DataMining - Gym

Diogo Cruz André Eiras

Dataset

```
#Importing the dataset
df <- read.csv("gym.csv", stringsAsFactors = TRUE)
df$Experience_Level <- as.factor(df$Experience_Level)
df_numeric <- df %>% select(where(is.numeric))
```

O dataset escolhido para este trabalho chama-se "*Gym Memebers Exercise Dataset*" tendo sido recolhido no *Kaggle*. O dataset tem informação detalhada sobre a rotina de exercícios, atributos físicos e dados demográficos de membros de um ginásio, sendo composto por 15 variáveis cada uma com 973 observações.

```
cat("Número de observações:",nrow(df), "\n")
```

```
Número de observações: 973
```

```
cat("Número de variávei:",ncol(df), "\n")
```

```
Número de variávei: 15
```

```
sapply(df, class)
```

```
            Age          Gender 
      "integer"        "factor" 
     Weight..kg.      Height..m. 
      "numeric"       "numeric" 
          Max_BPM          Avg_BPM 
      "integer"       "integer" 
```

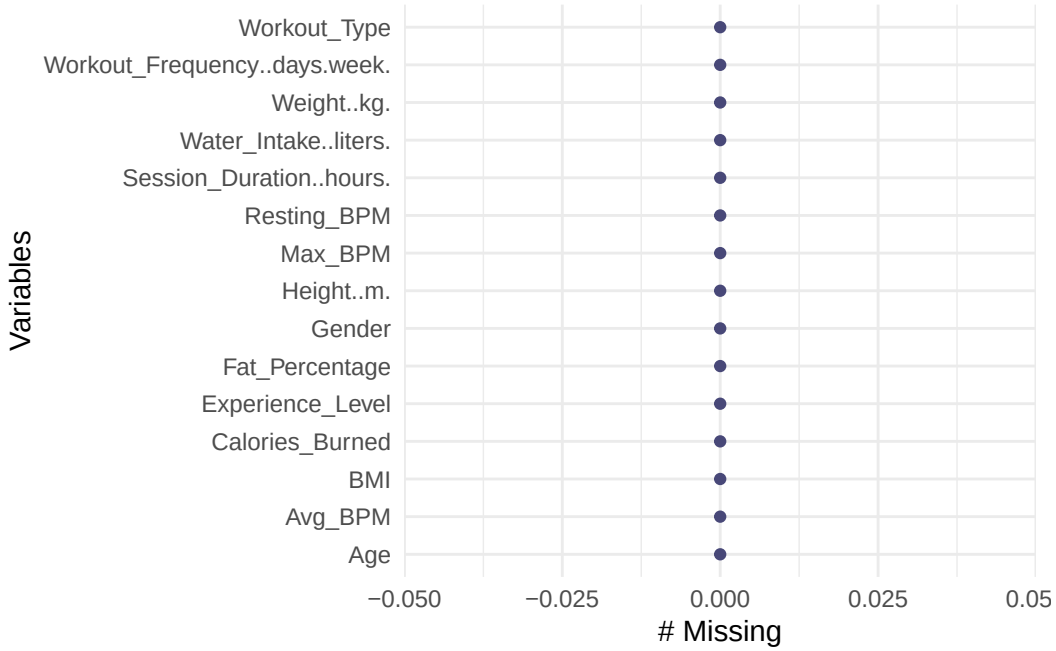
```
                  Resting_BPM      Session_Duration..hours.
                    "integer"                     "numeric"
               Calories_Burned                  Workout_Type
                    "numeric"                      "factor"
                Fat_Percentage         Water_Intake..liters.
                    "numeric"                     "numeric"
Workout_Frequency..days.week.               Experience_Level
                    "integer"                      "factor"
                          BMI
                    "numeric"
```

A lista seguinte indica o significado e tipo de cada uma das variáveis presentes.

- `Age` - Varíavel Quantitativa Contínua que indica a idade do membro.
- `Gender` - Varíavel Categórica que indica o género do membro.
- `Weigh` - Variável Quantitativa Contínua com a massa do membro em kg.
- `Height` - Variável Quantitativa Contínua com a altura do membro em metros.
- `Max BPM` - Varíavel Quantitativa Contínua que indica a frequência cardíaca máxima atingida pelo membro durante o seu treino em batimentos por minuto.
- `Avg BPM` - Varíavel Quantitativa Contínua que indica a frequência cardíaca média do membro durante o seu treino em batimentos por minuto.
- `Resting BPM` - Varíavel Quantitativa Contínua que indica a frequência cardíaca do membro em descanso antes do treino.
- `Session Duration` - Varíavel Quantitativa Contínua que indica a duração do treino em horas.
- `Calories Burned` - Variável Quantitativa Contínua que indica a quantidade de calorias gastas durante o treino em kCal.
- `Workout Type` - Varíavel Categórica que inidica o tipo de treino realizado.
- `Fat Percentage` - Variável Quantitativa Contínua que indica a percentagem de massa gorda do membro.
- `Water Intake` - Variável Quantitativa Contínua que indica o número de litros de água ingeridos diáriamente pelo membro.
- `Workout Frequency` - Variável Quantitativa Discreta que indica o número de dias por semana em que o membro treinou.
- `Experience Level` - Variável Quantitativa Discreta que indica o nível de experiência (1 a 3) do membro.
- `BMI` - Variável Quantitativa Contínua que indica o Índice de Massa Corporal do membro.

Como é possível ver no gráfico seguinte, o dataset não apresenta dados omissos.

```
gg_miss_var(df)
```

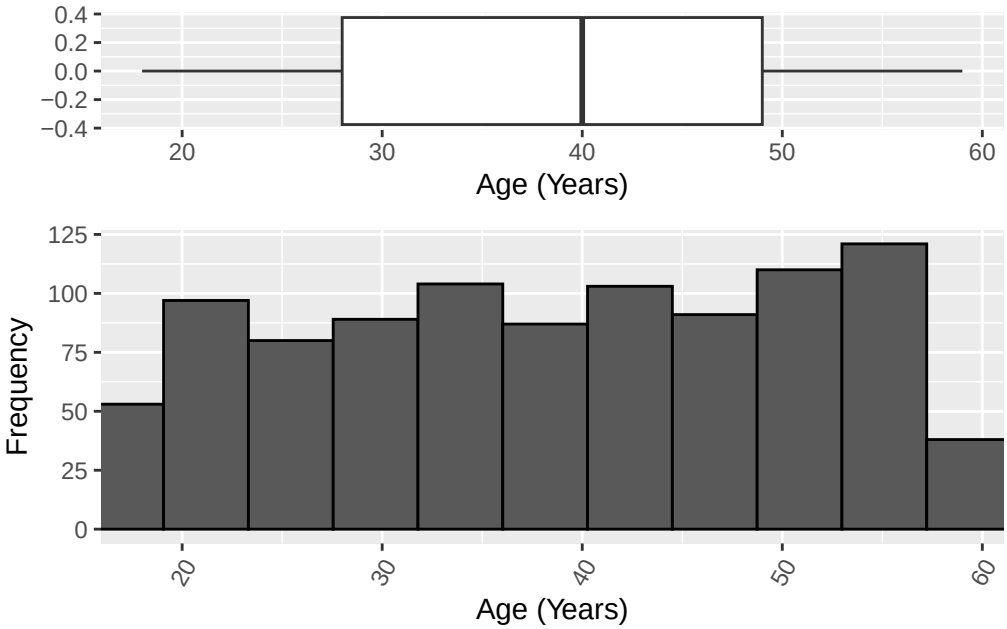


Análise Exploratória do Dataset

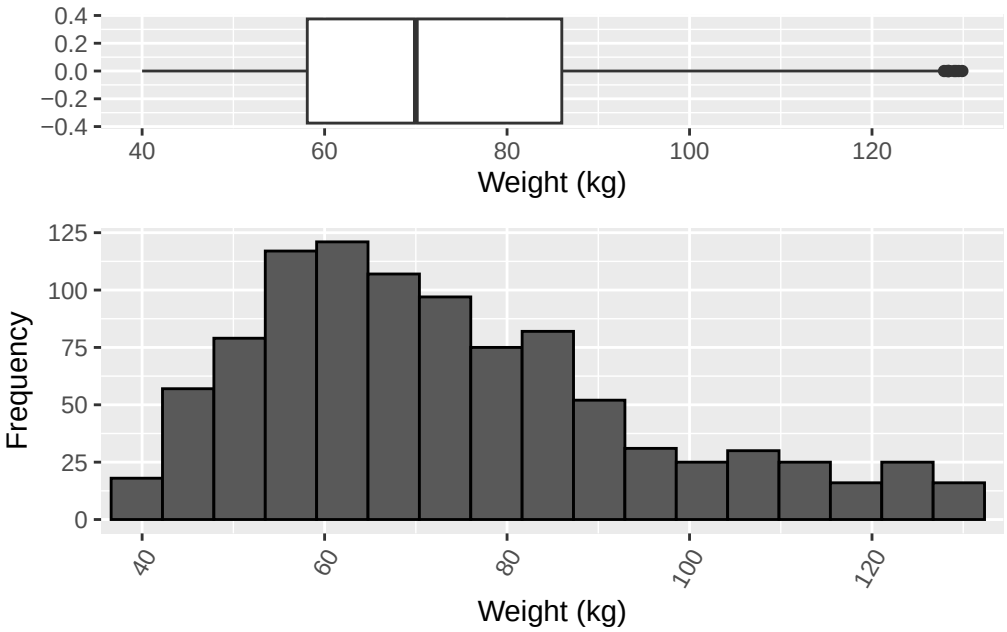
Análise Univariada

De modo a analisar a distribuição das nossas variáveis fizemos boxplot com histograma para cada variavel numérica. Podemos ver que a idade e as três variáveis referentes à frequência cardíaca dos mesmbros apresentam uma distribuição práticamente uniforme. Já as restantes variáveis apresentam distribuições com uma forma aproximadamente normal mas com assimetrias.

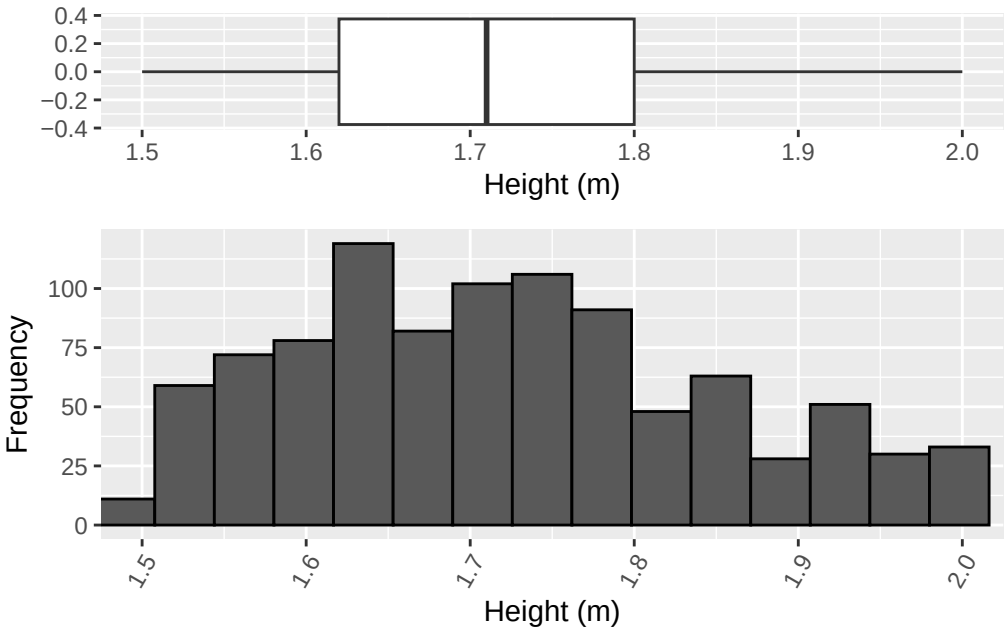
```
hist_and_box(df, df$Age, "Age (Years)")
```



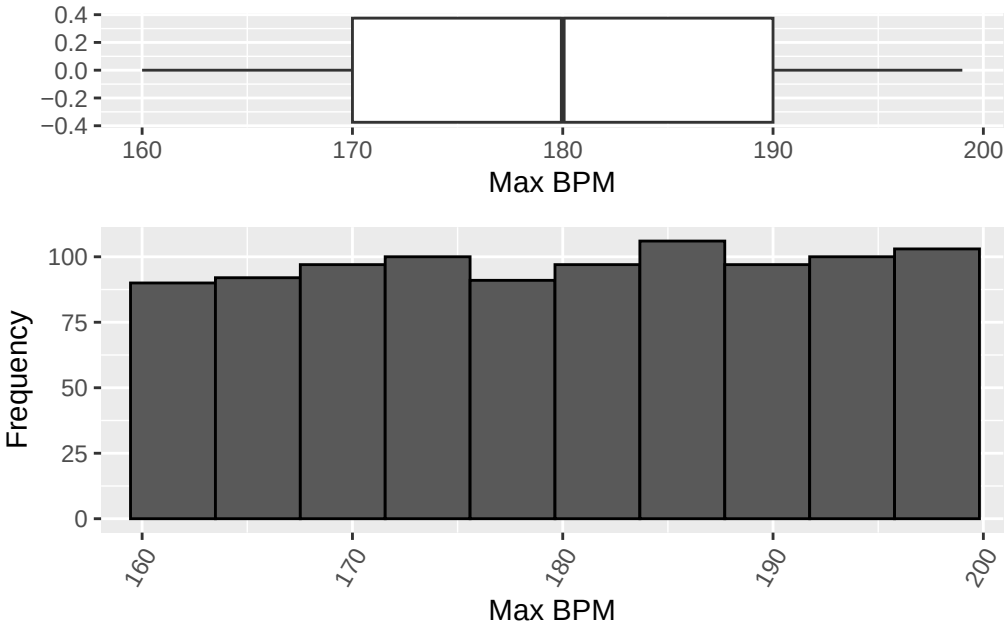

```
hist_and_box(df, df$Weight..kg., "Weight (kg)")
```



