   |
| Oil (mustard)        | Winter    | 121.98081843464443  | 49.219160289225634  | 40.34991806158158   |
| Oil (palm)           | Autumn    | 100.36812461820418  | 28.14379964768055   | 28.040575386596384  |
| Oil (palm)           | Monsoon   | 102.12359085290491  | 31.43967516428587   | 30.78590842890594   |
| Oil (palm)           | Summer    | 110.49919703520692  | 33.81725777260614   | 30.60407557697581   |
| Oil (palm)           | Winter    | 102.9246882793018   | 28.493524459511413  | 27.68385791190341   |
| Oil (soybean)        | Autumn    | 121.68612977099247  | 33.51874973617435   | 27.545250883773733  |
| Oil (soybean)        | Monsoon   | 123.62226608187152  | 35.53209085307518   | 28.742468472098132  |
| Oil (soybean)        | Summer    | 127.45671184022824  | 36.446462525588408  | 28.595169292670196  |
| Oil (soybean)        | Winter    | 124.34885074626871  | 31.621439423589873  | 25.4296193620813    |
| Oil (sunflower)      | Autumn    | 130.58737175792484  | 37.1967301916985    | 28.484170935495662  |
| Oil (sunflower)      | Monsoon   | 132.02059468822162  | 40.24712353905911   | 30.48548874825649   |
| Oil (sunflower)      | Summer    | 138.36912429378535  | 40.78388418521634   | 29.474699932785576  |
| Oil (sunflower)      | Winter    | 133.11793522267217  | 35.344641650796554  | 26.551374607541824  |
| Onions               | Autumn    | 35.2205943889681    | 17.560165449989633  | 49.857663547807356  |
| Onions               | Monsoon   | 26.1885430463576    | 10.989728908907981  | 41.963880500929484  |
| Onions               | Summer    | 20.67624406457738   | 7.820883055729173   | 37.82545336233451   |
| Onions               | Winter    | 27.431321671525744  | 11.943148801996658  | 43.53836444706886   |
| Potatoes             | Autumn    | 24.414113345521     | 10.348810871905663  | 42.388641051362406  |
```

Relevance

High CV: A high coefficient of variation indicates that a commodity's price is highly volatile relative to its average price. This suggests that the commodity is more susceptible to seasonal changes or external factors affecting its price.

Low CV: A low CV suggests that the commodity's price is relatively stable, with less fluctuation around the mean. This can indicate a more predictable pricing pattern, which is beneficial for planning and forecasting.

Q13) Identify the cheapest and costliest state for each commodity for a given year? Take the year 2024 for example.

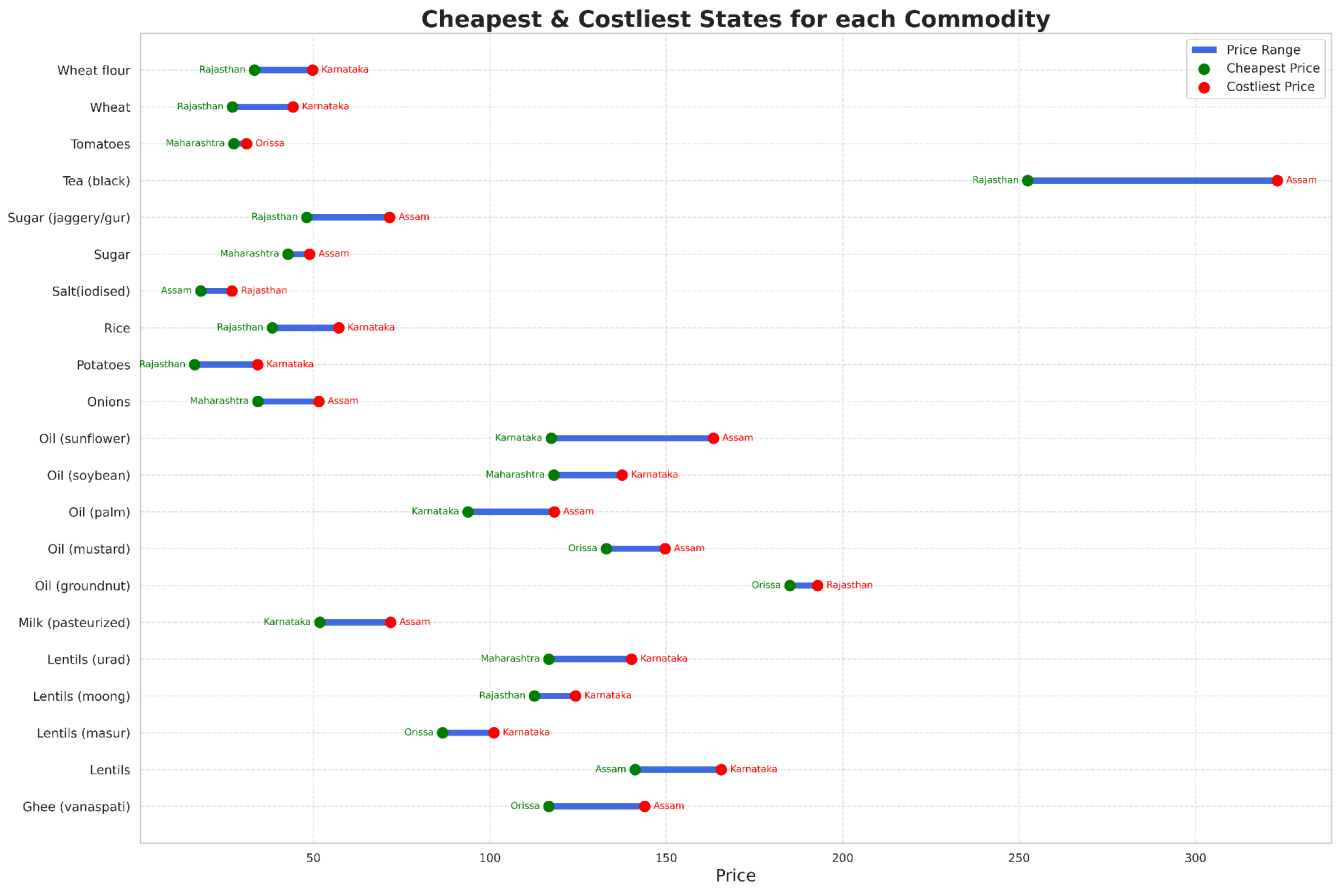
```
WITH price_extremes AS (
    SELECT
        commodity,
        state,
        MIN(price) AS min_price,
        MAX(price) AS max_price
    FROM food_prices
    WHERE year = 2024
    GROUP BY commodity, state
),
ranked_extremes AS (
    SELECT
        commodity,
        state AS cheapest_state,
        min_price AS cheapest_price,
        NULL AS costliest_state,
        NULL AS costliest_price,
