MSE: TSM_MarkFor | FS 2025
Daniel Kogan | Sven Träger | *Date: 01.05.2025*

Optimization of Beverage Sales

Load necessary libraries

In [1]:
```
library(tidyverse) # includes ggplot2, dplyr, tidyr, purrr
library(lubridate)
library(skimr)
library(janitor)
library(knitr)
library(rfm)
library(caret)
library(jtools)
library(broom)
library(Metrics)
```

```
Warning message:
"package 'tidyverse' was built under R version 4.4.3"
Warning message:
"package 'tidyr' was built under R version 4.4.3"
Warning message:
"package 'readr' was built under R version 4.4.3"
Warning message:
"package 'dplyr' was built under R version 4.4.3"
Warning message:
"package 'forcats' was built under R version 4.4.3"
Warning message:
"package 'lubridate' was built under R version 4.4.3"
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.1     ✓ tibble    3.2.1
✓ lubridate 1.9.4     ✓ tidyr     1.3.1
✓ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all confli
cts to become errors
Warning message:
"package 'skimr' was built under R version 4.4.3"
Warning message:
"package 'janitor' was built under R version 4.4.3"

Attaching package: 'janitor'


The following objects are masked from 'package:stats':

    chisq.test, fisher.test


Warning message:
"package 'rfm' was built under R version 4.4.3"
Warning message:
"package 'caret' was built under R version 4.4.3"
Loading required package: lattice


Attaching package: 'caret'


The following object is masked from 'package:purrr':

    lift


Warning message:
"package 'jtools' was built under R version 4.4.3"
Warning message:
"package 'broom' was built under R version 4.4.3"
Warning message:
"package 'Metrics' was built under R version 4.4.3"

Attaching package: 'Metrics'
```

```
The following objects are masked from 'package:caret':

    precision, recall
```

Load dataset

In [37]:
```
df <- read_csv("synthetic_beverage_sales_data.csv", show_col_types = FALSE) %>%
  clean_names()
head(df, 4) # print first 4 rows of data frame
glimpse(df) # print summary of data frame
```

A tibble: 4 × 11

order_id	customer_id	customer_type	product	category	unit_price	quantity	discou
<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<db
ORD1	CUS1496	B2B	Vio Wasser	Water	1.66	53	0.
ORD1	CUS1496	B2B	Evian	Water	1.56	90	0.
ORD1	CUS1496	B2B	Sprite	Soft Drinks	1.17	73	0.
ORD1	CUS1496	B2B	Rauch Multivitamin	Juices	3.22	59	0.

```
Rows: 8,999,910
Columns: 11
$ order_id      <chr> "ORD1", "ORD1", "ORD1", "ORD1", "ORD1", "ORD2", "ORD3", …
$ customer_id   <chr> "CUS1496", "CUS1496", "CUS1496", "CUS1496", "CUS1496", "…
$ customer_type <chr> "B2B", "B2B", "B2B", "B2B", "B2B", "B2C", "B2B", "B2B", …
$ product       <chr> "Vio Wasser", "Evian", "Sprite", "Rauch Multivitamin", "…
$ category      <chr> "Water", "Water", "Soft Drinks", "Juices", "Water", "Alc…
$ unit_price    <dbl> 1.66, 1.56, 1.17, 3.22, 0.87, 9.09, 2.14, 0.43, 1.21, 1.…
$ quantity      <dbl> 53, 90, 73, 59, 35, 2, 44, 13, 92, 3, 11, 8, 16, 3, 43, …
$ discount      <dbl> 0.10, 0.10, 0.05, 0.10, 0.10, 0.00, 0.10, 0.05, 0.10, 0.…
$ total_price   <dbl> 79.18, 126.36, 81.14, 170.98, 27.40, 18.18, 84.74, 5.31,…
$ region        <chr> "Baden-Württemberg", "Baden-Württemberg", "Baden-Württem…
$ order_date    <date> 2023-08-23, 2023-08-23, 2023-08-23, 2023-08-23, 2023-08…
```

Data Preperation

In [3]:
```
df[df == ""] <- NA # Replace empty strings with NA

colSums(is.na(df)) # Check for missing values

sum(duplicated(df)) # Check for duplicate entries

df %>% # Count rows with invalid or implausible values
  filter(
    unit_price <= 0 | # Unit price should be greater than 0
      quantity <= 0 | # Quantity should be greater than 0
      discount < 0 | discount > 1 | # Discount must be between 0 and 1
```

```
      total_price < 0 # Total price should not be negative
  ) %>%
  nrow() # Count number of rows that meet any of the above conditions
```

order_id: 0 **customer_id:** 0 **customer_type:** 0 **product:** 0 **category:** 0 **unit_price:** 0 **quantity:** 0 **discount:** 0 **total_price:** 0 **region:** 0 **order_date:** 0

0

0

Splitting into B2B and B2C & grouping into different dataframes:

line-wise, rfm-base, numeric-invoice-wise, full-invoice-wise

In [4]:

```
# Split into B2B and B2C segments
# Each row represents a product purchase/position(line item)
# by a customer on a specific day.
# A customer may appear multiple times per day for different products.

# B2B orders
line_wise_b2b <- df %>% filter(customer_type == "B2B")
# B2C orders
line_wise_b2c <- df %>% filter(customer_type == "B2C")

# Required for RFM: only customer_id, order_date, and revenue

# B2B orders
rfm_base_b2b <- line_wise_b2b %>%
  group_by(customer_id, order_date) %>% # Group by customer and date
  summarise(
    total_price = sum(total_price, na.rm = TRUE), # Total revenue per order
    .groups = "drop"
  )
# B2C orders
rfm_base_b2c <- line_wise_b2c %>%
  group_by(customer_id, order_date) %>%
  summarise(
    total_price = sum(total_price, na.rm = TRUE),
    .groups = "drop"
  )

# Numerical summary per invoice (for statistical modeling)
# One row per customer and date = one invoice

# B2B orders
numeric_invoice_wise_b2b <- line_wise_b2b %>%
  group_by(customer_id, order_date) %>%
  summarise(
    quantity = sum(quantity, na.rm = TRUE),
    discount = mean(discount, na.rm = TRUE), # Avg. discount per invoice
    unit_price = mean(unit_price, na.rm = TRUE), # Avg. unit price per invoice
    total_price = sum(total_price, na.rm = TRUE), # Total invoice value
    .groups = "drop"
  )
# B2C orders
numeric_invoice_wise_b2c <- line_wise_b2c %>%
  group_by(customer_id, order_date) %>%
```

```
  summarise(
    quantity = sum(quantity, na.rm = TRUE),
    discount = mean(discount, na.rm = TRUE),
    unit_price = mean(unit_price, na.rm = TRUE),
    total_price = sum(total_price, na.rm = TRUE),
    .groups = "drop"
  )

# Full invoice-level dataset including order_id and region
# Each row = one invoice (combination of order_id, customer_id, and date)

# B2B orders
full_invoice_wise_b2b <- line_wise_b2b %>%
  group_by(order_id, customer_id, order_date, region) %>%
  summarise(
    quantity = sum(quantity, na.rm = TRUE),
    discount = mean(discount, na.rm = TRUE),
    unit_price = mean(unit_price, na.rm = TRUE),
    total_price = sum(total_price, na.rm = TRUE),
    .groups = "drop"
  )
# B2C orders
full_invoice_wise_b2c <- line_wise_b2c %>%
  group_by(order_id, customer_id, order_date, region) %>%
  summarise(
    quantity = sum(quantity, na.rm = TRUE),
    discount = mean(discount, na.rm = TRUE),
    unit_price = mean(unit_price, na.rm = TRUE),
    total_price = sum(total_price, na.rm = TRUE),
    .groups = "drop"
  )
```

In [5]:

```
# ------------------------------ prints ------------------------------
print("line-wise, b2b")
glimpse(line_wise_b2b) # print summary of data frame
head(line_wise_b2b, 4) # print first 4 rows of data frame

print("rfm-base, b2b")
glimpse(rfm_base_b2b) # print summary of data frame
head(rfm_base_b2b, 4) # print first 4 rows of data frame

print("numeric-invoice-wise, b2b")
glimpse(numeric_invoice_wise_b2b) # print summary of data frame
head(numeric_invoice_wise_b2b, 4) # print first 4 rows of data frame

print("full-invoice-wise, b2b")
glimpse(full_invoice_wise_b2b) # print summary of data frame
head(full_invoice_wise_b2b, 4) # print first 4 rows of data frame
```

```
[1] "line-wise, b2b"
Rows: 3,204,505
Columns: 11
$ order_id      <chr> "ORD1", "ORD1", "ORD1", "ORD1", "ORD1", "ORD3", "ORD3", …
$ customer_id   <chr> "CUS1496", "CUS1496", "CUS1496", "CUS1496", "CUS1496", "…
$ customer_type <chr> "B2B", "B2B", "B2B", "B2B", "B2B", "B2B", "B2B", "B2B", …
$ product       <chr> "Vio Wasser", "Evian", "Sprite", "Rauch Multivitamin", "…
$ category      <chr> "Water", "Water", "Soft Drinks", "Juices", "Water", "Jui…
$ unit_price    <dbl> 1.66, 1.56, 1.17, 3.22, 0.87, 2.14, 0.43, 1.21, 1.38, 1.…
$ quantity      <dbl> 53, 90, 73, 59, 35, 44, 13, 92, 3, 8, 16, 3, 43, 44, 10,…
$ discount      <dbl> 0.10, 0.10, 0.05, 0.10, 0.10, 0.10, 0.05, 0.10, 0.05, 0.…
$ total_price   <dbl> 79.18, 126.36, 81.14, 170.98, 27.40, 84.74, 5.31, 100.19…
$ region        <chr> "Baden-Württemberg", "Baden-Württemberg", "Baden-Württem…
$ order_date    <date> 2023-08-23, 2023-08-23, 2023-08-23, 2023-08-23, 2023-08…
```

A tibble: 4 × 11

order_id	customer_id	customer_type	product	category	unit_price	quantity	discou
<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<db
ORD1	CUS1496	B2B	Vio Wasser	Water	1.66	53	0.
ORD1	CUS1496	B2B	Evian	Water	1.56	90	0.
ORD1	CUS1496	B2B	Sprite	Soft Drinks	1.17	73	0.
ORD1	CUS1496	B2B	Rauch Multivitamin	Juices	3.22	59	0.

```
[1] "rfm-base, b2b"
Rows: 935,300
Columns: 3
$ customer_id <chr> "CUS1000", "CUS1000", "CUS1000", "CUS1000", "CUS1000", "CU…
$ order_date  <date> 2021-01-03, 2021-01-04, 2021-01-06, 2021-01-11, 2021-01-1…
$ total_price <dbl> 827.07, 414.21, 142.12, 348.41, 1051.40, 570.97, 4350.27, …
```

A tibble: 4 × 3

customer_id	order_date	total_price
<chr>	<date>	<dbl>
CUS1000	2021-01-03	827.07
CUS1000	2021-01-04	414.21
CUS1000	2021-01-06	142.12
CUS1000	2021-01-11	348.41

```
[1] "numeric-invoice-wise, b2b"
Rows: 935,300
Columns: 6
$ customer_id <chr> "CUS1000", "CUS1000", "CUS1000", "CUS1000", "CUS1000", "CU…
$ order_date  <date> 2021-01-03, 2021-01-04, 2021-01-06, 2021-01-11, 2021-01-1…
$ quantity    <dbl> 228, 144, 22, 130, 241, 139, 107, 297, 30, 65, 293, 163, 1…
$ discount    <dbl> 0.08750000, 0.06250000, 0.05000000, 0.12500000, 0.08000000…
$ unit_price  <dbl> 2.887500, 2.522500, 6.800000, 2.435000, 4.976000, 5.104000…
$ total_price <dbl> 827.07, 414.21, 142.12, 348.41, 1051.40, 570.97, 4350.27, …
```

A tibble: 4 × 6

customer_id	order_date	quantity	discount	unit_price	total_price
<chr>	<date>	<dbl>	<dbl>	<dbl>	<dbl>
CUS1000	2021-01-03	228	0.0875	2.8875	827.07
CUS1000	2021-01-04	144	0.0625	2.5225	414.21
CUS1000	2021-01-06	22	0.0500	6.8000	142.12
CUS1000	2021-01-11	130	0.1250	2.4350	348.41

```
[1] "full-invoice-wise, b2b"
Rows: 1,068,808
Columns: 8
$ order_id    <chr> "ORD1", "ORD10", "ORD1000", "ORD1000000", "ORD1000006", "O…
$ customer_id <chr> "CUS1496", "CUS9472", "CUS9185", "CUS3347", "CUS3145", "CU…
$ order_date  <date> 2023-08-23, 2023-04-09, 2022-06-17, 2023-07-07, 2021-01-2…
$ region      <chr> "Baden-Württemberg", "Bayern", "Niedersachsen", "Thüringen…
$ quantity    <dbl> 310, 196, 161, 184, 75, 167, 11, 244, 359, 218, 340, 380, …
$ discount    <dbl> 0.09000000, 0.08333333, 0.08750000, 0.08333333, 0.10000000…
$ unit_price  <dbl> 1.696000, 1.820000, 6.480000, 30.736667, 1.430000, 1.98666…
$ total_price <dbl> 485.06, 301.74, 1018.33, 6293.73, 96.52, 370.17, 18.29, 22…
```

A tibble: 4 × 8

order_id	customer_id	order_date	region	quantity	discount	unit_price	to
<chr>	<chr>	<date>	<chr>	<dbl>	<dbl>	<dbl>	
ORD1	CUS1496	2023-08-23	Baden-Württemberg	310	0.09000000	1.69600	
ORD10	CUS9472	2023-04-09	Bayern	196	0.08333333	1.82000	
ORD1000	CUS9185	2022-06-17	Niedersachsen	161	0.08750000	6.48000	
ORD1000000	CUS3347	2023-07-07	Thüringen	184	0.08333333	30.73667	

