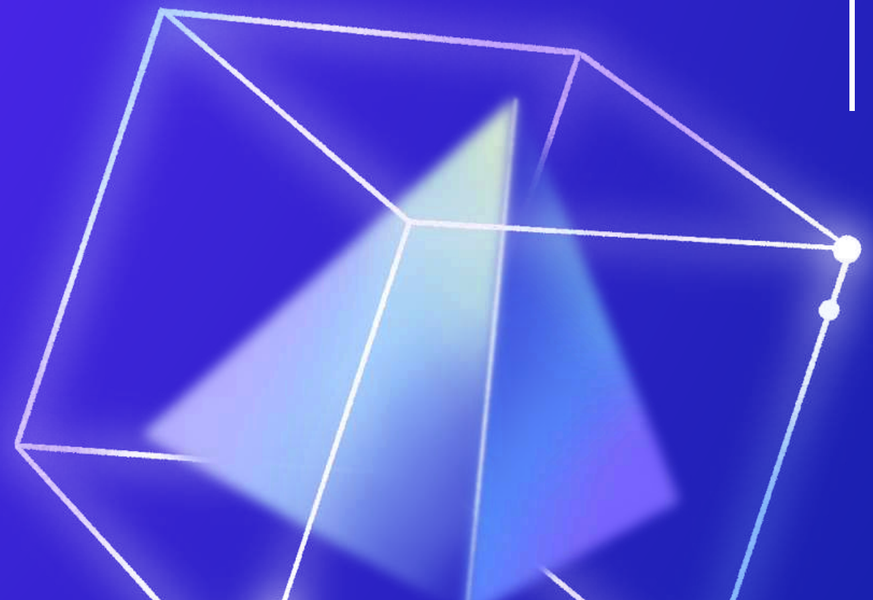
TIME SERIES FORECASTING

MARKET PRICE PREDICTION

By Prashant Pal
Machine Learning Intern- Mentorness

TABLE OF CONTENTS



INTRODUCTION

- Accurately predicting commodity prices is crucial today. Our project aims to develop precise machine learning models for forecasting market trends.

- Machine learning has revolutionized market analysis. Our dataset offers rich historical market data, providing insights into commodity quantities and prices across regions.

- Our goal is to build robust time series forecasting models. Leveraging advanced algorithms, we empower stakeholders with actionable insights for decision-making.

- The dataset includes features like quantity, prices, states, cities, and dates. Through preprocessing and feature engineering, we aim to extract valuable insights for model training.

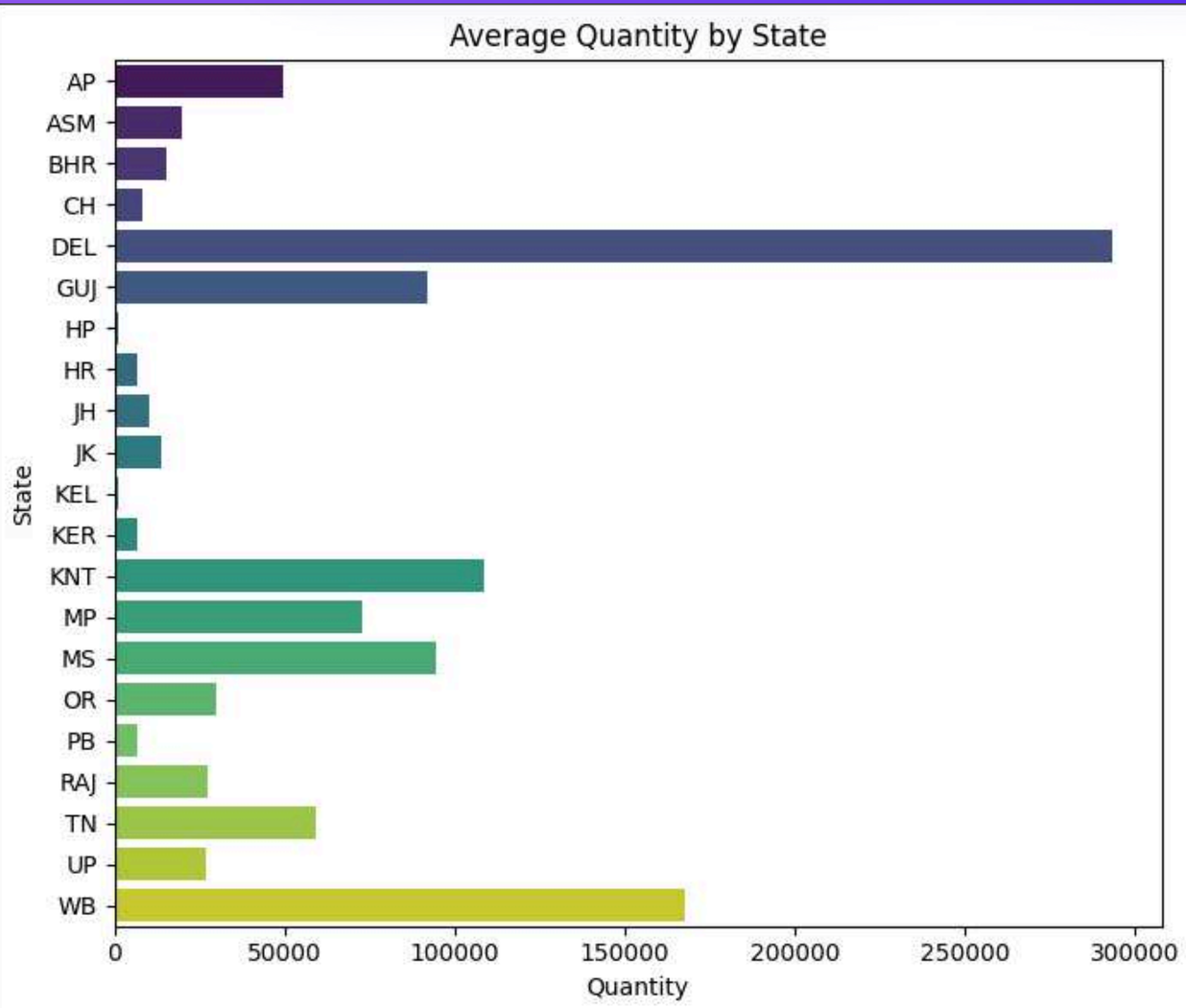
- Our project involves crucial tasks: data preprocessing, exploratory analysis, feature engineering, model selection, training, and validation. We strive to build reliable forecasting models through these steps.

