|  |  |
| --- | --- |
| Logotipo, Ícone  Descrição gerada automaticamente | UNIVERSIDADE DE BRASÍLIA  FACULDADE DE TECNOLOGIA  ENGENHARIA CIVIL |

# Análise da curva de Tensão-Deformação com auxílio da linguagem de programação estatística *R Software*

*Vítor Luís Costa Azevedo – Julho/2024*

## Procedimentos iniciais e importação dos dados.

O R é um software gratuito de computação estatística e gráficos desenvolvido pelos laboratórios Bell1 (atualmente *Lucent Technologies*), usado em diversas áreas da ciência por sua liberdade na representação gráfica e velocidade de análise de grandes bases de dados. O R também é aclamado pela grande disponibilidade de livrarias (*libraries*) que exponenciam as possibilidades de uso da linguagem.

Neste texto, será utilizado a livraria ggplot2, para representação gráfica dos dados, além de códigos encontrados na base do R. Para fins de explicação, o código utilizado será escrito conforme abaixo, que importa a livraria supracitada. Nele, o texto em vermelho será lido pelo R, enquanto o texto em azul é um comentário, colocado no texto para facilitar e explicar a leitura do código:

|  |
| --- |
| library(ggplot2) # Importação da livraria |

Assim, agora é necessário importar os dados para um *dataframe*, que irá armazenar em linhas e colunas. No código, o *dataframe* será chamado de *df.* No código abaixo, são colocadas *Booleans*(isto é, afirmações de verdadeiro ou falso) para evitar o a leitura incorreta pelo código. O primeiro trecho lê a tabela copiada pelo usuário (“*clipboard*”), e o converte num *dataframe*. O segundo, converte esse *dataframe* em um arquivo csv, para que os dados possam ser lidos sem necessidade de copiar os dados ao *clipboard*. O terceiro, ativo, importa os dados do arquivo salvo. Com a execução do código abaixo, é possível iniciar a análise.

|  |
| --- |
| data\_path <- "data.csv" # arquivo csv  # Importação do arquivo, do clipboard para um dataframe do R  get\_from\_clipboard <- FALSE  if (get\_from\_clipboard) {    df <- read.table(file = "clipboard", sep = "\t", header = TRUE)  }  write\_into\_csv <- FALSE  # Dos dados salvos em df, salvá-los em csv, para facilitar importação  if (write\_into\_csv) {  write.csv(df, file = data\_path, row.names=FALSE)  }  # Bool para evitar leitura incorreta  get\_from\_csv <- TRUE  if (get\_from\_csv) {    df <- read.csv(data\_path)    cat("Data from csv file read.\n")  } |

Podemos checar se os dados foram importados corretamente com o código abaixo, utilizando as funções *names*(), que imprime os nomes de cada coluna, e *head*(), que imprime as primeiras linhas da tabela.

|  |
| --- |
| > names(df)  [1] "tempo\_s" "deformacao" "forca\_N"  > head(df)  tempo\_s deformacao forca\_N  1 0.016667 -0.0008003 10.348  2 0.233330 0.0000000 51.742  3 0.316670 -0.0008003 93.135  4 0.333330 -0.0008003 196.620  5 0.350000 -0.0008003 320.800  6 0.416670 -0.0016007 362.190 |

## Inserção das constantes obtidas em laboratório.

Agora, é conveniente inserir os valores e constantes anotados em laboratório. No R, para associar uma variável a um valor, utiliza-se o operador “<-“. O código abaixo cria essas variáveis e os associa a um valor. As unidades de medida de cada variável estão dispostas em comentário.

|  |
| --- |
| # Constantes e Valores anotados  nominal\_diameter <- 12.5 #mm, diâmetro nominal  mass <- 545.8 #g, massa  l\_total <- 600 #mm, comprimento total  l\_exp <- 500 #mm, comprimento tracionado  mass\_esp <- 0.00785 #gmm-3, massa específica do aço |

