TRABALHO EM GRUPO (até 3 alunos)
Disciplina: Métodos Computacionais Aplicados.
Tema: *Case* de Otimização de Portfólio.

Valor máximo: 10,0 pontos.
Peso na disciplina: 60%.
Professor responsável: Rodrigo Togneri.
Versão: 2017.02

Matrícula	**Nome Completo**
130492/2017	Alberto Ribeiro Vallim
126704/2017	Alexandre Vasconcelos Lima
126693/2017	Luiz Hissashi Rocha

Com base nos conhecimentos de:

- Modelo de carteira eficiente de ativo financeiros (material de aula);
- *Case* específico de otimização de portfólio resolvido em aula com modelagem computacional em R e por algoritmos em *Simulated Annealing* (SelecaoPortifolio_c_SA_codRMT_3), *Genetic Algorithm* (SelecaoPortifolio_c_GA_codRMT_3) e *Particle Swarm Optimization* (SelecaoPortifolio_c_PSO_codRMT_3).

Estabelece-se a seguinte situação-problema:

Você é o principal responsável de *Analytics* de uma grande administradora de fundos de investimento. Você e sua equipe já desenvolveram um algoritmo de otimização de portfólio pelo método de Markowitz, e agora estão em fase de aplicação e ajustes à aplicação nos negócios da empresa.

Este modelo deverá ser aplicado em um fundo de ações, com as mesmas ações possíveis do problema visto em aula. A atualização do "melhor portfólio" será feita semanalmente para este fundo, e a carteira do fundo será exatamente aquela definida pelo modelo.

Problema 1: Questão de Parametrização para Otimização do Negócio (5,0 pontos).

A principal questão de negócio em aberto que você tem em mãos é:

Como definir qual o melhor período histórico (em número de semanas) a considerar para que o modelo tenha a melhor aderência à realidade?

Informações adicionais para a resolução da principal questão de negócio em aberto:

- O modelo funciona em base semanal (históricos em intervalos de semana e a previsão é para uma semana adiante). É possível avaliar a carteira escolhida ao

final de cada semana, confrontando o retorno esperado com o retorno real da carteira.

- Assim, uma abordagem corriqueira para avaliação de histórico é a simulação em dados históricos, com o confronto de valores esperados e reais.

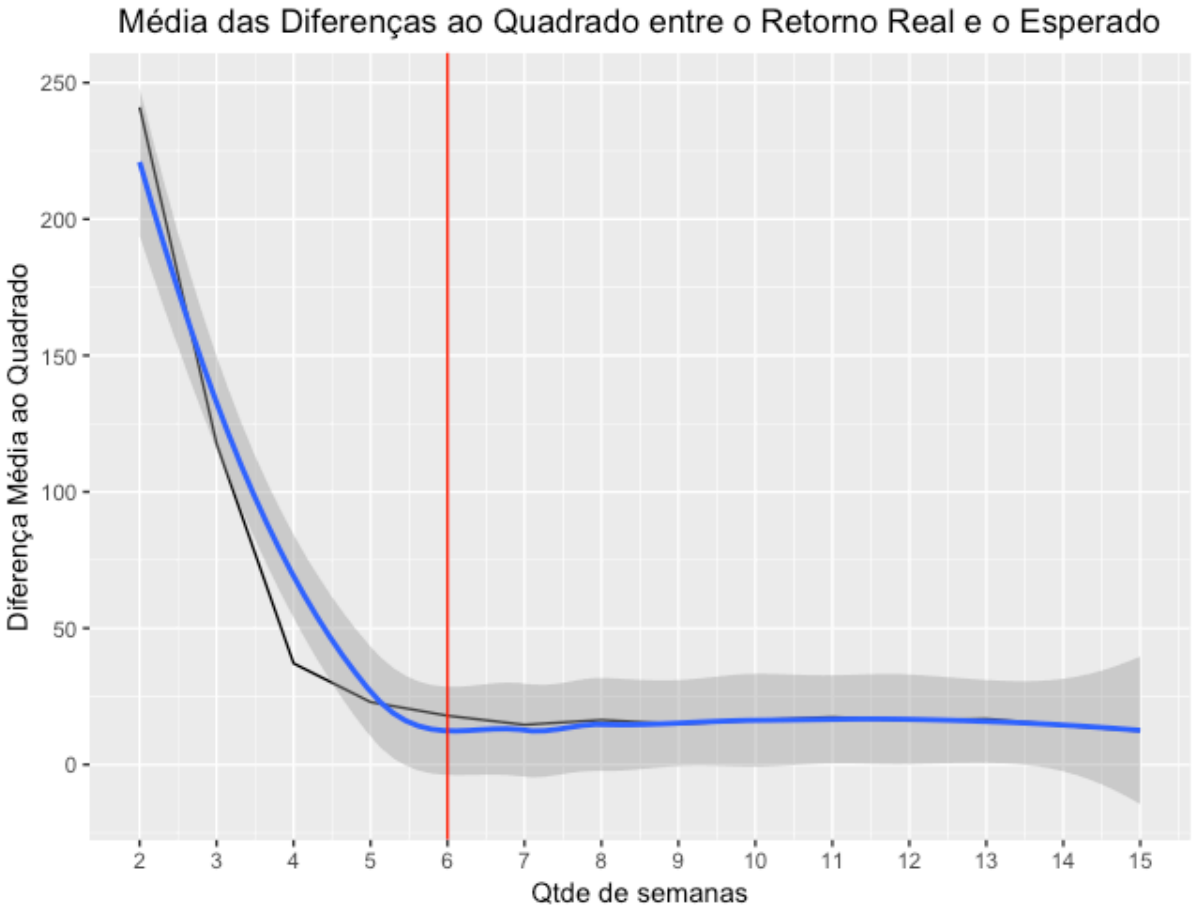
- O arquivo Ilustracao_estudo_hist propõe uma estrutura de safras de simulação para diversos tamanhos diferentes de histórico e o código SelecaoPortifolio_c_SA_HorHist_codRMT_1 executa a simulação para os cenários traçados. Os arquivos mencionados estão na seção 05 Trabalho em Grupo\01 Problema 1\01 Dados.