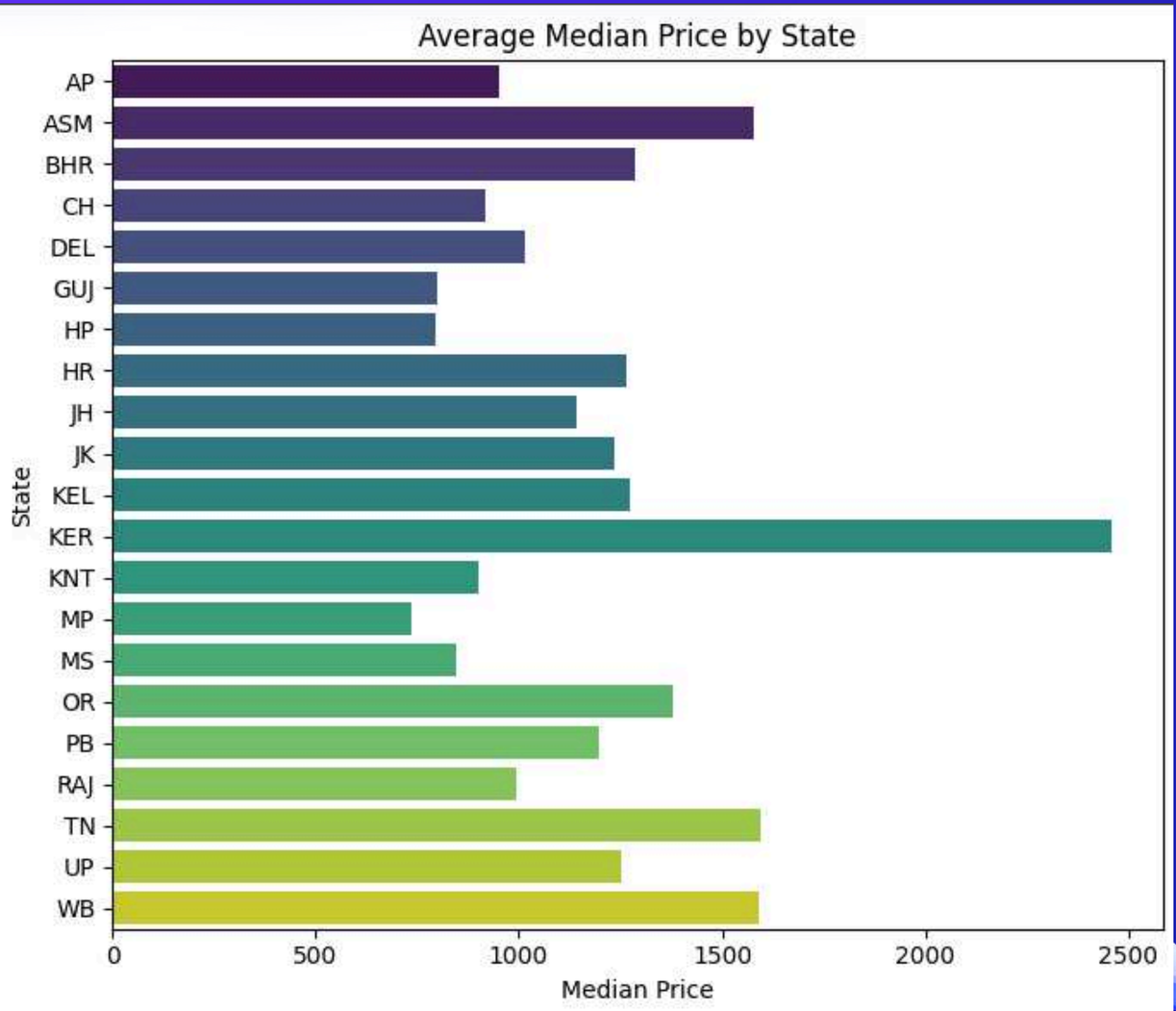
DATA PREPROCESSING

- Data from Marketpriceprediction.CSV file was loaded into DataFrame df.
- No missing values were observed, hence no imputation was required.
- Categorical variables were retained as-is for EDA and later dropped during model training.
- Date column was converted to datetime format, and year and month were extracted.
- Duplicate dates were removed by aggregating data to forecast on a monthly basis.
- Column names were lowercased for ease of coding.
- Calculated Median price by taking median of all prices for forecasting.

