PL2

Probabilidades e Variáveis aleatórias

Métodos Probabilísticos para Engenharia Informática

2020/2021

Pedro Daniel Fontes Lopes 97827

Rodrigo França Lima 98475

1 Introdução ao problema proposto

Considere uma empresa fabricante de brinquedos que produz um determinado brinquedo. O brinquedo é composto por dois componentes (1 e 2) que são produzidos separadamente e posteriormente montados. No final, os brinquedos são embalados para comercialização em caixas com n brinquedos cada. O processo de fabrico do Componente 1 produz p1 = 0,2% de componentes com defeito. O processo de fabrico do Componente 2 produz p2 = 0,5% de componentes com defeito. Um brinquedo está com defeito se pelo menos um de seus componentes estiver com defeito. O processo de montagem produz pa = 1% de brinquedos com defeito (mesmo quando nenhum dos 2 componentes está com defeito).

2 Desenvolvimento dos exercícios propostos

Para a simulação da companhia de construção de brinquedos recorreu-se à simulação de cada caixa com os seus *n* brinquedos. Primeiramente definiu-se a variável *N* cujo valor representa o número de caixas a serem fabricadas, este valor será elevado pois sendo este elevado futuramente se obterá resultados mais precisos. Cada caixa terá como enunciado *n* brinquedos sendo estes compostos por duas peças *p1* e *p2*. Para o cálculo dos defeitos possíveis por brinquedo recorre-se à função *rand()*, esta função criará uma matriz cujo valor estará compreendido entre 0 e 1 exclusivé. Neste caso mais específico recorreu-se ao uso de *rand(n,N)* que será a variável experiências. Esta variável experiências será uma matriz *nxN* cujos valores em cada célula serão pseudo aleatórios. Como cada brinquedo possui uma peça com uma chance de esta possuir defeito se aplicarmos uma comparação nomeadamente se a matriz experiências celula a celula é inferior à probabilidade da peça ter defeito obtém-se assim uma matriz em que as *N* colunas representam uma caixa de brinquedos e as n linhas representam as peças individuais de cada brinquedo em cada caixa.

Através deste método é assim possível calcular de maneira aleatória tendo em conta a probabilidade de defeito, as peças defeituosas. Tendo cada célula da matriz após a comparação com a probabilidade de defeito, 1 caso seja defeituosa ou 0 caso não seja.

Um brinquedo pode ter três fatores independentes de origem de defeitos. Tendo isto em conta para a simulação de defeitos em cada respectivo brinquedo deve-se então simular independente cada defeito. Isto é realizado através do método anteriormente definido e a re-simulação da matriz experiências para cada fator de defeito. Obtém-se assim três matrizes *p1*, *p2* e *pa* cujo tamanho é *nxN* e cuja soma destas representa célula a célula o número de defeitos em cada respectivo brinquedo sendo esta soma representada pela variável *ndefeitos*.

A verificação desta matriz célula a célula para ver se é superior a 0 resulta numa matriz *defeitos* que apresenta em cada célula 1 caso o brinquedo tenha pelo menos um defeito em si e 0 caso não o tenha, sendo a soma das linhas em cada coluna desta matriz a variável *nbrinquedos* que representa o número de brinquedos defeituosos em cada respectiva caixa.

```
N = 1e5;                                % numero de experiencias
n = 8;                                  % numero de brinquedos por caixa
experiencias = rand(n,N);               % fabrico de peças p1
p1  = experiencias < 0.002;             % peças p1 com defeito
experiencias = rand(n,N);               % fabrico de peças p2
p2  = experiencias < 0.005;             % peças p2 com defeito
experiencias = rand(n,N);               % processo de montagem pa
pa  = experiencias < 0.01;              % defeitos na montagem pa com defeito
ndefeitos = p1 + p2 + pa;               % numero de defeitos por brinquedo
defeitos = ndefeitos > 0;               % brinquedos com defeito
nbrinquedos = sum(defeitos);            % numero de brinquedos com defeito por caixa
```

Figura 1: Simulador do processo de fabrico de Brinquedos.

1. Considere o evento ”A - uma caixa de brinquedos tem pelo menos 1 brinquedo com defeito”.

(a) Estime por simulação a probabilidade do evento A quando n = 8 brinquedos.

```
N = 1e5;                                  % numero de experiencias
n = 8;                                    % numero de brinquedos por caixa
experiencias = rand(n,N);                 % fabrico de peças p1
p1  = experiencias < 0.002;               % peças p1 com defeito
experiencias = rand(n,N);                 % fabrico de peças p2
p2  = experiencias < 0.005;               % peças p2 com defeito
experiencias = rand(n,N);                 % processo de montagem pa
pa  = experiencias < 0.01;                % defeitos na montagem pa com defeito
ndefeitos = p1 + p2 + pa;                 % numero de defeitos por brinquedo
defeitos = ndefeitos > 0;                 % brinquedos com defeito
nbrinquedos = sum(defeitos);              % numero de brinquedos com defeito por caixa
caixas = nbrinquedos >=1;                 % soma das caixas com pelo menos um brinquedo com defeito
probA= sum(caixas)/N                      % calculo da probabilidade do acontecimento A
```

Figura 2: Simulador e cálculo da probabilidade de A

De forma a que a probabilidade por simulação possa ser próxima da probabilidade real foram usadas 100000 amostras de cada componente, o *N* usado não foi escolhido ao acaso uma vez que é um números suficientemente grande para que se reduza o erro da probabilidade do acontecimento *A* é suficientemente pequeno para não ser demasiado exigente do ponto de vista computacional.

Através do código apresentado na figura 1 e tal como explicado, é possível com aquele simulador calcule os casos favoráveis ao acontecimento A. Através da seguinte comparação *ndefeitos* > *0*, sendo *defeitos* a matriz que resulta da comparação anterior, e *nbrinquedos* a matriz linha que resulta da soma das colunas da matriz *defeitos*, em que a célula (1,j) mostra a quantidade de defeitos que existe na caixa. Portanto já temos toda a informação necessária para os calcular, logo criamos outra matriz linha: *caixas,* que tem valor 1 caso a caixa que representa contêm pelo menos um brinquedo com defeito, somando todos estes uns obtemos os casos favoráveis pelo que basta dividir o resultado obtido pela soma anterior por *N* para calcular o valor da probabilidade de *A* que é 0.1292.