- O resultado desta simulação se encontra no arquivo estudo_hist_resultados, localizado na seção 05 Trabalho em Grupo\01 Problema 1\01 Dados.

Comece explicando sucintamente os prós e contras de um histórico muito pequeno e de um histórico muito grande.

Em seguida, com base nos resultados das simulações, estabeleça uma análise e faça a escolha do melhor tamanho de histórico para este modelo.

A escolha do tamanho da série temporal pode alterar o resultado de suas análises. Uma série muito pequena pode trazer menor confiabilidade nos resultados, pois os erros entre o valor real e o esperado tende a ser maior, pois haverá menor número de observações. Essa escolha, portanto, passa pela análise dos erros entre o valor observado e sua estimativa.
Já uma série muito longa pode fazer com que dados antigos tenham peso na sua estimativa atual. Ou seja, valores antigos influenciariam os valores esperados futuros. Uma forma de mitigar esse problema é aplicar técnica de séries temporais, em que valores mais recentes possuem maior peso. Outra opção é o modelo LSTM (Long Short Term Memory), aplicado em Deep Learning.
De qualquer forma, valores muito antigos, que possuem peso baixo nos modelos não precisariam ser carregados e assim economizaria recurso computacional.



Analisando o gráfico acima, verifica-se que a partir da sexta semana a diferença entre o retorno real e o esperado se mantém em nível semelhante. Enquanto que para períodos inferiores, observa-se uma queda acentuada dessa diferença. Dessa forma, o melhor tamanho de histórico com base nos dados apresentados é 6 semanas.

```
################
### SCRIPT EM R ###
################


###############
### LIBRARIES ###
###############

library(dplyr)
library(ggplot2)

##############
### DATASET ###
##############

df1 <- read.csv2("estudo_hist_resultados.csv", stringsAsFactors = FALSE)
glimpse(df1)

####################
### DATA MANIPULATION ###
```

```
####################

# Calculando a diferença entre retorno esperado e retorno real
df1$dif <- df1$retorno_esp_modelo - df1$retorno_real

# Elevando ao quadrado para eliminar os valores negativos
df1$dif2 <- df1$dif^2

#############
### ANALYSIS ###
#############

# Calculando a média das diferenças para cada tamanho de histórico
df1 %>%
  group_by(qt_sem_hist) %>%
  summarise(diferenca_media = mean(dif2)) %>%
  ggplot(aes(x=qt_sem_hist, y=diferenca_media)) +
  geom_line() +
  geom_smooth() +
  xlab('Qtde de semanas') +
  ylab('Diferença Média ao Quadrado') +
  ggtitle('Média das Diferenças ao Quadrado entre o Retorno Real e o Esperado') +
  scale_x_continuous(breaks = unique(df1$qt_sem_hist)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_vline(xintercept = 6, color = 'red')
```

Problema 2: Questão de Parametrização para Melhor Gestão de Recursos Computacionais.

