PHE Rebuy Rate Analysis

Ryan Mersereau

2026-02-27

```
#Attaching packages
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.5.2

## Warning: package 'ggplot2' was built under R version 4.5.1

## Warning: package 'tidyr' was built under R version 4.5.2

## Warning: package 'readr' was built under R version 4.5.1

## Warning: package 'purrr' was built under R version 4.5.2

## Warning: package 'stringr' was built under R version 4.5.2

## Warning: package 'forcats' was built under R version 4.5.2

## Warning: package 'lubridate' was built under R version 4.5.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.2
## v purrr     1.2.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(dplyr)
library(ggplot2)
```

```
orders <- read.csv("C:/Users/gemer/Downloads/Finalorderscsv.csv")
```

```
#Converting date objects and adding time to delivery variable
orders$dateAndTime <- ymd_hms(orders$dateAndTime, tz = "UTC")
```

```
## Warning: 206 failed to parse.
```

```r
orders$dateAndTime <- as.Date(orders$dateAndTime)
orders$dateAndTime <- format(orders$dateAndTime, format = "%m/%d/%Y")

orders <- orders %>%
  rename(arrival_date = dateAndTime)

orders$arrival_date <- as.Date(orders$arrival_date, format = "%m/%d/%Y")
orders$shipping_date <- as.Date(orders$shipping_date, format = "%m/%d/%Y")

orders <- orders %>%
  mutate(time_to_delivery = difftime(orders$arrival_date, orders$shipping_date,
                                     units = "days"))

orders$time_to_delivery <- as.integer(orders$time_to_delivery)

orders <- drop_na(orders)
```

```r
#Table Stats
sum(orders$times_ordered == 1)
```

```
## [1] 7933
```

```r
sum(orders$times_ordered > 1)
```

```
## [1] 1529
```

```r
# n = 9462, 7933 orders from customers who ordered once,
# 1529 from customers who ordered previously or went on to order again
```

```r
sum(orders$time_to_delivery == 1)
```

```
## [1] 1
```

```r
n_distinct(orders$g_user_id)
```

```
## [1] 8432
```

```r
#Count number of unique guids, for repeat buyers select earliest date
# (first order)

orders %>%
  group_by(times_ordered) %>%
  summarise(count = n_distinct(g_user_id))
```

```
## # A tibble: 6 x 2
##   times_ordered count
##           <int> <int>
## 1             1  7704
## 2             2   589
## 3             3   102
## 4             4    25
## 5             5     6
## 6             6     6
```

```
mean(orders[orders$times_ordered == 1, "time_to_delivery"])
```

```
## [1] 4.857557
```

```
#Average for non-repeat customers: 4.857 days

mean(orders[orders$times_ordered > 1, "time_to_delivery"])
```

```
## [1] 4.805755
```

```
#Average for repeat customers: 4.805 days

mean(orders$time_to_delivery)
```

```
## [1] 4.849186
```

```
#Average delivery time: 4.849 days

sd(orders$time_to_delivery)
```

```
## [1] 1.959202
```

```
#Std dev: 1.959 days
```

```
#Create new firstorders table
firstorders <- orders %>%
  mutate(order_create_date = as.Date(order_create_date, "%m/%d/%Y")) %>%
  group_by(g_user_id) %>%
  filter(order_create_date == min(order_create_date)) %>%
  filter(1:n() == 1)
```

```
#Compare averages for first ship times by times ordered from first orders
firstorders %>%
  group_by(times_ordered) %>%
  summarise(minimum = min(time_to_delivery),
            Q1 = quantile(time_to_delivery, probs = .25),
            mean = mean(time_to_delivery),
            median = median(time_to_delivery),
            Q3 = quantile(time_to_delivery, probs = .75),
            maximum = max(time_to_delivery))
```

