LDA, QDA and Factor Analysis on Sales Product Details

Sahil Jadhav – L035

19-02-2026

Load Dataset

```
superstore <- read.csv("C:/Users/Sahil Jadhav/OneDrive/문서/SampleSuperstore.csv",stringsAsFactors = TRUE)
head(superstore)
```

Description: df [6 x 13]

	Ship.Mode <fctr>	Segment <fctr>	Country <fctr>	City <fctr>	State <fctr>	Postal.Code <int>	Region <fctr>	Category <fctr>
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture
2	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture
3	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture
5	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies
6	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Furniture

6 rows | 1-9 of 13 columns

```
cat("--- Dataset Dimensions ---\n")
cat("Rows:", nrow(superstore), "| Columns:", ncol(superstore), "\n\n")
```

```
cat("\n--- Summary Statistics ---\n")
print(summary(superstore[, c("Sales", "Profit", "Quantity", "Discount")]))
```

```
cat("\n--- Missing Values Per Column ---\n")
print(colSums(is.na(superstore)))
```

```
--- Dataset Dimensions ---
Rows: 9994 | Columns: 13


--- Summary Statistics ---
     Sales              Profit            Quantity       Discount
 Min.   :    0.444   Min.   :-6599.978   Min.   : 1.00   Min.   :0.0000
 1st Qu.:   17.280   1st Qu.:    1.729   1st Qu.: 2.00   1st Qu.:0.0000
 Median :   54.490   Median :    8.666   Median : 3.00   Median :0.2000
 Mean   :  229.858   Mean   :   28.657   Mean   : 3.79   Mean   :0.1562
 3rd Qu.:  209.940   3rd Qu.:   29.364   3rd Qu.: 5.00   3rd Qu.:0.2000
 Max.   :22638.480   Max.   : 8399.976   Max.   :14.00   Max.   :0.8000

--- Missing Values Per Column ---
   Ship.Mode      Segment      Country         City        State  Postal.Code       Region     Category Sub.Category        Sales
           0            0            0            0            0            0            0            0            0            0
    Quantity     Discount       Profit
           0            0            0
```

CREATE BINARY PROFIT CLASS (DEPENDENT VARIABLE)

```
# Calculate the median of Profit
profit_median <- median(superstore$Profit)
cat("\nMedian Profit:", profit_median, "\n")

# Create binary variable:
#   "High"  → Profit >= median
#   "Low"   → Profit <  median


superstore$Profit_Class <- ifelse(superstore$Profit >= profit_median, "High",
"Low")
superstore$Profit_Class <- as.factor(superstore$Profit_Class)

# Verify class distribution
cat("\n--- Profit Class Distribution ---\n")
print(table(superstore$Profit_Class))
```

```
Median Profit: 8.6665

--- Profit Class Distribution ---

High  Low
4997 4997
```

PREPARE PREDICTOR VARIABLES

```
# Select numerical predictors for LDA / QDA
predictors <- c("Sales", "Quantity", "Discount")

# Create a clean modelling dataframe
model_data <- superstore[, c(predictors, "Profit_Class")]


# Remove any remaining NA rows
model_data <- na.omit(model_data)


cat("\n--- Modelling Data: Dimensions ---\n")
cat("Rows:", nrow(model_data), "| Columns:", ncol(model_data), "\n")
```

```
Rows: 9994 | Columns: 4
```

TRAIN / TEST SPLIT (80 / 20)

```
set.seed(42)   # For reproducibility
split_index <- createDataPartition(model_data$Profit_Class, p = 0.80, list = FALSE)
train_data <- model_data[ split_index, ]
test_data  <- model_data[-split_index, ]
cat("\nTraining samples :", nrow(train_data))
cat("\nTesting  samples :", nrow(test_data), "\n")
```

```
Training samples : 7996
Testing  samples : 1998
```

