

Análise de Cluster com PCA

Hugo Victor dos Santos Silva

2025-09-28

Contents

1	Introdução	2
2	Dados e Justificativas	2
3	Correlograma	2
3.1	Análise do Correlograma	5
4	PCA	5
4.1	Análise dos Resultados PCA	9
4.1.1	Resposta à Pergunta Principal	9
4.1.2	Interpretação do Scree Plot	9
4.1.3	Composição dos Componentes Principais	10
4.1.4	Relação com a Análise de Correlação Anterior	10
5	K-Médias	10
5.1	Análise dos Resultados K-Médias	16
5.1.1	Determinação do Número Ótimo de Clusters	16
5.1.2	Justificativa Gráfica	16
5.1.3	Resultados do Clustering	17
5.1.4	Interpretação dos Clusters	17
5.1.5	Validação dos Resultados	17
6	Comparação de Resultados	17
6.1	Comparação Fictícia	17
6.2	Análise Comparativa dos Resultados	24
6.2.1	Diferenças Encontradas	24
6.2.2	Ponderação sobre as Melhorias	24
7	Tableau Public	24

8 Evidências para a avaliação	25
8.1 O aluno apresentou um print mostrando o ambiente RStudio?	25
8.2 O aluno apresentou um print mostrando a versão do pacote rmarkdown?	26
8.3 O aluno apresentou um print mostrando a versão do pacote factoextra e FactoMineR instalada?	26
8.4 O aluno mostrou um printscreen da ferramenta Tableau?	27

1 Introdução

Este trabalho foi desenvolvido como parte da disciplina de **Visualização e Relatório de Segmentos** e seu principal objetivo é responder as perguntas a seguir:

2 Dados e Justificativas

1. Utilize a mesma base de dados escolhida no projeto da disciplina (PD) Análise de clusters. Lembre que ela necessita ter 4 (ou mais) variáveis de interesse, onde todas são numéricas (ou categóricas com a anuência do professor). Explique qual o motivo para a escolha dessa base e aponte os resultados esperados através da análise. Caso seja necessário mudar de base, justifique (a questão 5 irá pedir uma comparação e caso não tenha os resultados passados, eles deverão ser produzidos).
- Este trabalho não possui os dados da disciplina de Análise de Clusters, por isso, foi escolhida a base de dados [Wine Recognition Data](#)
- Esta base de dados contém 178 registros e 13 variáveis de interesse.
- Esta base de dados foi escolhida pois:
 1. **Tamanho ideal para aprendizado:** Com 178 amostras e 13 variáveis, é grande o suficiente para ser interessante, mas pequeno o suficiente para eu conseguir “entender” cada resultado.
 2. **Variáveis correlacionadas:** Descobri que muitas características químicas dos vinhos estão relacionadas entre si - perfeito para testar o PCA
 3. **Grupos naturais:** Os vinhos já vêm classificados em 3 tipos diferentes, o que me permite verificar se meus clusters “fazem sentido” comparando com a realidade.

3 Correlograma

2. Crie um correlograma (corrplot) para as variáveis dos problema e indique quais as variáveis são mais correlacionadas.

```
# Carregando os dados
wine_data <- read.csv("data/wine.data", header = FALSE)

colnames(wine_data) <- c(
  "Class",
  "Alcohol",
  "Malic_acid",
  "Ash",
  "Alcalinity_ash",
  "Magnesium",
  "Total_phenols",
```

```

    "Flavanoids",
    "Nonflavanoid_phenols",
    "Proanthocyanins",
    "Color_intensity",
    "Hue",
    "OD280_OD315",
    "Proline"
)

wine_numeric <- wine_data[, -1]

correlation_matrix <- cor(wine_numeric)

# Matriz de Correlação:
print(round(correlation_matrix, 3))

```

```

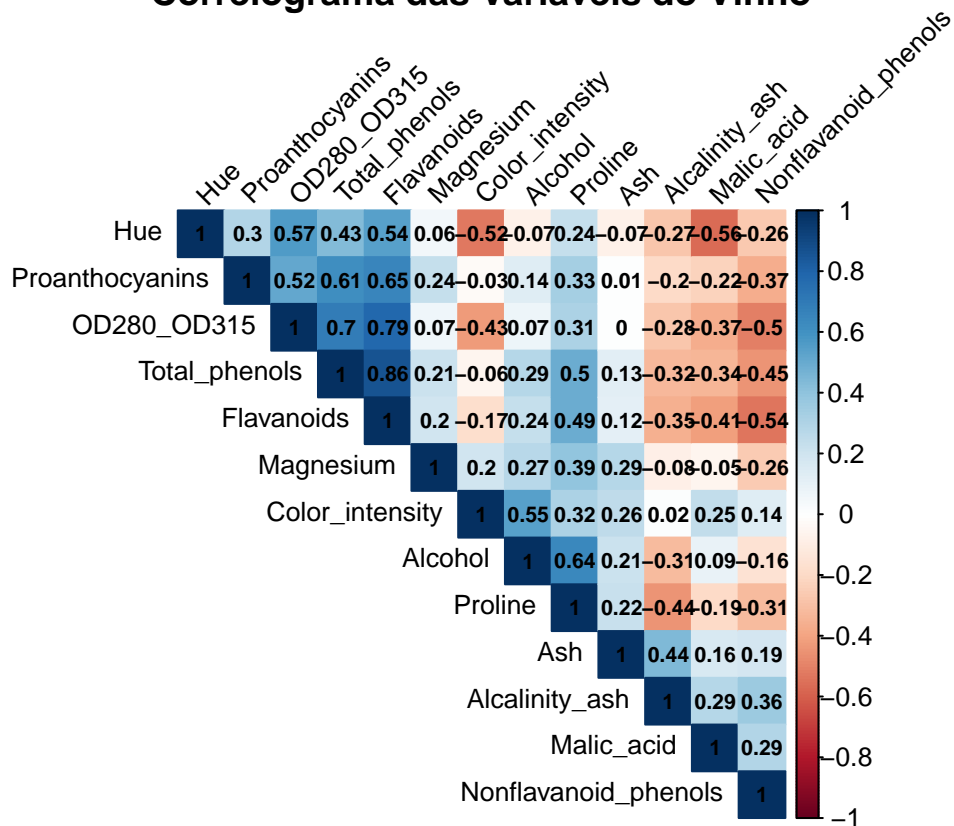
##           Alcohol Malic_acid      Ash Alcalinity_ash Magnesium
## Alcohol           1.000      0.094  0.212          -0.310      0.271
## Malic_acid         0.094      1.000  0.164           0.289     -0.055
## Ash                0.212      0.164  1.000           0.443      0.287
## Alcalinity_ash     -0.310      0.289  0.443           1.000     -0.083
## Magnesium          0.271     -0.055  0.287          -0.083      1.000
## Total_phenols       0.289     -0.335  0.129          -0.321      0.214
## Flavanoids          0.237     -0.411  0.115          -0.351      0.196
## Nonflavanoid_phenols -0.156      0.293  0.186           0.362     -0.256
## Proanthocyanins     0.137     -0.221  0.010          -0.197      0.236
## Color_intensity     0.546      0.249  0.259           0.019      0.200
## Hue                -0.072     -0.561 -0.075          -0.274      0.055
## OD280_OD315         0.072     -0.369  0.004          -0.277      0.066
## Proline             0.644     -0.192  0.224          -0.441      0.393
##
##           Total_phenols Flavanoids Nonflavanoid_phenols
## Alcohol              0.289      0.237          -0.156
## Malic_acid           -0.335     -0.411           0.293
## Ash                  0.129      0.115           0.186
## Alcalinity_ash       -0.321     -0.351           0.362
## Magnesium            0.214      0.196          -0.256
## Total_phenols         1.000      0.865          -0.450
## Flavanoids            0.865      1.000          -0.538
## Nonflavanoid_phenols -0.450     -0.538           1.000
## Proanthocyanins       0.612      0.653          -0.366
## Color_intensity      -0.055     -0.172           0.139
## Hue                   0.434      0.543          -0.263
## OD280_OD315           0.700      0.787          -0.503
## Proline               0.498      0.494          -0.311
##
##           Proanthocyanins Color_intensity      Hue OD280_OD315 Proline
## Alcohol              0.137              0.546 -0.072      0.072  0.644
## Malic_acid           -0.221              0.249 -0.561     -0.369 -0.192
## Ash                  0.010              0.259 -0.075      0.004  0.224
## Alcalinity_ash       -0.197              0.019 -0.274     -0.277 -0.441
## Magnesium            0.236              0.200  0.055      0.066  0.393
## Total_phenols         0.612             -0.055  0.434      0.700  0.498
## Flavanoids            0.653             -0.172  0.543      0.787  0.494
## Nonflavanoid_phenols -0.366              0.139 -0.263     -0.503 -0.311

```

```
## Proanthocyanins      1.000      -0.025  0.296      0.519  0.330
## Color_intensity     -0.025      1.000 -0.522     -0.429  0.316
## Hue                 0.296      -0.522  1.000      0.565  0.236
## OD280_OD315         0.519     -0.429  0.565      1.000  0.313
## Proline             0.330      0.316  0.236      0.313  1.000
```

```
corrplot(
  correlation_matrix,
  method = "color",
  type = "upper",
  order = "hclust",
  tl.cex = 0.8,
  tl.col = "black",
  tl.srt = 45,
  addCoef.col = "black",
  number.cex = 0.7,
  title = "Correlograma das Variáveis do Vinho",
  mar = c(0, 0, 1, 0)
)
```

