

Universidade do Minho

2ºSemestre 2020/21

(MIEI, 3ºAno)

Modelos Estocásticos de Investigação Operacional

Trabalho Prático

(Problema de Gestão de Inventários)

Identificação do Grupo

<u>Número</u>	<u>Nome Completo</u>
A89468	Júlio Miguel de Sá Lima Magalhães Alves
A89537	Leonardo de Freitas Marreiros
A89616	Eduardo Benjamim Lopes Coelho
A89618	Henrique Gabriel dos Santos Neto

Rubricas



Benjamim Coelho


Data de Entrega: 2022-04-30

1 Introdução e Contextualização

Este trabalho tem como objetivo a abordagem do processo de desenvolvimento de um modelo estocástico por forma a avaliar qual a melhor política a estabelecer para a gestão de inventário da empresa Café&Afins.

Café&Afins é uma empresa multinacional que importa café do Brasil e o distribui por vários países da Europa e tem vindo a ter um crescimento no volume de vendas nos últimos anos.

Durante os últimos anos, o responsável pela gestão do armazém da empresa tem notado que a política de encomenda atual tem trazido problemas como o stock excessivo em certas alturas do ano ou a sua falta no caso de atraso na entrega das encomendas.

Com isto, a Café&Afins pretende agora adotar uma política de gestão de inventário (s,S) que funciona exatamente como a política de ciclo de encomenda. Neste caso são feitas revisões em intervalos regulares de t , 4 semanas, as encomendas são feitas caso o stock em mão desça abaixo do nível de encomenda, s . Finalmente, a quantidade a encomendar é variável, igual ao nível de referência S menos o stock em mão. Além disto, nesta política não se pretende que haja mais do que uma quebra de stock a cada dois anos.

Quanto aos outros dados do problema, sabemos que o prazo de entrega pode variar, podendo ser uma, duas ou três semanas, cujas probabilidades p_1, p_2, p_3 de ocorrência são: 0.27, 0.53 e 0.20, respetivamente. A quantidade de encomenda e nível de encomenda atuais são 1700 e 1200 sacos de café. Os custos de encomenda são 1500 euros, os custos de posse anual são 15% e os custos de quebra estão estimados em 36 euros por saco. Finalmente, o preço de compra do café tem um preço médio de 115 euros por saco.

2 Formulação do Problema

Aumento na produtividade, melhoria no serviço ao cliente e uma visão mais clara do stock são consequências de um bom controlo de stock. Este controlo apenas é possível com a implementação de boas estratégias e ferramentas, de modo a manter resultados positivos. Uma gestão eficiente e equilibrada de stock garante a minimização e prevenção de erros operacionais.

Um bom controlo de stock acontece quando um negócio tem conhecimento de cada produto no armazém, conhece o seu valor, sabe que produtos vendem mais e os que correm o risco de ficar em excesso. Ao mesmo tempo, sabe quando é a altura certa de comprar novos produtos. Melhorar o controlo de stock é a solução necessária para aumentar os lucros da empresa.

Desta forma, o problema passa por, inicialmente, analisar os resultados de anos passados e identificar falhas ou pontos que poderiam ter sido melhorados como valores de quantidade e nível de encomenda que conduziriam a menores custos e/ou menor quantidade de quebras.

De seguida, é feita uma previsão dos valores ótimos para a nova política (s,S) para 2021 com base

nos valores obtidos nos anos anteriores.

Para terminar, é implementado um modelo de simulação do funcionamento do sistema de gestão pretendido tendo em conta diferentes valores de \underline{s} e \underline{S} e uma análise comparativa dos resultados obtidos identificando por fim a solução ótima que deve ser implementada.

3 Questões

3.1 Questão 1

Um bom controlo de stock passa pelo conhecimento de cada produto no armazém: o seu valor, quais os produtos vendem mais e os que correm o risco de ficar em excesso. Nesta visão, começamos por analisar os valores de anos transatos e por identificar cenários que iriam levar a menores custos e quebras.

Ao analisar os dados referentes às procuras semanais dos diferentes anos, foi identificado um padrão claro. Este padrão consiste em duas alturas do ano em que a diferença na procura de sacos de café é bastante distinta. Decidimos designar estas épocas por época alta e época baixa. Este padrão é claro no seguinte gráfico:

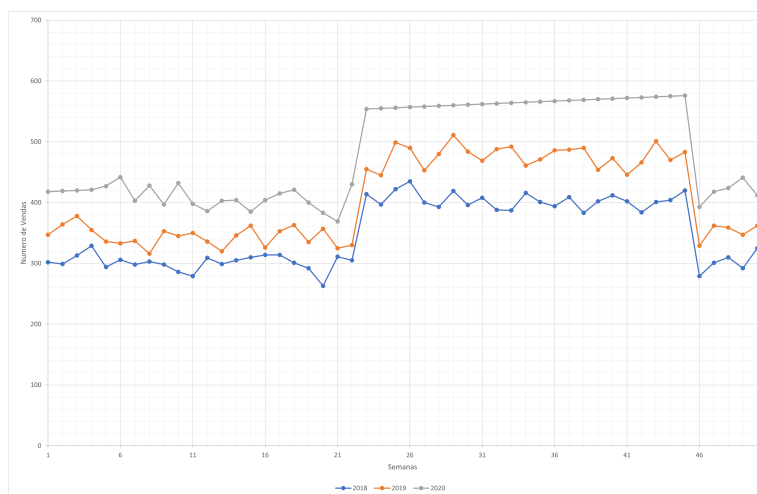


Figura 1: Valores das procuras médias semanais

Primeiro calculamos os valores que foram utilizados pelo gestor do armazém em 2020, para depois poder comparar com os dois estudos que foram realizados: uma política de nível de encomenda anual ou uma política de nível de encomenda com diferentes parâmetros para as diferentes épocas do ano.

Para todos os cálculos realizados a seguir, utilizámos os valores dados pelo enunciado:

- $i = 0.3\%$ semanal
- $b = 115 \text{ €/ unidade}$

- $c_3 = 1500 \text{ €/encomenda}$
- $c_2 = 36 \text{ €/unidade em atraso}$
- $c_1 = 0.345 \text{ €/unidade/semana}$
- $l = \{1, 2, 3\}$ semanas ($p_1 = 0.27$, $p_2 = 0.53$ e $p_3 = 0.20$)

3.1.1 Análise dos custos do ano 2020

Para conseguirmos saber 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, temos de calcular todos os custos que a empresa teve no ano 2020 com os parâmetros que o Sr. Gervásio utilizou.

- $q=1700$ unidades
- $S=1200$ unidades

Conhecendo o valor de S , se calcularmos os valores de $\mu_{DDL T}$ e $\sigma_{DDL T}$, conseguimos saber o valor de Z e, consequentemente N .

$$S = \mu_{DDL T} + Z * \sigma_{DDL T} \quad (1)$$

$$\mu_{DDL T} = r * l = 481.76 * 1.93 \approx 929.7968 \text{ u.} \quad (2)$$

$$\sigma_{DDL T} = \sqrt{l * \sigma_r^2 + r^2 * \sigma_l^2} \approx 109.5786 \text{ u.} \quad (3)$$

Assim,

$$Z \approx 2.4658, N = 82, 2^{\circ}integral = 0.001157 \quad (4)$$

$$E[DDL T > S] = 2^{\circ}integral * \sigma_{DDL T} \approx 0.1267 \quad (5)$$

Finalmente, podemos calcular o custo total semanal pela fórmula:

$$C = C_1 * \left(\frac{q}{2} + S - \mu_{DDL T}\right) + C_2 * \frac{r}{q} * E[DDL T > S] + C_3 * \frac{r}{q} \approx 812.8 \text{ €/semana} \quad (6)$$

Para obter o custo total do ano 2020 basta multiplicarmos este valor por 50 (assumindo que existem 50 semanas num ano):

$$C_{Tano} = 812.8 * 50 \approx 40642.2 \text{ €/ano} \quad (7)$$

Podemos também o calcular o número esperado de artigos em quebra por ano utilizando a fórmula:

$$N^{\circ}artigosemquebra = E[DDL T > S] * \frac{r}{q} = 0.1267 * \frac{481.76 * 50}{1700} \approx 2 \text{ sacos} \quad (8)$$

3.1.2 Estudo anual

Nesta iteração do processo, pretendemos calcular os valores ótimos de quantidade, nível de encomenda, custos e quebras utilizando para isso uma procura média semanal calculada a partir da média dos valores de todas as semanas de 2020 cujo resultado foi $r = 481.76$ com um desvio padrão $\sigma_r = 78.876$.

