SME0806 - Estatística Computacional - Trabalho 3

04/07/2021

Alunos:

- Aline Fernanda da Conceição, 9437275
- Diego J. Talarico Ferreira, 3166561
- Matheus Victal Cerqueira, 10276661
- Murilo Henrique Soave, 10688813
- Nelson Calsolari Neto, 10277022

Docente: Professor Dr. Mário de Castro

Introdução

O presente documento se trata de uma solução para os exercícios propostos no Trabalho 3 da disciplina SME0806 - Estatística Computacional, oferecida pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo no primeiro semestre de 2021. As temáticas abordadas são métodos computacionais para a obtenção do nível descritivo de determinados testes de hipóteses e métodos MCMC para geração de amostras aleatórias.

Desenvolvimento e Metodologia

Exercício 1

No primeiro exercício, tem-se o objetivo de implementar um gerador de amostras aleatórias para o vetor aleatório $\boldsymbol{X} = (X_1, X_2)'$, cuja função densidade é dada por:

$$f(x_1, x_2) \propto x_2^{65} exp\left\{\frac{-x_2}{2}\sum_{i=1}^{130}(y_i - x_1)^2\right\} exp\left\{-\frac{(x_1 - 50)^2}{200}\right\} x_2^{-0,999} exp\left\{-0,001x_2\right\}$$

se $x_1 \in \mathbb{R}$ e $x_2 > 0$ e $f(x_1, x_2) = 0$ caso contrário. Os valores de y_i, $i = 1, 2, \cdots, 130$ são conhecidos e dados pelo exercício.

A partir de desenvolvimento matemático em $f(x_1, x_2)$, podemos mostrar que as distribuições condicionais completas de X_1 e X_2 são dadas por:

$$X_1|X_2 = x_2 \sim N\left(\mu = \frac{1/2 + x_2\sum_{i=1}^{130} y_i}{130x_2 + 1/100}, \sigma^2 = \frac{1}{130x_2 + 1/100}\right)$$

$$X_2|X_1 = x_1 \sim Gamma\left(k = 65,001; \theta = \left(0,001 + \frac{\sum_{i=1}^{130}(y_i - x_1)^2}{2}\right)^{-1}\right)$$

Observação: O desenvolvimento para a obtenção das relações acima, bem como o núcleo obtido para as distribuições $X_1|X_2 = x_2$ e $X_2|X_1 = x_1$ que permitiram tais conclusões, podem ser encontrados no apêndice desse trabalho.

Assim, como já são conhecidas as distribuições condicionais completas de X_1 e X_2, é possível obter amostras pseudoaleatórias de tais distribuições a partir do software R. Logo, optou-se pela implementação de um amostrador de Gibbs, o qual possui uma implementação mais simples que o algoritmo Metropolis-Hastings, além de não ser baseado em um método de aceitação-rejeição, o que permite que aproveitemos mais iterações.

Primeiramente, foram iplementadas funções para o cálculo dos parâmetros das distribuições condicionais completas. Para isso, foi utilizado os 130 valores conhecidos do vetor y apresentado no enunciado do exercício. Portanto, toda vez que a variável "y" for apresentada no código, estará fazendo menção a esse vetor de valores. Seguem abaixo as funções para os parâmetros:

```
# soma dos valores de y
s <- sum(y)

# Parâmetros de X1|X2=x2 ~ N(mu,sigma2)
mu <- function(x2){
  return( (1/2+x2*s)/(130*x2+1/100) ) # obtenção da média mu
}

sigma2 <- function(x2){
  return( 1/(130*x2+1/100) ) # obtenção da variância sigma2
}


# Parâmetros de X2|X1=x1 ~ Gamma(k,theta)
k = 65.001 # o parâmetro de forma k é um valor constante

theta <- function(x1){
  return( 1 / ( 0.001 + sum((y-x1)^2)/2 ) ) # obtenção do parâmetro de escala
}
```

Com as funções para os parâmetros em mãos, basta implementar uma função do amostrador de Gibbs, como segue abaixo:

```
# a função recebe o número de iterações desejada (iter), os valores iniciais
# para x1 e x2 (x1.inicial e x2.inicial, respectivamente) e a semente para
# possível reprodutibilidade
amostrador <- function(iter, x1.inicial, x2.inicial, semente){

  set.seed(semente)

  x1 <- x2 <-c() # x1 e x2 são vetores vazios aqui inicializados para armazenar
  # os valores obtidos nas amostras de x1 e x2 pelo método de Gibbs

  # atribuição dos valores iniciais
  x1[1] <- x1.inicial
  x2[1] <- x2.inicial

  # Laço de aplicação do método de Gibbs
  for(j in 1:iter) {
    # Utilização das distribuições condicionais completas obtidas anteriormente
    # e das funções para o cálculo dos parâmetros a cada iteração
```