(b) Estime por simulação o número médio de brinquedos defeituosos apenas devido ao processo de montagem quando ocorre o evento A e n = 8 brinquedos

Com o intuito de simular este ambiente foram novamente geradas matrizes *nxN* para cada componente e para o processo de fabrico, com números aleatórios, tal como é possível ver na figura 1.

```
defeitos = ndefeitos > 0;                 % brinquedos com defeito
todosdef = sum(defeitos);                 % soma de todos os brinquedos defeitoosos por caixa
```

Figura 3: Matriz todosdef, que tem o valor de 1 nas colunas em que a respetiva caixa contém pelo menos um brinquedo com defeito

Contudo sendo agora o objetivo estimar a média de brinquedos defeituosos apenas devido ao processo de fabrico quando o evento *A* ocorre, foram primeiro identificadas as colunas em que isso se confirma na matriz *todosdef*.

```
mdefeitos = (p1+p2)>0;                          % calculo de defeitos provocados por peças
mdefeitos = (mdefeitos ~= pa) & pa;             % calculo de defeitos provocados só por montagem
```

Figura 4: Cálculo da matriz representante dos brinquedos com defeitos provocados só pelo processo de montagem

Para obter uma matriz relativa ao acontecimento *B* foi necessário somar as matrizes *p1* e *p2*, e ver quando estas eram superiores a 0, ou seja, quando um dos componentes que constitui o brinquedo tem defeito. De seguida identificamos quando é que o acontecimento *B* acontece com a seguinte expressao *(mdefeitos ~= pa) & pa*, que verifica quais os brinquedos com defeitos cujos defeitos são provenientes unicamente do processo de montagem. Para se obter este resultado verifica-se dos defeituosos devido a qualquer defeito e verifica-se quais são os que possuem esse defeito unicamente devido a pa, comparando-se assim a matriz de qualquer defeito *mdefeitos* com *pa* vendo quais são os diferentes e fazendo uma conjunção com *pa* obtendo assim os que são defeituosos unicamente pelo processo de montagem

```
for d = 1:N                                          % for para iteração das matrizes
    if(todosdef(d) ~=0)                              % verificação se existe pelo menos 1 brinquedo com defeito por caixa
        medias= medias + nmdefeitos(d)/todosdef(d);  % soma das médias de brinquedos defeitoosos devido a montagem em comparação aos defeitos totais por caixa
        counter = counter + 1;                       % contador de casos com pelo menos um defeito por caixa
    end                                              %
end                                                  %
```

Figura 5: Ciclo for que varre todosdef, à procura de caixas em que A acontece

De seguida com a ajuda de um for loop a matriz *todosdef* foi varrida, sendo que sempre que um valor igual 1 fosse encontrado era somado à variável *medias* o número de defeitos originados apenas pelo processo de montagem a dividir pelo número de defeitos encontrados nessa caixa, e o *counter* incrementado, para que no fim deste loop a variável *medias* tivesse o somatório das médias de brinquedos defeituosos devido à montagem em comparação aos defeitos totais por caixa, e *counter* o número de vezes que isso aconteceu, após isto basta dividir a variável *medias* por *counter* e obtêm-se o valor 0.5839.

```
result = medias/counter                         % média final das médias obtidas
```

Figura 6: Cálculo do número médio de brinquedos defeituosos devido unicamente ao processo de montagem

2. Considere o evento ”B - uma caixa de brinquedos não tem brinquedos com defeito”.

(a) Estime por simulação a probabilidade do evento B quando n = 8 brinquedos. Verifique a consistência deste resultado com o obtido na questão 1(a).

```
caixas = nbrinquedos ==0;                  % soma das caixas com nenhum brinquedo com nenhum defeito
probB= sum(caixas)/N                       % calculo da probabilidade do acontecimento B
```

Figura 7: Cálculo da probabilidade de B.

Para estimar o valor da probabilidade de *B* foi usada a mesma abordagem da alínea 1(a), implementando o simulador proposto na figura 1, contudo a matriz caixas contém o valor 1 caso a caixa *m(1:N)* não tenha brinquedos defeituosos, pelo que a soma de todos os valores da matriz caixas irá dar origem ao número de casos favoráveis ao acontecimento *B*, pelo que para obter a *probB* basta dividir o valor dessa soma por N. O resultado obtido para a probabilidade do acontecimento *B*, que foi de 0.8706 que é compatível com o já obtido resultado da probabilidade de *A*.

```
vrfy = probB + probA                          % verificação se os valores da probA e probB são compativeis
```

Figura 8: Verificação da compatibilidade entre probB e probA

Uma vez que o acontecimento *B* é o acontecimento complementar do acontecimento *A* era expectável que a soma das duas probabilidades resultasse num valor próximo de 1.

(b) Determine o valor teórico da probabilidade do evento B e compare-o com o valor estimado por simulação na questão 2(a). O que conclui?

Sendo que o evento *B* é um conjunto de experiências de Bernoulli, em que um brinquedo não ter defeito é considerado como um sucesso, declaramos X como a número de sucessos que desejamos, que devido à definição do acontecimento *B* tem de ser 8.

Como um conjunto de experiências de Bernoulli pode ser analisado como um experiência Binomial usamos a fórmula teórica para a obtenção da probabilidade de *B* $\frac{n! \times p^X \times (1-p)^{(n-X)}}{(n-X)! \times X!}$, pelo que no final devemos obter uma probabilidade por simulação muito próxima da obtida pela fórmula teórica, esta aproximação só é possível porque o número de amostras(N) utilizado foi consideravelmente grande para evitar grandes variações na probabilidade de simulação.

Concluindo, o valor previamente calculado por simulação que deu 0.8706, está muito próximo da probabilidade obtida pela fórmula teórica acima apresentada que deu 0.8724, já que estes dois valores são muito próximos retira-se que o simulador implementado retrata bem o problema.