O valor ótimo de q (q^*) numa política de nível de encomenda tem-se com a expressão:

$$q^* = \sqrt{\frac{2r(C_2E[DDLT > S] + C_3)}{C_1}} \quad (9)$$

No entanto, não é possível o cálculo com esta fórmula pois $E[DDLT > S]$ depende de S . Assim, temos de começar por calcular q como uma aproximação da QEE:

1ª Aproximação:

$$q = QEE = \sqrt{\frac{2rC_3}{C_1}} = \sqrt{\frac{2 * 481.76 * 1500}{0.345}} = 2047 \text{ u.} \quad (10)$$

De seguida calculamos o risco ótimo de quebra $P^*(DDLT > S)$:

$$P^*(DDLT > S) = \frac{C_1 q^*}{C_2 r} = \frac{0.345 * 2047}{36 * 481.76} \approx 0.040719618 \quad (11)$$

Através da função de densidade normal, para o primeiro integral igual a 0.040719618 temos que $N = 58$. Com isto, calculamos o $E[DDLT > S]$, calculando primeiro o σ_{DDLT} :

$$\sigma_{DDLT} = \sqrt{l\sigma_r^2 + r^2\sigma_l^2} = \sqrt{1.93 * 78.876^2 + 481.76^2 * 0} \approx 109.579 \quad (12)$$

$$E(DDLT > S) = 2^\circ \text{Integral} * \sigma_{DDLT} = 0.014502 * 109.579 \approx 1.589 \quad (13)$$

2ª Aproximação:

$$q = \sqrt{\frac{2r(C_2E[DDLT > S] + C_3)}{C_1}} = \sqrt{\frac{2 * 481.76 * (36 * 1.589 + 1500)}{0.345}} = 2086 \text{ u.} \quad (14)$$

De seguida calculamos o risco ótimo de quebra $P^*(DDLT > S)$:

$$P^*(DDLT > S) = \frac{C_1 q^*}{C_2 r} = \frac{0.345 * 2086}{36 * 481.76} \approx 0.04149542 \quad (15)$$

Através da função de densidade normal, para o primeiro integral igual a 0.04149542 temos que $N = 57$. Com isto, calculamos o $E[DDLT > S]$, sabendo que $\sigma_{DDLT} = 109.579$:

$$E(DDLT > S) = 2^\circ \text{Integral} * \sigma_{DDLT} = 0.01573 * 109.579 \approx 1.724 \quad (16)$$

3ª Aproximação:

$$q = \sqrt{\frac{2r(C_2 E[DDLT > S] + C_3)}{C_1}} = \sqrt{\frac{2 * 481.76 * (36 * 1.724 + 1500)}{0.345}} = 2089 \quad (17)$$

De seguida calculamos o risco ótimo de quebra $P^*(DDLT > S)$:

$$P^*(DDLT > S) = \frac{C_1 q^*}{C_2 r} = \frac{0.345 * 2089}{36 * 481.76} \approx 0.041555097 \quad (18)$$

Através da função de densidade normal, para o primeiro integral igual a 0.04149542 temos que $N = 57$ o que significa que o processo convergiu. Conseguimos agora calcular o valor de S com os seguintes passos através de (20) e por fim podemos calcular o custo com (21).

$$\mu_{DDLT} = r * l = 481.76 * 1.93 \approx 929.7968 \quad (19)$$

$$S = \mu_{DDLT} + Z * \sigma_{DDLT} = 929.7968 + \frac{3 * 57}{100} * 109.579 = 1118 \quad (20)$$

$$C = C_1 \left(\frac{q}{2} + S - \mu_{DDLT} \right) + C_2 \frac{r}{q} E(DDLT > S) + C_3 \frac{r}{q} = 785.519 \text{ €/semana} \quad (21)$$

Multiplicando o custo total semanal pelas 50 semanas obtemos o custo total anual que é igual a 39275.96€. Calculamos também o número esperado de artigos em quebra por ano:

$$E[DDLT > S] * \frac{r * 50}{q} = 1.724 * \frac{481.76 * 50}{2089} = 30 \quad (22)$$

3.1.3 Estudo por épocas

Pela análise dos valores obtidos, é possível verificar que existem claramente duas épocas distintas. Decidimos designar as semanas de $1 - 22 \wedge 46 - 50$ como época baixa, e as restantes $23 - 45$ como época alta. Desta forma, procedemos a fazer a análise tendo as diferentes épocas em consideração.

Tal como no estudo anual, também temos como objetivo calcular os valores ótimos de quantidade, nível de encomenda, custos e quebras, porém neste caso iremos utilizar uma procura média semanal calculada apenas a partir dos valores de todas as semanas de cada época.

Época baixa

Para a época baixa temos $r \sim N(410.852, 18.315^2)$

1ª Aproximação:

$$q = QEE = \sqrt{\frac{2rC_3}{C_1}} = \sqrt{\frac{2 * 410,852 * 1500}{0.345}} = 1891 \text{ u.} \quad (23)$$

$$P^*(DDLT > S) = \frac{C_1 q^*}{C_2 r} = \frac{0.345 * 1891}{36 * 410,852} = 0.04410 \quad (24)$$

Através do valor de $P[DDLT > S]$, retiramos que $N = 57$ e partimos para o cálculo de $E[DDLT > S]$.

$$E[DDLT > S] = 2^{\circ} \text{integral} * \sigma_{DDLT} \approx 0,4002 \quad (25)$$

2º Aproximação:

$$q^* = \sqrt{\frac{2r(C_2 E[DDLT > S] + C_3)}{C_1}} = 1900 \text{ u.} \quad (26)$$

$$P^*(DDLT > S) = \frac{C_1 q^*}{C_2 r} = \frac{0.345 * 1900}{36 * 410,852} = 0.0443 \quad (27)$$

Como o valor de N convergiu comparativamente à primeira iteração, consideramos os valores ótimos os obtidos na segunda iteração. Posteriormente iremos calcular as variáveis necessárias para saber o custo total.

$$\mu_{DDLT} = 792,944, \sigma_{DDLT} = 25,444, Z = 1,71, S = 837 \quad (28)$$

Obtemos então o valor de custo total semanal para a época baixa

$$C_{TEB} = C_1 \left(\frac{q}{2} + S - \mu_{DDLT} \right) + C_2 \frac{r}{q} E[DDLT > S] + C_3 \frac{r}{q} \approx 670,42 \text{ €/semana} \quad (29)$$

Multiplicando o custo total semanal pelas 28 semanas correspondentes à época baixa obtemos o custo total da época que é igual a 18771.81€.

Época alta

Para a época alta temos $r \sim N(565, 6.782^2)$ unidades/semana

1º Aproximação:

$$q = 2217 \text{ u.}; P[DDLT > S] = 0,0376; N = 59; E[DDLT > S] = 0,0906 \quad (30)$$

2º Aproximação:

$$q = 2219 \text{ u.}; P[DDLT > S] = 0,0376; N = 59; E[DDLT > S] = 0,0906 \quad (31)$$

Como N da segunda aproximação converge com o N da primeira aproximação, consideramos que encontramos os valores ótimos e vamos então partir para o cálculo do custo total.

$$\mu_{DDLT} = 1090,45, \sigma_{DDLT} = 9,4223, Z = 1,77, S = 1108 \quad (32)$$

$$C_{TEA} = C_1 * \left(\frac{q}{2} + S - \mu_{DDLT} \right) + C_2 * \frac{r}{q} * E[DDLT > S] + C_3 * \frac{r}{q} \approx 771,5915 \text{ €/semana} \quad (33)$$

Multiplicando o custo total semanal pelas 22 semanas correspondentes à época alta obtemos o custo total da época que é igual a 16975.01€.

Resultados totais com o ano dividido por épocas

Após calcular os custos que cada época acarreta, resta-nos somar os mesmos para obter o custo total anual.

$$C_{Tano} = C_{TEB} + C_{TEA} = 18771,81 + 16975,01 = 35746,82 \text{ €/ano} \quad (34)$$

3.1.4 Análise de resultados

Resultados do ano 2020 com os parâmetros utilizados pelo Sr. Gervásio:

- Custo total = 40642.2 €/ano
- N°artigos em quebra = 2

Resultados do ano 2020 utilizando os valores ótimos da política de nível de encomenda aplicados ao ano inteiro:

- Custo total = 39275.9 €/ano
- N°artigos em quebra = 20

Resultados do ano 2020 aplicando a política de nível de encomenda a cada uma das épocas:

- Custo total = 35746.8 €/ano
- N°artigos em quebra = 4

[noitemsep]

Podemos verificar que, ao aplicar a política de nível de encomenda o Sr. Gervásio poderia ter poupado 1366.3 € ao longo do ano 2020. Porém, esta abordagem não é perfeita, uma vez que o número de artigos em quebra é muito superior. Deste modo, a melhor opção seria aplicar parâmetros distintos a cada uma das épocas do ano (época alta e época baixa).