        RANK() OVER (PARTITION BY commodity ORDER BY min_price ASC) AS cheapest_rank,
        RANK() OVER (PARTITION BY commodity ORDER BY max_price DESC) AS
costliest_rank
    FROM price_extremes
    UNION ALL
    SELECT
        commodity,
        NULL AS cheapest_state,
        NULL AS cheapest_price,
        state AS costliest_state,
        max_price AS costliest_price,
        RANK() OVER (PARTITION BY commodity ORDER BY min_price ASC) AS cheapest_rank,
        RANK() OVER (PARTITION BY commodity ORDER BY max_price DESC) AS
costliest_rank
    FROM price_extremes
)
SELECT
    commodity,
    MAX(CASE WHEN cheapest_rank = 1 THEN cheapest_state END) AS cheapest_state,
    MAX(CASE WHEN cheapest_rank = 1 THEN cheapest_price END) AS cheapest_state_price,
    MAX(CASE WHEN costliest_rank = 1 THEN costliest_state END) AS costliest_state,
    MAX(CASE WHEN costliest_rank = 1 THEN costliest_price END) AS
costliest_state_price
FROM ranked_extremes
GROUP BY commodity
ORDER BY commodity;
```

```
0: jdbc:hive2://localhost:10000> WITH price_extremes AS ( SELECT commodity, state, MIN(price) AS min_price, MAX(price) AS max_price FROM food_prices WHERE year = 2024 GROUP BY commodity, state ), ranked_e
xtremes AS ( SELECT commodity, state AS cheapest_state, min_price AS cheapest_price, NULL AS costliest_state, NULL AS costliest_price, RANK() OVER (PARTITION BY commodity ORDER BY min_price ASC) AS cheape
st_rank, RANK() OVER (PARTITION BY commodity ORDER BY max_price DESC) AS costliest_rank FROM price_extremes UNION ALL SELECT commodity, NULL AS cheapest_state, NULL AS cheapest_price, state AS costliest_s
tate, max_price AS costliest_price, RANK() OVER (PARTITION BY commodity ORDER BY min_price ASC) AS cheapest_rank, RANK() OVER (PARTITION BY commodity ORDER BY max_price DESC) AS costliest_rank FROM price_
extremes ) SELECT commodity, MAX(CASE WHEN cheapest_rank = 1 THEN cheapest_state END) AS cheapest_state, MAX(CASE WHEN cheapest_rank = 1 THEN cheapest_price END) AS cheapest_state_price, MAX(CASE WHEN cos
tliest_rank = 1 THEN costliest_state END) AS costliest_state, MAX(CASE WHEN costliest_rank = 1 THEN costliest_price END) AS costliest_state_price FROM ranked_extremes GROUP BY commodity ORDER BY commodity
;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+--------------------+----------------+----------------------+-----------------+------------------------+
|      commodity     | cheapest_state | cheapest_state_price | costliest_state | costliest_state_price  |
+--------------------+----------------+----------------------+-----------------+------------------------+
| Ghee (vanaspati)   | Orissa         | 116.72               | Assam           | 143.88                 |
| Lentils            | Assam          | 141.18               | Karnataka       | 165.57                 |
| Lentils (masur)    | Orissa         | 86.61                | Karnataka       | 101.14                 |
| Lentils (moong)    | Rajasthan      | 112.64               | Karnataka       | 124.26                 |