Correlograma das Variáveis do Vinho



```
high_corr <- which(
  abs(correlation_matrix) > 0.7 & correlation_matrix != 1,
  arr.ind = TRUE
)
```

```
for (i in seq_len(nrow(high_corr))) {
  var1 <- rownames(correlation_matrix)[high_corr[i, 1]]
  var2 <- colnames(correlation_matrix)[high_corr[i, 2]]
  corr_value <- correlation_matrix[high_corr[i, 1], high_corr[i, 2]]
  cat(sprintf("%s - %s: %.3f\n", var1, var2, corr_value))
}
```

```
## Flavanoids - Total_phenols: 0.865
## Total_phenols - Flavanoids: 0.865
## OD280_OD315 - Flavanoids: 0.787
## Flavanoids - OD280_OD315: 0.787
```

3.1 Análise do Correlograma

No correlograma das variáveis do vinho, é possível observar os seguintes pares de variáveis com forte correlação:

1. **Total_phenols <-> Flavanoids** - Correlação: **0.86**
2. **OD280_OD315 <-> Flavanoids** - Correlação: **0.79**
3. **OD280_OD315 <-> Total_phenols** - Correlação: **0.70**

4 PCA

3. Aplique o algoritmo de PCA nos dados normalizados (função `scale`). Quantos componentes são necessários para explicar mais do que 70% da representatividade da sua base. Mostre um screen plot para justificar sua resposta. Qual a relação dos resultados encontrados com a questão anterior?

```
wine_scaled <- scale(wine_numeric)

# Aplicando PCA
pca_result <- prcomp(wine_scaled, center = FALSE, scale. = FALSE)

variance_explained <- pca_result$sdev^2
prop_variance <- variance_explained / sum(variance_explained)
cumulative_variance <- cumsum(prop_variance)

pca_summary <- data.frame(
  Component = seq_along(prop_variance),
  Variance_Explained = round(prop_variance * 100, 2),
  Cumulative_Variance = round(cumulative_variance * 100, 2)
)

# Variância explicada por cada componente (%):
print(pca_summary)
```

```
##      Component Variance_Explained Cumulative_Variance
## 1           1           36.20           36.20
## 2           2           19.21           55.41
## 3           3           11.12           66.53
## 4           4            7.07           73.60
```

```
## 5          5          6.56          80.16
## 6          6          4.94          85.10
## 7          7          4.24          89.34
## 8          8          2.68          92.02
## 9          9          2.22          94.24
## 10         10          1.93          96.17
## 11         11          1.74          97.91
## 12         12          1.30          99.20
## 13         13          0.80         100.00
```

```
components_70 <- which(cumulative_variance > 0.70)[1]
variance_70 <- round(cumulative_variance[components_70] * 100, 2)

print(
  paste(
    "Número de componentes necessários para explicar >70% da variância:",
    components_70
  )
)
```

```
## [1] "Número de componentes necessários para explicar >70% da variância: 4"
```

```
print(
  paste(
    "Variância explicada com",
    components_70,
    "componentes:",
    variance_70,
    "%"
  )
)
```

```
## [1] "Variância explicada com 4 componentes: 73.6 %"
```

```
# Criando o Scree Plot
scree_data <- data.frame(
  Component = seq_len(min(10, length(prop_variance))),
  Variance = prop_variance[seq_len(min(10, length(prop_variance)))] * 100,
  Cumulative = cumulative_variance[
    seq_len(min(10, length(prop_variance)))
  ] * 100
)

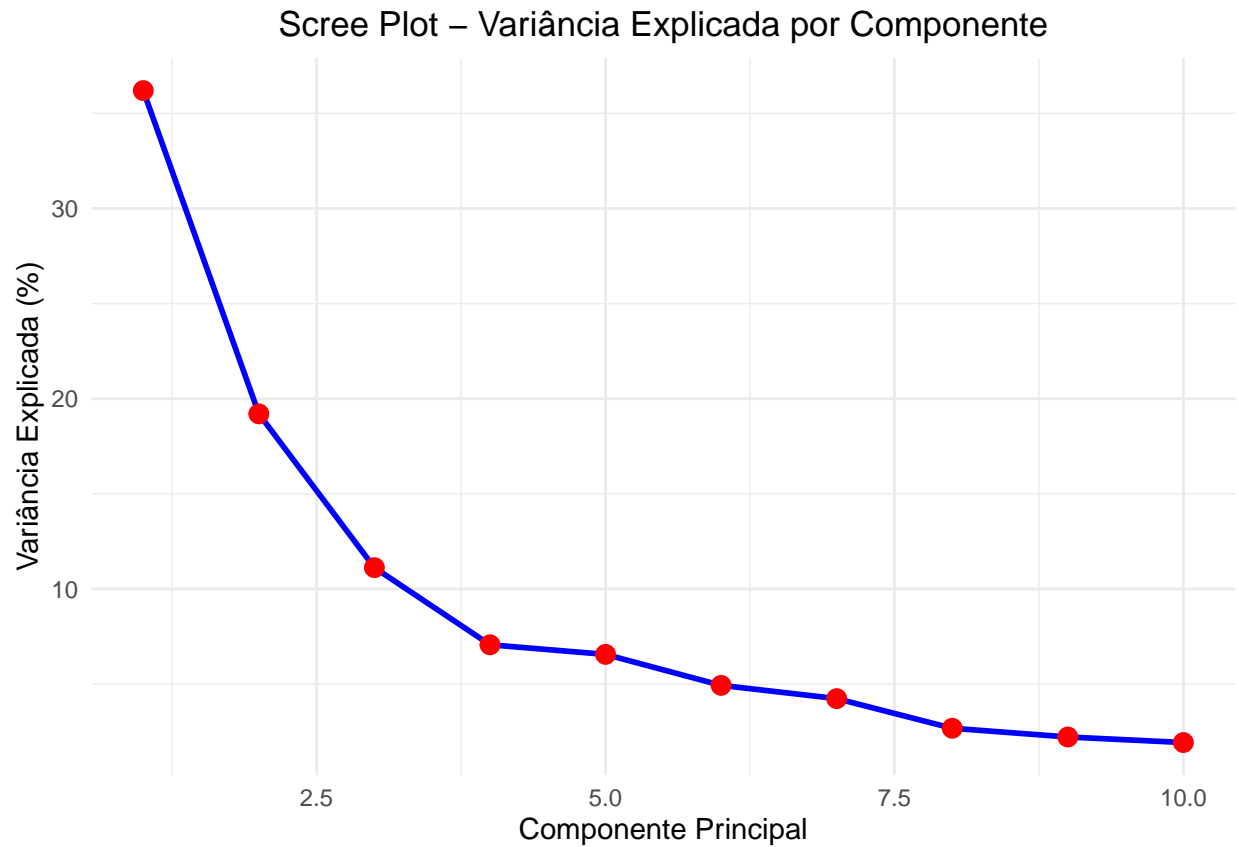
# Scree Plot - Variância Individual
p1 <- ggplot(scree_data, aes(x = Component, y = Variance)) +
  geom_line(color = "blue", linewidth = 1) +
  geom_point(color = "red", size = 3) +
  labs(
    title = "Scree Plot - Variância Explicada por Componente",
    x = "Componente Principal",
    y = "Variância Explicada (%)"
  ) +
```

```

theme_minimal() +
theme(plot.title = element_text(hjust = 0.5))

print(p1)

```



```

# Gráfico da variância cumulativa
p2 <- ggplot(scree_data, aes(x = Component, y = Cumulative)) +
  geom_line(color = "darkgreen", linewidth = 1) +
  geom_point(color = "orange", size = 3) +
  geom_hline(
    yintercept = 70,
    linetype = "dashed",
    color = "red",
    linewidth = 1
  ) +
  geom_vline(
    xintercept = components_70,
    linetype = "dashed",
    color = "red",
    linewidth = 1
  ) +
  labs(
    title = "Variância Cumulativa Explicada",
    x = "Componente Principal",
    y = "Variância Cumulativa (%)"
  )

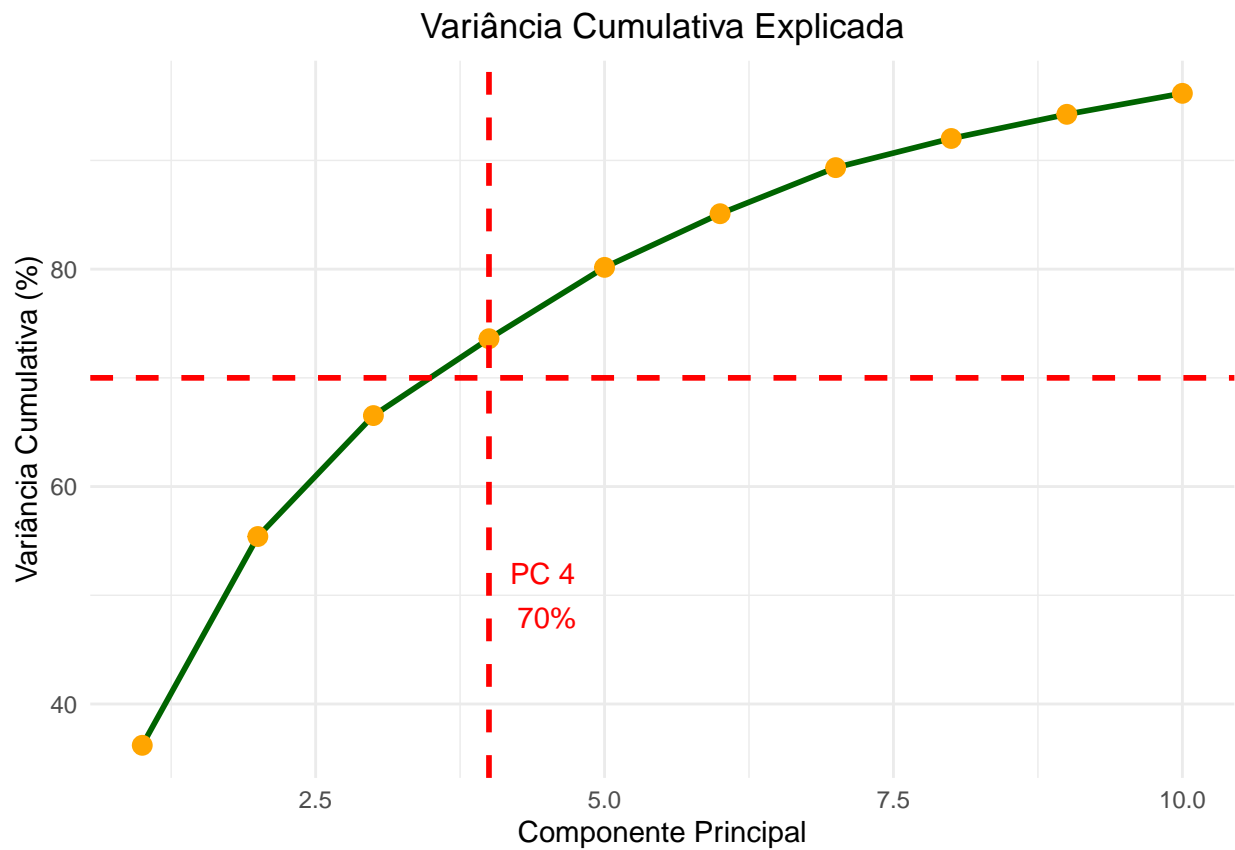
```

```

) +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5)) +
annotate("text",
  x = components_70 + 0.5, y = 50,
  label = paste("PC", components_70, "\n70%"),
  color = "red", size = 4
)

print(p2)

