ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

BÁO CÁO BÀI TẬP LỚN

MÔN HỌC : XÁC SUẤT THỐNG KÊ HỌC KỲ 231

Đề tài : 10 - Lớp L11

Giảng viên hướng dẫn: Nguyễn Thị Mộng Ngọc

-

Thành phố Hồ Chí Minh – 2023

Bảng phân công công việc

Sinh viên thực hiện	Mã số sinh viên	Phân công công việc	Tiến độ công việc
Bùi Trần Gia Hưng	2211353	Lý thuyết + tổng hợp	100%
Dương Gia Thuận	2213350	Thống kê suy diễn + tiền xử lý dữ liệu + Code R	100%
Bùi Việt Tiến	2213445	Lý thuyết + Powerpoint	100%
Trần Đình Thưởng	2213424	Code R + thống kê mô tả	100%
Nguyễn Trường Sơn	2212930	Lý thuyết + Powerpoint	100%

MỤC LỤC

I. Tổng quan dữ liệu ..... **Error! Bookmark not defined.**

II. Kiến thức nền ..... **Error! Bookmark not defined.**

1. Phân tích hồi quy ..... **Error! Bookmark not defined.**

2. Mô hình hồi quy logistic ..... **Error! Bookmark not defined.**

3. Phân tích phương sai một yếu tố (Anova) ..... **Error! Bookmark not defined.**

III. Tiền xử lý dữ liệu ..... **Error! Bookmark not defined.**

IV. Thống kê mô tả ..... **Error! Bookmark not defined.**

V. Thống kê suy diễn ..... **Error! Bookmark not defined.**

VI. Thảo luận và mở rộng ..... **Error! Bookmark not defined.**

VII. TÀI LIỆU THAM KHẢO ..... **Error! Bookmark not defined.**

I. TỔNG QUAN DỮ LIỆU

Hoạt động 1:

Tập tin "dirty_data"chứa thông tin về một cửa hàng điện tử trực tuyến. Cửa hàng có ba kho để giao hàng cho khách hàng. Dữ liệu gốc được cung cấp tại: https://www.kaggle.com/datasets/muhammadshahrayar/transactional-retail-dataset-of-electronics-store?select=dirty_data.csv

Các biến chính trong bộ dữ liệu:

- **order_price:** giá của đơn hàng trước khi được giảm giá (USD)
- **delivery_charges:** phí giao hàng
- **coupon_discount**: giảm giá (%)
- **order_total:** thành tiền (USD)
- **season:** mùa
- **is_expedited_delivery:** biến biểu thị khách hàng có yêu cầu giao hàng nhanh hay không (1/0)
- **distance_to_nearest_warehouse**: khoảng cách từ khách hàng đến kho gần nhất (km)
- **is_happy_customer:** biến biểu thị khách hàng có hàng lòng hay không (1/0)

Các bước thực hiện:

1. Đọc dữ liệu (Import data):

2. Làm sạch dữ liệu (Data cleaning): NA (dữ liệu khuyết)

3. Làm rõ dữ liệu: (Data visualization)

(a) Chuyển đổi biến (nếu cần thiết)

(b) Thống kê mô tả: dùng thống kê mẫu và dùng đồ thị

4. Xây dựng mô hình hồi quy logistic để đánh giá các nhân tố có thể ảnh hưởng đến việc khách hàng có hài lòng với đơn hàng đó hay không

5. Thực hiện dự báo khách có hài lòng với đơn hàng hay không

II. KIẾN THỨC NỀN

1.Phân tích hồi quy:

1.1 Định nghĩa:

Phân tích hồi qui là phân tích mối liên hệ phụ thuộc giữa một biến gọi là biến phụ thuộc (biến được giải thích, biến nội sinh) phụ thuộc vào một hoặc một số biến khác gọi là (các) biến giải thích (biến độc lập, biến ngoại sinh, biến hồi qui).

Phân tích hồi quy được sử dụng trong một số bối cảnh trong kinh doanh, tài chính và kinh tế. Ví dụ, nó được sử dụng để giúp các nhà quản lý đầu tư định giá tài sản và hiểu mối quan hệ giữa các yếu tố như giá cả hàng hóa và cổ phiếu của các doanh nghiệp kinh doanh những mặt hàng đó.

1.2 Một số khái niệm cơ bản:

• Biến phụ thuộc Y (dependent variable):

Biến số chịu ảnh hưởng của một biến số khác trong mô hình. Ví dụ, nhu cầu về một hàng hoá bị ảnh hưởng bởi giá cả của nó.

• Biến giải thích/ độc lập X (regressor(s))

Biếnsố tác động tới biến số khác (biến phụ thuộc) trong một mô hình kinh tế. Chẳng hạn, giá hàng hoá là biến số độc lập ảnh hưởng tới lượng cầu về nó.

• Tham số hồi quy

Tham số hồi quy (tổng thể), , là những con số cố định (fixed numbers) và không ngẫu nhiên (not random), mặc dù mình không thể biết giá trị thực của các Bs là bao nhiêu.

Phân tích hồi qui nghiên cứu mối liên hệ phụ thuộc giữa đại lượng ngẫu nhiên biến phụ thuộc phụ thuộc vào các giá trị xác định của (các) biến giải thích như thế nào.

1.3 Mục đích hồi quy:

- Ước lượng (Estimate) trung bình biến phụ thuộc và các tham số.
- Kiểm định (Hypothesis testing) về mối quan hệ.
- Dự báo (Forecast, Prediction) giá trị biến phụ thuộc khi biến giải thích thay đổi.

2. Mô hình hồi quy logistic:

1.1 Lý thuyết:

Hồi quy Binary Logistic là mô hình phổ biến trong nghiên cứu dùng để ước lượng xác suất một sự kiện sẽ xảy ra. Đặc trưng của hồi quy nhị phân là biến phụ thuộc chỉ có hai giá trị: 0 và 1. Trên thực tế, có rất nhiều hiện tượng tự nhiên, hiện tượng kinh tế, xã hội,… mà chúng ta cần dự đoán khả năng xảy ra của nó như chiến dịch quảng cáo có được chấp nhận hay không, người vay có trả được nợ hay hông, công ty có phá sản hay không, khách hàng có mua hay không,… Những biến nghiên cứu có hai biểu hiện như vậy được mã hóa thành hai giá trị 0 và 1, được gọi là biến nhị phân.

Khi biến phụ thuộc ở dạng nhị phân, chúng ta không thể phân tích với dạng hồi quy tuyến tính thông thường vì mô hình sẽ vi phạm các giả định hồi quy … Các giả định quan trọng này bị vi phạm sẽ làm mất hiệu lực thống kê của các kiểm định trong hồi quy, dẫn đến kết quả ước lượng không còn chính xác. Trong khi đó, hồi quy Binary Logistic lại không cần thiết phải thỏa mãn các giả định này

1.2 Phương trình hồi quy binary logistic:

Thay vì chúng ta ước lượng giá trị của biến phụ thuộc Y theo biến độc lập X như ở hồi quy đa biến, thì trong hồi quy Binary Logistic, chúng ta sẽ ước lượng xác suất xảy ra sự kiện Y (probability) khi biết giá trị X. Biến phụ thuộc Y có hai giá trị 0 và 1, với 0 là không xảy ra sự kiện và 1 là xảy ra sự kiện. Từ đặc điểm này,chúng ta có thể đánh giá được khả năng xảy ra sự kiện (Y = 1) nếu xác suất dự

đoán lớn hơn 0.5, ngược lại, khả năng không xảy ra sự kiện (Y = 0) nếu xác suất dự đoán nhỏ hơn 0.5. Ta có hàm xác suất như sau:

$$P_i = P(Y = 1) = E(Y = 1|X) = \frac{e^{-(\beta_0+\beta_1 X_{1i}+\beta_2 X_{2i}+\beta_3 X_{3i}+\ldots+\beta_k X_{ki})}}{1+e^{-(\beta_0+\beta_1 X_{1i}+\beta_2 X_{2i}+\beta_3 X_{3i}+\ldots+\beta_k X_{ki}+)}}$$

Trong đó P_i là xác suất xảy ra sự kiện. Thực hiện các phép chuyển đổi toán học, ta thu được hàm hồi quy mới như sau:

$$\log\left(\frac{P_i}{1+P_i}\right) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \ldots + \beta_k X_{ki}$$

Trong đó

P_i xác suất xảy ra sự kiện (Y=1

$1 - P_i$ xác suất xảy ra sự kiện (Y=0)

Lí do hàm hồi quy logistic có dạng như trên là vì giá trị xác suất bị ràng buộc trong phạm vi từ 0 đến 1. Trong khi đó, hàm hồi quy tuyến tính có giá trị từ âm vô cùng đến dương vô cùng. Và đối với các giá trị của hàm nằm ngoài khoảng (0,1) thì sự diễn giải theo mô hình hồi quy logistic không mang ý nghĩa thống kê.

Để giải quyết vấn đề này, có hai bước tiếp cận thông qua chúng ta thực hiện hai biến đổi:

- Thứ nhất, chúng ta sử dụng một chỉ số thống kê quan trọng đó là Odds:

$$Odd = \frac{P}{1-P}$$

Trong đó, Odd được định nghĩa là tỉ số của hai xác suất. Nếu p là xác suất mắc bệnh, thì 1-p là xác suất sự kiện không mắc bệnh. Như vậy, giá trị của Odd có thể lớn hơn 1 và khi xác suất P tiến dần tới 1 thì đồng thời Odd cũng tiến ra vô cùng. Giá trị của Odd bây giờ nằm trong khoảng từ 0 đến vô cùng, vì vậy mô hình hồi quy vẫn còn hạn chế vì Odd bị giới hạn bởi 0.

Để khắc phục điều này, ta thực hiện phép biến đổi tiếp theo, trình bày công thức của Odd ở trên theo logarit của Odd:

$$\text{Log(Odd)} = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \ldots + \beta_k X_{ki}$$

Từ đây việc hoàn chuyển từ P sang Odd và từ Odd sang log(Odd) làm cho vế phải của phương trình có dạng của hàm tuyến tính đa biến và giá trị của log(Odd) không bị giới hạn bởi cận trên, cận dưới nên việc phân tích các tham số dựa vào tích chất của hàm hồi quy đa biến sẽ dễ dàng hơn.

