Aula Prática 4 (ALN)

Raphael F. Levy

May 19, 2021

1 Introdução

Para a Aula Prática 4, o método a ser estudado é o Método dos Mínimos Quadrados, que é usado para encontrar uma aproximação linear para retas impossíveis dados os pontos passados. Para isso, é preciso que sejam passados as coordenadas dos pontos para que a matriz A e o vetor b possam ser criados:

$A=$
$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ ... & ... \\ 1 & x_n \end{bmatrix}$$

$b=$
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ ... \\ y_n \end{bmatrix}$$

Com isso, podemos achar o vetor $x = [a; b]$ que minimiza o erro quadrático desses pontos através da seguinte fórmula: $(A^T A)x = A^T b$, que pode ser resolvida utilizando o método *Gaussian_Elimination_4*, para separar x de $A^T A$.

2 Questão 1

Para a questão 1, o livro nos passa a seguinte tabela:

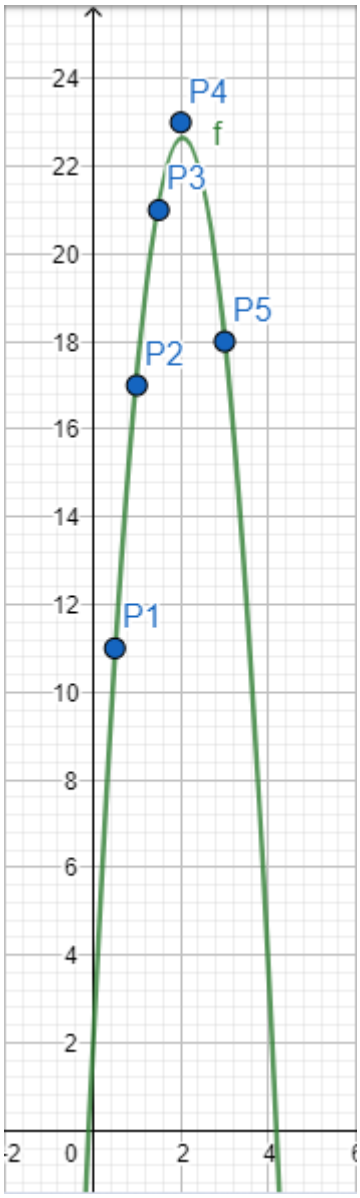
Tempo (s): 0.5; 1; 1.5; 2; 3
Altura (m): 11; 17; 21; 23; 18

1a) Encontre a aproximação quadrática por mínimos quadrados para esses dados.

Solução:

```
--> A=[1 0.5 0.25; 1 1 1; 1 1.5 2.25; 1 2 4; 1 3 9]
 A  =

   1.    0.5   0.25
   1.    1.    1.
   1.    1.5   2.25
   1.    2.    4.
   1.    3.    9.

--> A'
 ans  =

   1.      1.    1.      1.    1.
   0.5     1.    1.5     2.    3.
   0.25    1.    2.25    4.    9.

--> b=[11;17;21;23;18]
 b  =

   11.
   17.
   21.
   23.
   18.

--> AtA=A'*A
 AtA  =

   5.      8.      16.5
   8.      16.5    39.5
   16.5    39.5    103.125

--> Atb=A'*b
 Atb  =

   90.
   154.
   321.

--> [x]=Gaussian_Elimination_4_AP3(AtA,Atb)
 x  =

   1.9175258
   20.306333
  -4.9720177

--> x=AtA\Atb
 x  =

   1.9175258
   20.306333
  -4.9720177
```

Utilizando o cálculo de divisão de matrizes apenas para garantir a resposta, e confirmando que a *Gaussian_Elimination_4* encontrou o mesmo vetor x, descobrimos que a melhor aproximação linear para esse conjunto de pontos é $y = 1.9175258 + 20.306333x - 4.9720177x^2$.



Aproximação linear para os pontos dados

1b) Estime a altura na qual o objeto foi solto (em m), sua velocidade inicial (em m/s) e sua aceleração da gravidade (em m/s^2).

Solução: Sabendo que o vetor x encontrado é $[s_0, v_0, g/2]$, podemos afirmar que a altura inicial da qual o objeto foi lançado é de $1.9175258m$, a velocidade

é de $20.306333m/s$ e a gravidade será $-9.9440354m/s^2$.

1c) Quando, aproximadamente, o objeto irá atingir o chão?

