

IBM6250 Group Project - Coffee Vending

Min Gong

Jarrod Griffin

Ceren Unal

Eunice Won

2025-05-11

Table of contents

1	Introduction	2
1.1	This report:	2
2	Executive Summary with Actionable Recommendations	2
2.0.1	Executive Summary	2
2.0.2	Actionable Recommendations	3
3	Data and Assumptions	3
3.1	Raw Transaction Data	3
3.2	Products and Ingredients	5
3.2.1	Unique Products	5
3.2.2	Ingredients	5
3.2.3	Ingredient Logic	5
3.3	Combining Transaction Data and Recipes	6
3.4	Converting to Weekly Series	8
3.5	Final Data for use in Analysis	9
4	Exploratory Insights (EDA)	11
4.1	Total Revenue Series	11
4.1.1	Data Description & Exploratory Summary	17
4.2	Coffee Type Series	17
4.2.1	Data Description & Exploratory Summary	23
4.3	Ingredient Series	23
4.3.1	Data Description & Exploratory Summary	28

5	Modeling Method Choice and Diagnostics	28
5.1	Ingredients Model	28
5.1.1	Model Evaluation and Diagnostics	29
5.2	Coffee Type Model	31
5.2.1	With differencing (Model forecasting change)	31
5.2.2	No differencing (Model forecasting unit sales)	31
5.3	Revenue Model	31
6	Generate Forecasts Results	32
6.1	Ingredients Forecast	32
6.2	Coffee Type Forecast	32
6.2.1	With differencing (Model forecasting change)	32
6.2.2	Americano with Milk - Forecast (2 Difference)	33
6.2.3	No differencing (Model forecasting unit sales)	34
6.3	Revenue Forecast	35
6.3.1	Conclusion	36
7	Appendix	38
7.0.1	Machine 1	38

1 Introduction

Effective inventory control for coffee-vending machines hinges on anticipating weekly ingredient consumption while avoiding costly spoilage. We forecast demand using historical sales from two machines, delivering eight-week projections that guide stock levels and reorder cadence.

1.1 This report:

1. Imports & cleans transaction data from a coffee vending machine.
2. Explores key demand drivers.
3. Models weekly sales with Seasonal ARIMA.
4. Delivers eight-week forecasts and stocking recommendations.

2 Executive Summary with Actionable Recommendations

2.0.1 Executive Summary

This report delivers an integrated forecasting solution for a coffee vending machine using transaction data from March 2024 to March 2025. We modeled the trends in ingredient usage, coffee type unit sales, and overall revenue to generate eight-week forecasts that guide more precise inventory management and restocking decisions.

- **Ingredient Usage:** From the EDA, we can see that milk is the most volatile and heavily used ingredient, peaking at over 16,000mL in some weeks due to high demand for milk-based drinks. Coffee grounds remain stable, while the usage of chocolate and sugar show seasonal spikes. The seasonal spikes may be due to the cold weather and consumers preferring hot drinks, like hot chocolate and cocoa.

Seasonal ARIMA models effectively captured ingredient usage patterns, with Ljung-Box tests confirming white-noise residuals. All ingredient seasonal ARIMA models have uncorrelated residuals and is reliable for forecasting.

- **Drink Sales Patterns:** Americano with Milk and Latte was seen as the most popular and consistently sold drinks. Cortado and Espresso was seen to be less popular in sales and could be reconsidered for prioritization.
- **Revenue Trends:** Weekly revenue fluctuates and showed peaks that resembled “three hills”. The STL decomposition showed a trend, but it did not show significant seasonality. The ETS model projects stable short-term growth.

2.0.2 Actionable Recommendations

- **Focus on High-Selling Drinks:** Focus inventory on Americano with Milk and Latte ingredients to meet ongoing high sales volume. Make sure that there is an uninterrupted ingredient supply for the high-selling drinks.
- **Prioritize Milk Inventory:** Proactively stock milk with a buffer margin, especially during high-consumption weeks to avoid any out-of-stock situations.
- **Deprioritize Low-Selling Drinks:** Reevaluate stocking strategy for lower-selling drinks to minimize unnecessary ingredient usage. Can consider rotating low-selling drinks to optimize machine space and ingredient turnover.
- **Prepare for Seasonal Demand Spikes:** Slightly increase sugar and chocolate inventory in colder months or during promotional periods when the demand for hot drinks increase.
- **Automate Coffee Reorders:** Use consistent SARIMA forecasts to consider setting up auto-reordering for coffee grounds and to avoid overstocking.
- **Update Forecast Models Regularly:** Update forecast models regularly to see the changes in trend and seasonality and to take that into effect. Update SARIMA and ETS models quarterly to reflect changes in customer behavior or seasonal effects.

3 Data and Assumptions

Kaggle data is from two vending machines. Below we will import the two datasets and combine them.

3.1 Raw Transaction Data

Transaction data was taken from the following Kaggle link:

<https://www.kaggle.com/datasets/ihelon/coffee-sales>

```

machine1 <- read_csv("data/index_1.csv") %>%
  mutate(machine_id = "machine1")

sales <- machine1 |>
  mutate(date = as_date(date),
         datetime = as_datetime(datetime),
         coffee_name=toupper(coffee_name))

skim(sales)

```

Table 1: Data summary

Name	sales
Number of rows	3636
Number of columns	7
Column type frequency:	
character	4
Date	1
numeric	1
POSIXct	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
cash_type	0	1.00	4	4	0	2	0
card	89	0.98	19	19	0	1316	0
coffee_name	0	1.00	5	19	0	8	0
machine_id	0	1.00	8	8	0	1	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date	0	1	2024-03-01	2025-03-23	2024-10-06	381

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
money	0	1	31.75	4.92	18.12	27.92	32.82	35.76	40	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
datetime	0	1	2024-03-01 10:15:50	2025-03-23 18:11:38	2024-10-07 02:55:12	3636

```
unique(sales$coffee_name)
```

```
[1] "LATTE"           "HOT CHOCOLATE"   "AMERICANO"
[4] "AMERICANO WITH MILK" "COCOA"           "CORTADO"
[7] "ESPRESSO"        "CAPPUCCINO"
```

3.2 Products and Ingredients

In the dataset, only product names are given. In order to more accurately predict what ingredients are needed and when, we must decompose the product into its ingredients. See below for the assumptions made for each of the 8 unique products.

3.2.1 Unique Products

```
unique(sales$coffee_name)%>%sort()
```

```
[1] "AMERICANO"           "AMERICANO WITH MILK" "CAPPUCCINO"
[4] "COCOA"               "CORTADO"             "ESPRESSO"
[7] "HOT CHOCOLATE"      "LATTE"
```

3.2.2 Ingredients

```
recipes <- tribble(
  ~coffee_name,      ~coffeeG, ~milkML, ~chocolateG, ~sugarG, ~vanillaML,
  "AMERICANO",        18,      0,      0,      0,      0,
  "AMERICANO WITH MILK", 18,     60,      0,      0,      0,
  "CAPPUCCINO",       18,    100,      0,      0,      0,
  "COCOA",            0,    240,    22,    15,      0,
  "CORTADO",          18,     60,      0,      0,      0,
  "ESPRESSO",         18,      0,      0,      0,      0,
  "HOT CHOCOLATE",     0,    240,    30,    20,      0,
  "LATTE",            18,    240,      0,      0,    10
)
```

3.2.3 Ingredient Logic

Drink	Ingredient-logic rationale
Espresso	Straight double shot: 18 g ground coffee, no additives (Specialty Coffee Association 2018).
Americano	Same 18 g espresso diluted with $4 \times$ its volume of hot water; nothing else required (Specialty Coffee Association 2018).

Drink	Ingredient-logic rationale
Americano with Milk	Americano softened with 60 ml steamed milk – enough to mellow bitterness without turning it into a latte (Cordell 2024).
Cappuccino	Classic 1 : 1 : 1 build – espresso, 60 ml steamed milk, equal micro-foam – fills a 150–180 ml cup (Raffii 2024).
Cortado	Spanish “cut” drink: equal parts double espresso and 60 ml steamed milk (Wine Editors 2025).
Latte	U.S. latte stretches the shot with 240 ml milk (1 : 4–5 ratio); vanilla version adds 10 ml syrup (2 pumps) (Coffee Bros. 2024; Page 2025).
Cocoa	Non-coffee mix: 240 ml milk + 22 g cocoa powder + 15 g sugar – standard stovetop proportions (Hersheyland Test Kitchen 2025).
Hot Chocolate	Richer café blend: same milk but 30 g cocoa and 20 g sugar for modern sweetness level (Hersheyland Test Kitchen 2025).

3.3 Combining Transaction Data and Recipes

Below we will join the two tables on the coffee name, which will add ingredients to all rows in the transaction data. Explore the data we will use in our analysis below:

```
sales_ingredients <- sales |>
  left_join(recipes, by = "coffee_name") |>
  replace_na(list(
    coffeeG = 0, milkML = 0, chocolateG = 0, sugarG = 0, vanillaML = 0
  ))
```

Copy

CSV

Search:

	date	datetime	cash_type	card	money	coffee_name	machine_id	coffeeG	milkML	chocolateG	sugar
1	2024-03-01	2024-03-01T10:15:50Z	card	ANON-0000-0000-0001	38.7	LATTE	machine1	18	240	0	
2	2024-03-01	2024-03-01T12:19:22Z	card	ANON-0000-0000-0002	38.7	HOT CHOCOLATE	machine1	0	240	30	
3	2024-03-01	2024-03-01T12:20:18Z	card	ANON-0000-0000-0002	38.7	HOT CHOCOLATE	machine1	0	240	30	
4	2024-03-01	2024-03-01T13:46:33Z	card	ANON-0000-0000-0003	28.9	AMERICANO	machine1	18	0	0	
5	2024-03-01	2024-03-01T13:48:14Z	card	ANON-0000-0000-0004	38.7	LATTE	machine1	18	240	0	

Showing 1 to 5 of 3,636 entries

Previous

1

2

3

4

5

...

728

Next

3.4 Converting to Weekly Series

We aggregate to a weekly time series because the business decisions we are informing, like re-ordering coffee, milk, chocolate, etc, are made on a weekly cadence. Collapsing daily transactions into weeks smooths out erratic, day-to-day swings leaving a cleaner signal that aligns directly with the quantity we must predict.

