**02 - Teste de Wilcoxon**

[0:00] Mais um teste não paramétrico. Agora vamos conhecer o teste de Wilcoxon, que é um teste de comparação de populações, mas quando as amostras são dependentes.

[0:10] Vamos começar com um probleminha para que você já compreenda essa questão da dependência.

[0:14] "Um novo tratamento para acabar com o hábito de fumar está sendo empregado em um grupo de 35 pacientes voluntários."

[0:19] "De cada paciente testado foram obtidas as informações de quantidade de cigarros consumidos antes e depois do termino do tratamento."

[0:28] "Assumindo um nível de confiança de 95% é possível concluir que depois da aplicação do tratamento houve uma mudança no hábito de fumar no grupo de pacientes testados?"

[0:37] Essa é a questão da dependência, verificamos informações de um grupo de pessoas, de um grupo que a gente está testando, aplicamos determinado tratamento ou determinado evento acontece e depois verificamos esse mesmo grupo, que efeito teve esse tratamento, essa aplicação de evento. Por exemplo, uma campanha de marketing ou semelhantes. Podemos testar a eficiência através de testes como o de Wilcoxon.

[1:08] Então, vamos lá. Nesse teste, vamos seguir aqueles passos, mas eu não vou digitar muita coisa, porque senão a gente vai ter um vídeo muito longo.

[1:16] Então, eu já deixei preparado. Fique à vontade, faça no seu tempo, pode parar o vídeo, se quiser digitar, digite, não tem problema, mas é só para a gente fluir bem.

[1:28] Os passos são praticamente os mesmos, só que é um pouco mais trabalhoso para fazer manualmente. Vamos lá. Aqui estão os dados do problema.

fumo = {

'Antes': [39, 25, 24, 50, 13, 52, 21, 29, 10, 22, 50, 15, 36, 39, 52, 48, 24, 15, 40, 41, 17, 12, 21, 49, 14, 55, 46, 22, 28, 23, 37, 17, 31, 49, 49],

'Depois': [16, 8, 12, 0, 14, 16, 13, 12, 19, 17, 17, 2, 15, 10, 20, 13, 0, 4, 16, 18, 16, 16, 9, 9, 18, 4, 17, 0, 11, 14, 0, 19, 2, 9, 6]

}

significancia = 0.05

confianca = 1 - significancia

n = 35

COPIAR CÓDIGO

[1:34] Eu tenho aqui fumo, coloquei um JSON, um dicionário do Python, com 'Antes', que são as informações da quantidade de cigarros antes do tratamento. E 'Depois', que são as informações desses mesmos pacientes, de quantidade de cigarros que eles fumavam depois do tratamento. O nível de significância é de 5%. Confiança é 1 menos significância. O N é 35.

[1:57] Já foi dado tudo isso no problema. Eu estou colocando o fumo dentro de um Dataframe.

fumo = pd.DataFrame(fumo)

fumo.head()

COPIAR CÓDIGO

[2:05] Mostrei só os cinco primeiros.

| **Antes** | **Depois** |  |
| --- | --- | --- |
| 0 | 39 | 16 |
| 1 | 25 | 8 |
| 2 | 24 | 12 |
| 3 | 50 | 0 |
| 4 | 13 | 14 |

Então temos o "Antes" e o "Depois". Eu vou calcular a média do grupo antes, média de cigarros fumados do grupo antes e depois.

media\_antes = fumo.Antes.mean()

media\_antes

COPIAR CÓDIGO

[2:16] Eu já vejo uma diferença aqui. Mas, lembre-se, quando estamos lidando com testes não paramétricos, não mexemos nos parâmetros da nossa população.

[2:22] Então, isso é só para tirarmos uma conclusão final. Nesse tipo de teste, a hipótese nula é sempre assim: "não há diferença entre os grupos".

[2:37] A contra-hipótese alternativa, que é o contrário, de que existe uma diferença entre os grupos. Podemos modelá-la da seguinte maneira, que é o passo que a gente vai fazer agora, formulação das hipóteses.

[2:49] Na hipótese nula escrevemos que a média antes é igual a média depois, ou seja, não existe diferença entre esses grupos, o tratamento não surtiu efeito nenhum.

[3:00] E o outro caso, como eu tenho a média para comparar, eu posso colocar essa hipótese alternativa como se a média antes, de cigarros fumados, é maior que a média depois, ou seja, o meu tratamento foi eficiente.

[3:12] Olhando essa média aqui, eu posso tirar essa conclusão, mas vamos ver estatisticamente, utilizando um teste para resolver esse problema.

media\_antes = fumo.Antes.mean()

media\_antes

31.857142857142858

media\_depois = fumo.Depois.mean()

media\_depois

11.2

COPIAR CÓDIGO

[3:20] Escolha da distribuição adequada. Agora a gente tem que prestar atenção para não confundir isso daqui com um teste paramétrico.

