

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A01.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 13 e coluna 8

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 53 e da coluna 62 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 61 graus
- cosseno de 3 graus
- tangente de 41 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

36x_1 - 11x_2 - 30x_3 - 30x_4 - 35x_5 - 14x_6 - 25x_7 - 38x_8 + 27x_9 = -212
37x_1 - 38x_2 - 33x_3 - 33x_4 - 1x_5 + 7x_6 - 22x_7 - 20x_8 + 27x_9 = -277
27x_1 + 17x_2 - 26x_3 - 35x_4 + 32x_5 - 35x_6 - 32x_7 + 2x_8 - 25x_9 = 159
-29x_1 - 16x_2 + 40x_3 + 13x_4 + 22x_5 + 19x_6 - 23x_7 + 16x_8 + 27x_9 = -488
-4x_1 - 14x_2 + 28x_3 + 9x_4 + 3x_5 + 8x_6 + 3x_7 + 14x_8 + 2x_9 = -152
-13x_1 + 38x_2 - 15x_3 + 24x_4 - 25x_5 - 3x_6 + 27x_7 + 28x_8 + 24x_9 = 100
-24x_1 - 5x_2 + 2x_3 - 14x_4 + 33x_5 + 14x_6 + 32x_7 + 31x_8 - 5x_9 = 204
12x_1 - 26x_2 + 38x_3 - 8x_4 + 9x_5 + 6x_6 - 7x_7 + 32x_8 - 9x_9 = -176
-24x_1 - 27x_2 - 19x_3 + 36x_4 + 2x_5 - 11x_6 - 20x_7 - 5x_8 - 29x_9 = -192

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75260 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A02.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 5 e coluna 2

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 34 e da coluna 12 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 75 graus
- cosseno de 55 graus
- tangente de 37 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações
-28x_1 - 35x_2 + 34x_3 + 35x_4 - 13x_5 + 25x_6 + 6x_7 - 28x_8 + 24x_9 = 410
8x_1 + 6x_2 - 32x_3 + 19x_4 + 30x_5 + 10x_6 + 13x_7 - 7x_8 - 34x_9 = -270
-23x_1 + 3x_2 - 4x_3 + 8x_4 + 13x_6 + 40x_7 + 6x_8 + 16x_9 = 80
35x_1 - 29x_2 - 18x_3 - 29x_4 + 26x_5 - 33x_6 - 29x_7 + 37x_8 - 37x_9 = 33
-5x_1 + 13x_2 - 28x_3 + 16x_4 - 1x_5 - 11x_6 + 29x_7 + 15x_8 - 14x_9 = -26
11x_1 + 21x_2 + 7x_3 - 31x_4 + 10x_5 + 10x_6 + 13x_7 + 31x_8 - 12x_9 = -239
-23x_1 + 33x_2 - 10x_3 - 13x_4 + 13x_5 + 23x_6 + 26x_7 - 35x_8 + 28x_9 = -358
-24x_1 + 32x_2 - 5x_3 + 30x_4 - 16x_5 + 4x_6 + 30x_7 - 30x_8 - 3x_9 = -143
10x_1 - 14x_2 - 17x_3 + 10x_4 - 23x_5 + 2x_6 - 3x_7 - 18x_8 - 5x_9 = 77

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75277 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A03.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 10 e coluna 3

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 55 e da coluna 9 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 60 graus
- cosseno de 32 graus
- tangente de 64 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} 3x_1 - 31x_2 - 9x_3 - 31x_4 + 23x_5 + 34x_6 + 27x_7 - 23x_8 + 8x_9 &= 122 \\ -35x_1 + 33x_2 + 4x_3 - 17x_4 + 37x_5 - 36x_6 + 36x_7 + 30x_8 + 16x_9 &= -702 \\ 16x_1 + 25x_2 - 24x_3 - 29x_4 - 14x_5 - 4x_6 + 31x_7 - 15x_8 - 2x_9 &= -161 \\ 32x_1 - 12x_2 + 15x_3 + 21x_4 + 9x_5 + 31x_6 + 7x_7 + 2x_8 + 39x_9 &= 4 \\ 29x_1 + 5x_2 - 25x_3 - 20x_4 + 28x_5 - 15x_6 - 32x_7 + 5x_8 + 24x_9 &= -234 \\ 8x_1 + 9x_2 + 39x_3 + 38x_4 - 25x_5 + 26x_6 - 3x_7 - 25x_8 - 17x_9 &= 572 \\ -32x_1 + 17x_2 - 22x_3 - 30x_4 - 28x_5 - 21x_6 + 20x_7 - 19x_8 - 14x_9 &= -127 \\ -11x_1 - 36x_2 + 11x_3 - 31x_4 - 17x_5 - 9x_6 + 25x_7 + 19x_8 - 37x_9 &= -32 \\ 12x_1 - 5x_2 + 40x_3 - 14x_4 - 29x_5 - 15x_6 + 24x_7 + 3x_8 + 6x_9 &= -129 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

☞ Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75284 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A04.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 7 e coluna 7

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 24 e da coluna 21 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 3 graus
- cosseno de 57 graus
- tangente de 51 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} 15x_1 + 31x_2 - 34x_3 + 13x_4 + 3x_5 + 4x_6 + 3x_7 - 26x_8 + 21x_9 &= -136 \\ 27x_1 - 37x_2 - 26x_3 - 9x_4 + 36x_5 - 32x_6 + 13x_7 + 8x_8 + 24x_9 &= -254 \\ 11x_1 - 12x_2 + 30x_3 + 11x_4 + 17x_5 + 10x_6 + 32x_7 - 28x_8 + 3x_9 &= 87 \\ 33x_1 - 18x_2 + 27x_3 + 31x_4 + 13x_5 + 11x_6 + 33x_7 + 2x_8 - 39x_9 &= -351 \\ -32x_1 - 15x_2 + 5x_3 + 37x_4 - 25x_5 - 18x_6 - 27x_7 - 29x_8 + 20x_9 &= 309 \\ 27x_1 + 19x_2 - 35x_3 + 30x_4 + 31x_5 - 13x_6 + 20x_7 - 31x_8 - 7x_9 &= -264 \\ 30x_1 + 24x_2 - 32x_3 + 13x_4 - 11x_5 + 24x_6 - 24x_7 - 22x_8 + 32x_9 &= -125 \\ -8x_1 - 39x_2 + 23x_3 + 25x_4 - 29x_5 - 21x_6 + 16x_7 - 2x_8 - 38x_9 &= -92 \\ 13x_1 + 8x_2 + 7x_3 + 27x_4 + 32x_5 - 14x_6 + 39x_7 - 21x_8 + 36x_9 &= -132 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A05.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 16 e coluna 17

