# Análise de Cluster - Instacart

Renato de Camargo
09/08/2020

O objetivo desse estudo é encontrar grupos com padrões de frequência de compra semelhantes na base de pedidos do Instacart.

A base de dados vem do "The Instacart Online Grocery Shopping Dataset 2017", extraída do site https://www.instacart.com/datasets/grocery-shopping-2017 em agosto/2020.

Para a análise de cluster dos usuários, usaremos o método do Kmeans com definição do número de grupos pelo método do cotovelo, trabalhando dissimilaridade 'within-cluster' em função do número de grupos.

## Bibliotecas

```
if(!require(tidyverse)){install.packages("tidyverse");
library(tidyverse)}
if(!require(patchwork)){devtools::install_github("thomasp85/patchwork");
library(patchwork)}
if(!require(RColorBrewer)){install.packages("RColorBrewer");
library(RColorBrewer)}
if(!require(fmsb)){install.packages("fmsb"); library(fmsb)}
```

## Bases de dados

```
raw_orders <- read_csv("./data/orders.csv")
qtd_produto <- read_csv("./data/qtd_produtos.csv")
```

## Feature engineering

Para analisar a frequência dos usuários, nos basearemos na variável "days_since_prior_order". Como a primeira compra não possui compra anterior e traz esse campo vazio, vamos descartá-las.

Também baseado em "days_since_prior_order", vamos criar as variáveis: * Média de dia entre compras por usuário (day_mean) * Desvio padrão de dias entre compras por usuário (day_sd) * Correlação entre o número da compra e dias entre compras por usuário (Correl)

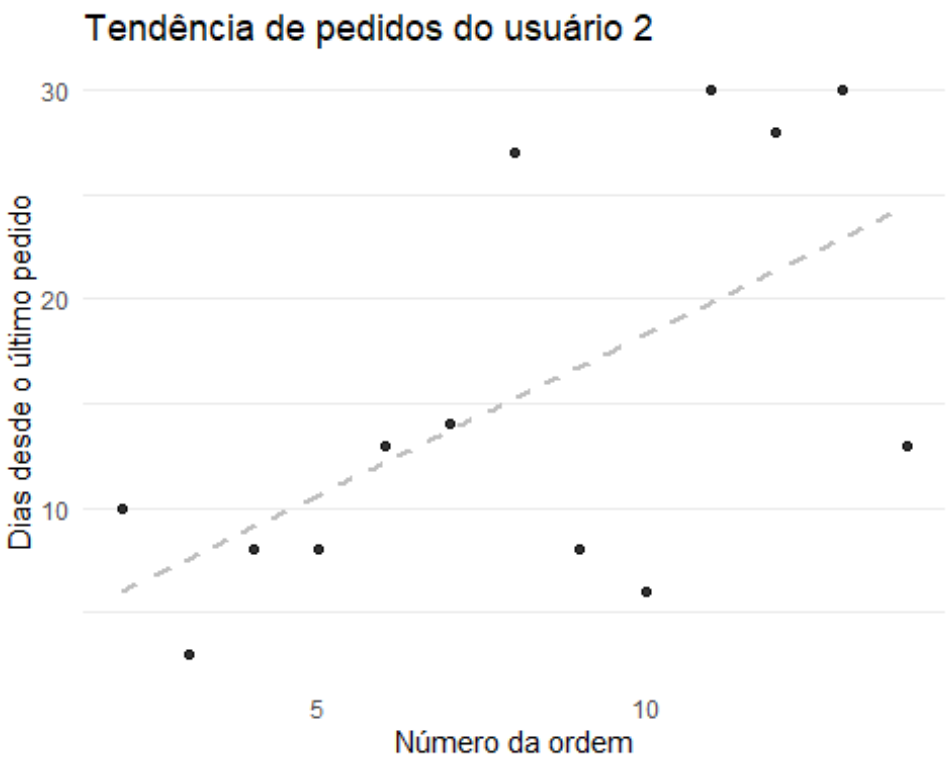
Essa última variável "correl" servirá de estimador para determinar se o usuário está com uma frequência crescente ou decrescente. Por exemplo, se essa correlação for positiva próxima de 1, indica que quanto mais compras o usuário faz, mais tempo ele demora entre uma compra e outra; portando possui uma frequência decrescente.

