Fall 2024 - IS 507
Data, Statistical Models, and Information

Fast Fashion Supply Chain Management

Project Report

Submitted to: Professor Yang Wang

Submitted By:

Alisha Shinde(ashinde3)
Akanksha Agrawal(Aa148)
Prisha Singhania(pds4)

Objective

The goal of this project is to enhance the fast fashion supply chain by tackling issues related to high return rates, profitability obstacles, and shipping inefficiencies. By leveraging data-driven insights, it aims to lower costs, improve product quality, streamline operations, and increase customer satisfaction. Utilizing data-driven strategies can help forecast returns, optimize operational processes, and refine shipping and warehousing practices. This methodology not only reduces costs but also enhances product quality, fosters customer loyalty, and promotes profitability and sustainability.

Problem Statement

The fast fashion industry needs to work on elevated return rates, operational inefficiencies, and challenges in pinpointing profitable products, which leads to higher costs and diminished customer satisfaction. Addressing these challenges is crucial for improving efficiency, cutting costs, and maintaining competitiveness.

Dataset Overview

Source: We have selected the Fast Fashion Supply Chain dataset from Kaggle (The dataset can be accessed here).
The dataset consists of four files, each capturing different aspects of the supply chain:

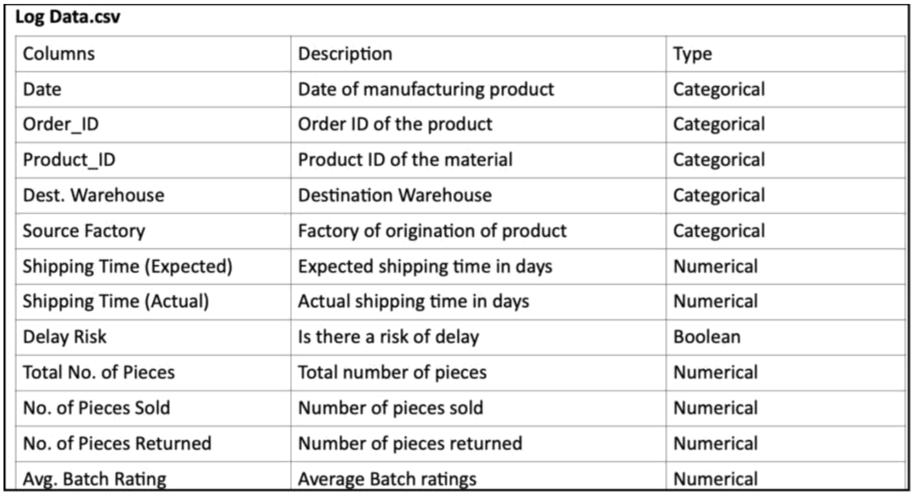
Log Data.csv (4.6 MB): Contains product order details, including manufacturing date, order and product IDs, source factory, and destination warehouse. It tracks expected and actual shipping times, delay risks, total pieces shipped, sold, and returned, along with average batch ratings.

Production Costs.csv (5 KB): Focuses on manufacturing costs for products across 5 factories, listing product IDs and factory-specific costs.

Products.csv (1.2 KB): Provides details of 40 unique clothing and accessory items, including product IDs, names, target gender, selling prices, and weights.

Warehouse Shipping Costs.csv (8.4 KB): Outlines shipping costs for 1,000 pieces of products from factories to 20 warehouses, with costs varying based on the product and factory.

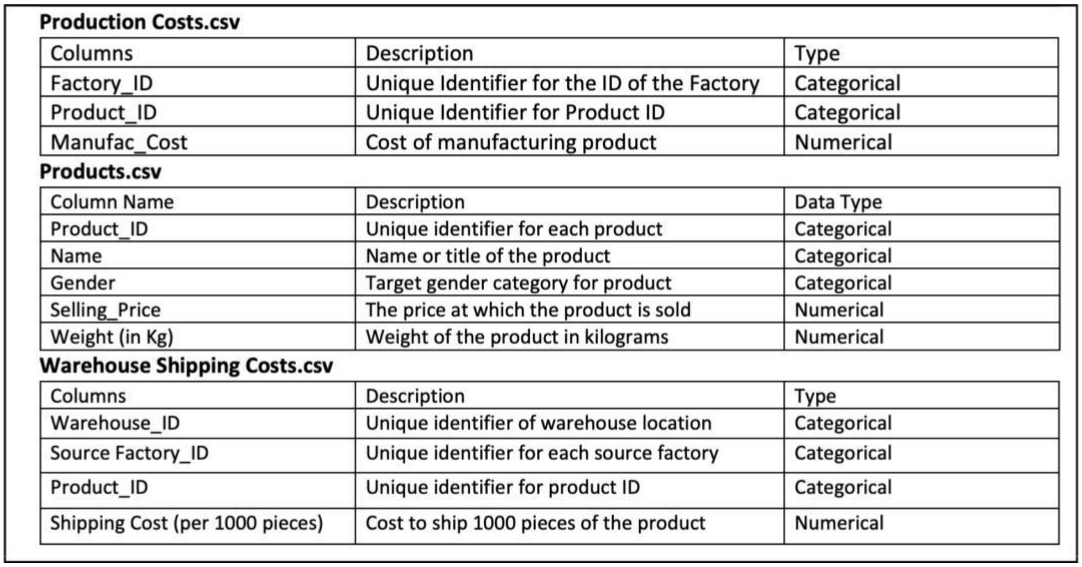
Together, these files offer a complete view of the supply chain, covering production, logistics, and sales.



Log Data.csv

Columns	Description	Type
Date	Date of manufacturing product	Categorical
Order_ID	Order ID of the product	Categorical
Product_ID	Product ID of the material	Categorical
Dest. Warehouse	Destination Warehouse	Categorical
Source Factory	Factory of origination of product	Categorical
Shipping Time (Expected)	Expected shipping time in days	Numerical
Shipping Time (Actual)	Actual shipping time in days	Numerical
Delay Risk	Is there a risk of delay	Boolean
Total No. of Pieces	Total number of pieces	Numerical
No. of Pieces Sold	Number of pieces sold	Numerical
No. of Pieces Returned	Number of pieces returned	Numerical
Avg. Batch Rating	Average Batch ratings	Numerical

Table 1: Log.csv



Production Costs.csv

Columns	Description	Type
Factory_ID	Unique Identifier for the ID of the Factory	Categorical
Product_ID	Unique Identifier for Product ID	Categorical
Manufac_Cost	Cost of manufacturing product	Numerical

Products.csv

Column Name	Description	Data Type
Product_ID	Unique identifier for each product	Categorical
Name	Name or title of the product	Categorical
Gender	Target gender category for product	Categorical
Selling_Price	The price at which the product is sold	Numerical
Weight (in Kg)	Weight of the product in kilograms	Numerical

Warehouse Shipping Costs.csv

Columns	Description	Type
Warehouse_ID	Unique identifier of warehouse location	Categorical
Source Factory_ID	Unique identifier for each source factory	Categorical
Product_ID	Unique identifier for product ID	Categorical
Shipping Cost (per 1000 pieces)	Cost to ship 1000 pieces of the product	Numerical

Table 2: Production Cost.csv, Products.csv, Warehouse Shipping

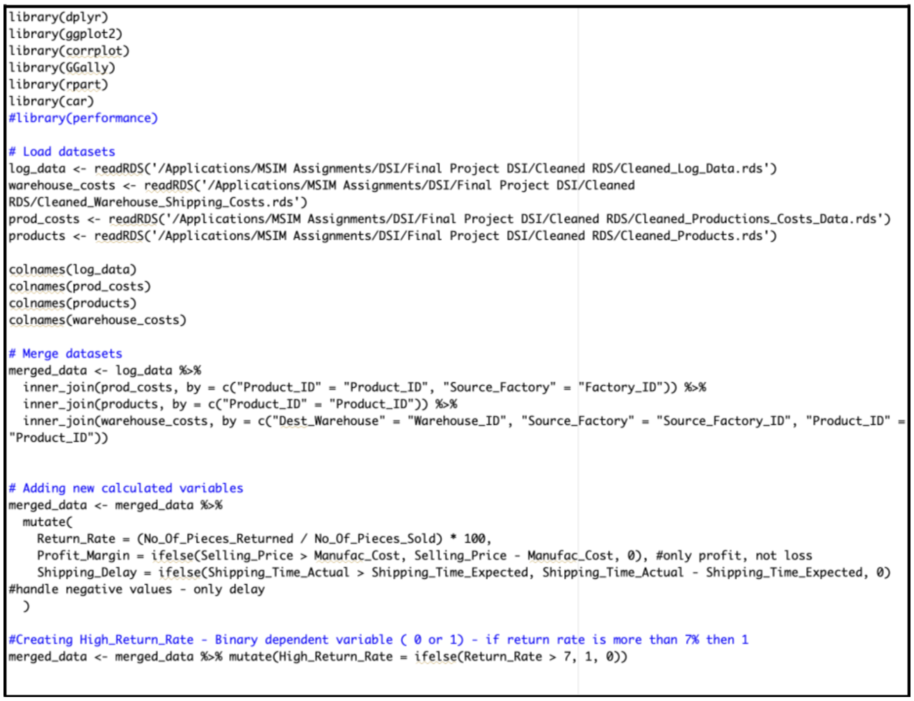
Costs.csv **Comparative Analysis of Fast Fashion Supply Chain Solutions:**

This project analyzes a fast fashion supply chain dataset to improve efficiency and profitability. Initial attempts at using logistic regression to predict returns were unsuccessful due to weak predictor relationships. Logistic regression was successfully used to predict high return risk and multiple linear regression to analyze profit margins and total costs. Key findings highlight the impact of shipping delays, warehouse selection, quantity of items, factory, and product weight on costs. The study doesn't compare its approach to existing literature. Further research is needed to validate the findings and explore more sophisticated modeling techniques.

Research Question 1- Part 1) Are products from certain factories more likely to be returned, and could a rating-based metric predict which products should undergo additional quality control?

For the first question, we wanted to see if products from certain factories were more likely to be returned and if a rating-based metric could help identify which products need extra quality checks. We chose logistic regression because it works well for yes-or-no problems, like predicting if a product has a high return rate. It also helps analyze how factors like factory origin, product ratings, and other characteristics impact returns.

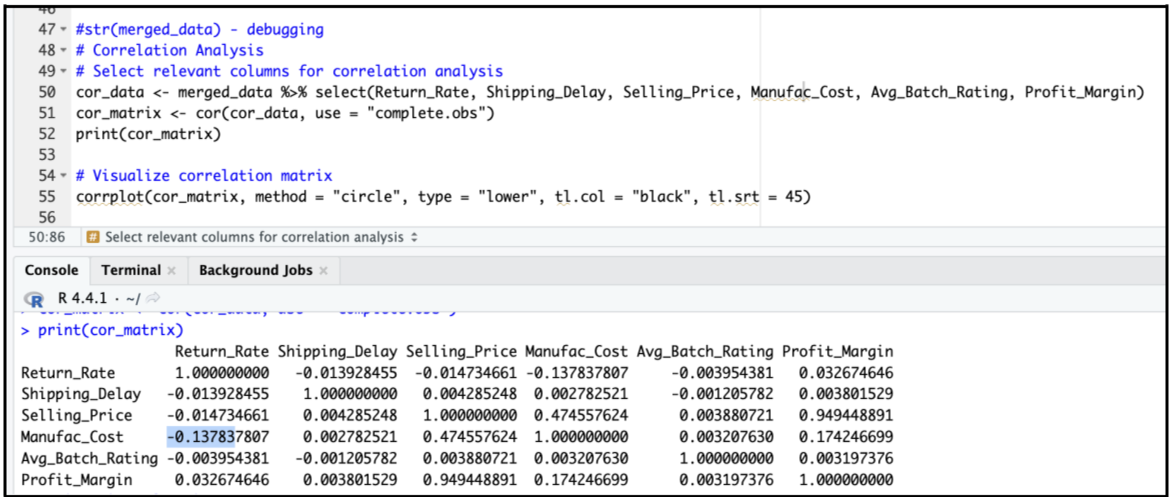
We used several statistical tests and Exploratory Data Analysis (EDA) to assess the applicability of logistic regression. Key fields like Product_ID, Source_Factory, and Dest_Warehouse were used to merge the cleaned datasets into a single, unified dataset. The following new variables were created: Shipping_Delay (shipping delays), Profit_Margin (profit per product), and Return_Rate (% of products returned). To find return rates higher than 7%, a binary target variable called High_Return_Rate was created as shown in Figure 1 below.



```
library(dplyr)
library(ggplot2)
library(corrplot)
library(GGally)
library(rpart)
library(car)
#library(performance)

# Load datasets
log_data <- readRDS('/Applications/MSIM Assignments/DSI/Final Project DSI/Cleaned RDS/Cleaned_Log_Data.rds')
warehouse_costs <- readRDS('/Applications/MSIM Assignments/DSI/Final Project DSI/Cleaned 
RDS/Cleaned_Warehouse_Shipping_Costs.rds')
prod_costs <- readRDS('/Applications/MSIM Assignments/DSI/Final Project DSI/Cleaned RDS/Cleaned_Productions_Costs_Data.rds')
products <- readRDS('/Applications/MSIM Assignments/DSI/Final Project DSI/Cleaned RDS/Cleaned_Products.rds')

colnames(log_data)
colnames(prod_costs)
colnames(products)
colnames(warehouse_costs)

# Merge datasets
merged_data <- log_data %>%
  inner_join(prod_costs, by = c("Product_ID" = "Product_ID", "Source_Factory" = "Factory_ID")) %>%
  inner_join(products, by = c("Product_ID" = "Product_ID")) %>%
  inner_join(warehouse_costs, by = c("Dest_Warehouse" = "Warehouse_ID", "Source_Factory" = "Source_Factory_ID", "Product_ID" =
"Product_ID"))


# Adding new calculated variables
merged_data <- merged_data %>%
  mutate(
    Return_Rate = (No_Of_Pieces_Returned / No_Of_Pieces_Sold) * 100,
    Profit_Margin = ifelse(Selling_Price > Manufac_Cost, Selling_Price - Manufac_Cost, 0), #only profit, not loss
    Shipping_Delay = ifelse(Shipping_Time_Actual > Shipping_Time_Expected, Shipping_Time_Actual - Shipping_Time_Expected, 0)
#handle negative values - only delay
  )

#Creating High_Return_Rate - Binary dependent variable ( 0 or 1) - if return rate is more than 7% then 1
merged_data <- merged_data %>% mutate(High_Return_Rate = ifelse(Return_Rate > 7, 1, 0))
```