We will also convert to a time series type object and verify it has no gaps in the series. If we see FALSE from `.gaps`, then we have no gaps.

```
weekly_sales <- sales_ingredients |>
  mutate(week = lubridate::floor_date(date, unit = "week")) |>
  group_by(week) |>
  summarise(across(coffeeG:vanillaML, sum, na.rm = TRUE),
            sales_n = n()) |>
  ungroup()

weekly_sales <- weekly_sales|>
  as_tsibble(index = week)

has_gaps(weekly_sales)
```

```
# A tibble: 1 x 1
  .gaps
  <lgl>
1 FALSE
```


3.5 Final Data for use in Analysis

Copy

CSV

Search:

	week	coffeeG	milkML	chocolateG	sugarG	vanillaML	sales_n
1	2024-02-25	252	2100	112	75	30	18
2	2024-03-03	792	7440	344	230	130	56
3	2024-03-10	846	4580	150	100	90	52
4	2024-03-17	612	4220	104	70	80	38
5	2024-03-24	666	5640	82	55	150	40

Showing 1 to 5 of 57 entries

Previous

1

2

3

4

5

...

12

Next

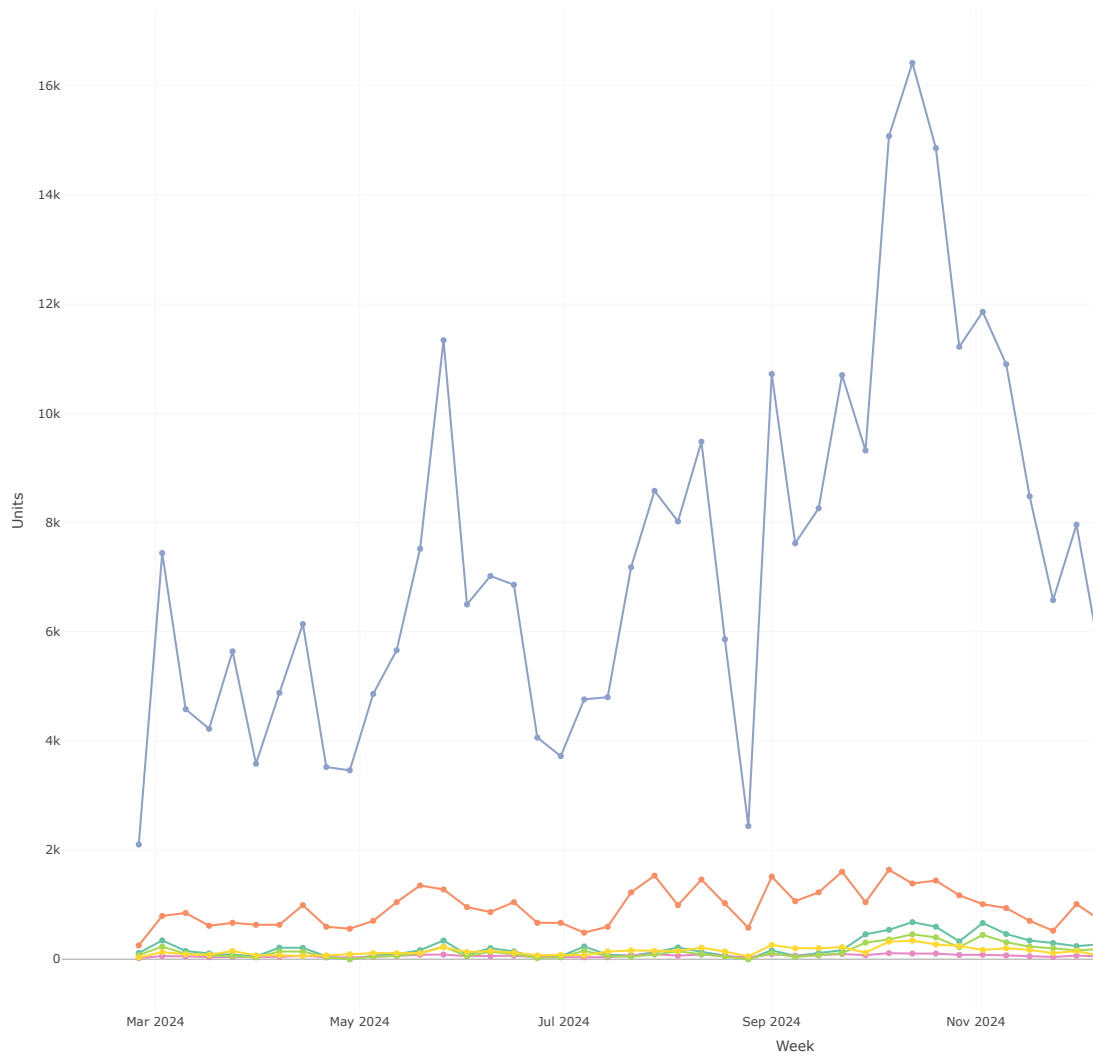


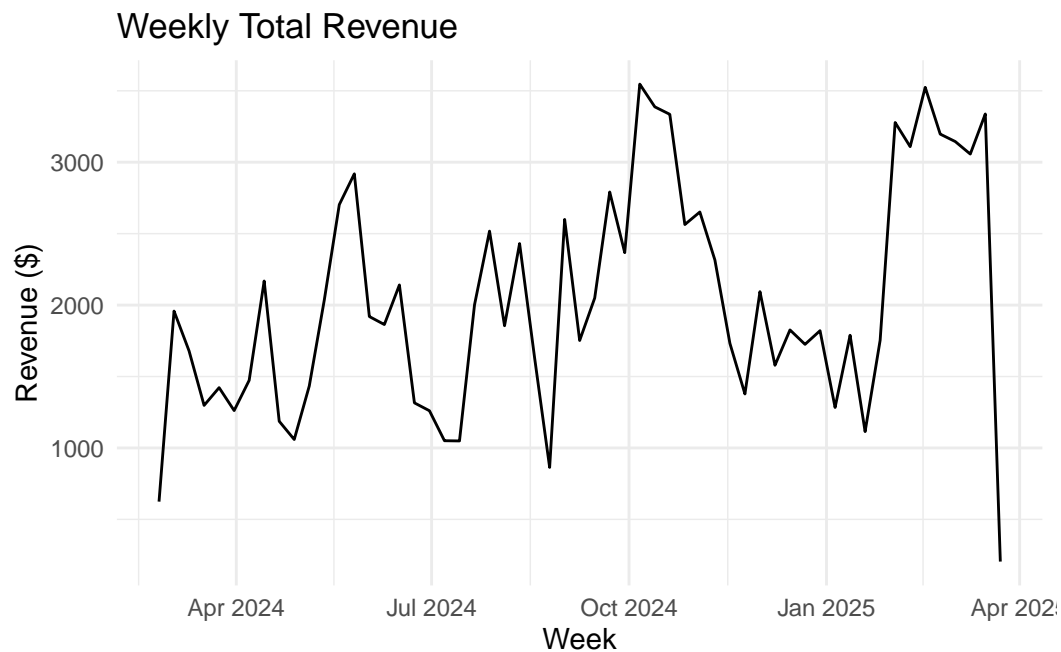
Figure 1: Weekly ingredient demand vs. cups sold

