

// laboratório codex 02 · microeconomia

python + ia

>_ Laboratório Codex 02

Microeconomia em Python, com IA

Escolha do consumidor, curvas de demanda e de Engel, excedentes, produção, monopólio, oligopólio e teoria dos jogos: os modelos da Microeconomia em programas que você testa e explica.

// apostila do aluno II · trilha de exercícios

01 A escolha do consumidor

03 Excedentes e bem-estar

05 O monopolista

07 Teoria dos jogos

02 Demanda e curva de Engel

04 A função de produção

06 Oligopólio de Cournot

08 Estruturas de mercado



Prof. Caio Haffner

Mestre em Desenvolvimento Econômico · Unicamp

1

Como usar esta apostila

**LEIA TUDO, NA ORDEM**

Leia esta apostila **desde a primeira palavra** e faça tudo o que ela manda, na ordem. Não pule direto para os exercícios: cada página prepara a seguinte. Se ainda não fez a **Apostila I** (inclusive a página "PARE. Leia antes de tudo"), comece por ela.

Esta é a **continuação da Apostila I** (Laboratório Codex 01). Lá você instalou as ferramentas e aprendeu a conversar com o Codex. Aqui você vai **praticar**: cada exercício é um **modelo da Microeconomia**, do consumidor às estruturas de mercado. Você programa o modelo, testa com números conhecidos e interpreta o resultado.

1. Revisar

Checklist e lembretes da Apostila I

2. Consumidor

Escolha, demanda e Engel

3. Bem-estar

Excedentes e produção

4. Mercados

Monopólio e Cournot

5. Comparar

Simulador de estruturas

Os sinais desta apostila

Quando aparecer	O que significa
Cartão roxo · PROMPT	Texto para copiar e colar no Codex. Leia antes de enviar.
Quadrado tracejado · CHECKPOINT	O que você deve estar vendo. Se não estiver, pare e peça ajuda.
Caixa azul · DICA	Atalho, pergunta para pensar ou jeito mais fácil de fazer.
Caixa laranja · ATENÇÃO	Ponto em que muita gente erra (e a IA também).

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Resolver a escolha do consumidor	Calculo a cesta ótima com Cobb-Douglas.	☐
Derivar curvas	Monto as curvas de demanda e de Engel.	☐
Medir bem-estar	Calculo excedentes e peso morto.	☐
Analisar a produção	Mostro rendimentos marginais e de escala.	☐
Modelar mercados	Resolvo monopólio e Cournot.	☐
Usar teoria dos jogos	Encontro o equilíbrio de Nash numa matriz 2×2.	☐

1

REGRA DE OURO

Você é responsável pelo código que entrega, mesmo quando a IA o escreveu. **Não entregue código que você não consegue explicar.** E aqui vale mais uma: **todo modelo simplifica**; saiba dizer as hipóteses de cada um.

2

Antes de começar

A instalação está na **Apostila I** (Laboratório Codex 01, seções 3 a 7). Se alguma coisa abaixo não estiver pronta, volte lá antes de seguir.

O que precisa estar pronto	Onde está na Apostila I	Ok
VS Code instalado e abrindo	Seção 4	☐
Python funcionando no terminal (<code>python --version</code> ou <code>py --version</code>)	Seção 5	☐
Extensão do Codex instalada e com login feito	Seção 6	☐
Você sabe criar uma pasta de projeto e abrir no VS Code	Seção 7	☐

Crie a pasta desta apostila

- 1 Na pasta Documentos, crie uma pasta chamada `Laboratorio_Codex_MICRO` .
- 2 No VS Code, use **Arquivo** → **Abrir Pasta** e escolha essa pasta.
- 3 Abra o painel do Codex e confira que ele está conectado à sua conta.

**CHECKPOINT**

O Explorer mostra a pasta `Laboratorio_Codex_MICRO` e o Codex responde a um "olá".

**NÃO TEM A APOSTILA I?**

Peça ao professor o PDF do **Laboratório Codex 01 · Apostila do aluno**. As seções 3 a 7 levam cerca de 30 minutos.



O CICLO DE TRABALHO COM A IA

Pedir um **plano** → ler → aprovar → **executar** → **testar** com contas feitas por você → **explicar** o código com suas palavras.