Solução: Resolvendo essa equação do segundo grau, achamos duas soluções: 4.176465316151968 e −0.09234208384730486. Como se trata de uma resposta em relação ao tempo, a resposta deve ser positiva, e portanto o tempo que o objeto vai levar pra atingir o chão é de aproximadamente 4.17 segundos.

3 Questão 2

Para a questão 2, o livro nos passa a seguinte tabela:

Ano: 1950; 1960; 1970; 1980; 1990; 2000
População (em milhões): 150; 179; 203; 227; 250; 281

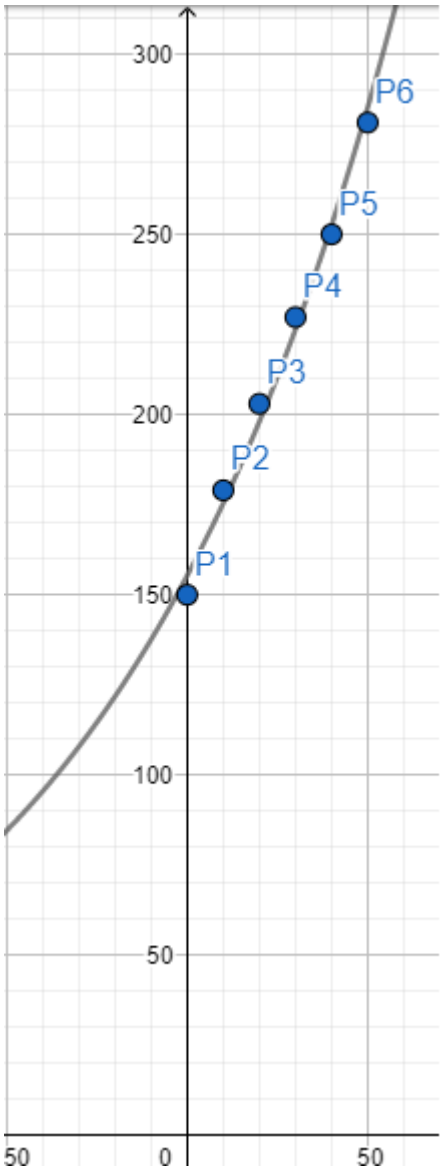
2a) Supondo um modelo de crescimento exponencial da forma $p(t) = ce^{kt}$, em que $p(t)$ é a população em um tempo t, utilize mínimos quadrados para encontrar a equação para a taxa de crescimento da população. [Sugestão: considere t=0 para 1950]

Solução: $p(t) = ce^{kt} \Rightarrow lnp = lnc + kt$

```
--> A=[1 0; 1 10; 1 20; 1 30; 1 40; 1 50]
 A  =

   1.   0.
   1.   10.
   1.   20.
   1.   30.
   1.   40.
   1.   50.

--> A'
 ans  =

   1.   1.    1.    1.    1.    1.
   0.   10.   20.   30.   40.   50.

--> p=[150;179;203;227;250;281]
 p  =

   150.
   179.
   203.
   227.
   250.
   281.

--> b=log(p)
 b  =

```

```
   5.0106353
   5.1873858
   5.313206
   5.42495
   5.5214609
   5.6383547

--> AtA=A'*A
 AtA  =

   6.      150.
   150.    5500.
--> Atb=A'*b
 Atb  =

   32.095993
   823.66265

--> [x]=Gaussian_Elimination_4_AP3(AtA,Atb)
 x  =

   5.0455774
   0.0121502

--> x=AtA\Atb
 x  =

   5.0455774
   0.0121502
```

