

Introdução à Programação

Aula 15: Módulos 'random' e 'statistics'

Prof. Eduardo Corrêa Gonçalves

04/06/2024

Introdução

- A biblioteca padrão do Python (módulos que vem junto com a linguagem) é composta por mais de 200 módulos.
 - Há desde módulos para acessar o conteúdo de páginas Web até módulos que permitem a programação paralela.
- Há 3 módulos especialmente interessantes para os estatísticos:
 - `'math'` (já apresentado no início do curso)
 - `'random'`
 - `'statistics'`
- Nessa aula, faremos uma breve apresentação dos módulos `'random'` e `'statistics'`

Módulo 'random'

- O módulo 'random' oferece recursos para a geração de números aleatórios.



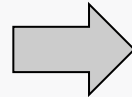
- O módulo pertence à biblioteca padrão do Python. Ele pode ser importado da seguinte maneira:
 - `import random`

Funções do módulo 'random' (1/5)

- Função **random()**
 - É a função mais geral do módulo (**não confunda** com o nome do módulo!)
 - Produz um número real x no intervalo $[0,1)$, isto é: $0.0 \leq x < 1.0$
 - O exemplo a seguir irá gerar 5 números aleatórios.
 - Como **não** foi estabelecida uma **semente**, os resultados são imprevisíveis...
 - Ao rodar no seu computador os valores gerados serão diferentes dos que estão apresentados abaixo.

```
import random
```

```
for i in range(5):  
    print(random.random())
```



```
0.15257583151155107  
0.5054613136195354  
0.36943166367880187  
0.18748915998230642  
0.9785997946304678
```

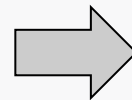
Funções do módulo 'random' (2/5)

- Função **seed()**
 - Permite definir uma **semente** para o gerador de aleatórios.
 - Assim, a mesma sequência será gerada toda vez que você executar o programa.
 - *Exemplo de aplicação: Em um jogo, serve para garantir que uma mesma fase sempre aconteça da mesma forma (mesmos inimigos, obstáculos nos mesmos locais, etc.)*
 - **Sintaxe:** **random.seed(semente)**
 - Onde a semente é um número inteiro.
 - Ao rodar o programa abaixo em sua máquina, os mesmos números apresentados serão gerados.

```
import random
```

```
random.seed(0)
```

```
for i in range(5):
    print(random.random())
```



```
0.8444218515250481
0.7579544029403025
0.420571580830845
0.25891675029296335
0.5112747213686085
```

Funções do módulo 'random' (3/5)

- Funções **randrange()** e **randint()**
 - Se você deseja números aleatórios inteiros, poderá usar as funções acima.
 - A função **randrange()** serve para gerar um inteiro dentro de uma faixa de valores
 - A faixa de valores é definida do mesmo jeito que na função **range()**

random.randrange(5) # gera um aleatório entre 0 e 4

random.randrange(1, 10) # gera um aleatório entre 1 e 9

random.randrange(0, 11, 2) # gera um aleatório pertencente ao
conjunto {0, 2, 4, 6, 8, 10}

Funções do módulo 'random' (4/5)

- Funções **randrange()** e **randint()**
 - Se você deseja números aleatórios inteiros, poderá usar as funções acima.
 - A função **randint()** é equivalente à `randrange(inicio, fim+1)`.
 - Ou seja, ela gera um inteiro aleatório na faixa [início, fim]
 - Os dois parâmetros, *início* e *fim*, são obrigatórios

random.randint(0, 5) # gera um aleatório entre 0 e 5

random.randint(1, 10) # gera um aleatório entre 1 e 10

Funções do módulo 'random' (5/5)

- Funções **choice()** e **sample()**

- A função **choice(seq)** escolhe aleatoriamente um elemento de uma sequência *seq* passada como entrada.

Exemplo: retorna um dos 4 números da lista [0, 50, 100, 1000]

random.choice([0, 50, 100, 1000])

- A função **sample(seq, n)** cria uma lista contendo *n* elementos selecionados aleatoriamente da sequência *seq*.