| Lentils (urad)     | Maharashtra    | 116.7                | Karnataka       | 140.15                 |
| Milk (pasteurized) | Karnataka      | 51.8                 | Assam           | 71.83                  |
| Oil (groundnut)    | Orissa         | 185.04               | Rajasthan       | 192.87                 |
| Oil (mustard)      | Orissa         | 133.0                | Assam           | 149.64                 |
| Oil (palm)         | Karnataka      | 93.78                | Assam           | 118.25                 |
| Oil (soybean)      | Maharashtra    | 118.11               | Karnataka       | 137.5                  |
| Oil (sunflower)    | Karnataka      | 117.42               | Assam           | 163.38                 |
| Onions             | Maharashtra    | 34.21                | Assam           | 51.54                  |
| Potatoes           | Rajasthan      | 16.27                | Karnataka       | 34.16                  |
| Rice               | Rajasthan      | 38.32                | Karnataka       | 57.15                  |
| Salt (iodised)     | Assam          | 18.05                | Rajasthan       | 26.89                  |
| Sugar              | Maharashtra    | 42.78                | Assam           | 48.9                   |
| Sugar (jaggery/gur)| Rajasthan      | 48.1                 | Assam           | 71.58                  |
| Tea (black)        | Rajasthan      | 252.46               | Assam           | 323.2                  |
| Tomatoes           | Maharashtra    | 27.41                | Orissa          | 31.03                  |
| Wheat              | Rajasthan      | 27.04                | Karnataka       | 44.19                  |
| Wheat flour        | Rajasthan      | 33.28                | Karnataka       | 49.74                  |
+--------------------+----------------+----------------------+-----------------+------------------------+
21 rows selected (13.208 seconds)
0: jdbc:hive2://localhost:10000>
```

Relevance

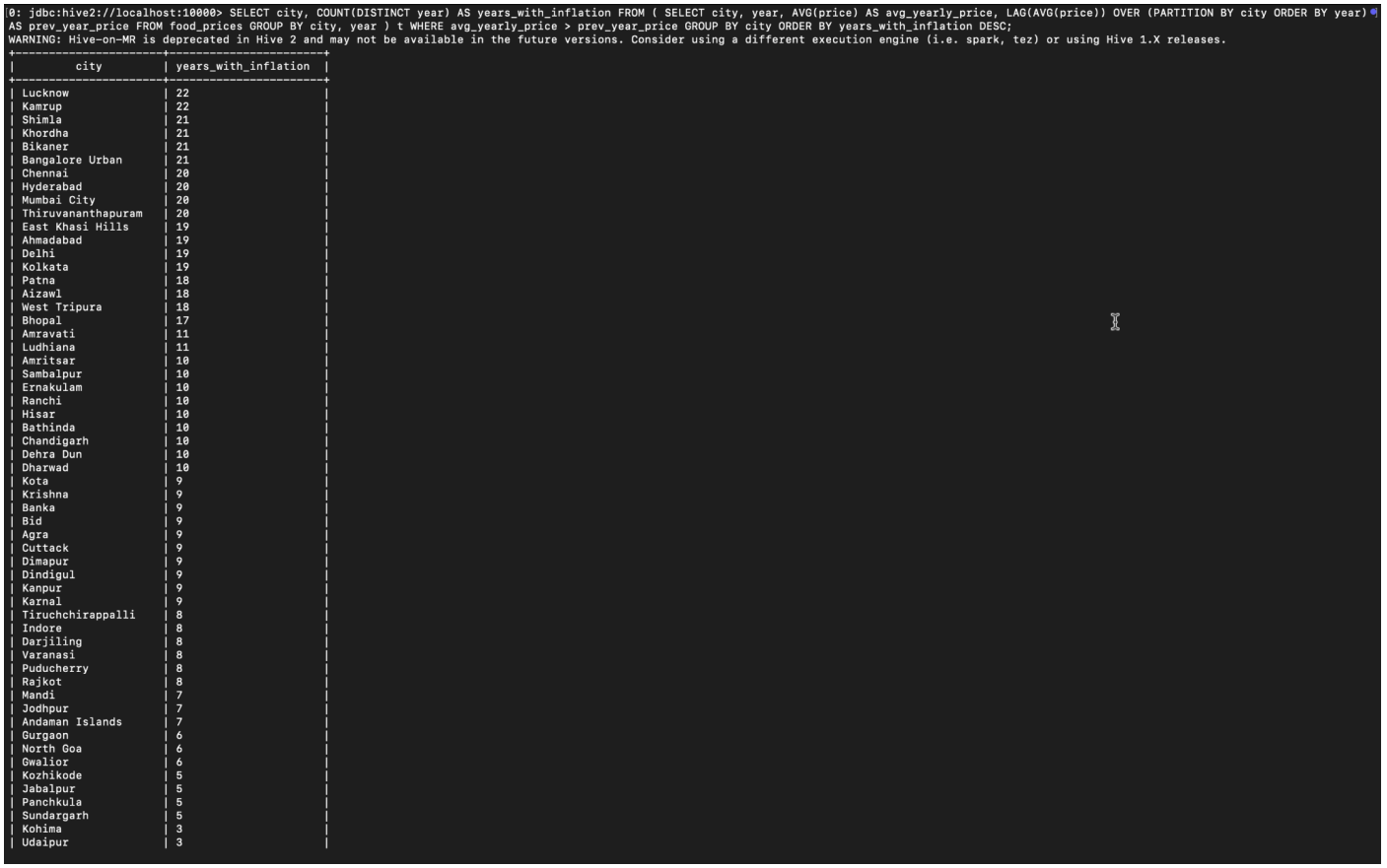
Comparing the cheapest and costliest states for each commodity reveals the extremes of India's regional price disparities. This stark contrast highlights opportunities for improving inter-state trade and distribution networks, while also helping consumers understand which foods might be worth purchasing from neighboring states when possible.

Visualization



Q14) Which cities experienced consistent food price inflation over the years?

```
SELECT
    city,
    COUNT(DISTINCT year) AS years_with_inflation
FROM (
    SELECT
        city,
        year,
        AVG(price) AS avg_yearly_price,
        LAG(AVG(price)) OVER (PARTITION BY city ORDER BY year) AS prev_year_price
    FROM food_prices
    GROUP BY city, year
) t
WHERE t.avg_yearly_price > t.prev_year_price
GROUP BY city
ORDER BY years_with_inflation DESC;
```



```