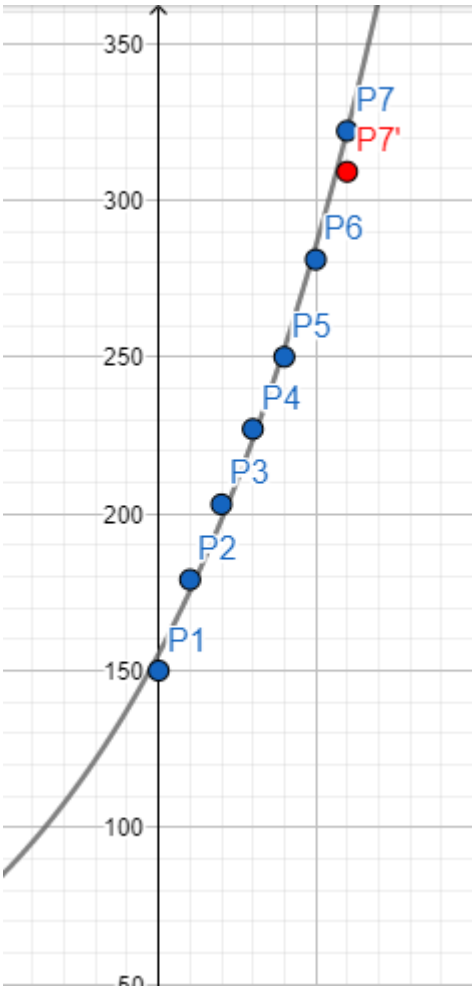
Utilizando o cálculo de divisão de matrizes apenas para garantir a resposta, e confirmando que a *Gaussian_Elimination_4* encontrou o mesmo vetor x, descobrimos que a melhor aproximação linear para esse conjunto de pontos é $ln(p) = 5.0455774 + 0.0121502t \Rightarrow p = e^{5.0455774+0.0121502t} \Rightarrow p = 155.334 * e^{0.0121502t}; c = 155.334, k = 0.0121502$.



Aproximação linear para os pontos dados

2b) Use a equação obtida para estimar a população dos Estados Unidos em 2010.

Solução: Em 2010, $t = 60$, então $p = 155.334 * e^{0.0121502*60} \approx 322$ milhões de habitantes. De acordo com o censo da população estadunidense em 2010, a população era de aproximadamente 309 milhões de habitantes.



Aproximação linear para a população em 2010

Pela figura, é possível ver que a estimativa da população em 2010 (P7) foi bem próxima da medida real (P7'), tendo ficado ainda mais próxima da aproximação calculada que o valor real.

Ainda, o livro nos mostra uma forma diferente de estimação para um modelo de crescimento exponencial:

CAS **Exemplo 7.29**

A tabela 7.2 apresenta a população do mundo para intervalos de dez anos, referente à segunda metade do século XX. Supondo um modelo de crescimento exponencial, encontre a taxa de crescimento relativo e preveja a população do mundo para 2010.

Tabela 7.2

Ano	População (em bilhões)
1950	2,56
1960	3,04
1970	3,71
1980	4,46
1990	5,28
2000	6,08

Fonte: U.S. Bureau of the Census, International Data Base

Solução Vamos medir o tempo t em intervalos de dez anos. Com isso, $t = 0$ é 1950, $t = 1$ é 1960, e assim por diante. Como $c = p(0) = 2{,}56$, a equação para a taxa de crescimento da população é

$$p = 2{,}56e^{kt}$$

Como podemos utilizar o método dos mínimos quadrados nessa equação? Se calcularmos o logaritmo neperiano de ambos os lados, alteramos a equação para uma linear:

$$\begin{aligned} \ln p &= \ln(2{,}56e^{kt}) \\ &= \ln 2{,}56 + \ln(e^{kt}) \\ &\approx 0{,}94 + kt \end{aligned}$$

Substituindo os valores de t e de p da tabela 7.2, chegamos ao seguinte sistema (onde arredondamos os valores para a terceira casa decimal):

$$\begin{aligned} 0{,}94 &= 0{,}94 \\ k &= 0{,}172 \\ 2k &= 0{,}371 \\ 3k &= 0{,}555 \\ 4k &= 0{,}724 \\ 5k &= 0{,}865 \end{aligned}$$

Podemos ignorar a primeira equação (que só corresponde à condição inicial $c = p(0) = 2{,}56$). As outras equações correspondem a um sistema $A\mathbf{x} = \mathbf{b}$, com

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \quad \text{e} \quad \mathbf{b} = \begin{bmatrix} 0{,}172 \\ 0{,}371 \\ 0{,}555 \\ 0{,}724 \\ 0{,}865 \end{bmatrix}$$

Como $A^TA = 55$ e $A^T\mathbf{b} = 9{,}80$, as equações normais correspondentes são somente a equação

$$55\bar{\mathbf{x}} = 9{,}80$$

Portanto, $k = \bar{\mathbf{x}} = 9{,}80/55 \approx 0{,}178$. Consequentemente, a solução por mínimos quadrados tem a forma $p = 2{,}56e^{0{,}178t}$ (veja a figura 7.16).