Deste modo, o Sr. Gervásio poderia ter poupado 4895.4 € ao longo do ano 2020, sem aumentar muito o número de artigos em quebra.

3.2 Questão 2

Para estimar analiticamente os valores da política (s, S) para o ano 2021 tivemos primeiro de criar uma regressão linear obtida pelas médias dos anos anteriores. Assim como na questão anterior, iremos estudar duas soluções possíveis, uma delas considerando apenas uma regressão com as médias anuais, a outra considerando duas regressões referentes às épocas alta e baixas. Os resultados destas regressões foram os seguintes:

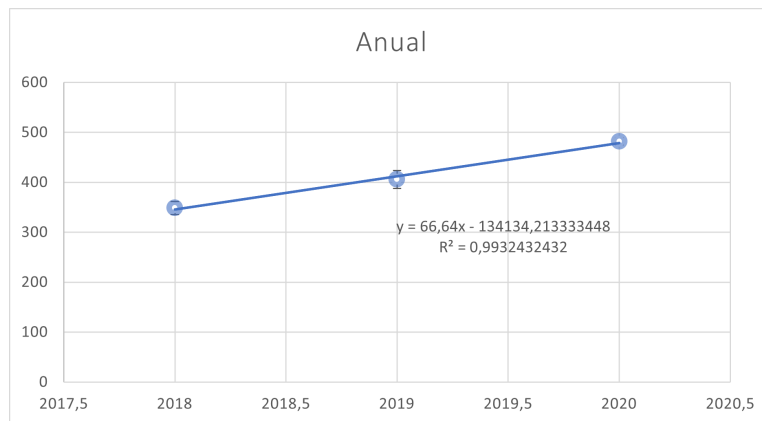
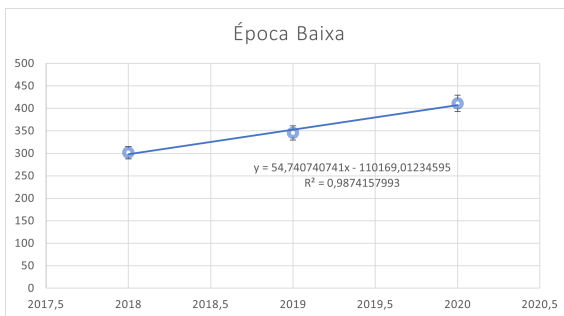
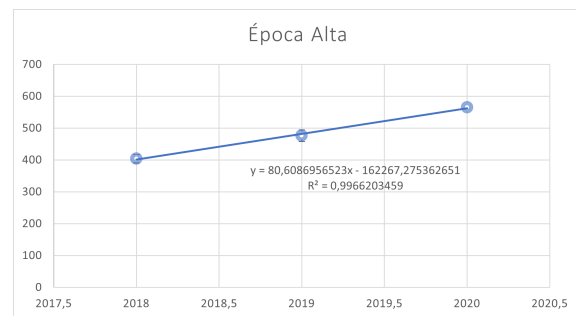


Figura 2: Regressão linear anual



(a) Época baixa



(b) Época alta

Figura 3: Regressões lineares divididas em épocas

Do enunciado, temos os seguintes dados:

- $i = 0.3\%$ semanal
- $b = 115 \text{ €/ unidade}$
- $c3 = 1500\text{€/ encomenda}$
- $c2 = 36\text{€/ unidade em atraso}$
- $c1 = 0.345 \text{ €/ unidade/ semana}$
- $l = \{1,2,3\}$ semanas ($p1 = 0.27$, $p2 = 0.53$ e $p3 = 0.20$)
- $t = 4$ semanas

3.2.1 Estudo anual

Para o estudo anual, começamos por obter o valor da procura média semanal (r) a partir da reta obtida na figura 2 substituindo o valor de x por 2021. Isto é:

$$y = 66.64x - 134134.2133 \Rightarrow$$

Substituindo x por 2021:

$$\begin{aligned} &= 66.64 * 2021 - 134134.2133 \\ &= 545.227. \end{aligned}$$

O desvio padrão de procura é obtido pela média dos anos anteriores, assim, obtemos $\sigma_r = 66.717$, pelo que ficamos com a procura norma $N(545.227; 66.717^2)$. Para o desvio padrão de entrega, assumimos que este é igual a 0. Tendo em consideração que o l não é sempre igual, calculamos o seu valor esperado que é igual a $1*0.27 + 2*0.53 + 3*0.20 = 1.93$.

Sabemos também que o número máximo de quebras por ano é uma quebra a cada dois anos pelo que calculamos o $P(DDPP > S)$ com a seguinte expressão:

$$P(DDPP > S) \leq \frac{1}{\#ciclosemdoisanos} = \frac{1}{\frac{2*50}{t}} = 0.04 \quad (35)$$

Apesar disto, decidimos calcular o risco ótimo de quebra $P^*(DDPP > S)$ com a expressão:

$$P(DDPP > S) = \frac{C_1 t}{C_2} = \frac{0.345 * 4}{36} \approx 0.0383 \quad (36)$$

Ora, como $0.0383 \leq 0.04$, a condição das quebras verificava-se, no entanto, o valor de N na tabela da função de densidade normal seria o mesmo, pelo que optamos por utilizar o valor 0.04 para uma simplificação nos cálculos. Com isto, conseguimos agora calcular o S através dos seguintes passos, utilizando fórmulas referentes ao ciclo de encomenda:

$$S = \mu_{DDPP} + Z * \sigma_{DDPP} \quad (37)$$

$$\mu_{DDPP} = r * (t + l) = 545.227 * (4 + 1.93) \approx 3233.194 \quad (38)$$

$$\sigma_{DDPP} = \sqrt{(t + l) * \sigma_r^2 + r^2 * \sigma_l^2} \approx 162.466 \quad (39)$$

Através da função de densidade normal, para o primeiro integral igual a 0.04 temos que **N = 58**. O $E(DDPP > S)$ pode ser calculado com:

$$E(DDPP > S) = 2^\circ Integral * \sigma_{DDPP} = 0.014502 * 162.466 \approx 2.356 \quad (40)$$

Finalmente, por (20) obtemos, substituindo:

$$S = 3233.194 * \frac{3 * 58}{100} * 162.466 = 3516 \quad (41)$$

Para obter o \underline{s} utilizamos:

$$s = S + \frac{r * t}{2} - q \quad (42)$$

Onde q = QEE por aproximação. Substituindo, temos:

$$s = 3516 + \frac{545.227 * 4}{2} - \sqrt{\frac{2 * 545.227 * 1500}{0.345}} \approx 2429 \quad (43)$$

Em seguida, calculamos o custo total. Começamos por calcular o custo total semanal:

$$C_t = C_1 * (S - r * l - \frac{r * t}{2}) + C_2 * \frac{1}{t} * E[DDPP > S] + C_3 * \frac{1}{t} \approx 869.98 \text{ €/semana} \quad (44)$$

Multiplicando o custo total semanal pelas 50 semanas obtemos o custo total anual que é igual a 43498.96€. Calculamos também o número esperado de artigos em quebra por ano:

$$E[DDPP > S] * \frac{r * 50}{q} = 2.356 * \frac{545.227 * 50}{2178} = 30 \quad (45)$$

3.2.2 Estudo por épocas

Pela análise dos valores obtidos, é possível verificar que existem claramente duas épocas distintas. Decidimos designar as semanas de $1 - 22 \wedge 46 - 50$ como época baixa, e as restantes $23 - 45$ como época alta. Com estes dados, ao invés do estudo anual, decidimos criar duas regressões: uma para a época alta e outra para a época baixa (Figura 3). Tal como anteriormente, foram feitos cálculos para determinar os parâmetros ótimos da política (s , S) e o seu custo associado.

