

MATLAB Avançado

Aula 3

Melissa Weber Mendonça
`melissa.mendonca@ufsc.br`

Relembrando...

```
>> inteiro = fscanf(arquivo, '%d')
```

Repita o exercício da aula anterior (ler UM dado de um arquivo), agora com um número real:

```
>> real = fscanf(arquivo, '%f')
```

e depois repita com um texto:

```
>> texto = fscanf(arquivo, '%s')
```

Tente colocar o formato errado e observe o erro produzido.

Lista de dados

Se quisermos ler uma lista de números inteiros, por exemplo, basta informar o *padrão* dos dados.

Exemplo: se no arquivo temos

1 2 3 4 5

podemos usar o comando

```
v = fscanf(arquivo,'%d')
```

v será um vetor *coluna*.

Exercício

Ler uma matriz 3×3 de dados de um arquivo .txt

Cuidado com a ordem dos dados!

matriz.txt

```
1 2 3  
4 5 6  
7 8 9
```

```
arquivo = fopen('matriz.txt');  
A = fscanf(arquivo,'%f');  
fclose(arquivo);  
A
```

Ler uma matriz

Por padrão, o MATLAB lê os dados em um vetor. Se quisermos especificar o tamanho da saída dos dados, devemos acrescentar um argumento à função **fscanf**:

```
A = fscanf(arquivo,'%f',[3 3])
```

```
arquivo = fopen('matriz.txt');  
A = fscanf(arquivo,'%f',[3 3]);  
fclose(arquivo);  
% A verdadeira matriz eh a transposta  
% da matriz que foi lida:  
A = A';  
A
```

Ler uma matriz desconhecida

Se não sabemos o tamanho da matriz que está no arquivo, não podemos informar seu formato. Mas podemos contar quantos elementos foram lidos do arquivo:

```
[A,contador] = fscanf(arquivo,'%d')
```

Formatos avançados

Suponha que temos no nosso arquivo também o nome do campo de dados:

`temperaturas.txt`

Hora	1,	Temperatura	20.6
Hora	3,	Temperatura	21.2
Hora	5,	Temperatura	23.1
Hora	6,	Temperatura	24.5
Hora	8,	Temperatura	25.0
Hora	9,	Temperatura	25.2
Hora	10,	Temperatura	25.8

Para ler apenas os números desta tabela, usamos o comando

```
A = fscanf(arquivo, 'Hora %d, Temperatura %f\n', [2 7])
```

Exemplo

Se não conhecemos quantos dados estão na lista (mas sabemos que são 2 por linha), podemos especificar

```
A = fscanf(arquivo, 'Hora %d, Temperatura %f\n', [2 Inf])
```


Exercício

Dado um arquivo com a tabela abaixo de nomes e idades, calcule a média de idades deste grupo.

Antonio	12
Bruno	20
Caio	34
Danilo	21
Eder	45
Fernando	78
Gustavo	20

Dica: para não ler o texto, e ler apenas as idades, podemos *pular* o campo de texto com o comando

```
idades = fscanf(arquivo,'%*s %d\n')
```

Exercícios: resposta

```
arquivo = fopen('tabela.txt');  
idades = fscanf(arquivo,'%*s %d\n');  
fclose(arquivo);  
% A funcao mean(vetor) calcula a media dos  
% valores do vetor.  
media = mean(idades);  
disp(['A media de idades do grupo eh ' ...  
      num2str(media) ' anos.'])
```

xlsread

Para ler arquivos de planilha gerados pelo Microsoft Excel, usamos

```
>> [dados,texto,resto] = xlsread(arquivo)
```

Para ler os dados de uma planilha específica do arquivo, usamos

```
>> [dados,texto,resto] = xlsread(arquivo,planilha)
```

Em sistemas com o Microsoft Excel instalado, pode-se usar

```
>> [dados,texto,resto] = xlsread(arquivo,-1)
```

para abrir uma janela do Excel e selecionar os dados a serem importados interativamente.

Escrita em arquivos: save

Para salvarmos alguma variável em um arquivo, podemos usar o comando

```
>> save('arquivo.mat','variavel')
```

Porém, este comando salva o arquivo no formato MAT, que é um formato próprio do MATLAB, ilegível para humanos. Assim, para salvarmos em um arquivo texto simples, acrescentamos a opção '-ascii'.

