

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



XÁC SUẤT THỐNG KÊ (MT2013)

Báo cáo Bài tập lớn

THỐNG KÊ DỮ LIỆU BÁN LẺ GIAO DỊCH CỦA CỬA HÀNG ĐIỆN TỬ

GVHD: Cô Nguyễn Kiều Dung

SV thực hiện:	Bùi Nguyễn Hoàng Thọ	2333017
	Trần Văn Được	2210782
	Trịnh Duy Nghiêm	2312256

Tp. Hồ Chí Minh, Tháng 8/2025



Tên nhóm: MT01

STT	Họ tên SV	MSSV	Tên lớp	Ngành / Khoa
1	Bùi Nguyễn Hoàng Thọ	2333017	DT03	KH-KT máy tính
2	Trần Văn Được	2210782	DT03	Điện - Điện tử
3	Trịnh Duy Nghiêm	2312256	DT03	KH-KT máy tính

Mục lục

1	Tóm Tắt Dữ Liệu	6
1.1	Ngữ cảnh	6
1.2	Cách dữ liệu được thu thập	6
1.3	Các loại biến và quan trắc	6
2	Cơ sở lý thuyết (Kiến thức nền)	7
2.1	Giới thiệu về công cụ sử dụng thống kê	7
2.1.1	Ngôn ngữ R	7
2.1.2	Công cụ RStudio	7
2.2	Các khái niệm về thành phần và phương pháp sử dụng thống kê	7
2.2.1	Biến định lượng	7
2.2.2	Phân phối chuẩn	7
2.2.3	Kiến thức thống kê mô tả	7
2.3	Kiến thức thống kê suy luận (Tóm tắt)	9
2.3.1	Lý thuyết Kiểm định thống kê	9
2.3.2	Kiểm định z-test cho 1 mẫu	9
2.3.3	Kiểm định t-test cho 2 mẫu độc lập	9
2.3.4	Kiểm định Kruskal-Wallis	10
2.3.5	Kiểm định Wilcoxon-Mann-Whitney	10
2.3.6	Kiểm định Wilcoxon Signed-Rank	10
2.3.7	Phân tích hậu kiểm	11
3	Tiền Xử Lý Dữ Liệu	12
3.1	Ghép dữ liệu	12
3.2	Xác định Dữ liệu Khuyết	13
3.3	Làm sạch dữ liệu	13
3.3.1	Order_price, Order_Total	13
3.3.2	season	14
3.3.3	is_happy_customer	14

3.3.4	Kiểm tra dữ liệu sau khi làm sạch	15
4	Mô Tả Thống Kê	16
4.1	Xử lý các điểm Outlier	16
4.2	Thống kê dữ liệu	18
4.2.1	Ma trận tương quan giữa order_price và total_order	19
4.2.2	Tổng đơn hàng theo mùa	20
4.2.3	Tổng phí giao hàng theo từng mùa	20
5	Thống Kê Suy Diễn	23
5.1	Kiểm định trung bình một mẫu	24
5.1.1	Bài toán thực tế	24
5.1.2	Tính giá trị kiểm định	24
5.1.3	Mô tả dữ liệu và công thức tính	24
5.1.4	Xử lý trên phần mềm RStudio	25
5.1.5	Kết luận	25
5.2	Kiểm định trung bình hai mẫu	26
5.2.1	Bài toán thực tế	26
5.2.2	Tính giá trị kiểm định	26
5.2.3	Mô tả dữ liệu và công thức tính	27
5.2.4	Xử lý trên phần mềm RStudio	27
5.2.5	Kết luận	28
5.3	Wilcoxon Signed rank	28
5.3.1	Mục tiêu	28
5.3.2	Kết quả kiểm định	28
5.3.3	Nhận xét	29
5.3.4	Kết luận	29
6	Thảo luận và Mở rộng	30
6.1	Nhận xét dữ liệu	30
6.2	Tính bao quát	30

6.3	Điểm mạnh và điểm yếu của các phương pháp đã sử dụng	31
6.4	Mối quan hệ giữa mục tiêu và phương pháp được chọn	31
6.5	Mở rộng: Dunn Test	32
6.6	Kết quả	32
7	Nguồn Dữ Liệu	33
8	Tài liệu Tham Khảo	33

Giới thiệu

Trong thời đại khoa học công nghệ phát triển mạnh mẽ, nhu cầu sử dụng linh kiện và thiết bị điện tử ngày càng gia tăng. Điều này đã đặt ra yêu cầu cao hơn về chất lượng sản phẩm và dịch vụ, buộc các cửa hàng điện tử phải đối mặt với sự cạnh tranh khốc liệt. Để thu hút và giữ chân khách hàng, nhiều chương trình khuyến mãi cùng các chính sách ưu đãi được triển khai liên tục, đáp ứng tốt hơn nhu cầu và nâng cao chất lượng đời sống người tiêu dùng.

Để đạt được những mục tiêu này, các cửa hàng điện tử không ngừng đầu tư vào việc phân tích dữ liệu giao dịch bán lẻ. Đây là một bước quan trọng, giúp khảo sát nhu cầu khách hàng, từ đó xây dựng các chiến lược kinh doanh hiệu quả và tối ưu hóa phương thức tiếp cận thị trường.

Với mong muốn có một góc nhìn tổng quát hơn về hoạt động phân tích trong kinh doanh, chúng em đã chọn đề tài “Thống kê dữ liệu bán lẻ giao dịch của cửa hàng điện tử”. Qua đó, chúng em hy vọng có thể tự mình thử nghiệm và áp dụng các kiến thức về xác suất và thống kê đã được tích lũy trong quá trình học tập.

Trong quá trình thực hiện báo cáo, nếu có bất kỳ thiếu sót nào, nhóm rất mong nhận được sự góp ý từ thầy để hoàn thiện tốt hơn.

1 Tóm Tắt Dữ Liệu

1.1 Ngữ cảnh

Bộ dữ liệu được sử dụng lấy từ Kaggle, mô phỏng các giao dịch bán lẻ tại một cửa hàng điện tử. Nó chứa thông tin chi tiết về đơn hàng, khách hàng, sản phẩm mua, và phản hồi của khách hàng.

1.2 Cách dữ liệu được thu thập

Bộ dữ liệu trên là tổng hợp những đơn hàng sản phẩm đã được thanh toán tại một cửa hàng điện tử. Nó đi kèm với ngày thanh toán, thông tin sản phẩm, và thông tin khách hàng.

1.3 Các loại biến và quan trắc

Nhóm đã tổng hợp lại được tổng cộng 1,000 giá trị quan trắc liên quan đến các giao dịch và 3 giá trị liên quan đến thông tin các kho hàng.

Trong quá trình tổng hợp, các biến trong bộ dữ liệu được đưa ra như sau:

- **Biến định tính (Categorical variables):**

- `order_id`: Mã đơn hàng.
- `customer_id`: ID khách hàng.
- `nearest_warehouse`: Tên kho hàng gần nhất.
- `season`: Mùa trong năm.
- `latest_customer_review`: Nhận xét từ khách hàng.

- **Biến định lượng (Numerical variables):**

- `order_price`: Giá trị đơn hàng trước khuyến mãi.
- `delivery_charges`: Phí giao hàng.
- `coupon_discount`: Giá trị khuyến mãi áp dụng.
- `order_total`: Giá trị đơn hàng sau khuyến mãi.
- `customer_lat` và `customer_long`: Vĩ độ và kinh độ của khách hàng.
- `distance_to_nearest_warehouse`: Khoảng cách đến kho gần nhất.

- **Biến Boolean:**

- `is_expedited_delivery`: Có giao hàng nhanh hay không.
- `is_happy_customer`: Khách hàng hài lòng hay không.

2 Cơ sở lý thuyết (Kiến thức nền)

2.1 Giới thiệu về công cụ sử dụng thống kê

2.1.1 Ngôn ngữ R

R là một công cụ rất mạnh cho học máy, thống kê và phân tích dữ liệu. Nó là một ngôn ngữ lập trình, có thể sử dụng trên bất kỳ hệ điều hành nào do tính chất độc lập nền tảng (*platform-independent*). R có khả năng tích hợp với các ngôn ngữ khác như C, C++ và cho phép tương tác với nhiều nguồn dữ liệu cũng như các gói thống kê như SAS, SPSS.

