

Análise de Perfil de Trabalho

Comitê de Avaliação de Projetos.

11/09/2019

Introdução

Com o intuito de criar ações específicas para cada petiano, respeitando suas características, o Comitê de Análise de Projetos (C.A.P.) resolveu analisar os dados dos petianos atuais do PET Engenharia Química UFC por meio do teste disponibilizado pela Fundação Estudar: [Teste de Estilos de Trabalho](#). Esse teste nos informa a pontuação de cada petiano em 8 características, sendo elas:

Agilidade: *Preferência por ambientes que buscam ação e velocidade, onde a calma é menos valorizada.*

Agressividade: *Preferência por um ambiente que valorize a competição e foco.*

Atenção aos Detalhes: *Preferência por ambientes analíticos, onde se preste atenção em tudo que está acontecendo e onde a qualidade seja valorizada.*

Ênfase em Recompensa: *Preferência por ambientes onde existem muitas oportunidades de crescimento e de ganho para quem está disposto a fazer o que é preciso para obtê-los.*

Estabilidade: *Preferência por ambientes organizados e onde as coisas são bem definidas.*

Informalidade: *Preferência por ambientes informais e livres, com poucas regras.*

Orientação para Resultados: *Preferência por ambientes voltados para execução, onde o que importa é o que foi feito e onde fazer as coisas de forma diferente é menos valorizado.*

Trabalho em Equipe: *Preferência por ambientes onde as decisões e responsabilidades são compartilhadas e cujo foco seja menos no individual e mais na equipe.*

Sumário

1. Procedimentos: Análises que foram feitas e como foram feitas.
2. Preparação: Ajustes iniciais na planilha.
3. Análises: Análises introdutórias.
 - K-Means: Definição do número de clusters, aplicação e interpretação.

- Análise de Componentes Principais: Aplicação e interpretação da ACP.
- PERMANOVA: Explicação, aplicação e interpretação.

Procedimentos

Para encontrarmos os grupos formados com base no perfil de trabalho, aplicamos uma **Análise de Clusterização (K-Means)**. Contudo, essa análise possui um problema por não encontrar o número de clusters, apenas acha as pessoas que formam os cluster com base no número de clusters pré-estabelecidos. Para encontrarmos o número ótimo de clusters, podemos realizar um Método da Silhueta Média. Após isso, conseguimos encontrar, com os grupos já formados, quais características mais estão relacionadas com cada grupo.

Contudo, precisamos saber quais variáveis estão correlacionadas pois esse efeito pode interferir na análise dos resultados. Podemos pensar da seguinte forma: se duas variáveis estão muito correlacionadas, ao incluímos ela, estamos “duplicando” sua informação.

Contudo, precisamos saber quais variáveis estão correlacionadas pois esse efeito pode interferir na análise dos resultados. Podemos pensar da seguinte forma: se duas variáveis estão muito correlacionadas, ao incluímos ela, estamos “duplicando” sua informação.

Após isso, podemos realizar uma **Análise de Componentes Principais (ACP)** para verificar quais características de perfil de trabalho estão mais relacionadas a cada grupo. Por último, foi comparado o grupo de homens e mulheres, assim como os grupos formado pelos anos de curso de cada integrante do PET. As validações de possíveis afirmações foram testada por meio da PERMANOVA.

Preparação

```
library(ggplot2)
library(gridExtra)
library(DT)
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-5
```

```
library(mvnormtest)
library(MVN)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

## sROC 0.1-2 loaded

library(mvoutlier)

## Loading required package: sgeostat

library(factoextra)

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
https://goo.gl/13EFCZ
```

Inicialmente, visualizando a tabela

```
pet <- read.table('_txt/petianos.txt', h = T, sep= '\t')
pet <- pet[1:15,];datatable(pet)
```

Show entries Search:

	Petiano	Agilidade	Agressividade	Detalhes	Recompensa	Estabilidade	Informalidade	Resultado	Equipe
1	Caio	3.8	1.7	4.6	6.9	5.3	2.9	6.7	6.9
2	Fabricio	4.2	2.1	4.6	3.8	5.8	4.2	5	4.4
3	GabrielaBr	8.7	2.5	6.7	5.6	4.5	1.3	4.6	3.8
4	GabrielaBa	7.5	2.9	6.3	7.5	4.5	2.1	5	5.6
5	Gerson	6.3	2.5	7.1	5.6	5.5	1.7	4.2	5.6
6	Larissa	5.4	4.2	5	6.9	8.3	0.8	5	3.8
7	LucasA	5	1.7	4.6	6.3	6.8	6.3	5	6.3
8	LucasB	5.4	2.5	7.1	3.8	7.3	2.5	4.6	7.8
9	Mariana	5.4	1.7	6.3	4.4	4.8	3.8	5.8	3.4
10	Matheus	3.8	6.3	7.5	5	5.5	2.9	3.3	3.8

Showing 1 to 10 of 15 entries Previous 2 Next

Podemos observar que a primeira coluna são os nomes das linhas. Podemos armazenar a primeira coluna da tabela (que são os nomes) e atribuí-las para os nomes das linhas da tabela.

```
row.names(pet) # nomes iniciais das linhas

## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13"
## [14] "14"
## [15] "15"

info <- pet[,1];info # atribuindo para o objeto 'info' os nomes dos
petianos
```

```
## [1] Caio      Fabrício   GabrielaBr GabrielaBa Gerson
## [6] Larissa    LucasA    LucasB     Mariana    Matheus
## [11] Mirela     Sarah     Thiago     Vitor      Vitoria
## 16 Levels: Caio Fabrício GabrielaBa GabrielaBr Gerson Larissa ...
Vitoria

row.names(pet) <- info
row.names(pet)

## [1] "Caio"      "Fabrício"   "GabrielaBr" "GabrielaBa" "Gerson"
## [6] "Larissa"    "LucasA"     "LucasB"      "Mariana"     "Matheus"
## [11] "Mirela"     "Sarah"      "Thiago"      "Vitor"       "Vitoria"
```