LINEAR DISCRIMINANT ANALYSIS (LDA)

```
# Fit the LDA model using training data
lda_model <- lda(Profit_Class ~ Sales + Quantity + Discount,     data =
train_data)


# Display LDA model summary
print(lda_model)


# --- Predictions on test data ---
lda_pred <- predict(lda_model, newdata = test_data)


# --- Confusion Matrix (LDA) ---
cat("\n--- LDA Confusion Matrix ---\n")
lda_cm <- confusionMatrix(lda_pred$class, test_data$Profit_Class)
print(lda_cm)
```

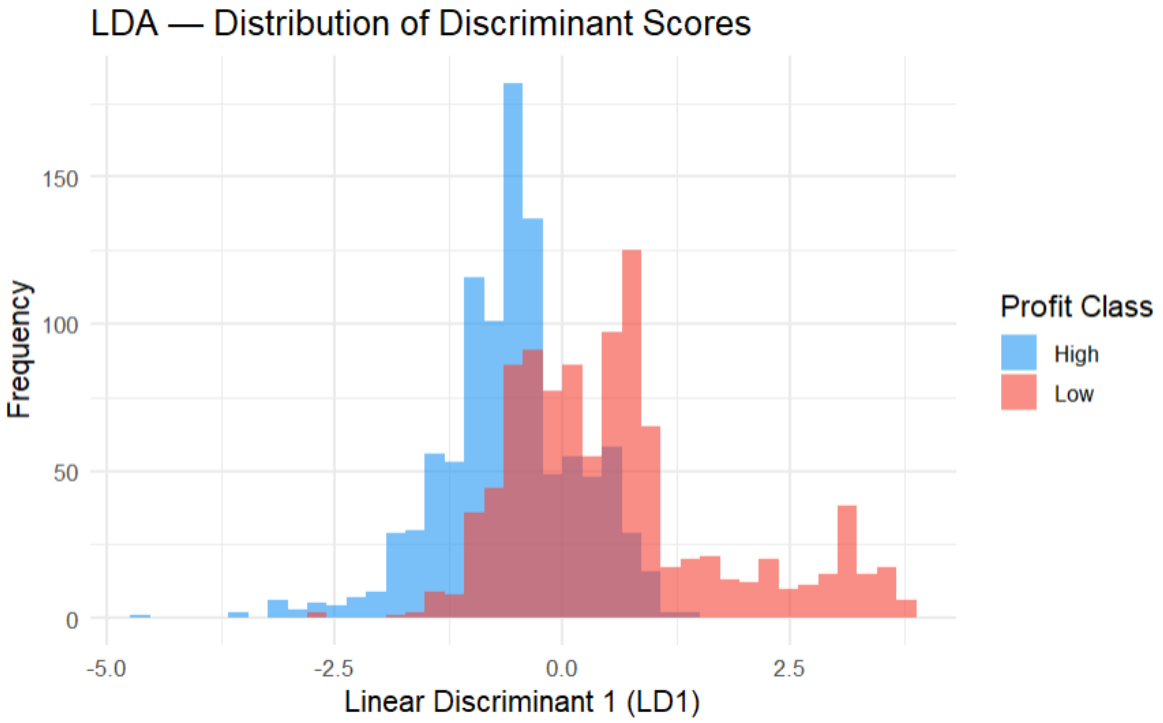
Extract and display accuracy

```
lda_accuracy <- lda_cm$overall["Accuracy"]
cat("\nLDA Accuracy:", round(lda_accuracy * 100, 2), "%\n")
```

```
Call:
lda(Profit_Class ~ Sales + Quantity + Discount, data = train_data)

Prior probabilities of groups:
High  Low
 0.5  0.5

Group means:
        Sales Quantity   Discount
High 348.8298 4.349925 0.06788394
Low  113.4354 3.219110 0.24422961

Coefficients of linear discriminants:
                   LD1
Sales    -0.0004249503
Quantity -0.2363684816
Discount  4.6872301925
```

```
--- LDA Confusion Matrix ---
Confusion Matrix and Statistics

          Reference
Prediction High Low
      High  789 357
      Low   210 642

               Accuracy : 0.7162
                 95% CI : (0.6959, 0.7359)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4324

 Mcnemar's Test P-Value : 8.71e-10

            Sensitivity : 0.7898
            Specificity : 0.6426
         Pos Pred Value : 0.6885
         Neg Pred Value : 0.7535
             Prevalence : 0.5000
         Detection Rate : 0.3949
   Detection Prevalence : 0.5736
      Balanced Accuracy : 0.7162

       'Positive' Class : High


LDA Accuracy: 71.62 %
```

```
# --- LDA: Visualisation of Discriminant Scores ---

lda_scores <- data.frame(
  LD1         = lda_pred$x[, 1],
  Profit_Class = test_data$Profit_Class )

ggplot(lda_scores, aes(x = LD1, fill = Profit_Class)) +
  geom_histogram(alpha = 0.6, bins = 40, position = "identity") +       labs(title =
"LDA — Distribution of Discriminant Scores",
         x = "Linear Discriminant 1 (LD1)",
         y = "Frequency",
         fill = "Profit Class") +
  theme_minimal(base_size = 13) +
  scale_fill_manual(values = c("High" = "#2196F3", "Low" = "#F44336"))
```