[3:25] A gente vai utilizar também a normal e a t, simplesmente porque a estatística de teste, a partir de determinado tamanho de n, se aproxima de uma normal.

[3:37] Então, não tem nada a ver com a nossa população ser distribuída como uma normal, isso é estatística de teste.

[3:46] Então, no caso, alguns livros dizem dez, alguns livros dizem 20, eu deixei aqui o 20.

[3:53] Quando o n for maior que 20, a gente pode utilizar nossa tabela normal, a z, como se fosse um Ztest, mas não é.

[4:02] Se for menor que isso, a gente usa o t de Student para fazer a comparação, criar as áreas de aceitação e rejeição. Então, como o n é maior que 20, sim, a gente vai usar a tabela normal.

[4:14] A seguir, já temos o procedimento para obter o z alfa sobre dois.

probabilidade = (0.5 + (confianca / 2))

probabilidade

z\_alpha\_2 = norm.ppf(probabilidade)

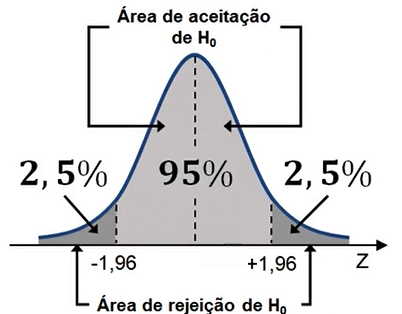
z\_alpha\_2.round(2)

1,96

COPIAR CÓDIGO

Isso aqui é um teste bicaudal, sempre é. São testes bicaudais, porque têm igualdade e a diferença, e ele consegue reportar para a gente.

[4:32] Aquele procedimento, como é 95%, a gente calcula aqui no gráfico. Isso aqui você já está cansado de saber. O 0,975, ponto do pontilhado da cauda direita até o final da cauda esquerda.



[4:43] Agora basta encontrar o PPF, o 1,96 que é o z e que vai fazer a divisão da minha área de aceitação com as minhas duas áreas de rejeição, porque o teste é bicaudal.

[4:56] Há algumas estatísticas que eu vou ter que obter. Na verdade, eu vou ter que obter esse o Z maiúsculo. Ele é o T, que é a menor soma de postos de mesmo sinal (nós faremos um passo a passo para melhor compreensão) menos o mi dividido pelo sigma. Esse Z, a partir de determinado n, se aproxima de uma normal, isto é, converge para uma distribuição normal. Não confunda com paramétrico.

[5:30] Seguindo, fumo é o Dataframe que eu criei. Vamos rodar. Está lá, correto, o Antes e o Depois.

fumo['Dif'] = fumo.Depois - fumo.Antes

fumo

COPIAR CÓDIGO

[5:37] A primeira coisa que eu tenho que fazer é pegar a diferença entre eles. Para isso, vou criar uma variável chamada 'Dif', de diferença, e vou pegar fumo.Depois e fazer a diferença entre o fumo.Antes.

[5:53] Criamos a Dif e precisamos fazer o Depois menos o Antes. Temos resultados positivos e negativos.

| **Antes** | **Depois** | **Dif** |  |
| --- | --- | --- | --- |
| 0 | 39 | 16 | -23 |
| 1 | 25 | 8 | -17 |
| 2 | 24 | 12 | -12 |
| 3 | 50 | 0 | -50 |
| 4 | 13 | 14 | 1 |
| 5 | 52 | 16 | -36 |
| 6 | 21 | 13 | -8 |
| 7 | 29 | 12 | -17 |
| 8 | 10 | 19 | 9 |
| 9 | 22 | 17 | -5 |
| 10 | 50 | 17 | -33 |
| 11 | 15 | 2 | -13 |
| 12 | 36 | 15 | -21 |
| 13 | 39 | 10 | -29 |
| 14 | 52 | 20 | -32 |
| 15 | 48 | 13 | -35 |
| 16 | 24 | 0 | -24 |
| 17 | 15 | 4 | -11 |
| 18 | 40 | 16 | -24 |
| 19 | 41 | 18 | -23 |
| 20 | 17 | 16 | -1 |
| 21 | 12 | 16 | 4 |
| 22 | 21 | 9 | -12 |
| 23 | 49 | 9 | -40 |
| 29 | 23 | 14 | -9 |
| 30 | 37 | 0 | -37 |
| 31 | 17 | 19 | 2 |
| 32 | 31 | 2 | -29 |
| 33 | 49 | 9 | -40 |
| ... |  |  |  |
| 29 | 23 | 14 | -9 |
| 30 | 37 | 0 | -37 |
| 31 | 17 | 19 | 2 |
| 32 | 31 | 2 | -29 |
| 33 | 49 | 9 | -40 |
| 34 | 49 | 6 | -43 |

[6:02] O próximo passo é, justamente, ignorar esses sinais, pegar o valor absoluto dessa diferença.

fumo['|Dif|'] = fumo.Dif.abs()

fumo

COPIAR CÓDIGO

[6:09] Chamei o Dif entre colchetes, que é como se fosse o valor absoluto da diferença, e, em seguida, fumo.Dif, que era a variável que a gente criou, e .abs(), que é justamente isso, ele ignora os sinais.