4 Exploratory Insights (EDA)

4.1 Total Revenue Series

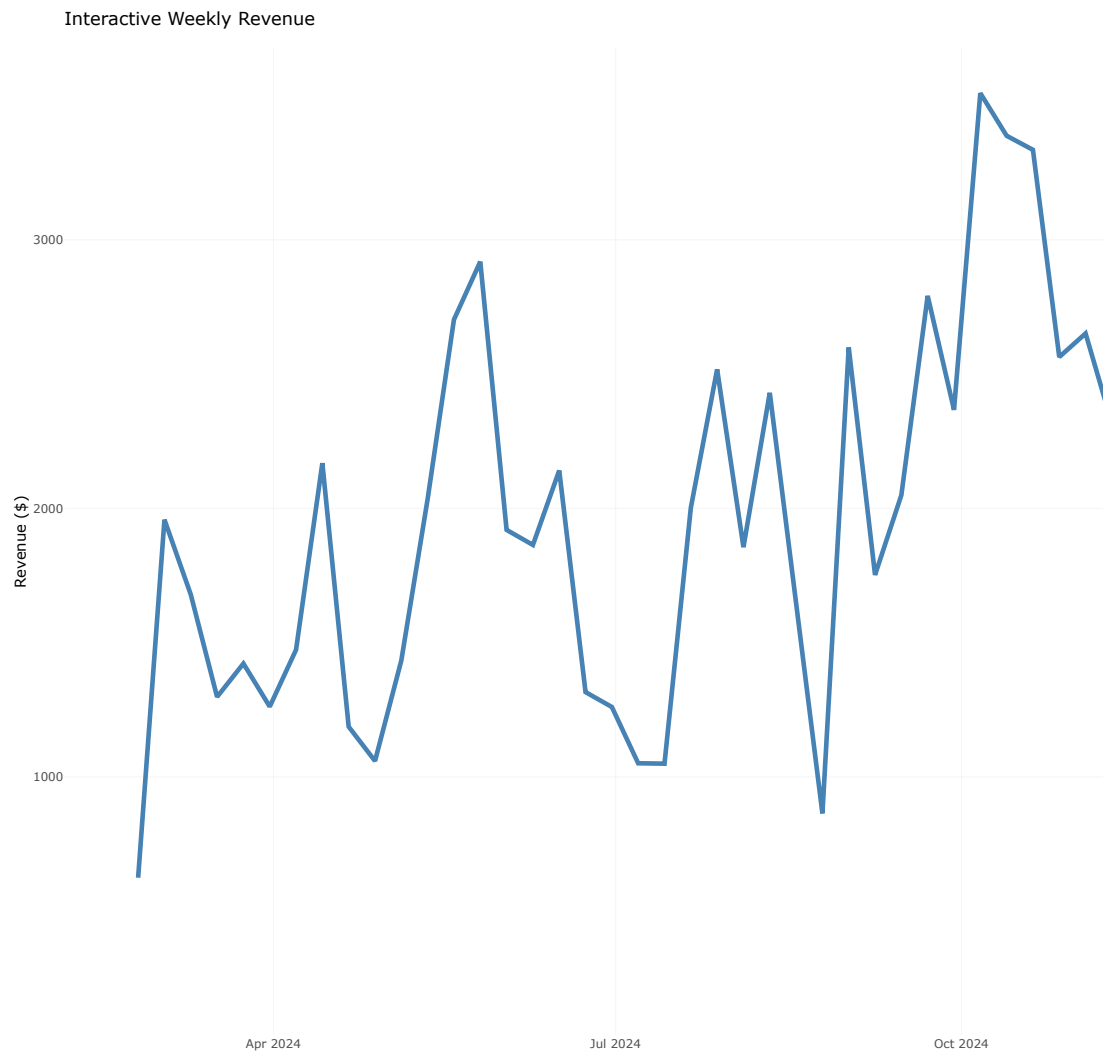
```
#weekly revenue tsibble
weekly_revenue <- sales %>%
  mutate(week = floor_date(date, unit = "week")) %>%
  group_by(week) %>%
  summarise(total_revenue = sum(money, na.rm = TRUE), .groups = "drop") %>%
  as_tsibble(index = week)
```

```
# weekly total revenue time series plot
autoplot(weekly_revenue, total_revenue) +
  labs(title = "Weekly Total Revenue",
       y = "Revenue ($)",
       x = "Week") +
  theme_minimal()
```



```
#Interactive weekly revenue
weekly_revenue %>%
  ggplot(aes(x = week, y = total_revenue)) +
  geom_line(color = "steelblue", size = 1.2) +
  labs(title = "Interactive Weekly Revenue",
       y = "Revenue ($)", x = NULL) +
  theme_minimal() -> p

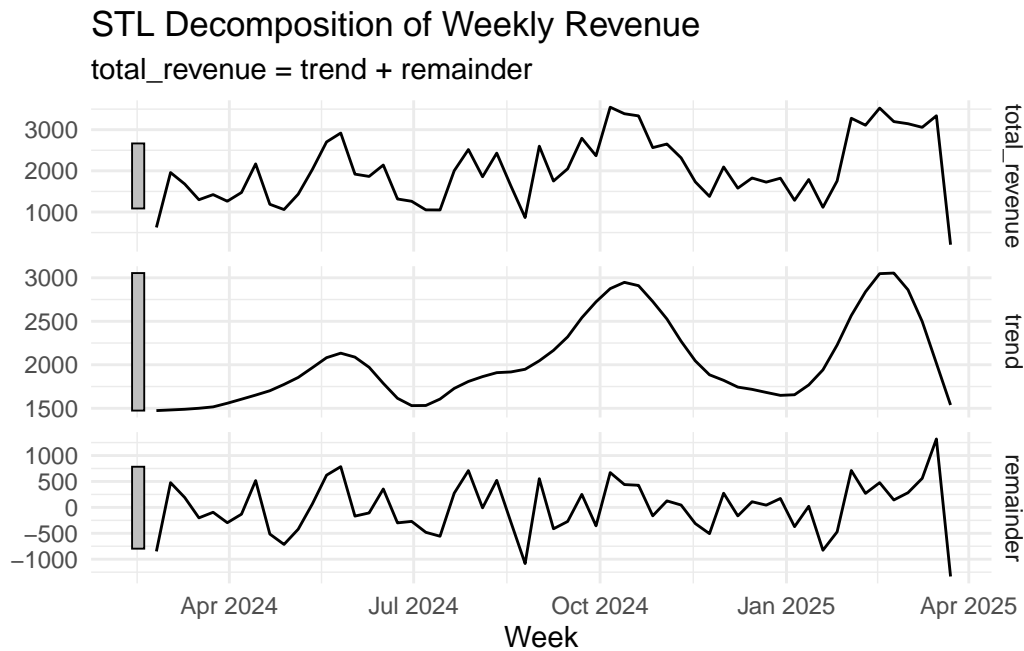
ggplotly(p)
```



```
#STL Decomposition  
decomposed_revenue <- weekly_revenue %>%
```

```
model(STL(total_revenue ~ season() + trend(), robust = TRUE)) %>%
  components()
```

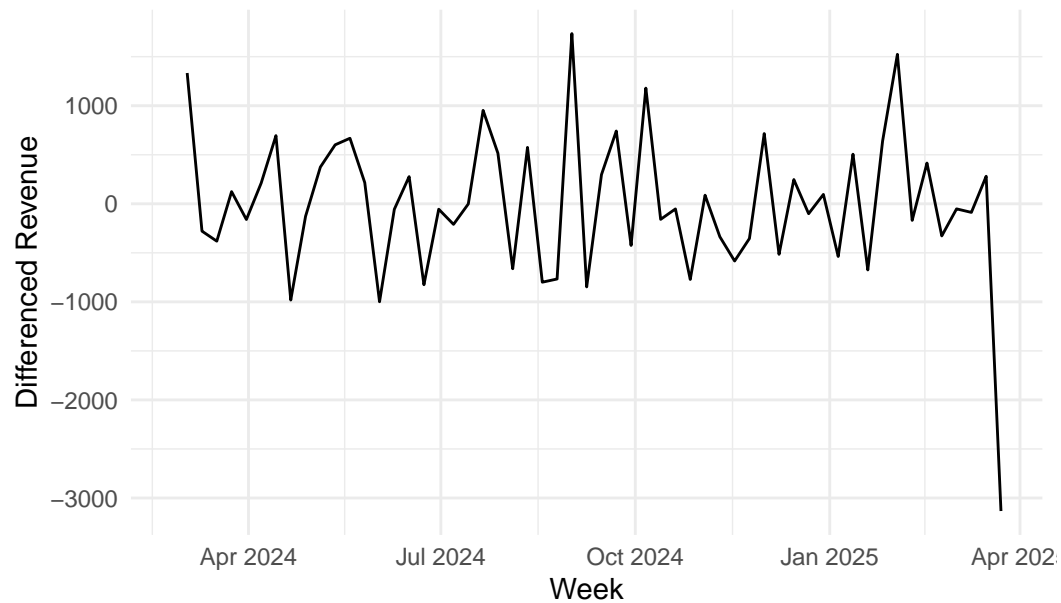
```
# plot component
autoplot(decomposed_revenue) +
  labs(title = "STL Decomposition of Weekly Revenue",
       x = "Week", y = NULL) +
  theme_minimal()
```



```
#Differencing
diff_revenue <- weekly_revenue %>%
  mutate(diff_revenue = difference(total_revenue, lag = 1))

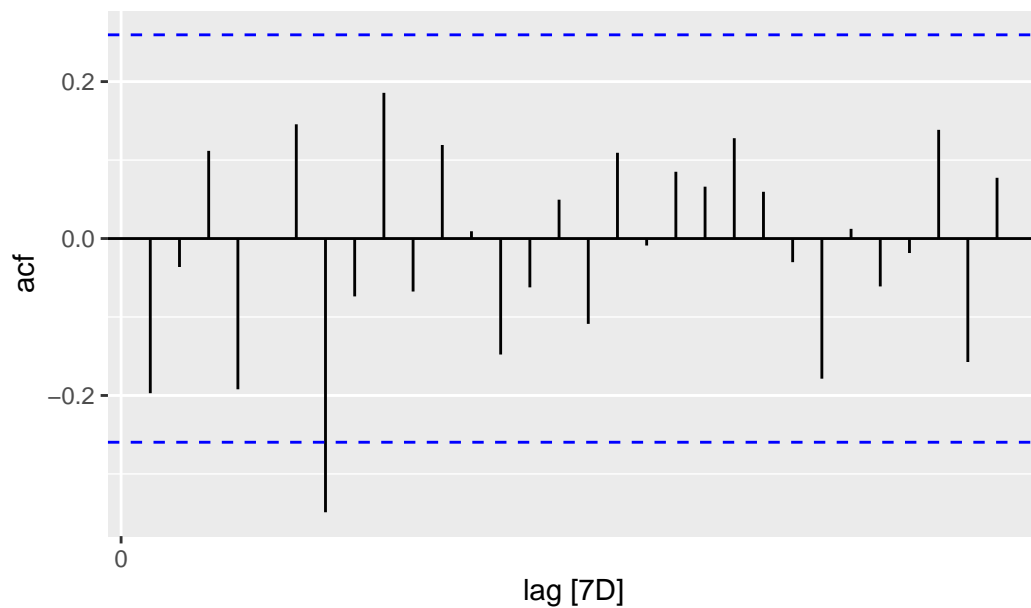
autoplot(diff_revenue, diff_revenue) +
  labs(title = "First Difference of Weekly Revenue",
       y = "Differenced Revenue",
       x = "Week") +
  theme_minimal()
```

First Difference of Weekly Revenue



```
# ACF
diff_revenue %>%
  ACF(diff_revenue, lag_max = 30) %>%
  autoplot() +
  labs(title = "ACF of Differenced Revenue")
```

ACF of Differenced Revenue



```
# Summary stats
summary_stats <- weekly_revenue %>%
  summarise(
```

```

    mean_revenue = mean(total_revenue, na.rm = TRUE),
    median_revenue = median(total_revenue, na.rm = TRUE),
    max_revenue = max(total_revenue, na.rm = TRUE),
    min_revenue = min(total_revenue, na.rm = TRUE)
  )

# Highest Weekly Sales
top_week <- weekly_revenue %>%
  filter(total_revenue == max(total_revenue, na.rm = TRUE))
top_week

```

```

# A tsibble: 1 x 2 [7D]
  week      total_revenue
  <date>      <dbl>
1 2024-10-06      3547.

```

```

datatable(summary_stats, caption = "Weekly Revenue Summary Stats")

```

Show entries

Search:

Weekly Revenue Summary Stats

	week	mean_revenue	median_revenue	max_revenue	min_revenue
1	2024-02-25	624.3999999999999	624.3999999999999	624.3999999999999	624.3999999999999
2	2024-03-03	1958.0000000000002	1958.0000000000002	1958.0000000000002	1958.0000000000002
3	2024-03-10	1679.0000000000001	1679.0000000000001	1679.0000000000001	1679.0000000000001
4	2024-03-17	1298	1298	1298	1298
5	2024-03-24	1422.1	1422.1	1422.1	1422.1
6	2024-03-31	1261.7	1261.7	1261.7	1261.7
7	2024-04-07	1473.2000000000001	1473.2000000000001	1473.2000000000001	1473.2000000000001
8	2024-04-14	2168.5	2168.5	2168.5	2168.5
9	2024-04-21	1186.84	1186.84	1186.84	1186.84
10	2024-04-28	1059.18	1059.18	1059.18	1059.18

Showing 1 to 10 of 57 entries

Previous

2

3

4

5

6

Next

4.1.1 Data Description & Exploratory Summary

- Dataset Overview

This dataset is derived from the raw sales table and represents the total weekly revenue, aggregated across all transactions from coffee vending machines, covering the period from March 2024 to March 2025, with data on 8 types of coffee.

