DA Assignment

Prerequisites

Import packages.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.1
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.1
```

```
## Warning: package 'tibble' was built under R version 4.2.1
```

```
## Warning: package 'tidyr' was built under R version 4.2.1
```

```
## Warning: package 'readr' was built under R version 4.2.1
```

```
## Warning: package 'purrr' was built under R version 4.2.1
```

```
## Warning: package 'dplyr' was built under R version 4.2.1
```

```
## Warning: package 'stringr' was built under R version 4.2.1
```

```
## Warning: package 'forcats' was built under R version 4.2.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.2.1
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.1
```

```
library(ggplot2)
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.2.1
```

```
setwd('C:/Personal/Data Analyst Job - Assingment/')
```

Load Data.

```
df <- read_excel("Dataset_DA_2022.xlsx")

glimpse(df)
```

```
## Rows: 8,368
## Columns: 4
## $ CUID         <chr> "CUIDADED1135A300.0A1", "CUIDA14669E9317B.0A2", "CUIDDC3E~
## $ subscription <chr> "Plus-yearly", "Plus-monthly", "Plus-yearly", "Plus-yearl~
## $ started_at   <chr> "28/06/2021", "25/06/2021", "17/06/2021", "16/06/2021", "~
## $ cancelled_at <chr> "30/06/2021", "26/06/2021", "23/06/2021", "23/06/2021", "~
```

Data Cleaning

Both 'started_at' and 'cancelled_at' fields types are 'char'

```
# Convert 'started_at' and 'cancelled_at' fields types to date
df <- mutate(df,
             started_at = as_date(started_at, format = '%d/%m/%Y'),
             cancelled_at = as_date(cancelled_at, format = '%d/%m/%Y'))

glimpse(df)
```

```
## Rows: 8,368
## Columns: 4
## $ CUID         <chr> "CUIDADED1135A300.0A1", "CUIDA14669E9317B.0A2", "CUIDDC3E~
## $ subscription <chr> "Plus-yearly", "Plus-monthly", "Plus-yearly", "Plus-yearl~
## $ started_at   <date> 2021-06-28, 2021-06-25, 2021-06-17, 2021-06-16, 2021-06-~
## $ cancelled_at <date> 2021-06-30, 2021-06-26, 2021-06-23, 2021-06-23, 2021-06-~
```

Issue: It seems date field records are consisting with different formats.

```
# Convert the excel file to a csv and load data from csv
df <- read_csv("Dataset_DA_2022.csv")
```

```
## Rows: 1028619 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): CUID, subscription, started_at, cancelled_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Convert 'started_at' and 'cancelled_at' fields types to date
df <- mutate(df,
             started_at = as_date(started_at, format = '%d/%m/%Y'),
             cancelled_at = as_date(cancelled_at, format = '%d/%m/%Y'))

glimpse(df)
```

```
## Rows: 1,028,619
## Columns: 4
## $ CUID         <chr> "CUIDADED1135A300.0A1", "CUIDA14669E9317B.0A2", "CUIDDC3E~
## $ subscription <chr> "Plus-yearly", "Plus-monthly", "Plus-yearly", "Plus-yearl~
## $ started_at   <date> 2021-06-28, 2021-06-25, 2021-06-17, 2021-06-16, 2021-06-~
## $ cancelled_at <date> 2021-06-30, 2021-06-26, 2021-06-23, 2021-06-23, 2021-06-~
```

```
#tail(df, 100)
```

Check for null values

```
dim(df)
```

```
## [1] 1028619       4
```

```
df %>% filter(is.na(CUID))
```

```
## # A tibble: 1,020,251 x 4
##    CUID  subscription started_at cancelled_at
##    <chr> <chr>        <date>     <date>
##  1 <NA>  <NA>         NA         NA
##  2 <NA>  <NA>         NA         NA
##  3 <NA>  <NA>         NA         NA
##  4 <NA>  <NA>         NA         NA
##  5 <NA>  <NA>         NA         NA
##  6 <NA>  <NA>         NA         NA
##  7 <NA>  <NA>         NA         NA
##  8 <NA>  <NA>         NA         NA
##  9 <NA>  <NA>         NA         NA
## 10 <NA>  <NA>         NA         NA
## # ... with 1,020,241 more rows
```

There are 1,020,251 null records from field CUID

```
# Removing null records
df <-  df %>% filter(!is.na(CUID))


# Verify null record removal
df %>% filter(is.na(CUID)) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     0
```

```
df %>% filter(is.na(subscription)) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     0
```

```
df %>% filter(is.na(started_at)) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     0
```

```
df %>% filter(is.na(cancelled_at)) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     0
```

```
dim(df)
```

```
## [1] 8368    4
```

Null values removed 8368 records with data.

Feature Engineering

Check subscription types

```
df %>% select(subscription) %>% unique()
```

```
## # A tibble: 6 x 1
##   subscription
##   <chr>
## 1 Plus-yearly
## 2 Plus-monthly
## 3 Growth-monthly
## 4 Growth-yearly
## 5 Plus-
## 6 Growth-
```

There are 6 subscription types in the data sets.

```
df
```

```
## # A tibble: 8,368 x 4
##    CUID                  subscription started_at cancelled_at
##    <chr>                 <chr>        <date>     <date>
##  1 CUIDADED1135A300.0A1  Plus-yearly  2021-06-28 2021-06-30
##  2 CUIDA14669E9317B.0A2  Plus-monthly 2021-06-25 2021-06-26
##  3 CUIDDC3E380248BD.0A3  Plus-yearly  2021-06-17 2021-06-23
##  4 CUID5F3A2B9392F9.0A4  Plus-yearly  2021-06-16 2021-06-23
##  5 CUID46C633ACC3B3.0A5  Plus-yearly  2021-06-15 2021-06-17
##  6 CUID21714D668730.0A6  Plus-monthly 2021-06-14 2021-06-16
##  7 CUID323190C0EC8E.0A7  Plus-monthly 2021-06-10 2021-06-11
##  8 CUIDCDE07D01714B.0A8  Plus-yearly  2021-06-08 2021-06-09
##  9 CUID8881EA074EC9.0A9  Plus-monthly 2021-06-08 2021-06-10
## 10 CUID5560339C0F99.0A10 Plus-yearly  2021-06-08 2021-06-10
## # ... with 8,358 more rows
```

```
summary(df)
```

```
##      CUID           subscription         started_at
##  Length:8368        Length:8368        Min.   :2021-01-01
##  Class :character   Class :character   1st Qu.:2021-02-09
##  Mode  :character   Mode  :character   Median :2021-03-07
##                                        Mean   :2021-03-09
##                                        3rd Qu.:2021-04-07
##                                        Max.   :2021-06-28
##   cancelled_at
##  Min.   :2021-01-02
##  1st Qu.:2021-03-25
##  Median :2021-04-25
##  Mean   :2021-04-24
##  3rd Qu.:2021-05-25
##  Max.   :2021-07-07
```

Introduce new features

```
# Introduce new feilds
# usage_days (cancelled_at - started_at)
# started_month
```