[6:22] De forma simples, é o valor absoluto. Está aqui o Dif e o valor absoluto de Dif. O módulo de Dif.

| **Antes** | **Depois** | **Dif** | **|Dif|** |  |
| --- | --- | --- | --- | --- |
| 4 | 13 | 14 | 1 | 1 |
| 20 | 17 | 16 | -1 | 1 |
| 31 | 17 | 19 | 2 | 2 |
| 21 | 12 | 16 | 4 | 4 |
| 24 | 14 | 18 | 4 | 4 |
| 9 | 22 | 17 | -5 | 5 |
| 6 | 21 | 13 | -8 | 8 |
| 29 | 23 | 14 | -9 | 9 |
| 8 | 10 | 19 | 9 | 9 |
| 17 | 15 | 4 | -11 | 11 |
| 2 | 24 | 12 | -12 | 12 |
| 22 | 21 | 9 | -12 | 12 |
| 11 | 15 | 2 | -13 | 13 |
| 28 | 28 | 11 | -17 | 17 |
| 1 | 25 | 8 | -17 | 17 |
| 7 | 29 | 12 | -17 | 17 |
| 12 | 36 | 15 | -21 | 21 |
| 27 | 22 | 0 | -22 | 22 |
| 19 | 41 | 18 | -23 | 23 |
| 0 | 39 | 16 | -23 | 23 |
| 16 | 24 | 0 | -24 | 24 |
| 18 | 40 | 16 | -24 | 24 |
| 26 | 46 | 17 | -29 | 29 |
| 13 | 39 | 10 | -29 | 29 |
| ... |  |  |  |  |
| 30 | 37 | 0 | -37 | 37 |
| 33 | 49 | 9 | -40 | 40 |
| 23 | 49 | 9 | -40 | 40 |
| 34 | 49 | 6 | -43 | 43 |
| 3 | 50 | 0 | -50 | 50 |
| 25 | 55 | 4 | -51 | 51 |

[6:35] Quando é positivo, fica positivo. Perfeito. Então é isso que a gente tem que fazer, já fizemos.

[6:40] Próximo passo: sorteá-los por essa nova variável que a gente acabou de criar, a variável que ignora os sinais negativos.

[6:47] Então eu faço fumo.sort\_Values by, em seguida, vem quem eu estou ordenando, que é o barra Dif, iplace igual a True, para efetivar a modificação.

fumo.sort\_values(by = '|Dif|', inplace = True)

fumo

COPIAR CÓDIGO

[7:00] Rodamos de novo, está aqui. Ele já está ordenado por esta coluna.

| **Antes** | **Depois** | **Dif** | **\Dif\** |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 13 | 14 | 1 | 1 |  |
| 20 | 17 | 16 | -1 | 1 |  |
| 31 | 17 | 19 | 2 | 2 |  |
| 21 | 12 | 16 | 4 | 4 |  |
| 24 | 14 | 18 | 4 | 4 |  |
| 9 | 22 | 17 | -5 | 5 |  |
| 6 | 21 | 13 | -8 | 8 |  |
| 29 | 23 | 14 | -9 | 9 |  |
| 8 | 10 | 19 | 9 | 9 |  |
| 17 | 15 | 4 | -11 | 11 | 10 |
| 2 | 24 | 12 | -12 | 12 | 11 |
| 22 | 21 | 9 | -12 | 12 | 12 |
| 11 | 15 | 2 | -13 | 13 | 13 |
| 28 | 28 | 11 | -17 | 17 | 14 |
| 1 | 25 | 8 | -17 | 17 | 15 |
| 7 | 29 | 12 | -17 | 17 | 16 |
| 12 | 36 | 15 | -21 | 21 | 17 |
| 27 | 22 | 0 | -22 | 22 | 18 |
| 19 | 41 | 18 | -23 | 23 | 19 |
| 0 | 39 | 16 | -23 | 23 | 20 |
| 16 | 24 | 0 | -24 | 24 | 21 |
| 18 | 40 | 16 | -24 | 24 | 22 |
| 26 | 46 | 17 | -29 | 29 | 23 |
| 13 | 39 | 10 | -29 | 29 | 24 |
| ... |  |  |  |  |  |
| 30 | 37 | 0 | -37 | 37 | 30 |
| 33 | 49 | 9 | -40 | 40 | 31 |
| 23 | 49 | 9 | -40 | 40 | 32 |
| 34 | 49 | 6 | -43 | 43 | 33 |
| 3 | 50 | 0 | -50 | 50 | 34 |
| 25 | 55 | 4 | -51 | 51 | 35 |

Você vê um, um, dois, quatro, quatro e por aí vai. São as diferenças ignorando os sinais.