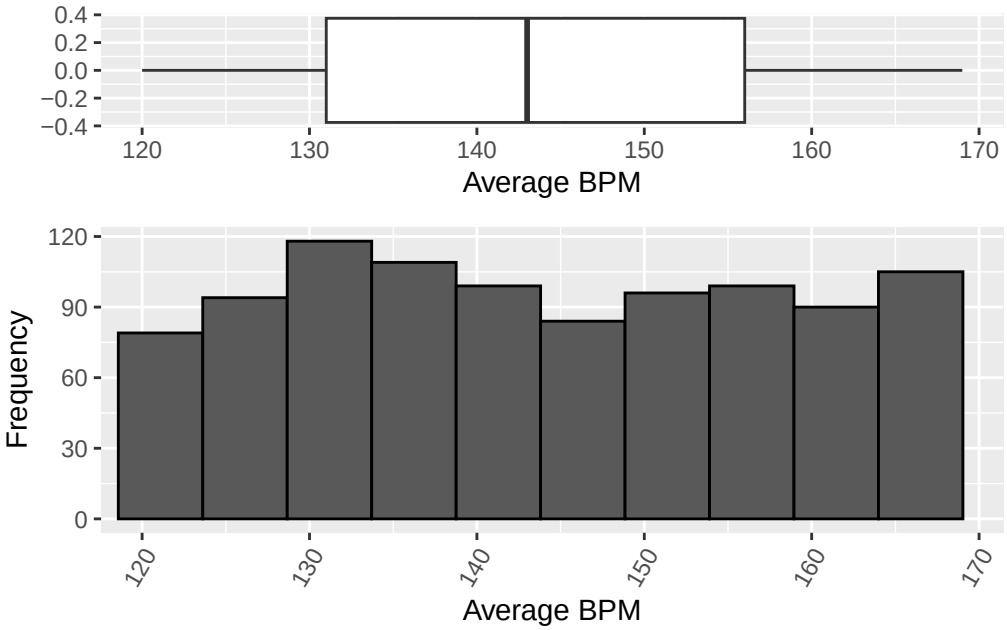
```
hist_and_box(df, df$Height..m., "Height (m)")
```



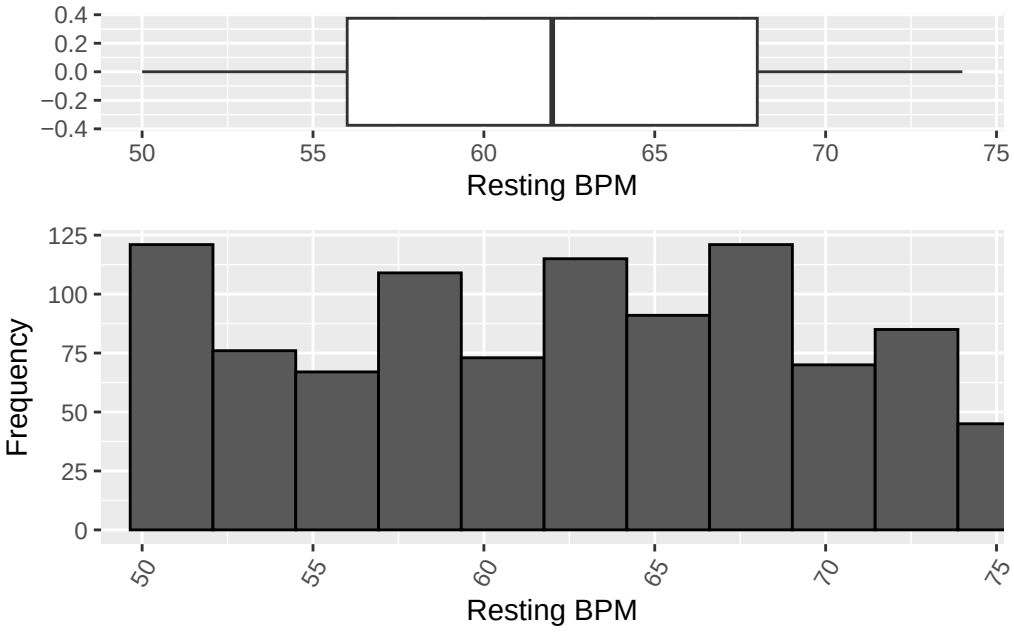

```
hist_and_box(df, df$Max_BPM, "Max BPM")
```



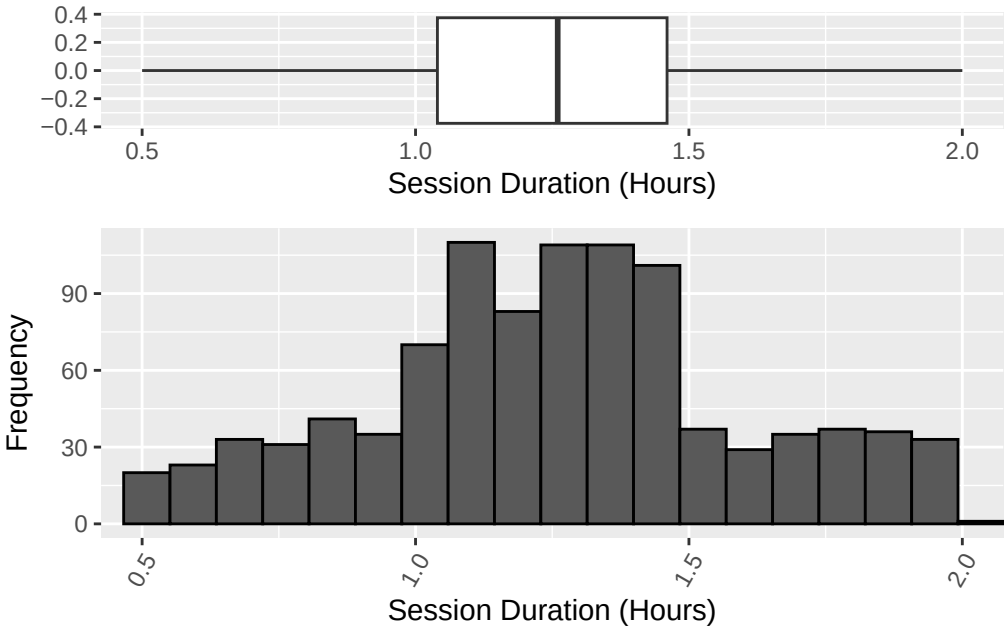
```
hist_and_box(df, df$Avg_BPM, "Average BPM")
```



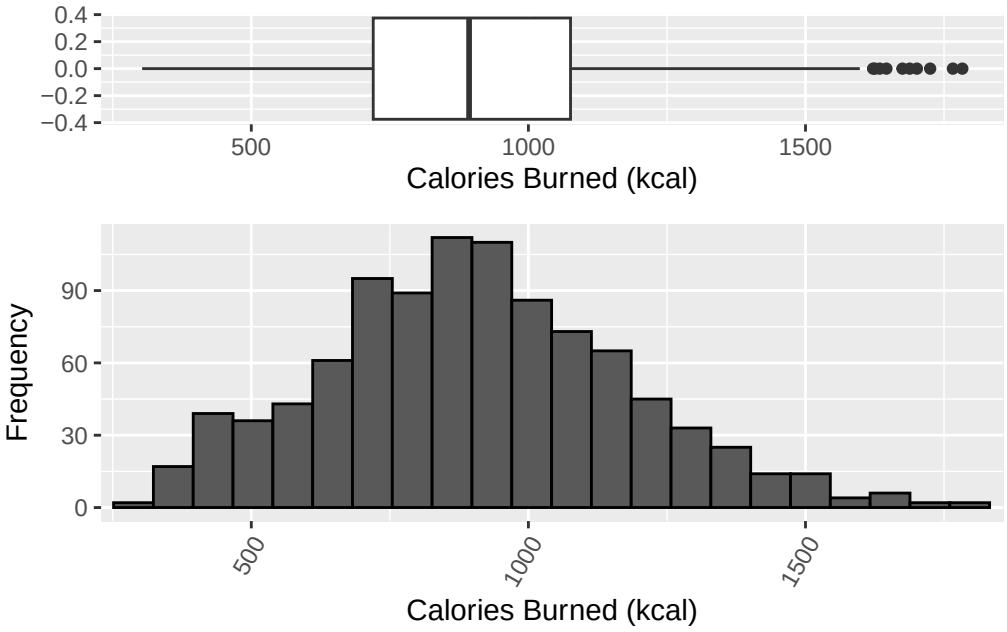

```
hist_and_box(df, df$Resting_BPM, "Resting BPM")
```



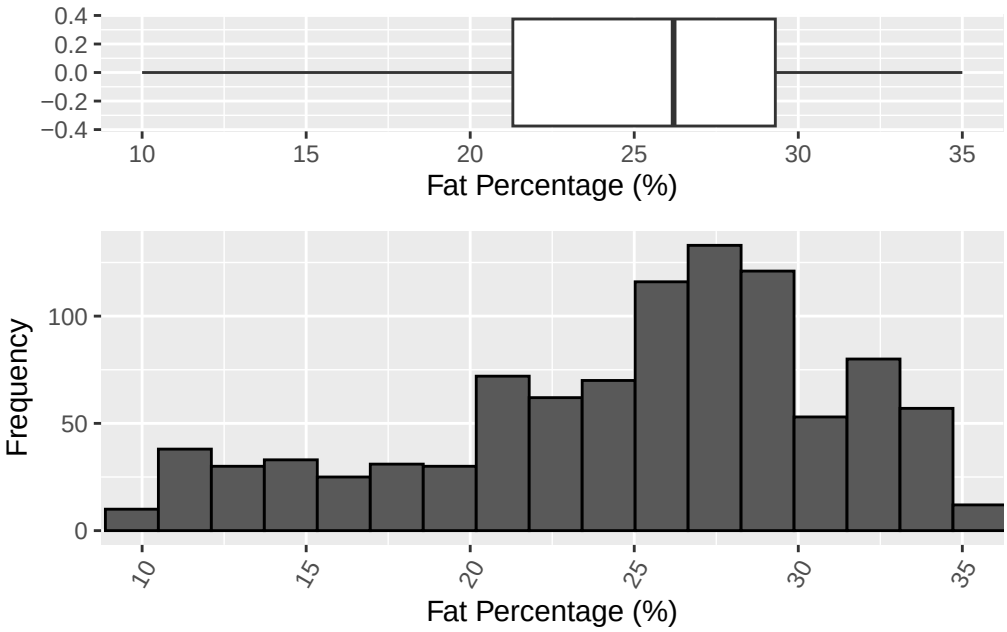
```r
hist_and_box(df, df$Session_Duration..hours., "Session Duration (Hours)")
```



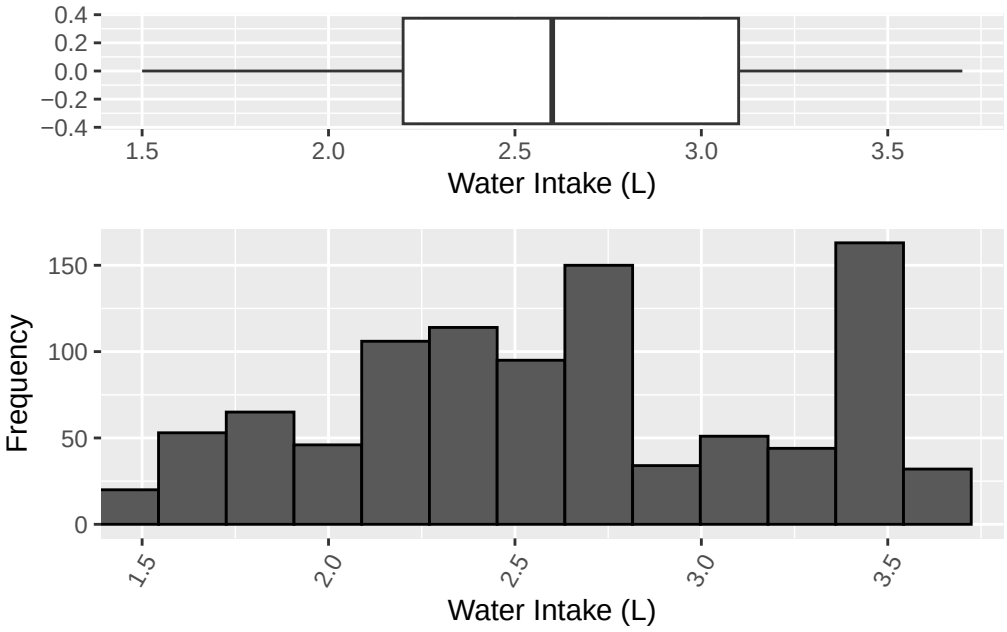

```r
hist_and_box(df, df$Calories_Burned, "Calories Burned (kcal)")
```



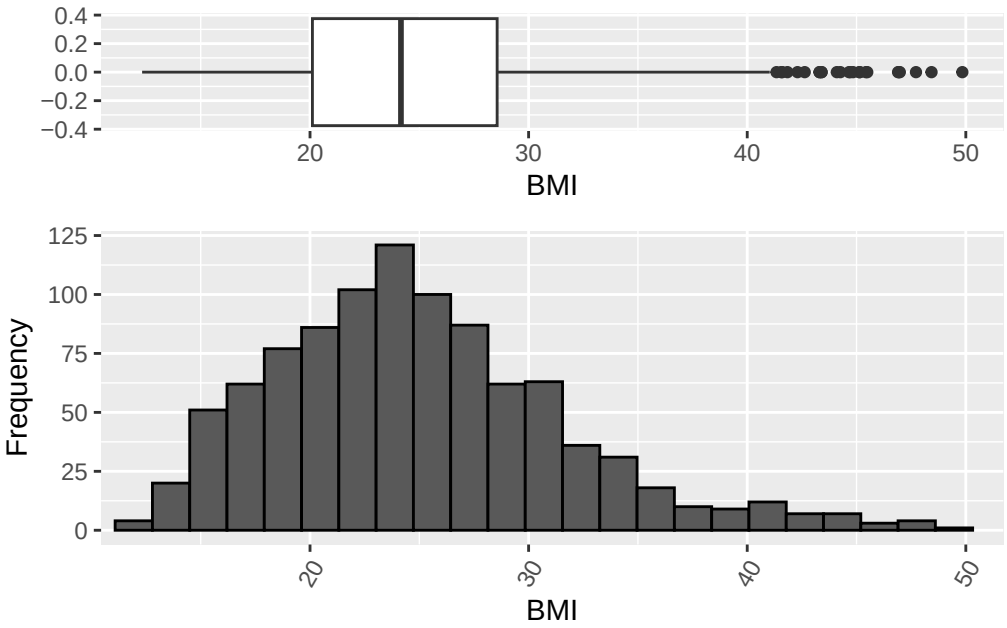
```
hist_and_box(df, df$Fat_Percentage, "Fat Percentage (%)")
```




```
hist_and_box(df, df$Water_Intake..liters., "Water Intake (L)")
```



```r
hist_and_box(df, df$BMI, "BMI")
```



