Métodos Probabilísticos para a Engenharia Informática

Relatório PL02

2020/2021

João Amaral, nº98373

Bruno Lemos, nº98221

Departamento de Eletrónica, Telecomunicações e Informática

Exercício 1

Segundo o enunciado da secção de avaliação, uma empresa fabricante de brinquedos compostos e divide o processo de produção em 3 fases independentes. Para isso, a componente 1 do brinquedo é produzida separadamente da componente 2, sendo posteriormente montadas uma à outra. Na faze de produção, a componente 1 do brinquedo produz 0,2% brinquedos com defeito, a componente 2 produz 0,5% brinquedos com defeito e, por fim, o processo de montagem produz 1% de brinquedos com defeito.

Tendo isto em conta, o evento “A – uma caixa de brinquedos tem pelo menos 1 brinquedo com defeito”, sabendo que cada caixa tem 8 brinquedos. Para determinar a probabilidade deste evento:

```
%#ok<*NOPTS>
exp = 1e5;
n = 8;

comp1 = randi([1,1000], n, exp) < 3; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 1 em 8 brinquedos em milhares de caixas

comp2 = randi([1,1000], n, exp) < 6; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 2 em 8 brinquedos em milhares de caixas

montagem = randi([1,100], n, exp) < 2;  %põe a 1 sempre que pelo menos um brinquedo tiver defeito na montagem em 8 brinquedos em milhares de caixas

total = comp1 | comp2 | montagem;   %une todas as partes e e põe a 1 sempre que um brinquedo tiver pelo menos um defeito (seja em qual das partes foi)
sucessos = sum(total) >= 1;     %conta o numero de brinquedos com defeito e põe a 1 sempre que pelo menos um(>=1) brinquedo tenha defeito em numa caixa de 8 brinquedos
defeituosas = sum(sucessos);    %soma todas as vezes que pelo menos um brinquedo tenha defeito em todas as caixas
probA = defeituosas/exp   %divide o nummero de pelo menos um brinquedo defeituoso (em cada caixa) por todas as caixas produzidas, obtendo a probabilidade de
                                %pelo menos um brinquedo ser defeituoso numa caixa com 8
```

```
mmand Window
>> ex1_a

probA =

    0.1276
```

Como é explicado nos comentários, usando matrizes lógicas para obter números aleatórios (brinquedos), onde o valor 1 corresponde a um brinquedo defeituoso[1], é possível ter uma ideia de quantos brinquedos estão defeituosos numa caixa de 8 brinquedos, em milhares de caixas. Usando o operador lógico “OU” (‘|’), é possível reunir os três acontecimentos independentes, onde o valor 1 é obtido sempre que um brinquedo seja defeituoso na componente 1, componente 2 ou montagem, ou mesmo em várias componentes o mesmo tempo.

[1] Usando a função randi[1,1000], obtemos valores gerados aleatoriamente entre 1 e 1000. Como a probabilidade de, por exemplo, a componente 1 ter um brinquedo defeituoso é 0.2%, significa que 2 em cada 1000 brinquedos têm defeito e, por isso, põe-se a 1 sempre que o valor gerado aleatoriamente for 1 ou 2

Deste modo, obtemos uma matriz onde cada coluna (caixa) tem 8 linhas (brinquedos) correspondentes, onde cada valor pode ser 1 (defeituoso) ou 0 (não defeituoso). Usando a função 'sum', é possível obter um vetor que contêm o número de brinquedos defeituosos em cada caixa, pondo a 1 sempre que uma caixa tiver pelo menos um (>=1) brinquedo defeituoso. No fim, somamos todos os valores desse vetor e dividimos pelo número de caixas produzidas (colunas), obtendo assim a probabilidade associada ao evento A, sendo esta à volta dos ~12.8%.