Por fim, vamos contar o número total de pedidos do usuário (qtd_order) e a média de produtos por pedido do usuário (qtd_produtos)

```
#pegar apenas o segundo pedido em diante, calcular o numero de pedidos e
achar Media e Desvio padrao dos dias entre pedidos
#também calcularemos a variável Correl

freq <- raw_orders %>%
  filter(order_number!= 1) %>%
  group_by(user_id) %>%
  mutate (  day_mean = mean(days_since_prior_order),
            day_sd = sd(days_since_prior_order),
            qtd_order = max(order_number),
            correl = cor(order_number, days_since_prior_order, method =
c("pearson"))) %>%
  ungroup()

#Para a variável Correl, transformaremos os NA em Zero.
#os NAs formaram quando os days_since_prior_order são constantes
freq <- freq %>%
  mutate( correl = coalesce(correl,0 ))


#colocar a quantidade de pridutos em um pedido, vindo da base
"qtd_produto")

freq <-  left_join (freq, qtd_produto, by = "order_id") %>%
  select(-X1) %>%
  filter(!is.na(qtd_produtos))
```

Aprofundando na variável “Correl”, analisando um usuário que há correlação negativa para ver se a tendência é realmente de aumento de dias desde o último pedido

```
freq %>%
  filter(user_id ==     2) %>%
  ggplot(aes(x = order_number, y = days_since_prior_order)) +
  geom_point(color = "black", alpha = 0.8) +
  geom_smooth(method = "lm", se = FALSE, color = "grey", linetype =
"dashed", alpha = 0.8) +
  theme_minimal() +
  theme(axis.line=element_blank(), panel.border =element_blank(),
panel.grid.major.x =element_blank(),
        panel.grid.minor.x =element_blank()) +
  labs(x = "Número da ordem",
       y = "Dias desde o último pedido",
       title = "Tendência de pedidos do usuário 2")
```




Por fim, vamos subir a granularidade dos dados. Ao invés de olhar por pedido, vamos agregá-los e olhar por usuário

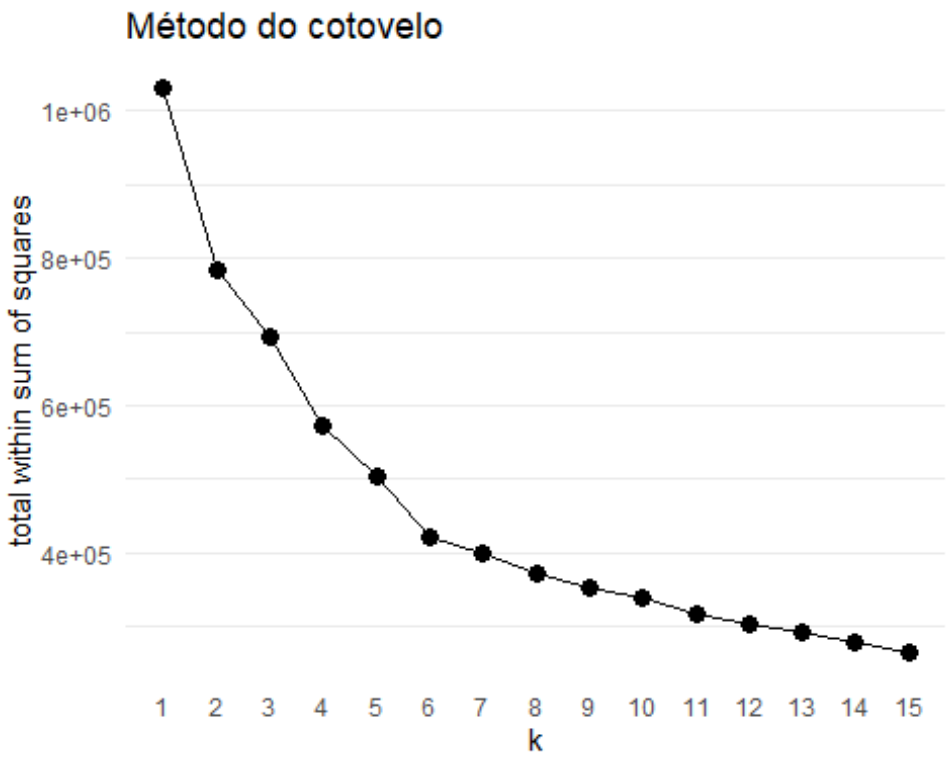
```
freq_user <- freq %>%
  group_by(user_id) %>%
  summarise(day_mean = max(day_mean), day_sd = max(day_sd),
            correl = max(correl), qtd_order = max(qtd_order),
            qtd_produtos = mean(qtd_produtos))
```