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 20 e da coluna 59 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 53 graus
- cosseno de 68 graus
- tangente de 87 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} 11x_1 + 27x_2 + 25x_3 - 22x_4 + 18x_5 - 32x_6 - 14x_7 + 40x_8 - 25x_9 &= 196 \\ -33x_1 - 4x_2 - 1x_3 - 2x_4 + 27x_5 + 19x_6 - 29x_7 - 28x_8 - 9x_9 &= 109 \\ 20x_1 - 28x_2 + 5x_3 - 10x_4 - 29x_5 + 4x_6 - 32x_7 - 38x_8 + 9x_9 &= 259 \\ 6x_1 + 3x_2 - 8x_3 - 12x_4 - 39x_5 - 14x_6 + 9x_7 - 28x_8 - 38x_9 &= 191 \\ -9x_1 - 28x_2 - 9x_3 - 4x_4 + 13x_5 - 22x_6 - 35x_7 + 33x_8 - 31x_9 &= -70 \\ -21x_1 - 25x_2 - 24x_3 - 38x_4 - 26x_5 + 19x_6 - 23x_7 + 28x_8 - 12x_9 &= 212 \\ -28x_1 + 35x_2 + 11x_3 - 19x_4 - 1x_5 - 8x_6 + 29x_7 + 10x_8 - 14x_9 &= 49 \\ 24x_1 - 28x_2 + 30x_3 - 32x_4 + 7x_5 - 20x_6 + 17x_7 + 25x_8 - 22x_9 &= 204 \\ -39x_1 - 11x_2 - 26x_3 + 37x_4 - 9x_5 - 7x_6 - 39x_7 - 8x_8 + 17x_9 &= -421 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A06.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 19 e coluna 18

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 4 e da coluna 66 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 58 graus
- cosseno de 39 graus
- tangente de 24 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

24x_1 - 19x_2 + 15x_3 + 16x_4 - 20x_5 - 21x_6 - 28x_7 + 17x_8 + 32x_9 = -495
-32x_1 - 7x_2 + 36x_3 + 30x_4 - 16x_5 + 36x_6 + 2x_7 - 10x_8 - 11x_9 = 263
13x_1 - 1x_2 - 37x_3 - 15x_4 - 34x_5 - 9x_6 + 18x_7 - 1x_8 + 17x_9 = 377
7x_1 + 2x_2 - 32x_3 + 11x_4 + 26x_5 - 23x_6 - 29x_7 - 10x_8 - 32x_9 = -77
-30x_1 - 1x_2 - 20x_3 + 7x_4 - 32x_5 + 24x_6 - 34x_7 + 30x_8 - 5x_9 = 360
-24x_1 - 31x_2 - 10x_3 - 20x_4 + 8x_5 - 24x_6 - 12x_7 + 9x_8 + 25x_9 = -225
-38x_1 + 35x_2 - 11x_3 + 25x_4 + 12x_5 + 27x_6 + 20x_7 + 13x_8 - 6x_9 = 500
-33x_1 - 24x_2 - 38x_3 - 35x_4 + 31x_5 - 25x_6 - 12x_7 + 23x_8 + 38x_9 = -203
7x_1 + 24x_2 + 19x_3 - 36x_4 + 21x_5 + 23x_6 + 39x_7 - 19x_8 - 1x_9 = 91

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75310 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A07.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 7 e coluna 2

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 37 e da coluna 69 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 42 graus
- cosseno de 77 graus
- tangente de 54 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

-9x_1 - 22x_2 - 39x_3 - 12x_4 + 2x_5 - 12x_6 + 12x_7 + 25x_8 + 4x_9 = 248
-23x_1 - 22x_2 + 39x_3 + 26x_4 + 22x_5 + 33x_6 + 25x_7 + 31x_8 + 37x_9 = 127
30x_1 + 2x_2 - 23x_3 - 28x_4 + 16x_5 - 33x_6 + 16x_7 - 35x_8 = -219
40x_1 + 15x_2 - 28x_3 - 19x_4 + 35x_5 + 10x_6 - 37x_7 + 12x_8 + 18x_9 = -595
-20x_1 + 37x_2 - 4x_3 - 25x_4 + 5x_5 - 12x_6 - 20x_7 - 33x_8 + 4x_9 = -78
20x_1 - 10x_2 + 2x_3 + 12x_4 + 29x_5 + 26x_6 - 27x_7 + 20x_8 + 21x_9 = -455
8x_1 - 37x_2 - 15x_3 + 4x_4 + 40x_5 - 2x_6 - 1x_7 + 38x_8 + 39x_9 = -286
-1x_1 - 19x_2 - 2x_3 - 32x_4 - 1x_5 + 40x_6 + 35x_7 - 13x_8 - 1x_9 = 453
6x_1 - 12x_2 + 4x_3 + 32x_4 - 16x_5 + 7x_6 - 11x_7 + 14x_8 - 15x_9 = -41

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75327 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A08.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 13 e coluna 15

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 17 e da coluna 32 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 36 graus
- cosseno de 5 graus
- tangente de 85 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} 12x_1 + 4x_2 - 38x_3 - 32x_4 + 7x_5 + 9x_6 - 31x_7 + 4x_8 + 28x_9 &= 140 \\ -1x_1 - 9x_2 - 9x_3 - 34x_4 - 14x_5 - 12x_6 + 5x_7 - 6x_8 - 11x_9 &= 175 \\ 28x_1 + 28x_2 - 14x_3 - 29x_4 + 5x_5 + 31x_6 + 37x_7 + 7x_8 + 13x_9 &= 181 \\ -37x_1 - 10x_2 + 30x_3 + 34x_4 + 19x_5 + 5x_6 + 30x_7 - 1x_9 &= 194 \\ 4x_1 + 16x_2 - 16x_3 - 27x_4 - 35x_5 - 27x_6 + 20x_7 - 24x_8 - 7x_9 &= 194 \\ 2x_1 - 11x_2 - 29x_3 - 14x_4 - 3x_5 + 9x_6 + 4x_7 + 11x_8 - 2x_9 &= 244 \\ -21x_1 + 11x_2 + 31x_4 - 11x_5 + 27x_6 + 22x_7 + 20x_8 - 20x_9 &= 98 \\ -2x_1 - 33x_2 - 18x_3 - 3x_4 + 25x_5 + 6x_6 - 2x_7 - 15x_8 - 34x_9 &= 40 \\ -24x_1 - 1x_2 + 2x_3 - 10x_4 - 11x_5 - 16x_6 + 16x_7 + 22x_8 + 4x_9 &= 276 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

☞ Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75334 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A09.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 2 e coluna 8