Temos agora duas colunas com os nomes, mas a primeira coluna da planilha ainda continua sendo os nomes dos petianos. Como já atribuímos os nomes das linhas, podemos eliminar a primeira coluna

```
pet <- pet[, -1]; datatable(pet)
```

Show entries Search:

	Agilidade	Agressividade	Detalhes	Recompensa	Estabilidade	Informalidade	Resultado	Equipe
Caio	3.8	1.7	4.6	6.9	5.3	2.9	6.7	6.9
Fabricio	4.2	2.1	4.6	3.8	5.8	4.2	5	4.4
GabrielaBr	8.7	2.5	6.7	5.6	4.5	1.3	4.6	3.8
GabrielaBa	7.5	2.9	6.3	7.5	4.5	2.1	5	5.6
Gerson	6.3	2.5	7.1	5.6	5.5	1.7	4.2	5.6
Larissa	5.4	4.2	5	6.9	8.3	0.8	5	3.8
LucasA	5	1.7	4.6	6.3	6.8	6.3	5	6.3
LucasB	5.4	2.5	7.1	3.8	7.3	2.5	4.6	7.8
Mariana	5.4	1.7	6.3	4.4	4.8	3.8	5.8	3.4
Matheus	3.8	6.3	7.5	5	5.5	2.9	3.3	3.8

Showing 1 to 10 of 15 entries Previous 2 Next

Análises

Analisaremos um pouco os dados obtidos:

Tentaremos analisar algumas informações através de gráficos de densidade e Box Plot, utilizando a função `geom_density` e a função `geom_boxplot`, respectivamente, do pacote **ggplot2**.

```
plot_density <- function(data, coluna, string) {
  ggplot(data = data) +
    geom_density(aes(coluna), fill = '#F0F8FF', alpha = 0.5) +
```

```

ggtitle(paste("Densidade", string, sep=" ")) +
ylab('Densidade') +
xlab(string) +
ylim(c(0,0.4)) +
xlim(c(0,10)) +
geom_vline(xintercept = c(min(coluna), max(coluna)), alpha = 0.5) +
theme_bw()
}

```

```

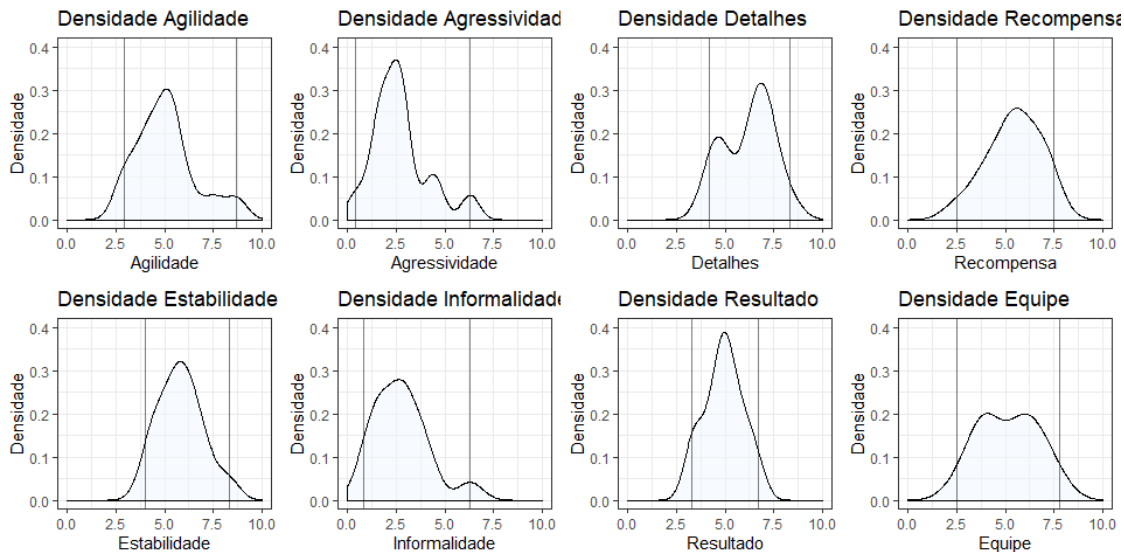
graf_agi <- plot_density(pet, pet$Agilidade, 'Agilidade')
graf_agress <- plot_density(pet, pet$Agressividade, 'Agressividade')
graf_det <- plot_density(pet, pet$Detalhes, 'Detalhes')
graf_recomp <- plot_density(pet, pet$Recompensa, 'Recompensa')
graf_est <- plot_density(pet, pet$Estabilidade, 'Estabilidade')
graf_infor <- plot_density(pet, pet$Informalidade, 'Informalidade')
graf_res <- plot_density(pet, pet$Resultado, 'Resultado')
graf_eq <- plot_density(pet, pet$Equipe, 'Equipe')

```

```

grid.arrange(graf_agi, graf_agress, graf_det, graf_recomp, graf_est,
             graf_infor, graf_res, graf_eq, ncol = 4, nrow = 2)

```



OBS: É interessante ressaltar que não precisamos padronizar, ainda, os dados para o gráfico de densidade, pois ele leva em conta apenas a frequência. Caso padronizássemos e plotássemos os gráficos de densidade, teríamos o mesmo resultado.

```

tema <- theme(plot.title = element_text(hjust = 0.5),
              panel.background = element_rect(fill = "white", colour =
"grey50"),
              panel.border = element_rect(linetype = 'solid' , fill =
NA))

plot_boxplot <- function(data, coluna, string){
  ggplot(data = data) +
    geom_boxplot(aes(y = coluna)) +

