Descrevendo Tempo de Execução Usando Análise Exploratória de Dados

Coloque seu nome aqui (ID=5)

Data de entrega: 30/04/2021

Descrição da atividade

O objetivo desta atividade é aplicar as técnicas de análise exploratória de dados (AED) vistas em aula:

- medidas de tendência central (média, mediana, moda, quartis, percentis);
- medidas de dispersão (amplitude, variância, desvio padrão, coeficiente de variação);
- gráficos para inspeção visual (dispersão, histograma, boxplot, ...).

A métrica de desempenho usada será o tempo de execução de uma determinada função. Você irá realizar diversas medições do tempo de execução, aplicar as técnicas de AED para investigar os dados obtidos, e descrever a métrica com base nas conclusões dessa investigação.

Algumas recomendações:

- Se você não estiver habituado com R Markdown, acostume-se a processar com frequência o documento, usando o botão **Knit**. Isso permitirá que eventuais erros no documento ou no código R sejam identificados rapidamente, pouco depois de terem sido cometidos, o que facilitará sua correção. Na verdade, é uma boa ideia você fazer isso **agora**, para garantir que seu ambiente esteja configurado corretamente. Se você receber uma mensagem de erro do tipo *Error in library(foo)*, isso significa que o pacote `foo` não está instalado. Para instalar um pacote, execute o comando `install.packages("foo")` no Console, ou clique em *Tools -> Install Packages*.
- A seção "Análise" deste documento é o seu *playground*. Brinque nela o quanto quiser, e não preocupe-se em remover partes desnecessárias antes de entregar a atividade. Certifique-se apenas que o arquivo .Rmd pode ser processado sem erros.
- Após concluir a atividade, você deverá submeter no Moodle um arquivo ZIP contendo:
  - o arquivo fonte .Rmd;
  - a saída processada (PDF ou HTML) do arquivo .Rmd;
  - outros arquivos necessários ao processamento do arquivo .Rmd (se houver).

Configuração

Em um documento R Markdown, é praxe que o primeiro bloco (*chunk*) contenha comandos de inicialização e configuração do ambiente, como carga de pacotes. Cada bloco do documento pode receber um nome. Neste documento, o bloco de inicialização (logo abaixo) recebeu o nome de `config`, mas outro nome (ou mesmo nome nenhum) poderia ter sido usado.

No bloco de inicialização, a linha 1 carrega o pacote `microbenchmark`, que será usada para efetuar as medições de tempo. Caso a biblioteca não esteja instalada, veja as recomendações na seção anterior.

A linha 2 carrega o arquivo `asc-tempo-func.R`, que contém a função `func()`, cujo tempo de execução será medido.

A linha 3 fixa a semente aleatória usada pelo R, o que é essencial para garantir que seus resultados reproduzíveis (ou seja, que você obterá sempre os mesmos resultados ao processar o documento).

A linha 4 atribui à variável `id` o identificador (ID) numérico usado na atividade. Você receberá um ID único, que garante que cada aluno obtenha resultados distintos dos demais. Edite essa linha, substituindo o número 99 pelo ID que você recebeu.

ATENÇÃO: você não deve alterar as três primeiras linhas deste bloco.

```
library(microbenchmark)      # carrega pacote
source("asc-tempo-func.R")   # carrega arquivo que contem a funcao
set.seed(1234)               # fixa semente para garantir resultados reproduziveis
id <- 5                     # substitua 99 pelo seu proprio ID
```

Realização do experimento

O tempo de execução de uma função raramente é constante. Na verdade, o normal é que ocorram variações, tanto porque o código efetivamente executado pode não ser sempre o mesmo (se houver desvios condicionais, por exemplo) quanto devido a flutuações de carga no sistema durante a execução. Portanto, para conhecer o tempo de execução de uma função, é preciso medir esse tempo um certo número de vezes, e analisar o conjunto de medições.

Para medir o tempo de execução será usado o pacote `microbenchmark`. A função `microbenchmark()` executa um código R um número especificado de vezes, medindo o tempo de cada execução. A função retorna um *data frame*, no qual a coluna `time` armazena os tempos de execução, **em nanossegundos (ns)**.

O blocos abaixo usa `microbenchmark()` para obter os tempos de execução e armazena esses tempos de execução, **já convertidos para milissegundos (ms)**, no vetor `tempos.ms`. A variável `nrep`, definida na linha 2, determina quantas vezes a função será executada. Caso você ache necessário aumentar ou diminuir o número de repetições, basta alterar o valor de `nrep` (e processar novamente o documento).

```
# quantas repeticoes serao executadas
nrep <- 3
# executa 'nrep' repeticoes de 'func(id)'
mbm <- microbenchmark(func(id), times=nrep)
# microbenchmark() retorna tempos em ns, convertendo para ms
tempos.ms <- mbm$time/1e6
```

Análise

Nesta seção você deve realizar a análise numérica e gráfica dos tempos de execução contidos na variável `tempos.ms`. Recomenda-se que cada comando fique em um bloco separado. Anote suas observações sobre o resultado de cada comando ou grupo de comandos, mantendo um registro do que você estava pensando durante a análise; isso tornará mais fácil escrever as suas conclusões, evitando que você precise confiar em sua memória.

Abaixo há dois exemplos (o primeiro com `mean()` e `median()` e o segundo com `plot()`); você deve adaptar esses exemplos para suas necessidades, acrescentando, modificando e/ou removendo comandos.