MÉTODO DE DEPURAÇÃO

Diagnóstico → explicação → solução → alteração → teste. **Peça o diagnóstico antes da correção.**



A VÍRGULA BRASILEIRA

No Python, o decimal é com **ponto** (**13.75**). Em todos os programas desta apostila, peça ao Codex para **aceitar vírgula ou ponto**. E digite os milhares **sem ponto** (**1000** , e não **1.000**): um programa que aceita vírgula ou ponto pode ler 1.000 como 1.



SEGURANÇA

Só aprove o que você entende.
Trabalhe só na pasta do projeto. Nunca coloque senhas ou chaves no código.

PROMPT UNIVERSAL DE DIAGNÓSTICO

serve para qualquer erro

Estou com este problema no meu projeto.

Erro/mensagem: [COLE A MENSAGEM]

Arquivo envolvido: [NOME DO ARQUIVO]

Antes de alterar qualquer coisa:

1. explique a causa provável;
2. diga como confirmar a causa;
3. proponha a menor correção;
4. aguarde minha autorização para alterar o arquivo.

4

Exercício 1 · A escolha do consumidor

Com uma renda limitada, o consumidor escolhe a cesta que lhe dá mais satisfação. Com a utilidade Cobb-Douglas, a resposta tem uma fórmula simples.



$$U = x^{\alpha} \cdot y^{1-\alpha} \quad \cdot \quad x^* = \alpha \cdot R \div P_x \quad \cdot \quad y^* = (1 - \alpha) \cdot R \div P_y$$

$$\text{Restrição orçamentária: } P_x \cdot x + P_y \cdot y = R$$

PROMPT 1

Crie um programa em Python chamado `consumidor.py`.

O usuário informa a renda, os preços dos dois bens e o parâmetro α da utilidade Cobb-Douglas.

O programa deve:

1. calcular a cesta ótima e a utilidade;
2. confirmar que a cesta gasta exatamente a renda;
3. testar algumas cestas vizinhas na restrição orçamentária e mostrar que nenhuma tem utilidade maior;
4. recalcular a cesta se o preço de x mudar.

Antes de criar o arquivo, apresente um plano curto.

$R = 1.000 \cdot P_x = 10 \cdot P_y = 20 \cdot \alpha = 0,4$

Você espera

O programa mostrou

Cesta ótima (x, y)

Utilidade

Nova cesta com $P_x = 8$

5 Exercício 2 · Da utilidade à curva de demanda

Repetindo a escolha ótima para vários preços, nasce a **curva de demanda**. Repetindo para várias rendas, nasce a **curva de Engel**.

PROMPT 2

Crie um programa em Python chamado `curvas.py` que use a função do Exercício 1 para:

1. montar a curva de demanda do bem x : quantidade ótima para P_x de 5 a 25 (de 5 em 5);
2. montar a curva de Engel: quantidade ótima de x para rendas de 500 a 2.500;
3. classificar o bem como normal ou inferior e calcular a elasticidade-renda entre $R = 1.000$ e $R = 1.500$.

Mostre as duas tabelas. Antes de escrever o código, apresente um plano curto.

P_x	5	10	15	20	25
x^*					

Renda	500	1.000	1.500	2.000	2.500
x^*					



PARA PENSAR

Com a Cobb-Douglas, a elasticidade-renda é sempre 1. Por quê? Que tipo de utilidade daria um bem inferior?

6

Exercício 3 · Excedentes e bem-estar

Quanto os consumidores e os produtores ganham com o mercado? Os **excedentes** medem isso, e mostram o custo de um tabelamento.



excedente do consumidor = área entre a demanda e o preço
excedente do produtor = área entre o preço e a oferta

Com curvas lineares, as áreas são triângulos (ou trapézios)

PROMPT 3

Crie um programa em Python chamado bem_estar.py.

Com demanda $Q_d = a - b \cdot P$ e oferta $Q_s = -c + d \cdot P$, o programa deve calcular o equilíbrio,

o excedente do consumidor, o do produtor e o total. Depois, com um preço máximo abaixo do equilíbrio, deve calcular a quantidade transacionada, os novos excedentes e o peso morto (a perda total de bem-estar).

Use integração numérica simples ou geometria. Antes de escrever, apresente um plano.