2.3 Phương pháp ước lượng hợp lí cực đại (Maximum likelihood)

Bởi vì hồi quy logistic hoạt động về một biến phân loại, phương pháp của bình phương nhỏ nhất (OLS) là không thể sử dụng (nó giả định một biến phụ thuộc được phân phối chuẩn). Do vậy, một phương pháp ước lượng chung hơn được sử dụng để phát hiện giá trị phù hợp tốt của các tham số. Điều này được gọi là "ước lượng hợp lí cực đại" (Maximum likelihood estimation).

Hợp lí cực đại là một kĩ thuật ước lượng tương tác để chọn các ước lượng tham số rằng cực đại sự hợp lí của bộ dữ liệu mẫu được quan sát. Trong hồi quy logistic, hợp lí cực đại chọn các ước lượng hệ số rằng sự cực đại về logarit của xác suất của quan sát bộ giá trị cụ thể của biến phụ thuộc trong mẫu cho một bộ đã cho của các giá trị X.

Bởi vì hồi quy logistic sử dụng phương pháp hợp lí cực đại, hệ số xác định R^2 có thể không được ước lượng trực tiếp.

3. Phân tích phương sai một yếu tố (Anova)

• Lý thuyết phân tích phương sai một yếu tố

Phân tích phương sai một yếu tố (còn gọi là oneway anova) dùng để kiểm định giả thuyết trung bình bằng nhau của các nhóm mẫu với khả năng phạm sai lầm chỉ là 5%. Phân tích phương sai một yếu tố là phân tích ảnh hưởng của một yếu tố nguyên nhân (định tính) đến một yếu tố kết quả (định lượng).

• Phương pháp phân tích phương sai một yếu tố

Phép phân tích phương sai được dùng trong các trắc nghiệm để so sánh các giá trị trung bình của hai hay nhiều mẫu được lấy từ các phân số. Đây có thể được xem như phần mở rộng các trắc nghiệm t hay z (so sánh hai giá trị trung bình).

• Mục đích của sự phân tích phương sai 1 yếu tố:

Đánh giá sự ảnh hưởng của một yếu tố (nhân tạo hay tự nhiên) nào đó trên các giá trị quan sát, Y_i(i=0,1,2,…k).

• Mô hình phân tích phương sai 1 yếu tố:

Thứ Tự	k mẫu quan sát				
	1	2	3	...	k
1	X_{11}	X_{12}	X_{13}		X_{1k}
2	X_{21}	X_{22}	X_{23}		X_{2k}
...	...	...	...		...
...					
N	X_{N1}	X_{N2}	X_{N3}		X_{Nk}
Trung bình mẫu	T_1	T_2	T_3		T_K
	$\overline{X_1}$	$\overline{X_2}$	$\overline{X_3}$		$\overline{X_k}$

• **Bảng Anova**

Nguồn sai số	Bậc sai số	Tổng số bình phương	Bình phương trung bình
Yếu tố	$k-1$	$SSF = \sum_{i=1}^{k} \frac{T_i^2}{N} - \frac{T^2}{N}$	$MSF = \frac{SSF}{k-1}$
Sai số	$N-k$	$SSE = SST - SSF$	$MSE = \frac{SSE}{N-k}$
Tổng cộng	$N-1$	$SST = \sum_{i=1}^{k} \sum_{j=1}^{n} Y_n^2 - \frac{T^2}{N}$	

Một số giả định khi phân tích ANOVA:
– Các nhóm so sánh phải độc lập và được chọn một cách ngẫu nhiên.
– Các nhóm so sánh phải có phân phối chuẩn hoặc cỡ mẫu phải đủ lớn để được xem như tiệm cận phân phối chuẩn.
– Phương sai của các nhóm so sánh phải đồng nhất.
Lưu ý: nếu giả định tổng thể có phân phối chuẩn với phương sai bằng nhau không đáp ứng được thì bạn có thể dùng kiểm định phi tham số Kruskal-Wallis sẽ để thay thế cho ANOVA.

III. TIỀN XỬ LÝ SỐ LIỆU

NHIỆM VỤ 1: Xử lí dữ liệu file dirty_data

1.1 Đọc dữ liệu

```
dirty_data <- read.csv("C:\\Users\\Admin\\Downloads\\dirty_data.csv")
view(dirty_data)
```

Dưới đây là bảng dữ liệu được đọc:

order_id	customer_id	date	nearest_warehouse	shopping_cart	order_price	delivery_charges	customer_lat	customer_long	coupon_discount	order_total	season	is_expec
ORD182494	ID6197211592	2019-06-22	Thompson	[('Lucent 330S', 1), ('Thunder line', 2), ('iStream', 2),...	12200	79.89	-37.81511	144.9328	10	11059.89	Winter	True
ORD395518	ID0282825849	2019-12-29	Thompson	[('Thunder line', 1), ('Universe Note', 2)]	9080	62.71	-37.80274	144.9511	0	9142.71	Summer	False
ORD494479	ID0579391891	2019-03-02	Nickolson	[('Thunder line', 1), ('pearTV', 2)]	10670	65.87	-37.82130	144.9576	10	9668.87	Autumn	False
ORD019224	ID4544561904	2019-01-12	Nickolson	[('Universe Note', 1), ('Alcon 10', 2), ('Olivia x460', 1...	24800	57.61	-37.81142	144.9731	15	21137.61	Summer	False
ORD104032	ID6231506320	2019-11-28	Nickolson	[('Universe Note', 1), ('Olivia x460', 1), ('iStream', 1)...	9145	75.54	37.82386	144.9699	25	6934.29	Spring	False
ORD146760	ID0311654900	2019-09-16	Bakers	[('Thunder line', 2), ('Universe Note', 1)]	7810	71.22	37.82025	145.0149	10	7100.22	Spring	False
ORD337984	ID3394768956	2019-09-14	Thompson	[('Candle Inferno', 1), ('Alcon 10', 1), ('Toshika 750', ...	13700	74.84	-37.80774	144.9516	5	13089.84	Spring	False
ORD072312	ID0774517121	2019-05-23	Thompson	[('Universe Note', 1), ('Thunder line', 2), ('iStream', 1)]	7960	52.28	-37.80634	144.9595	5	10789.79	Autumn	False
ORD377837	ID4769265355	2019-10-09	Bakers	[('Alcon 10', 2), ('Thunder line', 1), ('Candle Inferno',...	25390	107.58	-37.81081	145.0141	10	22958.58	Spring	True
ORD462194	ID5301568579	2019-03-21	Thompson	[('Universe Note', 1), ('Lucent 330S', 1), ('Toshika 75...	13320	62.26	-37.80868	144.9423	15	11384.26	winter	True
ORD034800	ID4283908179	2019-08-03	Bakers	[('Alcon 10', 2), ('pearTV', 2), ('iStream', 1), ('Olivia ...	31895	78.25	-37.81133	145.0087	0	31973.25	Winter	True
ORD361636	ID0589500304	2019-12-05	Nickolson	[('Lucent 330S', 1), ('pearTV', 2)]	13850	77.29	-37.82023	144.9574	25	10464.79	Summer	False
ORD124395	ID0702352304	2019-02-11	Thompson	[('Alcon 10', 1), ('Universe Note', 1), ('pearTV', 1), ('i...	19010	94.75	-37.80543	144.9413	0	926057.25	Summer	True
ORD255642	ID3085953531	2019-12-24	Nickolson	[('iAssist Line', 2), ('Alcon 10', 1), ('pearTV', 1)]	19710	75.64	-37.81617	144.9753	0	19785.64	Summer	True
ORD496722	ID0589449820	2019-04-09	Nickolson	[('pearTV', 2), ('iStream', 1), ('Lucent 330S', 1), ('Alc...	31900	79.78	-37.80946	144.9724	0	31979.78	Autumn	True
ORD449130	ID0356449717	2019-05-17	Bakers	[('Toshika 750', 2), ('Alcon 10', 1), ('Thunder line', 1)...	20200	46.35	-37.80813	144.9868	25	15196.35	Autumn	False
ORD036056	ID0767733196	2019-08-10	Nickolson	[('Alcon 10', 1), ('Olivia x460', 2), ('Lucent 330S', 1),...	14810	80.69	-37.80423	144.9717	25	11188.19	Winter	True
ORD428910	ID2180614753	2019-07-15	Nickolson	[('Olivia x460', 1), ('Candle Inferno', 1)]	1655	63.27	-37.81411	144.9714	25	1304.52	Winter	False
ORD007249	ID0767586831	2019-02-24	Nickolson	[('iAssist Line', 2), ('Candle Inferno', 2), ('Olivia x460...	6535	60.64	-37.80918	144.9729	5	6268.89	Summer	False
ORD232940	ID2020835025	2019-01-04	Nickolson	[('iAssist Line', 1), ('Toshika 750', 1)]	6545	77.11	-37.81492	144.9680	5	6294.86	Summer	True
ORD178590	ID6167441029	2019-11-20	Thompson	[('Candle Inferno', 1), ('Toshika 750', 2), ('Thunder li...	13710	77.61	-37.80931	145.0186	0	13787.61	Spring	False
ORD157688	ID1404219802	2019-12-15	Bakers	[('Candle Inferno', 1), ('Toshika 750', 1)]	4750	73.05	-37.80282	144.9924	0	4823.05	winter	False
ORD314979	ID2187972536	2019-09-17	Nickolson	[('Lucent 330S', 1), ('Alcon 10', 2), ('iStream', 1), ('U...	22730	63.51	-37.80960	144.9638	25	17111.01	Spring	False
ORD339672	ID0129952519	2019-10-07	Bakers	[('Olivia x460', 1), ('iAssist Line', 2)]	5675	73.03	-37.80698	144.9942	5	5464.28	Spring	False
ORD239838	ID1888086809	2019-04-07	Thompson	[('Universe Note', 2), ('Toshika 750', 1), ('pearTV', 1)...	18390	68.69	-37.79960	144.9605	5	17539.19	Autumn	False
ORD340641	ID3130999251	2019-11-19	Bakers	[('Candle Inferno', 1), ('Thunder line', 1), ('iStream', ...	2910	91.48	-37.81329	145.0104	25	2273.98	Spring	True