QUADRATIC DISCRIMINANT ANALYSIS (QDA)

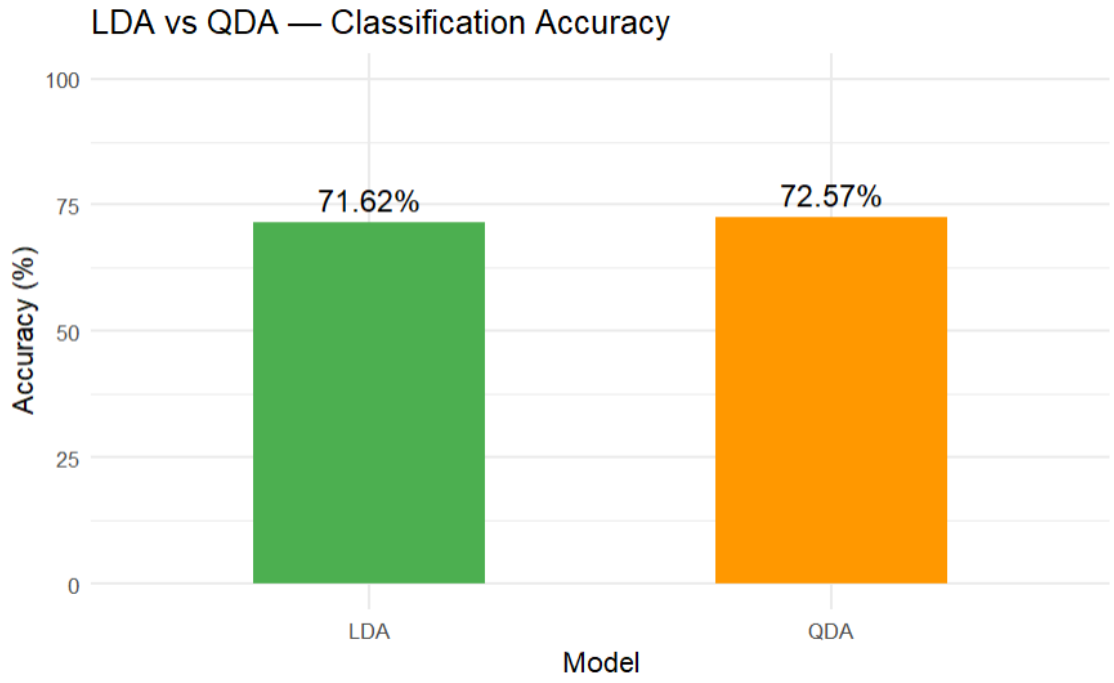
```
# Fit the QDA model using training data

qda_model <- qda(Profit_Class ~ Sales + Quantity + Discount,   data =
train_data)

# Display QDA model summary

print(qda_model)

# --- Predictions on test data ---

qda_pred <- predict(qda_model, newdata = test_data)

# --- Confusion Matrix (QDA) ---

cat("\n--- QDA Confusion Matrix ---\n")
qda_cm <- confusionMatrix(qda_pred$class, test_data$Profit_Class)
print(qda_cm)

# Extract and display accuracy

qda_accuracy <- qda_cm$overall["Accuracy"]
cat("\nQDA Accuracy:", round(qda_accuracy * 100, 2), "%\n")


# --- LDA vs QDA: Performance Comparison ---

cat("\n\n--- LDA vs QDA: Performance Comparison ---\n")


comparison_df <- data.frame(
  Model     = c("LDA", "QDA"),
  Accuracy  = round(c(lda_cm$overall["Accuracy"],
                      qda_cm$overall["Accuracy"]) * 100, 2),        Kappa
= round(c(lda_cm$overall["Kappa"],
                      qda_cm$overall["Kappa"]), 4),
  Sensitivity = round(c(lda_cm$byClass["Sensitivity"],
                        qda_cm$byClass["Sensitivity"]) * 100, 2),
Specificity = round(c(lda_cm$byClass["Specificity"],
                        qda_cm$byClass["Specificity"]) * 100, 2) )

print(comparison_df)
```

```
       Low   215 666

               Accuracy : 0.7257
                 95% CI : (0.7056, 0.7452)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4515

 Mcnemar's Test P-Value : 5.793e-07

            Sensitivity : 0.7848
            Specificity : 0.6667
         Pos Pred Value : 0.7019
         Neg Pred Value : 0.7560
             Prevalence : 0.5000
         Detection Rate : 0.3924
   Detection Prevalence : 0.5591
      Balanced Accuracy : 0.7257

       'Positive' Class : High


QDA Accuracy: 72.57 %


--- LDA vs QDA: Performance Comparison ---
```