1. A média é menor que a mediana. Será que essa diferença é relevante?

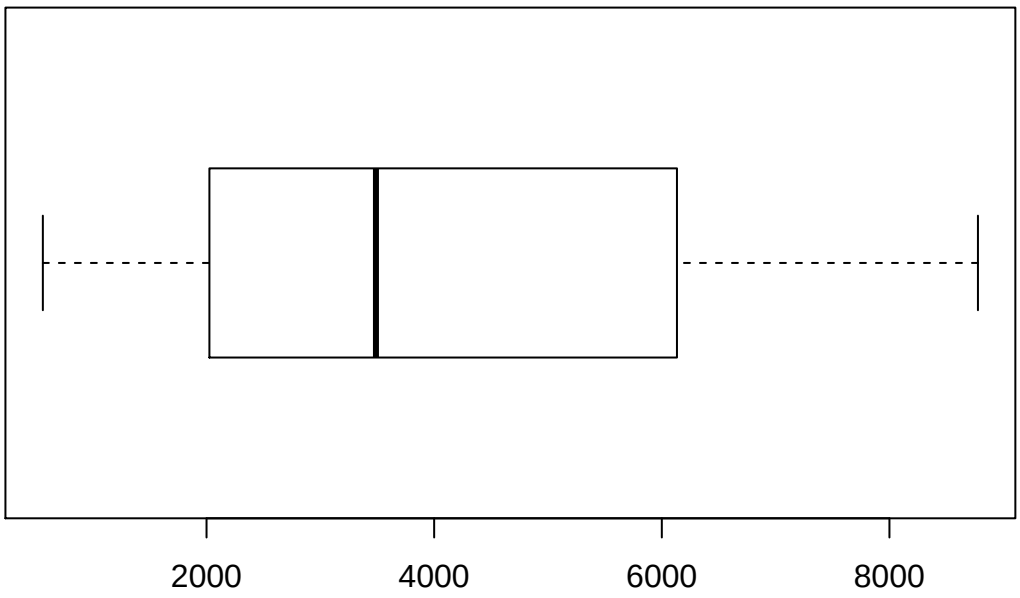
```
mean(tempos.ms)
```

```
## [1] 4276.047
```

```
median(tempos.ms)
```

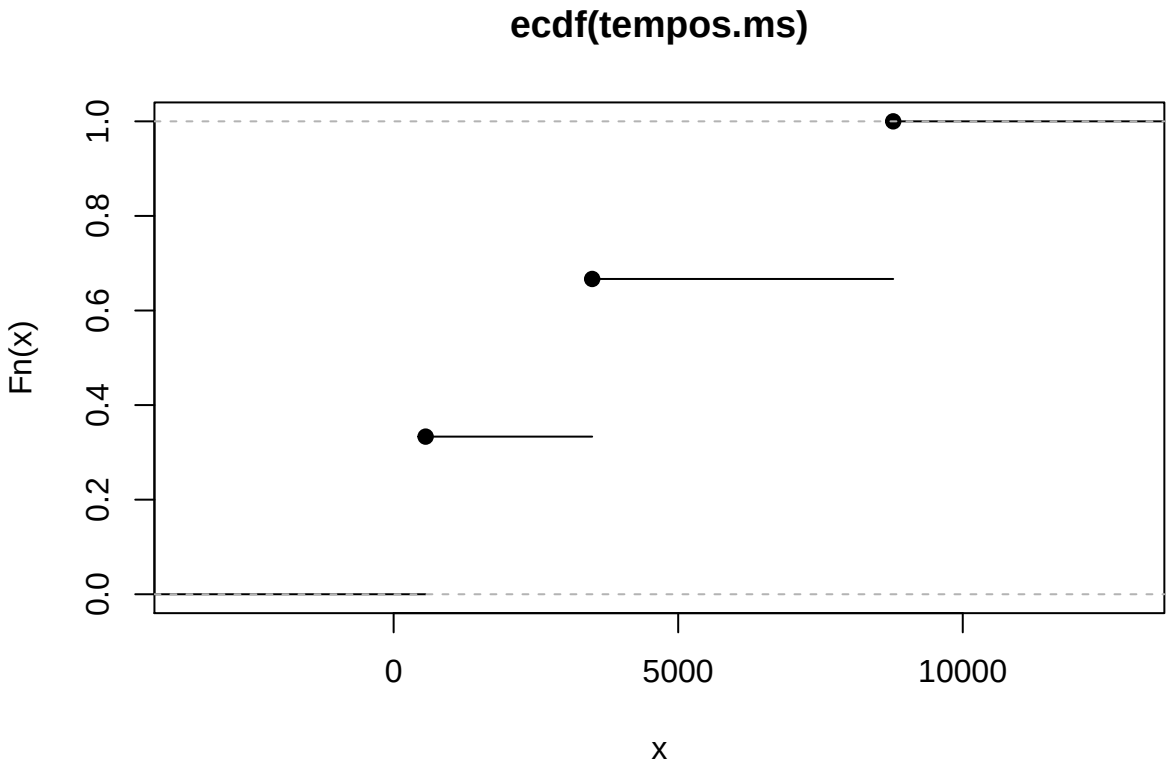
```
## [1] 3488.613
```

```
boxplot(tempos.ms, horizontal = T)
```