Hình 3.1 *Đọc dữ liệu file dirty_data.csv*

1.2 Làm sạch dữ liệu

Tạo một tệp con bao gồm các biến cần phân tích(bỏ đi 1 số biến không cần thiết) như là order_id, customer_id,…:

```
new_data_0<-
dirty_data[,c("nearest_warehouse","order_price","delivery_charges","customer_lat","customer_long","coupon_discount","order_total","season","is_expedited_delivery","distance_to_nearest_warehouse","is_happy_customer")]
```

1.3 Xử lí những cột biến phân loại có cách viết chưa đồng nhất

•Kiểm tra dữ liệu khuyết trong tập tin mới:

```
> apply(is.na(new_data_0),2,which)
integer(0)
```

Nhận xét: Tệp tin không có dữ liệu khuyết

Lần lượt làm sạch các dữ liệu trong những cột có kiểu dữ liệu character: nearest_warehouse, season

Đầu tiên, kiểm tra tính đồng nhất của dữ liệu trong cột nearest_warehouse, sử dụng hàm unique():

```
> table(new_data_0$nearest_warehouse, useNA = "always")

 Bakers nickolson Nickolson  thompson  Thompson      <NA>
    119         3       181         4       193         0
```

Nhận xét: không có dữ liệu khuyết nhưng có một số cách viết khác nhau trong cột nearest_warehouse

Khắc phục: sử dụng lệnh recode để tái đinh nghĩa lại cột "nearest_warehouse" theo cột hiện tại "nearest_warehouse", nhưng với các thay đổi mã hoá được chỉ định (thompson ->Thompson, nickolson -> Nickolson)

•Kiểm tra cột nearest_warehouse:

```
new_data_0<- new_data_0 %>%
mutate(nearest_warehouse = recode(nearest_warehouse,
                                  "Thompson"   = "Thompson",
                                  "thompson"   = "Thompson",
                                  "Nickolson"  = "Nickolson",
                                  "nickolson"  = "Nickolson",
                                  "Bakers"     = "Bakers"        ))
```

• Kiểm tra lại:

```
> table(new_data_0$nearest_warehouse, useNA = "always")

 Bakers Nickolson  Thompson      <NA>
    119       184       197         0
```

Hình 3.2 *Đồng nhất cách viết trong cột nearest_warehouse*

•Thực hiện tương tự với cột season.

•Kiểm tra lại:

```
> table(new_data_0$season, useNA = "always")

Autumn Spring Summer Winter   <NA>
   127    134    124    115      0
```

Hình 3.3 *Đồng nhất cách viết trong cột season*

NHIỆM VỤ 2: tìm và xử lí dữ liệu khuyết trong file dữ liệu missing_data

2.1 Đọc dữ liệu

Tương tự như phần trên, đọc dữ liệu file missing_data vào trong R

```
missing_data <- read.csv("C:\\Users\\Admin\\Downloads\\missing_data.csv")
```

2.2 Làm sạch dữ liệu

2.2.1 Xử lí cột season:

Xây dựng hàm thời gian để suy ra các ô mùa (season) còn trống: nhận thấy thời gian của cột date với season chưa tương ứng nhau, vì vậy cần xử lí đồng thời ô dữ liệu trống và ô mùa chưa phù hợp với date

Code sử dụng

```
      as.Date(missing_data$date)
missing_data <- missing_data %>%   mutate(date_onset = as.Date(date, format = "%Y/%m/%d"))
fix_season <- function(date) {
 month <- as.POSIXlt(date)$mon +1
 if (month %in% c(1, 2, 3)) {
  return("Spring")
 } else if (month %in% c(4, 5, 6,7)) {
  return("Summer")
 } else if (month %in% c(7, 8, 9,10)) {
  return("Autumn")
 } else {
  return("Winter")
 }
}
missing_data$season <-sapply(missing_data$date, fix_season)
```

Nhận xét: sau khi xử lí các ô mùxa bị thiếu đã được điền đầy đủ và những ô mùa chưa đúng cũng được sửa.

2.2.2 Xử lí cột nearest_warehouse

Dựa vào kinh độ, vĩ độ từ file warehouses để tìm kho gần nhất:

```
warehouses <- read.csv("C:\\Users\\Admin\\Downloads\\warehouses.csv")
```

Xây dựng hàm tìm địa chỉ kho gần nhất, sử dụng hàm distVincentySphere trong package geosphere để tính khoảng cách giữa khách hàng và các kho và trả về tên kho gần nhất.

```
install.packages("geosphere")
find_nearest_warehouse <- function(customer, warehouses) {
  distances <- geosphere::distVincentySphere(
    cbind(customer["customer_long"], customer["customer_lat"]),
    cbind(warehouses$lon, warehouses$lat)
  )
  nearest_warehouse <- warehouses[which.min(distances), ]
  return(nearest_warehouse$names)
}
empty_rows <- missing_data$nearest_warehouse == ""
missing_data$nearest_warehouse[empty_rows] <- apply(missing_data[empty_rows, c("customer_long", "customer_lat")], 1, function(row) find_nearest_warehouse(row, warehouses))
```

Nhận xét: những ô trống trong cột nearest_warehouse đã được điền đầy đủ.

2.2.3 Xử lí cột is_happy_customer

Xây dựng hàm nhận biết xem khách hàng có vừa lòng với đơn hàng hay không dựa vào đánh giá từ cột latest_customer_review, những khách hàng hài lòng là những khách hàng có review tốt, có những từ khoá thể hiện sự hài lòng.

```
get_sentiment <- function(review) {
  positive_keywords <- c("good", "excellent", "love", "like", "amazing", "happy", "recommend", "great", "nice")
  return(any(grepl(paste(positive_keywords, collapse = "|"), tolower(review))))
}
missing_data$is_happy_customer <- ifelse(missing_data$is_happy_customer == "", sapply(missing_data$latest_customer_review, get_sentiment), missing_data$is_happy_customer)
```

Lúc này, những ô dữ liệu trống trong cột biến is_happy_customer đã được gán đầy đủ nhưng chưa đồng nhất về cách viết do trong file dữ liệu là True/False còn hàm get_setiment mặc định trả về TRUE/FALSE, vì vậy ta cần thêm một bước để đồng nhất cột dữ liệu này.

```
missing_data$is_happy_customer[missing_data$is_happy_customer == 'TRUE'] <- "True"
```

```
missing_data$is_happy_customer[missing_data$is_happy_customer == 'FALSE'] <- "False"
```

2.2.4 Xử lí cột order_price

Ô order_price được liên hệ bởi công thức:

order_price= (order_total-delivery_charges)/(1-coupon_discount/100).

Từ công thức trên xây dựng hàm trả về kết quả cần tìm và lưu vào những ô NA của order_price.

```
get_price <- function(total, discount, deli){
  price= (total-deli)/(1-discount/100)
  return(price)
}
missing_data$order_price[is.na(missing_data$order_price)] <-
get_price(missing_data$order_total[is.na(missing_data$order_price)],missing_data$coupon_discount[is.na(mi
ssing_data$order_price)],missing_data$delivery_charges[is.na(missing_data$order_price)])
```

2.2.5 Xử lí cột order_total

Ô order_total được liên hệ bằng công thức:

order_total= order_price*(1-coupon_discount/100)+delivery_charges

Từ công thức trên xây dựng hàm trả về kết quả cần tìm và lưu vào những ô NA của order_total.

```
get_total <- function(price, discount, deli){
  total= price*(1-discount/100)+deli
  return(total)
}
missing_data$order_total[is.na(missing_data$order_total)] <-
get_total(missing_data$order_price[is.na(missing_data$order_total)],missing_data$coupon_discount[is.na(mi
ssing_data$order_total)],missing_data$delivery_charges[is.na(missing_data$order_total)])
```

2.2.6 Xử lí cột distance_to_nearest_warehouse

Dựa trên hàm tìm nearest_warehouse để xây dựng nên hàm tìm distance_to_nearest_warehouse.

```
get_distance <- function(customer, warehouses){
  distances <- geosphere::distVincentySphere(
    cbind(customer["customer_long"], customer["customer_lat"]),
    cbind(warehouses$lon, warehouses$lat)
```

```
  )
  distance <- min(distances)
  return(distance/100)
}
missing_data$distance_to_nearest_warehouse[is.na(missing_data$distance_to_nearest_warehouse)] <- apply(missing_data[is.na(missing_data$distance_to_nearest_warehouse), c("customer_long", "customer_lat")], 1, function(row2) get_distance(row2, warehouses))
```

2.2.7 Xử lí những ô còn lại là NA

Những ô NA còn lại là những ô dữ liệu không quan trọng vì vậy ta có thể xoá những hàng chứa dữ liệu NA mà không ảnh hưởng đến kết quả

```
list_na <- colnames(missing_data)[ apply(missing_data, 2, anyNA) ]
list_na
install.packages("dplyr")
library(dplyr)
missing_data_drop <-missing_data %>%
na.omit()
```

IV.THỐNG KÊ MÔ TẢ

1. Chuyển đổi kiểu dữ liệu

Tạo một tệp new_data mới bao gồm các biến cần phân tích(bỏ đi 1 số biến không cần thiết) như là order_id, customer_id,…:

```
new_data<-
missing_data[,c("nearest_warehouse","order_price","delivery_charges","customer_lat","customer_long","coupon_discount","order_total","season","is_expedited_delivery","distance_to_nearest_warehouse","is_happy_customer")]
```

Từ hàm str(), ta có thể nhận thấy một số cột chứa loại dữ liệu là kiểu kí tự (character). Trong R Studio, kiểu kí tự không có tác dụng trong thống kê, vì vậy cần chuyển đổi những cột có dữ liệu kiểu ký tự (nearest_warehouse, season) sang kiểu Factor. Đối với biến chỉ có giá trị False/True như **is_expedited_delivery** và **is_happy_customer**, thay False=0, True=1.

```
new_data_0$is_expedited_delivery[new_data$is_expedited_delivery == 'True'] <- 1
new_data$is_expedited_delivery[new_data$is_expedited_delivery == 'False'] <- 0
new_data$is_expedited_delivery <- as.integer(new_data$is_expedited_delivery)
```

```
new_data$is_happy_customer[new_data$is_happy_customer == 'True'] <- 1
new_data$is_happy_customer[new_data$is_happy_customer == 'False'] <- 0
```

Sau khi chuyển đổi sang kiểu Factor, cần thực hiện sắp xếp trật tự thứ bậc của các dữ liệu trong cột "season":

```
> new_data$season <- factor((new_data$season),   levels= c("Spring","Summer", "Autumn","Winter"))
> levels(new_data$season)
[1] "Spring" "Summer" "Autumn" "Winter"
```

```
> new_data$nearest_warehouse <- factor((new_data$nearest_warehouse),   levels= c("Bakers","Nickolson","Thompson"))
> levels(new_data$nearest_warehouse)
[1] "Bakers"    "Nickolson" "Thompson"
```

2. Phân tích thống kê mô tả

2.1 Tổng quát về dữ liệu

Trong phân tích thống kê, thống kê mô tả là bước quan trọng vì bước này cho cái nhìn trực quan về số liệu của mẫu thống kê. Hàm summary() là một trong những hàm hữu ích được tích hợp sẵn trong R Studio nhằm giúp việc thống kê mô tả dễ dàng hơn. Sử dụng hàm summary() để xem mô tả tổng quan về dữ liệu trong file dữ liệu đã được làm sạch new_data.

