**06 - Aplicando p-valor**

[0:00] Realizamos o nosso teste Qui-quadrado, que é um teste de adequação, um teste não paramétrico, e chegamos à conclusão de que rejeitamos a hipótese nula de que as frequências observadas e esperadas são iguais.

[0:10] Ou seja, elas são diferentes, então a sua moeda é viciada e precisa ser substituída. O que eu vou mostrar agora é a forma de calcular p valor. Nesse caso será um pouco diferente dos que calculamos anteriormente, e se a gente for fazer manualmente, precisaremos disso.

[0:28] A ferramenta que eu vou mostrar logo depois, que é a forma mais simples de realizar esse teste, a gente não precisa se preocupar com isso, ele já vai colocar o nosso p valor corretamente, sem nenhuma preocupação.

[0:37] Vamos chamar o chi\_2. É a estatística de teste que obtivemos, X² = 5,12. E é com ele que eu obtenho o p valor, como nos outros testes, com a estatística de teste.

[0:57] Estamos tratando de uma variável estatística padronizada elevada ao quadrado que obtivemos a partir da função chi do Scipy.

[1:06] Ou seja, para eu obter o meu p valor, eu vou ter que voltar para o chi, que não vai ser mais Qui-quadrado, eu vou tirar esse quadrado, vou passar uma raiz quadrada nesse número e ele voltará a ser Qui.

[1:18] E aí sim eu aplico a mesma função que a gente vem aplicando nos outros testes, que é o SF. O normal, o t de Student, é o mesmo critério.

[1:29] Para fazer isso, eu vou chamar de raiz\_chi\_2. Não vou chamar de chi, porque senão vai gerar um conflito com a função chi que a gente já importou do Scipy.

[1:39] Que vai ser igual a np.sqrt, que é a função para extrair a raiz quadrada de um número, e eu passo somente o qui quadrado.

raiz\_chi\_2 = np.sqrt(chi\_2)

raiz\_chi\_2

2.262741699796952

COPIAR CÓDIGO

[1:51] Feito isso, já temos um número para obter o p valor. Daqui para frente é a mesma coisa.

raiz\_chi\_2 = np.sqrt(chi\_2)

raiz\_chi\_2

2.262741699796952

p\_valor = chi.sf(raiz\_chi\_2, df=1)

p\_valor

0.023651616655355978

COPIAR CÓDIGO

[2:00] Então vai ser o chi.sf, a mesma função. Dentro eu passo a raiz de chi e também, nesse caso, como na t de Student, os graus de liberdade com o parâmetro df.

[2:17] Na linha seguinte, p\_valor, que é igual a 0.0236. Mais abaixo eu já fiz um teste, p\_valor <= significancia. Lembrando que o p valor menor ou igual a significância indica que a gente tem que rejeitar a hipótese nula.

[2:31] Está nas nossas anotações: valor p = rejeitar se p valor é menor ou igual a alfa. Vamos executar, pois já temos o p valor e a significância.

p\_valor <= significancia

True

COPIAR CÓDIGO

[2:39] True, ou seja, rejeitamos H0, a mesma conclusão que a gente chegou fazendo com o critério do valor crítico.

[2:49] Agora, a forma mais simples de calcular tudo isso, de executar esse teste com uma linha de código, é a seguinte: from scipy.stats import chisquare. Repare que aqui já veio a função Qui-quadrado, chisquare.

[3:08] Eu rodo isso. Acessando [este link](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chisquare.html), você encontra a documentação da funcionalidade dentro do Scipy.

[3:15] Nessa documentação encontramos o f\_obs, que é o parâmetro que vai fornecer a frequência observada. Também temos o f\_exp, que é a frequência esperada, e por aí vai.

[3:27] Vamos executar o teste. Chamaremos o chisquare() e passaremos para ele esses dois parâmetros: f\_obs é igual a frequência observada, ou seja, f\_obs=F\_Observada. Esse valor nós já temos, está mais acima no código. O outro parâmetro é o f\_exp, que é a frequência esperada, f\_exp=F\_Esperada.

chisquare(f\_obs=F\_Observada, f\_exp=F\_Esperada)

Power\_divergenceResult(statistic=5.12, pvalue=0.023651616655356)

COPIAR CÓDIGO

[3:56] Feito isso, é só executar. Está ali a nossa estatística: 5,12, igual ao que a gente calculou, e o nosso p valor é pvalue=0.023651616655356, que a gente pode comparar com esse trecho que está mais acima.

raiz\_chi\_2 = np.sqrt(chi\_2)

raiz\_chi\_2

2.262741699796952

p\_valor = chi.sf(raiz\_chi\_2, df=1)

p\_valor

0.023651616655355978

COPIAR CÓDIGO

[4:06] Nunca dá exatamente o mesmo valor por causa de algum problema de arredondamento e a forma de obtenção dos valores, mas a gente vê que é um valor bem próximo.

[4:18] Como nos outros vídeos, podemos chamar chisquare(f\_obs=F\_Observada, f\_exp=F\_Esperada) de chi\_2 vírgula p\_valor e a gente pode fazer um print() dos dois.

chi\_2, p\_valor = chisquare(f\_obs=F\_Observada, f\_exp=F\_Esperada)

print(chi\_2)

print(p\_valor)

5.12

0.023651616655356

COPIAR CÓDIGO

[4:35] Isso aqui estava vindo como uma estatística, então vai ser chisquare mesmo, e o p valor também. Print p valor.

[4:53] E ele já disponibiliza e separa as variáveis para podermos executar coisas desse tipo aqui embaixo no código, e verificar tudo corretamente.

[5:03] Era isso que eu queria mostrar, pessoal. A gente já realizou o primeiro teste não paramétrico, agora vamos estudar mais dois. O próximo é o teste de Wilcoxon. Beleza? Até lá.