$Q_d = 100 - 2P$ · $Q_s = -20 + 2P$	Equilíbrio	Teto de R\$ 25
Preço e quantidade		
Excedente do consumidor		
Excedente do produtor		
Total / peso morto		

7

Exercício 4 · A função de produção

Como a produção responde a mais trabalhadores? A lei dos **rendimentos marginais decrescentes** aparece nos números.



$$Q = A \cdot K^{\alpha} \cdot L^{\beta} \quad \cdot \quad PMe = Q \div L \quad \cdot \quad PMg = Q(L) - Q(L - 1)$$

Exemplo: $A = 10$, $K = 16$ máquinas, $\alpha = \beta = 0,5$

PROMPT 4

Crie um programa em Python chamado `producao.py`.

Com a função Cobb-Douglas e o capital fixo, o programa deve montar uma tabela para L de 1 a 25 trabalhadores com a produção, o produto médio e o produto marginal, apontar a partir de quando o produto marginal cai e testar os rendimentos de escala (o que acontece com Q se K e L dobrarem).

Antes de escrever o código, apresente um plano curto.

Trabalhadores (L)	Produção	Produto médio	Produto marginal
1			
4			
9			
16			
25			

Com K e L dobrando (32 e 32), a produção passa de _____ para _____: rendimentos _____ de escala.

8 Exercício 5 · O monopolista

Sem concorrentes, a empresa escolhe a quantidade em que a **receita marginal** iguala o custo marginal, e cobra o que a demanda aceitar.



$$\text{demanda } P = a - b \cdot Q \quad \cdot \quad \text{RMg} = a - 2b \cdot Q \quad \cdot \quad \text{RMg} = \text{CMg}$$

$$\text{índice de Lerner} = (P - \text{CMg}) \div P$$

$$\text{Exemplo: } P = 100 - 2Q, \text{CMg} = 20$$

PROMPT 5

Crie um programa em Python chamado monopolio.py.

Com a demanda inversa $P = a - b \cdot Q$ e custo marginal constante, o programa deve calcular

a quantidade, o preço e o lucro do monopolista; o resultado em concorrência perfeita ($P = \text{CMg}$); o peso morto do monopólio e o índice de Lerner.

Antes de escrever o código, apresente um plano curto.

	Monopólio	Concorrência perfeita
Quantidade		
Preço		
Lucro / excedente		

Peso morto do monopólio: _____ · Índice de Lerner: _____

9

Exercício 6 · Oligopólio de Cournot

Poucas empresas, cada uma escolhendo quanto produzir, de olho na outra. No equilíbrio de **Cournot**, nenhuma quer mudar sua quantidade.

★ $P = a - (q_1 + \dots + q_n) \cdot \text{custo } c \cdot \text{cada empresa: } q = (a - c) \div (n + 1)$

Exemplo: $a = 100, c = 10$

PROMPT 6

Crie um programa em Python chamado `cournot.py`.

Com a demanda $P = a - Q$ e custo c igual para todas, o programa deve calcular o equilíbrio

de Cournot para n empresas (quantidade de cada uma, total, preço e lucro de cada uma) e mostrar uma tabela para $n = 1, 2, 3, 5$ e 10 . Depois, encontre o equilíbrio de 2 empresas

"na prática": comece com quantidades quaisquer e faça cada empresa responder à outra (melhor resposta: $q = (a - c - q_{\text{outra}}) \div 2$) até parar de mudar.

Antes de escrever o código, apresente um plano curto.

Empresas	Quantidade total	Preço	Lucro de cada
1 (monopólio)			
2			
3			
5			
10			



PARA PENSAR

O que acontece com o preço quando o número de empresas cresce? Para onde ele tende?

10 Desafio extra · Teoria dos jogos

Duas empresas decidem o preço ao mesmo tempo: alto ou baixo. A matriz de payoffs mostra o lucro de cada uma, e o **equilíbrio de Nash** revela um dilema.

PROMPT 7

desafio

Crie um programa em Python chamado jogos.py que receba uma matriz 2x2 de payoffs (lucro da empresa A e da B para cada combinação de estratégias) e encontre: as melhores respostas, os equilíbrios de Nash (em estratégias puras) e se existe estratégia dominante. Teste com: (Alto, Alto) = (10, 10); (Alto, Baixo) = (4, 14); (Baixo, Alto) = (14, 4); (Baixo, Baixo) = (6, 6). Explique o dilema dos prisioneiros.