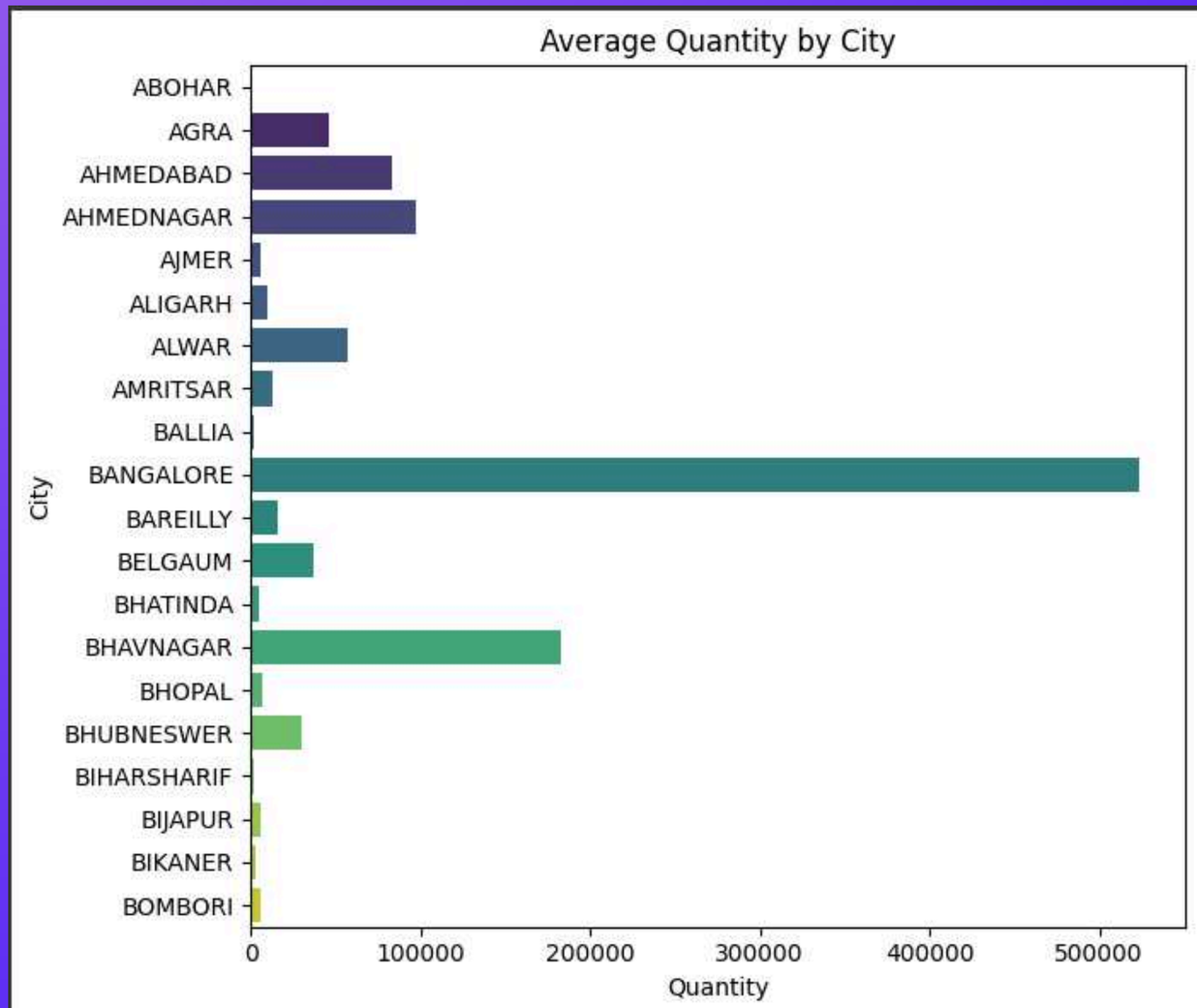
EDA

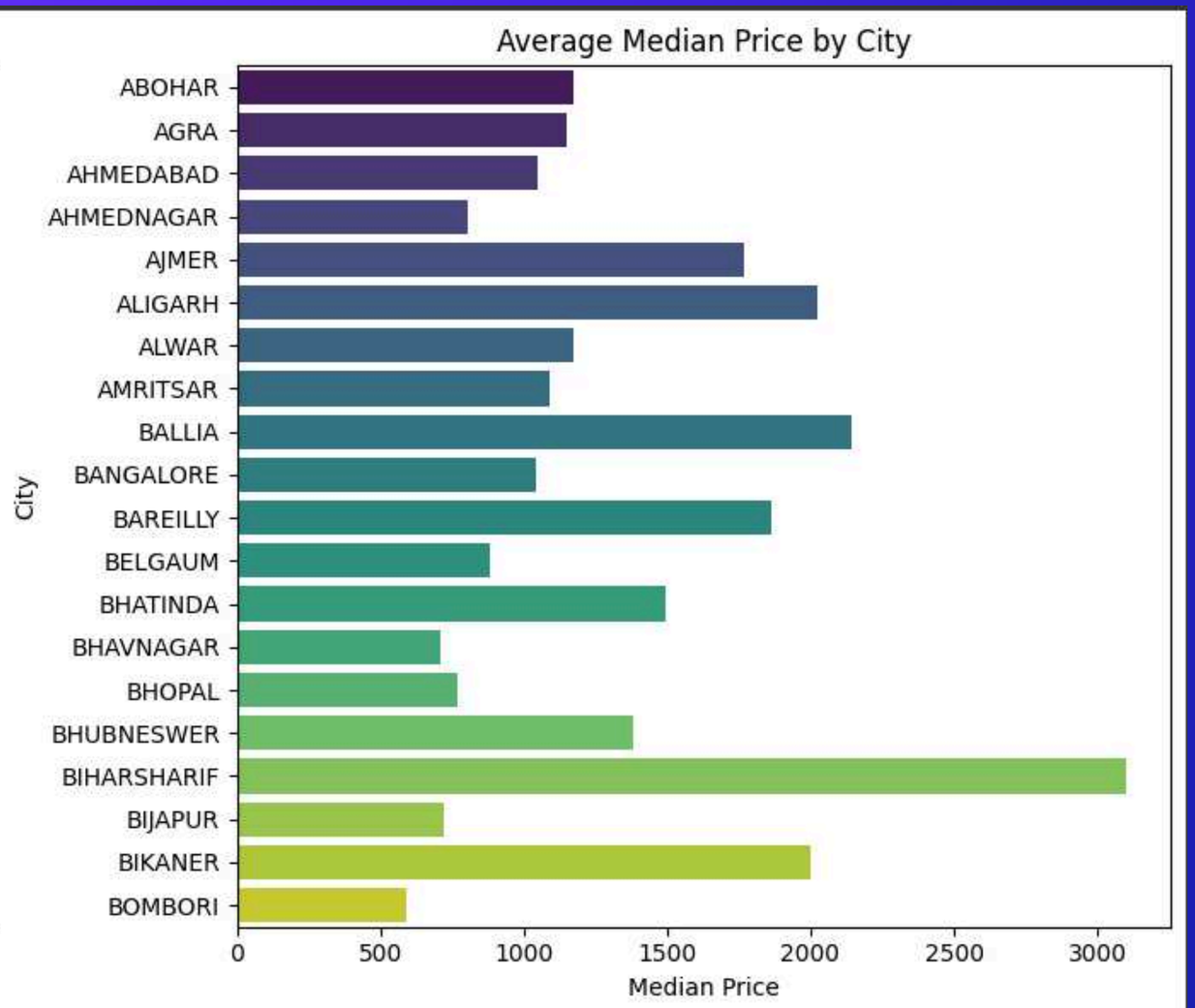
AVERAGE QUANTITY & MEDIAN PRICE BY STATE