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 11 e da coluna 64 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 46 graus
- cosseno de 5 graus
- tangente de 15 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

19x_1 - 33x_2 - 19x_3 + 6x_4 + 11x_5 - 18x_6 + 34x_7 - 5x_8 - 11x_9 = -371
6x_1 - 15x_2 + 13x_3 + 26x_4 - 5x_5 + 7x_6 - 39x_7 - 35x_8 + 31x_9 = 103
3x_1 + 34x_2 - 14x_3 + 24x_4 - 10x_5 - 11x_6 - 36x_7 - 2x_8 - 30x_9 = 61
7x_1 + 13x_2 - 22x_3 - 8x_4 - 23x_5 - 33x_6 - 16x_7 - 39x_8 + 3x_9 = 326
2x_1 - 20x_2 + 23x_3 + 24x_4 + 26x_5 + 21x_6 + 18x_7 - 26x_8 + 9x_9 = -448
34x_1 + 40x_2 + 11x_3 + 38x_4 - 17x_5 + 34x_6 - 3x_7 - 21x_8 + 33x_9 = 463
-39x_1 - 13x_2 + 7x_3 - 7x_4 - 19x_5 + 7x_6 + 30x_7 - 38x_8 - 27x_9 = -486
29x_1 + 11x_2 - 32x_3 + 10x_4 - 12x_5 - 9x_6 - 37x_7 - 27x_8 + 7x_9 = 325
-38x_1 + 3x_2 - 18x_3 - 35x_4 - 13x_5 - 4x_6 + 22x_7 - 29x_8 + 32x_9 = 171

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gdc + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75341 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A10.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 8 e coluna 4

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 23 e da coluna 7 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 47 graus
- cosseno de 58 graus
- tangente de 41 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} -8x_1 + 3x_2 + 24x_3 + 37x_4 + 13x_5 - 1x_6 + 22x_7 + 39x_8 - 35x_9 &= -219 \\ -37x_1 - 27x_2 - 21x_3 + 14x_4 - 36x_5 - 23x_6 + 4x_7 + 30x_8 + 11x_9 &= 173 \\ 4x_1 - 11x_3 - 13x_4 - 5x_5 - 3x_6 + 34x_7 - 38x_8 + 14x_9 &= 14 \\ 4x_1 - 20x_2 + 40x_3 + 38x_4 + 19x_5 + 11x_6 - 19x_7 + 6x_8 - 3x_9 &= -185 \\ -33x_1 + 37x_2 - 24x_3 - 18x_4 - 8x_5 + 36x_6 - 9x_7 - 35x_8 + 39x_9 &= -86 \\ -19x_1 + 16x_2 + 17x_3 + 33x_4 - 17x_5 + 35x_6 - 32x_7 - 17x_8 + 34x_9 &= -128 \\ 4x_1 + 14x_2 - 16x_3 + 7x_4 - 8x_5 - 17x_6 - 6x_7 + 16x_8 - 18x_9 &= -26 \\ -24x_1 + 4x_2 - 13x_3 + 20x_4 + 28x_5 + 10x_6 + 36x_7 + 39x_8 + 31x_9 &= -281 \\ -11x_1 + 31x_2 - 15x_3 + 16x_4 - 7x_5 - 32x_6 + 24x_7 + 31x_8 - 15x_9 &= -300 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

☞ Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A11.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 11 e coluna 18

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 30 e da coluna 17 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 82 graus
- cosseno de 53 graus
- tangente de 54 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} -38x_1 + 21x_2 - 34x_3 + 20x_4 - 12x_5 - 8x_6 - 24x_7 + 40x_8 - 14x_9 &= 272 \\ -23x_1 - 15x_2 - 14x_3 - 23x_4 + 20x_5 + 39x_6 + 23x_7 + 16x_8 - 4x_9 &= -277 \\ -6x_2 - 14x_3 - 6x_4 - 39x_5 - 36x_6 - 25x_7 + 36x_8 - 12x_9 &= 172 \\ -18x_1 + 8x_3 - 15x_4 - 35x_5 + 22x_6 + 12x_7 + 24x_8 + 9x_9 &= -14 \\ 17x_1 - 28x_2 + 34x_3 + 37x_4 + 31x_5 - 21x_6 - 27x_7 + 29x_8 + 24x_9 &= -150 \\ 35x_1 - 12x_2 + 29x_3 + 25x_4 + 27x_5 - 11x_6 + 20x_7 - 23x_8 + 6x_9 &= -83 \\ 16x_1 + 16x_2 + 11x_3 - 1x_4 + 9x_5 - 39x_6 - 34x_7 - 34x_8 + 13x_9 &= -28 \\ 18x_1 + 32x_2 - 29x_3 - 23x_4 - 5x_5 + 32x_6 - 39x_7 - 7x_8 + 23x_9 &= 430 \\ 39x_1 + 29x_2 - 17x_3 - 17x_4 + 12x_5 + 20x_6 - 6x_7 - 24x_8 - 8x_9 &= 269 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (géc + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

☞ Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75365 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A12.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 2 e coluna 6

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 9 e da coluna 60 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 63 graus
- cosseno de 37 graus
- tangente de 54 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações
-23x1 - 9x2 + 29x3 + 22x4 - 32x5 - 31x6 - 7x7 - 11x8 + 4x9 = -327
10x1 - 26x2 + 34x3 + 4x4 - 19x5 + 11x6 - 16x7 + 28x8 - 33x9 = -292
-27x1 - 24x2 + 19x3 - 29x4 - 2x5 - 30x6 + 17x7 - 15x8 - 28x9 = -268
6x1 + 40x2 - 9x3 - 32x4 - 3x5 + 37x6 + 37x7 + 24x8 = 364
37x1 + 4x2 + 30x3 + 36x4 - 34x5 + 15x6 - 37x7 - 35x8 - 1x9 = 468
7x1 - 33x2 - 1x3 + 15x4 + 11x5 - 31x6 + 10x7 - 32x8 - 36x9 = -233
5x1 - 36x2 - 26x3 - 5x4 + 12x5 + 10x6 - 17x7 - 38x8 - 23x9 = 58
18x1 - 14x2 - 34x3 - 34x4 - 32x5 + 20x6 - 28x7 - 11x8 + 4x9 = 309
7x1 + 20x2 - 6x3 + 14x4 + 5x5 - 38x6 - 39x7 - 25x8 + 4x9 = 166

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r1) e o da última (r9).

