

Universidade do Minho

2º Semestre 2020/21


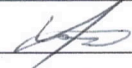
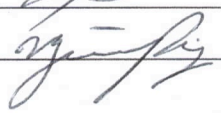
(MIEI, 3º Ano)

Modelos Estocásticos de Investigação Operacional

Trabalho Prático

(Problema de Gestão de Inventários)

Identificação do Grupo

<u>Número:</u>	<u>Nome completo:</u>	<u>Rubrica:</u>
A89513	Francisco Alves Andrade	
A89465	Paulo Silva Sousa	Paulo Sousa
A89444	Luís Filipe Cruz Sobral	
A89520	João Figueiredo Martins Leite dos Santos	

Data de entrega: 2021-04-26

Índice

1	Introdução	1
2	Dados do problema	1
3	Questões	2
3.1	Questão 1	2
3.2	Questão 2	4
3.3	Questão 3	6
3.3.1	Resultados	9
4	Conclusões	10
A	Anexo	11
A.1	Valores de Procura	11
B	Código da Aplicação	12

1 Introdução

No mundo do negócio a gestão do inventário é algo crucial e que não deve ser descartado. O inventário é "dinheiro disfarçado na forma de bens ou stock", e, como tal, deve ser protegido e gerido da maneira mais inteligente possível. Esta gestão recorre a práticas como a simulação e a utilização de situações hipotéticas que nos permitem decidir qual a melhor decisão a tomar em cada uma delas para melhor preservar a economia do negócio em questão.

Neste caso vai ser utilizada uma política (s, S) , que consiste na fusão das políticas de nível de encomenda (que obriga a realização de encomendas sempre um valor fixo assim que o stock estiver abaixo de um determinado valor) e a de ciclo de encomenda (que recorre à realização de encomendas periódicas sem o valor fixo para encomendar).

2 Dados do problema

Sendo o nosso maior número de aluno o número A89520, iremos determinar os dados do nosso problema a partir deste. Assim, concluímos que:

$$d_1 = 5, d_2 = 2, d_3 = 0$$

$$p_1 = 0.21 + \frac{5}{100} = 0.26$$

$$p_2 = 0.52 + \frac{2}{100} = 0.54$$

$$p_3 = 1 - 0.26 - 0.54 = 0.2$$

$$C_2 = 20 + 2 * 0 = 20$$

3 Questões

3.1 Questão 1

Estime analiticamente os valores dos parâmetros da política nível de encomenda que teriam sido mais adequados para o último ano (2020). Quanto é que a empresa poderia ter poupado em custos e ou evitado em quebras de stock, ao longo do último ano, se tivesse usado parâmetros mais racionais na sua política de gestão?

Através da análise dos dados fornecidos, utilizando ferramentas Excel (Average e Std-dev) obtivemos que $r = 462$ e $\sigma_r = 50$. Além disso, através da análise do enunciado temos que $C_1 = 0.345$ €/unidade/semana, $C_2 = 20$ €/encomenda em atraso e $C_3 = 1500$ €/encomenda.

De acordo com os dados fornecidos pelo sistema de encomendas empregue pelo Sr. Gervásio obtivemos o custo deste sistema da seguinte forma:

$$LT = 1 * p1 + 2 * p2 + 3 * p3 = 1 * 0.26 + 2 * 0.54 + 3 * 0.2 = 1.94 \text{ semanas}$$

$$\mu_{DDLT} = LT * r = 1.94 * 462 = 896.28 \text{ artigos}$$

$$\sigma_{LT}^2 = (1 - LT)^2 * p1 + (2 - LT)^2 * p2 + (3 - LT)^2 * p3 = 0.4564$$

$$\sigma_{DDLT} = \sqrt{LT * \sigma_r^2 + r^2 * \sigma_{LT}^2} = \sqrt{1.94 * 50^2 + 462^2 * 0.4564} = 319.79$$

$$S = 1200 \text{ artigos}$$

$$Z = \frac{S - \mu_{DDLT}}{\sigma_{DDLT}} = \frac{1200 - 896.28}{319.79} = 0.94975$$