```r
df <- mutate(df,
             usage_days = difftime(cancelled_at, started_at, units = "days"),
             started_month = month(ymd(started_at),
                                   label = FALSE,
                                   abbr = FALSE),
             started_month_nm = month(ymd(started_at),
                                      label = TRUE,
                                      abbr = FALSE))

df$usage_days <- as.numeric(df$usage_days, units="days")
df$usage_months <- as.numeric(ceiling(df$usage_days / 30), units = "month")

summary(df)
```

```
##      CUID           subscription         started_at
##  Length:8368        Length:8368        Min.   :2021-01-01
##  Class :character   Class :character   1st Qu.:2021-02-09
##  Mode  :character   Mode  :character   Median :2021-03-07
##                                        Mean   :2021-03-09
##                                        3rd Qu.:2021-04-07
##                                        Max.   :2021-06-28
##
##   cancelled_at          usage_days     started_month   started_month_nm
##  Min.   :2021-01-02   Min.   :  0.00   Min.   :1.000   March   :2315
##  1st Qu.:2021-03-25   1st Qu.: 30.00   1st Qu.:2.000   February:2125
##  Median :2021-04-25   Median : 31.00   Median :3.000   April   :1603
##  Mean   :2021-04-24   Mean   : 45.89   Mean   :2.771   January :1530
##  3rd Qu.:2021-05-25   3rd Qu.: 61.00   3rd Qu.:4.000   May     : 716
##  Max.   :2021-07-07   Max.   :181.00   Max.   :6.000   June    :  79
##                                                        (Other) :   0
##   usage_months
##  Min.   :0.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :2.039
##  3rd Qu.:3.000
##  Max.   :7.000
##
```

New fields: usage_days - usage period in days usage_months - usage period in months started_month - subscribed month started_month_nm - subscribed month name

```r
summary(df)
```

```
##      CUID           subscription         started_at
##  Length:8368        Length:8368        Min.   :2021-01-01
##  Class :character   Class :character   1st Qu.:2021-02-09
##  Mode  :character   Mode  :character   Median :2021-03-07
##                                        Mean   :2021-03-09
##                                        3rd Qu.:2021-04-07
##                                        Max.   :2021-06-28
##
```

```
##   cancelled_at          usage_days     started_month   started_month_nm
##  Min.   :2021-01-02   Min.   :  0.00   Min.   :1.000   March   :2315
##  1st Qu.:2021-03-25   1st Qu.: 30.00   1st Qu.:2.000   February:2125
##  Median :2021-04-25   Median : 31.00   Median :3.000   April   :1603
##  Mean   :2021-04-24   Mean   : 45.89   Mean   :2.771   January :1530
##  3rd Qu.:2021-05-25   3rd Qu.: 61.00   3rd Qu.:4.000   May     : 716
##  Max.   :2021-07-07   Max.   :181.00   Max.   :6.000   June    :  79
##                                                        (Other) :   0
##   usage_months
##  Min.   :0.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :2.039
##  3rd Qu.:3.000
##  Max.   :7.000
##
```

Detailed Analysis

1) Sales

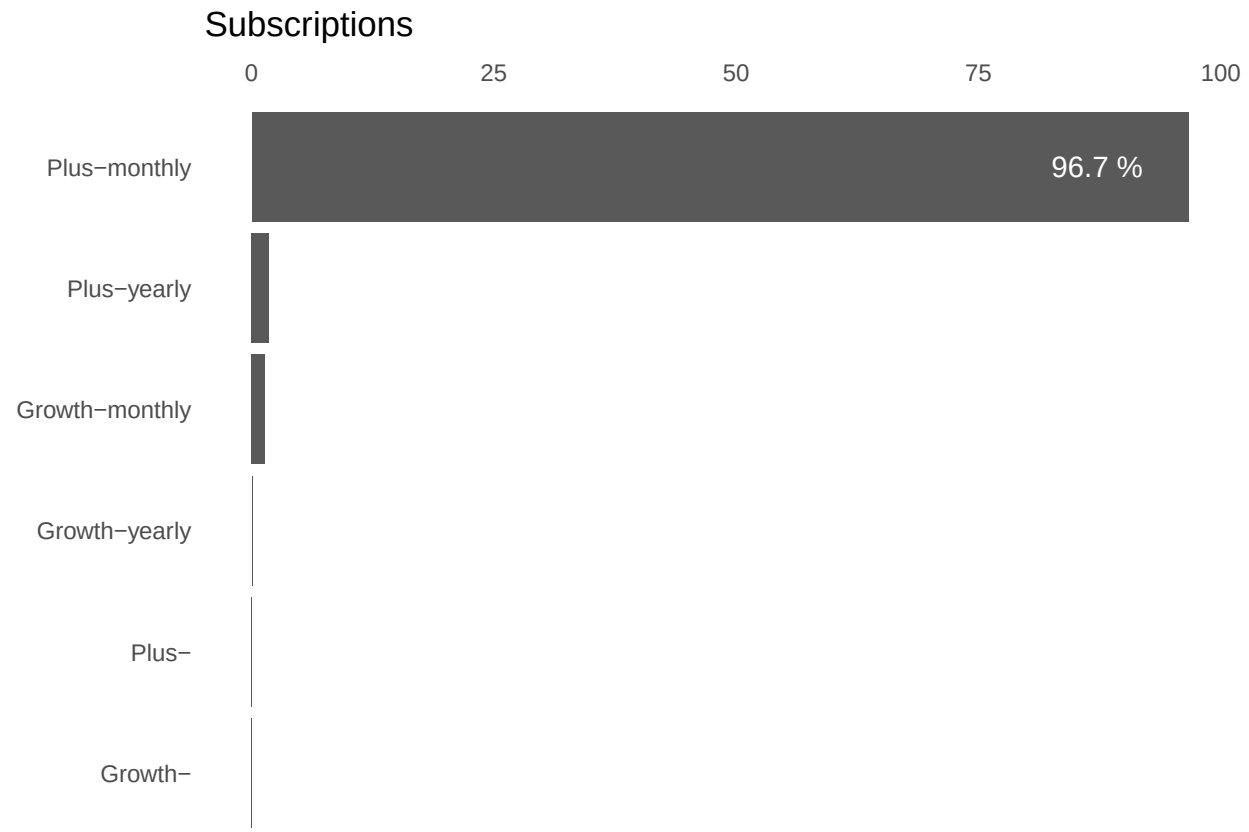
```
subscriptions <- df %>% group_by(subscription) %>%
  count(name = 'No_of_subscriptions')
subscriptions
```

```
## # A tibble: 6 x 2
## # Groups:   subscription [6]
##   subscription   No_of_subscriptions
##   <chr>                        <int>
## 1 Growth-                          1
## 2 Growth-monthly                 115
## 3 Growth-yearly                    8
## 4 Plus-                            5
## 5 Plus-monthly                  8089
## 6 Plus-yearly                    150
```

```
total_subs <- as.numeric(count(df))

subscriptions = subscriptions %>% mutate(
  subscribed_percentage = No_of_subscriptions / total_subs*100)

ggplot(subscriptions, aes(y = reorder(subscription, No_of_subscriptions),
                          x = subscribed_percentage))+
  geom_bar(stat = "summary")+
  scale_x_continuous(position = "top") +
  geom_text(aes(label = paste0(round(subscribed_percentage,1), " %")), colour = "white", hjust = 1.5)+
  scale_fill_manual(name = "Reviews", values=c("#F7C815","grey50")) +
  labs(title = "Subscriptions",
       x = NULL,
       y = NULL) +
```

```
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
    plot.caption = element_text(face = 'italic'),
    panel.grid.major = element_line('white', size = 0.5),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank(),
    panel.ontop = FALSE
  )
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
#ggsave("1_overall_subcriptions.jpg", width = 10, height = 8, units = "cm")
```

It is a clear highlight that, 'Plus-monthly' is the most commonly subscribed package.