Época baixa

$$y = 54.74x - 110169.012 \Rightarrow$$

Substituindo x por 2021:

$$\begin{aligned} &= 54.74 * 2021 - 110169.012 \\ &= 462.025. \end{aligned}$$

O desvio padrão é obtido pela média das épocas baixas dos anos anteriores, assim, obtemos $\sigma_r = 16.010$, pelo que ficamos com a procura normal $N(462.025; 16.010^2)$ unidades/semana.

Para o desvio padrão de entrega, mais uma vez assumimos que este é igual a 0. Temos novamente que: $l = 1*0.27 + 2*0.53 + 3*0.20 = 1.93$.

Tal como anteriormente, o valor de $P(DDPP > S)$ utilizado foi 0.04 que garante que não haja mais do que uma quebra a cada dois anos.

Com isto, conseguimos agora calcular o S através dos seguintes passos, utilizando fórmulas referentes ao ciclo de encomenda:

$$S = \mu_{DDPP} + Z * \sigma_{DDPP} \quad (46)$$

$$\mu_{DDPP} = r * (t + l) = 462.025 * (4 + 1.93) \approx 2739.806 \quad (47)$$

$$\sigma_{DDPP} = \sqrt{(t + l) * \sigma_r^2 + r^2 * \sigma_l^2} \approx 38.987 \quad (48)$$

$$(49)$$

Através da função de densidade normal, para o primeiro integral igual a 0.04 temos que **N = 58**. O $E(DDPP > S)$ pode ser calculado com:

$$E(DDPP > S) = 2^\circ Integral * \sigma_{DDPP} = 0.014502 * 38.987 \approx 0.565 \quad (50)$$

Finalmente, por (46) obtemos, substituindo:

$$S = 2739.806 * \frac{3 * 58}{100} * 38.987 = 2808 \quad (51)$$

E tendo em conta que $q \approx QEE$ resulta que s toma o valor de:

$$s = S + \frac{r * t}{2} - q = 2808 + \frac{462.025 * 4}{2} - \sqrt{\frac{2 * 462.025 * 1500}{0.345}} \approx 1728 \quad (52)$$

Em seguida, calculamos o custo total. Começamos por calcular o custo total semanal:

$$C_{TEB} = C_1 * (S - r * l - \frac{r * t}{2}) + C_2 * \frac{1}{t} * E[DDPP > S] + C_3 * \frac{1}{t} \approx 722.41 \text{ €/semana} \quad (53)$$

Multiplicando o custo total semanal pelas 28 semanas correspondentes à época baixa obtemos o custo total da época que é igual a 20227.55€.

Calculamos também o número esperado de artigos em quebra por época baixa:

$$E[DDPP > S] * \frac{r * 28}{q} = 0.565 * \frac{462.025 * 50}{2005} = 4 \quad (54)$$

Época alta

$$y = 80.61x - 162267.275 \Rightarrow$$

Substituindo x por 2021:

$$\begin{aligned} &= 80.61 * 2021 - 162267.275 \\ &= 642.9. \end{aligned}$$

O desvio padrão é obtido pela média das épocas altas dos anos anteriores, assim, obtemos $\sigma_r = 12.748$, pelo que ficamos com a procura normal:

$$N(642.9; 12.748^2) \text{ unidades/semana}$$

Assumindo o desvio padrão de entrega igual a 0, $l = 1*0.27 + 2*0.53 + 3*0.20 = 1.93$ e $P(DDPP > S) = 0.04$ conseguimos agora calcular o S através dos seguintes passos, utilizando fórmulas referentes ao ciclo de encomenda:

$$S = \mu_{DDPP} + Z * \sigma_{DDPP} \quad (55)$$

$$\mu_{DDPP} = r * (t + l) = 642.9 * (4 + 1.93) \approx 3812.388 \quad (56)$$

$$\sigma_{DDPP} = \sqrt{(t + l) * \sigma_r^2 + r^2 * \sigma_l^2} \approx 31.044 \quad (57)$$

Através da função de densidade normal, para o primeiro integral igual a 0.04 temos que $N = 58$. Desta forma $E(DDPP > S)$ pode ser calculado com:

$$E(DDPP > S) = 2^\circ \text{Integral} * \sigma_{DDPP} = 0.014502 * 31.044 \approx 0.450 \quad (58)$$

Finalmente, por (55) obtemos, substituindo:

$$S = 3812.388 * \frac{3 * 58}{100} * 31.044 = 3867 \quad (59)$$

Semelhante à época anterior $q \approx QEE$, e desta forma:

$$s = S + \frac{r * t}{2} - q = 3867 + \frac{642.9 * 4}{2} - \sqrt{\frac{2 * 642.9 * 1500}{0.345}} \approx 2788 \quad (60)$$

Em seguida, calculamos o custo total. Começamos por calcular o custo total semanal:

$$C_{TEA} = C_1 * (S - r * l - \frac{r * t}{2}) + C_2 * \frac{1}{t} * E[DDPP > S] + C_3 * \frac{1}{t} \approx 841.49 \text{ €/semana} \quad (61)$$

Multiplicando o custo total semanal pelas 22 semanas correspondentes à época alta obtemos o custo total da época que é igual a 18512.84€.

Calculamos também o número esperado de artigos em quebra por época alta:

$$E[DDPP > S] * \frac{r * 22}{q} = 0.450 * \frac{642.9 * 50}{2365} = 3 \quad (62)$$

Resultados totais com o ano dividido por épocas

Com os resultados obtidos de cada época, falta agora somar esses valores para obter informação acerca da totalidade do ano e para assim poder comparar resultado.

$$C_{Tano} = C_{TEB} + C_{TEA} = 20227.55 + 18512.84 = 38740.39 \text{ €} \quad (63)$$

$$E[DDPP > S]_{Tano} = E[DDPP > S]_{TEB} + E[DDPP > S]_{TEA} = 3 + 4 = 7 \quad (64)$$

3.2.3 Análise de resultados

Resta agora comparar os resultados para períodos diferentes do ano. Para o estudo anual, o custo total anual foi igual a 43498.96 com um número de sacos esperados em quebra de 30 unidades; quanto ao estudo por época, o custo anual total foi igual a 38740.39 com um número de sacos esperados em quebra de 7 unidades.

Podemos então concluir que a Café&Afins beneficiaria da implementação de uma política dividida para diferentes períodos do ano que se revelou menos custosa e com menos quebras do que uma política igual para todo o ano. Isto acontece pois, devido às disparidades entre valores de procura entre a época alta e a época baixa, a média de procura semanal equivale a um valor onde, na época baixa são encomendadas demasiadas unidades e na época alta sejam encomendadas um número insuficiente de unidades, o que causa um número elevado de quebras. Com a utilização desta política com diferentes parâmetros em diferentes épocas do ano, é possível corrigir este erro e atingir níveis ideais com um menor custo e menores quebras.

Dito isto, o gestor do armazém da empresa deve: para a época baixa entre as semanas 1 a 22 e 46 a 50 implementar a política (s, S) com $s = 1728$ e $S = 2808$; para a época alta entre as semanas 23 a 45 implementar a política (s, S) com $s = 2788$ e $S = 3867$;

3.3 Questão 3

Para realizar a simulação do ano 2021, optámos por utilizar a linguagem de programação Python, uma vez que permite a implementação de programas relativamente complexos com uma maior facilidade. Juntamente com o módulo numPy e Math, somos capazes de gerar procuras semanais para o ano 2021 utilizando uma distribuição normal para gerar valores aleatórios baseados nos cálculos realizados na questão anterior.

Como inferimos, anteriormente, que aplicar parâmetros diferentes às épocas alta e baixa, decidimos pedir valores s e S diferentes para as mesmas de modo a minimizar os custos. De seguida, num processo iterativo, para cada semana calculámos o stock, tendo em conta a procura, e, em cada período de revisão verificamos se é necessário fazer uma encomenda, isto é, se o nível de stock é menor ou igual que o nível de referência s . Caso seja preciso fazer uma encomenda, aumentamos o

stock da semana em que a encomenda chega pelo volume de encomenda calculado (seja x a semana atual, então $x + lt$ é a semana em que a encomenda chega. Seja u o stock atual, então o volume de encomenda a realizar é $S-u$). Para além disto, em caso de quebra de stock, ou seja, caso o stock atual seja ≤ 0 é importante calcular os custos implicados por esta situação. Assim, assumindo que 40% das vendas são perdidas em caso de quebra, os custos de quebra serão a soma entre o produto do custo de posse e 40% da procura durante a quebra e o produto do custo de quebra por 60% da procura. Para controlarmos todos os custos da operação da empresa, temos também de somar os custos de encomenda. Deste modo, todas as semanas somámos aos custos totais o produto do custo de encomenda pela procura dessa mesma semana.