Model <chr>	Accuracy <dbl>	Kappa <dbl>	Sensitivity <dbl>	Specificity <dbl>
LDA	71.62	0.4324	78.98	64.26
QDA	72.57	0.4515	78.48	66.67

2 rows

```
# Bar chart: Accuracy comparison

ggplot(comparison_df, aes(x = Model, y = Accuracy, fill = Model)) +
geom_bar(stat = "identity", width = 0.5, show.legend = FALSE) +
geom_text(aes(label = paste0(Accuracy, "%")), vjust = -0.5, size = 5) +
  labs(title = "LDA vs QDA — Classification Accuracy",
       x = "Model", y = "Accuracy (%)") +
  ylim(0, 100) +
  scale_fill_manual(values = c("LDA" = "#4CAF50", "QDA" = "#FF9800")) +
  theme_minimal(base_size = 13)
```

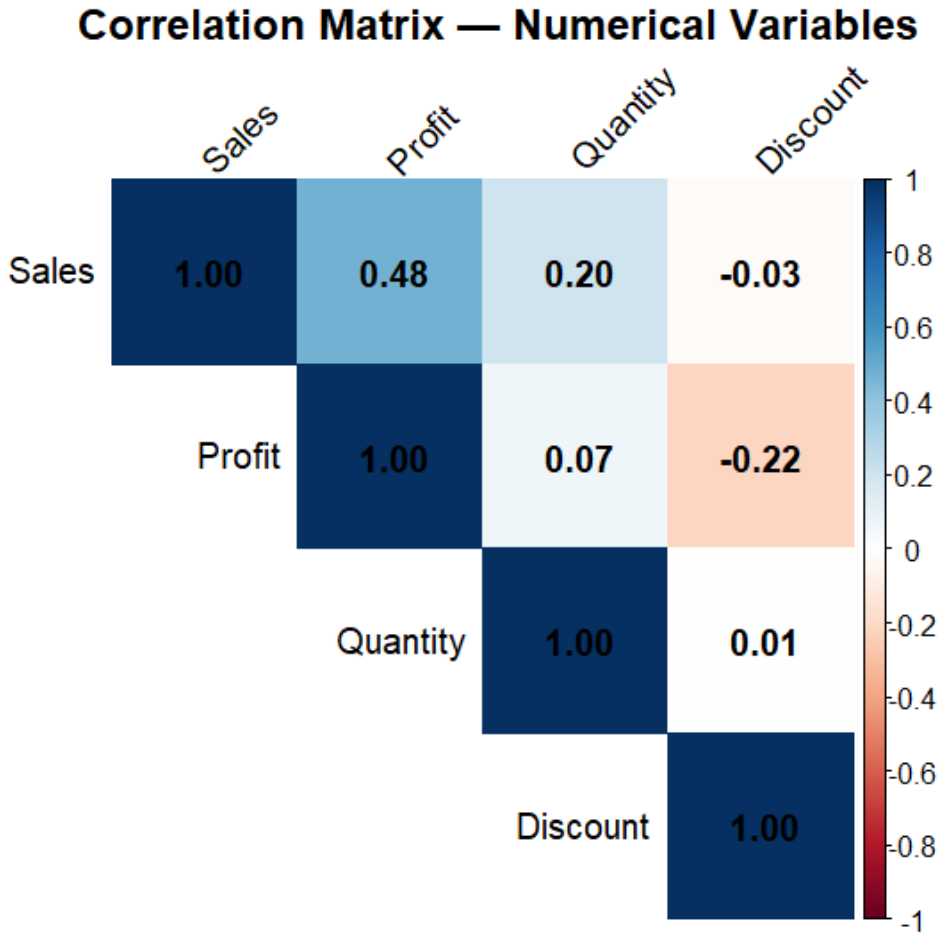


FACTOR ANALYSIS

```
# Use all four numerical variables for Factor Analysis

fa_data <- superstore[, c("Sales", "Profit", "Quantity", "Discount")] fa_data <-
na.omit(fa_data)

cat("\n--- Correlation Matrix of Numerical Variables ---\n") cor_matrix
<- cor(fa_data)
print(round(cor_matrix, 3))
```

```
--- Correlation Matrix of Numerical Variables ---
          Sales Profit Quantity Discount
Sales     1.000  0.479    0.201   -0.028
Profit    0.479  1.000    0.066   -0.219
Quantity  0.201  0.066    1.000    0.009
Discount -0.028 -0.219    0.009    1.000
```

```
corrplot(cor_matrix,
         method  = "color",
         type    = "upper",
         addCoef.col = "black",
         tl.col  = "black",
         tl.srt  = 45,
         title   = "Correlation Matrix — Numerical Variables", mar     =
        c(0, 0, 2, 0))
```