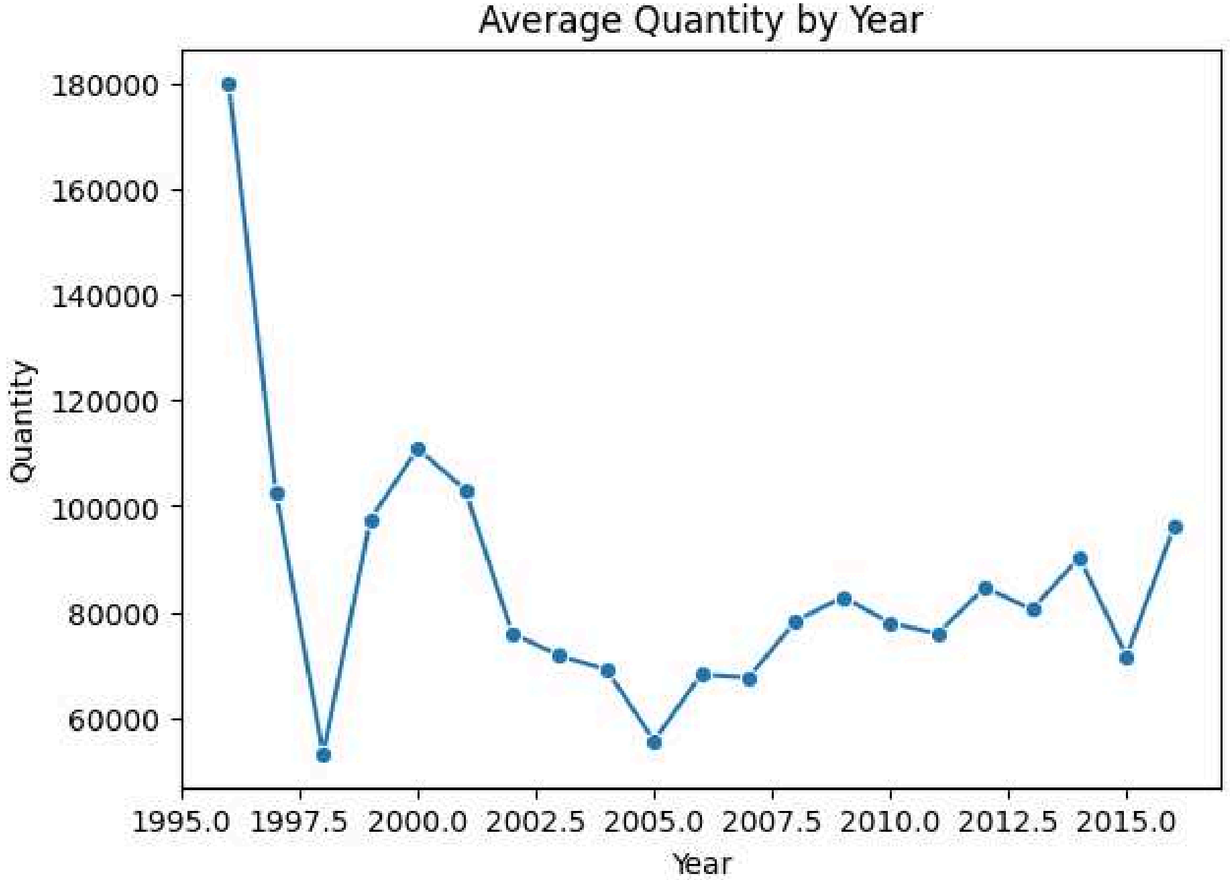


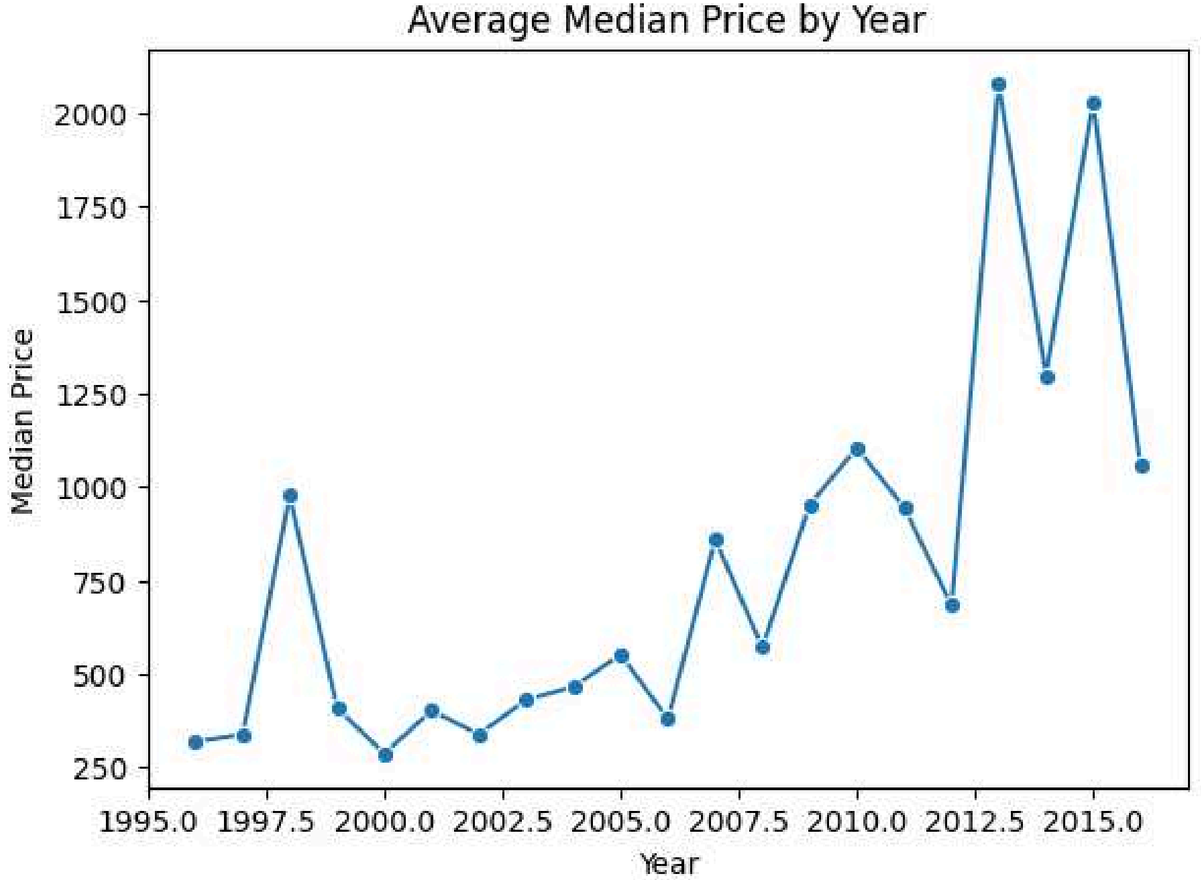
AVERAGE QUANTITY & MEDIAN PRICE BY CITY