$$Z = \frac{3N}{100} \Leftrightarrow N = \frac{100 * Z}{3} = \frac{0.94975 * 100}{3} \approx 32$$

$$E[DDLT > S] = 2^{\circ} \text{integral} * \sigma_{DDLT} = 0.086740 * 319.79 = 27.7386$$

$$\begin{aligned} CT &= C_1 \left(\frac{q^*}{2} + S - \mu_{DDLT} \right) + C_2 \frac{r}{q^*} E[DDLT > S] + C_3 \frac{r}{q^*} = \\ &= 0.345 * \left(\frac{1700}{2} + 1200 - 896.28 \right) + 20 * \frac{462}{1700} * 27.7386 + 1500 * \frac{462}{1700} = 956.45 \text{ €/semana} \end{aligned}$$

Assim prosseguimos à estimativa de parâmetros de um modelo de nível de encomenda de modo a reduzir o custo total.

$$q^* = \sqrt{\frac{2rC_3}{C_1}} = \sqrt{\frac{2 * 462 * 1500}{0.345}} = 2005 \text{ artigos}$$

$$P[DDLT > S] = \frac{C_1 * q}{C_2 * r} = \frac{0.345 * 2005}{20 * 462} = 0.07486$$

$$N = 48$$

$$E[DDLT > S] = 2^{\circ}integral * \sigma_{DDLT} = 0.031070 * 319.79 = 9.9384$$

$$q^* = \sqrt{\frac{2r(C_2 E[DDLT > S] + C_3)}{C_1}} = \sqrt{\frac{2 * 462 * (20 * 9.9384 + 1500)}{0.345}} = 2134 \text{ artigos}$$

$$P[DDLT > S] = \frac{C_1 * q}{C_2 * r} = \frac{0.345 * 2134}{20 * 462} = 0.07978$$

$$N = 47$$

$$E[DDLT > S] = 2^{\circ}integral * \sigma_{DDLT} = 0.033350 * 319.79 = 10.665$$

$$q^* = \sqrt{\frac{2r(C_2 E[DDLT > S] + C_3)}{C_1}} = \sqrt{\frac{2 * 462 * (20 * 10.665 + 1500)}{0.345}} = 2143 \text{ artigos}$$

$$P[DDLT > S] = \frac{C_1 * q}{C_2 * r} = \frac{0.345 * 2143}{20 * 462} = 0.08$$

$$N = 47 \Rightarrow \text{convergiu}$$

$$Z = \frac{3N}{100} = \frac{3 * 47}{100} = 1.41$$

$$S = \mu_{DDLT} + Z * \sigma_{DDLT} = 896.28 + 1.41 * 319.79 = 1347.1839 \text{ artigos}$$

$$\begin{aligned} CT &= C_1 \left(\frac{q^*}{2} + S - \mu_{DDLT} \right) + C_2 \frac{r}{q^*} E[DDLT > S] + C_3 \frac{r}{q^*} = \\ &= 0.345 * \left(\frac{2143}{2} + 1347.1839 - 896.28 \right) + 20 * \frac{462}{2143} * 10.665 + 1500 * \frac{462}{2143} = 894.59 \text{ €/semana} \end{aligned}$$

Assim, a diferença de custos entre os dois métodos é dada por:

$$\Delta_{custos} = 956.45 - 894.59 = 61.86 \text{ €/semana}$$

Desta forma concluímos que os parâmetros por nós estimados trariam benefícios ao Sr. Gervásio, reduzindo o custo, tornando a implementação de um modelo de nível de encomenda mais apetecível. A quantidade ótima de encomenda seria de 2143 artigos a cada 1,94 semanas, com um nível de encomenda de 1347,1839 artigos, com um custo semanal de 894,59 €.

3.2 Questão 2

Estime analiticamente os valores dos parâmetros da política (s , S) para o ano em curso (2021). Considere, para a média dos valores da procura semanal, uma estimativa que consiste na extrapolação do valor segundo a regressão linear dos valores médios homólogos verificados nos últimos anos.

