Dicas para Resolução da Atividade Final

Preparação Inicial

Antes de começar as análises específicas, verifique se:

1. Você instalou e carregou todos os pacotes necessários
2. Os dados foram carregados corretamente
3. Transformou as variáveis categóricas em fatores
4. Tratou adequadamente os valores ausentes

```
# Verificando dados ausentes
sum(is.na(dados_heart))

# Removendo linhas com valores ausentes
dados_heart_clean <- na.omit(dados_heart)

# Verificando a estrutura dos dados
glimpse(dados_heart_clean)
summary(dados_heart_clean)
```

Parte 1: Análise Discriminante Linear (LDA)

Dicas para Construção do Modelo LDA

1. **Seleção de variáveis**: Use variáveis numéricas relevantes baseando-se na matriz de correlação. Variáveis altamente correlacionadas entre si podem causar problemas no modelo.
2. **Construção do modelo**: Use a função `lda()` do pacote MASS com a fórmula relacionando a variável alvo com as preditoras selecionadas.

```
# Exemplo de seleção de variáveis relevantes
modelo_lda <- lda(target ~ age + trestbps + chol + thalach + oldpeak, data = dados_treino)
```

3. **Análise dos coeficientes**: Examine `modelo_lda$scaling` para ver quais variáveis têm maior peso na função discriminante. Quanto maior o valor absoluto, mais importante é a variável para a discriminação.
4. **Visualização da função discriminante**: Use `ldahist()` para visualizar a separação dos grupos.

```
# Visualizando a separação entre os grupos
ldahist(data = predict(modelo_lda, dados_treino)$x, g = dados_treino$target)
```

Dicas para Avaliação do Modelo LDA

1. **Faça predições**: Use `predict()` para aplicar o modelo ao conjunto de teste.

```
predicoes_lda <- predict(modelo_lda, dados_teste)
```

2. **Avalie o desempenho**: Crie uma matriz de confusão e calcule métricas como acurácia, sensibilidade e especificidade.

```
# Matriz de confusão simples
matriz_confusao <- table(Previstos = predicoes_lda$class, Reais = dados_teste$target)

# Matriz de confusão detalhada com métricas
confusionMatrix(predicoes_lda$class, dados_teste$target)
```

3. **Interpretação**: Ao interpretar os resultados, preste atenção em:
    - Quais variáveis têm maior peso na discriminação
    - Quão bem o modelo classifica os pacientes saudáveis (especificidade)
    - Quão bem o modelo classifica os pacientes doentes (sensibilidade)
    - Qual o erro total do modelo (1 - acurácia)

Parte 2: Análise de Cluster

Dicas para Determinação do Número de Clusters

1. **Padronização dos dados**: Sempre padronize os dados antes de aplicar algoritmos de cluster para evitar que variáveis com escalas maiores dominem o agrupamento.

```
dados_cluster_scaled <- scale(dados_cluster)
```

2. **Método do Cotovelo**: Busque o ponto onde a adição de mais clusters não reduz substancialmente a soma dos quadrados dentro dos grupos (WSS).

```
fviz_nbclust(dados_cluster_scaled, kmeans, method = "wss") +
  labs(title = "Método do Cotovelo")
```

3. **Método da Silhueta**: Busque o número de clusters que maximiza o coeficiente de silhueta médio.

```
fviz_nbclust(dados_cluster_scaled, kmeans, method = "silhouette") +
  labs(title = "Método da Silhueta")
```

Dicas para Aplicação do K-means

1. **Defina uma semente aleatória**: Para garantir reprodutibilidade dos resultados.

```
set.seed(123)
```

2. **Use múltiplos pontos de partida**: Configure `nstart > 1` para aumentar a chance de encontrar o agrupamento ótimo.

```
km <- kmeans(dados_cluster_scaled, centers = 3, nstart = 25)
```

3. **Visualize os clusters**: Use fviz_cluster() para visualizar os clusters em um espaço bidimensional.

```
fviz_cluster(list(data = dados_cluster_scaled, cluster = km$cluster),
             palette = c("#00AFBB", "#FC4E07", "#E7B800"),
             ellipse.type = "convex",
             repel = TRUE,
             ggtheme = theme_minimal())
```

Dicas para Caracterização dos Clusters

1. **Adicione a informação de cluster aos dados originais**:

```
dados_com_clusters <- dados_heart_clean %>%
  mutate(cluster = factor(km$cluster))
```

2. **Calcule estatísticas descritivas por cluster**: Use group_by() e summarise().

```
cluster_profile <- dados_com_clusters %>%
  group_by(cluster) %>%
  summarise(
    N = n(),
    Idade_Média = mean(age),
    Colesterol_Médio = mean(chol),
    # Adicione outras estatísticas relevantes
    Perc_Doença = mean(target == "Disease") * 100
  )
```

3. **Visualize as características de cada cluster**: Use gráficos de dispersão, boxplots ou barras para comparar os clusters.

```
# Exemplo: visualizando idade vs. frequência cardíaca máxima
ggplot(dados_com_clusters, aes(x = age, y = thalach, color = factor(cluster))) +
  geom_point() +
  labs(title = "Idade vs. FC Máxima por Cluster")
```

4. **Nomeie os clusters**: Com base em suas características distintivas (ex: "Jovens saudáveis", "Idosos de alto risco", etc.).