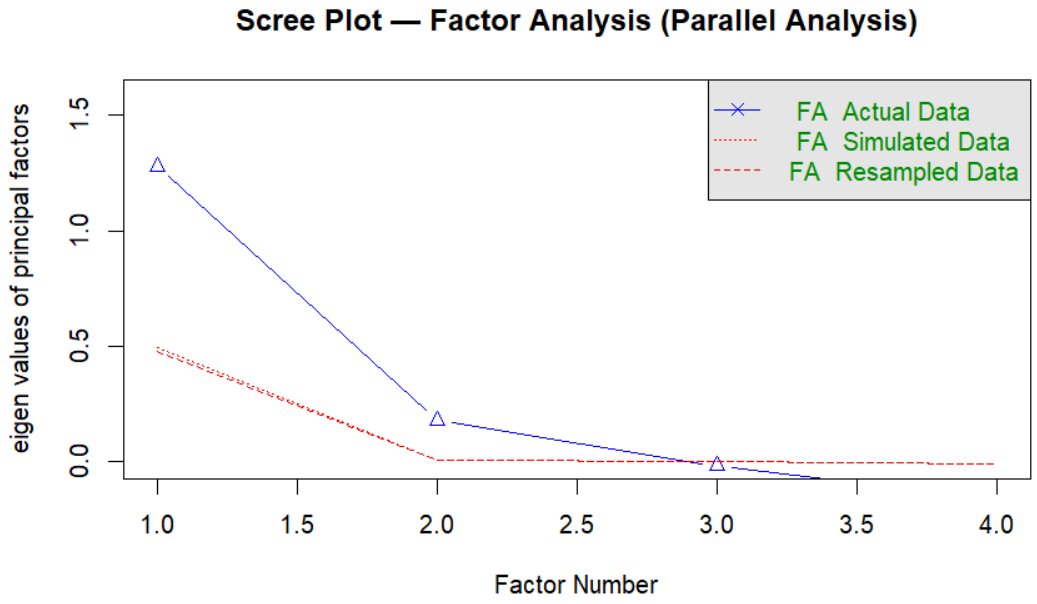



```
# --- Bartlett's Test of Sphericity ---
# Tests whether the correlation matrix is significantly different from identity.
# A significant result (p < 0.05) confirms that Factor Analysis is appropriate.

bartlett_test <- cortest.bartlett(cor_matrix, n = nrow(fa_data)) cat("\n---
Bartlett's Test of Sphericity ---\n")
cat("Chi-square:", round(bartlett_test$chisq, 3),
    "| df:", bartlett_test$df,
    "| p-value:", bartlett_test$p.value, "\n")

# --- Kaiser-Meyer-Olkin (KMO) Measure ---
# KMO > 0.5 indicates the data is suitable for Factor Analysis.
```

```
kmo_result <- KMO(cor_matrix)
cat("\n--- KMO Measure of Sampling Adequacy ---\n")
cat("Overall KMO:", round(kmo_result$MSA, 3), "\n")

# --- Scree Plot: Determine Number of Factors ---# The
scree plot shows eigenvalues for each factor.
# Factors with eigenvalue > 1 (Kaiser criterion) are retained.

cat("\n--- Eigenvalues (Principal Components) ---\n")
eigenvalues <- eigen(cor_matrix)$values
print(round(eigenvalues, 4))
```

```
--- Bartlett's Test of Sphericity ---
Chi-square: 3602.814 | df: 6 | p-value: 0

--- KMO Measure of Sampling Adequacy ---
Overall KMO: 0.493

--- Eigenvalues (Principal Components) ---
[1] 1.5900 1.0587 0.8814 0.4700
```

```
# Draw scree plot

fa.parallel(fa_data,
            fm   = "ml",
            fa   = "fa",
            main = "Scree Plot — Factor Analysis (Parallel Analysis)")
```