Descriptive Statistical Analysis

In [6]:

```
# Summary statistics based on line-item level (each row = one product purchase)
# Not aggregated by invoice/customer

# B2B orders
line_wise_summary_b2b <- line_wise_b2b %>%
  summarise(
    discount    = list(discount),
    quantity    = list(quantity),
    total_price = list(total_price),
    unit_price  = list(unit_price)
  ) %>%
  pivot_longer(everything(), names_to = "variable", values_to = "values") %>%
  mutate(
    count  = map_int(values, ~ length(.x)),
    mean   = map_dbl(values, ~ mean(.x, na.rm = TRUE)),
```

```r
    sd     = map_dbl(values, ~ sd(.x, na.rm = TRUE)),
    min    = map_dbl(values, ~ min(.x, na.rm = TRUE)),
    q25    = map_dbl(values, ~ quantile(.x, 0.25, na.rm = TRUE)),
    median = map_dbl(values, ~ median(.x, na.rm = TRUE)),
    q75    = map_dbl(values, ~ quantile(.x, 0.75, na.rm = TRUE)),
    max    = map_dbl(values, ~ max(.x, na.rm = TRUE))
  ) %>%
  select(variable, count, mean, sd, min, q25, median, q75, max)
# B2C orders
line_wise_summary_b2c <- line_wise_b2c %>%
  summarise(
    discount    = list(discount),
    quantity    = list(quantity),
    total_price = list(total_price),
    unit_price  = list(unit_price)
  ) %>%
  pivot_longer(everything(), names_to = "variable", values_to = "values") %>%
  mutate(
    count  = map_int(values, ~ length(.x)),
    mean   = map_dbl(values, ~ mean(.x, na.rm = TRUE)),
    sd     = map_dbl(values, ~ sd(.x, na.rm = TRUE)),
    min    = map_dbl(values, ~ min(.x, na.rm = TRUE)),
    q25    = map_dbl(values, ~ quantile(.x, 0.25, na.rm = TRUE)),
    median = map_dbl(values, ~ median(.x, na.rm = TRUE)),
    q75    = map_dbl(values, ~ quantile(.x, 0.75, na.rm = TRUE)),
    max    = map_dbl(values, ~ max(.x, na.rm = TRUE))
  ) %>%
  select(variable, count, mean, sd, min, q25, median, q75, max)

# Summary statistics based on invoice-level data
# Each row represents one invoice (aggregated per customer and order date)

# B2B orders
invoice_wise_summary_b2b <- numeric_invoice_wise_b2b %>%
  summarise(
    discount    = list(discount),
    quantity    = list(quantity),
    total_price = list(total_price),
    unit_price  = list(unit_price)
  ) %>%
  pivot_longer(everything(), names_to = "variable", values_to = "values") %>%
  mutate(
    count  = map_int(values, ~ length(.x)),
    mean   = map_dbl(values, ~ mean(.x, na.rm = TRUE)),
    sd     = map_dbl(values, ~ sd(.x, na.rm = TRUE)),
    min    = map_dbl(values, ~ min(.x, na.rm = TRUE)),
    q25    = map_dbl(values, ~ quantile(.x, 0.25, na.rm = TRUE)),
    median = map_dbl(values, ~ median(.x, na.rm = TRUE)),
    q75    = map_dbl(values, ~ quantile(.x, 0.75, na.rm = TRUE)),
    max    = map_dbl(values, ~ max(.x, na.rm = TRUE))
  ) %>%
  select(variable, count, mean, sd, min, q25, median, q75, max)
# B2C orders
invoice_wise_summary_b2c <- numeric_invoice_wise_b2c %>%
  summarise(
    discount    = list(discount),
    quantity    = list(quantity),
    total_price = list(total_price),
    unit_price  = list(unit_price)
  ) %>%
```

```
  pivot_longer(everything(), names_to = "variable", values_to = "values") %>%
  mutate(
    count  = map_int(values, ~ length(.x)),
    mean   = map_dbl(values, ~ mean(.x, na.rm = TRUE)),
    sd     = map_dbl(values, ~ sd(.x, na.rm = TRUE)),
    min    = map_dbl(values, ~ min(.x, na.rm = TRUE)),
    q25    = map_dbl(values, ~ quantile(.x, 0.25, na.rm = TRUE)),
    median = map_dbl(values, ~ median(.x, na.rm = TRUE)),
    q75    = map_dbl(values, ~ quantile(.x, 0.75, na.rm = TRUE)),
    max    = map_dbl(values, ~ max(.x, na.rm = TRUE))
  ) %>%
  select(variable, count, mean, sd, min, q25, median, q75, max)
```

In [7]:
```
# ------------------------------ Print summary tables --------------------------
kable(line_wise_summary_b2b, caption = "Line-wise summary statistics of B2B nume
kable(line_wise_summary_b2c, caption = "Line-wise summary statistics of B2C nume
kable(invoice_wise_summary_b2b, caption = "Invoice-level summary statistics of B
kable(invoice_wise_summary_b2c, caption = "Invoice-level summary statistics of B
```

Table: Line-wise summary statistics of B2B numeric variables

variable	count	mean	sd	min	q25	median	q75	max
discount	3204505	0.08	0.03	0.05	0.05	0.10	0.10	0.15
quantity	3204505	50.52	28.87	1.00	26.00	51.00	76.00	100.00
total_price	3204505	281.36	810.75	0.30	31.35	77.20	167.58	14295.30
unit_price	3204505	5.63	13.21	0.32	1.03	1.81	3.04	169.53

Table: Line-wise summary statistics of B2C numeric variables

variable	count	mean	sd	min	q25	median	q75	max
discount	5795405	0.00	0.00	0.00	0.00	0.00	0.00	0.00
quantity	5795405	8.00	4.32	1.00	4.00	8.00	12.00	15.00
total_price	5795405	47.46	143.17	0.46	6.20	13.02	26.28	1686.90
unit_price	5795405	5.92	15.46	0.46	1.07	1.75	3.21	112.46

Table: Invoice-level summary statistics of B2B orders

variable	count	mean	sd	min	q25	median	q75	max
discount	935300	0.08	0.02	0.05	0.07	0.08	0.10	0.15
quantity	935300	173.08	110.04	1.00	89.00	159.00	237.00	1166.00
total_price	935300	963.99	1582.90	0.30	161.36	365.37	945.58	23490.22
unit_price	935300	5.63	8.55	0.32	1.55	2.30	6.12	154.50

Table: Invoice-level summary statistics of B2C orders

variable	count	mean	sd	min	q25	median	q75	max
discount	1689025	0.00	0.00	0.00	0.00	0.00	0.00	0.00
quantity	1689025	27.45	17.22	1.00	14.00	25.00	38.00	199.00
total_price	1689025	162.85	277.79	0.46	27.51	59.74	142.02	4067.90
unit_price	1689025	5.92	9.94	0.46	1.57	2.24	4.81	112.46

RFM Analysis

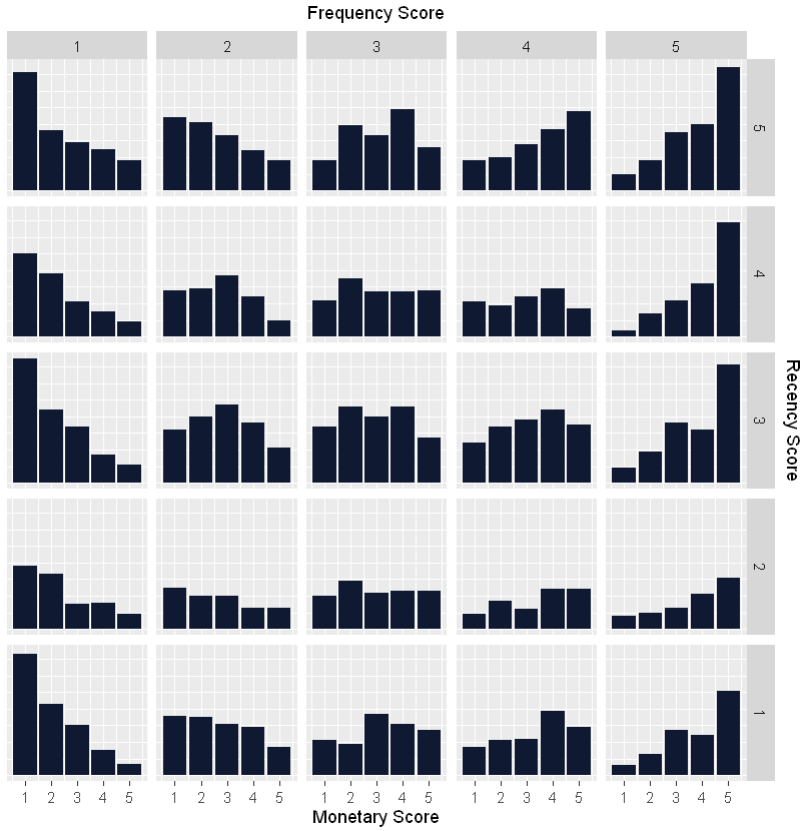
In [8]:
```r
analysis_date <- as.Date("2023-12-31") # Analysis date for RFM:last day of year
# B2B orders
rfm_result_b2b <- rfm_table_order(
  data = rfm_base_b2b,
  customer_id = customer_id, # Column identifying each customer
  order_date = order_date, # Column with the date of each order
  revenue = total_price, # Column with the monetary value of the order
  analysis_date = analysis_date # Reference point for recency calculation
)
# B2C orders
rfm_result_b2c <- rfm_table_order(
  data = rfm_base_b2c,
  customer_id = customer_id,
  order_date = order_date,
  revenue = total_price,
  analysis_date = analysis_date
)
```

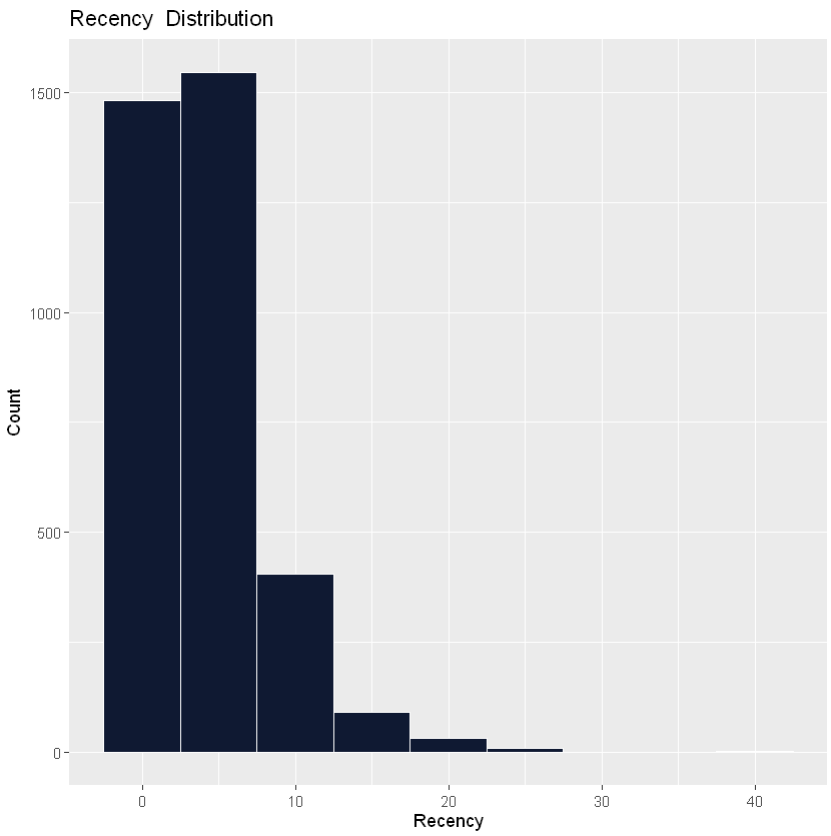
RFM results & visualizations

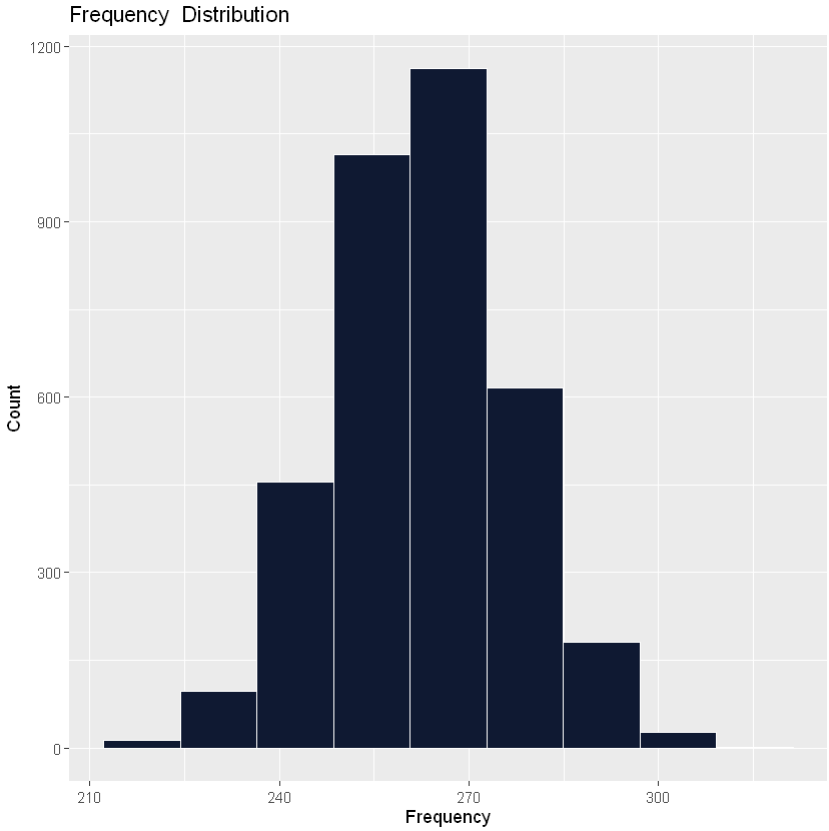
In [9]:
```r
glimpse(rfm_result_b2b$rfm)
```

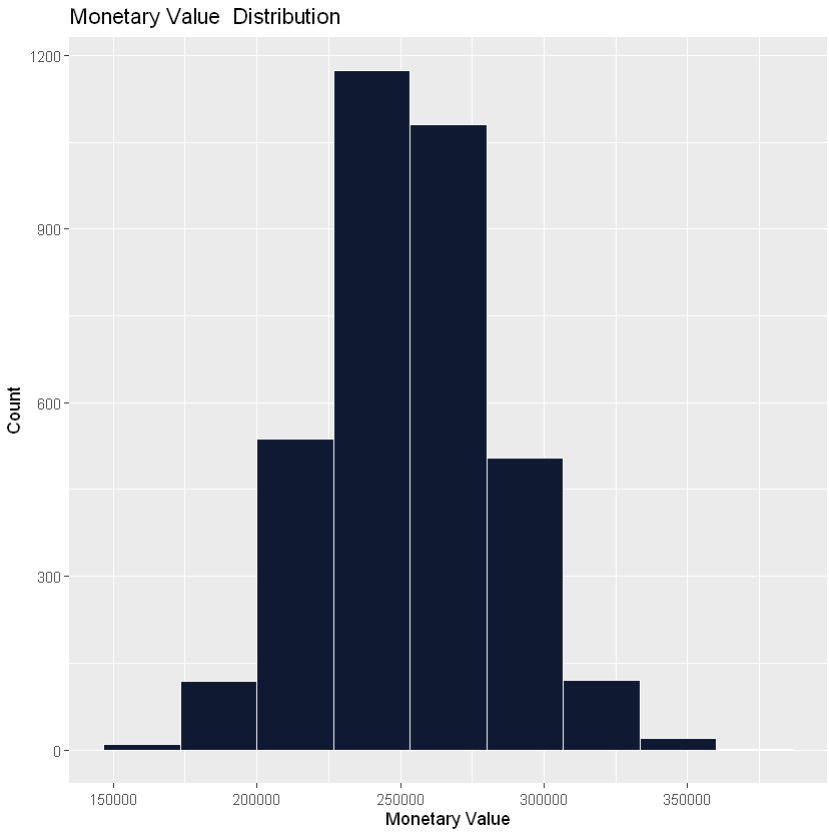
```
Rows: 3,563
Columns: 8
$ customer_id       <chr> "CUS1000", "CUS10000", "CUS1002", "CUS1005", "CUS100…
$ recency_days      <dbl> 2, 8, 5, 2, 6, 6, 5, 6, 2, 1, 8, 2, 1, 6, 4, 3, 1, 1…
$ transaction_count <int> 258, 259, 241, 258, 266, 247, 262, 249, 267, 267, 26…
$ amount            <dbl> 271186.9, 253436.7, 247201.9, 259063.1, 277493.9, 26…
$ recency_score     <int> 4, 1, 2, 4, 2, 2, 2, 2, 4, 5, 1, 4, 5, 2, 3, 3, 5, 5…
$ frequency_score   <int> 2, 2, 1, 2, 3, 1, 3, 1, 4, 4, 3, 2, 1, 1, 1, 2, 5, 3…
$ monetary_score    <int> 4, 3, 3, 3, 4, 4, 1, 1, 1, 3, 3, 4, 1, 1, 3, 1, 3, 2…
$ rfm_score         <dbl> 424, 123, 213, 423, 234, 214, 231, 211, 441, 543, 13…
```