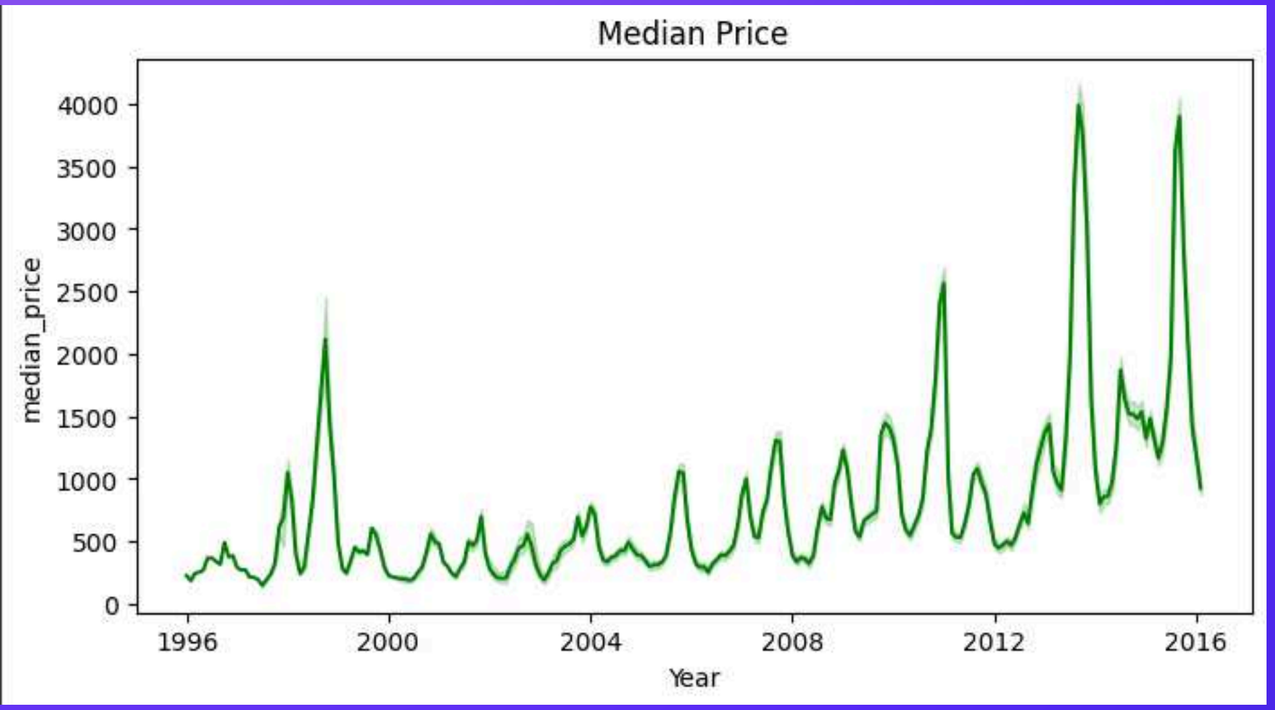


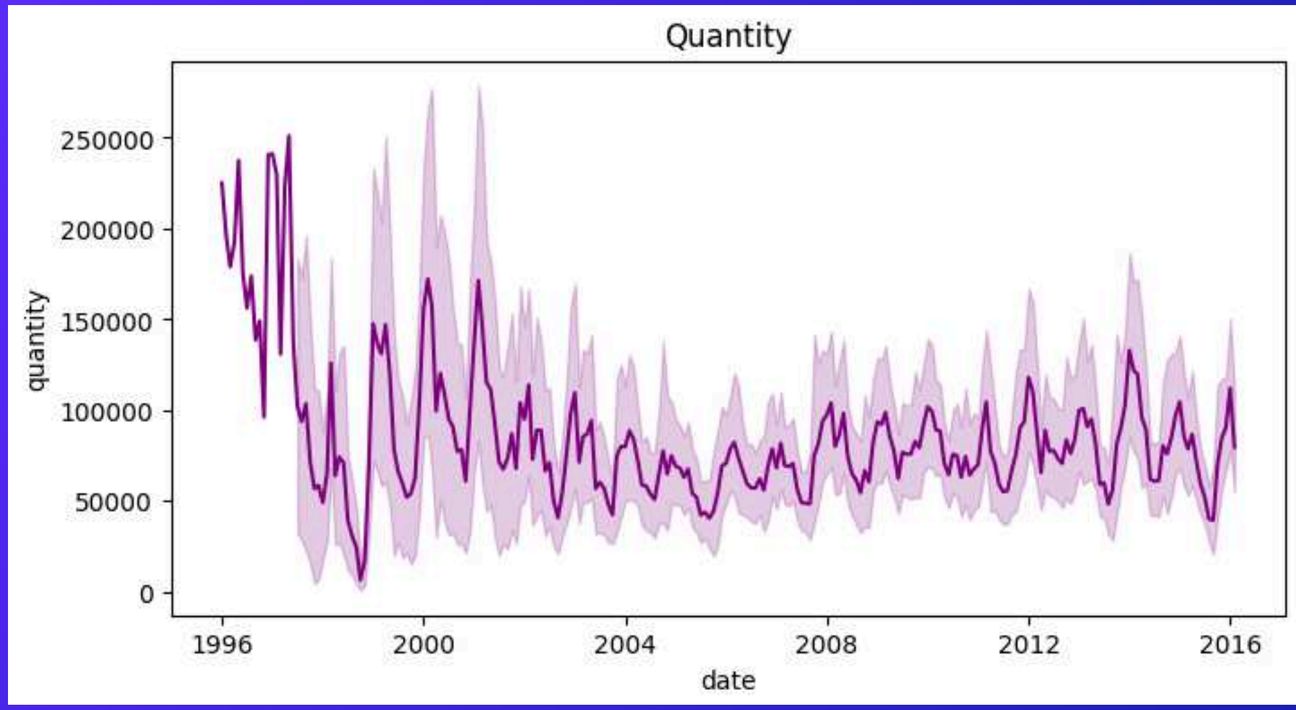
EXPLORATORY DATA ANALYSIS



