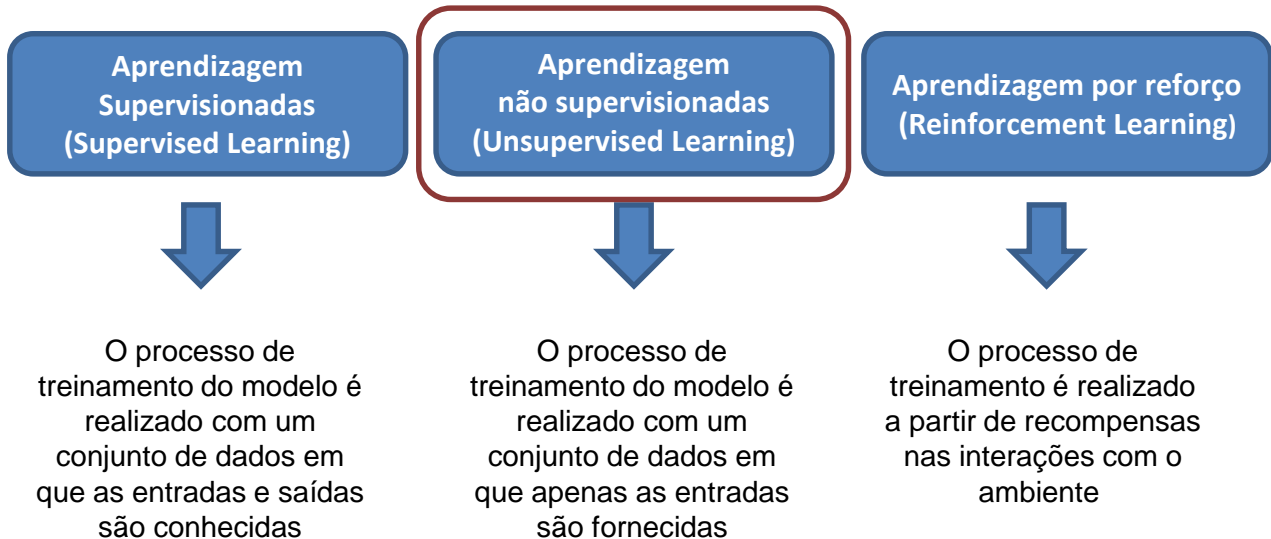


# Aplicação de Técnicas de Aprendizagem de Máquina utilizando R

Mário de Noronha Neto e Richard Demo Souza

# Alguns tipos de aprendizado de máquina



# Técnicas de Aprendizagem não Supervisionada abordadas neste curso

**Associação - Market Basket Analysis**



Utilizada para identificar associações úteis nos dados

**Agrupamento (Clustering)**



Utilizada para segmentação de grupos com algumas semelhanças

# Clustering



A técnica de *Clustering* busca dividir automaticamente o conjunto de dados em grupos de itens similares. Esta técnica não é utilizada para predições, mas sim para extrair conhecimento que podem fornecer informações relevantes sobre o agrupamento natural encontrado nos dados.

É baseada no princípio de que itens/elementos dentro de um mesmo grupo devem ser muito similares entre si, mas bem distintos dos itens/elementos que não estão no mesmo grupo.

Este processo cria um "novo dado" em que exemplos não rotulados recebem o rótulo de um *cluster*. Por esta razão, esta técnica também pode ser entendida como uma técnica de classificação não supervisionada pelo fato de classificar exemplos não rotulados.