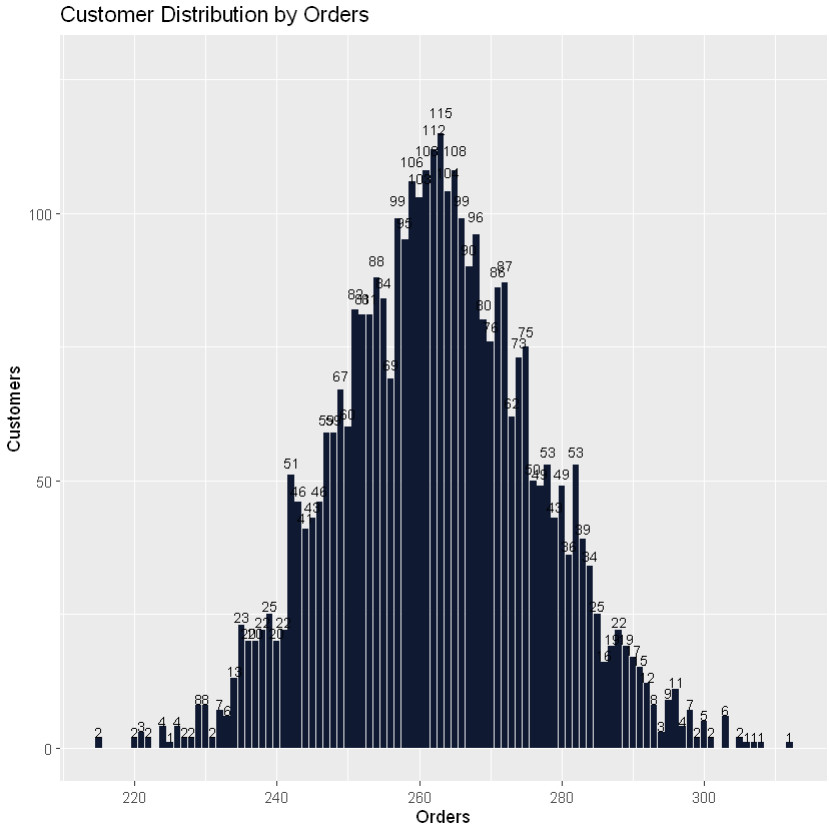
In [10]:
```r
# B2B customers
# Visualize distribution of RFM scores (count of customers per total RFM score)
rfm_plot_bar_chart(rfm_result_b2b)
# Distribution of recency scores (how recently customers made a purchase)
rfm_plot_histogram(rfm_result_b2b, metric = "recency")
# Distribution of frequency scores (how often customers purchased)
rfm_plot_histogram(rfm_result_b2b, metric = "frequency")
# Distribution of monetary scores (how much customers spent)
rfm_plot_histogram(rfm_result_b2b, metric = "monetary")
# Frequency of RFM score combinations (e.g., RFM = 555, 444, etc.)
rfm_plot_order_dist(rfm_result_b2b)
# Heatmap of average monetary value by Recency and Frequency segments
rfm_plot_heatmap(rfm_result_b2b)
```

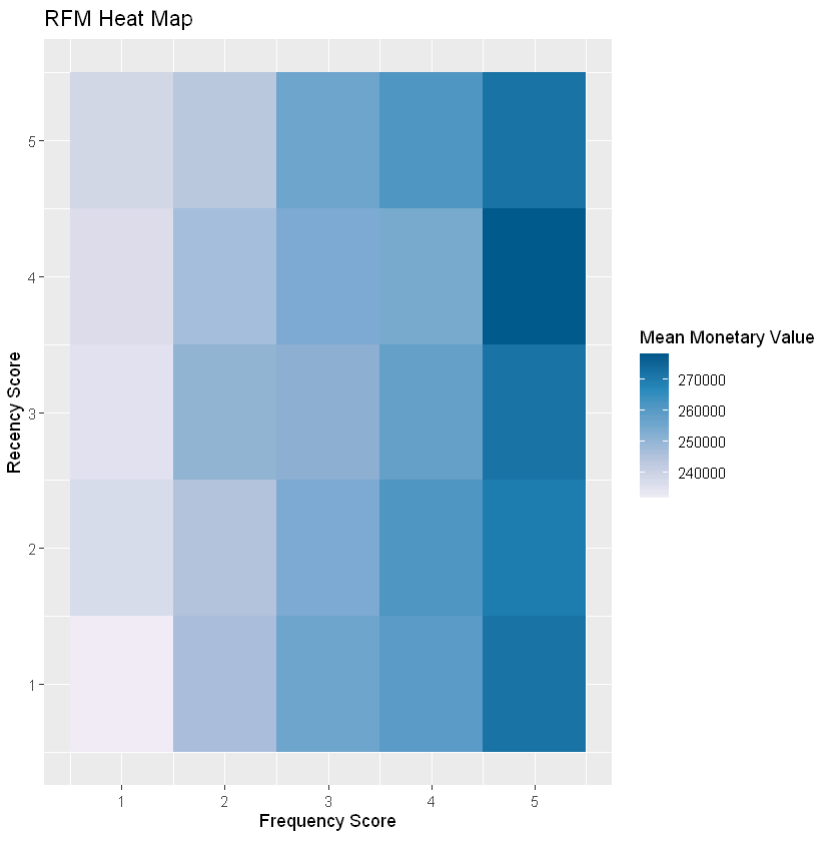

Frequency Score
1
2
3
4
5
Recency Score
Monetary Score


Recency Distribution
Count
1500
1000
500
0
0
10
20
30
40
Recency

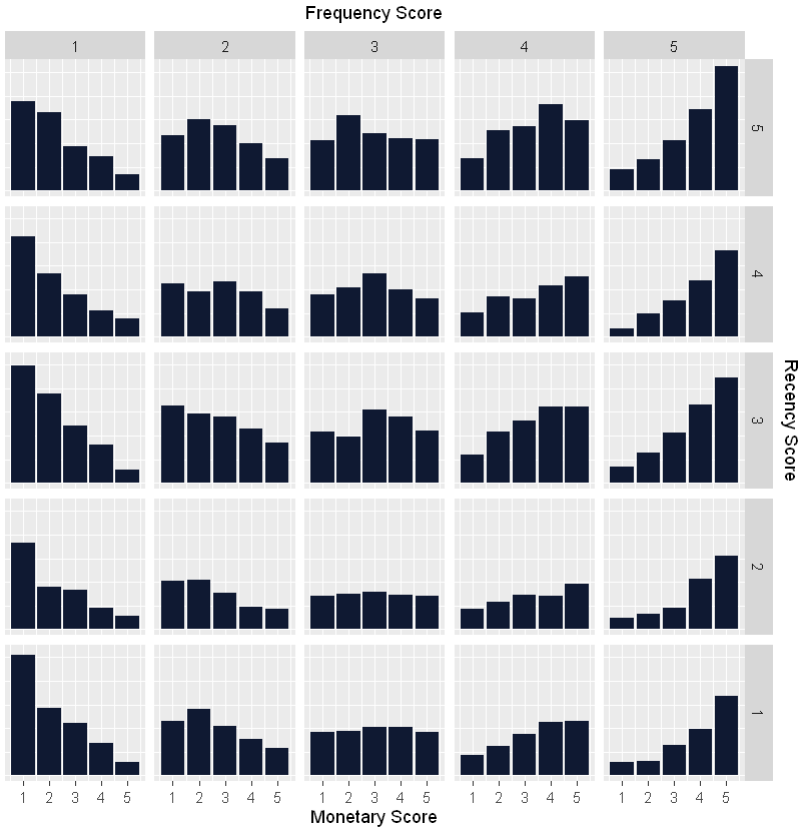

Frequency Distribution
1200
900
600
300
0
Count
210
240
270
300
Frequency

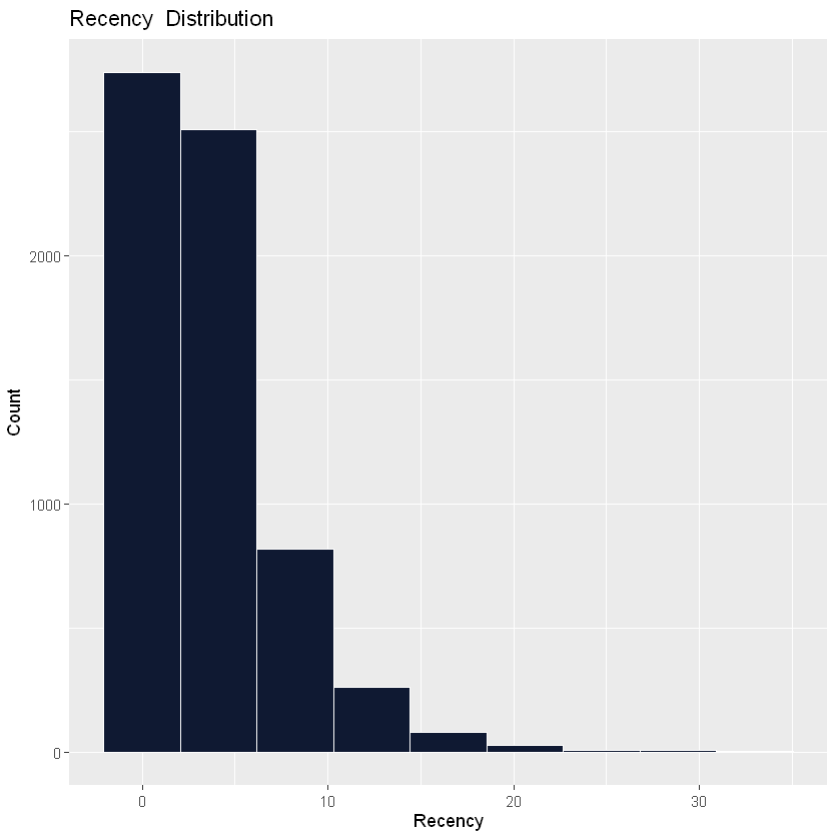

Monetary Value Distribution
1200
900
600
300
0
Count
150000
200000
250000
300000
350000
Monetary Value

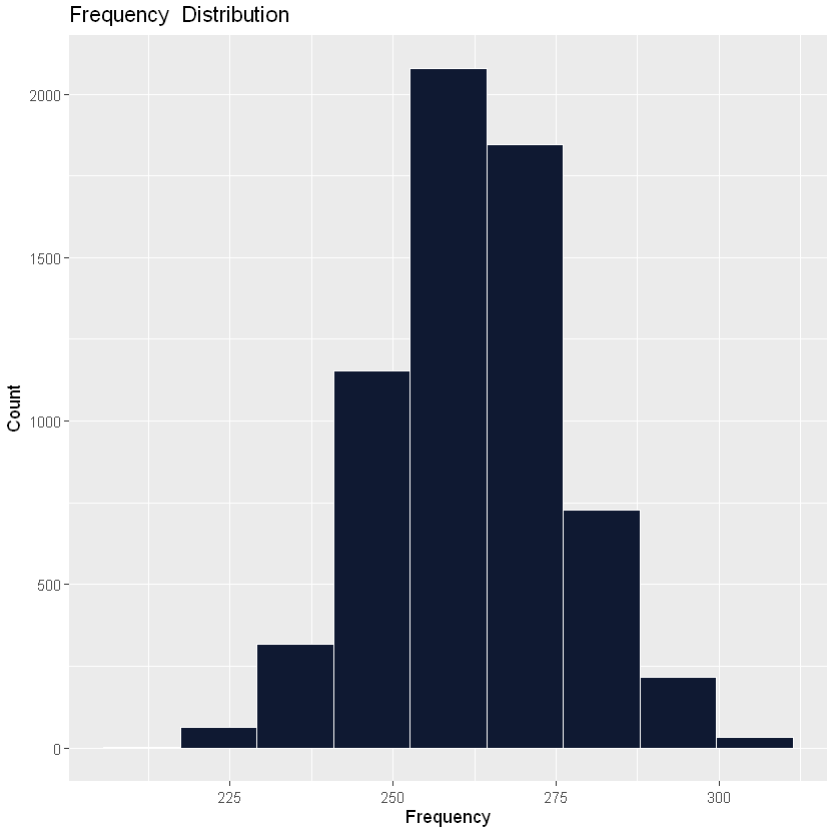


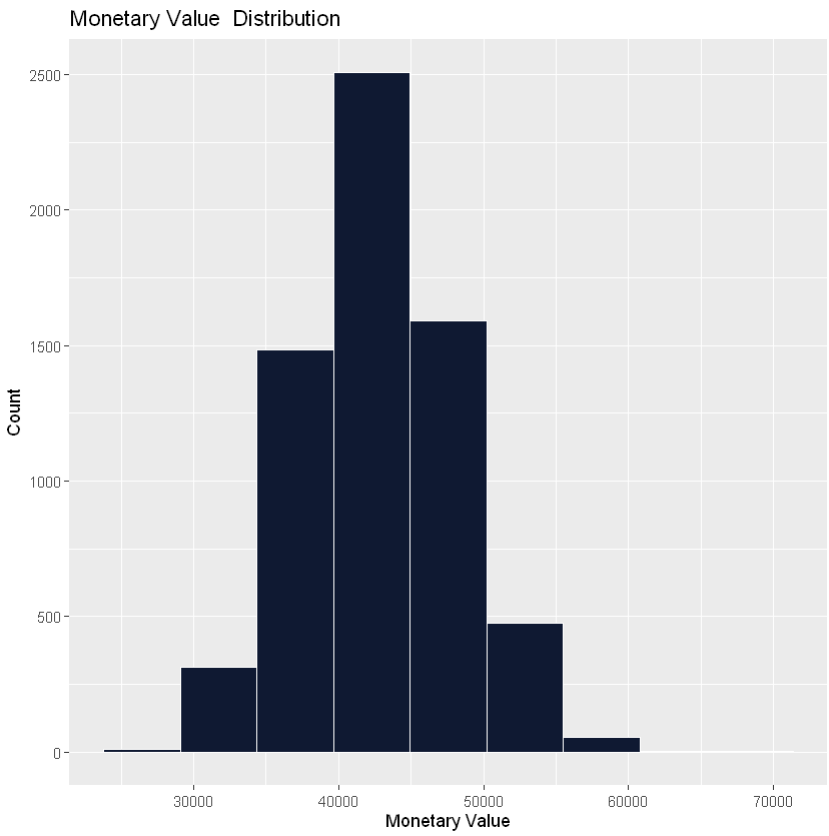


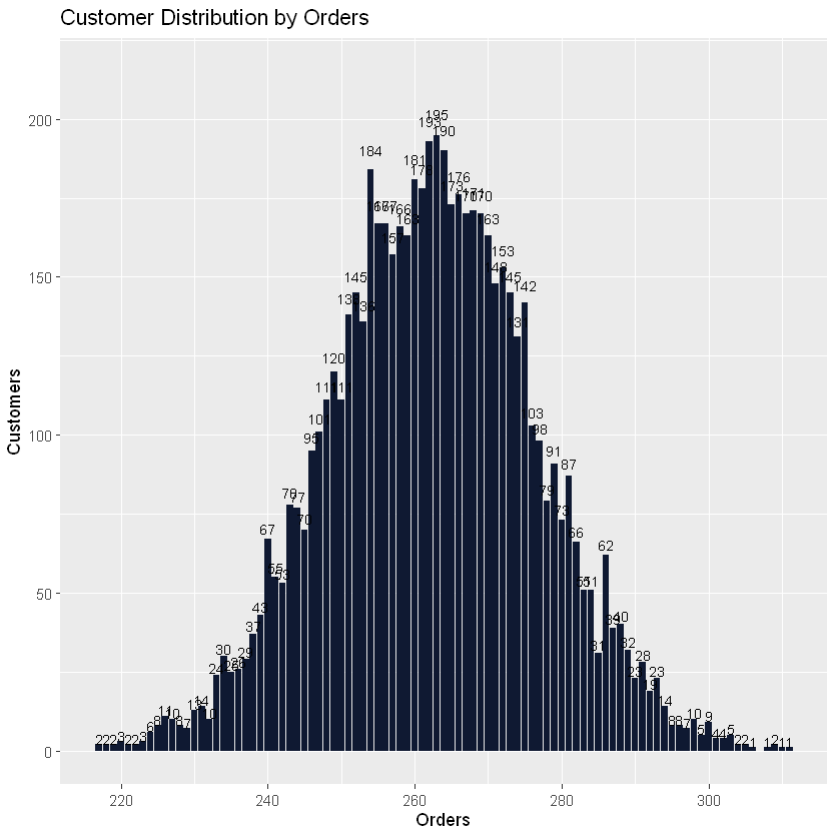
In [11]:
```
# B2C customers
# Same set of RFM visualizations applied to B2C segment
rfm_plot_bar_chart(rfm_result_b2c)
rfm_plot_histogram(rfm_result_b2c, metric = "recency")
rfm_plot_histogram(rfm_result_b2c, metric = "frequency")
rfm_plot_histogram(rfm_result_b2c, metric = "monetary")
rfm_plot_order_dist(rfm_result_b2c)
rfm_plot_heatmap(rfm_result_b2c)
```