Com o objetivo de analisar os custos totais, realizamos algumas simulações cujos resultados de uma dessas iterações foram os seguintes valores, inserindo variâncias de 5% nos valores de S e s :

- $S_{alta} = 3867$, $s_{alta} = 2788$, $S_{baixa} = 2808$, $s_{baixa} = 1728$, $C_{total} = 42471.15$ €, N°de quebras = 1
- $S_{alta} = 4061$, $s_{alta} = 2788$, $S_{baixa} = 2949$, $s_{baixa} = 1728$, $C_{total} = 43703.88$ €, N°de quebras = 1
- $S_{alta} = 3867$, $s_{alta} = 2928$, $S_{baixa} = 2808$, $s_{baixa} = 1815$, $C_{total} = 47331.75$ €, N°de quebras = 1
- $S_{alta} = 3674$, $s_{alta} = 2788$, $S_{baixa} = 2668$, $s_{baixa} = 1728$, $C_{total} = 49976.61$ €, N°de quebras = 3
- $S_{alta} = 3867$, $s_{alta} = 2649$, $S_{baixa} = 2808$, $s_{baixa} = 1642$, $C_{total} = 46256.055$ €, N°de quebras = 2



Figura 4: Gráficos de stocks das simulações 1,2,3,4,5, respetivamente.

Com isto podemos concluir que, para $S_{alta} = 3867$, $s_{alta} = 2788$, $S_{baixa} = 2808$, $s_{baixa} = 1728$: aumentar nível de referência (S) ou o nível de encomenda (s) produziram um aumento dos custos e das quebras de stock. Isto significa que os parâmetros encontrados são ótimos.

4 Conclusões

A gestão de stocks é um aspeto essencial para gerir uma empresa de modo eficiente. Uma boa gestão de stocks permite: aumentar os níveis de produtividade e a eficiência, ajuda a organizar o espaço de armazenamento, poupar tempo e dinheiro, melhora a exatidão das encomendas de inventário, etc.

Com a realização deste projeto, conseguimos experienciar alguns destes fatores, principalmente os últimos dois. Através da análise dos dados do ano anterior da Café&Afins, conseguimos perceber a importância de uma política de gestão de inventário com valores racionais e a forma como estes parâmetros otimizados levam a uma diminuição dos custos. Conseguimos também comparar diferentes abordagens e estudos de épocas para conseguir o máximo de poupança em custos e/ou a diminuição de quebras do stock.

De seguida, a partir de uma média estimada para os valores de 2021, foi possível estimar analiticamente quais os valores ótimos para a política (s, S) que pretendia ser implementada e, desta forma, analisar resultados de estudos diferentes e ponderar o *trade-off* entre custos mais baixos mas maiores quebras de stock com custos mais altos mas menos quebras.

Através do modelo de simulação do sistema, foi obtida uma política de gestão de inventário aproximadamente ótima.

Finalmente, é importante referir a importância da análise dos diferentes desempenhos obtidos com as diferentes políticas. Com este projeto conseguimos comparar políticas de nível de encomenda (incluindo modelos de vendas perdidas), ciclo de encomenda, e políticas de gestão de stocks mistas (s, S) e, desta forma, conseguimos sugerir o conjunto de valores mais recomendados para implementar pelo Sr. Gervásio na Café&Afins.

Anexos

A Dados Fornecidos

Grupo de Trabalho 2 - MIEI-MEIO 2020/21

VALORES DA PROCURA (EMPRESA Café&Afins)

<u>Semana</u>	<u>ANOS</u>		
	<u>2018</u>	<u>2019</u>	<u>2020</u>
1	302	347	418
2	299	364	388
3	313	378	429
4	329	355	410
5	294	336	427
6	306	333	442
7	298	337	403
8	303	316	428
9	298	353	397
10	286	345	432
11	279	350	398
12	309	336	386
13	299	320	403
14	305	346	404
15	310	362	385
16	314	326	404
17	314	353	415
18	301	363	421
19	292	335	400
20	263	357	383
21	311	325	369
22	305	330	430
23	414	455	554
24	397	445	567
25	422	499	534

<u>Semana</u>	<u>ANOS</u>		
	<u>2018</u>	<u>2019</u>	<u>2020</u>
26	435	490	556
27	400	453	545
28	393	480	560
29	419	511	548
30	396	484	553
31	408	469	541
32	388	488	555
33	387	492	560
34	416	461	563
35	401	471	561
36	394	486	569
37	409	487	600
38	383	490	526
39	402	454	538
40	412	473	538
41	402	446	578
42	384	466	538
43	401	501	536
44	404	470	558
45	420	483	587
46	279	329	393
47	301	362	418
48	310	359	424
49	292	347	441
50	325	362	412

ANEXO: Tabela de dados

Grupo de Trabalho 1

MIEI-MEIO 2020/21

VALORES DA PROCURA (EMPRESA Café&Afins)

ANOS

Semana

2018

2019

2020

1

302

347

418

2

299

364

419

3

313

378

420

4

329

355

421

5

294

336

427

6

306

333

442

7

296

337

403

8

303

316

428

9

298

353

397

10

286

345

432

11

279

350

398

12

309

336

386

13

299

320

403

14

305

346

404

15

310

362

385

16

314

326

404

17

314

353

415

18

301

363

421

19

292

335

400

20

263

357

383

21

311

325

369

22

305

330

430

23

414

455

554

24

397

445

555

25

422

499

556

26

435

490

557

27

400

453

558

28

393

480

559

29

419

511

560

30

396

484

561

31

408

469

562

32

388

488

563

33

387

492

564

34

416

461

565

35

401

471

566

36

394

486

567

37

409

487

568

38

383

490

569

39

402

454

570

40

412

473

571

41

402

446

572

42

384

466

573

43

401

501

574

44

404

470

575

45

420

483

576

46

279

329

393

47

301

362

418

48

310

359

424

49

292

347

441

50

325

363

412

DADOS

C1 (€/semana)

0,345

C2 (€/saco)

36

C3 (t)

1500

t (semanas)

1,93

t (semanas)

4

Época Baixa

Ano

Média

Desvio

2018

301,3703704

14,00590026

2019

345,4074074

15,70952494

2020

410,8518519

18,31498694

Desvio Padrão Médio

16,01013738

m

54,7407407

b

-110169,0124

Segundo a regressão

Ano

Média

-

2018

297,8023826

-

2019

352,5431233

-

2020

407,283864

-

2021

462,0246047

-

Época Alta

Ano

Média

Desvio

2018

403,7826087

13,27320103

2019

476,2608696

18,18895117

2020

565

6,782329983

Desvio Padrão Médio

12,74816073

m

80,60869565

b

-162267,2754

Segundo a regressão

Ano

Média

-

2018

401,0724637

-

2019

481,6811593

-

2020

562,289855

-

2021

642,8985506

-

Política Anual

Ano

Média

Desvio

2018

348,48

53,30705815

2019

405,6

67,96727904

2020

481,76

78,87643656

Desvio Padrão Médio

66,71692459

m

66,64

b

-134134,2133

Segundo a regressão

Ano

Média

-

2018

345,3066666

-

2019

411,9466666

-

2020

478,5866666

-

2021

545,2266666

-

Época Baixa

y = 54,7407407x - 110169,0124595

R² = 0,9874157993

2018

2018,5

2019

2019,5

2020

2020,5

Época Alta

y = 80,6086956323x - 162267,275362651

R² = 0,996620459

2018

2018,5

2019

2019,5

2020

2020,5

Anual

y = 66,64x - 134134,213333448

R² = 0,993242432

2018

2018,5

2019

2019,5

2020

2020,5

Época Baixa

judpp

oddp

r

σr

C1

C2

C3

LT

t

2739,805906

38,98723225

462,0246047

16,01013738

0,345

36

1500

1,93

4

P[DOPP+S]<=

q

E[DOPP]

0,04

2005

0,565392842

N

S=

s=

C=

C (Epoca Baixa)

Artigos_Quebra

58

2808

1728

722,41

20227,55

4

CT=

Artigos_Quebra_Ano

38740,39

7

Época Alta

judpp

oddp

r

σr

C1

C2

C3

LT

t

3812,388405

31,04380002

642,8985506

12,74816073

0,345

36

1500

1,93

4

P[DOPP+S]<=

q

E[DOPP]