Figure 1: Merging datasets and creating new variables

Correlation analysis revealed weak or negligible relationships between Return_Rate and other predictors, with Manufac_Cost showing a weak negative correlation (-0.13) and other variables close to zero (Figures 2 and 3). Despite this, logistic regression was applied to uncover non-linear patterns. A summary of applying the GLM function is shown in Figure 4.



```r
#str(merged_data) - debugging
# Correlation Analysis
# Select relevant columns for correlation analysis
cor_data <- merged_data %>% select(Return_Rate, Shipping_Delay, Selling_Price, Manufac_Cost, Avg_Batch_Rating, Profit_Margin)
cor_matrix <- cor(cor_data, use = "complete.obs")
print(cor_matrix)

# Visualize correlation matrix
corrplot(cor_matrix, method = "circle", type = "lower", tl.col = "black", tl.srt = 45)

```

```
> print(cor_matrix)
                 Return_Rate Shipping_Delay Selling_Price Manufac_Cost Avg_Batch_Rating Profit_Margin
Return_Rate       1.000000000   -0.013928455  -0.014734661 -0.137837807     -0.003954381   0.032674646
Shipping_Delay   -0.013928455    1.000000000   0.004285248  0.002782521     -0.001205782   0.003801529
Selling_Price    -0.014734661    0.004285248   1.000000000  0.474557624      0.003880721   0.949448891
Manufac_Cost     -0.137837807    0.002782521   0.474557624  1.000000000      0.003207630   0.174246699
Avg_Batch_Rating -0.003954381   -0.001205782   0.003880721  0.003207630      1.000000000   0.003197376
Profit_Margin     0.032674646    0.003801529   0.949448891  0.174246699      0.003197376   1.000000000
```

Figure 2: Correlation Analysis

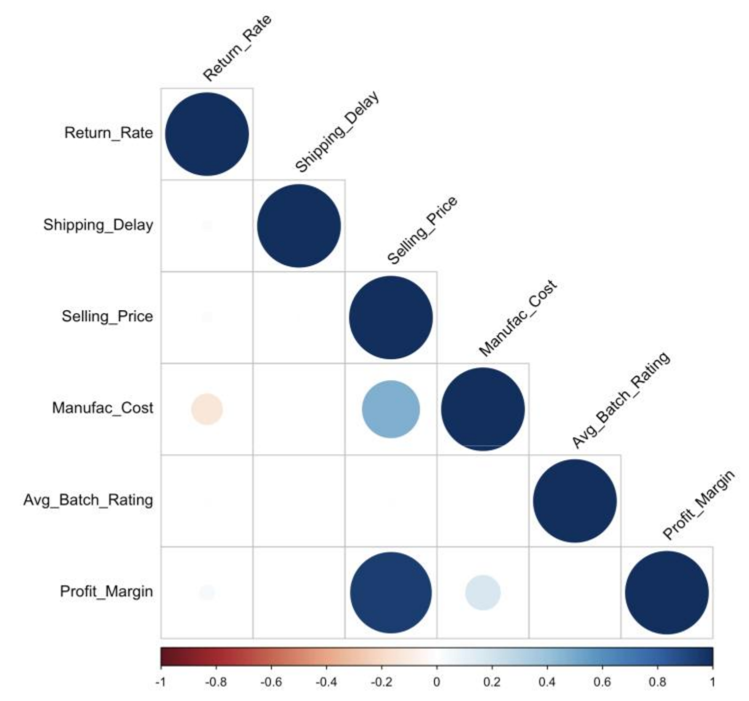
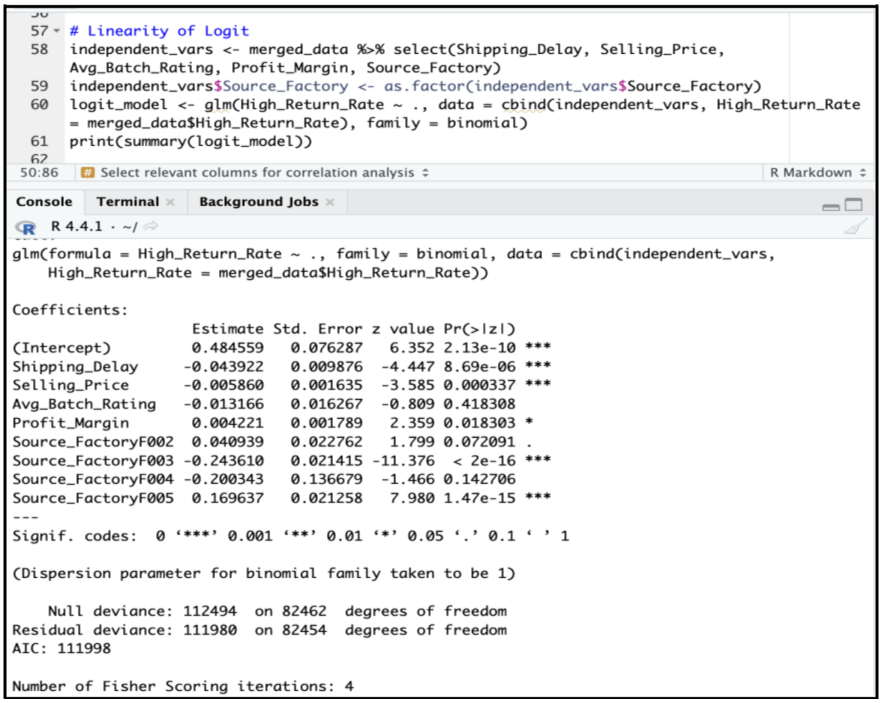


Figure 3: Correlation Analysis plot

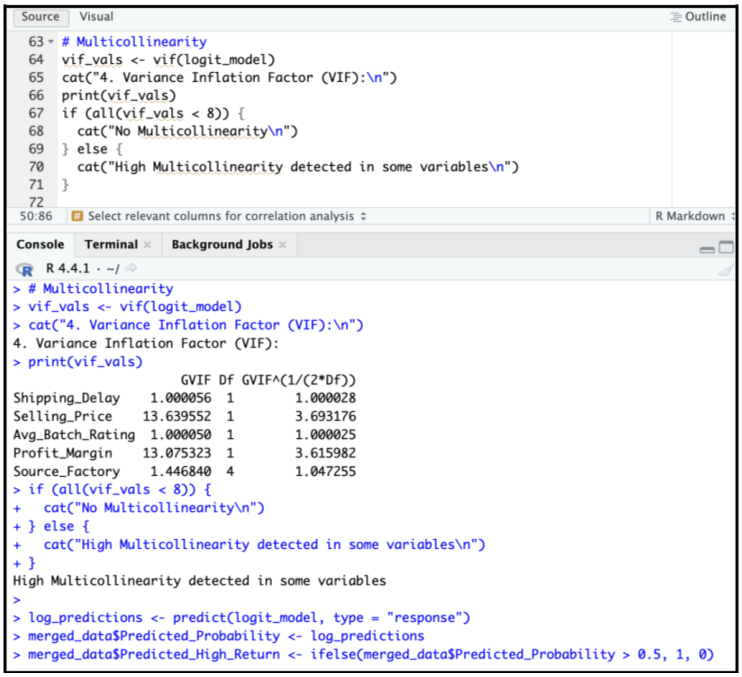


```
# Linearity of Logit
independent_vars <- merged_data %>% select(Shipping_Delay, Selling_Price,
Avg_Batch_Rating, Profit_Margin, Source_Factory)
independent_vars$Source_Factory <- as.factor(independent_vars$Source_Factory)
logit_model <- glm(High_Return_Rate ~ ., data = cbind(independent_vars, High_Return_Rate
= merged_data$High_Return_Rate), family = binomial)
print(summary(logit_model))
```

```
glm(formula = High_Return_Rate ~ ., family = binomial, data = cbind(independent_vars,
    High_Return_Rate = merged_data$High_Return_Rate))

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.484559   0.076287   6.352 2.13e-10 ***
Shipping_Delay    -0.043922   0.009876  -4.447 8.69e-06 ***
Selling_Price     -0.005860   0.001635  -3.585 0.000337 ***
Avg_Batch_Rating  -0.013166   0.016267  -0.809 0.418308
Profit_Margin      0.004221   0.001789   2.359 0.018303 *
Source_FactoryF002  0.040939   0.022762   1.799 0.072091 .
Source_FactoryF003 -0.243610   0.021415 -11.376  < 2e-16 ***
Source_FactoryF004 -0.200343   0.136679  -1.466 0.142706
Source_FactoryF005  0.169637   0.021258   7.980 1.47e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 112494  on 82462  degrees of freedom
Residual deviance: 111980  on 82454  degrees of freedom
AIC: 111998

Number of Fisher Scoring iterations: 4
```

Figure 4: Applying GLM and checking the summary

Significant predictors included Shipping_Delay, Selling_Price, and Profit_Margin, while Avg_Batch_Rating was insignificant. However, multicollinearity issues were observed between Selling_Price and Profit_Margin (figure 5), indicating potential redundancy in predictors and impacting the model's performance.



```
# Multicollinearity
vif_vals <- vif(logit_model)
cat("4. Variance Inflation Factor (VIF):\n")
print(vif_vals)
if (all(vif_vals < 8)) {
  cat("No Multicollinearity\n")
} else {
  cat("High Multicollinearity detected in some variables\n")
}
```