A tabela seguinte mostra as estatísticas descritivas das variáveis numéricas do dataset.

```r
summary_stats <- data.frame(
  Variable = c("Age (Years)", "Weight (kg)", "Height (m)", "Max BPM", "Average BPM",
               "Resting BPM", "Session Duration (Hours)", "Calories Burned (kcal)",
               "Fat Percentage (%)", "Water Intake (l)", "BMI"),
  Mean = round(c(mean(df$Age, na.rm = TRUE), mean(df$Weight..kg., na.rm = TRUE),
                 mean(df$Height..m., na.rm = TRUE), mean(df$Max_BPM, na.rm = TRUE),
                 mean(df$Avg_BPM, na.rm = TRUE), mean(df$Resting_BPM, na.rm = TRUE),
                 mean(df$Session_Duration..hours., na.rm = TRUE), mean(df$Calories_Burned, na
                 mean(df$Fat_Percentage, na.rm = TRUE), mean(df$Water_Intake..liters., na.rm
                 mean(df$BMI, na.rm = TRUE)), 2),
  SD = round(c(sd(df$Age, na.rm = TRUE), sd(df$Weight..kg., na.rm = TRUE),
               sd(df$Height..m., na.rm = TRUE), sd(df$Max_BPM, na.rm = TRUE),
               sd(df$Avg_BPM, na.rm = TRUE), sd(df$Resting_BPM, na.rm = TRUE),
               sd(df$Session_Duration..hours., na.rm = TRUE), sd(df$Calories_Burned, na.rm =
               sd(df$Fat_Percentage, na.rm = TRUE), sd(df$Water_Intake..liters., na.rm = TRU
               sd(df$BMI, na.rm = TRUE)), 2),
  Median = round(c(median(df$Age, na.rm = TRUE), median(df$Weight..kg., na.rm = TRUE),
                   median(df$Height..m., na.rm = TRUE), median(df$Max_BPM, na.rm = TRUE),
                   median(df$Avg_BPM, na.rm = TRUE), median(df$Resting_BPM, na.rm = TRUE),
```

```
                  median(df$Session_Duration..hours., na.rm = TRUE), median(df$Calories_Burn
                  median(df$Fat_Percentage, na.rm = TRUE), median(df$Water_Intake..liters.,
                  median(df$BMI, na.rm = TRUE)), 2),
  IQR = round(c(IQR(df$Age, na.rm = TRUE), IQR(df$Weight..kg., na.rm = TRUE),
                IQR(df$Height..m., na.rm = TRUE), IQR(df$Max_BPM, na.rm = TRUE),
                IQR(df$Avg_BPM, na.rm = TRUE), IQR(df$Resting_BPM, na.rm = TRUE),
                IQR(df$Session_Duration..hours., na.rm = TRUE), IQR(df$Calories_Burned, na.rm
                IQR(df$Fat_Percentage, na.rm = TRUE), IQR(df$Water_Intake..liters., na.rm = T
                IQR(df$BMI, na.rm = TRUE)), 2),
  Min = round(c(min(df$Age, na.rm = TRUE), min(df$Weight..kg., na.rm = TRUE),
                min(df$Height..m., na.rm = TRUE), min(df$Max_BPM, na.rm = TRUE),
                min(df$Avg_BPM, na.rm = TRUE), min(df$Resting_BPM, na.rm = TRUE),
                min(df$Session_Duration..hours., na.rm = TRUE), min(df$Calories_Burned, na.rm
                min(df$Fat_Percentage, na.rm = TRUE), min(df$Water_Intake..liters., na.rm = T
                min(df$BMI, na.rm = TRUE)), 2),
  Max = round(c(max(df$Age, na.rm = TRUE), max(df$Weight..kg., na.rm = TRUE),
                max(df$Height..m., na.rm = TRUE), max(df$Max_BPM, na.rm = TRUE),
                max(df$Avg_BPM, na.rm = TRUE), max(df$Resting_BPM, na.rm = TRUE),
                max(df$Session_Duration..hours., na.rm = TRUE), max(df$Calories_Burned, na.rm
                max(df$Fat_Percentage, na.rm = TRUE), max(df$Water_Intake..liters., na.rm = T
                max(df$BMI, na.rm = TRUE)), 2)
)

# Exibir a tabela com kable (na ordem: Mean, SD, Median, IQR, Min, Max)
kable(summary_stats, format = "pipe", digits = 2, caption = "Estatísticas Descritivas das Var
```

Table 1: Estatísticas Descritivas das Variáveis

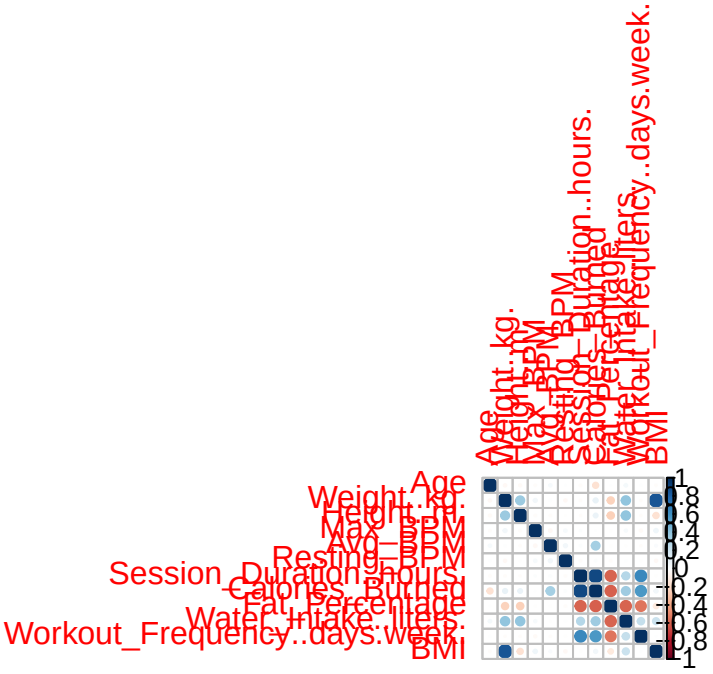
Variable	Mean	SD	Median	IQR	Min	Max
Age (Years)	38.68	12.18	40.00	21.00	18.00	59.00
Weight (kg)	73.85	21.21	70.00	27.90	40.00	129.90
Height (m)	1.72	0.13	1.71	0.18	1.50	2.00
Max BPM	179.88	11.53	180.00	20.00	160.00	199.00
Average BPM	143.77	14.35	143.00	25.00	120.00	169.00
Resting BPM	62.22	7.33	62.00	12.00	50.00	74.00
Session Duration (Hours)	1.26	0.34	1.26	0.42	0.50	2.00
Calories Burned (kcal)	905.42	272.64	893.00	356.00	303.00	1783.00
Fat Percentage (%)	24.98	6.26	26.20	8.00	10.00	35.00
Water Intake (l)	2.63	0.60	2.60	0.90	1.50	3.70
BMI	24.91	6.66	24.16	8.45	12.32	49.84

Análise de Correlação

A matriz de correlação permite-nos perceber a relação entre as variáveis numéricas do dataset. Através da matriz de correlação conseguimos perceber quais as variáveis que estão mais correlacionadas entre si.

As variáveis com maior correlação são as Calorias queimadas durante o exercício e a duração da sessão de treino, com uma correlação de 0.91. Este valor é expéctavel uma vez que quanto mais tempo de treino, mais calorias são queimadas. A massa e a o índice de massa corporal também apresentam uma forte correlação positiva. Por outro lado, a maior correlação negativa é entre as calorias queimadas e a percentagem de massa gorda, sendo esta correlação quase tão fortes como a correlação entre a ingestão de água e a percentagem de massa gorda.

```
cor_matrix <- cor(df_numeric, use = "pairwise.complete.obs")
cor_long <- as.data.frame(as.table(cor_matrix))
short_names <- abbreviate(colnames(cor_matrix), minlength = 6)

corrplot(cor_matrix)
```




```
cor_long <- cor_long %>% filter(Var1 != Var2)

cor_long <- cor_long %>%
  mutate(pair = apply(cor_long[, 1:2], 1, function(x) paste(sort(x), collapse = "_"))) %>%
```

```
  distinct(pair, .keep_all = TRUE) %>%
  select(-pair)


top_5 <- cor_long %>% arrange(desc(Freq)) %>% head(5)
bottom_5 <- cor_long %>% arrange(Freq) %>% head(5)

print("Top 5 maiores correlações:")
```

```
[1] "Top 5 maiores correlações:"
```

```
print(top_5)
```

```
                          Var1                      Var2      Freq
1               Calories_Burned Session_Duration..hours. 0.9081404
2                           BMI              Weight..kg. 0.8531577
3 Workout_Frequency..days.week. Session_Duration..hours. 0.6441404
4 Workout_Frequency..days.week.           Calories_Burned 0.5761501
5         Water_Intake..liters.              Weight..kg. 0.3942757
```

```
print("Top 5 menores correlações:")
```

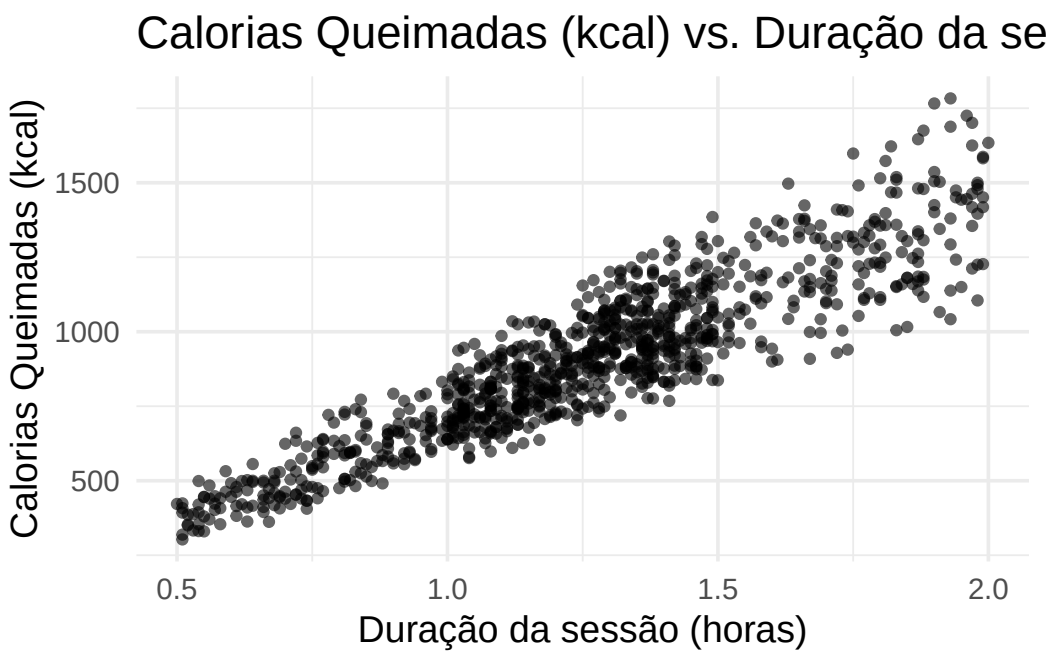
```
[1] "Top 5 menores correlações:"
```

```
print(bottom_5)
```

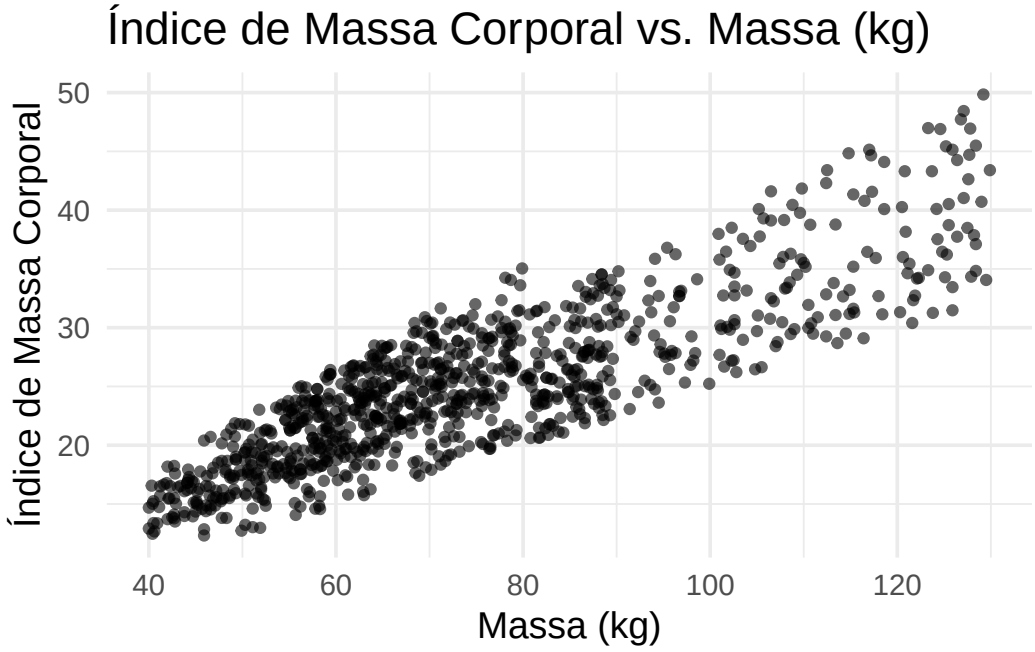
```
                          Var1                      Var2       Freq
1               Fat_Percentage           Calories_Burned -0.5976152
2         Water_Intake..liters.            Fat_Percentage -0.5886828
3               Fat_Percentage Session_Duration..hours. -0.5815198
4 Workout_Frequency..days.week.            Fat_Percentage -0.5370595
5               Fat_Percentage                Height..m. -0.2355209
```

```
scatter_plot <- function(data, x_var, y_var, x_label, y_label, title = "Scatter Plot") {
  ggplot(data, aes(x = x_var, y = y_var)) +
    geom_point(color = "black", alpha = 0.6) +  # Blue points with transparency
    labs(x = x_label, y = y_label, title = paste(y_label, "vs.", x_label)) +
    theme_minimal(base_size = 14)  # Clean theme with readable font size
}
```

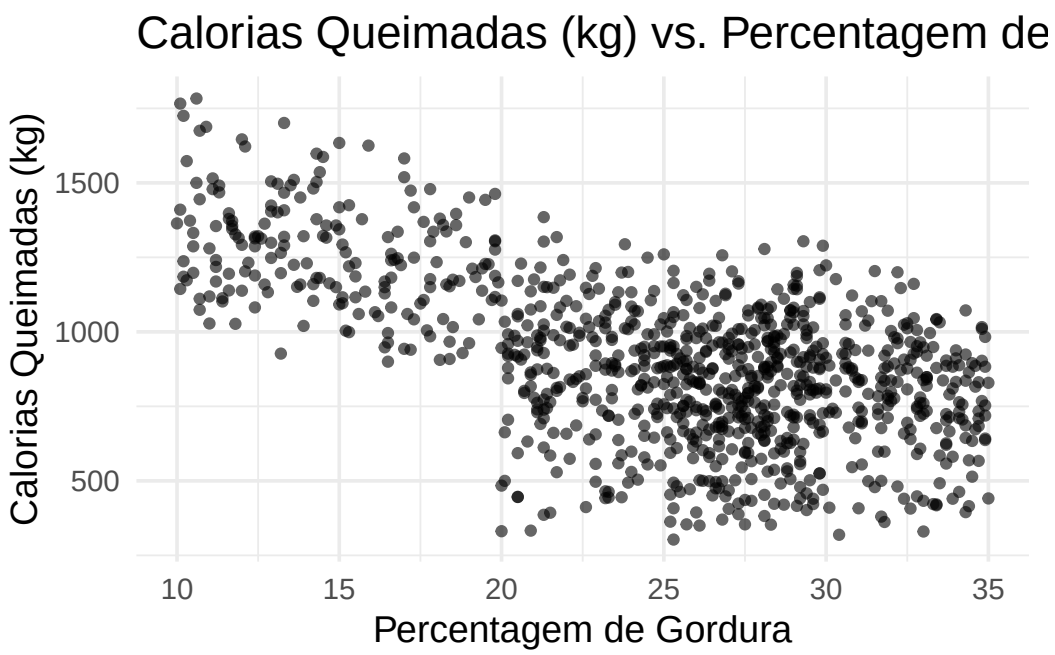
```
scatter_plot(df, df$Session_Duration..hours., df$Calories_Burned, "Duração da sessão (horas)"
```