- EDA Findings

The weekly revenue time series is complete with no missing weeks. The revenue exhibits a distinct trend, characterized by three major peaks, resembling a “three hills” pattern. STL decomposition confirmed a clear trend but did not detect significant seasonality. After first-order differencing, the series became stationary, with the ACF indicating white noise. The highest revenue occurred in the week of October 6, 2024, amounting to \$3,546, while the lowest revenue was recorded in the week of March 23, 2025, at \$204.76. The observed fluctuation pattern suggests that external drivers, such as promotions or demand cycles, may play a role.

- Assumptions

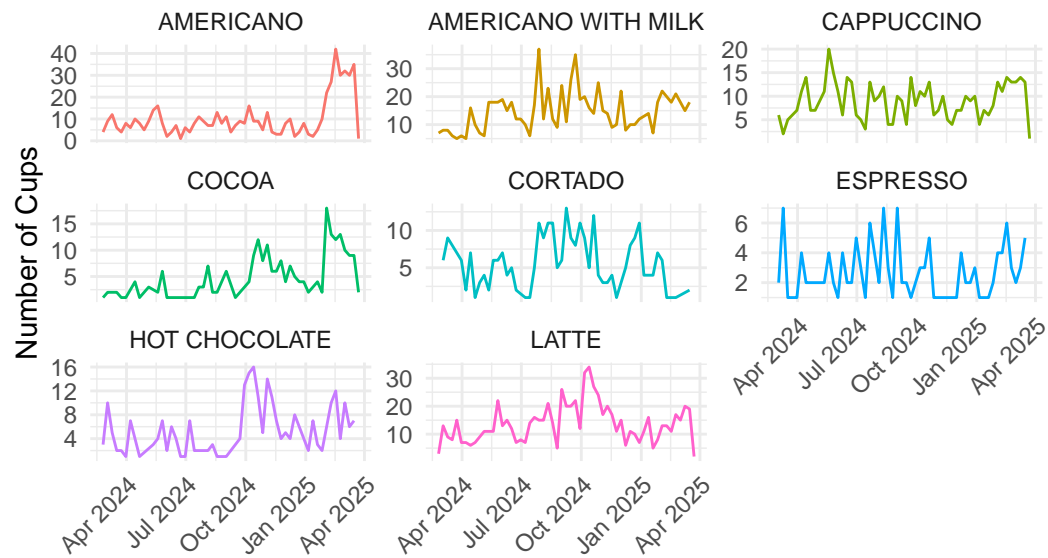
Revenue can be modeled primarily as a trend-driven process, with the data-generating process in the next 8 weeks expected to mirror that of the recent 8 weeks. Weekly sales are projected to grow in the upcoming week, continuing the observed trend.

4.2 Coffee Type Series

```
#sales of each coffee type
weekly_coffee_volume <- sales %>%
  mutate(week = floor_date(date, unit = "week")) %>%
  group_by(week, coffee_name) %>%
  summarise(
    total_volume = n(),
    .groups = "drop"
  ) %>%
  as_tsibble(index = week, key = coffee_name)
```

```
# Weekly cups sold by coffee type
weekly_coffee_volume %>%
  autoplot(total_volume) +
  facet_wrap(~coffee_name, scales = "free_y") +
  labs(title = "Weekly Cups Sold by Coffee Type",
       y = "Number of Cups", x = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```

Weekly Cups Sold by Coffee Type

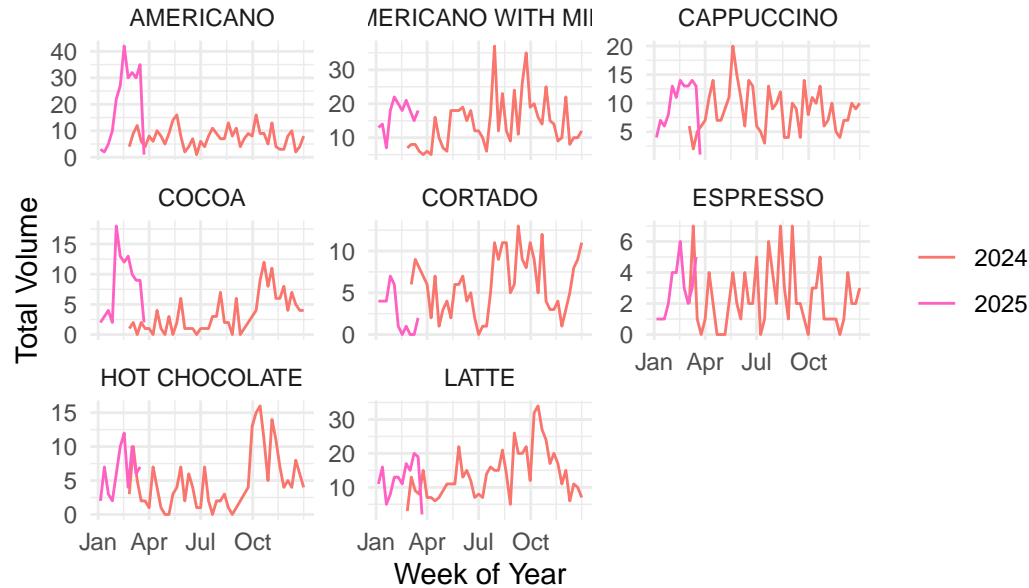


```
# Check NA
missing_data <- weekly_coffee_volume %>%
  group_by(coffee_name) %>%
  summarise(missing_weeks = sum(is.na(total_volume)))

#fill gap
weekly_volume_filled <- weekly_coffee_volume %>%
  fill_gaps(total_volume = 0)

#gg_season
weekly_volume_filled%>%
  gg_season(total_volume) +
  facet_wrap(~ coffee_name, scales = "free_y") +
  labs(title = "Seasonal Plot of Weekly Coffee Volume",
       x = "Week of Year", y = "Total Volume") +
  theme_minimal()
```

Seasonal Plot of Weekly Coffee Volume

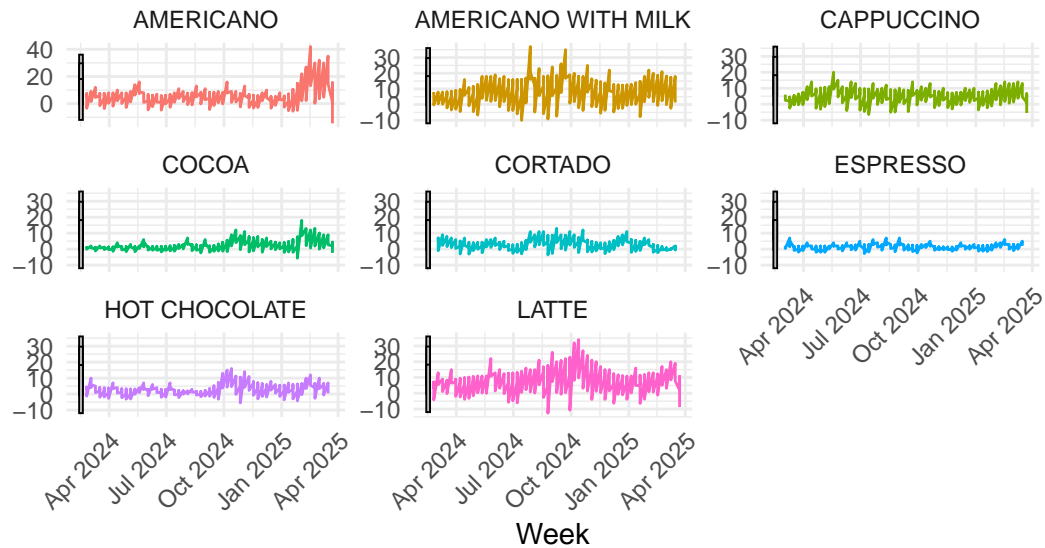


```
#decomposition
decomposed_volume <- weekly_volume_filled %>%
  model(
    STL(total_volume ~ season(window = 52) + trend(window = 7), robust = TRUE)
  ) %>%
  components()

autoplot(decomposed_volume) +
  facet_wrap(~coffee_name, scales = "free_y") +
  labs(title = "STL Decomposition of Weekly Volume by Coffee Type",
       x = "Week", y = NULL) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```

STL Decomposition of Weekly Volume by Coffee Type

total_volume = trend + remainder



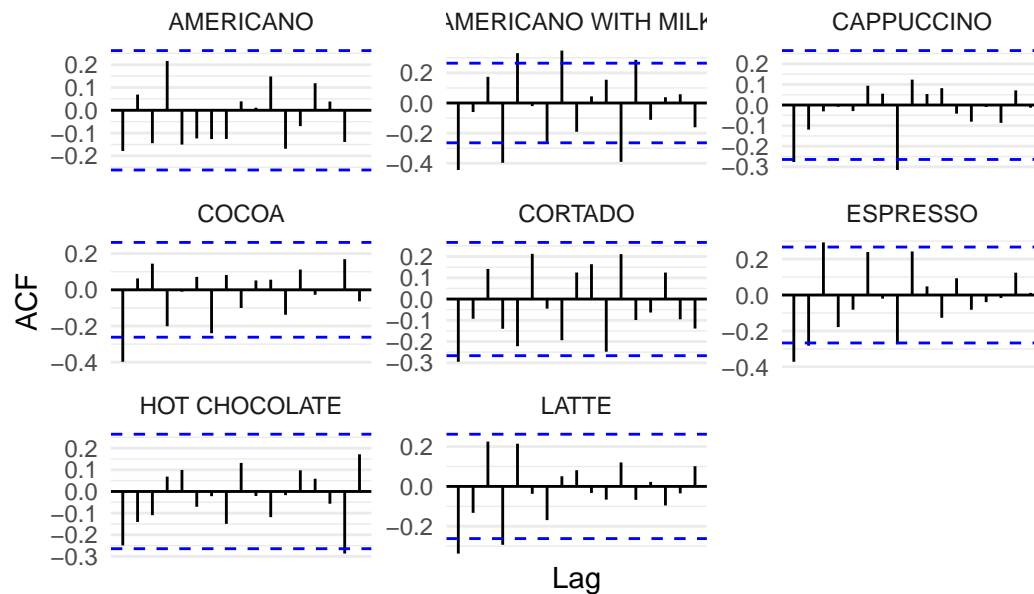
```
#Differencing for all coffee type
weekly_volume_diff <- weekly_volume_filled %>%
  group_by(coffee_name) %>%
  mutate(diff_volume = difference(total_volume)) %>%
  ungroup()
```

```
#tsibble
weekly_volume_diff_tsibble <- weekly_volume_diff %>%
  filter(!is.na(diff_volume)) %>%
  as_tsibble(index = week, key = coffee_name)
```

```
# ACF
acf_diff_data <- weekly_volume_diff_tsibble %>%
  ACF(diff_volume)
```

```
#facet ACF
acf_diff_data %>%
  autoplot() +
  facet_wrap(~ coffee_name, scales = "free_y") +
  labs(title = "ACF After Differencing (by Coffee Type)",
       x = "Lag", y = "ACF") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```

ACF After Differencing (by Coffee Type)