(c) Faça as simulações necessárias para desenhar um gráfico plot da probabilidade do evento B em função da capacidade da caixa n. Considere todos os valores de n de 2 a 20. Descreva e justifique os resultados obtidos.

```
y = (2:20);                                   % probabilidade do acontecimento B em função da capacidade da caixa
N=1e5;                                        % numero de experiencias
for n =  2:20                                 % numero de brinquedos por caixa
    experiencias = rand(n,N);                 % fabrico de peças p1
    p1  = experiencias < 0.002;               % peças p1 com defeito
    experiencias = rand(n,N);                 % fabrico de peças p2
    p2  = experiencias < 0.005;               % peças p2 com defeito
    experiencias = rand(n,N);                 % processo de montagem pa
    pa  = experiencias < 0.01;                % defeitos na montagem pa com defeito
    ndefeitos = p1 + p2 + pa;                 % numero de defeitos por brinquedo
    defeitos = ndefeitos > 0;                 % brinquedos com defeito
    nbrinquedos = sum(defeitos);              % numero de brinquedos com defeito por caixa
    caixas = nbrinquedos ==0;                 % soma das caixas sem nenhum brinquedo com defeito
    y(n-1) = sum(caixas)/N;                   % probabilidade experimental das caixas de n = 2:20 não terem brinquedos defeituosos
end
plot(2:20,y,"r.-");
xlabel ("Capacidade da caixa");
ylabel ("Probabilidade de B");
legend ("Probabilidade de B como funções da capacidade da caixa n");
```

Figura 9: Ciclo for que percorre o vetor (2:20) para calcular a probabilidade de B para os diferentes número de brinquedos por caixa.

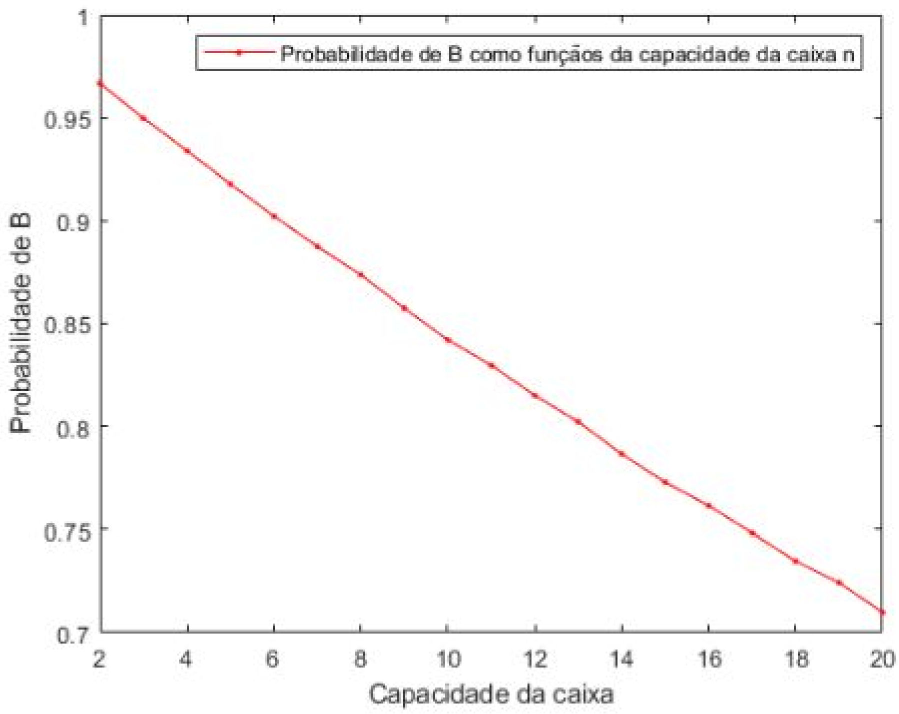

Figura 10: Probabilidade de B em função da capacidade da caixa.

Para a obtenção das probabilidades do acontecimento *B* em função de n, foi utilizado o código da pergunta 1(b) tendo sido criado um novo vetor de dimensão *1X21* para acomodar os valores das diferentes probabilidades para os diferentes n, e um ciclo for que irá a cada iteração calcular o valor da probabilidade *B* para um valor de n diferente e colocar na posição *(n-1)* do vetor y o valor da probabilidade obtida.

Analisando-os valores obtidos com o aumento do número brinquedos por caixa a probabilidade do acontecimento *B* diminui, isto significa que com o aumento de número de brinquedos por caixa existe também uma subida no número de defeitos por caixa.

(d) Analisando o gráfico traçado na questão anterior, 2(c), qual deve ser a capacidade máxima da caixa se a empresa quiser garantir que a probabilidade de cada caixa não ter brinquedos com defeito seja de pelo menos 90%?

Acrescentado uma linha reta que marca o valor de 90% de probabilidade de uma caixa não ter brinquedos defeituosos ao gráfico da figura 6, torna-se visível que se o objetivo da empresa é maximizar o número de brinquedos por caixa assegurando que pelo menos 90% dessas caixas não tem brinquedos com defeito, pelo que devem ser colocados 6 brinquedos por caixa.

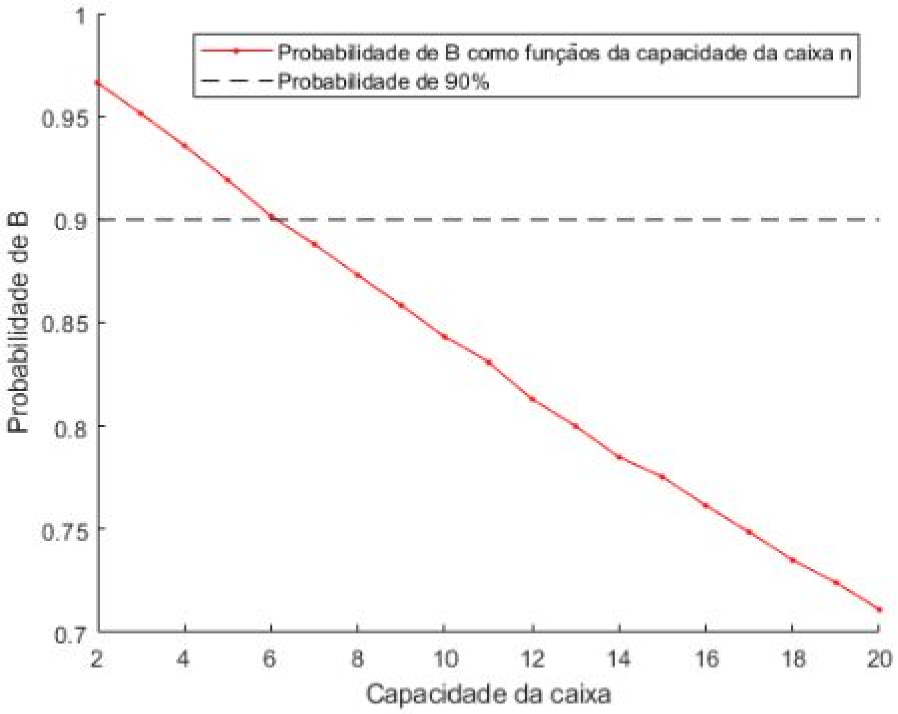

Figura 11: Probabilidade de B em função da capacidade da caixa com uma linha reta que marca a probabilidade de B ser 90%.