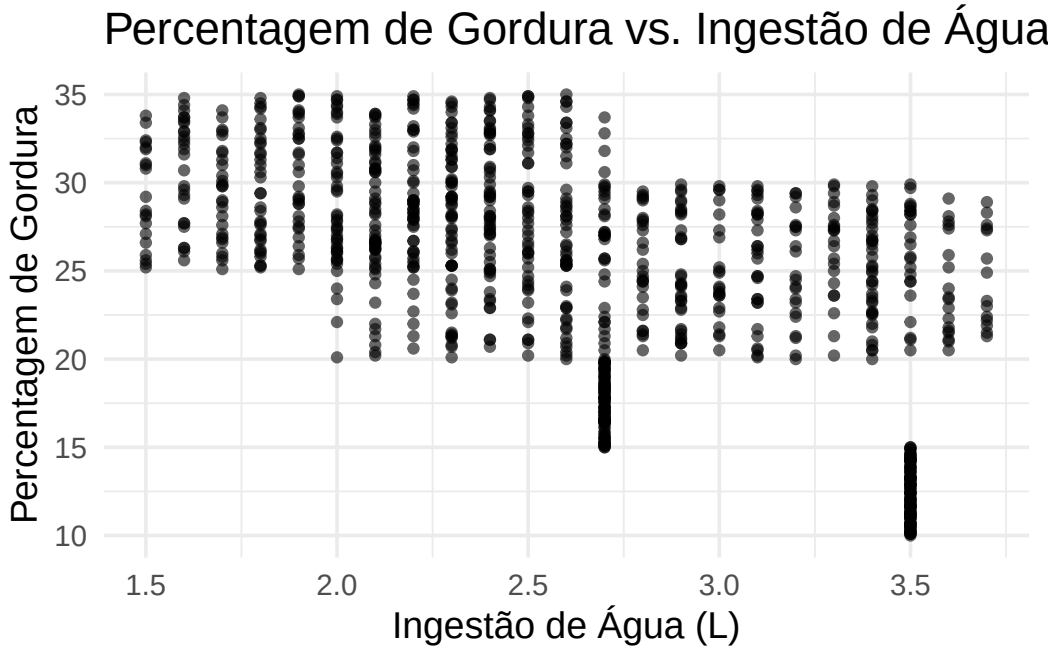

```
scatter_plot(df, df$Weight..kg., df$BMI, "Massa (kg)", "Índice de Massa Corporal")
```



```
scatter_plot(df, df$Fat_Percentage, df$Calories_Burned, "Percentagem de Gordura", "Calorias (
```




```
scatter_plot(df, df$Water_Intake..liters., df$Fat_Percentage, "Ingestão de Água (L)", "Percen
```



Análise das Componentes Principais

A Análise das Componentes Principais (PCA) é uma técnica de redução de dimensionalidade que permite representar os dados num espaço de dimensão inferior. A PCA é uma técnica muito útil para visualizar a estrutura dos dados e identificar padrões. Para realizar a PCA, é necessário normalizar os dados, de modo a que todas as variáveis tenham a mesma escala.

A tabela seguinte mostra o resumo da PCA, onde é possível ver a variância explicada por cada componente principal. A primeira componente principal explica 28% da variância total, enquanto a segunda componente principal explica 17% da variância total. Juntas, as duas primeiras componentes principais explicam 44% da variância total.

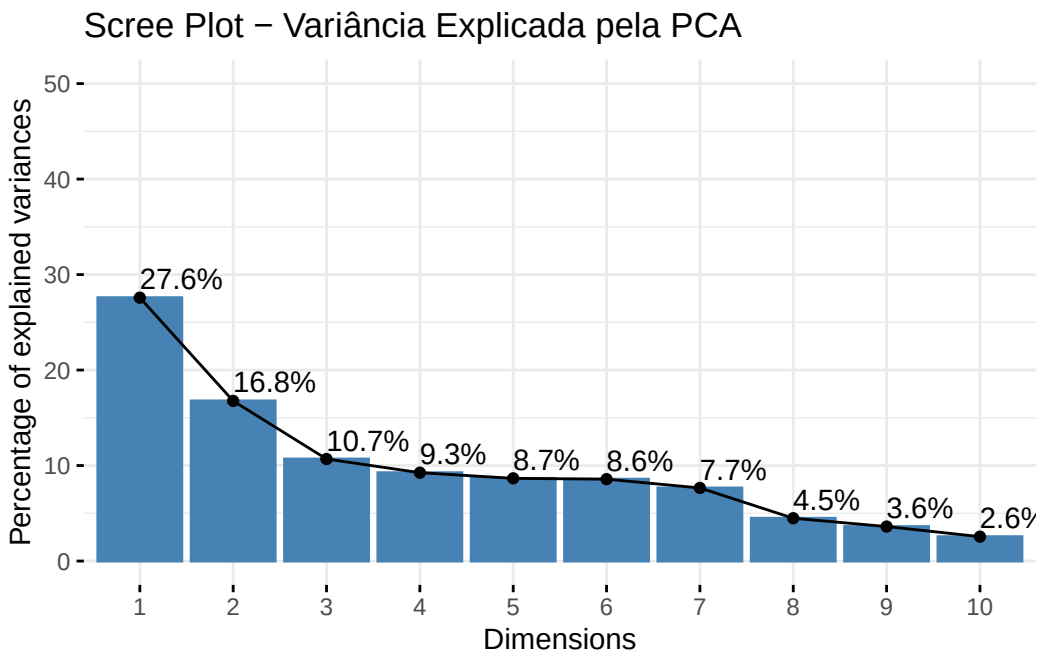
```
df_scaled <- scale(df_numeric)
pca <- prcomp(df_scaled, center = TRUE, scale. = TRUE)
pca_summary <- round(summary(pca)$importance, 2)
kable(pca_summary, caption = "Resumo da ACP")
```

Table 2: Resumo da ACP

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Standard deviation	1.82	1.42	1.13	1.05	1.02	1.01	0.96	0.73	0.66	0.55	0.12	0.08
Proportion of Variance	0.28	0.17	0.11	0.09	0.09	0.09	0.08	0.04	0.04	0.03	0.00	0.00
Cumulative Proportion	0.28	0.44	0.55	0.64	0.73	0.82	0.89	0.94	0.97	1.00	1.00	1.00

Pela análise do Scree Plot, é possível perceber que a partir da terceira componente principal, a variância explicada por cada componente diminui consideravelmente. Assim, podemos concluir que as três primeiras componentes principais são as mais importantes para explicar a variância dos dados.

```
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 50)) +
  labs(title = "Scree Plot - Variância Explicada pela PCA")
```




```
loadings_3_pcas <- pca$rotation[, 1:3]
kable(round(loadings_3_pcas,3), caption = "Loadings dos 3 Primeiros Componentes Principais")
```

Table 3: Loadings dos 3 Primeiros Componentes Principais

	PC1	PC2	PC3
Age	-0.033	0.004	0.090
Weight..kg.	0.192	0.634	-0.110
Height..m.	0.142	0.229	0.689
Max_BPM	0.011	0.081	-0.106
Avg_BPM	0.073	-0.061	-0.255
Resting_BPM	0.001	-0.034	0.009
Session_Duration..hours.	0.453	-0.274	-0.135
Calories_Burned	0.478	-0.201	-0.168
Fat_Percentage	-0.458	-0.003	-0.151
Water_Intake..liters.	0.355	0.252	0.329
Workout_Frequency..days.week.	0.387	-0.238	-0.089
BMI	0.130	0.546	-0.494

A tabela seguinte mostra as três variáveis mais influentes em cada componente principal. A primeira componente principal é influenciada principalmente pelas calorias queimadas, pela

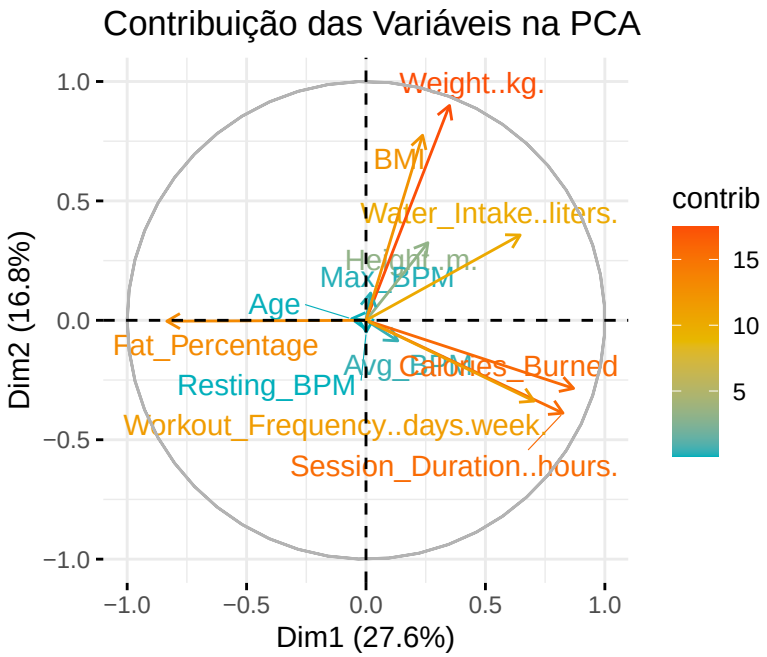
percentagem de massa gorda e pela duração do treino. A segunda componente principal é influenciada principalmente pela massa, pelo índice de massa corporal e pela duração do treino. Já a terceira componente principal é influenciada principalmente pela altura, pelo índice de massa corporal e pela ingestão diária de água.

```
top_variables <- apply(loadings_3_pcas, 2, function(x) {
  names(sort(abs(x), decreasing = TRUE)[1:3])
})


top_variables_df <- as.data.frame(top_variables)
colnames(top_variables_df) <- c("PC1", "PC2", "PC3")

kable(top_variables_df, caption = "Três Variáveis Mais Influentes em Cada Componente Principa
```

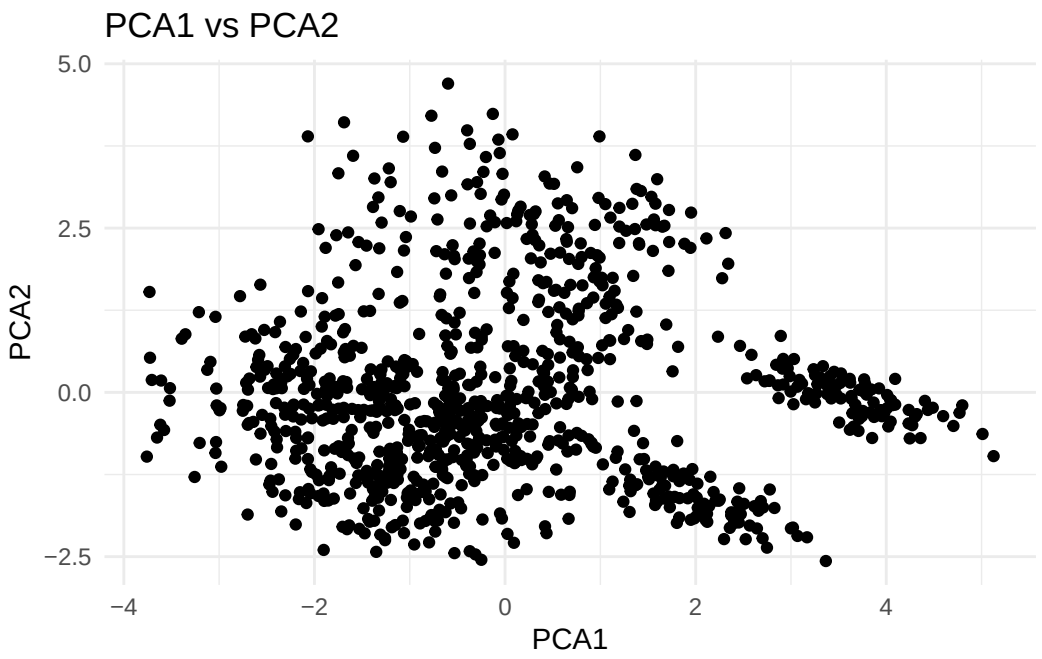
Table 4: Três Variáveis Mais Influentes em Cada Componente Principal

PC1	PC2	PC3
Calories_Burned	Weight..kg.	Height..m.
Fat_Percentage	BMI	BMI
Session_Duration..hours.	Session_Duration..hours.	Water_Intake..liters.