2.1.2 Công cụ RStudio

RStudio là một môi trường phát triển tích hợp (IDE - Integrated Development Environment) dành riêng cho R. RStudio cung cấp giao diện trực quan và các chức năng thuận tiện để làm việc với R mà không cần chạy R trực tiếp.

2.2 Các khái niệm về thành phần và phương pháp sử dụng thống kê

2.2.1 Biến định lượng

Biến định lượng là những biến có thể đo lường được bằng các con số và thực hiện các phép toán số học như cộng, trừ, nhân, chia. Biến định lượng cung cấp thông tin về "mức độ" hoặc "số lượng" của một đặc điểm cụ thể.

2.2.2 Phân phối chuẩn

Khái niệm: Phân phối chuẩn, còn gọi là phân phối Gauss (theo tên nhà toán học Carl Friedrich Gauss), là một loại phân phối liên tục. Đặc điểm nổi bật của phân phối chuẩn là:

- Đối xứng qua giá trị trung bình.
- Có hình dạng chuông.
- Các giá trị được phân bố đều xung quanh trung bình và giảm dần khi xa trung bình.

2.2.3 Kiến thức thống kê mô tả

Các chỉ số đo lường xu hướng tập trung:

- **Trung bình (Mean):** Là giá trị trung tâm của bộ dữ liệu, được tính bằng công thức:

$$\text{Mean} = \frac{\sum x_i}{n}$$

trong đó x_i là các giá trị quan sát, n là số lượng giá trị.

- **Trung vị (Median):** Là giá trị nằm ở giữa khi các giá trị được sắp xếp theo thứ tự tăng dần.
- **Mốt (Mode):** Là giá trị xuất hiện nhiều nhất trong bộ dữ liệu.

Các chỉ số đo lường sự phân tán

- **Phương sai (Variance):** Đo lường mức độ phân tán của dữ liệu xung quanh giá trị trung bình, được tính bằng công thức:

$$\text{Variance} = \frac{\sum (x_i - \bar{x})^2}{n}$$

- **Độ lệch chuẩn (Standard Deviation):** Là căn bậc hai của phương sai:

$$\text{SD} = \sqrt{\text{Variance}}$$

- **Khoảng (Range):** Là hiệu giữa giá trị lớn nhất và giá trị nhỏ nhất trong bộ dữ liệu.

Các chỉ số mô tả sự phân bố

- **Tứ phân vị (Quartiles):** Là các giá trị chia dữ liệu thành bốn phần bằng nhau.
- **Khoảng tứ phân vị hay độ khoảng giữa (Interquartile Range - IQR):** Là khoảng cách giữa tứ phân vị thứ ba (Q_3) và tứ phân vị thứ nhất (Q_1), được tính bằng:

$$\text{IQR} = Q_3 - Q_1$$

IQR là một chỉ số quan trọng giúp xác định sự phân tán dữ liệu và phát hiện các giá trị ngoại lai (outliers). Các giá trị được coi là ngoại lai nếu nằm ngoài khoảng:

$$[Q_1 - 1.5 \times \text{IQR}, Q_3 + 1.5 \times \text{IQR}]$$

Biểu đồ sử dụng thống kê

- **Biểu đồ cột (Barplot):** Dùng so sánh các giá trị khác nhau.
- **Biểu đồ hộp (Box plot):** Thể hiện các tứ phân vị và phát hiện giá trị ngoại lai.
- **Biểu đồ Q-Q (Quantile-Quantile plot):** Kiểm tra dữ liệu có tuân theo phân phối chuẩn hay không.
- **Biểu đồ corrplot:** thể hiện sự tương quan của dữ liệu.

2.3 Kiến thức thống kê suy luận (Tóm tắt)

2.3.1 Lý thuyết Kiểm định thống kê

Kiểm định là quá trình sử dụng dữ liệu mẫu để đưa ra quyết định về một giả thuyết nào đó liên quan đến tổng thể.

- **Giả thuyết không (Null Hypothesis - H_0):** Giả thuyết ban đầu cho rằng không có sự khác biệt hoặc mối quan hệ nào giữa các biến.
- **Giả thuyết thay thế (Alternative Hypothesis - H_1):** Giả thuyết cho rằng có sự khác biệt hoặc mối quan hệ giữa các biến.
- **Mức ý nghĩa (Significance Level - α):** Xác định ngưỡng để bác bỏ giả thuyết không, thường là 0.05 hoặc 0.01.
- **Giá trị p (p-value):** Xác suất để quan sát được dữ liệu mẫu nếu giả thuyết không là đúng. Nếu p-value nhỏ hơn mức ý nghĩa α , ta bác bỏ giả thuyết không.

2.3.2 Kiểm định z-test cho 1 mẫu

2.3.3 Kiểm định t-test cho 2 mẫu độc lập

- **Định nghĩa:** Kiểm định t-test là một phương pháp thống kê dùng để so sánh trung bình của hai nhóm độc lập hoặc hai mẫu liên quan nhằm xác định xem có sự khác biệt có ý nghĩa thống kê giữa chúng hay không.
- **Ý nghĩa:**
 - Phù hợp cho dữ liệu có phân phối tùy ý và mẫu lớn ($n > 30$)
 - Giúp xác định xem hai nhóm có khác biệt về trung bình hay không.

Công thức tổng quát cho hai mẫu độc lập:

$$T = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Trong đó:

- \bar{X}_1, \bar{X}_2 : trung bình mẫu của hai nhóm (ví dụ: mùa xuân và mùa hè)
- s_1, s_2 : độ lệch chuẩn mẫu của hai nhóm
- n_1, n_2 : kích thước mẫu của hai nhóm

Bậc tự do (degrees of freedom) được ước lượng theo công thức Welch–Satterthwaite:

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}}$$

So sánh giá trị $|T|$ với ngưỡng $t_{\alpha/2, df}$ để đưa ra kết luận.

2.3.4 Kiểm định Kruskal-Wallis

- **Định nghĩa:** Kiểm định phi tham số dùng để so sánh trung vị của **ba hoặc nhiều nhóm độc lập** nhằm xác định xem có sự khác biệt đáng kể giữa các nhóm khi dữ liệu không theo phân phối chuẩn.
- **Ý nghĩa:**
 - Phù hợp cho dữ liệu không chuẩn hoặc dữ liệu thứ bậc.
 - Kiểm tra sự khác biệt trung vị giữa các nhóm độc lập.

2.3.5 Kiểm định Wilcoxon-Mann-Whitney

- **Định nghĩa:** Kiểm định phi tham số dùng để so sánh **hai mẫu độc lập**, kiểm tra xem hai mẫu có cùng phân phối hay không.
- **Ý nghĩa:**
 - Thích hợp khi dữ liệu không chuẩn hoặc có ngoại lệ lớn.
 - Ứng dụng cho dữ liệu định lượng hoặc thứ bậc để kiểm tra sự khác biệt về phân phối.

2.3.6 Kiểm định Wilcoxon Signed-Rank

- **Định nghĩa:** Kiểm định phi tham số dùng để so sánh trung vị của **hai tập dữ liệu liên quan (cặp đôi)**, thay thế kiểm định tham số khi dữ liệu không tuân theo phân phối chuẩn.
- **Ý nghĩa:**

- Kiểm tra sự khác biệt giữa hai tập dữ liệu liên quan.
- Thích hợp cho dữ liệu định lượng không chuẩn hoặc dữ liệu thứ bậc.

2.3.7 Phân tích hậu kiểm

Post hoc test: được sử dụng sau khi có kết quả có ý nghĩa thống kê để xác định sự khác biệt giữa các nhóm. Kiểm định **Bonferroni** là phương pháp đơn giản, điều chỉnh mức ý nghĩa α bằng cách chia cho số lượng so sánh, giúp kiểm soát tỷ lệ lỗi loại I khi thực hiện các bài kiểm tra tính độc lập giữa các nhóm.

