# Analytics for Supply Chain and Distribution Plan (OM 286)

# Final Project: Forecasting Store Sales

Prepared For: Dr. Genaro Gutierrez


Written By:

Final Project Group 2

Ashley Hattendorf (ahh864)

Brooks Li (zl22449)

Brinda Asuri (bva246)

Sarah Dominguez (sad3396)

Twinkle Panda (tp26545)

# 1. Executive Summary

This project aims to develop a comprehensive forecasting model that accurately predicts sales across 54 stores and 33 product families of an Ecuador company, *Corporacion Favorita*. The project implements advanced forecasting techniques, including ARIMA and Bayesian modeling to achieve this end goal. These methodologies are applied to explore trends, seasonality, and external factors within the datasets that can potentially influence sales. In addition, the project also incorporates a range of supplementary data, such as promotional information, transaction records, holiday event schedules, and daily oil prices. Combining these factors, these variables provide a detailed understanding of the dynamics influencing store sales performance across different store locations and their product families in different countries and regions.

Accurate sales forecasting is the cornerstone of effective inventory management, resource planning, and supply chain optimization. Companies often face the challenges of avoiding stockouts, which frustrate customers and negatively impact store revenue, and preventing overstocking, which increases costs and produces additional waste. An easy-to-use and robust forecasting model can address these challenges by allowing businesses to make informed decisions about inventory, promotions, and resource allocation. This project tackles this challenge by applying analytical techniques to capture the complexities of sales patterns while considering external factors that will influence consumer demand.

In terms of the dataset that we used, it is a store sales dataset used for a Kaggle competition, aiming to predict sales for product families sold at Favorita stores across Ecuador and other South American countries. The challenge involves leveraging time-series forecasting to accurately predict future sales using historical data and other supplementary information. The dataset includes historical sales figures, promotional data, and holiday events, allowing us to develop models that account for various factors that can influence sales trends.

The dataset comprises several key features, including a target variable and daily sales. The time-series data spans a specific period (2013-2017), with store information such as *store_nbr* (identifying the store) and metadata such as city, state, store type, and clusters of similar stores. Product-related details include *family*, which specifies the product type, and *onpromotion*, showing the number of promoted items within a product family on a given date. The target variable is daily unit sales, with some fractional values representing non-integer unit

sales. Additional features include daily oil prices, reflecting Ecuador and other South American countries' sensitivity to oil price fluctuations, and holiday metadata, such as holidays, transferred holidays, bridge days, and workdays, which have shown to have some effect on consumer behavior and sales patterns.

A significant project component involves Exploratory Data Analysis (EDA) and feature engineering. Through EDA, we uncovered trends and patterns in sales, seasonality, and the effect of external factors like holidays and promotions. EDA also helped us identify anomalies and trends influenced by socio-economic events, such as the bi-weekly wage payments and the April 2016 earthquake in Ecuador. On the other hand, feature engineering allowed us to improve and organize the dataset.

We performed several steps to clean, organize, and enhance the datasets for analysis and modeling. We started by importing multiple datasets, including training, test, store, holiday, and oil data, and ensured consistency by converting date columns to a standard format and removing duplicate rows. To enrich the holiday data, we created a binary "holiday_flag" column to indicate the presence of holidays or events, simplifying the inclusion of special dates in the analysis. We also grouped 33 detailed product categories into 7 broader groups (e.g., "Food" and "Electronics") to simplify analysis and reduce noise. Sales and promotional data were aggregated by date, store, and product category, resulting in a more concise and organized dataset.

We merged datasets to create a comprehensive view by incorporating information like holiday effects and oil price fluctuations. Missing values in numerical columns were handled using null imputation, replacing NA values with zeros to ensure consistency and interpret missing data as an absence of activity. Finally, categorical variables were transformed into a numerical format using one-hot encoding when necessary. The final processed datasets were saved for analysis or model training. These steps ensured the data was clean, consistent, and well-structured for subsequent tasks.

We experimented with ~6 ARIMA modeling approaches. Our champion model was a SARIMA() model trained with 3 regressors to predict the sales of 30 days for a particular store cluster, with MAPE of 6.51% and MAE of 1.58%. Our challenger model was a base ARIMA() model, forecasting the day-level sales with a MAPE of 7.18% and an MAE of -2.39%. Both models significantly outperformed the baseline naïve forecast, with a MAPE of 35.03% and an MAE of -34.90%.

| Model | Level | MPE (%) | MAPE (%) |
|---|---|---|---|
| ARIMA() | Day | -2.39 | 7.18 |
| ARIMA() + 3 Regressors | Day | -0.86 | 9.28 |
| SARIMA() + 3 Regressors | Day | 11.95 | 13.36 |
| SARIMA() + 3 Regressors | Category-Day | -2.30 | 9.71 |
| ARIMA() | StoreCluster - Day | 12.95 | 13.55 |
| ARIMA() + 3 Regressors | StoreCluster - Day | 5.48 | 7.84 |
| SARIMA() + 3 Regressors | StoreCluster - Day | 1.58 | 6.51 |
| ARIMA() + Lagged Onpromo | Day | 8.26 | 10.55 |
| Naive | Day | -34.90 | 35.03 |

Our analysis showed that forecast accuracy for retail sales data can be greatly increased by considering the proper level of aggregation and adding both seasonal components and external regressors. Aggregating data at the store cluster level proved particularly effective, with the SARIMA model achieving a MAPE of 6.51%, the highest accuracy among all models. This approach captured consistent patterns across similar stores while leveraging critical external factors such as promotions, holidays, and oil prices. Additionally, incorporating lagged promotions revealed the lasting impact of past marketing efforts on future sales, emphasizing the importance of strategic promotional planning. While the initial analysis began with approximately 10 features, many proved unsuitable for the final models. Future iterations could incorporate new engineered features to enhance predictive power further. Implementation of forecasting at various hierarchical levels and experimenting with hybrid or ensemble approaches that combine multiple model strengths could also improve accuracy beyond the current champion model's MAPE of 6.51% and the challenger model's MAPE of 7.18%.

In addition to ARIMA modeling, we experimented with Bayesian forecasting to address any additional complexities in the dataset. By additionally incorporating Bayesian models, we could account for uncertainty more effectively, especially when considering external factors such as holidays and oil prices. To build this model, we integrated seasonal components, such as sine and cosine harmonics, holiday indicators, and macroeconomic factors, such as oil prices. However, our Bayesian forecasts were less accurate due to several challenges and yielded more uninterpretable results. Our original dataset comprised over three million rows, and our processed dataset comprised over 600 thousand rows. and when running the Bayesian mode,l the project teams' computers could not handle running simulations on such large chunks of data. This resulted in the team using a subset of the data to gain some insights into future sales, but the model still showed high levels of variability and an inability to capture extreme demand spikes. Due to the computational demands of running the Baysian model and the project time frame, the team deduced that Baysian forecasting was impractical to interpret and that our ARIMA models yielded more confident, accurate, and interpretable results.

