

Universidade Eduardo Mondlane  
Faculdade de Ciências  
Departamento de Matemática e Informática  
Programação Estatística  
Ficha de exercícios- Visualização de dados

**Exercício 1: Customização de gráficos**

1. O arquivo `weight_chart.txt` contém dados para um gráfico de crescimento de um bebê durante os primeiros 9 meses de vida. Leia estes dados usando a função `read.delim`. Use a função `plot` para visualizar um gráfico de ponto e linha (`type="b"`) com as seguintes alterações. Aplique cada um dos pontos abaixo para ter o feito desejado.
  - Altere o caracter de ponto para um quadrado preenchido
  - Altere o tamanho do ponto no gráfico para 1.5x o tamanho normal (`cex=1.5`)
  - Altere a espessura da linha para o dobro do tamanho padrão
  - Altere o eixo y para uma escala entre 2 e 10 kg.
  - Altere o título do eixo y para Peso (kg)
  - Adicione um título adequado ao topo do gráfico
2. O ficheiro `feature_counts.txt` contém um resumo do número de características de diferentes tipos no genoma do mouse `GRCm38`. Leia os dados usando `read.delim` e, em seguida, faça um gráfico de barras usando a função `barplot`. Observe que os dados que você precisa para fazer o gráfico de barras estão contidos na coluna `Count` da data `frame`.

Depois de ter o gráfico inicial, faça as seguintes alterações.

- Coloque as barras na horizontal
- Dê um nome apropriado ao eixo x.
- As características devem ser adicionados ao eixo y.
- Dê um título apropriado ao gráfico.
- Coloque as `labels` das barras na horizontal (`las=1`). Observe passar este argumento para a função `par`.
- As margens devem ser ajustadas para acomodar os rótulos (`labels`) (parâmetro `par` `mar`). Você precisa fornecer um vetor de 4 elementos para os valores das margens inferior, esquerda, superior e direita. Observe o valor de `par()$mar` para ver quais são os valores padrão para saber por onde começar. Observe que você terá que re-visualizar o gráfico de barras depois de fazer as alterações com o `par`.

## Exerício 2: Aplicação de cores

1. O ficheiro `male_female_counts.txt` contém uma série temporal dividida em valores de contagem masculina e feminina.
  - Visualize estes dados usando um gráfico de barras.
  - Aplique cores diferentes para cada barra usando a função `rainbow`.
  - Refaça o gráfico e aplique cores diferentes para as barras dos homens e mulheres. Neste caso, as amostras masculinas e femininas se alternam, então você pode apenas passar um vetor de duas cores.
2. O ficheiro `up_down_expression.txt` contém um conjunto de dados de comparação de expressões, mas possui uma coluna extra que classifica as linhas em um dos 3 grupos (`up`, `down` ou `unchanging`). visualize o gráfico como um gráfico de dispersão (`plot`) com o `up` sendo vermelho, o `down` sendo azul e `unchanging` sendo cinza.
  - Leia os dados usando a função `read.delim`
  - Comece apenas plotando a coluna `Condition1` contra a coluna `Condition2` em um

gráfico.

- Passe a coluna `State` para o argumento `col` (`col=up.down$State` por exemplo). Isso definirá a cor de acordo com o estado de cada ponto, mas as cores serão definidas automaticamente a partir da saída da paleta. Execute a `palette()` para ver quais cores existem inicialmente e verifique se você pode ver como elas se relacionam com as cores obtidas no seu gráfico.
- Execute `levels()` nos dados da coluna `State` e combine isso com o que você viu na `palette()` para ver como cada cor foi selecionada. Descubra quais cores você precisaria colocar na paleta para obter a seleção de cores que realmente deseja.
- Use a função de `palette` para definir as cores correspondentes que deseja usar (por exemplo, `palette(c("red", "green", "blue"))`).

### Exercício 3: Sobreposição de gráficos

1. O ficheiro `cromossomo_position_data.txt` contém dados de contagem posicional para 3 conjuntos de dados diferentes (um WT e duas mutações). Plote isso como um gráfico de linhas (usando `plot(type="l")`) mostrando os 3 conjuntos de dados no mesmo gráfico.
  - Você precisará fazer uma plotagem inicial com `type="l"` especificado seguido por dois layers adicionais usando a função de `lines()`. Os valores x serão a coluna `Position` em cada caso. Os valores y serão a coluna `WT` para o gráfico inicial e, em seguida, as colunas `Mut1` e `Mut2` para as sobreposições
  - Lembre-se de calcular o intervalo completo de valores em todos os 3 conjuntos de dados ao fazer o gráfico inicial para que todos os dados caibam na área do gráfico. Você pode passar toda a parte de dados do quadro de dados (excluindo as posições) para o intervalo para fazer o cálculo, por exemplo: `range(chr_positions[,2:4])`
  - Coloque uma legenda no gráfico
2. O arquivo `brain_bodyweight.txt` contém dados para o cérebro `log10` e peso corporal

para uma variedade de espécies, juntamente com uma medida SEM para cada ponto. Plote esses dados em um gráfico de dispersão com barras de erro (error bar) mostrando a média  $\pm$  SEM e os nomes dos conjuntos de dados em cada ponto.

- Inicialmente, você precisará fazer um gráfico passando as colunas `Bodyweight` e `Brainweight` como dados. Certifique-se de que isso funcione antes de tentar adicionar barras de erro.
- Você precisará adicionar 2 conjuntos de barras de erro, um para o `Bodyweight` e outro para o `Brainweight`. Estas serão duas camadas adicionais separadas, cada uma consistindo em uma função de `arrows`.
- As posições inicial e final de uma barra de erro serão o valor original mais e menos o SEM correspondente.
- Para adicionar os nomes das espécies logo abaixo de cada ponto de dados, você usaria a função de `text`. Os valores x e y serão os mesmos do gráfico original (`Brainweight` e `Bodyweight`).