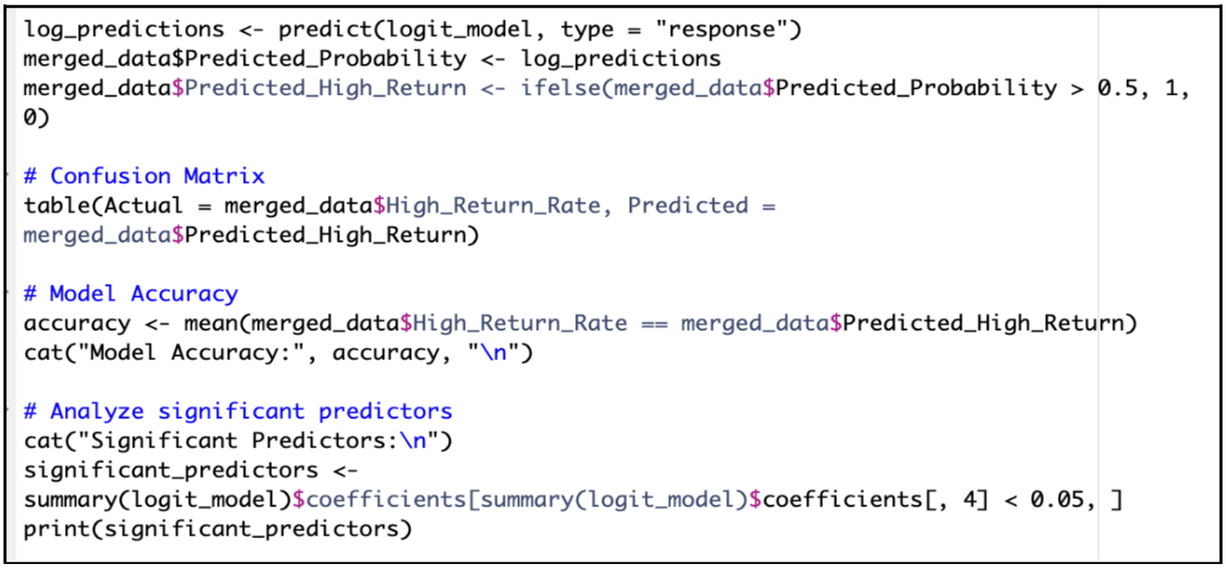
```
> # Multicollinearity
> vif_vals <- vif(logit_model)
> cat("4. Variance Inflation Factor (VIF):\n")
4. Variance Inflation Factor (VIF):
> print(vif_vals)
                      GVIF Df GVIF^(1/(2*Df))
Shipping_Delay    1.000056  1        1.000028
Selling_Price    13.639552  1        3.693176
Avg_Batch_Rating  1.000050  1        1.000025
Profit_Margin    13.075323  1        3.615982
Source_Factory    1.446840  4        1.047255
> if (all(vif_vals < 8)) {
+   cat("No Multicollinearity\n")
+ } else {
+   cat("High Multicollinearity detected in some variables\n")
+ }
High Multicollinearity detected in some variables
>
> log_predictions <- predict(logit_model, type = "response")
> merged_data$Predicted_Probability <- log_predictions
> merged_data$Predicted_High_Return <- ifelse(merged_data$Predicted_Probability > 0.5, 1, 0)
```

Figure 5: Checking Multicollinearity

Predictions were generated using the logistic regression model, and observations were classified based on a probability threshold of 0.5 (as shown in Figure 6).

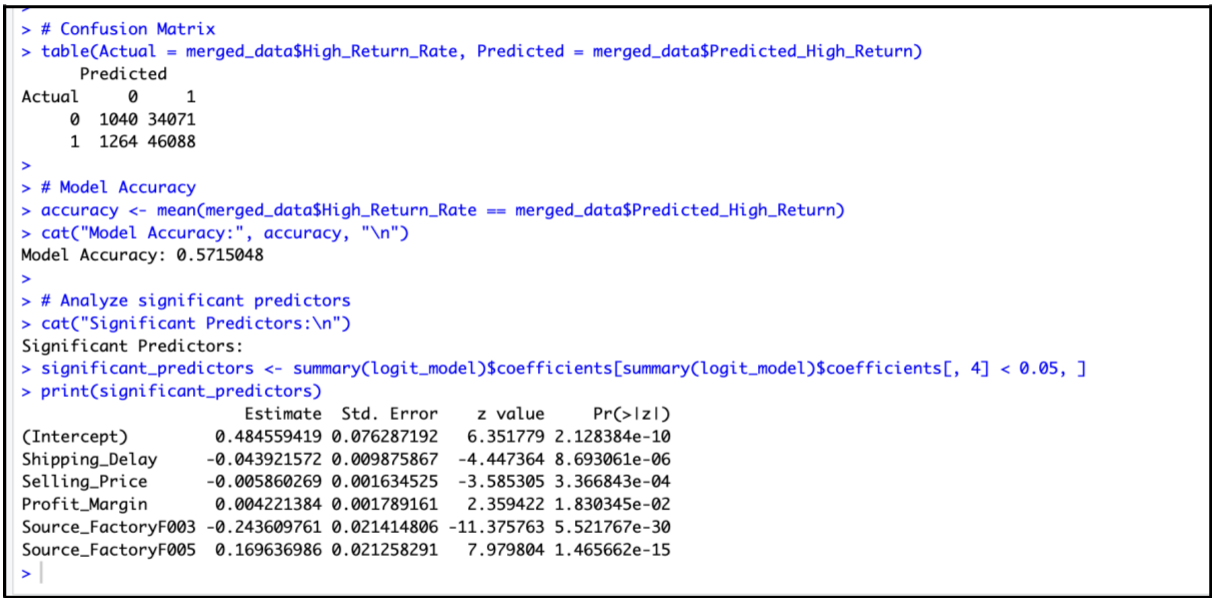
The Confusion Matrix showed that while the model correctly predicted many high-return cases (true positives), it also generated a significant number of false positives. (Figure 7 highlights the output of the confusion matrix).

The model achieved an accuracy of 57.15%, which is only slightly better than random guessing. (Figure 7 shows the Model accuracy).



```
log_predictions <- predict(logit_model, type = "response")
merged_data$Predicted_Probability <- log_predictions
merged_data$Predicted_High_Return <- ifelse(merged_data$Predicted_Probability > 0.5, 1,
0)

# Confusion Matrix
table(Actual = merged_data$High_Return_Rate, Predicted =
merged_data$Predicted_High_Return)

# Model Accuracy
accuracy <- mean(merged_data$High_Return_Rate == merged_data$Predicted_High_Return)
cat("Model Accuracy:", accuracy, "\n")

# Analyze significant predictors
cat("Significant Predictors:\n")
significant_predictors <-
summary(logit_model)$coefficients[summary(logit_model)$coefficients[, 4] < 0.05, ]
print(significant_predictors)
```

Figure 6: Predicting Probability, Confusion Matrix, Model Accuracy



```
> # Confusion Matrix
> table(Actual = merged_data$High_Return_Rate, Predicted = merged_data$Predicted_High_Return)
      Predicted
Actual     0     1
     0  1040 34071
     1  1264 46088
>
> # Model Accuracy
> accuracy <- mean(merged_data$High_Return_Rate == merged_data$Predicted_High_Return)
> cat("Model Accuracy:", accuracy, "\n")
Model Accuracy: 0.5715048
>
> # Analyze significant predictors
> cat("Significant Predictors:\n")
Significant Predictors:
> significant_predictors <- summary(logit_model)$coefficients[summary(logit_model)$coefficients[, 4] < 0.05, ]
> print(significant_predictors)
                      Estimate  Std. Error    z value     Pr(>|z|)
(Intercept)        0.484559419 0.076287192   6.351779 2.128384e-10
Shipping_Delay    -0.043921572 0.009875867  -4.447364 8.693061e-06
Selling_Price     -0.005860269 0.001634525  -3.585305 3.366843e-04
Profit_Margin      0.004221384 0.001789161   2.359422 1.830345e-02
Source_FactoryF003 -0.243609761 0.021414806 -11.375763 5.521767e-30
Source_FactoryF005  0.169636986 0.021258291   7.979804 1.465662e-15
>
```

Figure 7: Output of statistical tests

Note: In the proposal, we assumed that the first research question could be solved using logistic regression. However, after EDA, we found that logistic regression struggles when the relationships between predictors and the target variable are weak or complex. To properly explore and understand logistic regression, we focused on a different research question better suited for this model, as explained in the next section.

Research Question 1- Part 2) Modified Question - What factors contribute to a high risk of product returns, and how effectively can we predict this risk based on the metric rating available in the sales data?

Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no.

This model analyses the data to find the dependency of the variable high_return_risk on various factors such as sales_efficiency, Avg_Batch_rating, and source_factory.

After the data is entered into the pipeline to perform the analysis, we mutate the model to form new columns and establish new relations to generate insightful results.

$$= \overline{\hspace{6cm}}$$

$$= \overline{\hspace{7cm}}$$

This is a binary target variable in which 1 means the return rate is above the median (high risk) and 0 means the return rate is below the median (low risk).

In R, "set.seed(123)" means to set the starting point for the random number generator to the value 123, ensuring that any random numbers generated afterward in your code will be the same each time you run it, making your analysis more reproducible; essentially, it "seeds" the random number generator with a specific value to produce consistent results.

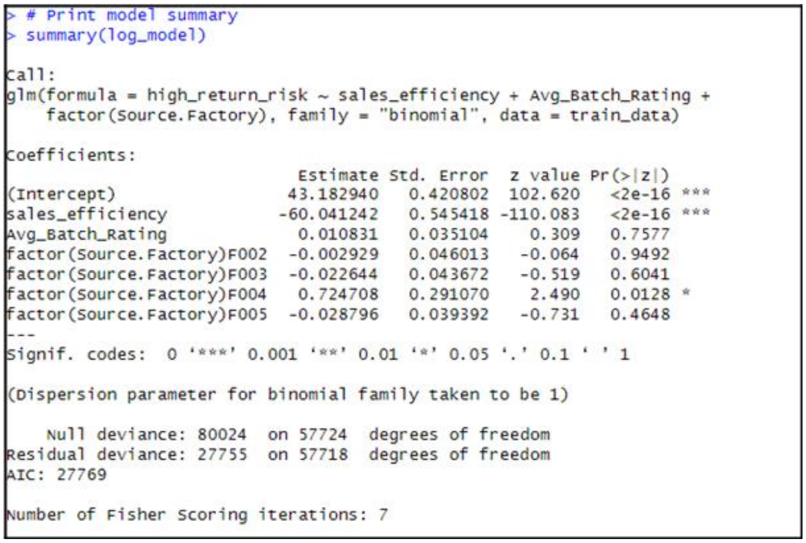
Splitting into 70% training and 30% testing data to ensure proportional representation of high_return_risk.

Logistic regression is suitable for binary classification problems.

Glm is a **Generalized Linear Model** to predict high_return_risk based on sales_efficiency, avg_batch_rating, and factor (Source.Factory) (treating this as a categorical variable) and family(binomial) to tell that the dependent variable is binary.

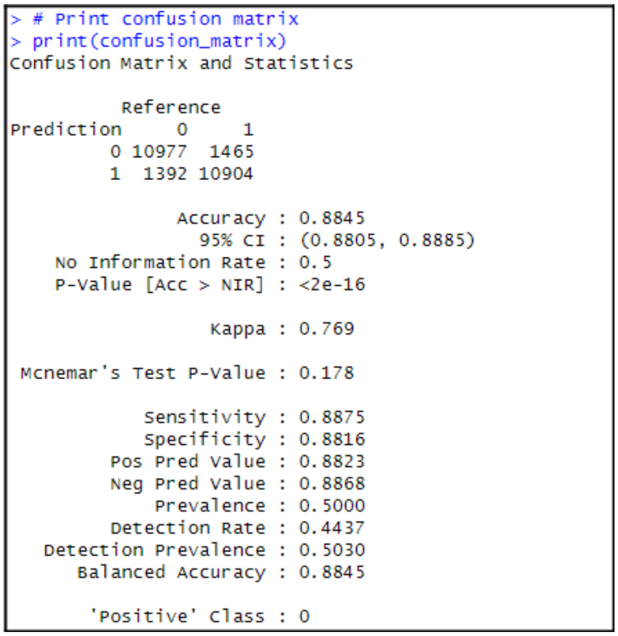
GLM provides the framework to handle the non-normal distribution of binary data in logistic regression analysis. We are assuming that the data is going to be non-normal.

Then a confusion matrix is created to obtain the outputs.



```
> # Print model summary
> summary(log_model)

Call:
glm(formula = high_return_risk ~ sales_efficiency + Avg_Batch_Rating +
    factor(Source.Factory), family = "binomial", data = train_data)

Coefficients:
                            Estimate Std. Error  z value Pr(>|z|)
(Intercept)                43.182940   0.420802  102.620   <2e-16 ***
sales_efficiency          -60.041242   0.545418 -110.083   <2e-16 ***
Avg_Batch_Rating            0.010831   0.035104    0.309   0.7577
factor(Source.Factory)F002 -0.002929   0.046013   -0.064   0.9492
factor(Source.Factory)F003 -0.022644   0.043672   -0.519   0.6041
factor(Source.Factory)F004  0.724708   0.291070    2.490   0.0128 *
factor(Source.Factory)F005 -0.028796   0.039392   -0.731   0.4648
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 80024  on 57724  degrees of freedom
Residual deviance: 27755  on 57718  degrees of freedom
AIC: 27769

Number of Fisher Scoring iterations: 7
```

Figure 8A: Summary model for the regression analysis



```
> # Print confusion matrix
> print(confusion_matrix)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10977  1465
         1  1392 10904

               Accuracy : 0.8845
                 95% CI : (0.8805, 0.8885)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.769

 Mcnemar's Test P-Value : 0.178

            Sensitivity : 0.8875
            Specificity : 0.8816
         Pos Pred Value : 0.8823
         Neg Pred Value : 0.8868
             Prevalence : 0.5000
         Detection Rate : 0.4437
   Detection Prevalence : 0.5030
      Balanced Accuracy : 0.8845

       'Positive' Class : 0
```

Figure 8B: Confusion Matrix analysis outputs

The next step is to generate an ROC Curve which is nothing but a Receiver Operating Characteristic curve, is a graphical representation of a classification model's performance across different classification thresholds, plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various cut-off points.