```

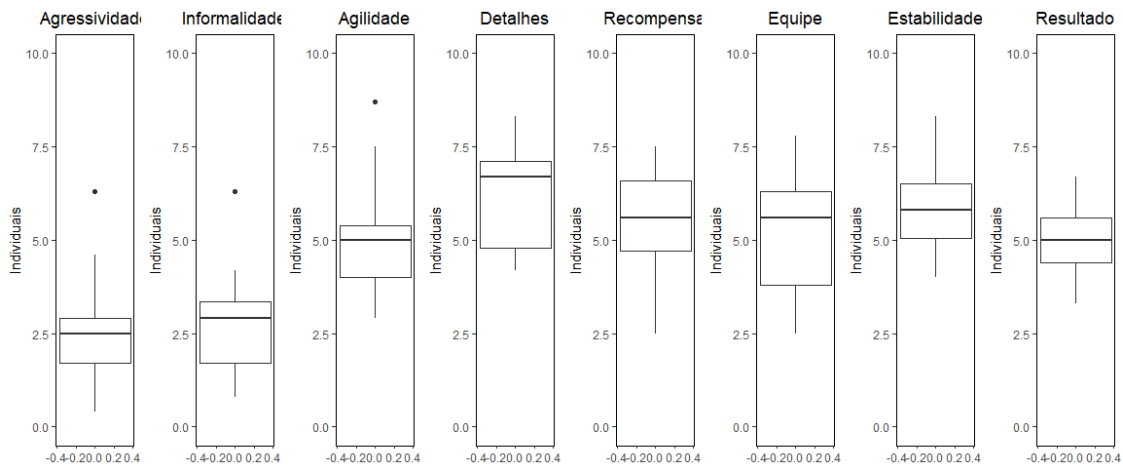
```

ylab('Individuais') +
ylim(0,10) +
ggtitle(string) +
tema
}

c1 <- plot_boxplot(pet, pet$Agressividade, 'Agressividade')
c2 <- plot_boxplot(pet, pet$Informalidade, 'Informalidade')
c3 <- plot_boxplot(pet, pet$Agilidade, 'Agilidade')
c4 <- plot_boxplot(pet, pet$Detalhes, 'Detalhes')
c5 <- plot_boxplot(pet, pet$Recompensa, 'Recompensa')
c6 <- plot_boxplot(pet, pet$Equipe, 'Equipe')
c7 <- plot_boxplot(pet, pet$Estabilidade, 'Estabilidade')
c8 <- plot_boxplot(pet, pet$Resultado, 'Resultado')

grid.arrange(c1,c2,c3,c4,c5,c6,c7,c8,
             nrow = 1, ncol = 8)

```



RH - Concluimos que:

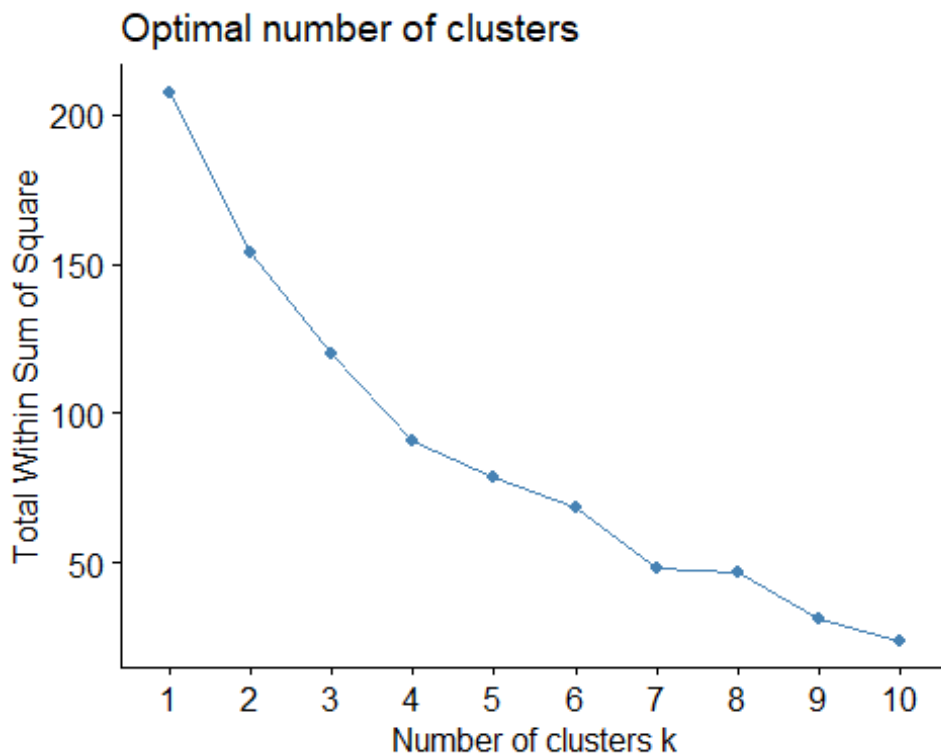
1. Conseguimos observar que **Atenção aos Detalhes** é a característica que apresenta maiores valores no grupo.
2. **Trabalho em Equipe** é a característica que possui maior variação no grupo. Em contrapartida, **Orientação para Resultados** é a que possui menos variação.
3. Aparentemente, temos dois “grupos” de características: (Agressividade, Informalidade) e (Agilidade, Detalhes, Recompensa, Equipe, Estabilidade, Resultado).
4. **Agressividade, Informalidade e Agilidade** possuem um outlier cada.

K-Means

No R, é extremamente simples realizar uma K-Means. O comando é `kmeans`. Contudo, antes, precisamos determinar o melhor número de clusters. Para isso há diversos métodos.

Um deles é o **Método de Elbow**. O código já está presente na biblioteca `factoextra`, por meio da função `fviz_nbclust`.