0,04

2365

0,450197188

N

S=

s=

C=

C (Epoca Alta)

Artigos_Quebra

58

3867

2788

841,49

18512,84

3

CT=

Artigos_Quebra_Ano

38740,39

7

Anual

judpp

oddp

r

σr

C1

C2

C3

LT

t

3233,194133

162,4663282

545,2266666

66,71692459

0,345

36

1500

1,93

4

P[DOPP+S]<=

q

E[DOPP]

0,04

2178

2,356086691

N

S=

s=

C=

C (anual)

Artigos_Quebra_Ano

58

3516

2429

869,98

43498,96

30

CT=

Artigos_Quebra_Ano

38740,39

7

C Simulações

Nas paginas seguintes encontraremos 5 tabelas correspondentes a 5 simulações feitas com os dados :

1. $S_{alta} = 3867$, $s_{alta} = 2788$, $S_{baixa} = 2808$, $s_{baixa} = 1728$, $C_{total} = 42471.15$ €, N°de quebras = 1
2. $S_{alta} = 4061$, $s_{alta} = 2788$, $S_{baixa} = 2949$, $s_{baixa} = 1728$, $C_{total} = 43703.88$ €, N°de quebras = 1
3. $S_{alta} = 3867$, $s_{alta} = 2928$, $S_{baixa} = 2808$, $s_{baixa} = 1815$, $C_{total} = 47331.75$ €, N°de quebras = 1
4. $S_{alta} = 3674$, $s_{alta} = 2788$, $S_{baixa} = 2668$, $s_{baixa} = 1728$, $C_{total} = 49976.61$ €, N°de quebras = 3
5. $S_{alta} = 3867$, $s_{alta} = 2649$, $S_{baixa} = 2808$, $s_{baixa} = 1642$, $C_{total} = 46256.055$ €, N°de quebras = 2

Semana	Vendas	Stock	Pedido	Chegada	Lucro	Custo	Atrasos	Perdas
1	472	1256	True	False	53846,68	1933,32000000000002	0	0
2	475	2333	False	True	53820,115	804,88500000000001	0	0
3	462	1871	False	False	52484,505	645,495	0	0
4	456	1415	False	False	51951,825	488,17500000000007	0	0
5	439	976	True	False	50148,28	1836,72	0	0
6	440	536	False	False	50415,08	184,92000000000002	0	0
7	450	86	False	False	51720,33	29,67	0	0
8	468	1450	False	True	53319,75	500,25000000000006	0	0
9	469	981	True	False	53596,555	1838,44500000000002	0	0
10	443	2365	False	True	50129,075	815,92500000000001	0	0
11	473	1892	False	False	53742,26	652,74	0	0
12	472	1420	False	False	53790,1	489,90000000000003	0	0
13	474	946	True	False	54183,63	1826,37	0	0
14	448	498	False	False	51348,19	171,81	0	0
15	425	1935	False	True	48207,425	667,575	0	0
16	479	1456	False	False	54582,68	502,32000000000005	0	0
17	490	966	True	False	56016,73	1833,27	0	0
18	453	2355	False	True	51282,525	812,475	0	0
19	476	1879	False	False	54091,745	648,25500000000001	0	0
20	485	1394	False	False	55294,07	480,93000000000006	0	0
21	469	925	True	False	53615,875	1819,125	0	0
22	449	476	False	False	51470,78	164,22000000000003	0	0
23	637	2781	False	True	72295,555	959,445	0	0
24	661	2120	False	False	75283,6	731,40000000000001	0	0
25	650	1470	True	False	74242,85	2007,15	0	0
26	631	839	False	False	72275,545	289,45500000000004	0	0
27	653	186	False	False	75030,83	64,17	0	0
28	677	1906	False	True	77197,43	657,57	0	0
29	646	1260	True	False	73855,3	1934,7	0	0
30	617	643	False	False	70733,165	221,835	0	0
31	642	2608	False	True	72930,24	899,76000000000001	0	0
32	637	1971	False	False	72575,005	679,995	0	0
33	632	1339	True	False	72218,045	1961,955	0	0
34	654	685	False	False	74973,675	236,32500000000002	0	0
35	642	2571	False	True	72943,005	886,99500000000001	0	0
36	647	1924	False	False	73741,22	663,78000000000001	0	0
37	645	1279	True	False	73733,745	1941,255	0	0
38	613	666	False	False	70265,23	229,77	0	0
39	658	8	False	False	75667,24	2,7600000000000002	0	0
40	633	1963	False	True	72117,765	677,235	0	0
41	672	1291	True	False	76834,605	1945,395	0	0
42	641	650	False	False	73490,75	224,25000000000003	0	0
43	655	-3	False	False	74987,0	108,0	3	2
44	673	1900	False	True	76739,5	655,5	0	0
45	646	1254	True	False	73857,37	1932,63	0	0
46	459	795	False	False	52510,725	274,27500000000003	0	0
47	486	1863	False	True	55247,265	642,735	0	0
48	452	1411	False	False	51493,205	486,795	0	0
49	446	965	True	False	50957,075	1832,925	0	0
50	454	511	False	False	52033,705	176,29500000000002	0	0

Semana	Vendas	Stock	Pedido	Chegada	Lucro	Custo	Atrasos	Perdas
1	472	1256	True	False	53846,68	1933,32000000000002	0	0
2	475	781	False	True	54355,555	269,44500000000005	0	0
3	462	319	False	False	53019,945	110,055	0	0
4	456	1556	False	True	51903,18	536,82	0	0
5	439	1117	True	False	50099,635	1885,365	0	0
6	440	2509	False	True	49734,395	865,605	0	0
7	450	2059	False	False	51039,645	710,355	0	0
8	468	1591	False	True	53271,105	548,89500000000001	0	0
9	469	1122	True	False	53547,91	1887,09000000000001	0	0
10	443	679	False	True	50710,745	234,25500000000002	0	0
11	473	2033	False	True	53693,615	701,38500000000001	0	0
12	472	1561	False	False	53741,455	538,54500000000001	0	0
13	474	1087	True	False	54134,985	1875,015	0	0
14	448	639	False	False	51299,545	220,455	0	0
15	425	2076	False	True	48158,78	716,22	0	0
16	479	1597	False	False	54534,035	550,965	0	0
17	490	1107	True	False	55968,085	1881,915	0	0
18	453	2496	False	True	51233,88	861,12000000000001	0	0
19	476	2020	False	False	54043,1	696,90000000000001	0	0
20	485	1535	False	False	55245,425	529,575	0	0
21	469	1066	True	False	53567,23	1867,77	0	0
22	449	617	False	False	51422,135	212,865	0	0
23	637	-12	False	True	71903,0	432,0	12	8
24	661	2322	False	True	75213,91	801,09	0	0
25	650	1672	True	False	74173,16	2076,84	0	0
26	631	1041	False	False	72205,855	359,14500000000004	0	0
27	653	388	False	False	74961,14	133,86	0	0
28	677	2100	False	True	77130,5	724,50000000000001	0	0
29	646	1454	True	False	73788,37	2001,63	0	0
30	617	837	False	False	70666,235	288,76500000000004	0	0
31	642	195	False	True	73762,725	67,275	0	0
32	637	2165	False	True	72508,075	746,92500000000001	0	0
33	632	1533	True	False	72151,115	2028,885	0	0
34	654	879	False	False	74906,745	303,25500000000005	0	0
35	642	237	False	True	73748,235	81,765	0	0
36	647	2118	False	True	73674,29	730,71	0	0
37	645	1473	True	False	73666,815	2008,185	0	0
38	613	860	False	False	70198,3	296,70000000000005	0	0
39	658	2790	False	True	74707,45	962,55000000000001	0	0
40	633	2157	False	True	72050,835	744,16500000000001	0	0
41	672	1485	True	False	76767,675	2012,325	0	0
42	641	844	False	False	73423,82	291,18	0	0
43	655	189	False	False	75259,795	65,20500000000001	0	0
44	673	2092	False	True	76673,26	721,74	0	0
45	646	1446	True	False	73791,13	1998,87000000000001	0	0
46	459	2490	False	True	51925,95	859,05000000000001	0	0
47	486	2004	False	True	55198,62	691,38000000000001	0	0
48	452	1552	False	False	51444,56	535,44	0	0
49	446	1106	True	False	50908,43	1881,57000000000002	0	0
50	454	652	False	False	51985,06	224,94000000000003	0	0