[7:11] Próximo passo: eu vou criar uma variável que eu estou chamando de posto, que é justamente um *range*. É uma contagem, começa no primeiro, segundo, terceiro e por aí vai.

fumo = fumo.merge(posto, left\_on='|Dif|', right\_on='|Dif|', how = 'left')

fumo

COPIAR CÓDIGO

[7:23] Então, vamos criá-la aqui. Eu começo contando, é como se fosse um índice que começa do 1 e vai até o 35. É isso o posto.

| **Antes** | **Depois** | **Dif** | **|Dif|** | **Posto** |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 13 | 14 | 1 | 1 | 1.5 |
| 1 | 17 | 16 | -1 | 1 | 1.5 |
| 2 | 17 | 19 | 2 | 2 | 3.0 |
| 3 | 12 | 16 | 4 | 4 | 4.5 |
| 4 | 14 | 18 | 4 | 4 | 4.5 |
| 5 | 22 | 17 | -5 | 5 | 6.0 |
| 6 | 21 | 13 | -8 | 8 | 7.0 |
| 7 | 23 | 14 | -9 | 9 | 8.5 |
| 8 | 10 | 19 | 9 | 9 | 8.5 |
| 9 | 15 | 4 | -11 | 11 | 10.0 |
| 10 | 24 | 12 | -12 | 12 | 11.5 |
| 11 | 21 | 9 | -12 | 12 | 11.5 |
| 12 | 15 | 2 | -13 | 13 | 13.0 |
| 13 | 28 | 11 | -17 | 17 | 15.0 |
| 14 | 25 | 8 | -17 | 17 | 15.0 |
| 15 | 29 | 12 | -17 | 17 | 15.0 |
| 16 | 36 | 15 | -21 | 21 | 17.0 |
| 17 | 22 | 0 | -22 | 22 | 18.0 |
| 18 | 41 | 18 | -23 | 23 | 19.5 |
| 19 | 39 | 16 | -23 | 23 | 19.5 |
| 20 | 24 | 0 | -24 | 24 | 21.5 |
| 21 | 40 | 16 | -24 | 24 | 21.5 |
| 22 | 46 | 17 | -29 | 29 | 24.0 |
| 23 | 39 | 10 | -29 | 29 | 24.0 |
| ... |  |  |  |  |  |
| 29 | 37 | 0 | -37 | 37 | 30.0 |
| 30 | 49 | 9 | -40 | 40 | 31.5 |
| 31 | 49 | 9 | -40 | 40 | 31.5 |
| 32 | 49 | 6 | -43 | 43 | 33.0 |
| 33 | 50 | 0 | -50 | 50 | 34.0 |
| 34 | 55 | 4 | -51 | 51 | 35.0 |

[7:37] Próximo passo é criar um novo arquivo, um novo *series* no Dataframe, onde eu pego do Dataframe fumo somente essas duas variáveis: o Dif ignorando os sinais, o Dif absoluto, e Posto, que é essa variável que acabamos de criar.

[7:59] Faço um groupby, segundo essa variável Dif, e tiro a média.

posto = fumo[['|Dif|', 'Posto']].groupby(['|Dif|']).mean()

posto

COPIAR CÓDIGO

[8:06] O que vai acontecer? Eu vou pegar a os dados a partir da tabela, fazer um groupby segundo a variável que está na coluna |Dif|, linhas 4 e 20, o 1 e 1 que se repetem.

[8:16] Quando ele repete, eu pego a média. 1 mais 2 é igual a 3. Dividido por 2, é igual a 1,5 .

[8:23] Nesse caso aqui só tem um 2, então ele vai ser o próprio 3. A média de um número é o próprio número.

[8:28] Continuando na coluna |Dif|, temos duas repetições de 4, então, vai ser 4 mais 5, que é igual a 9. Dividido por 2, é igual a 4,5.