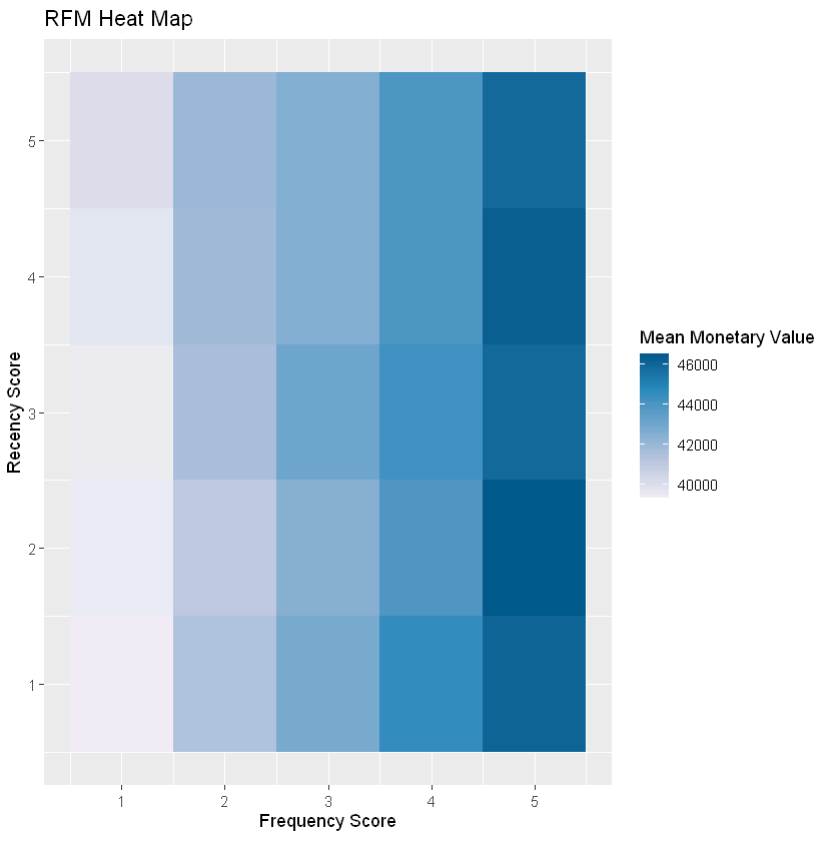

Frequency Score
1
2
3
4
5
Recency Score
Monetary Score


Recency Distribution
Count
2000
1000
0
0
10
20
30
Recency


Frequency Distribution
2000
1500
1000
500
0
Count
225
250
275
300
Frequency


Monetary Value Distribution
2500
2000
1500
1000
500
0
Count
30000
40000
50000
60000
70000
Monetary Value





RFM Segmentation

In [12]:
```
# Manually assign RFM-based customer segments based on score combinations
# These rules are based on the RFM segmentation taught in the lecture
# B2B customers
rfm_segmented_b2b <- rfm_result_b2b$rfm %>%
  mutate(
    segment = case_when(
```

```
      recency_score %in% 4:5 & frequency_score %in% 4:5 & monetary_score %in% 4:
      recency_score %in% 2:5 & frequency_score %in% 3:5 & monetary_score %in% 3:
      recency_score %in% 3:5 & frequency_score %in% 1:3 & monetary_score %in% 1:
      recency_score %in% 4:5 & frequency_score <= 1 & monetary_score <= 1 ~ "New
      recency_score %in% 3:4 & frequency_score <= 1 & monetary_score <= 1 ~ "Pro
      recency_score %in% 2:3 & frequency_score %in% 2:3 & monetary_score %in% 2:
      recency_score %in% 2:3 & frequency_score <= 2 & monetary_score <= 2 ~ "Abo
      recency_score <= 2 & frequency_score %in% 2:5 & monetary_score %in% 2:5 ~
      recency_score <= 1 & frequency_score %in% 4:5 & monetary_score %in% 4:5 ~
      recency_score %in% 1:2 & frequency_score %in% 1:2 & monetary_score %in% 1:
      recency_score <= 2 & frequency_score <= 2 & monetary_score <= 2 ~ "Lost",
      TRUE ~ "Uncategorized"
    )
  )
# B2C customers
rfm_segmented_b2c <- rfm_result_b2c$rfm %>%
  mutate(
    segment = case_when(
      recency_score %in% 4:5 & frequency_score %in% 4:5 & monetary_score %in% 4:
      recency_score %in% 2:5 & frequency_score %in% 3:5 & monetary_score %in% 3:
      recency_score %in% 3:5 & frequency_score %in% 1:3 & monetary_score %in% 1:
      recency_score %in% 4:5 & frequency_score <= 1 & monetary_score <= 1 ~ "New
      recency_score %in% 3:4 & frequency_score <= 1 & monetary_score <= 1 ~ "Pro
      recency_score %in% 2:3 & frequency_score %in% 2:3 & monetary_score %in% 2:
      recency_score %in% 2:3 & frequency_score <= 2 & monetary_score <= 2 ~ "Abo
      recency_score <= 2 & frequency_score %in% 2:5 & monetary_score %in% 2:5 ~
      recency_score <= 1 & frequency_score %in% 4:5 & monetary_score %in% 4:5 ~
      recency_score %in% 1:2 & frequency_score %in% 1:2 & monetary_score %in% 1:
      recency_score <= 2 & frequency_score <= 2 & monetary_score <= 2 ~ "Lost",
      TRUE ~ "Uncategorized"
    )
  )
```

In [14]:

```
# ------------------------------ Segment Summary ------------------------------
# Calculate total number of customers for B2B
total_customers <- nrow(rfm_segmented_b2b)
# Define all possible segments to ensure complete reporting
all_segments <- c(
  "Champions", "Loyal Customers", "Potential Loyalist", "New Customers",
  "Promising", "Need Attention", "About To Sleep", "At Risk",
  "Can’t Lose Them", "Hibernating", "Lost", "Uncategorized"
)
# Aggregate segment metrics for B2B: size, avg. spending, transactions, recency
segment_analysis_b2b <- rfm_segmented_b2b %>%
  group_by(segment) %>%
  summarise(
    no_customers = n(),
    avg_spending = round(mean(amount, na.rm = TRUE), 2),
    avg_transactions = round(mean(transaction_count, na.rm = TRUE), 2),
    avg_recency_days = round(mean(recency_days, na.rm = TRUE), 1),
    .groups = "drop"
  ) %>%
  mutate(
    total_customers = total_customers,
    percentage = round(100 * no_customers / total_customers, 1)
  ) %>%
  right_join(tibble(segment = all_segments), by = "segment") %>%
  replace_na(list(
    no_customers = 0,
    avg_spending = 0,
```

```
    avg_transactions = 0,
    avg_recency_days = 0,
    percentage = 0,
    total_customers = total_customers
  )) %>%
  arrange(factor(segment, levels = all_segments))
```

In [15]:
```
# Repeat the same steps for B2C
total_customers <- nrow(rfm_segmented_b2c)

segment_analysis_b2c <- rfm_segmented_b2c %>%
  group_by(segment) %>%
  summarise(
    no_customers = n(),
    avg_spending = round(mean(amount, na.rm = TRUE), 2),
    avg_transactions = round(mean(transaction_count, na.rm = TRUE), 2),
    avg_recency_days = round(mean(recency_days, na.rm = TRUE), 1),
    .groups = "drop"
  ) %>%
  mutate(
    total_customers = total_customers,
    percentage = round(100 * no_customers / total_customers, 1)
  ) %>%
  right_join(tibble(segment = all_segments), by = "segment") %>%
  replace_na(list(
    no_customers = 0,
    avg_spending = 0,
    avg_transactions = 0,
    avg_recency_days = 0,
    percentage = 0,
    total_customers = total_customers
  )) %>%
  arrange(factor(segment, levels = all_segments))
```

In [16]:
```
# ------------------------------ Output Tables ------------------------------
# Summary tables per segment for reporting
kable(segment_analysis_b2b, caption = "RFM Segment Summary – B2B", digits = 2)
kable(segment_analysis_b2c, caption = "RFM Segment Summary – B2C", digits = 2)

# Display a few examples of customers that could not be assigned to a segment
rfm_segmented_b2c %>%
  filter(segment == "Uncategorized") %>%
  select(customer_id, recency_score, frequency_score, monetary_score, amount, tr
  head(10)
```

Table: RFM Segment Summary – B2B

segment	no_customers	avg_spending	avg_transactions	avg_recency_days	total_customers	percentage
Champions	346	286637.7	278.65	1.4	3563	9.7
Loyal Customers	861	270568.3	270.97	3.2	3563	24.2
Potential Loyalist	923	230295.2	252.08	2.2	3563	25.9
New Customers	0	0.0	0.00	0.0	3563	0.0
Promising	0	0.0	0.00	0.0	3563	0.0
Need Attention	69	241827.5	258.45	5.4	3563	1.9
About To Sleep	96	218367.9	246.46	5.4	3563	2.7
At Risk	505	264282.5	267.70	9.5	3563	14.2
Can’t Lose Them	0	0.0	0.00	0.0	3563	0.0
Hibernating	152	217886.3	244.72	10.3	3563	4.3
Lost	0	0.0	0.00	0.0	3563	0.0
Uncategorized	611	249902.5	260.27	3.9	3563	17.1

Table: RFM Segment Summary – B2C

segment	no_customers	avg_spending	avg_transactions	avg_recency_days	total_customers	percentage
Champions	652	48376.31	277.98	1.4	6437	10.1
Loyal Customers	1589	45834.36	271.18	3.3	6437	24.7
Potential Loyalist	1646	38665.34	251.68	2.2	6437	25.6
New Customers	0	0.00	0.00	0.0	6437	0.0
Promising	0	0.00	0.00	0.0	6437	0.0
Need Attention	127	40608.60	258.06	5.4	6437	2.0
About To Sleep	187	36348.92	245.88	5.5	6437	2.9
At Risk	844	44800.99	267.18	9.4	6437	13.1
Can’t Lose Them	0	0.00	0.00	0.0	6437	0.0
Hibernating	257	36537.57	244.69	10.5	6437	4.0
Lost	0	0.00	0.00	0.0	6437	0.0
Uncategorized	1135	42193.40	260.33	3.9	6437	17.6

A tibble: 10 × 7

customer_id	recency_score	frequency_score	monetary_score	amount	transaction_count
<chr>	<int>	<int>	<int>	<dbl>	<int>
CUS1001	5	4	1	35749.85	268
CUS1003	5	2	4	45890.74	254
CUS1015	4	4	2	41273.83	268
CUS1039	5	5	1	34475.56	275
CUS1040	4	4	1	32852.74	273
CUS1041	4	4	2	40552.85	270
CUS1050	4	5	1	35897.40	283
CUS1056	2	4	1	37257.30	268
CUS1087	1	3	1	36951.99	261
CUS1107	4	4	1	29230.56	267

Custom RFM Segmentation

In [17]:

```r
# Define extended set of RFM segments (custom categories)
# This includes standard segments and new ones like:
# "Active High Value", "Dormant High Value", "Occasional Shoppers", etc.
all_segments <- c(
  "Champions", "Loyal Customers", "Potential Loyalist",
  "New Customers", "Need Attention", "Promising",
  "About To Sleep", "At Risk", "Can't Lose Them",
  "Hibernating", "Lost", "Active High Value",
  "Active Medium Value", "Dormant High Value",
  "Occasional Shoppers", "Dormant Big Spenders"
)

# B2C Segmentation

# Assign each B2C customer to a segment based on detailed RFM scoring logic
rfm_segmented_b2c <- rfm_result_b2c$rfm %>%
  mutate(
    segment = case_when(
      recency_score >= 5 & frequency_score >= 5 & monetary_score >= 5 ~ "Champio
      recency_score >= 4 & frequency_score >= 4 & monetary_score >= 4 ~ "Loyal C
      recency_score >= 4 & frequency_score >= 3 & monetary_score >= 3 ~ "Potenti
      recency_score >= 4 & frequency_score >= 2 & monetary_score >= 2 ~ "New Cus
      recency_score >= 3 & frequency_score == 2 ~ "Occasional Shoppers",
      recency_score >= 3 & frequency_score >= 3 & monetary_score >= 3 ~ "Need At
      recency_score >= 3 & frequency_score == 1 ~ "Promising",
      recency_score >= 2 & frequency_score >= 3 ~ "At Risk",
      recency_score >= 2 & frequency_score == 2 ~ "About To Sleep",
      recency_score >= 1 & frequency_score >= 3 ~ "Can't Lose Them",
      recency_score >= 1 & frequency_score <= 2 & monetary_score <= 3 ~ "Hiberna
      recency_score <= 3 & frequency_score <= 1 & monetary_score <= 2 ~ "Lost",
      recency_score >= 3 & monetary_score >= 3 ~ "Active High Value",
      recency_score >= 3 & monetary_score >= 2 ~ "Active Medium Value",
      recency_score < 3 & monetary_score >= 2 ~ "Dormant High Value",
      TRUE ~ "Uncategorized"
    )
  )

# Calculate total number of B2C customers (needed for percentage computation)
total_customers <- nrow(rfm_segmented_b2c)

# Aggregate statistics per segment for B2C
segment_analysis_b2c <- rfm_segmented_b2c %>%
  group_by(segment) %>%
  summarise(
    no_customers = n(),
    avg_spending = round(mean(amount, na.rm = TRUE), 2),
    avg_transactions = round(mean(transaction_count, na.rm = TRUE), 2),
    avg_recency_days = round(mean(recency_days, na.rm = TRUE), 1),
    .groups = "drop"
  ) %>%
  mutate(
    total_customers = total_customers,
    percentage = round(100 * no_customers / total_customers, 1)
  ) %>%
  right_join(tibble(segment = all_segments), by = "segment") %>%
  replace_na(list(
    no_customers = 0,
    avg_spending = 0,
    avg_transactions = 0,
```

```
    avg_recency_days = 0,
    percentage = 0,
    total_customers = total_customers
  )) %>%
  arrange(factor(segment, levels = all_segments))
```