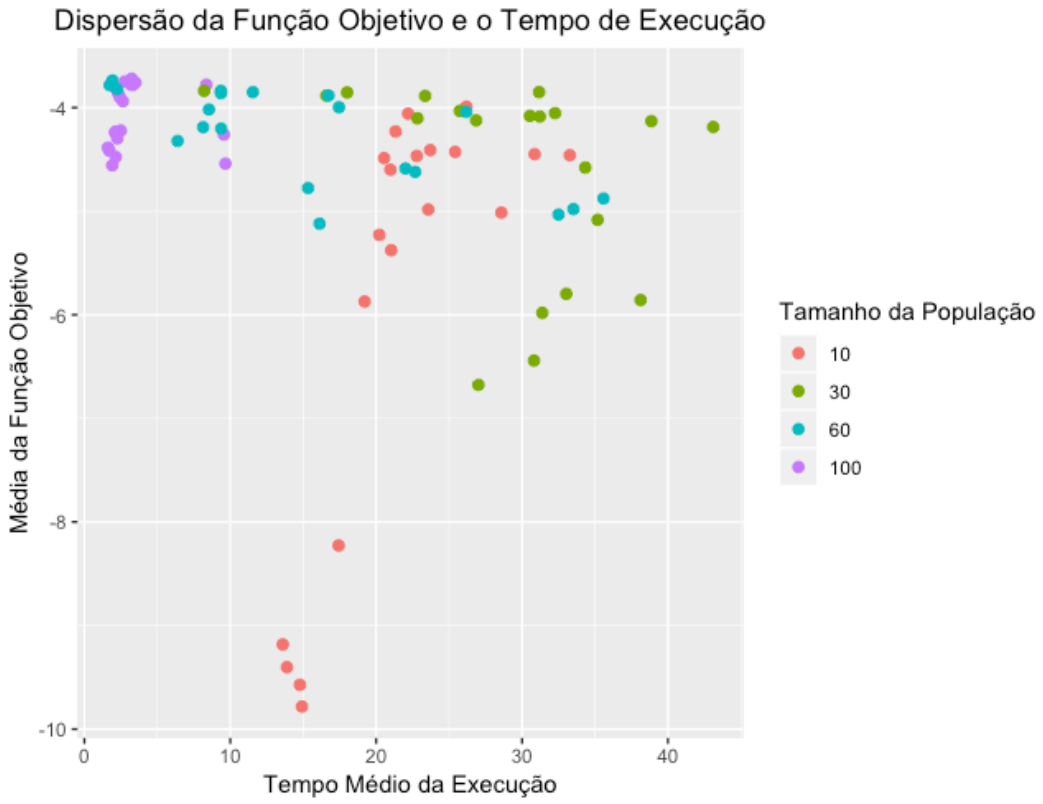
A principal questão técnica que você tem em mãos é:

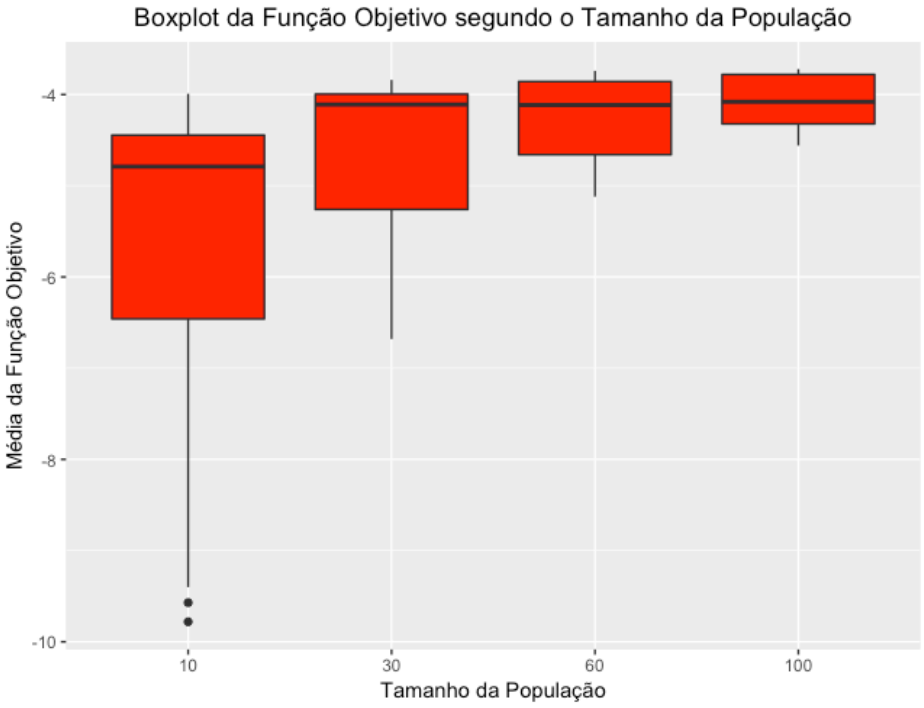
Como posso fazer para que o modelo escolhido rode sempre que necessário alcançando a melhor relação de compromisso entre os desempenhos de função objetivo e de tempo de processamento?