Na alínea b) do exercício 1, é pedido o número médio de brinquedos defeituosos apensa devido ao processo de montagem quando ocorre o evento A. Para isso, idealmente seria criar uma matriz que pusesse a 1 sempre que um brinquedo não fosse defeituoso na componente 1 e 2, mas que fosse defeituoso devido ao processo de montagem. Para isso:

```
%#ok<*NOPTS>
exp = 1e6;
n = 8;

comp1 = randi([1,1000], n, exp) < 3;

comp2 = randi([1,1000], n, exp) < 6;

montagem = randi([1,100], n, exp) < 2;

%so pela montagem

total_montagem = 1-comp1 & 1-comp2 & montagem;  %põe a 1 sempre que um brinquedo não seja defeituoso em nenhuma das componentes mas que falhou na montagem
sucessos_montagem = sum(total_montagem)>=1;      %soma todas as vezes que pelo meno sum brinquedo sai defeituoso devido apenas à montagem
defeituosas_montagem = sum(sucessos_montagem);   %soma o numero total de brinquedos defeituosos em todas as caixas

%acontecimento A (universo)

totalA = comp1 | comp2 | montagem;
sucessosA = sum(totalA) >= 1;
defeituosasA = sum(sucessosA);

nMedio = defeituosas_montagem/defeituosasA
```

```
mmand Window
>> ex1_b

nMedio =

    0.6010
```

Desta vez, para obtermos a matriz pedida, é necessário usar o operador lógico "E" ('&'), que faz a interseção das matrizes onde está a 1 os brinquedos que não têm defeito nem na componente 1 nem na componente 2 (1 – defeito) mas que têm defeito apenas devido ao processo de montagem.

Somando o número de vezes em que pelo menos um brinquedo é defeituoso em cada caixa e em todas as caixas, obtemos o número de brinquedos total que falharam apenas no processo de montagem. Dividindo esse valor pelo número vezes que pelo menos um brinquedo falhou numa caixa, em qualquer um dos processos, obtemos o valor médio desejado, equivalente a ~0.6 brinquedos.

Exercício 2

No exercício 2, o evento a considerar é "B – uma caixa de brinquedos não tem brinquedos com defeito". Ora, o acontecimento complementar ao A é exatamente o B, e, teoricamente, conseguimos calcular o valor deste evento apenas fazendo probB = 1 – probA.

a) No entanto, para obter a probabilidade por simulação:

```
%#ok<*NOPTS>
exp = 1e5;
n = 8;

comp1 = randi([1,1000], n, exp) >= 3;   %põe a 1 sempre que pelo um brinquedo não tiver defeito na componente 1 em 8 brinquedos em milhares de caixas

comp2 = randi([1,1000], n, exp) >= 6;   %põe a 1 sempre que pelo um brinquedo não tiver defeito na componente 2 em 8 brinquedos em milhares de caixas

montagem = randi([1,100], n, exp) >= 2; %põe a 1 sempre que pelo um brinquedo não tiver defeito na montagem em 8 brinquedos em milhares de caixas

total = comp1 & comp2 & montagem;   %faz a interceção de todas as partes e põe a 1 todos os brinquedos que foram produzidos sem defeito
sucessos = sum(total) == n;         %cria um vetor com 1 sempre que o numero de brinquedos produzidos sem defeito seja igual ao numero de brinquedos por caixa
corretas = sum(sucessos);           %soma todas as vezes em que isso aconteceu
probCorretas = corretas/exp         %divide todos esses acontecimentos pelo numero de caixas feitas, para obter a probabilidade
```

```
mmand Window
>> ex2_a

probCorretas =

    0.8722
```

A alteração desta simulação para a anterior é vista no processo de geração aleatória de valores na matriz, onde desta vez se considera que o valor 1 é atribuído sempre que um brinquedo não for defeituoso. Na matriz total, que envolve os 3 processos de construção de um brinquedo, é usada a interseção, pois queremos garantir que cada brinquedo não tem defeito na componente 1, nem na componente 2 e nem na montagem.

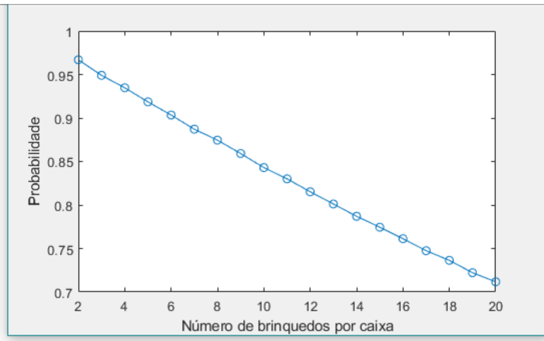
Semelhante ao exercício anterior, é feita a soma dos brinquedos sem defeito em cada coluna (caixa) e é atribuído o valor 1 sempre que o número de brinquedos sem defeito for o número de brinquedos da caixa, neste caso 8.