# Clustering



Alguns exemplos de aplicações da técnica de *Clustering*:

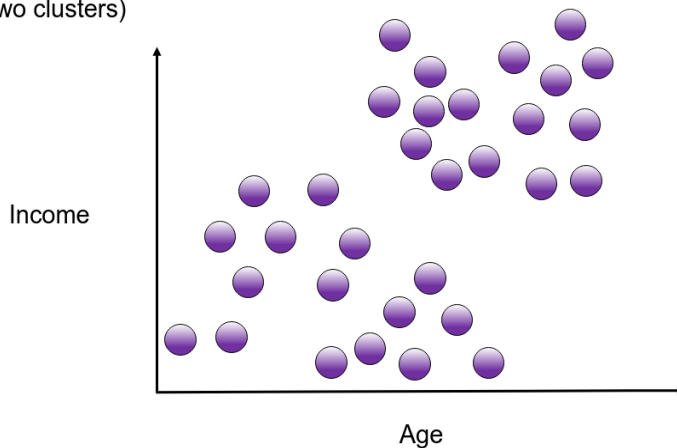
- Segmentação de clientes em grupos com dados demográficos ou padrões de compra semelhantes para campanhas de marketing direcionadas
- Detecção de comportamento anômalo, como intrusões de rede não autorizadas, identificando padrões de uso que estão fora dos clusters conhecidos
- Simplificação de conjuntos de dados extremamente grandes através do agrupamento de características com valores semelhantes em um conjunto menor de categorias.

# Algoritmo k-means



## K-Means Algorithm

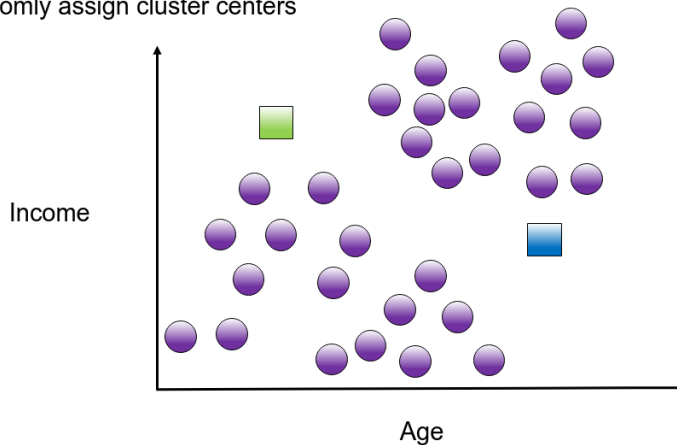
$K = 2$  (find two clusters)