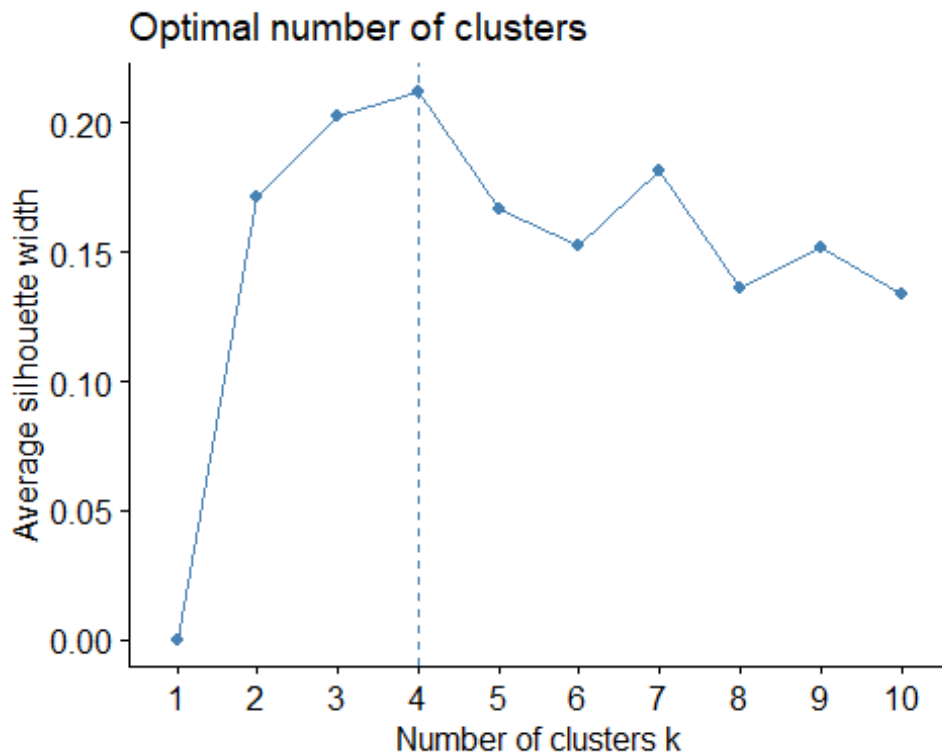
```
fviz_nbclust(pet, kmeans, method = "wss")
```



Contudo, esse método se mostra complicado de prever um número de clusters. Para isso podemos utilizar outro método.

Outra forma, é através da **Average Silhouette Method**, ou Método da Silhueta Média (tradução livre). A abordagem de silhueta média mede a qualidade de um agrupamento. Isto é, determina o quão bem cada objeto está dentro de seu cluster. Uma largura média alta da silhueta indica um bom agrupamento. O método de silhueta média calcula a silhueta média das observações para diferentes valores de k . O número ideal de clusters k é aquele que maximiza a silhueta média em uma faixa de valores. Também já está implementado na função `fviz_nbclust`, devemos apenas mudar o método.

```
fviz_nbclust(pet, kmeans, method = "silhouette")
```



O gráfico nos mostra que devemos escolher quatro clusters. Podemos fazer isso utilizando a função `kmeans` e plotamos com a função `fviz_nbclust`.

```
km_pet <- kmeans(pet,4,iter.max=100000);km_pet

## K-means clustering with 4 clusters of sizes 4, 5, 3, 3
##
## Cluster means:
##   Agilidade Agressividade Detalhes Recompensa Estabilidade
## Informalidade
## 1    6.975      2.700000 7.100000    6.075000    4.625000
## 2.000000
## 2    4.260      1.520000 4.860000    5.280000    5.840000
## 4.020000
## 3    4.600      5.033333 6.400000    6.266667    6.766667
## 1.800000
## 4    4.300      2.233333 6.966667    3.966667    6.433333
## 2.533333
##   Resultado Equipe
## 1  4.900000 5.150000
## 2  5.580000 5.080000
## 3  4.866667 3.366667
## 4  3.900000 7.000000
##
## Clustering vector:
##      Caio   Fabrício GabrielaBr GabrielaBa      Gerson
## Larissa
##      2         2         1         1         1
```

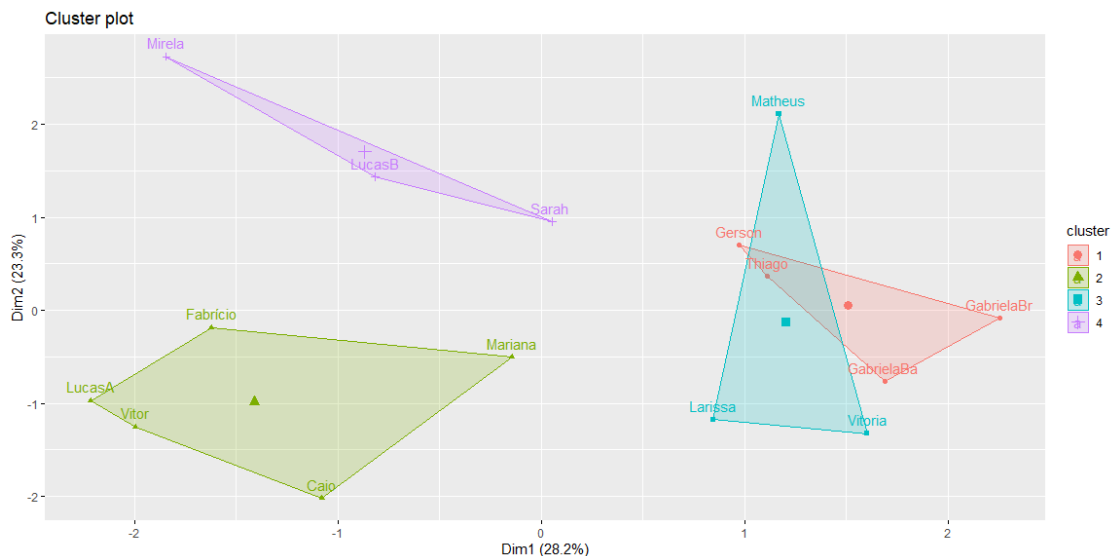


```

3
##      LucasA      LucasB      Mariana      Matheus      Mirela      Sarah
##          2          4          2          3          4
4
##      Thiago      Vitor      Vitoria
##          1          2          3
##
## Within cluster sum of squares by cluster:
## [1] 17.71250 36.09600 21.33333 15.85333
## (between_SS / total_SS = 56.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"

fviz_cluster(km_pet, data = pet)

```



O gráfico acima nos mostra o padrão geral do grupo atual de petianos. Entretanto, podemos nos questionar quais características esses grupos possuem. Isso será visto posteriormente com a PCA.