In [18]:

```
# B2B Segmentation
# Apply same custom logic to B2B customers
rfm_segmented_b2b <- rfm_result_b2b$rfm %>%
  mutate(
    segment = case_when(
      recency_score >= 5 & frequency_score >= 5 & monetary_score >= 5 ~ "Champio
      recency_score >= 4 & frequency_score >= 4 & monetary_score >= 4 ~ "Loyal C
      recency_score >= 4 & frequency_score >= 3 & monetary_score >= 3 ~ "Potenti
      recency_score >= 4 & frequency_score >= 2 & monetary_score >= 2 ~ "New Cus
      recency_score >= 3 & frequency_score == 2 ~ "Occasional Shoppers",
      recency_score >= 3 & frequency_score >= 3 & monetary_score >= 3 ~ "Need At
      recency_score >= 3 & frequency_score == 1 ~ "Promising",
      recency_score >= 2 & frequency_score >= 3 ~ "At Risk",
      recency_score >= 2 & frequency_score == 2 ~ "About To Sleep",
      recency_score >= 1 & frequency_score >= 3 ~ "Can’t Lose Them",
      recency_score >= 1 & frequency_score <= 2 & monetary_score <= 3 ~ "Hiberna
      recency_score <= 3 & frequency_score <= 1 & monetary_score <= 2 ~ "Lost",
      recency_score >= 3 & monetary_score >= 3 ~ "Active High Value",
      recency_score >= 3 & monetary_score >= 2 ~ "Active Medium Value",
      recency_score < 3 & monetary_score >= 2 ~ "Dormant High Value",
      TRUE ~ "Uncategorized"
    )
  )

# Calculate total number of B2B customers
total_customers <- nrow(rfm_segmented_b2b)

# Aggregate statistics per segment for B2B
segment_analysis_b2b <- rfm_segmented_b2b %>%
  group_by(segment) %>%
  summarise(
    no_customers = n(),
    avg_spending = round(mean(amount, na.rm = TRUE), 2),
    avg_transactions = round(mean(transaction_count, na.rm = TRUE), 2),
    avg_recency_days = round(mean(recency_days, na.rm = TRUE), 1),
    .groups = "drop"
  ) %>%
  mutate(
    total_customers = total_customers,
    percentage = round(100 * no_customers / total_customers, 1)
  ) %>%
  right_join(tibble(segment = all_segments), by = "segment") %>%
  replace_na(list(
    no_customers = 0,
    avg_spending = 0,
    avg_transactions = 0,
    avg_recency_days = 0,
    percentage = 0,
    total_customers = total_customers
  )) %>%
  arrange(factor(segment, levels = all_segments))
```

In [19]:
```
# ------------------------- Display Tables ----------------------------------
# Output final segment analysis tables
kable(segment_analysis_b2b, caption = "RFM Analysis – B2B")
kable(segment_analysis_b2c, caption = "RFM Analysis – B2C")
```

Table: RFM Analysis – B2B

segment	no_customers	avg_spending	avg_transactions	avg_recency_days	total_customers	percentage
Champions	74	298748.1	284.18	1.0	3563	2.1
Loyal Customers	272	283342.9	277.15	1.5	3563	7.6
Potential Loyalist	299	263104.2	267.87	1.4	3563	8.4
New Customers	361	248663.4	261.05	1.5	3563	10.1
Need Attention	369	273916.7	272.87	3.4	3563	10.4
Promising	493	236679.6	244.05	2.1	3563	13.8
About To Sleep	91	244986.9	255.12	5.4	3563	2.6
At Risk	545	242683.4	269.68	4.2	3563	15.3
Can’t Lose Them	384	262006.2	271.51	10.1	3563	10.8
Hibernating	334	228209.3	246.63	9.0	3563	9.4
Lost	0	0.0	0.00	0.0	3563	0.0
Active High Value	0	0.0	0.00	0.0	3563	0.0
Active Medium Value	0	0.0	0.00	0.0	3563	0.0
Dormant High Value	93	274826.7	250.49	9.0	3563	2.6
Occasional Shoppers	248	239119.5	255.75	2.8	3563	7.0
Dormant Big Spenders	0	0.0	0.00	0.0	3563	0.0

Table: RFM Analysis – B2C

segment	no_customers	avg_spending	avg_transactions	avg_recency_days	total_customers	percentage
Champions	132	50804.62	283.86	1.0	6437	2.1
Loyal Customers	520	47759.89	276.48	1.5	6437	8.1
Potential Loyalist	527	44447.95	267.71	1.4	6437	8.2
New Customers	709	42069.03	261.40	1.4	6437	11.0
Need Attention	679	46423.69	272.65	3.4	6437	10.5
Promising	877	39725.04	243.83	2.2	6437	13.6
About To Sleep	186	41095.01	255.69	5.4	6437	2.9
At Risk	958	41183.06	270.35	4.1	6437	14.9
Can’t Lose Them	646	44344.23	270.96	9.8	6437	10.0
Hibernating	612	38367.77	246.15	8.9	6437	9.5
Lost	0	0.00	0.00	0.0	6437	0.0
Active High Value	0	0.00	0.00	0.0	6437	0.0
Active Medium Value	0	0.00	0.00	0.0	6437	0.0
Dormant High Value	151	46831.07	249.87	9.1	6437	2.3
Occasional Shoppers	440	40028.77	255.54	2.9	6437	6.8
Dormant Big Spenders	0	0.00	0.00	0.0	6437	0.0

Multiple Linear Regression

In [20]:
```r
# -------------------- Top 10 Products by Revenue (B2B and B2C) ----------------
# B2B: Calculate total revenue per product and return the top 10
top_revenue_b2b <- line_wise_b2b %>%
  group_by(product) %>%
  summarise(
    total_revenue = sum(total_price, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(desc(total_revenue)) %>%
  slice_head(n = 10)
# B2C: Same logic applied to B2C customers
top_revenue_b2c <- line_wise_b2c %>%
  group_by(product) %>%
  summarise(
    total_revenue = sum(total_price, na.rm = TRUE),
    .groups = "drop"
```

```
  ) %>%
  arrange(desc(total_revenue)) %>%
  slice_head(n = 10)
kable(top_revenue_b2b, caption = "Top 10 Products by Revenue – B2B", digits = 2)
kable(top_revenue_b2c, caption = "Top 10 Products by Revenue – B2C", digits = 2)
```

Table: Top 10 Products by Revenue – B2B

product	total_revenue
Moët & Chandon	139543793
Veuve Clicquot	137528584
Johnnie Walker	75422676
Jack Daniels	66066217
Tanqueray	65537046
Havana Club	50321112
Bacardi	50282108
Riesling	19929136
Chardonnay	18193599
Sauvignon Blanc	18080395

Table: Top 10 Products by Revenue – B2C

product	total_revenue
Veuve Clicquot	65053315
Moët & Chandon	35544293
Jack Daniels	29814786
Johnnie Walker	21839125
Tanqueray	20606609
Bacardi	8185396
Havana Club	7725853
Cranberry Juice	5860585
Tomato Juice	5448523
Rotkäppchen Sekt	5401697

MLR per Product

To identify the impact of discount, region, and month on quantity sold

In [32]:
```
# Prepare B2B data: define region, discount, and month as predictors
b2b_prepped <- line_wise_b2b %>%
  mutate(
    region   = as.factor(region),
    product  = as.factor(product),
    discount = as.numeric(discount),
    month    = factor(format(order_date, "%m")) # adds seasonal effect
  ) %>%
  select(product, quantity, discount, region, month)

# Define a list of top-selling products for B2B
top_products_b2b <- c(
  "Moët & Chandon", "Veuve Clicquot", "Johnnie Walker", "Jack Daniels",
  "Tanqueray", "Havana Club", "Bacardi", "Riesling", "Chardonnay",
  "Sauvignon Blanc"
)

# Store regression summaries for each product
```

```r
model_summaries_b2b <- list()

for (prod in top_products_b2b) {
  df_product <- b2b_prepped %>% filter(product == prod)

  if (nrow(df_product) >= 50) { # Ensure sufficient sample size
    model_b2b <- lm(quantity ~ discount + region + month, data = df_product)
    model_summaries_b2b[[prod]] <- summary(model_b2b)
  }
}

# Output regression results per product
for (prod in names(model_summaries_b2b)) {
  cat("\n============================================\n")
  cat("Regression Summary for:", prod, "\n")
  cat("============================================\n")
  print(model_summaries_b2b[[prod]])
}
```

```
========================================
Regression Summary for: Moët & Chandon
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-79.162 -18.488   0.308  18.681  65.819

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 14.093005   0.722052  19.518   <2e-16 ***
discount                   428.153693   3.650070 117.300   <2e-16 ***
regionBayern                 0.236709   0.714150   0.331   0.7403
regionBerlin                 1.595406   0.701322   2.275   0.0229 *
regionBrandenburg            0.487704   0.706469   0.690   0.4900
regionBremen                -0.179946   0.736223  -0.244   0.8069
regionHamburg                0.783017   0.689596   1.135   0.2562
regionHessen                -1.083263   0.686308  -1.578   0.1145
regionMecklenburg-Vorpommern 1.304782   0.698827   1.867   0.0619 .
regionNiedersachsen         -0.167752   0.718700  -0.233   0.8154
regionNordrhein-Westfalen   -0.464631   0.718517  -0.647   0.5179
regionRheinland-Pfalz        0.681972   0.713432   0.956   0.3391
regionSaarland               0.211781   0.697368   0.304   0.7614
regionSachsen                0.518367   0.719127   0.721   0.4710
regionSachsen-Anhalt         0.006386   0.724012   0.009   0.9930
regionSchleswig-Holstein    -0.056651   0.721917  -0.078   0.9375
regionThüringen              0.693046   0.709794   0.976   0.3289
month02                      0.205582   0.616196   0.334   0.7387
month03                      0.250845   0.597759   0.420   0.6747
month04                     -0.314866   0.604910  -0.521   0.6027
month05                     -0.374115   0.607883  -0.615   0.5383
month06                     -0.236064   0.607229  -0.389   0.6975
month07                      0.847617   0.602988   1.406   0.1598
month08                      0.909860   0.598558   1.520   0.1285
month09                      0.220568   0.602847   0.366   0.7145
month10                      0.656214   0.598503   1.096   0.2729
month11                      0.842569   0.603847   1.395   0.1629
month12                      0.391583   0.602892   0.650   0.5160
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.94 on 40743 degrees of freedom
Multiple R-squared:  0.2531,    Adjusted R-squared:  0.2526
F-statistic: 511.3 on 27 and 40743 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Veuve Clicquot
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-79.519 -18.697   0.363  18.710  64.927
```

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  14.902694   0.724241  20.577   <2e-16 ***
discount                    426.249745   3.657041 116.556   <2e-16 ***
regionBayern                 -0.612039   0.710719  -0.861   0.3892
regionBerlin                 -0.343230   0.704118  -0.487   0.6259
regionBrandenburg             0.481918   0.716938   0.672   0.5015
regionBremen                 -0.590547   0.735311  -0.803   0.4219
regionHamburg                 0.055797   0.686863   0.081   0.9353
regionHessen                  0.825016   0.690192   1.195   0.2320
regionMecklenburg-Vorpommern  1.170118   0.703308   1.664   0.0962 .
regionNiedersachsen          -0.529106   0.722475  -0.732   0.4640
regionNordrhein-Westfalen    -0.357786   0.719985  -0.497   0.6192
regionRheinland-Pfalz         0.681774   0.705774   0.966   0.3341
regionSaarland                0.314694   0.697013   0.451   0.6516
regionSachsen                -0.733701   0.721691  -1.017   0.3093
regionSachsen-Anhalt         -0.107426   0.725557  -0.148   0.8823
regionSchleswig-Holstein     -0.320071   0.718801  -0.445   0.6561
regionThüringen              -0.189339   0.698289  -0.271   0.7863
month02                       0.508834   0.611711   0.832   0.4055
month03                       0.191535   0.597758   0.320   0.7487
month04                       0.207505   0.610536   0.340   0.7340
month05                      -0.408282   0.600773  -0.680   0.4968
month06                       0.109215   0.609527   0.179   0.8578
month07                      -0.286021   0.602905  -0.474   0.6352
month08                       0.001226   0.603973   0.002   0.9984
month09                       0.200634   0.606196   0.331   0.7407
month10                      -0.141442   0.605240  -0.234   0.8152
month11                      -0.237890   0.605380  -0.393   0.6944
month12                      -0.107716   0.600329  -0.179   0.8576
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.99 on 40693 degrees of freedom
Multiple R-squared:  0.2508,    Adjusted R-squared:  0.2503
F-statistic: 504.5 on 27 and 40693 DF,  p-value: < 2.2e-16


==========================================
Regression Summary for: Johnnie Walker
==========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-80.019 -18.278   0.256  18.480  65.365

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  13.2360     0.7052  18.770   <2e-16 ***
discount                    439.3066     3.6173 121.446   <2e-16 ***
regionBayern                  0.1618     0.6997   0.231   0.8172
regionBerlin                  0.9694     0.6947   1.395   0.1629
regionBrandenburg             0.6164     0.6987   0.882   0.3777
regionBremen                  0.0113     0.7304   0.015   0.9877
regionHamburg                -0.3788     0.6743  -0.562   0.5743
regionHessen                  0.1570     0.6782   0.231   0.8170
regionMecklenburg-Vorpommern  0.5553     0.6936   0.801   0.4234
```

```
regionNiedersachsen            1.3382     0.7012   1.909   0.0563 .
regionNordrhein-Westfalen      1.0365     0.6979   1.485   0.1375
regionRheinland-Pfalz          0.3005     0.6970   0.431   0.6664
regionSaarland                 0.5376     0.6783   0.792   0.4281
regionSachsen                  0.3220     0.7031   0.458   0.6470
regionSachsen-Anhalt           0.6204     0.7146   0.868   0.3853
regionSchleswig-Holstein       0.9744     0.7077   1.377   0.1686
regionThüringen               -0.5608     0.6860  -0.817   0.4137
month02                        0.5490     0.6128   0.896   0.3704
month03                        0.4907     0.5964   0.823   0.4106
month04                        0.3522     0.5998   0.587   0.5571
month05                        1.1356     0.6007   1.890   0.0587 .
month06                       -0.1876     0.6023  -0.311   0.7554
month07                        0.6369     0.5970   1.067   0.2861
month08                       -0.5042     0.5994  -0.841   0.4003
month09                       -0.3051     0.6021  -0.507   0.6124
month10                        0.5294     0.5949   0.890   0.3735
month11                        0.1675     0.5983   0.280   0.7795
month12                        0.4215     0.5983   0.705   0.4811
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.77 on 40598 degrees of freedom
Multiple R-squared:  0.267,     Adjusted R-squared:  0.2665
F-statistic: 547.6 on 27 and 40598 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Jack Daniels
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
   Min     1Q Median     3Q    Max
-79.49 -18.67   0.45  18.83  65.09

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  14.86522    0.70318  21.140   <2e-16 ***
discount                    429.89957    3.62783 118.500   <2e-16 ***
regionBayern                 -0.48863    0.69663  -0.701    0.483
regionBerlin                 -0.47936    0.69066  -0.694    0.488
regionBrandenburg             0.14438    0.70311   0.205    0.837
regionBremen                 -0.89719    0.73041  -1.228    0.219
regionHamburg                -0.27313    0.68095  -0.401    0.688
regionHessen                 -1.04903    0.68319  -1.535    0.125
regionMecklenburg-Vorpommern  0.05150    0.69095   0.075    0.941
regionNiedersachsen          -0.24736    0.70673  -0.350    0.726
regionNordrhein-Westfalen    -0.77188    0.70457  -1.096    0.273
regionRheinland-Pfalz        -0.30891    0.70372  -0.439    0.661
regionSaarland               -0.52571    0.68676  -0.765    0.444
regionSachsen                 0.64178    0.70414   0.911    0.362
regionSachsen-Anhalt         -0.14867    0.71014  -0.209    0.834
regionSchleswig-Holstein      0.14315    0.71935   0.199    0.842
regionThüringen              -0.71189    0.69064  -1.031    0.303
month02                      -0.67473    0.60676  -1.112    0.266
month03                       0.19818    0.59636   0.332    0.740
month04                      -0.24509    0.59678  -0.411    0.681
```