3. Considere a variável aleatória X que representa o número de brinquedos defeituosos numa caixa

(a) Estime por simulação a função massa de probabilidade pX(x) de X quando n = 8 brinquedos e desenhe-a num gráfico stem.

```
N = 1e5;                                % numero de experiencias
n = 8;                                  % numero de brinquedos por caixa
experiencias = rand(n,N);               % fabrico de peças p1
p1  = experiencias < 0.002;             % peças p1 com defeito
experiencias = rand(n,N);               % fabrico de peças p2
p2  = experiencias < 0.005;             % peças p2 com defeito
experiencias = rand(n,N);               % processo de montagem pa
pa  = experiencias < 0.01;              % defeitos na montagem pa com defeito
ndefeitos = p1 + p2 + pa;               % numero de defeitos por brinquedo
defeitos = ndefeitos > 0;               % brinquedos com defeito
nbrinquedos = sum(defeitos);            % numero de brinquedos com defeito
xi = (0:8);                             % conjunto dos valores possiveis de serem assumidos
px = zeros(1,9);                        % probabilidade associada a cada um dos valores possiveis de serem assumidos
for val = nbrinquedos                   % iteração pelos valores obtidos por simulação
    switch(val)                         % filtração dos valores
    case 0                              % caso a caixa tenha 0 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 0 brinquedos com defeito
    case 1                              % caso a caixa tenha 1 brinquedo com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 1 brinquedo com defeito
    case 2                              % caso a caixa tenha 2 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 2 brinquedos com defeito
    case 3                              % caso a caixa tenha 3 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 3 brinquedos com defeito
    case 4                              % caso a caixa tenha 4 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 4 brinquedos com defeito
    case 5                              % caso a caixa tenha 5 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 5 brinquedos com defeito
    case 6                              % caso a caixa tenha 6 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 6 brinquedos com defeito
    case 7                              % caso a caixa tenha 7 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 7 brinquedos com defeito
    case 8                              % caso a caixa tenha 8 brinquedos com defeito
        px(val+1) = px(val+1) + 1;      % contar numero de caixas com 8 brinquedos com defeito
    otherwise                           % verificação de erro
        fprintf("Ocorreu um erro\n");   % aviso de erro
    end                                 %
end                                     %
px = px/(N)                             % cálculo da probabilidade associada a cada um dos valores possiveis de serem assumidos
```

Figura 12: Simulador de fabrico de brinquedos e código respetivo ao problema a resolver

Para a o cálculo da função probabilidade *pX(x)* de X (número de brinquedos defeituosos numa caixa) quando a caixa contém 8 brinquedos, recorreu-se ao uso do cálculo do número de brinquedos com defeitos em cada caixa de maneira igual à apresentada na figura 1.

Após o cálculo e armazenamento do número de brinquedos defeituosos por caixa através da variável *nbrinquedos* ,sendo esta uma matriz *1xN* cujas células apresentam cada uma respetivamente uma caixa de *n* brinquedos e cujo valor é o número de brinquedos defeituosos contidos nessa mesma caixa, criou-se uma matriz *xi* cujo tamanho é *1x9* e criou-se a matriz *px* de tamanho *1x9* e cujos valores são todos zeros para mais tarde inserir-se os valores pretendidos nela.

Tendo os valores de quantos brinquedos defeituosos se tem por caixa, fez-se uso de um for para iteração pelas N caixas obtidas e o uso de um *switch* para filtração dos valores obtidos em cada caixa.

```matlab
for val = nbrinquedos                          % iteração pelos valores obtidos por simulação
    switch(val)                                % filtração dos valores
    case 0                                     % caso a caixa tenha 0 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 0 brinquedos com defeito
    case 1                                     % caso a caixa tenha 1 brinquedo com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 1 brinquedo com defeito
    case 2                                     % caso a caixa tenha 2 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 2 brinquedos com defeito
    case 3                                     % caso a caixa tenha 3 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 3 brinquedos com defeito
    case 4                                     % caso a caixa tenha 4 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 4 brinquedos com defeito
    case 5                                     % caso a caixa tenha 5 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 5 brinquedos com defeito
    case 6                                     % caso a caixa tenha 6 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 6 brinquedos com defeito
    case 7                                     % caso a caixa tenha 7 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 7 brinquedos com defeito
    case 8                                     % caso a caixa tenha 8 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 8 brinquedos com defeito
    otherwise                                  % verificação de erro
        fprintf("Ocorreu um erro\n");          % aviso de erro
    end                                        %
```

Figura 13: For loop e switch

O switch irá assim verificar se o valor de brinquedos defeituosos é um valor compreendido entre 0 e 8 e por sua vez irá somar 1 ao valor da célula da matriz <u>px</u> cuja posição é <u>*val*</u>+1.

Tem-se assim então <u>*k*</u> a posição <u>*px(1)*</u> corresponderá ao número de caixas que obtiveram 0 brinquedos com defeito e a posição <u>*px(2)*</u> corresponderá assim o número de caixas que obtiveram 1 brinquedos com defeito tendo-se assim todo o número de caixas que se obtiveram com o dentro dos números possíveis de se saírem <u>*xi*</u>.

Caso se obtenha um erro também se recorre à componente *otherwise* do *switch* para alertar do erro.

```matlab
otherwise                                  % verificação de erro
    fprintf("Ocorreu um erro\n");          % aviso de erro
end                                        %
```

Figura 14: Declaração de erro

Para cálculo da probabilidade recorre-se à divisão dos valores contados e guardados na matriz <u>*px*</u> pelo número de experiências realizadas <u>*N*</u>. Tendo-se assim então a probabilidade *pX(<u>x</u>)*.