0: jdbc:hive2://localhost:10000> SELECT city, COUNT(DISTINCT year) AS years_with_inflation FROM ( SELECT city, year, AVG(price) AS avg_yearly_price, LAG(AVG(price)) OVER (PARTITION BY city ORDER BY year)
AS prev_year_price FROM food_prices GROUP BY city, year ) t WHERE avg_yearly_price > prev_year_price GROUP BY city ORDER BY years_with_inflation DESC;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+----------------------+-----------------------+
|         city         | years_with_inflation  |
+----------------------+-----------------------+
| Lucknow              | 22                    |
| Kamrup               | 22                    |
| Shimla               | 21                    |
| Khordha              | 21                    |
| Bikaner              | 21                    |
| Bangalore Urban      | 21                    |
| Chennai              | 20                    |
| Hyderabad            | 20                    |
| Mumbai City          | 20                    |
| Thiruvananthapuram   | 20                    |
| East Khasi Hills     | 19                    |
| Ahmadabad            | 19                    |
| Delhi                | 19                    |
| Kolkata              | 19                    |
| Patna                | 18                    |
| Aizawl               | 18                    |
| West Tripura         | 18                    |
| Bhopal               | 17                    |
| Amravati             | 11                    |
| Ludhiana             | 11                    |
| Amritsar             | 10                    |
| Sambalpur            | 10                    |
| Ernakulam            | 10                    |
| Ranchi               | 10                    |
| Hisar                | 10                    |
| Bathinda             | 10                    |
| Chandigarh           | 10                    |
| Dehra Dun            | 10                    |
| Dharwad              | 10                    |
| Kota                 | 9                     |
| Krishna              | 9                     |
| Banka                | 9                     |
| Bid                  | 9                     |
| Agra                 | 9                     |
| Cuttack              | 9                     |
| Dimapur              | 9                     |
| Dindigul             | 9                     |
| Kanpur               | 9                     |
| Karnal               | 9                     |
| Tiruchchirappalli    | 8                     |
| Indore               | 8                     |
| Darjiling            | 8                     |
| Varanasi             | 8                     |
| Puducherry           | 8                     |
| Rajkot               | 8                     |
| Mandi                | 7                     |
| Jodhpur              | 7                     |
| Andaman Islands      | 7                     |
| Gurgaon              | 6                     |