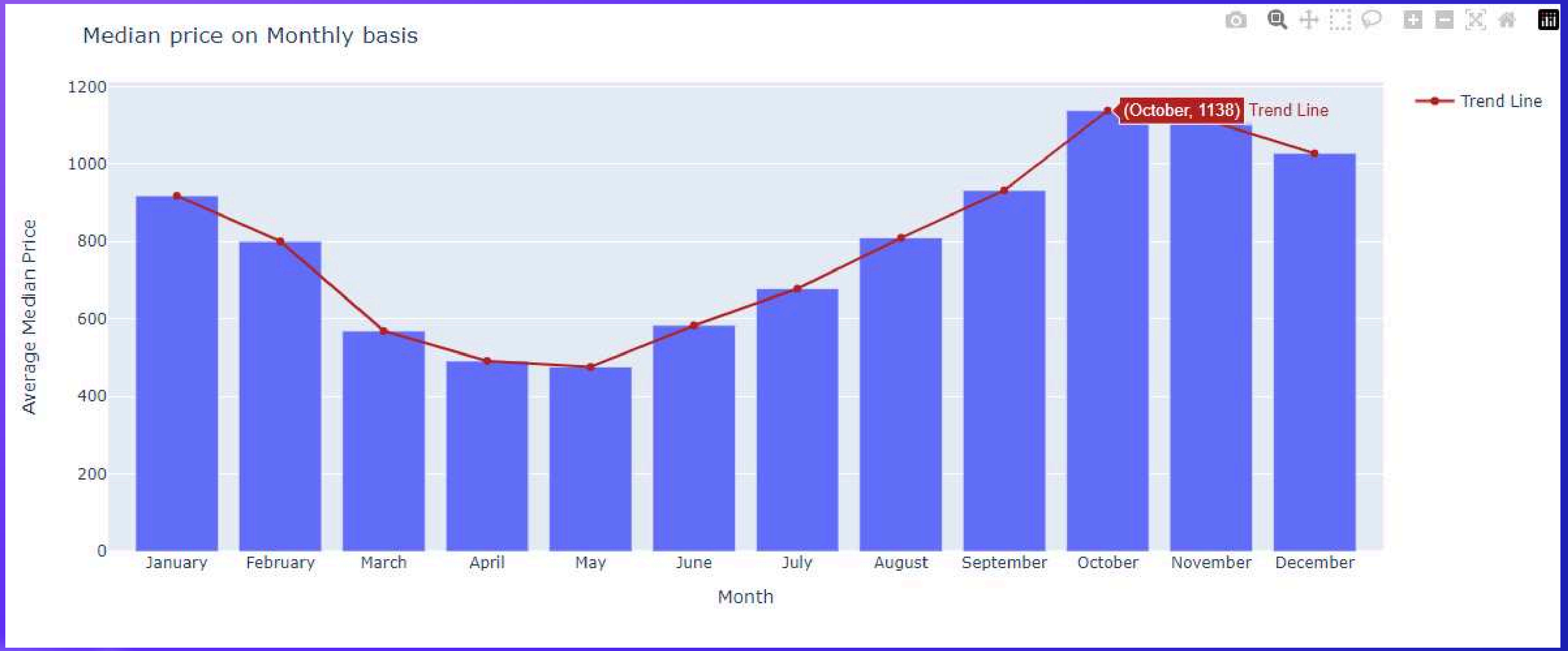


LINEPLOT FOR MEDIAN PRICE AND QUANTITY

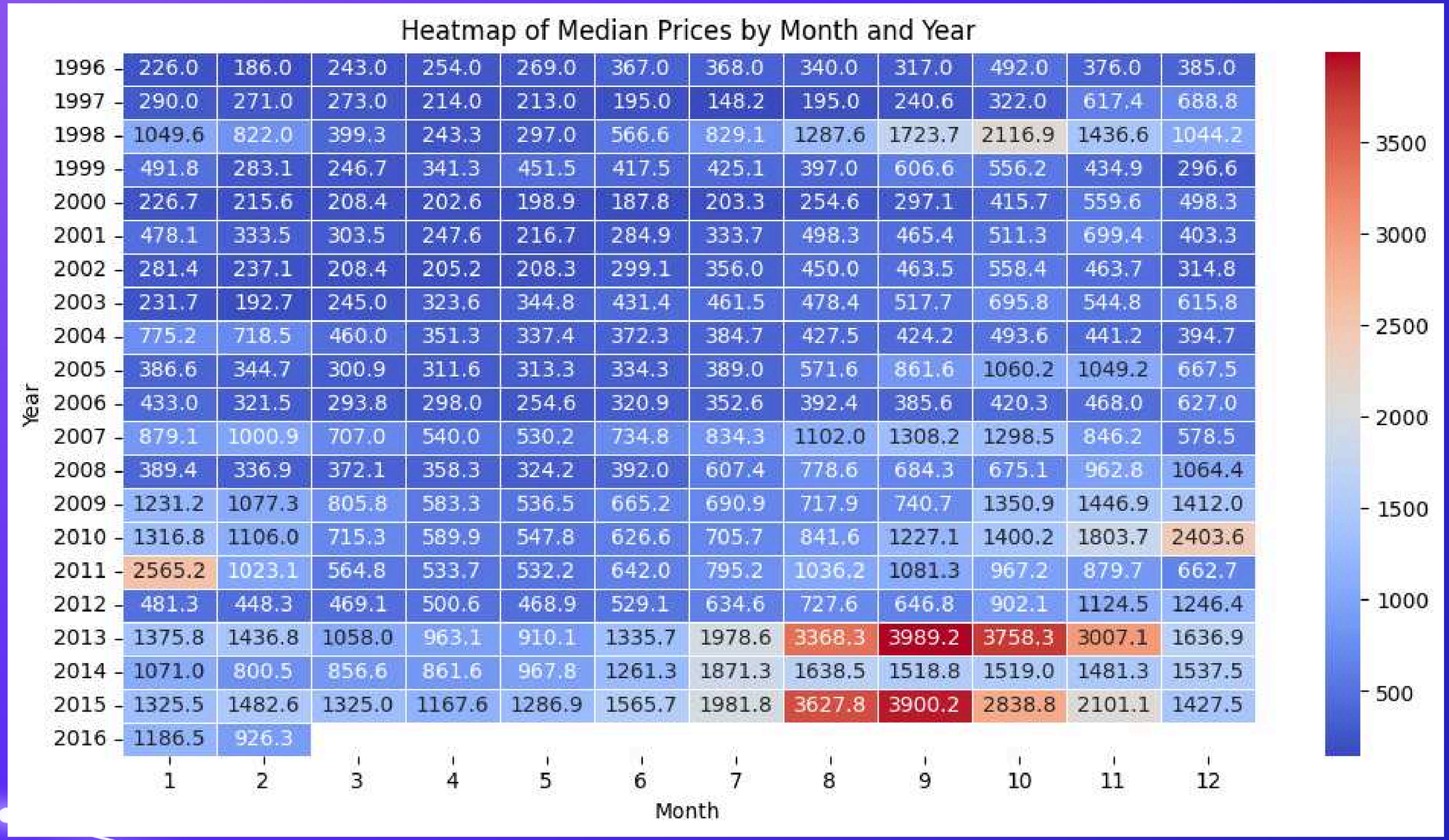