```
df %>% group_by(subscription) %>% count()
```

```
## # A tibble: 6 x 2
## # Groups:   subscription [6]
##   subscription       n
##   <chr>          <int>
## 1 Growth-            1
```

```
## 2 Growth-monthly   115
## 3 Growth-yearly      8
## 4 Plus-              5
## 5 Plus-monthly    8089
## 6 Plus-yearly      150
```

This shows that only 'Plus-monthly', 'Growth-monthly' and 'Plus-yearly' subscription types have a minimal of 100 subscriptions. Other subscription types not even exceed 10 subscriptions.

Overall subscriptions by month

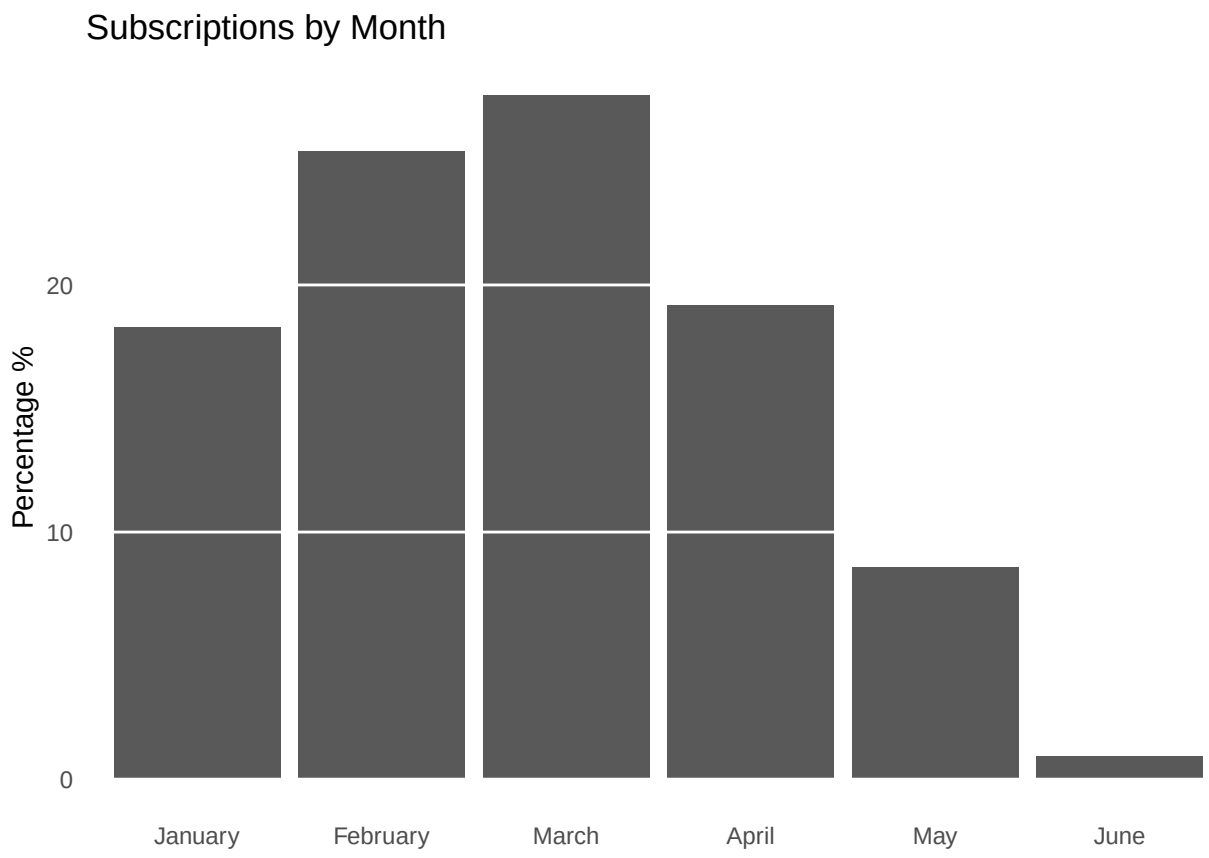
```
monthly_subscriptions3 <- df %>%
  group_by(started_month_nm) %>%
  count() %>%
  arrange(started_month_nm)

total_subs <- as.numeric(count(df))

monthly_subscriptions3 = monthly_subscriptions3 %>% mutate(
  subscribed_percentage = n / total_subs*100)


monthly_subscriptions3
```

```
## # A tibble: 6 x 3
## # Groups:   started_month_nm [6]
##   started_month_nm     n subscribed_percentage
##   <ord>            <int>                 <dbl>
## 1 January           1530                18.3
## 2 February          2125                25.4
## 3 March             2315                27.7
## 4 April             1603                19.2
## 5 May                716                 8.56
## 6 June                79                 0.944
```

```
#sum(monthly_subscriptions3$subscribed_percentage)
```

```
ggplot(monthly_subscriptions3, aes(y = subscribed_percentage,
                      x = started_month_nm))+
  geom_bar(stat = "summary")+
  labs(title = "Subscriptions by Month",
       x = NULL,
       y = 'Percentage %') +
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
    plot.caption = element_text(face = 'italic'),
    panel.grid.major = element_line('white', size = 0.5),
    panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.ontop = TRUE
  )
```

```
## No summary function supplied, defaulting to 'mean_se()'
```




```
#ggsave("2_subscriptions_by_month.jpg", width = 16, height = 12, units = "cm")
```

Overall it indicate a good subscription sales from January to April. Then there is a considerable drop of sales.

Subscriptions by Subscription Types

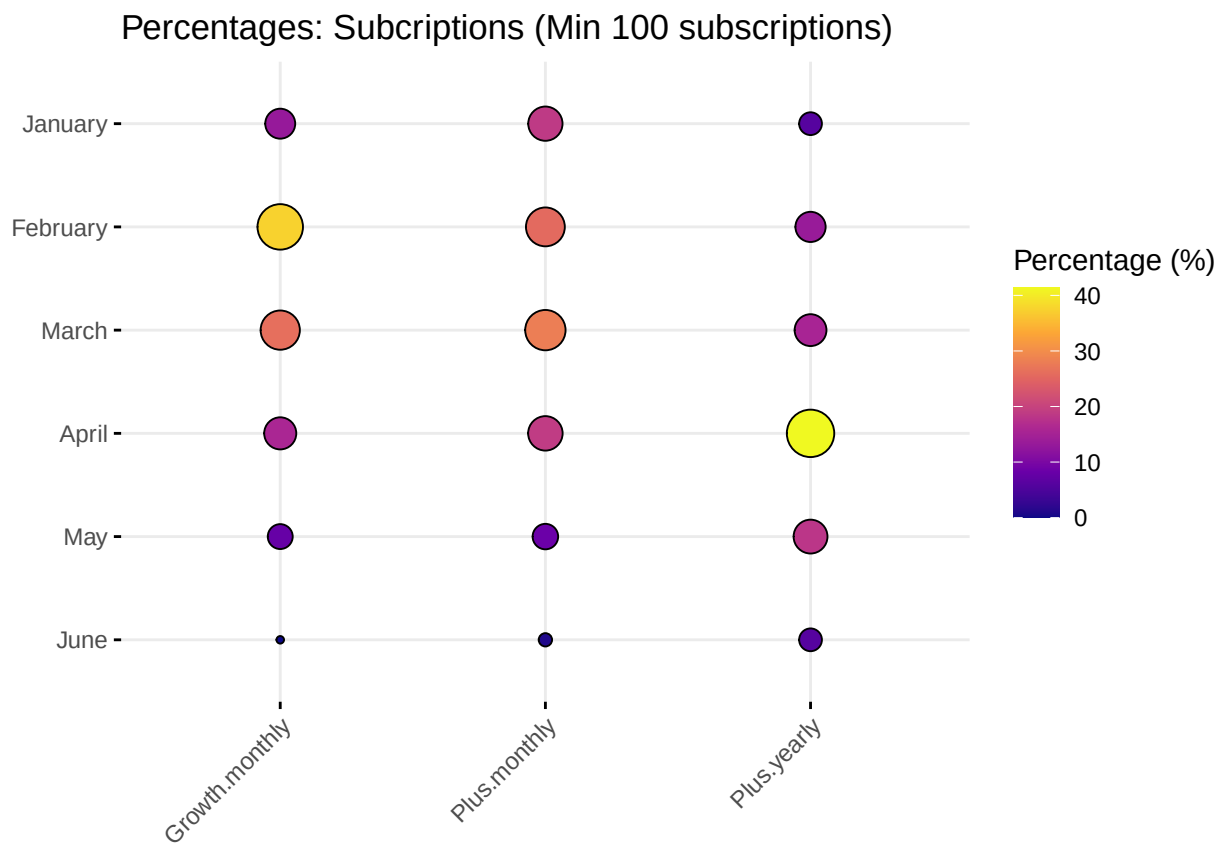
```
monthly_subscriptions <- df %>% filter((subscription == 'Growth-monthly')|
                                       (subscription == 'Plus-monthly')|
                                       (subscription == 'Plus-yearly')) %>%
  group_by(started_month_nm, subscription) %>%
  count(subscription, name = "n") %>%
  pivot_wider(names_from = subscription, values_from = n, values_fill = 0) %>%
  ungroup()

monthly_subscriptions <-data.frame(monthly_subscriptions, row.names = 1)

growth_m_sum <- sum(monthly_subscriptions$Growth.monthly)
plus_m_sum <- sum(monthly_subscriptions$Plus.monthly)
plus_y_sum <- sum(monthly_subscriptions$Plus.yearly)
```