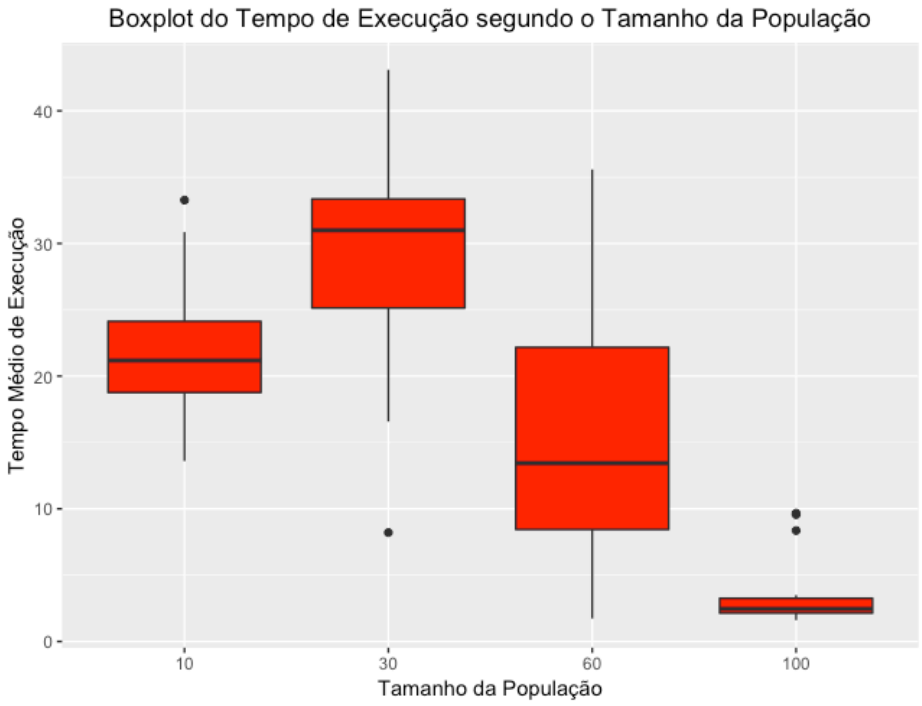
Sabe-se, do que se aprendeu em aula, que podemos tratar este problema por duas abordagens complementares: a) a busca pela melhor configuração de recursos computacionais; e b) a busca pelos melhores parâmetros de *tuning* do algoritmo utilizado. Para este trabalho, não trataremos da primeira abordagem citada, e sim apenas da segunda.

Desta forma, foi feita a simulação de cenários de parametros de *tuning* para a resolução do problema por Algoritmo Genético (pelo código no arquivo sim_GA_otim_par_codRMT_v1), sendo coletados os resultados analíticos (arquivo df_resultados_sim_GA) e consolidados (arquivo df_resultados_sim_GA_csdda). Os arquivos citados acima estão na seção 05 Trabalho em Grupo\02 Problema 2\01 Dados.