```
coffee_summary <- weekly_coffee_volume %>%
  group_by(coffee_name) %>%
  summarise(
    total_cups = sum(total_volume, na.rm = TRUE),
    avg_weekly_cups = mean(total_volume, na.rm = TRUE),
    max_weekly_cups = max(total_volume, na.rm = TRUE),
    week_of_max_sales = week[which.max(total_volume)],
    min_weekly_cups = min(total_volume, na.rm = TRUE),
    week_of_min_sales = week[which.min(total_volume)],
    .groups = "drop"
  ) %>%
  mutate(
    percent_of_total = total_cups / sum(total_cups) * 100
  ) %>%
  arrange(desc(total_cups))

datatable(coffee_summary,
  caption = "Coffee Type Weekly Sales Summary",
  options = list(pageLength = 8, scrollX = TRUE),
  rownames = FALSE) %>%
  formatRound(columns = c("avg_weekly_cups", "percent_of_total"), digits = 2)
```

Show

8

 entries

Search:

coffee_name		week	total_cups	avg_weekly_cups
AMERICANO	2025-02-16	42	42.00	42 20%
AMERICANO WITH MILK	2024-07-28	37	37.00	37 20%
AMERICANO	2025-03-16	35	35.00	35 20%
AMERICANO WITH MILK	2024-09-22	35	35.00	35 20%
LATTE	2024-10-13	34	34.00	34 20%
AMERICANO	2025-03-02	32	32.00	32 20%
LATTE	2024-10-06	32	32.00	32 20%
AMERICANO	2025-02-23	30	30.00	30 20%

Showing 1 to 8 of 427 entries

4.2.1 Data Description & Exploratory Summary

- Dataset Overview

This dataset represents the **weekly sales volume** (cups sold) for 8 different coffee types from March 2024 to March 2025.

- EDA Findings

The dataset is mostly complete, with only a few missing values for certain coffee types in specific weeks. Among the weeks with over 30 cups sold, Americano appeared 5 times, Americano with Milk 2 times, and Latte 2 times. While `gg_season()` plots suggest that each coffee type exhibits weak seasonal patterns, the short time span of the data (approximately one year) was not enough for STL decomposition to confidently detect seasonality.

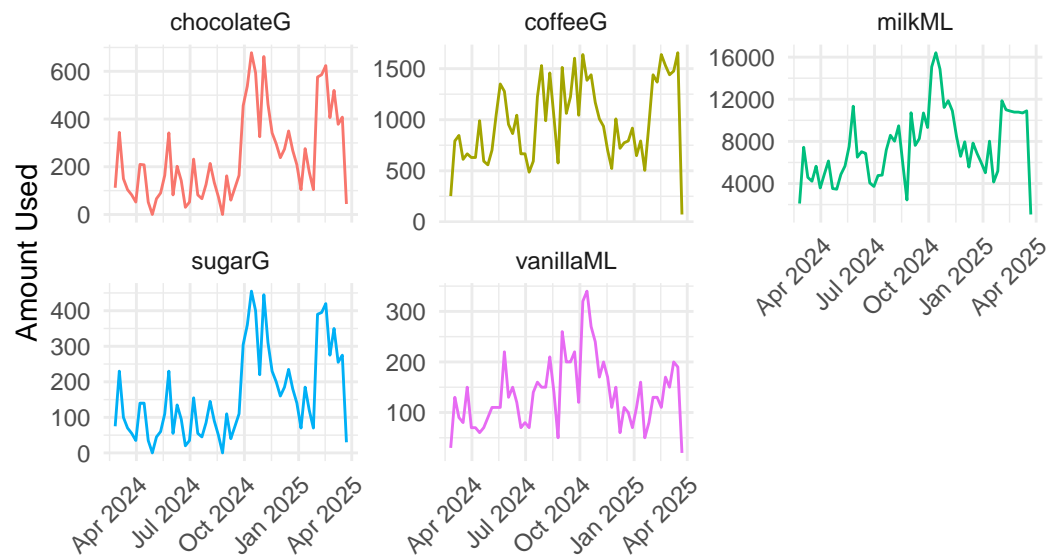
- Assumptions

Americano with Milk is expected to remain the top-selling coffee over the next 8 weeks, continuing its strong performance observed historically. However, customer preferences may shift across coffee types due to changes in weather. Any growth patterns are likely to be gradual rather than abrupt, with no coffee type expected to suddenly double or halve in volume.

4.3 Ingredient Series

```
# Weekly Ingredient Usage visualization
weekly_long %>%
  filter(metric != "sales_n") %>%
  autoplot(value) +
  facet_wrap(~metric, scales = "free_y") +
  labs(title = "Weekly Ingredient Usage",
       y = "Amount Used",
       x = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```

Weekly Ingredient Usage

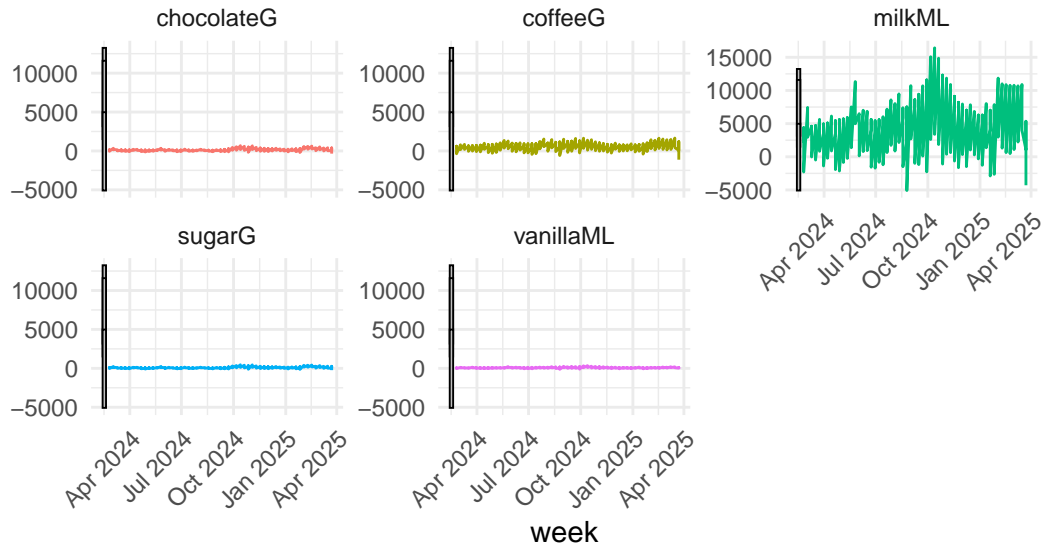


```
#STL decomposition
decomp_recipe <- weekly_long %>%
  filter(metric != "sales_n") %>%
  model(
    STL(value ~ trend(window = 7), robust = TRUE)
  ) %>%
  components()

autoplot(decomp_recipe) +
  facet_wrap(~metric, scales = "free_y") +
  labs(title = "STL Decomposition of Ingredient Usage") +
  theme_minimal() +
  theme(legend.position = "none")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

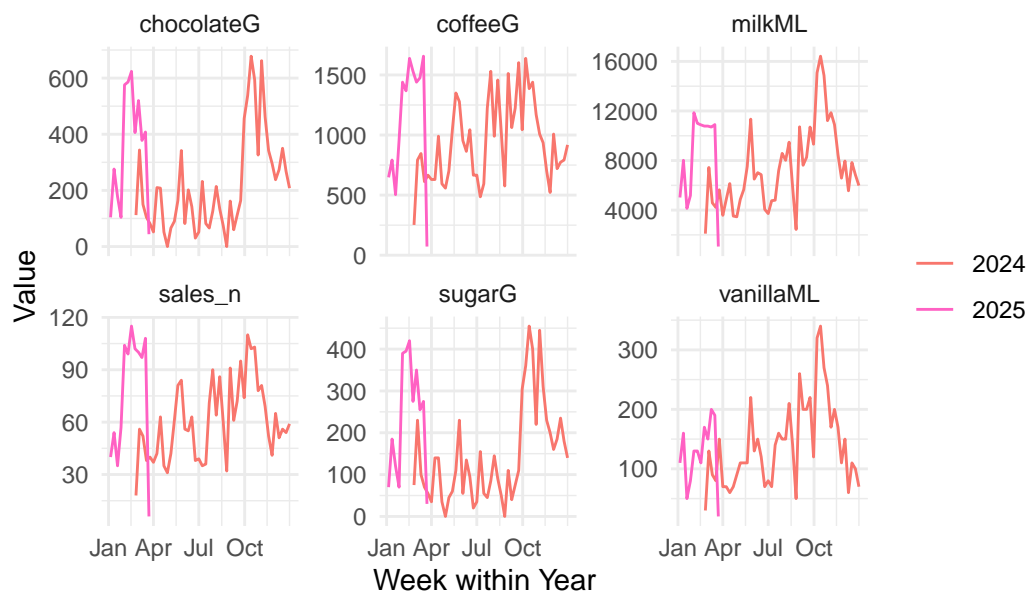

STL Decomposition of Ingredient Usage

value = trend + remainder