```
monthly_subscriptions$Growth.monthly <- monthly_subscriptions$Growth.monthly / growth_m_sum * 100
monthly_subscriptions$Plus.monthly <- monthly_subscriptions$Plus.monthly / plus_m_sum * 100
monthly_subscriptions$Plus.yearly <- monthly_subscriptions$Plus.yearly / plus_y_sum * 100


head(monthly_subscriptions)
```

```
##          Growth.monthly Plus.monthly Plus.yearly
## January        13.043478   18.5437013     6.00000
## February       37.391304   25.4790456    13.33333
## March          26.086957   27.9268142    15.33333
## April          15.652174   18.8156756    41.33333
## May             7.826087    8.3693905    18.00000
## June            0.000000    0.8653727     6.00000
```

```
ggballoonplot(monthly_subscriptions, fill = "value", size.range = c(1, 8)) +
  scale_fill_viridis_c(option = "C")+
  guides(size = FALSE) +
  labs(title = "Percentages: Subcriptions (Min 100 subscriptions)",
       fill = 'Percentage (%)',
       x = NULL,
       y = NULL)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



This shows the percentages of sales per each subscription type (top 3). But this is not the best graphical representation.

Another approach.

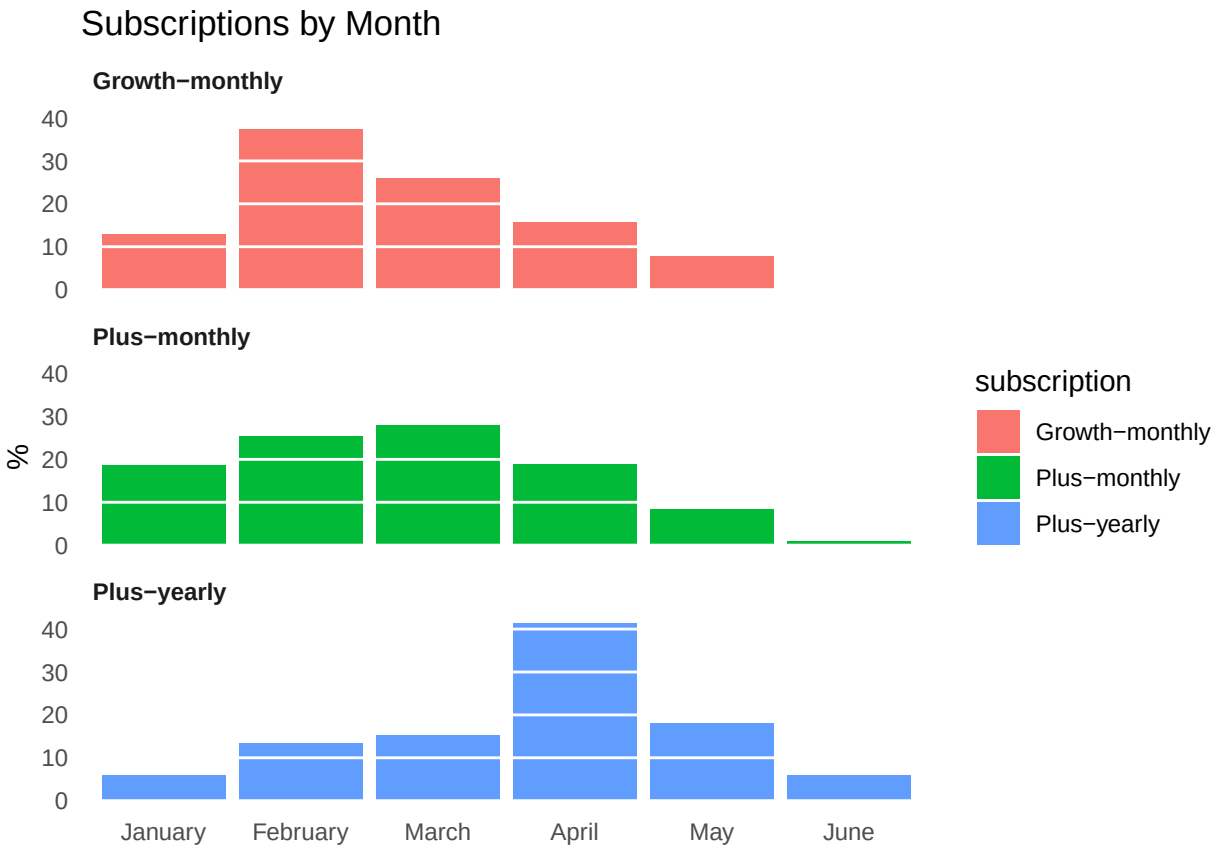
```
monthly_subscriptions2 <- df %>%
  filter((subscription == 'Growth-monthly')| (subscription == 'Plus-monthly')| (subscription == 'Plus-y
  group_by(subscription, started_month_nm) %>%
  count() %>%
  arrange(subscription, started_month_nm)

monthly_subscriptions2 = monthly_subscriptions2 %>% mutate(
  subscribed_percentage = case_when(subscription == "Growth-monthly"~ n/growth_m_sum*100,
                                    subscription == "Plus-monthly"~ n/ plus_m_sum * 100,
                                    subscription == "Plus-yearly"~ n/plus_y_sum * 100)
 )


monthly_subscriptions2
```

```
## # A tibble: 17 x 4
## # Groups:   subscription, started_month_nm [17]
##    subscription   started_month_nm     n subscribed_percentage
##    <chr>          <ord>            <int>                 <dbl>
##  1 Growth-monthly January             15                13.0
##  2 Growth-monthly February            43                37.4
##  3 Growth-monthly March               30                26.1
##  4 Growth-monthly April               18                15.7
##  5 Growth-monthly May                  9                 7.83
##  6 Plus-monthly   January           1500                18.5
##  7 Plus-monthly   February          2061                25.5
##  8 Plus-monthly   March             2259                27.9
##  9 Plus-monthly   April             1522                18.8
## 10 Plus-monthly   May                677                 8.37
## 11 Plus-monthly   June                70                 0.865
## 12 Plus-yearly    January              9                 6
## 13 Plus-yearly    February            20                13.3
## 14 Plus-yearly    March               23                15.3
## 15 Plus-yearly    April               62                41.3
## 16 Plus-yearly    May                 27                18
## 17 Plus-yearly    June                 9                 6
```

```
ggplot(monthly_subscriptions2, aes(y = subscribed_percentage,
                      x = started_month_nm, fill = subscription))+
  geom_bar(stat = "summary")+
  facet_wrap(~ subscription, ncol= 1)+
  labs(title = "Subscriptions by Month",
       x = NULL,
       y = '%') +
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
    plot.caption = element_text(face = 'italic'),
    panel.grid.major = element_line('white', size = 0.5),
```

```
    panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.ontop = TRUE
  )
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



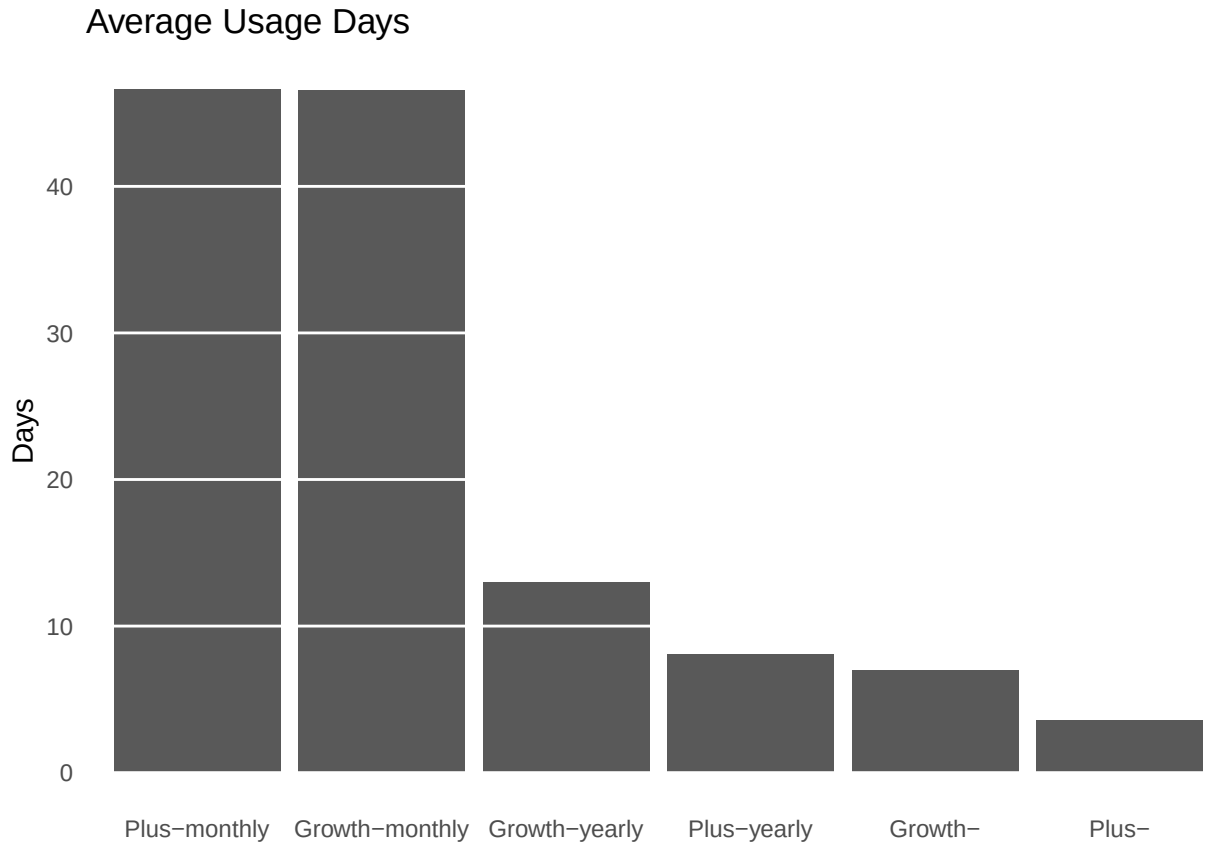