Parte 3: Análise Fatorial

Dicas para Verificação da Adequação dos Dados

1. **Verifique correlações**: As variáveis devem ter correlações substanciais entre si (> 0.3).

```
cor_matrix_fatorial <- cor(dados_fatorial)
corrplot(cor_matrix_fatorial)
```

2. **Teste KMO**: Valores acima de 0.6 são considerados adequados.

```
KMO(dados_fatorial)
```

3. **Teste de Esfericidade de Bartlett**: O p-valor deve ser < 0.05.

```
cortest.bartlett(cor(dados_fatorial), n=nrow(dados_fatorial))
```

Dicas para Determinação do Número de Fatores

1. **Critério de Kaiser**: Selecione fatores com autovalores > 1.

```
eigen_valores <- eigen(cor(dados_fatorial))$values
data.frame(
  Fator = 1:length(eigen_valores),
  Autovalor = eigen_valores
) %>%
  ggplot(aes(x = Fator, y = Autovalor)) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red")
```

Dicas para Aplicação da Análise Fatorial

1. **Escolha o método de extração**: Maximum Likelihood (ml) é comum em análises confirmatórias, enquanto Principal Axis Factoring (pa) é comum em análises exploratórias.
2. **Escolha o método de rotação**: Varimax (ortogonal) é mais fácil de interpretar, enquanto oblimin (oblíqua) permite correlações entre fatores.

```
modelo_fa <- fa(dados_fatorial, nfactors = 2, rotate = "varimax", fm = "ml")
```

3. **Interpretação das cargas fatoriais**: Considere significativas cargas acima de 0.3 em valor absoluto.

```
print(modelo_fa$loadings, cutoff=0.3)
```

4. **Visualização das cargas fatoriais**:

```
data.frame(
  Variável = rownames(modelo_fa$loadings),
  Fator1 = modelo_fa$loadings[,1],
  Fator2 = modelo_fa$loadings[,2]
) %>%
  ggplot(aes(x = Fator1, y = Fator2, label = Variável)) +
  geom_point() +
  geom_text_repel()
```

Dicas para Cálculo e Análise dos Escores Fatoriais

1. **Calcule os escores fatoriais**:

```
escores_fatoriais <- factor.scores(dados_fatorial, modelo_fa)$scores
```

2. **Adicione os escores aos dados originais**:

```
dados_com_fatores <- dados_heart_clean %>%
  cbind(escores_fatoriais)
```

3. **Analise a relação entre os fatores e o diagnóstico**:

```
# Visualização
ggplot(dados_com_fatores, aes(x = ML1, y = ML2, color = target)) +
  geom_point() +
  labs(title = "Escores Fatoriais por Diagnóstico")

# Comparação estatística
t.test(ML1 ~ target, data = dados_com_fatores)
```

Parte 4: Integração das Técnicas

Dicas para Combinação das Análises

1. **Crie um dataset integrado** com os resultados das três técnicas:

```
dados_integrados <- dados_heart_clean %>%
  # Adicionar clusters
  mutate(cluster = factor(km$cluster)) %>%
  # Adicionar escores fatoriais
  cbind(escores_fatoriais) %>%
  # Adicionar predições da LDA
  mutate(lda_pred = predict(modelo_lda, dados_heart_clean)$class)
```

2. **Explore relações entre as técnicas**:

```
# Clusters x Fatores
ggplot(dados_integrados, aes(x = ML1, y = ML2, color = cluster)) +
  geom_point() +
  labs(title = "Clusters no Espaço dos Fatores")

# Clusters x Predição LDA
table(dados_integrados$cluster, dados_integrados$lda_pred)
```

3. **Crie visualizações integradas**:

```
ggplot(dados_integrados, aes(x = ML1, y = ML2,
                             color = cluster, shape = target)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Integração de Clusters, Fatores e Diagnóstico")
```

Dicas para o Relatório Final

Para escrever um relatório coeso e completo:

1. **Introdução**: Contextualize o problema de pesquisa e a importância das técnicas multivariadas para a análise de dados cardiovasculares.
2. **Metodologia**: Descreva brevemente as técnicas utilizadas e os passos seguidos para a análise.
3. **Resultados**: Para cada técnica, apresente:
    - Os principais achados
    - As interpretações clínicas
    - Visualizações relevantes
4. **Integração**: Explique como os resultados das três técnicas se complementam, criando uma visão mais abrangente do perfil de risco cardiovascular.
5. **Implicações**: Discuta como os resultados podem ser aplicados na prática clínica e na gestão em saúde.
6. **Limitações e Pesquisas Futuras**: Reconheça as limitações da análise e sugira caminhos para futuras investigações.
7. **Conclusão**: Resuma os principais achados e responda diretamente à pergunta principal de pesquisa.

Dicas Gerais para o Processo de Análise

1. **Documentação**: Comente seu código de forma clara e explicativa.
2. **Iteração**: Não hesite em refazer análises com diferentes parâmetros ou variáveis.
3. **Interpretação clínica**: Sempre relacione os achados estatísticos com o contexto clínico da doença cardíaca.
4. **Visualização**: Crie visualizações claras e informativas que comuniquem efetivamente os padrões encontrados.
5. **Consistência**: Mantenha a consistência na interpretação dos resultados ao longo de todas as análises.

Boa atividade!