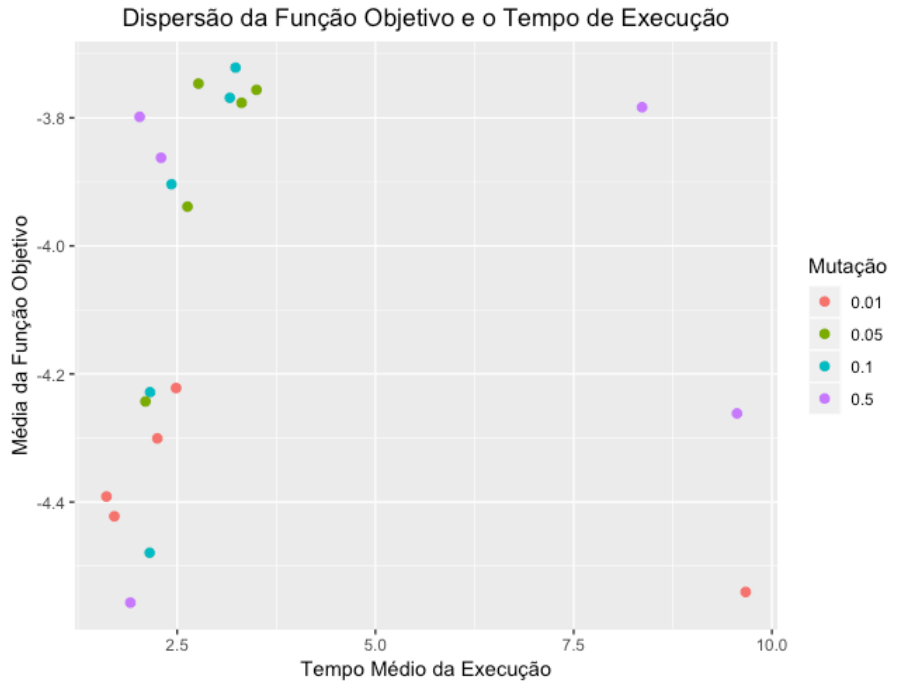
Utilize os resultados analíticos e (ou) consolidados para escolher bons parâmetros de *tuning* para o Algoritmo Genético aplicado ao nosso problema.