Exemplo:

```
>> dados = rand(3,4);  
>> save('dadosout.txt','dados','-ascii')
```

Escrita

Para escrever em um arquivo existente, a sintaxe é similar, mas devemos avisar ao MATLAB que vamos escrever neste arquivo:

```
>> arquivo = fopen('info.txt','w')
>> fprintf(arquivo,'%d',1)
>> fclose(arquivo)
```

Obs. As opções do comando `fopen` são:

```
>> arquivo = fopen('info.txt','r')
>> arquivo = fopen('info.txt','w')
>> arquivo = fopen('info.txt','a')
```

Métodos para Análise Estatística

Média aritmética simples

Podemos calcular o valor da média aritmética simples para um conjunto de números armazenados em um vetor `x` usando o comando

```
>> mean(x)
```

Para calcularmos a média aritmética simples **de cada coluna** de uma matriz e armazenarmos essas médias em um vetor linha, podemos usar o comando

```
>> mean(matriz)
```

Para calcularmos a média aritmética simples **de cada linha** de uma matriz e armazenarmos o resultado em uma matriz coluna, usamos o comando

```
>> mean(matriz,2)
```

trimmmean

O comando

```
>> m = trimmmean(X,pc)
```

calcula a média excluindo os k maiores e menores valores de um vetor X , em que

$$k = \frac{n * \frac{pc}{100}}{2}$$

e onde n é o número de valores em X .

Mediana

Para calcularmos a mediana de um conjunto de dados armazenados em um vetor, usamos o comando

```
>> median(x)
```

Para calcularmos a mediana das colunas de uma matriz, e retornar as medianas em um vetor linha, usamos o comando

```
>> median(matriz)
```

Desvio Padrão

Existem duas definições para o desvio padrão:

$$s = \left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \quad (1)$$

ou

$$s = \left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \quad (2)$$

onde

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Desvio Padrão

Para calcularmos o desvio padrão usando a fórmula (1), usamos o comando

```
>> std(x)
```

O resultado é a raiz quadrada da variância.

Se quisermos calcular um vetor linha contendo o desvio padrão calculado para cada coluna de uma matriz, usamos

```
>> std(matriz)
```

Se quisermos calcular o desvio padrão dos elementos de um vetor usando a fórmula (2), usamos

```
>> std(x,1)
```

Variância

Para calcularmos a variância dos elementos de um vetor, usamos o comando

```
>> var(x)
```

Para calcularmos um vetor linha com as variâncias de cada coluna da matriz, usamos o comando

```
>> var(matriz)
```

O comando **var** normaliza os dados por $n - 1$, se temos $n > 1$ dados. Se desejamos normalizar por n , usamos o comando

```
>> var(x,1)
```

Covariância

Para calcularmos a matriz de covariância entre 2 variáveis de dados, usamos o comando

```
>> cov(X)
```

Podemos ainda obter outras informações desta matriz:

```
>> diag(cov(X))
```

é o vetor de variâncias para cada coluna de dados (idem a **var**)

```
>> sqrt(diag(cov(X)))
```

é desvio padrão (idem a **std**).

X pode ser um vetor ou uma matriz. Para uma matriz $m \times n$, a matriz de covariância é $n \times n$.

Coeficientes de Correlação

Se tivermos uma matriz em que cada coluna contém observações de uma variável, podemos calcular os coeficientes de correlação entre as variáveis desta matriz usando o comando

```
>> R = corrccoef(X)
```

Os coeficientes vão de -1 (correlação negativa) até 1 (correlação positiva). Valores próximos de 0 indicam que não há correlação linear entre as variáveis.

Se também quisermos saber o *p-value* de cada correlação, usamos o comando

```
>> [R, P] = corrccoef(X)
```

Exemplo

Calcular a matriz de correlação e os *p-values* entre as colunas da matriz X:

```
>> [R,P] = corrcoef(X)
```

Encontrar todos os índices da matriz de correlação para os quais o *p-value* é menor que 0.05:

```
>> [i,j] = find(p<0.05)
```

Correlação (2)

Para encontrarmos a matriz de correlação entre variáveis e seus respectivos *p-values*, também podemos usar a função **corr**, com mais opções:

```
>> [RHO,PVAL] = corr(X,Y,'nome',valor)
```

Exemplos:

```
>> [RHO,PVAL] = corr(X,Y,'type','Pearson')
```

```
>> [RHO,PVAL] = corr(X,Y,'type','Kendall')
```

```
>> [RHO,PVAL] = corr(X,Y,'rows','all')
```

```
>> [RHO,PVAL] = corr(X,Y,'rows','complete') : pula  
linhas com NaN!
```


Histograma

Um histograma pode ser criado com o comando

```
>> n = hist(Y)
```

em que o vetor Y é distribuído em 10 caixas igualmente espaçadas, e n é o número de elementos em cada caixa. O comando

```
>> n = hist(Y,nbins)
```

divide os dados em nbins caixas.

Box Plot

O comando

```
>> boxplot(X)
```

cria um gráfico de caixas dos dados em X. Se X for uma matriz, existirá uma caixa por coluna; se X for um vetor, existirá apenas uma caixa.

Em cada caixa:

- ▶ a marca central é a mediana;
- ▶ os limites da caixa representam o 25° e o 75° percentil;
- ▶ os marcadores externos sinalizam os pontos extremos dos dados (sem considerar *outliers*;
- ▶ os *outliers*, se existirem, serão marcados individualmente no gráfico.