6: r1	6: r9
-------	-------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a1,1 x a2,2 - a2,1 x a1,2. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r1	6.r9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A13.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 13 e coluna 12

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 43 e da coluna 25 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 30 graus
- cosseno de 28 graus
- tangente de 43 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} -12x_1 + 38x_2 + 21x_3 + 18x_4 + 17x_5 - 26x_6 + 21x_7 - 36x_8 - 36x_9 &= 389 \\ -15x_1 - 20x_2 - 39x_3 - 33x_4 + 18x_5 - 36x_6 - 2x_7 - 4x_8 - 39x_9 &= -326 \\ 27x_2 + 35x_3 + 6x_4 - 12x_5 + 12x_6 - 33x_7 - 32x_8 + 28x_9 &= 750 \\ -17x_1 + 4x_2 + 13x_3 - 39x_4 + 16x_5 + 26x_6 - 39x_7 + 34x_8 - 22x_9 &= -127 \\ -35x_1 + 16x_2 + 23x_3 - 12x_4 - 28x_5 + 9x_6 - 21x_7 + 5x_8 + 21x_9 &= 98 \\ -5x_1 - 19x_2 + 36x_3 - 17x_4 - 26x_5 - 3x_6 + 28x_7 + 18x_8 - 37x_9 &= -171 \\ 31x_1 - 7x_2 - 5x_3 - 26x_4 + 30x_5 - 30x_6 - 18x_7 + 4x_8 + 23x_9 &= 288 \\ -10x_1 - 17x_2 + 29x_3 - 23x_4 + 14x_5 - 14x_6 - 7x_7 + 31x_8 - 3x_9 &= -113 \\ -33x_1 - 32x_2 - 22x_3 + 31x_4 - 3x_5 - 20x_6 - 29x_7 + 30x_8 + 22x_9 &= -383 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r1) e o da última (r9).

6: r1	6: r9
-------	-------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a1,1 x a2,2 - a2,1 x a1,2. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gac + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r1	6.r9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75389 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A14.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 7 e coluna 6

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 55 e da coluna 11 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 38 graus
- cosseno de 59 graus
- tangente de 14 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} 23x_1 + 2x_2 + 3x_3 + 6x_4 + 14x_5 - 9x_6 + 19x_7 - 12x_8 + 28x_9 &= -10 \\ 35x_1 + 37x_2 + 31x_3 + 39x_4 - 6x_5 - 13x_6 + 5x_7 - 25x_8 - 24x_9 &= 248 \\ -9x_1 - 14x_2 - 11x_3 - 2x_4 - 32x_5 - 5x_6 + 4x_7 + 21x_8 - 21x_9 &= 80 \\ 32x_1 - 1x_2 + 34x_3 - 32x_4 - 35x_5 - 2x_6 + 38x_7 + 19x_8 + 22x_9 &= -161 \\ 24x_1 - 38x_2 + 8x_3 + 15x_4 - 16x_5 - 30x_6 - 37x_7 + 17x_8 - 34x_9 &= -111 \\ -34x_1 + 16x_2 + 18x_3 + 2x_4 - 39x_5 + 37x_6 + 8x_7 - 2x_8 - 38x_9 &= 310 \\ -30x_1 + 30x_2 - 37x_3 + 38x_4 + 29x_5 - 3x_6 - 11x_7 + 13x_8 - 4x_9 &= -14 \\ 4x_1 - 39x_2 - 12x_3 - 39x_4 - 21x_5 - 5x_6 + 27x_7 + 32x_8 - 37x_9 &= 128 \\ 36x_1 + 12x_2 - 22x_3 + 40x_4 + 36x_5 + 9x_6 + 6x_7 + 3x_8 + 36x_9 &= 189 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75396 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A15.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 5 e coluna 11

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 53 e da coluna 50 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 45 graus
- cosseno de 85 graus
- tangente de 10 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações
-28x₁ - 33x₂ + 38x₃ - 22x₄ + 24x₅ + 14x₆ - 34x₇ + 22x₈ + 6x₉ = -667
-12x₁ - 39x₂ + 23x₃ - 19x₄ - 24x₅ - 24x₆ + 3x₇ + 22x₈ - 18x₉ = -408
-26x₁ - 39x₂ + 16x₃ - 12x₄ - 14x₅ + 3x₆ + 4x₇ - 34x₈ - 5x₉ = -248
10x₁ + 10x₂ - 12x₃ - 27x₄ - 21x₆ + 37x₇ - 4x₈ - 33x₉ = 431
6x₁ - 4x₂ - 35x₃ - 17x₄ - 23x₅ + 9x₆ - 27x₇ - 7x₈ - 28x₉ = -111
-32x₁ - 25x₂ - 30x₃ + 11x₄ - 15x₅ + 29x₆ + 2x₇ + 22x₈ - 12x₉ = -583
4x₁ + 8x₂ + 28x₃ + 7x₅ + 33x₆ - 18x₇ - 4x₈ - 7x₉ = 10
-19x₁ - 7x₂ - 7x₄ - 1x₅ + 37x₆ - 13x₇ - 22x₈ - 15x₉ = -151
-26x₁ - 21x₂ - 34x₃ - 30x₄ - 13x₅ + 26x₆ + 14x₇ + 28x₈ + 35x₉ = -766

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75408 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A16.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 5 e coluna 3

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 12 e da coluna 7 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 58 graus
- cosseno de 10 graus
- tangente de 31 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} 21x_1 - 23x_2 - 28x_3 + 20x_4 + 5x_5 + 11x_6 + 7x_7 + 28x_8 + 14x_9 &= 282 \\ 32x_1 + 37x_2 - 10x_3 + 26x_4 + 2x_5 - 36x_6 - 18x_7 - 2x_8 + 31x_9 &= -44 \\ 29x_1 + 19x_2 - 17x_3 + 3x_4 - 15x_5 + 14x_6 + 33x_7 - 20x_8 + 26x_9 &= 422 \\ 6x_1 + 30x_2 + 23x_3 - 33x_4 + 31x_5 + 26x_6 - 28x_7 + 28x_8 + 8x_9 &= -446 \\ 6x_1 + 11x_2 - 17x_3 - 19x_4 + 24x_5 + 30x_6 - 16x_7 - 7x_8 - 6x_9 &= -284 \\ -27x_1 - 20x_2 - 22x_3 - 24x_4 - 28x_5 + 16x_6 - 32x_7 + 22x_8 + 4x_9 &= -167 \\ -22x_1 + 4x_2 + 11x_3 - 36x_4 - 7x_5 - 5x_6 - 38x_7 + 18x_8 - 35x_9 &= -703 \\ -14x_1 + 24x_2 + 8x_3 - 7x_4 + 32x_5 + 2x_6 - 5x_7 + 26x_8 + 25x_9 &= -150 \\ -11x_1 + 21x_2 - 17x_3 - 9x_4 - 38x_5 + 29x_6 + 9x_7 - 33x_8 - 21x_9 &= -21 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75415 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A17.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 8 e coluna 1