No fim, apenas dividimos o número de caixas com brinquedos sem defeito nenhum pelo número de caixas total, chegando ao valor ~87.2%.

b) O valor teórico do evento corresponde ao complementar do evento A, neste caso seria probB = 100% – 12.8% = 87.2%, o que coincide perfeitamente com o valor calculado por simulação.

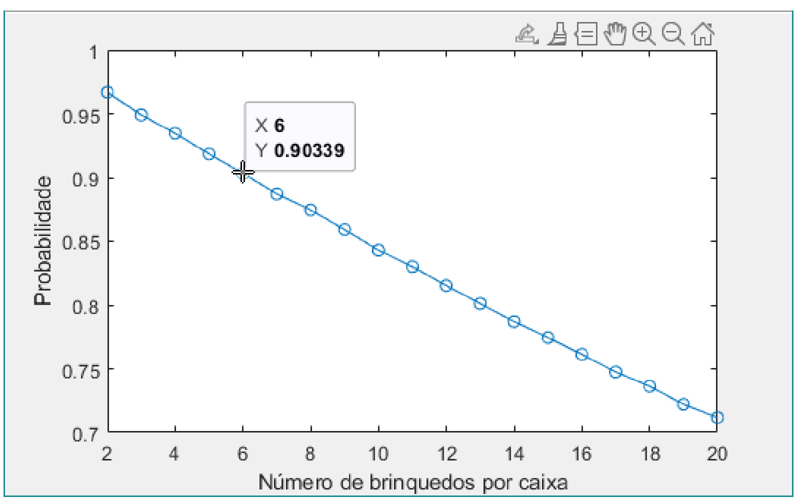
c) Nesta alínea, é necessário desenhar um gráfico *plot* da probabilidade do evento B, onde o número de brinquedos por caixa varia entre 2 e 20 brinquedos:

```
n = 8;
probs = [];
nums = [];

for k = 2:20
    probK = calcula(k, exp);
    probs = [probs probK];
    nums = [nums k];
end

probs
nums

plot(nums, probs, '-o')
xlabel("Número de brinquedos por caixa")
ylabel("Probabilidade")


function [probCorretas] = calcula(n, exp)
     comp1 = randi([1,1000], n, exp) >= 3;          %coloca a 1 sempre que a componente 1 for bem construida

     comp2 = randi([1,1000], n, exp) >= 6;          %coloca a 1 sempre que a componente 2 for bem construida

     montagem = randi([1,100], n, exp) >= 2;        %coloca a 1 sempre que a montagem for bem construida

     total = comp1 & comp2 & montagem;              %interseta os tres acontecimentos para termos a certeza que o brinquedo é bem construido nas tres fases
     sucessos = sum(total) == n;                    %põe a 1 sempre que o numero de brinquedos sem defeitos for igual ao total de brinquedos construidos
     corretas = sum(sucessos);                      %soma todos esses acontecimentos
     probCorretas = corretas/exp;                   %divide esses acontecimentos pelo numero de experiencias feitas para obter a probabilidade do sucesso
end
```



Ao analisarmos o gráfico, podemos observar que quando menor for o número de brinquedos numa caixa, maior é a probabilidade dessa caixa não ter brinquedos defeituosos (~ 96.7%, m=2), pois quantos menos brinquedos produzidos forem para cada caixa, menor é a probabilidade de essa caixa ter pelo menos um brinquedo defeituoso. À medida que vamos aumentando o número de brinquedos por cada caixa, a probabilidade de nenhum brinquedo ter defeito já é menor (~71.1%, m=20), pois em cada caixa, a probabilidade de pelo menos um brinquedo ser defeituoso aumenta significativamente, dado que cada caixa tem consideravelmente mais brinquedos.