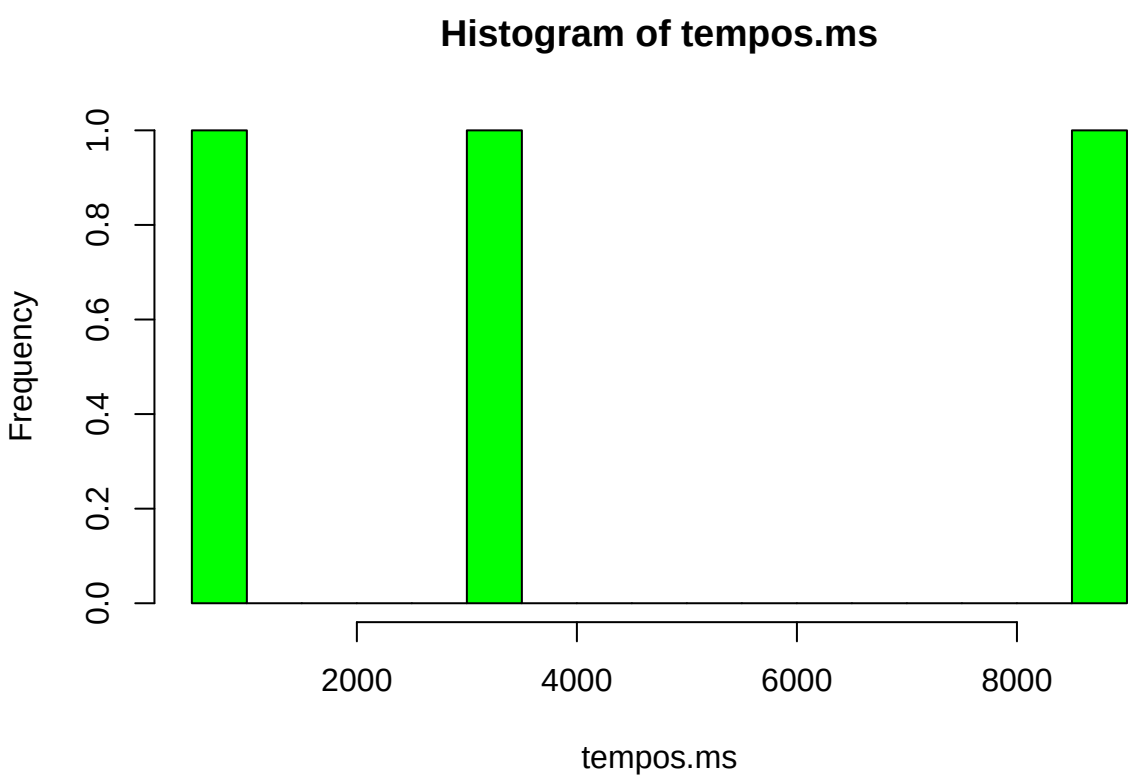
- Observando apenas com o boxplot fica difícil dizer que o gráfico será simétrico, mas ao menos não existem outliers, mas como a máxima está bem distante da mediana eu arriscaria dizer que isso seria uma assimétrica à direita

```
plot(ecdf(tempos.ms))
```



- Colocando o ecdf piorou a análise

```
hist(tempos.ms, col = "GREEN", breaks=20)
```



- Com o histograma é possível ver algo uniforme, mas também muito distantes entre si

```
tempos.ms
```

```
## [1]  561.9634 8777.5662 3488.6125
```

```
length(tempos.ms)
```

```
## [1] 3
```

- Printei a quantidade e existem poucos dados para realizar a análise plotando o gráfico de ecdf ou histograma

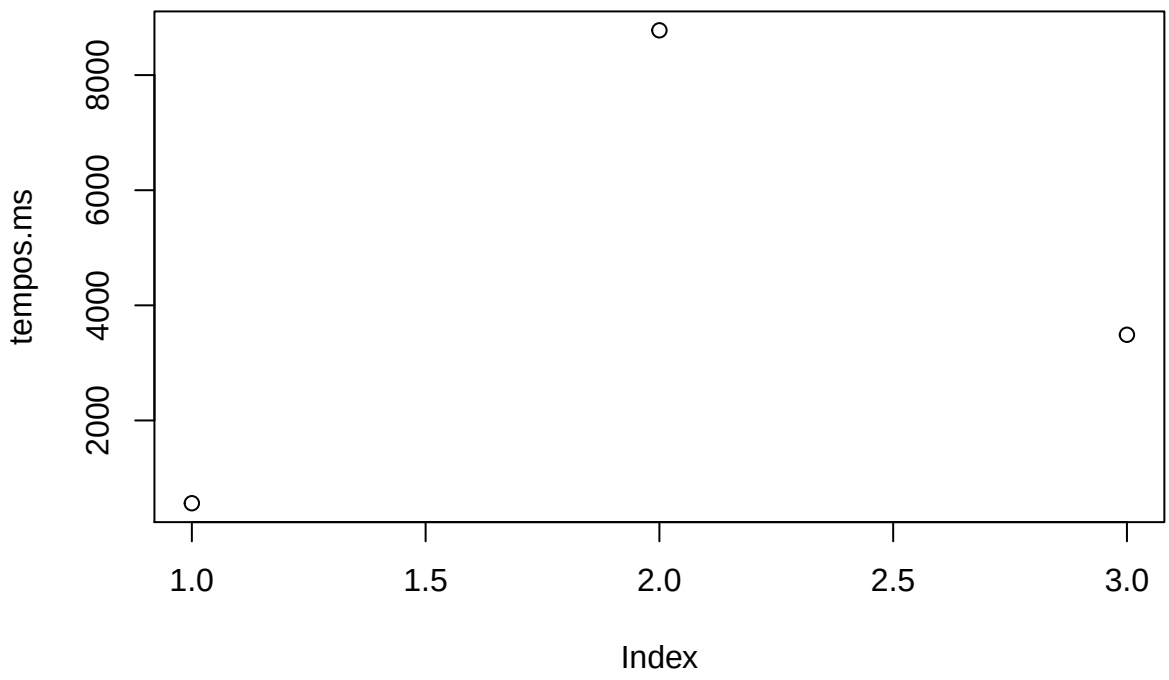
```
summary(tempos.ms)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     562    2025    3489    4276    6133    8778
```

- Acredito que essa diferença não é tão relevante, pois existem poucos dados no conjunto para ser analisado, então como so existem outros dois dados alem da mediana, fica complexo dizer que a diferença entre eles tem relevância

2. O gráfico mostra que as observações estão espalhadas entre 850 e 1100, então talvez dê para considerar que a média e a mediana são equivalentes. Acho que preciso de um número maior de medições para concluir alguma coisa.

```
plot(tempos.ms)
```

