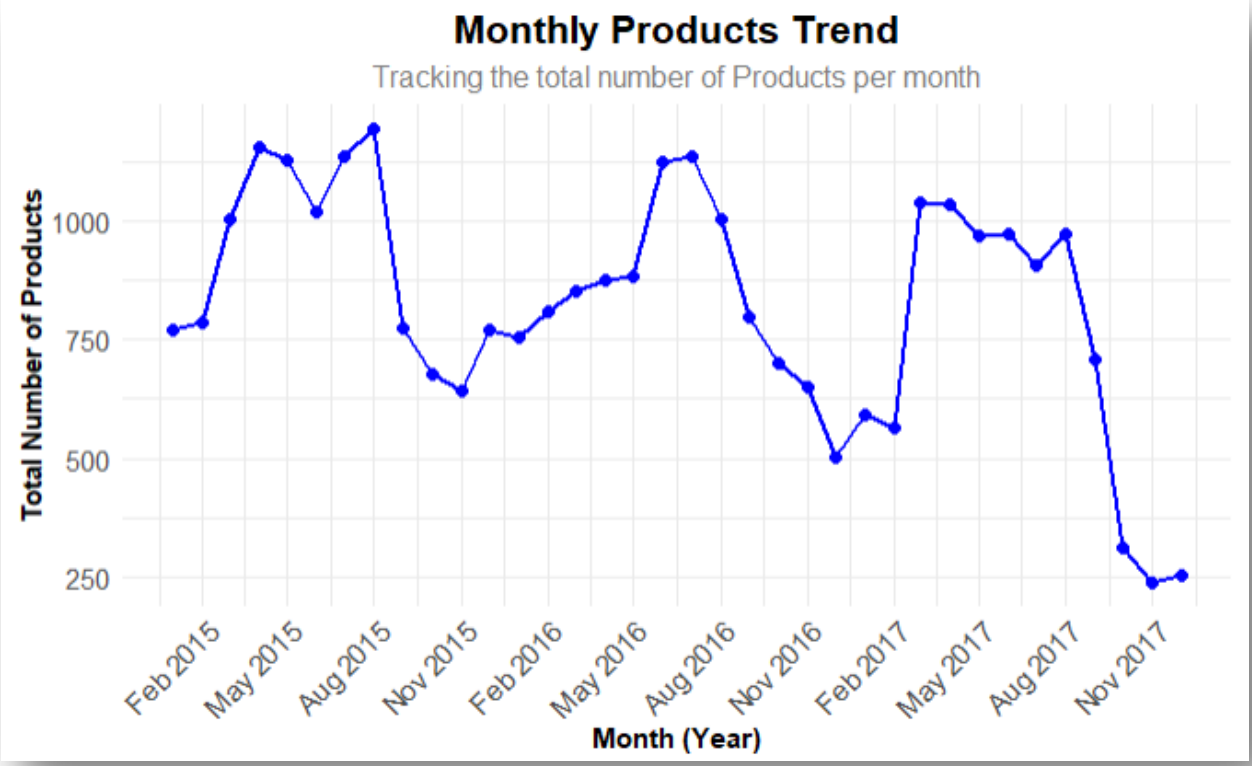
Group 9 - Data-Driven Optimisation of Supply Chain Management

Data mining problem 2 - To determine how we predict the future order demand in order to optimise supply chain capacity

```
library(forecast)
library(ggplot2)
library(dplyr)
library(lubridate)
library(readxl)
library(scales)

# Data mining problem 2 - How can we predict the future order demand in order
 to optimise supply chain capacity?

