K-means Clustering

Heitor Gabriel S. Monteiro

02/11/2021

Contents

1 Prelúdio

Nosso objetivo é formar um classificador de grupos de interesse baseado em dados do número de vezes que certas palavras apareceram em publicações de jovens norte-americanos, em uma determinada rede social. Para essa tarefa, vamos usar o Tidyverse e o Tidymodels para importar, tratar os dados, visualizá-los e aplicar o modelo do algoritmo K-Means. Esse exercício é muito útil, por exemplo, se quisermos determinar precisamente o perfil de interesse para uma propaganda direcionada e personalizada para cada subgrupo.

```
setwd('/home/heitor/Área de Trabalho/R Projects/Análise Macro/Labs/Lab 12')

library(tidyverse)
library(plotly)
library(tidymodels)
library(knitr)
```

```
aa <- read_csv("snsdata.csv") %>% as_tibble()
attach(aa)
```

2 Importação e Tratamento

Ao ver os gráficos pela primeira vez, vemos que uma quantidade grande de dados têm a idade faltante além de observações que reportam idades absurdas. Além de uma predominância feminina e observações sem categorização do gênero.

```
aa %>% summary()
```

```
##     gradyear       gender                age              friends
##  Min.   :2006   Length:30000       Min.   :  3.086   Min.   :  0.00
##  1st Qu.:2007   Class :character   1st Qu.: 16.312   1st Qu.:  3.00
##  Median :2008   Mode  :character   Median : 17.287   Median : 20.00
##  Mean   :2008                      Mean   : 17.994   Mean   : 30.18
##  3rd Qu.:2008                      3rd Qu.: 18.259   3rd Qu.: 44.00
##  Max.   :2009                      Max.   :106.927   Max.   :830.00
##                                    NA's   :5086
##    basketball         football           soccer           softball
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
##  Mean   : 0.2673   Mean   : 0.2523   Mean   : 0.2228   Mean   : 0.1612
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   :24.0000   Max.   :15.0000   Max.   :27.0000   Max.   :17.0000
##
##    volleyball         swimming        cheerleading       baseball
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median : 0.0000   Median :0.0000   Median : 0.0000
##  Mean   : 0.1431   Mean   : 0.1344   Mean   :0.1066   Mean   : 0.1049
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.: 0.0000
##  Max.   :14.0000   Max.   :31.0000   Max.   :9.0000   Max.   :16.0000
##
##      tennis             sports            cute              sex
##  Min.   : 0.00000   Min.   : 0.00   Min.   : 0.0000   Min.   :  0.0000
##  1st Qu.: 0.00000   1st Qu.: 0.00   1st Qu.: 0.0000   1st Qu.:  0.0000
##  Median : 0.00000   Median : 0.00   Median : 0.0000   Median :  0.0000
##  Mean   : 0.08733   Mean   : 0.14   Mean   : 0.3229   Mean   :  0.2094
##  3rd Qu.: 0.00000   3rd Qu.: 0.00   3rd Qu.: 0.0000   3rd Qu.:  0.0000
##  Max.   :15.00000   Max.   :12.00   Max.   :18.0000   Max.   :114.0000
```

```
##
##       sexy               hot              kissed            dance
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
##  Mean   : 0.1412   Mean   : 0.1266   Mean   : 0.1032   Mean   : 0.4252
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   :18.0000   Max.   :10.0000   Max.   :26.0000   Max.   :30.0000
##
##       band            marching           music             rock
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
##  Mean   : 0.2996   Mean   : 0.0406   Mean   : 0.7378   Mean   : 0.2433
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.: 0.0000
##  Max.   :66.0000   Max.   :11.0000   Max.   :64.0000   Max.   :21.0000
##
##       god              church            jesus             bible
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.00000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.00000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.00000
##  Mean   : 0.4653   Mean   : 0.2482   Mean   : 0.1121   Mean   : 0.02133
##  3rd Qu.: 1.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.00000
##  Max.   :79.0000   Max.   :44.0000   Max.   :30.0000   Max.   :11.00000
##
##       hair             dress             blonde              mall
##  Min.   : 0.0000   Min.   :0.000   Min.   :  0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.:0.000   1st Qu.:  0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median :0.000   Median :  0.0000   Median : 0.0000
##  Mean   : 0.4226   Mean   :0.111   Mean   :  0.0989   Mean   : 0.2574
##  3rd Qu.: 0.0000   3rd Qu.:0.000   3rd Qu.:  0.0000   3rd Qu.: 0.0000
##  Max.   :37.0000   Max.   :9.000   Max.   :327.0000   Max.   :12.0000
##
##     shopping          clothes          hollister         abercrombie
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median : 0.000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   : 0.353   Mean   :0.1485   Mean   :0.06987   Mean   :0.05117
##  3rd Qu.: 1.000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :11.000   Max.   :8.0000   Max.   :9.00000   Max.   :8.00000
##
##       die              death             drunk              drugs
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :0.00000   Min.   : 0.00000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.: 0.00000
##  Median : 0.0000   Median : 0.0000   Median :0.00000   Median : 0.00000
```

```
##  Mean   : 0.1841   Mean   : 0.1142   Mean   :0.08797   Mean   : 0.06043
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:0.00000   3rd Qu.: 0.00000
##  Max.   :22.0000   Max.   :14.0000   Max.   :8.00000   Max.   :16.00000
##
```

```
aa <- aa %>% mutate(gender = factor(gender))
aa$gender %>% summary()
```

```
##     F     M  NA's
## 22054  5222  2724
```

```
aa$gender %>% table() %>%
    prop.table() %>%
  round(digits = 4)
```

```
## .
##      F      M
## 0.8085 0.1915
```