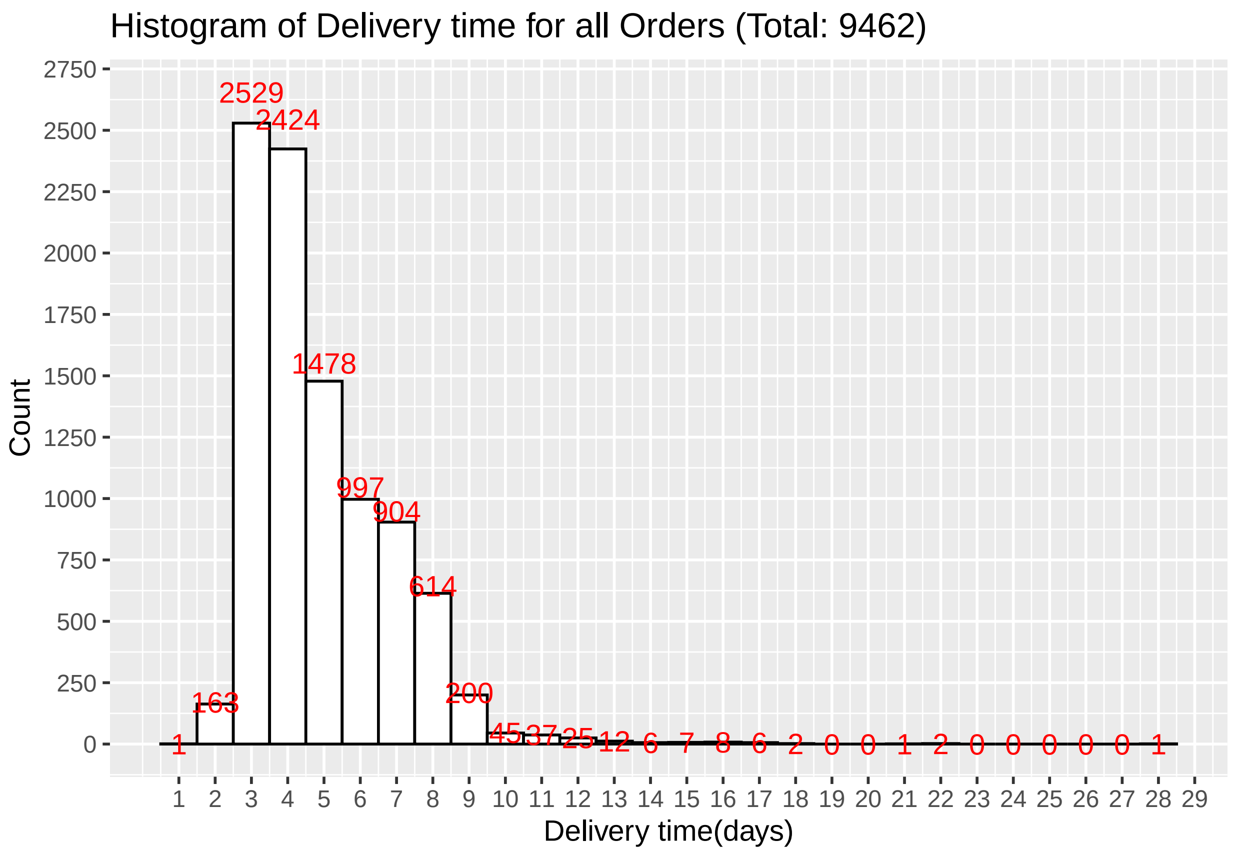
```
## # A tibble: 6 x 7
##   times_ordered minimum    Q1  mean median    Q3 maximum
##           <int>   <int> <dbl> <dbl>  <dbl> <dbl>   <int>
## 1             1       1     3  4.86      4     6      28
## 2             2       2     3  4.84    4    6         17
## 3             3       2     3  4.65    4    6         11
## 4             4       3     4  5.68    6    7         12
## 5             5       3     4  4.83    4.5  5.75       7
## 6             6       4     4  4.5     4    4.75       6
```

```
#Compare average number of rebuys by delivery time from first orders
firstorders %>%
  group_by(time_to_delivery) %>%
  summarise(minimum = min(times_ordered),
            Q1 = quantile(times_ordered, probs = .25),
            mean = mean(times_ordered),
            median = median(times_ordered),
            Q3 = quantile(times_ordered, probs = .75),
            maximum = max(times_ordered))
```