# Algoritmo k-means

## K-Means Algorithm

K = 2, Randomly assign cluster centers

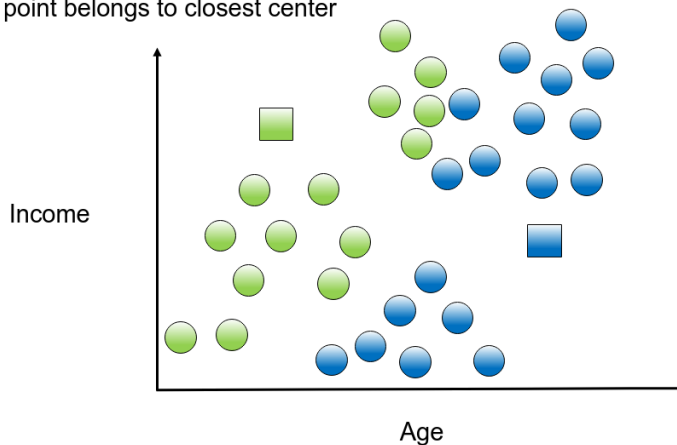


# Algoritmo k-means



## K-Means Algorithm

K = 2, Each point belongs to closest center



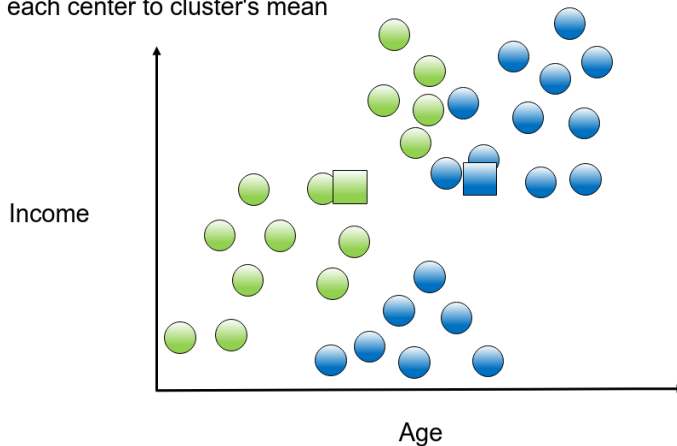


# Algoritmo k-means



## K-Means Algorithm

$K = 2$ , Move each center to cluster's mean

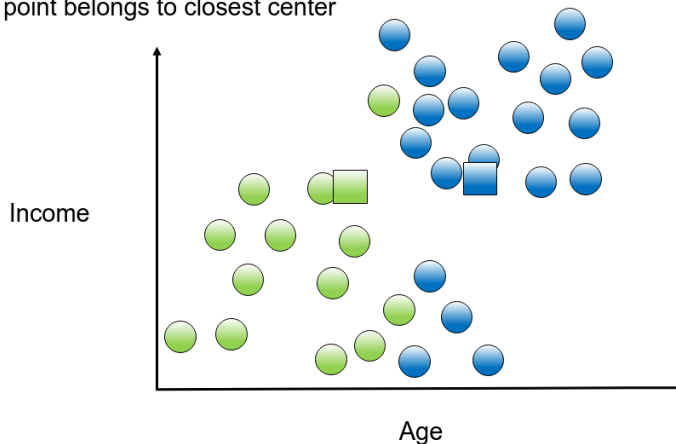


# Algoritmo k-means



## K-Means Algorithm

$K = 2$ , Each point belongs to closest center

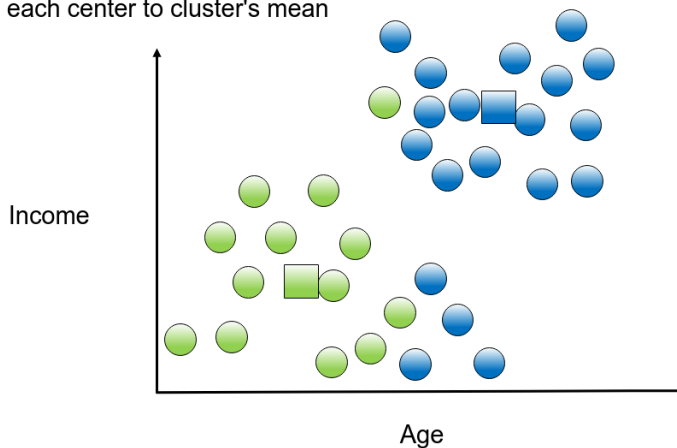


# Algoritmo k-means



## K-Means Algorithm

$K = 2$ , Move each center to cluster's mean