RH - Concluimos que:

- (Mirela, Lucas B. e Sarah), (Larissa, Vitória e Matheus), (Gerson, Gabriela Ba. e Gabriela Br.) e (Fabrício, Mariana, Lucas A., Vitor e Caio) formam, de fato, clusters.
- Há um grupo com grande variação: Fabrício, Mariana, Lucas A., Vitor e Caio

Devemos analisar agora as correlações entre as variáveis. Utilizamos o Coeficiente de Correlação de Pearson, indicado para variáveis quantitativas e, preferencialmente, contínuas.

```
cor(pet[, -9], method = 'pearson')
```

```
##           Agilidade Agressividade  Detalhes  Recompensa
## Agilidade    1.00000000    0.04004402  0.2958389  0.39375372
## Agressividade 0.04004402    1.00000000  0.4384319  0.14099548
## Detalhes     0.29583886    0.43843190  1.00000000 -0.16042913
## Recompensa    0.39375372    0.14099548 -0.1604291  1.00000000
## Estabilidade -0.36424019    0.08751955 -0.4179836 -0.02535771
## Informalidade -0.37922292   -0.26717264 -0.3552556 -0.34159949
## Resultado     0.03255388   -0.25784227 -0.3681399  0.48720167
## Equipe       -0.13708601   -0.39356936  0.0528093 -0.22687247
##           Estabilidade Informalidade  Resultado  Equipe
## Agilidade    -0.36424019   -0.37922292  0.03255388 -0.13708601
## Agressividade  0.08751955   -0.26717264 -0.25784227 -0.39356936
## Detalhes     -0.41798362   -0.35525558 -0.36813989  0.05280930
## Recompensa    -0.02535771   -0.34159949  0.48720167 -0.22687247
## Estabilidade  1.00000000   -0.03533155 -0.11025725  0.05877571
## Informalidade -0.03533155    1.00000000  0.05656687  0.24072664
## Resultado    -0.11025725    0.05656687  1.00000000 -0.22047622
## Equipe       0.05877571    0.24072664 -0.22047622  1.00000000
```

Com exceção da comparação de um variável com a mesma variável, não temos um valor relativamente alto para podermos considerar que há correlação. Assim, optamos por continuar com todas as variáveis.

Análise de Componentes Principais (ACP)

Agora, veremos as características que esses grupos possuem. Para isso, plotaremos uma PCA e veremos os loads que mais se aproximam dos clusters.

Contudo, para delimitarmos visualmente na PCA os grupos, criaremos uma coluna extra na tabela pet, contendo a que grupo cada pessoa pertence. Em seguida, poderemos utilizar o comando `ordihull` para delimitarmos os grupos.

```
km_pet$cluster
```

```
##           Caio    Fabrício  GabrielaBr  GabrielaBa    Gerson
Larissa
##           2         2             1             1             1
3
##           LucasA    LucasB    Mariana    Matheus    Mirela    Sarah
##           2         4             2             3             4
4
##           Thiago    Vitor    Vitoria
##           1         2             3
```

```
pet$Grupo <- factor(c(4,4,2,2,2,1,4,3,4,1,3,3,2,4,1))
datatable(pet)
```

Show entries Search:

	Agilidade	Agressividade	Detalhes	Recompensa	Estabilidade	Informalidade	Resultado	Equipe	Grupo
Caio	3.8	1.7	4.6	6.9	5.3	2.9	6.7	6.9	4
Fabricao	4.2	2.1	4.6	3.8	5.8	4.2	5	4.4	4
GabrielaBr	8.7	2.5	6.7	5.6	4.5	1.3	4.6	3.8	2
GabrielaBa	7.5	2.9	6.3	7.5	4.5	2.1	5	5.6	2
Gerson	6.3	2.5	7.1	5.6	5.5	1.7	4.2	5.6	2
Larissa	5.4	4.2	5	6.9	8.3	0.8	5	3.8	1
LucasA	5	1.7	4.6	6.3	6.8	6.3	5	6.3	4
LucasB	5.4	2.5	7.1	3.8	7.3	2.5	4.6	7.8	3
Mariana	5.4	1.7	6.3	4.4	4.8	3.8	5.8	3.4	4
Matheus	3.8	6.3	7.5	5	5.5	2.9	3.3	3.8	1

Showing 1 to 10 of 15 entries Previous 2 Next

```
dim(pet)

## [1] 15 9

pca <- prcomp(pet[, -9])
summary(pca)

## Importance of components:
##
##               PC1      PC2      PC3      PC4      PC5      PC6
PC7
## Standard deviation      2.1609 1.6991 1.6586 1.3068 1.09094 0.98773
0.62730
## Proportion of Variance 0.3152 0.1949 0.1857 0.1153 0.08033 0.06585
0.02656
## Cumulative Proportion 0.3152 0.5101 0.6957 0.8110 0.89136 0.95721
0.98377
##
##               PC8
## Standard deviation      0.49029
## Proportion of Variance 0.01623
## Cumulative Proportion 1.00000

pca_scores <- pca$x
loads <- t(cor(pca_scores, pet[, -9]))

par(mfrow = c(1,3))

plot(pca_scores[,1:2], xlim = c(-4,4), ylim = c(-4,4),
     xlab = 'PC1 (31.52%)', ylab = 'PC2 (19.49%)',
     main = 'PCA - Perfil de Trabalho dos Petianos')
abline(v=0,h=0,lty=2)
```

```

arrows(0,0,loads[,1]*3,loads[,2]*3)
text(loads[,1:2]*3.6,rownames(loads))
ordihull(pca$x[,c(1,2)], pet$Grupo, draw = 'polygon', show.groups = '1',
alpha = 0.3, col = 3)
ordihull(pca$x[,c(1,2)], pet$Grupo, draw = 'polygon', show.groups = '2',
alpha = 0.3, col = 4)
ordihull(pca$x[,c(1,2)], pet$Grupo, draw = 'polygon', show.groups = '3',
alpha = 0.3, col = 5)
ordihull(pca$x[,c(1,2)], pet$Grupo, draw = 'polygon', show.groups = '4',
alpha = 0.3, col = 6)
legend('bottomright', fill = c(3:6), legend = c('Grupo 1', 'Grupo 2',
'Grupo 3', 'Grupo 4'))