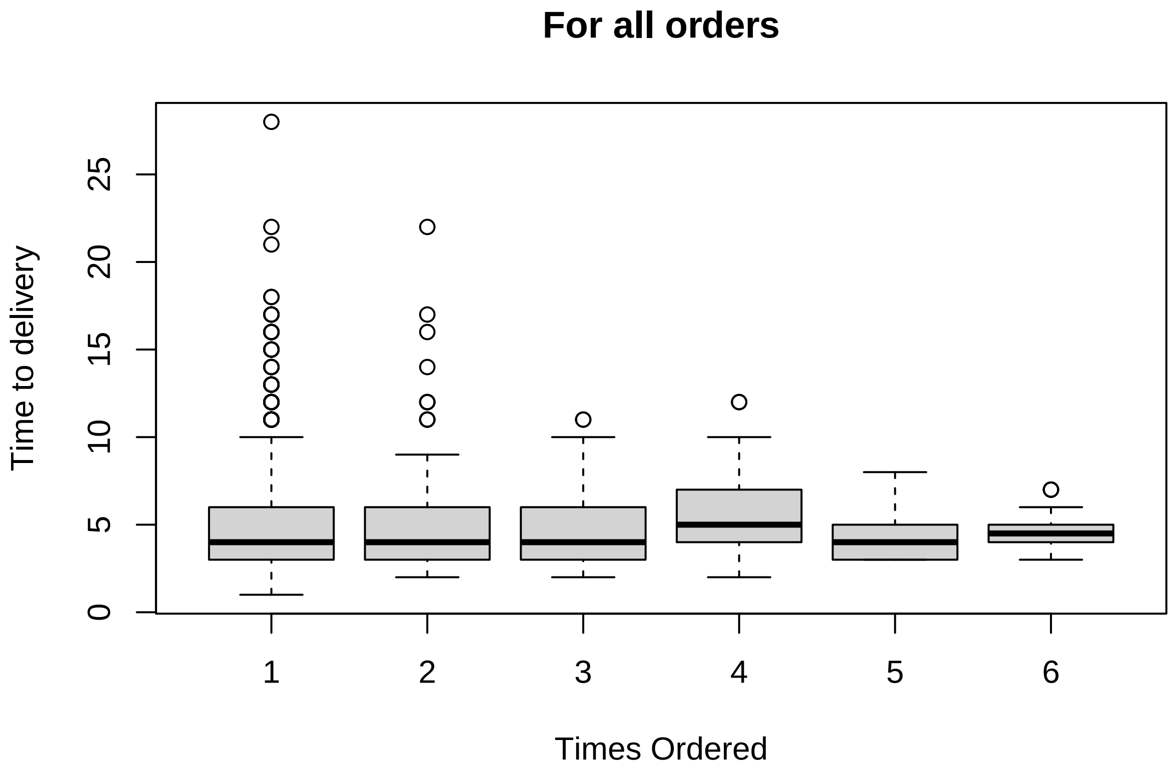
```
## # A tibble: 21 x 7
##    time_to_delivery minimum    Q1  mean median    Q3 maximum
##               <int>   <int> <dbl> <dbl>  <dbl> <dbl>   <int>
##  1                1       1     1  1          1     1       1
##  2                2       1     1  1.12       1     1       3
##  3                3       1     1  1.11       1     1       5
##  4                4       1     1  1.11       1     1       6
##  5                5       1     1  1.11       1     1       6
##  6                6       1     1  1.11       1     1       6
##  7                7       1     1  1.12       1     1       5
##  8                8       1     1  1.10       1     1       4
##  9                9       1     1  1.12       1     1       3
## 10               10       1     1  1.05       1     1       3
## # i 11 more rows
```

```
# trim <- function(x){
#   x[(x > mean(x)-1.5*IQR(x)) & (x < mean(x)+1.5*IQR(x))]
# }
#
# trimmedorders <- orders %>%
#   mutate(orders$time_to_delivery = trim(orders$time_to_delivery)) %>%
#

#Histogram of delivery time for all orders

hist <- ggplot(orders, aes(x=time_to_delivery)) +
  geom_histogram(binwidth = 1, color = "black", fill = "White") +
  stat_bin(binwidth = 1, geom = 'text', aes(label = ..count..), color = "red",
           position = position_stack(vjust = 1.05)) +
  scale_y_continuous(breaks = seq(0, 3000, 250)) +
  scale_x_continuous(breaks = seq(1, 30, 1)) +
  labs(y = "Count", x = "Delivery time(days)") +
  ggtitle("Histogram of Delivery time for all Orders (Total: 9462)")

hist
```