Previmos os valores da procura semanal e do seu desvio padrão através da regressão linear. Utilizamos funcionalidades do Excel que nos permitiram definir a reta de regressão linear para a procura semanal e o seu desvio padrão. Como pontos base optamos por usar as procuras semanais dos anos anteriores, onde $X \in \{1, 2, 3\}$ e $Y \in \{r_{2018}, r_{2019}, r_{2020}\}$ para a regressão linear da procura semanal e $X \in \{1, 2, 3\}$ e $Y \in \{\sigma_{r2018}, \sigma_{r2019}, \sigma_{r2020}\}$ para o desvio padrão. Daqui retiramos as expressões das regressões lineares pretendidas.

Para r :

$$y = 62,73x + 271,64$$

e para o desvio padrão:

$$y = 7,54911x + 25,8325666666667$$

Para $x=4$, de modo a estimar os valores para 2021 obtivemos:

$$r = 523 \text{ artigos}$$

$$\sigma_r = 57$$

	2018	2019	2020
Media	336,16	393,52	461,62
Desvio Padrao	34,88573	37,92268	49,98395

	Coefficients
Intercept	271,64
X Variable 1	62,73

	Coefficients
Intercept	25,83256667
X Variable 1	7,54911

Figura 1: r e desvio padrão conhecidos

Figura 2: Regressão Linear de r

Figura 3: Regressão Linear de σ_r

Dados:

$$t = 4 \text{ semanas}$$

Ocorrência de uma situação de quebra a cada dois anos, no máximo.

$$p_1 = 0,26$$

$$p_2 = 0,54$$

$$p_3 = 0,2$$

$$C_1 = 0,345 \text{ €/artigos/semana}$$

$$C_2 = 20 \text{ €/encomenda em atraso}$$

$$C_3 = 1500 \text{ €/encomenda}$$

$$LT = 1,94 \text{ semanas}$$

$$\sigma_{LT}^2 = 0,4564$$

$$\mu_{DDPP} = r * (t + LT) = 523 * (4 + 1,94) = 3106,62$$

$$\sigma_{DDPP} = \sqrt{(t + LT) * \sigma_r^2 + r^2 * \sigma_{LT}^2} = \sqrt{(4 + 1,94) * 57^2 + 523^2 * 0,4564} = 379,65$$

$$P(DDPP > S) \Leftarrow \frac{1}{\#?em2anos} = \frac{1}{\frac{2*50}{4}} = 0,04 \Rightarrow N = 58$$

$$Z = \frac{3 * N}{100} \Leftrightarrow Z = \frac{3 * 58}{100} = 1,74$$

$$S = \mu_{DDPP} + Z\Gamma_{DDPP} = 3106,62 + 1,74 * 379,65 = 3767,211 \text{ artigos}$$

$$E[DDPP > S] = 2integral * \Gamma_{DDPP} = 0,014502 * 379,65 = 5,506$$

$$S = \sqrt{\frac{2 * r * C_3}{C_1}} + s - \frac{rt}{2} \Leftrightarrow s = S + \frac{rt}{2} - \sqrt{\frac{2 * rC_3}{C_1}} \Leftrightarrow$$

$$\Leftrightarrow s = 3767,211 + \frac{523 * 4}{0,345} - \sqrt{\frac{2 * 523 * 1500}{0,345}} \Leftrightarrow s = 2600,65 \text{ artigos}$$

Desta forma prevemos que para o ano 2021 a procura semanal será de 523 artigos com desvio padrão de 57 unidades, com um nível de referência de 3767,211 artigos e um nível de encomenda de 2600,65 artigos.

3.3 Questão 3

Utilizando uma folha de cálculo ou uma linguagem de programação, implemente um modelo de simulação do funcionamento do sistema de gestão pretendido (para 2021). Inclua, na sua folha ou programa, o cálculo das medidas de desempenho que achar adequadas para realizar as análises estatísticas subsequentes. Por exemplo, será adequado estimar o stock médio, as quebras, os custos, etc., para inferir a eficácia e a eficiência relativa das diversas instâncias numéricas da política de gestão a simular. Simule o funcionamento do sistema para conjuntos alternativos dos valores dos parâmetros se S , faça uma análise comparativa dos respetivos desempenhos, e sugira o conjunto ou conjuntos de valores (s , S) mais recomendados para implementar, indicando claramente ao Sr. Gervásio como deve proceder.