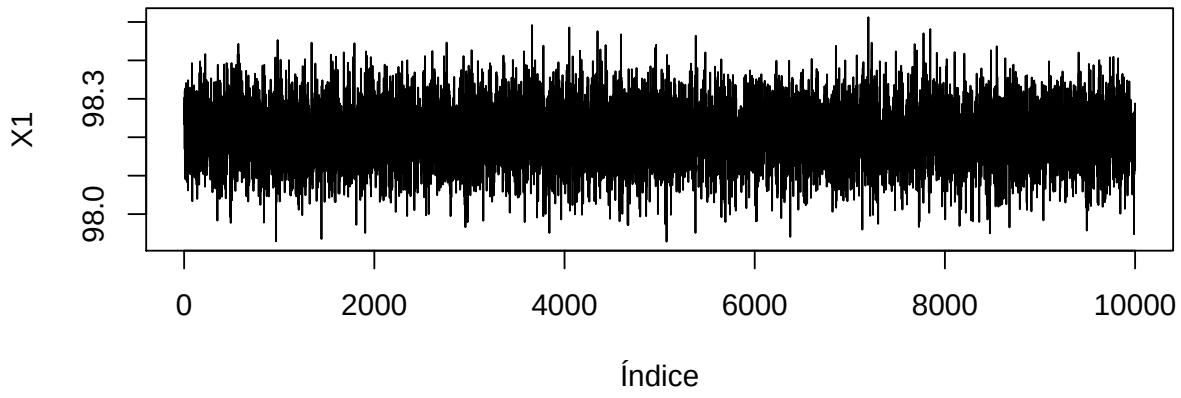
```
    x1[j+1] <-rnorm(1, mu(x2[j]), sqrt(sigma2(x2[j])) )
    x2[j+1] <-rgamma(1, shape = k, scale = theta(x1[j+1]) )

    }
  return(list(x1,x2)) # retorna uma lista contendo os valores dos dois vetores
}
```

Para efeito de sucintez, não foram incluídos neste relatório todos os testes realizados em relação à sensibilidade da convergência à escolha dos valores iniciais. Mas tais testes foram feitos para diferentes valores iniciais e ocorreu convergência em torno dos mesmos pontos na reta real para valores iniciais da cadeia consideravemente distintos, além de serem obtidas medidas resumo muito próximas entre os casos estudados. Assim, utilizando da função *amostrador* criada anteriormente para a obtenção de dois vetores (um de amostras de X_1 e outro de amostras de X_2) com 10.000 iterações e apresentando os resultados para as escolhas de valores iniciais 98 para x_1 e 1 para x_2, como segue abaixo:

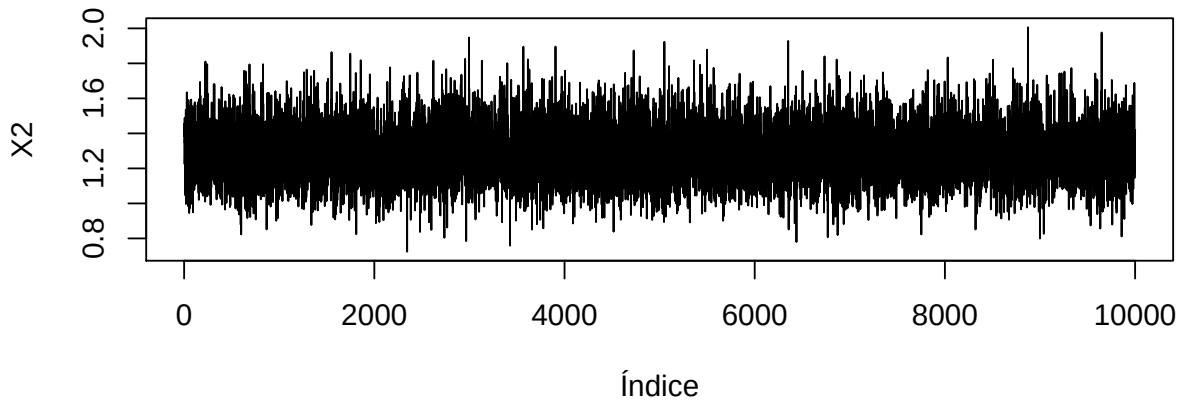
```
iter = 10000 # número de iterações

# obtenção da lista com os vetores de amostras através da função amostradora
# semente utilizada = 2112
amostras <- amostrador(iter = iter, x1.inicial = 98, x2.inicial = 1,
                       semente = 2112)

# Transformação da estrutura de dados que contém os valores obtidos de lista
# para vetor numérico.
ax1 <- unlist(amostras[1], recursive = TRUE, use.names = TRUE)
ax2 <- unlist(amostras[2], recursive = TRUE, use.names = TRUE)

# remoção do valor inicial da cadeia, o qual é um chute
ax1 <- ax1[-1]
ax2 <- ax2[-1]
```