```
#ggsave("3_subscription_sales_by_subscription_type.jpg", width = 16, height = 16, units = "cm")
```

2) Subscription cancellations

Average subscription usage period of each subscription type.

```
usage <- df %>% group_by(subscription) %>% summarise(average_usage_days = mean(usage_days))
#usage

ggplot(usage, aes(x = reorder(subscription, -average_usage_days), y = average_usage_days)) +
  geom_bar(stat = "summary", fun = "mean") +
  labs(title = "Average Usage Days",
       x = NULL,
```

```
      y = "Days")+
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
    plot.caption = element_text(face = 'italic'),
    panel.grid.major = element_line('white', size = 0.5),
    panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.ontop = TRUE
  )
```



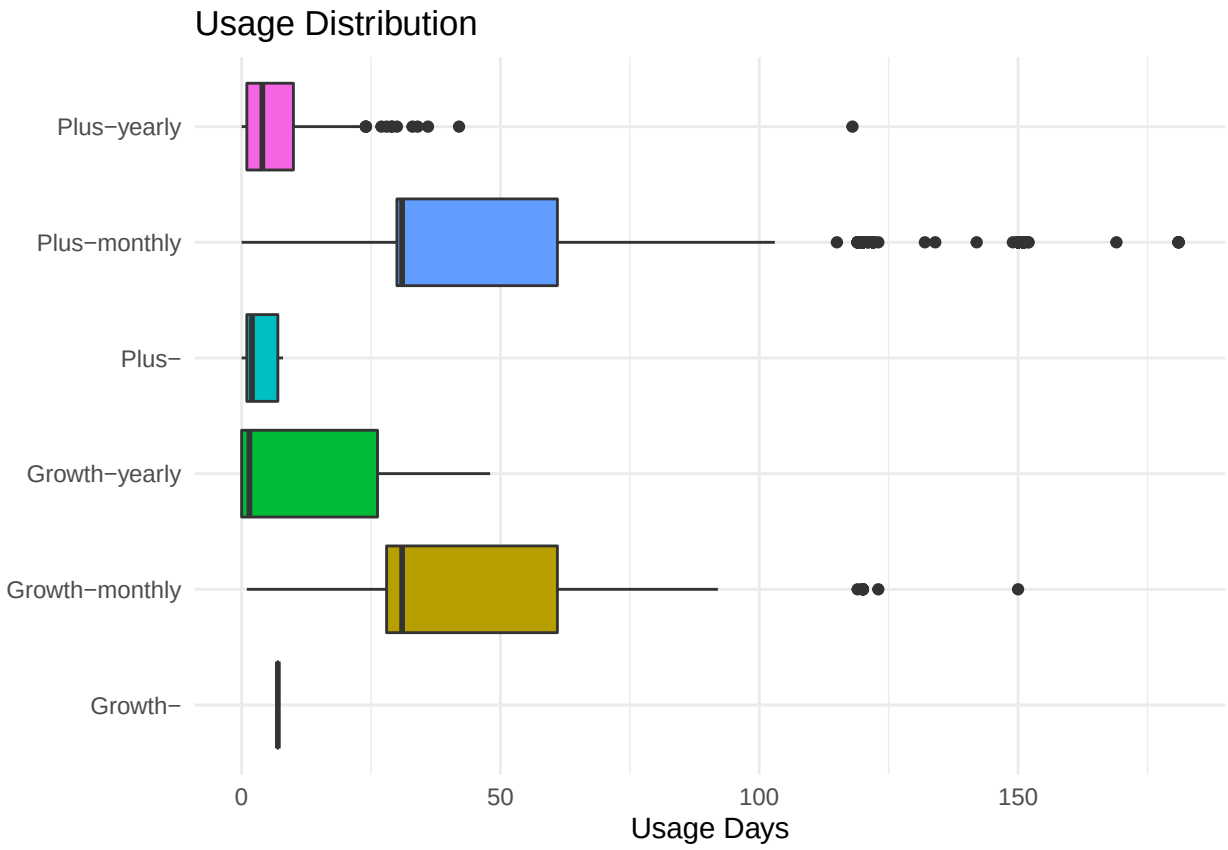

```
#ggsave("4_", width = 16, height = 16, units = "cm")
```

This indicate that only 'Plus-monthly' and 'Growth-monthly' subscribers use the service comparatively longer than other subscription types. But still the average usage duration is less than 50 days.

Spread of subscription usage period by each subscription type.