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 65 e da coluna 79 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 18 graus
- cosseno de 30 graus
- tangente de 39 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} -5x_1 - 9x_2 + 20x_3 - 17x_4 + 21x_5 + 34x_6 + 10x_7 + 12x_8 + 33x_9 &= 302 \\ -26x_1 + 35x_2 - 36x_3 + 7x_4 + 5x_5 + 16x_6 - 17x_7 + 34x_8 + 9x_9 &= 26 \\ -32x_1 + 27x_2 - 24x_3 + 4x_5 - 33x_6 - 8x_7 + 9x_8 + 26x_9 &= -279 \\ -32x_1 + 21x_2 + 8x_3 - 25x_4 + 21x_5 - 12x_6 - 37x_7 + 33x_8 - 19x_9 &= 728 \\ 5x_1 - 5x_2 - 33x_3 + 8x_4 + 18x_5 - 26x_6 - 24x_7 + 4x_8 - 18x_9 &= -105 \\ -37x_1 - 9x_2 + 24x_3 - 17x_4 + 7x_5 + 6x_6 + 33x_7 + 24x_8 - 38x_9 &= 628 \\ 36x_1 - 12x_2 + 5x_3 - 22x_4 + 37x_5 + 3x_6 - 19x_7 + 7x_8 - 21x_9 &= 565 \\ 4x_1 - 33x_2 - 39x_3 + 29x_4 + 3x_5 - 2x_6 - 26x_7 - 20x_8 - 17x_9 &= -499 \\ -31x_1 + 2x_2 - 34x_3 - 35x_4 - 16x_5 + 38x_6 - 29x_7 + 21x_8 + 28x_9 &= 61 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

🔗 Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A18.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 20 e coluna 3

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 17 e da coluna 36 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 71 graus
- cosseno de 44 graus
- tangente de 19 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

-15x_1 + 6x_2 - 36x_3 - 2x_4 + 29x_5 - 9x_6 - 23x_7 - 34x_8 - 36x_9 = 329
-13x_1 + 3x_2 + 22x_3 - 11x_4 + 39x_5 + 15x_6 - 28x_7 - 7x_8 - 28x_9 = 683
-35x_1 - 24x_2 + 31x_3 + 12x_4 + 29x_5 + 32x_6 + 31x_7 - 32x_8 - 33x_9 = 431
-14x_1 + 32x_2 - 33x_3 + 23x_4 - 32x_5 + 21x_6 + 24x_7 - 20x_8 - 30x_9 = 25
34x_1 - 12x_2 + 31x_3 - 28x_4 - 2x_5 + 11x_6 + 37x_7 + 20x_8 + 11x_9 = -269
-7x_1 - 32x_3 + 11x_4 + 9x_6 + 19x_7 - 8x_8 + 28x_9 = -448
-27x_1 - 18x_2 - 11x_3 + 2x_4 + 16x_5 + 17x_6 - 10x_7 - 15x_8 - 27x_9 = 255
37x_1 + 23x_2 - 17x_3 - 20x_4 + 24x_5 + 35x_6 + 17x_7 + 38x_8 - 31x_9 = -16
-12x_1 + 26x_2 + 5x_3 + 30x_4 + 18x_5 + 40x_6 - 15x_7 + 6x_8 + 6x_9 = 382

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gac + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A19.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 1 e coluna 1

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 39 e da coluna 70 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 27 graus
- cosseno de 86 graus
- tangente de 58 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} -14x_1 - 4x_2 - 17x_3 + 11x_4 + 3x_5 + 11x_6 + 35x_7 - 6x_8 + 7x_9 &= -10 \\ 10x_1 - 32x_2 - 19x_3 - 31x_4 - 16x_5 - 37x_6 - 10x_8 - 29x_9 &= 69 \\ -29x_1 - 21x_2 - 9x_3 - 8x_4 - 37x_5 + 6x_6 + 32x_7 + 37x_8 + 35x_9 &= -38 \\ -11x_1 + 11x_2 + 10x_3 + 22x_4 + 13x_5 - 22x_6 - 28x_7 + 24x_8 - 15x_9 &= 125 \\ 34x_1 + 12x_2 + 8x_3 + 4x_4 - 3x_5 + 24x_6 - 26x_7 + 39x_8 - 10x_9 &= 259 \\ -22x_1 + 27x_2 - 31x_3 - 20x_4 - 5x_5 + 23x_6 + 12x_7 - 33x_8 + 32x_9 &= -48 \\ -37x_1 - 7x_2 - 7x_3 + 11x_4 - 20x_5 + 30x_6 + 23x_7 - 11x_8 + 3x_9 &= -343 \\ 26x_1 + 16x_2 + 9x_3 + 28x_4 - 38x_5 - 14x_6 - 31x_7 + 40x_8 - 13x_9 &= 202 \\ -8x_1 + 24x_2 - 15x_3 + 6x_4 + 4x_5 + 21x_6 - 19x_7 + 5x_8 + 36x_9 &= 56 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (géc + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

🔗 Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75446 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A20.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 8 e coluna 12

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 7 e da coluna 63 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 26 graus
- cosseno de 9 graus
- tangente de 61 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações
-37x1 + 19x2 - 33x3 - 13x4 + 3x5 - 32x6 + 5x7 + 13x8 - 6x9 = 588
7x1 - 19x2 - 20x3 - 35x4 - 7x5 - 1x6 + 22x7 - 29x8 - 32x9 = -205
13x1 + 18x2 - 7x3 - 26x4 - 39x5 - 2x6 + 6x7 - 17x8 - 24x9 = -198
-11x1 - 27x2 + 8x3 + 19x4 - 23x5 - 33x6 - 33x7 + 31x8 - 9x9 = 329
13x1 - 6x2 + 27x3 - 36x4 - 29x5 + 14x6 - 20x7 + 27x8 + 35x9 = -512
32x1 + 16x2 - 27x3 - 23x4 + 5x5 + 9x6 + 19x7 + 6x8 + 39x9 = -205
-12x1 - 26x2 - 22x3 - 3x4 - 30x5 + 20x6 - 23x7 - 37x8 + 22x9 = -219
-38x1 - 9x2 - 2x3 - 37x4 + 37x5 + 39x6 - 17x7 + 32x8 + 19x9 = -58
23x1 - 31x2 + 6x3 + 26x4 + 18x5 + 34x6 - 5x7 - 29x8 - 16x9 = -328

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r1) e o da última (r9).