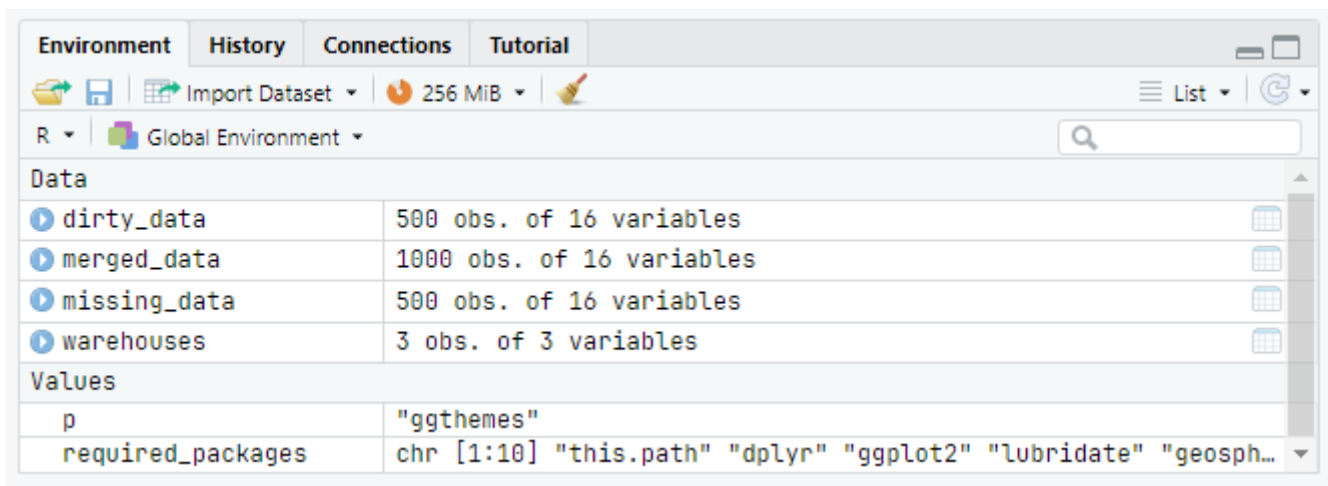
3 Tiền Xử Lý Dữ Liệu

3.1 Ghép dữ liệu

Trước hết, nhóm cần cài đặt các thư viện cần được sử dụng trong suốt quá trình làm sạch và xử lý dữ liệu.

```
1 required_packages <- c("this.path", "dplyr", "ggplot2", "lubridate", "geosphere", "readr", "
  corrplot", "faraway", "car", "ggthemes", "gt", "nortest", "knitr", "FSA", "ggcorrplot", "dunn.
  test")
2 for (p in required_packages) {
3   if (!require(p, character.only = TRUE)) install.packages(p)
4   library(p, character.only = TRUE)
5 }
6 # Load đường dẫn hiện tại của thư mục data chứa các file CSV mẫu
7 setwd(this.path::here())
8 dirty_data <- read_csv("data/dirty_data.csv")
9 missing_data <- read_csv("data/missing_data.csv")
10 # Chuyển định dạng tháng ngày năm cột date
11 dirty_data$date <- parse_date_time(dirty_data$date, orders = c("mdy", "ymd", "dmy"))
12 missing_data$date <- parse_date_time(missing_data$date, orders = c("mdy", "ymd", "dmy"))
13 merged_data <- rbind(dirty_data, missing_data)
```

Listing 1: Ghép dữ liệu trong R



Environment	
History	
Connections	
Tutorial	
Import Dataset	
256 MiB	
List	
R	
Global Environment	
Data	
dirty_data	500 obs. of 16 variables
merged_data	1000 obs. of 16 variables
missing_data	500 obs. of 16 variables
warehouses	3 obs. of 3 variables
Values	
p	"ggthemes"
required_packages	chr [1:10] "this.path" "dplyr" "ggplot2" "lubridate" "geosph...

Hình 1: Sau khi gộp 2 files CSV

Giải thích: Thư viện **this.path** giúp tự động cập nhật đường dẫn thư mục gốc của dự án tại nơi chứa file R. Sau khi tải dữ liệu từ các file CSV mẫu, nhóm gộp hai file này thành một bảng duy nhất để thuận tiện xử lý.

3.2 Xác định Dữ liệu Khuyết

Trong quá trình phân tích dữ liệu mẫu, nhóm đã tiến hành kiểm tra sự tồn tại của các giá trị bị khuyết (NA) trong bộ dữ liệu. Kết quả cho thấy một số cột có chứa giá trị khuyết, cụ thể như sau:

```
1 > na_cout<- colSums(is.na(merged_data ))
2 > print(na_cout)
3           order_id           customer_id           date
4                0                0                0
5 nearest_warehouse shopping_cart order_price
6                10                0                10
7 delivery_charges customer_lat customer_long
8                0                10                10
9 coupon_discount order_total season
10               0                10                10
11 is_expedited_delivery distance_to_nearest_warehouse latest_customer_review
12                0                10                0
13 is_happy_customer
14                10
```

Listing 2: Các cột dữ liệu chứa giá trị bị khuyết

Để phục vụ mục đích làm bài tập lớn và vẽ đồ thị, nhóm đã quyết định bỏ đi ba cột **distance_to_nearest_warehouse**, **customer_long** và **customer_lat** và **nearest_warehouses**.

3.3 Làm sạch dữ liệu

3.3.1 Order_price, Order_Total

Để xử lý các giá trị bị khuyết ở hai cột **order_price** và **order_total**, nhóm áp dụng công thức:

$$\text{order_total} = \frac{\text{order_price} \times (100 - \text{coupon_discount})}{100} + \text{delivery_charges} \quad (1)$$

$$\text{order_price} = \frac{(\text{order_total} - \text{delivery_charges}) \times 100}{100 - \text{coupon_discount}} \quad (2)$$

```
1 merged_data <- merged_data %>%
2   mutate(
3     order_total = ifelse(is.na(order_total), order_price * (100 - coupon_discount) / 100 +
4       delivery_charges, order_total),
5     order_price = ifelse(is.na(order_price), (order_total - delivery_charges) * 100 / (100 -
6       coupon_discount), order_price)
7   )
```

Listing 3: Xử lý cột **order_total** và **order_price**

Giải thích: Hàm **mutate** của thư viện **dplyr** được sử dụng để thêm hoặc cập nhật cột trong bảng dữ liệu.

3.3.2 season

Ở cột **season**, nhóm nhận thấy một số giá trị không đồng nhất về cách viết hoa hoặc viết thường, dẫn đến dữ liệu thiếu đồng bộ được thể hiện dưới đây:

```
1 season_unique_before<-unique(merged_data$season)
2 print(season_unique_before)
```

Listing 4: Giá trị cột **season** ban đầu

```
1 merged_data$season <- tolower(merged_data$season) # Đổi tất cả giá trị mùa dạng viết thường
2 month_value <- month(merged_data$date) # lấy tháng trong cột $date
3 merged_data <- merged_data %>%
4   # Toán tử %>% : Truyền kết quả của phép toán hoặc hàm vào hàm tiếp theo .
5   # mutate: Tạo ra cột mới hoặc thay đổi giá trị các cột trong data frame
6   mutate(season = case_when(
7     !is.na(season) ~ season,
8     month_value %in% c(12, 1, 2) ~ "winter",
9     month_value %in% c(3, 4, 5) ~ "spring",
10    month_value %in% c(6, 7, 8) ~ "summer",
11    TRUE ~ "autumn"
12  ))
```

Listing 5: Xử lý cột **season**

Giải thích: Nhóm đã lấy giá trị **tháng** trong cột **date** của dữ liệu mẫu và sử dụng hàm **case_when** xuất ra mùa và thế vào các giá trị bị khuyết.

```
1 > season_unique_after<-unique(merged_data$season)
2 > print(season_unique_after)
3 [1] "winter" "summer" "autumn" "spring"
```

Listing 6: Dữ liệu cột **season** sau khi xử lý

3.3.3 is_happy_customer

```
1 median_happy_customer <- round(median(merged_data$is_happy_customer, na.rm = TRUE), digits =
  0)
2 merged_data$is_happy_customer[is.na(merged_data$is_happy_customer)] <- median_happy_customer
```

Listing 7: Làm tròn trung vị

Giải thích: Lí do nhóm sử dụng giá trị trung vị cho cột **is_happy_customer** để **phản ánh độ hài lòng của khách hàng**, chúng chỉ có rơi vào một trong hai trường hợp là 0 và 1 (Hài lòng hoặc không hài lòng) và không ảnh hưởng bởi các giá trị ngoại lai .