```
ggplot(df) +
  geom_boxplot(aes(x = usage_days, y = subscription, fill = subscription))+
  #scale_x_continuous(limits = c(0, 5000)) +
  labs(title = "Usage Distribution",
       x = "Usage Days",
```

```
      y = NULL) +
  theme_minimal() +
  theme(legend.position="none")
```




```
#ggsave("5_subscription_usage_distribution.jpg", width = 16, height = 12, units = "cm")
```

This indicate that 'Plus-monthly', 'Growth-monthly' and 'Plus-yearly' subscription types shows wider usage spread. It would be easy to understand view subscription cancellation percentages by usage months for each subscription type.

Subscription cancellation percenages by period of months

Filter subscription types with minimum 100 subscribers.

```
# Alter data
min_usage <- df %>% filter((subscription == 'Growth-monthly')|
                                      (subscription == 'Plus-monthly')|
                                      (subscription == 'Plus-yearly')) %>%
  group_by(subscription, usage_months) %>%
  count() %>%
  arrange(subscription, usage_months)

min_usage = min_usage %>% mutate(
  cancel_percentage = case_when(subscription == "Growth-monthly"~ n/growth_m_sum*100,
```

```
                                subscription == "Plus-monthly"~ n/ plus_m_sum * 100,
                                subscription == "Plus-yearly"~ n/plus_y_sum * 100)
 )

min_usage = min_usage %>% mutate(
  cancellation_type = case_when(usage_months == 0~ "Same Day",
                                usage_months <= 1~ "Within 1 Month",
                                usage_months <= 2~ "Within 2 Months",
                                usage_months <= 3~ "Within 3 Months",
                                usage_months <= 4~ "Within 4 Months",
                                usage_months <= 5~ "Within 5 Months",
                                usage_months <= 6~ "Within 6 Months",
                                usage_months <= 7~ "Within 7 Months")
 )

# Verify percentages

test <- min_usage %>% filter(subscription == "Growth-monthly")
test
```

```
## # A tibble: 5 x 5
## # Groups:   subscription, usage_months [5]
##   subscription   usage_months     n cancel_percentage cancellation_type
##   <chr>                 <dbl> <int>             <dbl> <chr>
## 1 Growth-monthly            1    43             37.4  Within 1 Month
## 2 Growth-monthly            2    41             35.7  Within 2 Months
## 3 Growth-monthly            3    19             16.5  Within 3 Months
## 4 Growth-monthly            4    10              8.70 Within 4 Months
## 5 Growth-monthly            5     2              1.74 Within 5 Months
```

```
sum(test$cancel_percentage)
```

```
## [1] 100
```

```
test <- min_usage %>% filter(subscription == "Plus-monthly")
test
```

```
## # A tibble: 8 x 5
## # Groups:   subscription, usage_months [8]
##   subscription usage_months     n cancel_percentage cancellation_type
##   <chr>               <dbl> <int>             <dbl> <chr>
## 1 Plus-monthly            0    57            0.705  Same Day
## 2 Plus-monthly            1  2400           29.7    Within 1 Month
## 3 Plus-monthly            2  3396           42.0    Within 2 Months
## 4 Plus-monthly            3  1620           20.0    Within 3 Months
## 5 Plus-monthly            4   514            6.35   Within 4 Months
## 6 Plus-monthly            5    40            0.494  Within 5 Months
## 7 Plus-monthly            6    57            0.705  Within 6 Months
## 8 Plus-monthly            7     5            0.0618 Within 7 Months
```

```
sum(test$cancel_percentage)
```

```
## [1] 100
```

```
test <- min_usage %>% filter(subscription == "Plus-yearly")
test
```

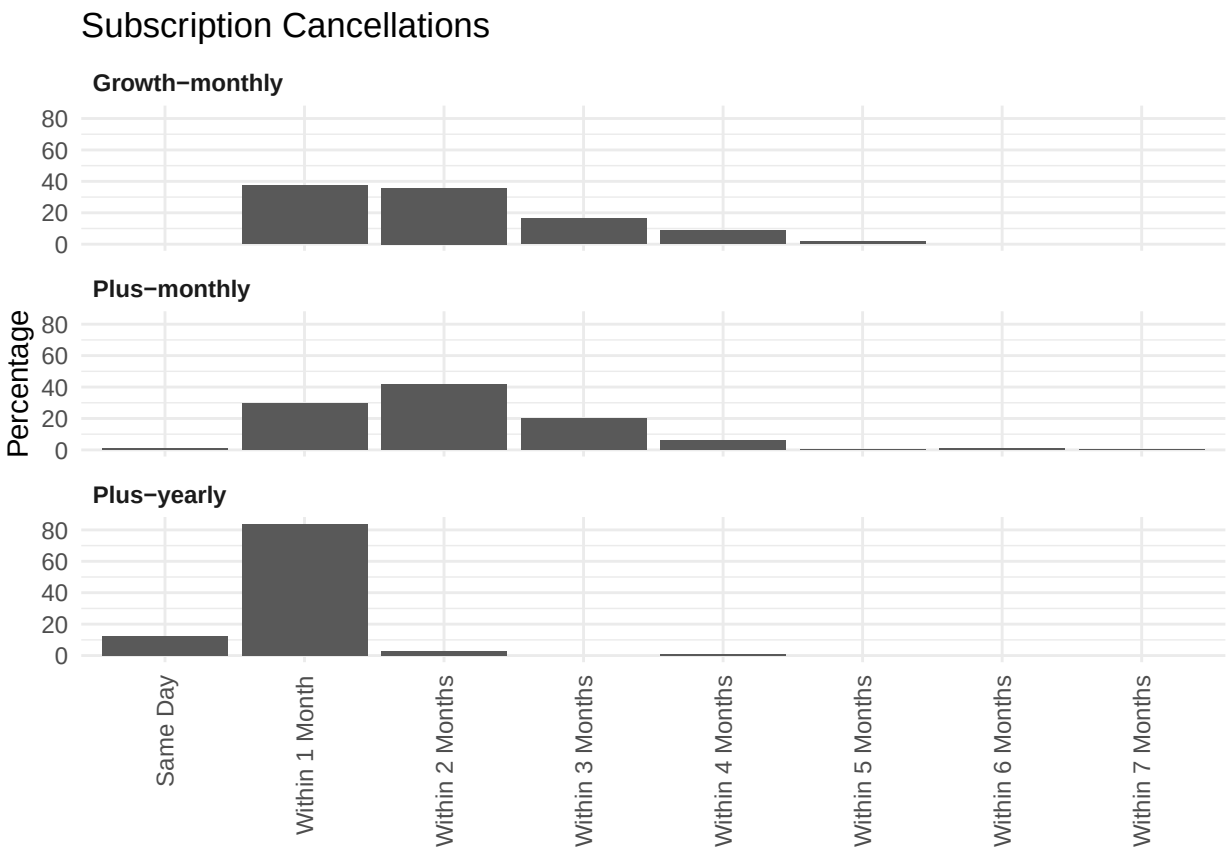
```
## # A tibble: 4 x 5
## # Groups:   subscription, usage_months [4]
##   subscription usage_months     n cancel_percentage cancellation_type
##   <chr>               <dbl> <int>             <dbl> <chr>
## 1 Plus-yearly             0    19            12.7   Same Day
## 2 Plus-yearly             1   126            84     Within 1 Month
## 3 Plus-yearly             2     4             2.67  Within 2 Months
## 4 Plus-yearly             4     1             0.667 Within 4 Months
```

```
sum(test$cancel_percentage)
```

```
## [1] 100
```

```
ggplot(min_usage, aes(y = cancel_percentage,
                      x = cancellation_type))+
  geom_bar(stat = "summary")+
  facet_wrap(~ subscription, ncol= 1)+
  labs(title = "Subscription Cancellations",
       x = NULL,
       y = 'Percentage') +
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  )
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



This shows subscription cancellation percentages for each subscription type by usage period. But this is not the perfect view to present inside.