```
# gg_season to check seasonal pattern
gg_season(weekly_long, value) +
  facet_wrap(~ metric, scales = "free_y") +
  labs(title = "Seasonal Plot by Ingredient",
       x = "Week within Year", y = "Value") +
  theme_minimal()
```

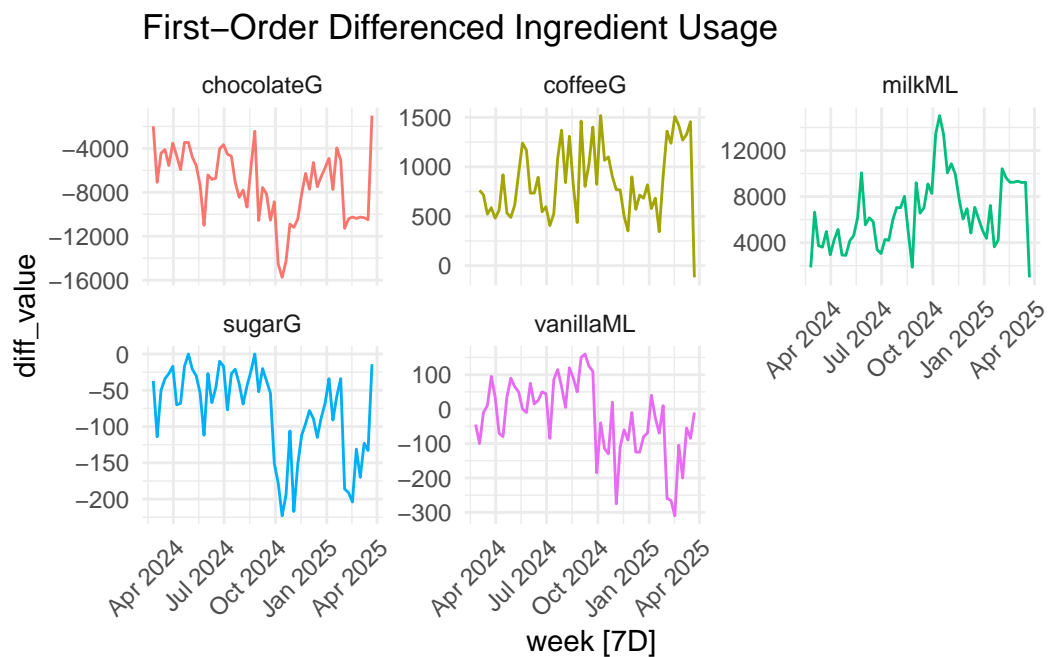
Seasonal Plot by Ingredient



```
#differencing
diff_recipe <- weekly_long %>%
```

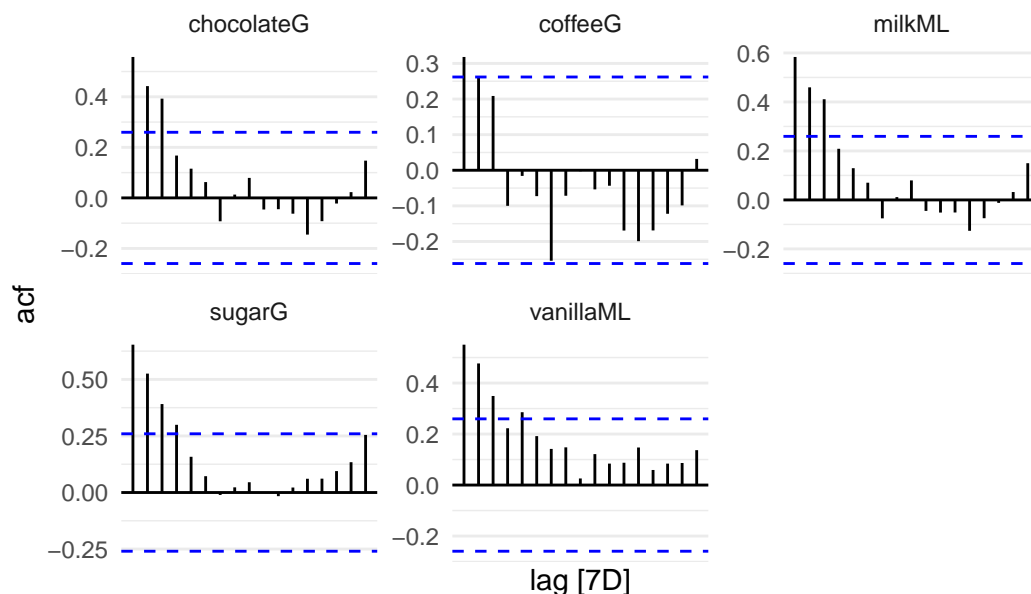
```
filter(metric != "sales_n") %>%
mutate(diff_value = difference(value))
```

```
diff_recipe %>%
  autoplot(diff_value) +
  facet_wrap(~metric, scales = "free_y") +
  labs(title = "First-Order Differenced Ingredient Usage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  theme(legend.position = "none")
```



```
#ACF
diff_recipe %>%
  filter(!is.na(diff_value)) %>%
  ACF(diff_value) %>%
  autoplot() +
  facet_wrap(~metric, scales = "free_y") +
  labs(title = "ACF of First-Order Differenced Ingredient Usage") +
  theme_minimal()
```

ACF of First-Order Differenced Ingredient Usage



```
ingredient_summary <- weekly_long %>%
  filter(metric != "sales_n") %>%
  group_by(metric) %>%
  summarise(
    avg_weekly_usage = mean(value, na.rm = TRUE),
    min_weekly_usage = min(value, na.rm = TRUE),
    max_weekly_usage = max(value, na.rm = TRUE),
    total_usage = sum(value, na.rm = TRUE),
    weeks_count = n()
  ) %>%
  arrange(desc(avg_weekly_usage))

ingredient_summary
```

```
# A tsibble: 285 x 7 [7D]
# Key:      metric [5]
  metric week      avg_weekly_usage min_weekly_usage max_weekly_usage
  <chr> <date>          <dbl>             <dbl>             <dbl>
1 milkML 2024-10-13      16420             16420             16420
2 milkML 2024-10-06      15080             15080             15080
3 milkML 2024-10-20      14860             14860             14860
4 milkML 2024-11-03      11860             11860             11860
5 milkML 2025-02-02      11860             11860             11860
6 milkML 2024-05-26      11340             11340             11340
7 milkML 2024-10-27      11220             11220             11220
8 milkML 2025-02-09      11000             11000             11000
9 milkML 2024-11-10      10900             10900             10900
10 milkML 2025-03-16      10900             10900             10900
# i 275 more rows
# i 2 more variables: total_usage <dbl>, weeks_count <int>
```

4.3.1 Data Description & Exploratory Summary

- Dataset Overview

This dataset represents weekly ingredient usage for a coffee vending machines from March 2024 to March 2025, including key ingredients like milk, coffee, chocolate, sugar, and vanilla.

- EDA Findings

Among all ingredients, milk usage exhibits the highest variation and fluctuation, particularly evident in the STL decomposition. As the ingredient with the highest demand, milk also demonstrates significant volatility in its weekly usage. In contrast, other ingredients like coffee and chocolate show more stable, stationary patterns, with coffee being the second most demanded ingredient, displaying only slight fluctuations over time. Although the STL decomposition did not reveal clear seasonal components, the `gg_season()` plots suggest that some ingredients may have weak seasonal patterns. This discrepancy arises because STL decomposition typically requires at least two full seasonal cycles to robustly detect seasonality, while our dataset spans only around one year. Therefore, while statistical methods may not identify strong seasonal trends, visual tools like `gg_season()` remain valuable for uncovering subtle, recurring usage patterns across weeks of the year.

- Assumptions

Given that milk consistently demonstrates the highest demand, it is reasonable to expect that it will remain the most used ingredient over the next 8 weeks. Meanwhile, ingredients such as coffee and chocolate are likely to maintain stable usage patterns, with only minor fluctuations potentially arising from consumer trends or specific events.

5 Modeling Method Choice and Diagnostics

We model each weekly ingredient time series using Seasonal ARIMA (SARIMA), selected for its ability to capture both autoregressive and seasonal dynamics that is present in our exploratory data analysis. We want to produce stable 8-week forecasts to inform weekly inventory restocking decisions.

5.1 Ingredients Model

We first examine each ingredient's stationarity and structure. After confirming data quality and transformation needs, we fit ARIMA models with seasonal terms. The March 23, 2025 week is excluded due to incomplete data.

```
# Reshaping to time series per ingredient
ingredients <- c("coffeeG", "milkML", "chocolateG", "sugarG", "vanillaML")

ingredient_ts <- weekly_sales %>%
  select(week, coffeeG, milkML, chocolateG, sugarG, vanillaML) %>%
  pivot_longer(-week, names_to = "ingredient", values_to = "amount") %>%
  as_tsibble(index = week, key = ingredient)

# Exclude March 23, 2025
ingredient_ts_filtered <- ingredient_ts %>%
  filter(week != "2025-03-23")
```

```
# Long format tsibble
weekly_long_ingredients <- weekly_sales %>%
  pivot_longer(cols = coffeeG:vanillaML, names_to = "ingredient", values_to = "value") %>%
  as_tsibble(index = week, key = ingredient)

# Fitting ARIMA to each ingredient using key grouping
ingredient_models <- weekly_long_ingredients %>%
  model(ARIMA(value))

# Fitting Seasonal ARIMA (SARIMA) to each ingredient series
arma_models <- ingredient_ts %>%
  model(ARIMA(value))
```

5.1.1 Model Evaluation and Diagnostics

For each ingredient, we examined residual plots from SARIMA to assess randomness, autocorrelation, and normality.

The Ljung-Box test results shown below evaluate whether the **residuals** from the ARIMA models are **uncorrelated**. It shows if the model has sufficiently captured all patterns in the time series.