Como verificação do resultado obtido previamente recorreu-se à soma das probabilidades obtidas previamente. Sendo para o resultado estar certo a soma ser obrigatoriamente com resultado igual a 1, resultado que fora obtido nesta simulação.

```matlab
sum(px)                                    % verificação de resultados (soma tem que ser igual a 1)
```

Figura 15: Soma das probabilidades obtidas

Após a obtenção dos resultados guardados em px e a sua respetiva verificação, realizou-se assim o desenho do gráfico da função probabilidade *pX(x)* de X (número de brinquedos defeituosos numa caixa) através da função *stem()*.

```
stem(xi,px,"filled",'r.')                          % stem da função massa
xlabel("x");
ylabel("pX(x)");
```

Figura 16: Stem e labels do pretendido

Obtendo-se assim o respetivo resultado:

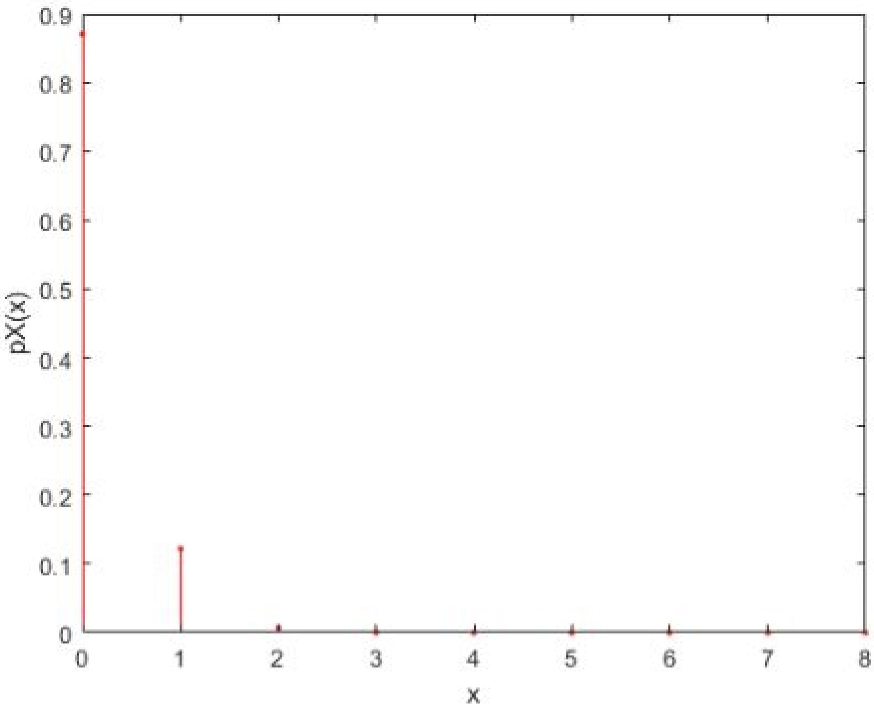

Figura 17: Histograma da função massa de probabilidade

Os resultados obtidos por simulação obtidos foram respectivamente:

- *pX(0)* ≈ 0.87085;
- *pX(1)* ≈ 0.12151;
- *pX(2)* ≈ 0.00739;
- *pX(3)* ≈ 0.00024;
- *pX(4)* ≈ 1.00000e-05;
- *pX(5)* ≈ 0;
- *pX(6)* ≈ 0;
- *pX(7)* ≈ 0;
- *pX(8)* ≈ 0;

Verifica-se então através dos resultados obtidos que quando o número de brinquedos por caixa n é igual a 8 a probabilidade de se ter xi brinquedos com defeito por caixa vai diminuindo drasticamente ao longo do aumento de xi. Verifica-se também que a probabilidade de se ter xi brinquedos com defeito por caixa equivale a zero se xi for igual a

5. Estes valores vão em conformidade à não elevada probabilidade de um brinquedo defeituoso.

Comparando-se assim os resultados obtidos nesta simulação com os resultados obtidos no ponto 2 a) verifica-se que:

```
px(1)                                   % probabilidade obtida por simulação de não existirem brinquedos defeitoosos numa caixa com n brinquedos
```

Figura 18: verificação de que px(1) = probB

- O resultado obtido nesta simulação para quando se pretende que nenhum brinquedo tenha defeito (evento B) foi 0.8709.

```
probB                                   % probabilidade de não existirem brinquedos defeitoosos numa caixa com n brinquedos
```

Figura 19: verificação de que px(1) = probB

- O resultado obtido no ponto 2 a) quando se pretende que nenhum brinquedo tenha defeito (evento B) foi 0.8724.

Concluímos assim que os resultados permanecem consistentes mesmo após diferentes simulações destes.

(b) Com base em pX(x), calcule a probabilidade de X >= 2.

```
px2 = sum(px(3:9));                     % probabilidade obtida por simulação
```

Figura 20: Cálculo da probabilidade de uma caixa ter pelo menos dois brinquedos sem defeito

Como foi obtido na alínea anterior as diferentes probabilidades para cada valor de número de sucessos basta calcular a soma das probabilidades cujo o número de sucessos é superior ou igual a 2, sendo possível atingir este objetivo somando as colunas da matriz _px_ a partir de 3 até ao fim da matriz, que resulta na probabilidade de 0.0075.

O que conclui?

Sendo que a probabilidade de haver 2 ou mais brinquedos defeituosos numa caixa de 8 brinquedos é de 0.0075, este é um evento que vai acontecer muito raramente, ou seja, em 400 caixas é possível que aconteça em 3 delas.

(c) Com base em pX(x), estime o valor esperado, a variância e o desvio padrão de X.

Tem-se que o valor esperado seja igual a $\sum_{n=0}^{16} x_n \, p(x_n)$, pelo que se recorreu a operações matriciais para calcular esse valor. Com a expressão _xi_ .* _px_, multiplicou-se cada probabilidade de ter _xi_ brinquedos com defeitos com _xi_, após esta operação usa-se a função soma para somar todas as células dando o resultado de 0.1339 para o valor expectado.

```
E = sum(xi .* px)                    % valor esperado, tem-se que o valor esperado seja igual ao somatório de xi*p(xi)
```

Figura 21: Cálculo do valor esperado

Sabe-se que o valor da variância seja igual a $\sum_{n=0}^{8} [x_n - E(X)]^2 \, p(x_n)$. Novamente recorreu-se a operações matriciais, subtraiu-se o valor expectado à matriz <u>*xi*</u>, na qual se obteve o quadrado de cada célula usando ".^2" por fim multiplicou-se os valores de <u>*xi*</u> aos respetivos valores da probabilidade, para chegar finalmente ao valor de 0.1316.

```
var = sum((xi - E) .^2 .* px)        % variancia, tem-se que o valor da variancia seja igual ao somatório de ((xi-E(X))^2)*p(xi)
```

Figura 22: Cálculo da variância

Por fim para calcular o desvio padrão usou-se a fórmula: $\sqrt{var}$, através da função MATLAB *sqrt,* obtendo o valor de 0.3628 para o desvio padrão.

```
devi = sqrt(var);                    % desvio padrão, este é calculado fazendo a raiz quadrada da variancia
```

Figura 23: Cálculo do desvio padrão

(d) Repita as questões 3(a), 3(b) e 3(c), mas agora considerando n = 16 brinquedos. Compare todos os resultados com os obtidos anteriormente (com n = 8 brinquedos) e justifique as diferenças.