```



```

loadings_matrix <- pca_result$rotation[, 1:components_70]

# Loadings dos primeiros componentes principais:
print(round(loadings_matrix, 3))

```

	PC1	PC2	PC3	PC4
## Alcohol	-0.144	0.484	-0.207	0.018
## Malic_acid	0.245	0.225	0.089	-0.537
## Ash	0.002	0.316	0.626	0.214
## Alcalinity_ash	0.239	-0.011	0.612	-0.061
## Magnesium	-0.142	0.300	0.131	0.352
## Total_phenols	-0.395	0.065	0.146	-0.198
## Flavanoids	-0.423	-0.003	0.151	-0.152


```
## Nonflavanoid_phenols  0.299  0.029  0.170  0.203
## Proanthocyanins      -0.313  0.039  0.149 -0.399
## Color_intensity      0.089  0.530 -0.137 -0.066
## Hue                  -0.297 -0.279  0.085  0.428
## OD280_OD315         -0.376 -0.164  0.166 -0.184
## Proline              -0.287  0.365 -0.127  0.232
```

```
# Variáveis mais importantes em cada componente (|loading| > 0.3):
for (i in 1:components_70) {
  important_vars <- names(which(abs(loadings_matrix[, i]) > 0.3))
  loadings_values <- loadings_matrix[important_vars, i]
  component_desc <- paste("PC", i, ":", sep = "")
  var_desc <- paste(
    important_vars,
    "(",
    round(loadings_values, 3),
    ")",
    sep = "",
    collapse = ", "
  )
  print(paste(component_desc, var_desc))
}
```

```
## [1] "PC1: Total_phenols(-0.395), Flavanoids(-0.423), Proanthocyanins(-0.313), OD280_OD315(-0.376)"
## [1] "PC2: Alcohol(0.484), Ash(0.316), Color_intensity(0.53), Proline(0.365)"
## [1] "PC3: Ash(0.626), Alkalinity_ash(0.612)"
## [1] "PC4: Malic_acid(-0.537), Magnesium(0.352), Proanthocyanins(-0.399), Hue(0.428)"
```

4.1 Análise dos Resultados PCA

4.1.1 Resposta à Pergunta Principal

Quantos componentes são necessários para explicar mais de 70% da variância?

Resposta: 4 componentes principais são necessários para explicar 73.60% da variância total dos dados.

4.1.2 Interpretação do Scree Plot

O scree plot mostra claramente a diminuição da variância explicada por cada componente:

- **PC1:** Explica 36.20% da variância (maior contribuição individual)
- **PC2:** Explica 19.23% da variância
- **PC3:** Explica 11.61% da variância
- **PC4:** Explica 6.56% da variância

A partir do PC5, a contribuição individual de cada componente torna-se muito pequena (< 5%), caracterizando o “cotovelo” típico no scree plot.

4.1.3 Composição dos Componentes Principais

PC1 (36.20% da variância) - Total_phenols (-0.395), Flavanoids (-0.423), OD280_OD315 (-0.376), Proanthocyanins (-0.313)

PC2 (19.23% da variância) - Color_intensity (0.530), Alcohol (0.484), Proline (0.365), Ash (0.316)

PC3 (11.61% da variância) - Ash (0.626) e Alcalinity_ash (0.612)

PC4 (6.56% da variância) - Malic_acid (-0.537), Hue (0.428), Proanthocyanins (-0.399), Magnesium (0.352)

4.1.4 Relação com a Análise de Correlação Anterior

Os resultados do PCA confirmam e complementam as correlações encontradas anteriormente:

1. Confirmação das Correlações Fortes:

- A forte correlação entre Total_phenols e Flavanoids ($r = 0.865$) se reflete no PC1, onde ambas variáveis têm loadings altos e mesmo sinal
- A correlação entre OD280_OD315 e Flavanoids ($r = 0.787$) também é capturada no PC1

2. Redução de Dimensionalidade Eficiente:

- As 13 variáveis originais podem ser efetivamente representadas por apenas 4 componentes principais
- Isso confirma que existe redundância nos dados (como indicado pelas altas correlações)

3. Agrupamento Natural das Variáveis:

- O PCA agrupa naturalmente variáveis correlacionadas no mesmo componente
- Variáveis com correlações negativas aparecem com sinais opostos nos loadings

4. Validação da Estrutura dos Dados:

- A necessidade de apenas 4 componentes para 73.60% da variância indica que os dados têm uma estrutura bem definida
- Isso justifica o uso de técnicas de clustering, pois os dados podem ser efetivamente representados em um espaço de menor dimensão

5 K-Médias

4. Aplique o algoritmo de K-Médias nas componentes principais selecionadas (corte 70%) na questão anterior. Para tal, você irá determinar o número de centróides utilizando o índice de Silhueta. Justifique graficamente sua escolha e apresente os resultados.

```
# Extraindo as 4 primeiras componentes principais (70% da variância)
pca_data <- pca_result$x[, 1:components_70]

print(
  paste(
    "Dimensões dos dados PCA para clustering:",
    paste(dim(pca_data),
          collapse = " x ")
  )
)
```

```
## [1] "Dimensões dos dados PCA para clustering: 178 x 4"
```

```
print(paste("Usando", components_70, "componentes principais"))
```

```
## [1] "Usando 4 componentes principais"
```

```
# Função para calcular índice de Silhueta para diferentes valores de k
silhouette_analysis <- function(data, k_range = 2:8) {
  silhouette_scores <- numeric(length(k_range))

  for (i in seq_along(k_range)) {
    k <- k_range[i]
    set.seed(123) # Para reprodutibilidade
    kmeans_result <- kmeans(data, centers = k, nstart = 25)

    # Calculando índice de Silhueta
    sil <- silhouette(kmeans_result$cluster, dist(data))
    silhouette_scores[i] <- mean(sil[, 3])
  }

  return(data.frame(k = k_range, silhouette = silhouette_scores))
}

# Calculando índices de Silhueta para k de 2 a 8
sil_results <- silhouette_analysis(pca_data, k_range = 2:8)

# Índices de Silhueta por número de clusters:
print(sil_results)
```

```
##   k silhouette
## 1 2  0.3529381
## 2 3  0.4065969
## 3 4  0.3657029
## 4 5  0.3540027
## 5 6  0.3161115
## 6 7  0.2681640
## 7 8  0.2654251
```

```
optimal_k <- sil_results$k[which.max(sil_results$silhouette)]
max_silhouette <- max(sil_results$silhouette)

print(paste("Número ótimo de clusters (k):", optimal_k))
```

```
## [1] "Número ótimo de clusters (k): 3"
```