## Cálculo dos clusters

Baseado nas variáveis que criamos iremos criar grupos de usuários com o método do Kmeans com definição do número de grupos pelo método do cotovelo, trabalhando dissimilaridade 'within-cluster' em função do número de grupos.

```
set.seed(123)

#Escalonar e centralizar os dados, remover o user_id
freq_pad <- freq_user %>%
  select(-user_id) %>%
  scale()

#definição do critério de dissimilaaridade within cluster
criterio <- function(k) kmeans(freq_pad, k)$tot.withinss
```

```
#coleta da dissimilaridade olhando 15 grupos
estudo <- tibble(k = 1:15) %>%
  mutate(w = map_dbl(k, criterio))

#montagem do gráfico
estudo %>%
  ggplot(aes(k, w)) +
  geom_point(size = 3) +
  geom_line() +
  labs(y = "total within sum of squares", x = "k", title = "Método do
cotovelo") +
  scale_x_continuous(breaks = 1:15) +
  theme_minimal() +
  theme(axis.line=element_blank(), panel.border =element_blank(),
panel.grid.major.x =element_blank(),
        panel.grid.minor.x =element_blank())
```




Visualizando o gráfico, vemos que do 6 ao 7 grupo não há redução expressiva das dissimilaridades; portanto iremos trabalhar com 6 grupos.

```
# Passando o valor de k = 6 ao modelo
set.seed(666)
kmedias <- kmeans(freq_pad, 6)

#adicionando o resultado ao banco de dados
freq_user <- freq_user %>%
```

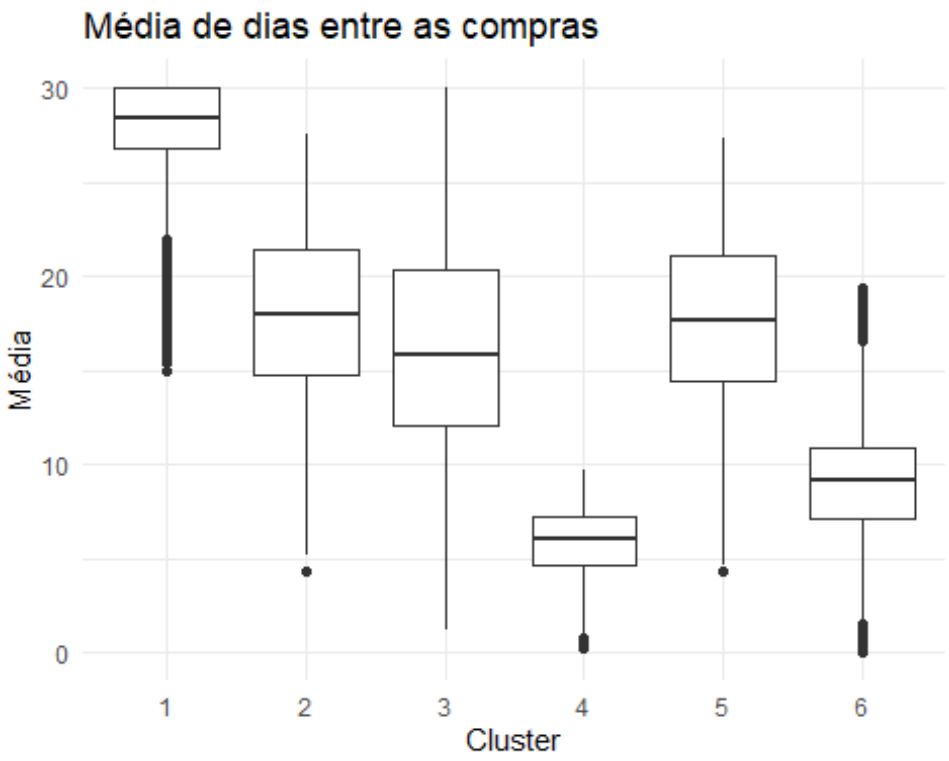
```
  mutate(cluster = kmedias$cluster)

#Tabela resumo de cada cluster
freq_user %>%
  group_by(cluster) %>%
  summarise(n = n())
```

```
## # A tibble: 6 x 2
##   cluster     n
##     <int> <int>
## 1       1 16843
## 2       2 47386
## 3       3 25041
## 4       4 16931
## 5       5 51604
## 6       6 48404
```