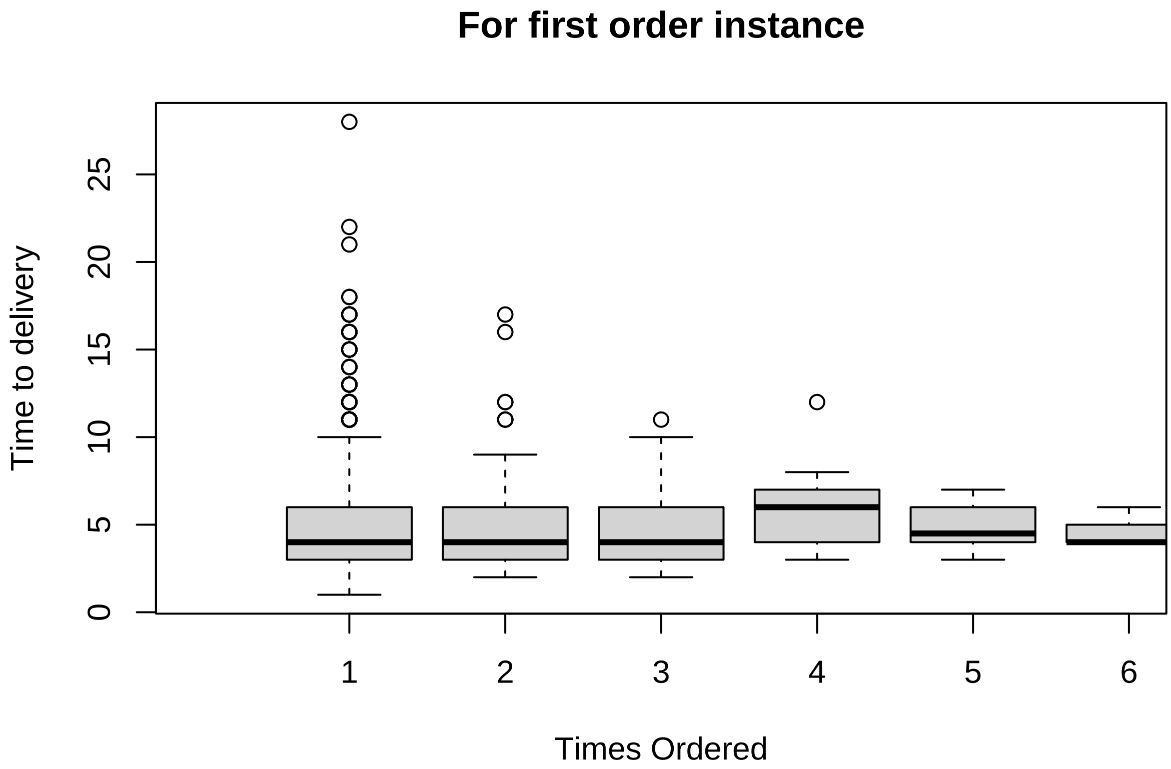
```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(count)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