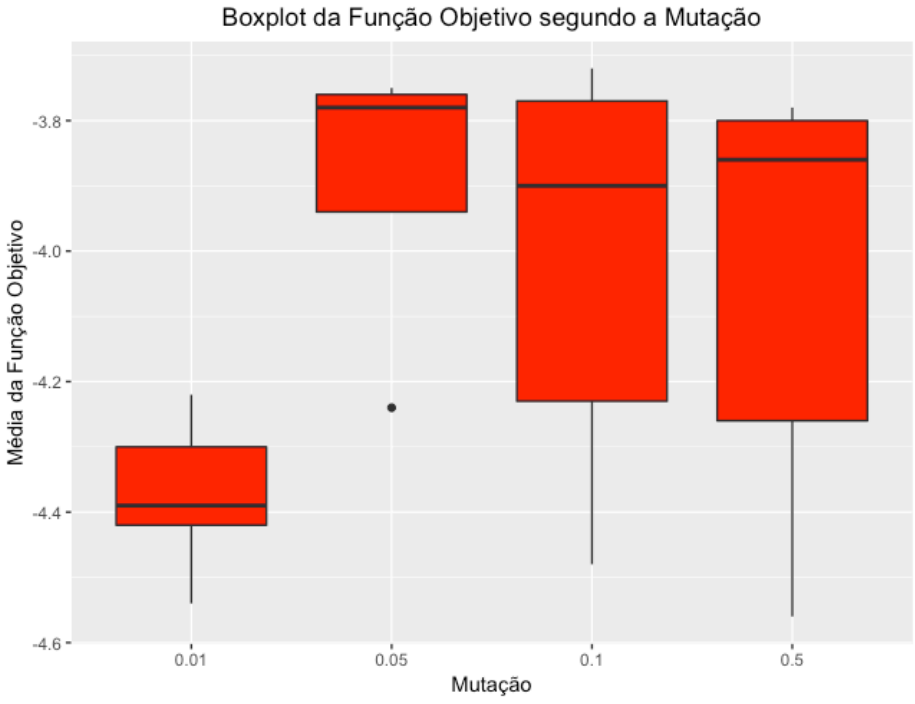


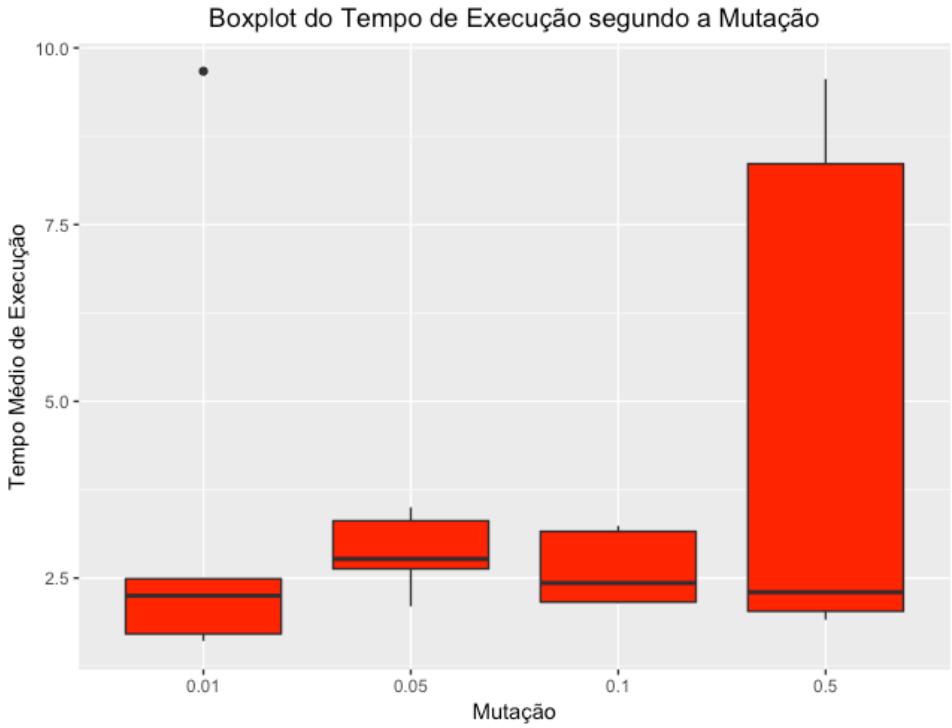




Nota-se que quando utilizamos o tamanho da população de 100 temos uma relação função objetivo e tempo médio de execução melhor do que os outros valores do parâmetro. Apesar da média da função objetivo ser semelhante entre os valores 30, 60 e 100, a dispersão (conforme o boxplot da função objetivo segundo a população) apresenta menor desvio interquartil, o que mostra menor dispersão dos dados. Quando verificamos o boxplot com o tempo de execução fica evidente que o tamanho da população apresentou tempo significativamente menor do que os outros valores para o parâmetro da população.

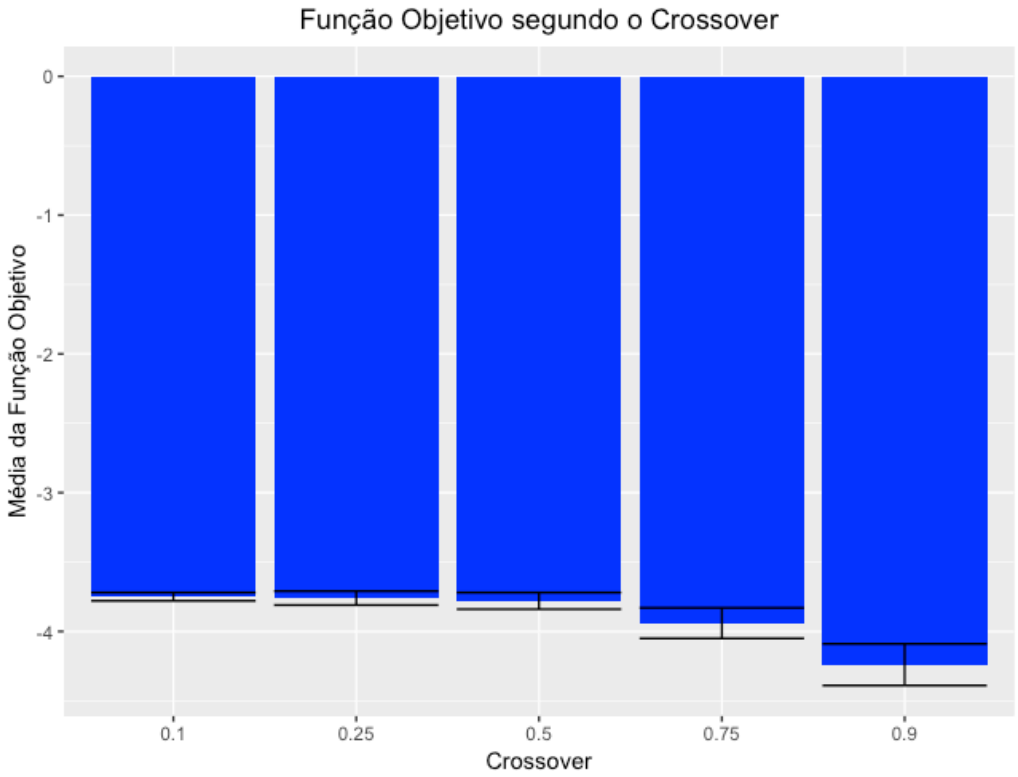

Dispersão da Função Objetivo e o Tempo de Execução
Média da Função Objetivo
-3.8
-4.0
-4.2
-4.4
2.5
5.0
7.5
10.0
Tempo Médio da Execução
Mutação
0.01
0.05
0.1
0.5