## Massa linear, diâmetro efetivo, alongamento e coluna de tensão

Para calcular a Massa Linear do aço, basta dividir sua massa pelo comprimento total. Note que, no R, assim como em outras linguagens de programação, o operador de divisão será o “/”. Deste modo, têm-se que:

|  |
| --- |
| # Cálculo da Massa Linear  linear\_mass <- mass / l\_total |

Para o cálculo do diâmetro efetivo e alongamento, por questões de simplificação, é interessante definir uma função para cada. No R, pi já definido, o que já facilita o cálculo:

|  |
| --- |
| # Define a função para calcular diametro efetivo  calc\_effective\_diameter <- function(mass, gamma\_esp, length) {    d <- 2 \* sqrt(mass / (pi \* gamma\_esp \* length))    return(d)  }  # Calcula com base nas constantes já definidas  effective\_diameter <- calc\_effective\_diameter(    mass = mass,    gamma\_esp = mass\_esp,    length = l\_total  )  # Cálculo do alongamento  calc\_lengthening <- function(L0, Lf) {    return((100 \* (Lf - L0) / L0))  }  lengthening <- calc\_lengthening(L0 = 120, Lf = 138.1) |

Por fim, como pôde ser observado na primeira seção, o *dataframe* df apresenta apenas força, em newtons, e não há coluna para tensão. Isso não é um problema, pois o R permite a edição de seus *dataframes* de forma intuitiva:

|  |
| --- |
| # Obter tensão  area\_section <- mass / (mass\_esp \* l\_total) # Área da seção, mm²  df["tensao\_MPa"] <- df["forca\_N"] / area\_section # Cria nova coluna  > names(df)  [1] "tempo\_s" "deformacao" "forca\_N" "tensao\_MPa"  > head(df)  tempo\_s deformacao forca\_N tensao\_MPa  1 0.016667 -0.0008003 10.348 0.08929842  2 0.233330 0.0000000 51.742 0.44650938  3 0.316670 -0.0008003 93.135 0.80371171  4 0.333330 -0.0008003 196.620 1.69673910  5 0.350000 -0.0008003 320.800 2.76835471  6 0.416670 -0.0016007 362.190 3.12553115 |

## Desenho do gráfico e análise do limite de escoamento

Como mencionado, foi usada a livraria ggplot2 para desenho dos gráficos. Deste modo, foi criado o objeto que representará o gráfico, chamado *plot*, a serem adicionados demais características. Por exemplo, os pontos do gráfico são adicionados por geom\_point(), enquanto a reta de tendência elástica é adicionada por geom\_abline(). Para determinar o limite elástico, foi utilizado um *while loop*, que analisa, linha por linha, se o valor da tensão é crescente. No momento que este se torna decrescente, é justo afirmar que acaba o limite elástico. As linhas, chamadas *rows*, dos dados que compõe o *dataframe* df são copiadas a um segundo, chamado de *elastic\_df*. Por fim, é interessante analisar o momento durante o escoamento, e para isso realiza-se um “zoom”, por meio da função ­*get\_zoom\_plot()*, onde define-se um segundo gráfico, mais robusto.