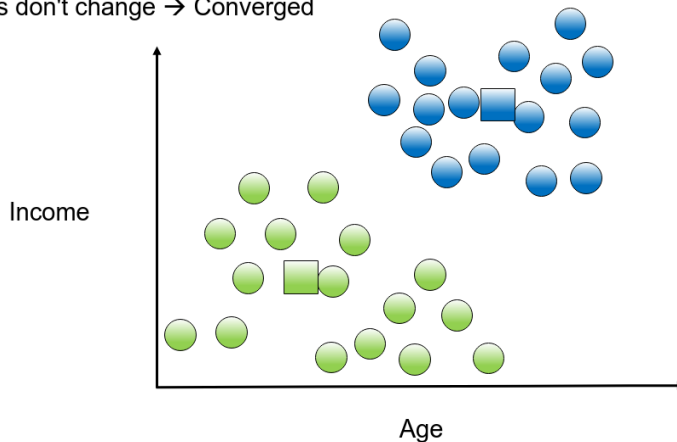


# Algoritmo k-means



## K-Means Algorithm

K = 2, Points don't change → Converged

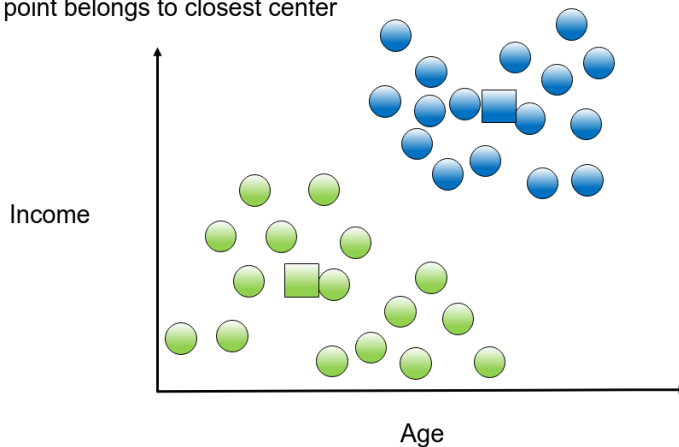


# Algoritmo k-means



## K-Means Algorithm

$K = 2$ , Each point belongs to closest center

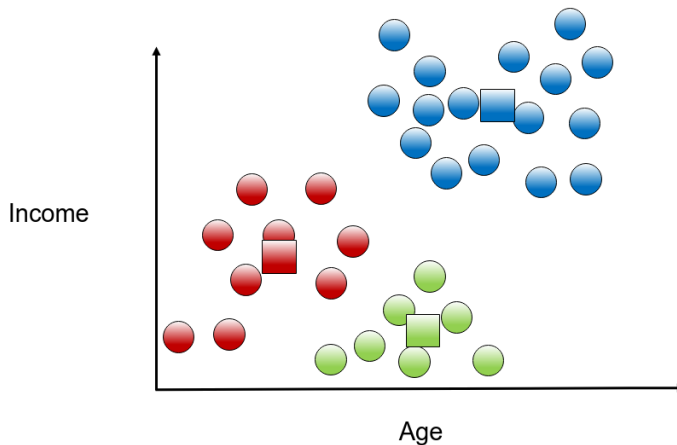


# Algoritmo k-means



## K-Means Algorithm

$K = 3$

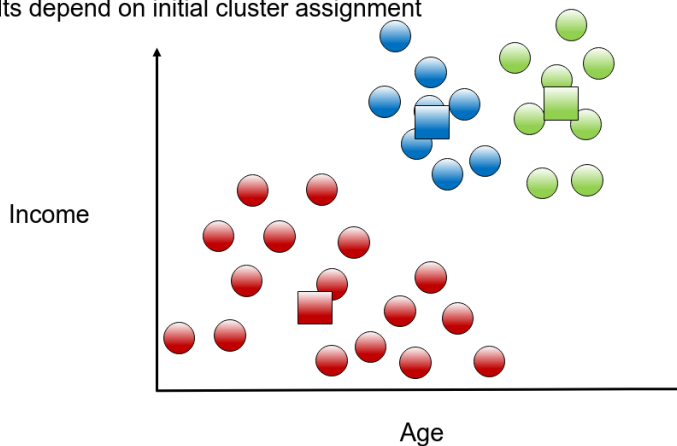


# Algoritmo k-means



## K-Means Algorithm

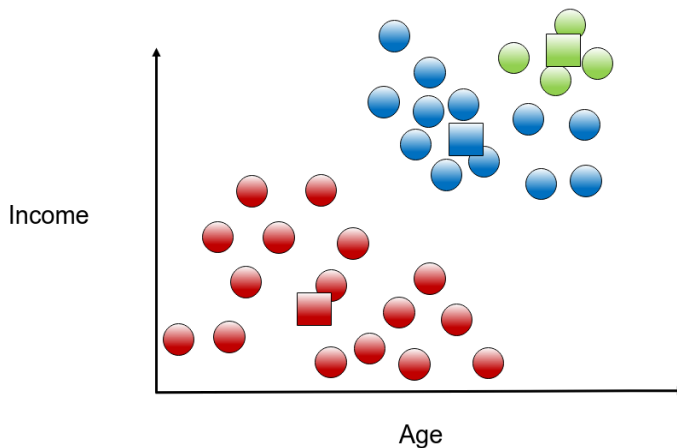
K = 3, Results depend on initial cluster assignment



# Algoritmo k-means



Which Model is the Right One?