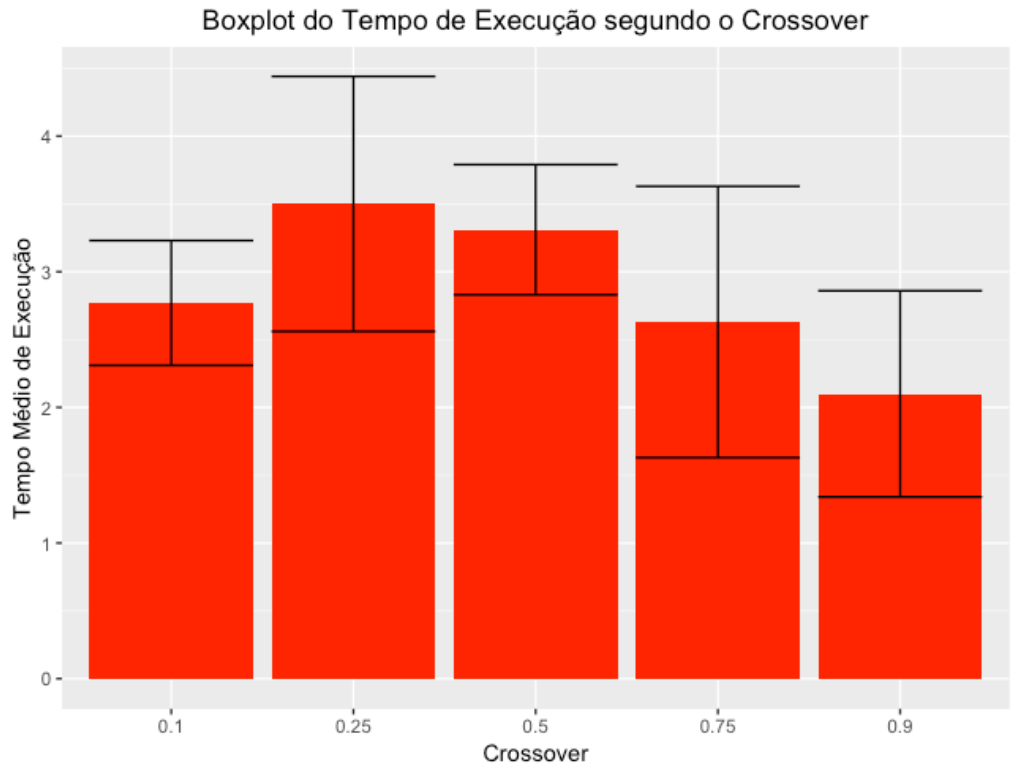

Boxplot da Função Objetivo segundo a Mutação
Média da Função Objetivo
-3.8
-4.0
-4.2
-4.4
-4.6
0.01
0.05
0.1
0.5
Mutação



Analisando o gráfico de Boxplot da Função Objetivo segundo a Mutação, percebe-se que o maior valor da função objetivo ocorre quando a probabilidade de mutação é igual a 0.05. Apesar de ter outlier na distribuição dos dados, este valor para o parâmetro também apresenta a menor dispersão para a média da função objetivo (menor intervalo interquartil).

Quando verificamos o tempo de execução, nota-se que, quando temos a probabilidade de mutação igual a 0.05, o tempo é um pouco mais elevado do que os outros valores para o parâmetro. Entretanto, essa diferença não é significativa e a dispersão dos valores de tempo para esse parâmetro é baixa.





Por fim, o parâmetro do crossover foi escolhido analisando os gráficos de tempo de execução e da função objetivo. A função objetivo é maior quando os valores do parâmetro são iguais a 0.1, 0.25 e 0.5. Analisando o tempo, vemos que o tempo de execução é menor

para o valor de 0.9, em seguida estão os valores do parâmetro de 0.75 e 0.1. Nesse sentido, temos que, dentre os valores que apresentam melhor desempenho na Função Objetivo, o menor tempo é para o valor de crossover de 0.1.