```
#ggplot(orders, aes(x=time_to_delivery, fill=as.factor(times_ordered))) +
#  geom_histogram(binwidth = 1, alpha = .5, position = "identity")

#ggplot(orders, aes(x=time_to_delivery, fill=as.factor(times_ordered))) +
#  geom_density(alpha = .3)

boxplot(orders$time_to_delivery ~ orders$times_ordered,
        main = "For all orders",
        xlab = "Times Ordered",
        ylab = "Time to delivery")
```



```r
boxplot(firstorders$time_to_delivery ~ firstorders$times_ordered,
        main = "For first order instance",
        xlab = "Times Ordered",
        ylab = "Time to delivery",
        xlim = c(0,6))
```



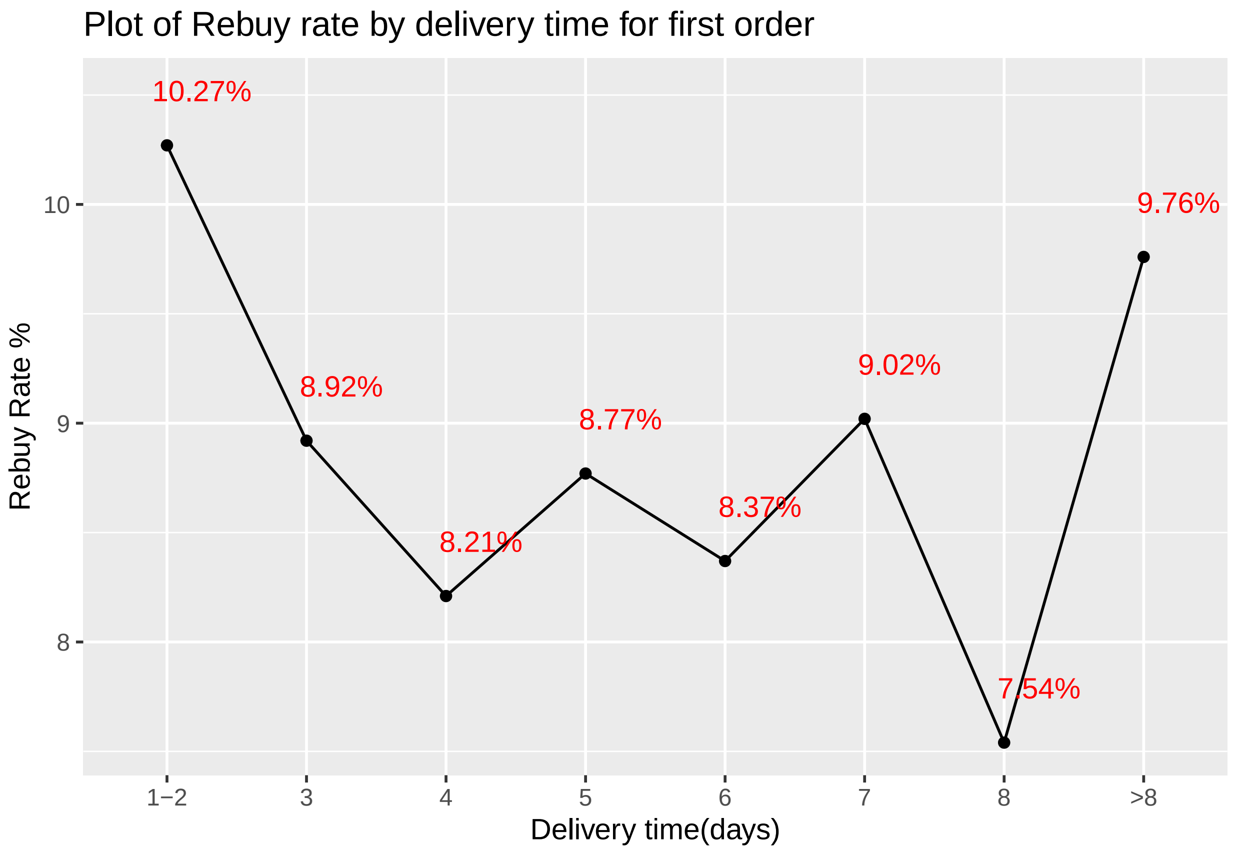
```
#Making a dataframe for each cohort