(a) Estime por simulação a função massa de probabilidade pX(x) de X quando n = 8 brinquedos e desenhe-a num gráfico stem.

```
N = 1e5;                             % numero de experiencias
n = 16;                              % numero de brinquedos por caixa
experiencias = rand(n,N);            % fabrico de peças p1
p1  = experiencias < 0.002;          % peças p1 com defeito
experiencias = rand(n,N);            % fabrico de peças p2
p2  = experiencias < 0.005;          % peças p2 com defeito
experiencias = rand(n,N);            % processo de montagem pa
pa  = experiencias < 0.01;           % defeitos na montagem pa com defeito
ndefeitos = p1 + p2 + pa;            % numero de defeitos por brinquedo
defeitos = ndefeitos > 0;            % brinquedos com defeito
nbrinquedos = sum(defeitos);         % numero de brinquedos com defeito
xi = (0:8);                          % conjunto dos valores possiveis de serem assumidos
px = zeros(1,9);                     % probabilidade associada a cada um dos valores possiveis de serem assumidos
```

Figura 24a: For loop e switch.

```
for val = nbrinquedos                          % iteração pelos valores obtidos por simulação
    switch(val)                                % filtração dos valores
    case 0                                     % caso a caixa tenha 0 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 0 brinquedos com defeito
    case 1                                     % caso a caixa tenha 1 brinquedo com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 1 brinquedo com defeito
    case 2                                     % caso a caixa tenha 2 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 2 brinquedos com defeito
    case 3                                     % caso a caixa tenha 3 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 3 brinquedos com defeito
    case 4                                     % caso a caixa tenha 4 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 4 brinquedos com defeito
    case 5                                     % caso a caixa tenha 5 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 5 brinquedos com defeito
    case 6                                     % caso a caixa tenha 6 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 6 brinquedos com defeito
    case 7                                     % caso a caixa tenha 7 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 7 brinquedos com defeito
    case 8                                     % caso a caixa tenha 8 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 8 brinquedos com defeito
    case 9                                     % caso a caixa tenha 0 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 0 brinquedos com defeito
    case 10                                    % caso a caixa tenha 10 brinquedo com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 10 brinquedos com defeito
    case 11                                    % caso a caixa tenha 11 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 11 brinquedos com defeito
    case 12                                    % caso a caixa tenha 12 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 12 brinquedos com defeito
    case 13                                    % caso a caixa tenha 13 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 13 brinquedos com defeito
    case 14                                    % caso a caixa tenha 14 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 14 brinquedos com defeito
    case 15                                    % caso a caixa tenha 15 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 15 brinquedos com defeito
    case 16                                    % caso a caixa tenha 16 brinquedos com defeito
        px(val+1) = px(val+1) + 1;             % contar numero de caixas com 16 brinquedos com defeito
    otherwise                                  % verificação de erro
        fprintf("Ocorreu um erro\n");          % aviso de erro
    end                                        %
end                                            %
```

Figura 24b: For loop e switch

Como é possível observar, a totalidade do código usado para responder a questão 3 a) para $n = 8$, tendo sido apenas acrescentados mais "*cases*" para filtrar a possibilidade de uma caixa ter mais que 8 insucessos. Por esta razão também se mudou a dimensão das matrizes xi e px, para suportarem as novas ocorrências. O código acima apresentando não será o mais eficiente contudo para efeitos de clareza foi este que foi usado.

De qualquer das maneiras outra implementação possível para a solução deste exercício seria:

```
for val = nbrinquedos                    %
    px(val + 1) = px(val + 1) + 1;       %
end                                      %
```

Figura 25: For loop alternativo

Em que cada iteração val tomaria um valor entre 0 e 16, correspondente ao número de brinquedos com defeito, contudo devido à forma de implementação de matrizes na linguagem de programação utilizada é necessário acrescentar o valor 1.

Logo *px(1)* deve ser igual a *P(B)* e o $\sum_{n=2}^{17} px(n) = probA$, tal como pode ser demonstrado pelos resultados obtidos nas figuras seguintes.

```
px(1)                                        % probabilidade obtida por simulação de não existirem brinquedos defeitoosos numa caixa com n brinquedos
```

Figura 26: Probabilidade do acontecimento B para n = 16

Desta linha de código resulta o valor de 0.7600 que é compatível com a probabilidade do acontecimento para *n* = *16* tal como é possível verificar na figura 6.

(b) Com base em pX(x), calcule a probabilidade de X >= 2.

```
px2 = sum(px(3:9));                          % probabilidade obtida por simulação
```

Figura 27: Probabilidade de existir pelo menos 2 brinquedos com defeito

Para o cálculo de *pX(x)* para *X* >=2, somou-se apenas os valores de probabilidade em que existe pelo menos dois brinquedos com defeito obtendo-se o resultado 0.0291.

Tal como seria de esperar esta probabilidade aumenta, sendo agora mais provável que pelo menos 2 brinquedos tenham defeito dentro da mesma caixa, uma vez que passamos a ter um maior universo de amostras por caixa.

(c) Com base em pX(x), estime o valor esperado, a variância e o desvio padrão de X.