Note: There are considerable percentage of same day cancellations for "Plus-yearly" subscription type.

I want to show how sooner the business looses their subscribers. It would be more useful to use cumulative subscription cancellation percentage rather than individual percentages by usage period.

Cumulative subscription cancellation percentages by period of months

Overall subscription cancellations

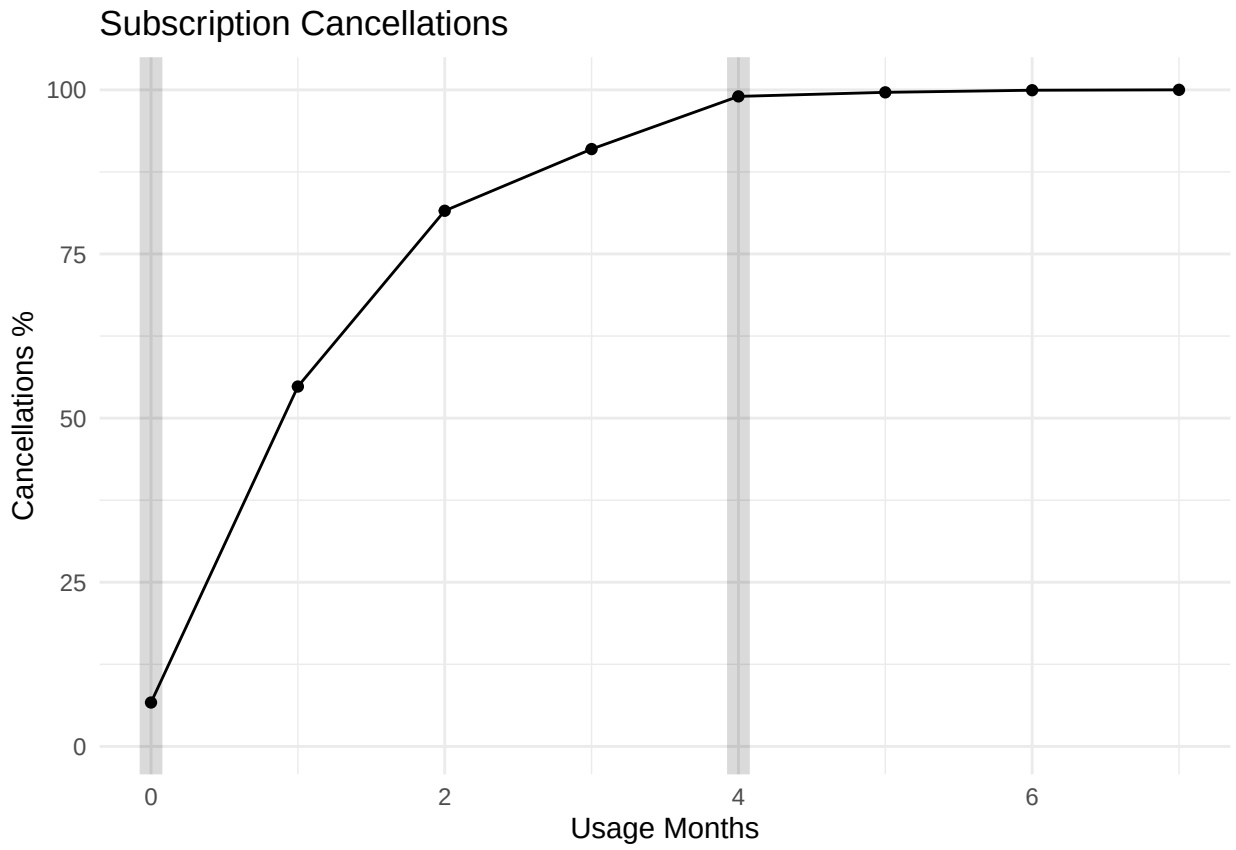
```
min_usage2 <- df %>% filter((subscription == 'Growth-monthly')|
                            (subscription == 'Plus-monthly')|
                            (subscription == 'Plus-yearly')) %>%
  group_by(subscription, usage_months) %>%
  summarise(proportion = n()) %>%
  mutate(Perc = cumsum(100*proportion/sum(proportion))) %>%
  select(-proportion)
```

```
## `summarise()` has grouped output by 'subscription'. You can override using the
## `.groups` argument.
```

```
ggplot(min_usage2, aes(y = Perc,
                       x = usage_months))+
  geom_point(stat = "summary") +
```

```
  geom_line(stat = "summary") +
  geom_vline(xintercept = 4, size = 4, alpha = 0.15) +
  geom_vline(xintercept = 0, size = 4, alpha = 0.15) +
  labs(title = "Subscription Cancellations",
       x = 'Usage Months',
       y = 'Cancellations %') +
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
  )
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```




```
ggsave("6_overall_subscription_cancellations.jpg", width = 16, height = 12, units = "cm")
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```

This shows that, Overall the business looses almost 100% of the subscribers after 4 months There is about 6-8% same day cancellation subscribers.

Cumulative subscription cancellation percentages by period of months per each subscription type

```
min_usage3 <- df %>% filter((subscription == 'Growth-monthly')|
                            (subscription == 'Plus-monthly')|
                            (subscription == 'Plus-yearly')) %>%
  group_by(subscription, usage_months) %>%
  summarise(proportion = n()) %>%
  mutate(Perc = cumsum(100*proportion/sum(proportion))) %>%
  select(-proportion)
```