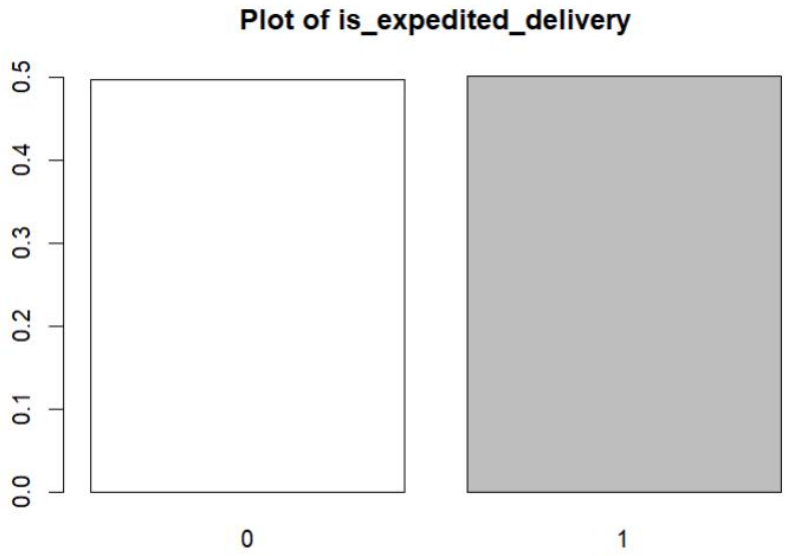
```
> summary(new_data)
 nearest_warehouse  order_price    delivery_charges  customer_lat    customer_long   coupon_discount  order_total        season
 Bakers    :105     Min.   :  580   Min.   : 46.20   Min.   :-37.83   Min.   :144.9   Min.   : 0.00   Min.   :  568.6   Spring:110
 Nickolson:173     1st Qu.: 7012   1st Qu.: 67.14   1st Qu.:-37.82   1st Qu.:145.0   1st Qu.: 5.00   1st Qu.: 6271.4   Summer:143
 Thompson :202     Median :12220   Median : 77.39   Median :-37.81   Median :145.0   Median :10.00   Median :10687.6   Autumn:134
                   Mean   :13242   Mean   : 77.84   Mean   :-37.81   Mean   :145.0   Mean   :11.21   Mean   :11863.7   Winter: 93
                   3rd Qu.:18323   3rd Qu.: 85.41   3rd Qu.:-37.81   3rd Qu.:145.0   3rd Qu.:15.00   3rd Qu.:16231.7
                   Max.   :37300   Max.   :110.99   Max.   :-37.79   Max.   :145.0   Max.   :25.00   Max.   :37362.5
 is_expedited_delivery distance_to_nearest_warehouse is_happy_customer
 Min.   :0.0000        Min.   : 0.0549               Min.   :0.0000
 1st Qu.:0.0000        1st Qu.: 0.7324               1st Qu.:1.0000
 Median :1.0000        Median : 1.0576               Median :1.0000
 Mean   :0.5021        Mean   : 1.2638               Mean   :0.7729
 3rd Qu.:1.0000        3rd Qu.: 1.4199               3rd Qu.:1.0000
 Max.   :1.0000        Max.   :18.0980               Max.   :1.0000
```

Hình 4.1 *Tổng quan dữ liệu*

Đối với biến định lượng, hàm summary() cho biết các giá trị đặc trưng của mẫu như giá trị max,min, mean, tứ phân vị thứ nhất và phần tư, trung vị, còn đối với biến phân loại hàm này chủ yếu thống kê số lượng của từng biến phân loại.

2.2 Phân tích chi tiết

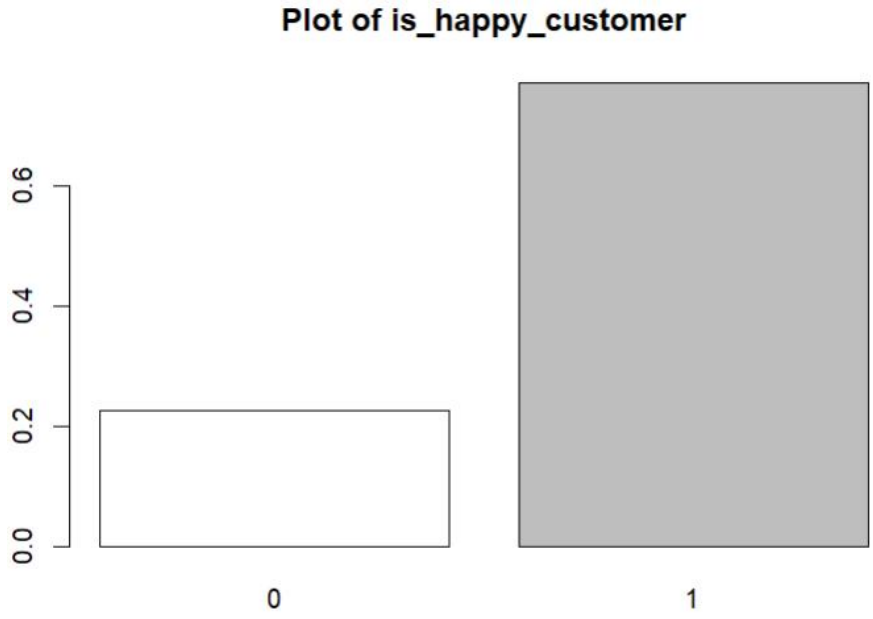
Đối với biến phân loại như is_expedited_delivery, bar plot là đồ thị thích hợp để biểu diễn. Vẽ đồ thị bar plot đối với biến is_expedited_delivery (đơn vị là %):



Hình 4.2 *Biểu đồ thể hiện tỉ lệ khách hàng hàng yêu cầu giao hàng nhanh*

Nhận xét: từ biểu đồ ta thấy, có 50% khách có nhu cầu giao hàng nhanh.

Đồ thị bar plot cho biến is_customer_happy (đơn vị %):

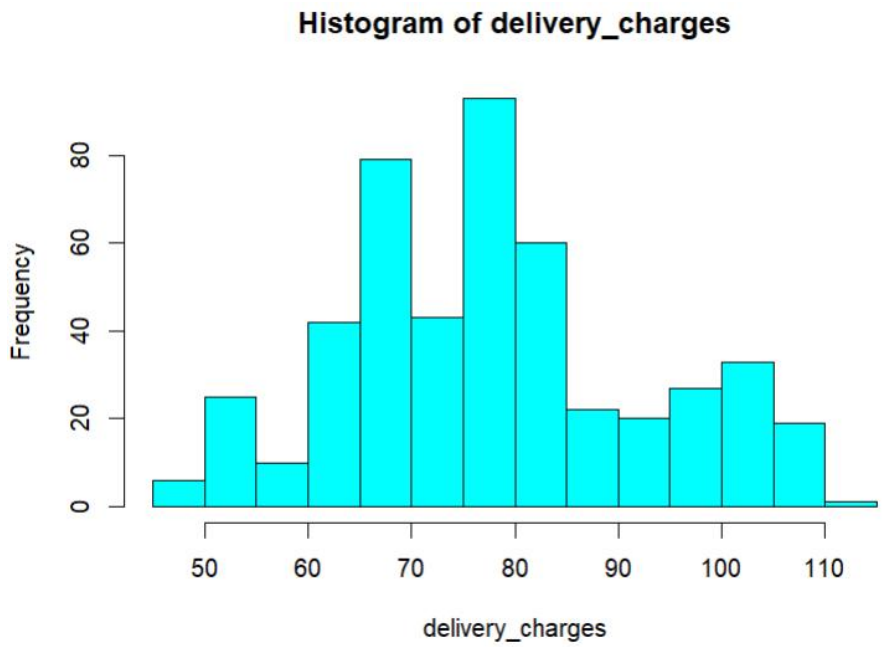


Hình 4.3 *Biểu đồ thể hiện tỉ lệ hài lòng của khách hàng*

Nhận xét: tỷ lệ khách hàng hài lòng chiếm hầu hết.

Đối với biến định lượng, đồ thị bar plot không còn thích hợp, thay vào đó, histogram là đồ thị thích hợp hơn cho việc biểu diễn sự phân bố của các biến định lượng.