Tem-se que o valor esperado seja igual a $\sum_{n=0}^{16} x_n\, p(x_n)$, pelo que se recorreu a operações matriciais para calcular esse valor. Com a expressão *xi .* px*, multiplicou-se cada probabilidade de ter *xi* brinquedos com defeitos com *xi*, após esta operação usa-se a função *sum()* para somar todas as células dando o resultado de 0.2725 para o valor expectado.

```
E = sum(xi .* px)                            % valor esperado, tem-se que o valor esperado seja igual ao somatório de xi*p(xi)
```

Figura 28: Cálculo do valor esperado

Sabe-se que o valor da variância seja igual a $\sum_{n=0}^{8} [x_n - E(X)]^2\, p(x_n)$. Novamente recorreu-se a operações matriciais, subtraiu-se o valor expectado à matriz *xi*, na qual se obteve o quadrado de cada célula usando ".^2" por fim multiplicou-se os valores de *xi* aos respetivos valores da probabilidade, para chegar finalmente ao valor de 0.2663.

```
var = sum((xi - E) .^2 .* px)                % variancia, tem-se que o valor da variancia seja igual ao somatório de ((xi-E(X))^2)*p(xi)
```

Figura 29: Cálculo da variância

Por fim para calcular o desvio padrão usou-se a fórmula: $\sqrt{var}$, através da função MATLAB *sqrt(),* obtendo o valor de 0.5160 para o desvio padrão.

```
devi = sqrt(var);                            % desvio padrão, este é calculado fazendo a raiz quadrada da variancia
```

Figura 30: Cálculo do desvio padrão

Uma vez que todos os brinquedos têm igual probabilidade de terem defeito, tal como todas as faces de um dado equilibrado tem igual probabilidade de ficarem voltadas para cima, conclui-se que irá haver uma maior dispersão do número de brinquedos com defeito, já que para _n_ = *16* a variância e o desvio padrão são ligeiramente maiores dos valores obtidos para _n_ = *8*, o valor do centro de gravidade da função massa também irá aumentar.

4. Suponha agora que a empresa pretende comercializar os brinquedos em caixas de n = 20 brinquedos garantindo que a probabilidade de uma caixa comercializada não ter brinquedos com defeito seja de pelo menos 90%.

Para atingir este objetivo, o processo de montagem foi melhorado reduzindo pa para 0,1% e foi implementado um processo de garantia de qualidade da seguinte forma: uma amostra de m brinquedos (com 1 ≤ m < 20) e selecionada de cada caixa para teste; a caixa não é comercializada se pelo menos um dos brinquedos selecionados estiver com defeito, ou é comercializada caso contrário.

Para a resolução do problema proposto recorreu-se mais uma vez ao método de simulação previamente usado. Para este problema tivesse de mudar certos parâmetros do método usado previamente. Os parâmetros que receberam mudança foram _n_ pois as caixas albergam 20 brinquedos em vez de 8 tendo assim _n_ o valor de 20 agora e o valor de pa que previamente era uma matriz que era baseada na probabilidade de _pa_ ser igual a 1% e agora é baseada em este ser 0.1%.

O método de simulação do acontecimento de defeitos nas caixas é assim feito pelo código:

```
N = 1e5;                                % numero de experiencias
n = 20;                                 % numero de brinquedos por caixa
experiencias = rand(n,N);               % fabrico de peças p1
p1  = experiencias < 0.002;             % peças p1 com defeito
experiencias = rand(n,N);               % fabrico de peças p2
p2  = experiencias < 0.005;             % peças p2 com defeito
experiencias = rand(n,N);               % processo de montagem pa
pa  = experiencias < 0.001;             % defeitos na montagem pa com defeito
ndefeitos = p1 + p2 + pa;               % numero de defeitos por brinquedo
defeitos = ndefeitos > 0;               % brinquedos com defeito
```

Figura 31: Simulador do fabrico de brinquedos

(a) Estime por simulação a probabilidade de uma caixa ser comercializada quando o processo de garantia de qualidade é implementado com m = 1.

Para simular as amostras de brinquedos retiradas por caixa para verificação de defeito começou-se por se definir uma variável representativa do número de amostras a retirar por caixa, sendo o nome desta variável _m_.

```
m = 1;                                  % numero de amostras retiradas por caixa
```

Figura 32: Variável m

As amostras são retiradas aleatoriamente por caixa, tendo isto em conta uma maneira de se garantir isto na simulação é recorrer-se á função "*randperm(n)*", esta função cria uma matriz *1x(n-1)* cujos valores em cada célula é um número entre 1 a *n*-1 numa posição indexada aleatória. Com este método é nos assim possível retirar um brinquedo de uma caixa numa posição aleatória.

```
ord = randperm(n-1);                          % ordem random de retirada das m amostras
```

Figura 33: Uso da função randperm()

Após retirar-se as amostras de cada caixa estas serão guardadas numa matriz *1xN* onde cada célula é representativa da caixa com o mesmo índice da matriz *defeitos*, inicialmente esta matriz é preenchida com zeros em cada célula e posteriormente inserida com as amostras.

A inserção das amostras é feita através de dois ciclos *for*, o primeiro ciclo *for* é o número de vezes que se quer retirar amostras em cada caixa, mais concretamente é a variável *m*. O segundo ciclo *for* é para se iterar pelo número de caixas a serem simuladas sendo neste caso representadas pelo valor *N*. Consegue-se assim retirar de cada caixa e guardar na matrix *amostras* a amostra da matriz *defeitos* na posição aleatória dentro da caixa.

```
for i = 1:m                                      % iteração pelo numero de amostras a retirar por caixa
    for c = 1:N                                  % iteração pelas caixas
        amostras(c) = amostras(c) + defeitos(ord(i),c); % retiro de uma amostra random por caixa e guardar seu valor
    end                                          %
end                                              %
```

Figura 34: Ciclo for utilizado

Após a simulação de retirar *m* amostras de cada caixa verifica-se quais das caixas têm pelo menos um brinquedo defeituoso. Recorrendo-se a uma comparação da matriz *amostras* com o valor 0 e verificando-se célula a célula se o valor é superior a 0 e caso seja essa célula fica com valor 1 e caso contrário fica com valor 0.

```
amostras = amostras > 0;                      % caixas com pelo menos um brinquedo com defeito
```

Figura 35: Comparação da matriz amostras

O cálculo da probabilidade de uma caixa ser comercializada sendo que para esta ser comercializada não pode ter um brinquedo defeituoso por caixa, foi feito através do cálculo da probabilidade de as caixas não serem comercializadas. Como a probabilidade de serem comercializadas é igual a 1 menos a probabilidade destas não serem, calculou-se então a probabilidade de não serem recorrendo-se à soma da matriz *amostras* que e dividindo pelo número de experiências realizadas *N* obtendo-se assim a probabilidade das caixas não serem comercializadas.