| North Goa            | 6                     |
| Gwalior              | 6                     |
| Kozhikode            | 5                     |
| Jabalpur             | 5                     |
| Panchkula            | 5                     |
| Sundargarh           | 5                     |
| Kohima               | 3                     |
| Udaipur              | 3                     |
```

Relevance

Tracking cities with consistent food price inflation identifies urban areas facing chronic affordability challenges. These locations may have structural issues that continuously drive prices upward, such as rapid population growth, limited agricultural land in surrounding areas, or inefficient distribution systems that need targeted policy interventions.

Q15) Identify the top 10 regions in India where food prices are the most volatile, by calculating the average price variance across nearby markets (within a 100 km radius).

```
WITH markets AS (
  SELECT
    state, city, market,
    AVG(price) AS avg_price,
    MAX(latitude) AS latitude,
    MAX(longitude) AS longitude,
    ROW_NUMBER() OVER (PARTITION BY state, city, market) AS rn
  FROM food_prices
  GROUP BY state, city, market
),
nearby_markets AS (
    SELECT
    m1.state, m1.city, m1.market, m1.avg_price AS price1, m2.market AS nearby_market,
    m2.avg_price AS price2,
    111.045 * DEGREES(ACOS(
      LEAST(1.0, COS(RADIANS(m1.latitude)) * COS(RADIANS(m2.latitude)) *
      COS(RADIANS(m2.longitude) - RADIANS(m1.longitude)) +
      SIN(RADIANS(m1.latitude)) * SIN(RADIANS(m2.latitude)))
    )) AS distance_km
  FROM markets m1 JOIN markets m2 ON m1.market != m2.market
  WHERE
  111.045 * DEGREES(ACOS(
      LEAST(1.0, COS(RADIANS(m1.latitude)) * COS(RADIANS(m2.latitude)) *
      COS(RADIANS(m2.longitude) - RADIANS(m1.longitude)) +
      SIN(RADIANS(m1.latitude)) * SIN(RADIANS(m2.latitude)))
    )) <= 100
)
SELECT
  state, city, market,
  AVG(ABS(price1 - price2)) AS avg_price_volatility,
  COUNT(DISTINCT nearby_market) AS nearby_market_count
FROM nearby_markets
GROUP BY state, city, market
HAVING COUNT(DISTINCT nearby_market) > 2
ORDER BY avg_price_volatility DESC LIMIT 10;
```

```
0: jdbc:hive2://localhost:10000> WITH markets AS ( SELECT state, city, market, AVG(price) AS avg_price, MAX(latitude) AS latitude, MAX(longitude) AS longitude, ROW_NUMBER() OVER (PARTITION BY state, city,
 market) AS rn FROM food_prices GROUP BY state, city, market ), nearby_markets AS ( SELECT m1.state, m1.city, m1.market, m1.avg_price AS price1, m2.market AS nearby_market, m2.avg_price AS price2, 111.045
 * DEGREES(ACOS( LEAST(1.0, COS(RADIANS(m1.latitude)) * COS(RADIANS(m2.latitude)) * COS(RADIANS(m2.longitude) - RADIANS(m1.longitude)) + SIN(RADIANS(m1.latitude)) * SIN(RADIANS(m2.latitude))) )) AS distan
ce_km FROM markets m1 JOIN markets m2 ON m1.market != m2.market WHERE 111.045 * DEGREES(ACOS( LEAST(1.0, COS(RADIANS(m1.latitude)) * COS(RADIANS(m2.latitude)) * COS(RADIANS(m2.longitude) - RADIANS(m1.long