#1 to 2 days
one_to_two_days <- firstorders %>%
  filter(time_to_delivery <= 2)

#Table of rebuy rate for delivery time of 1-2 days
rebuy_one_to_two <- one_to_two_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)

rr_1to2 <- 100 - rebuy_one_to_two$percent[1]

#3 days
three_days <- firstorders %>%
  filter(time_to_delivery == 3)

#
rebuy_3 <- three_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)
```

```
rr_3 <- 100 - rebuy_3$percent[1]

#4 days
four_days <- firstorders %>%
  filter(time_to_delivery == 4)

#
rebuy_4 <- four_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)

rr_4 <- 100 - rebuy_4$percent[1]

#5 days
five_days <- firstorders %>%
  filter(time_to_delivery == 5)

#
rebuy_5 <- five_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)

rr_5 <- 100 - rebuy_5$percent[1]

#6 days
six_days <- firstorders %>%
  filter(time_to_delivery == 6)

#
rebuy_6 <- six_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)

rr_6 <- 100 - rebuy_6$percent[1]

#7 days
seven_days <- firstorders %>%
  filter(time_to_delivery == 7)

#
rebuy_7 <- seven_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)

rr_7 <- 100 - rebuy_7$percent[1]
```

```r
#8 days
eight_days <- firstorders %>%
  filter(time_to_delivery == 8)

#
rebuy_8 <- eight_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)

rr_8 <- 100 - rebuy_8$percent[1]

#9 days or more
nine_days <- firstorders %>%
  filter(time_to_delivery >= 9)

#
rebuy_9_or_more <- nine_days %>%
  group_by(times_ordered) %>%
  summarise(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(percent = (n/sum(n)) * 100)

rr_9_or_more <- 100 - rebuy_9_or_more$percent[1]

#Rounding and putting into dataframe
rr_all <- c(rr_1to2, rr_3, rr_4, rr_5, rr_6, rr_7, rr_8,
            rr_9_or_more)

rr_all <- round(rr_all, digits = 2)

rebuyrates <- data.frame(Days_to_delivery = c('1-2', '3', '4', '5', '6',
                         '7', '8', '>8' ),
                         Rebuy_rates = rr_all )

rebuyrates
```

```
##   Days_to_delivery Rebuy_rates
## 1              1-2       10.27
## 2                3        8.92
## 3                4        8.21
## 4                5        8.77
## 5                6        8.37
## 6                7        9.02
## 7                8        7.54
## 8               >8        9.76
```

```r
linegraph <- ggplot(data = rebuyrates, aes(x = factor(Days_to_delivery,
      level = c('1-2', '3', '4', '5', '6',
                '7', '8', '>8' )),
      y = Rebuy_rates, group = 1)) +
  geom_line() +
  geom_point() +
  scale_y_continuous(breaks = seq(6, 11, 1)) +
  labs(y = "Rebuy Rate %", x = "Delivery time(days)") +
  ggtitle("Plot of Rebuy rate by delivery time for first order") +
  geom_text(aes(label = paste0(Rebuy_rates, "%")), color = "red",
            nudge_x = .25, nudge_y = .25)

linegraph
```


Plot of Rebuy rate by delivery time for first order
Rebuy Rate %
>8
Delivery time(days)