# Considerações - Algoritmo k-means

O k-means normalmente utiliza a distância Euclidiana para ajustar definir os cluster, embora outras distâncias possam ser utilizadas.

**Distância Euclidiana**

$$\text{dist}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Esta técnica é muito sensível aos posicionamentos dos pontos centrais iniciais.

Da mesma forma, o número de *cluster* influencia bastante o desempenho. Para a definição do número de *clustes* é importante que se tenha uma noção dos possíveis grupos que o conjunto possui.

# Exemplo: Identificando segmentos de mercado para adolescentes utilizando o k-means



## Passo 1: Coleta de dados

### Dataset utilizado:

O *dataset* utilizado neste exemplo representa uma amostra aleatória de 30.000 estudantes de *High School* dos Estados Unidos, os quais possuíam seu perfis publicados em uma rede social bem conhecida no ano de 2006. Os dados foram amostrados uniformemente nos quatro anos da *High School*. Portanto, podemos supor que este conjunto de dados seja representativo para avaliar o comportamento dos adolescentes naquela época. Usando um processo automatizado, o texto completo dos perfis pode-se registrar o sexo, a idade e o número de amigos do SNS de cada adolescente. Uma ferramenta de mineração foi utilizada para dividir o conteúdo restante das páginas em palavras. Das 500 palavras que mais apareceram em todas as páginas, 36 foram escolhidas para representar 5 categorias de interesse: **Atividades extracurriculares, modas, religião, romance e comportamento antissocial**. Estas palavras incluem termos como *futebol, sexy, beijo, bíblia, compra, morte e drogas*. Este conjunto de dados foi compilado por *Brett Lantz*.

## Passo 2: Explorando e preparando os dados

```
> teens <- read.csv("snsdata.csv")
```

```
> str(teens)
'data.frame':   30000 obs. of  40 variables:
 $ gradyear    : int   2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 ...
 $ gender      : Factor w/ 2 levels "F","M": 2 1 2 1 NA 1 1 2 1 1 ...
 $ age        : num   19 18.8 18.3 18.9 19 ...
 $ friends     : int    7 0 69 0 10 142 72 17 52 39 ...
 $ basketball  : int    0 0 0 0 0 0 0 0 0 0 ...
 $ football    : int    0 1 1 0 0 0 0 0 0 0 ...
 $ soccer      : int    0 0 0 0 0 0 0 0 0 0 ...
 $ softball    : int    0 0 0 0 0 0 0 1 0 0 ...
 $ volleyball  : int    0 0 0 0 0 0 0 0 0 0 ...
```

Observe que o conjunto de dados possui 4 variáveis indicando características pessoais e 36 palavras indicando interesse.

## Passo 2: Explorando e preparando os dados

```
> table(teens$gender)
```

F	M
22054	5222

```
> table(teens$gender, useNA = "ifany")
```

F	M	<NA>
22054	5222	2724

```
> summary(teens$age)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
3.086	16.310	17.290	17.990	18.260	106.900	5086

Observe que aproximadamente 9% não cadastraram a opção *gênero* e 17% a opção *idade*.  
Observe também que os valores mínimo e máximo estão estranhos.

## Passo 2: Explorando e preparando os dados

```
> teens$age <- ifelse(teens$age >= 13 & teens$age < 20,  
                      teens$age, NA)
```

```
> summary(teens$age)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
13.03	16.30	17.26	17.25	18.22	20.00	5523

Todas as idades que estiverem fora deste intervalo serão tratadas com o valores NA.