```
# Check residuals
ingredient_models %>%
  augment() %>%
  features(.resid, ljung_box, lag = 8) %>%
  datatable(caption = "Ljung-Box Test Results for Ingredient ARIMA Residuals")
```

Show entries

Search:

Ljung-Box Test Results for Ingredient ARIMA Residuals

	ingredient	.model	lb_stat	lb_pvalue
1	chocolateG	ARIMA(value)	4.50342466877707	0.8090903821120865
2	coffeeG	ARIMA(value)	10.45168062058496	0.2347431280961428
3	milkML	ARIMA(value)	11.15073842714544	0.1933035734092685
4	sugarG	ARIMA(value)	4.563854180452648	0.80301122214714
5	vanillaML	ARIMA(value)	7.179052909963448	0.5174433508101526

Showing 1 to 5 of 5 entries

Previous Next

Since the p-value is for all ingredients are greater than 0.05 ($p > 0.05$), there is no significant autocorrelation in residuals. This means that the model is adequate and that the residuals resembles white noise. All ingredient SARIMA models have uncorrelated residuals and can be confidently used for forecasting.

5.2 Coffee Type Model

We may forecast coffee sales in two different ways: The unit sales forecast and change in unit sales forecast. As the week of March 23 is incomplete and therefore, causes a sharp drop in revenue, we will be excluding that week from our models.

5.2.1 With differencing (Model forecasting change)

Having made most of our weekly coffee sales data set stationary after differencing once, we use ARIMA to create our model.

```
#model with differencing
coffee_type_models_diff <- weekly_volume_diff %>%
  filter(coffee_name != "AMERICANO WITH MILK",
         week != "2025-03-23") %>%
  model(ARIMA(diff_volume))
```

We difference “Americano with Milk” coffee type a second time as differencing it once did not fix autocorrelation.

```
americano_milk_diff <- weekly_volume_diff %>%
  filter(coffee_name == "AMERICANO WITH MILK",
         week != "2025-03-23") %>%
  mutate(diff_volume = difference(total_volume, differences = 2))

americano_milk_models_diff <- americano_milk_diff %>%
  model(ARIMA(diff_volume))
```

5.2.2 No differencing (Model forecasting unit sales)

```
#model without differencing

coffee_type_models <- weekly_volume_diff %>%
  filter(week != "2025-03-23") %>%
  model(ARIMA(total_volume))
```

5.3 Revenue Model

Due to the significant trend with increasing variance, we use ETS to train the revenue model. This model does not require differencing.

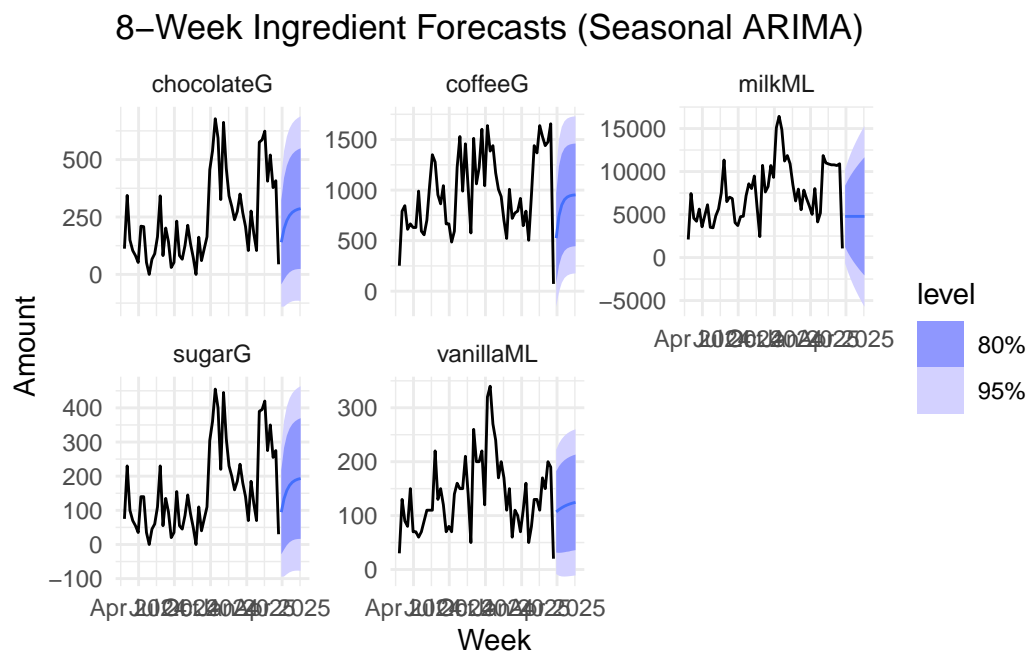
As the week of March 23 is incomplete and therefore, causes a sharp drop in revenue, we will be excluding that week from our model.

```
revenue_models <- weekly_revenue %>%
  filter(week != "2025-03-23") %>%
  model(ETS(total_revenue))
```

6 Generate Forecasts Results

6.1 Ingredients Forecast

```
forecast_ingredients <- ingredient_models %>%  
  forecast(h = "8 weeks")  
  
autoplot(forecast_ingredients, weekly_long_ingredients) +  
  labs(title = "8-Week Ingredient Forecasts (Seasonal ARIMA)",  
        y = "Amount", x = "Week") +  
  facet_wrap(~ingredient, scales = "free_y") +  
  theme_minimal()
```



6.2 Coffee Type Forecast

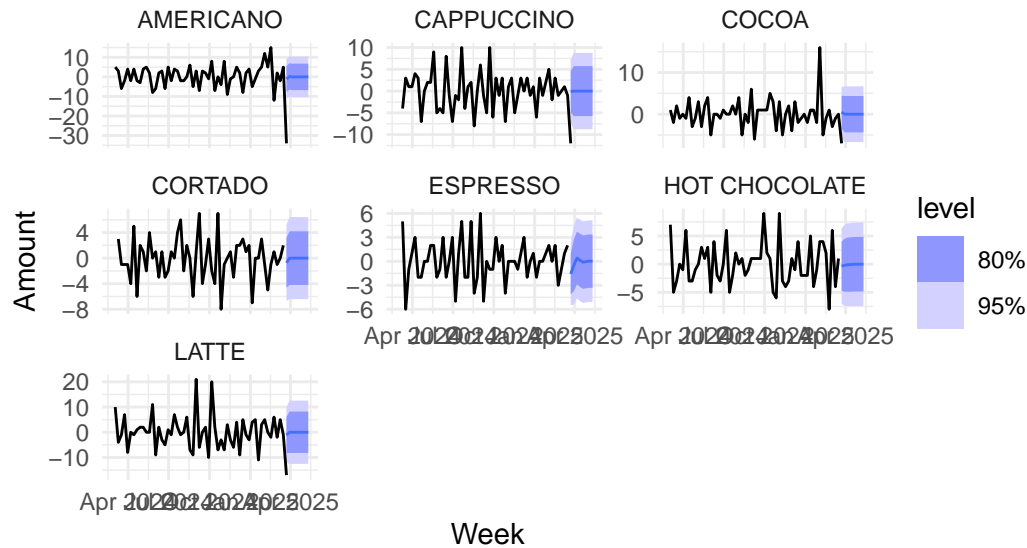
6.2.1 With differencing (Model forecasting change)

When forecasting differenced time series, we predict w much sales will change over time, not sales themselves. For most coffee types the ARIMA model expects those changes to average around zero.

```
forecast_coffee_type_diff <- coffee_type_models_diff %>%  
  forecast(h = "8 weeks")  
  
autoplot(forecast_coffee_type_diff, weekly_volume_diff) +  
  labs(title = "8-Week Coffee Type Forecasts (ARIMA)",  
        subtitle = "1 Difference",  
        y = "Amount", x = "Week") +  
  facet_wrap(~coffee_name, scales = "free_y") +  
  theme_minimal()
```


8-Week Coffee Type Forecasts (ARIMA)

1 Difference



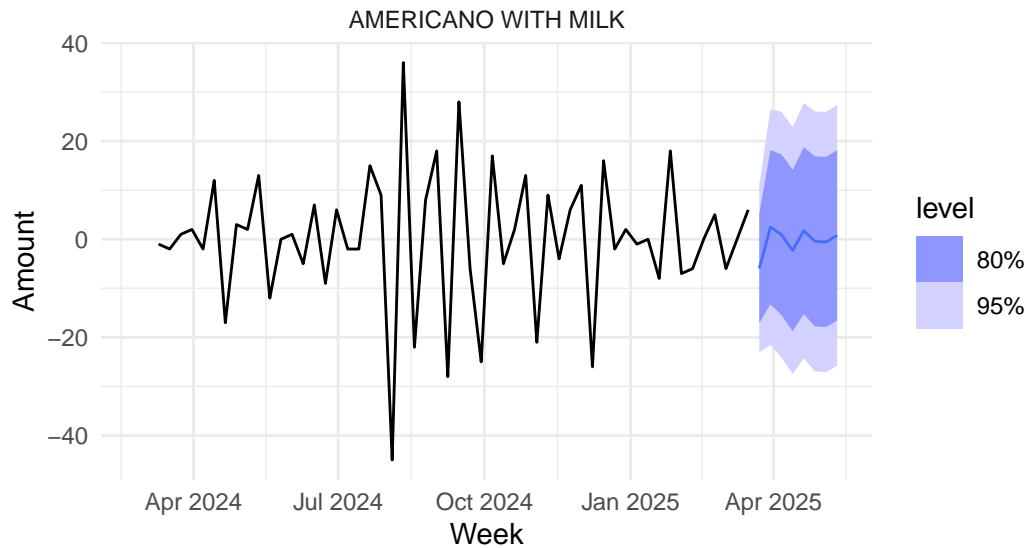
6.2.2 Americano with Milk - Forecast (2 Difference)

To remove autocorrelation, “americano with milk” coffee type was differenced twice. The ARIMA model predicts changes around zero, albeit with more fluctuation compared to other coffee types.