itude)) + SIN(RADIANS(m1.latitude)) * SIN(RADIANS(m2.latitude))) )) <= 100 ) SELECT state, city, market, AVG(ABS(price1 - price2)) AS avg_price_volatility, COUNT(DISTINCT nearby_market) AS nearby_market_c
ount FROM nearby_markets GROUP BY state, city, market HAVING COUNT(DISTINCT nearby_market) > 2 ORDER BY avg_price_volatility DESC LIMIT 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20250323104630_a3427828-823c-4942-b25c-e34678988187.log
2025-03-23 10:46:40     Starting to launch local task to process map join;      maximum memory = 477626368
2025-03-23 10:46:40     Dump the side-table for tag: 1 with group count: 1 into file: file:/tmp/root/135576f2-db96-4567-bd79-7b7c57dfcc70/hive_2025-03-23_10-46-30_811_5025985908235006507-1/-local-10008/Ha
shTable-Stage-6/MapJoin-mapfile121--.hashtable
2025-03-23 10:46:41     Uploaded 1 File to: file:/tmp/root/135576f2-db96-4567-bd79-7b7c57dfcc70/hive_2025-03-23_10-46-30_811_5025985908235006507-1/-local-10008/HashTable-Stage-6/MapJoin-mapfile121--.hasht
able (6736 bytes)
2025-03-23 10:46:41     End of local task; Time Taken: 0.224 sec.
+------------------+------------------+------------+---------------------+----------------------+
|      state       |       city       |   market   | avg_price_volatility | nearby_market_count  |
+------------------+------------------+------------+---------------------+----------------------+
| Karnataka        | Bangalore Urban  | Bengaluru  | 31.77778056046837    | 4                    |
| Assam            | Kamrup           | Guwahati   | 29.682731774791034   | 3                    |
| Meghalaya        | East Khasi Hills | Shillong   | 25.650885913298822   | 3                    |
| Bihar            | Banka            | Bhagalpur  | 23.12405612351834    | 4                    |
| Jharkhand        | Ranchi           | Ranchi     | 20.426011016500635   | 3                    |
| Kerala           | Thrissur         | Thrissur   | 20.08355450667978    | 4                    |
| Himachal Pradesh | Solan            | Solan      | 19.22529582612383    | 5                    |
| Uttar Pradesh    | Saharanpur       | Saharanpur | 19.210837360623998   | 3                    |
| Uttar Pradesh    | Agra             | Agra       | 19.119423700261184   | 3                    |
| Jharkhand        | Gumla            | Gumla      | 16.196884757246558   | 4                    |
+------------------+------------------+------------+---------------------+----------------------+
10 rows selected (14.258 seconds)
0: jdbc:hive2://localhost:10000>
```

Relevance

We wanted to approach this problem using geospatial analysis to uncover localized price inconsistencies that wouldn't be visible through standard market analysis. Hive doesn't natively support geospatial functions, but you can use the Haversine formula to calculate the distance between two latitude-longitude pairs.