This project is particularly relevant in retail and consumer goods industries, where accurate future sales forecasting is the key to optimizing inventory and supply chain management to reduce inefficiencies and enhance customer satisfaction. By integrating advanced forecasting methods with industry-specific knowledge, this project will be able to demonstrate the value of data when it comes to deciding how much inventory to purchase in the future. Furthermore, it provides a replicable framework for tackling similar forecasting challenges in other industries and regions.

In summary, our goal is to combine advanced forecasting and analytical techniques, gain an understanding of retail operations, and focus on real-world applicability. By fully addressing each complexity of sales forecasting, it demonstrates the power of using data to make informed inventory optimization decisions and drive meaningful improvements in a company's performance and customer satisfaction rate.

# 2. Scope, Objective, and Relevance

## 2.1 Scope:

The scope of this project focuses on developing a comprehensive forecasting model to predict daily sales volume across all Favorita stores and product families sold within each store. The

dataset spans multiple years of historical sales data from Ecuadorian stores, providing a good foundation for analyzing sales trends, seasonality, and the impact of external factors each year. Our project will implement advanced forecasting methods, such as ARIMA, to address the complex relationship of real-world retail data. We will also attempt Bayesian forecasting to capture additional seasonality and any additional variance. In addition, the project also focuses on understanding the broader context in which sales occur by integrating external variables, such as holiday schedules, oil prices, and other macroeconomic and socio-political indicators, to understand the influence of sales better.

## 2.2 Objective:

This project aims to develop an accurate and reliable forecasting model to predict daily sales volume for Favorita stores and their various product families using the Kaggle dataset "Store Sales - Time Series Forecasting." Many companies worldwide face problems managing inventory, especially around holidays when demand may be high, and promotions are more relevant. Understanding how such factors will influence demand and forecasting sales, particularly with a foreign company in a country like Ecuador, where there are various cultural and work holidays and generally many external factors, poses an interesting challenge. Favorita will be able to understand better how they should manage inventory and operations with future sales predictions.

## 2.3 Relevance:

This project is highly relevant to daily corporate decision-making, as accurate sales forecasting is a key component of supply chain management. Companies across all industries need to understand how demand will change over time for each product to optimize their revenue, inventory management, and general daily operations. For example, companies will know when to push promotional sales and enhance pricing strategies to maximize profits. Additionally, with inventory management, precise forecasts will enable businesses to maintain the right inventory levels, reducing any risk of under or overstocking, thereby minimizing costs. Knowing approximate future sales can also allow companies to be more efficient in their warehousing and distribution processes, ensuring that the number of products obtained meets demand requirements and is delivered in a timely manner. This can also increase customer satisfaction, as

products will be more available when customers need them. Without being able to forecast customer demand, companies would be very disorganized and uncertain of how to optimize their cash flow and product materials and generally make long-term decisions that foster growth. By choosing this project, we can practice tackling what companies deal with daily: understanding how various factors may affect sales and future demand results and how to drive effective supply chain management to achieve a successful business venture.

# 3. Data Used

The Store Sales - Time Series Forecasting dataset is part of a competition to predict sales for thousands of product families sold at **Favorita stores in Ecuador**. The challenge is to use time-series forecasting to accurately predict future sales, leveraging a rich set of historical data and supplementary information. This dataset includes historical sales figures, promotional information, and holiday events, enabling the development of models that account for various factors influencing sales trends. The goal is to help grocery retailers optimize inventory, minimize waste, and improve customer satisfaction.

## 3.1 Features and Target:

1. **Dates**: Time-series data spanning a specific period
2. **Store Information**:
    - *store_nbr*: Identifies the store where sales occurred
    - Metadata like city, state, store type, and clusters (groups of similar stores)
3. **Product Information**:
    - *family*: Indicates the product type being sold
4. **Promotions**:
    - *onpromotion*: Number of promoted items within a product family on a given date
5. **Sales**: The target variable representing daily unit sales. Fractional values indicate sales in non-integer units (e.g., 1.5 kg of cheese).
6. **Oil Prices**: Daily oil prices, reflecting Ecuador's dependency on oil and its economic sensitivity to price changes.

7. **Holidays and Events**:
    - Includes metadata on holidays, transferred holidays, bridge days, and work days, which affect consumer behavior and sales.

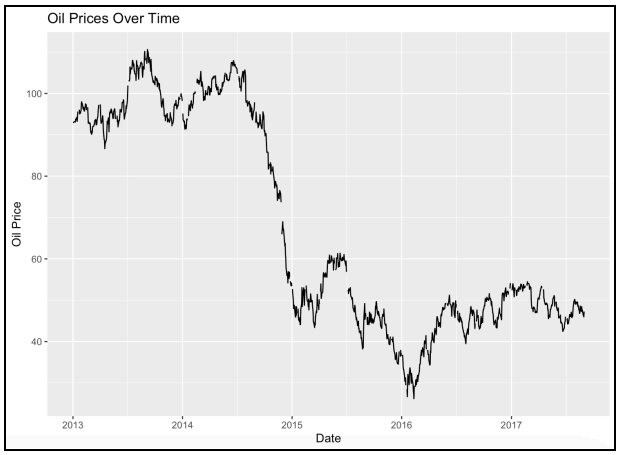
### 3.2 Files in the Dataset

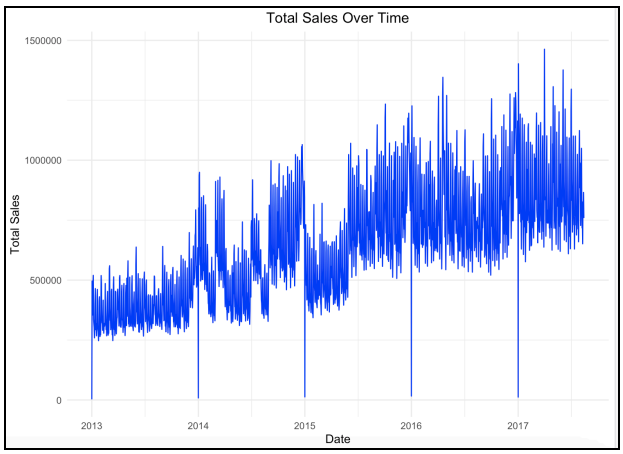
1. **train.csv**:
    - Historical sales data for training the model, including *store_nbr, family, onpromotion*, and the target variable *sales*
2. **test.csv**:
    - Contains the same features as the training set but without sales. Predictions are required for these dates
3. **stores.csv**:
    - Metadata about stores, such as their city, state, type, and cluster
4. **oil.csv**:
    - Daily oil price data for both training and testing periods
5. **holidays_events.csv**:
    - Includes holiday information, with a special focus on transferred holidays and additional days that may impact sales

## 4. Exploratory Data Analysis

### 4.1 General Trends

To begin our analysis, we first analyzed some generic trends in the data, focusing on oil prices and total sales over time. The oil price graph, as shown below, revealed significant fluctuations between 2013 and 2017, with a sharp decline starting in mid-2014. Prices fell from over $100 per barrel to below $40 per barrel by early 2016. This drop likely reflects a major economic shock, particularly for Ecuador, an oil-dependent country. This period of low and fluctuating oil prices may have influenced consumer spending patterns and overall economic activity, indirectly impacting store sales.