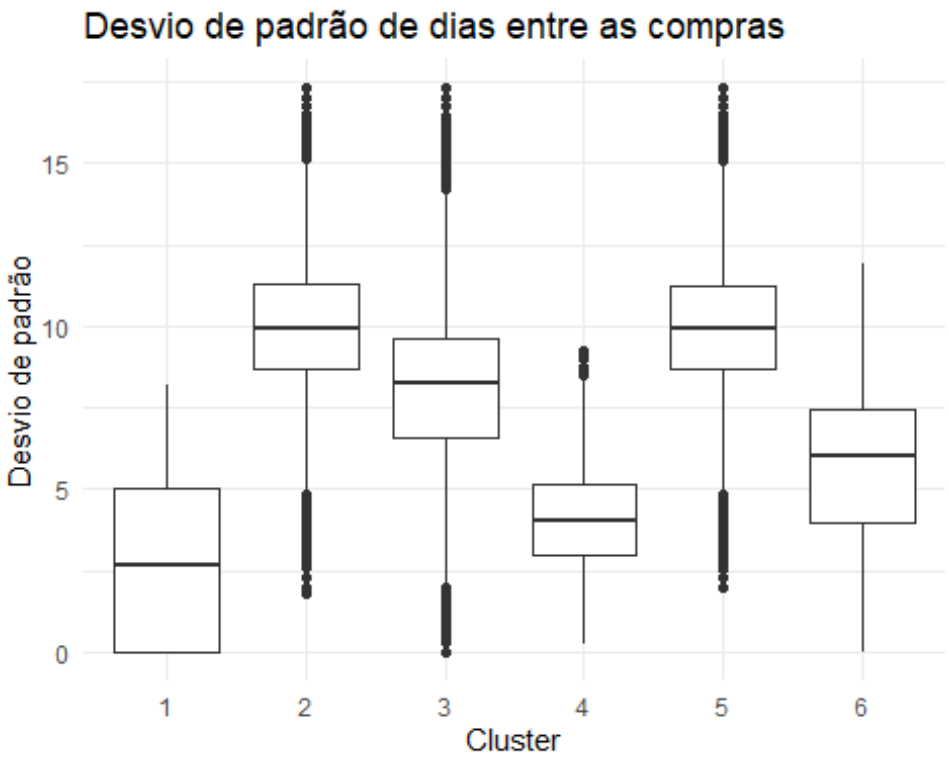
## Interpretação dos clusters

para leitura dos cluster iremos fazer gráficos de boxplot para cada variável

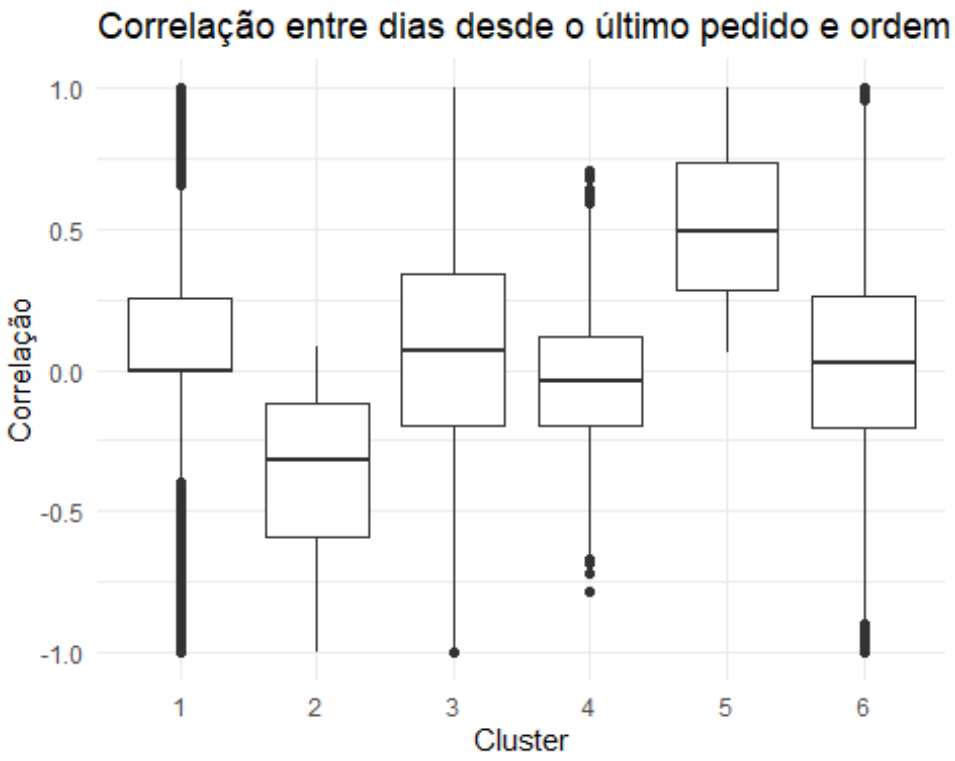
```
freq_user %>%
  ggplot( aes(x=as.character(cluster) , y=day_mean)) +
  geom_boxplot() +
  theme(legend.position="none",
        plot.title = element_text(size=11),
        axis.line=element_blank(),
        panel.border =element_blank(),
        panel.grid.major.x =element_blank()) +
  theme_minimal() +
  labs(x = "Cluster",
       y = "Média",