According to the formula, distance between two geographical points is:

$$d = 2r \cdot arcsin\left(\sqrt{sin^2\left(\frac{\Phi 2 - \Phi 1}{2}\right) + cos(\Phi 1) \cdot cos(\Phi 1) \cdot sin^2\left(\frac{\lambda 2 - \lambda 1}{2}\right)}\right)$$

Which in SQL converts to:

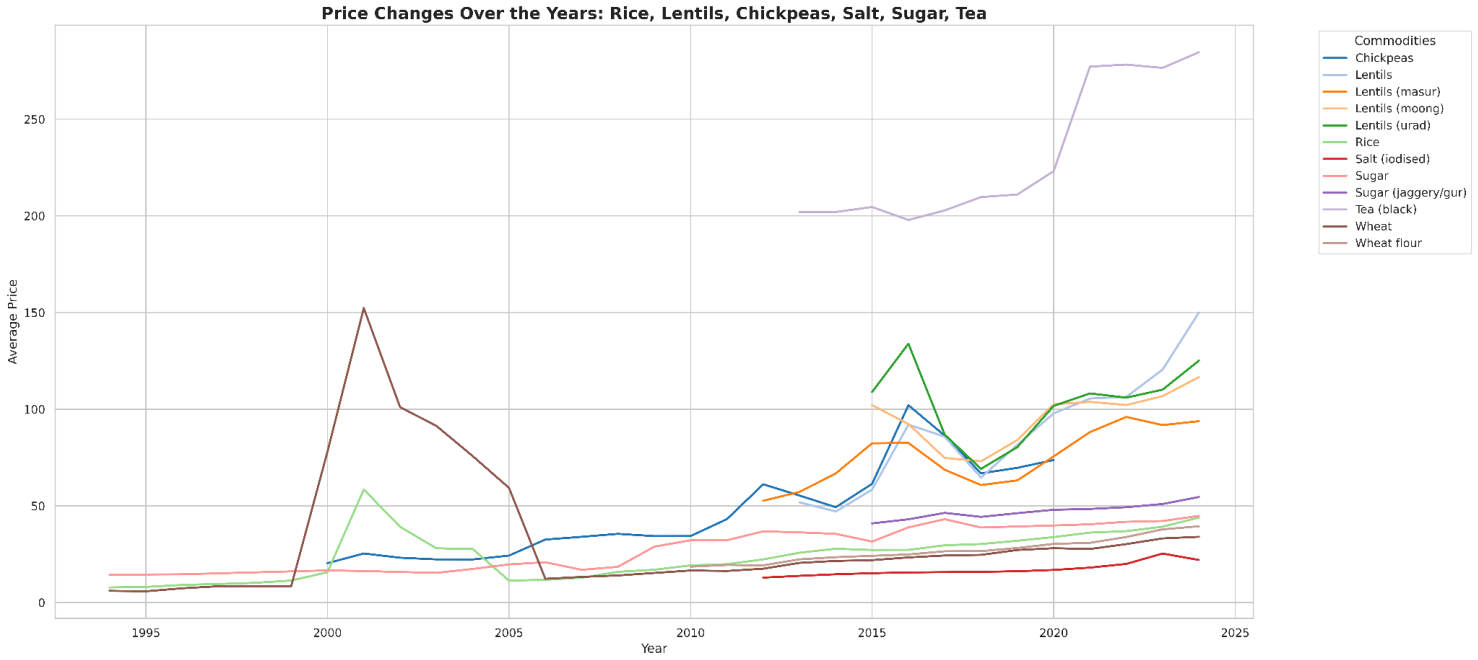
```
111.045 * DEGREES(ACOS(LEAST(1.0,COS(RADIANS(lat1)) * COS(RADIANS(lat2)) *
            COS(RADIANS(long1) - RADIANS(long2)) +
            SIN(RADIANS(lat1)) * SIN(RADIANS(lat2)))))
```

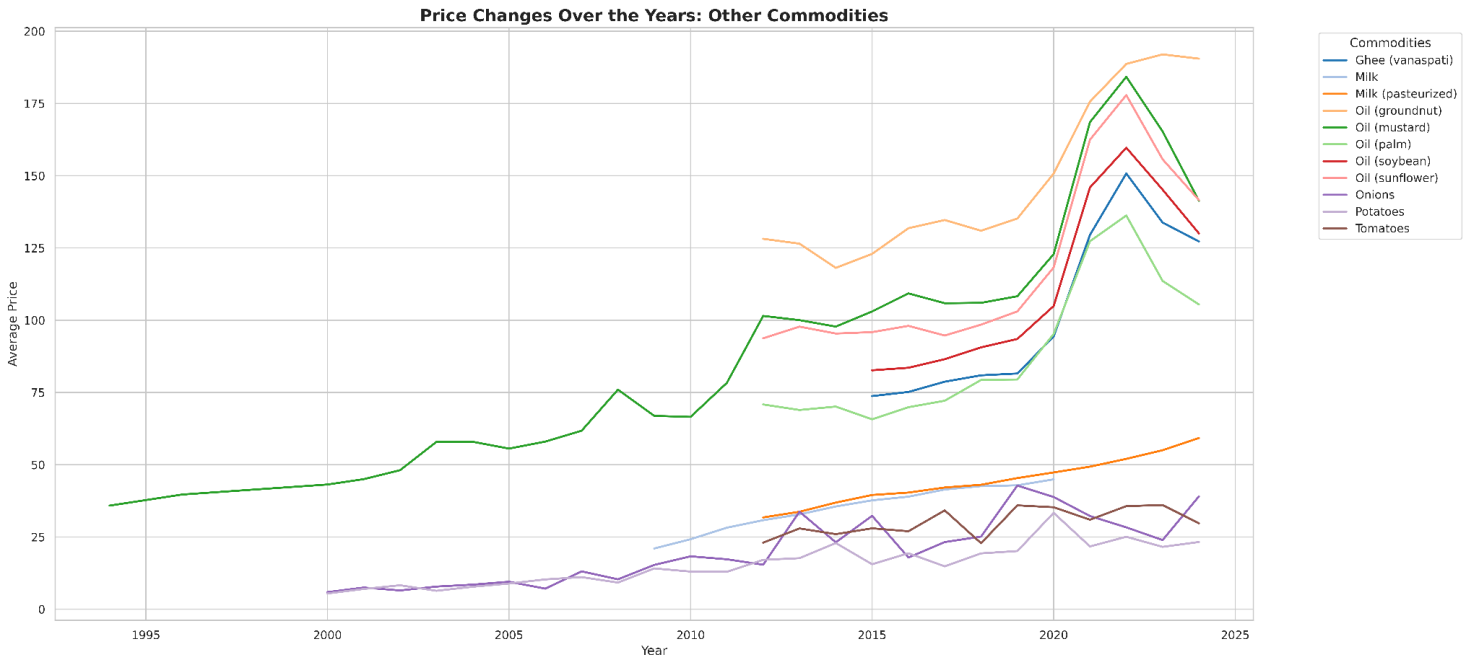
where 111.045 is a conversion factor representing approximately how many kilometers one degree of latitude equals at the equator. Reference - Fast SQL location finder (geolocation) using Haversines

Mapping regions with the highest food price volatility within a 100km radius reveals local market integration problems. Areas where nearby markets show wildly different prices likely suffer from poor transportation infrastructure, information gaps between markets, or fragmented supply chains. This spatial analysis helps prioritize logistics improvements that could stabilize regional food access.

Conclusion

The analysis of Indian food prices reveals several key insights into market dynamics and regional variations. Monsoon emerges as the most expensive season. Notably, commodities such as tea, wheat, and ghee exhibit high volatility across different markets, with a *standard deviation* exceeding **45**, marking them as products with substantial price increases over the years. Tea and oil consistently rank as the most expensive products across nearly all states.