Para a probabilidade de as caixas serem comercializadas simplesmente subtraiu-se a 1 a probabilidade de não serem obtendo-se assim a probabilidade pretendida.

```
prob = 1 - (sum(amostras)/N)                  % probabilidade das caixas serem comercializadas
```

Figura 36: Cálculo da probabilidade de as caixas serem comercializadas

Sendo a probabilidade obtida de uma caixa ser comercializada por método de simulação ≈ 0.9922 ou ≈ 99.22%.

Concluído assim que retirando-se uma amostra por caixa composta por 20 brinquedos e verificando se tem defeito resulta em que ≈ 99.22% das caixas são comercializadas.

(b) Estime por simulação o menor valor de m necessário para atingir o objectivo desejado.

Pretende-se agora averiguar qual é o número de amostras a serem retiradas por caixa (*m*)para se garantir que a probabilidade de comercializar uma caixa que não tenha brinquedos defeituosos seja pelo menos 90%, tendo em conta que *m* terá que ser compreendido entre 1 e 19 inclusivé. Para simular as caixas a serem avaliadas recorreu-se mais uma vez ao método anteriormente descrito.

```
N = 1e5;                                 % numero de experiencias
n = 20;                                  % numero de brinquedos por caixa
experiencias = rand(n,N);                % fabrico de peças p1
p1  = experiencias < 0.002;              % peças p1 com defeito
experiencias = rand(n,N);                % fabrico de peças p2
p2  = experiencias < 0.005;              % peças p2 com defeito
experiencias = rand(n,N);                % processo de montagem pa
pa  = experiencias < 0.001;              % defeitos na montagem pa com defeito
ndefeitos = p1 + p2 + pa;                % numero de defeitos por brinquedo
defeitos = ndefeitos > 0;                % brinquedos com defeito
caixas = sum(defeitos)>0;                % caixas defeitoosas
```

Figura 36: Simulador do fabrico de brinquedos

Como se pretende avaliar o *m* mínimo para se conseguir atingir o objetivo pretendido, usou-se um ciclo for para se verificar *m* a *m* de entre os valores possíveis de *m* o mínimo pretendido.

```
M = 19;                                  % numero de amostras que poderão ser retirados
for m = 1: M                             % numero de amostras retiradas por caixa
```

Figura 37: Iteração de m a m

De maneira semelhante à descrita previamente retirou-se aleatoriamente *m* amostras e guardou-se na matrix *amostras* os resultados obtidos por caixa.

```
M = 19;                                              % numero de amostras que poderão ser retirados
for m = 1: M                                         % numero de amostras retiradas por caixa
    amostras = zeros(1,N);                           % amostras
    ord = randperm(19);                              % ordem random de retirada das m amostras
    for i = 1:m                                      % iteração pelo numero de amostras a retirar por caixa
        for c = 1:N                                  % iteração pelas caixas
            amostras(c)=amostras(c)+defeitos(ord(i),c); % retiro de uma amostra random por caixa e guardar seu valor
        end                                          %
    end                                              %
    amostras = amostras > 0;                         % caixas com pelo menos um brinquedo com defeito apanhadas pelo processo de garantia de qualidade
```

Figura 38: Ciclo for utilizado

Obtendo-se assim através da verificação celula a celula se a matriz <u>*amostras*</u> é superior a zero as caixas que possuem brinquedos defeituosos e que o nosso método de amostragem conseguiu apanhar.

Tenta-se agora com os resultados obtidos averiguar quais as caixas que escaparam ao nosso método de verificação através da amostragem. É verificado isto assim através da matriz <u>*escaparam*</u> que é de tamanho *1x<u>N</u>* e cujos valores são representativos da subtração célula a célula da matriz caixas ,que apresenta as caixas com defeitos através de 1 caso pelo menos um brinquedo com defeito e 0 caso não tenha, com a matriz <u>*amostras*</u>. Obtendo-se assim uma matriz escaparam cujos valores são 1 se existirem caixas com pelo menos um brinquedo defeituoso e que a nossa verificação não detetou e 0 caso a caixa não tenha pelo menos um brinquedo defeituoso ou que a nossa verificação detetou.

```
escaparam = caixas - amostras;                % caixas com defeitos que escaparam ao processo de garantia de qualidade
```

Figura 39: Matriz escaparam

Para cálculo da probabilidade das caixas serem comercializadas com nenhum brinquedo defeituoso calculou-se primeiro a probabilidade das caixas serem comercializadas com pelo menos um brinquedo defeituoso e subtrai-se a 1 a probabilidade obtida, obtendo-se assim a probabilidade pretendida.

O cálculo da probabilidade das caixas serem comercializadas com pelo menos um brinquedo defeituoso foi feito através da soma de todas as células da matriz <u>*escaparam*</u> obtendo-se assim o número de caixas com pelo menos um brinquedo defeituoso e dividiu-se pela diferença entre o número de experiências realizadas <u>*N*</u> e a soma de todas as células da matriz <u>*amostras*</u> tendo-se assim o número de caixas que não foram retiradas devido a confirmação de brinquedos defeituosos nas caixas, pelo processo de amostragem de <u>*m*</u> amostras.

Tem-se assim após a subtração de 1 à probabilidade calculada, a probabilidade pretendida <u>*prob*</u>.

```
prob = 1 - (sum(escaparam)/(N-sum(amostras)));     % probabilidade das caixas serem comercializadas sem defeitos
```

Figura 16: Cálculo da probabilidade das caixas serem comercializadas sem defeito

Agora basta verificar se a probabilidade obtida verifica os parâmetros pretendidos, isto é se esta é pelo menos 90%. Para averiguação deste parâmetro usou-se um if statement no qual caso a probabilidade obtida prob seja maior ou igual a <u>*90/100*</u> a variável <u>*result*</u> fica com o valor <u>*m*</u> a ser executado no respectivo momento e acaba o *ciclo* for com um *break*.

```
if prob >= 90/100                    % verificação se a probabilidade de comercialização da caixa é pelo menos 90/100
    result = m                       % valor minimo para garantir que a probabilidade de comercialização da caixa é pelo menos 90/100
    break                            %
end                                  %
```

Figura 16: Verificação da probabilidade obtida

Obtendo-se assim que o número mínimo de amostras a serem retiradas por caixa (<u>*m*</u>) para se garantir que a probabilidade de comercializar uma caixa que não tenha brinquedos defeituosos seja pelo menos 90%, tendo em conta que <u>*m*</u> terá que ser compreendido entre 1 e 19 inclusivé, é 7 amostras.