[8:34] Então vamos fazer isso para todo mundo. Está aqui. Rodamos, está lá, criou essa variável, 1,5 ; 3; 4, do jeito que eu tinha falado.

posto = fumo[['|Dif|', 'Posto']].groupby(['|Dif|']).mean()

posto

COPIAR CÓDIGO

| **|Dif|** | **Posto** |
| --- | --- |
| 1 | 1.5 |
| 2 | 3.0 |
| 4 | 4.5 |
| 5 | 6.0 |
| 8 | 7.0 |
| 9 | 8.5 |
| 11 | 10.0 |
| 12 | 11.5 |
| 13 | 13.0 |
| 17 | 15.0 |
| 21 | 17.0 |
| 22 | 18.0 |
| 23 | 19.5 |
| 24 | 21.5 |
| 29 | 24.0 |
| 32 | 26.0 |
| 33 | 27.0 |
| 35 | 28.0 |
| 36 | 29.0 |
| 37 | 30.0 |
| 40 | 31.5 |
| 43 | 33.0 |
| 50 | 34.0 |
| 51 | 35.0 |

[8:45] Próximo passo: eu quero utilizar essa variável |Dif| para fazer um Merge depois com o arquivo de Fumo. Ele transformou ela num índice, porque a gente fez o groupby por ela.

[8:57] Vou tirar esse |Dif| daqui e vou torná-lo uma variável novamente, fazendo o reset\_index, seguido do inplace = True, e rodar.

posto.reset\_index(inplace = True)

postoCOPIAR CÓDIGO

|  | **|Dif|** | **Posto** |
| --- | --- | --- |
| 0 | 1 | 1.5 |
| 1 | 2 | 3.0 |
| 2 | 4 | 4.5 |
| 3 | 5 | 6.0 |
| 4 | 8 | 7.0 |
| 5 | 9 | 8.5 |
| 6 | 11 | 10.0 |
| 7 | 12 | 11.5 |
| 8 | 13 | 13.0 |
| 9 | 17 | 15.0 |
| 10 | 21 | 17.0 |
| 11 | 22 | 18.0 |
| 12 | 23 | 19.5 |
| 13 | 24 | 21.5 |
| 14 | 29 | 24.0 |
| 15 | 32 | 26.0 |
| 16 | 33 | 27.0 |
| 17 | 35 | 28.0 |
| 18 | 36 | 29.0 |
| 19 | 37 | 30.0 |
| 20 | 40 | 31.5 |
| 21 | 43 | 33.0 |
| 22 | 50 | 34.0 |
| 23 | 51 | 35.0 |

[9:05] Ele criou um índice normal: 0, 1, 2, 3, pegou o Dif e transformou numa variável de novo, foi só isso.

[9:11] Próximo passo: lembra que a gente criou um posto no arquivo Fumo. Nesse arquivo Posto que a gente criou agora. Ele também tem uma variável Posto.

[9:21] Para evitar conflito, porque eu vou juntar os dois agora, eu vou eliminar a variável Posto do arquivo Fumo.

[9:29] Se você quiser ir fazendo com calma, porque parece que está meio confuso, vai fazendo com calma que você vai entender o passo a passo. É bem simples.

[9:35] O arquivo Fumo já está sem aquela variável Posto. Agora eu consigo fazer um Merge.

| **Antes** | **Depois** | **Dif** | **|Dif|** |  |
| --- | --- | --- | --- | --- |
| 4 | 13 | 14 | 1 | 1 |
| 20 | 17 | 16 | -1 | 1 |
| 31 | 17 | 19 | 2 | 2 |
| 21 | 12 | 16 | 4 | 4 |
| 24 | 14 | 18 | 4 | 4 |
| 9 | 22 | 17 | -5 | 5 |
| 6 | 21 | 13 | -8 | 8 |
| 29 | 23 | 14 | -9 | 9 |
| 8 | 10 | 19 | 9 | 9 |
| 17 | 15 | 4 | -11 | 11 |
| 2 | 24 | 12 | -12 | 12 |
| 22 | 21 | 9 | -12 | 12 |
| 11 | 15 | 2 | -13 | 13 |
| 28 | 28 | 11 | -17 | 17 |
| 1 | 25 | 8 | -17 | 17 |
| 7 | 29 | 12 | -17 | 17 |
| 12 | 36 | 15 | -21 | 21 |
| 27 | 22 | 0 | -22 | 22 |
| 19 | 41 | 18 | -23 | 23 |
| 0 | 39 | 16 | -23 | 23 |
| 16 | 24 | 0 | -24 | 24 |
| 18 | 40 | 16 | -24 | 24 |
| 26 | 46 | 17 | -29 | 29 |
| 13 | 39 | 10 | -29 | 29 |
| ... |  |  |  |  |
| 30 | 37 | 0 | -37 | 37 |
| 33 | 49 | 9 | -40 | 40 |
| 23 | 49 | 9 | -40 | 40 |
| 34 | 49 | 6 | -43 | 43 |
| 3 | 50 | 0 | -50 | 50 |
| 25 | 55 | 4 | -51 | 51 |