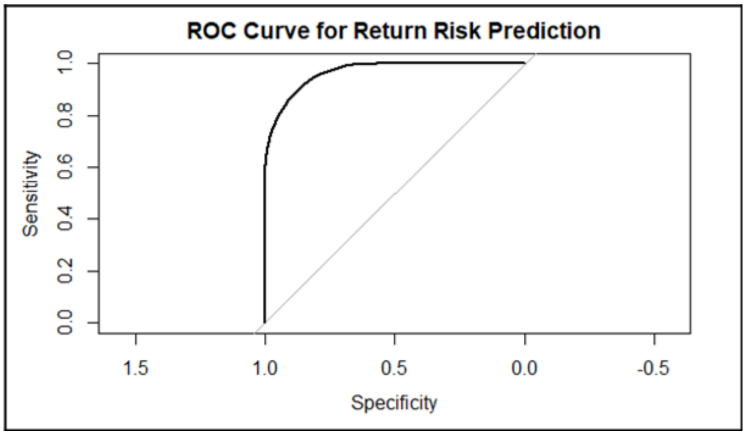


Figure 9: ROC Curve plot

The final step is to evaluate the model performance based on several characteristics.

Akaike Information Criterion (AIC) -The Akaike information criterion (AIC) is a mathematical method for evaluating how well a model fits the data it was generated from. In statistics, AIC is used to compare different possible models and determine which one is the best fit for the data.

Research Question 2 - Part 1) Using ANOVA

Are there specific product categories or factories that yield higher profit margins due to lower manufacturing costs or higher selling prices?

This paper analyzes the suitability of ANOVA for comparing profit margins across product categories and factories. While ANOVA offers advantages for multiple group comparisons, key assumption violations necessitate exploring alternative statistical methods for robust analysis of complex profit margin data.

Analysis of Variance in Profit Margin Research: Applications, Limitations, and Alternatives

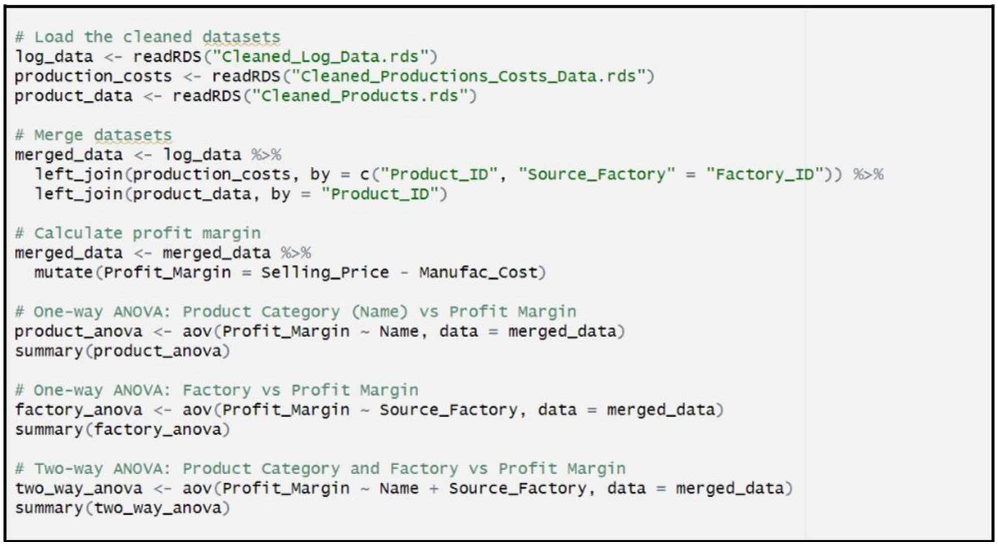
The analysis of profit margins across diverse product categories and manufacturing locations presents a complex challenge in business research. Analysis of Variance (ANOVA) is often employed as a primary statistical tool for such investigations. ANOVA is a common statistical tool usually employed in such investigations. This paper examines the appropriateness of ANOVA in this context, its potential limitations, and alternative approaches that may provide more robust analyses.

The Rationale for ANOVA Application

ANOVA's appeal for analyzing profit margins stems from its ability to simultaneously compare means across multiple product categories and factory locations. Profit margins' continuous nature aligns well with ANOVA's design, while the categorical nature of product type and factory location fits its framework. Two-way ANOVA, particularly useful with the large dataset (82,000+ observations), can reveal interaction effects between these factors, providing powerful insights into their joint influence on profit margins.

Limitations of ANOVA in Profit Margin Analysis

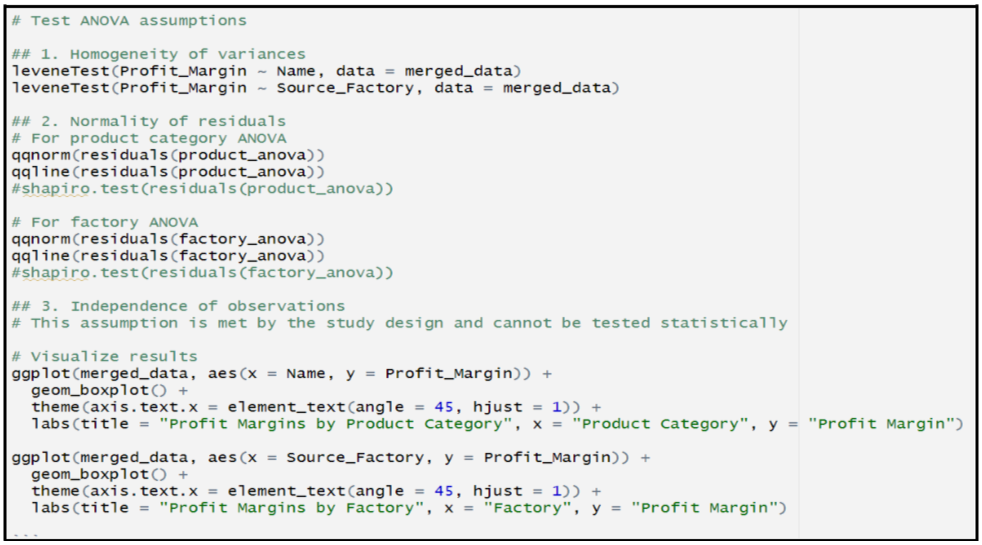
ANOVA's application is challenged by the violation of its homogeneity of variance assumption (Levene's test p < 2.2e-16), indicating unequal variability across groups and potentially biased results. Furthermore, the possibility of non-normal profit margins, outliers, and non-linear relationships further compromises the reliability of standard ANOVA.



```r
# Load the cleaned datasets
log_data <- readRDS("Cleaned_Log_Data.rds")
production_costs <- readRDS("Cleaned_Productions_Costs_Data.rds")
product_data <- readRDS("Cleaned_Products.rds")

# Merge datasets
merged_data <- log_data %>%
  left_join(production_costs, by = c("Product_ID", "Source_Factory" = "Factory_ID")) %>%
  left_join(product_data, by = "Product_ID")

# Calculate profit margin
merged_data <- merged_data %>%
  mutate(Profit_Margin = Selling_Price - Manufac_Cost)

# One-way ANOVA: Product Category (Name) vs Profit Margin
product_anova <- aov(Profit_Margin ~ Name, data = merged_data)
summary(product_anova)

# One-way ANOVA: Factory vs Profit Margin
factory_anova <- aov(Profit_Margin ~ Source_Factory, data = merged_data)
summary(factory_anova)

# Two-way ANOVA: Product Category and Factory vs Profit Margin
two_way_anova <- aov(Profit_Margin ~ Name + Source_Factory, data = merged_data)
summary(two_way_anova)
```

Figure 10: Code for ANOVA

Test ANOVA assumptions



```r
# Test ANOVA assumptions

## 1. Homogeneity of variances
leveneTest(Profit_Margin ~ Name, data = merged_data)
leveneTest(Profit_Margin ~ Source_Factory, data = merged_data)

## 2. Normality of residuals
# For product category ANOVA
qqnorm(residuals(product_anova))
qqline(residuals(product_anova))
#shapiro.test(residuals(product_anova))

# For factory ANOVA
qqnorm(residuals(factory_anova))
qqline(residuals(factory_anova))
#shapiro.test(residuals(factory_anova))

## 3. Independence of observations
# This assumption is met by the study design and cannot be tested statistically

# Visualize results
ggplot(merged_data, aes(x = Name, y = Profit_Margin)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Profit Margins by Product Category", x = "Product Category", y = "Profit Margin")

ggplot(merged_data, aes(x = Source_Factory, y = Profit_Margin)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Profit Margins by Factory", x = "Factory", y = "Profit Margin")
```

Figure 11 Code for ANOVA assumptions check

Output:

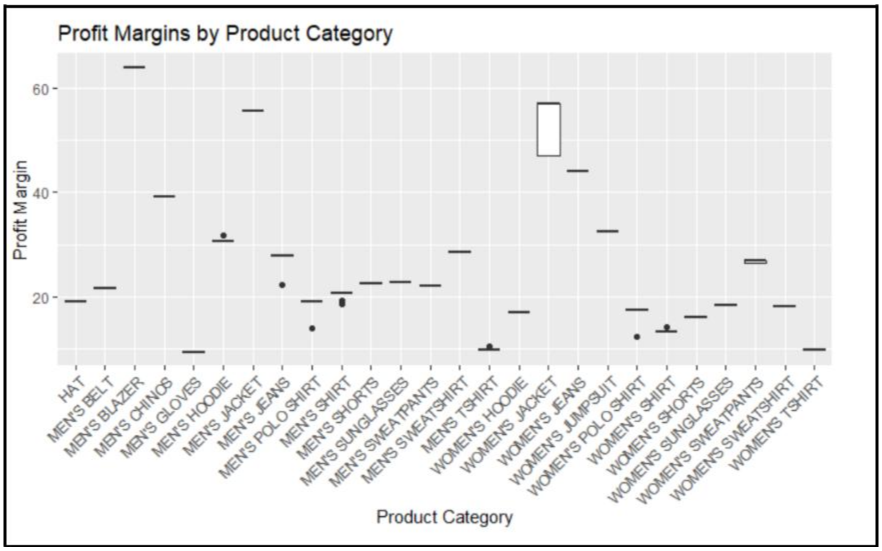


Figure 12: Output for profit margin by Product Category

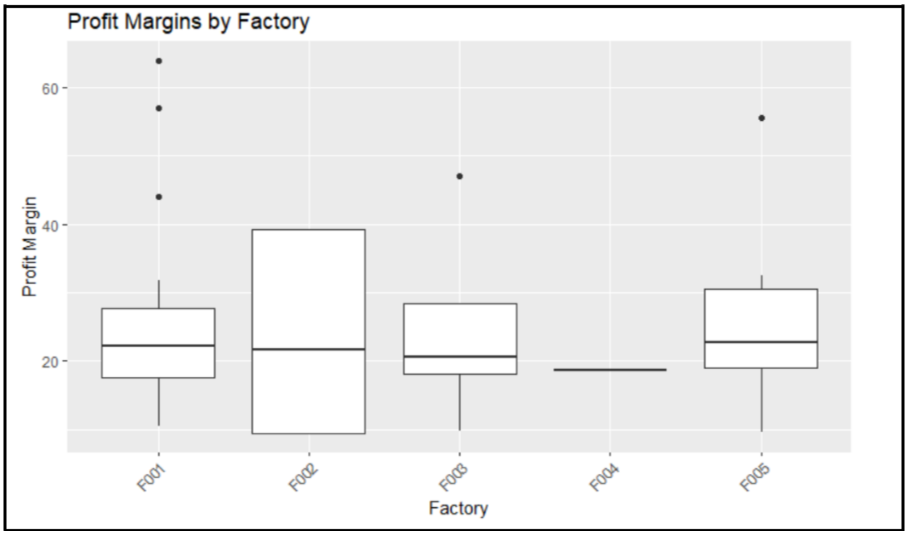
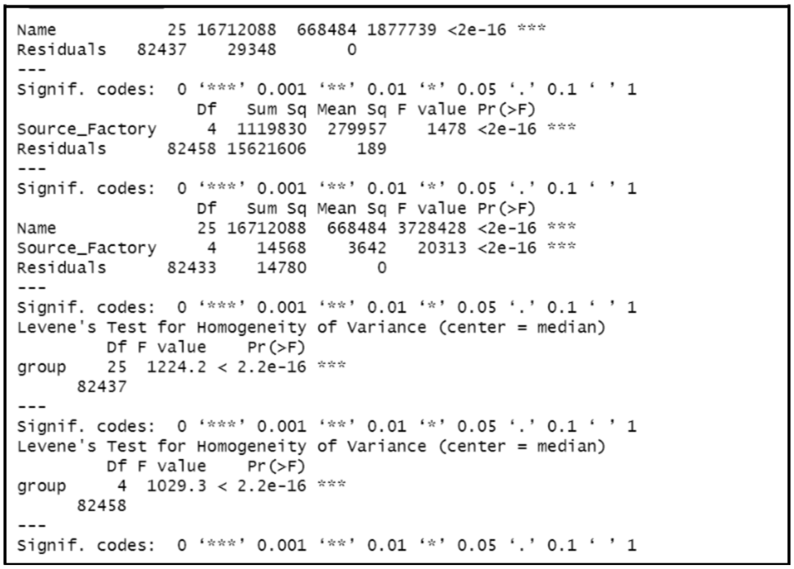