# Load the dataset
data <- read_excel("C:/Users/Xi/Desktop/incom2024_delay_example_dataset.xlsx
")

# Data Preprocessing
data <- na.omit(data)  # Remove rows with missing values
data <- unique(data)  # Remove duplicate rows
data <- data[data$customer_state != '91732', ]  # Filter out rows where custo
mer_state is '91732'
data$order_date <- as.Date(data$order_date, format = "%Y-%m-%d")  # Convert o
rder_date to Date format
data <- data %>%
  filter(order_date < as.Date("2018-01-01"))  # Keep only records with order_
date before 2018

data <- data %>%
  filter(!order_status %in% c("CLOSED", "CANCELED"))  # Exclude orders with s
tatus "CLOSED" or "CANCELED"

# Grouping data by month
monthly_data <- data %>%
  group_by(month = floor_date(order_date, "month")) %>%  # Group by month
  summarise(total_orders = sum(order_item_quantity, na.rm = TRUE))  # Calcula
te total orders per month

# Plotting the monthly product trend
ggplot(monthly_data, aes(x = month, y = total_orders)) +
    geom_line(color = 'blue', linewidth = 1) +
    geom_point(color = 'blue', size = 2) +
    labs(
        title = 'Monthly Product Trend',
        subtitle = 'Tracking the total number of products per month',
        x = 'Month (Year)',
        y = 'Total Number of Products'
```

```
    ) +
    scale_x_date(date_labels = "%b %Y", date_breaks = "3 months") +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
        plot.subtitle = element_text(hjust = 0.5, size = 12, color = "gray50
"),
        axis.title.x = element_text(face = "bold"),
        axis.title.y = element_text(face = "bold"),
        axis.text.x = element_text(angle = 45, hjust = 1, size = 11),
        axis.text.y = element_text(size = 11)
    )
```




```r
#The trend of order volume over time

# Convert data to a time series format
ts_data <- ts(monthly_data$total_orders, frequency = 12, start = c(2015, 1))

# Train the ARIMA model
arima_model <- auto.arima(ts_data)
summary(arima_model)
```

```
## Series: ts_data
## ARIMA(0,0,1)(1,1,0)[12] with drift
```

```
##
## Coefficients:
##          ma1     sar1    drift
##       0.3923  -0.5433  -8.8379
## s.e.  0.1623   0.2074   2.3312
##
## sigma^2 = 17695:  log likelihood = -152.01
## AIC=312.01   AICc=314.12   BIC=316.72
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE     MASE       A
CF1
## Training set 1.968449 101.5968 69.70962 -4.29687 12.39157 0.448951 0.06160
744

# Generate forecast for the next 12 months (2018)
arima_forecast <- forecast(arima_model, h = 12)

# Build a forecast dataframe for plotting
forecast_df <- data.frame(
  month = seq(from = as.Date("2018-01-01"), by = "month", length.out = 12),
  Forecast = arima_forecast$mean,
  Lower_CI = arima_forecast$lower[, 2],
  Upper_CI = arima_forecast$upper[, 2]
)

# Plot actual vs. forecasted values with customized titles and labels
ggplot() +
  geom_line(data = monthly_data, aes(x = month, y = total_orders), color = 'b
lue', linewidth = 1) +
  geom_point(data = monthly_data, aes(x = month, y = total_orders), color = '
blue', size = 2) +  # Points for actual values
  geom_line(data = forecast_df, aes(x = month, y = Forecast), color = 'red',
linewidth = 1) +
  geom_point(data = forecast_df, aes(x = month, y = Forecast), color = 'red',
 size = 2) +  # Points for forecasted values
  geom_ribbon(data = forecast_df, aes(x = month, ymin = Lower_CI, ymax = Uppe
r_CI), fill = 'lightgrey', alpha = 0.4) +
  labs(
    title = 'Comparison of Actual and Forecasted Monthly Product Totals',
    subtitle = 'Blue represents actual data (2015-2017), while red shows fore
casted values for 2018',
    x = 'Month (Year)',
    y = 'Total Number of Products'
  ) +
  scale_x_date(date_labels = "%b %Y", date_breaks = "3 months") +  # Monthly
labels with quarterly breaks for readability
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
```

```
    plot.subtitle = element_text(hjust = 0.5, size = 12, color = "gray50"),
    axis.title.x = element_text( size = 12),
    axis.title.y = element_text( size = 12),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10))
```