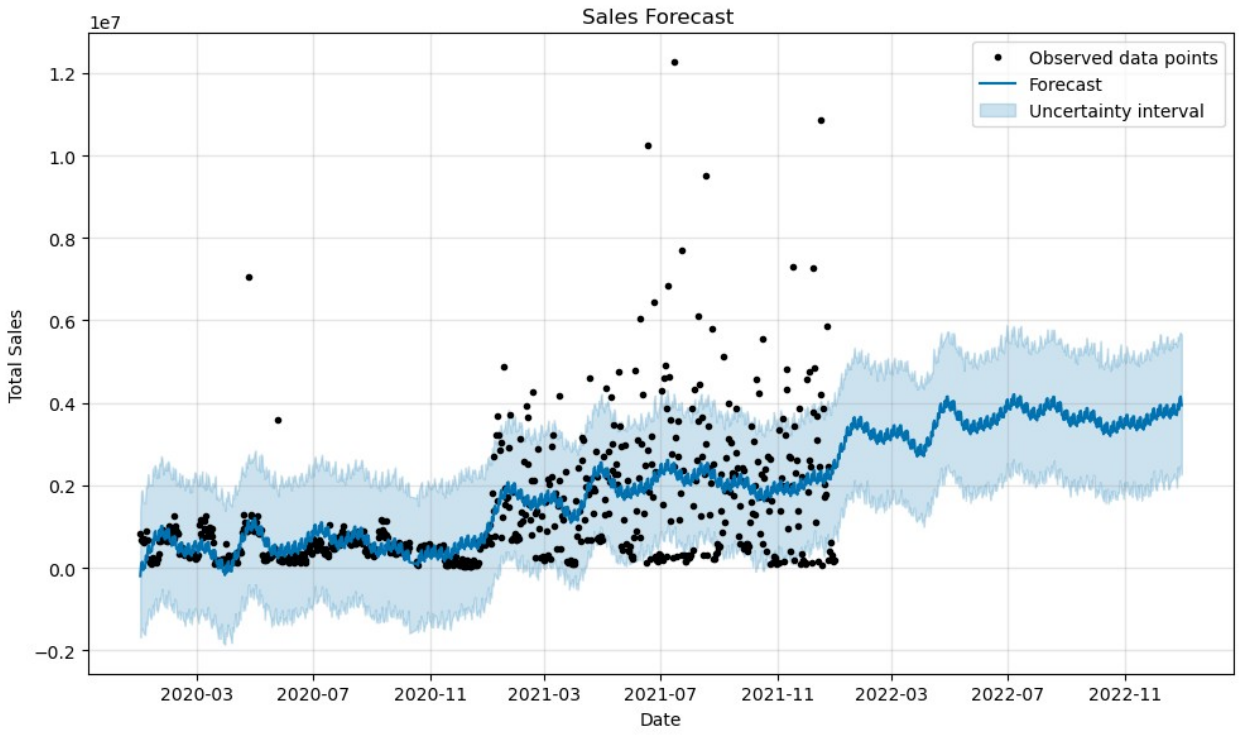
```
import pandas as pd
df=pd.read_excel(r"C:\Users\Dell i5\OneDrive - Cape Peninsula
University of Technology\Desktop\sales.xlsx")
df.head()
```