Tendo os vetores de amostras geradas para X_1 (*ax1*)e X_2 (*ax2*), foi realizado um estudo de convergência e de autocorrelação das cadeias para assegurar que as amostras geradas, de fato, são amostras aleatórias de X_1 e X_2. Abaixo tem-se gráficos para a evolução das cadeias geradas:

```
# Gráficos de evolução da cadeia
par(mfrow =c(1, 1))
plot(ax1, xlab = "Índice", ylab = "X1", type = "l")
```




```
plot(ax2, xlab = "Índice", ylab = "X2", type = "l")
```

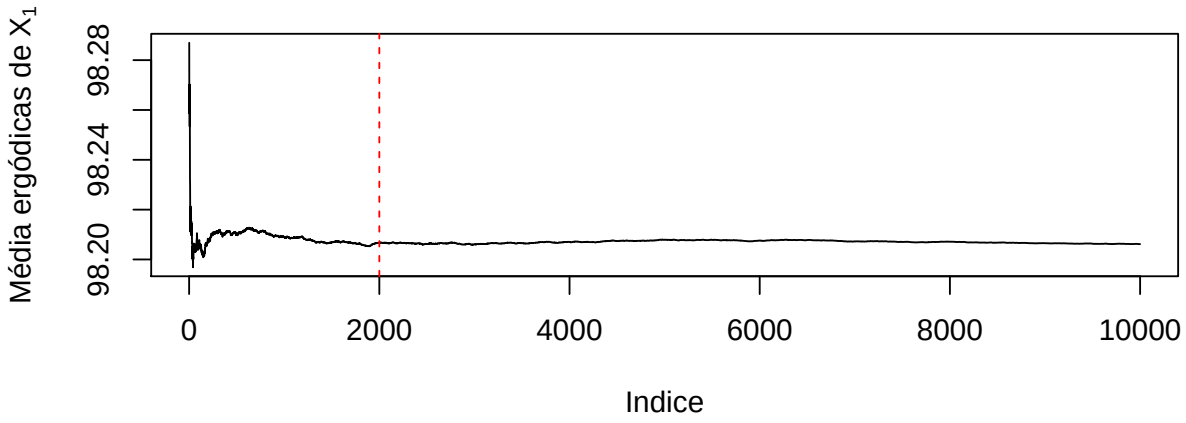


Dado os gráficos acima, parece não existir um padrão de evolução nas amostras geradas ao longo das iterações. Sendo um bom indicador em relação à propreidade de aleatoriedade da amostra que pretende-se atingir.

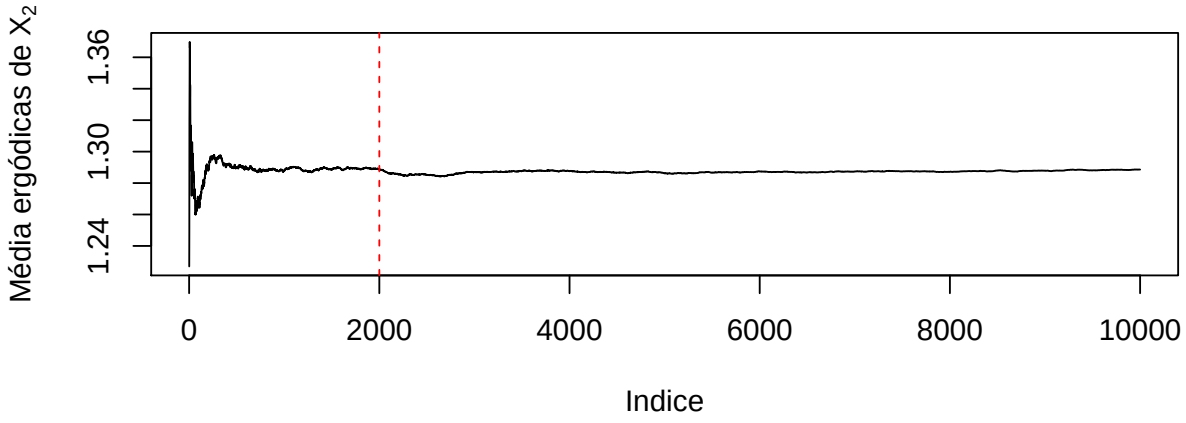
Prosseguindo com a análise, a partir dos gráficos da evolução das médias ergódicas será avaliado a convergência da cadeia gerada. A partir da Observação do comportamento dos gráficos, foi proposto um *burn-in* (queima) dos primeiros 2000 valores da cadeia original. Como foram geradas 10.000 observações para cada uma das duas variáveis de interesse, ainda permanece uma grande quantidade de valores para análise das propriedades estatísticas posteriormente. Abaixo, encontram-se os gráficos da evolução das médias ergódicas:

```
#burn-in proposto:
descarte = 2000
```

```r
# Gráficos para as médias ergódicas

# Obs.: o R é uma linguagem matricial de alto nível, o que permite a obtenção
# dso valores das médias ergódicas pela simples sintaxe cumsum(ax1)/(1:iter)

plot(cumsum(ax1)/(1:iter), type = "l", xlab = "Indice",
     ylab =expression(paste("Média ergódicas de ", X[1])))
abline(v = descarte, lty = 2, col = "red")
```