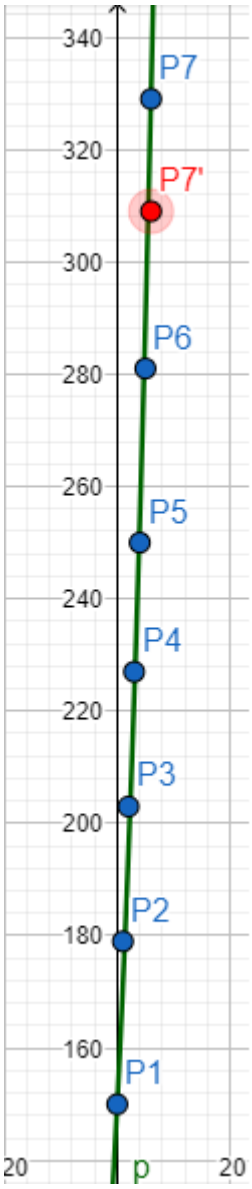
A população mundial em 2010 corresponde a $t = 6$, de onde obtemos

$$p(6) = 2{,}56e^{0{,}178(6)} \approx 7{,}448$$

Nesse caso, consideraríamos $c = p(0) = 150$ e $p = ce^{kt} \Rightarrow ln(p) = ln(c) + kt \Rightarrow kt = ln(p) - ln(c) = ln(p) - 5.01$:

```
-->  A=[0; 1; 2; 3; 4; 5];

--> A'
 ans  =

   0.   1.   2.   3.   4.   5.

--> p=[150;179;203;227;250;281];

--> b=log(p);

--> kt=(b-5.01)
 kt  =

   0.0006353
   0.1773858
   0.303206
   0.41495
   0.5114609
   0.6283547

--> AtA=A'*A
 AtA  =

   55.

--> Atb=A'*kt
 Atb  =

   7.2162648

--> [x]=Gaussian_Elimination_4_AP3(AtA,Atb)
 x  =

   0.1312048
```

Assim, nossa equação seria $p = ce^kt = 150e^{0.1312048*t}$, sendo $t = 6 \Rightarrow 150e^{0.1312048*6} = 329.6$



Aproximação linear para a população em 2010

Pela figura, é possível ver que essa equação também é uma boa aproximação para a população estadunidense, embora o valor estimado em 2010 seja ainda maior que o verdadeiro nesse caso, indicado por P7'. Assim, embora o algoritmo não estime quem é c nesse caso, é visível que ambas as formas, tanto a indicada pelo livro quanto a ensinada em sala, são boas aproximações para o cálculo da população.

4 Questão 3

Para a questão 3, o livro nos passa a seguinte tabela:

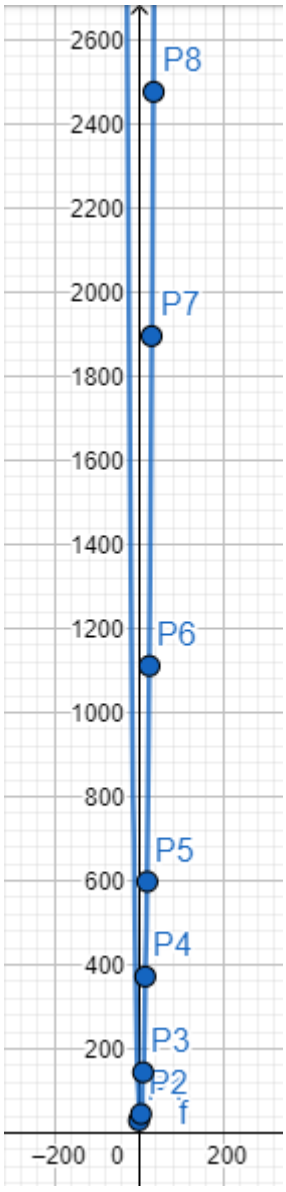
Ano: 1970; 1975; 1980; 1985; 1990; 1995; 2000; 2005
Média salarial (em milhares): 29.3; 44.7; 143.8; 371.6; 597.5; 1110.8; 1895.6; 2476.6

3a) Encontre a aproximação quadrática por mínimos quadrados para esses dados. [Observação: usaremos $t = 0$ para 1970]

```
--> A=[1 0 0; 1 5 25; 1 10 100; 1 15 225; 1 20 400; 1 25 625; 1 30
    900; 1 35 1225]
 A  =

   1.   0.    0.
   1.   5.    25.
   1.   10.   100.
   1.   15.   225.
   1.   20.   400.
   1.   25.   625.
   1.   30.   900.
   1.   35.   1225.

--> A'
 ans  =

   1.   1.    1.     1.     1.     1.     1.     1.
   0.   5.    10.    15.    20.    25.    30.    35.
   0.   25.   100.   225.   400.   625.   900.   1225.

--> b=[29.3; 44.7; 143.8; 371.6; 597.5; 1110.8; 1895.6; 2476.6]
 b  =

   29.3
   44.7
   143.8
   371.6
   597.5
   1110.8
   1895.6
   2476.6

--> AtA=A'*A
 AtA  =

   8.       140.     3500.
   140.     3500.    98000.
   3500.    98000.   2922500.

--> Atb=A'*b
 Atb  =

   6669.9
   190504.5
   5772232.5