3.3.4 Kiểm tra dữ liệu sau khi làm sạch

Sau khi dữ liệu được làm sạch, kiểm tra có cột nào còn dính giá trị bị khuyết nữa không.

```
1 > na_cout<- colSums(is.na(merged_data ))
2 > print(na_cout)
3           order_id           customer_id           date
4                0                0                0
5 nearest_warehouse shopping_cart order_price
6                10                0                0
7 delivery_charges customer_lat customer_long
8                0                10                10
9 coupon_discount order_total season
10               0                0                0
11 is_expedited_delivery distance_to_nearest_warehouse latest_customer_review
12                0                10                0
13 is_happy_customer
14                0
15 >
```

Listing 8: Kiểm tra dữ liệu bị khuyết sau khi làm sạch

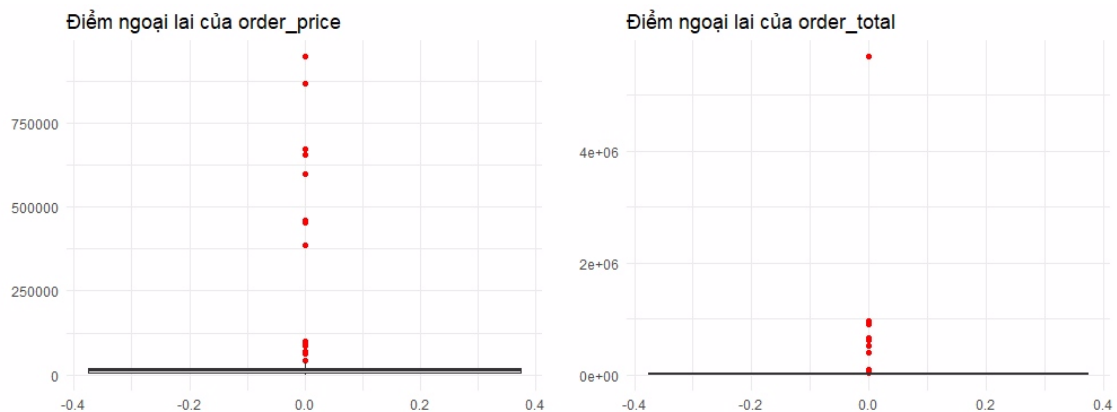
4 Mô Tả Thống Kê

4.1 Xử lý các điểm Outlier

Sau khi xử lý dữ liệu gốc, nhóm phát hiện rằng một số cột trong bộ dữ liệu xuất hiện những điểm bất thường, dẫn đến sai lệch nghiêm trọng. Điều này có thể bắt nguồn từ lỗi trong bộ dữ liệu gốc hoặc phát sinh trong quá trình xử lý. Những giá trị này được nhận diện là outliers. Để xác định các cột chứa giá trị ngoại lai, nhóm sử dụng *Borplot* để trực quan hóa dữ liệu.

```
1 ggplot(data = merged_data, aes(y = order_price)) +  
2   geom_boxplot(outlier.shape = 16, outlier.colour = "red", outlier.fill = "red") +  
3   theme_minimal() +  
4   labs(  
5     title = "Điểm ngoại lai của order_price",  
6     y = ""  
7   )  
8 ggplot(data = merged_data, aes(y = order_total)) +  
9   geom_boxplot(outlier.shape = 16, outlier.colour = "red", outlier.fill = "red") +  
10  theme_minimal() +  
11  labs(  
12    title = "Điểm ngoại lai của order_total",  
13    y = ""  
14  )
```

Listing 9: Phát hiện giá trị outlier



Hình 2: Điểm ngoại lai được đánh dấu đỏ

Qua phân tích, các cột được phát hiện có chứa outliers bao gồm:

- `order_price`
- `order_total`

Để loại bỏ điểm ngoại lai này, nhóm áp dụng phương pháp tính Độ trải giữa (*Interquartile Range*, viết tắt là IQR). Cụ thể, nhóm đặt cận trên *Whisker Upper* và cận dưới theo công thức sau:

- Cận trên

$$(Q3 + 1.5) \times IQR \text{ (độ trải giữa)}$$

- Cận dưới

$$(Q1 - 1.5) \times IQR \text{ (độ trải giữa)}$$

Những giá trị vượt qua cận trên hoặc dưới sẽ được thay thế bằng chính giá trị tương ứng của những cận này.

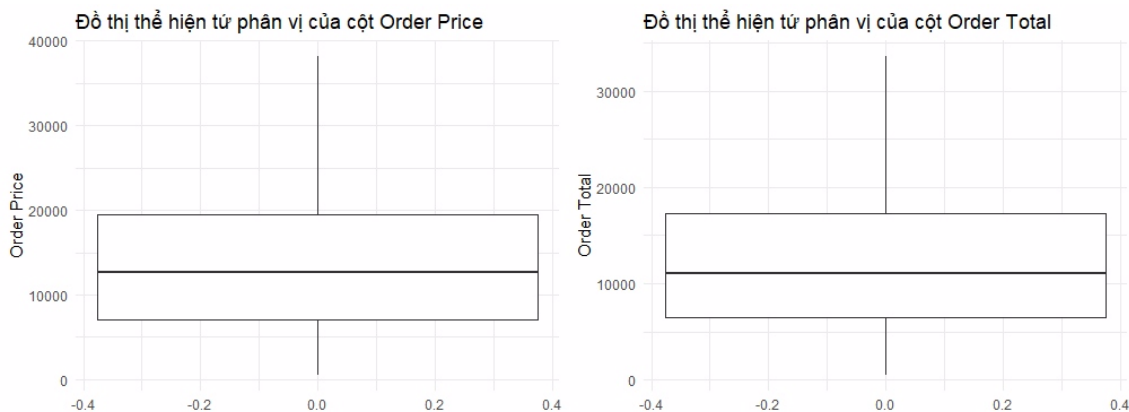
```
1 # Tính toán các tứ phân vị cho các cột order_price và order_total
2 quantiles_price <- quantile(merged_data$order_price)
3 quantiles_total <- quantile(merged_data$order_total)
4 # Trong R , hàm quantile trả về 5 giá trị cùng [index]
5 # [1]: 0%
6 # [2]: 25% [Q1]
7 # [3]: 50%
8 # [4]: 75% [Q3]
9 # [5]: 100%
10 # Hàm tứ phân vị trong R
11 q1_price <- quantiles_price[2]
12 q3_price <- quantiles_price[4]
13 q1_total <- quantiles_total[2]
14 q3_total <- quantiles_total[4]
15 # Tính IQR cho order_price và order_total
16 IQR_price <- q3_price - q1_price
17 IQR_total <- q3_total - q1_total
18
19 # Hàm tính giá trị biên dưới (lower) và biên trên (upper) của IQR
20 calc_lower <- function(Q1, IQR) { return(Q1 - 1.5 * IQR) }
21 calc_upper <- function(Q3, IQR) { return(Q3 + 1.5 * IQR) }
22
23 # Tính giá trị cận dưới và cận trên cho order_price và order_total
24 lower_price <- calc_lower(q1_price, IQR_price)
25 upper_price <- calc_upper(q3_price, IQR_price)
26 lower_total <- calc_lower(q1_total, IQR_total)
27 upper_total <- calc_upper(q3_total, IQR_total)
28 # Điều chỉnh giá trị order_total ra ngoài phạm vi IQR
29 for (i in 1:length(merged_data$order_total)) {
30   Tạo một vòng lặp cho cột order_total
31   if (merged_data$order_total[i] > upper_total) {
32     merged_data$order_total[i] = upper_total # Thay giá trị cận trên của order_total
33   } else if(merged_data$order_total[i] < lower_total){
34     merged_data$order_total[i] = lower_total # Thay giá trị cận dưới của order_total
35   }
36 }
37 # Điều chỉnh giá trị order_price ra ngoài phạm vi IQR
38 ## Tạo một vòng lặp cho cột order_price
39 for (i in 1:length(merged_data$order_price)) {
40   if (merged_data$order_price[i] > upper_price) {
41     merged_data$order_price[i] = upper_price # Thay giá trị cận trên của order_price
```

```

42 } else if(merged_data$order_price[i] < lower_price){
43   merged_data$order_price[i] = lower_price # Thay giá trị dưới của order_price
44 }
45 }

```

Listing 10: Xử lý giá trị ngoại lai bằng phương pháp IQR



Hình 3: Sau khi dùng tứ phân vị

4.2 Thống kê dữ liệu

Mục tiêu của nhóm là hiểu rõ hơn về chi phí và doanh thu trong từng mùa, từ đó đưa ra các quyết định hợp lý như thay đổi chiến lược giao hàng hoặc cải thiện hiệu quả kinh doanh trong các mùa. Hướng đến mục tiêu chung là cung cấp dịch vụ tốt nhất cho khách hàng theo từng thời điểm trong năm, nhóm đã thực hiện các phân tích thống kê về những hạng mục quan trọng bao gồm:

- Kiểm tra sự tương quan giữa `order_price` và `order_total`
- Tổng số đơn hàng theo từng mùa.
- Tổng chi phí giao hàng.

```

1 season_summary <- merged_data %>%
2   group_by(season) %>%
3   summarise(
4     total_orders = n(),
5     avg_order_total = mean(order_total, na.rm = TRUE),
6     total_delivery_charges = sum(delivery_charges, na.rm = TRUE)
7   )
8 print(season_summary)

```

Listing 11: Thống kê

```

1 > print(season_summary)
2   season total_orders avg_order_total total_delivery_charges
3   <chr>      <int>      <dbl>      <dbl>

```

4	1	autumn	247	12992.	17277.
5	2	spring	266	12320.	23526.
6	3	summer	249	12487.	19892.
7	4	winter	238	12009.	16475.