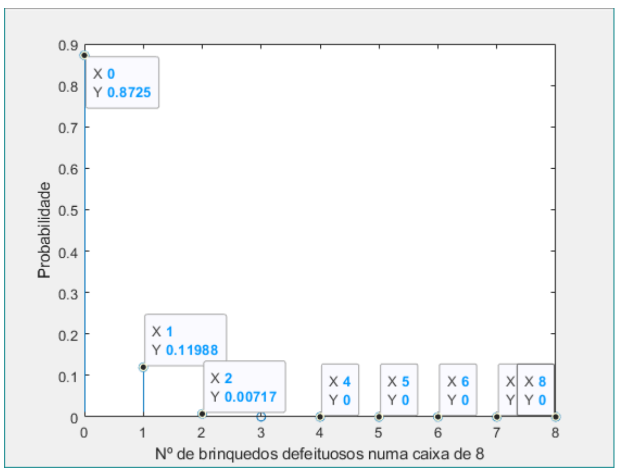
d) Na última alínea, analisando o gráfico com detalhe, a capacidade máxima da caixa para que a probabilidade de cada caixa não ter brinquedos defeituosos é m=6, onde a sua probabilidade respetiva é ~90.2%:



Exercício 3

a) Considerando agora a variável aleatória X, que representa o número de brinquedos defeituosos numa caixa, a sua respetiva função massa de probabilidade quando n = 8 é :

```
%#ok<*NOPTS>
%#ok<*AGROW>
exp = 1e5;
n = 0:8;
caixa = 8;
px = [];

for i = 1:length(n)
    comp1 = randi([1,1000], caixa, exp) < 3; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 1 em 8 brinquedos em milhares de caixas

    comp2 = randi([1,1000], caixa, exp) < 6; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 2 em 8 brinquedos em milhares de caixas

    montagem = randi([1,100], caixa, exp) < 2;  %põe a 1 sempre que pelo menos um brinquedo tiver defeito na montagem em 8 brinquedos em milhares de caixas

    total = comp1 | comp2 | montagem;   %une todas as partes e e põe a 1 sempre que um brinquedo tiver pelo menos um defeito (seja em qual das partes foi)
    sucessos = sum(total) == n(i);     %conta o numero de brinquedos com defeito e põe a 1 sempre o nº de brinquedos com defeito for igual ao desejado
    defeituosas = sum(sucessos);    %soma todas essas vezes em todas as caixas
    probDefeito = defeituosas/exp;  %divide o nmumero de pelo menos um brinquedo defeituoso por todas as caixas produzidas
    px = [px probDefeito];
end

n
px
stem(n,px)
xlabel("Nº de brinquedos defeituosos numa caixa de 8")
ylabel("Probabilidade")
```



Como podemos analisar, os resultados obtidos vão de acordo com os exercícios anteriores. Por exemplo, a probabilidade de numa caixa de 8 brinquedos não haver brinquedos com defeito(0) é de ~87.2%, o que coincide perfeitamente com o resultado obtido na questão 2(a). Também é possível observar que a probabilidade de pelo menos um brinquedo ser defeituoso numa caixa de 8 brinquedos (somatório de todas as probabilidades para valores de xi >=1) é aproximadamente ~12.7%, o que é próximo o suficiente do valor obtido na questão 1(a).

b) Com base px(x), a probabilidade de X >= 2 é aproximadamente ~0.77%, resultado obtido através da soma de todas as probabilidades para valores de xi >= 2 :

```
(...)
pM2 = 0;

for i =3:length(n)
    pM2 = pM2 + px(i);          %soma todos os valores de probabilidade para xi>=2
end
px
pM2
```

mmand Window

```
>> ex3_b

px =

    0.8736    0.1193    0.0074    0.0003    0.0000         0         0         0         0


pM2 =

    0.0077
```

c) O valor esperado corresponde a $E[x] = \sum_i$ xi $*$ p(xi), onde cada valor da variável aleatória é multiplicado pela correspondente probabilidade. Já a variância, esta é calculada pela expressão $E[x] = \sum_i$ $[\mathrm{x(i)} - E[x]^2 * p(xi)$, onde o desvio padrão corresponde à raiz quadrada da variância:

```
(...)
ex = 0;
for i= 1:length(x)
    ex = ex + x(i)*px(i); %calcula o valor esperado do px
end

Valor_esperado = ex

varx = 0;
for i = 1:length(x)
   varx = varx + (((x(i)-ex)^2)* px(i)); %calcula a variancia do px
end
Variancia = varx
Desvio_padrao = sqrt(varx)  %calcula o desvio padrão de px
```