```

```

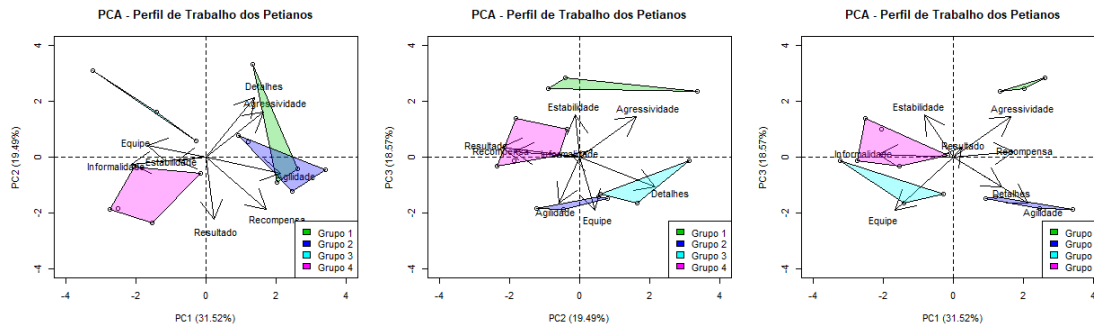
plot(pca_scores[,2:3], xlim = c(-4,4), ylim = c(-4,4),
      xlab = 'PC2 (19.49%)', ylab = 'PC3 (18.57%)',
      main = 'PCA - Perfil de Trabalho dos Petianos')
abline(v=0,h=0,lty=2)
arrows(0,0,loads[,2]*3,loads[,3]*3)
text(loads[,2:3]*3.6,rownames(loads))
ordihull(pca$x[,c(2,3)], pet$Grupo, draw = 'polygon', show.groups = '1',
alpha = 0.3, col = 3)
ordihull(pca$x[,c(2,3)], pet$Grupo, draw = 'polygon', show.groups = '2',
alpha = 0.3, col = 4)
ordihull(pca$x[,c(2,3)], pet$Grupo, draw = 'polygon', show.groups = '3',
alpha = 0.3, col = 5)
ordihull(pca$x[,c(2,3)], pet$Grupo, draw = 'polygon', show.groups = '4',
alpha = 0.3, col = 6)
legend('bottomright', fill = c(3:6), legend = c('Grupo 1', 'Grupo 2',
'Grupo 3', 'Grupo 4'))

```

```

plot(pca_scores[,1],pca_scores[,3], xlim = c(-4,4), ylim = c(-4,4),
      xlab = 'PC1 (31.52%)', ylab = 'PC3 (18.57%)',
      main = 'PCA - Perfil de Trabalho dos Petianos')
abline(v=0,h=0,lty=2)
arrows(0,0,loads[,1]*3,loads[,3]*3)
text(loads[,1]*3.6,loads[,3]*3.6,rownames(loads))
ordihull(pca$x[,c(1,3)], pet$Grupo, draw = 'polygon', show.groups = '1',
alpha = 0.3, col = 3)
ordihull(pca$x[,c(1,3)], pet$Grupo, draw = 'polygon', show.groups = '2',
alpha = 0.3, col = 4)
ordihull(pca$x[,c(1,3)], pet$Grupo, draw = 'polygon', show.groups = '3',
alpha = 0.3, col = 5)
ordihull(pca$x[,c(1,3)], pet$Grupo, draw = 'polygon', show.groups = '4',
alpha = 0.3, col = 6)
legend('bottomright', fill = c(3:6), legend = c('Grupo 1', 'Grupo 2',
'Grupo 3', 'Grupo 4'))

```



Seria útil, também, analisarmos os grupos pelo sexo, e observar se há maior discrepância em alguns deles. Para isso, criaremos uma nova coluna na planilha pet2, com o sexo de cada petiano.

```
par(mfrow = c(1,3))
```

```
pet$Sexo <- factor(c('H', 'H', 'M', 'M', 'H', 'M',
                     'H', 'H', 'M', 'H', 'M', 'M',
                     'H', 'H', 'M'))
```

```
plot(pca_scores[,1:2], xlim = c(-4,4), ylim = c(-4,4),
     xlab = 'PC1 (31.52%)', ylab = 'PC2 (19.49%)',
     main = 'PCA - Perfil de Trabalho dos Petianos')
abline(v=0,h=0,lty=2)
arrows(0,0,loads[,1]*3,loads[,2]*3)
text(loads[,1:2]*3.6,rownames(loads))
ordihull(pca$x[,1:2], pet$Sexo, draw = 'polygon', show.groups = 'H',
alpha = 0.2, col = 6)
ordihull(pca$x[,1:2], pet$Sexo, draw = 'polygon', show.groups = 'M',
alpha = 0.2, col = 7)
legend('topright', fill = c(6:7), legend = c('Homem', 'Mulher'))
```