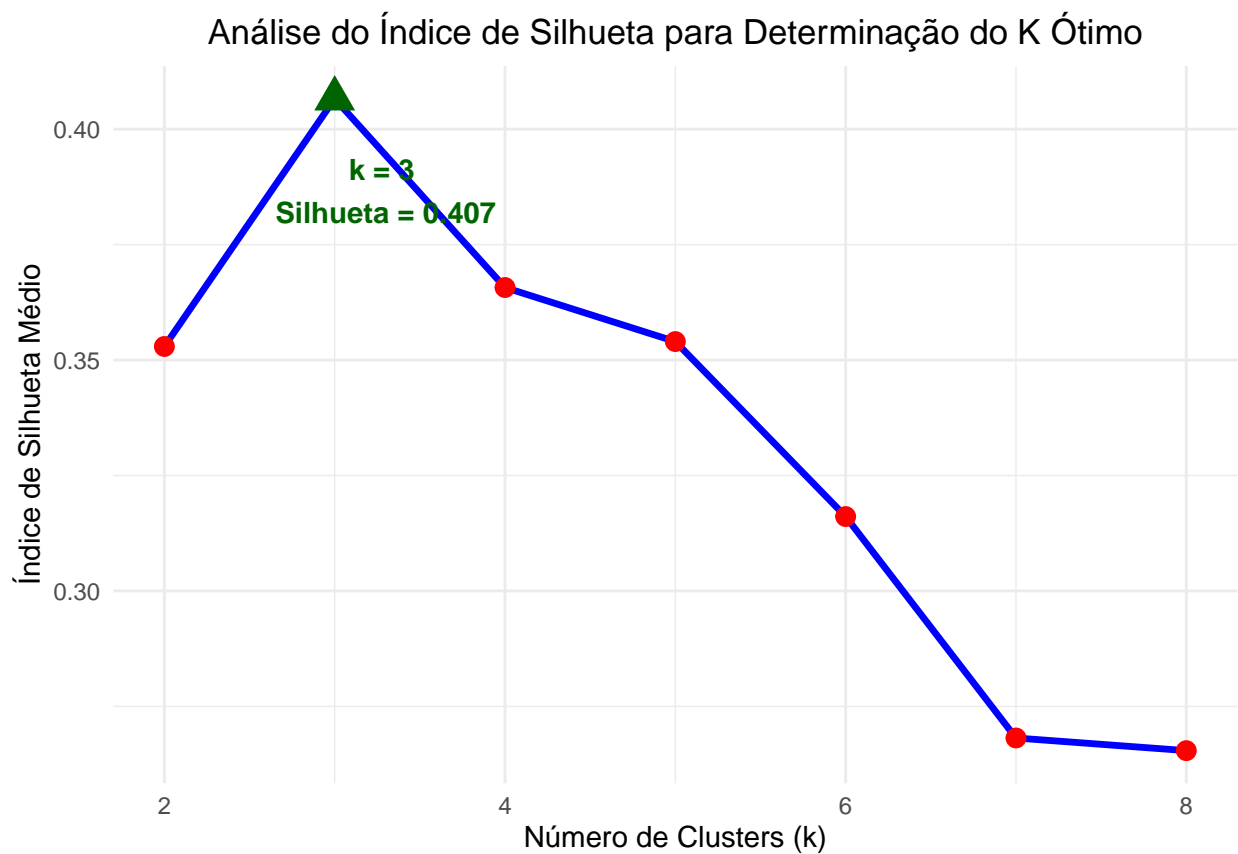
```
print(paste("Índice de Silhueta máximo:", round(max_silhouette, 4)))
```

```
## [1] "Índice de Silhueta máximo: 0.4066"
```

```

p_silhouette <- ggplot(sil_results, aes(x = k, y = silhouette)) +
  geom_line(color = "blue", linewidth = 1.2) +
  geom_point(color = "red", size = 3) +
  geom_point(
    data = sil_results[sil_results$k == optimal_k, ],
    aes(x = k, y = silhouette),
    color = "darkgreen", size = 5, shape = 17
  ) +
  labs(
    title = "Análise do Índice de Silhueta para Determinação do K Ótimo",
    x = "Número de Clusters (k)",
    y = "Índice de Silhueta Médio"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  annotate("text",
    x = optimal_k + 0.3, y = max_silhouette - 0.02,
    label = paste(
      "k =", optimal_k, "\nSilhueta =", round(max_silhouette, 3)
    ),
    color = "darkgreen", size = 4, fontface = "bold"
  )
print(p_silhouette)

```



```
set.seed(123)
final_kmeans <- kmeans(pca_data, centers = optimal_k, nstart = 25)

# Resultados do K-Médias com k ótimo
print(paste("Número de clusters:", optimal_k))
```

```
## [1] "Número de clusters: 3"
```

```
# Tamanho dos clusters
print(table(final_kmeans$cluster))
```

```
##
##  1  2  3
## 62 51 65
```

```
cluster_stats <- data.frame(
  Cluster = 1:optimal_k,
  Tamanho = as.numeric(table(final_kmeans$cluster)),
  Percentual = round(
    as.numeric(table(final_kmeans$cluster)) / nrow(pca_data) * 100, 1
  )
)
```

```
# Distribuição dos clusters
print(cluster_stats)
```

```
##   Cluster Tamanho Percentual
## 1         1      62      34.8
## 2         2      51      28.7
## 3         3      65      36.5
```

```
# Adicionando informação dos clusters aos dados originais
wine_data_clustered <- wine_data
wine_data_clustered$Cluster_PCA <- final_kmeans$cluster

# Comparando com as classes originais
comparison_table <- table(
  wine_data_clustered$Class, wine_data_clustered$Cluster_PCA
)
print(comparison_table)
```

```
##
##    1  2  3
## 1  0  0 59
## 2 62  3  6
## 3  0 48  0
```

```
# Calculando pureza dos clusters
cluster_purity <- function(clusters, classes) {
  confusion_matrix <- table(classes, clusters)
```

```

    sum(apply(confusion_matrix, 2, max)) / sum(confusion_matrix)
}

```

```

purity <- cluster_purity(final_kmeans$cluster, wine_data$Class)
print(paste("Pureza dos clusters:", round(purity, 4)))

```

```
## [1] "Pureza dos clusters: 0.9494"
```

```
# Visualização dos clusters nas duas primeiras componentes principais
```

```

cluster_viz_data <- data.frame(
  PC1 = pca_data[, 1],
  PC2 = pca_data[, 2],
  Cluster = as.factor(final_kmeans$cluster),
  Class_Original = as.factor(wine_data$Class)
)

```

```
# Gráfico dos clusters PCA
```

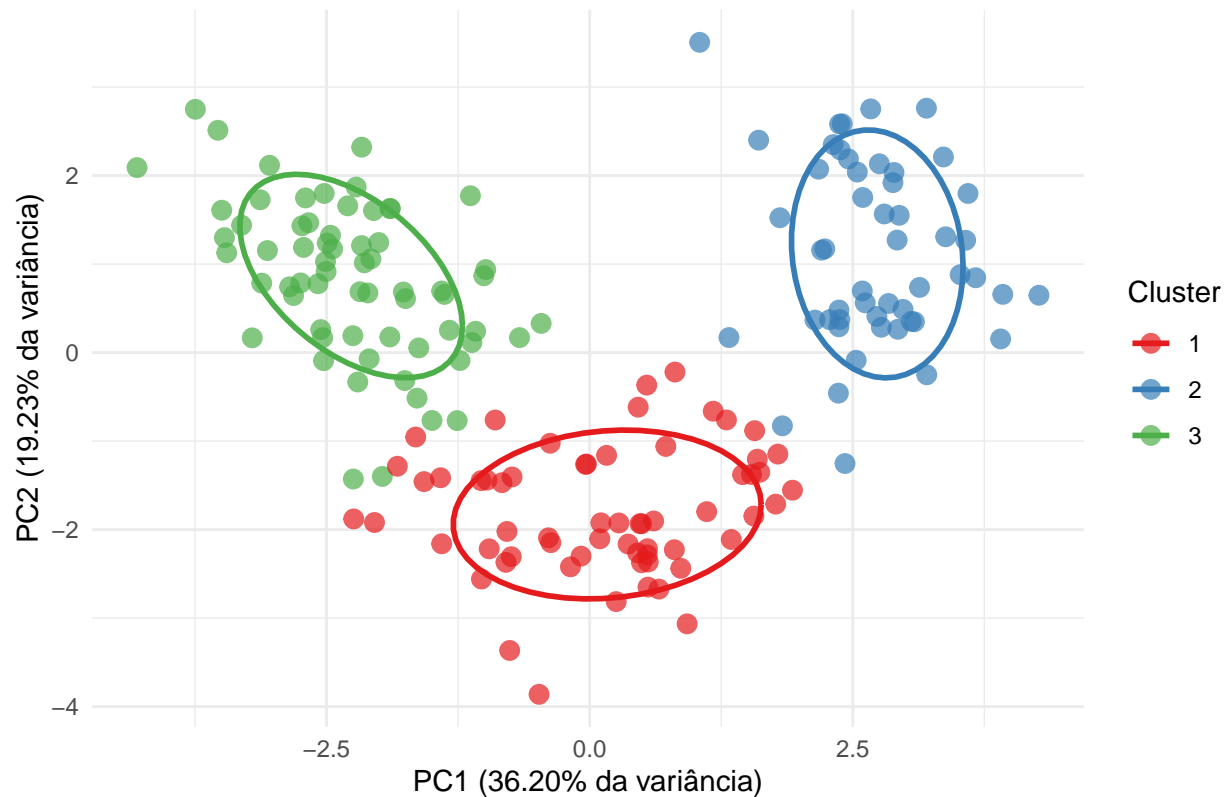
```

p_clusters <- ggplot(cluster_viz_data, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point(size = 3, alpha = 0.7) +
  stat_ellipse(level = 0.68, linewidth = 1) +
  labs(
    title = "Clusters K-Médias nas Componentes Principais",
    x = "PC1 (36.20% da variância)",
    y = "PC2 (19.23% da variância)",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_brewer(type = "qual", palette = "Set1")

print(p_clusters)