Listing 12: Thông số từng hạng mục theo mùa

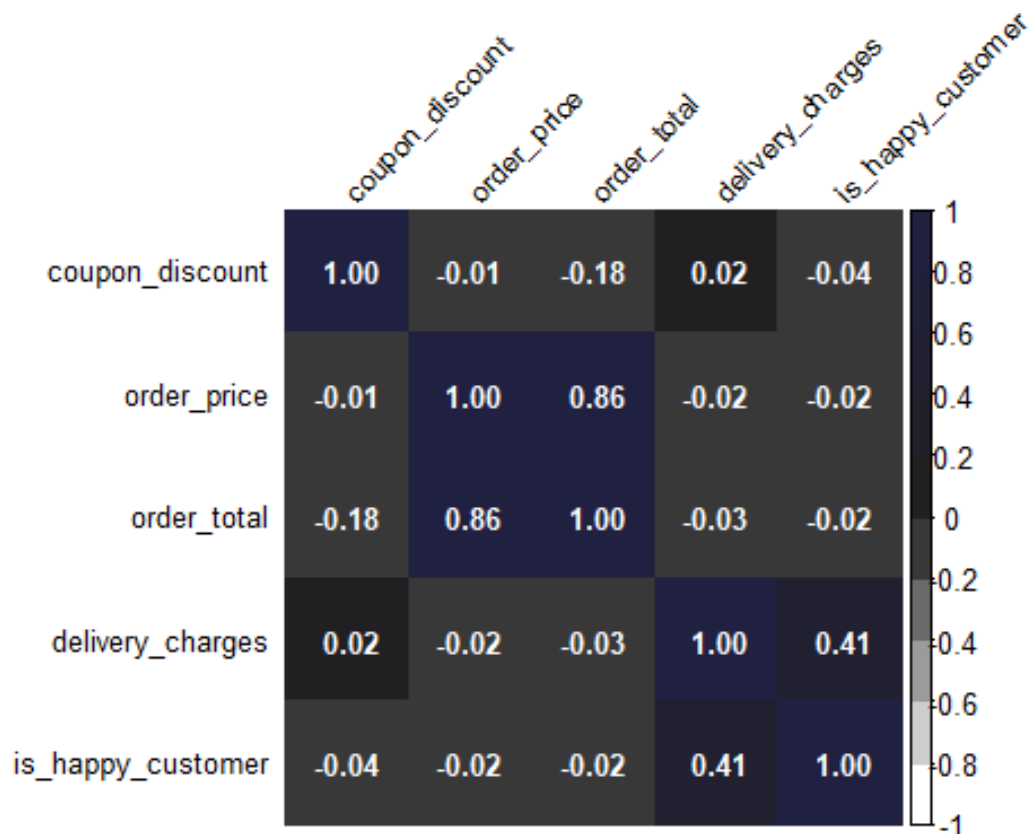
4.2.1 Ma trận tương quan giữa `order_price` và `total_order`

Trước khi vẽ đồ thị, nhóm muốn kiểm tra liệu có mối quan hệ giữa `order_total` và `order_price` hay không?

Nhóm chọn phương pháp ma trận `cor` để thể hiện độ tương quan

```
1 overview_data <- merged_data[c("order_price", "delivery_charges", "coupon_discount", "order_
  total", "is_expedited_delivery", "is_happy_customer", "shopping_cart")]
2 numeric_data <- overview_data[sapply(overview_data, is.numeric)]
3 cor_matrix <- cor(numeric_data)
4 # Hiện thị ma trận
5 corplot(
6   cor_matrix,
7   method = "color",
8   col= colorRampPalette(c("white", "#202020", "#202040"))(10) ,
9   type = "full",
10  order = "hclust",
11  tl.col = "black",
12  tl.srt = 45,
13  addCoef.col = "white",
14  number.cex = 0.8,
15  tl.cex = 0.8,
16  diag = TRUE,
17  cl.pos = "r"
18 )
```

Listing 13: Code kiểm tra độ tương quan



Hình 4: Biểu đồ colorgram

Nhận xét: Có sự tương quan giữa order_total và order_price.

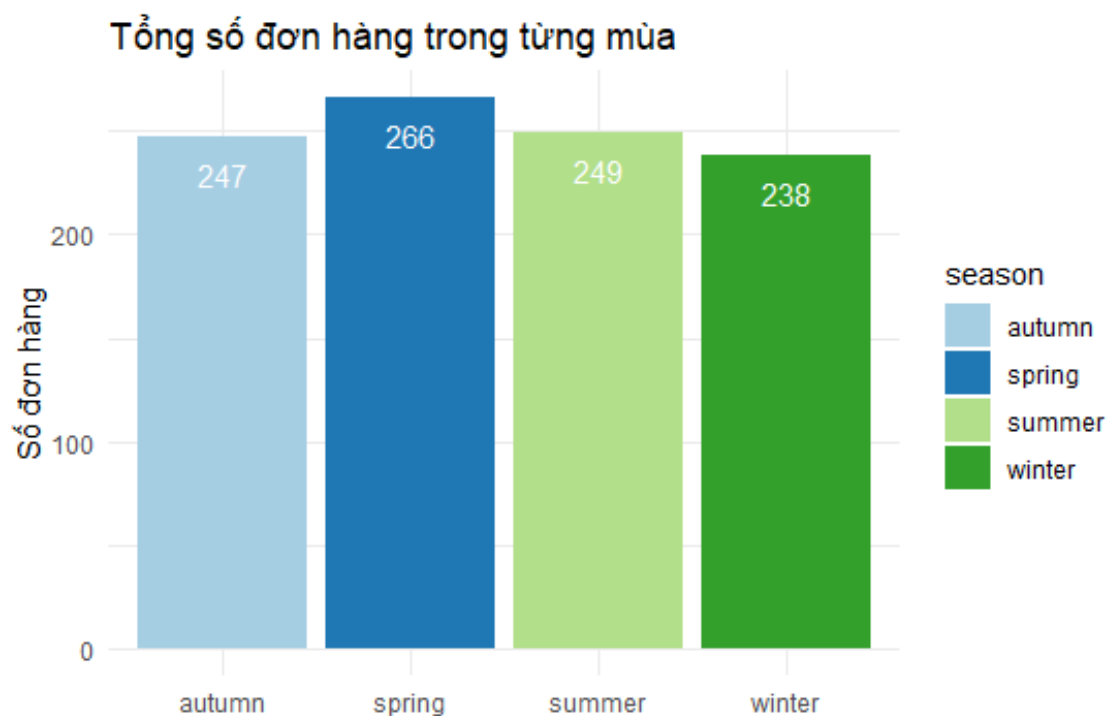
4.2.2 Tổng đơn hàng theo mùa

```
1 ggplot(data = season_summary, aes(x = season, y = total_orders, fill = season)) +
2   geom_bar(stat = "identity") +
3   geom_text(aes(label = total_orders), vjust = 2, color = "white", ) +
4   theme_minimal() +
5   labs(
6     title = "Tổng số đơn hàng trong từng mùa",
7     x = " ",
8     y = " " # Để trống
9   ) +
10  scale_fill_brewer(palette = "Paired")
```

Listing 14: Đồ thị thể hiện tổng số đơn hàng theo từng mùa

Nhận xét: Vào mùa xuân (spring), khách hàng thường mua sắm nhiều hơn nên tổng số đơn hàng tập trung nhiều vào mùa xuân. Nhưng sự chênh lệch không đáng kể.