Para não retirar a grande quantidade de `NA` em `gender` e `age`, criaremos uma variável binária para feminino/não-feminino e outra para gênero conhecido e desconhecido. Quem for, por exemplo, homem, será então não-feminino e gênero conhecido.

```
aa <- aa %>%
    mutate( female =
                case_when(gender=='M'   ~ 0,
                          is.na(gender) ~ 0,
                          gender=='F'   ~ 1),
            unk_gender =
                case_when(is.na(gender) ~ 1,
                          gender=='M'   ~ 0,
                          gender=='F'   ~ 0))
```

Vemos que as quantidades de `female` e `unk_gender` batem com os números anteriores de `age` para `F` e para `NA`.

```
sum(aa$female==1)
```

```
## [1] 22054
```

```
sum(aa$unk_gender==1)
```

```
## [1] 2724
```

```
aa$age <- replace(aa$age,
                  aa$age<=13|aa$age>20,
                  NA)
summary(aa$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   13.03   16.30   17.27   17.25   18.22   20.00    5523
```

Para contornar o problema das idades faltantes sem deletar as observações, aplicaremos a idade média de cada ano de conclusão da *high school*. A função `agregate()` aplica determinada função, no caso: `mean`, em `age`, mas dividindo-o em subgrupos de acordo com `gradyear`. A função `ave()` aplica a média em cada observação do vetor `age`, restando ao `if_else()` aplicar na observação com `NA` e devolver o vetor transformado ao nosso banco de dados `aa`.

```
aggregate(data = aa,
          age ~ gradyear,
          mean, na.rm = TRUE)
```

```
##   gradyear      age
## 1     2006 18.65586
## 2     2007 17.70617
## 3     2008 16.76770
## 4     2009 15.81957
```

```
ave_age <- ave(age, gradyear,
               FUN = function(x) mean(x, na.rm = TRUE))

aa$age <- if_else(is.na(age), ave_age, aa$age)
summary(aa$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   13.03   16.50   17.42   17.37   18.38   20.00     437
```

Já que nossas variáveis são a frequência de menção a cada assunto, em cada observação, ganharemos mais informação se compararmos essa tal frequência ao número médio de frequências daquele tópico: o quanto cada usuário citou o tópico relativo aos demais. Portanto aplicaremos a função `scale` para tranformar `aa` em `cc`; sem mexer nas variáveis `c(gradyear, gender, female, unk_gender)`.