Bihar stands out with the most significant price difference between high and low-valued commodities, highlighting regional disparities in pricing. Interestingly, average prices across

seasons show that metropolitan areas tend to be cheaper compared to the rest of India, suggesting more competitive pricing in urban centers. Karnataka and Maharashtra consistently appear in the top five for the highest average price of each commodity, indicating their strong market presence.

Cities like Lucknow, Kamrup, Shimla, Khordha, Bikaner, Bengaluru, Chennai, and Mumbai City have experienced constant inflation over the past **22** years, reflecting ongoing economic pressures. Additionally, Bengaluru, Guwahati, and Shillong are particularly prone to price volatility, with significant price variance observed across nearby markets within a *100KM* range. These findings provide valuable insights for stakeholders aiming to navigate the complexities of the Indian food market.

References

[1] Configuring a MySQL Database for the Hive Metastore – https://docs.ezmeral.hpe.com/datafabric-customer-managed/79/Hive/Config-RemoteMySQLForHiveMetastore.html

[2] Importing data to partitioned and clustered Hive tables – https://www.ibm.com/docs/en/psfa/1.7.2?topic=movement-importing-data-partitioned-clustered-hive-tables

[3] Bubble Charts – https://plotly.com/python/bubble-charts/

[4] Common Table Expression (CTE) – https://hive.apache.org/docs/latest/common-table-expression_38572242/

[5] Hive Window Functions – https://www.linkedin.com/pulse/hive-lag-rank-over-explained-example-fu-an-yang/

[6] Fast nearest-location finder for SQL – https://www.plumislandmedia.net/mysql/haversine-mysql-nearest-loc/