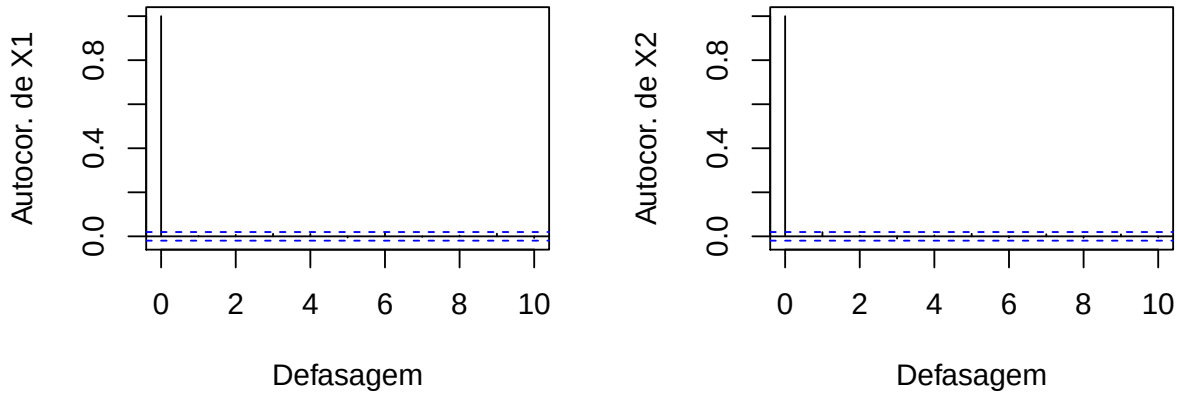

```r
plot(cumsum(ax2)/(1:iter), type = "l", xlab = "Indice",
     ylab =expression(paste("Média ergódicas de ", X[2])))
abline(v = descarte, lty = 2, col = "red")
```



Como discutido, as médias ergódicas parecem se estabilizar em torno de um ponto na reta real a partir do valor 2000 para o índice.

Por fim, foi verificado a autocorrelação da cadeia para desafagens de 1 a 10 a fim de averiguar se era necessáio um processo de *thinning*. A seguir os gráficos de autocorrelação:

```
par(mfrow =c(1, 2))
acf(ax1, lag.max = 10, xlab = "Defasagem", main = "",
    ylab = "Autocor. de X1")
acf(ax2, lag.max = 10, xlab = "Defasagem", main = "",
    ylab = "Autocor. de X2")
```



A partir dos gráficos, é possível concluir que a autocorrelação é muito pequena na cadeia gerada e dentro dos intervalos de confiança propostos pela função *acf* em torno de 0. Conclui-se portanto, que não é necessário desprender esforço computacional em um processo de *thinning*. Assim, será aplicado o *burn-in* proposto para a obtenção de um vetor com as amostras desejadas após confirmação dos tratamentos necessários pelas análises anteriores.

```
# Obtenção das amostras de X1 e X2 após o tratamento sugerido

# descarte proposto de 2000 e thinning não aplicado (step = 1)
indices <- seq(descarte+1, iter)

ax1.f <- ax1[indices]
ax2.f <- ax2[indices]
```

Depois da obtenção das amostras aleatórias de X_1 e X_2 a partir de um amostrador de Gibbs e de um tratamento baseado na análise de convergência e autocorrelação da cadeia, pode-se calcular estatísticas descritivas para analisar o comportameto de tais variáveis aleatórias.

Abaixo, segue algumas estatísticas descritivas para as amostras finais obtidas.