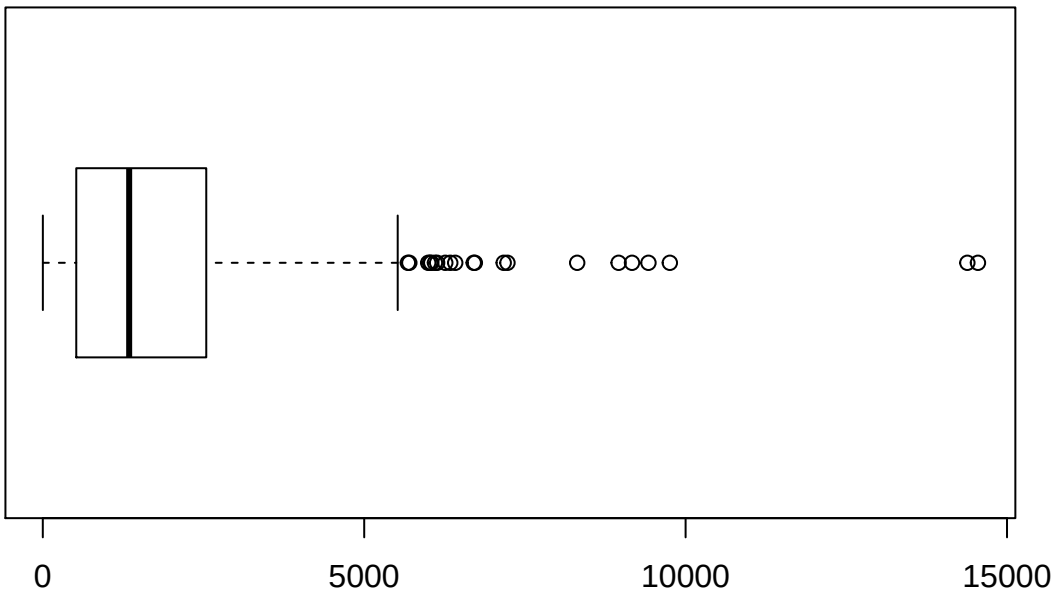


- Bom, na verdade as observações estão espalhadas entre 561.963376 e 8777.566154. De fato, para realizar uma análise mais acertiva sobre os dados é necessário uma maior quantidade de medições
- Lembrei que é possível aumentar a quantidade de repetições da função func para analisar novamente

```
# executa 'nrep' repeticoes de 'func(id)'
mbm2 <- microbenchmark(func(id), times=300)
# microbenchmark() retorna tempos em ns, convertendo para ms
tempos2.ms <- mbm2$time/1e6
```

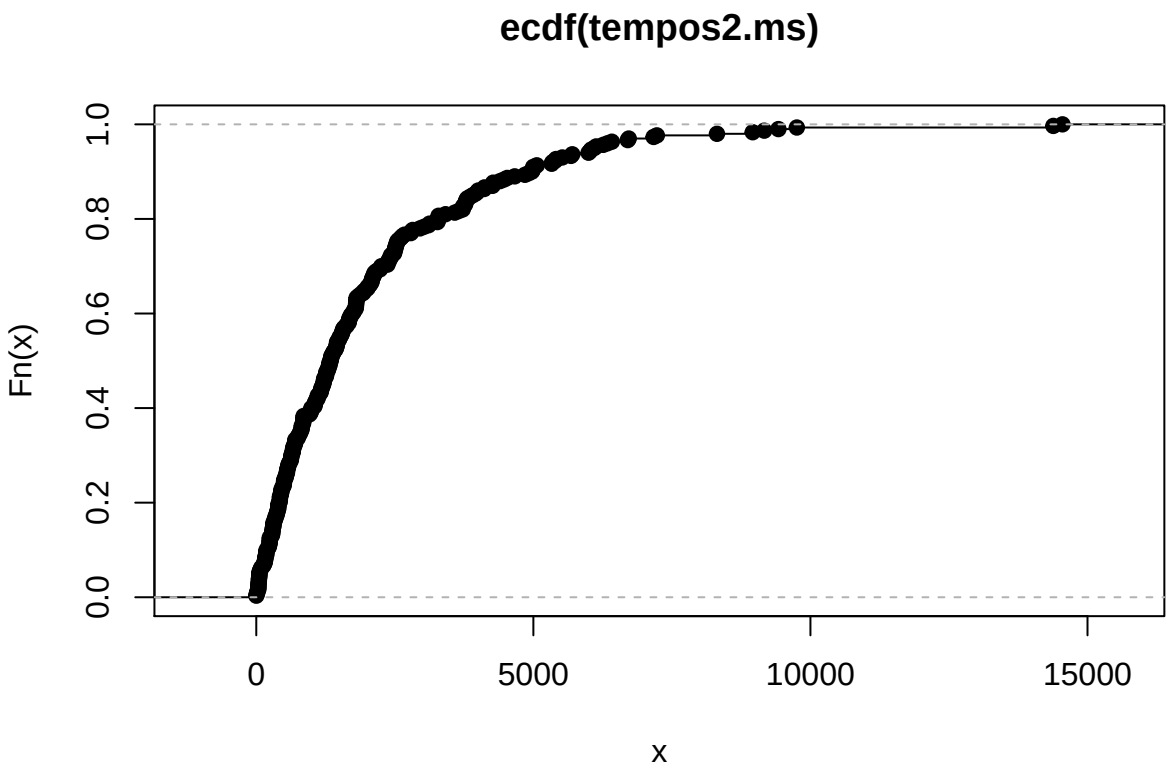
- testado com 3000 reps travou... vamos tentar com 300
- talvez com 50
- demorou mas foi com 50, então vou printar todos as analises e graficos com 300 e esperar o build

```
boxplot(tempos2.ms, horizontal = T)
```