6: r1	6: r9
-------	-------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a1,1 x a2,2 - a2,1 x a1,2. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

☞ Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r1	6.r9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A21.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 20 e coluna 11

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 3 e da coluna 55 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 58 graus
- cosseno de 62 graus
- tangente de 34 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

-23x_1 + 6x_2 + 40x_3 - 24x_4 + 21x_5 - 38x_6 + 9x_7 - 14x_8 + 34x_9 = 38
9x_1 + 30x_2 + 24x_3 + 3x_4 - 17x_5 + 4x_6 - 31x_7 + 22x_8 - 9x_9 = 32
-39x_1 - 30x_2 - 25x_3 + 2x_4 + 35x_5 - 23x_6 - 19x_7 - 7x_8 + 18x_9 = 324
-10x_1 - 11x_2 + 22x_3 - 36x_4 - 34x_5 - 2x_6 - 26x_7 + 37x_8 + 20x_9 = 242
6x_1 - 13x_2 - 11x_3 - 17x_4 - 24x_5 - 3x_6 - 9x_7 - 24x_8 + 37x_9 = 150
17x_1 - 19x_2 + 14x_3 - 23x_4 + 3x_5 - 39x_6 - 12x_7 - 23x_8 + 2x_9 = -46
15x_1 - 39x_2 - 29x_3 + 10x_4 + 26x_5 + 21x_6 + 20x_7 + 16x_8 + 26x_9 = 508
20x_1 + 7x_2 - 38x_3 + 29x_4 - 22x_5 - 8x_6 + 19x_7 - 21x_8 + 10x_9 = -289
11x_1 + 6x_2 - 26x_3 - 10x_4 - 20x_5 - 15x_6 + 5x_7 + 22x_8 - 18x_9 = -271

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75460 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A22.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 7 e coluna 18

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 69 e da coluna 18 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 75 graus
- cosseno de 30 graus
- tangente de 15 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} -19x_1 + 32x_2 + 19x_3 + 5x_4 + 10x_5 + 27x_6 + 39x_7 - 35x_8 + 14x_9 &= 121 \\ 8x_1 - 34x_2 - 27x_3 + 22x_4 + 21x_5 + 8x_6 + 11x_7 + 6x_8 - 25x_9 &= -148 \\ -22x_1 - 29x_2 - 8x_3 - 4x_4 + 18x_5 + 8x_6 - 37x_7 + 30x_8 - 13x_9 &= -368 \\ -10x_1 + 8x_2 - 16x_3 - 29x_4 + 21x_5 + 33x_6 - 5x_7 + 22x_8 + 6x_9 &= 178 \\ 5x_1 - 11x_2 + 2x_3 + 31x_4 - 22x_5 - 20x_6 + 30x_7 + 18x_8 + 35x_9 &= -213 \\ -18x_1 + 18x_2 - 16x_3 - 28x_4 + 7x_5 - 39x_6 - 8x_7 - 27x_8 - 18x_9 &= 423 \\ 25x_1 + 34x_2 - 5x_3 - 38x_4 + 31x_5 + 24x_6 - 15x_7 - 24x_8 + 18x_9 &= 645 \\ -34x_1 + 28x_2 + 29x_3 + 28x_4 + 15x_5 + 3x_6 - 39x_7 + 35x_8 - 24x_9 &= -331 \\ -23x_1 + 14x_2 + 2x_3 + 6x_4 + 30x_5 + 20x_6 - 29x_7 + 36x_8 - 16x_9 &= -116 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r1) e o da última (r9).

6: r1	6: r9
-------	-------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a1,1 x a2,2 - a2,1 x a1,2. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r1	6.r9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75477 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A23.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 16 e coluna 1

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 70 e da coluna 9 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 51 graus
- cosseno de 85 graus
- tangente de 50 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} 12x_1 + 33x_2 + 13x_3 + 28x_4 - 38x_5 + 22x_6 - 9x_7 - 2x_8 - 7x_9 &= 635 \\ 19x_1 + 18x_2 + 40x_3 + 30x_4 - 22x_5 - 12x_6 + 12x_7 + 32x_8 - 25x_9 &= -153 \\ 26x_1 + 6x_2 + 33x_3 + 23x_4 + 23x_5 + 28x_6 - 4x_7 + 24x_8 - 5x_9 &= 5 \\ 21x_1 - 36x_2 + 36x_3 - 36x_4 - 35x_5 + 22x_6 - 11x_7 + 15x_8 + 36x_9 &= -261 \\ 24x_1 - 28x_2 - 33x_3 + 39x_4 - 13x_5 + 14x_6 + 15x_7 + 28x_8 - 4x_9 &= 173 \\ -22x_1 + 16x_2 + 13x_3 + 4x_4 + 27x_5 + 40x_6 - 4x_7 - 6x_8 + 11x_9 &= 238 \\ 7x_1 + 30x_2 + 12x_3 + 32x_4 + 13x_5 - 28x_6 - 36x_7 - 17x_8 - 4x_9 &= 252 \\ -34x_1 + 11x_2 + 3x_3 - 26x_4 + 12x_5 - 24x_6 - 38x_7 + 4x_8 - 9x_9 &= -282 \\ -17x_1 + 40x_2 - 20x_3 - 19x_4 - 38x_5 - 12x_6 - 31x_7 + 37x_8 + 30x_9 &= 180 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (géc + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75484 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A24.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 11 e coluna 20

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 17 e da coluna 67 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 83 graus
- cosseno de 23 graus
- tangente de 68 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

-38x_1 - 34x_2 + 20x_3 + 20x_4 + 18x_5 - 37x_6 - 25x_7 + 40x_8 - 25x_9 = -593
-22x_1 - 7x_2 + 38x_3 - 21x_4 - 5x_5 + 22x_6 - 25x_7 + 18x_8 + 25x_9 = 479
-26x_1 - 11x_2 - 14x_3 - 3x_4 + 20x_5 - 25x_6 - 3x_7 + 28x_8 - 24x_9 = -515
38x_1 - 13x_2 + 12x_3 - 21x_4 + 26x_5 + 30x_6 + 30x_7 - 4x_8 + 20x_9 = 120
-21x_1 - 31x_2 - 16x_3 + 4x_4 + 4x_5 + 25x_6 + 22x_7 - 6x_8 + 6x_9 = -292
27x_1 - 15x_2 - 23x_3 + 27x_4 - 6x_5 + 37x_6 - 22x_7 - 29x_8 - 18x_9 = 230
-33x_1 + 28x_2 + 2x_3 - 31x_4 - 18x_5 - 12x_6 + 30x_7 + 19x_8 + 34x_9 = 5
30x_1 - 2x_2 + 28x_3 + 27x_4 + 32x_5 + 38x_6 - 15x_7 + 29x_8 + 6x_9 = 187
28x_1 - 10x_2 - 32x_3 - 9x_4 - 15x_5 - 5x_6 + 17x_7 + 26x_8 + 27x_9 = -162