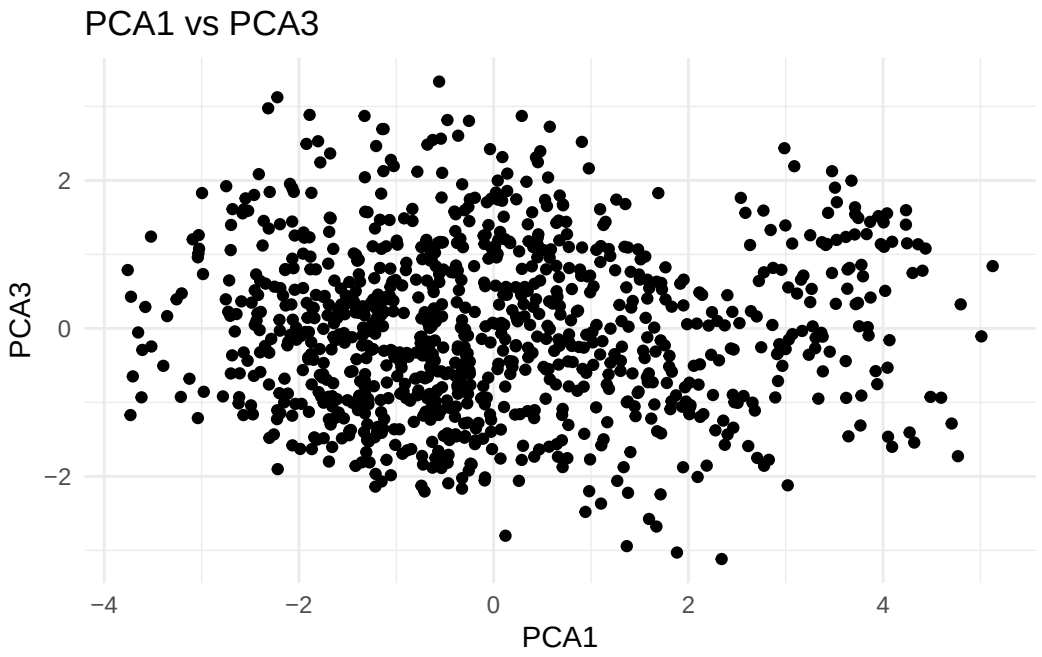
```
fviz_pca_var(pca,
             col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE) +
  labs(title = "Contribuição das Variáveis na PCA")
```



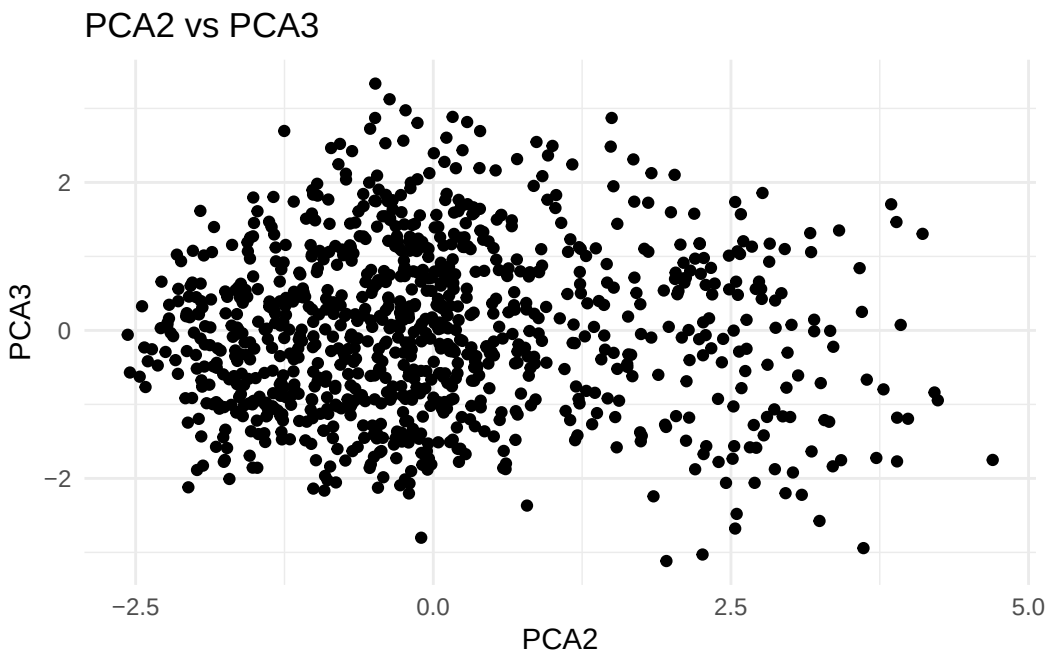

```r
pca_scores <- pca$x[, 1:3]

ggplot(data.frame(PCA1 = pca_scores[, 1], PCA2 = pca_scores[, 2]), aes(x = PCA1, y = PCA2)) +
  geom_point() +
  labs(x = "PCA1", y = "PCA2", title = "PCA1 vs PCA2") +
  theme_minimal()
```




```
ggplot(data.frame(PCA1 = pca_scores[, 1], PCA3 = pca_scores[, 3]), aes(x = PCA1, y = PCA3)) +
  geom_point() +
  labs(x = "PCA1", y = "PCA3", title = "PCA1 vs PCA3") +
  theme_minimal()
```




```
ggplot(data.frame(PCA2 = pca_scores[, 2], PCA3 = pca_scores[, 3]), aes(x = PCA2, y = PCA3)) +
  geom_point() +
  labs(x = "PCA2", y = "PCA3", title = "PCA2 vs PCA3") +
  theme_minimal()
```



Clusterização

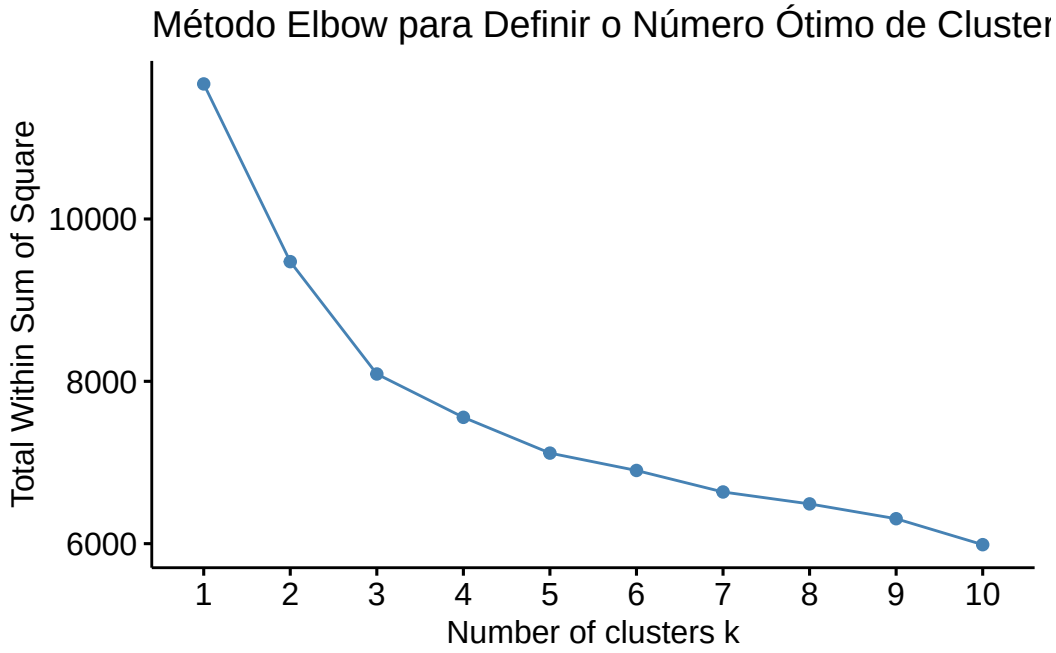
A clusterização é uma técnica de aprendizagem não supervisionada que permite agrupar os dados em grupos com características semelhantes. Para realizar a clusterização, é necessário normalizar os dados, de modo a que todas as variáveis tenham a mesma escala. Iremos testar diferentes métodos de clusterização, nomeadamente o K-means, o DBSCAN, o Hierarchical Clustering, o método de mistura de Gaussianas e o método de clustering espectral.

K-means

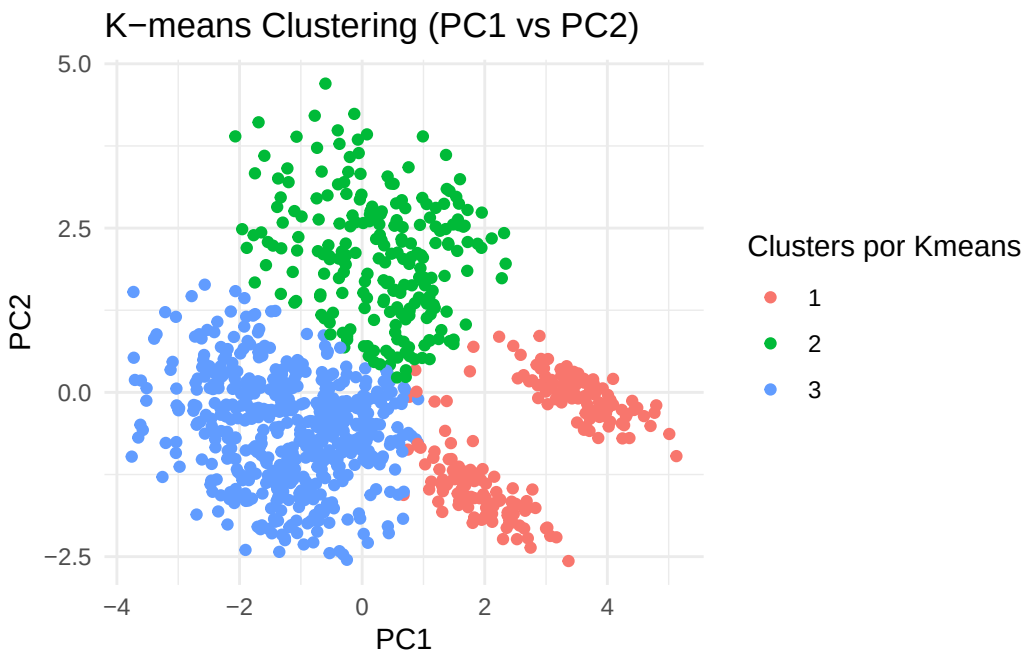
O K-means é um dos métodos de clusterização mais populares. O K-means agrupa os dados em K clusters, onde K é um número pré-definido. O K-means é um método iterativo que tenta minimizar a soma dos quadrados das distâncias entre os pontos e os centroides dos clusters.

Através do gráfico e pelo método do cotovelo, é possível perceber que o número ótimo de clusters é 3.

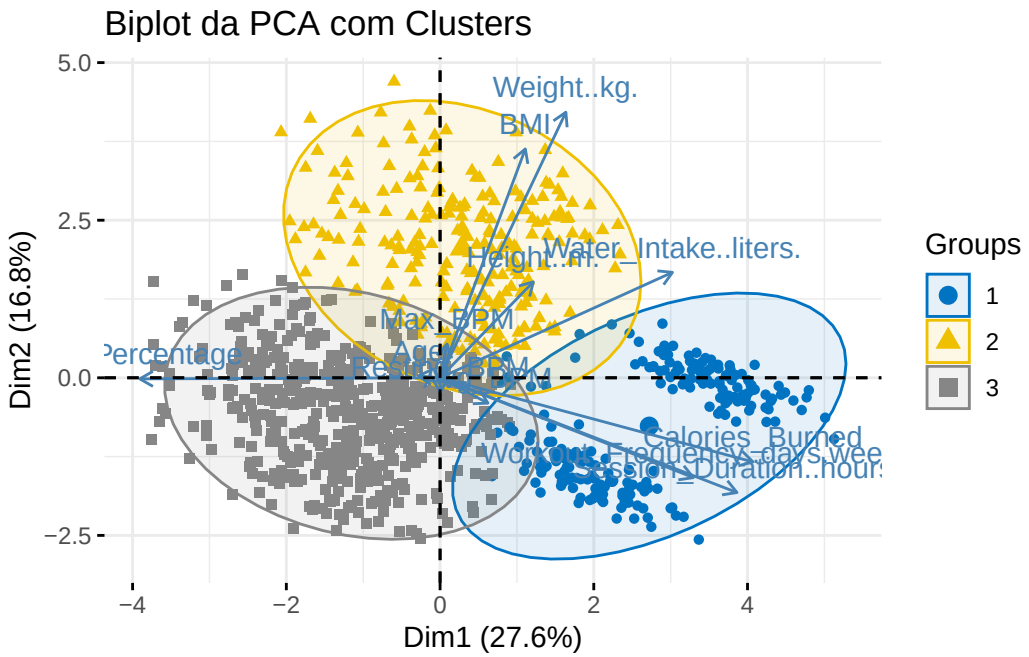
```
fviz_nbclust(df_scaled, kmeans, method = "wss") +
  labs(title = "Método Elbow para Definir o Número Ótimo de Clusters")
```




```r
kmeans_clusters <- kmeans(df_scaled, centers = 3, nstart = 25)
df$kmeanscluster <- as.factor(kmeans_clusters$cluster)

pca_scores <- as.data.frame(pca$x[, 1:3])

pca_scores$kmeanscluster <- as.factor(kmeans_clusters$cluster)

ggplot(pca_scores, aes(x = PC1, y = PC2, color = kmeanscluster)) +
  geom_point() +
  labs(title = "K-means Clustering (PC1 vs PC2)", color= "Clusters por Kmeans") +
  theme_minimal()
```




```
fviz_pca_biplot(pca,
                label = "var",
                habillage = df$kmeanscluster,
                addEllipses = TRUE,
                ellipse.level = 0.95,
                palette = "jco") +
  labs(title = "Biplot da PCA com Clusters")
```



Através de testes ANOVA e Tukey conseguimos ver que as variáveis que são significativas para a clusterização são:

- `Weight` - Que é significativamente diferente entre os 3 clusters.
- `Height` - Que é significativamente diferente entre os 3 clusters.
- `Session Duartion` - Que é significativamente diferente entre os clusters 1-2 e entre os clusters 1-3, não havendo evidência estatística para diferença entre os clusters 3-2.
- `Calories Burned` - Que é significativamente diferente entre os 3 clusters.
- `Fat Percentage` - Que é significativamente diferente entre os 3 clusters.
- `Water Intake` - Que é significativamente diferente entre os 3 clusters.
- `Workout Frequency` - Que é significativamente diferente entre os clusters 1-1 e entre os clusters 1-3, não havendo evidência estatística para diferença entre os clusters 3-2.
- `BMI` - Que é significativamente diferente entre os 3 clusters.

```
# Realizando a ANOVA para cada variável numérica e depois o teste de Tukey
resultados_posthoc <- lapply(df[, sapply(df, is.numeric)], function(x) {
  aov_result <- aov(x ~ kmeanscluster, data = df)

  # Realizar o teste de Tukey se a ANOVA for significativa
  if (summary(aov_result)[[1]]$`Pr(>F)`[1] < 0.05) {
```

```r
    tukey_result <- TukeyHSD(aov_result)
    return(tukey_result)
  } else {
    return(NULL)  # Se não for significativo, retornamos NULL
  }
})

# Filtrando variáveis com resultados de Tukey
resultados_posthoc_significativos <- resultados_posthoc[sapply(resultados_posthoc, function(

# Função para gerar uma tabela com os resultados do Tukey
tukey_table <- function(var_name, tukey_result) {
  # Criando a tabela para a variável
  result_df <- data.frame(
    Comparação = rownames(tukey_result),
    Diferença_Média = tukey_result[, "diff"],
    Intervalo_Confiança = paste("[", round(tukey_result[, "lwr"], 2), ", ", round(tukey_resul
    Valor_p_Ajustado = round(tukey_result[, "p adj"], 3)
  )

  # Renomeando as colunas
  colnames(result_df) <- c("Comparação", paste("Diferença Média (", var_name, ")", sep = "")

  return(result_df)
}

# Aplicando a função para cada variável significativa
resultados_tukey_all <- lapply(names(resultados_posthoc_significativos), function(var_name) {
  tukey_result <- resultados_posthoc_significativos[[var_name]]
  tukey_table(var_name, tukey_result[[1]])
})

# Gerando a tabela simples em LaTeX
lapply(resultados_tukey_all, function(tabela) {
  var_name <- names(tabela)[2]

  # Gerando a tabela com kable (formato latex)
  kable(tabela,
        caption = paste("Resultados do Teste de Tukey para a variável", var_name),
        digits = c(0, 2, 0, 3),
        align = "lccc",
        col.names = c("Comparação", "Diferença Média", "Intervalo de Confiança", "Valor p Aju
```

```
})
```

```
[[1]]
```

Table: Resultados do Teste de Tukey para a variável Diferença Média (Weight..kg.)