THE SEASONAL (MONTHLY) TRENDS



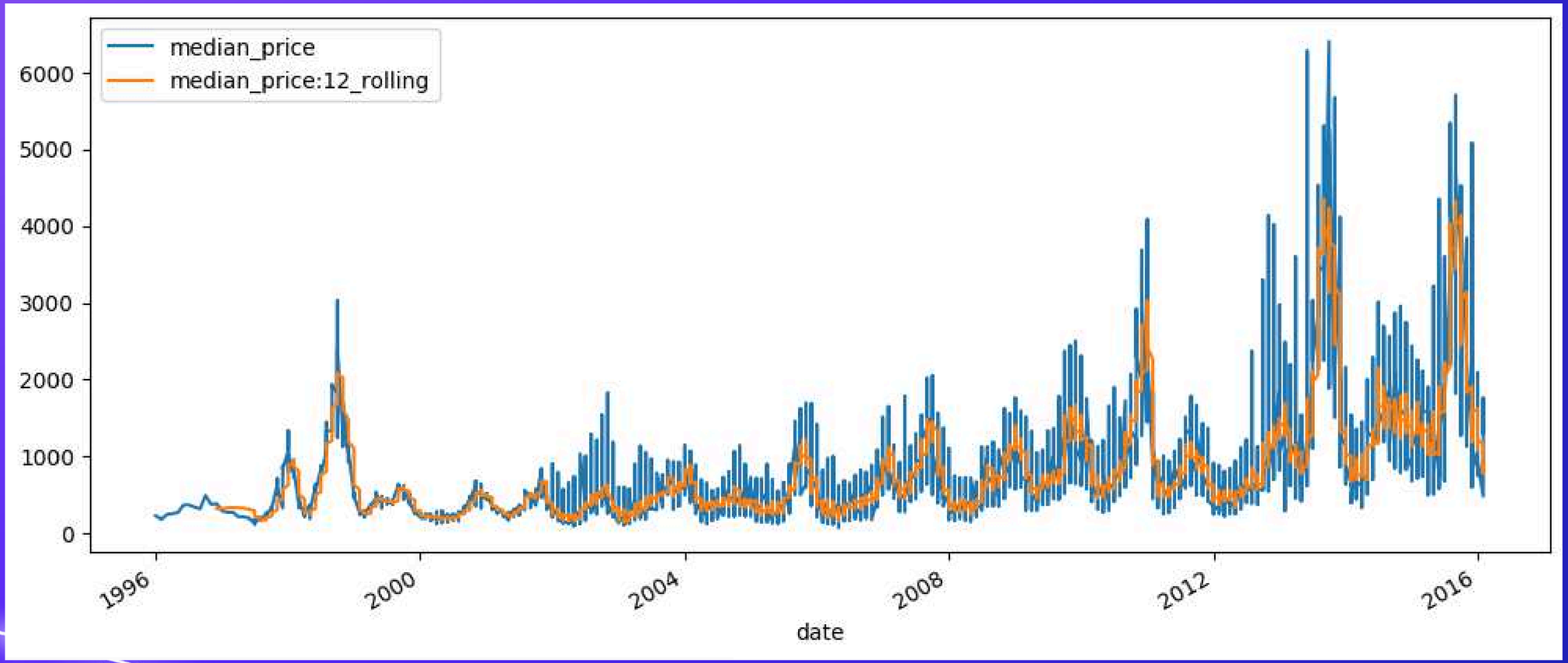
HEATMAP OF MEDIAN PRICES BY MONTH AND YEAR