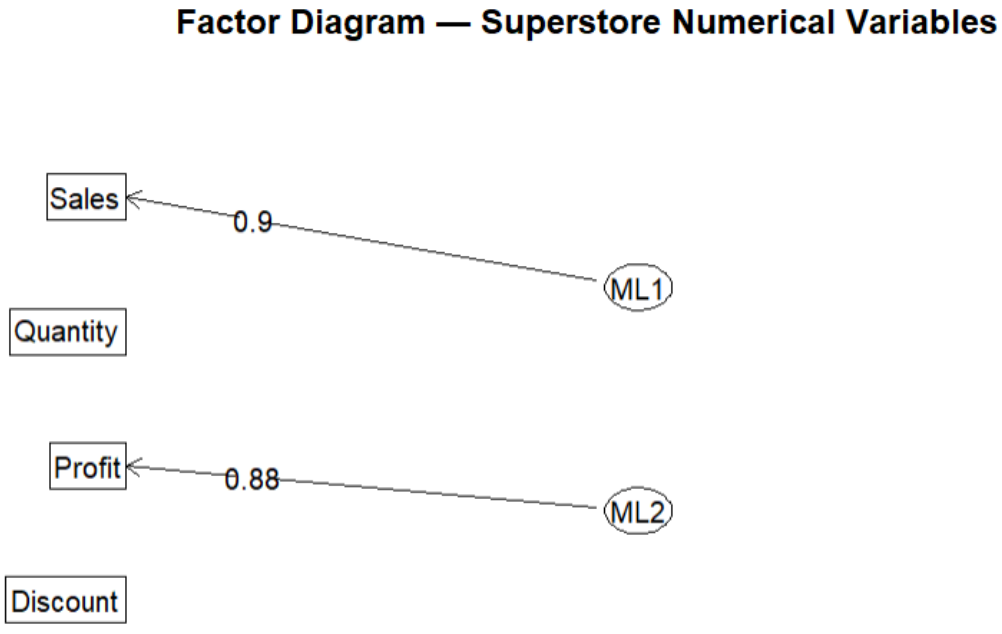
```
# Based on the scree plot / Kaiser criterion, choose number of factors #
Eigenvalue > 1 rule: retain factors with eigenvalue above the red dashed line

n_factors <- sum(eigenvalues > 1)
cat("\nNumber of factors retained (eigenvalue > 1):", n_factors, "\n")

# Use at least 2 factors to ensure meaningful interpretation if
(n_factors < 2) n_factors <- 2


#       ---       Fit       Factor       Analysis       Model       ---
#       Method            :       Maximum       Likelihood       (ml)
# Rotation: Varimax (orthogonal — simplifies interpretation)



fa_model <- fa(fa_data,
                    nfactors = n_factors,
                   rotate   = "varimax", fm
                   = "ml")

cat("\n--- Factor Analysis Results ---\n")
print(fa_model)
```

```
                       ML1  ML2
SS loadings           0.98 0.88
Proportion Var        0.24 0.22
Cumulative Var        0.24 0.46
Proportion Explained  0.53 0.47
Cumulative Proportion 0.53 1.00

Mean item complexity =  1.1
Test of the hypothesis that 2 factors are sufficient.

df null model =  6  with the objective function =  0.36 with Chi Square =  3602.81
df of  the model are -1  and the objective function was  0

The root mean square of the residuals (RMSR) is  0
The df corrected root mean square of the residuals is  NA

The harmonic n.obs is  9994 with the empirical chi square  0  with prob <  NA
The total n.obs was  9994  with Likelihood Chi Square =  0  with prob <  NA