	Comparação	Diferença Média	Intervalo de Confiança	Valor p Ajustado
2-1	2-1	30.85	[28.02, 33.68]	0
3-1	3-1	-11.52	[-13.92, -9.12]	0
3-2	3-2	-42.37	[-44.7, -40.04]	0

```
[[2]]
```

Table: Resultados do Teste de Tukey para a variável Diferença Média (Height..m.)

	Comparação	Diferença Média	Intervalo de Confiança	Valor p Ajustado
2-1	2-1	0.06	[0.04, 0.09]	0
3-1	3-1	-0.04	[-0.06, -0.02]	0
3-2	3-2	-0.10	[-0.13, -0.08]	0

```
[[3]]
```

Table: Resultados do Teste de Tukey para a variável Diferença Média (Session_Duration..hours

	Comparação	Diferença Média	Intervalo de Confiança	Valor p Ajustado
2-1	2-1	-0.60	[-0.65, -0.55]	0.000
3-1	3-1	-0.61	[-0.65, -0.56]	0.000
3-2	3-2	-0.01	[-0.05, 0.04]	0.888

```
[[4]]
```

Table: Resultados do Teste de Tukey para a variável Diferença Média (Calories_Burned)

	Comparação	Diferença Média	Intervalo de Confiança	Valor p Ajustado

2-1	2-1	-401.88	[-447.91, -355.84]	0.000
3-1	3-1	-457.81	[-496.88, -418.74]	0.000
3-2	3-2	-55.94	[-93.79, -18.08]	0.002

[[5]]

Table: Resultados do Teste de Tukey para a variável Diferença Média (Fat_Percentage)

	Comparação	Diferença Média	Intervalo de Confiança	Valor p Ajustado
2-1	2-1	9.67	[8.87, 10.47]	0
3-1	3-1	13.18	[12.5, 13.86]	0
3-2	3-2	3.51	[2.85, 4.17]	0

[[6]]

Table: Resultados do Teste de Tukey para a variável Diferença Média (Water_Intake..liters.)

	Comparação	Diferença Média	Intervalo de Confiança	Valor p Ajustado
2-1	2-1	-0.15	[-0.26, -0.05]	0.002
3-1	3-1	-0.83	[-0.92, -0.74]	0.000
3-2	3-2	-0.68	[-0.76, -0.59]	0.000

[[7]]

Table: Resultados do Teste de Tukey para a variável Diferença Média (Workout_Frequency..days

	Comparação	Diferença Média	Intervalo de Confiança	Valor p Ajustado
2-1	2-1	-1.53	[-1.68, -1.37]	0.000
3-1	3-1	-1.45	[-1.58, -1.32]	0.000
3-2	3-2	0.08	[-0.05, 0.21]	0.329

[[8]]

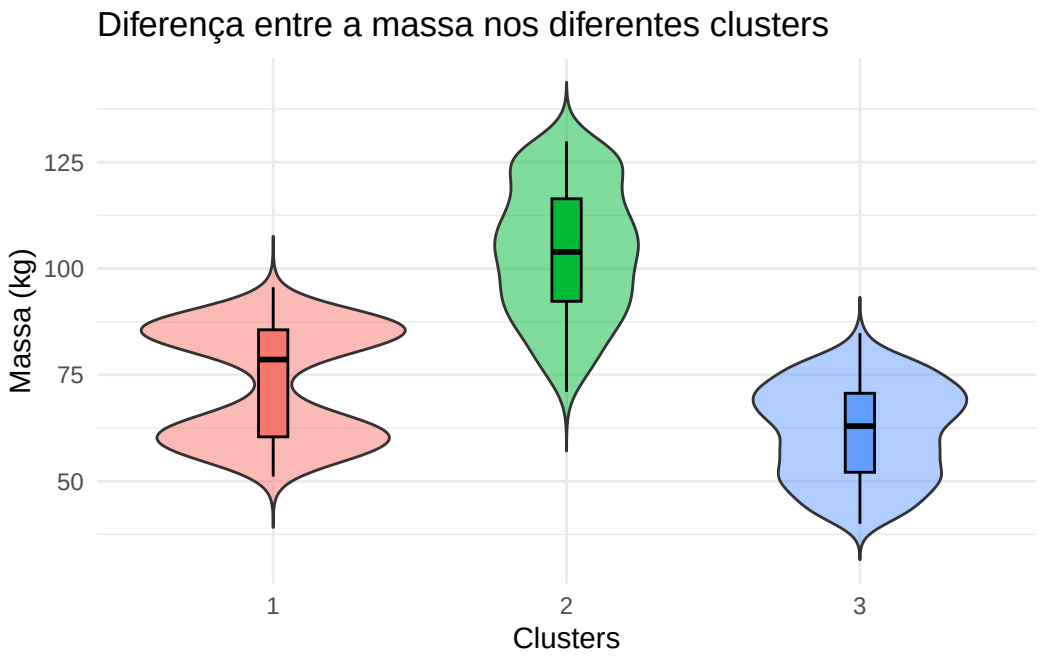
Table: Resultados do Teste de Tukey para a variável Diferença Média (BMI)

| |Comparação | Diferença Média | Intervalo de Confiança | Valor p Ajustado |

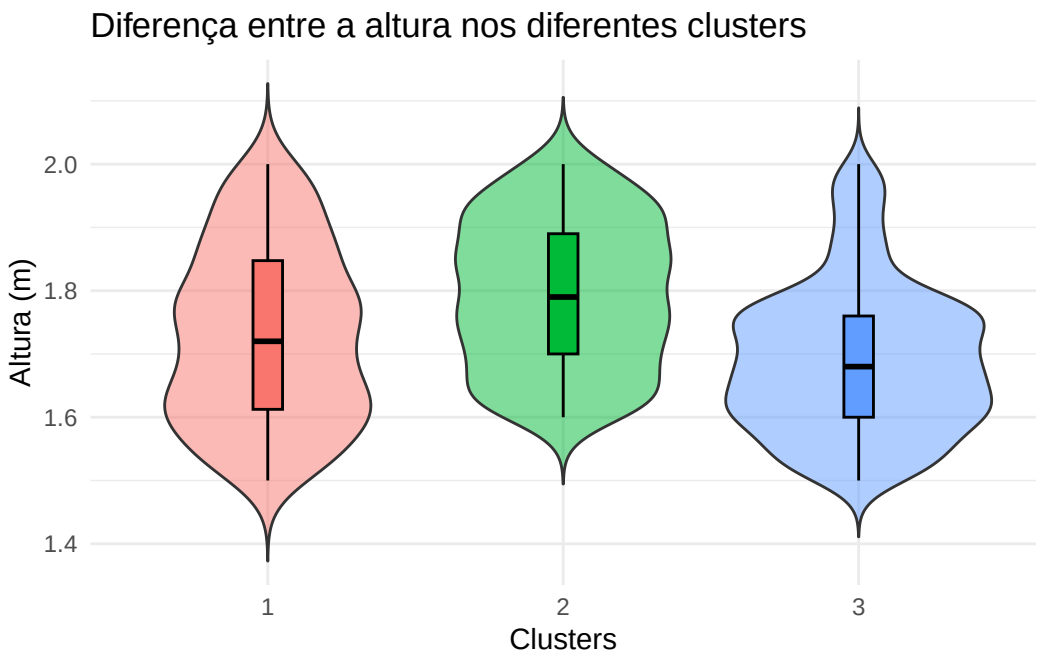
```
|:---|:----------|:--------------:|:---------------------:|:---------------:|
|2-1 |2-1        |      8.12      |      [6.99, 9.26]     |        0        |
|3-1 |3-1        |     -2.72      |     [-3.68, -1.75]    |        0        |
|3-2 |3-2        |    -10.84      |     [-11.77, -9.9]    |        0        |
```

```r
violin_boxplot <- function(df, variavel, separacao, titulo, xtitulo, ytitulo) {
  ggplot(df, aes(x = separacao, y = variavel, fill = separacao)) +
    geom_violin(trim = FALSE, alpha = 0.5) +  # Gráfico de violino
    geom_boxplot(width = 0.1, outlier.shape = NA, color = "black") +  # Boxplot embutido
    labs(title = titulo, x = xtitulo, y = ytitulo) +
    theme_minimal() +
    theme(legend.position = "none")  # Remove a legenda
}
```

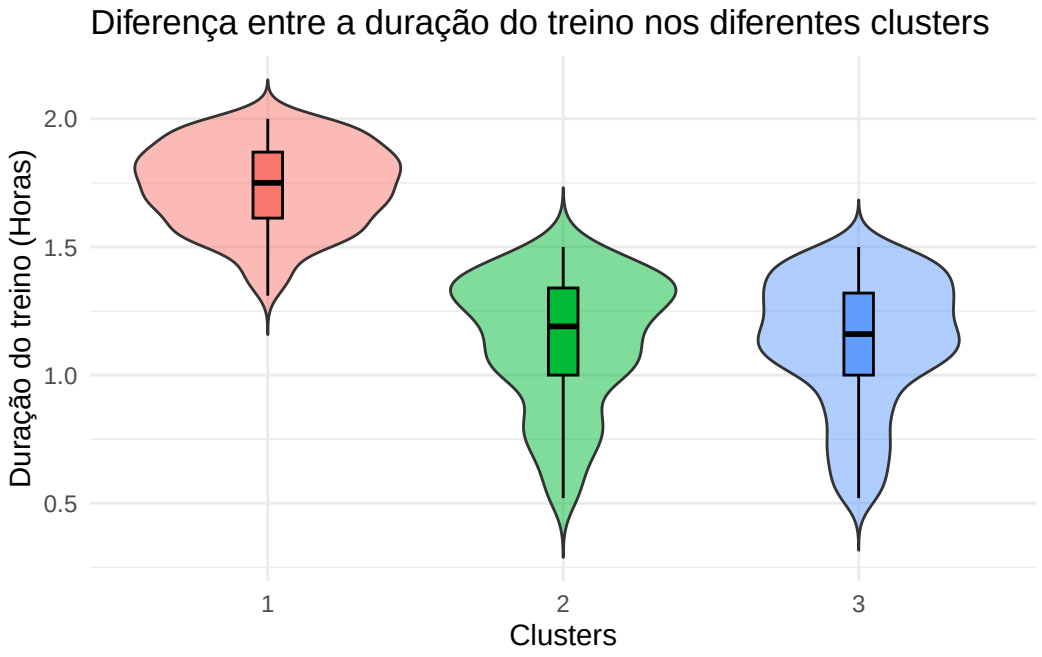
```r
violin_boxplot(df, df$Weight..kg., df$kmeanscluster, "Diferença entre a massa nos diferentes
```



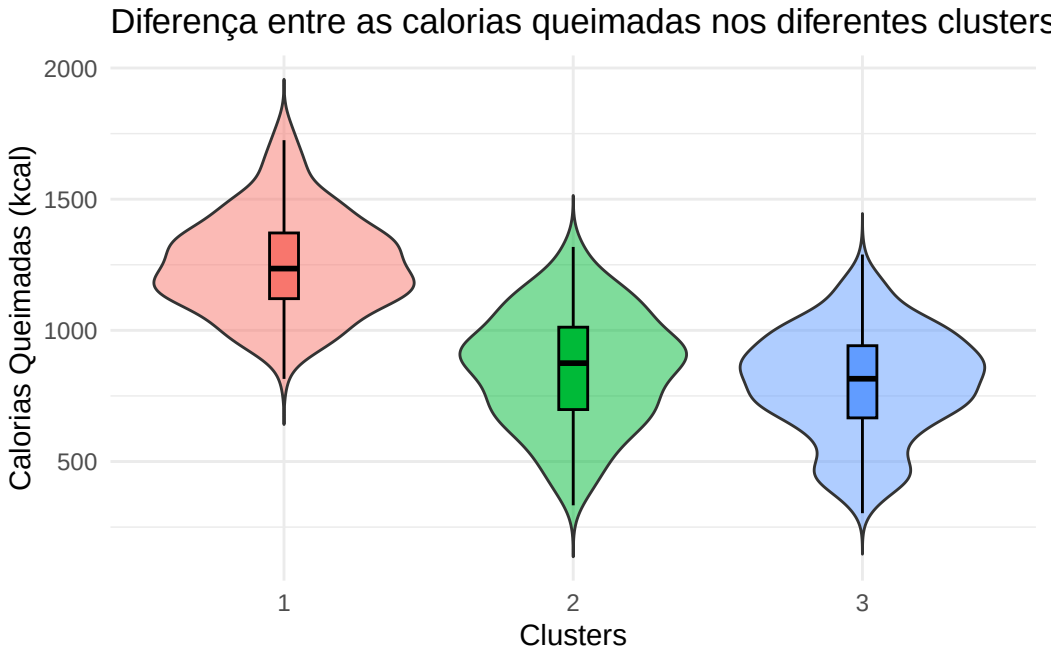

```r
violin_boxplot(df, df$Height..m., df$kmeanscluster, "Diferença entre a altura nos diferentes
```



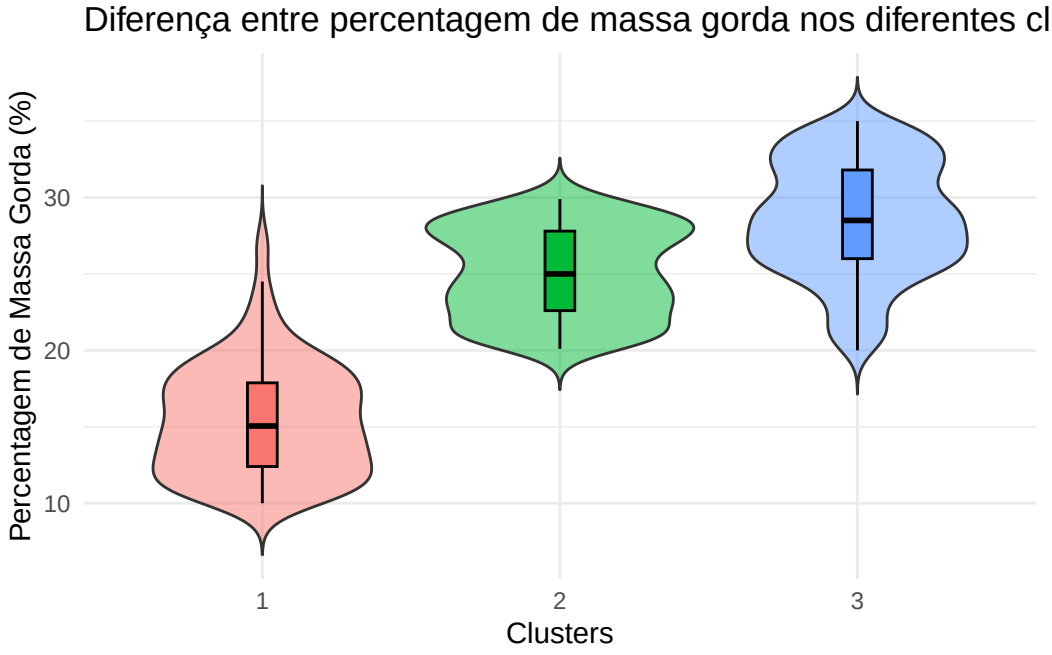

```
violin_boxplot(df, df$Session_Duration..hours., df$kmeanscluster, "Diferença entre a duração
```