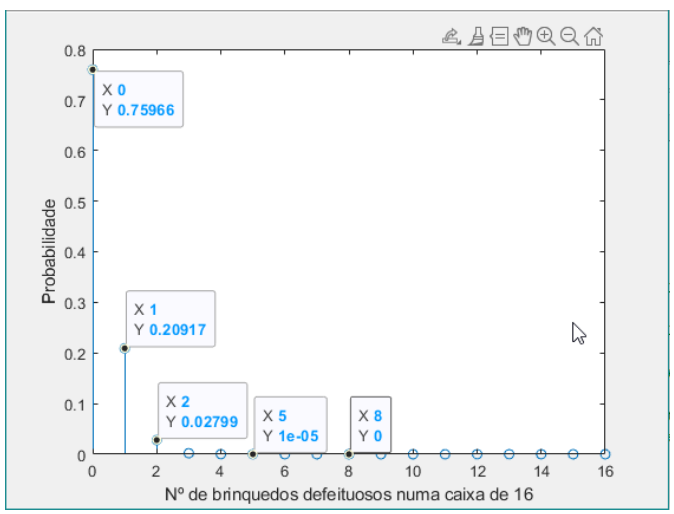
mmand Window

```
Valor_esperado =

    0.1356


Variancia =

    0.1329


Desvio_padrao =

    0.3646
```

d) Mudando agora o número de brinquedos por caixa para 16, os valores obtidos são relativamente diferentes em qualquer um dos aspetos:

```
exp = 1e5;
x = 0:16;
caixa = 16;
px = [];

for i = 1:length(x)
    comp1 = randi([1,1000], caixa, exp) < 3; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 1 em 16 brinquedos em milhares de caixas

    comp2 = randi([1,1000], caixa, exp) < 6; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 2 em 16 brinquedos em milhares de caixas

    montagem = randi([1,100], caixa, exp) < 2;  %põe a 1 sempre que pelo menos um brinquedo tiver defeito na montagem em 16 brinquedos em milhares de caixas

    total = comp1 | comp2 | montagem;   %une todas as partes e e põe a 1 sempre que um brinquedo tiver pelo menos um defeito (seja em qual das partes foi)
    sucessos = sum(total) == x(i);     %conta o numero de brinquedos com defeito e põe a 1 sempre o nº de brinquedos com defeito for igual ao desejado
    defeituosas = sum(sucessos);    %soma todas essas vezes em todas as caixas
    probDefeito = defeituosas/exp;  %divide o nmumero de pelo menos um brinquedo defeituoso por todas as caixas produzidas
    px = [px probDefeito];
end

stem(x,px)
xlabel("Nº de brinquedos defeituosos numa caixa de 16")
ylabel("Probabilidade")

ex = 0;
for i= 1:length(x)
    ex = ex + x(i)*px(i); %calcula o valor esperado
end

Valor_esperado = ex

varx = 0;
for i = 1:length(x)
   varx = varx + (((x(i)-ex)^2)* px(i));    %calcula a variancia
end
Variancia = varx
Desvio_padrao = sqrt(varx)  %calcula o desvio padrão
```

```
Valor_esperado =

    0.2718


Variancia =

    0.2679


Desvio_padrao =

    0.5176
```



É possível observar que tanto o valor esperado, a variância e o desvio padrão aumentaram, mas não uma quantidade qualquer. Mais especificamente, o valor esperado e a variância aumentaram para o dobro do valor correspondente a 8 brinquedos por caixa. Com isto, podemos concluir que aumentando para o dobro número de brinquedos por caixa, aumentamos também linearmente o valor esperado e a variância correspondentes. No entanto, os valores da função massa de probabilidade são diferentes, havendo uma diferença significativa nas probabilidades de cada valor.

