Global Superstore Sales Forecasting

A Time Series Analysis

Group 4:

Festus , Fiyinfoluwa, Yemisi, Zion , Sophia, Franklyn,
Temitope, Tobi, Darasimi, Ibrahim, Fiyin, Gift

Project Goal

Overview:

- **Problem:** We started with a large, raw dataset of a global superstore's sales over four years.
- **Objective:** Our goal was to prepare this data, analyze its patterns, and then use a time series model to forecast sales for the next seven days.

Speaking Notes: Clearly state the project's purpose and how you will walk the audience through the process.

Data Preparation

What we did:

- **Data Cleaning:** We converted key columns like dates and sales values to the correct data types.
- **Aggregation:** We transformed the individual sales transactions into a single, comprehensive record of daily sales.
- **Continuity:** We filled in any days with no sales with a value of zero to create a continuous time series.

Speaking Notes: This slide explains the crucial first step of making the data usable for a time series model.

Data Preparation for Time Series

The initial raw data, which was already cleaned of data type and duplication errors in a previous step, was at the transactional level. To perform a time series analysis, it was aggregated to a daily level by summing sales for each unique date. Additionally, the dataset was checked for any missing dates to ensure the time series was continuous.

The aggregation and data cleaning process is documented in the following code.

```
# Load necessary libraries
library(dplyr)

library(ggplot2)

library(forecast)

library(tseries)

# Load the cleaned dataset

coffee_data <- read.csv("../Data/Coffee_Cleaned.csv")

# Ensure the Order.Date is in the correct format

# The format is year-month-day

coffee_data$Order.Date <- as.Date(coffee_data$Order.Date)
```

```
# Aggregate sales to a daily level

daily_sales <- coffee_data %>%

group_by(Order.Date) %>%

summarise(Total.Sales = sum(Sales)) %>%

ungroup()

# Handle missing dates by creating a continuous date sequence

daily_sales <- daily_sales[!is.na(daily_sales$Order.Date), ] full_date_range <-
data.frame(Order.Date = seq(min(daily_sales$Order.Date),
max(daily_sales$Order.Date), by = "day"))
```

```r
# Join the full date range with the sales data, filling in missing days with NA
daily_sales <- left_join(full_date_range, daily_sales, by = "Order.Date")

# Replace any NA values with 0
daily_sales$Total.Sales[is.na(daily_sales$Total.Sales)] <- 0

# Check the new structure to confirm changes

str(daily_sales)
```

Output: The `str()` output confirms a data frame with 1440 observations, representing 1440 days, with both the `Order.Date` and `Total.Sales` columns in the correct format.

Code:

```
colSums(is.na(daily_sales))

last_date <- max(daily_sales$Order.Date)

print(paste("The last date in the dataset is:", last_date))
```

Output: The `colSums(is.na(daily_sales))` output of `0` confirms that all missing sales values were successfully replaced, and the last date in the dataset is `2018-12-11`.

Exploratory Data Analysis (EDA)

Insights from Visualizations:

- **Trend:** The data shows a clear upward trend over the four-year period, indicating steady business growth.
- **Seasonality:** There is a strong, repeating yearly pattern, with sales peaking significantly at the end of each year.
- **Remainder:** This is the random fluctuation in sales not explained by the trend or seasonality.

Speaking Notes: This is where you explain what you learned from the data. Point to the different components of the decomposition plot to illustrate your findings.