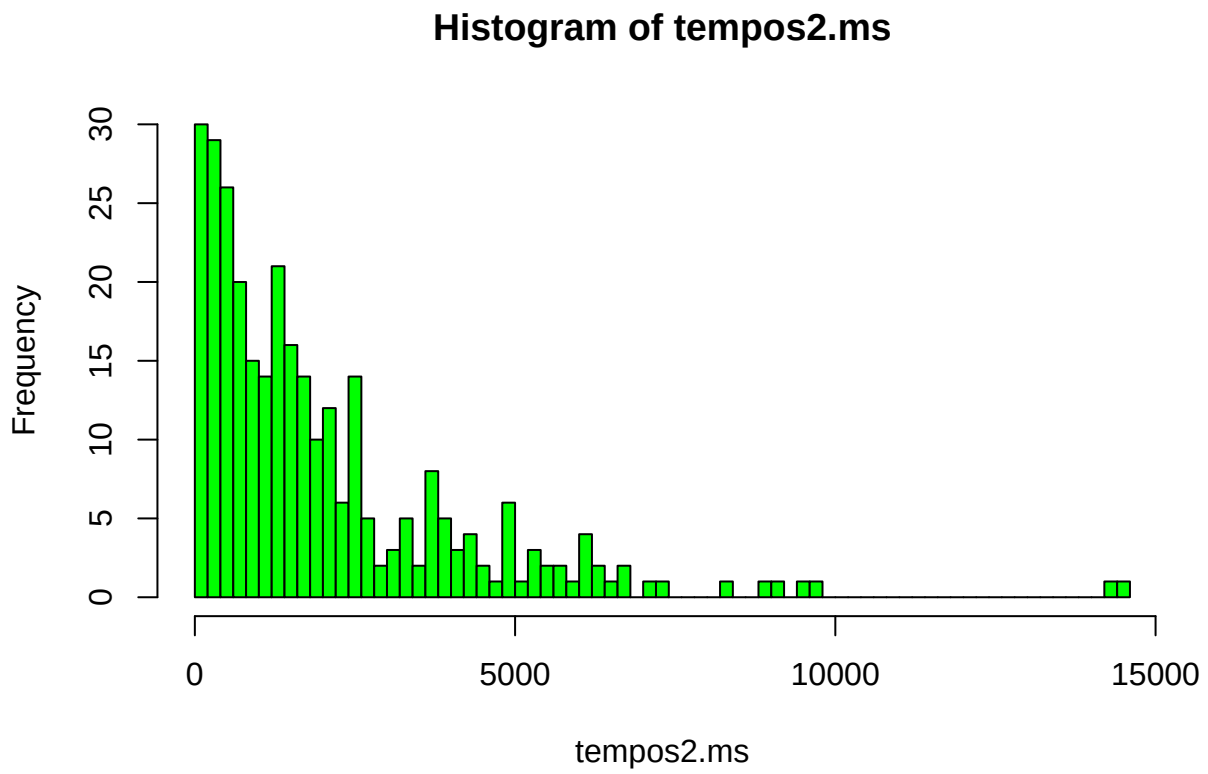
- Agora com mais amostras, ja aparecem os outliers, antes realmente era impossível ter outliers com apenas 3 registros pois um deles era o minimo, outro o máximo e o outro a mediana.
- O boxplot ja auxilia apontando que a distribuição é assimétrica e possívelmente à direita pois o 3 quartil esta mais esticado do que o primeiro, além dos outliers à direita

```
plot(ecdf(tempos2.ms))
```

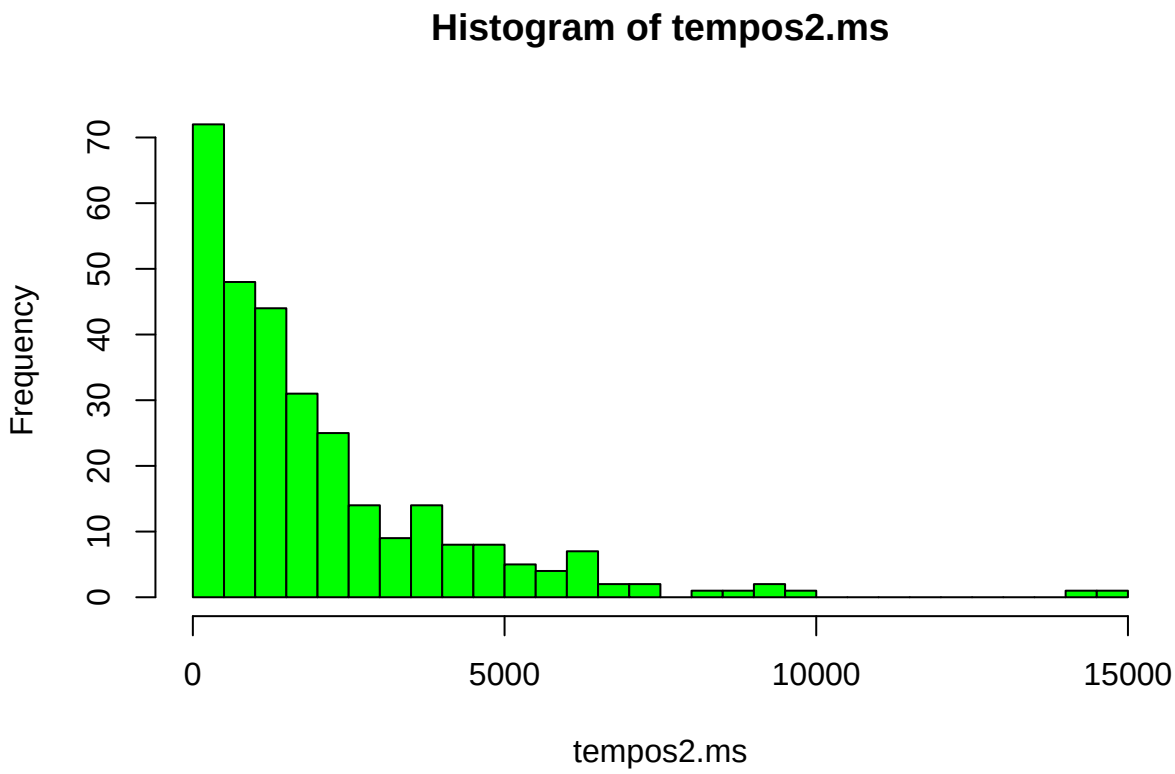


- Agora sim o ecdf faz sentido e pode ser analisado em que basicamente 80% dos dados estão abaixo de 2500, criando uma curva bem acentuada a direita, auxiliando no fundamento de que a distribuição será assimétrica à direita