Exercício 4

a) Mudando agora o processo de produção e implementado um processo de garantia de qualidade, a fase de montagem foi melhorada, tendo agora um erro apenas de 0.1%, e um método de garantia de qualidade que retira aleatoriamente um número determinado de brinquedos de uma caixa de 20 e verifica se têm defeito, comercializando a caixa se esta não apresentar nenhum brinquedo defeituoso nesse processo:

```
%#ok<*NOPTS>
exp = 1e5;
n = 20;
m = 1;

comp1 = randi([1,1000], n, exp) < 3; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 1 em 8 brinquedos em milhares de caixas

comp2 = randi([1,1000], n, exp) < 6; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 2 em 8 brinquedos em milhares de caixas

montagem = randi([1,1000], n, exp) < 2;  %põe a 1 sempre que pelo menos um brinquedo tiver defeito na montagem em 8 brinquedos em milhares de caixas

total = comp1 | comp2 | montagem;   %une todas as partes e e põe a 1 sempre que um brinquedo tiver pelo menos um defeito (seja em qual das partes foi)

sucessos = 0;

%Processo de garantia de qualidade

for i = 1:exp                          %ciclo que percorre todas as caixas produzidas
    vetor = randperm(20,m);            %cria um vetor de m indices ao calhas entre 1 e 20
    column = total(:,i);
    row = column';                     %pega em cada caixa e torna num vetor
    soma = 0;
    for k = 1:length(vetor)            %percorre todos os indices do vetor de indices random
        soma = soma + row(k);          %soma os sucessos
    end
    if(soma == 0)
        sucessos = sucessos + 1;       %somamos aos sucessos sempre que na amostra não houver nenhum brinquedo com defeito
    end
end

comercializadas_em_100000 = sucessos
prob = sucessos/exp
```

```
mmand Window
comercializadas_em_100000 =

       99199


prob =

    0.9920
```

Para isso, o processo de gestão de qualidade é implementado da seguinte maneira : primeiro, geramos um vetor aleatório de números (neste caso apenas um número pois m = 1) entre 1 e 20 sem repetição, usando a função Matlab *randperm*, que vão corresponder aos índices dos brinquedos que vamos verificar, de modo a garantir a máxima aleatoriedade na seleção de uma amostra de brinquedos.

Após isso, percorremos cada caixa (coluna que foi agora transformada numa linha) e somamos o valor atribuído ao brinquedo selecionado (1 se for defeituoso e 0 se não for), obtendo assim o número de brinquedos defeituosos na amostra selecionada. De seguida, verificamos se o número de brinquedos defeituosos é 0 e, se assim for, consideramos a caixa comercializada. No fim, dividimos o número de caixas comercializadas pelo número total de caixas produzidas, chegando o valor da probabilidade de uma caixa ser comercializada ~99.2%.

Trata-se de um valor muito alto pois se apenas verificarmos só um brinquedo numa caixa de 20, há uma grande chance na mesmo de um dos outros brinquedos da caixa ter defeito.