Pergunta

Resposta

Equilíbrio de Nash

Estratégia dominante

O resultado é o melhor para as duas?

11 Use o Codex como professor

Escolha um dos seus programas e use os três prompts abaixo. Anote no diário de prompts o que aprendeu.

EXPLICAR

Explique este código linha por linha para um estudante de Microeconomia que conhece o conteúdo, mas nunca programou em Python. Não reescreva o código.

REVISAR O CONTEÚDO

Revise meu programa como um professor de Microeconomia. Não faça alterações. Confira se os conceitos e as contas estão corretos (as fórmulas dos modelos, as condições de ótimo ($RMg = CMg$), as áreas dos excedentes e as hipóteses) e liste os riscos de entrada inválida.

TESTAR

Crie uma tabela de casos de teste para este programa, com o resultado esperado de cada caso: um caso normal, um caso-limite, valores com centavos ou decimais e uma entrada inválida.

**CONFIRA A TABELA DE TESTES**

Os "resultados esperados" que a IA escreve também podem estar errados. Confira pelo menos dois na calculadora.

Junte tudo num programa que compara, para o mesmo mercado, a **concorrência perfeita**, o **oligopólio de Cournot** e o **monopólio**: preço, quantidade, lucros e bem-estar.

Requisitos

- ☐ Ler a demanda ($P = a - b \cdot Q$) e o custo marginal
- ☐ Calcular concorrência perfeita, Cournot com n empresas e monopólio
- ☐ Para cada estrutura: preço, quantidade, lucro total, excedente do consumidor
- ☐ Calcular o peso morto em relação à concorrência perfeita
- ☐ Mostrar tudo numa tabela comparativa
- ☐ Explicar em uma frase o que muda de uma estrutura para outra



COM B DIFERENTE DE 1

Na Apostila II usamos $b = 1$ no Cournot. Generalize: $q = (a - c) \div (b \cdot (n + 1))$. Teste com os números do monopólio do Exercício 5.

PROMPT A

planejamento

Quero criar em Python um simulador que compare estruturas de mercado (concorrência perfeita, Cournot com n empresas e monopólio) para uma demanda $P = a - b \cdot Q$ e custo marginal c . Antes de escrever código, proponha as funções, as fórmulas de cada estrutura e casos de teste. Não escreva o código ainda.

PROMPT B

implementação

Agora implemente em `mercados.py`, reaproveitando os exercícios 3, 5 e 6.

$P = 100 - 2Q \cdot c = 20$	Preço	Quantidade	Peso morto
Concorrência perfeita			
Cournot (2 empresas)			
Monopólio			

13 Entrega e reflexão

Diário de prompts

#	O que pedi (resumo)	O que a IA fez	Precisei corrigir?
1			
2			
3			

O que entregar

Item	Entregue
Pasta do projeto com os arquivos .py dos exercícios e da atividade final	☐
As tabelas de teste desta apostila preenchidas	☐
Um erro de conteúdo que a IA cometeu (ou que você encontrou) e como foi corrigido	☐
O diário de prompts (acima) e a reflexão final (abaixo)	☐
Registro da auditoria humana (opcional, vale ponto extra)	☐

Reflexão individual

1. Qual modelo ficou mais claro depois que você o programou?

2. Em que momento a IA errou uma fórmula ou uma condição de ótimo?

3. Por que o monopólio gera peso morto, e o que acontece com mais concorrentes?

4. Dê um exemplo real de dilema dos prisioneiros entre empresas.

14 Como você será avaliado

Critério	0	1	2	3
Conteúdo da disciplina	Conceitos errados	Acertou alguns	Acertou quase todos	Todos corretos e conferidos
Uso do Codex	Não conseguiu	Só copiou	Usou com alguma compreensão	Usou de forma crítica
Programação	Não funciona	Funciona em parte	Funciona	Funciona e está bem organizado
Testes	Não testou	Um teste	Vários testes	Testes e casos-limite
Depuração	Não identificou	Precisou da solução pronta	Identificou com ajuda	Diagnosticou e corrigiu
Explicação	Não explica	Explica pouco	Explica a lógica	Explica a lógica e o significado econômico

Total máximo: 18 pontos, convertidos para a escala da disciplina.