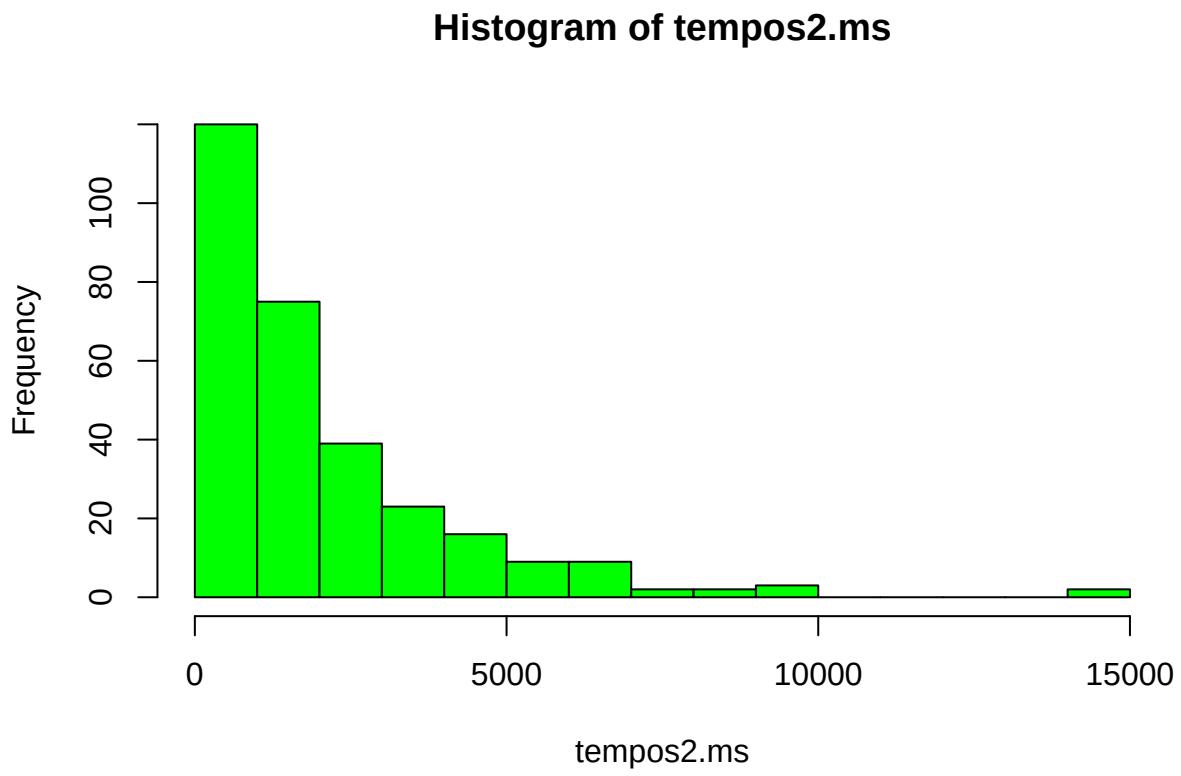
```
hist(tempos2.ms, col = "GREEN", breaks=100)
```



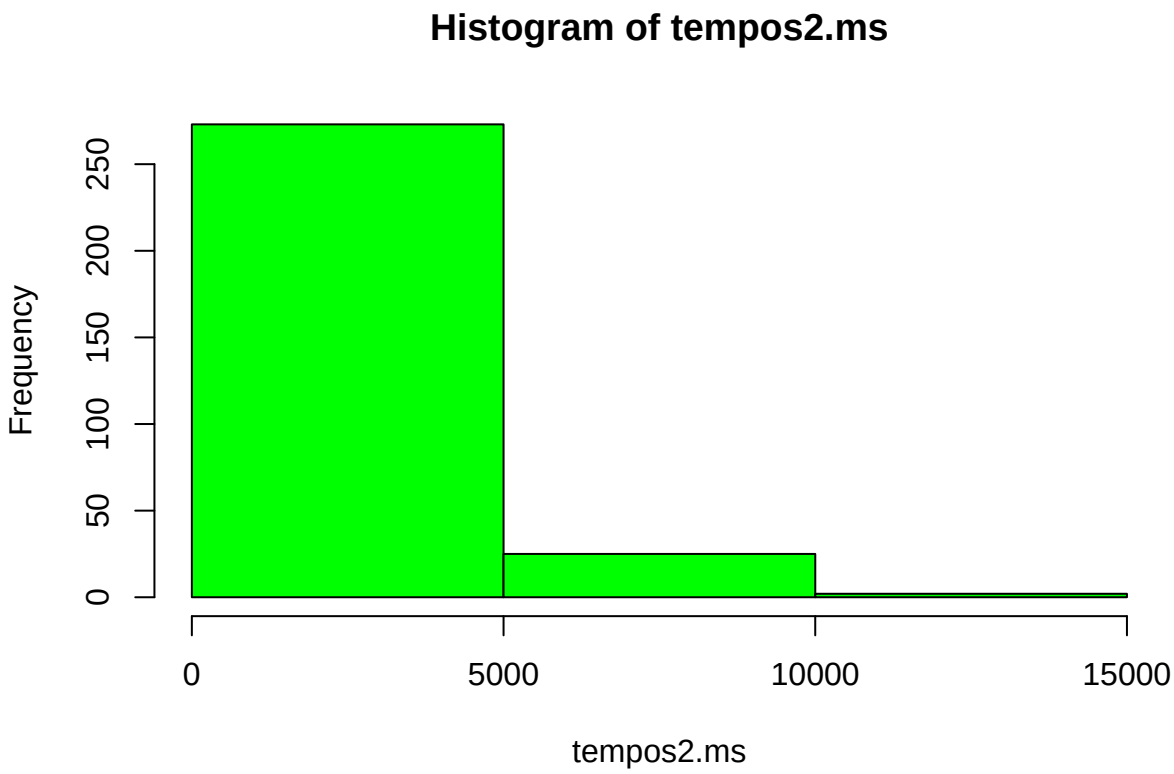

```
hist(tempos2.ms, col = "GREEN", breaks=50)
hist(tempos2.ms, col = "GREEN", breaks=30)
```




```
hist(tempos2.ms, col = "GREEN", breaks=20)
```