4.2.3 Tổng phí giao hàng theo từng mùa



Hình 5: Tổng số đơn hàng theo từng mùa

```

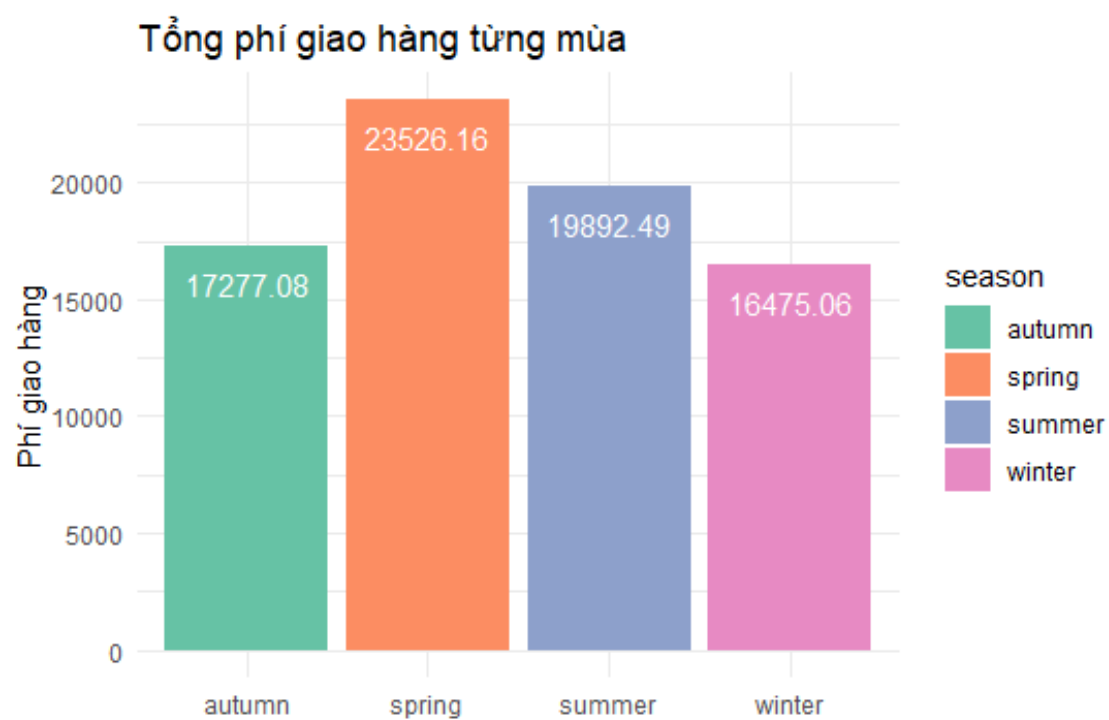
1 ggplot(data = season_summary, aes(x = season, y = total_delivery_charges, fill = season)) +
2   geom_bar(stat = "identity") +
3   geom_text(aes(label = total_delivery_charges), vjust = 2, color = "white", ) +
4   theme_minimal() +
5   labs(
6     title = "Tổng phí giao hàng từng mùa",
7     x = " ",
8     y = " " #Để trống
9   ) +
10  scale_fill_brewer(palette = "Set2")

```

Listing 15: Đồ thị Tổng chi phí giao hàng theo từng mùa

Nhận xét: Lượng chi phí đơn hàng đạt giá trị cao nhất vào mùa xuân và chúng nó độ chênh lệch cao giữa các mùa.

Tổng kết: Việc phân tích các chỉ số trên sẽ giúp nhóm có cái nhìn tổng quan và đưa ra các chiến lược phù hợp nhằm tối ưu hóa các yếu tố kinh doanh theo từng mùa, đáp ứng nhu cầu của khách hàng một cách tối đa.



Hình 6: Tổng phí đơn hàng theo từng mùa

5 Thống Kê Suy Diễn

Kiểm tra phân phối chuẩn

Trước khi lựa chọn các phương pháp thống kê, nhóm phải kiểm tra dữ liệu **season**, **price_total**, **order_total** có tuân theo bảng phân phối chuẩn hay không với phương pháp **shapiro**.

```
1 # Trích xuất các cột order_price và order_total từ merged_data
2 get_order_price <- merged_data$order_price
3 get_order_total <- merged_data$order_total
4
5 # Kiểm tra phân phối chuẩn với Shapiro-Wilk cho cột order_total
6 shapiro_order_total <- shapiro.test(get_order_total)
7 # Kiểm tra phân phối chuẩn với Shapiro-Wilk cho cột order_price
8 shapiro_order_price <- shapiro.test(get_order_price)
9
10 # In kết quả kiểm tra Shapiro-Wilk cho toàn bộ dữ liệu
11 print(shapiro_order_price) # Kết quả kiểm tra cho order_price
12 print(shapiro_order_total) # Kết quả kiểm tra cho order_total
13 # Kiểm tra phân phối chuẩn theo từng mùa
14 seasons <- c("spring", "summer", "fall", "winter")
15 # Tạo danh sách để lưu giá trị order_price theo từng mùa
16 order_price_by_season <- list()
17 # Lặp qua từng mùa, trích xuất dữ liệu và lưu vào danh sách
18 for (season in seasons) {
19   # Trích xuất dữ liệu của từng mùa bằng subset()
20   season_data <- subset(merged_data, season == season)
21   # Lưu giá trị order_price của từng mùa vào danh sách
22   order_price_by_season[[season]] <- season_data$order_price
23 }
24 # Tạo bảng tóm tắt kết quả kiểm tra Shapiro-Wilk cho từng mùa
25 shapiro_summary <- data.frame(
26   Season = names(order_price_by_season), # Tên các mùa
27   P_value = sapply(order_price_by_season, function(order_price) {
28     # Tính p-value từ kiểm tra Shapiro-Wilk cho từng mùa
29     shapiro.test(order_price)$p.value
30   })
31 )
32 # Đánh giá phân phối chuẩn: nếu p-value > 0.05 thì phân phối chuẩn
33 shapiro_summary$Normal_Distribution <- ifelse(shapiro_summary$P_value > 0.05, "True", # Phân
   phối chuẩn
34 "False") # Không phải phân phối chuẩn
35 # In bảng tóm tắt kết quả kiểm tra Shapiro-Wilk theo mùa
36 print(shapiro_summary)
```

Listing 16: Shapiro Test

Mùa	P-value	Có phải phân phối chuẩn?
spring	7.348016e-18	Không
summer	7.348016e-18	Không
fall	7.348016e-18	Không
winter	7.348016e-18	Không

Bảng 1: Kiểm tra phân phối chuẩn của mùa

```

1 > print(shapiro_order_price) # Kết quả kiểm tra cho order_price
2 Shapiro-Wilk normality test
3 data:  get_order_price
4 W = 0.95034, p-value < 2.2e-16
5 > print(shapiro_order_total) # Kết quả kiểm tra cho order_total
6 Shapiro-Wilk normality test
7 data:  get_order_total
8 W = 0.94557, p-value < 2.2e-16
9
10 % Giới thiệu

```

Listing 17: Hai cột còn lại

Nhận xét: Từ những dữ kiện trên, ta có thể kết luận chúng không tuân theo phân phối chuẩn.

5.1 Kiểm định trung bình một mẫu

5.1.1 Bài toán thực tế

Sử dụng t-test, nhóm muốn kiểm tra giả thuyết rằng doanh thu bán hàng của cửa hàng trong mùa Xuân có bằng với giá trị kì vọng ($\mu = 50$) không? Nhóm đưa ra hai giả thuyết như sau: **Giả thuyết kiểm định:**

- $H_0: \mu = 50$ (Doanh thu trung bình mùa xuân bằng kỳ vọng)
- $H_1: \mu > 50$ (Doanh thu trung bình mùa xuân lớn hơn kỳ vọng)

Giải thích: Lí do nhóm sử dụng **t-test** vì mẫu lớn hơn 30 mẫu, chúng không tuân theo phân phối chuẩn.