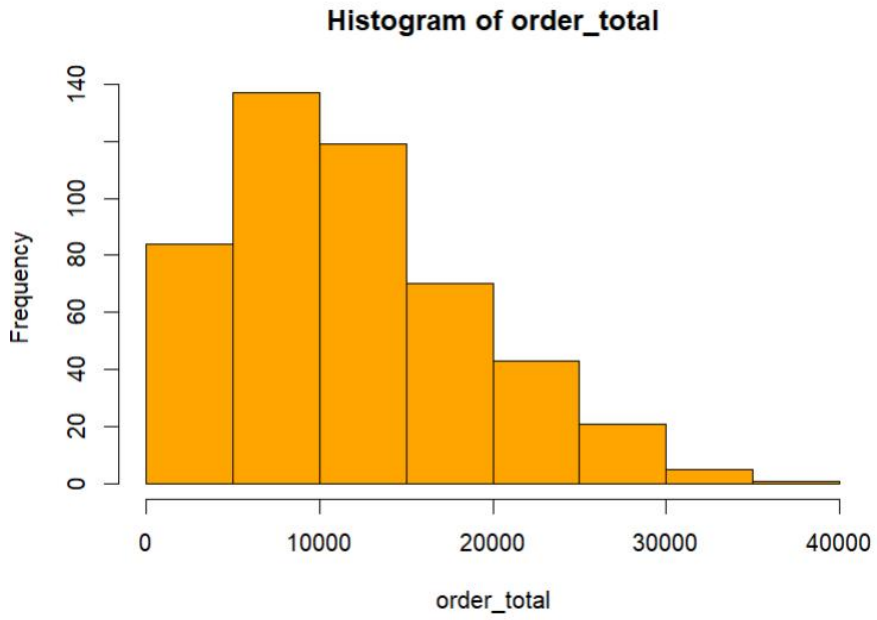
•Đồ thị histogram cho biến delivery_charges:



Hình 4.4 *Biểu đồ phân phối của biến delivery_charges*

Nhận xét: Chi phí giao hàng chủ yếu dao động từ giá trị 60 đến 80.

•Đồ thị histogram cho biến order_total:



Hình 4.6 Biểu đồ phân phối của biến order_total

Nhận xét: chi phí đơn hàng sau khi được áp mã giảm giá chủ yếu phân bố dao động quanh giá trị 10000..

2.2 Xử lí ngoại lai

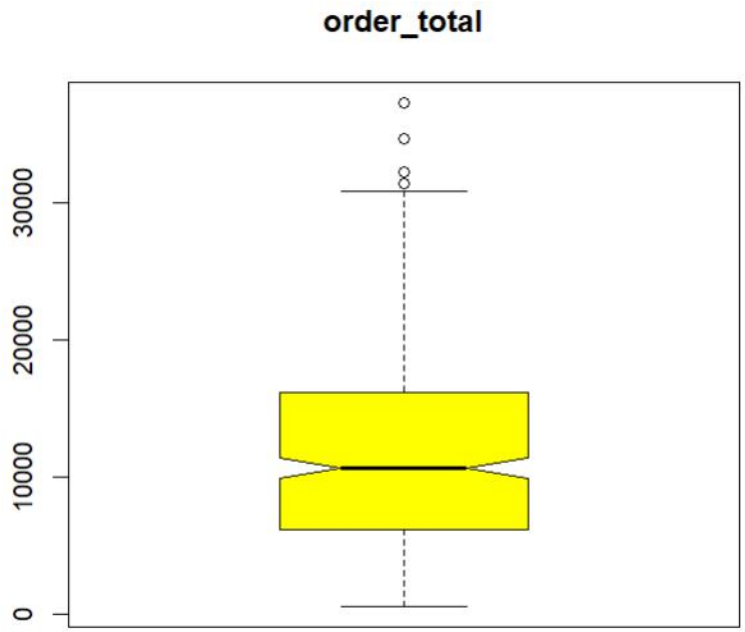
Dữ liệu ngoại lai là những giá trị dữ liệu(records) được ghi nhận có sự khác biệt bất thường so với những giá trị dữ liệu khác, không theo một quy tắc chung nào và có thể gây ra sự sai lệch trong kết quả phân tích và việc xây dựng thuật toán dự đoán.

Ngoài đồ thị histogram, boxplot cũng là một trong những dạng đồ thị phổ biến để biểu thị phân bố của biến, boxplot có một ứng dụng quan trọng có ý nghĩa trong thống kê là biểu thị những điểm ngoại lai. Từ đây ta có thể nhận biết được những biến nào có giá trị ngoại lai cần loại bỏ thông qua việc sử dụng đồ thị boxplot.

Đối với boxplot, dữ liệu ngoại lai được xác định là những dấu chấm ở 2 đầu của biểu đồ.

Có nhiều phương pháp thống kê để xử lí ngoại lai, trong bài báo cáo này, nhóm sử dụng phương pháp IQR(Interquartile Range) để xác định giới hạn của các giá trị ngoại lai.

•Đồ thị boxplot cho biến order_total:



Hình 4.7 Biểu đồ thể hiện phân bố và ngoại lai của biến order_total

•Đồ thị boxplot cho biến order_price:

Hình 4.8 *Biểu đồ thể hiện phân bố và ngoại lai của biến order_price*

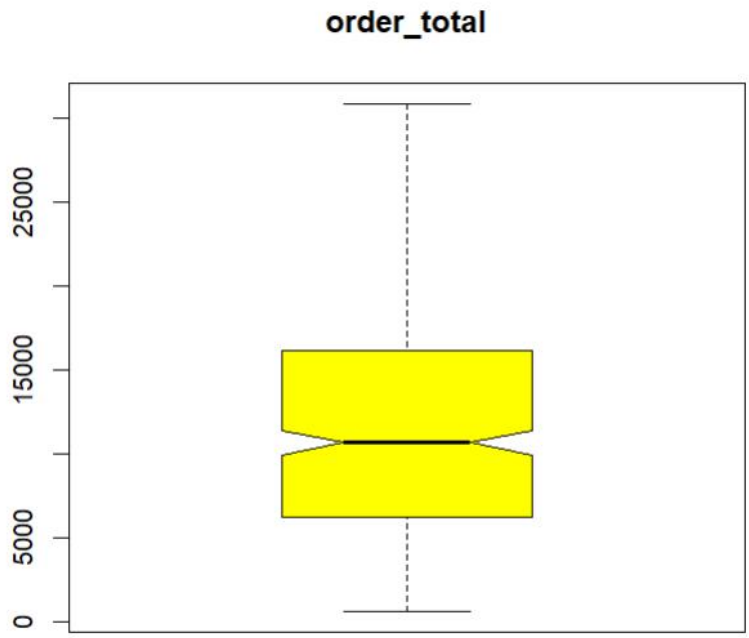
Từ đồ thị trên cho thấy có nhiều điểm ngoại lai khiến ta không thể nhìn rõ được phân phối, cần xử lí những điểm ngoại lai.

Xây dựng hàm xử lí ngoại lai:

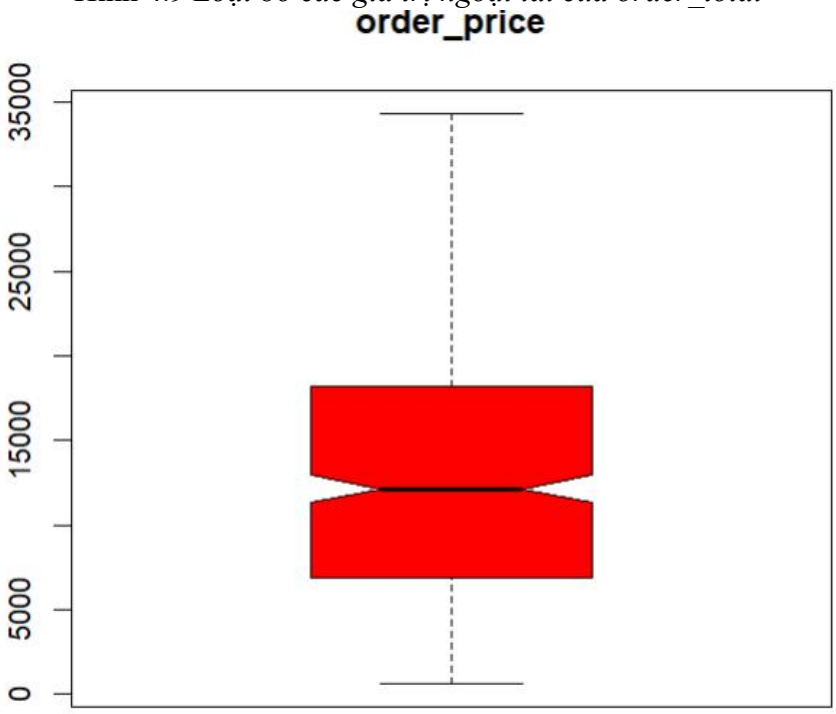
```
detect_outlier <- function(x, na.rm = TRUE, ...){
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  y <- x
  y[x < (qnt[1] - H)] <- NA
  y[x > (qnt[2] + H)] <- NA
  y
}
new_data$order_total <-detect_outlier(new_data$order_total)
new_data$order_price<-detect_outlier(new_data$order_price)
new_data<-na.omit(new_data)
```

Những biến nằm ngoài khoảng IQR sẽ được gán cho giá trị NA, từ đó chỉ cần loại bỏ những biến NA ta sẽ thu được file dữ liệu sạch đã xoá các biến ngoại lai

•Kiểm tra lại bằng đồ thị boxplot:

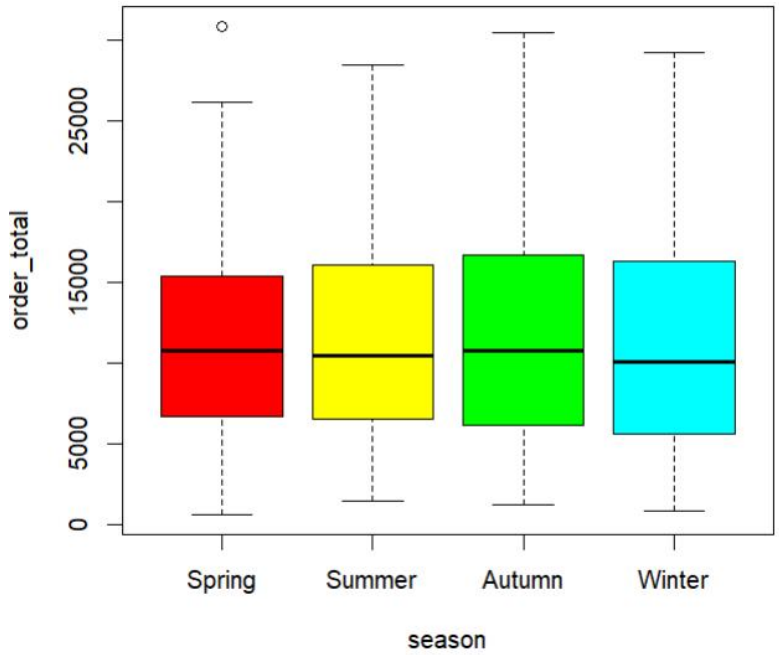


Hình 4.9 *Loại bỏ các giá trị ngoại lai của order_total*



Hình 4.10 *Loại bỏ các giá trị ngoại lai của order_price*

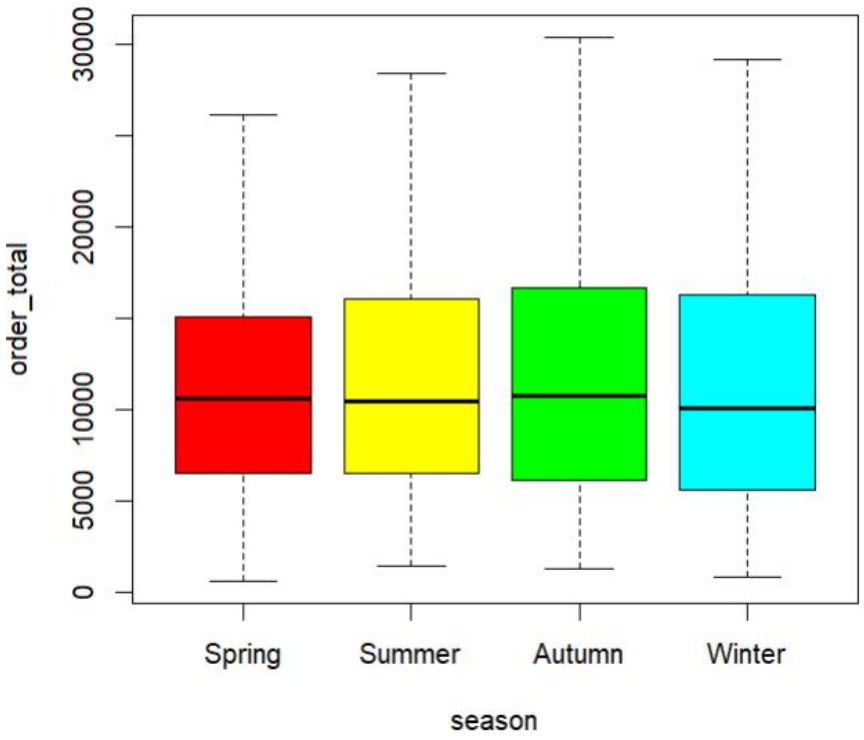
•Bây giờ tiến hành kiểm tra ngoại lai của biến order_total theo từng mùa:



Hình 4.11 *Biểu đồ thể hiện phân bố và ngoại lai của biến order_total theo từng mùa*

Nhận xét: khi xét theo từng mùa riêng lẻ, biến order_total xuất hiện 1 điểm ngoại lai ứng với mùa xuân, vì vậy ta cần tiếp tục xử lí điểm ngoại lai này.

```
Spring_data = subset(new_data,new_data$season =="Spring")
Spring_data$order_total = detect_outlier(Spring_data$order_total)
Summer_data = subset(new_data,new_data$season =="Summer")
Summer_data$order_total = detect_outlier(Summer_data$order_total)
Autumn_data = subset(new_data,new_data$season =="Autumn")
Autumn_data$order_total = detect_outlier(Autumn_data$order_total)
Winter_data = subset(new_data,new_data$season =="Winter")
Winter_data$order_total = detect_outlier(Winter_data$order_total)
new_data_2 <- rbind(Spring_data,Summer_data,Autumn_data,Winter_data)
apply(is.na(new_data_2),2,sum)
apply(is.na(new_data_2),2,mean)
new_data_2<-na.omit(new_data_2)
```



Hình 4.12 *Tất cả các điểm ngoại lai của order_total đã được xử lí*

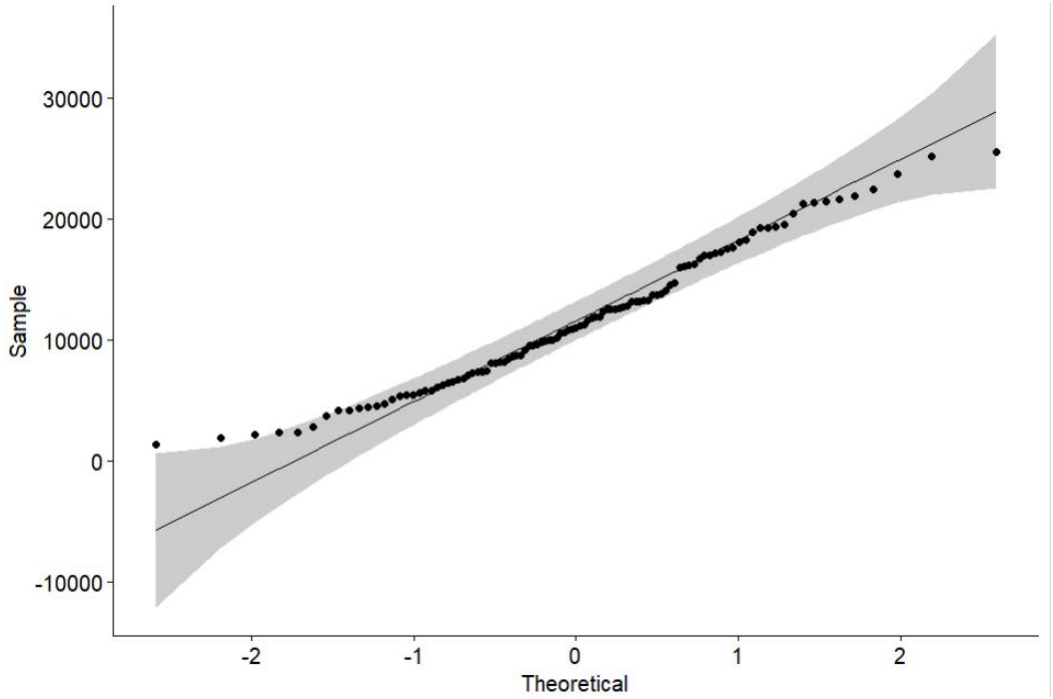
2.3 Kiểm tra giả thiết phân phối chuẩn của biến order_total

Kiểm tra các giả định:

+ Giả định 1: chi phí đặt hàng ở các kho hàng tuân theo phân phối chuẩn:

```
Bakers_data_2<-subset(new_data_2,new_data_2$nearest_warehouse =="Bakers")
qqnorm(Bakers_data_2$order_total)
qqline(Bakers_data_2$order_total)
```

shapiro.test(Bakers_data_2$order_total)



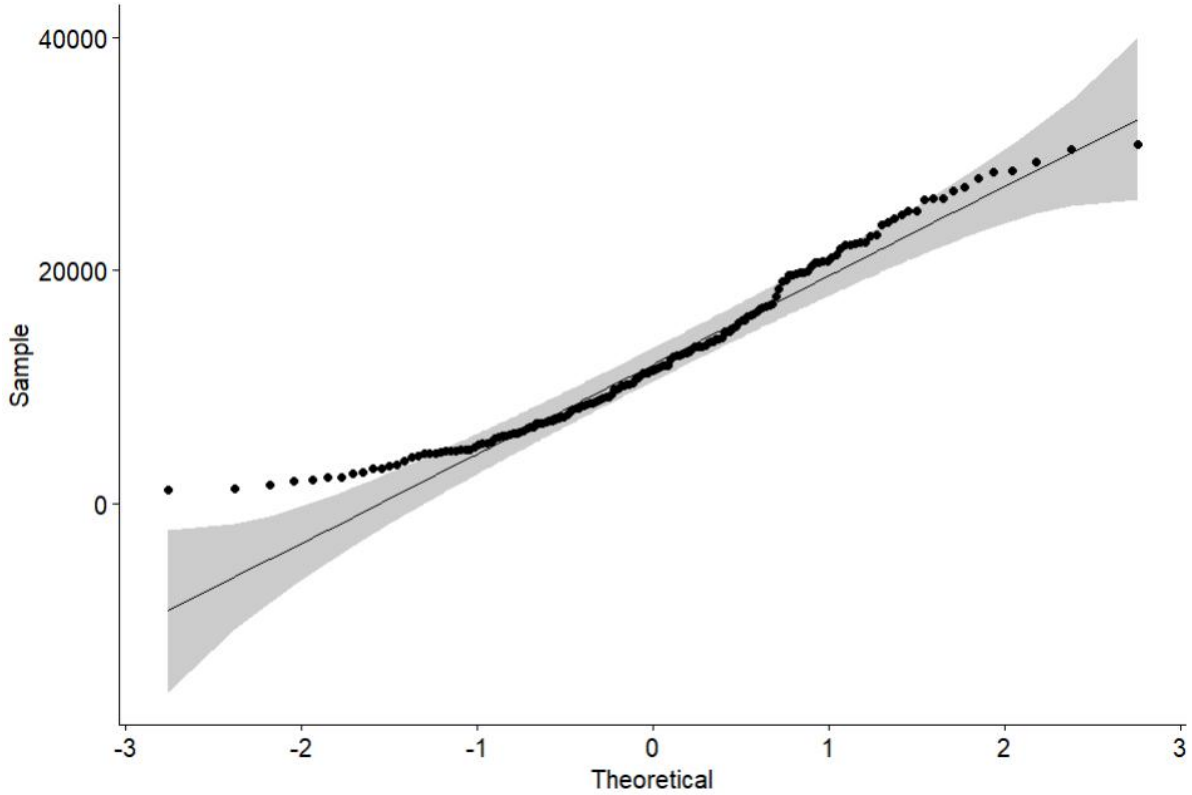

```
> shapiro.test(Bakers_data_2$order_total)

        Shapiro-Wilk normality test

data:  Bakers_data_2$order_total
W = 0.97346, p-value = 0.03318
```