```
forecast_americano_milk <- americano_milk_models_diff %>%
  forecast(h = "8 weeks")

autoplot(forecast_americano_milk, americano_milk_diff) +
  labs(title = "8-Week Americano With Milk Forecast (ARIMA)",
       subtitle = "2 Difference",
       y = "Amount", x = "Week") +
  facet_wrap(~coffee_name, scales = "free_y") +
  theme_minimal()
```

8-Week Americano With Milk Forecast (ARIMA) 2 Difference



6.2.3 No differencing (Model forecasting unit sales)

Forecasting without differencing manually lets ARIMA do the differencing automatically and then revert the forecast back to the original scale, allowing us to forecast unit sales over time.

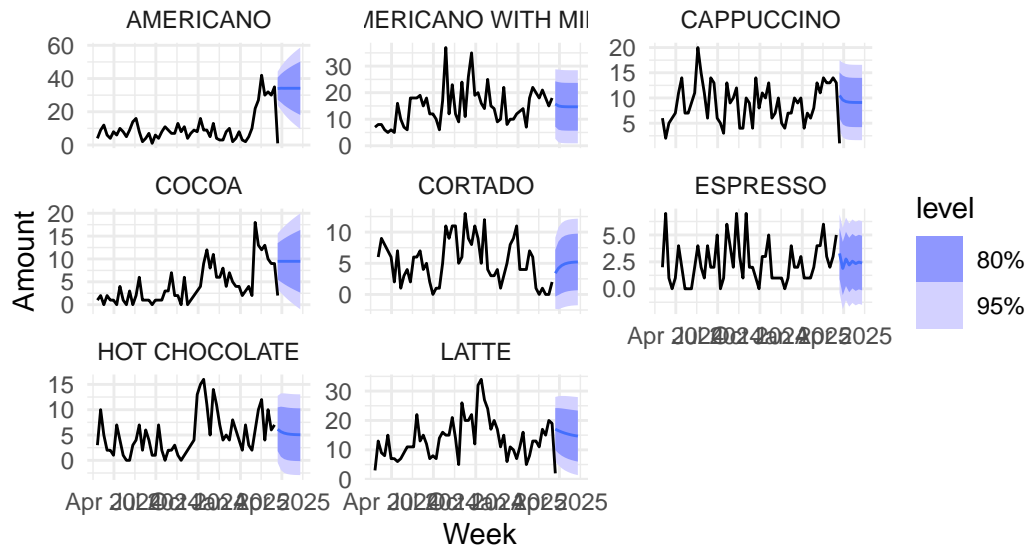
As the differenced ARIMA model suggests, the sales are expected to be relatively stable for most coffee types over the next 8 weeks. Cortado, latte and hot chocolate may see some decrease in sales. We should plan to decrease supply for those.

```
forecast_coffee_type <- coffee_type_models %>%
  forecast(h = "8 weeks")

autoplot(forecast_coffee_type, weekly_volume_diff) +
  labs(title = "8-Week Coffee Type Forecasts (ARIMA)",
        subtitle = "No Differencing",
        y = "Amount", x = "Week") +
  facet_wrap(~coffee_name, scales = "free_y") +
  theme_minimal()
```

8-Week Coffee Type Forecasts (ARIMA)

No Differencing



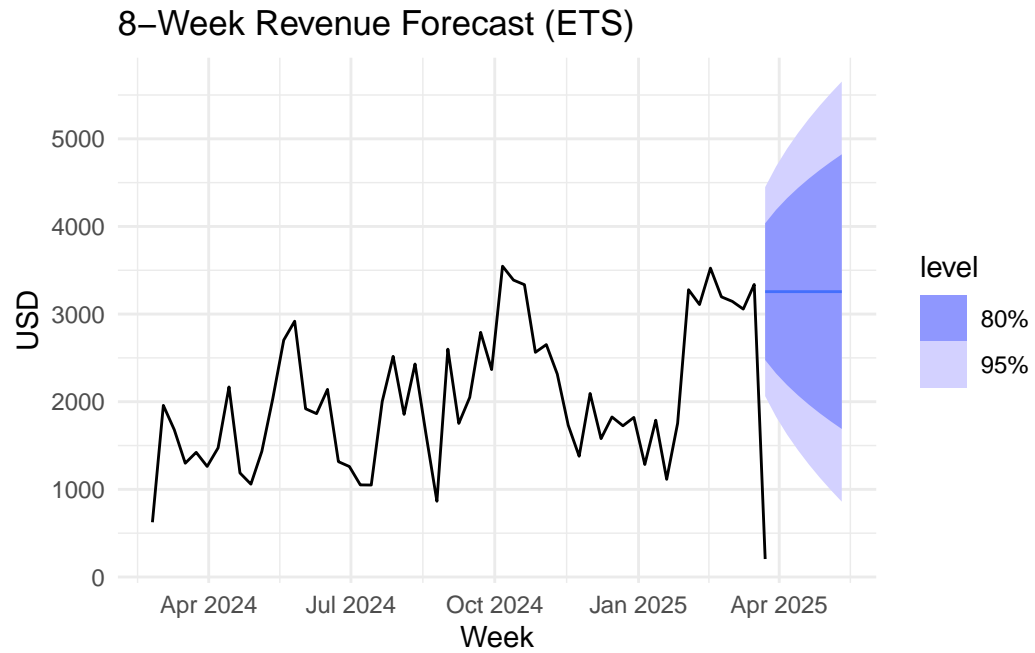
6.3 Revenue Forecast

Given ETS, places more significance onto recent values, the forecasts predicts the revenue will be stable over the next 8 weeks.

Since the week of March 23 was incomplete, it was excluded from model training. As a result, while the current data (black line) show the weekly revenue around \$300, the model predicts it will rise to \$3000 by the end of the week based on the previous period.

```
forecast_revenue <- revenue_models %>%
  forecast(h = "8 weeks")

autoplot(forecast_revenue, weekly_revenue) +
  labs(title = "8-Week Revenue Forecast (ETS)",
       y = "USD", x = "Week") +
  theme_minimal()
```



6.3.1 Conclusion

By using this report in aligning weekly inventory decisions with the forecasts, the coffee vending machine can reduce waste, maintain product availability, and improve cost-efficiency. This ensures customers to consistently find their preferred drinks stocked and ready.

7 Appendix

7.0.1 Machine 1

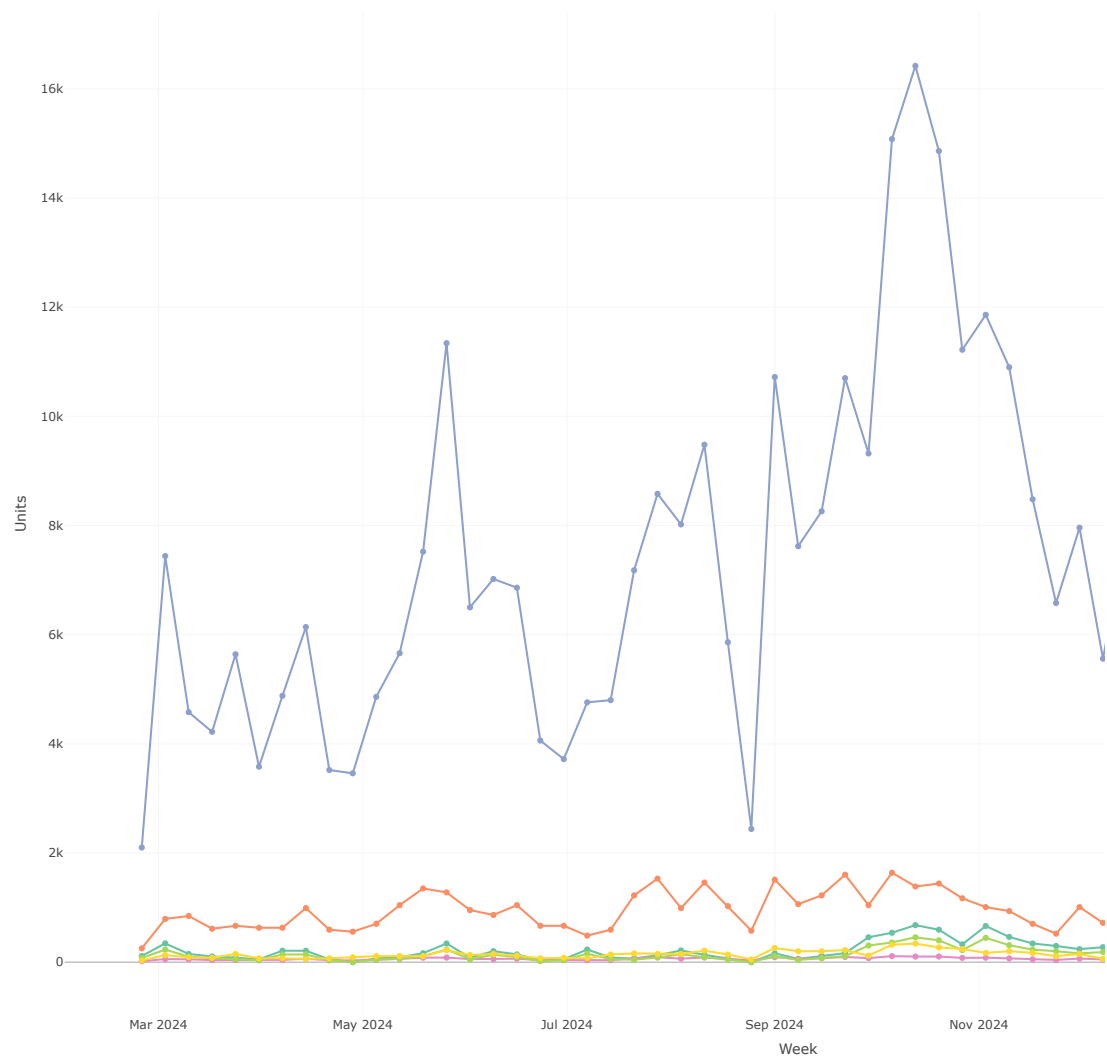


Figure 2: Machine 1 Weekly ingredient demand vs. cups sold

Coffee Bros. 2024. “Milk-to-Espresso Ratio Calculator.” 2024. <https://coffeebros.com/pages/milk-to-espresso-ratio-calculator>.

Cordell, George. 2024. “Americano Coffee with Milk Recipe.” 2024. <https://coffeelikers.com/americano-coffee-with-milk/>.

Hersheyland Test Kitchen. 2025. “Hot Cocoa for One.” 2025. <https://www.hersheyland.com/recipes/hot-cocoa-for-one.html>.

Page, Amazon Product. 2025. “Torani Vanilla Syrup Pump – 8 Ml Per Pump.” 2025. <https://www.amazon.com/dp/B09P49HWKK>.

Raffii, Ahlam. 2024. “How to Make the Perfect Cappuccino.” 2024. <https://www.thespruceeats.com/how-to-make-cappuccinos-766116>.

Specialty Coffee Association. 2018. “Defining the Ever-Changing Espresso.” <https://sca.coffee/sca-news/25-magazine/issue-3/defining-ever-changing-espresso-25-magazine-issue-3>.

Wine Editors, Food &. 2025. “8 Types of Coffee Drinks to Order Around the World.” 2025. <https://www.foodandwine.com/types-of-coffee-drinks-11724561>.