Figure 13: Output for profit margin by factory

Validating Initial Hypotheses: A Comparison of Proposal and Results

Two-way ANOVA showed highly significant (p < 2e-16) effects of both product category and factory location on profit margins, indicated by large F-statistics (3,728,428 and 20,313, respectively). Both factors independently influence profit margin variability.



```
Name           25 16712088  668484 1877739 <2e-16 ***
Residuals   82437    29348       0
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
                 Df   Sum Sq Mean Sq F value Pr(>F)
Source_Factory     4  1119830  279957    1478 <2e-16 ***
Residuals      82458 15621606     189
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
                 Df   Sum Sq Mean Sq F value Pr(>F)
Name              25 16712088  668484 3728428 <2e-16 ***
Source_Factory     4    14568    3642   20313 <2e-16 ***
Residuals      82433    14780       0
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Levene's Test for Homogeneity of Variance (center = median)
         Df F value    Pr(>F)
group    25  1224.2 < 2.2e-16 ***
      82437
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Levene's Test for Homogeneity of Variance (center = median)
         Df F value    Pr(>F)
group     4  1029.3 < 2.2e-16 ***
      82458
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 14: ANOVA assumption check result

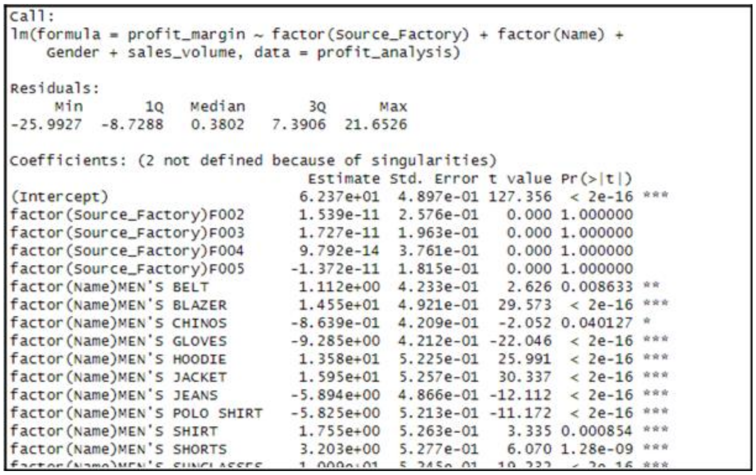
A critical violation of homogeneity of variance (Levene's test p < 2.2e-16) was detected, potentially affecting ANOVA reliability. While the large sample size (n > 82,000) mitigates normality concerns, strong F-statistics and low p-values still suggest significant effects of both product category and factory location on profit margins, particularly for the product category. Post-hoc tests (e.g., Games-Howell) are recommended. Future analysis should use methods accounting for unequal variances.

<u>Research Question 2- Part 2)modified to fit the model</u>

<u>Are there specific product categories or factories that yield higher profit margins due to lower manufacturing costs or higher selling prices?</u>

Solution - Multiple Linear Regression

To perform this regression model, we will first load the necessary libraries and then mutate the dataset as required to remove any null values using the function. Then the regression model can be made using the profit_margin variable which is a dependent variable, and the remaining ones such as Source_Factory, name, gender, and sales_volume. A summary of the profit model can be obtained as follows:



```
Call:
lm(formula = profit_margin ~ factor(Source_Factory) + factor(Name) +
    Gender + sales_volume, data = profit_analysis)

Residuals:
     Min       1Q   Median       3Q      Max
-25.9927  -8.7288   0.3802   7.3906  21.6526

Coefficients: (2 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.237e+01  4.897e-01 127.356  < 2e-16 ***
factor(Source_Factory)F002  1.539e-11  2.576e-01   0.000 1.000000
factor(Source_Factory)F003  1.727e-11  1.963e-01   0.000 1.000000
factor(Source_Factory)F004  9.792e-14  3.761e-01   0.000 1.000000
factor(Source_Factory)F005 -1.372e-11  1.815e-01   0.000 1.000000
factor(Name)MEN'S BELT      1.112e+00  4.233e-01   2.626 0.008633 **
factor(Name)MEN'S BLAZER    1.455e+01  4.921e-01  29.573  < 2e-16 ***
factor(Name)MEN'S CHINOS   -8.639e-01  4.209e-01  -2.052 0.040127 *
factor(Name)MEN'S GLOVES   -9.285e+00  4.212e-01 -22.046  < 2e-16 ***
factor(Name)MEN'S HOODIE    1.358e+01  5.225e-01  25.991  < 2e-16 ***
factor(Name)MEN'S JACKET    1.595e+01  5.257e-01  30.337  < 2e-16 ***
factor(Name)MEN'S JEANS    -5.894e+00  4.866e-01 -12.112  < 2e-16 ***
factor(Name)MEN'S POLO SHIRT -5.825e+00  5.213e-01 -11.172  < 2e-16 ***
factor(Name)MEN'S SHIRT     1.755e+00  5.263e-01   3.335 0.000854 ***
factor(Name)MEN'S SHORTS    3.203e+00  5.277e-01   6.070 1.28e-09 ***
factor(Name)MEN'S SUNGLASSES 1.000e+01  5.245e-01  19.232  < 2e-16 ***
```

Figure 15: Summary model for the linear regression analysis

Estimate represents the best guess for a population parameter based on data from a sample.

Std. Error: A measure of how much error is expected in the estimated value, essentially the standard deviation of the sampling distribution.

t value: The t-value, or t-score, is a ratio of the difference between the mean of the two sample sets and the variation that exists within the sample sets.

Pr(>|t|): The probability of observing data as extreme as the observed data if the null hypothesis is true; a small p-value indicates evidence against the null hypothesis.

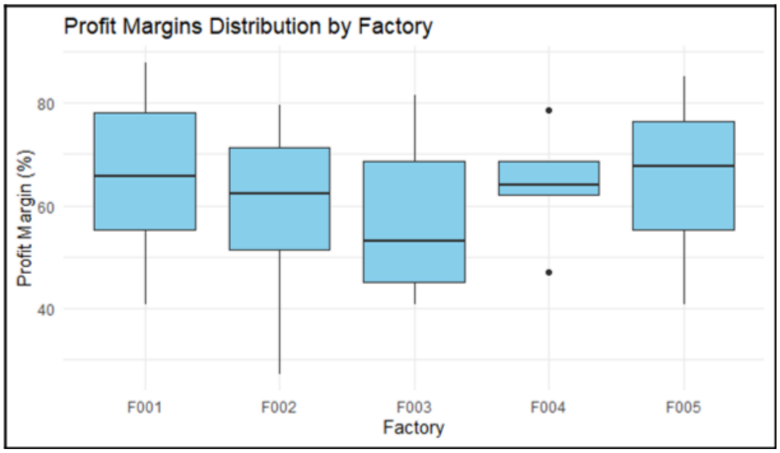
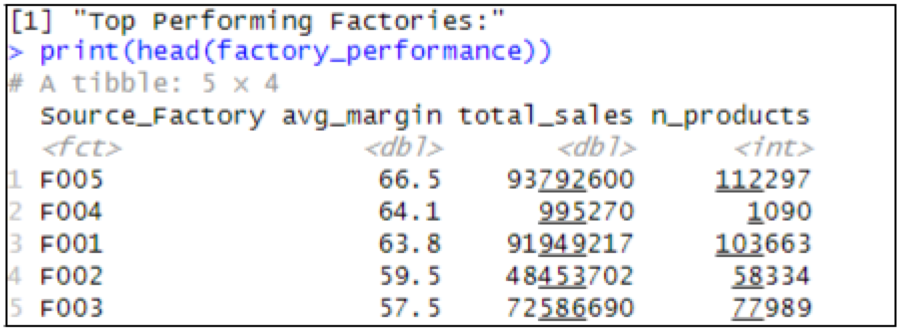


Figure 16: Vertical boxplot for factories by profit margin



```
[1] "Top Performing Factories:"
> print(head(factory_performance))
# A tibble: 5 x 4
  Source_Factory avg_margin total_sales n_products
  <fct>               <dbl>       <dbl>      <int>
1 F005                 66.5    93792600     112297
2 F004                 64.1      995270       1090
3 F001                 63.8    91949217     103663
4 F002                 59.5    48453702      58334
5 F003                 57.5    72586690      77989
```

Figure 17: Tabular data showing Factories analysis

This visualization is a vertical boxplot showing the factories and their profit margins as obtained on the graph.

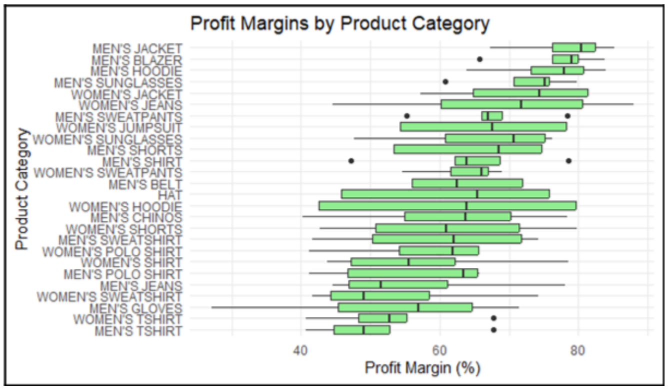
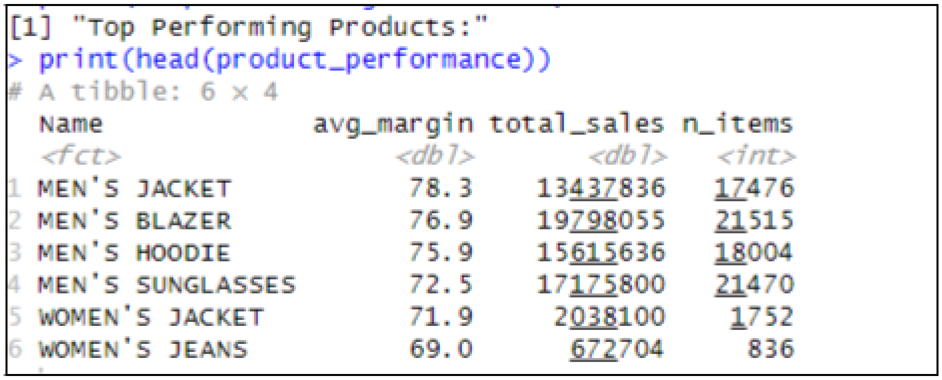


Figure 18: Horizontal boxplot of Products by profit margin



```
[1] "Top Performing Products:"
> print(head(product_performance))
# A tibble: 6 x 4
  Name             avg_margin total_sales n_items
  <fct>                 <dbl>       <dbl>   <int>
1 MEN'S JACKET           78.3    13437836   17476
2 MEN'S BLAZER           76.9    19798055   21515
3 MEN'S HOODIE           75.9    15615636   18004
4 MEN'S SUNGLASSES       72.5    17175800   21470
5 WOMEN'S JACKET         71.9     2038100    1752
6 WOMEN'S JEANS          69.0      672704     836
```

Figure 19: Tabular data showing Product analysis

The horizontal box plot shows the products in a format describing all the profit margins of the product along with a result that has been obtained in the console.

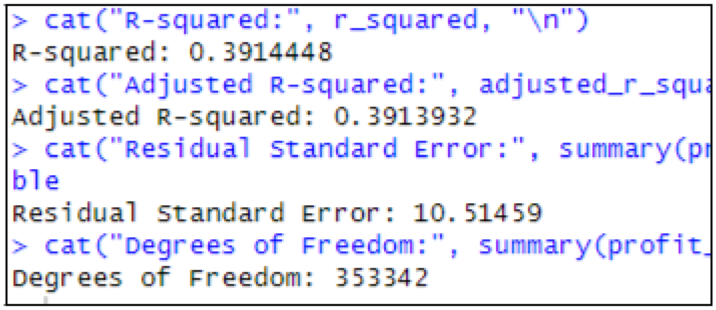
The code calculates performance metrics for factories and products:

factory_performance: Grouped by Source_Factory, it calculates the average margin, total sales, and number of products for each factory.

product_performance: Grouped by Name (product category), it calculates similar metrics for each product type.

Both summaries are arranged in descending order of average margin.

Finally, the code prints the top-performing factories and products based on the calculated metrics.