Semana	Vendas	Stock	Pedido	Chegada	Lucro	Custo	Atrasos	Perdas
1	472	1256	True	False	53846,68	1933,32000000000002	0	0
2	475	781	False	True	54355,555	269,44500000000005	0	0
3	462	319	False	False	53019,945	110,055	0	0
4	456	1415	False	True	51951,825	488,17500000000007	0	0
5	439	976	True	False	50148,28	1836,72	0	0
6	440	2368	False	True	49783,04	816,96	0	0
7	450	1918	False	False	51088,29	661,71	0	0
8	468	1450	False	True	53319,75	500,25000000000006	0	0
9	469	981	True	False	53596,555	1838,44500000000002	0	0
10	443	538	False	True	50759,39	185,61	0	0
11	473	65	False	True	54372,575	22,425	0	0
12	472	1420	False	True	53790,1	489,90000000000003	0	0
13	474	946	True	False	54183,63	1826,37	0	0
14	448	2360	False	True	50705,8	814,2	0	0
15	425	1935	False	True	48207,425	667,575	0	0
16	479	1456	False	False	54582,68	502,32000000000005	0	0
17	490	966	True	False	56016,73	1833,27	0	0
18	453	513	False	True	51918,015	176,985	0	0
19	476	1879	False	True	54091,745	648,25500000000001	0	0
20	485	1394	False	False	55294,07	480,93000000000006	0	0
21	469	925	True	False	53615,875	1819,125	0	0
22	449	476	False	False	51470,78	164,22000000000003	0	0
23	637	-97	False	True	62403,0	3492,0	97	64
24	661	2184	False	True	75261,52	753,48	0	0
25	650	1534	True	False	74220,77	2029,23	0	0
26	631	3236	False	True	71448,58	1116,42	0	0
27	653	2583	False	False	74203,865	891,13500000000001	0	0
28	677	1906	False	True	77197,43	657,57	0	0
29	646	1260	True	False	73855,3	1934,7	0	0
30	617	643	False	False	70733,165	221,835	0	0
31	642	2608	False	True	72930,24	899,76000000000001	0	0
32	637	1971	False	True	72575,005	679,995	0	0
33	632	1339	True	False	72218,045	1961,955	0	0
34	654	685	False	False	74973,675	236,32500000000002	0	0
35	642	43	False	True	73815,165	14,835	0	0
36	647	1924	False	True	73741,22	663,78000000000001	0	0
37	645	1279	True	False	73733,745	1941,255	0	0
38	613	3254	False	True	69372,37	1122,63	0	0
39	658	2596	False	True	74774,38	895,62000000000001	0	0
40	633	1963	False	True	72117,765	677,235	0	0
41	672	1291	True	False	76834,605	1945,395	0	0
42	641	650	False	False	73490,75	224,25000000000003	0	0
43	655	2571	False	True	74438,005	886,99500000000001	0	0
44	673	1898	False	True	76740,19	654,81000000000001	0	0
45	646	1252	True	False	73858,06	1931,94	0	0
46	459	793	False	True	52511,415	273,58500000000004	0	0
47	486	1863	False	True	55247,265	642,735	0	0
48	452	1411	False	False	51493,205	486,795	0	0
49	446	965	True	False	50957,075	1832,925	0	0
50	454	511	False	False	52033,705	176,29500000000002	0	0

Semana	Vendas	Stock	Pedido	Chegada	Lucro	Custo	Atrasos	Perdas
1	472	1256	True	False	53846,68	1933,32000000000002	0	0
2	475	781	False	True	54355,555	269,445000000000005	0	0
3	462	1731	False	True	52532,805	597,195	0	0
4	456	1275	False	True	52000,125	439,875000000000006	0	0
5	439	836	True	False	50196,58	1788,42	0	0
6	440	396	False	True	50463,38	136,62	0	0
7	450	1778	False	True	51136,59	613,410000000000001	0	0
8	468	1310	False	True	53368,05	451,950000000000005	0	0
9	469	841	True	False	53644,855	1790,145	0	0
10	443	398	False	True	50807,69	137,31	0	0
11	473	1752	False	True	53790,56	604,44	0	0
12	472	1280	False	True	53838,4	441,6	0	0
13	474	806	True	False	54231,93	1778,070000000000002	0	0
14	448	358	False	True	51396,49	123,51	0	0
15	425	1795	False	True	48255,725	619,275000000000001	0	0
16	479	1316	False	False	54630,98	454,020000000000004	0	0
17	490	826	True	False	56065,03	1784,97	0	0
18	453	373	False	True	51966,315	128,685	0	0
19	476	1739	False	True	54140,045	599,955	0	0
20	485	1254	False	False	55342,37	432,630000000000005	0	0
21	469	785	True	False	53664,175	1770,825	0	0
22	449	3225	False	True	50522,375	1112,625	0	0
23	637	2588	False	True	72362,14	892,860000000000001	0	0
24	661	1927	False	True	75350,185	664,815	0	0
25	650	1277	True	False	74309,435	1940,565	0	0
26	631	3043	False	True	71515,165	1049,835	0	0
27	653	2390	False	False	74270,45	824,550000000000001	0	0
28	677	1713	False	True	77264,015	590,985	0	0
29	646	1067	True	False	73921,885	1868,115	0	0
30	617	450	False	False	70799,75	155,25	0	0
31	642	-115	False	True	60835,0	4140,0	115	77
32	637	1855	False	True	72615,025	639,975	0	0
33	632	1223	True	False	72258,065	1921,935	0	0
34	654	569	False	False	75013,695	196,305	0	0
35	642	-44	False	True	68911,0	1584,0	44	29
36	647	1760	False	True	73797,8	607,2	0	0
37	645	1115	True	False	73790,325	1884,675	0	0
38	613	502	False	True	70321,81	173,190000000000003	0	0
39	658	-94	False	True	65156,0	3384,0	94	62
40	633	1832	False	True	72162,96	632,040000000000001	0	0
41	672	1160	True	False	76879,8	1900,2	0	0
42	641	3033	False	True	72668,615	1046,385	0	0
43	655	2378	False	True	74504,59	820,410000000000001	0	0
44	673	1705	False	True	76806,775	588,225	0	0
45	646	1059	True	False	73924,645	1865,355	0	0
46	459	600	False	True	52578,0	207,000000000000003	0	0
47	486	114	False	True	55850,67	39,3300000000000005	0	0
48	452	1271	False	True	51541,505	438,495000000000006	0	0
49	446	825	True	False	51005,375	1784,625	0	0
50	454	371	False	False	52082,005	127,995	0	0

Semana	Vendas	Stock	Pedido	Chegada	Lucro	Custo	Atrasos	Perdas
1	472	1256	True	False	53846,68	1933,32000000000002	0	0
2	475	781	False	True	54355,555	269,44500000000005	0	0
3	462	1871	False	True	52484,505	645,495	0	0
4	456	1415	False	True	51951,825	488,17500000000007	0	0
5	439	976	True	False	50148,28	1836,72	0	0
6	440	536	False	True	50415,08	184,92000000000002	0	0
7	450	1918	False	True	51088,29	661,71	0	0
8	468	1450	False	True	53319,75	500,25000000000006	0	0
9	469	981	True	False	53596,555	1838,44500000000002	0	0
10	443	538	False	True	50759,39	185,61	0	0
11	473	1892	False	True	53742,26	652,74	0	0
12	472	1420	False	True	53790,1	489,90000000000003	0	0
13	474	946	True	False	54183,63	1826,37	0	0
14	448	498	False	True	51348,19	171,81	0	0
15	425	1935	False	True	48207,425	667,575	0	0
16	479	1456	False	False	54582,68	502,32000000000005	0	0
17	490	966	True	False	56016,73	1833,27	0	0
18	453	2355	False	True	51282,525	812,475	0	0
19	476	1879	False	True	54091,745	648,25500000000001	0	0
20	485	1394	False	False	55294,07	480,93000000000006	0	0
21	469	925	True	False	53615,875	1819,125	0	0
22	449	476	False	True	51470,78	164,22000000000003	0	0
23	637	-97	False	True	62403,0	3492,0	97	64
24	661	2184	False	True	75261,52	753,48	0	0
25	650	1534	True	False	74220,77	2029,23	0	0
26	631	903	False	True	72253,465	311,535	0	0
27	653	2583	False	True	74203,865	891,13500000000001	0	0
28	677	1906	False	True	77197,43	657,57	0	0
29	646	1260	True	False	73855,3	1934,7	0	0
30	617	643	False	False	70733,165	221,835	0	0
31	642	2608	False	True	72930,24	899,76000000000001	0	0
32	637	1971	False	True	72575,005	679,995	0	0
33	632	1339	True	False	72218,045	1961,955	0	0
34	654	685	False	False	74973,675	236,32500000000002	0	0
35	642	2571	False	True	72943,005	886,99500000000001	0	0
36	647	1924	False	True	73741,22	663,78000000000001	0	0
37	645	1279	True	False	73733,745	1941,255	0	0
38	613	666	False	True	70265,23	229,77	0	0
39	658	2596	False	True	74774,38	895,62000000000001	0	0
40	633	1963	False	True	72117,765	677,235	0	0
41	672	1291	True	False	76834,605	1945,395	0	0
42	641	650	False	True	73490,75	224,25000000000003	0	0
43	655	-3	False	True	74987,0	108,0	3	2
44	673	1900	False	True	76739,5	655,5	0	0
45	646	1254	True	False	73857,37	1932,63	0	0
46	459	795	False	True	52510,725	274,27500000000003	0	0
47	486	1863	False	True	55247,265	642,735	0	0
48	452	1411	False	True	51493,205	486,795	0	0
49	446	965	True	False	50957,075	1832,925	0	0
50	454	511	False	False	52033,705	176,29500000000002	0	0