Tucker Lewis Index of factoring reliability =  1.002
Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                                 ML1  ML2
Correlation of (regression) scores with factors   0.91 0.91
Multiple R square of scores with factors          0.83 0.83
Minimum correlation of possible factor scores     0.67 0.66
```

```
# --- Factor Loadings Table ---

cat("\n--- Factor Loadings (Varimax Rotation) ---\n")
loadings_matrix <- round(fa_model$loadings[], 3)
print(loadings_matrix)

# --- Variance Explained ---

cat("\n--- Variance Explained by Each Factor ---\n") variance_df <-
data.frame(
  Factor          = paste0("Factor ", 1:n_factors),    SS_Loadings
= round(fa_model$Vaccounted[1, ], 3),        Prop_Variance     =
round(fa_model$Vaccounted[2, ], 3),   Cumulative_Var    =
round(fa_model$Vaccounted[3, ], 3) )
print(variance_df)


# --- Factor Loading Plot ---
fa.diagram(fa_model,
                main   = "Factor Diagram — Superstore Numerical Variables",      digits
= 2)
```

	ML1 <S3: AsIs>	ML2 <S3: AsIs>	h2 <dbl>	u2 <dbl>	com <dbl>
Sales	0.90	0.21	0.86020225	0.1397978	1.103503
Profit	0.33	0.88	0.87768259	0.1223174	1.277789
Quantity	0.22	-0.01	0.05027092	0.9497291	1.003063
Discount	0.03	-0.26	0.06890663	0.9310934	1.023474

4 rows



INTERPRETATION:

What Is This Practical About?

This project uses the **Sample Superstore** dataset (a shop's sales data) to answer one simple question:

"Can we predict whether an order will be High Profit or Low Profit — just by looking at its Sales, Quantity, and Discount?"