Nickolson_data_2<-subset(new_data_2,new_data_2$nearest_warehouse =="Nickolson")
ggqqplot(Nickolson_data_2$order_total)

shapiro.test(Nickolson_data_2$order_total)



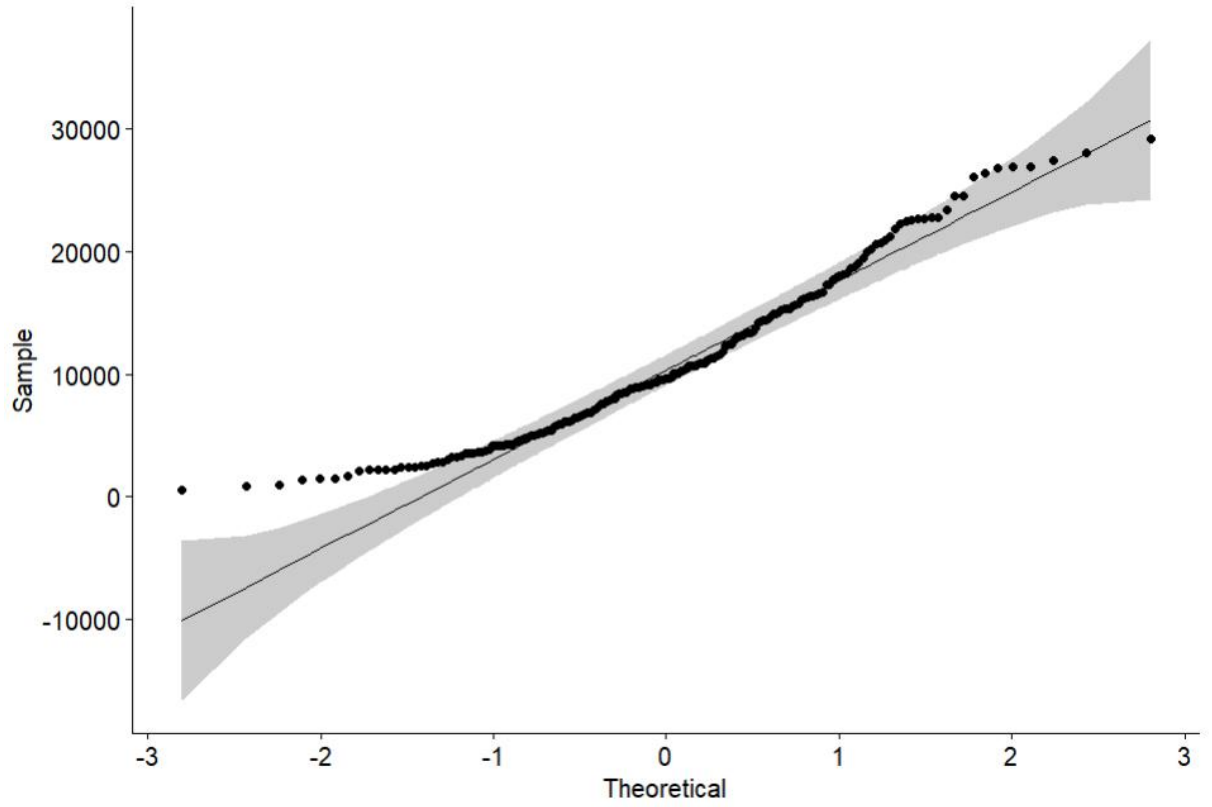

```
> shapiro.test(Nickolson_data_2$order_total)

        Shapiro-Wilk normality test

data:  Nickolson_data_2$order_total
W = 0.95201, p-value = 1.438e-05
```

Thompson_data_2<-subset(new_data_2,new_data_2$nearest_warehouse =="Thompson")
ggqqplot(Thompson_data_2$order_total)

shapiro.test(Thompson_data_2$order_total)




```
> shapiro.test(Thompson_data_2$order_total)

        Shapiro-Wilk normality test

data:  Thompson_data_2$order_total
W = 0.94858, p-value = 1.381e-06
```

Nhận xét: Dựa trên phân tích QQ-plot, có thể nhận thấy rằng các quan sát không khớp với đường thẳng, đề xuất rằng chi phí đặt hàng tại các kho không tuân theo phân phối chuẩn. Thêm vào đó, p-value của các kiểm định cũng đều rất thấp, thậm chí bé hơn mức ý nghĩa 5%, điều này cung cấp thêm chứng cứ cho việc chi phí đặt hàng ở các kho không tuân theo phân phối chuẩn. Điều này có thể có ảnh hưởng đáng kể đến quá trình quản lý và dự đoán chi phí đặt hàng, và cần được xem xét và điều chỉnh để đảm bảo tính chính xác và đáng tin cậy của mô hình.

V. THỐNG KÊ SUY DIỄN

1. So sánh trung bình về chi phí đặt hàng của khách hàng ở 3 kho hàng để xem có kho hàng nào mà chi phí đặt hàng nhiều hơn không?

Giả thiết H0: Phương sai chi phí đặt hàng ở 3 kho hàng bằng nhau.
Giả thiết đối H1: có ít nhất 2 kho hàng có phương sai chi phí đặt hàng khác nhau.
Với giá trị p-value là 0.4631, lớn hơn mức ý nghĩa 5%, chúng ta không có đủ bằng chứng để bác bỏ giả thuyết H0. Do đó, có thể kết luận rằng không có sự khác biệt đáng kể về phương sai của chi phí đặt hàng giữa ba kho hàng.Thực hiện phân tích phương sai 1 nhân tố:

```
> one.way <- aov(order_total~nearest_warehouse,data=new_data_2)
> summary(one.way)
                   Df    Sum Sq   Mean Sq F value Pr(>F)
nearest_warehouse   2 2.591e+08 129557157   2.834 0.0598 .
Residuals         473 2.162e+10  45714882
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Gỉa thiết H0: chi phí đặt hàng trung bình ở 3 kho hàng bằng nhau.
Giả thiết H1: có ít nhất 2 kho hàng có chi phí đặt hàng trung bình khác nhau.

Với giá trị p-value là 0.059, lớn hơn mức ý nghĩa 5%, chúng ta không có đủ bằng chứng để bác bỏ giả thuyết H0. Do đó, có thể kết luận rằng không có sự khác biệt đáng kể về chi phí đặt hàng trung bình của khách hàng giữa ba kho hàng.

2. So sánh trung bình về chi phí đặt hàng của khách hàng ở 4 mùa để xem có mùa nào khách hàng đặt hàng nhiều nhất không?

Thực hiện phân tích phương sai 1 nhân tố:

```
> one.way_2 <- aov(order_total~season,data=new_data)
> summary(one.way_2)
             Df    Sum Sq  Mean Sq F value Pr(>F)
season        1 1.108e+06  1108029   0.022  0.882
Residuals   478 2.387e+10 49943160
```

Gỉa thiết H0: chi phí đặt hàng trung bình ở 4 mùa bằng nhau.
Giả thiết H1: có ít nhất 2 mùa có chi phí đặt hàng trung bình khác nhau.
Với giá trị p-value là 0.882, lớn hơn mức ý nghĩa 5%, chúng ta không có đủ bằng chứng để bác bỏ giả thuyết H0. Do đó, có thể kết luận rằng không có sự khác biệt về chi phí đặt hàng trung bình của khách hàng giữa bốn mùa trong năm.

3. Xây dựng mô hình hồi quy logistic

•Phân tích các yếu tố ảnh hưởng đến việc hài lòng của khác hàng.
•Đổi kiểu dữ liệu của các biến phân loại thành integer trước khi đưa vào mô hình hồi quy:

```
new_data_4<-  new_data[,c("nearest_warehouse","delivery_charges","coupon_discount",
"season","order_total","distance_to_nearest_warehouse","is_expedited_delivery","is_happy_custome
r")]
new_data_4$season <- as.integer(new_data_4$season)
new_data_4$nearest_warehouse <- as.integer(new_data_4$nearest_warehouse)
```

Xét mô hình hồi quy logistic gồm biến is_happy_customer là biến phụ thuộc và các biến còn lại là biến độc lập.

new_data_4<-
new_data[,c("nearest_warehouse","delivery_charges","customer_lat","customer_long","coupon_discount", "season", "order_total", "distance_to_nearest_warehouse", "is_expedited_delivery", "is_happy_customer")]

```
> model1 <- glm(is_happy_customer~., family="binomial", data=new_data_4)
> summary(model1)

Call:
glm(formula = is_happy_customer ~ ., family = "binomial", data = new_data_4)

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                  3.724e+03  2.680e+03   1.390  0.16464
nearest_warehouse           -6.077e-01  4.647e-01  -1.308  0.19089
delivery_charges             2.286e-01  2.334e-02   9.797  < 2e-16 ***
customer_lat                 1.693e+01  2.162e+01   0.783  0.43355
customer_long               -2.135e+01  1.717e+01  -1.244  0.21368
coupon_discount             -2.412e-02  1.761e-02  -1.370  0.17080
season                      -5.416e-01  1.798e-01  -3.012  0.00259 **
order_total                 -1.177e-05  2.225e-05  -0.529  0.59682
distance_to_nearest_warehouse -1.503e-01  8.171e-02  -1.840  0.06577 .
is_expedited_delivery       -3.847e+00  4.783e-01  -8.043 8.75e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 514.30  on 479  degrees of freedom
Residual deviance: 293.08  on 470  degrees of freedom
AIC: 313.08

Number of Fisher Scoring iterations: 6
```

Nhận xét: các biến nearest_warehouse, customer_lat, customer_long, coupon_discount, order_total, distace_to_nearest_warehouse có p-value> 0,05 nên không có giá trị thống kê.

Xét mô hình hồi quy mới sau khi loại bỏ các biến không có giá trị thống kê trên.

new_data_5<- new_data[,c("delivery_charges", "season", "is_expedited_delivery", "is_happy_customer")]
model2 <- glm(is_happy_customer~., family="binomial", data=new_data_5)

```
> model2 <- glm(is_happy_customer~., family="binomial", data=new_data_5)
> summary(model2)

Call:
glm(formula = is_happy_customer ~ ., family = "binomial", data = new_data_5)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -11.94754    1.36216  -8.771  < 2e-16 ***
delivery_charges        0.22076    0.02238   9.866  < 2e-16 ***
season                 -0.48753    0.17487  -2.788   0.0053 **
is_expedited_delivery  -3.61257    0.45468  -7.945 1.94e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 514.30  on 479  degrees of freedom
Residual deviance: 299.71  on 476  degrees of freedom
AIC: 307.71

Number of Fisher Scoring iterations: 6
```

Nhận xét: việc loại bỏ những biến không có giá trị thống kê trong mô hình 1 cho mô hình 2 chỉ bao gồm các biến có giá trị thống kê. Giá trị AIC của mô hình 2 nhỏ hơn (307,71<313,08)

4. Kiểm định mô hình bằng phương pháp phân tích phương sai (ANOVA)

Giả thuyết H0: 2 mô hình có hiệu quả như nhau.

Giả thuyết H1: 2 mô hình có hiệu quả khác nhau.