```
month05                       -0.67785    0.59243  -1.144    0.253
month06                        0.31223    0.60025   0.520    0.603
month07                        0.49873    0.59694   0.835    0.403
month08                        0.07801    0.59701   0.131    0.896
month09                        0.12812    0.59681   0.215    0.830
month10                       -0.06361    0.59196  -0.107    0.914
month11                        0.20958    0.59969   0.349    0.727
month12                        0.04837    0.59198   0.082    0.935
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.97 on 41193 degrees of freedom
Multiple R-squared:  0.2546,    Adjusted R-squared:  0.2541
F-statistic: 521.1 on 27 and 41193 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Tanqueray
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-79.095 -18.541   0.371  18.681  65.070

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 14.6512     0.7111  20.604   <2e-16 ***
discount                   432.1756     3.6511 118.369   <2e-16 ***
regionBayern                -0.7713     0.7072  -1.091   0.2754
regionBerlin                 0.3874     0.6943   0.558   0.5768
regionBrandenburg            0.3288     0.7089   0.464   0.6428
regionBremen                -0.6421     0.7309  -0.879   0.3797
regionHamburg               -1.2015     0.6859  -1.752   0.0798 .
regionHessen                 0.5206     0.6831   0.762   0.4460
regionMecklenburg-Vorpommern -1.1130     0.6944  -1.603   0.1090
regionNiedersachsen         -0.9654     0.7005  -1.378   0.1681
regionNordrhein-Westfalen   -0.4172     0.7177  -0.581   0.5610
regionRheinland-Pfalz       -1.3303     0.6972  -1.908   0.0564 .
regionSaarland               0.3098     0.6840   0.453   0.6506
regionSachsen               -0.7985     0.7049  -1.133   0.2573
regionSachsen-Anhalt        -0.6988     0.7246  -0.964   0.3349
regionSchleswig-Holstein    -1.4620     0.7096  -2.060   0.0394 *
regionThüringen              0.1897     0.6930   0.274   0.7843
month02                      0.7413     0.6093   1.217   0.2237
month03                     -0.5480     0.5978  -0.917   0.3593
month04                      0.1350     0.6029   0.224   0.8228
month05                     -0.9417     0.5972  -1.577   0.1149
month06                      0.3071     0.5963   0.515   0.6066
month07                     -0.5093     0.5970  -0.853   0.3936
month08                      0.1823     0.5980   0.305   0.7605
month09                      0.1260     0.6015   0.209   0.8341
month10                      0.2023     0.5949   0.340   0.7339
month11                      0.4385     0.6032   0.727   0.4673
month12                     -0.4545     0.6021  -0.755   0.4503
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 24.89 on 40832 degrees of freedom
Multiple R-squared:  0.2562,    Adjusted R-squared:  0.2557
F-statistic: 520.9 on 27 and 40832 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Havana Club
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-79.398 -18.553   0.295  18.429  65.675

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  13.96251    0.70651  19.763   <2e-16 ***
discount                    440.87127    3.61643 121.908   <2e-16 ***
regionBayern                  0.66441    0.69736   0.953   0.3407
regionBerlin                 -0.05188    0.69680  -0.074   0.9407
regionBrandenburg             0.62239    0.70851   0.878   0.3797
regionBremen                 -0.55416    0.72864  -0.761   0.4469
regionHamburg                -0.15826    0.67489  -0.235   0.8146
regionHessen                  0.15648    0.68151   0.230   0.8184
regionMecklenburg-Vorpommern  0.18250    0.68909   0.265   0.7911
regionNiedersachsen          -0.06123    0.70141  -0.087   0.9304
regionNordrhein-Westfalen     0.28261    0.70104   0.403   0.6869
regionRheinland-Pfalz        -0.02817    0.69988  -0.040   0.9679
regionSaarland                0.07602    0.68585   0.111   0.9117
regionSachsen                 0.13655    0.70651   0.193   0.8467
regionSachsen-Anhalt          0.29123    0.71512   0.407   0.6838
regionSchleswig-Holstein      0.23107    0.71138   0.325   0.7453
regionThüringen               0.27213    0.69391   0.392   0.6949
month02                      -0.34955    0.61429  -0.569   0.5693
month03                      -0.55451    0.59222  -0.936   0.3491
month04                      -0.42011    0.59994  -0.700   0.4838
month05                      -0.31722    0.59671  -0.532   0.5950
month06                      -0.38151    0.59924  -0.637   0.5243
month07                       0.15600    0.59147   0.264   0.7920
month08                      -0.68649    0.59658  -1.151   0.2499
month09                      -0.35271    0.60195  -0.586   0.5579
month10                      -1.12686    0.59560  -1.892   0.0585 .
month11                      -1.03827    0.60371  -1.720   0.0855 .
month12                      -0.57588    0.59976  -0.960   0.3370
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.78 on 40969 degrees of freedom
Multiple R-squared:  0.2664,    Adjusted R-squared:  0.2659
F-statistic: 551.1 on 27 and 40969 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Bacardi
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)
```

```
Residuals:
   Min     1Q Median     3Q    Max
-79.52 -18.38   0.32  18.48  65.69

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   13.46152    0.71097  18.934   <2e-16 ***
discount                     434.02983    3.63683 119.343   <2e-16 ***
regionBayern                   0.38213    0.70433   0.543   0.5874
regionBerlin                  -0.23138    0.69965  -0.331   0.7409
regionBrandenburg             -0.35688    0.69837  -0.511   0.6093
regionBremen                  -0.94638    0.73016  -1.296   0.1949
regionHamburg                 -0.05188    0.67507  -0.077   0.9387
regionHessen                   0.32431    0.68460   0.474   0.6357
regionMecklenburg-Vorpommern  -0.27746    0.69087  -0.402   0.6880
regionNiedersachsen           -0.36965    0.70949  -0.521   0.6024
regionNordrhein-Westfalen      0.23990    0.70853   0.339   0.7349
regionRheinland-Pfalz         -0.40542    0.69550  -0.583   0.5600
regionSaarland                 0.02110    0.68467   0.031   0.9754
regionSachsen                  1.34297    0.70270   1.911   0.0560 .
regionSachsen-Anhalt           0.09062    0.71252   0.127   0.8988
regionSchleswig-Holstein       0.10159    0.71413   0.142   0.8869
regionThüringen                1.26311    0.69219   1.825   0.0680 .
month02                        0.94995    0.61519   1.544   0.1226
month03                        0.79471    0.59501   1.336   0.1817
month04                        0.09616    0.60641   0.159   0.8740
month05                        0.01008    0.60464   0.017   0.9867
month06                        1.56352    0.60532   2.583   0.0098 **
month07                        1.12457    0.60010   1.874   0.0609 .
month08                        1.09457    0.59950   1.826   0.0679 .
month09                        1.10688    0.60460   1.831   0.0671 .
month10                        0.61251    0.60556   1.011   0.3118
month11                        0.66606    0.60326   1.104   0.2696
month12                        0.46972    0.60030   0.782   0.4339
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.85 on 40895 degrees of freedom
Multiple R-squared:  0.2591,    Adjusted R-squared:  0.2586
F-statistic: 529.6 on 27 and 40895 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Riesling
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-79.583 -18.644   0.297  18.598  65.039

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   13.68155    0.69568  19.666  < 2e-16 ***
discount                     427.46482    3.49367 122.354  < 2e-16 ***
regionBayern                   1.81314    0.69953   2.592  0.00955 **
regionBerlin                   0.37086    0.69717   0.532  0.59476
```

```
regionBrandenburg             -0.02086    0.70132  -0.030  0.97627
regionBremen                   0.14751    0.72846   0.203  0.83953
regionHamburg                 -0.29125    0.67576  -0.431  0.66647
regionHessen                   1.25443    0.67594   1.856  0.06349 .
regionMecklenburg-Vorpommern   0.90032    0.69359   1.298  0.19427
regionNiedersachsen            0.69342    0.70635   0.982  0.32625
regionNordrhein-Westfalen     -0.49277    0.71111  -0.693  0.48834
regionRheinland-Pfalz          0.86721    0.58803   1.475  0.14028
regionSaarland                 0.62355    0.68150   0.915  0.36021
regionSachsen                  0.01766    0.69985   0.025  0.97987
regionSachsen-Anhalt           0.83995    0.71675   1.172  0.24125
regionSchleswig-Holstein       0.24036    0.71100   0.338  0.73532
regionThüringen                0.49864    0.69470   0.718  0.47289
month02                        0.96829    0.59366   1.631  0.10289
month03                        1.20553    0.57322   2.103  0.03546 *
month04                       -0.02248    0.58071  -0.039  0.96912
month05                       -0.01709    0.57238  -0.030  0.97618
month06                        0.65137    0.57900   1.125  0.26060
month07                        0.62787    0.57245   1.097  0.27273
month08                        0.59468    0.57568   1.033  0.30161
month09                        0.83118    0.58119   1.430  0.15269
month10                        1.00168    0.57388   1.745  0.08091 .
month11                        0.19789    0.58191   0.340  0.73381
month12                        0.78132    0.57659   1.355  0.17540
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.9 on 44416 degrees of freedom
Multiple R-squared:  0.2527,    Adjusted R-squared:  0.2522
F-statistic: 556.1 on 27 and 44416 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Chardonnay
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-79.236 -18.788   0.379  19.006  64.300

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  15.896211   0.714918  22.235   <2e-16 ***
discount                    425.450845   3.657883 116.311   <2e-16 ***
regionBayern                 -0.800657   0.698860  -1.146   0.2519
regionBerlin                 -0.994445   0.699520  -1.422   0.1551
regionBrandenburg            -0.978562   0.714935  -1.369   0.1711
regionBremen                 -0.806889   0.733671  -1.100   0.2714
regionHamburg                -1.064598   0.681405  -1.562   0.1182
regionHessen                 -0.868586   0.687875  -1.263   0.2067
regionMecklenburg-Vorpommern -0.498077   0.689303  -0.723   0.4699
regionNiedersachsen          -0.819826   0.710743  -1.153   0.2487
regionNordrhein-Westfalen    -0.229060   0.719043  -0.319   0.7501
regionRheinland-Pfalz        -0.337338   0.709281  -0.476   0.6344
regionSaarland               -0.239647   0.687260  -0.349   0.7273
regionSachsen                -0.482231   0.714329  -0.675   0.4996
regionSachsen-Anhalt         -0.679207   0.718432  -0.945   0.3445
```

```
regionSchleswig-Holstein      -1.239889   0.710443  -1.745   0.0810 .
regionThüringen               -1.163909   0.699981  -1.663   0.0964 .
month02                        0.138219   0.615121   0.225   0.8222
month03                        0.003331   0.600835   0.006   0.9956
month04                        0.271656   0.607083   0.447   0.6545
month05                        0.751286   0.601665   1.249   0.2118
month06                        0.092614   0.600589   0.154   0.8774
month07                       -0.178672   0.599890  -0.298   0.7658
month08                        0.053951   0.598148   0.090   0.9281
month09                       -0.184228   0.606527  -0.304   0.7613
month10                       -0.141196   0.597890  -0.236   0.8133
month11                       -0.228630   0.598017  -0.382   0.7022
month12                       -0.212030   0.600397  -0.353   0.7240
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.09 on 41069 degrees of freedom
Multiple R-squared:  0.2482,    Adjusted R-squared:  0.2477
F-statistic: 502.1 on 27 and 41069 DF,  p-value: < 2.2e-16


========================================
Regression Summary for: Sauvignon Blanc
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-79.778 -18.756   0.507  18.665  65.410

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 14.18112    0.71315  19.885   <2e-16 ***
discount                   432.26388    3.64928 118.452   <2e-16 ***
regionBayern                 0.44026    0.71149   0.619   0.5361
regionBerlin                -0.34055    0.69722  -0.488   0.6252
regionBrandenburg           -0.48919    0.71260  -0.686   0.4924
regionBremen                -0.06533    0.74413  -0.088   0.9300
regionHamburg               -0.17635    0.67919  -0.260   0.7951
regionHessen                -0.20172    0.68404  -0.295   0.7681
regionMecklenburg-Vorpommern  0.48857    0.69266   0.705   0.4806
regionNiedersachsen          0.83048    0.71433   1.163   0.2450
regionNordrhein-Westfalen    0.96836    0.71150   1.361   0.1735
regionRheinland-Pfalz       -0.52415    0.70855  -0.740   0.4595
regionSaarland               0.35323    0.69365   0.509   0.6106
regionSachsen                0.04445    0.70517   0.063   0.9497
regionSachsen-Anhalt        -0.52943    0.72271  -0.733   0.4638
regionSchleswig-Holstein     0.59825    0.72095   0.830   0.4067
regionThüringen             -0.38796    0.69741  -0.556   0.5780
month02                      0.29945    0.61816   0.484   0.6281
month03                      0.92689    0.60110   1.542   0.1231
month04                      0.37668    0.61081   0.617   0.5374
month05                     -0.19017    0.60405  -0.315   0.7529
month06                      0.26185    0.60794   0.431   0.6667
month07                      0.09822    0.60471   0.162   0.8710
month08                     -1.13939    0.60267  -1.891   0.0587 .
month09                     -0.07212    0.61256  -0.118   0.9063
month10                      0.46484    0.60153   0.773   0.4397
```

```
month11                      -0.01477    0.60819  -0.024   0.9806
month12                      -0.38663    0.60658  -0.637   0.5239
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.05 on 41050 degrees of freedom
Multiple R-squared:  0.2555,    Adjusted R-squared:  0.255
F-statistic: 521.9 on 27 and 41050 DF,  p-value: < 2.2e-16
```

In [33]:

```r
b2c_prepped <- line_wise_b2c %>%
  mutate(
    region   = as.factor(region),
    product  = as.factor(product),
    discount = as.numeric(discount),
    month    = factor(format(order_date, "%m"))
  ) %>%
  select(product, quantity, discount, region, month)

# Define a list of top-selling products for B2C
top_products_b2c <- c(
  "Veuve Clicquot", "Moët & Chandon", "Jack Daniels", "Johnnie Walker",
  "Tanqueray", "Bacardi", "Havana Club", "Cranberry Juice", "Tomato Juice",
  "Rotkäppchen Sekt"
)

# Store regression summaries for each product
model_summaries_b2c <- list()

for (prod in top_products_b2c) {
  df_product <- b2c_prepped %>% filter(product == prod)

  if (nrow(df_product) >= 50) {
    model_b2c <- lm(quantity ~ discount + region + month, data = df_product)
    model_summaries_b2c[[prod]] <- summary(model_b2c)
  }
}

# Output regression results per product
for (prod in names(model_summaries_b2c)) {
  cat("\n==========================================\n")
  cat("Regression Summary for:", prod, "\n")
  cat("==========================================\n")
  print(model_summaries_b2c[[prod]])
}
```

```
========================================
Regression Summary for: Veuve Clicquot
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2691 -3.9539 -0.0261  3.9051  7.2657