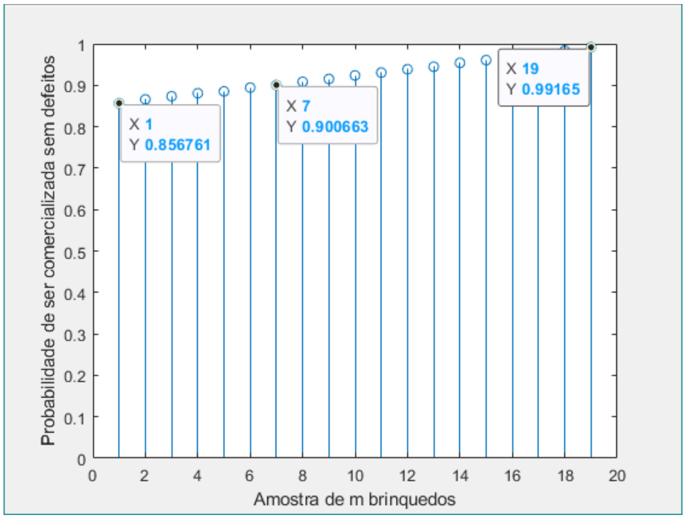
b) O objetivo desejado corresponde à garantia de que a probabilidade de uma caixa de 20 brinquedos comercializada não ter brinquedos com defeito seja de pelo menos 90%. Para isso, a melhor implementação de uma solução seria criar um gráfico para analisar todos os valores de m, sendo assim mais fácil de identificar o menor valor para cumprir o objetivo:

```
probsM = [];

for  k = 1:19
    prob = calcula(k, n, exp);
    probsM = [probsM prob];
end

m
probsM
stem(m,probsM)
ylabel("Probabilidade de ser comercializada sem defeitos")
xlabel("Amostra de m brinquedos")


function [probSemDefeito] = calcula(m,n, exp)
    comp1 = randi([1,1000], n, exp) < 3; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 1 em 8 brinquedos em milhares de caixas
    comp2 = randi([1,1000], n, exp) < 6; %põe a 1 sempre que pelo menos um brinquedo tiver defeito na componente 2 em 8 brinquedos em milhares de caixas
    montagem = randi([1,1000], n, exp) < 2;  %põe a 1 sempre que pelo menos um brinquedo tiver defeito na montagem em 8 brinquedos em milhares de caixas
    total = comp1 | comp2 | montagem;   %une todas as partes e e põe a 1 sempre que um brinquedo tiver pelo menos um defeito (seja em qual das partes foi)
    sucessos = [];
    for i = 1:exp                        %ciclo que percorre todas as caixas produzidas
       indices = randperm(20,m);         %cria um vetor de m indices ao calhas entre 1 e 20 para retirar da caixa o(s) brinquedo(s) que correspondem a esse(s) indice(s)
       column = total(:,i);
       row = column';                    %pega em cada caixa e torna num vetor
       soma = 0;
       for k = 1:length(indices)           %percorre todos os indices do vetor de indices random
          soma = soma + row(indices(k));         %soma se os brinquedos escolhidos têm defeito
       end
       if(soma > 0)
           sucessos = [sucessos 0];      %se a soma for > 0, entao houve defeito e por isso não é comercializada (0)
       else
           sucessos = [sucessos 1];      %se a soma for = 0, entao nao houve defeito e por isso é comercializada (1)
       end
    end                                  %o vetor sucessos contem 0 sempre que a caixa correspondete nao foi comercializada e 1 caso contrario
    comercializadas = sum(sucessos);     %somamos todos os valores do vetor sucessos que nos dá o numero de caixas que foram comercializadas
    semDefeitos = 0;                     %variavel usada para contar o numero de caixas comercializadas mas que têm >=1 brinquedo com defeito
    somaTotal = sum(total);              %vetor que contem o numero de brinquedos defeituosos por cada caixa
    for i = 1:exp
       if(sucessos(i)==1 && somaTotal(i) == 0)   %se a caixa foi comercializada e se a mesma caixa nao tiver brinquedos com defeito
           semDefeitos = semDefeitos + 1;       %incrementamos o numero de caixas comercializadas com defeito
       end
    end
    probSemDefeito = semDefeitos/comercializadas;
end
```

É importante ter em nota que é necessário contabilizar o número de caixas que foram comercializadas, mas que tinham pelo menos um brinquedo com defeito. Para isso, o vetor sucessos contem o valor 1 sempre que a caixa respetiva (coluna) for comercializada. Assim, basta verificar que se a caixa foi comercializada (sucessos(i)==1) e o número de brinquedos com defeito dessa caixa for 0 (somaTotal(i)==0), então considera-se que a caixa foi comercializada sem defeitos.

No fim, dividimos o número de caixas comercializadas sem defeito pelo número total de caixas produzidas, obtendo assim a probabilidade de uma caixa ser comercializada sem defeitos.

Para além disso diferença nesta alínea, é que percorremos todos os valores possíveis de ‘m’ e guardamos as suas respetivas probabilidades noutro vetor, para podermos criar um gráfico e observar mais facilmente os resultados da simulação:



Conclui-se então que o menor valor de m para que o processo de garantia de qualidade comercialize pelo menos 90% das caixas sem brinquedos com defeito é m = 7, onde a sua probabilidade é de ~90.1%. Ao analisarmos o gráfico, podemos observar que á medida que aumentamos o tamanho da nossa amostra, a probabilidade da caixa ser comercializada sem defeitos aumenta. Isto, pois ao recolhermos uma amostra maior, a probabilidade de encontrar pelo menos um brinquedo com defeito aumenta, reduzindo assim o número total de caixas comercializadas com defeito, aumentando assim a probabilidade da caixa ser comercializada sem nenhum brinquedo com defeito, daí uma probabilidade maior.