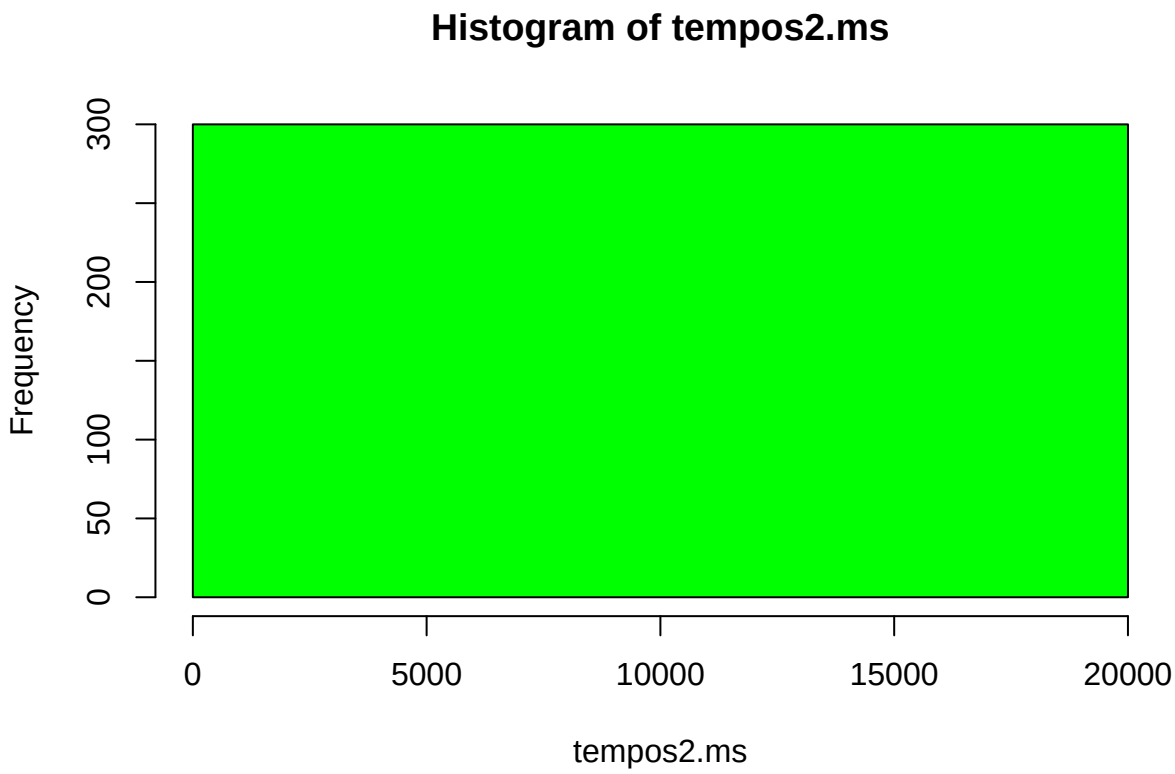



```
hist(tempos2.ms, col = "GREEN", breaks=5)
hist(tempos2.ms, col = "GREEN", breaks=3)
```



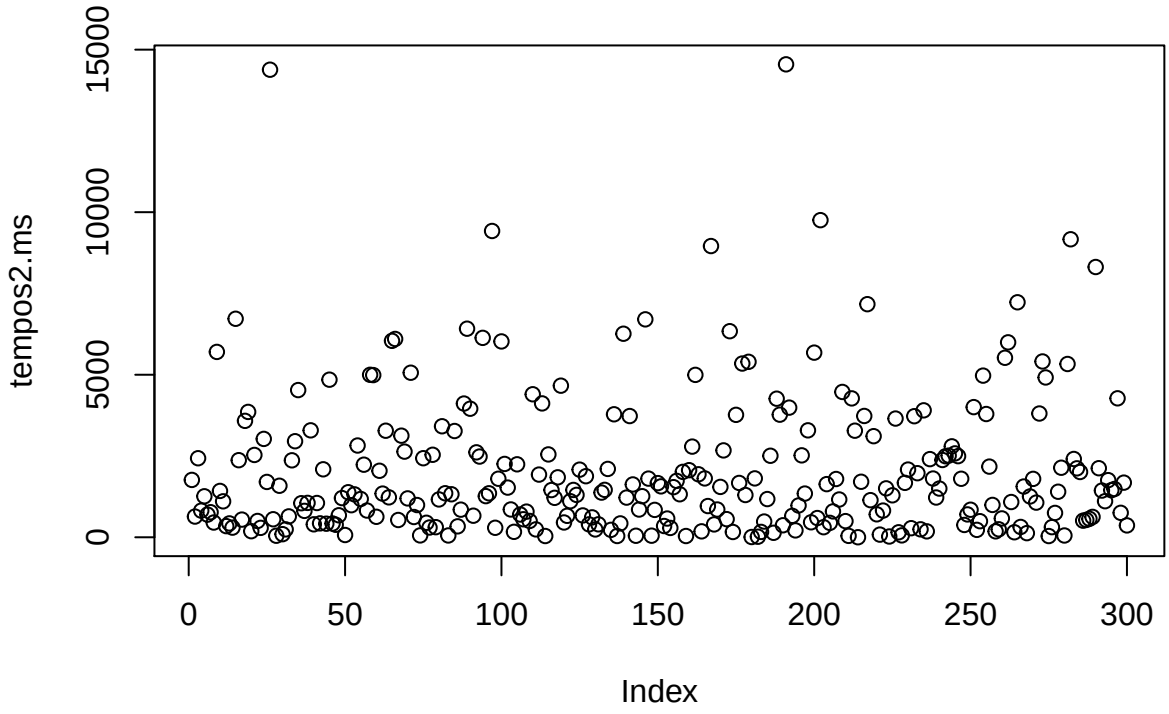
- Todos os histogramas (menos o com 1 break) confirmaram a afirmação de distribuição assimétrica à direita

```
hist(tempos2.ms, col = "GREEN", breaks=1)
```