Observação: O resumo das medidas abaixo foram obtidas a partir das funções *summary* e *sd* do R. Esses foram aplicados às amostras finais contidas nas variáveis *ax1.f* e *ax2.f*. O código foi omitido para efeito de organização do relatório.

Estatísticas resumo de X_1:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   97.93   98.15   98.21   98.21   98.26   98.51
```

```
## Desvio-padrão: 0.07792675
```

Logo, conclui-se que X_1 possui um desvio-padrão pequeno, em torno de 0,077. Assim, os valores que ela assume se concentram em torno da média 98,210. Valor o qual também corresponde à sua mediana para a precisão analisada.

Estatísticas resumo de X_2:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7239  1.1768  1.2814  1.2886  1.3951  2.0065
```

```
## Desvio-padrão: 0.1615919
```

Já para X_2, o desvio-padrão é maior do que no caso de X_1, mas ainda é um valor pequeno, em torno de 0,162. Os valores que ela assume se encontram em torno da média 1,289. Essa média não corresponde ao valor da mediana com a mesma precisão do caso de X_1 mas ainda é próxima de seu valor de 1,281.

Exercício 2

Sejam as variáveis aleatórias T_1: tempos de queima do tipo de combustível 1 e T_2: tempos de queima do tipo de combustível 2 (em minutos). Abaixo, são apresentadas amostras observadas de T_1 e T_2:

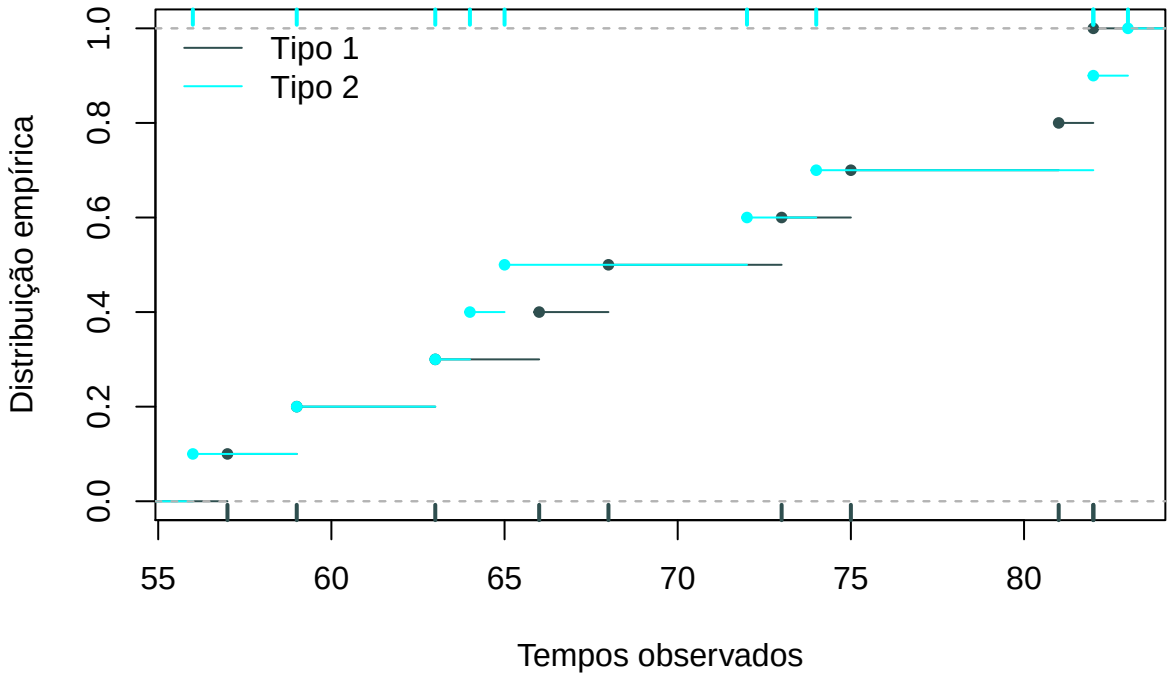
```r
tempos1 <- c(63, 82, 81, 68, 57, 59, 66, 75, 82, 73)

tempos2 <- c(64, 56, 72, 63, 83, 74, 59, 82, 65, 82)
```