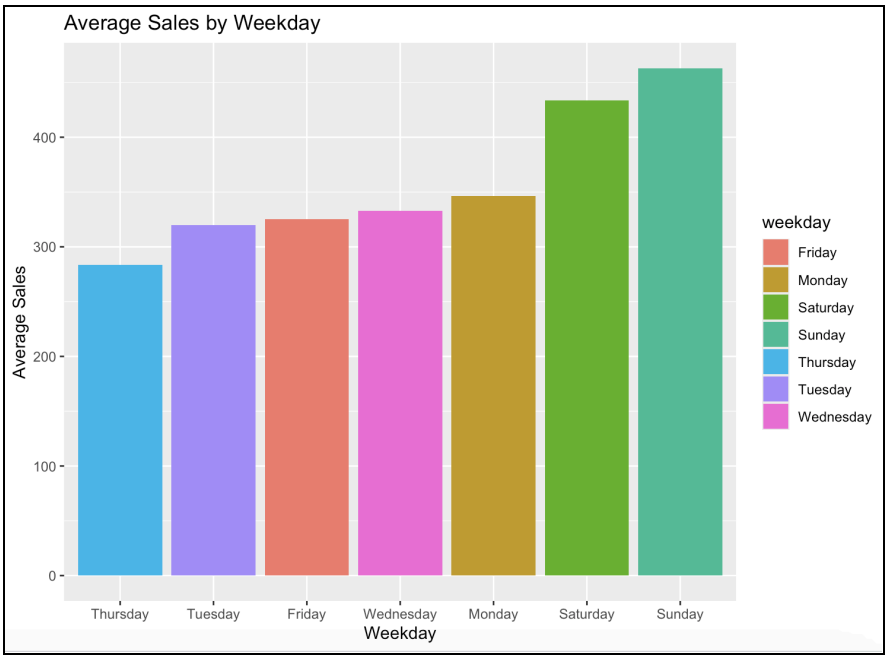




Interestingly, despite the drop in oil prices, the total sales graph showed an upward trend from 2013 to 2017, indicating steady growth in sales over the period. This does not necessarily mean that oil prices had no impact on sales. Instead, it suggests that other factors may have mitigated or offset this potential negative effect. For instance, the business could have expanded its operations by opening new stores, increasing product offerings, or launching successful promotional campaigns that drove sales growth.

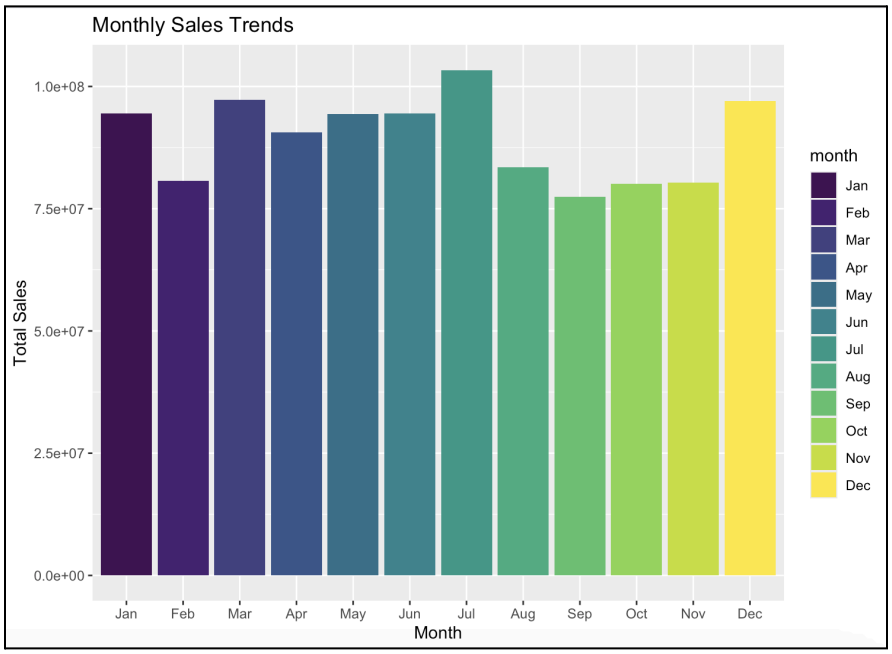
Overall, this upward trajectory highlights the business's resilience and adaptability to external challenges. While the declining oil prices might have indirectly influenced sales patterns, the observed growth suggests that strategic efforts, consumer demand for essential goods, and effective promotions played a significant role in sustaining and driving sales during this period.

## 4.2 Weekly and Monthly Analysis

We then continued our analysis by looking at the data weekly and monthly. The Average Sales by Weekday bar chart, as shown below, highlights consumer behavior patterns across the week. Weekends, particularly Sunday and Saturday, exhibit the highest average sales, reflecting a strong preference for shopping during this time. On the other hand, weekdays such as Tuesday and Thursday have the lowest average sales. This remains consistent with our general beliefs, allowing businesses to capitalize on these trends by allocating additional staff and inventory to handle higher customer traffic on weekends while leveraging promotions or discounts on lower sales weekdays to balance weekly sales distribution.

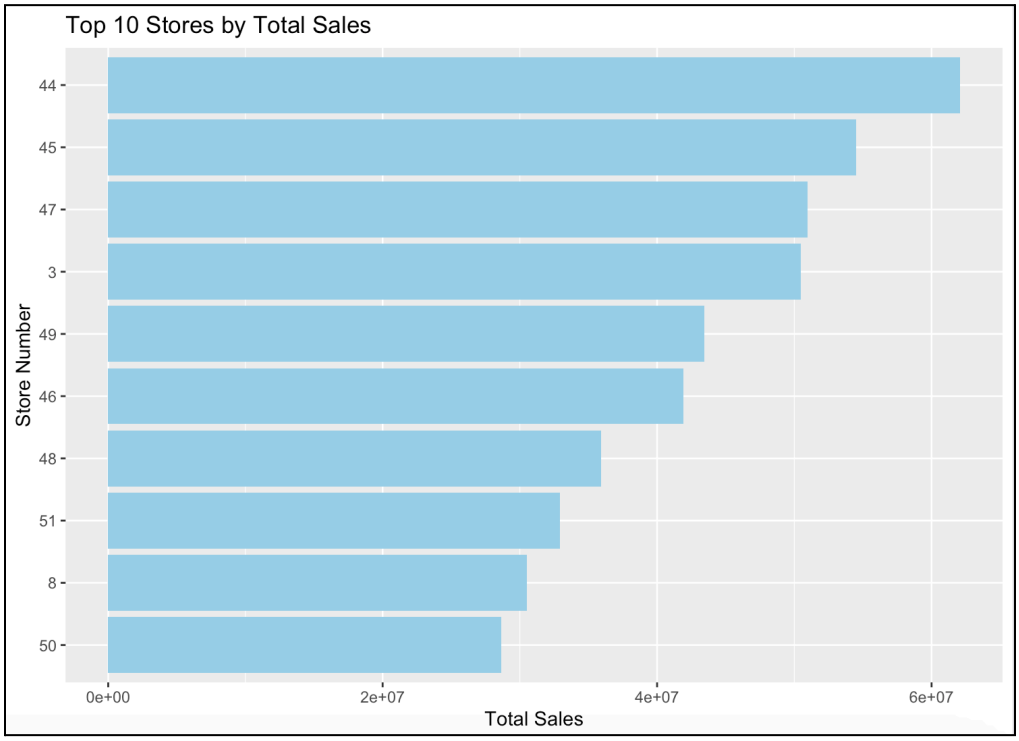



Looking at sales on a monthly level, the Monthly Sales Trends bar chart reveals significant patterns in sales performance across months. January, March, July, and December stand out as the months with the highest total sales, while February and September show relatively lower sales. This seasonality indicates opportunities to allocate resources strategically, such as ramping up inventory and marketing efforts during high-sales months. Conversely, targeted promotions or campaigns during underperforming months could help boost sales during these periods.