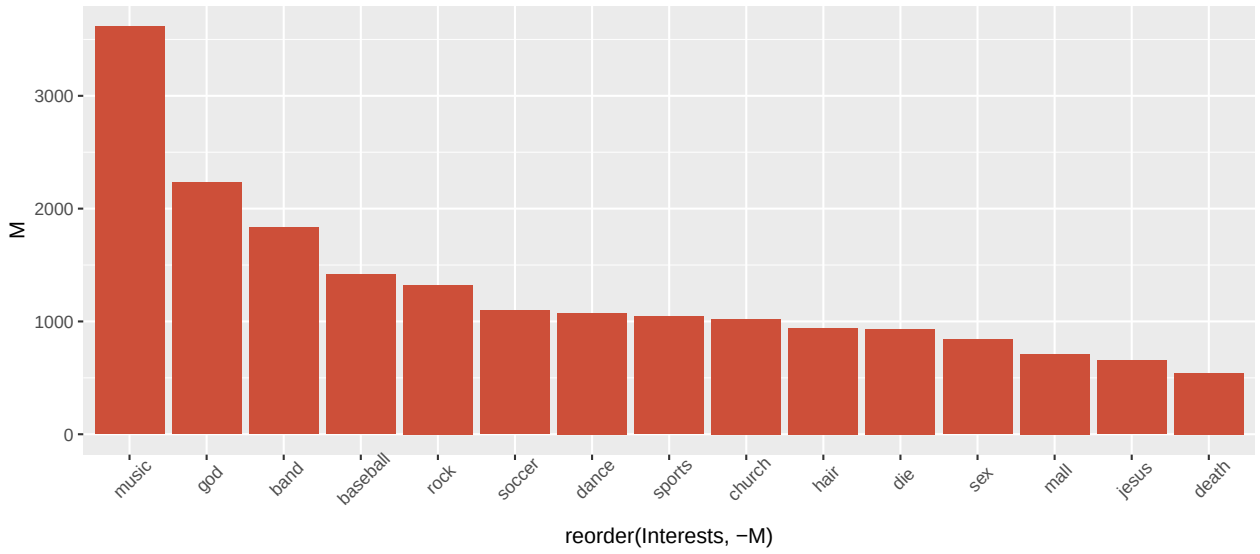
```
bb <- aa %>% select(! c(gradyear, gender,
                        age, female, unk_gender))

bb <- as.data.frame(
    lapply(bb, scale))

cc <- cbind(bb,
            'gradyear'   = aa$gradyear,
            'gender'     = aa$gender,
            'female'     = aa$female,
            'unk_gender' = aa$unk_gender)
rm(bb)
```

3 Visualizações e Descrições

Vamos entender melhor os tópicos mais abortados por gênero, para exemplificação.