[9:41] Eu pego o fumo.merge com posto, eu junto as duas usando como variáveis de ligação, para o arquivo fumo o barra Dif, o Dif absoluto, e para esse right\_one='|Dif|'.

fumo = fumo.merge(posto, left\_on='|Dif|', right\_on='|Dif|', how = 'left')

fumo

COPIAR CÓDIGO

[9:54] De que maneira? 'left'. Eu quero que todo mundo do left, que é o fumo, seja mantido. É isso.

| **Antes** | **Depois** | **Dif** | **|Dif|** | **Posto** |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 13 | 14 | 1 | 1 | 1.5 |
| 1 | 17 | 16 | -1 | 1 | 1.5 |
| 2 | 17 | 19 | 2 | 2 | 3.0 |
| 3 | 12 | 16 | 4 | 4 | 4.5 |
| 4 | 14 | 18 | 4 | 4 | 4.5 |
| 5 | 22 | 17 | -5 | 5 | 6.0 |
| 6 | 21 | 13 | -8 | 8 | 7.0 |
| 7 | 23 | 14 | -9 | 9 | 8.5 |
| 8 | 10 | 19 | 9 | 9 | 8.5 |
| 9 | 15 | 4 | -11 | 11 | 10.0 |
| 10 | 24 | 12 | -12 | 12 | 11.5 |
| 11 | 21 | 9 | -12 | 12 | 11.5 |
| 12 | 15 | 2 | -13 | 13 | 13.0 |
| 13 | 28 | 11 | -17 | 17 | 15.0 |
| 14 | 25 | 8 | -17 | 17 | 15.0 |
| 15 | 29 | 12 | -17 | 17 | 15.0 |
| 16 | 36 | 15 | -21 | 21 | 17.0 |
| 17 | 22 | 0 | -22 | 22 | 18.0 |
| 18 | 41 | 18 | -23 | 23 | 19.5 |
| 19 | 39 | 16 | -23 | 23 | 19.5 |
| 20 | 24 | 0 | -24 | 24 | 21.5 |
| 21 | 40 | 16 | -24 | 24 | 21.5 |
| 22 | 46 | 17 | -29 | 29 | 24.0 |
| 23 | 39 | 10 | -29 | 29 | 24.0 |
| ... |  |  |  |  |  |
| 29 | 37 | 0 | -37 | 37 | 30.0 |
| 30 | 49 | 9 | -40 | 40 | 31.5 |
| 31 | 49 | 9 | -40 | 40 | 31.5 |
| 32 | 49 | 6 | -43 | 43 | 33.0 |
| 33 | 50 | 0 | -50 | 50 | 34.0 |
| 34 | 55 | 4 | -51 | 51 | 35.0 |

[10:05] Aquela variável posto já está aqui e também a média média.

[10:13] Próximo passo, que é o passo final: eu vou criar uma coluna para posto quando a diferença for positiva e quando a diferença for negativa.

[10:24] Aqui, Posto positivo eu estou chamando. Eu vou pegar fumo e usar a função apply e dentro dela eu vou usar uma função lambda.

[10:33] Onde eu vou atribuir, para cada linha, que quando o Dif for maior que zero, ou seja, positivo, eu vou colocar a minha variável Posto no lugar da minha variável Posto positivo.

[10:54] Vamos fazer aqui. Quando não for, ele vai colocar um zero. O axis igual a 1 é justamente que eu quero fazer isso linha a linha.

fumo['Posto (+)'] = fumo.apply(lambda x: x.Posto if x.Dif > 0 else 0, axis = 1)

fumo

COPIAR CÓDIGO

[11:03] Rodou. Quando o Dif é maior que zero, eu peguei o Posto positivo. Se não é maior que zero, eu coloco um zero.