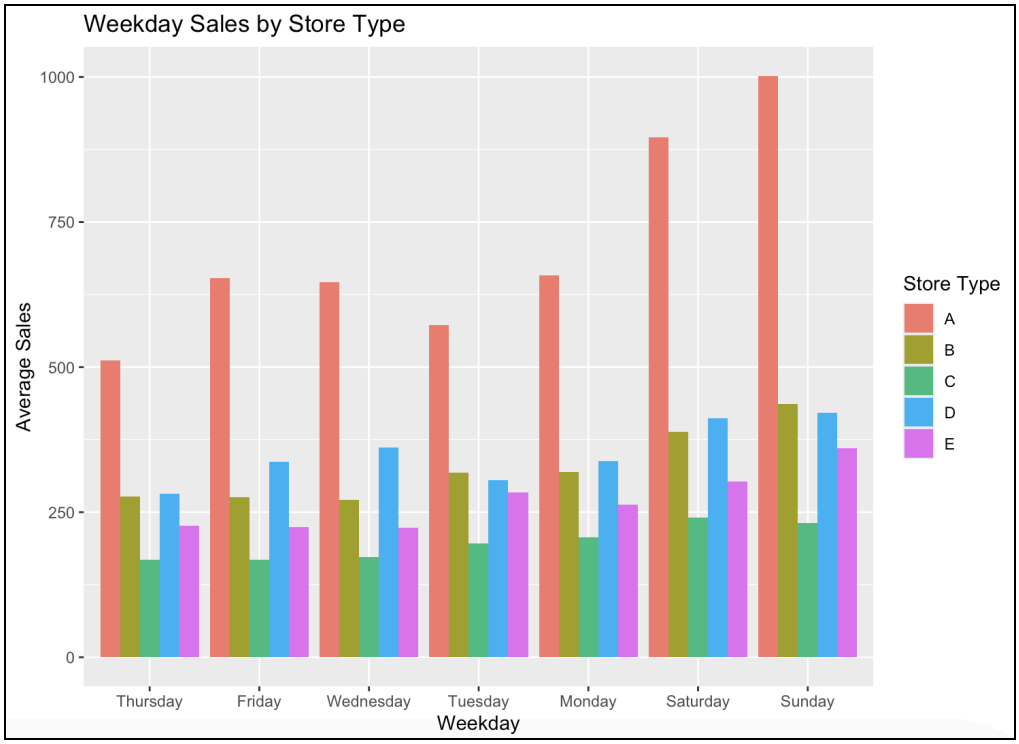
## 4.3 Individual Store Analysis

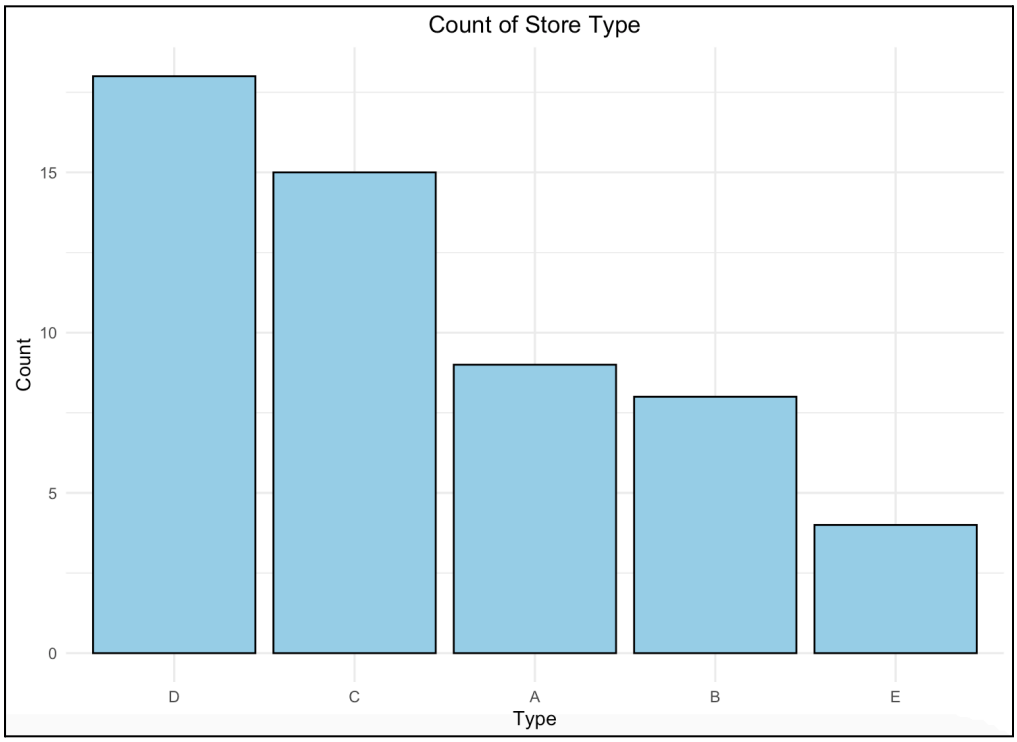
The bar chart below highlights the highest-performing stores based on their total sales, with store number 44 leading, followed closely by stores 45 and 47. High-performing stores contribute significantly to overall revenue, likely due to factors such as location, effective sales strategies, or strong customer loyalty. Analyzing top-performing stores' operations, customer demographics, and marketing strategies can help identify best practices to implement across the business and in stores that may not perform well. Additionally, sustaining the performance of top-ranking stores is crucial. Conducting deeper analyses into factors such as location and product offerings can explain gaps in performance and guide Favorita's decisions to help maximize revenue across all stores.




## 4.4 Store Type Analysis

Similarly, we looked at whether the store type impacted sales.




The Count of Store Type bar chart on the bottom provides an overview of the distribution of store types in Ecuador. Store type D has the highest count, followed by type C, with type E

having the least representation. This distribution suggests that types D and C form the backbone of the store network, but their performance relative to other types raises questions about efficiency and profitability.

The Weekday Sales by Store Type bar chart on top highlights the average sales performance for each store type across weekdays. Store type A consistently outperforms all other types in sales throughout the week, with sales peaking on weekends, keeping consistent with our results from above. In contrast, despite their higher count, type D stores show only moderate sales performance, while type C stores lag significantly behind.