|  |
| --- |
| ## Desenhar gráfico ##  # Definir valores limites do gráfico  max\_value <- c(max(df$tensao\_MPa), max(df$deformacao))  plot\_breaks\_x <- seq(from = 0, to = max\_value[2], by = .1)  plot\_breaks\_y <- seq(from = 0, to = max\_value[1], by = (max\_value[1]) / 10)  # Cria um objeto de plot, com base em df  plot <- ggplot(df, aes(x = deformacao, y = tensao\_MPa))  # Inserir linhas horizontais  for (y in plot\_breaks\_y) {  plot <- plot + geom\_hline(yintercept = y, color = "grey")  }  # Separar dados presentes apenas no regime elástico  # Em outro dataframe, elastic\_df  i <- 1  n\_rows <- nrow(df)  elastic\_df <- data.frame()  # Aqui o código analisa se o próximo valor de tensão é superior ao anterior.  # Se não, assume-se que não há mais regime elástico.  while (i < n\_rows && df$tensao\_MPa[i + 1] > df$tensao\_MPa[i]) {  elastic\_df <- rbind(elastic\_df, df[i, ])  i <- i + 1  }  # Calcular coeficiente de elasticidade  elastic\_lm <- lm(tensao\_MPa ~ deformacao, data = elastic\_df) # modelo linear  elastic\_coef <- elastic\_lm$coefficients[2]  y\_cross\_value <- elastic\_lm$coefficients[1]  plot <- plot +    theme\_classic() + # define a aparência    geom\_point(size = .1,               col = "#000000",               aes(color = "Data points")) + # Adiciona os pontos    scale\_x\_continuous(limits = c(0, max\_value[2]),                       breaks = plot\_breaks\_x) + # Define as quebras no eixo x    scale\_y\_continuous(limits = c(0, max\_value[1]),                       breaks = plot\_breaks\_y) + # Define as quebras no eixo y    geom\_abline(slope = elastic\_coef,                intercept = y\_cross\_value,                color = "red") + # Adiciona a reta de elasticidade, em vermelho    labs(x = "deformação (mm/mm)",         y = "Tensão (MPa)") # Define o nome dos eixos  # A tensão de escoamento é a tensão máxima no regime elástico  tensao\_esc <- max(elastic\_df$tensao\_MPa)  # Função para analizar uma região específica dos dados  get\_zoom\_plot <- function(x0, x1, dframe) {  n\_rows <- nrow(dframe)  zoom\_df <- data.frame()  for (n in seq(n\_rows)) {  if (x1 >= dframe$deformacao[n] && dframe$deformacao[n] >= x0) { #nolint  zoom\_df <- rbind(zoom\_df, dframe[n, ])  }  }  zoom\_plot <- ggplot() +  geom\_point(data = zoom\_df,  aes(x = zoom\_df$deformacao, y = zoom\_df$tensao\_MPa)) +  theme\_classic()  return(zoom\_plot)  }  # Observar o limite de elasticidade, escolhendo pontos específicos  zoom\_plot\_elastic <- get\_zoom\_plot(.15, .2, df) |

Os gráficos obtidos serão demonstrados na seção 6.

## Tensão de ruptura e tensão máxima

Por fim, é possível definir a tensão de ruptura, a última tensão no *dataframe* e tensão máxima, a tensão de maior magnitude obtida no ensaio. É possível obter esses dados por meio das funções *tail()* e *max()*, respectivamente.

|  |
| --- |
| tensao\_max <- max(df$tensao\_MPa) # máximo de df$tensao  tensao\_rup <- tail(df, n = 1)$tensao\_MPa # último valor de df$tensao |

## Resultados e outras considerações

Para exibição dos resultados finais, definiu-se uma função. Note que *cat()* é a função que imprime os dados no terminal do R, enquanto ggsave salva os gráficos criados em imagens:

|  |
| --- |
| print\_results <- function() {    cat(rep("-", 20), "\n")    cat("Tensão de Escoamento (MPa):", tensao\_esc, "\n")    cat("Tensão Máxima (MPa):", tensao\_max, "\n")    cat("Tensão de Ruptura (MPa):", tensao\_rup, "\n")    cat("Alongamento (%):", lengthening, "\n")    cat("Massa Linear (g/mm):", linear\_mass, "\n")    cat("Módulo de Elasticidade:", elastic\_coef, "\n")    cat(rep("-", 20), "\n")    ggsave(file = "Main\_Plot.png", plot = plot)    ggsave(file = "Zoom\_Plot.png", plot = zoom\_plot\_elastic)  } |

Assim, roda-se a função:

|  |
| --- |
| > print\_results()  - - - - - - - - - - - - - - - - - - - -  Tensão de Escoamento (MPa): 623.3199  Tensão Máxima (MPa): 770.1342  Tensão de Ruptura (MPa): 398.1063  Alongamento (%): 15.08333  Massa Linear (g/mm): 0.9096667  Módulo de Elasticidade: 4151.402  - - - - - - - - - - - - - - - - - - - -  Saving 3.06 x 7.31 in image  Saving 3.06 x 7.31 in image |

As imagens salvas estão representadas na figura (1) e figura (2):

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figura (1) – Gráfico de tensão-deformação (preto) com reta de tendência do regime elástico (vermelho)

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Figura (2) – Gráfico de tensão-deformação amplificado no limite de escoamento

## Fontes

1. Site, <https://www.r-project.org/about.html>