Coefficients: (1 not defined because of singularities)
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 8.032687   0.081428  98.648   <2e-16 ***
discount                          NA         NA      NA       NA
regionBayern                0.030239   0.088819   0.340   0.7335
regionBerlin               -0.096100   0.089943  -1.068   0.2853
regionBrandenburg          -0.186910   0.090368  -2.068   0.0386 *
regionBremen                0.106696   0.093107   1.146   0.2518
regionHamburg              -0.077091   0.087599  -0.880   0.3788
regionHessen               -0.014950   0.091097  -0.164   0.8696
regionMecklenburg-Vorpommern -0.041695   0.090414  -0.461   0.6447
regionNiedersachsen        -0.015802   0.087066  -0.181   0.8560
regionNordrhein-Westfalen  -0.028012   0.088795  -0.315   0.7524
regionRheinland-Pfalz       0.115740   0.090125   1.284   0.1991
regionSaarland              0.046225   0.088906   0.520   0.6031
regionSachsen              -0.031922   0.087914  -0.363   0.7165
regionSachsen-Anhalt       -0.009142   0.090208  -0.101   0.9193
regionSchleswig-Holstein   -0.131531   0.087736  -1.499   0.1338
regionThüringen            -0.090528   0.088965  -1.018   0.3089
month02                    -0.006574   0.078851  -0.083   0.9336
month03                    -0.111499   0.077216  -1.444   0.1487
month04                     0.025926   0.077813   0.333   0.7390
month05                    -0.009589   0.076868  -0.125   0.9007
month06                     0.032006   0.077408   0.413   0.6793
month07                     0.120655   0.077286   1.561   0.1185
month08                     0.017308   0.077068   0.225   0.8223
month09                     0.105893   0.077443   1.367   0.1715
month10                     0.120080   0.077328   1.553   0.1205
month11                     0.020248   0.077240   0.262   0.7932
month12                     0.055161   0.077634   0.711   0.4774
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.317 on 74116 degrees of freedom
Multiple R-squared:  0.0005227, Adjusted R-squared:  0.0001721
F-statistic: 1.491 on 26 and 74116 DF,  p-value: 0.05139


========================================
Regression Summary for: Moët & Chandon
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1658 -3.9615 -0.0028  3.9257  7.1398
```

```
Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 7.9731484  0.0813260  98.039   <2e-16 ***
discount                           NA         NA      NA       NA
regionBayern                0.1786718  0.0883453   2.022   0.0431 *
regionBerlin                0.0996393  0.0889774   1.120   0.2628
regionBrandenburg           0.0723883  0.0900572   0.804   0.4215
regionBremen                0.1341516  0.0936339   1.433   0.1519
regionHamburg               0.1790726  0.0873683   2.050   0.0404 *
regionHessen               -0.0070771  0.0902877  -0.078   0.9375
regionMecklenburg-Vorpommern  0.0287511  0.0912507   0.315   0.7527
regionNiedersachsen         0.0314070  0.0867322   0.362   0.7173
regionNordrhein-Westfalen  -0.0131678  0.0880922  -0.149   0.8812
regionRheinland-Pfalz       0.1089061  0.0893715   1.219   0.2230
regionSaarland              0.1403371  0.0887065   1.582   0.1136
regionSachsen               0.0419417  0.0876295   0.479   0.6322
regionSachsen-Anhalt        0.1112342  0.0905975   1.228   0.2195
regionSchleswig-Holstein   -0.0008748  0.0874226  -0.010   0.9920
regionThüringen             0.0657567  0.0889028   0.740   0.4595
month02                    -0.0360793  0.0788310  -0.458   0.6472
month03                    -0.0094499  0.0769242  -0.123   0.9022
month04                    -0.0434324  0.0777733  -0.558   0.5765
month05                    -0.0602515  0.0762862  -0.790   0.4296
month06                    -0.0048435  0.0775802  -0.062   0.9502
month07                     0.0064297  0.0762560   0.084   0.9328
month08                     0.0014981  0.0764842   0.020   0.9844
month09                    -0.0613098  0.0775865  -0.790   0.4294
month10                    -0.0745587  0.0766184  -0.973   0.3305
month11                     0.0136181  0.0770972   0.177   0.8598
month12                    -0.0998155  0.0768362  -1.299   0.1939
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.314 on 74109 degrees of freedom
Multiple R-squared:  0.0002843, Adjusted R-squared:  -6.644e-05
F-statistic: 0.8106 on 26 and 74109 DF,  p-value: 0.7381


========================================
Regression Summary for: Jack Daniels
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2215 -3.9324 -0.0009  3.9315  7.2379

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 7.923548   0.081999  96.630   <2e-16 ***
discount                          NA         NA      NA       NA
regionBayern                0.110239   0.089536   1.231   0.2182
regionBerlin               -0.004368   0.090072  -0.048   0.9613
regionBrandenburg           0.044963   0.089967   0.500   0.6172
regionBremen               -0.105209   0.093757  -1.122   0.2618
regionHamburg              -0.011899   0.087666  -0.136   0.8920
regionHessen                0.117950   0.091441   1.290   0.1971
regionMecklenburg-Vorpommern  0.030901   0.090779   0.340   0.7336
```

```
regionNiedersachsen           0.040725   0.087475   0.466   0.6415
regionNordrhein-Westfalen     0.130680   0.089129   1.466   0.1426
regionRheinland-Pfalz         0.110887   0.090511   1.225   0.2205
regionSaarland                0.020515   0.089681   0.229   0.8191
regionSachsen                 0.024856   0.087574   0.284   0.7765
regionSachsen-Anhalt          0.046805   0.090417   0.518   0.6047
regionSchleswig-Holstein      0.015357   0.088389   0.174   0.8621
regionThüringen               0.032607   0.088478   0.369   0.7125
month02                       0.043470   0.078547   0.553   0.5800
month03                       0.049523   0.077065   0.643   0.5205
month04                       0.095847   0.077355   1.239   0.2153
month05                       0.030537   0.076729   0.398   0.6906
month06                       0.167254   0.077652   2.154   0.0313 *
month07                      -0.031223   0.077200  -0.404   0.6859
month08                      -0.023767   0.076816  -0.309   0.7570
month09                       0.044330   0.078042   0.568   0.5700
month10                      -0.056250   0.076837  -0.732   0.4641
month11                       0.100021   0.077304   1.294   0.1957
month12                       0.072384   0.077313   0.936   0.3492
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.308 on 74095 degrees of freedom
Multiple R-squared:  0.0003645, Adjusted R-squared:  1.371e-05
F-statistic: 1.039 on 26 and 74095 DF,  p-value: 0.4085


========================================
Regression Summary for: Johnnie Walker
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1982 -3.9610 -0.0163  3.9365  7.2451

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 8.0609798  0.0814852  98.926   <2e-16 ***
discount                           NA         NA      NA       NA
regionBayern               -0.1389064  0.0886544  -1.567   0.1172
regionBerlin               -0.0547705  0.0893954  -0.613   0.5401
regionBrandenburg           0.0154222  0.0904826   0.170   0.8647
regionBremen               -0.0339433  0.0935154  -0.363   0.7166
regionHamburg              -0.0406306  0.0876804  -0.463   0.6431
regionHessen               -0.0748253  0.0906399  -0.826   0.4091
regionMecklenburg-Vorpommern  0.0018211  0.0906322   0.020   0.9840
regionNiedersachsen        -0.1963471  0.0868952  -2.260   0.0238 *
regionNordrhein-Westfalen  -0.0297348  0.0887097  -0.335   0.7375
regionRheinland-Pfalz      -0.0460527  0.0902236  -0.510   0.6098
regionSaarland             -0.0152922  0.0892710  -0.171   0.8640
regionSachsen              -0.0013768  0.0875215  -0.016   0.9874
regionSachsen-Anhalt       -0.0244157  0.0909456  -0.268   0.7883
regionSchleswig-Holstein   -0.0292391  0.0883260  -0.331   0.7406
regionThüringen            -0.0451285  0.0889471  -0.507   0.6119
month02                     0.0322235  0.0791370   0.407   0.6839
month03                    -0.0154834  0.0770039  -0.201   0.8406
month04                     0.0694693  0.0772626   0.899   0.3686
```

```
month05                      -0.0948997  0.0769190  -1.234   0.2173
month06                      -0.0539523  0.0772054  -0.699   0.4847
month07                       0.0375105  0.0763372   0.491   0.6232
month08                       0.0380878  0.0763253   0.499   0.6178
month09                      -0.0236719  0.0776523  -0.305   0.7605
month10                      -0.1097597  0.0771338  -1.423   0.1547
month11                       0.0007305  0.0770270   0.009   0.9924
month12                       0.1218145  0.0767022   1.588   0.1123
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.325 on 74432 degrees of freedom
Multiple R-squared:  0.0003753, Adjusted R-squared:  2.613e-05
F-statistic: 1.075 on 26 and 74432 DF,  p-value: 0.3612


========================================
Regression Summary for: Tanqueray
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2049 -3.9481 -0.0025  3.9240  7.1843

Coefficients: (1 not defined because of singularities)
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  7.87688    0.08112  97.101  < 2e-16 ***
discount                          NA         NA      NA       NA
regionBayern                -0.01242    0.08836  -0.141  0.88825
regionBerlin                 0.02637    0.08851   0.298  0.76575
regionBrandenburg            0.04596    0.09035   0.509  0.61099
regionBremen                 0.02666    0.09239   0.289  0.77290
regionHamburg               -0.02462    0.08701  -0.283  0.77716
regionHessen                -0.02840    0.09022  -0.315  0.75296
regionMecklenburg-Vorpommern -0.02141    0.09084  -0.236  0.81367
regionNiedersachsen         -0.04704    0.08721  -0.539  0.58959
regionNordrhein-Westfalen    0.03609    0.08852   0.408  0.68353
regionRheinland-Pfalz        0.09128    0.08983   1.016  0.30953
regionSaarland               0.09373    0.08929   1.050  0.29388
regionSachsen                0.05441    0.08692   0.626  0.53132
regionSachsen-Anhalt        -0.02360    0.09041  -0.261  0.79411
regionSchleswig-Holstein     0.10112    0.08718   1.160  0.24610
regionThüringen              0.01598    0.08822   0.181  0.85628
month02                     -0.01414    0.07896  -0.179  0.85786
month03                      0.04918    0.07661   0.642  0.52095
month04                      0.11893    0.07704   1.544  0.12267
month05                      0.09800    0.07615   1.287  0.19813
month06                      0.09585    0.07721   1.241  0.21447
month07                      0.13694    0.07659   1.788  0.07377 .
month08                      0.22694    0.07680   2.955  0.00313 **
month09                      0.20411    0.07713   2.646  0.00814 **
month10                      0.12352    0.07664   1.612  0.10705
month11                      0.15406    0.07704   2.000  0.04553 *
month12                      0.12022    0.07706   1.560  0.11873
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 4.304 on 74302 degrees of freedom
Multiple R-squared:  0.0003659, Adjusted R-squared:  1.614e-05
F-statistic: 1.046 on 26 and 74302 DF,  p-value: 0.3989


========================================
Regression Summary for: Bacardi
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1633 -3.9254 -0.0017  3.9584  7.1704

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  8.070728   0.081950  98.483   <2e-16 ***
discount                           NA         NA      NA       NA
regionBayern                -0.159969   0.089340  -1.791   0.0734 .
regionBerlin                -0.069044   0.090210  -0.765   0.4441
regionBrandenburg           -0.030390   0.090121  -0.337   0.7360
regionBremen                -0.058927   0.093428  -0.631   0.5282
regionHamburg               -0.030148   0.087735  -0.344   0.7311
regionHessen                -0.030348   0.091041  -0.333   0.7389
regionMecklenburg-Vorpommern -0.037505   0.090757  -0.413   0.6794
regionNiedersachsen         -0.150258   0.087077  -1.726   0.0844 .
regionNordrhein-Westfalen    0.005997   0.088849   0.067   0.9462
regionRheinland-Pfalz       -0.059913   0.089723  -0.668   0.5043
regionSaarland              -0.125203   0.089344  -1.401   0.1611
regionSachsen               -0.177200   0.087290  -2.030   0.0424 *
regionSachsen-Anhalt        -0.156866   0.091497  -1.714   0.0865 .
regionSchleswig-Holstein    -0.018021   0.088190  -0.204   0.8381
regionThüringen             -0.076654   0.088488  -0.866   0.3863
month02                      0.004119   0.079017   0.052   0.9584
month03                     -0.030924   0.077589  -0.399   0.6902
month04                     -0.024122   0.077746  -0.310   0.7564
month05                     -0.006566   0.077275  -0.085   0.9323
month06                     -0.057505   0.077913  -0.738   0.4605
month07                      0.001232   0.077182   0.016   0.9873
month08                      0.086596   0.077311   1.120   0.2627
month09                     -0.053478   0.077811  -0.687   0.4919
month10                     -0.063902   0.077224  -0.827   0.4080
month11                      0.011562   0.077775   0.149   0.8818
month12                     -0.014804   0.077231  -0.192   0.8480
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.316 on 74018 degrees of freedom
Multiple R-squared:  0.0002731, Adjusted R-squared:  -7.809e-05
F-statistic: 0.7776 on 26 and 74018 DF,  p-value: 0.7811


========================================
Regression Summary for: Havana Club
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-7.1843 -3.9451 -0.0087  3.9405  7.2000

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  7.9879992  0.0816468  97.836   <2e-16 ***
discount                            NA         NA      NA       NA
regionBayern                -0.0558154  0.0893062  -0.625    0.532
regionBerlin                 0.0459995  0.0896781   0.513    0.608
regionBrandenburg            0.0808468  0.0903907   0.894    0.371
regionBremen                 0.0638827  0.0940010   0.680    0.497
regionHamburg               -0.0612688  0.0870518  -0.704    0.482
regionHessen                 0.0694729  0.0904893   0.768    0.443
regionMecklenburg-Vorpommern 0.0235366  0.0907191   0.259    0.795
regionNiedersachsen          0.1399598  0.0869408   1.610    0.107
regionNordrhein-Westfalen   -0.0137159  0.0894774  -0.153    0.878
regionRheinland-Pfalz       -0.0034727  0.0903029  -0.038    0.969
regionSaarland               0.0880005  0.0894655   0.984    0.325
regionSachsen                0.0169219  0.0876121   0.193    0.847
regionSachsen-Anhalt         0.0556129  0.0908211   0.612    0.540
regionSchleswig-Holstein     0.0275938  0.0877861   0.314    0.753
regionThüringen             -0.0812300  0.0889479  -0.913    0.361
month02                     -0.0013445  0.0785052  -0.017    0.986
month03                      0.0159043  0.0762838   0.208    0.835
month04                     -0.0069267  0.0773354  -0.090    0.929
month05                     -0.0312545  0.0764091  -0.409    0.683
month06                     -0.0515542  0.0770588  -0.669    0.503
month07                      0.0244307  0.0768859   0.318    0.751
month08                      0.0100586  0.0763525   0.132    0.895
month09                      0.0563879  0.0772497   0.730    0.465
month10                     -0.0006927  0.0762627  -0.009    0.993
month11                     -0.1067688  0.0770196  -1.386    0.166
month12                      0.0141164  0.0769692   0.183    0.854
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.315 on 74408 degrees of freedom
Multiple R-squared:  0.000271,  Adjusted R-squared:  -7.835e-05
F-statistic: 0.7757 on 26 and 74408 DF,  p-value: 0.7835


========================================
Regression Summary for: Cranberry Juice
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1028 -3.9688  0.0009  3.9758  7.1132

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  7.939174   0.049045 161.875  < 2e-16 ***
discount                           NA         NA      NA       NA
regionBayern                -0.007795   0.053584  -0.145  0.88434
regionBerlin                 0.026652   0.054256   0.491  0.62326
```