Exemplo: retorna uma lista com 2 dentre os 4 números
da lista [0, 50, 100, 1000]

random.sample([0, 50, 100, 1000], 2)

Módulo 'statistics'

- Fornece funções para cálculos estatísticos básicos sobre conjuntos de dados.
- Os dados a serem processados devem ser especificados em **uma lista**
 - O módulo pertence à biblioteca padrão do Python. Ele pode ser importado da seguinte maneira:
 - `import statistics`

Funções do módulo ‘statistics’ (1/3)

- Há funções para o cálculo das medidas de tendência central e de variabilidade. Algumas são apresentadas no quadro a seguir.
 - Considere que “lst” é um conjunto de dados representado em uma lista.

statistics.mean(lst): média dos valores de lst

statistics.median(lst): mediana de lst (os valores não precisam estar ordenados)

statistics.harmonic_mean(lst): média harmônica;

statistics.mode(lst): moda de uma lista de valores discretos ou categóricos (a moda é retornada em uma lista, pois os dados de entrada podem ser bimodais ou multimodais). *Não funciona no lab, pois a versão do Python não é a mais recente.*

statistics.median_low(lst): quando lst possui um número par de valores, retorna o menor dos dois valores que seriam usados para computar a mediana. Exemplo: para [1,2,3,4], retorna 2;

statistics.median_high(lst): quando lst possui um número par de valores, retorna o maior dos dois valores que seriam usados para computar a mediana. Exemplo: para [1,2,3,4], retorna 3;

statistics.stdev(lst): desvio padrão de amostra;

statistics.pstdev(lst): desvio padrão de população;

statistics.variance(lst): variância da amostra;

statistics.pvariance(lst): variância da população;

Funções do módulo 'statistics' (2/3)

- **Exemplo:** Considere um site sobre cinema que armazena as notas do público para 7 diferentes filmes do diretor Milos Forman.
- Utilizando o módulo 'statistics', compute estatísticas básicas para as notas.

Filme	Nota do Público (média)
Um Estranho no Ninho (1975)	8.7
Hair (1979)	7.6
Na Época do Ragtime (1981)	7.3
Amadeus (1984)	8.3
Valmont - Uma História de Seduções (1989)	7.0
O Povo Contra Larry Flint (1996)	7.3
O Mundo de Andy (1999)	7.4

Funções do módulo 'statistics' (3/3)

- Resolução:

```
import statistics

nomes_filmes = ['Um Estranho no Ninho (1975)', 'Hair (1979)',
'Na Época do Ragtime (1981)', 'Amadeus (1984)',
'Valmont - Uma História de Seduções (1989)', 'O Povo Contra Larry
Flint (1996)', 'O Mundo de Andy (1999)',
]

avaliacao_filmes = [8.7, 7.6, 7.3, 8.3, 7.0, 7.3, 7.4]

print('média = ', statistics.mean(avaliacao_filmes))
print('mediana = ', statistics.median(avaliacao_filmes))
print('variância = ', statistics.variance(avaliacao_filmes))
print('desvio padrão = ', statistics.stdev(avaliacao_filmes))
```

Saída:

```
>>>
média = 7.6571428571428575
mediana = 7.4
variância = 0.37619047619047613
desvio padrão = 0.6133436852128471
```

Exercícios

(1) Em uma avaliação de autoritarismo (escores mais altos refletem tendência para preconceito, etnocentrismo e submissão a autoridade), sete estudantes alcançaram o seguinte resultado*:

1 6 6 3 7 4 10

Armazene estes escores em uma lista e obtenha a média, mediana, variância e desvio padrão para esses escores usando o módulo `'statistics'`.

(2) Construa o seguinte jogo: o computador deve sortear um número entre 1 e 15 - use o módulo `'random'` - e armazena-lo em uma variável (sem mostrar para o usuário).

Em seguida ele deve pedir para o usuário adivinhar o número sorteado. O computador deve dar 3 chances para o usuário acertar. Se o usuário errar na primeira e na segunda chance, o computador deve dar uma dica para o usuário. Essa dica consiste em indicar se o número digitado pelo usuário é maior ou menor do que o número que foi sorteado pelo computador.

* Exemplo retirado de: Levin, J. e Fox, J. A. "Estatísticas para Ciências Sociais". 9ª edição, Pearson, 2004