       title = "Média de dias entre as compras")
```



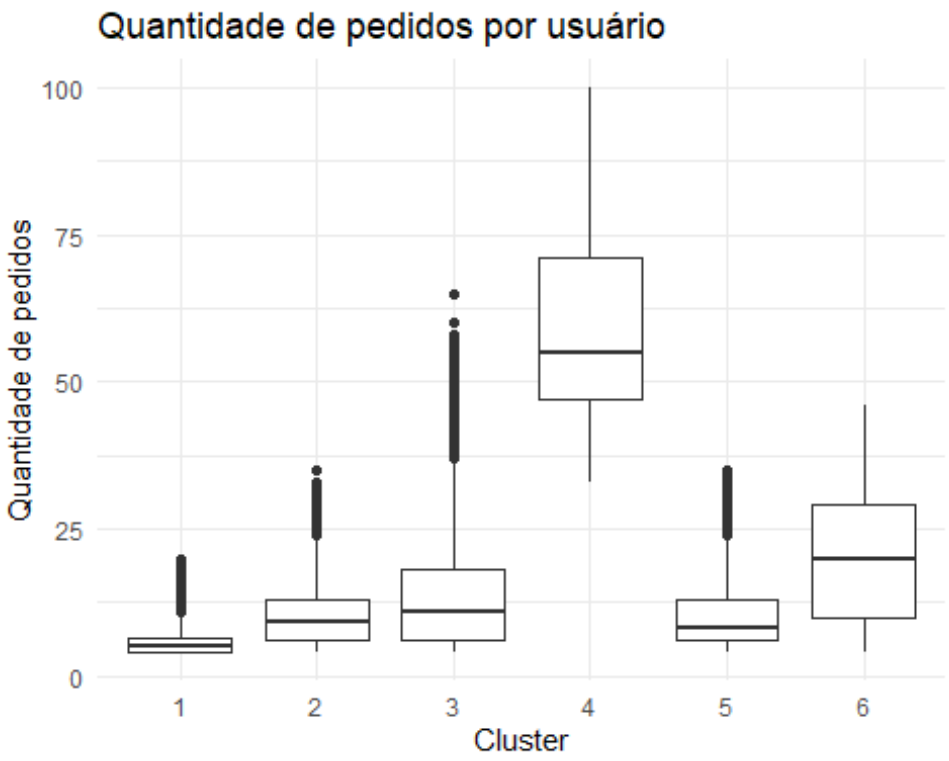

```
freq_user %>%
  ggplot( aes(x=as.character(cluster) , y=day_sd)) +
  geom_boxplot() +
  theme(legend.position="none",
        plot.title = element_text(size=11),
        axis.line=element_blank(),
        panel.border =element_blank(),
        panel.grid.major.y =element_blank()) +
  theme_minimal() +
  labs(x = "Cluster",
       y = "Desvio de padrão",
       title = "Desvio de padrão de dias entre as compras")
```