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

☞ Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75491 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A25.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 5 e coluna 16

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 23 e da coluna 20 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 25 graus
- cosseno de 12 graus
- tangente de 64 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

21x_1 - 7x_2 + 26x_3 + 25x_4 + 25x_5 + 12x_6 + 11x_7 + 32x_8 - 27x_9 = 433
34x_1 - 38x_2 - 29x_3 + 4x_4 - 24x_5 + 22x_6 + 29x_7 + 39x_8 - 23x_9 = 99
22x_1 - 36x_2 - 26x_3 - 6x_4 - 7x_5 + 23x_6 - 15x_7 + 37x_8 + 6x_9 = -51
-22x_1 - 21x_2 + 31x_3 + 8x_4 - 38x_5 - 36x_6 - 3x_7 - 11x_8 - 26x_9 = 371
3x_1 + 35x_2 + 5x_3 - 39x_4 + 36x_5 - 3x_6 + 30x_7 - 3x_8 + 19x_9 = 197
-34x_1 + 39x_2 - 29x_3 - 8x_4 - 14x_5 - 30x_6 + 33x_7 - 25x_8 - 38x_9 = 47
-38x_1 + 11x_2 - 9x_3 - 2x_4 + 34x_5 + 8x_6 + 40x_7 + 13x_8 - 5x_9 = 127
12x_1 - 8x_2 + 19x_3 + 13x_4 - 37x_5 + 14x_6 + 29x_7 + 27x_8 + 31x_9 = -93
5x_1 - 7x_2 + 14x_4 + 40x_5 - 14x_6 + 39x_7 + 10x_8 - 5x_9 = 193

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gdc + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75503 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A26.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 1 e coluna 15

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 6 e da coluna 24 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 22 graus
- cosseno de 38 graus
- tangente de 65 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

14x_1 + 25x_2 + 4x_3 - 6x_4 + 39x_5 - 37x_7 + 22x_8 - 31x_9 = -365
12x_1 + 32x_2 + 15x_3 - 6x_5 + 35x_6 - 2x_7 - 20x_8 + 27x_9 = -54
11x_1 + 30x_2 + 14x_3 - 39x_4 + 12x_5 + 29x_6 + 31x_7 - 24x_8 + 40x_9 = 84
25x_1 - 14x_2 + 23x_3 + 17x_4 - 11x_5 - 17x_6 - 25x_7 + 28x_8 + 13x_9 = 109
6x_1 - 36x_2 - 15x_3 + 5x_4 - 26x_5 + 5x_6 - 23x_7 + 30x_8 - 2x_9 = 62
7x_1 + 16x_2 - 20x_3 - 16x_4 + 18x_5 - 15x_6 + 35x_7 + 13x_8 - 24x_9 = 25
-12x_1 - 35x_2 - 8x_3 - 38x_4 - 17x_5 - 4x_6 - 14x_7 + 28x_8 - 19x_9 = -180
19x_1 - 2x_2 - 36x_3 - 22x_4 + 38x_5 - 4x_6 - 10x_7 + 34x_8 - 6x_9 = 80
34x_1 + 3x_2 - 11x_3 - 38x_4 - 17x_5 - 9x_6 - 26x_7 - 19x_8 - 18x_9 = -477

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75510 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A27.mxd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 11 e coluna 14

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A^{-1} à matriz inversa de A, tem-se A.A^{-1} = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 56 e da coluna 48 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 51 graus
- cosseno de 38 graus
- tangente de 80 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

2x_1 + 8x_2 - 4x_3 + 40x_4 - 37x_5 - 10x_6 - 23x_7 - 39x_8 - 32x_9 = -250
21x_1 + 32x_2 + 5x_3 + 16x_4 - 22x_5 - 27x_6 - 10x_7 - 16x_8 - 26x_9 = -400
3x_1 - 14x_2 + 31x_3 - 23x_4 + 15x_5 + 7x_6 + 8x_7 - 24x_8 + 14x_9 = 41
38x_1 + 38x_2 + 7x_3 + 12x_4 - 12x_5 + 27x_6 + 12x_7 - 25x_8 + 25x_9 = 60
17x_1 - 5x_2 - 13x_3 + 31x_4 - 24x_6 - 30x_7 - 32x_8 + 25x_9 = -425
37x_1 + 31x_2 + 37x_3 - 33x_4 - 9x_5 - 15x_6 - 12x_7 + 38x_8 - 17x_9 = -323
-36x_1 - 38x_2 - 22x_3 + 19x_4 - 5x_5 + 3x_6 + 29x_7 - 23x_8 + 23x_9 = 479
-11x_1 + 9x_2 + 31x_3 - 28x_4 + 28x_5 - 2x_6 + 17x_7 + 9x_8 + 7x_9 = 72
-36x_1 - 12x_2 + 12x_3 + 22x_4 - 23x_5 + 39x_6 - 4x_7 - 39x_8 - 13x_9 = 435

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r_1) e o da última (r_9).

6: r_1	6: r_9
--------	--------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} \times a_{2,2} - a_{2,1} \times a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for \neq 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r_1	6.r_9
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75527 - /

Prática em Numpy

O pacote Numpy é a base do desenvolvimento matemático em Python. Ele sozinho responde por boa parte da disciplina de Cálculo Numérico em qualquer faculdade de engenharia. Além disso, é pre-requisito para muitos outros pacotes (tensorflow, matplotlib, sympy, ...). Nesta folha imagina-se que logo no início da sessão Python emitiu-se o comando

```
>>> import numpy as np
```

Por essa razão todos os comandos de numpy vão ser prefixados por np. Na sequência, você deve ler e processar o arquivo de nome

F936A28.myd

que deve estar publicado no lugar usual. Cada conjunto de dados está separado dos demais por uma linha de comentários que deve separar os arquivos de maneira que cada exercício leia apenas o bloco de seus dados. Cabe a você fazer estas manipulações manualmente (ou automaticamente, você escolhe).

1. O primeiro assunto é a leitura de dados sequenciais e o cálculo automatizado de diversas medidas estatísticas. No início do arquivo você deve encontrar uma matriz de 100 linhas por 8 colunas (800 valores) inteiros. Deve ler tais dados usando o comando np.loadtxt e depois deve comandar o cálculo da média (x.aver()) e do desvio padrão da amostra (x.var()*0.5).