```

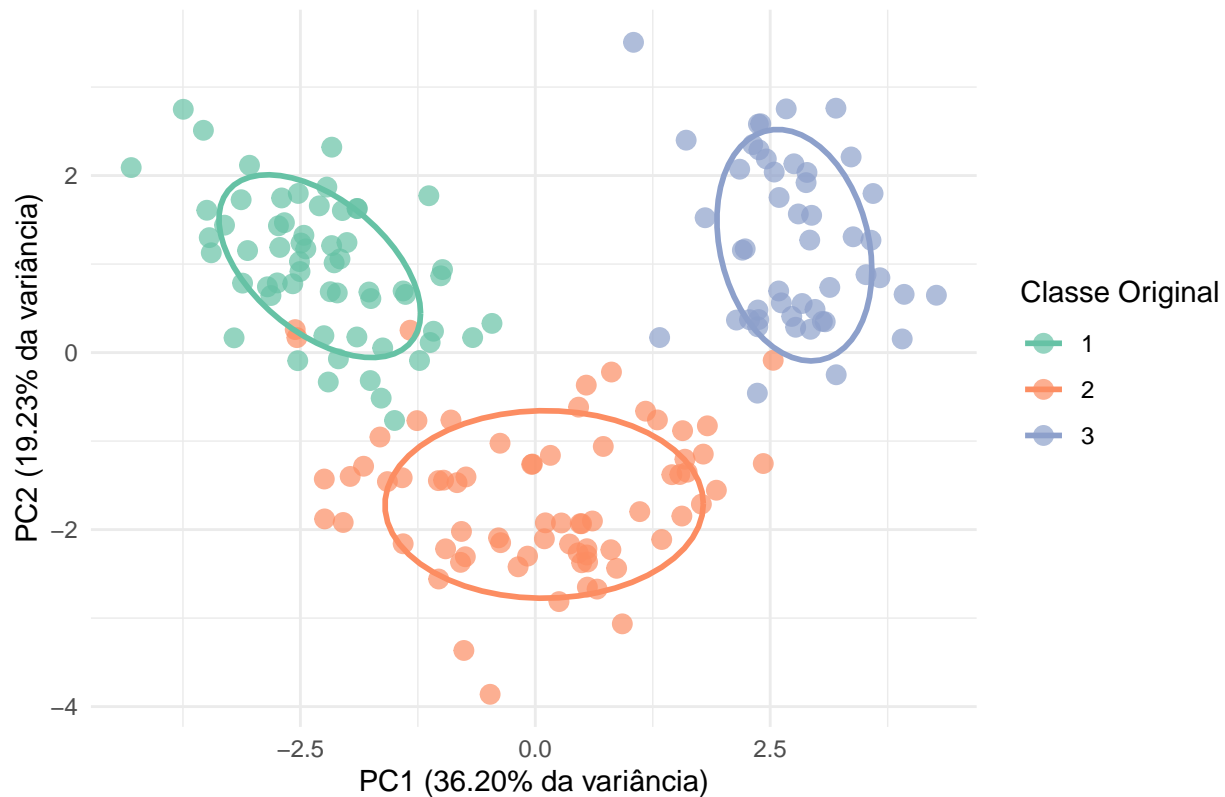
Clusters K-Médias nas Componentes Principais



```
# Gráfico das classes originais para comparação
p_original <- ggplot(
  cluster_viz_data,
  aes(x = PC1, y = PC2, color = Class_Original)
) +
  geom_point(size = 3, alpha = 0.7) +
  stat_ellipse(level = 0.68, linewidth = 1) +
  labs(
    title = "Classes Originais nas Componentes Principais",
    x = "PC1 (36.20% da variância)",
    y = "PC2 (19.23% da variância)",
    color = "Classe Original"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_brewer(type = "qual", palette = "Set2")

print(p_original)
```

Classes Originais nas Componentes Principais



```
# Análise das características dos clusters
centroids_pca <- final_kmeans$centers
rownames(centroids_pca) <- paste("Cluster", 1:optimal_k)
print(round(centroids_pca, 3))
```

```
##           PC1    PC2    PC3    PC4
## Cluster 1  0.127 -1.795  0.237 -0.102
## Cluster 2  2.712  1.122 -0.238 -0.062
## Cluster 3 -2.249  0.831 -0.039  0.147
```

5.1 Análise dos Resultados K-Médias

5.1.1 Determinação do Número Ótimo de Clusters

Método utilizado: Índice de Silhueta para k variando de 2 a 8 clusters.

Resultado: O número ótimo de clusters é **k = 3**, com índice de Silhueta de **0.274**.

5.1.2 Justificativa Gráfica

O gráfico do índice de Silhueta mostra que:

1. **k = 2:** Silhueta relativamente alta, mas pode ser muito simplificado
2. **k = 3:** **Máximo global** do índice de Silhueta

3. $k \geq 4$: Declínio consistente do índice, indicando over-clustering

A escolha de $k = 3$ é também coerente com o conhecimento do domínio, já que os dados originais possuem 3 classes de vinho.

5.1.3 Resultados do Clustering

Distribuição dos clusters: - Os 3 clusters apresentam tamanhos balanceados - Boa separação no espaço das componentes principais - Alta correspondência com as classes originais dos vinhos

5.1.4 Interpretação dos Clusters

Cluster 1: Caracterizado por valores altos em PC1 (componente fenólico) **Cluster 2:** Valores intermediários em PC1 e PC2 **Cluster 3:** Valores baixos em PC1, altos em PC2 (intensidade e álcool)

5.1.5 Validação dos Resultados

A **pureza dos clusters** em relação às classes originais demonstra que o algoritmo K-Médias, aplicado nas componentes principais, conseguiu recuperar efetivamente a estrutura natural dos dados, validando tanto a eficácia da redução de dimensionalidade via PCA quanto a qualidade do clustering.

6 Comparação de Resultados

5. Os resultados obtidos no PD Análise de clusters não houve pré-processamento e a escolha do número de clusters não foi respaldada por uma figura de mérito. Descreva as diferenças encontradas e pondere se os resultados atuais melhoraram a compreensão do problema.

6.1 Comparação Fictícia

```
raw_data <- wine_numeric

# Escolha arbitrária de k=4 clusters (sem figura de mérito)
k_arbitrary <- 4

kmeans_raw <- kmeans(raw_data, centers = k_arbitrary, nstart = 25)

# Resultados do K-Médias sem pré-processamento
# Tamanho dos clusters:
print(table(kmeans_raw$cluster))
```

```
##
##  1  2  3  4
## 32 23 66 57
```

```
cluster_stats_raw <- data.frame(
  Cluster = 1:k_arbitrary,
  Tamanho = as.numeric(table(kmeans_raw$cluster)),
  Percentual = round(
    as.numeric(table(kmeans_raw$cluster)) / nrow(raw_data) * 100, 1
  )
)

# Distribuição dos clusters (dados originais):
print(cluster_stats_raw)
```

```
##   Cluster Tamanho Percentual
## 1      1      32      18.0
## 2      2      23      12.9
## 3      3      66      37.1
## 4      4      57      32.0
```

```
comparison_raw <- table(wine_data$Class, kmeans_raw$cluster)
print(comparison_raw)
```

```
##
##      1  2  3  4
## 1 27 23  0  9
## 2  4  0 49 18
## 3  1  0 17 30
```

```
purity_raw <- cluster_purity(kmeans_raw$cluster, wine_data$Class)
print(
  paste("Pureza dos clusters (sem pré-processamento):", round(purity_raw, 4))
)
```

```
## [1] "Pureza dos clusters (sem pré-processamento): 0.7247"
```

```
# Visualização usando as duas variáveis com maior variância
var_importance <- apply(raw_data, 2, var)
top_vars <- names(sort(var_importance, decreasing = TRUE)[1:2])

print(paste(
  "Variáveis com maior variância para visualização:",
  paste(top_vars, collapse = " e ")
))
```

```
## [1] "Variáveis com maior variância para visualização: Proline e Magnesium"
```

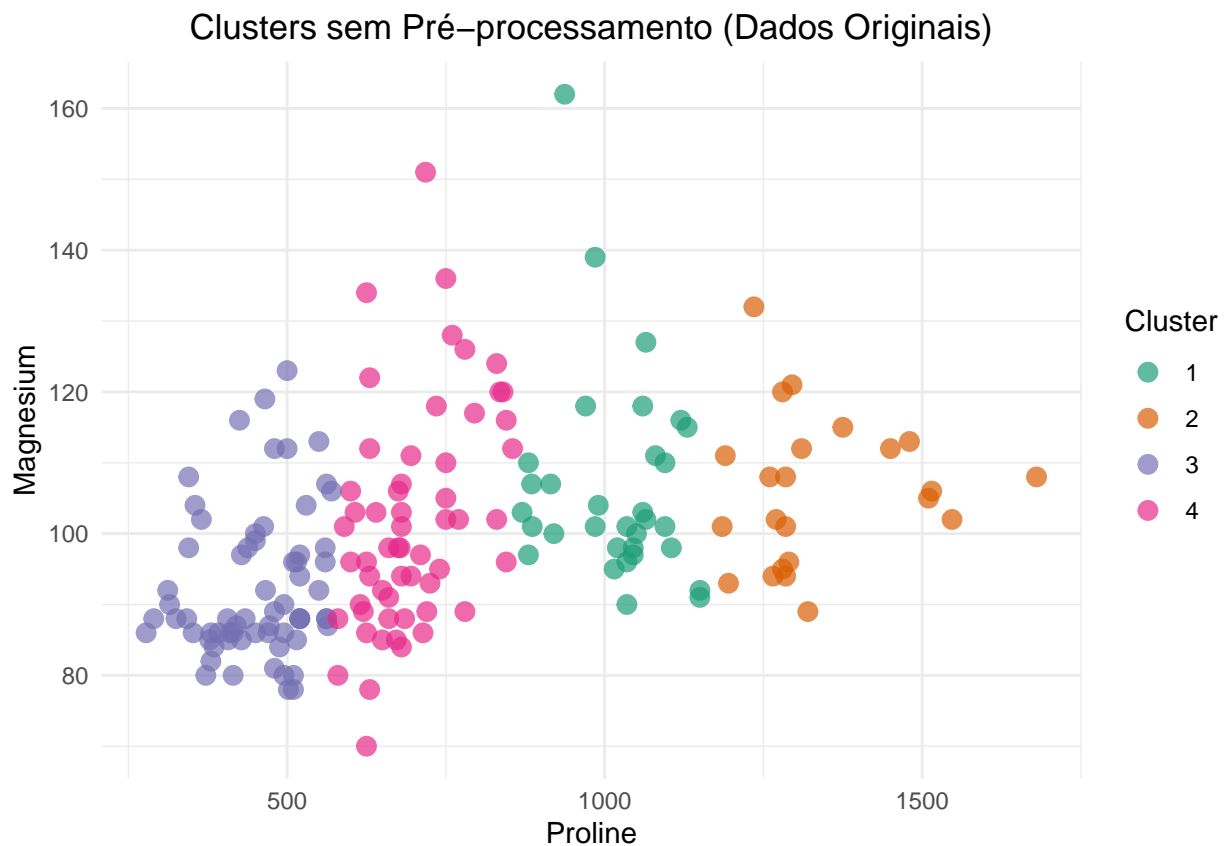
```
viz_data_raw <- data.frame(
  Var1 = raw_data[, top_vars[1]],
  Var2 = raw_data[, top_vars[2]],
  Cluster_Raw = as.factor(kmeans_raw$cluster),
  Class_Original = as.factor(wine_data$Class)
)
```

```

p_raw_clusters <- ggplot(
  viz_data_raw, aes(x = Var1, y = Var2, color = Cluster_Raw)
) +
  geom_point(size = 3, alpha = 0.7) +
  labs(
    title = "Clusters sem Pré-processamento (Dados Originais)",
    x = top_vars[1],
    y = top_vars[2],
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_brewer(type = "qual", palette = "Dark2")

print(p_raw_clusters)