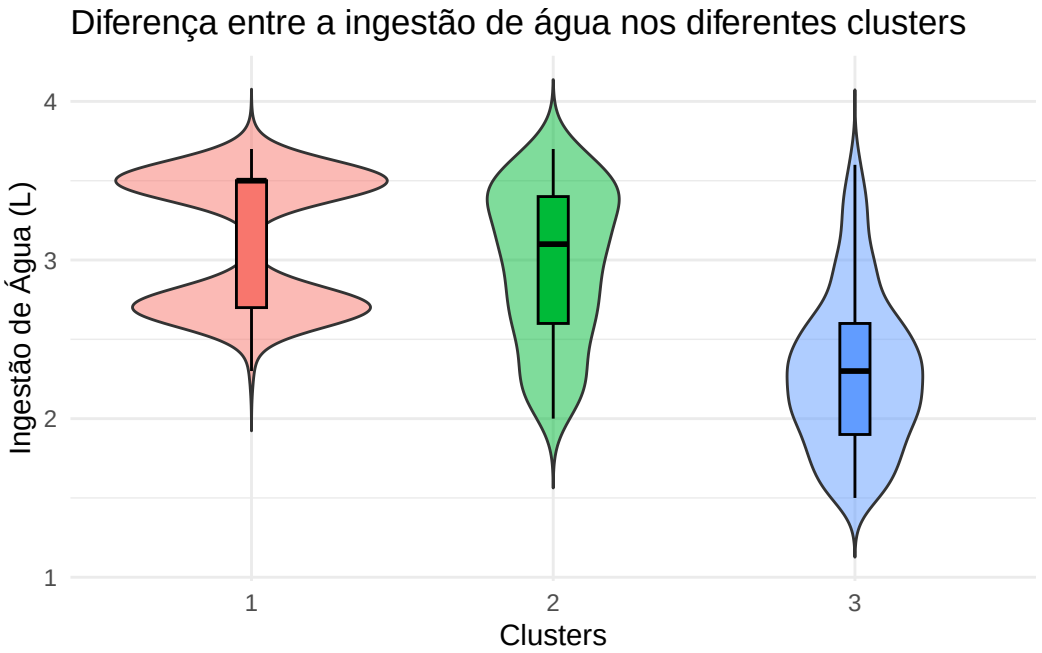
```
violin_boxplot(df, df$Calories_Burned, df$kmeanscluster, "Diferença entre as calorias queimad
```



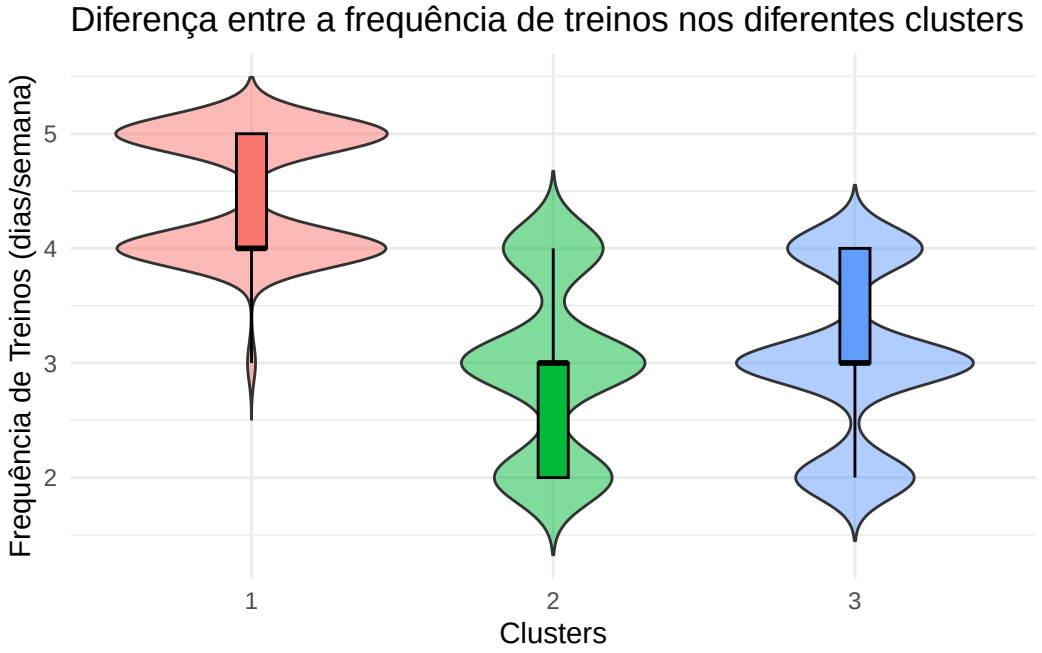

```
violin_boxplot(df, df$Fat_Percentage, df$kmeanscluster, "Diferença entre percentagem de massa
```



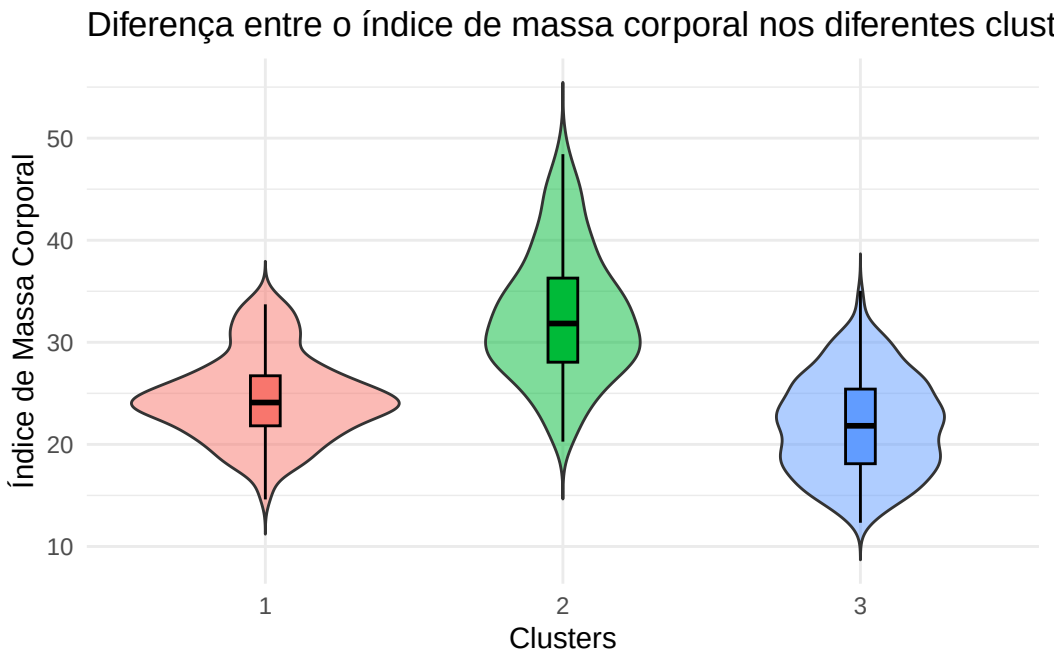
```
violin_boxplot(df, df$Water_Intake..liters., df$kmeanscluster, "Diferença entre a ingestão de
```




```
violin_boxplot(df, df$Workout_Frequency..days.week., df$kmeanscluster, "Diferença entre a fre
```

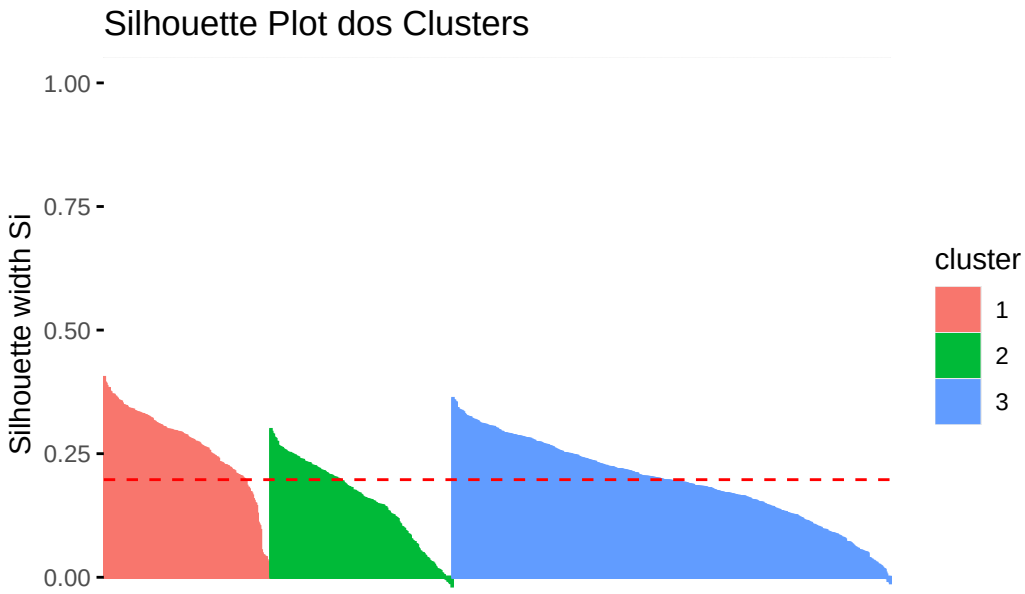


```r
violin_boxplot(df, df$BMI, df$kmeanscluster, "Diferença entre o índice de massa corporal nos
```




```r
sil <- silhouette(kmeans_clusters$cluster, dist(df_scaled))
fviz_silhouette(sil) +
  labs(title = "Silhouette Plot dos Clusters")
```

```
  cluster size ave.sil.width
1       1  206          0.26
2       2  225          0.15
3       3  542          0.19
```

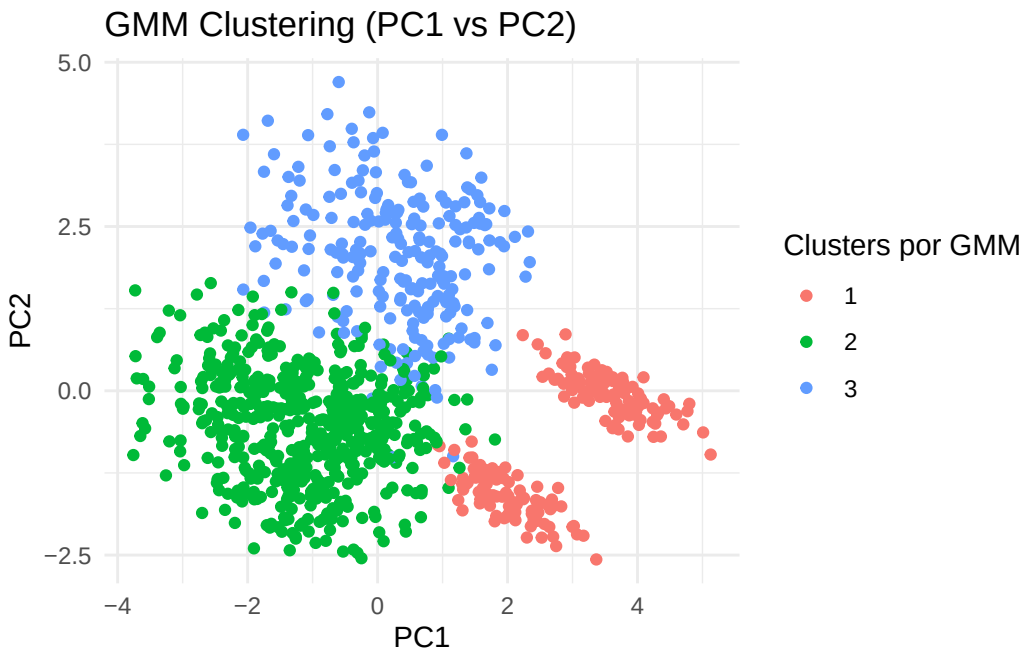


Gaussian Mixture Model

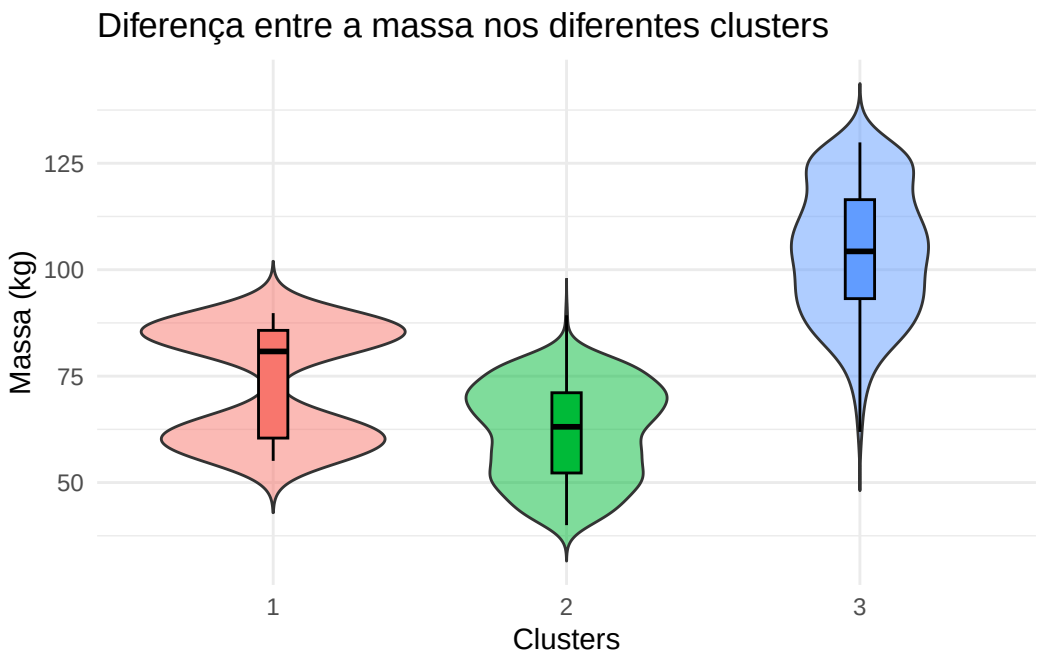
```r
gmm_clusters <- Mclust(df_scaled, G = 3)
summary(gmm_clusters)
```