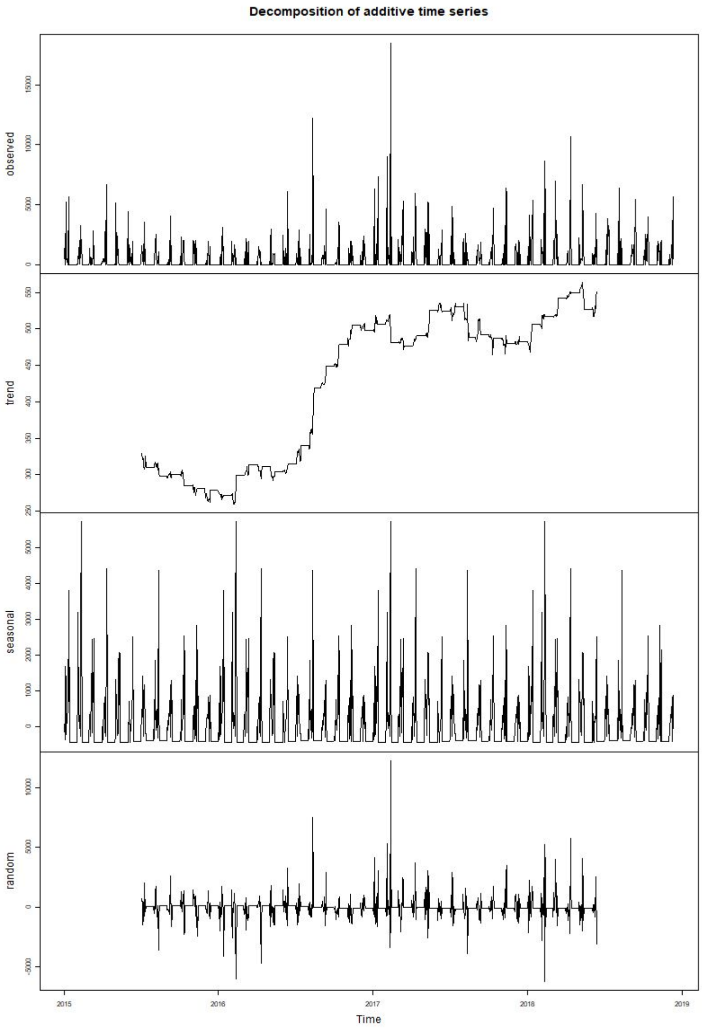
EDA

The next step in the analysis involved visualizing the sales data over time to identify underlying patterns such as trend and seasonality. This was done by converting the daily data into a time series object and plotting it.

```
# Convert the data to a time series object

start_date <- min(daily_sales$Order.Date)

sales_ts <- ts(daily_sales$Total.Sales, start = c(as.numeric(format(start_date, "%Y")), as.numeric(format(start_date,
"%j"))), frequency = 365)

# Plot the time series plot

(sales_ts, main = "Total Daily Sales", xlab = "Date", ylab = "Sales")

# Decompose the time series to visualize trend, seasonality, and remainder

sales_ts_decomp <- decompose(sales_ts) plot(sales_ts_decomp)
```

Analysis:

The `Total Daily Sales` plot reveals a clear upward trend over the four-year period, indicating consistent business growth. The decomposition plot further isolates this trend and also shows a strong yearly seasonal pattern, with sales typically peaking at the end of each year.



Decomposition of additive time series
observed
trend
seasonal
random
Time
2015
2016
2017
2018
2019

Statistical Modeling

Model: ARIMA (AutoRegressive Integrated Moving Average)

- **Why ARIMA?** This model is a great choice because it can effectively handle data with both a clear trend and a seasonal component.
- **How it worked:** We used an automated function in R to find the perfect-fitting ARIMA model for our specific dataset.

Speaking Notes: Explain the model in simple terms. You can say something like, "The model uses past sales trends and patterns to predict what will happen next."

An ARIMA model was chosen for forecasting due to its suitability for handling both trend and seasonal components in a time series. The `auto.arima()` function was used to automatically find the best-fitting model for the data, which was then used to forecast sales for the next seven days.

```
# Fit an ARIMA model to the sales data

fit_model <- auto.arima(sales_ts)

# Use the model to forecast sales for the next 7 days

forecast_sales <- forecast(fit_model, h = 7)

# Print the forecast results

print(forecast_sales)
```

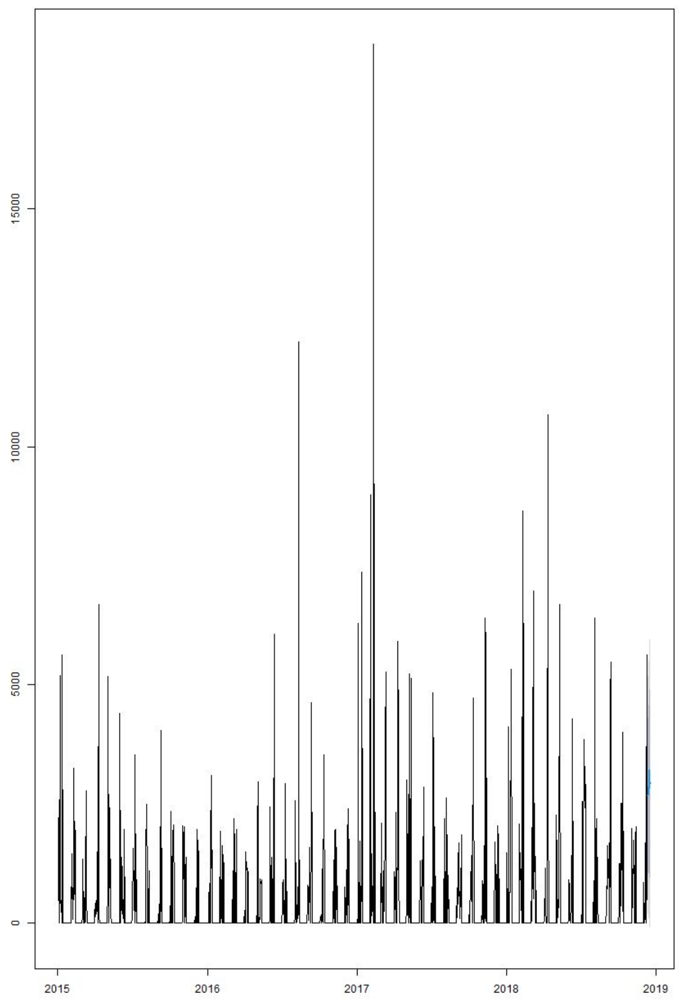
The 7-Day Sales Forecast

Our Forecast: We used the fitted model to generate a sales forecast for the week after the dataset's final date.

- Show the table with the point forecasts and confidence intervals.

Speaking Notes: Present the final results. Explain that the "Point Forecast" is the most likely prediction, and the `Lo` and `Hi` columns show the range of possibilities.

7-Day Sales Forecast



Conclusion & Implications

Key Findings:

1. The business shows a strong, growing trend in sales.
2. Sales are highly seasonal, which is an important factor for planning.
3. The forecasted sales for the upcoming week provide a solid basis for business decisions.

Implications: These predictions can help management make informed decisions about inventory, staffing, and marketing strategies for the upcoming week.

Speaking Notes: Conclude by summarizing your main points and highlighting the real-world value of your analysis.

Model Prediction

Based on the model's predictions, the following sales values are expected for the next seven days:

Day	Point Forecast
Day 1	$2679.42
Day 2	$2857.34
Day 3	$2806.79
Day 4	$2876.76
Day 5	$3215.43
Day 6	$2934.66
Day 7	$2944.32