As contas desta apostila

Cesta ótima	$x = \alpha R/P_x \cdot y = (1-\alpha)R/P_y$
Excedentes	áreas sob/sobre o preço
PMg	$Q(L) - Q(L-1)$
Monopólio	$RMg = CMg$
Lerner	$(P - CMg)/P$
Cournot	$q = (a-c)/(b(n+1))$

Comandos

TERMINAL

```
python arquivo.py    # executa um
programa
Ctrl + C             # interrompe um
programa travado
```

Glossário

Utilidade

Medida da satisfação do consumidor.

Restrição orçamentária

As cestas que a renda permite comprar.

Curva de Engel

Quantidade consumida conforme a renda.

Excedente

Ganho de consumidores ou produtores com o mercado.

Peso morto

Bem-estar perdido por uma distorção.

Rendimentos decrescentes

Cada trabalhador a mais acrescenta menos produção.

Monopólio

Uma só empresa vendendo.

Cournot

Oligopólio em que as empresas escolhem quantidades.

Equilíbrio de Nash

Ninguém ganha mudando sozinho de estratégia.

1

LEMBRE-SE

A IA é uma ferramenta. Conhecimento, julgamento, validação e responsabilidade continuam sendo de quem programa, e de quem usa os modelos para entender os mercados.

Todo material produzido com IA precisa de auditoria humana. Sempre. A IA escreve rápido e com segurança, mas pode errar uma conta, inventar uma regra ou usar um dado desatualizado. E ela escreve o erro com a mesma confiança do acerto.

Esta apostila foi escrita com ajuda de IA e revisada pelo professor, conta por conta. Mesmo assim, algum erro pode ter escapado. Agora o auditor é você: encontre, comprove e relate.



REGRA DO LABORATÓRIO: IA SEMPRE PASSA POR AUDITORIA HUMANA

A IA não sabe quando errou: o texto errado sai tão convincente quanto o certo.

Quem assina o resultado é uma pessoa: você no trabalho, o analista no relatório, o professor na apostila.

Auditar não é desconfiar da ferramenta. É a etapa que transforma um rascunho da IA num material confiável.

Vale para tudo: código, contas, textos, gráficos e esta apostila.

O que procurar

Tipo de erro	Exemplo	Como conferir
Conta ou número	Um total que não fecha, uma porcentagem errada	Refaça na calculadora
Regra da disciplina	Fórmula trocada, conceito mal explicado	Compare com o livro ou o material da aula
Referência	"Veja a seção 9", quando o assunto está na 11	Vá até a seção indicada
Prompt	Um prompt que faz o Codex errar sempre do mesmo jeito	Rode o prompt duas vezes
Dado desatualizado	Selic, IPCA, regra de imposto	Confira no site oficial (Banco Central, IBGE, Receita Federal)
Texto	Frase confusa, erro de português	Leia o trecho em voz alta

Como auditar com o Codex

PROMPT DE AUDITORIA

use com cuidado

Vou colar um trecho da minha apostila de [DISCIPLINA].

Atue como revisor. Não reescreva o texto.

1. refaça cada conta e mostre o cálculo;
2. aponte afirmações que podem estar erradas ou desatualizadas;
3. para cada problema, diga como eu posso confirmar sozinho.

Trecho: [COLE AQUI]



A IA TAMBÉM ERRA NA AUDITORIA

O Codex pode apontar erros que não existem. Só relate o que você confirmou com uma conta, uma fonte oficial ou um teste.

Registro da auditoria

#	Seção e página	O que está escrito	O que deveria ser	Como confirmei
1				
2				
3				

Como enviar

- 1 Preencha a tabela acima.
- 2 Envie cada erro pelo formulário: **laboratorio-codex.pages.dev/auditoria**
- 3 Aguarde a confirmação do professor.



O QUE VOCÊ GANHA

Cada erro confirmado vale **+1 ponto extra** na avaliação (até 3 pontos), e o seu nome entra na **lista de revisores** da próxima versão da apostila. Se duas pessoas relatarem o mesmo erro, vale o primeiro envio.