## Passo 2: Explorando e preparando os dados

Uma solução fácil para tratar os dados sem valor (NA) é excluí-los do conjunto de dados. Entretanto, eles podem representar uma parcela significativa do *dataset*, principalmente se consideramos que os NAs de uma variável podem ser diferentes dos NAs de outra. Uma outra solução é utilizar o conceito de *dummy coding* para criar valores binários para cada nível da variável.

```
> teens$female <- ifelse(teens$gender == "F" &
                          !is.na(teens$gender), 1, 0)
> teens$no_gender <- ifelse(is.na(teens$gender), 1, 0)

> table(teens$gender, useNA = "ifany")
   F      M  <NA>
22054  5222  2724

> table(teens$female, useNA = "ifany")
   0      1
7946 22054

> table(teens$no_gender, useNA = "ifany")
   0      1
27276  2724
```

## Passo 2: Explorando e preparando os dados

Para o caso da variável numérica, como é o caso da variável *idade*, não faz sentido criar uma nova categoria com valores desconhecidos. Neste caso, uma alternativa seria identificar a idade típica do estudante no ano de graduação que ele está cursando e atribuir este valor aos respectivos valores NAs.

```
> ave_age <- ave(teens$age, teens$gradyear, FUN =  
                 function(x) mean(x, na.rm = TRUE))  
  
> teens$age <- ifelse(is.na(teens$age), ave_age, teens$age)  
  
> summary(teens$age)  
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
 13.03  16.28   17.24   17.24   18.21   20.00
```

## Passo 3: Treinando o modelo

### Clustering syntax

using the `kmeans()` function in the `stats` package

#### Finding clusters:

```
myclusters <- kmeans(mydata, k)
```

- `mydata` is a matrix or data frame with the examples to be clustered
- `k` specifies the desired number of clusters

The function will return a cluster object that stores information about the clusters.

#### Examining clusters:

- `myclusters$cluster` is a vector of cluster assignments from the `kmeans()` function
- `myclusters$centers` is a matrix indicating the mean values for each feature and cluster combination
- `myclusters$size` lists the number of examples assigned to each cluster

#### Example:

```
teen_clusters <- kmeans(teens, 5)  
teens$cluster_id <- teen_clusters$cluster
```



## Passo 3: Treinando o modelo

A função `kmeans()` requer que o *dataframe* contenha apenas variáveis numéricas e um parâmetro especificando o número de *clusters*. Para evitar que uma variável se sobressaia sobre outra por ter uma escala maior, vamos utilizar a função `scale()` para aplicar uma padronização z-score. Desta forma, todas as variáveis terão média zero e desvio padrão unitário.

$$\text{z-score: } X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

```
> interests <- teens[5:40]
> interests_z <- as.data.frame(lapply(interests, scale))
```

Como mencionado anteriormente, a definição do número de *cluster* pode ser feita com base em informações *a priori* do que se deseja analisar. Neste estudo, foi escolhido inicialmente um  $k = 5$ .

```
> set.seed(2345)
> teen_clusters <- kmeans(interests_z, 5)
```

## Passo 4: Analisando o modelo

```
> teen_clusters$size
```

```
[1] 871 600 5981 1034 21514
```

```
> teen_clusters$centers
```

	basketball	football	soccer	softball
1	0.16001227	0.2364174	0.10385512	0.07232021
2	-0.09195886	0.0652625	-0.09932124	-0.01739428
3	0.52755083	0.4873480	0.29778605	0.37178877
4	0.34081039	0.3593965	0.12722250	0.16384661
5	-0.16695523	-0.1641499	-0.09033520	-0.11367669