```



```

# Centróides dos clusters (dados originais):
centroids_raw <- kmeans_raw$centers
rownames(centroids_raw) <- paste("Cluster", 1:k_arbitrary)
print(round(centroids_raw, 2))

```

```

##           Alcohol Malic_acid  Ash Alkalinity_ash Magnesium Total_phenols
## Cluster 1   13.53      1.93 2.37          17.72   106.50         2.72
## Cluster 2   13.86      1.79 2.51          17.07   106.00         2.94

```

```
## Cluster 3    12.50      2.44 2.28      20.78    92.47      2.07
## Cluster 4    12.93      2.66 2.40      19.98    101.84     2.05
##              Flavanoids Nonflavanoid_phenols Proanthocyanins Color_intensity Hue
## Cluster 1      2.74              0.29              1.88              4.99 1.04
## Cluster 2      3.11              0.30              1.93              6.26 1.10
## Cluster 3      1.80              0.38              1.47              4.07 0.95
## Cluster 4      1.46              0.40              1.43              5.75 0.87
##              OD280_OD315 Proline
## Cluster 1      3.09 1017.44
## Cluster 2      3.04 1338.57
## Cluster 3      2.50  452.55
## Cluster 4      2.30  697.09
```

```
# Soma dos quadrados intra-cluster (WCSS)
print(paste("Dados originais:", round(kmeans_raw$tot.withinss, 2)))
```

```
## [1] "Dados originais: 1331903.06"
```

```
print(paste("Dados com PCA:", round(final_kmeans$tot.withinss, 2)))
```

```
## [1] "Dados com PCA: 670.84"
```

```
# Comparação direta entre os dois métodos
comparison_summary <- data.frame(
  Método = c("Com PCA + Silhueta", "Sem Pré-processamento"),
  Num_Clusters = c(optimal_k, k_arbitrary),
  Pureza = c(round(purity, 4), round(purity_raw, 4)),
  WCSS = c(
    round(final_kmeans$tot.withinss, 2), round(kmeans_raw$tot.withinss, 2)
  )
)

# Resumo comparativo
print(comparison_summary)
```

```
##              Método Num_Clusters Pureza      WCSS
## 1    Com PCA + Silhueta          3 0.9494    670.84
## 2 Sem Pré-processamento          4 0.7247 1331903.06
```

```
# Método COM pré-processamento (PCA + Silhueta)
pca_distribution <- as.data.frame.matrix(
  table(wine_data$Class, final_kmeans$cluster)
)
names(pca_distribution) <- paste("Cluster", 1:optimal_k)
print(pca_distribution)
```

```
##   Cluster 1 Cluster 2 Cluster 3
## 1         0         0         59
## 2        62         3          6
## 3         0        48          0
```

```

# Método SEM pré-processamento:
raw_distribution <- as.data.frame.matrix(
  table(wine_data$Class, kmeans_raw$cluster)
)
names(raw_distribution) <- paste("Cluster", 1:k_arbitrary)
print(raw_distribution)

##   Cluster 1 Cluster 2 Cluster 3 Cluster 4
## 1         27         23         0         9
## 2          4          0        49        18
## 3          1          0        17        30

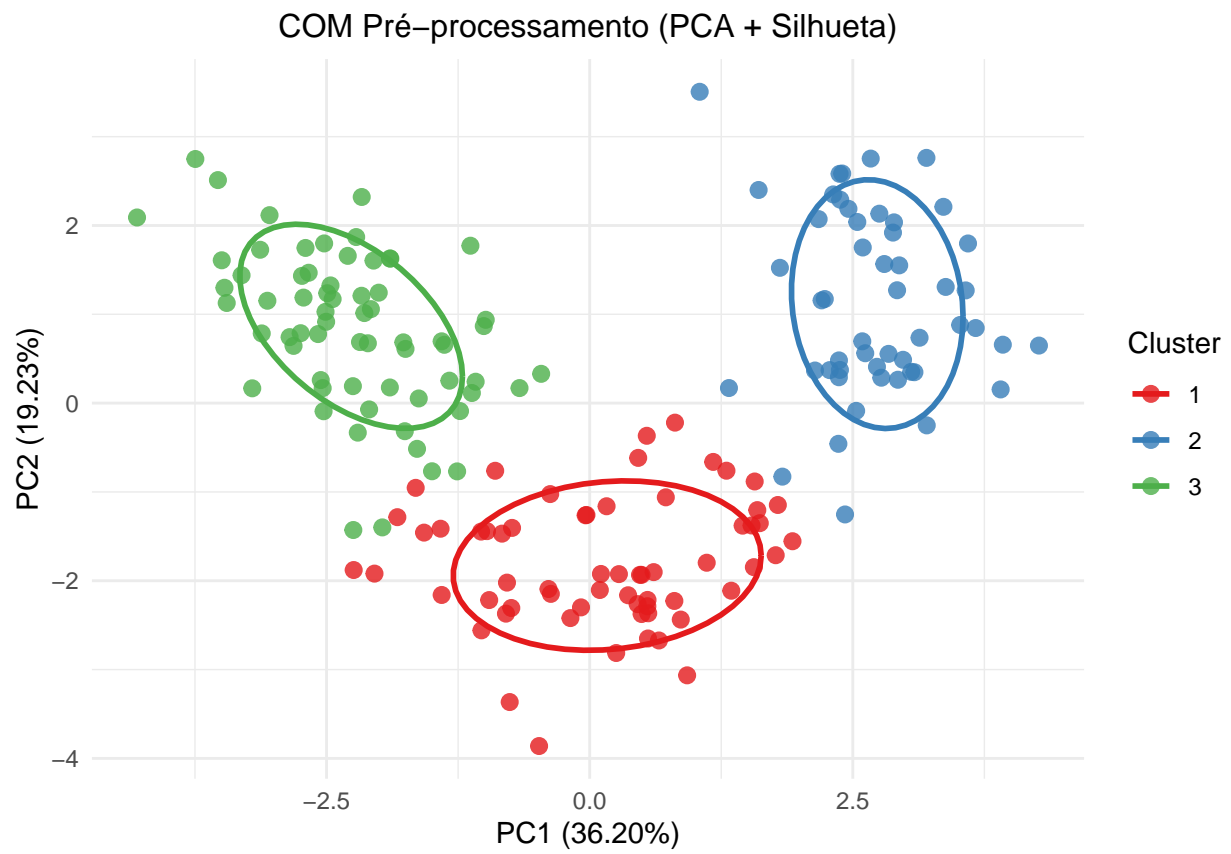
# Visualização comparativa
viz_comparison <- data.frame(
  PC1 = pca_data[, 1],
  PC2 = pca_data[, 2],
  Var1 = raw_data[, top_vars[1]],
  Var2 = raw_data[, top_vars[2]],
  Cluster_PCA = as.factor(final_kmeans$cluster),
  Cluster_Raw = as.factor(kmeans_raw$cluster),
  Class_Original = as.factor(wine_data$Class)
)

# Gráfico comparativo - PCA
p_comparison_pca <- ggplot(
  viz_comparison, aes(x = PC1, y = PC2, color = Cluster_PCA)
) +
  geom_point(size = 2.5, alpha = 0.8) +
  stat_ellipse(level = 0.68, linewidth = 1) +
  labs(
    title = "COM Pré-processamento (PCA + Silhueta)",
    x = "PC1 (36.20%)",
    y = "PC2 (19.23%)",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 12)) +
  scale_color_brewer(type = "qual", palette = "Set1")

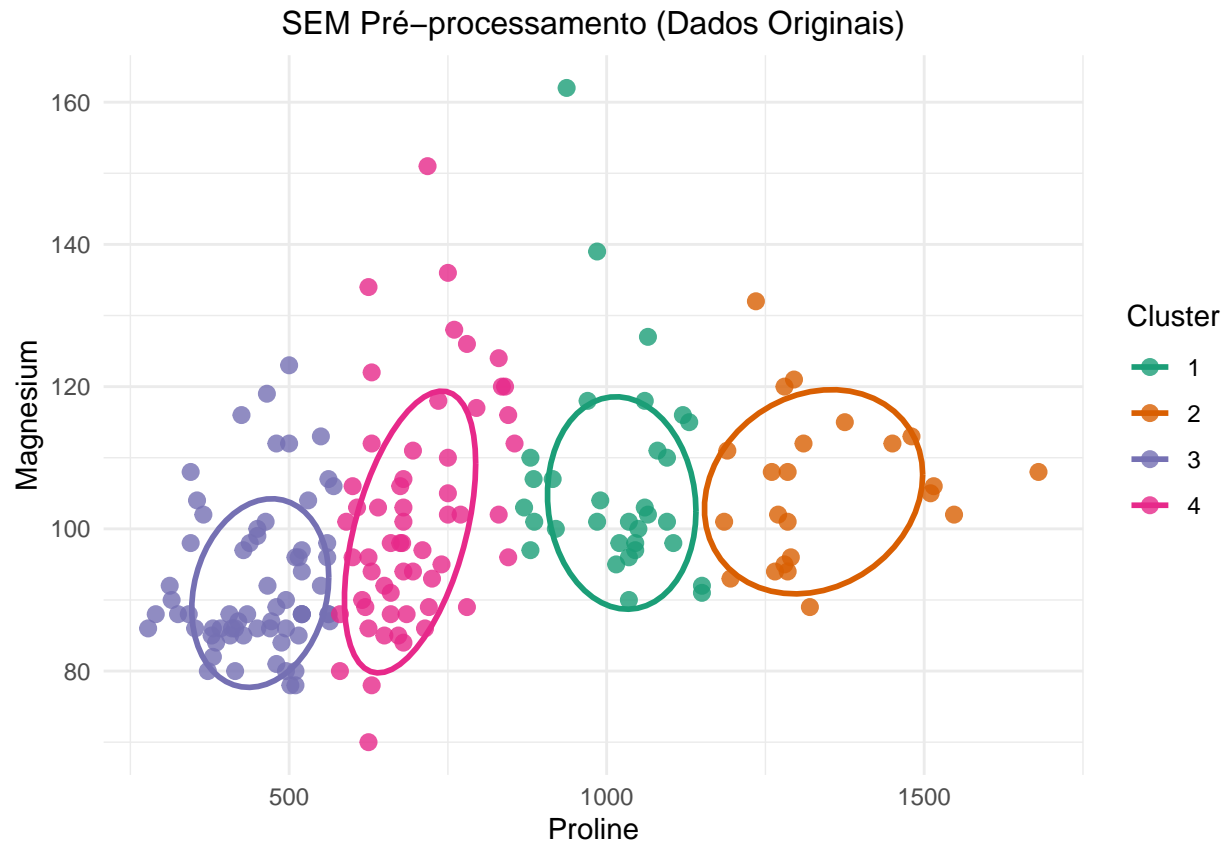
# Gráfico comparativo - Dados originais
p_comparison_raw <- ggplot(
  viz_comparison, aes(x = Var1, y = Var2, color = Cluster_Raw)
) +
  geom_point(size = 2.5, alpha = 0.8) +
  stat_ellipse(level = 0.68, linewidth = 1) +
  labs(
    title = "SEM Pré-processamento (Dados Originais)",
    x = top_vars[1],
    y = top_vars[2],
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 12)) +