O objetivo é realizar um teste de hipóteses que permita obter conclusões acerca da igualdade ou desigualdade das variâncias populacionais de T_1 e T_2 utilizando métodos puramente computacionais. Para tal, foi utilizada a teoria de testes de permutação para testar as seguintes hipóteses:

$$H_0 : F_1 = F_2 \quad \text{vs} \quad H_1 : F_1 \neq F_2$$

Sendo F_1 e F_2 as funções de distribuição de T_1 e T_2 respectivamente. Portanto, primeiramente será realizada uma análise qualitativa da diferença entre as médias de T_1 e T_2:

```r
# Gráficos de distribuição acumulada empírica
dados <- c(tempos1,tempos2)

plot(ecdf(tempos1), main = "", pch = 20, col = "darkslategrey",
     xlab = "Tempos observados",
     ylab = "Distribuição empírica",
     xlim = range(dados))
lines(ecdf(tempos2), pch = 20, col = "cyan")
rug(tempos1, col = "darkslategrey", lwd = 2)
rug(tempos2, col = "cyan", lwd = 2, side = 3)
legend("topleft",c("Tipo 1", "Tipo 2"), lty = 1,
       col =c("darkslategrey", "cyan"),
       bty = "n")
```




```
# Médias para os tempos observados
mean(tempos1)
```

```
## [1] 70.6
```

```
mean(tempos2)
```

```
## [1] 70
```

A partir dos resultados do gráfico das distribuições empíricas de T_1 e T_2 criadas através das amostras, pode-se dizer que as médias são consideravelmente próximas. Isso permite a utilização do nível descritivo obtido no teste de permutação com mais segurança para concluir sobre a igualdade das variâncias, já que o teste de permutação compara as funções de distribuição e não as variâncias diretamente.

Para realizar o teste de permutações, utilizou-se a estatística de teste a seguir:

$$F = \frac{S_1^2}{S_2^2} \quad (I)$$

sendo,

$$S_k^2 = \frac{\sum_{i=1}^{n_i}(t_{ki} - \bar{t_k})}{n_k - 1}$$

Onde n_k é o tamanho da amostra obtida de T_k, $t_{ki}, i = 1, \cdots, n_k$ são as observações dos tempos de T_k e $\bar{t_k}$ é a média amostral de tais tempos, para $k \in \{1, 2\}$. Assim, foram obtidas as informações necessárias para realizar um teste de permutações utilizando-se a estatística de teste (I).

```
# Teste de permutação:

## Estatística de teste: s2_1/s2_2;
## Aplicar um teste de permutação (não há suposições sobre a distribuição
## original);
## Muitas permutações são possíveis: abordagem por método de Monte Carlo.

# tamanhos das amostras
n1 <- length(tempos1)
n2 <- length(tempos2)

# soma das amostras para realizar a permutação
n <- n1 + n2

dados <- c(tempos1,tempos2)

choose(n, n1) # permutações possíveis
```

```
## [1] 184756
```

```
F_obs <- var(tempos1)/var(tempos2) # estatística de teste observada
F_obs
```