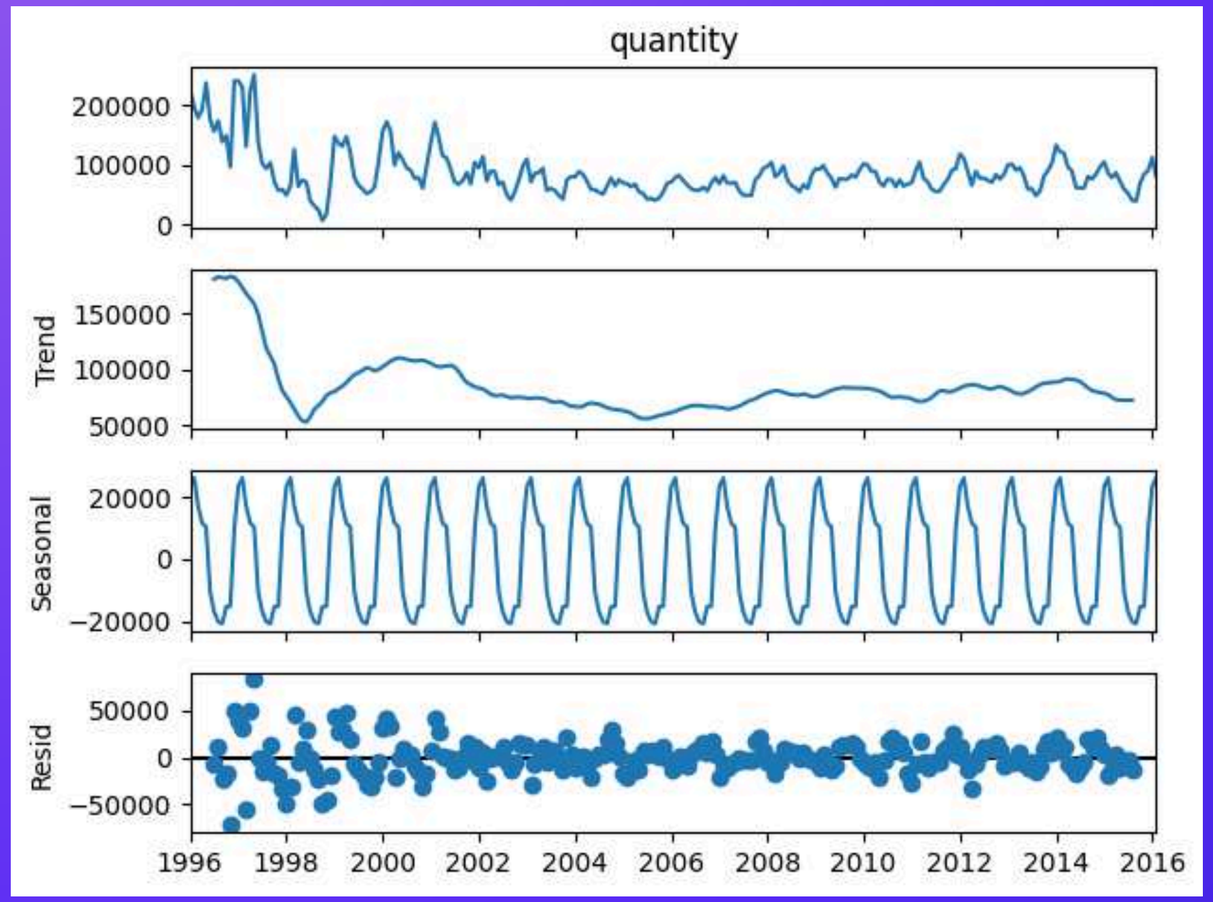
FEATURE ENGINEERING

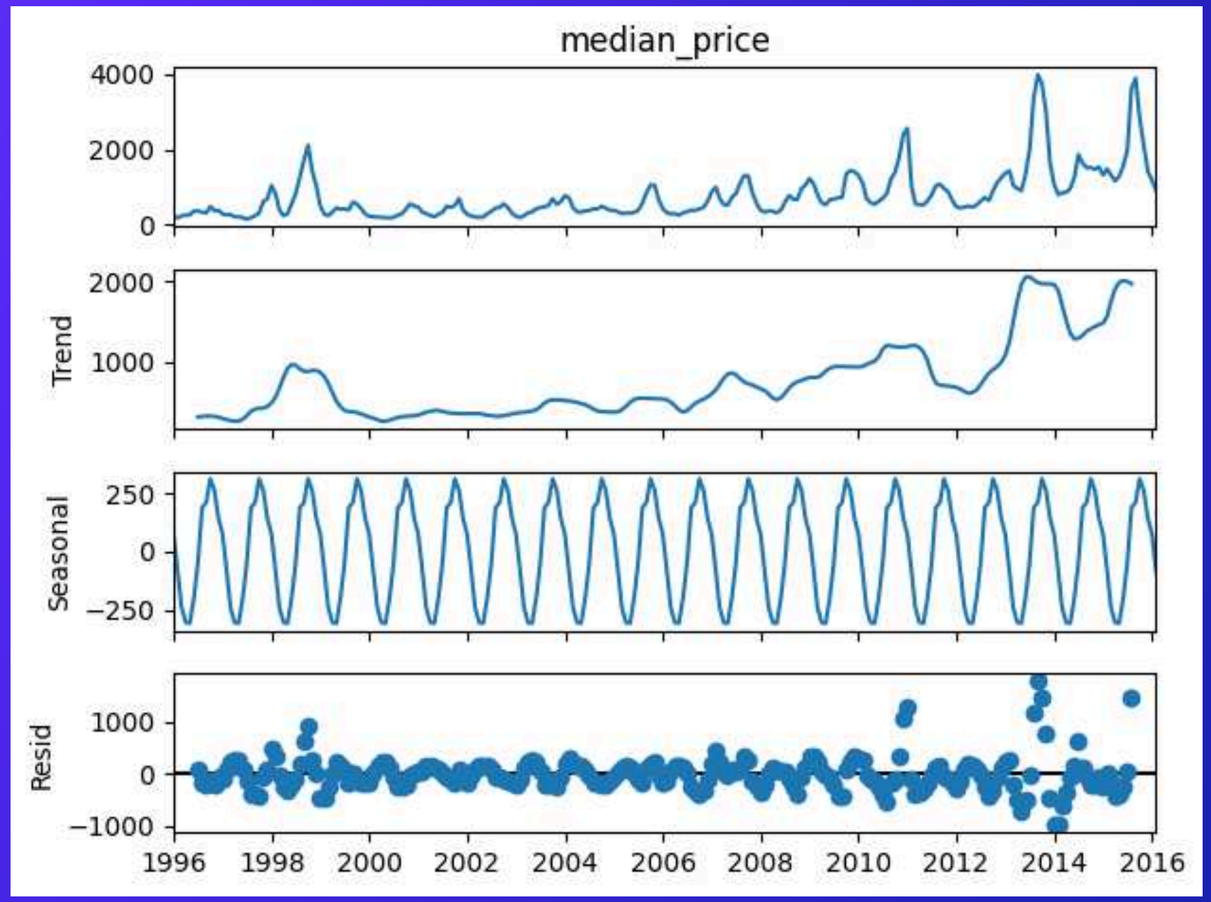
7. Rolling Statistics

```
[931] #Rolling Statistics
      df['median_price:12_rolling']=df['median_price'].rolling(12).mean()
```



SEASONAL DECOMPOSE FOR QUANTITY & MEDIAN PRICE





MODELS USED FOR FORECASTING

01

ARIMA

(Autoregressive Integrated Moving Average)

02

SARIMA

(Seasonal Autoregressive Integrated Moving Average)

03

PROPHET

04

LSTM

(Long Short-Term Memory)

EVALUATION OF MODEL FOR QUANTITY FORECASTING

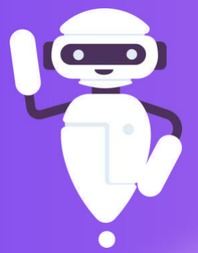
```
# Create a dictionary to store the error metrics for quantity
quantity_metrics = {
    'Model': ['ARIMA', 'SARIMAX', 'PROPHET', 'LSTM'],
    'MAE': [ARIMA_Qty_mae, SARIMAX_Qty_mae, Prophet_Qty_mae, LSTM_Qty_mae],
    'MSE': [ARIMA_Qty_mse, SARIMAX_Qty_mse, Prophet_Qty_mse, LSTM_Qty_mse],
    'RMSE': [ARIMA_Qty_rmse, SARIMAX_Qty_rmse, Prophet_Qty_rmse, LSTM_Qty_rmse]
}

# Create a DataFrame for quantity metrics
quantity_df = pd.DataFrame(quantity_metrics)

quantity_df=quantity_df.set_index('Model')

print("Evaluation of Quantity Metrics:")
print(quantity_df)
```

```
Evaluation of Quantity Metrics:
                MAE           MSE           RMSE
Model
ARIMA     18077.992139  5.159257e+08  22713.997036
SARIMAX   11753.369780  2.033790e+08  14261.101350
PROPHET   11432.407343  1.991221e+08  14111.062194
LSTM      76780.395480  6.179510e+09  78609.860969
```

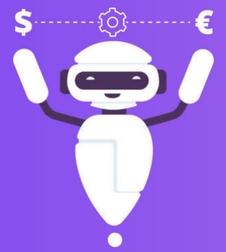


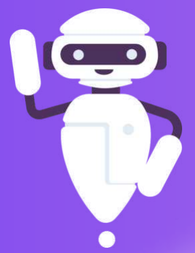
EVALUATION OF MODEL FOR QUANTITY FORECASTING

```python
# Create a dictionary to store the error metrics for price
price_metrics = {
    'Model': ['ARIMA', 'SARIMAX', 'PROPHET', 'LSTM'],
    'MAE': [ARIMA_price_mae, SARIMAX_price_mae, Prophet_price_mae, LSTM_price_mae],
    'MSE': [ARIMA_price_mse, SARIMAX_price_mse, Prophet_price_mse, LSTM_price_mse],
    'RMSE': [ARIMA_price_rmse, SARIMAX_price_rmse, Prophet_price_rmse, LSTM_price_rmse]
}

# Create a DataFrame for price metrics
price_df = pd.DataFrame(price_metrics)
price_df=price_df.set_index('Model')
print("\nEvaluation of Price Metrics:")
print(price_df)
```

```
Evaluation of Price Metrics:
                MAE           MSE         RMSE
Model
ARIMA     899.862680  1.582562e+06  1257.999148
SARIMAX   890.316787  1.555606e+06  1247.239280
PROPHET   508.817160  6.354909e+05   797.176847
LSTM     1175.788680  1.963411e+06  1401.217835
```





CONCLUSION

01

For quantity forecasting, Prophet and SARIMAX models outshine ARIMA and LSTM, with Prophet demonstrating superior accuracy. However, LSTM, while visually accurate, exhibits higher error metrics, indicating potential limitations in capturing underlying patterns.

02

In price forecasting, Prophet emerges as the top-performing model, followed closely by ARIMA and SARIMAX. LSTM, similar to its performance in quantity forecasting, shows higher error metrics, suggesting limitations in predicting commodity prices accurately.

Based on the Evaluation, Prophet stands out as the most suitable model for both quantity and price forecasting, offering the most accurate predictions.

THANK YOU!