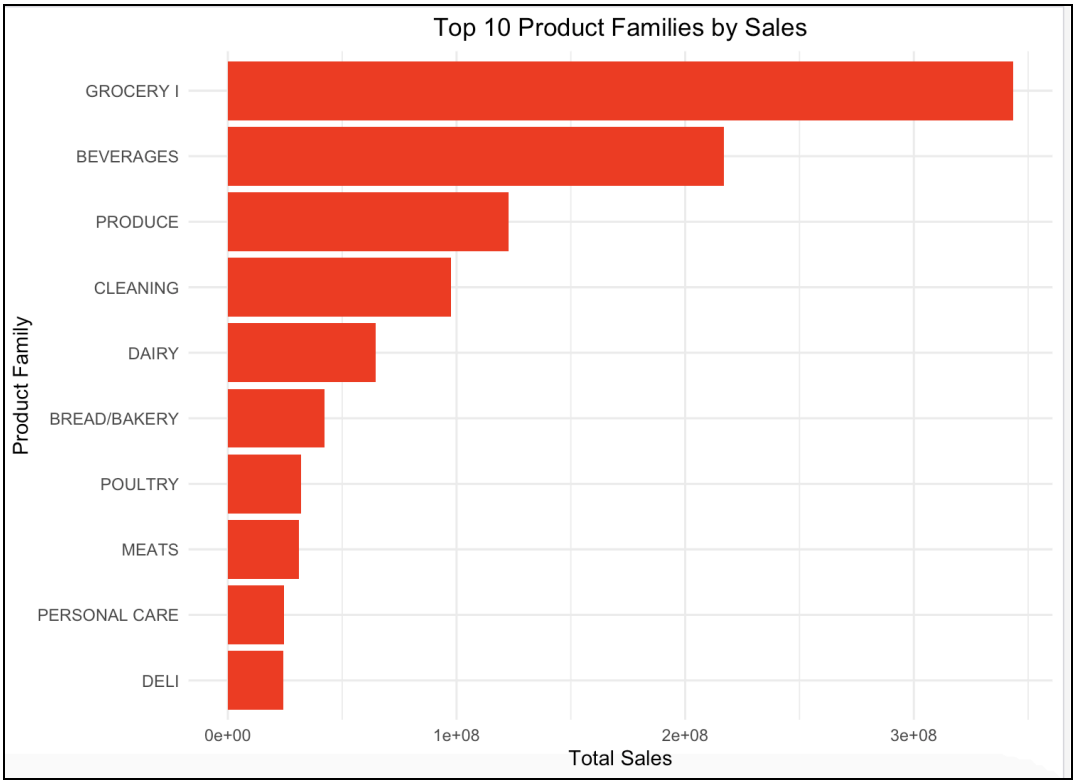
The high count of store types D and C presents an opportunity to improve their average sales by adopting strategies from type A stores, demonstrating strong sales performance despite its smaller presence. With C stores' low representation and weak sales, they should be evaluated for long-term viability. Additionally, type A stores could leverage their success by expanding their presence or further enhancing their operations.

## 4.5 Product Family Analysis

Finally, we examined the product families with the highest sales. Unsurprisingly, grocery and beverages were the two families selling the most products. These categories are critical areas for maintaining and expanding market share. Favorita should prioritize inventory management, promotions, and customer satisfaction within these categories to sustain its strong performance.

Categories like Cleaning and Dairy demonstrate solid performance, indicating steady demand. Efforts to cross-promote these products with higher-performing categories could boost sales further. Additionally, analyzing customer preferences and pricing strategies could reveal growth opportunities.

Lastly, Bread/Bakery, Poultry, Meats, Personal Care, and Deli are important but underperforming compared to the top tiers. Favorita should evaluate these categories to identify potential issues, such as limited product variety, pricing concerns, or insufficient marketing, and implement strategies to enhance their performance.




## 5. Data Pre-Processing

In our data preprocessing pipeline, we undertook a series of steps to ensure the data was clean, consistent, and ready for analysis or model training. Initially, we imported all the datasets, including train, test, store, holiday, and oil data. Each dataset serves a unique purpose in enriching the final dataset. We ensured consistency across these datasets by converting date-related columns into the same standard Date format, enabling efficient time-based operations and analysis. Duplicate rows were also removed from all datasets if they were found to maintain data quality and avoid redundancy.

A key enhancement was applied to the holiday data, where we created a new column named “holiday_flag” that marks these dates with a binary indicator (1 for holidays/events and 0 otherwise). This step allowed us to incorporate the impact of special dates into our model without additional complexity, such as what the specific holiday was.

In the product family column, we reduced 33 detailed categories that explained what items were being bought in a day into 7 broader groups, such as "Food," "Non-Food," and "Electronics." This grouping simplified the analysis and helped reduce noise by focusing on more meaningful category distinctions. We combined sales and promotional data by grouping it based on date, store, and product category. In this process, we added up each group's total sales and promotion counts, creating a simpler and more organized dataset that focuses on the key metrics.

Next, the datasets were merged to create a comprehensive view. This involved joining the holiday data, oil prices, and store information with the training data based on shared columns like date and store_nbr. This integration allowed us to include external influences such as holiday effects and fluctuations in oil prices, which could impact store operations and sales.

We applied a process called null imputation to handle missing values in the data. For all numerical columns in both the training and testing datasets, we replaced any missing (NA) values with zeros. This approach ensured that any gaps in the data, such as missing transactions or oil price values, would not disrupt our analysis or modeling process. We maintained consistency across the datasets by filling these (NA) values with zeros. We treated the missing data as an absence of activity (e.g., no sales or recorded oil price for a specific date).

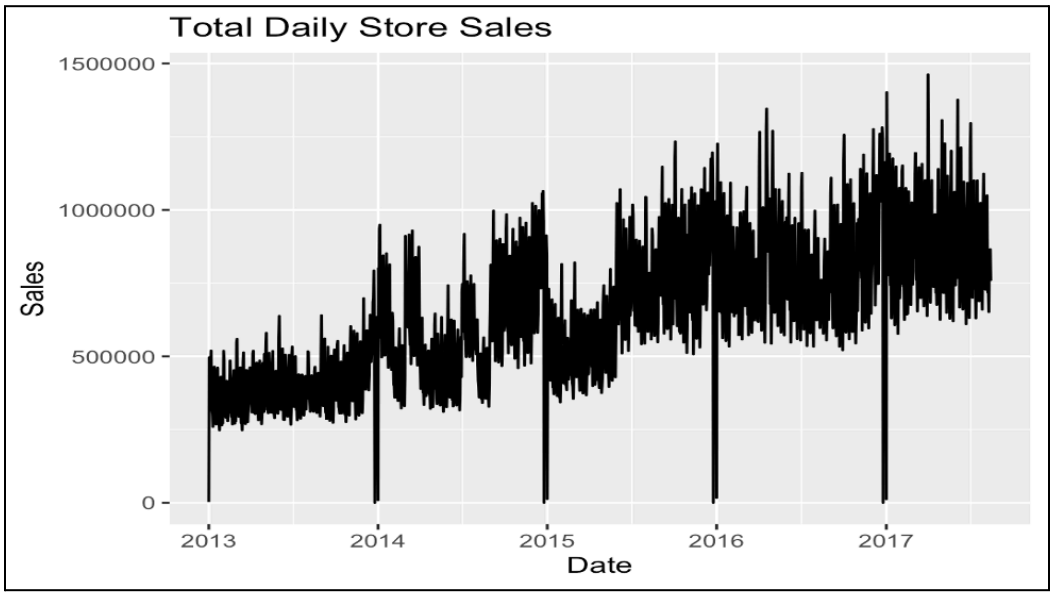
Finally, we transformed categorical variables into numerical representations through one-hot encoding if needed, and if there weren’t any categorical columns, everything would return to the original data frame. The final outputs were saved as processed datasets, ready for further analysis or model training.