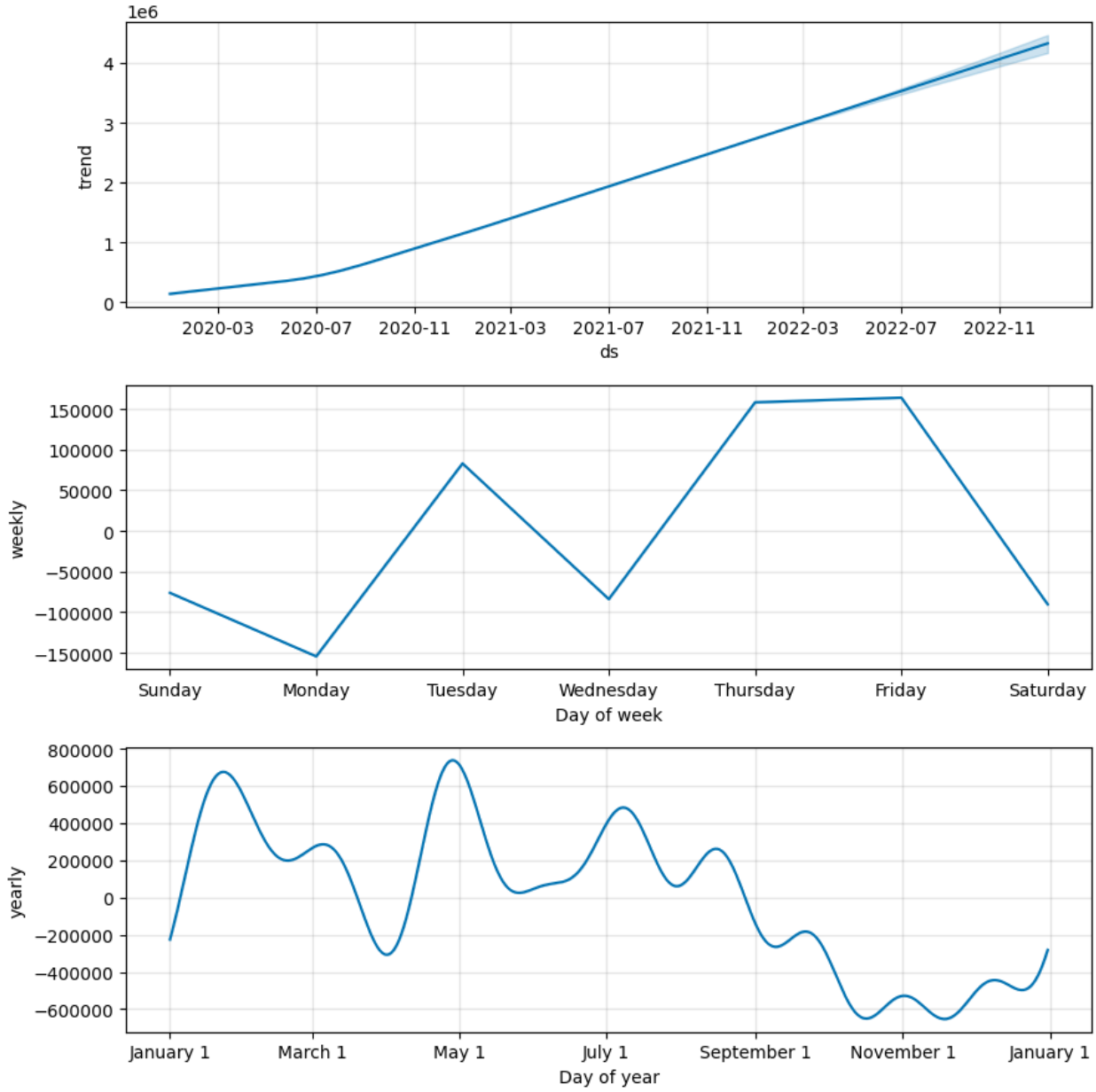
```
      Retailer  Retailer ID Invoice Date     Region     State
City  \
0  Foot Locker      1185732   2020-01-01  Northeast  New York  New
York
1  Foot Locker      1185732   2020-01-02  Northeast  New York  New
York
2  Foot Locker      1185732   2020-01-03  Northeast  New York  New
York
3  Foot Locker      1185732   2020-01-04  Northeast  New York  New
York
4  Foot Locker      1185732   2020-01-05  Northeast  New York  New
York

                     Product  Price per Unit  Units Sold  Total Sales
\
0      Men's Street Footwear            50.0        1200     600000.0

1    Men's Athletic Footwear            50.0        1000     500000.0

2    Women's Street Footwear            40.0        1000     400000.0

3  Women's Athletic Footwear            45.0         850     382500.0

4              Men's Apparel            60.0         900     540000.0


   Operating Profit  Operating Margin Sales Method
0          300000.0                0.50     In-store
1          150000.0                0.30     In-store
2          140000.0                0.35     In-store
3          133875.0                0.35     In-store
4          162000.0                0.30     In-store
```

```
import pandas as pd
from prophet import Prophet
import matplotlib.pyplot as plt


df['Invoice Date'] = pd.to_datetime(df['Invoice Date'])
daily_sales = df[['Invoice Date', 'Total Sales']].copy()
daily_sales = daily_sales.groupby('Invoice Date').sum().reset_index()
daily_sales.columns = ['ds', 'y']

# Fit Prophet model
model = Prophet()
model.fit(daily_sales)
```

```
# Create future dates
future = model.make_future_dataframe(periods=365, freq='D')

# Forecast
forecast = model.predict(future)

# Plot forecast
fig1 = model.plot(forecast)
plt.title("Sales Forecast ")
plt.xlabel("Date")
plt.ylabel("Total Sales")
plt.tight_layout()
plt.legend()
plt.show()

# Plot forecast components (trend, seasonality)
model.plot_components(forecast)
plt.tight_layout()
plt.show()
```

```
10:10:09 - cmdstanpy - INFO - Chain [1] start processing
10:10:09 - cmdstanpy - INFO - Chain [1] done processing
```





Overall trend of sales shows a steady increase.the forecast results shows that there is a strong seasonality with multiple peaks and troughs.

Tuesday, Thursday, and Friday show positive values, meaning sales tend to be higher on these days. Sunday, Monday, Wednesday, and Saturday show negative values, indicating sales drop relative to the baseline on these days.

February and May are months where we have sale peaks and moreover April and November and December shows a deep in sales.