

Computação Científica com MATLAB

Melissa Weber Mendonça
melissa.mendonca@ufsc.br

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.

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)
```

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 = corrcoef(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] = corrcoef(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!
```

Regressão

Podemos calcular automaticamente um modelo de regressão (usando quadrados mínimos) através da janela de um gráfico. Exemplo:

```
>> load census  
>> plot(cdate, pop, 'ro')
```

Na janela do gráfico, podemos selecionar

Tools → Basic Fitting

Norma dos resíduos

Podemos calcular a norma dos resíduos para um *fit* realizado através do comando

```
>> sqrt(sum(resids.^2))
```

Podemos também extrapolar dados usando a interface gráfica do MATLAB, novamente em

Tools → Basic Fitting

Finalmente, podemos usar o comando

File → Generate Code

para criarmos uma função que reproduz o gráfico obtido.

Interpolação polinomial: `polyfit`

O comando

```
>> p = polyfit(x,y,n)
```

encontra os coeficientes do polinômio $p(x)$ de grau n que aproxima os dados $y(i) = p(x(i))$, em um sentido de mínimos quadrados. O vetor p resultante contém os coeficientes do polinômio em ordem decendente de potências. O comando

```
>> [p,S] = polyfit(x,y,n)
```

retorna os coeficientes do polinômio em p e uma estrutura S que pode ser usada com o comando `polyval`.

A estrutura S contém os campos `R`, `df` e `normr`.

Se os dados y são aleatórios, uma estimativa da covariância de p é

```
>> (inv(R)*inv(R)')*normr^2/df
```

Avaliação de polinômios: `polyval`

O comando

```
>> y = polyval(p,x)
```

retorna o valor de um polinômio de grau n (armazenado no vetor `p`) em `x`. O comando

```
>> [y,delta] = polyval(p,x,S)
```

usa a estrutura `S` gerada pelo comando `polyfit` para gerar `delta`, que é uma estimativa do desvio padrão do erro obtido ao se tentar calcular $p(x)$.

Ajuste polinomial

O comando

```
>> polytool(x,y)
```

ajusta uma reta aos vetores x e y e mostra um gráfico interativo do resultado.

```
>> polytool(x,y,n)
```

ajusta um polinômio de grau n aos dados.

Só disponível na Statistics Toolbox!

Curve Fitting Toolbox

Para fazermos o ajuste de curvas de maneira interativa, podemos usar o comando

```
>> cftool
```

Só disponível com a Curve Fitting Toolbox