| **Antes** | **Depois** | **Dif** | **|Dif|** | **Posto** | **Posto (+)** |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 13 | 14 | 1 | 1 | 1.5 | 1.5 |
| 1 | 17 | 16 | -1 | 1 | 1.5 | 0.0 |
| 2 | 17 | 19 | 2 | 2 | 3.0 | 3.0 |
| 3 | 12 | 16 | 4 | 4 | 4.5 | 4.5 |
| 4 | 14 | 18 | 4 | 4 | 4.5 | 4.5 |
| 5 | 22 | 17 | -5 | 5 | 6.0 | 0.0 |
| 6 | 21 | 13 | -8 | 8 | 7.0 | 0.0 |
| 7 | 23 | 14 | -9 | 9 | 8.5 | 0.0 |
| 8 | 10 | 19 | 9 | 9 | 8.5 | 8.5 |
| 9 | 15 | 4 | -11 | 11 | 10.0 | 0.0 |
| 10 | 24 | 12 | -12 | 12 | 11.5 | 0.0 |
| 11 | 21 | 9 | -12 | 12 | 11.5 | 0.0 |
| 12 | 15 | 2 | -13 | 13 | 13.0 | 0.0 |
| 13 | 28 | 11 | -17 | 17 | 15.0 | 0.0 |
| 14 | 25 | 8 | -17 | 17 | 15.0 | 0.0 |
| 15 | 29 | 12 | -17 | 17 | 15.0 | 0.0 |
| 16 | 36 | 15 | -21 | 21 | 17.0 | 0.0 |
| 17 | 22 | 0 | -22 | 22 | 18.0 | 0.0 |
| 18 | 41 | 18 | -23 | 23 | 19.5 | 0.0 |
| 19 | 39 | 16 | -23 | 23 | 19.5 | 0.0 |
| 20 | 24 | 0 | -24 | 24 | 21.5 | 0.0 |
| 21 | 40 | 16 | -24 | 24 | 21.5 | 0.0 |
| 22 | 46 | 17 | -29 | 29 | 24.0 | 0.0 |
| 23 | 39 | 10 | -29 | 29 | 24.0 | 0.0 |
| ... |  |  |  |  |  |  |
| 29 | 37 | 0 | -37 | 37 | 30.0 | 0.0 |
| 30 | 49 | 9 | -40 | 40 | 31.5 | 0.0 |
| 31 | 49 | 9 | -40 | 40 | 31.5 | 0.0 |
| 32 | 49 | 6 | -43 | 43 | 33.0 | 0.0 |
| 33 | 50 | 0 | -50 | 50 | 34.0 | 0.0 |
| 34 | 55 | 4 | -51 | 51 | 35.0 | 0.0 |

[11:12] Se é maior que zero, o Posto está em Posto (+). E assim sucessivamente.

[11:17] A mesma coisa para o negativo.

fumo['Posto (-)'] = fumo.apply(lambda x: x.Posto if x.Dif < 0 else 0, axis = 1)

fumo

COPIAR CÓDIGO

Rodamos, eu crio um negativo.

| **Antes** | **Depois** | **Dif** | **|Dif|** | **Posto** | **Posto (+)** | **Posto (-)** |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 13 | 14 | 1 | 1 | 1.5 | 1.5 | 0.0 |
| 1 | 17 | 16 | -1 | 1 | 1.5 | 0.0 | 1.5 |
| 2 | 17 | 19 | 2 | 2 | 3.0 | 3.0 | 0.0 |
| 3 | 12 | 16 | 4 | 4 | 4.5 | 4.5 | 0.0 |
| 4 | 14 | 18 | 4 | 4 | 4.5 | 4.5 | 0.0 |
| 5 | 22 | 17 | -5 | 5 | 6.0 | 0.0 | 6.0 |
| 6 | 21 | 13 | -8 | 8 | 7.0 | 0.0 | 7.0 |
| 7 | 23 | 14 | -9 | 9 | 8.5 | 0.0 | 8.5 |
| 8 | 10 | 19 | 9 | 9 | 8.5 | 8.5 | 0.0 |
| 9 | 15 | 4 | -11 | 11 | 10.0 | 0.0 | 10.0 |
| 10 | 24 | 12 | -12 | 12 | 11.5 | 0.0 | 11.5 |
| 11 | 21 | 9 | -12 | 12 | 11.5 | 0.0 | 11.5 |
| 12 | 15 | 2 | -13 | 13 | 13.0 | 0.0 | 13.0 |
| 13 | 28 | 11 | -17 | 17 | 15.0 | 0.0 | 15.0 |
| 14 | 25 | 8 | -17 | 17 | 15.0 | 0.0 | 15.0 |
| 15 | 29 | 12 | -17 | 17 | 15.0 | 0.0 | 15.0 |
| 16 | 36 | 15 | -21 | 21 | 17.0 | 0.0 | 17.0 |
| 17 | 22 | 0 | -22 | 22 | 18.0 | 0.0 | 18.0 |
| 18 | 41 | 18 | -23 | 23 | 19.5 | 0.0 | 19.5 |
| 19 | 39 | 16 | -23 | 23 | 19.5 | 0.0 | 19.5 |
| 20 | 24 | 0 | -24 | 24 | 21.5 | 0.0 | 21.5 |
| 21 | 40 | 16 | -24 | 24 | 21.5 | 0.0 | 21.5 |
| 22 | 46 | 17 | -29 | 29 | 24.0 | 0.0 | 24.0 |
| 23 | 39 | 10 | -29 | 29 | 24.0 | 0.0 | 24.0 |
| 29 | 37 | 0 | -37 | 37 | 30.0 | 0.0 | 30.0 |
| 30 | 49 | 9 | -40 | 40 | 31.5 | 0.0 | 31.5 |
| 31 | 49 | 9 | -40 | 40 | 31.5 | 0.0 | 31.5 |
| 32 | 49 | 6 | -43 | 43 | 33.0 | 0.0 | 33.0 |
| 33 | 50 | 0 | -50 | 50 | 34.0 | 0.0 | 34.0 |
| 34 | 55 | 4 | -51 | 51 | 35.0 | 0.0 | 35.0 |

[11:26] Quando ele for menor que zero, ele vai ser igual ao Posto. Caso contrário, ele vai ser zero. Agora, o que a gente tem que fazer?