```
> anova(model1, model2, test= "LRT")
Analysis of Deviance Table

Model 1: is_happy_customer ~ nearest_warehouse + delivery_charges + customer_lat +
    customer_long + coupon_discount + season + order_total +
    distance_to_nearest_warehouse + is_expedited_delivery
Model 2: is_happy_customer ~ delivery_charges + season + is_expedited_delivery
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       470     293.08
2       476     299.71 -6  -6.6347    0.356
```

Df là bậc tự do, tổng bình phương là Sum Sq, giá trị trung bình của tổng bình phương là Mean Sq, giá trị của F là F value và p-value.

Nhận xét: từ kết quả phân tích phương sai trên ta thấy giá trị p-value= 0,356 >0,05 nên chưa đủ cơ sở bác bỏ H0. Ở mức ý nghĩa 5%, chúng ta có thể kết luận rằng cả hai mô hình có hiệu quả như nhau, ta chọn mô hình 2 vì mô hình này chỉ chứa các biến có ý nghĩa thống kê và giá trị AIC nhỏ hơn.

Ta tính được logit(P) (với P là xác suất khách hàng hài lòng):

Logit(P) = 0,22076 x **deliver_charges** – 0,4875x**season** – 3,6125 x **is_expedited_delivery** -11,9475.

Ý nghĩa của các tham số trong mô hình hồi quy:

Hệ số chặn(intercept)=- 11,9475là giá trị logit(P) khi các biến độc lập còn lại bằng 0. Trong trường hợp này logit(P)= -11,9475.

Null deviance: Độ lệch khi không có biến độc lập trong mô hình.

Residual deviance: Độ lệch khi có biến độc lập trong mô hình.

Hệ số hồi quy của biến độc lập vừa phản ánh mức độ tác động đồng thời cũng thể hiện chiều tác động của biến độc lập lên biến phụ thuộc. Ví dụ: đối với chi phí giao hàng (delivery_charges), nếu tăng chi phí lên 1 đơn vị thì giá trị của logit(P) tăng lên 0,22076, điều này ảnh hưởng đến mức độ hài lòng của khách hàng.

VI. Thảo luận và mở rộng

1. Hồi quy từng bước

Ngoài cách kiểm định anova, ta có thể dùng hồi quy từng bước để chọn ra mô hợp lí hơn với sai số dự đoán của mô hình sau nhỏ hơn mô hình trước. Đối với R studio, phần mềm cung cấp hàm step() để công việc hồi quy từng bước trở nên dễ dàng hơn, hàm này chọn ra mô hình có sai số dự đoán(AIC) nhỏ nhất.

Sử dụng mô hình 3 với đầy đủ các biến biến độc lập, hàm step() thực hiện hồi quy từng bước để loại ra những biến không phù hợp.

```
Call:
glm(formula = is_happy_customer ~ delivery_charges + season +
    distance_to_nearest_warehouse + is_expedited_delivery, family = "binomial",
    data = new_data_4)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -11.81740    1.36979  -8.627  < 2e-16 ***
delivery_charges             0.22397    0.02273   9.855  < 2e-16 ***
season                      -0.54246    0.17951  -3.022  0.00251 **
distance_to_nearest_warehouse -0.13693    0.07983  -1.715  0.08627 .
is_expedited_delivery       -3.72883    0.46603  -8.001 1.23e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 514.30  on 479  degrees of freedom
Residual deviance: 297.07  on 475  degrees of freedom
AIC: 307.07

Number of Fisher Scoring iterations: 6
```

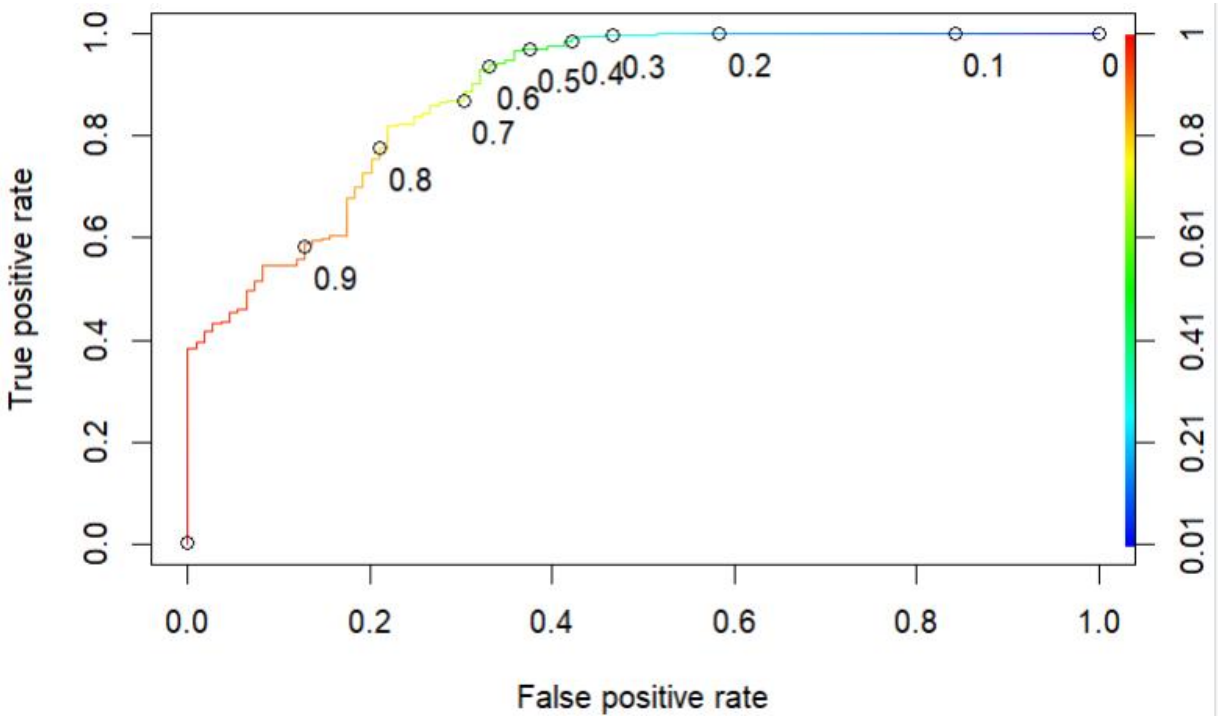
Nhận xét: sau khi thực hiện hồi quy từng bước, mô hình bây giờ có thêm sự xuất hiện của biến distance_to_nearest_warehouse, ta nhận thấy giá trị AIC=307,07 nhỏ hơn 2 mô hình trên. Điều này chứng tỏ đây là mô hình có sai số dự đoán nhỏ nhất. Tuy nhiên, trong trường hợp này, số lượng biến độc lập nhiều hơn so với mô hình 2 và p-value của biến distance_to_nearest_warehouse >5% vì vậy ta vẫn chọn mô hình 2. Ta nhận thấy rằng hồi quy từng bước thích hợp với những mô hình lớn với nhiều biến độc lập, hồi quy từng bước giúp lọc nhanh chóng những biến không cần thiết và giữ lại những biến quan trọng cho mô hình.

2. Xác định tính phù hợp của mô hình bằng đường cong ROC

Đường cong ROC là một biểu đồ cho thấy hiệu suất của mô hình phân loại ở tất cả các ngưỡng phân loại.

Mỗi điểm trên đường cong ROC là tọa độ tương ứng với tần suất dương tính thật (độ nhạy) trên trục tung và tần suất dương tính giả (1-độ đặc hiệu) trên trục hoành. Đường biểu diễn càng lệch về phía bên trên và bên trái thì sự phân biệt giữa 2 trạng thái (ví dụ có bệnh hoặc không bệnh) càng rõ.

```
dubao = predict(model2, type= "response", newdata = new_data_6)
install.packages("ROCR")
library(ROCR)
ROCRpred = prediction(dubao, new_data_6$is_happy_customer)
ROCRperf = performance(ROCRpred, "tpr", "fpr")
plot(ROCRperf, colorize=TRUE, print.cutoffs.at= seq(0,1,by=0.1), text.adj=c(-0.2,1.7))
```



Nhận xét: đường cong ROC cho thấy đây là mô hình cho dự đoán có độ chính xác rất tốt.

VII. TÀI LIỆU THAM KHẢO

1. Nguồn code
https://drive.google.com/drive/folders/19ljw9STSOQ2zHbGxqfliAgqViN6VfRkp?usp=sharing
2. Coolican, H. (2018). Research methods and statistics in psychology. Routledge
3. Hanneman, R. A., Kposowa, A. J., & Riddle, M. D. (2012). Basic statistics for social research (Vol. 38). John Wiley & Sons.
4. Hoàng Trọng và Chu Nguyễn Mộng Ngọc, Phân tích dữ liệu nghiên cứu với SPSS. Nhà xuất bản thống kê năm 2005
5. Nguyễn Văn Tuấn. 2007. Phân tích hồi qui logistic trong: Phân tích số liệu và tạo biểu đồ bằng R. Nhà Xuất bản Khoa học và Kỹ thuật. trang 215-218.
6. Michael (2023), (thảo luận về Latitude Longitude Coordinates to State Code in R), truy cập từ https://stackoverflow.com/questions/8751497/latitude-longitude-coordinates-to-state-code-in-r