Variáveis de  
interesses

Clusters

Valor médio do  
cluster para o  
interesse listado

Exemplo: O *cluster* 3 é o *cluster* que tem mais interesse em futebol

## Passo 4: Analisando o modelo

Analisando se o *cluster* fica acima ou abaixo do valor médio de cada categoria de interesse, ou analisando os valores máximo e mínimo de cada categoria, podemos notar determinados comportamentos que diferenciam os grupos

```
> teen_clusters$centers
```

	basketball	football	soccer	softball	volleyball	swimming
1	0.16001227	0.2364174	0.10385512	0.07232021	0.18897158	0.23970234
2	-0.09195886	0.0652625	-0.09932124	-0.01739428	-0.06219308	0.03339844
3	0.52755083	0.4873480	0.29778605	0.37178877	0.37986175	0.29628671
4	0.34081039	0.3593965	0.12722250	0.16384661	0.11032200	0.26943332
5	-0.16695523	-0.1641499	-0.09033520	-0.11367669	-0.11682181	-0.10595448

	cheerleading	baseball	tennis	sports	cute	sex
1	0.3931445	0.02993479	0.13532387	0.10257837	0.37884271	0.020042068
2	-0.1101103	-0.11487510	0.04062204	-0.09899231	-0.03265037	-0.042486141
3	0.3303485	0.35231971	0.14057808	0.32967130	0.54442929	0.002913623
4	0.1856664	0.27527088	0.10980958	0.79711920	0.47866008	2.028471066
5	-0.1136077	-0.10918483	-0.05097057	-0.13135334	-0.18878627	-0.097928345

	sexy	hot	kissed	dance	band	marching	music
1	0.11740551	0.41389104	0.06787768	0.22780899	-0.10257102	-0.10942590	0.1378306
2	-0.04329091	-0.03812345	-0.04554933	0.04573186	4.06726666	5.25757242	0.4981238
3	0.24040196	0.38551819	-0.03356121	0.45662534	-0.02120728	-0.10880541	0.2844999
4	0.51266080	0.31708549	2.97973077	0.45535061	0.38053621	-0.02014608	1.1367885
5	-0.09501817	-0.13810894	-0.13535855	-0.15932739	-0.12167214	-0.11098063	-0.1532006

## Passo 4: Analisando o modelo

A tabela abaixo mostra o interesse dominante em cada *cluster*. Observe que o *cluster* 5 não possui nenhuma categoria de interesse que predominante.

Cluster 1 (N = 3,376)	Cluster 2 (N = 601)	Cluster 3 (N = 1,036)	Cluster 4 (N = 3,279)	Cluster 5 (N = 21,708)
swimming cheerleading cute sexy hot dance dress hair mall hollister abercrombie shopping clothes	band marching music rock	sports sex sexy hot kissed dance music band die death drunk drugs	basketball football soccer softball volleyball baseball sports god church Jesus bible	???

## Passo 4: Analisando o modelo

Colocando a identificação dos *clusters* na sequência de dados original (*teens*)

```
> teens$cluster <- teen_clusters$cluster
```

```
> teens[1:5, c("cluster", "gender", "age", "friends")]
```

	cluster	gender	age	friends
1	5	M	18.982	7
2	3	F	18.801	0
3	5	M	18.335	69
4	5	F	18.875	0
5	4	<NA>	18.995	10

## Passo 4: Analisando o modelo

```
> aggregate(data = teens, age ~ cluster, mean)
```

	cluster	age
1	1	16.86497
2	2	17.39037
3	3	17.07656
4	4	17.11957
5	5	17.29849

```
> aggregate(data = teens, female ~ cluster, mean)
```

	cluster	female
1	1	0.8381171
2	2	0.7250000
3	3	0.8378198
4	4	0.8027079
5	5	0.6994515

```
> aggregate(data = teens, friends ~ cluster, mean)
```

	cluster	friends
1	1	41.43054
2	2	32.57333
3	3	37.16185
4	4	30.50290
5	5	27.70052