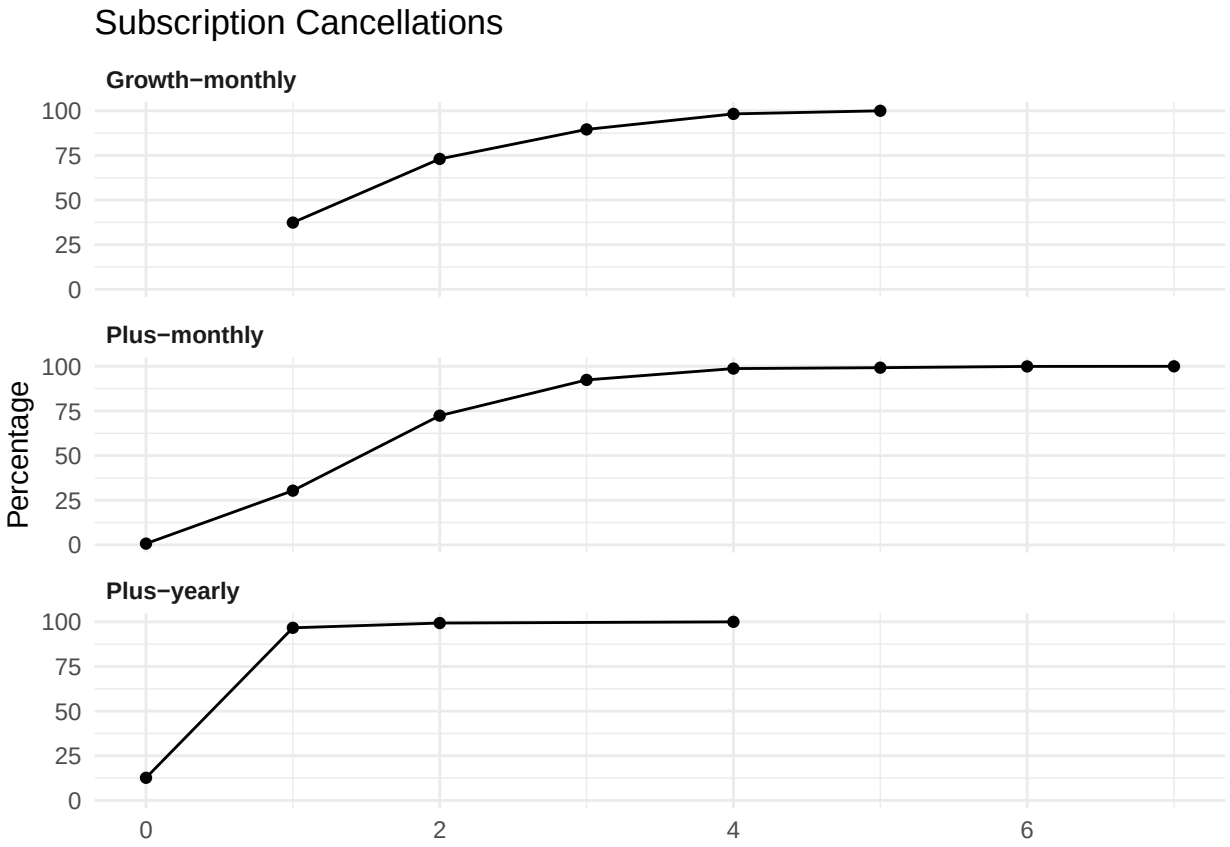
```
## `summarise()` has grouped output by 'subscription'. You can override using the
## `.groups` argument.
```

```
min_usage3
```

```
## # A tibble: 17 x 3
## # Groups:   subscription [3]
##    subscription   usage_months    Perc
##    <chr>                 <dbl>   <dbl>
##  1 Growth-monthly            1  37.4
##  2 Growth-monthly            2  73.0
##  3 Growth-monthly            3  89.6
##  4 Growth-monthly            4  98.3
##  5 Growth-monthly            5 100
##  6 Plus-monthly              0   0.705
##  7 Plus-monthly              1  30.4
##  8 Plus-monthly              2  72.4
##  9 Plus-monthly              3  92.4
## 10 Plus-monthly              4  98.7
## 11 Plus-monthly              5  99.2
## 12 Plus-monthly              6  99.9
## 13 Plus-monthly              7 100
## 14 Plus-yearly               0  12.7
## 15 Plus-yearly               1  96.7
## 16 Plus-yearly               2  99.3
## 17 Plus-yearly               4 100
```

```
ggplot(min_usage3, aes(y = Perc,
                       x = usage_months))+
  geom_point(stat = "summary")+
  geom_line(stat = "summary")+
  facet_wrap(~ subscription, ncol= 1)+
  labs(title = "Subscription Cancellations",
       x = NULL,
       y = 'Percentage') +
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
  )
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



This shows the cumulative cancellation percentages by each subscription type. It is noticeable that 'Plus-yearly' package losses its subscribers more sooner than other two types.

But still this does not show gives the seriousness of the issue. It would be better to look at things in a different angle. It will give a strong message if I show how long it will take to loose a certain percentage of subscribers.

Time duration to loose 90% of subscribers per each subscription type

```
loss_3_months <- min_usage2 %>% filter(usage_months <= 3)  %>% slice_max(Perc)
loss_3_months
```

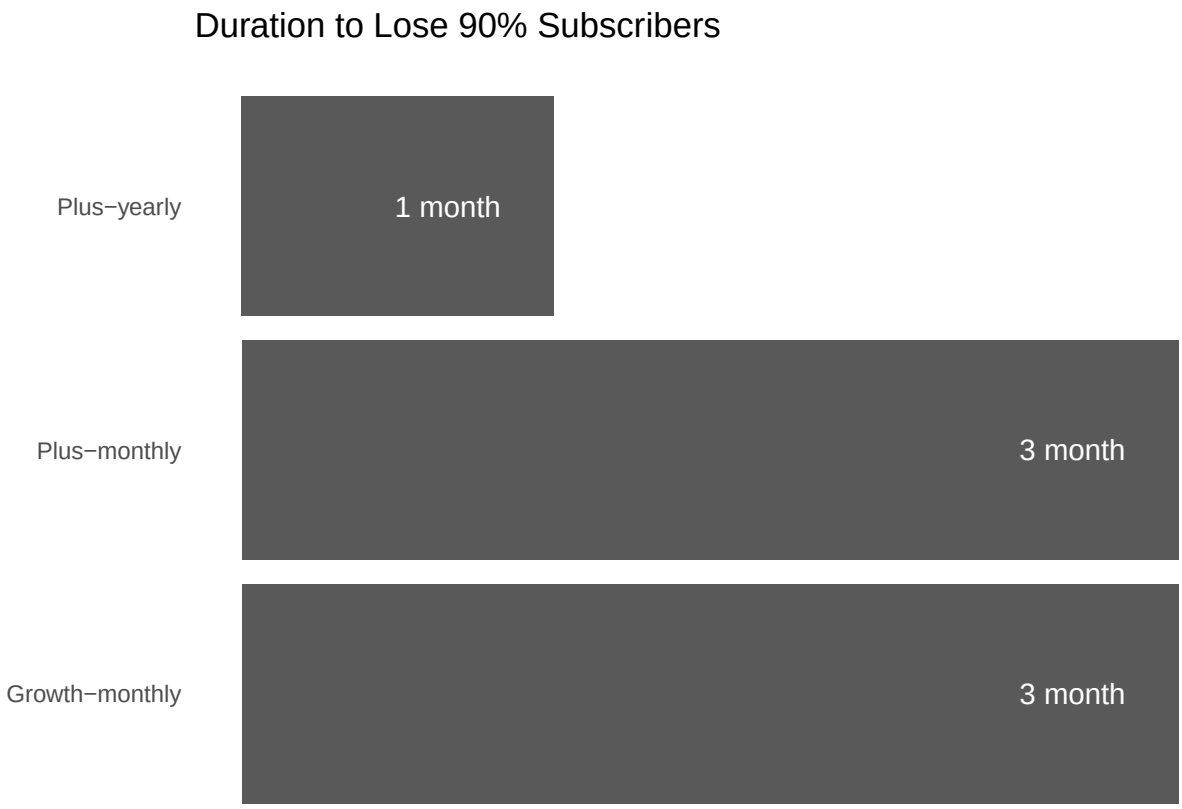
```
## # A tibble: 3 x 3
## # Groups:   subscription [3]
##   subscription   usage_months  Perc
##   <chr>                 <dbl> <dbl>
## 1 Growth-monthly            3  89.6
## 2 Plus-monthly              3  92.4
## 3 Plus-yearly               2  99.3
```

```r
months_90_cancellations <- min_usage2 %>% filter(Perc >= 89) %>% slice(1) # 89% rounded up to 90%
months_90_cancellations
```

```
## # A tibble: 3 x 3
## # Groups:   subscription [3]
##   subscription   usage_months  Perc
##   <chr>                 <dbl> <dbl>
## 1 Growth-monthly            3  89.6
## 2 Plus-monthly              3  92.4
## 3 Plus-yearly               1  96.7
```

```r
ggplot(months_90_cancellations, aes(y = subscription,
                      x = usage_months))+
  geom_bar(stat = "summary")+
  #geom_text(aes(label = usage_months), colour = "white", hjust = 3)+ #paste0(seq(0, 0.6, by = 0.1), "%
  geom_text(aes(label = paste0(usage_months, " month")), colour = "white", hjust = 1.5)+
  labs(title = "Duration to Lose 90% Subscribers",
       #x = 'Months',
       x = NULL,
       y = NULL) +
  theme_minimal() +
  theme(
    strip.text = element_text(face = 'bold', hjust = 0),
    plot.caption = element_text(face = 'italic'),
    panel.grid.major = element_line('white', size = 0.5),
    panel.grid.minor = element_blank(),
    axis.text.x=element_blank(),
    panel.grid.major.x = element_blank(),
    panel.ontop = FALSE
  )
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
#ggsave("7_duration_90_percent_lose_subscribers.jpg", width = 12, height = 8, units = "cm")
```

This plots is more meaningful.

'Plus-yearly' subscription type looses 90% of their subscribers just within 1 month.

Both 'Plus-monthly' and 'Growth-monthly' subscription packages looses 90% of their subscribers within 3 months.