5.1.2 Tính giá trị kiểm định

5.1.3 Mô tả dữ liệu và công thức tính

```

1 spring_data <- merged_data %>%
2   filter(season == "spring")
3 ## Xuất ra trung bình mẫu, sd, mua

```

```
4 spring_stats <- spring_data %>%
5   summarise(
6     mean = mean(order_total),
7     sd = sd(order_total),
8     n = n()
9   )
```

Listing 18: Tính các thông số

Thông số	Giá trị
Trung bình mẫu \bar{X}	12,320
Giá trị kỳ vọng μ_0	50
Độ lệch chuẩn mẫu s	7,757
Số quan sát n	266 (vì df = 265)

Bảng 2: Các thông số thống kê dùng trong bài toán kiểm định một mẫu

Áp dụng lý thuyết t-test cho một mẫu, ta có công thức tính giá trị kiểm định như sau:

$$Z_{qs} = \frac{\bar{X} - \mu_0}{\frac{s}{\sqrt{n}}}$$

Thay số vào:

$$Z = \frac{12320 - 50}{\frac{7757}{\sqrt{266}}} = \frac{12270}{474.89} \approx 25.83$$

5.1.4 Xử lý trên phần mềm RStudio

Xử lý trên phần mềm RStudio, nhóm sử dụng hàm `z.test` để tính giá trị kiểm định và p-value. Cụ thể, nhóm sử dụng hàm này với các tham số như sau:

```
1 z.test(x = spring_data$order_total,
2       mu = 50,
3       sigma.x = sd(spring_data$order_total),
4       alternative = "greater")
```

Listing 19: Kiểm định một mẫu

5.1.5 Kết luận

```
1 One-sample z-Test
2 data:  spring_data$order_price
3 z = 28.704, p-value < 2.2e-16
4 alternative hypothesis: true mean is greater than 50
```

```
5 95 percent confidence interval:
6 12920.34      NA
7 sample estimates:
8 mean of x
9 13702.69
```

Listing 20: Kết quả sau khi dùng t-test

Kết luận: $p\text{-value} < 0.05$, như vậy ta có thể khẳng định, không đủ điều kiện để bác bỏ H_0

5.2 Kiểm định trung bình hai mẫu

5.2.1 Bài toán thực tế

Sau khi đã xác định được doanh thu của cửa hàng tại mùa xuân không vượt qua giá trị kì vọng, nhóm tiếp tục đem so sánh dữ liệu **mùa xuân** so với **mùa hè** với mức ý nghĩa $\alpha = 0.05$

Nhóm đặt giả thuyết như sau

- **Giả thuyết không (H_0):** $\mu_1 = \mu_2$ (Doanh thu trung bình mùa xuân bằng mùa hè)
- **Giả thuyết đối (H_1):** $\mu_1 \neq \mu_2$ (Có sự khác biệt giữa 2 mùa)

5.2.2 Tính giá trị kiểm định

Để tính giá trị kiểm định, nhóm sử dụng phương pháp tách 2 mẫu **xuân** và **hè** ra làm 2 và lọc ra thông số độ lệch chuẩn, n (số mẫu) và mean (trung bình mẫu)

```
1
2 \begin{lstlisting}[language=R, caption= z-test]
3 group_stats <- merged_data %>%
4   group_by(season) %>%
5   summarise(
6     mean = mean(order_total),
7     sd = sd(order_total),
8     n = n(),
9     .groups = "drop" # tránh cảnh báo nhóm
10  ) %>%
11  filter(season %in% c("spring", "summer"))
12
13 print(group_stats)
14
15 # Độ lệch chuẩn 2 mùa xuân và hè
16 sd_spring <- group_stats$sd[group_stats$season == "spring"]
17 sd_summer <- group_stats$sd[group_stats$season == "summer"]
18
19 ## Tổng giá trị đơn hàng trong 2 mùa
```

```
20 spring_data <- merged_data$order_total[merged_data$season == "spring"]
21 summer_data <- merged_data$order_total[merged_data$season == "summer"]
22 ### Z -test
23 z_test_result <- z.test(x = spring_data, y = summer_data,
24     mu = 0,
25     sigma.x = sd_spring,
26     sigma.y = sd_summer,
27     alternative = "two.sided")
28
29 print(z_test_result)
```

Listing 21: Các thông số thống kê dùng trong kiểm định trung bình 2 mẫu

5.2.3 Mô tả dữ liệu và công thức tính

Bảng 3: Chỉ số bài toán `order_total` theo mùa Xuân và Hè

Mùa	Trung bình (μ)	Độ lệch chuẩn (s)	Số lượng (n)
Xuân (Spring)	12,320	7,757	266
Hè (Summer)	12,487	7,697	249

Trong phần lý thuyết, nhóm đã đề cập phương pháp t-test cho hai mẫu. Do đó nhóm sẽ sử dụng công thức tính giá trị kiểm định và thông số trong bảng mô tả dữ liệu như sau:

$$Z = \frac{12319.55 - 12486.75}{\sqrt{\frac{7757.41^2}{266} + \frac{7697.42^2}{249}}} \approx -0.24542$$

5.2.4 Xử lý trên phần mềm RStudio

Xử lý trên phần mềm RStudio, nhóm sử dụng hàm `t.test` để tính giá trị kiểm định và p-value. Cụ thể, nhóm sử dụng hàm này với các tham số như sau:

```
1 t_test_result_two_sample<-t.test(order_total ~ season,
2     data = merged_data %>% filter(season %in% c("spring", "summer")),
3     var.equal = FALSE) # giả định phương sai không bằng nhau
4 print(t_test_result_two_sample)
```

Listing 22: Các thông số thống kê dùng trong kiểm định trung bình 2 mẫu

```
1 Welch Two Sample t-test
2
3 data: order_total by season
4 t = -0.24542, df = 511.26, p-value = 0.8062
5 alternative hypothesis: true difference in means between group spring and group summer is not
   equal to 0
6 95 percent confidence interval:
```

```
7  -1505.705  1171.298
8  sample estimates:
9  mean in group spring mean in group summer
10                12319.55                12486.75
```

Listing 23: Kết quả t-test

5.2.5 Kết luận

Với $p_{value} = 0.8602 > 0.05$, nhóm không có đủ bằng chứng để bác bỏ giả thuyết H_0 tức là không có sự khác biệt trong doanh thu giữa hai mùa.

5.3 Wilcoxon Signed rank

5.3.1 Mục tiêu

Mục tiêu của phân tích này là **kiểm tra sự khác biệt giữa giá trị gốc (order_price) và giá trị thực tế (order_total)**. Cụ thể, chúng ta muốn phân tích xem có sự khác biệt đáng kể giữa hai giá trị này hay không, để hiểu rõ hơn về mối quan hệ giữa giá trị gốc và thực tế của các đơn hàng.

5.3.2 Kết quả kiểm định

Để kiểm tra sự khác biệt giữa `order_total` và `order_price`, chúng ta thực hiện kiểm định Wilcoxon signed-rank test. Quy trình thực hiện như sau:

- **B1:** Xác định cả 2 mẫu là phân phối không chuẩn. Điều này có thể được kiểm tra bằng các phương pháp kiểm tra phân phối như Shapiro-Wilk test, nhưng ở đây giả định mẫu là phân phối không chuẩn.
- **B2:** Dùng phương pháp Wilcoxon signed-rank test, một phương pháp thống kê không tham số, để so sánh giá trị giữa `order_total` và `order_price` trong cùng một nhóm.
- **B3:** Thực hiện kiểm định Wilcoxon signed-rank test trong R:

```
1  total_price <- merged_data[, c("order_price", "order_total")] # Tạo một
   dataframe mới cho riêng order_price và order_total
2  Wilcoxon_signed_rank_sample <- total_price %>%
3    slice_sample(n = 200) # Lấy 200 mẫu bất kì từ bộ dữ liệu
4  Wilcoxon_signed_rank_result <- wilcox.test(Wilcoxon_signed_rank_sample$order_
   price, Wilcoxon_signed_rank_sample$order_total, paired = TRUE) # Kiểm định Wilcoxon_
   signed_rank
5  print(Wilcoxon_signed_rank_result) # In kết quả
6
```

Listing 24: "Wilcoxin signed-rank test"

```
1 # Kết quả
2     Wilcoxon signed rank test
3 data:  merged_data$order_total and merged_data$order_price
4 V = 12345, p-value = 0.042
5 alternative hypothesis: true location shift is not equal to 0
```

Listing 25: Kết quả

Giả thuyết

- H_0 : Không có sự khác biệt giữa hai mẫu
- H_1 : Có sự khác biệt giữa hai mẫu

5.3.3 Nhận xét

Dựa trên kết quả kiểm định Wilcoxon signed-rank test, với **p-value** < **0.05**, chúng ta có đủ cơ sở để bác bỏ giả thuyết H_0 . Điều này khẳng định rằng có sự khác biệt đáng kể giữa giá trị của *order_price* và *order_total* sau khi kết hợp với các yếu tố **delivery_charges** và **coupon_discount**.