```
> cat("R-squared:", r_squared, "\n")
R-squared: 0.3914448
> cat("Adjusted R-squared:", adjusted_r_squa
Adjusted R-squared: 0.3913932
> cat("Residual Standard Error:", summary(pr
ble
Residual Standard Error: 10.51459
> cat("Degrees of Freedom:", summary(profit_
Degrees of Freedom: 353342
```

Figure 20: Various statistical analyses executed to evaluate metrics performance

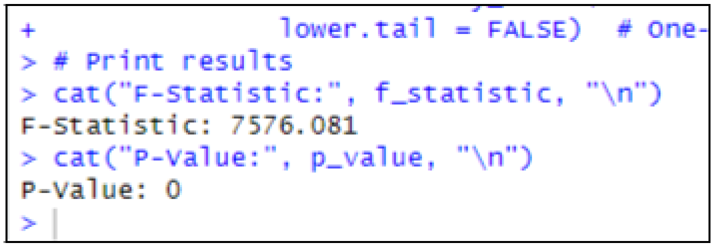
The value of the obtained parameters should be as follows :

R-squared should be close to 1.

Adjusted R-squared should be closer to 1.

The residual standard error should be as small as possible.

The degree of freedom can be anything greater than 30.



```
+                lower.tail = FALSE)  # One-
> # Print results
> cat("F-Statistic:", f_statistic, "\n")
F-Statistic: 7576.081
> cat("P-Value:", p_value, "\n")
P-Value: 0
> 
```

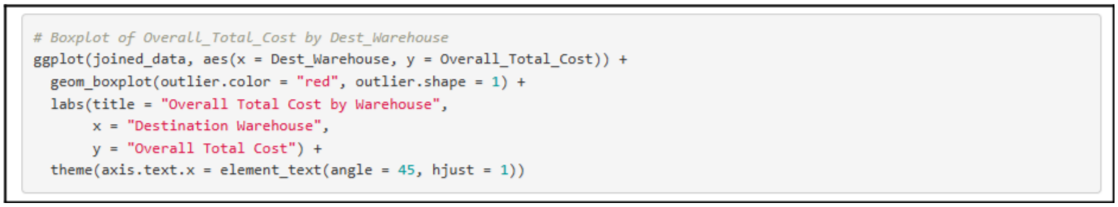
Figure 21: Various statistical analyses executed to evaluate metrics

performance F statistic should be relatively large.

p-value should be relatively smaller, typically less than 0.05 so that at least one of the independent variables significantly contributes to explaining the variation in the dependent variable.

Research Question 3: How do specific operational decisions (e.g., timing of shipments, and warehouse selection) influence the total cost incurred for production and shipping?

Step 1: Exploratory data analysis



```
# Boxplot of Overall_Total_Cost by Dest_Warehouse
ggplot(joined_data, aes(x = Dest_Warehouse, y = Overall_Total_Cost)) +
  geom_boxplot(outlier.color = "red", outlier.shape = 1) +
  labs(title = "Overall Total Cost by Warehouse",
       x = "Destination Warehouse",
       y = "Overall Total Cost") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Figure 22:Code for boxplot of Overall Total Cost vs Destination Warehouse

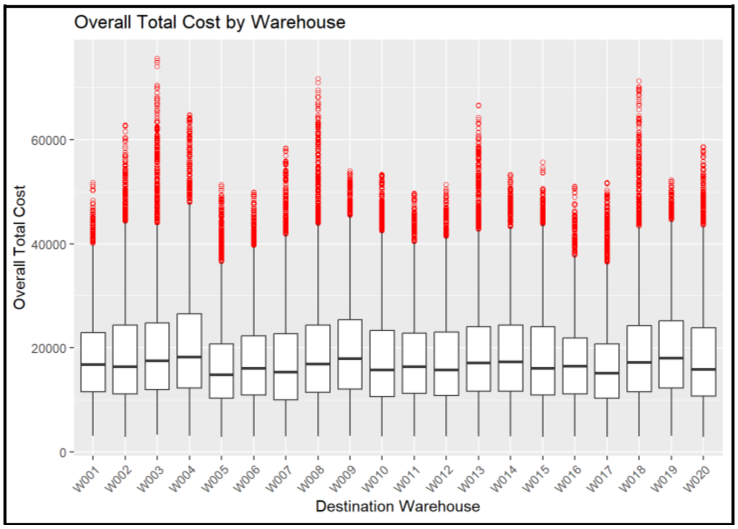
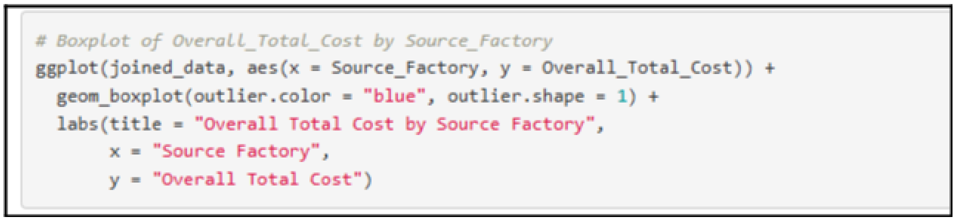


Figure 23: Boxplot output of Overall Total Cost by Warehouse

As shown in Figure 22, the code creates a boxplot to visualize the distribution of the Overall_Total_Cost variable for each Dest_Warehouse in the dataset joined_data. It highlights outliers in red with a specific shape and adds labels for the plot title and axes. The x-axis labels are rotated by 45 degrees for better readability.

The boxplot illustrated in Figure 23, gives the distribution of production and shipping expenses across 20 warehouses, with the whiskers indicating the cost range within 1.5 times the interquartile range (IQR). Warehouses such as W004, W009, and W020 present high-cost outliers that extend beyond the whiskers, signifying potential inefficiencies. In contrast, the majority of warehouses maintain stable costs within the IQR. Warehouses W005 and W007 demonstrate lower costs and a reduced number of outliers, implying greater efficiency. These findings can assist in pinpointing areas for enhancement in shipping, labor, and warehouse operations to improve cost management.



```
# Boxplot of Overall_Total_Cost by Source_Factory
ggplot(joined_data, aes(x = Source_Factory, y = Overall_Total_Cost)) +
  geom_boxplot(outlier.color = "blue", outlier.shape = 1) +
  labs(title = "Overall Total Cost by Source Factory",
       x = "Source Factory",
       y = "Overall Total Cost")
```

Figure 24:Code for Boxplot of Overall Total Cost vs Source Factory

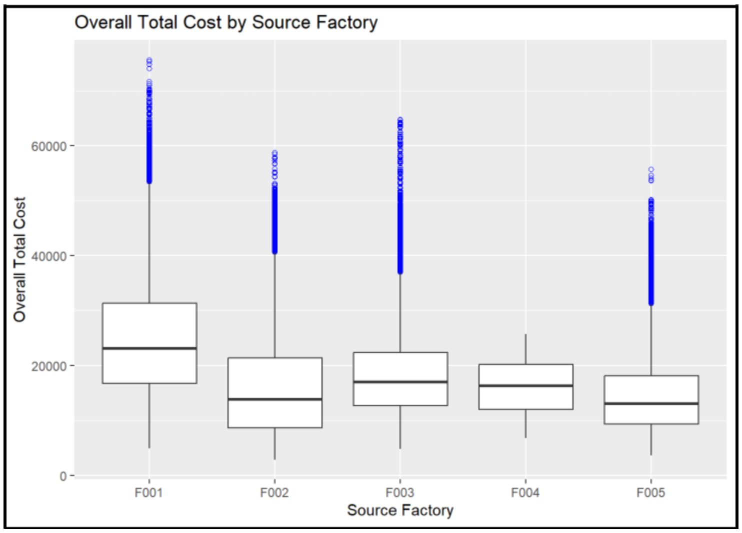
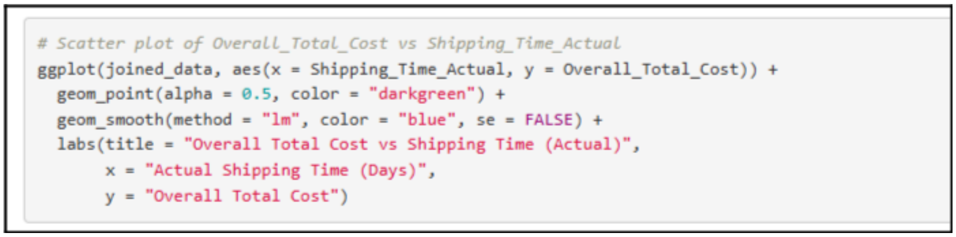


Figure 25: Boxplot of Overall Total Cost by Source Factory

As shown in Figure 24, the code generates a boxplot to illustrate the distribution of Overall_Total_Cost among various source factories. It emphasizes outliers in blue and presents the spread, median, and interquartile range (IQR) of costs for each factory. This visualization facilitates the comparison of cost variability and aids in identifying potential inefficiencies.

Figure 25 illustrates that F001 exhibits the highest median cost along with considerable variability, accompanied by several outliers, which indicates potential inconsistencies in cost management practices. Conversely, F002 presents the lowest median cost and demonstrates stable expenses, with a reduced number of outliers, reflecting superior cost control. F003 and F005 reveal moderate median costs but are characterized by a greater number of outliers, suggesting sporadic increases in expenditures. In contrast, F004 maintains a consistent cost profile with minimal fluctuations and few outliers, signifying effective cost management.



```
# Scatter plot of Overall_Total_Cost vs Shipping_Time_Actual
ggplot(joined_data, aes(x = Shipping_Time_Actual, y = Overall_Total_Cost)) +
  geom_point(alpha = 0.5, color = "darkgreen") +
  geom_smooth(method = "lm", color = "blue", se = FALSE) +
  labs(title = "Overall Total Cost vs Shipping Time (Actual)",
       x = "Actual Shipping Time (Days)",
       y = "Overall Total Cost")
```

Figure 26: Code for Scatter plot of Overall Total Cost vs Shipping time

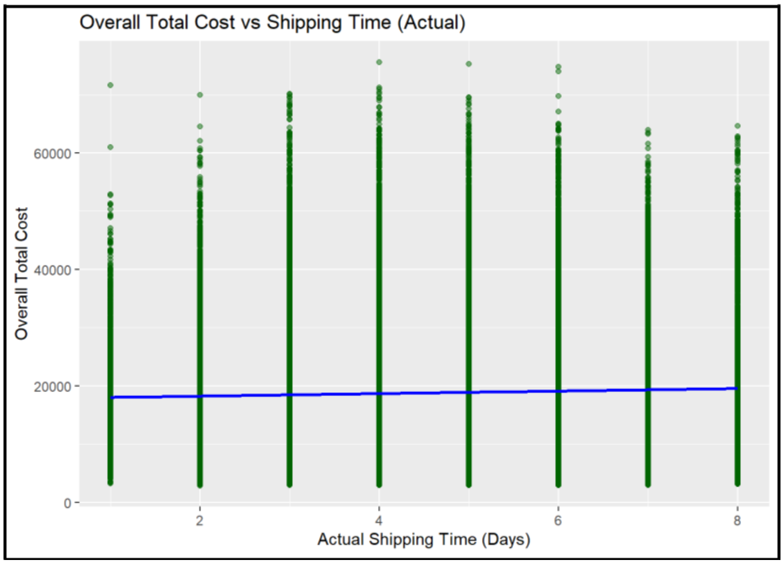
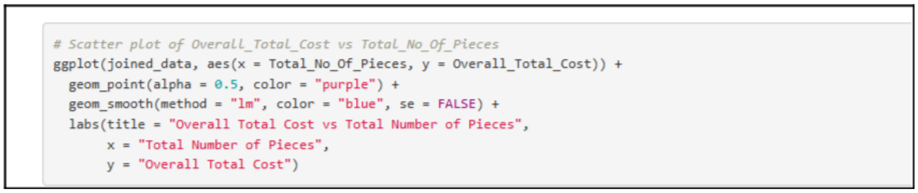


Figure 27:Scatter plot of Overall Total Cost by Shipping time

Figure 26 illustrates a scatter plot that depicts the correlation between Actual Shipping Time (Days) on the x-axis and Overall Total Cost on the y-axis. A blue trend line has been incorporated to represent the overall trend of the data, facilitating the identification of any potential correlations or patterns between these two variables.