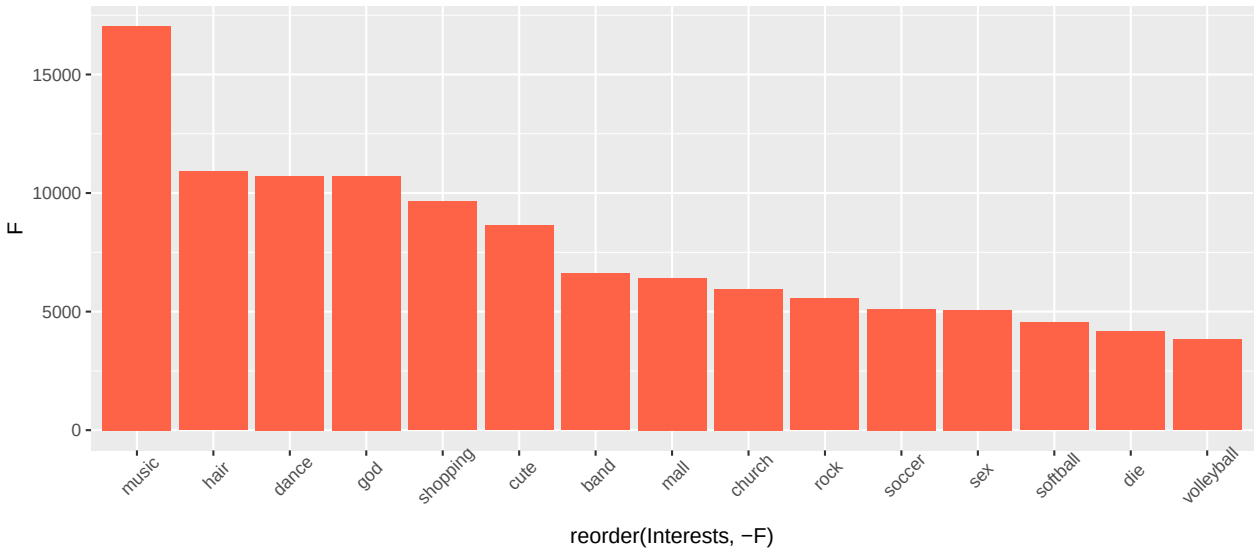
```
a2 <- aa %>%
    select(! c(age, female, unk_gender, gradyear)) %>%
    group_by(gender) %>%
    summarise( across(.cols = 4:37,
                      .fns = sum))%>%
    pivot_longer(cols = soccer:drugs,
                 names_to = 'Interests') %>%
    pivot_wider(names_from = gender,
                values_from = value)
```

```
gg5 <- a2 %>%
    select(Interests, M) %>%
    arrange(desc(M)) %>%
    slice_head(n = 15) %>%
    ggplot(aes(x=reorder(Interests, -M),
               y=M)) +
    geom_col(fill = 'tomato3')+
    theme(axis.text.x =
            element_text(angle = 45,
                         vjust = 0.8))
#ggplotly(
    gg5
```




```
    #)
```

```
gg6 <- a2 %>%
    select(Interests, `F`) %>%
    arrange(desc(`F`)) %>%
    slice_head(n = 15) %>%
    ggplot(aes(x=reorder(Interests, -`F`),
               y=`F`)) +
    geom_col(fill = 'tomato')+
    theme(axis.text.x =
            element_text(angle = 45,
                         vjust = 0.8))
#ggplotly(
    gg6
```




```
    #)
```

Podemos também, com a base escalonada, observar se há discrepâncias na sitação de determinados tópicos entre os sexos, por exemplo:

```
cc %>%
    dplyr::group_by(gender) %>%
    dplyr::summarise(mean_music = mean(music, na.rm=T),
                     mean_god = mean(god, na.rm=T),
                     mean_die = mean(die, na.rm=T),
                     mean_rock = mean(rock, na.rm=T),
                     mean_sex = mean(sex, na.rm=T)) %>%
  kable()
```

gender	mean_music	mean_god	mean_die	mean_rock	mean_sex
F	0.0277637	0.0155712	0.0071777	0.0111742	0.0179150
M	-0.0362341	-0.0280564	-0.0093125	0.0133748	-0.0435469
NA	-0.1553177	-0.0722823	-0.0402593	-0.1161086	-0.0615619

4 O Modelo

A regra de bolso é escolhermos o número de grupos de acordo com $\sqrt{\frac{n}{2}}$.

```
set.seed(123)
sqrt(nrow(cc)/2)
```