# 6. Models

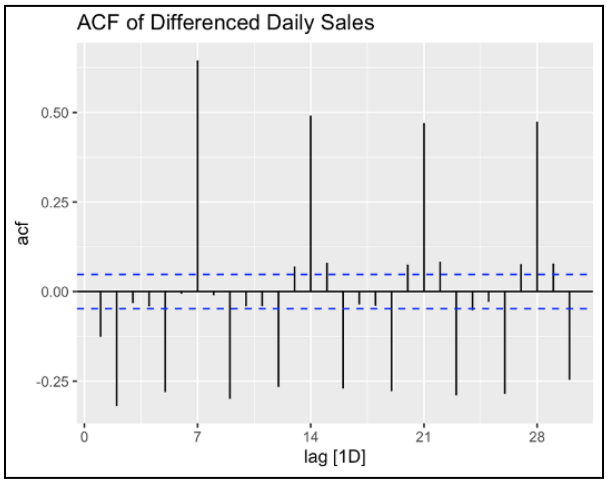
## 6.1 ARIMA Forecasting

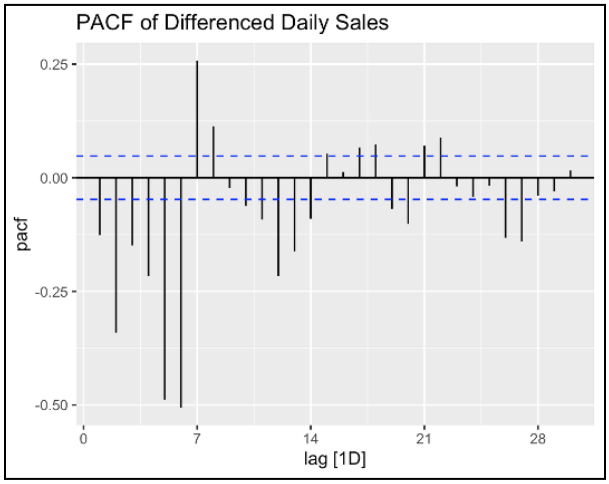
### 6.1.1 Data Diagnostics

### a. Sales Trend




### b. ACF & PACF




**Model Implications**

These patterns suggest the data exhibits strong weekly seasonality (7-day cycle). The presence of significant PACF spikes at early lags suggests an AR component. The alternating pattern in the ACF suggests a possible combination of seasonal and non-seasonal components. This suggests considering a seasonal ARIMA model with period=7, including both AR and MA terms, using seasonal and regular differencing for ARIMA modeling. A possible model structure could be SARIMA(P,D,Q)7.

**c. ADF Test Result**

```
        Augmented Dickey-Fuller Test

data:  .
Dickey-Fuller = -7.3576, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary
```

This result suggests that our time series is likely stationary. The p-value ($0.01 < 0.05$) rejects the null hypothesis of the ADF test. The null hypothesis of the ADF test is that the time series has a unit root, which implies non-stationarity. The large negative value of the test statistic (-7.3576) also supports the conclusion of stationarity, as more negative values provide stronger evidence against the null hypothesis. This result suggests that we may not need to apply differencing to your sales data, as it appears to be stationary. We could potentially use ARIMA models with d=0, focusing on identifying appropriate values for p and q.

### 6.1.2 Modelling Approaches

**Train Test Split:** We used all data except the last 30 days for training and only the last 30 days as a test dataset.

**1. Day Level Forecasting Models - ARIMA() , ARIMA() with Regressors**

We summarized the metrics across all stores and Family categories at a day level to run the model and produce forecasts at a specific day level. We built 2 models, ARIMA() and ARIMA() with 3 regressors: onpromotion,  holiday_flag, dcoilwtico.

```
# Fit the model on training data
fit <- train_sales %>%
  model(
    arima_auto = ARIMA(sales),
    arima_with_regressors = ARIMA(sales ~ onpromotion + holiday_flag + dcoilwtico)
  )
```

```
fit %>% report()
A tibble: 2 × 8
.model                  sigma2 log_lik    AIC   AICc    BIC ar_roots  ma_roots
<chr>                    <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>    <list>
arima_auto          10809534660. -21417. 42846. 42846. 42879. <cpl [0]> <cpl [17]>
arima_with_regressors 10895853066. -21446. 42915. 42915. 42974. <cpl [1]> <cpl [17]>
```

**Evaluation**

| | .model | .type | ME | RMSE | MAE | MPE | MAPE | MASE | RMSSE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | arima_auto | Test | -18327.175 | 92189.24 | 61439.43 | -2.3998381 | 7.182327 | NaN | NaN | 0.5968671 |
| 2 | arima_with_regressors | Test | 5107.369 | 97140.64 | 79098.46 | -0.8659563 | 9.283530 | NaN | NaN | 0.6503678 |

MASE and RMSSE could be NaN because of insufficient seasonal data or scaling factor issues.

**2. Day Level Forecasting Model - Seasonal ARIMA() with Regressors**

We include seasonal components based on the ACF and PACF plots, which display clear weekly seasonality (lag 7) and significant spikes.

```
# Fit seasonal ARIMA model with regressors
fit_seasonal <- train_sales %>%
  model(
    sarima_reg = ARIMA(sales ~ onpromotion + holiday_flag + dcoilwtico + 1 +
                         pdq(2,1,1) + PDQ(1,1,1,7))
  )
```

**Evaluation**

| | .model | .type | ME | RMSE | MAE | MPE | MAPE | MASE | RMSSE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | sarima_reg | Test | 99707.66 | 138672.2 | 111497.6 | 11.95249 | 13.35976 | NaN | NaN | 0.6578433 |

**3. Category-Day Level Forecasting Model - Seasonal ARIMA() with Regressors**

We tried to split the data across categories and tried to forecast for 1 category- 'FOOD.'

```
fit_food <- train_sales %>%
  model(
    sarima_food = ARIMA(sales ~ onpromotion + holiday_flag + dcoilwtico + 1 +
                          pdq(6,1,1) + PDQ(1,1,1,7))
  )
```

**Evaluation**

| | .model | .type | ME | RMSE | MAE | MPE | MAPE | MASE | RMSSE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | sarima_reg | Test | -17518.91 | 91902.34 | 71213.34 | -2.304861 | 9.714255 | NaN | NaN | 0.584238 |