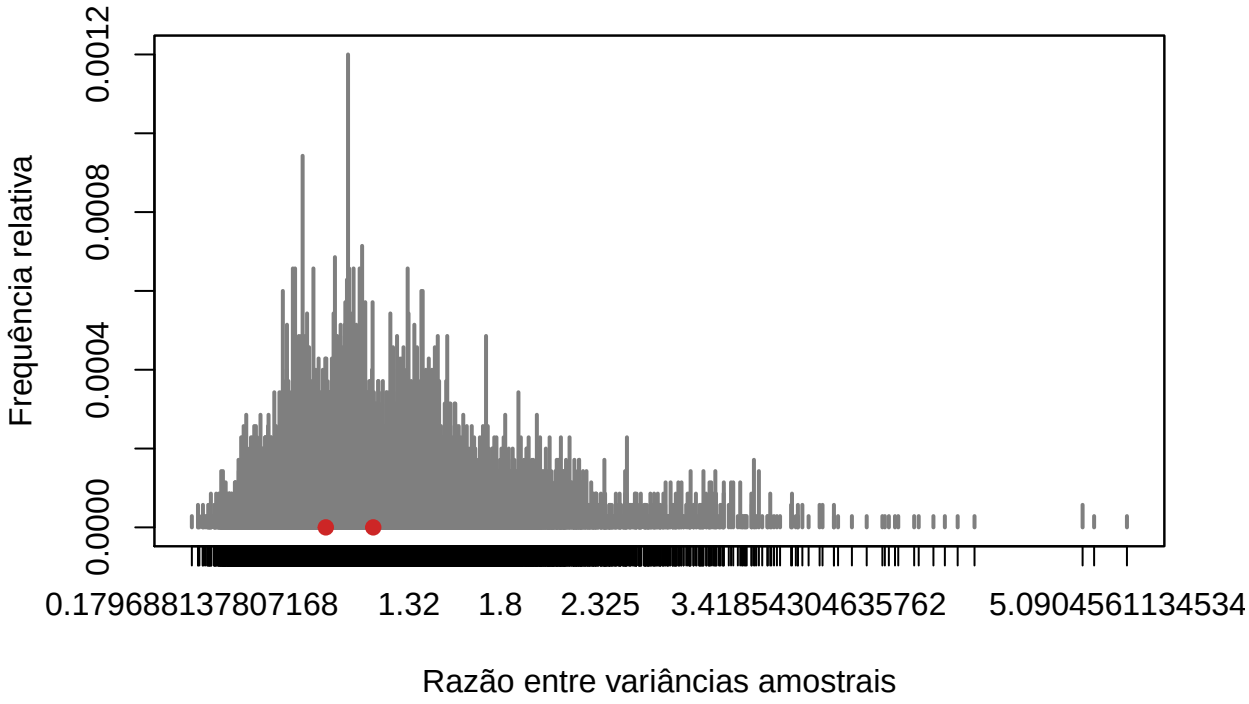
```
## [1] 0.8831858
```

Logo, são possíveis 184.756 permutações diferentes com as amostras fornecidas. Para poupar esforço computacional, foram obtidos resultados por simulações de Monte Carlo com 35.000 permutações, obtendo portanto, uma aproximação para o p-valor e não o seu valor condicional exato. A estatística observada $F_{obs} = 0,883$ será utilizada para a obtenção do p-valor. Como o teste realizado é bilateral, é necessário dois pontos na reta os quais serão utilizados como indicativo de observações extremas. São eles: F_{obs} e $1/F_{obs}$, fundamentais para realizar o cáculo do p-valor. Desse modo, serão observados os valores obtidos para a estatística (I) para as diferentes permutações obtidas pelo método de Monte Carlo.

```
R <- 35000 # Número de simulações
F. <-c() # vetor para armazenar os valores da estatística (I)

set.seed(2112)

for(j in 1:R) {
  dadosl <- sample(dados) # permutação
  t1l <- dadosl[1:n1] # atribui os n1 primeiros como observações de 1
  t2l <- dadosl[(n1+1):n] # atribui o restante como observações de 2
  F.[j] <- var(t1l)/var(t2l) # obtém a estatística para a permutação
}
```

Segue o comportamento da estatística de teste nas permutações:

```r
# gráfico das frequências relativas dos valores assumidos por (I)
plot(table(F.)/R, col = "gray50",
     xlab = "Razão entre variâncias amostrais",
     ylab = "Frequência relativa")
# pontos indicativos de observações extremas (estatística observada e seu
# inverso)
points(F_obs, 0, pch = 19, col = "firebrick3")
points(1/F_obs, 0, pch = 19, col = "firebrick3")
box()
```



A partir do gráfico acima e a localização dos pontos da estatística observada para definir observações extremas, tem-se suspeitas fortes de que o nível descritivo para o teste será alto. Obteve-se assim, o p-valor com a teoria para testes de permutações.

Observação: Sabe-se que para este caso $F_{obs} = 0,883 < 1/F_{obs}$. Assim, as observações extremas estarão à direita de $1/F_{obs}$ e à esquerda de F_{obs}.

```r
# observações extremas à esquerda de F_obs
ke <- sum(F. <= F_obs)
# observações extremas à direita de 1/F_obs
kd <- sum(F. >= 1/F_obs)

# p-valores bilaterais
p1 <- (ke+1)/(R+1)
p2 <- (kd+1)/(R+1)
```

```
cat("\n nível descritivo para teste unilateral à esquerda aprox. =", (ke+1)/(R+1), "\n")
```

```
##
##  nível descritivo para teste unilateral à esquerda aprox. = 0.3681323
```

```
cat("\n nível descritivo para teste unilateral à direita aprox. =", (kd+1)/(R+1), "\n")
```

```
##
##  nível descritivo para teste unilateral à direita aprox. = 0.3706466
```

```
#nível descritivo é igual a duas vezes o menor p-valor dos testes bilaterais
p <- 2*min(p1,p2)

cat("\n nível descritivo aprox. para o teste de permutações bilateral=", p, "\n")
```

```
##
##  nível descritivo aprox. para o teste de permutações bilateral= 0.7362647
```

A partir dos valores encontrados, o nível descritivo obtido pelo teste de permutação bilateral foi de $\alpha = 0,736$. Portanto, há indícios de que não rejeitamos a hipótese nula do teste de permutação, ou seja, que as distribuições F_1 e F_2 são iguais. Este resultado juntamente com a análise descritiva sobre a média, nos permite concluir que as variâncias dos dois grupos podem ser consideradas iguais. Dessa forma, podemos dizer que o tempo de queima dos 2 tipos de combustíveis são estatísticamente iguais.