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 3
components:

 log-likelihood   n  df       BIC       ICL
      -10198.98 973 272 -22269.43 -22301.62

Clustering table:
  1   2   3
191 559 223
```

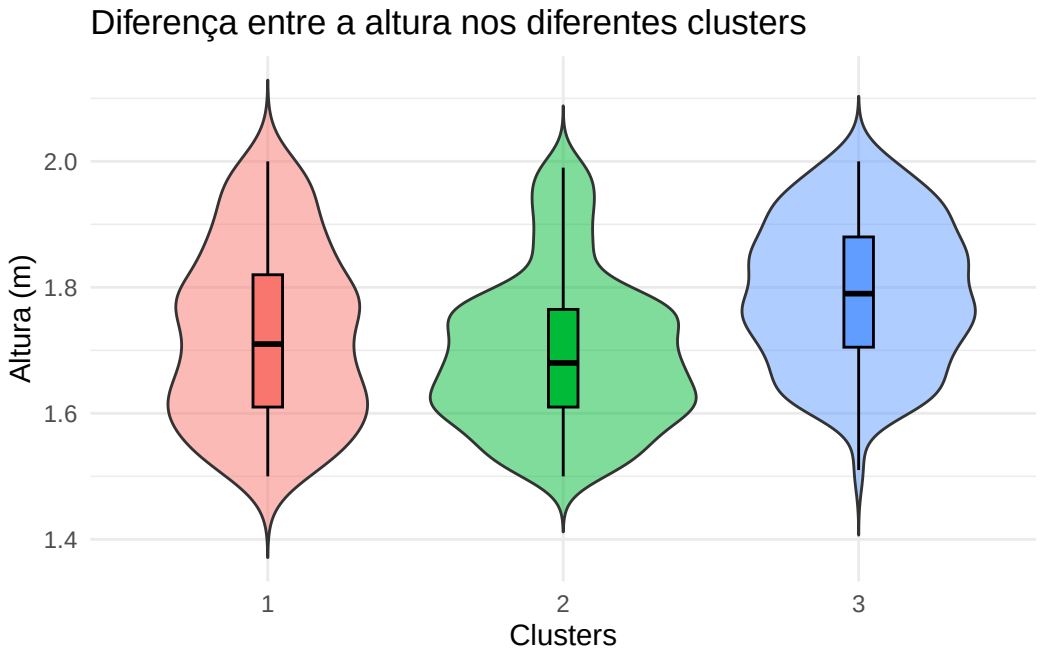
```r
df$gmmcluster <- as.factor(gmm_clusters$classification)
ggplot(pca_scores, aes(x = PC1, y = PC2, color = df$gmmcluster)) +
  geom_point() +
  labs(title = "GMM Clustering (PC1 vs PC2)", color= "Clusters por GMM") +
  theme_minimal()
```



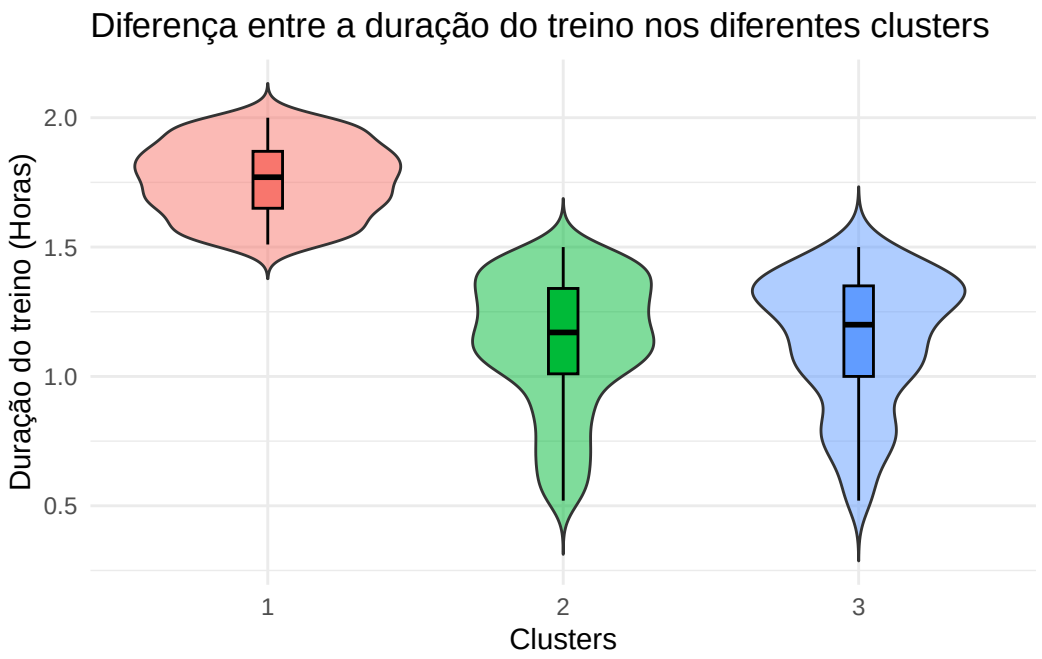

```r
violin_boxplot(df, df$Weight..kg., df$gmmcluster, "Diferença entre a massa nos diferentes clu
```



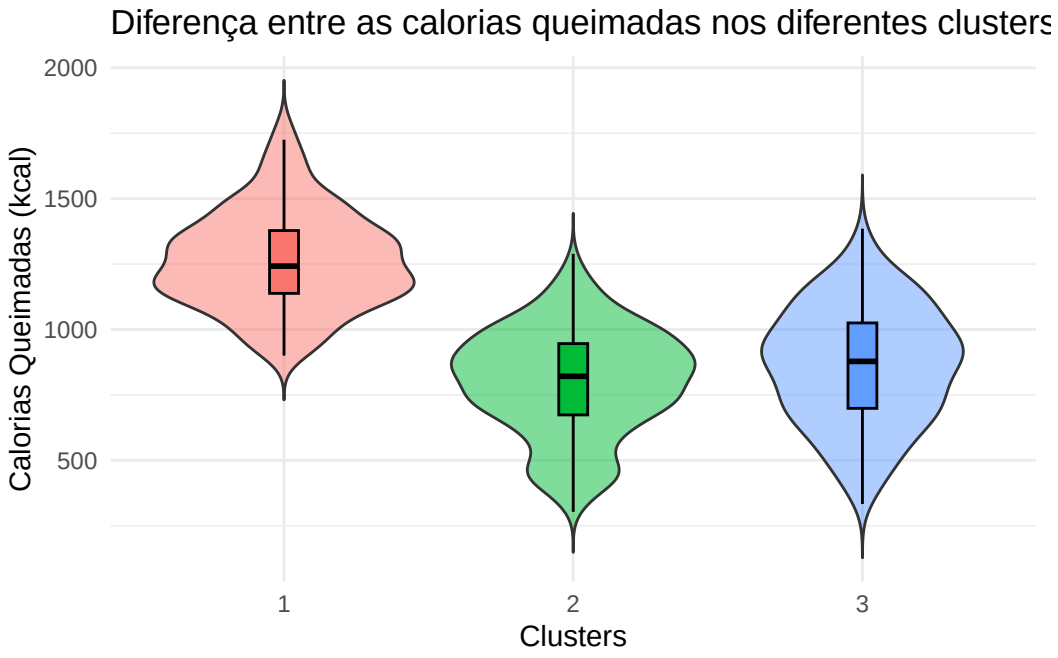

```
violin_boxplot(df, df$Height..m., df$gmmcluster, "Diferença entre a altura nos diferentes clu
```



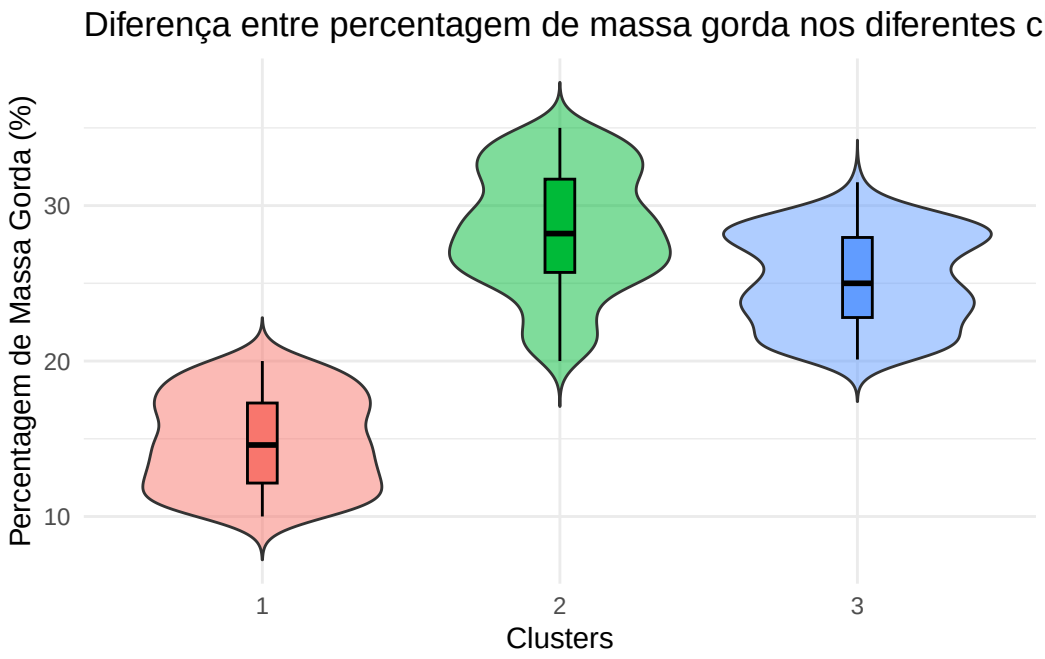
```
violin_boxplot(df, df$Session_Duration..hours., df$gmmcluster, "Diferença entre a duração do
```



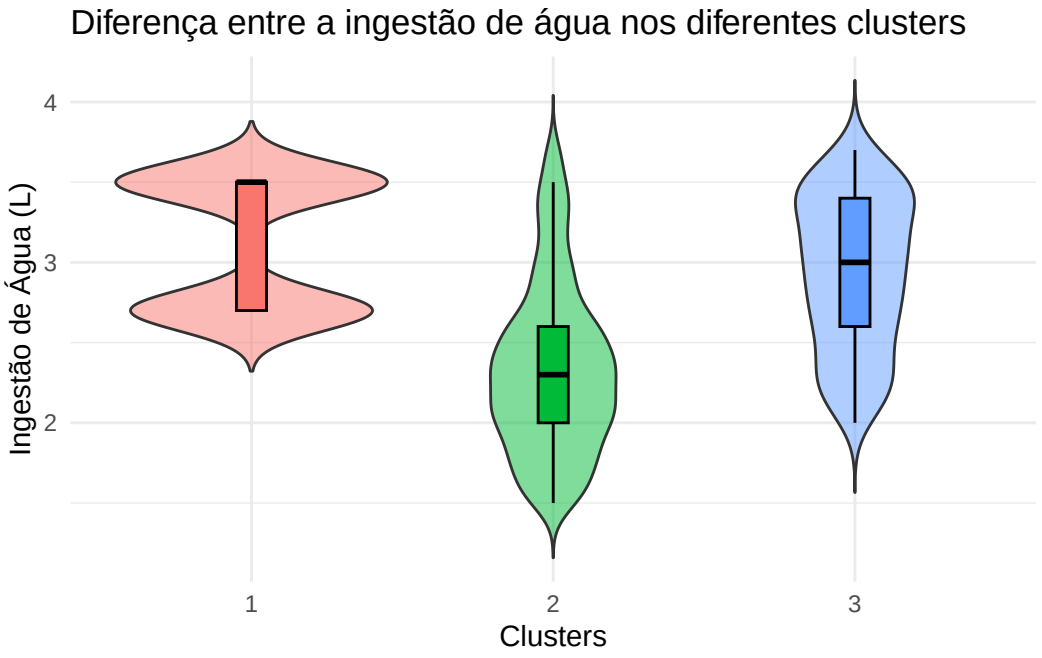

```
violin_boxplot(df, df$Calories_Burned, df$gmmcluster, "Diferença entre as calorias queimadas
```



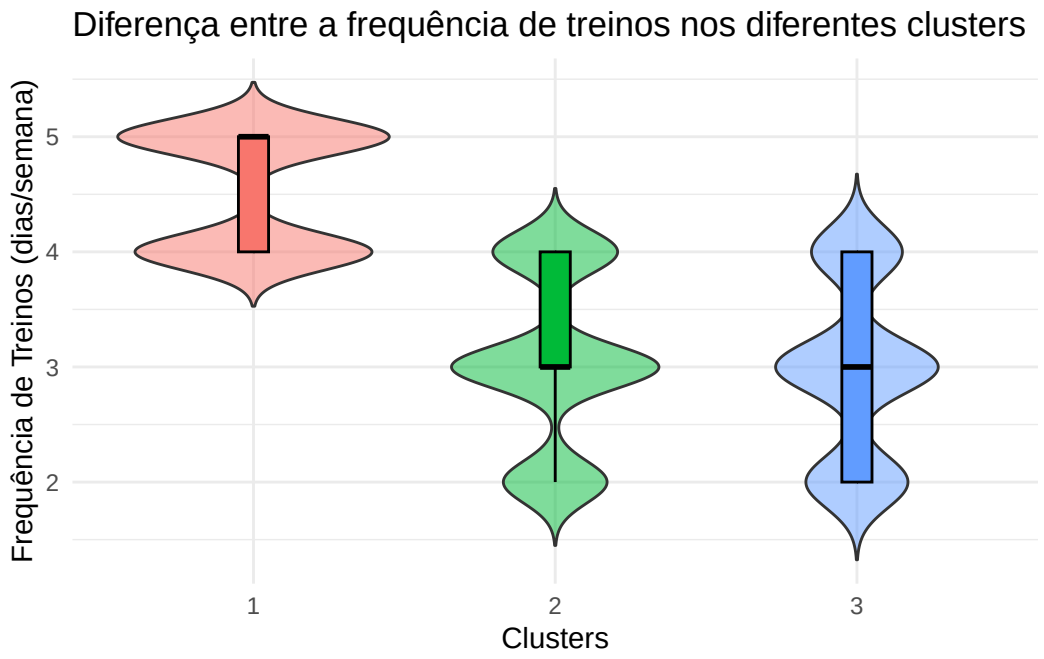
```
violin_boxplot(df, df$Fat_Percentage, df$gmmcluster, "Diferença entre percentagem de massa g
```




```
violin_boxplot(df, df$Water_Intake..liters., df$gmmcluster, "Diferença entre a ingestão de á
```



```
violin_boxplot(df, df$Workout_Frequency..days.week., df$gmmcluster, "Diferença entre a frequê
```




```
violin_boxplot(df, df$BMI, df$gmmcluster, "Diferença entre o índice de massa corporal nos di
```