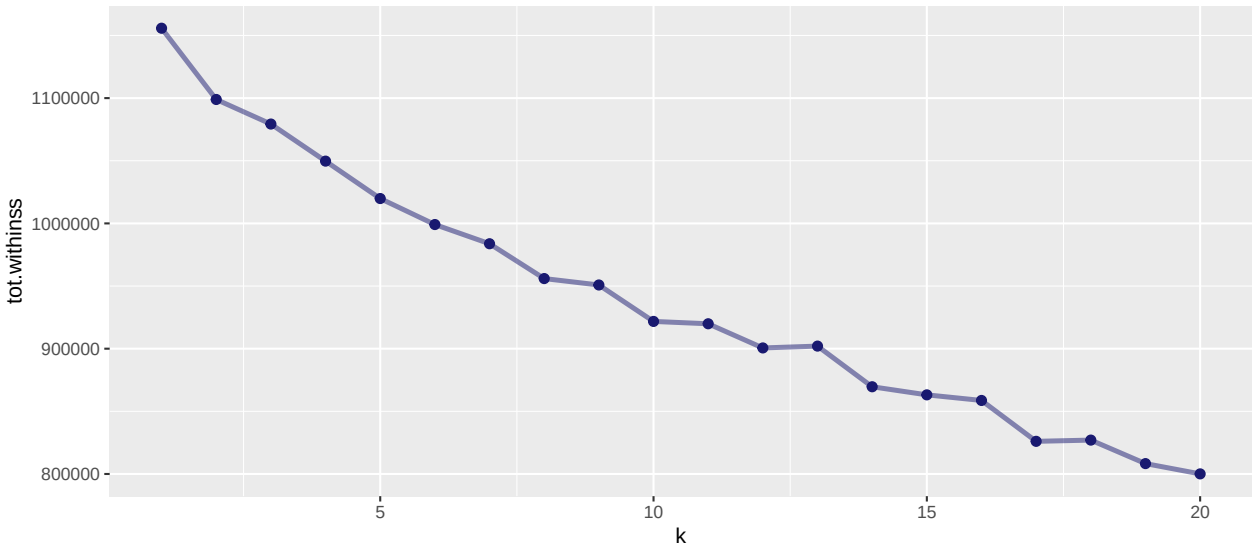
```
## [1] 122.4745
```

Vamos treinar modelos de 1 até 20 clasificações por conta do custo de processamento computacional. É super recomendado conhecer o Broom, ferramenta que nos ajuda a manipular os resultados e informações do modelo. A função `tidy()` resume em um nível por cluster, `augment()` adiciona as classificações de cada observação ao conjunto de dados original e a função `relance()` extrai um resumo em uma única linha.

```
kclusts <-
    tibble(k = 1:20) %>%
    mutate(
        kclust = map(k,
                     ~kmeans(dplyr::select(cc, -gender),
                             .x)),
        tidied = map(kclust, tidy),
        glanced = map(kclust, glance),
        augmented = map(kclust,
                        augment,
                        dplyr::select(cc, -gender)))

km_clusters <-
    kclusts %>%
    unnest(cols = c(tidied))

km_assignments <-
    kclusts %>%
    unnest(cols = c(augmented))

km_clusterings <-
    kclusts %>%
    unnest(cols = c(glanced))
```