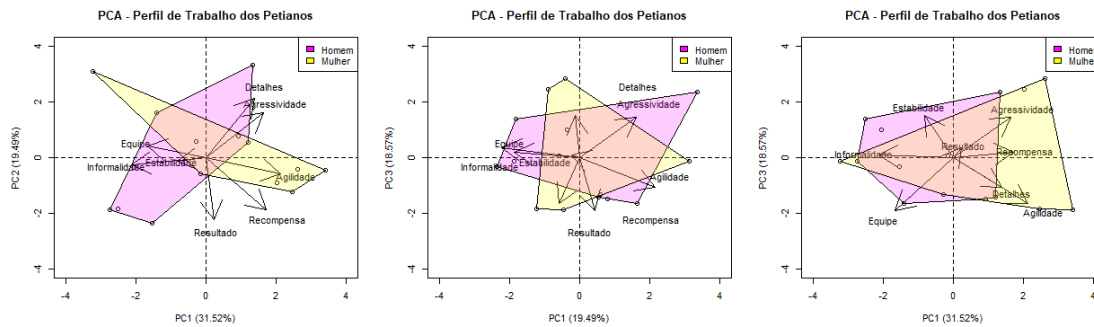
```
plot(pca_scores[,2:3], xlim = c(-4,4), ylim = c(-4,4),
     xlab = 'PC1 (19.49%)', ylab = 'PC3 (18.57%)',
     main = 'PCA - Perfil de Trabalho dos Petianos')
abline(v=0,h=0,lty=2)
arrows(0,0,loads[,2]*3,loads[,3]*3)
text(loads[,1:2]*3.6,rownames(loads))
ordihull(pca$x[,2:3], pet$Sexo, draw = 'polygon', show.groups = 'H',
alpha = 0.2, col = 6)
ordihull(pca$x[,2:3], pet$Sexo, draw = 'polygon', show.groups = 'M',
alpha = 0.2, col = 7)
legend('topright', fill = c(6:7), legend = c('Homem', 'Mulher'))
```

```
plot(pca_scores[,c(1,3)], xlim = c(-4,4), ylim = c(-4,4),
     xlab = 'PC1 (31.52%)', ylab = 'PC3 (18.57%)',
     main = 'PCA - Perfil de Trabalho dos Petianos')
```

```

abline(v=0,h=0,lty=2)
arrows(0,0,loads[,1]*3,loads[,3]*3)
text(loads[,c(1,3)]*3.6,rownames(loads))
ordihull(pca$x[,c(1,3)], pet$Sexo, draw = 'polygon', show.groups = 'H',
alpha = 0.2, col = 6)
ordihull(pca$x[,c(1,3)], pet$Sexo, draw = 'polygon', show.groups = 'M',
alpha = 0.2, col = 7)
legend('topright', fill = c(6:7), legend = c('Homem', 'Mulher'))

```



Aparentemente, os grupo Homem e Mulher não diferem com base no perfil de trabalho.

Podemos ver, também, quanto ao ano de cada petiano.

```

pet$Ano <- factor(c(2,3,1,2,1,2,3,2,1,3,3,1,3,3,1))

par(mfrow = c(1,3))

plot(pca_scores[,1:2], xlim = c(-4,4), ylim = c(-4,4),
      xlab = 'PC1 (31.52%)', ylab = 'PC2 (19.49%)',
      main = 'PCA - Perfil de Trabalho dos Petianos / Por Ano')
abline(v=0,h=0,lty=2)
arrows(0,0,loads[,1]*3,loads[,2]*3)
text(loads[,1:2]*3.6,rownames(loads))
ordihull(pca$x[,1:2], pet$Ano, draw = 'polygon', show.groups = '1', alpha
= 0.25, col = 9)
ordihull(pca$x[,1:2], pet$Ano, draw = 'polygon', show.groups = '2', alpha
= 0.2, col = 10)
ordihull(pca$x[,1:2], pet$Ano, draw = 'polygon', show.groups = '3', alpha
= 0.2, col = 11)
legend('topright', fill = c(9:11), legend = c('1º Ano', '2º Ano', '3º
Ano'))

plot(pca_scores[,2:3], xlim = c(-4,4), ylim = c(-4,4),
      xlab = 'PC2 (19.49%)', ylab = 'PC3 (18.57%)',
      main = 'PCA - Perfil de Trabalho dos Petianos / Por Ano')
abline(v=0,h=0,lty=2)
arrows(0,0,loads[,2]*3,loads[,3]*3)
text(loads[,2:3]*3.6,rownames(loads))
ordihull(pca$x[,2:3], pet$Ano, draw = 'polygon', show.groups = '1', alpha

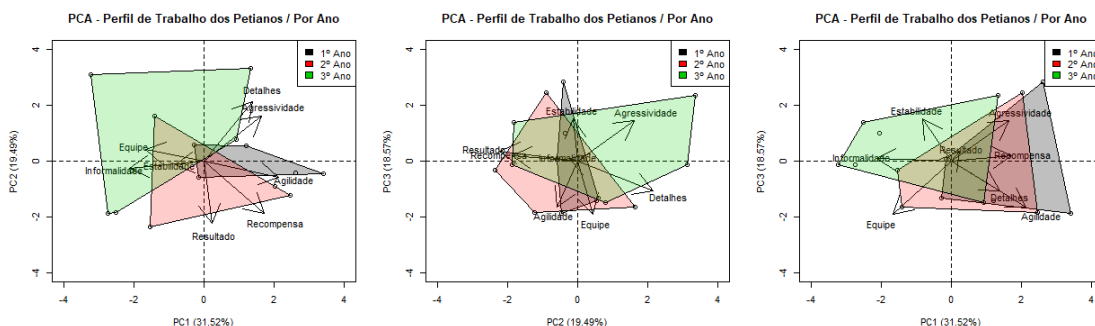
```

```

= 0.25, col = 9)
ordihull(pca$x[,2:3], pet$Ano, draw = 'polygon', show.groups = '2', alpha
= 0.2, col = 10)
ordihull(pca$x[,2:3], pet$Ano, draw = 'polygon', show.groups = '3', alpha
= 0.2, col = 11)
legend('topright', fill = c(9:11), legend = c('1º Ano', '2º Ano', '3º
Ano'))

plot(pca_scores[,c(1,3)], xlim = c(-4,4), ylim = c(-4,4),
      xlab = 'PC1 (31.52%)', ylab = 'PC3 (18.57%)',
      main = 'PCA - Perfil de Trabalho dos Petianos / Por Ano')
abline(v=0,h=0,lty=2)
arrows(0,0,loads[,1]*3,loads[,3]*3)
text(loads[,c(1,3)]*3.6,rownames(loads))
ordihull(pca$x[,c(1,3)], pet$Ano, draw = 'polygon', show.groups = '1',
alpha = 0.25, col = 9)
ordihull(pca$x[,c(1,3)], pet$Ano, draw = 'polygon', show.groups = '2',
alpha = 0.2, col = 10)
ordihull(pca$x[,c(1,3)], pet$Ano, draw = 'polygon', show.groups = '3',
alpha = 0.2, col = 11)
legend('topright', fill = c(9:11), legend = c('1º Ano', '2º Ano', '3º
Ano'))

```



RH - Concluimos que:

- O grupo do 3º Ano, no PC1, apresenta maiores valores de Equipe, Informalidade e Estabilidade. No PC2, a maioria dos membros, apresenta maiores valores em Detalhes e Agressividade enquanto o restante apresenta valores altos de Recompensa e Resultado. Por último, no PC3, alguns pontos com valores altos em Agressividade e Estabilidade enquanto um ponto com altos valores em Detalhes, Agilidade e Equipe.
- O grupo do 2º Ano, no PC1, metade dos membros apresenta altos valores de Recompensa, Agilidade, Detalhes e Agressividade, enquanto a outra metade apresenta altos valores de Equipe, Informalidade e Estabilidade. No PC2, a maioria dos membros apresenta altos valores em Resultado e Recompensa. Já no PC3, um ponto com alto valor em Estabilidade enquanto a maioria com valores altos em Agilidade, Equipe e Detalhes.