De modo a resolver esta questão, decidimos recorrer a um script de Python.

Primeiramente, introduzimos os dados no nosso script e calculamos os valores da procura para o ano de 2021. Para isso, utilizamos a biblioteca *numpy* do python.

```
media = 523
desvio = 57
initialS = 3767
initials = 2601

p1 = 0.26
p2 = 0.54
p3 = 0.2

C1 = 0.345
C2 = 20
C3 = 1500

procura = random.normal(loc=media, scale=desvio, size=50).astype(int).tolist()
```

Posteriormente, criamos uma função que calcula o custo total, o stock médio e o número de quebras de acordo com o s e com o S . Para isso, inicialmente, criamos uma lista para o stock, tempo de encomenda, custo de posse, custo de quebra e custo de encomenda.

```
def run(s, S, title):
    stock = []
    tempo_encomenda = []
    custo_posse = []
    custo_quebra = []
    custo_encomenda = []
```

Definimos também duas funções, *calculaTempoEncomenda* e *calculaCustos*, que calculam o tempo de entrega da encomenda e os custos para uma respetiva semana.

Para a primeira, geramos um número aleatório entre 0 e 1 e verificamos a que intervalos de probabilidade pertencia, adicionando assim o respetivo tempo de entrega à lista *tempo_encomenda*. Na eventualidade das condições não serem verificadas para realizar a encomenda, seria adicionado um 0.

Para a segunda, adicionamos os custos de posse, quebra e encomenda às respetivas listas.

```

def calculaTempoEncomenda(index, s, tempo_encomenda, stock):
    if (index%ciclos+1==ciclos) and stock[index]<s:
        tempo = random.random()
        if(tempo<p1):
            tempo_encomenda.append(1)
        elif(tempo<p1+p2):
            tempo_encomenda.append(2)
        else:
            tempo_encomenda.append(3)
    else:
        tempo_encomenda.append(0)

def calculaCustos(index, stock, custo_posse, custo_quebra,
                  custo_encomenda, tempo_encomenda):
    if stock[index] > 0:
        custo_posse.append(stock[index]*C1)
        custo_quebra.append(0)
    else:
        custo_posse.append(0)
        custo_quebra.append(abs(stock[index]*C2))
    if tempo_encomenda[index] == 0:
        custo_encomenda.append(0)
    else:
        custo_encomenda.append(C3)

```

Para a primeira semana do ano, o stock foi calculado subtraindo a procura dessa semana ao valor de S .

```

stock.append(S-procura[0])
calculaTempoEncomenda(0, s, tempo_encomenda, stock)
calculaCustos(0, stock, custo_posse, custo_quebra, custo_encomenda,
              tempo_encomenda)

```

Para as restantes verificamos se cada uma delas corresponde à semana de entrega de uma encomenda. Se sim, calculamos o valor do stock, subtraindo a procura dessa semana ao stock da semana passada e acrescentando a quantidade de artigos que foi encomendada. Caso contrário, apenas se subtrai a procura dessa semana ao stock da semana passada.

```

for i in range(1,semanas):
    cycleIndex = i%ciclos + 1
    cycle = (i//ciclos) * ciclos - 1

    if tempo_encomenda[cycle]==cycleIndex and cycleIndex != ciclos:
        stock.append(stock[i-1]-procura[i]+S-stock[cycle])
    else:
        stock.append(stock[i-1]-procura[i])

    calculaTempoEncomenda(i, s, tempo_encomenda, stock)
    calculaCustos(i, stock, custo_posse, custo_quebra,
                  custo_encomenda, tempo_encomenda)

```

Por último, calculamos o custo total, o stock médio e o número de quebras e imprimimos os resultados no ecrã.

```
custoTotal = sum(custo_posse) + sum(custo_quebra) + sum(custo_encomenda)
stockMedio = sum(stock) / len(stock)
quebras = len(list(filter(lambda x: (x<0), stock)))
```

```
printCustos(custoTotal, stockMedio, quebras, title)
```

```
def printCustos(custoTotal, stockMedio, quebras, title):
    custo = "{:.2f}".format(custoTotal)
    print("\n-----")
    print(title)
    print("-----")
    print("Custo Total: " + str(custo) + "€")
    print("Stock Médio: " + str(int(stockMedio)))
    print("Número Quebras: " + str(quebras))
    print("-----")
```