Portanto os valores de tuning para o problema são:

- Tamanho da População = 100
- Probabilidade de Mutação = 0.05
- Taxa de Crossover = 0.1

```
################
### SCRIPT EM R ###
################

#############
### LIBRARY ###
#############

library(dplyr)
library(ggplot2)

##############
### DATASET ###
##############

arquivos <- list.files()
arquivos
df2 <- read.csv2(arquivos[1], stringsAsFactors = FALSE)

glimpse(df2)

#####################
### DATA MANIPULATION ###
#####################

df2 <-
  apply(df2[,2:10], 2, as.numeric) %>%
  as.data.frame() %>%
  glimpse()

##############
### ANALYSIS ###
##############

df2 %>%
  ggplot(aes(x=tempo_execucao_media, y=funcao_objetivo_media, color = factor(popsize))) +
  geom_point(position = 'jitter', size = 2) +
  ggtitle('Gráfico de Dispersão da Função Objetivo e o Tempo de Execução') +
  xlab('Tempo Médio da Execução') +
  ylab('Média da Função Objetivo') +
  labs(color = 'Tamanho da População') +
  theme(plot.title = element_text(hjust = 0.5))

df2 %>%
  ggplot(aes(x=factor(popsize), y=funcao_objetivo_media)) +
  geom_boxplot(fill='red') +
  ggtitle('Boxplot da Função Objetivo segundo o Tamanho da População') +
  xlab('Tamanho da População') +
```

```
  ylab('Média da Função Objetivo') +
  theme(plot.title = element_text(hjust = 0.5))

df2 %>%
  ggplot(aes(x=factor(popsize), y=tempo_execucao_media)) +
  geom_boxplot(fill='red') +
  ggtitle('Boxplot do Tempo de Execução segundo o Tamanho da População') +
  xlab('Tamanho da População') +
  ylab('Tempo Médio de Execução') +
  theme(plot.title = element_text(hjust = 0.5))


df2 %>%
  filter(popsize == 100) %>%
  ggplot(aes(x=tempo_execucao_media, y=funcao_objetivo_media, color = factor(pmutation))) +
  geom_point(position = 'jitter', size = 2) +
  ggtitle('Gráfico de Dispersão da Função Objetivo e o Tempo de Execução') +
  xlab('Tempo Médio da Execução') +
  ylab('Média da Função Objetivo') +
  labs(color = 'Mutação') +
  theme(plot.title = element_text(hjust = 0.5))

df2 %>%
  filter(popsize == 100) %>%
  ggplot(aes(x=factor(pmutation), y=funcao_objetivo_media)) +
  geom_boxplot(fill='red') +
  ggtitle('Boxplot da Função Objetivo segundo a Mutação') +
  xlab('Mutação') +
  ylab('Média da Função Objetivo') +
  theme(plot.title = element_text(hjust = 0.5))

df2 %>%
  filter(popsize == 100) %>%
  ggplot(aes(x=factor(pmutation), y=tempo_execucao_media)) +
  geom_boxplot(fill='red') +
  ggtitle('Boxplot do Tempo de Execução segundo a Mutação') +
  xlab('Mutação') +
  ylab('Tempo Médio de Execução') +
  theme(plot.title = element_text(hjust = 0.5))


df2 %>%
  filter(popsize == 100 & pmutation == 0.05) %>%
  ggplot(aes(x=factor(pcrossover), y=funcao_objetivo_media)) +
  geom_col(fill='blue') +
  geom_errorbar(aes(ymin=funcao_objetivo_media - funcao_objetivo_desvpad,
                    ymax=funcao_objetivo_media + funcao_objetivo_desvpad)) +
  ggtitle('Função Objetivo segundo o Crossover') +
  xlab('Crossover') +
  ylab('Média da Função Objetivo') +
  theme(plot.title = element_text(hjust = 0.5))

df2 %>%
  filter(popsize == 100 & pmutation == 0.05) %>%
  ggplot(aes(x=factor(pcrossover), y=tempo_execucao_media)) +
  geom_col(fill='red') +
  geom_errorbar(aes(ymin=tempo_execucao_media - tempo_execucao_desvpad,
                    ymax=tempo_execucao_media + tempo_execucao_desvpad)) +
  ggtitle('Boxplot do Tempo de Execução segundo o Crossover') +
  xlab('Crossover') +
```

```
ylab('Tempo Médio de Execução') +
theme(plot.title = element_text(hjust = 0.5))
```