- Utilizar o histograma com apenas um break não foi a melhor ideia

```
plot(tempos2.ms)
```




```
length(tempos2.ms)
```

```
## [1] 300
```

- Após fazer as análises, estava observando no console e mesmo utilizando 300 reps, o conjunto de amostras de tempo tem apenas 300 amostras.

```
summary(tempos2.ms)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     0.771   525.948  1343.763  2013.504  2540.489 14547.986
```

```
diffMeanMedian = diff(range(median(tempos2.ms), mean(tempos2.ms)))
diffMaxMin = diff(range(tempos2.ms))
diffMeanMedianPercentage = (diffMeanMedian * 100) / diffMaxMin
```

- No fim a média e a mediana são relativamente próximas, pois se for levar em consideração que a diferença da distribuição entre o máximo 1.454799×10^4 e o mínimo 0.77 é de 1.454722×10^4 representando 100% do range dos dados, a diferença entre a média 2013.5 e a mediana 1343.76 é de 670 o que representaria uma diferença de apenas 4.6%

Conclusão

Nesta seção do documento você deve apresentar sua análise final. Usando os conceitos vistos em aula, descreva o tempo de execução da função `func()`. Atente para fatores como:

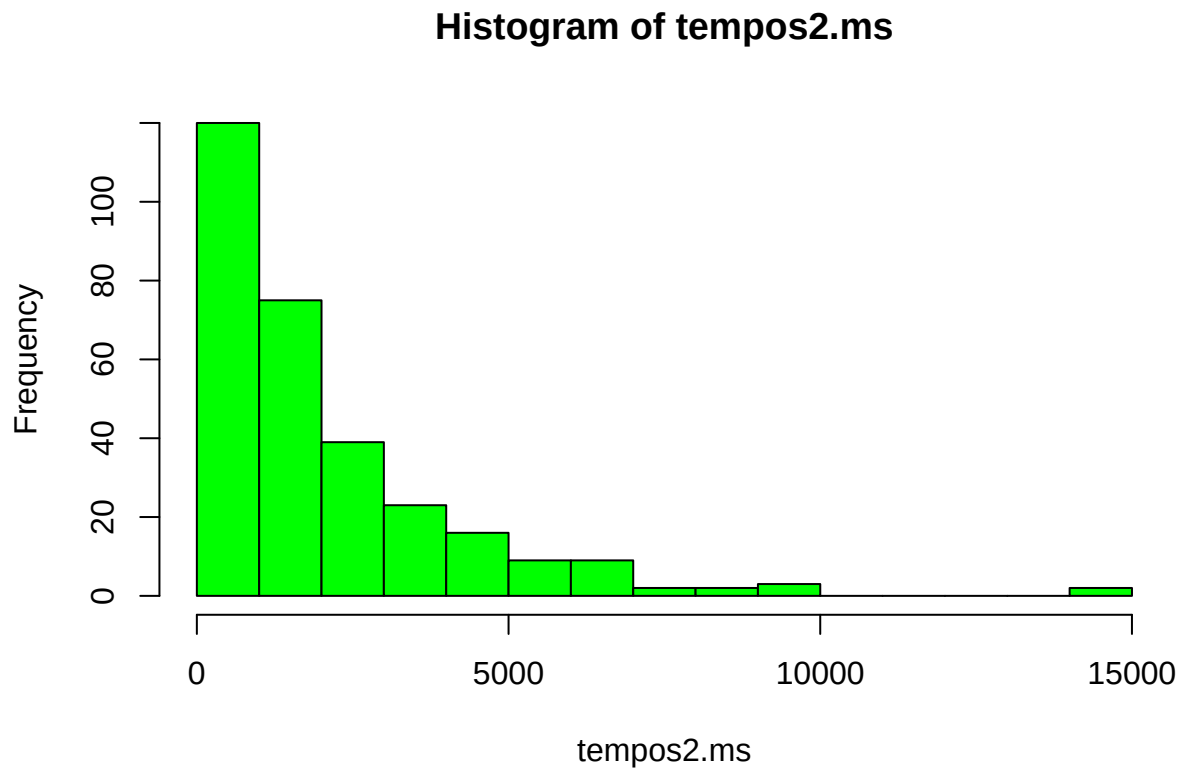
- a medida de tendência central mais apropriada;
- a variabilidade das medidas;
- se usar algum gráfico se faz necessário.

Note que o objetivo não é mostrar todas as medidas de tendência e variabilidade nem incluir todos os gráficos – o objetivo é que *você* analise os dados e apresente-os da forma mais sucinta possível que permita interpretá-los corretamente.

Um exemplo de conclusão poderia ser o seguinte:[1]

O histograma abaixo mostra que o tempo de execução da função `func()` tem distribuição assimétrica à direita. A mediana é de 1343.76 ms, e a média truncada excluindo 10% das extremidades fica com 1630.35 e o desvio padrão de 2165.47 ms. 83% dos tempos de execução mensurados estão abaixo de 3766.9 ms.

```
hist(tempos2.ms, col = "GREEN", breaks=20)
```



[1] O histograma do exemplo foi incluído como uma figura, mas no seu documento você deve usar comandos do R para gerar os gráficos pertinentes.