O script invoca a função run mostrada anteriormente para os valores dos parametros (s,S) , $(s+5\%,S)$, $(s-5\%,S)$, $(s,S+5\%)$ e $(s,S-5\%)$.

```
print("\nValores para 2021")
print(procura)

run(initials, initialS, "Para s e S")
run(initials*1.05, initialS, "Para s+5% e S")
run(initials*0.95, initialS, "Para s-5% e S")
run(initials, initialS*1.05, "Para s e S+5%")
run(initials, initialS*0.95, "Para s e S-5%")
```

3.3.1 Resultados

Correndo o script obtivemos os seguintes valores de procura para o ano de 2021:

```
Valores para 2021  
[404, 560, 611, 576, 556, 522, 621, 519, 519, 602, 537, 507, 543, 488, 391, 566, 549, 438, 524, 532, 530, 541,  
586, 532, 417, 511, 534, 554, 404, 406, 501, 507, 591, 552, 536, 518, 540, 599, 634, 448, 464, 537, 554, 523,  
467, 512, 587, 539, 378, 599]
```

Figura 4: Valores da Procura para 2021

Os resultados para os diferentes valores de s e S encontram de seguida:

```
-----  
Para s e S  
-----  
Custo Total: 51424.63€  
Stock Médio: 1937  
Número Quebras: 0  
-----
```

Figura 5: Resultado para s e S

```
-----  
Para s+5% e S  
-----  
Custo Total: 55931.37€  
Stock Médio: 2198  
Número Quebras: 0  
-----
```

Figura 6: Resultado para $s+5\%$ e S

```
-----  
Para s-5% e S  
-----  
Custo Total: 50891.95€  
Stock Médio: 1906  
Número Quebras: 0  
-----
```

Figura 7: Resultado para $s-5\%$ e S

```
-----  
Para s e S+5%  
-----  
Custo Total: 54905.17€  
Stock Médio: 2139  
Número Quebras: 0  
-----
```

Figura 8: Resultado para s e $S+5\%$

```
-----  
Para s e S-5%  
-----  
Custo Total: 51190.55€  
Stock Médio: 1924  
Número Quebras: 0  
-----
```

Figura 9: Resultado para s e $S-5\%$

Avaliando os resultados acima, concluímos que o custo mais baixo foi verificado para os parâmetros $(s-5\%, S)$, ou seja, $(2471, 3767)$.

Deste modo, concluímos que o Senhor Gervásio deve ter um stock máximo de 3767 unidades e deve realizar uma nova encomenda quando o stock baixa das 2471 unidades.

4 Conclusões

Com o objetivo de determinar os valores mais adequados para aplicar em cada situação que nos foi proposta, criamos um modelo de simulações do sistema de gestão.

Ao longo do desenvolvimento deste mesmo sistema, deparamo-nos com algumas dificuldades que fomos ultrapassando, com mais ou menos dificuldade, relativamente às questões colocadas e aos métodos que deveríamos utilizar para as resolver.

Após a obtenção dos resultados com as diversas simulações realizadas chegamos à solução que julgamos mais acertada em termos de redução de custos e de quantidades ótimas de encomenda, por exemplo.

Assim, este projeto ajudou-nos imenso na aprendizagem e desenvolvimento de diversos conceitos alusivos à temática abordada nas aulas. Atribuímos, então, um balanço positivo a este trabalho que nos elucidou sobre alguns dos processos e métodos a utilizar numa situação de gestão de stocks e nos ajudou a conhecer uma nova ferramenta que nos auxilia aquando dessa mesma organização na gestão dos stocks.