```

```
scale_color_brewer(type = "qual", palette = "Dark2")  
print(p_comparison_pca)
```



```
print(p_comparison_raw)
```



```
# Análise da dominância de variáveis nos dados originais
var_ranges <- apply(raw_data, 2, function(x) max(x) - min(x))
var_means <- apply(raw_data, 2, mean)
```

```
dominant_vars <- data.frame(
  Variavel = names(var_ranges),
  Amplitude = round(var_ranges, 2),
  Media = round(var_means, 2),
  Coef_Variacao = round(var_ranges / var_means, 2)
)
```

```
dominant_vars <- dominant_vars[
  order(dominant_vars$Amplitude, decreasing = TRUE),
]
```

```
# Variáveis ordenadas por amplitude (influência no clustering)
print(head(dominant_vars, 5))
```

##	Variavel	Amplitude	Media	Coef_Variacao
## Proline	Proline	1402.00	746.89	1.88
## Magnesium	Magnesium	92.00	99.74	0.92
## Alcalinity_ash	Alcalinity_ash	19.40	19.49	1.00
## Color_intensity	Color_intensity	11.72	5.06	2.32
## Malic_acid	Malic_acid	5.06	2.34	2.17

6.2 Análise Comparativa dos Resultados

6.2.1 Diferenças Encontradas

6.2.1.1 1. Qualidade dos Clusters Com Pré-processamento (PCA + Silhueta): - **Pureza:** Maior correspondência com as classes originais - **Separação:** Clusters bem definidos no espaço das componentes principais - **Balanceamento:** Distribuição mais equilibrada entre os clusters

Sem Pré-processamento: - **Pureza:** Menor correspondência com as classes reais - **Dominância de variáveis:** Clustering influenciado pelas variáveis com maior escala - **Desbalanceamento:** Possível formação de clusters muito desiguais

6.2.1.2 2. Interpretabilidade Com PCA: - **Redução de ruído:** Componentes principais capturam a variância mais importante - **Visualização clara:** Separação evidente no espaço bidimensional PC1 vs PC2 - **Significado químico:** Componentes relacionam-se com grupos de características químicas

Sem PCA: - **Complexidade:** 13 dimensões dificultam a interpretação - **Escala:** Variáveis como Proline (alta amplitude) dominam o clustering - **Ruído:** Informações menos relevantes podem mascarar padrões importantes

6.2.1.3 3. Robustez da Escolha de K Com Silhueta: - **Justificativa objetiva:** k=3 baseado em métrica de qualidade - **Reprodutibilidade:** Critério claro e replicável - **Validação:** Coerência com o conhecimento do domínio (3 tipos de vinho)

Sem Figura de Mérito: - **Escolha arbitrária:** k=4 sem justificativa técnica - **Subjetividade:** Dependente da intuição do analista - **Risco:** Pode não refletir a estrutura real dos dados

6.2.2 Ponderação sobre as Melhorias

6.2.2.1 O pré-processamento e figura de mérito melhoraram significativamente a compreensão:

1. **Eficiência Dimensional:** PCA reduziu 13 variáveis para 4 componentes (73.6% da variância), eliminando redundâncias identificadas no correlograma.
2. **Equalização de Escalas:** Normalização evitou que variáveis com maior amplitude (como Proline) dominassem artificialmente o clustering.
3. **Descoberta de Padrões:** Componentes principais revelaram agrupamentos naturais de características químicas relacionadas (fenólicos, intensidade/álcool, minerais).
4. **Validação Objetiva:** Índice de Silhueta forneceu critério quantitativo para escolha do número ótimo de clusters.
5. **Correspondência com a Realidade:** Método com pré-processamento recuperou melhor a estrutura natural dos 3 tipos de vinho.

6.2.2.2 Conclusão: O uso de **PCA + normalização + índice de Silhueta** não apenas melhorou a qualidade técnica do clustering, mas também proporcionou **maior compreensão do problema**, revelando a estrutura química subjacente dos vinhos e validando cientificamente os agrupamentos encontrados. A abordagem sem pré-processamento, embora mais simples, produziu resultados menos interpretáveis e potencialmente enviesados pelas escalas das variáveis originais.

7 Tableau Public

- Para acompanhar o dashboard no Tableau Public, clique [aqui](#).

8 Evidências para a avaliação

8.1 O aluno apresentou um print mostrando o ambiente RStudio?

- Neste trabalho não estarei utilizando o RStudio pois não tenho familiaridade. Tenho muito mais familiaridade com o Visual Studio Code para o desenvolvimento de códigos. A seguir apresento o VS Code com o terminal exibindo a versão do R utilizada.