D Código para Simulação questão 3

```
import numpy as np
import math
import re

weeks = 50 # N de semanas a simular
i = 0.15 / weeks # Taxa semanal de existencia
b = 115 # Preço unitario de um saco
t = 4 # Tamanho do periodo de revisão
revisao1 = 1 # Semana da primeira revisão (opcional)
C1 = i * b # Custo de existencia semanal
C2 = 36 # Custo de quebra por elemento
C3 = 1500 # Custo de passagem de encomenda
pLT = [0.27, 0.53, 0.2] # [p1,p2,p3]
LT = pLT[0] + 2 * pLT[1] + 3 * pLT[2] # Tempo de entrega médio
pPerda = 0.4 # Probabilidade de quebra nas encomendas em
    atraso
inicioEpocaAlta = 23 # Inicio da epoca contigua
fimEpocaAlta = 45 # Inicio da epoca contigua
#
S_alta = S_baixa = s_alta = s_baixa = 0
# Procura  $N(r_x, dv_x)$ ,  $x \in \{alta, baixa\}$ 
r_alta = 642.8985506
dv_alta = 12.74816073
r_baixa = 462.0246047
dv_baixa = 16.01013738

# Objeto representativo de uma Semana
class Semana:
    lucro = custo = 0.0
    id = perdas = atrasos = vendas = stock = 0
    pedido = False
    chegada = False
```

```
# Inicializador de uma semana
def __init__(self, id_):
    self.id = id_

# Converte uma semana em uma linha de texto CSV
def __str__(self):
    return (" " + str(self.id) + ";" +
            str(self.vendas) + ";" +
            str(self.stock) + ";" +
            str(self.pedido) + ";" +
            str(self.chegada) + ";—" +
            str(self.lucro) + ";" + str(self.custo) + ";" +
            str(self.atrasos) + ";" + str(self.perdas) + ";"
            "
    )

# Limpa os dados atuais menos a identificação da semana e a procura da mesma
def clean(self):
    self.perdas = self.atrasos = self.stock = 0
    self.lucro = self.custo = 0.0

# Regista uma encomenda e os seus custos devolvendo por fim o tempo de demorará a chegar
def encomendar(self):
    self.pedido = True
    self.custo += C3
    return np.random.choice([1, 2, 3], 1, pLT)[0]

# Recebe uma encomenda e acrescenta-a ao stock
def receber(self, inventario):
    self.chegada = True
    self.stock += inventario

# Efetua as vendas da respetiva semana e respetiva contabilidade
```

```
def negociar(self):
    # Se ainda nao estivermos em quebra
    if self.stock > 0:
        # Efetuar as vendas
        self.stock -= self.vendas
        # Se houve uma quebra no stock calcular as quebras
        if self.stock < 0:
            self.perdas = round(pPerda * abs(self.stock))
            self.stock += self.perdas
            self.atrasos = abs(self.stock)
        # Senão calcular o custo de posse dos elementos
        restantes
        else:
            self.custo += C1 * self.stock
    # Senão já estamos em quebra
    else:
        self.perdas = round(pPerda * self.vendas)
        self.atrasos = self.vendas - self.perdas
        self.stock -= self.atrasos
    self.custo += C2 * self.atrasos
    self.lucro = (self.vendas - self.perdas) * b - self.
    custo

# Herdas o stock anterior (incluindo o que falta entregar)
def herdar(self, week):
    self.stock += week.stock

# Inicializar a tabela da simulação
def init_table(stock_inicial):
    # Criar as instancias das semanas
    table = [Semana(i) for i in range(weeks + 1)]
    # Definir as vendas para as instancias criadas
    for i in range(1, weeks + 1):
        if inicioEpocaAlta <= i <= fimEpocaAlta:
            table[i].vendas = int(np.random.normal(r_alta ,
```

```
        dv_alta , size=None))
    else :
        table[i].vendas = int(np.random.normal(r_baixa ,
        dv_baixa , size=None))

    # Definir o stock inicial (semana 0)
    table[0].stock = stock_inicial

    return table

# Elimina os dados simulados para se poder efetuar uma nova
# simulação sobre a mesma população
def cleanup_table(table):
    for i in range(1, weeks + 1):
        table[i].clean()
    return table

# Indica se a semana dada deve adotar a politica alta ou baixa
def politica_alta(week):
    anticipation = math.ceil(LT)
    return (inicioEpocaAlta - anticipation) <= week <= (
        fimEpocaAlta - anticipation)

# Efetua a simulação
def simulation(table , S_alta , S_baixa , s_alta , s_baixa , t ,
weeks):
    for index in range(1, weeks + 1):
        # Herdar o stock da semana anterior (e mais so que não
        # sei como o CI é aplicado sema a semana :) )
        table[index].herdar(table[index - 1])
        # Vender as unidades em procura
        table[index].negociar()
        # Se estamos no periodo de revisao
```

```
    if (index - revisao1) % t == 0:
        # Determinos os parametros para a epoca em que
        # estamos
        if politica_alta(index):
            s = s_alta
            S = S_alta
        else:
            s = s_baixa
            S = S_baixa

        # Se o stock estiver abaixo do ponto predefinido
        if table[index].stock < s:
            # Encomendamos
            L = table[index].encomendar()
            # Receber a encomenda se a semana de chegada
            # ainda pertencer à simulação
            if index + L < weeks:
                table[index + L].receber(S - table[index].
                    stock)

    return table

# Lê dados do utilizador
def readFromUser():
    global S_alta, S_baixa, s_alta, s_baixa
    print("Simulação")
    S_alta = int(input("Insira o S da epoca alta: "))
    s_alta = int(input("Insira o s da epoca alta: "))
    S_baixa = int(input("Insira o S da epoca baixa: "))
    s_baixa = int(input("Insira o s da epoca baixa: "))

# Escreve a simulação num ficheiro CSV
def write(simcoiunt, table):
    name = 'simulacao' + str(simcoiunt) + '.csv'
    print("Resultado escrito no ficheiro", name)
```

```

with open(name, 'w') as file :
    file.write("Semana;Vendas;Stock;Pedido;Chegada;-;lucro;
               Custo;Atrasos;Perdas;\n")
    for i in range(1, 51):
        # Substituir todos os pontos por virgulas (Para
          funcionar direito em EXCEL)
        # e escrever a linha da tabela no ficheiro
        file.write(re.sub(r'\.', r',', str(table[i])) + "\n
                  ")

# Main
# Algoritmo Base
simcount = 1
readFromUser()
table = init_table(s_baixa) # Criar um possivel conjunto de
                             vendas para o ano 2020
while True:
    table = cleanup_table(table) # Limpar dados da simulação (
                                  se existirem)
    table = simulation(table, S_alta, S_baixa, s_alta, s_baixa,
                       t, weeks) # Efetuar a simulação
    write(simcount, table) # Escrever o resultado num ficheiro
    simcount += 1 # Aumentar o n da simulação
    # Se o utilizador nao insirir s ou si ou sim ou S ou SI ou
      SIM ou Si ou sI ou sIM ou sIm então termina o programa
    if re.search(r'(?i:si?m?)',
                 input("Deseja fazer outra simulação com os
                       mesmos dados de procura? (Sim/Não) ")) is
      None:
        break
readFromUser()

```