### 4. Store Cluster-Day Level Forecasting Models

We tried to split the data across clusters and then forecast for 1 store cluster – 13. We built 3 models: ARIMA(), ARIMA() with 3 regressors and SARIMA() with 3 regressors

```
# ARIMA
fit_arima_sc <- train_sales %>%
  model(
    arima_auto = ARIMA(sales),
    arima_with_regressors = ARIMA(sales ~ onpromotion + holiday_flag + dcoilwtico)
  )
```

```
# SARIMA
fit_sarima_sc <- train_sales %>%
  model(
    sarima_sc = ARIMA(sales ~ onpromotion + holiday_flag + dcoilwtico + 1 +
                              pdq(6,1,1) + PDQ(1,1,1,7))
  )
```

**Evaluation**

| | .model | .type | ME | RMSE | MAE | MPE | MAPE | MASE | RMSSE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | arima_auto | Test | 7370.086 | 9642.995 | 7648.882 | 12.950588 | 13.55137 | NaN | NaN | 0.2617650 |
| 2 | arima_with_regressors | Test | 3309.830 | 5855.396 | 4388.145 | 5.483493 | 7.84188 | NaN | NaN | 0.6032456 |

| | .model | .type | ME | RMSE | MAE | MPE | MAPE | MASE | RMSSE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | sarima_sc | Test | 992.6364 | 4955.519 | 3411.24 | 1.588075 | 6.519514 | NaN | NaN | 0.5837512 |

### 5. Day Level Forecasting Models Including Lagged *onpromotion* Regressors

We engineered new features that were derived from the *onpromotion* column by introducing different levels of lags. We included these new lagged and old features to build a SARIMA model.

```
# Fit SARIMA model with lagged promotions
fit_lagged <- train_sales %>%
  model(
    sarima_promo = ARIMA(sales ~ onpromotion + promo_lag1 + promo_lag2 +
                           promo_lag3 + promo_lag7 + promo_lag14 +
                           holiday_flag + dcoilwtico + 1 +
                           pdq(2,1,1) + PDQ(1,1,1,7))
  )
```

**Evaluation**

| | .model | .type | ME | RMSE | MAE | MPE | MAPE | MASE | RMSSE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | sarima_promo | Test | 69128.48 | 113167.6 | 88069.97 | 8.269816 | 10.55246 | NaN | NaN | 0.6436025 |

**6. Naïve Forecast Model**

We built a simple naïve forecast model that will take the most recent observation to predict future data points. This will serve as a baseline for model comparison.

```
# Naive forecast (using previous day)
naive_forecast <- train_sales %>%
  model(
    NAIVE(sales)
  )
```

**Evaluation**

| | .model | .type | ME | RMSE | MAE | MPE | MAPE | MASE | RMSSE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | NAIVE(sales) | Test | -267959.5 | 294761.2 | 269489.4 | -34.9012 | 35.03735 | NaN | NaN | 0.4647935 |

### 6.1.2 Model Comparison

We used the MAPE evaluation metric to compare all the models built. Our champion model was the SARIMA() with 3 regressors where the predictions were made at a store-cluster level with a MAPE of 6.51%. Our challenger model was a base ARIMA(), forecasting the day-level sales

with a MAPE of 7.18%. Both models significantly outperformed the baseline naïve forecast, which had a MAPE of 35.03%. (Refer to Section 7)

## 6.2 Bayesian Forecasting

In addition to using an ARIMA model, we attempted to forecast sales using the Bayesian method to account for both aleatory and epistemic variance. The team wanted to ideally implement both models to get a more well-rounded idea of our forecast variability and confidence. Unfortunately, our dataset of more than three million rows was too large for proper forecasting on our computers, but we could gain a little insight by looking at the first few thousand rows.

### 6.2.1 Creating Harmonics

After loading the data and undergoing preprocessing, we defined a function to generate harmonic components. These harmonics or implementations of sine and cosine waves help model seasonal patterns within our time-series data components. As the code image below shows, we generated a sine or cosine wave matrix based on our input parameters. The k value is the harmonic number that ranges from 1 to K, t represents the time indices, and P is the period. The period is annual or 365 for yearly seasonality. The matrix output, H, with N rows and K columns, would represent one harmonic wave. Furthermore, we set the number of harmonics equal to 5, so the function will create 5 sine and 5 cosine harmonics for yearly seasonality.

```r
harmonics <- function(SC, K, P, N) {
  if (SC == "S") {
    H <- sapply(1:K, \(k) sin(2 * pi * k * (0:(N - 1)) / P))
  } else {
    H <- sapply(1:K, \(k) cos(2 * pi * k * (0:(N - 1)) / P))
  }
  H %>% matrix(nrow = N, ncol = K)
}

#number of harmonics for yearly seasonality
MK <- 5  # Adjust based on the complexity of seasonality
N <- nrow(df)

#create harmonics for yearly seasonality
MS <- harmonics("S", MK, 365, N)
MC <- harmonics("C", MK, 365, N)
```

### 6.2.2 Stan Model

Before creating a Stan file for our Bayesian analysis, we prepared a list of data to be passed through the model. We set the number of observations equal to the length of the dataset, noted

the target variable (sales), set the time variable ‘t,’ and defined our harmonics that were set in the previous code chunk. Furthermore, the holiday variable was converted into a binary integer variable for more interpretable results, and the oil price was set as an external factor for any modeling dependencies.

```
stan_data <- list(
  N = N,
  d = df$sales,
  t = df$t,
  MK = MK,
  MS = MS,
  MC = MC,
  holiday = as.integer(df$holiday),
  oil_price = df$oil_price
)
```

The Stan file was then created to help incorporate seasonality, holidays, and oil prices as predictors. First, we had the data block, which specifies the inputs needed for the model. The inputs included the number of observations, the number of harmonics, the defined response variable of sales, a matrix of sine harmonics such that each row would correspond to an observation, and a matrix of cosine harmonics to model any cyclical patterns. Second is the parameter block, where real theta represents the mean demand level, the sigma value is the standard deviation of the error term, and phi MS and phi MK represent vector coefficients for sine and cosine harmonics. Next, the model block defines the likelihood function and helps quantify how well model parameters explain the data. For each observation, the predicted mean demand is calculated as the baseline demand plus the sine and cosine harmonics, interacting with the effect of holidays and oil prices. Finally, the generated quantities block generates posterior predictive samples of demand. In other words, this allows for simulated future demand based on the estimated parameters. A vector was defined to store the predicted values, and for each observation, this predicted demand is generated using the normal_rng function. This helps generate random samples from a Gaussian, normal distribution.

```
data {
  int<lower=1> N;           //Number of observations
  int<lower=1> MK;          //Number of harmonics
  vector[N] d;              //Demand data
  matrix[N, MK] MS;         //Sine harmonics
  matrix[N, MK] MC;         //Cosine harmonics
  vector[N] holiday;        //Holiday indicator
  vector[N] oil_price;      //Oil prices
}

parameters {
  real theta;               //Mean demand level
  real<lower=0> sigma;      //Error term
  vector[MK] phiMS;         //Coefficients for sine harmonics
  vector[MK] phiMC;         //Coefficients for cosine harmonics
  real beta_holiday;        //Coefficient for holiday effect
  real beta_oil;            //Coefficient for oil price effect
}

model {
  vector[N] mu;

  for (i in 1:N) {
    mu[i] = theta + dot_product(MS[i], phiMS) + dot_product(MC[i], phiMC) +
            beta_holiday * holiday[i] + beta_oil * oil_price[i];
  }

  d ~ normal(mu, sigma);    //Likelihood
}

generated quantities {
  vector[N] d_out;

  for (i in 1:N) {
    d_out[i] = normal_rng(theta + dot_product(MS[i], phiMS) + dot_product(MC[i], phiMC) +
                          beta_holiday * holiday[i] + beta_oil * oil_price[i], sigma);
  }
}
```

The text above was stored as a Stan file, where these defined inputs were initially iterated 1000 times, and 4 with Markov Chain Monte Carlo chain simulations. Given that our dataset had over 3 million rows, running this many iterations and chains caused our team's computers to crash or be unable to handle such large data components. So, we tried shortening the amount of iterations to 500 with 2 Markov chains. However, since our dataset was simply too large, our computers could still not handle such a large amount of information. Once we sampled the number of rows considered to be 3000, we were able to simulate some generated demand with 1000 iterations and 4 Markov chains.