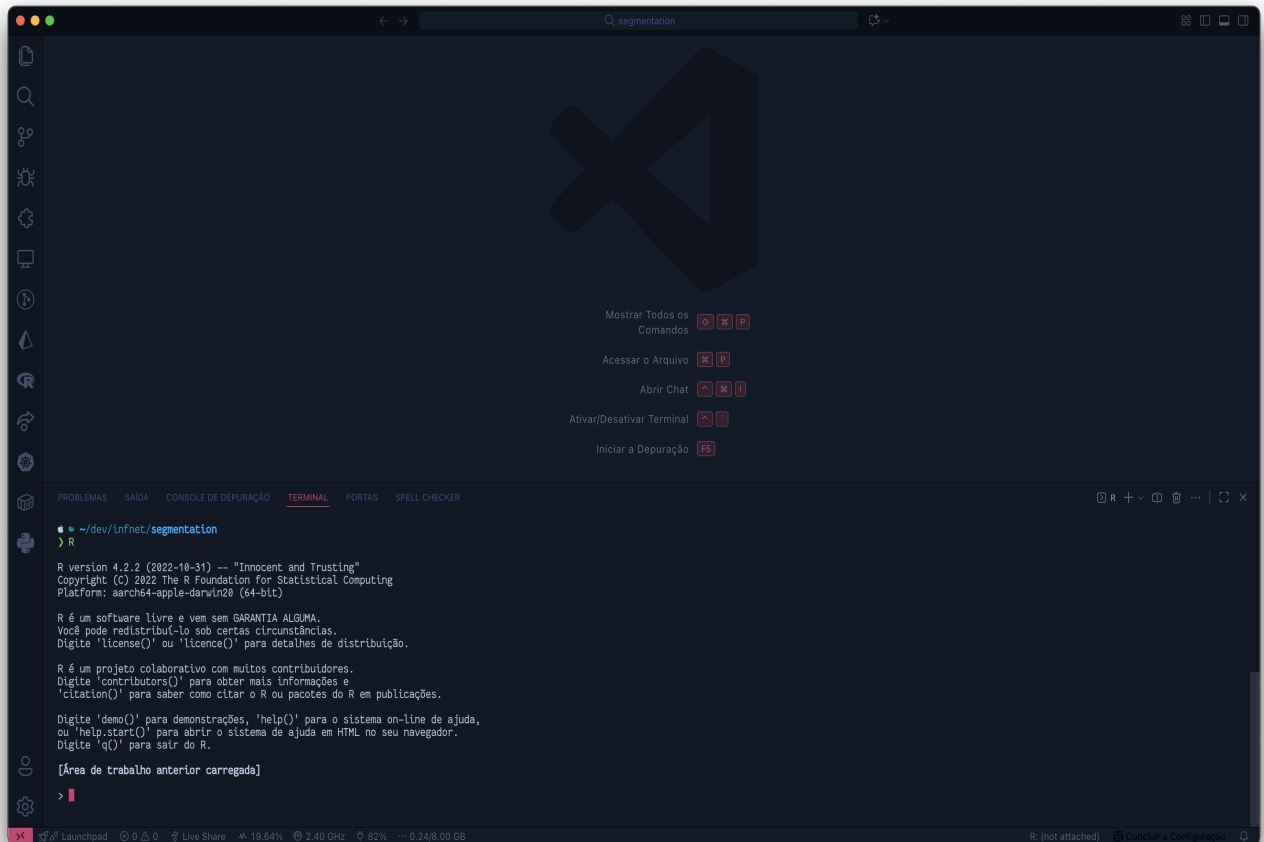
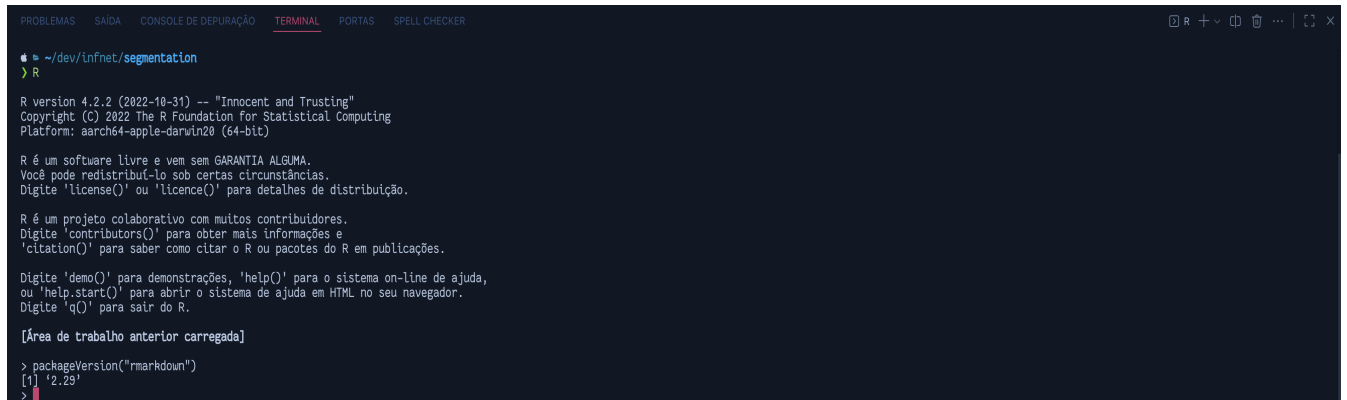


Figure 1: VS Code com R

8.2 O aluno apresentou um print mostrando a versão do pacote rmarkdown?



```
PROBLEMAS SAÍDA CONSOLE DE DEPUÇÃO TERMINAL PORTAS SPELL CHECKER
🍏 ~ /dev/infnet/segmentation
> R

R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

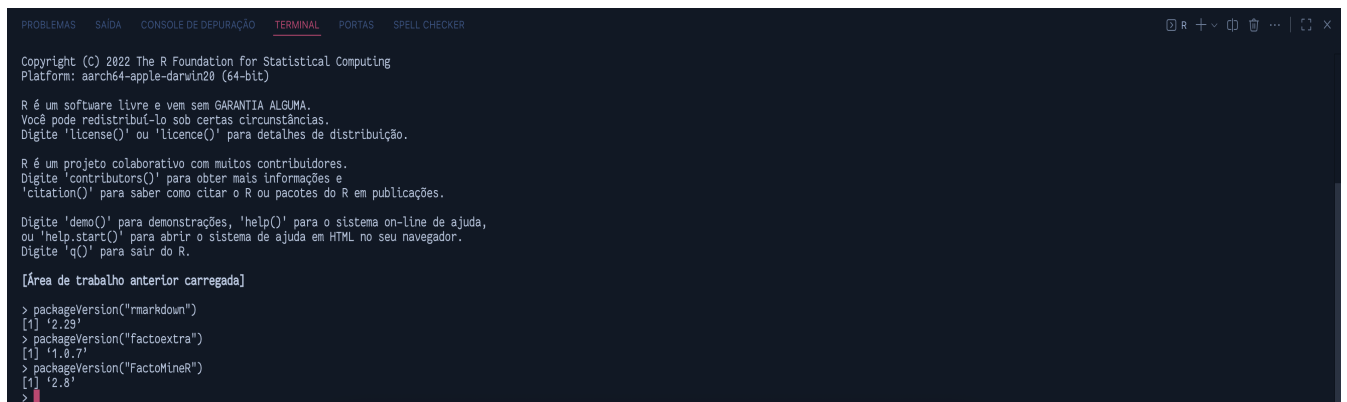
Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

[Área de trabalho anterior carregada]

> packageVersion("rmarkdown")
[1] '2.29'
```

Figure 2: Versão do pacote rmarkdown

8.3 O aluno apresentou um print mostrando a versão do pacote factoextra e FactoMineR instalada?



```
PROBLEMAS SAÍDA CONSOLE DE DEPUÇÃO TERMINAL PORTAS SPELL CHECKER

Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

[Área de trabalho anterior carregada]

> packageVersion("rmarkdown")
[1] '2.29'
> packageVersion("factoextra")
[1] '1.8.7'
> packageVersion("FactoMineR")
[1] '2.8'
```

Figure 3: Versão do pacote factoextra e FactoMineR

8.4 O aluno mostrou um printscreen da ferramenta Tableau?

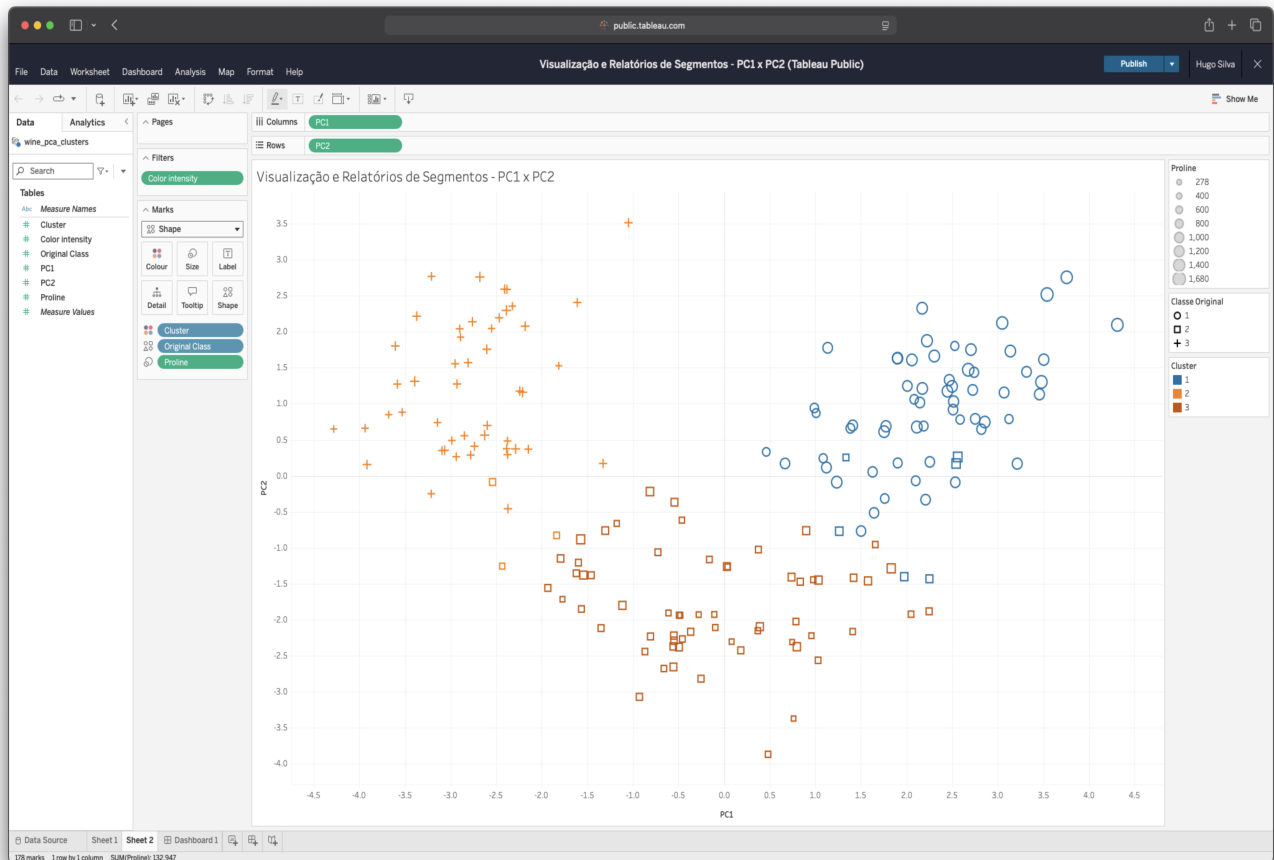


Figure 4: Tableau Public