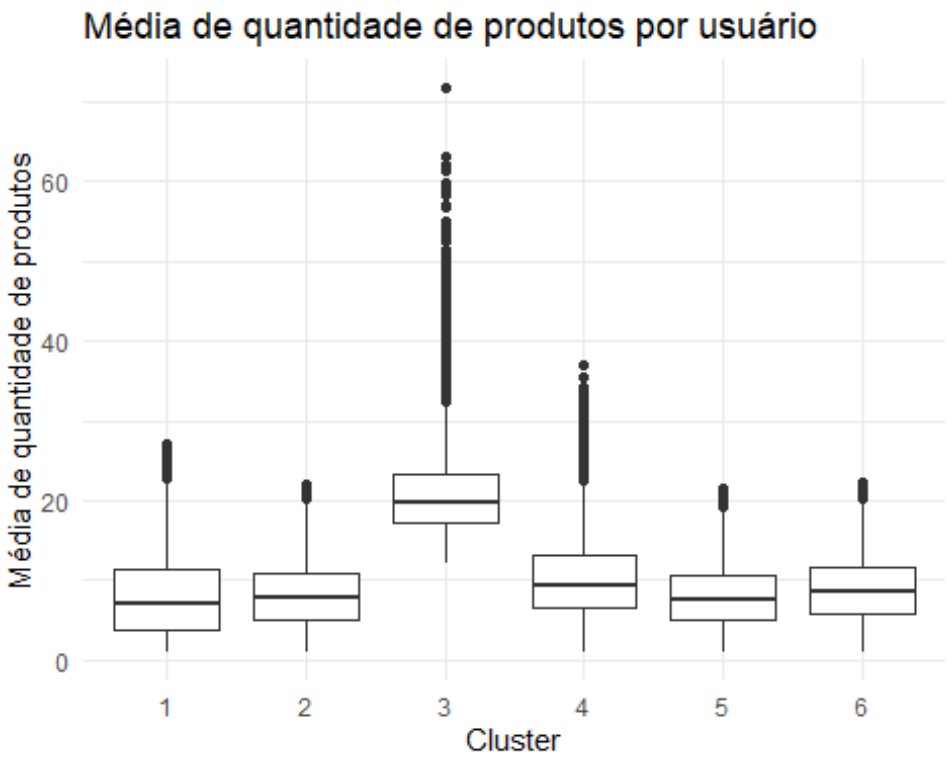

```
freq_user %>%
  ggplot( aes(x=as.character(cluster) , y= correl )) +
  geom_boxplot() +
  theme(legend.position="none",
        plot.title = element_text(size=11),
        axis.line=element_blank(),
        panel.border =element_blank(),
        panel.grid.major.y =element_blank()) +
  theme_minimal() +
  labs(x = "Cluster",
       y = "Correlação",
       title = "Correlação entre dias desde o último pedido e ordem do
pedido")
```




```
freq_user %>%
  ggplot( aes(x=as.character(cluster) , y= qtd_order )) +
  geom_boxplot() +
  theme(legend.position="none",
        plot.title = element_text(size=11),
        axis.line=element_blank(),
        panel.border =element_blank(),
        panel.grid.major.y =element_blank()) +
  theme_minimal() +
  labs(x = "Cluster",
       y = "Quantidade de pedidos",
       title = "Quantidade de pedidos por usuário")
```




```
freq_user %>%
  ggplot( aes(x=as.character(cluster) , y= qtd_produtos )) +
  geom_boxplot() +
  theme(legend.position="none",
        plot.title = element_text(size=11),
        axis.line=element_blank(),
        panel.border =element_blank(),
        panel.grid.major.x =element_blank(),
        panel.grid.minor.x =element_blank()) +
  theme_minimal() +
  labs(x = "Cluster",
       y = "Média de quantidade de produtos",
       title = "Média de quantidade de produtos por usuário")
```