Apêndice

Aqui apresenta-se o desenvolvimento matemático por trás da obtenção das distribuições condicionais completas no exercício 1.

$$f(x_1, x_2) \propto x_2^{65} exp\left\{\frac{-x_2}{2}\sum_{i=1}^{130}(y_i - x_1)^2\right\} exp\left\{-\frac{(x_1-50)^2}{200}\right\} x_2^{-0,999} exp\left\{-0,001x_2\right\}$$

Resolvendo para cada uma das expressões temos:

$$f_1(x_1) \propto exp\left\{-\frac{x_2}{2}\sum_{i=1}^{130}(y_i - x_1)^2\right\} exp\left\{-\frac{(x_1-50)^2}{200}\right\} =$$

$$= exp\left\{-\frac{x_2}{2}\cdot\sum_{i=1}^{130}(y_i^2 - 2y_i x_1 + x_1^2) - \frac{(x_1^2 - 100x_1 + 2500)}{200}\right\} =$$

$$= exp\left\{-\frac{x_2}{2}\cdot\left(\sum_{i=1}^{130} y_i^2 - 2x_1\sum_{i=1}^{130} y_i + 130x_1^2\right) - \frac{x_1^2}{200} + \frac{x_1}{2} - \frac{25}{2}\right\} =$$

$$= exp\left\{-\frac{1}{2}\cdot\left(x_2\sum_{i=1}^{130} y_i^2 - 2x_1x_2\sum_{i=1}^{130} y_i + 130x_2x_1^2\right) + \frac{x_1^2}{100} - x_1\right\} exp(25) =$$

onde $\sum_{i=1}^{130} y_i = S$

$$= exp\left\{-\frac{1}{2}\cdot\left(-2x_1x_2S + 130x_2x_1^2 + \frac{x_1^2}{100} - x_1\right)\right\} =$$

$$= exp\left\{-\frac{1}{2}\cdot\left[\left(130x_2 + \frac{1}{100}\right)x_1^2 - (1 + 2x_2S)\,x_1\right]\right\} =$$

$$= exp\left\{-\frac{1}{2}\cdot\left[\frac{x_1^2}{\left(\frac{1}{130x_2+\frac{1}{100}}\right)} - 2\frac{\left(\frac{1}{130x_2+\frac{1}{100}}\right)\cdot\left(\frac{1}{2} + x_2S\right)x_1}{\left(\frac{1}{130x_2+\frac{1}{100}}\right)}\right]\right\} =$$

$$= exp\left\{-\frac{1}{2}\cdot\frac{\left[x_1^2 - 2\left(\frac{1}{130x_2+\frac{1}{100}}\right)\left(\frac{1}{2} + x_2S\right)x_1\right]}{\left(\frac{1}{130x_2+\frac{1}{100}}\right)}\right\}$$

Desenvolvendo-se a dsitribuição de uma variável aleatória $X \sim N(\mu, \sigma^2)$, tem-se que:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot exp\left\{-\frac{1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}\right\} \propto exp\left\{-\frac{1}{2} \cdot \frac{(x^2 - 2x\mu + \mu^2)}{\sigma^2}\right\} \propto exp\left\{-\frac{1}{2} \cdot \frac{(x^2 - 2x\mu)}{\sigma^2}\right\}$$

Então, comparando-se o *kernel* (núcleo) de $f_1(x_1) obtido com o núcleo da distribuição normal apresentado acima ,temos que:

$$X_1|X_2 = x_2 \sim N\left(\mu = \frac{1/2 + x_2 \sum_{i=1}^{130} y_i}{130x_2 + 1/100}, \sigma^2 = \frac{1}{130x_2 + 1/100}\right)$$

Já para x_2 temos a seguinte função:

$$f_2(x_2) \propto x_2^{64,001} exp\left\{-\frac{x_2}{2}\sum_{i=1}^{130}(y_i - x1)^2 - 0,001x_2\right\} =$$

$$f_2(x_2) \propto x_2^{64,001} exp\left\{-x_2\left[0,001 + \frac{\sum_{i=1}^{130}(y_i - x1)^2}{2}\right]\right\} =$$

Dado que a distribuição Gamma tem os seguintes parâmetros e a seguinte forma para o núcleo:

$$\frac{1}{\Gamma(k)\theta^k} x^{k+1} \exp\left\{-\frac{x}{\theta}\right\} \propto x^{k+1} \exp\left\{-\frac{x}{\theta}\right\}$$

Temos que:

$$X_2|X_1 = x_1 \sim Gamma\left(k = 65,001; \theta = \left(0,001 + \frac{\sum_{i=1}^{130}(y_i - x_1)^2}{2}\right)^{-1}\right)$$