

Profissão: Analista de dados

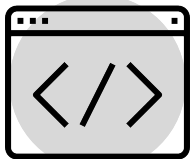


DE OLHO NO CÓDIGO



Fundamentos de Estatística

 Aprenda
média e
variância



Confira boas práticas da
linguagem Python por assunto
relacionado às aulas.



Aprenda média e variância

Média

- **Verifique os dados**
Antes de calcular a média, certifique-se de que os dados estejam no formato correto. Por exemplo, se os dados estiverem em uma lista, certifique-se de que todos os elementos da lista sejam números.

- **Verifique se a média faz sentido**
Sempre verifique se a média faz sentido em relação aos dados. Por exemplo, se você estiver lidando com uma lista de temperaturas em graus Celsius, verifique se a média faz sentido em relação aos valores esperados para a escala Celsius.



Aprenda média e variância

Média

• Use a função "mean"

O módulo statistics do Python possui a função "mean" para calcular a média de um conjunto de dados. Para usá-la, importe o módulo e chame a função passando a lista de dados como argumento. Exemplo:

```
import statistics
dados = [2, 4, 6, 8, 10]
media = statistics.mean(dados)
print(media)
```

Este código irá imprimir a média dos valores na lista, que é 6.



Aprenda média e variância

Média

● Considere usar o NumPy

Se você estiver lidando com grandes conjuntos de dados ou precisar realizar outras operações matemáticas além do cálculo da média, considere usar a biblioteca NumPy. Ela possui uma função "mean" que é otimizada para lidar com arrays grandes e pode ser mais rápida do que a função "mean" do módulo statistics. Por exemplo:

```
import numpy as np
dados = [2, 4, 6, 8, 10]
media = np.mean(dados)
print(media)
```

Este código também irá imprimir a média dos valores na lista, que é 6.



Aprenda média e variância

Média

● **Lide com dados ausentes**

Se seus dados contiverem valores ausentes (como None ou NaN), você deve lidar com eles antes de calcular a média. Você pode remover esses valores da lista ou substituí-los por um valor padrão. Por exemplo:

```
import statistics

dados = [2, 4, None, 8, 10]
dados_limpos = [x for x in dados if x is not None] #
remove valores None
media = statistics.mean(dados_limpos)
print(media)
```

Este código irá remover os valores None da lista antes de calcular a média.



Aprenda média e variância

Média

• Use a média com cuidado

A média é uma métrica útil, mas pode não ser a melhor escolha para todos os conjuntos de dados. Considere outras medidas de centralidade, como a mediana, se a distribuição dos seus dados não for simétrica ou se houver valores discrepantes (outliers).



Aprenda média e variância

Variância

• Use a função "var"

O módulo statistics do Python possui a função "var" para calcular a variância de um conjunto de dados. Para usá-la, importe o módulo e chame a função passando a lista de dados como argumento. Por exemplo:

```
import statistics

dados = [2, 4, 6, 8, 10]
variância = statistics.variance(dados)
print(variância)
```

Este código irá imprimir a variância dos valores na lista, que é 10.



Aprenda média e variância

Variância

● Considere usar o NumPy

Se você estiver lidando com grandes conjuntos de dados ou precisar realizar outras operações matemáticas além do cálculo da variância, considere usar a biblioteca NumPy. Ela possui uma função "var" que é otimizada para lidar com arrays grandes e pode ser mais rápida do que a função "var" do módulo statistics. Por exemplo:

```
import numpy as np

dados = [2, 4, 6, 8, 10]
variancia = np.var(dados)
print(variancia)
```

Este código também irá imprimir a variância dos valores na lista, que é 10.



Aprenda média e variância

Variância

● Lide com dados ausentes

Se seus dados contiverem valores ausentes (como None ou NaN), você deve lidar com eles antes de calcular a variância. Você pode remover esses valores da lista ou substituí-los por um valor padrão. Por exemplo:

```
import statistics

dados = [2, 4, None, 8, 10]
dados_limpos = [x for x in dados if x is not None] #
remove valores None
variancia = statistics.variance(dados_limpos)
print(variancia)
```

Este código irá remover os valores None da lista antes de calcular a variância.



Aprenda média e variância

O **desvio padrão** é uma medida de dispersão estatística que indica o quão distantes os valores de um conjunto de dados estão da média. Em Python, é possível calcular o desvio padrão utilizando a biblioteca NumPy ou a biblioteca statistics. Acompanhe algumas dicas para calcular o desvio padrão em Python.



● Use a função "std"

A biblioteca NumPy possui a função "std" que calcula o desvio padrão de um conjunto de dados. Para usá-la, importe a biblioteca e chame a função passando a lista de dados como argumento. Por exemplo:

```
import numpy as np

dados = [2, 4, 6, 8, 10]
desvio_padrao = np.std(dados)
print(desvio_padrao)
```

Este código irá imprimir o desvio padrão dos valores na lista, que é aproximadamente 2,83.

Aprenda média e variância

O **desvio padrão** é uma medida de dispersão estatística que indica o quão distantes os valores de um conjunto de dados estão da média. Em Python, é possível calcular o desvio padrão utilizando a biblioteca NumPy ou a biblioteca statistics. Acompanhe algumas dicas para calcular o desvio padrão em Python.



● Use a função "stdev"

A biblioteca statistics também possui uma função "stdev" que calcula o desvio padrão de um conjunto de dados. Para usá-la, importe a biblioteca e chame a função passando a lista de dados como argumento. Por exemplo:

```
import statistics

dados = [2, 4, 6, 8, 10]

desvio_padrao = statistics.stdev(dados)

print(desvio_padrao)
```

Este código também irá imprimir o desvio padrão dos valores na lista, que é aproximadamente 2,83.

Aprenda média e variância

O **desvio padrão** é uma medida de dispersão estatística que indica o quão distantes os valores de um conjunto de dados estão da média. Em Python, é possível calcular o desvio padrão utilizando a biblioteca NumPy ou a biblioteca statistics. Acompanhe algumas dicas para calcular o desvio padrão em Python.



● Lide com dados ausentes

Se seus dados contiverem valores ausentes (como None ou NaN), você deve lidar com eles antes de calcular o desvio padrão. Você pode remover esses valores da lista ou substituí-los por um valor padrão.

```
import statistics

dados = [2, 4, None, 8, 10]
dados_limpos = [x for x in dados if x is not None] #
remove valores None
desvio_padrao = statistics.stdev(dados_limpos)
print(desvio_padrao)
```

Este código irá remover os valores None da lista antes de calcular o desvio padrão.

Aprenda média e variância

O **desvio padrão** é uma medida de dispersão estatística que indica o quão distantes os valores de um conjunto de dados estão da média. Em Python, é possível calcular o desvio padrão utilizando a biblioteca NumPy ou a biblioteca statistics. Acompanhe algumas dicas para calcular o desvio padrão em Python.



● Considere usar outras medidas de dispersão

O desvio padrão é uma métrica útil, mas pode ser sensível a valores discrepantes (outliers) e pode não ser a melhor escolha para todos os conjuntos de dados. Considere outras medidas de dispersão, como a amplitude interquartil (IQR), se a distribuição dos seus dados não for simétrica ou se houver valores discrepantes.

Bons estudos!