Leitura de cada cluster:

Cluster 1 - Constante mensal com poucos pedidos

Cluster 4 - Constante semanalcom muitos pedidos

Cluster 2 - Frequência crescente

Cluster 5 - Frequência decrescente

Cluster 3 - Muitos produtos sem frequencia clara

Cluster 6 - Intermediário = Frequente 10 dias, constância média, pedidos intermediario,produto médio

## Gráfico resumo

```
#ajustes dos dados
personas <-
  freq_user %>%
  group_by(cluster) %>%
    summarise(day_mean = mean(day_mean),
              day_sd = mean(day_sd),
              correl = mean(correl),
```

```
              qtd_order = mean(qtd_order),
              qtd_produtos = mean(qtd_produtos)) %>%
  select(-cluster) %>%
  mutate(day_mean = (day_mean - min(day_mean)) / (max(day_mean) -
min(day_mean)) * 100,
         day_sd = (day_sd - min(day_sd)) / (max(day_sd) - min(day_sd)) *
100,
         correl = (correl - min(correl)) / (max(correl) - min(correl)) *
100,
         qtd_order = (qtd_order - min(qtd_order)) / (max(qtd_order) -
min(qtd_order)) * 100,
         qtd_produtos = (qtd_produtos - min(qtd_produtos)) /
(max(qtd_produtos) - min(qtd_produtos)) * 100)


# Adiciona os valores min e máx dos eixos
personas <- rbind(rep(100,5), rep(0, 5), personas)
personas
```

```
## # A tibble: 8 x 5
##   day_mean day_sd correl qtd_order qtd_produtos
##      <dbl>  <dbl>  <dbl>     <dbl>        <dbl>
## 1    100    100    100      100         100
## 2      0      0      0        0           0
## 3    100      0     47.4      0           0.381
## 4     54.9  100      0        7.85        1.03
## 5     46.9   71.3   49.7     13.6       100
## 6      0     18.6   37.5    100          17.3
## 7     53.6   99.2  100        7.24        0
## 8     14.2   40.1   44.9     26.5         6.15
```

```
# prepara cores
borda= alpha(c('#1b9e77', '#d95f02', '#7570b3', '#e7298a', '#66a61e',
'#e6ab02'), 1)
inter=alpha(c('#1b9e77', '#d95f02', '#7570b3', '#e7298a', '#66a61e',
'#e6ab02'), 0.5)

# títulos
mytitle <- c('Cluster_1', 'Cluster_2', 'Cluster_3',
             'Cluster_4', 'Cluster_5', 'Cluster_6')

borda
```

```
## [1] "#1B9E77" "#D95F02" "#7570B3" "#E7298A" "#66A61E" "#E6AB02"
```

```
# separa em 6 telas
par(mar=rep(0.8,4))
par(mfrow=c(2,3))

# Loop para cada plot
```

```r
for(i in 1:6){

  # customização de cada plot
  radarchart( personas[c(1,2,i+2),], axistype=1,

              #poligono
              pcol=borda[i] , pfcol=inter[i] , plwd=4, plty=1 ,

              #grid
              cglcol="grey", cglty=1, axislabcol="grey", ccglwd=0.8,

              #labels
              vlcex=0.8,

              #title
              title=mytitle[i]
  )
}
```