

Manufacturing Downtime Analysis

Project Report

Digital Egypt Pioneers Initiative (DEPI)

Google Data Analysis Specialist Track

Contributors

Abd Al-Rahman Mohamed Amin

Ahmed Mohamed

Mohamed Shebl Azab

Omar Mohamed Shebl

November 2024 – April 2025

Table of Contents

[Executive Summary 2](#_Toc194693200)

[Phase I: Data Cleaning and Preprocessing 3](#_Toc194693201)

[Data Overview 3](#_Toc194693202)

[Data Tables Normalization 4](#_Toc194693203)

[The Data Cleaning Process 4](#_Toc194693204)

# Executive Summary

This project investigates the factors causing production downtime in a soda drinks company. Using data cleaning, modeling, exploratory analysis, and forecasting, we identify the major contributors to downtime and predict future trends. The findings provide actionable insights to optimize operations and minimize downtime.

The project follows a structured data analysis approach. First, raw operational data is cleaned and preprocessed to ensure accuracy. A data model is then built to establish relationships between key variables affecting downtime. Exploratory analysis is conducted to uncover patterns, followed by forecasting models that predict future downtime trends. Finally, a Tableau dashboard visualizes the findings for decision-making.

The dataset consists of downtime records collected over a period of 5 days. The dataset included categories for causing factors, details on the batches produced such as production date, and details on the products produced by the company such as product flavors and package sizes. This allowed analysis of downtime by product features and causing factors.

The analysis found that five out of possible twelve causing factors contributed to 80% of the downtime. Machine adjustments were the primary cause behind downtime accounting for 23.92% of the downtime during the observed period. Prediction?

**Recommendations**

(To be added based on findings)

This project provides data-driven insights to enhance manufacturing efficiency and reduce operational disruptions. By implementing the suggested strategies, the company can achieve higher productivity and cost savings.

# Phase I: Data Cleaning and Preprocessing

Data cleaning is a crucial step before analysis because raw data often contains errors, inconsistencies, and missing values that can lead to inaccurate insights. By removing duplicates, handling missing data, correcting errors, and standardizing formats, data cleaning ensures the dataset is reliable and consistent.

## Data Overview

The raw dataset contains records on produced batches and downtime caused by different factors. The table below describes the dataset in details:

|  |  |  |
| --- | --- | --- |
| **Entity** | **Entity Type** | **Description** |
| **Downtime Factors** | **Dimension Table** | **Includes details on each downtime factor.** |
| Factor | Attribute | Unique identifier for each downtime factor. |
| Description | Attribute | Descriptive text for each factor. |
| Operator Error | Attribute | Whether a factor involves a human error. |
| **Line Downtime** | **Fact Table** | **Records downtime caused by each factor in production batches.** |
| Batch | Attribute | Unique identifier for produced batches. |
| Factor | Attribute | Downtime minutes for each factor in a wide format (12 columns). |
| **Line Productivity** | **Fact Table** | **Records details on each of the batches produced.** |
| Batch | Attribute | Unique identifier for the batch. |
| Date | Attribute | Production date. |
| Product | Attribute | Unique identifier for products. |
| Operator | Attribute | Name of the operator in charge during batch production. |
| Start Time | Attribute | Time at which batch production started. |
| End Time | Attribute | Time at which batch production ended. |
| **Products** | **Dimension Table** | **Includes details on the products produced by the company.** |
| Product | Attribute | Unique identifier for products. |
| Flavor | Attribute | Soda flavor of the product e.g., cola, lemon …etc. |
| Size | Attribute | Pack volume in milli liters. |
| Min Batch Time | Attribute | Minimum time required to produce a single batch without delay. |

## Data Tables Normalization

Data table normalization is the process of organizing a database to reduce redundancy and improve data integrity. It involves structuring tables according to a series of normal forms (NF), such as First Normal Form (1NF), Second Normal Form (2NF), and so on. The process eliminates duplicate data, ensures logical dependencies, and minimizes anomalies in data insertion, updating, and deletion. Normalization was applied to each of the provided tables to the third normal form (3NF):

* First Normal Form (1NF): ensure no repeating groups or multivalued attributes and that each column contains atomic values.
* Second Normal Form (1NF): ensure all non-key attributes are fully functionally dependent on the entire primary key.
* Third Normal Form (3NF): eliminate transitive dependencies so non-key attributes depend only on the primary key.

**Findings:**

1. The Line Downtime table violated 1NF because downtime factors are stored as column headers. Microsoft Excel Power Query’s unpivot columns feature was used to convert the table into a long format with only 3 columns: batch, factor, and downtime.
2. The Products table violated 3NF and was found to have a transitive dependency where Min Batch Time depended on Size rather than the primary key. Min Batch Time and Size were separated in a new table to resolve this issue.

## The Data Cleaning Process

Python was used to ensure that each table was clean and ready for analysis:

1. Check for missing values: No value was found to be missing in any of the tables.

1. # Check for null values in each column

2. line\_prod.isnull().sum()

1. Check that the number of unique IDs is equal to the number of rows in the table: all rows were found to have unique IDs in all the tables.

1. # Count the number of unique batch IDs

2. # The number of unique batch IDs should be equal to the number of rows in the data frame

3. line\_prod.Batch.nunique()

1. Check for duplicate records: no duplicate records were found in any of the tables.

1. # Find the number of duplicate records

2. line\_prod.duplicated().sum()

1. Check data type and formatting consistency in each column

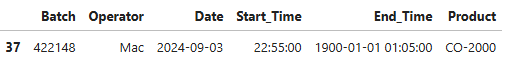
1. # Check that all End Time values are of the type datetime.time

2. line\_prod["End\_Time"].apply(lambda x: isinstance(x, datetime.time)).all()

The data types of values in the End Time column were not consistent. The following code was used to isolate rows where the type was not datetime.time where only one row was found to have a datetime.datetime data type:

1. # Find values in End Time that are not of the type datetime.time

2. line\_prod[line\_prod["End\_Time"].apply(lambda x: not(isinstance(x, datetime.time)))]



The time component of was extracted and used to overwrite the value at index 37:

1. # Extract the time component and rewrite the value

2. line\_prod.loc[37, "End\_Time"] = line\_prod.loc[37, "End\_Time"].time()