The scatter plot presented in Figure 27 does not demonstrate a distinct relationship between shipping time and overall total cost, as evidenced by the horizontal trend line. While the majority of data points are concentrated below 20,000, indicating relatively stable costs, there are notable outliers across various shipping durations, with costs exceeding 60,000. These outliers appear regardless of the shipping time, suggesting the influence of additional factors on elevated costs. The data indicates a uniform distribution of costs, with shipping time exerting minimal influence; however, the presence of high-cost outliers necessitates further analysis to uncover potential inefficiencies.



```r
# Scatter plot of Overall_Total_Cost vs Total_No_Of_Pieces
ggplot(joined_data, aes(x = Total_No_Of_Pieces, y = Overall_Total_Cost)) +
  geom_point(alpha = 0.5, color = "purple") +
  geom_smooth(method = "lm", color = "blue", se = FALSE) +
  labs(title = "Overall Total Cost vs Total Number of Pieces",
       x = "Total Number of Pieces",
       y = "Overall Total Cost")
```

Figure 28: Code for Scatter plot of Overall Total Cost vs Total number of Pieces

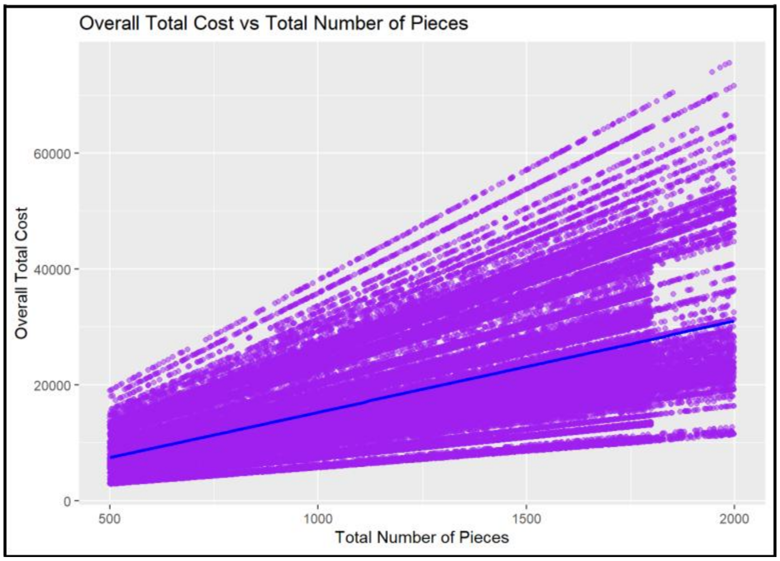
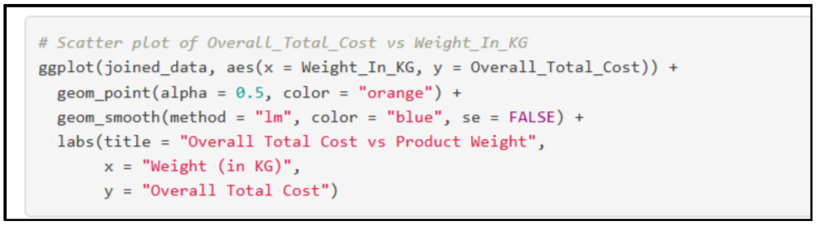


Figure 29: Scatter plot of Overall Total Cost vs Total number of pieces

The scatter plot presented in Figure 28 illustrates the relationship between the Total Number of Pieces (x-axis) and Overall Total Cost (y-axis). The data points are represented in purple, while a blue linear regression line is superimposed to emphasize the trend. This line serves as a fitted model to enhance the visualization of any potential correlation.

As depicted in Figure 29, the scatter plot indicates a slight upward trend, implying a positive correlation between the total number of pieces produced and the overall cost. An increase in the number of pieces corresponds with a rise in overall costs, suggesting that larger production volumes lead to higher expenses. This observation is reinforced by the blue regression line, which indicates that greater production volumes are typically linked to increased costs, likely due to higher resource consumption and logistical demands.



```
# Scatter plot of Overall_Total_Cost vs Weight_In_KG
ggplot(joined_data, aes(x = Weight_In_KG, y = Overall_Total_Cost)) +
  geom_point(alpha = 0.5, color = "orange") +
  geom_smooth(method = "lm", color = "blue", se = FALSE) +
  labs(title = "Overall Total Cost vs Product Weight",
       x = "Weight (in KG)",
       y = "Overall Total Cost")
```

Figure 30: Code for Scatter plot of Overall Total Cost vs Weight

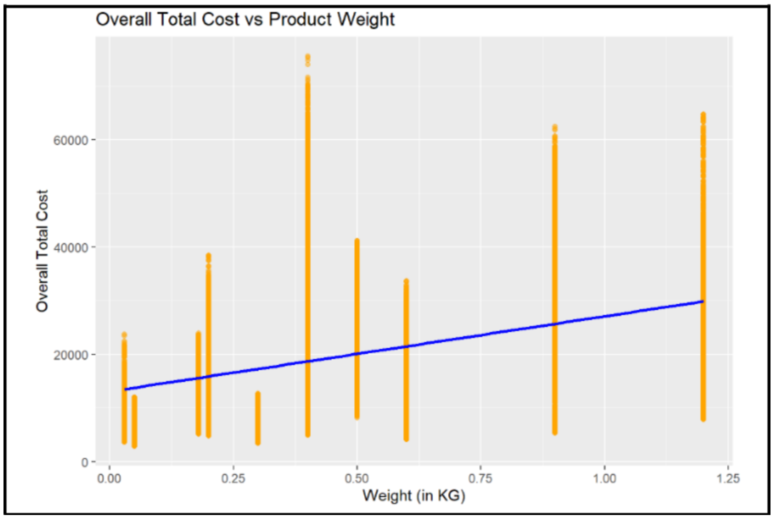
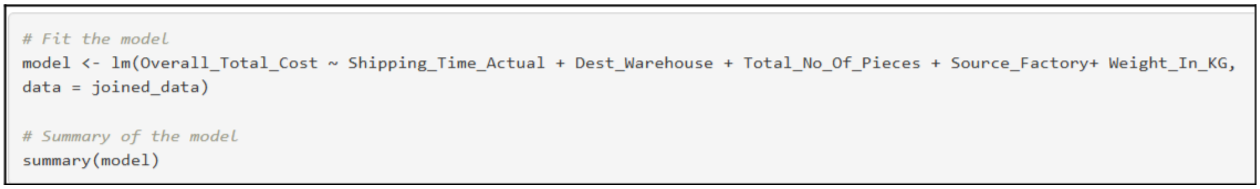


Figure 31: Scatter plot of Overall Total Cost By Weight

The scatter plot illustrated in Figure 30 employs `ggplot` to depict the relationship between the total number of items produced and the corresponding total cost incurred. Individual data points are represented by purple markers, while a blue regression line conveys the overall trend between these two variables.

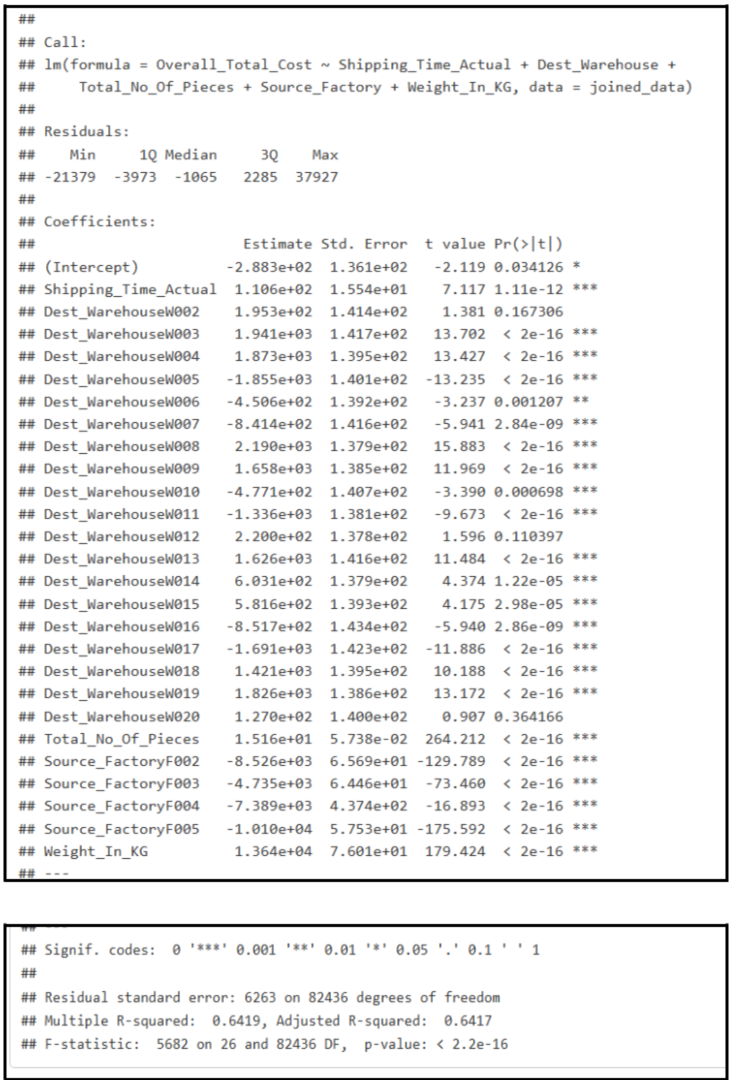
The scatter plot presented in Figure 31 demonstrates a positive relationship between production volume and total cost, suggesting that an increase in the number of items produced correlates with a rise in overall expenses. This pattern indicates that enhancing production levels results in higher costs, likely attributable to greater resource utilization, logistics demands, and labor requirements. Although the majority of data points conform to the regression line, some outliers may arise from operational inefficiencies or discrepancies in shipping methods. To effectively manage costs, it is essential to optimize production scaling, resource distribution, and logistics to mitigate cost increases.