```
regionBrandenburg            -0.036653   0.054349  -0.674  0.50006
regionBremen                 -0.009085   0.053782  -0.169  0.86586
regionHamburg                 0.042061   0.053073   0.793  0.42806
regionHessen                  0.013523   0.055520   0.244  0.80756
regionMecklenburg-Vorpommern -0.023835   0.054837  -0.435  0.66382
regionNiedersachsen           0.005046   0.053222   0.095  0.92447
regionNordrhein-Westfalen    -0.036069   0.053069  -0.680  0.49671
regionRheinland-Pfalz        -0.024156   0.053638  -0.450  0.65245
regionSaarland                0.017117   0.053998   0.317  0.75125
regionSachsen                 0.004813   0.052879   0.091  0.92747
regionSachsen-Anhalt         -0.003501   0.054117  -0.065  0.94842
regionSchleswig-Holstein     -0.052420   0.052990  -0.989  0.32255
regionThüringen               0.020557   0.054439   0.378  0.70571
month02                       0.079765   0.047413   1.682  0.09250 .
month03                       0.045394   0.046139   0.984  0.32519
month04                       0.080102   0.046641   1.717  0.08591 .
month05                       0.090697   0.046069   1.969  0.04899 *
month06                       0.042786   0.046515   0.920  0.35767
month07                       0.068996   0.046081   1.497  0.13433
month08                       0.060793   0.046222   1.315  0.18844
month09                       0.084859   0.046460   1.826  0.06778 .
month10                       0.024597   0.046114   0.533  0.59377
month11                       0.045926   0.046571   0.986  0.32406
month12                       0.121607   0.046454   2.618  0.00885 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.323 on 205632 degrees of freedom
Multiple R-squared:  8.62e-05,  Adjusted R-squared:  -4.023e-05
F-statistic: 0.6818 on 26 and 205632 DF,  p-value: 0.8855


========================================
Regression Summary for: Tomato Juice
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1013 -3.9549  0.0062  3.9674  7.2154

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  7.893962   0.049266 160.232  < 2e-16 ***
discount                           NA         NA      NA       NA
regionBayern                 0.114200   0.053764   2.124  0.03366 *
regionBerlin                 0.108095   0.054323   1.990  0.04661 *
regionBrandenburg            0.160252   0.054557   2.937  0.00331 **
regionBremen                 0.160437   0.053716   2.987  0.00282 **
regionHamburg                0.140798   0.053052   2.654  0.00796 **
regionHessen                 0.138868   0.055920   2.483  0.01302 *
regionMecklenburg-Vorpommern 0.101359   0.054964   1.844  0.06517 .
regionNiedersachsen          0.164047   0.053439   3.070  0.00214 **
regionNordrhein-Westfalen    0.107567   0.053293   2.018  0.04355 *
regionRheinland-Pfalz        0.119713   0.053676   2.230  0.02573 *
regionSaarland               0.161010   0.054061   2.978  0.00290 **
regionSachsen                0.143454   0.052824   2.716  0.00661 **
regionSachsen-Anhalt         0.088448   0.054282   1.629  0.10322
```

```
regionSchleswig-Holstein      0.104176   0.053154   1.960  0.05001 .
regionThüringen               0.099641   0.054380   1.832  0.06690 .
month02                      -0.017217   0.047667  -0.361  0.71795
month03                      -0.043660   0.046253  -0.944  0.34520
month04                       0.043309   0.046706   0.927  0.35378
month05                      -0.009973   0.046485  -0.215  0.83013
month06                      -0.025353   0.046719  -0.543  0.58736
month07                      -0.079816   0.046482  -1.717  0.08595 .
month08                       0.018932   0.046182   0.410  0.68185
month09                       0.027283   0.046665   0.585  0.55878
month10                      -0.075050   0.046337  -1.620  0.10531
month11                      -0.109354   0.046843  -2.334  0.01957 *
month12                      -0.038681   0.046671  -0.829  0.40722
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.322 on 204799 degrees of freedom
Multiple R-squared:  0.0001877, Adjusted R-squared:  6.08e-05
F-statistic: 1.479 on 26 and 204799 DF,  p-value: 0.05493


========================================
Regression Summary for: Rotkäppchen Sekt
========================================

Call:
lm(formula = quantity ~ discount + region + month, data = df_product)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1117 -3.9667  0.0021  3.9664  7.1183

Coefficients: (1 not defined because of singularities)
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 7.918330   0.074253 106.640   <2e-16 ***
discount                          NA         NA      NA       NA
regionBayern                0.075698   0.084698   0.894    0.371
regionBerlin                0.108197   0.084940   1.274    0.203
regionBrandenburg           0.067515   0.086060   0.785    0.433
regionBremen                0.013656   0.088489   0.154    0.877
regionHamburg               0.029547   0.082407   0.359    0.720
regionHessen               -0.024849   0.086272  -0.288    0.773
regionMecklenburg-Vorpommern  0.038724   0.086315   0.449    0.654
regionNiedersachsen         0.086230   0.082120   1.050    0.294
regionNordrhein-Westfalen   0.080344   0.083500   0.962    0.336
regionRheinland-Pfalz       0.047866   0.084873   0.564    0.573
regionSaarland              0.040155   0.083820   0.479    0.632
regionSachsen              -0.014115   0.082231  -0.172    0.864
regionSachsen-Anhalt        0.053928   0.069523   0.776    0.438
regionSchleswig-Holstein    0.099596   0.083274   1.196    0.232
regionThüringen             0.089624   0.084177   1.065    0.287
month02                     0.021375   0.075178   0.284    0.776
month03                     0.069227   0.073415   0.943    0.346
month04                    -0.011732   0.074152  -0.158    0.874
month05                     0.019863   0.073247   0.271    0.786
month06                     0.048651   0.074165   0.656    0.512
month07                     0.073631   0.073631   1.000    0.317
month08                     0.027328   0.073465   0.372    0.710
month09                     0.085213   0.073945   1.152    0.249
month10                     0.025642   0.073420   0.349    0.727
```

```
month11                     -0.005587   0.074056  -0.075    0.940
month12                      0.005638   0.073777   0.076    0.939
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.327 on 81411 degrees of freedom
Multiple R-squared:  0.0001292, Adjusted R-squared:  -0.0001901
F-statistic: 0.4047 on 26 and 81411 DF,  p-value: 0.9969
```

Model Evaluation

In [34]:

```r
# Store evaluation metrics
model_metrics_b2b <- tibble(
  product       = character(),
  train_r2      = numeric(),
  train_adj_r2  = numeric(),
  test_r2       = numeric(),
  rmse_test     = numeric(),
  n_train       = integer(),
  n_test        = integer()
)

for (prod in top_products_b2b) {
  df_product <- b2b_prepped %>% filter(product == prod)

  if (nrow(df_product) >= 50) {
    set.seed(123) # reproducibility

    # Create training (80%) and test (20%) split
    train_index_b2b <- createDataPartition(df_product$quantity, p = 0.8, list =
    train_data_b2b <- df_product[train_index_b2b, ]
    test_data_b2b <- df_product[-train_index_b2b, ]

    # Fit model on training set
    model_b2b <- lm(quantity ~ discount + region + month, data = train_data_b2b)

    # Predict on test set
    predictions_b2b <- predict(model_b2b, newdata = test_data_b2b)

    # Calculate evaluation metrics
    rmse_val_b2b <- rmse(actual = test_data_b2b$quantity, predicted = prediction
    r2_val_b2b <- summary(model_b2b)$r.squared # training R²
    adj_r2_b2b <- summary(model_b2b)$adj.r.squared

    # Compute R² on test set
    ss_total_b2b <- sum((test_data_b2b$quantity - mean(test_data_b2b$quantity))^
    ss_res_b2b <- sum((test_data_b2b$quantity - predictions_b2b)^2)
    test_r2_b2b <- 1 - (ss_res_b2b / ss_total_b2b)

    # Store metrics
    model_metrics_b2b <- model_metrics_b2b %>%
      add_row(
        product       = prod,
        train_r2      = round(r2_val_b2b, 3),
        train_adj_r2  = round(adj_r2_b2b, 3),
        test_r2       = round(test_r2_b2b, 3),
        rmse_test     = round(rmse_val_b2b, 3),
        n_train       = nrow(train_data_b2b),
```

```
        n_test        = nrow(test_data_b2b)
      )
  }
}

# Display evaluation results
kable(model_metrics_b2b, caption = "Model Performance on Test Set - B2B")
```

Table: Model Performance on Test Set - B2B

product	train_r2	train_adj_r2	test_r2	rmse_test	n_train	n_test
Moët & Chandon	0.252	0.252	0.255	24.948	32619	8152
Veuve Clicquot	0.257	0.256	0.226	25.375	32578	8143
Johnnie Walker	0.268	0.267	0.263	24.783	32502	8124
Jack Daniels	0.257	0.256	0.246	25.145	32979	8242
Tanqueray	0.255	0.254	0.261	24.931	32689	8171
Havana Club	0.270	0.269	0.252	24.916	32799	8198
Bacardi	0.261	0.261	0.249	25.028	32740	8183
Riesling	0.253	0.252	0.251	24.987	35556	8888
Chardonnay	0.251	0.250	0.238	25.166	32879	8218
Sauvignon Blanc	0.254	0.254	0.260	25.065	32865	8213

In [35]:
```
#------------------- Model evaluation -------------------
# Store evaluation metrics
model_metrics_b2c <- tibble(
  product       = character(),
  train_r2      = numeric(),
  train_adj_r2  = numeric(),
  test_r2       = numeric(),
  rmse_test     = numeric(),
  n_train       = integer(),
  n_test        = integer()
)

for (prod in top_products_b2c) {
  df_product <- b2c_prepped %>% filter(product == prod)

  if (nrow(df_product) >= 50) {
    set.seed(123) # reproducibility

    # Create training (80%) and test (20%) split
    train_index <- createDataPartition(df_product$quantity, p = 0.8, list = FALS
    train_data <- df_product[train_index, ]
    test_data <- df_product[-train_index, ]

    # Fit model on training set
    model <- lm(quantity ~ discount + region + month, data = train_data)

    # Predict on test set
    predictions <- predict(model, newdata = test_data)

    # Calculate evaluation metrics
    rmse_val <- rmse(actual = test_data$quantity, predicted = predictions)
    r2_val <- summary(model)$r.squared # training R²
    adj_r2 <- summary(model)$adj.r.squared

    # Compute R² on test set
    ss_total <- sum((test_data$quantity - mean(test_data$quantity))^2)
```

```
    ss_res <- sum((test_data$quantity - predictions)^2)
    test_r2 <- 1 - (ss_res / ss_total)

    # Store metrics
    model_metrics <- model_metrics %>%
      add_row(
        product       = prod,
        train_r2      = round(r2_val, 3),
        train_adj_r2  = round(adj_r2, 3),
        test_r2       = round(test_r2, 3),
        rmse_test     = round(rmse_val, 3),
        n_train       = nrow(train_data),
        n_test        = nrow(test_data)
      )
  }
}

# Display evaluation results
kable(model_metrics, caption = "Model Performance on Test Set – B2B")
```

Table: Model Performance on Test Set – B2B

product	train_r2	train_adj_r2	test_r2	rmse_test	n_train	n_test
Rotkäppchen Sekt	0.000	0	0.000	4.321	65151	16287
Veuve Clicquot	0.001	0	-0.001	4.323	59316	14827
Moët & Chandon	0.000	0	0.000	4.315	59311	14825
Jack Daniels	0.000	0	0.000	4.315	59300	14822
Johnnie Walker	0.001	0	-0.001	4.318	59569	14890
Tanqueray	0.000	0	0.000	4.289	59465	14864
Bacardi	0.001	0	-0.001	4.320	59238	14807
Havana Club	0.000	0	0.000	4.325	59550	14885
Cranberry Juice	0.000	0	0.000	4.328	164529	41130
Tomato Juice	0.000	0	0.000	4.324	163863	40963
Rotkäppchen Sekt	0.000	0	0.000	4.331	65151	16287

Weekday vs Weekend analysis

In [36]:
```
#------------------- Seeing if theres a difference in weekdays vs weekends -----
# Only one product is selected for this analysis
df_jack_weekend <- line_wise_b2c %>%
  filter(product == "Jack Daniels") %>%
  mutate(
    region      = as.factor(region),
    discount    = as.numeric(discount),
    unit_price  = as.numeric(unit_price),
    month       = factor(month(order_date)),
    is_weekend  = ifelse(wday(order_date) %in% c(1, 7), 1, 0) # Sonntag (1) und
  ) %>%
  select(quantity, discount, unit_price, region, month, is_weekend) %>%
  na.omit()

# Split & Modell
set.seed(123)
idx <- createDataPartition(df_jack_weekend$quantity, p = 0.8, list = FALSE)
train <- df_jack_weekend[idx, ]
test <- df_jack_weekend[-idx, ]
```

```r
model_weekend <- lm(quantity ~ discount + unit_price + region + month + is_weeke
summary(model_weekend)

# Evaluation
pred <- predict(model_weekend, newdata = test)
rmse <- sqrt(mean((test$quantity - pred)^2))
r2 <- 1 - sum((test$quantity - pred)^2) / sum((test$quantity - mean(test$quantit

cat("RMSE (is_weekend):", round(rmse, 2), "\n")
cat("R² (is_weekend):", round(r2, 4), "\n")
```

```
Call:
lm(formula = quantity ~ discount + unit_price + region + month +
    is_weekend, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2259 -3.9227 -0.0069  3.9203  7.2822

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  8.401287   0.738606  11.375   <2e-16 ***
discount                           NA         NA      NA       NA
unit_price                  -0.009493   0.014588  -0.651   0.5152
regionBayern                 0.006630   0.100175   0.066   0.9472
regionBerlin                -0.026115   0.100749  -0.259   0.7955
regionBrandenburg            0.082576   0.100157   0.824   0.4097
regionBremen                -0.125364   0.104630  -1.198   0.2309
regionHamburg               -0.015420   0.098253  -0.157   0.8753
regionHessen                 0.109414   0.102059   1.072   0.2837
regionMecklenburg-Vorpommern 0.026462   0.101528   0.261   0.7944
regionNiedersachsen          0.072084   0.097926   0.736   0.4617
regionNordrhein-Westfalen    0.135519   0.100008   1.355   0.1754
regionRheinland-Pfalz        0.125751   0.101392   1.240   0.2149
regionSaarland               0.061534   0.099855   0.616   0.5377
regionSachsen                0.029847   0.097776   0.305   0.7602
regionSachsen-Anhalt         0.060980   0.101153   0.603   0.5466
regionSchleswig-Holstein     0.012295   0.098843   0.124   0.9010
regionThüringen              0.054957   0.099014   0.555   0.5789
month2                      -0.007860   0.087828  -0.089   0.9287
month3                       0.073314   0.086347   0.849   0.3959
month4                       0.123468   0.086505   1.427   0.1535
month5                       0.041514   0.085806   0.484   0.6285
month6                       0.149765   0.087032   1.721   0.0853 .
month7                      -0.066778   0.086403  -0.773   0.4396
month8                      -0.033893   0.086068  -0.394   0.6937
month9                       0.064680   0.087507   0.739   0.4598
month10                     -0.062488   0.085928  -0.727   0.4671
month11                      0.129143   0.086570   1.492   0.1358
month12                      0.068941   0.086550   0.797   0.4257
is_weekend                   0.002459   0.039215   0.063   0.9500
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.306 on 59271 degrees of freedom
Multiple R-squared:  0.000478,  Adjusted R-squared:  5.814e-06
F-statistic: 1.012 on 28 and 59271 DF,  p-value: 0.4463
RMSE (is_weekend): 4.32
R² (is_weekend): -5e-04
R² (is_weekend): -5e-04
```