1: med	1: dp
--------	-------

2. O poker é um jogo fascinante. No mundo da programação, simular jogos de baralho é excelente ferramenta para lidar com simulações e aleatoriedade. Vai se trabalhar aqui com um baralho francês de cartas, representadas por um número inteiro, a saber 1:A, 2 a 10: a própria carta, 11: dama, 12: valete e 13: rei. Os naipes também serão representados por um inteiro. Aqui, 1: ouros, 2: espadas, 3: copas e 4: paus. Cada carta será um conjunto de 2 inteiros: a carta e o naipe. Você deve ler no mesmo arquivo acima citado, um conjunto de 100 baralhos devidamente embaralhados. Para cada baralho pegue as primeiras 25 cartas e distribua-as para 5 jogadores. Haverá portanto 500 mãos a analisar. Dessas 500 mãos, conte quantas delas têm:

- 1. Quadra: 4 das 5 cartas são do mesmo valor
- 2. Full-hand: 3 cartas são iguais entre si e as outras duas também são iguais entre si.
- 3. Flush: as 5 cartas são do mesmo naipe
- 4. Sequência: as 5 cartas são seguidas, independentemente do naipe.

O comando a usar aqui é a **Leitura**. A leitura dos dados dos 100 baralhos deve ser feita usando-se a=np.loadtxt('arq'). Depois deve-se ficar apenas com as primeiras 25 cartas desprezando-se as demais desse baralho. Finalmente, as 2500 cartas (ou 500 mãos de 5 cartas cada) devem ser analisadas para buscar quantos jogos quadra somada a full-hand, flush e sequência apareceram.

2: quadras+fh	2: flushes	2: seqs
---------------	------------	---------

3. A multiplicação matricial é uma operação importante em diversos ramos da ciência e da tecnologia, em particular na busca de soluções para sistemas lineares. Como se sabe, dadas as matrizes A e B, a multiplicação R = A.B é dada por

$$r_{i,k} = \sum_{j=1}^n a_{i,j} \times b_{j,k}$$

. Neste exercício você deve ler a matriz A de 20 linhas por 5 colunas e a matriz B de 5 colunas por 30 linhas e calcular a matriz R de 20 linhas por 30 colunas. Em numpy, a multiplicação matricial é calculada pelo comando np.dot(a,b). lembre que esta multiplicação não é comutativa. Responda o valor da linha 4 e coluna 18

3. elemento individual

4. Associado à multiplicação matricial, temos o conceito da matriz inversa, que é aquela que quando multiplicada matricialmente à matriz que lhe deu origem gera a matriz identidade. Chamando A⁻¹ à matriz inversa de A, tem-se A.A⁻¹ = I onde I é a identidade, formada por zeros, exceto a diagonal principal, que vale 1. Em numpy a matriz inversa é calculada pelo comando np.linalg.inv(a) Leia no arquivo a matriz de 80 linhas por 80 colunas e calcule a sua inversa. ache o elemento da inversa da linha 11 e da coluna 8 multiplique-o por 1000 e informe o resultado encontrado.

4. elemento da inversa multiplicado por 1000:

5. Tabela de senos, cossenos e tangente. O exercício agora é a produção de uma tabela contendo 4 colunas: o arco, de grau em grau, começando em zero e terminando em 90 (91 linhas) e as colunas de seno, cosseno e tangente para cada arco. Faça isso e depois responda, quanto é:

- seno de 37 graus
- cosseno de 68 graus
- tangente de 4 graus.

5: seno	5: cosseno	5: tangente
---------	------------	-------------

6. Solução de sistemas lineares. Suponha um sistema com 9 equações e 9 incógnitas, formado pelas equações

$$\begin{aligned} -25x_1 + 33x_2 - 29x_3 + 23x_4 - 11x_5 - 29x_6 - 19x_7 - 4x_8 + 37x_9 &= -10 \\ -10x_1 - 26x_2 - 22x_3 + 4x_4 - 29x_5 - 10x_6 + 8x_7 + 36x_8 - 11x_9 &= -20 \\ -37x_1 - 18x_2 + 24x_3 - 23x_4 - 28x_5 - 26x_6 - 35x_7 + 3x_8 + 8x_9 &= -350 \\ 2x_1 - 10x_2 - 4x_3 - 31x_4 + 7x_5 + 2x_6 + 37x_7 - 28x_8 - 14x_9 &= 148 \\ 16x_1 + 23x_2 - 4x_3 + 32x_4 + 30x_5 - 12x_6 - 3x_7 + 6x_8 + 32x_9 &= 401 \\ 33x_1 + 19x_2 + 12x_3 - 30x_4 + 5x_5 - 17x_6 + 16x_7 + 30x_8 + 13x_9 &= 367 \\ -8x_1 - 1x_2 + 29x_3 + 23x_5 - 6x_6 + 2x_7 - 36x_8 &= 54 \\ -3x_1 - 18x_2 + 15x_3 - 8x_4 - 14x_5 + 35x_6 + 23x_7 - 4x_8 + 2x_9 &= -222 \\ 36x_1 - 8x_2 - 18x_3 - 14x_4 - 38x_5 + 6x_6 + 40x_7 - 20x_8 - 8x_9 &= 202 \end{aligned}$$

Resolva-o, usando o comando np.linalg.solve(a,b) onde a é a matriz de coeficientes do sistema e b o vetor de termos independentes. Informe no quadro a seguir o valor da primeira raiz (r₁) e o da última (r₉).

6: r ₁	6: r ₉
-------------------	-------------------

7. O determinante de uma matriz quadrada é um número real que é calculado sobre os elementos da matriz. Em uma matriz 1, 1 que só tem 1 elemento o determinante é o próprio. Em uma matriz A de dimensões 2, 2, ele é resultado de a_{1,1} × a_{2,2} - a_{2,1} × a_{1,2}. Para uma matriz A de dimensões 3, 3 e contendo os elementos a, b, c na primeira linha e d, e, f na segunda e g, h, i na terceira o determinante é obtido calculando-se aei + bfg + dhc - (gec + ahf + dbi). Para a quarta ordem e as seguintes, há teoremas que permitem rebaixar a ordem, sempre pagando o preço de calcular inúmeros determinantes de ordem menor. Uma função imediata do determinante é saber se uma matriz tem ou não inversa: ela só terá se o seu determinante for ≠ 0. O comando que acha o determinante em numpy é np.linalg.det(a). Leia no arquivo uma matriz quadrada de ordem 5 (5 linhas por 5 colunas) calcule e transcreva aqui o valor do seu determinante.

7. determinante

🔗 Para você fazer

Responda aqui:

1.m	1.dp	2.q+fh	2.f
2.s	3.mm	4.i	5.s
5.c	5.t	6.r ₁	6.r ₉
7.d	///////// ///////// /////////	///////// ///////// /////////	///////// ///////// /////////



104-75534 - /