- O grupo do 1º Ano, no PC1, a maioria apresenta altos valores de Agilidade, Recompensa, Detalhes e Agressividade; a outra parte desse grupo se assemelha ao 3º Ano e apresentam maiores valores de Equipe, Informalidade e Estabilidade. No PC2, apresentam todos apresentam valores muito baixos quando comparado aos outros grupos. No PC3, percebemos dois pontos que puxam para maiores valores de Agilidade e Equipe.

Para podermos afirmar se os grupos diferem ou não, de fato, podemos realizar uma outra análise: PERMANOVA.

PERMANOVA

PERMANOVA é uma ANOVA multivariada permutacional. Uma PERMANOVA permite determinar estatisticamente se os centros (centróides) do grupo de amostras para um grupo difere do centro das amostras de outro grupo. Assim como uma ANOVA, podemos dizer se o valor médio difere entre os grupos de tratamento, de modo que a PERMANOVA permite determinar se os centróides diferem nas ordenações. Em outras palavras, a PERMANOVA lhe diz as chances de observar a ordenação que você observou, ou uma com menos sobreposição, não havendo diferença entre um grupo e outro.

OBS: Alguns pressupostos da análise não foram expostos nesse trabalho.

Primeiro, precisamos calcular os índices de dissimilaridade com a função `vegdist` (pacote **vegan**), passando a distância adequado para os dados. Como temos variáveis contínuas e não há presença de zeros, optamos pela distância euclidiana.

```
dist <- vegdist(pet[, -c(9:12)], method='eucl');dist
```

```
##           Caio Fabrício GabrielaBr GabrielaBa      Gerson  Larissa
## Fabrício    4.583667
## GabrielaBr  6.926038 6.219325
## GabrielaBa  4.923413 5.970762  3.067572
## Gerson      4.924429 4.741308  3.237283  2.765863
## Larissa     5.905929 5.812917  5.762812  5.245951 4.725463
## LucasA      4.342810 4.008740  7.478636  5.953150 5.748913 6.758698
## LucasB      5.332917 4.970915  6.290469  5.648008 3.595831 6.079474
## Mariana     5.081338 2.765863  4.590207  4.890808 3.978693 6.019967
## Matheus     7.374958 5.716642  6.685806  6.380439 5.167204 5.654202
## Mirela      6.235383 4.215448  7.950472  7.485987 5.352569 7.548510
## Sarah       4.534314 4.877499  5.248809  4.476606 2.321637 5.327288
## Thiago      4.854894 5.100980  4.593474  3.681032 2.942788 6.407027
## Vitor       3.956008 2.918904  7.226341  6.656576 5.482700 5.754129
## Vitoria     5.988322 5.698245  5.625833  5.223026 4.916299 3.334666
##           LucasA  LucasB  Mariana  Matheus  Mirela  Sarah
Thiago
## Fabrício
```



```

## GabrielaBr
## GabrielaBa
## Gerson
## Larissa
## LucasA
## LucasB      5.513620
## Mariana      5.095096 5.511806
## Matheus      7.422264 6.300000 5.764547
## Mirela      5.902542 3.760319 5.603570 5.470832
## Sarah      5.842944 3.367492 4.793746 5.955670 4.698936
## Thiago      6.025778 4.708503 3.629049 5.182663 5.583010 3.961060
## Vitor      4.834253 5.788782 4.191658 7.230491 5.519964 4.631414
6.125357
## Vitoria      7.121798 6.861487 4.880574 4.572745 7.626270 5.817216
5.032892
##          Vitor
## Fabrício
## GabrielaBr
## GabrielaBa
## Gerson
## Larissa
## LucasA
## LucasB
## Mariana
## Matheus
## Mirela
## Sarah
## Thiago
## Vitor
## Vitoria      6.020797

```

Com a matriz de distâncias, podemos realizar a PERMANOVA - função `adonis2` (novamente, pacote **vegan**).

```

permanova <- adonis2(dist~pet$Sexo, data=pet, permutations = 1000000,
method="euc");permanova

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 1e+06
##
## adonis2(formula = dist ~ pet$Sexo, data = pet, permutations = 1e+06,
method = "euc")
##          Df SumOfSqs      R2      F Pr(>F)
## pet$Sexo  1   14.122 0.06809 0.9498 0.4877
## Residual 13  193.285 0.93191
## Total    14  207.407 1.00000

permanova2 <- adonis2(dist~pet$Ano, data=pet, permutations = 1000000,
method="euc");permanova2

```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 1e+06
##
## adonis2(formula = dist ~ pet$Ano, data = pet, permutations = 1e+06,
method = "euc")
##           Df SumOfSqs      R2      F Pr(>F)
## pet$Ano    2   41.582 0.20049 1.5046  0.116
## Residual  12  165.824 0.79951
## Total     14  207.407 1.00000
```

O objeto nos dá muitas informações. Temos que $p > 0.05$ para ambos os casos, o que nos indica que devemos aceitar a hipótese nula: os grupos não diferem. Isto é, **homens e mulheres do PET Engenharia Química UFC não diferem em suas características e/ou não há diferença entre petianos de diferentes anos**. Além disso, o objeto nos informa o R^2 (correlação). Caso R^2 fosse 1, teríamos a formação de grupos perfeitos. Esse resultado nos diz, então, que há grupos bastantes **heterogêneos**, o que resulta em um R^2 baixo.