Chúng ta có thể nhìn ra sự khác biệt thông qua các giá trị quan trắc như sau

```
1 difference <- Wilcoxon_signed_rank_sample$order_price - Wilcoxon_signed_rank_sample$order_
  total # Tính phép toán trừ giữa order_price và order_total và lưu kết quả vào một đối tượng
2 summary(difference) # Gọi hàm summary ra các giá trị quan trắc
```

Listing 26: Gọi các giá trị quan trắc

```
1 > summary(difference) # Gọi hàm summary ra các giá trị quan trắc
2      Min.   1st Qu.   Median     Mean  3rd Qu.    Max.
3 -31109.3   263.9     795.3   1334.6   2302.1   33520.3
4 >
```

Listing 27: Quan sát các giá trị quan trắc

5.3.4 Kết luận

Với kết quả kiểm định thống kê, ta kết luận rằng **có sự khác biệt giữa giá trị gốc và giá trị thực tế của các đơn hàng**. Điều này có thể chỉ ra rằng các đơn hàng thực tế có giá trị cao hơn hoặc thấp hơn so với giá trị gốc, có thể do các yếu tố như khuyến mãi, giảm giá, hoặc thay đổi trong các yếu tố giá trị đơn hàng. Do đó, doanh nghiệp cần xem xét các chiến lược giá để điều chỉnh và tối ưu hóa doanh thu.

6 Thảo luận và Mở rộng

6.1 Nhận xét dữ liệu

- Trong mô hình tổng quan, có những cột dữ liệu không tuân theo phân phối chuẩn như **order_total** và **order_price**. Điều này có thể do sự hiện diện của các giá trị ngoại lai hoặc phân phối không đồng nhất.
- Phương sai không đồng nhất, có thể do sự biến động lớn trong giá trị đơn hàng giữa các mùa.
- Có những cột phụ thuộc với nhau, ví dụ **order_price** ảnh hưởng bởi **coupon_discount**.

6.2 Tính bao quát

Nhóm đã phân tích các yếu tố ảnh hưởng đến sự thay đổi của dữ liệu, cũng như phân tích và kiểm định trong từng trường hợp như sau: Các kiểm định được chọn phù hợp với mục tiêu và đặc điểm dữ liệu (không chuẩn, không đồng nhất phương sai, có cặp). Ngoài ra, nhóm cũng mở rộng bằng kiểm định hậu nghiệm (Dunn Test) để phân tích sâu hơn sau Kruskal-Wallis. Do đó, phân tích được đánh giá là có tính bao quát tốt và hợp lý về mặt phương pháp. Cụ thể các phương pháp đã được nhóm sử dụng như sau:

- Phân tích 1 mẫu với **z-test**
- Phân tích 2 mẫu với **t-test**
- Phân tích Anova bằng Kruskal-Wallis
- Phân tích hồi quy tuyến tính

6.3 Điểm mạnh và điểm yếu của các phương pháp đã sử dụng

Bảng 4: Tổng hợp ưu điểm và nhược điểm của các phương pháp kiểm định

Phương pháp	Ưu điểm	Nhược điểm
Z-test một mẫu	<ul style="list-style-type: none"> Dễ hiểu và triển khai. Hiệu quả với cỡ mẫu lớn ($n > 30$). Không yêu cầu phân phối chuẩn nhờ định lý giới hạn trung tâm. 	<ul style="list-style-type: none"> Giả định dữ liệu độc lập, mẫu đại diện. Không phù hợp với cỡ mẫu nhỏ.
T-test hai mẫu (Welch)	<ul style="list-style-type: none"> So sánh hai nhóm độc lập. Không yêu cầu phương sai bằng nhau. 	<ul style="list-style-type: none"> Nhạy cảm với ngoại lai nếu dữ liệu không chuẩn. Cần số lượng mẫu đủ lớn nếu phân phối không chuẩn.
Kruskal-Wallis trong Anova	<ul style="list-style-type: none"> Thay thế ANOVA khi dữ liệu không chuẩn. So sánh ≥ 3 nhóm độc lập. Không yêu cầu phương sai đồng nhất. 	<ul style="list-style-type: none"> Không chỉ ra cặp nhóm nào khác biệt. Cần kiểm định hậu nghiệm (như Dunn Test).

6.4 Mối quan hệ giữa mục tiêu và phương pháp được chọn

- Phân tích hồi quy thường dùng để xây dựng mô hình dự đoán hoặc mô hình hóa mối quan hệ tuyến tính giữa biến phụ thuộc và biến độc lập.
- Trong khi đó, bài toán của nhóm tập trung vào việc kiểm tra:
 - Sự khác biệt giữa các nhóm (mùa):** Dùng Kruskal-Wallis và Wilcoxon-Mann-Whitney.
 - Sự khác biệt giữa các giá trị trong cùng một nhóm:** Dùng Wilcoxon Signed Rank.
- Việc lựa chọn kiểm định không tham số phản ánh mục tiêu phân tích cụ thể mà không phụ thuộc vào các giả định phức tạp về phân phối dữ liệu.

Kết luận: Phân tích hồi quy không được chọn vì:

- Dữ liệu vi phạm các giả định quan trọng (phân phối chuẩn, phương sai đồng nhất).
- Mục tiêu của nhóm không phải dự đoán giá trị mà là kiểm tra sự khác biệt thống kê.

Các kiểm định không tham số được chọn phù hợp hơn, chính xác hơn, và cung cấp kết quả giải thích trực tiếp liên quan đến câu hỏi bài toán.

6.5 Mở rộng: Dunn Test

Sau khi đưa ra những điểm mạnh hay giới hạn những phương pháp nhóm đã sử dụng ở trên. Nhóm quyết định sẽ mở rộng phương pháp **Kruskal-Wallis** Như đề cập ở trên, phương pháp kiểm định KW chỉ đưa ra được "bề nổi" hay chỉ kiểm tra sự khác biệt tổng thể giữa các nhóm.

Nhóm sẽ áp dụng phương pháp hậu nghiệm (pos-hoc) để tìm ra cụ thể nhóm nào có sự khác biệt. Nhóm đề xuất sử dụng **Dunn Test**.

```
1 dunn_result <- dunn.test(merged_data$order_price, merged_data$season, method = "bonferroni",
  list = TRUE)
2 # Tạo bảng kết quả
3 table <- cbind.data.frame(
4   Comparison = dunn_result$comparisons, # Các cặp so sánh (cụ thể mùa)
5   Z_value = dunn_result$Z, # Giá trị Z
6   P_adjusted = dunn_result$P.adjusted # P-value đã điều chỉnh
7 )
8 # Sắp xếp bảng theo p-value đã điều chỉnh
9 table <- table[order(table$P_adjusted), ]
10 # Tạo bảng với gt và thêm tiêu đề
11 table %>%
12   gt() %>%
13   tab_header(
14     title = md("#### Kết quả phân tích pos-hoc (Dunn's Test)"),
15     subtitle = "So sánh sự khác nhau giữa các nhóm mùa bằng Bonerroni"
16   )
```

Listing 28: Thực hiện phân tích Pos Hoc trong R

6.6 Kết quả

Các mùa	Z-value	P.adjusted
autumn - winter	1.4580097	0.4345136
autumn - spring	0.8823710	1.0000000
autumn - summer	0.6008488	1.0000000
spring - summer	-0.2722941	1.0000000
spring - winter	0.6104089	1.0000000
summer - winter	0.8656661	1.0000000

Bảng 5: Post-hoc Analysis Results (Dunn's Test)

Kết luận: Sau khi xuất ra kết quả Dunn Test, tổng quan các mùa đều có khác biệt gì lớn.

7 Nguồn Dữ Liệu

- Dữ liệu mẫu: <https://www.kaggle.com/datasets/muhammadshahrayar/transactional-retail-dataset-of-electronics-store>
- Link R của bài tập lớn: https://github.com/bnhtho/btl_xtsk_241/blob/main/2333017_Assignment.R

8 Tài liệu Tham Khảo

Tài liệu

- [1] Anderson, D. R., Sweeney, D. J., & Williams, T. A. (2016). *Statistics for Business and Economics* (11th ed.). Cengage Learning Vietnam Company Limited.
- [2] Peter Dalgaard. *Introductory Statistics with R* (8th ed)
- [3] Nguyễn Đình Huy. *Giáo trình xác suất và thống kê(2023)*, Nxb. Đại học Quốc Gia, Thành phố Hồ Chí Minh.