A Anexo

A.1 Valores de Procura

Semana	2018	2019	2020	Semana	2018	2019	2020
1	304	370	410	26	267	345	406
2	310	381	431	27	379	441	494
3	291	362	412	28	363	440	505
4	361	393	459	29	350	422	512
5	321	394	397	30	407	420	532
6	292	352	452	31	359	455	524
7	308	371	416	32	387	424	513
8	302	355	433	33	341	458	506
9	307	349	446	34	367	423	527
10	311	376	411	35	389	470	544
11	313	354	415	36	365	391	510
12	315	355	401	37	394	426	522
13	295	328	405	38	337	425	496
14	331	364	413	39	405	453	548
15	308	379	392	40	383	435	511
16	319	398	437	41	360	434	514
17	310	361	408	42	386	417	532
18	328	358	407	43	343	441	495
19	306	356	449	44	383	419	539
20	305	361	410	45	367	447	523
21	318	376	416	46	361	434	501
22	312	368	415	47	372	411	506
23	310	349	410	48	320	381	434
24	284	360	422	49	312	335	457
25	312	410	432	50	338	349	431

Tabela 1: Valores de procura

B Código da Aplicação

```
import random
from numpy import random

ciclos = 4
semanas = 50

media = 523
desvio = 57
initialS = 3767
initials = 2601

p1 = 0.26
p2 = 0.54
p3 = 0.2

C1 = 0.345
C2 = 20
C3 = 1500

procura = random.normal(loc=media, scale=desvio, size=50).astype(int).tolist()

def calculaTempoEncomenda(index, s, tempo_encomenda, stock):
    if (index%ciclos+1==ciclos) and stock[index]<s:
        tempo = random.random()
        if(tempo<p1):
            tempo_encomenda.append(1)
        elif(tempo<p1+p2):
            tempo_encomenda.append(2)
        else:
            tempo_encomenda.append(3)
    else:
        tempo_encomenda.append(0)

def calculaCustos(index, stock, custo_posse, custo_quebra,
                  custo_encomenda, tempo_encomenda):
    if stock[index] > 0:
        custo_posse.append(stock[index]*C1)
        custo_quebra.append(0)
    else:
        custo_posse.append(0)
        custo_quebra.append(abs(stock[index]*C2))
    if tempo_encomenda[index] == 0:
        custo_encomenda.append(0)
    else:
        custo_encomenda.append(C3)

def printCustos(custoTotal, stockMedio, quebras, title):
    custo = "{:.2f}".format(custoTotal)
    print("\n-----")
```

```

print(title)
print("-----")
print("Custo Total: " + str(custo) + "€")
print("Stock Médio: " + str(int(stockMedio)))
print("Número Quebras: " + str(quebras))
print("-----")

def run(s, S, title):
    stock = []
    tempo_encomenda = []
    custo_posse = []
    custo_quebra = []
    custo_encomenda = []

    stock.append(S-procura[0])
    calculaTempoEncomenda(0, s, tempo_encomenda, stock)
    calculaCustos(0, stock, custo_posse, custo_quebra,
                  custo_encomenda, tempo_encomenda)

    for i in range(1,semanas):
        cycleIndex = i%ciclos + 1
        cycle = (i//ciclos) * ciclos - 1

        if tempo_encomenda[cycle]==cycleIndex and cycleIndex != ciclos:
            stock.append(stock[i-1]-procura[i]+S-stock[cycle])
        else:
            stock.append(stock[i-1]-procura[i])

        calculaTempoEncomenda(i, s, tempo_encomenda, stock)
        calculaCustos(i, stock, custo_posse, custo_quebra,
                      custo_encomenda, tempo_encomenda)

    custoTotal = sum(custo_posse) + sum(custo_quebra) + sum(custo_encomenda)
    stockMedio = sum(stock) / len(stock)
    quebras = len(list(filter(lambda x: (x<0), stock)))

    printCustos(custoTotal, stockMedio, quebras, title)

print("\nValores para 2021")
print(procura)

run(initials, initialS, "Para s e S")
run(initials*1.05, initialS, "Para s+5% e S")
run(initials*0.95, initialS, "Para s-5% e S")
run(initials, initialS*1.05, "Para s e S+5%")
run(initials, initialS*0.95, "Para s e S-5%")

```