[11:32] Agora, só para deixar organizado, bonito, eu vou tirar o Posto.

fumo.drop(['Posto'], axis = 1, inplace = True)

fumo

COPIAR CÓDIGO

Pronto, fiquei só com o que me interessa no arquivo.

[11:41] Para o passo seguinte, precisamos voltar nas informações anteriores. Para criar a nossa estatística, o Z, eu preciso de T maiúsculo menos um MiT e o SigmaT.

[11:52] Quem é o T? É justamente o mínimo entre a soma dessas duas variáveis que a gente criou aqui agora, Posto (+), Posto (-). O valor mínimo é o meu T.

[12:05] Então, T vai ser igual a Min, que é o mínimo, entre fumo Posto positivo, essa é a variável, .sum().

T = min(fumo['Posto (+)'].sum(), fumo['Posto (-)'].sum())

T

22.0

COPIAR CÓDIGO

[12:15] E depois temos o Posto negativo. Eu somo tudo e verifico qual dos dois é o menor e pego.

[12:24] O menor da soma dos dois é 22. Agora, vamos calcular o MiT, que é a fórmula: n, a gente já tem lá de cima, que é o 35, vezes n mais 1, dividido por quatro. É simples, é isso aí. 315.

mu\_T = (n \* (n + 1)) / 4

mu\_T

315.0

COPIAR CÓDIGO

[12:43] O Sigma T é uma fórmula maior: np.sqrt, que é a raiz quadrada, e n vezes n mais 1, vezes, duas vezes n mais um, divididos por 24.

sigma\_T = np.sqrt((n \* (n + 1) \* ((2 \* n) + 1)) / 24)

sigma\_T

61.053255441458646

COPIAR CÓDIGO

[13:02] Esse é o Sigma T: 61,05. Agora vem o Z, que é o Zteste que eu estou chamando aqui em cima, é o Z maiúsculo.

[13:12] Eu calculo ele dessa forma: T menos o MiT, divididos pelo Sigma. Aqui é bem simples, é só isso. E a gente tem a nossa estatística de teste, finalmente.

Z = (T - mu\_T) / sigma\_T

Z

-4.799088891843698

COPIAR CÓDIGO

[13:20] Estamos fazendo isso tudo para entender como o processo é feito manualmente. Lógico que você não vai precisar fazer tudo isso toda vez que você for fazer esse teste.

[13:26] Eu vou mostrar para você, no próximo vídeo, como fazer de maneira simples utilizando o Python, aqui é só para a gente entender.

[13:34] O - 4,8 fica dentro da área de rejeição. Então, aqui, aquele mesmo processo: a estatística de teste, o H0 e o H1, e as situações de rejeição e aceitação.

[13:47] Vamos testar a primeira: Z é menor ou igual a menos Z alfa sobre dois? Sim.

Z <= -z\_alpha\_2

True

Z >= z\_alpha\_2

False

COPIAR CÓDIGO

[14:01] Ele está dentro da área de rejeição, portanto, eu rejeito H0, que é a hipótese de que não há diferença entre os grupos, ou seja, existe uma diferença entre os grupos.

[14:14] Então, conclusão, rapidamente para passarmos para o outro vídeo.

[14:17] "Rejeitamos a hipótese de que não existe diferença entre os grupos, isto é, existe uma diferença entre as médias de cigarros fumados pelos pacientes antes e depois do tratamento".

[14:27] Ou seja, o tratamento parece ser eficiente. "E como é possível verificar através das médias de cigarros fumados por dia antes", lembra aquela média que a gente calculou lá no começo?

[14:37] "31,86, e depois ,11,2, do tratamento, podemos concluir que o tratamento apresentou resultado satisfatório." Legal esse teste.

[14:48] A gente consegue fazer esse tipo de averiguação quando estamos aplicando um tratamento novo, por exemplo, em um grupo de pacientes, que é o nosso caso aqui.

[14:59] No próximo vídeo eu vou mostrar como fazer isso de forma fácil, para a gente não precisar fazer tudo isso de posto, já tenho uma função que resolve tudo para a gente. Até lá.