Step 2: Machine learning model- Multiple linear regression



```
# Fit the model
model <- lm(Overall_Total_Cost ~ Shipping_Time_Actual + Dest_Warehouse + Total_No_Of_Pieces + Source_Factory+ Weight_In_KG,
data = joined_data)

# Summary of the model
summary(model)
```

Figure 32: Code for Predicting Overall_Total_Cost with a Multiple Linear Regression Model



```
##
## Call:
## lm(formula = Overall_Total_Cost ~ Shipping_Time_Actual + Dest_Warehouse +
##     Total_No_Of_Pieces + Source_Factory + Weight_In_KG, data = joined_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21379  -3973  -1065   2285  37927
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)          -2.883e+02  1.361e+02   -2.119 0.034126 *
## Shipping_Time_Actual  1.106e+02  1.554e+01    7.117 1.11e-12 ***
## Dest_WarehouseW002    1.953e+02  1.414e+02    1.381 0.167306
## Dest_WarehouseW003    1.941e+03  1.417e+02   13.702  < 2e-16 ***
## Dest_WarehouseW004    1.873e+03  1.395e+02   13.427  < 2e-16 ***
## Dest_WarehouseW005   -1.855e+03  1.401e+02  -13.235  < 2e-16 ***
## Dest_WarehouseW006   -4.506e+02  1.392e+02   -3.237 0.001207 **
## Dest_WarehouseW007   -8.414e+02  1.416e+02   -5.941 2.84e-09 ***
## Dest_WarehouseW008    2.190e+03  1.379e+02   15.883  < 2e-16 ***
## Dest_WarehouseW009    1.658e+03  1.385e+02   11.969  < 2e-16 ***
## Dest_WarehouseW010   -4.771e+02  1.407e+02   -3.390 0.000698 ***
## Dest_WarehouseW011   -1.336e+03  1.381e+02   -9.673  < 2e-16 ***
## Dest_WarehouseW012    2.200e+02  1.378e+02    1.596 0.110397
## Dest_WarehouseW013    1.626e+03  1.416e+02   11.484  < 2e-16 ***
## Dest_WarehouseW014    6.031e+02  1.379e+02    4.374 1.22e-05 ***
## Dest_WarehouseW015    5.816e+02  1.393e+02    4.175 2.98e-05 ***
## Dest_WarehouseW016   -8.517e+02  1.434e+02   -5.940 2.86e-09 ***
## Dest_WarehouseW017   -1.691e+03  1.423e+02  -11.886  < 2e-16 ***
## Dest_WarehouseW018    1.421e+03  1.395e+02   10.188  < 2e-16 ***
## Dest_WarehouseW019    1.826e+03  1.386e+02   13.172  < 2e-16 ***
## Dest_WarehouseW020    1.270e+02  1.400e+02    0.907 0.364166
## Total_No_Of_Pieces    1.516e+01  5.738e-02  264.212  < 2e-16 ***
## Source_FactoryF002   -8.526e+03  6.569e+01 -129.789  < 2e-16 ***
## Source_FactoryF003   -4.735e+03  6.446e+01  -73.460  < 2e-16 ***
## Source_FactoryF004   -7.389e+03  4.374e+02  -16.893  < 2e-16 ***
## Source_FactoryF005   -1.010e+04  5.753e+01 -175.592  < 2e-16 ***
## Weight_In_KG          1.364e+04  7.601e+01  179.424  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6263 on 82436 degrees of freedom
## Multiple R-squared:  0.6419, Adjusted R-squared:  0.6417
## F-statistic:  5682 on 26 and 82436 DF,  p-value: < 2.2e-16
```

Figure 33: Regression analysis

The code illustrated in Figure 32 establishes a multiple linear regression model aimed at forecasting the `**Overall_Total_Cost**` by utilizing predictors such as `**Shipping_Time_Actual**`, `**Dest_Warehouse**`, `**Total_No_Of_Pieces**`, `**Source_Factory**`, and `**Weight_In_KG**`. The model is constructed using the `**lm()**` function, while the `**summary()**` function delivers a comprehensive output that includes coefficients, standard errors, t-values, and p-values for each of the predictors.

The regression analysis presented in Figure 33 indicates that significant variables, including `**Total_No_Of_Pieces**` (coefficient = 15.16) and `**Weight_In_KG**` (coefficient = 13640), have a notable influence on the overall cost. The R-squared value of the model is 0.6419, suggesting that 64.19% of the variability in `**Overall_Total_Cost**` can be accounted for by the predictors. Furthermore, certain destination warehouses (for instance, `**Dest_WarehouseW003**` with a coefficient of 1941) and source factories (such as `**Source_FactoryF002**` with a coefficient of

-8526) demonstrate significant impacts on cost. The F-statistic is recorded at 5682 with a p-value less than 2. 2e-16, which affirms the overall statistical significance of the model.

Summary Analysis

Model Overview

The **Multiple linear regression** model assesses the total cost by considering various factors such as shipping duration, choice of warehouse, quantity of items, originating factory, and weight of the product.

Key Findings

Shipping Time: Delays greatly raise expenses (p = 1.11e-12).

Warehouse Selection: W003/W004 increases expenses, W005 decreases them, and W002 has no impact.

Total Items: Highly significant (p < 2e-16); an increase in items raises expenses.

Origin Factory: F002-F005 reduces expenses through negative coefficients.

Product Weight: Heavier items greatly increase expenses (p < 2e-16).

Fit Metrics

R-squared (0.6419): Accounts for 64% of the variability in costs.

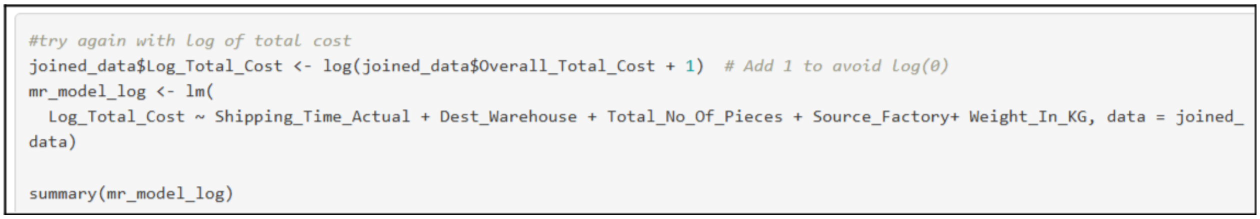
Adjusted R-squared (0.6417): Strong fit with no overfitting.

Conclusion

To reduce overall costs, it is essential to address shipping delays, optimize warehouse selection, and manage product weight effectively. Shipping duration and product weight are identified as key factors influencing costs.

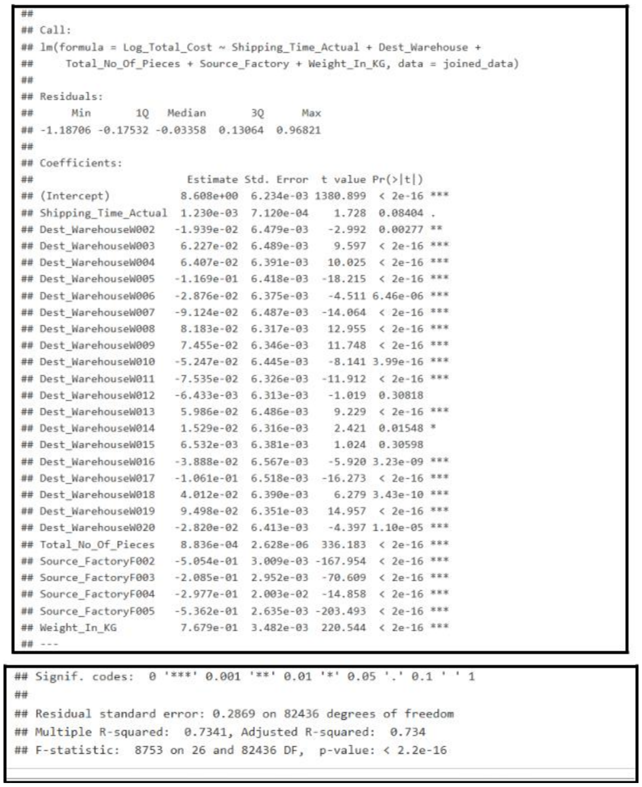
Model with Log-Transformed Total Cost

To correct the uneven distribution of the Overall Total Cost, we implemented a log transformation, Log_Total_Cost = log(Overall_Total_Cost + 1), to standardize the data, stabilize variance, and enhance model fit. This modification lessens the influence of outliers and improves the accuracy and interpretability of the regression model. The revised model enhances the understanding of connections among elements such as shipping duration, warehouse choice, product mass, and quantity of items, yielding more dependable forecasts and actionable recommendations.



```
#try again with log of total cost
joined_data$Log_Total_Cost <- log(joined_data$Overall_Total_Cost + 1)  # Add 1 to avoid log(0)
mr_model_log <- lm(
  Log_Total_Cost ~ Shipping_Time_Actual + Dest_Warehouse + Total_No_Of_Pieces + Source_Factory+ Weight_In_KG, data = joined_
data)

summary(mr_model_log)
```

Figure 34: Code for Predicting Log_Total_Cost with a Multiple Linear Regression Model



```
##
## Call:
## lm(formula = Log_Total_Cost ~ Shipping_Time_Actual + Dest_Warehouse +
##     Total_No_Of_Pieces + Source_Factory + Weight_In_KG, data = joined_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.18706 -0.17532 -0.03358  0.13064  0.96821
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           8.608e+00  6.234e-03 1380.899  < 2e-16 ***
## Shipping_Time_Actual  1.230e-03  7.120e-04    1.728  0.08404 .
## Dest_WarehouseW002   -1.939e-02  6.479e-03   -2.992  0.00277 **
## Dest_WarehouseW003    6.227e-02  6.489e-03    9.597  < 2e-16 ***
## Dest_WarehouseW004    6.407e-02  6.391e-03   10.025  < 2e-16 ***
## Dest_WarehouseW005   -1.169e-01  6.418e-03  -18.215  < 2e-16 ***
## Dest_WarehouseW006   -2.876e-02  6.375e-03   -4.511 6.46e-06 ***
## Dest_WarehouseW007   -9.124e-02  6.487e-03  -14.064  < 2e-16 ***
## Dest_WarehouseW008    8.183e-02  6.317e-03   12.955  < 2e-16 ***
## Dest_WarehouseW009    7.455e-02  6.346e-03   11.748  < 2e-16 ***
## Dest_WarehouseW010   -5.247e-02  6.445e-03   -8.141 3.99e-16 ***
## Dest_WarehouseW011   -7.535e-02  6.326e-03  -11.912  < 2e-16 ***
## Dest_WarehouseW012   -6.433e-03  6.313e-03   -1.019  0.30818
## Dest_WarehouseW013    5.986e-02  6.486e-03    9.229  < 2e-16 ***
## Dest_WarehouseW014    1.529e-02  6.316e-03    2.421  0.01548 *
## Dest_WarehouseW015    6.532e-03  6.381e-03    1.024  0.30598
## Dest_WarehouseW016   -3.888e-02  6.567e-03   -5.920 3.23e-09 ***
## Dest_WarehouseW017   -1.061e-01  6.518e-03  -16.273  < 2e-16 ***
## Dest_WarehouseW018    4.012e-02  6.390e-03    6.279 3.43e-10 ***
## Dest_WarehouseW019    9.498e-02  6.351e-03   14.957  < 2e-16 ***
## Dest_WarehouseW020   -2.820e-02  6.413e-03   -4.397 1.10e-05 ***
## Total_No_Of_Pieces    8.836e-04  2.628e-06  336.183  < 2e-16 ***
## Source_FactoryF002   -5.054e-01  3.009e-03 -167.954  < 2e-16 ***
## Source_FactoryF003   -2.085e-01  2.952e-03  -70.609  < 2e-16 ***
## Source_FactoryF004   -2.977e-01  2.003e-02  -14.858  < 2e-16 ***
## Source_FactoryF005   -5.362e-01  2.635e-03 -203.493  < 2e-16 ***
## Weight_In_KG          7.679e-01  3.482e-03  220.544  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2869 on 82436 degrees of freedom
## Multiple R-squared:  0.7341, Adjusted R-squared:  0.734
## F-statistic:  8753 on 26 and 82436 DF,  p-value: < 2.2e-16
```

Figure 35: Output of Log-Transformed Total Cost regression model

Coefficients and Significance

Intercept: Baseline log-transformed cost is 8.608 when predictors are zero.

Shipping Time: Coefficient 0.00123 (p = 0.084) shows marginally significant cost increases with delays.

Destination Warehouse: Warehouse W003 raises costs (0.06227), while W005 lowers them (-0.1169).

Total Pieces: Coefficient 15.16 (p < 2e-16) strongly links higher piece counts to increased costs.

Source Factory: Factories F002 and F005 significantly reduce costs due to efficiencies.

Weight: Coefficient 13.64e+03 (p < 2e-16) confirms higher weight increases costs**.**

Statistical Significance

Most variables, including weight, shipping time, and total pieces, significantly impact costs. Adjusted R^2 (0.6417) indicates 64% variance explained.

Conclusion

The analysis emphasizes essential elements affecting overall expenses in the supply chain, where weight and shipment volume emerge as the most significant cost drivers, succeeded by warehouse choice and shipping delays. Adjusting these variables can greatly lower expenses, boost operational effectiveness, and increase customer satisfaction. By concentrating on enhancing warehouse efficiency, overseeing shipment dimensions and weight, and tackling shipping delays, companies can attain improved cost management and more efficient supply chain operations, leading to increased profitability and competitiveness.

Future Research Directions: Improving the Fast Fashion Supply Chain Analysis

This data-driven analysis of a fast-fashion supply chain examined quality control, resource allocation, and logistics optimization. Findings revealed correlations between factory performance and return rates, suggesting targeted quality control measures. Profit margin analysis highlighted the importance of prioritizing high-margin product categories and factories. Logistics optimization requires strategic warehouse selection (W005 and W007 showed lower costs) and consideration of product weight. Continuous data collection and analysis are crucial for ongoing improvement. Future research should address ANOVA assumption violations and explore more sophisticated modeling techniques.