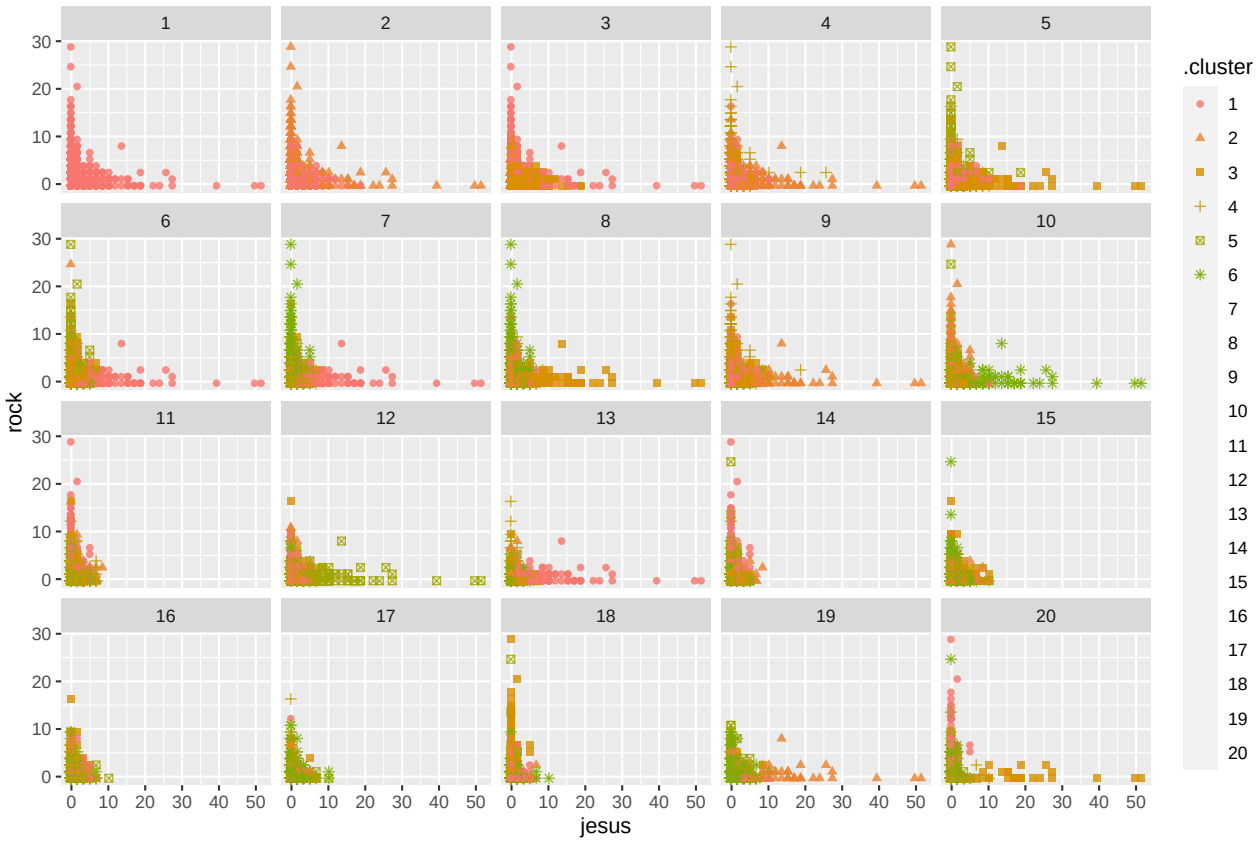
Vemos que o ganho marginal de especificação de clusters ainda é alto:

```
km_clusterings %>%
    ggplot(aes(k, tot.withinss)) +
    geom_line(alpha = 0.5,
              size = 1.2,
              color = "midnightblue") +
    geom_point(size = 2,
               color = "midnightblue")
```



Asseguir, podemos ter uma ideia de como estão organizadas os grupos de dois a dois especificadores, vamos lembrar que o agrupamento é feito com 39 variáveis:

```
gg7 <- km_assignments %>%
    ggplot(aes(x = jesus, y = rock)) +
    geom_point(aes(color = .cluster,
                   shape = .cluster), alpha = 0.8) +
    facet_wrap(~ k)

#ggplotly(
    gg7
```




```
    #)
```

Como, no nosso exemplo de aplicação em uma situação real, os gastos de marketing para mais de 20 propagandas específicas seria muito alta, vamos supor que o departamento de *marketing* tem orçamento para 5 propagandas específicas, então vamos separar esses dez grupos de usuários:

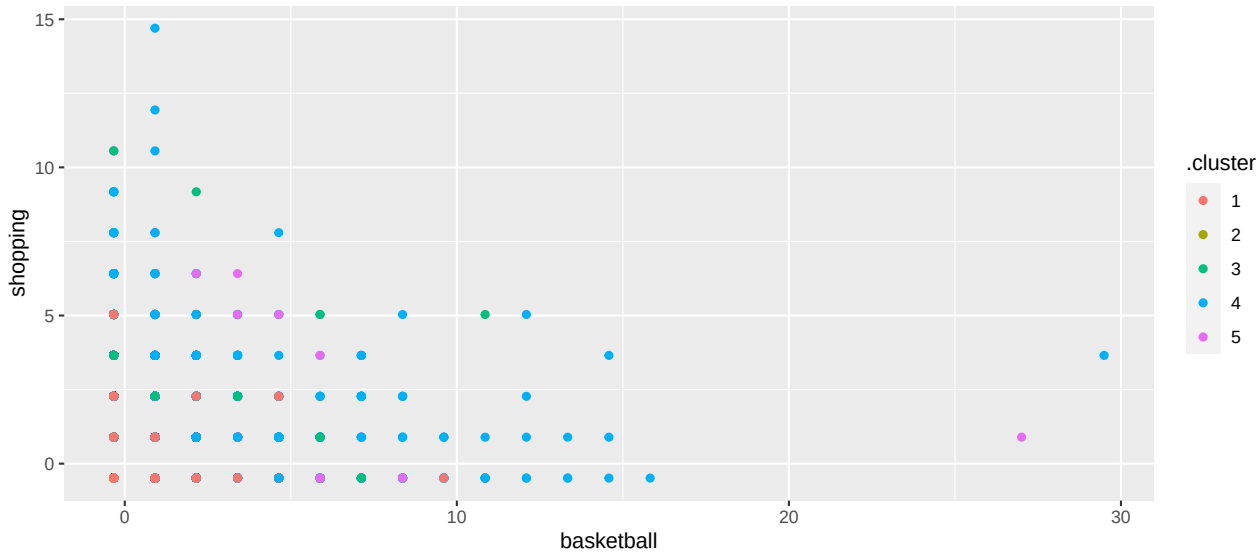
```
km05 <- kmeans(dplyr::select(cc, -gender),
               centers = 5)

summary(km05)
```

```
##              Length Class  Mode
## cluster      30000  -none- numeric
## centers        200  -none- numeric
## totss            1  -none- numeric
## withinss         5  -none- numeric
## tot.withinss     1  -none- numeric
## betweenss        1  -none- numeric
## size             5  -none- numeric
## iter             1  -none- numeric
## ifault           1  -none- numeric
```

```
broom::tidy(km05) %>%
    dplyr::select(35:43) %>%
    kable()
```

death	drunk	drugs	gradyear	female	unk_gender	size	withinss	cluster
-0.0815409	-0.1062799	-0.1086062	2008.504	0.7245744	0.0953739	11219	178717.39	1
-0.0714819	-0.0625924	-0.1097016	2006.476	0.6879037	0.1034399	12384	191094.73	2
0.1051973	0.0347360	0.0337144	2007.855	0.8375286	0.0755149	874	59022.89	3
0.1652253	0.0061577	-0.0546623	2007.724	0.8566985	0.0548767	4501	405611.93	4
0.9436574	1.8683205	2.7334304	2007.604	0.8003914	0.0587084	1022	185394.79	5

```
#ggplotly(
  broom::augment(km05, cc) %>%
    ggplot(aes(basketball, shopping,
               color=.cluster))+
    geom_point()
```




```
#  )
```

5 Referências:

- *Machine Learning with R Expert techniques for predictive modeling, 3rd Edition by Brett Lantz*

- *Getting started with k-means and employment status. TidyTuesday with Julia Silge*
- *K-means clustering with tidy data principles*
- *Análise Macro*