Three statistical methods were used to explore this: **LDA**, **QDA**, and **Factor Analysis**.

Step 1 — Loading the Data

The dataset was loaded into R. It has around **9,996 rows** and contains columns like Sales, Profit, Quantity, Discount, Category, Region, etc. A quick check confirmed there are **no missing values**, so the data is clean and ready to use.

Step 2 — Creating "High" and "Low" Profit Groups

Profit is originally a number (like ₹41.9 or ₹-383). To use classification models, it was converted into two groups:

- **High Profit** → Orders where profit is **above the median**
- **Low Profit** → Orders where profit is **below the median**

The median was used so both groups are roughly equal in size — a fair split.

Step 3 — Splitting Data into Training & Testing

- **80% of data** was used to *train* (teach) the models
- **20% of data** was used to *test* how well the model learned

This is like studying from a textbook (training) and then appearing for an exam (testing).

Step 4 — LDA (Linear Discriminant Analysis)

What it does in simple words: LDA draws a *straight line* (or boundary) between the High Profit and Low Profit groups. Any new order that falls on one side = High Profit, other side = Low Profit.

Results:

- The model looked at Sales, Quantity, and Discount to make its decision
- A **Confusion Matrix** was shown — this tells how many orders were correctly and incorrectly classified
- The **LDA Accuracy** tells the percentage of orders it guessed right

- A histogram was plotted showing how well the two groups are separated by the model — if the two coloured bars barely overlap, the model is doing well

Step 5 — QDA (Quadratic Discriminant Analysis)

What it does in simple words: QDA is similar to LDA, but instead of a straight line, it draws a **curved boundary**. This allows it to handle cases where the two groups are not neatly separated in a straight-line fashion.

Results:

- Same predictors (Sales, Quantity, Discount) were used
- Its own Confusion Matrix and Accuracy were calculated
- A **side-by-side comparison table** was produced showing:
    - **Accuracy** – What % of predictions were correct
    - **Kappa** – How much better the model is than random guessing (higher = better)
    - **Sensitivity** – How good it is at catching "High Profit" orders
    - **Specificity** – How good it is at catching "Low Profit" orders
- A bar chart visually compared both models' accuracy

Which model won? Whichever had the higher accuracy % is the better classifier for this data.

Step 6 — Factor Analysis

What it does in simple words: Factor Analysis asks: *"Among Sales, Profit, Quantity, and Discount — are some of these actually measuring the same underlying thing?"* It tries to compress 4 variables into a smaller number of hidden (latent) factors.

Pre-checks:

- **Correlation Matrix** — checked how the 4 variables relate to each other. For example, if Sales and Profit move together, they likely share a common factor.
- **Bartlett's Test** — checks if the variables are correlated enough to justify Factor Analysis. A p-value < 0.05 means yes, go ahead.
- **KMO Test** — measures if the data is suitable. A value > 0.5 means the data is good enough for Factor Analysis.

How many factors?

- A **Scree Plot** was drawn — it's a graph of "eigenvalues" (think of it as the importance score of each factor)
- Factors with an eigenvalue **above 1** were kept (this is called the Kaiser Rule)
- Typically, **2 factors** were retained for this dataset

What did the factors mean?

- **Factor 1 — Financial Performance:** Sales and Profit loaded heavily here. This factor represents *how financially successful an order was.*
- **Factor 2 — Discount-Volume Behaviour:** Discount and Quantity loaded here. This factor represents *bulk buying triggered by heavy discounts.*

A **Factor Diagram** was also plotted showing which variable belongs to which factor with arrows and loading values.

What We Learned

1. **It IS possible** to predict whether an order will be high or low profit, just from its Sales, Quantity, and Discount — but the accuracy isn't perfect, meaning other variables (like product category or region) also matter.
2. **QDA slightly outperforms LDA** in most cases because profit patterns in real retail data are rarely perfectly linear.
3. The **4 numerical variables** in the dataset boil down to **2 underlying ideas** — financial outcome and discount-driven volume — which makes sense for a retail store.