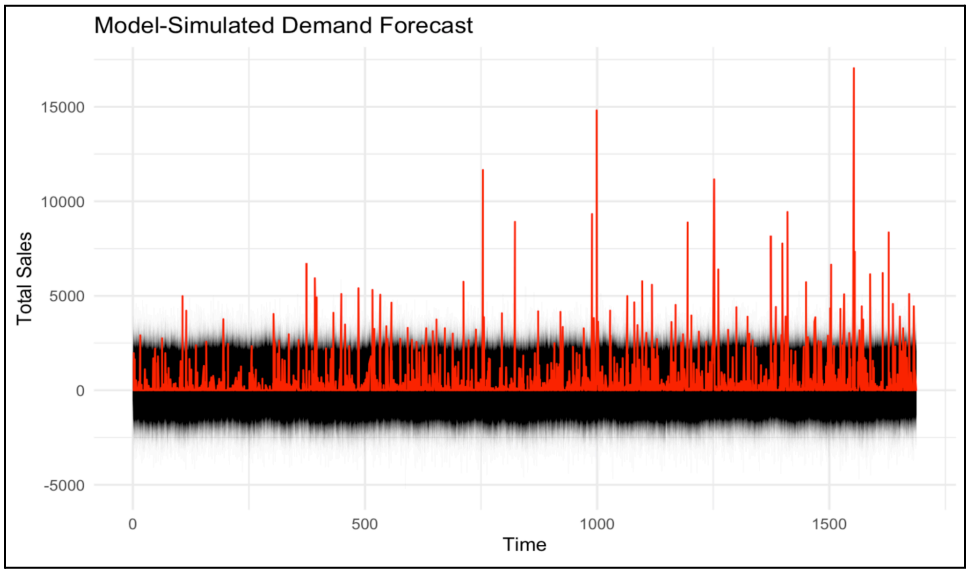
```
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)

#model
fit <- stan(file = "/Users/brindaasuri/Desktop/MSBA/demand_forecasting.st
an",
            data = stan_data,
            iter = 1000, chains = 4, verbose = TRUE)

# Print summary of the parameters
print(fit, pars = c("theta", "sigma", "beta_holiday", "beta_oil",
"phiMS", "phiMC"))
```

The model below shows the visual comparison between model-simulated demand forecasts and actual demand data. The x-axis represents the time period of the observations and helps track the

progression of demand. The y-axis represents the level of demand or sales. The black lines in the model are meant to represent the range of forecast uncertainty, where narrower regions suggest higher prediction confidence and wider regions are more uncertain. From this element alone, we may see that this model is not a great representation. The black lines extending below zero indicate that sales could drop below zero, which does not make sense. The red line, on the other hand, shows the actual demand over time. This line shows high variability and frequent spikes. Although we only took a subset of the actual sample, we can see from this graph how fluctuating actual sales demand can be. Periods of stability and sharp increase may be due to external factors such as promotions or holidays.




Although this attempt at Bayesian forecasting could not be accurately completed due to computation and time constraints, there are ways to avoid this in the future and obtain more interpretable results. For example, we can apply log transformations to stabilize variance if demand experiences random spikes or scale numerical features to standard ranges to help stabilize the model's performance. Furthermore, to handle the vast number of rows, we could conduct analysis on large random samples of the dataset to get more holistic results. Given that this model does not yield interpretable results, the team will base our predictions on the ARIMA models.

## 7. Results

Our champion model was the SARIMA() with 3 regressors (promotions, holidays, and oil prices) where the predictions were made at a store-cluster level with a MAPE of 6.51% and MAE of 1.58%. Our challenger model was a base ARIMA(), forecasting the day-level sales with a MAPE of 7.18% and an MAE of -2.39%. Both models significantly outperformed the baseline naïve forecast, with a MAPE of 35.03% and an MAE of -34.90%. This high accuracy produced by SARIMA indicates the value of aggregating data across similar stores to capture consistent sales patterns while incorporating critical external factors that significantly influence demand. Seasonal patterns, such as weekly and holiday trends, were effectively captured by SARIMA, outperforming simpler ARIMA models and including lagged promotions, revealing the lasting impact of past promotional efforts on future sales. These insights emphasize the importance of leveraging external factors and aggregated data for improved inventory management, optimized promotions, and better allocation of resources across stores.

| Model | Level | MPE (%) | MAPE (%) |
|---|---|---|---|
| ARIMA() | Day | -2.39 | 7.18 |
| ARIMA() + 3 Regressors | Day | -0.86 | 9.28 |
| SARIMA() + 3 Regressors | Day | 11.95 | 13.36 |
| SARIMA() + 3 Regressors | Category-Day | -2.30 | 9.71 |
| ARIMA() | StoreCluster - Day | 12.95 | 13.55 |
| ARIMA() + 3 Regressors | StoreCluster - Day | 5.48 | 7.84 |
| SARIMA() + 3 Regressors | StoreCluster - Day | 1.58 | 6.51 |
| ARIMA() + Lagged Onpromo | Day | 8.26 | 10.55 |
| Naive | Day | -34.90 | 35.03 |

## 8. Insights & Future Scope

- Our analysis showed that forecast accuracy for retail sales data can be greatly increased by considering the proper level of aggregation and adding both seasonal components and external regressors.
- While the initial analysis began with approximately 10 features, many proved unsuitable for the final models. Future iterations could incorporate new engineered features, including:
    - Demographic indicators
    - Regional characteristics
- Implementation of forecasting at various hierarchical levels could be beneficial. We would experiment with forecasting at different levels, like Store-Type Leve,l where we run forecasts for each store type and aggregate in the last step.
- We would like to experiment with hybrid approaches, such as combining results from the store-cluster level and lagged feature models. Creating ensemble approaches that leverage multiple model strengths could improve our forecasts.

These enhancements could improve the current champion model's MAPE of 6.51% and the challenger model's MAPE of 7.18%.

## 9. References


- Dataset: For access to the dataset, please visit the Kaggle - Store Sales.
- https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322
- https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8
- https://blogs.oracle.com/ai-and-datascience/post/introduction-to-forecasting-with-arima-in-r
- https://people.duke.edu/~rnau/411arim.htm