```

```
--> [x]=Gaussian_Elimination_4_AP3(AtA,Atb)
 x  =

   57.0625
  -20.334643
   2.5886429

--> x=AtA\Atb
 x  =

   57.0625
  -20.334643
   2.5886429
```

Fazendo a aproximação quadrática, encontramos $y = 57.0625 - 20.334643x + 2.5886429x^2$



Aproximação quadrática para os pontos dados

Apesar do gráfico não ser muito claro devido à proximidade dos pontos no eixo x e à grande distância no eixo y, é possível perceber que os pontos se encaixaram bem à reta criada.

```
--> Ea1=29.3-(57.0625-20.334643*(0)+2.5886429*((0)^2))
 Ea1   =
```

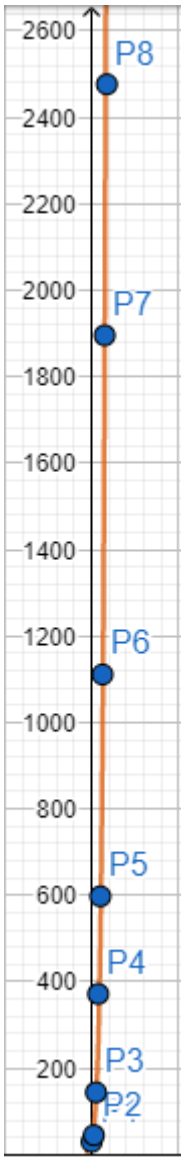
```
  -27.7625

--> Ea2=44.7-(57.0625-20.334643*(5)+2.5886429*((5)^2))
 Ea2  =

   24.594643

--> Ea3=143.8-(57.0625-20.334643*(10)+2.5886429*((10)^2))
 Ea3  =

   31.219640

--> Ea4=371.6-(57.0625-20.334643*(15)+2.5886429*((15)^2))
 Ea4  =

   37.112493

--> Ea5=597.5-(57.0625-20.334643*(20)+2.5886429*((20)^2))
 Ea5  =

  -88.326800

--> Ea6=1110.8-(57.0625-20.334643*(25)+2.5886429*((25)^2))
 Ea6  =

  -55.798237

--> Ea7=1895.6-(57.0625-20.334643*(30)+2.5886429*((30)^2))
 Ea7  =

   118.79818

--> Ea8=2476.6-(57.0625-20.334643*(35)+2.5886429*((35)^2))
 Ea8  =

  -39.837548

--> ea=sqrt((Ea1)^2+(Ea2)^2+(Ea3)^2+(Ea4)^2+(Ea5)^2+(Ea6)^2+(Ea7)
    ^2+(Ea8)^2)
 ea  =

   174.19173
```

3b) Encontre a aproximação exponencial por mínimos quadrados para esses dados. [Observação: usaremos $t = 0$ para 1970]

```
--> A=[1 0; 1 5; 1 10; 1 15; 1 20; 1 25; 1 30; 1 35]
 A  =

   1.   0.
   1.   5.
   1.   10.
   1.   15.
   1.   20.
   1.   25.
   1.   30.
   1.   35.

```

```
--> A'
 ans  =

   1.   1.   1.    1.    1.    1.    1.    1.
   0.   5.   10.   15.   20.   25.   30.   35.

--> p=[29.3; 44.7; 143.8; 371.6; 597.5; 1110.8; 1895.6; 2476.6]
 p  =

   29.3
   44.7
   143.8
   371.6
   597.5
   1110.8
   1895.6
   2476.6

--> b=log(p)
 b  =

   3.3775875
   3.7999735
   4.9684234
   5.917818
   6.3927543
   7.0128358
   7.5472907
   7.8146419

--> AtA=A'*A
 AtA  =

   8.     140.
   140.   3500.

--> Atb=A'*b
 Atb  =

   46.831325
   960.55854

--> [x]=Gaussian_Elimination_4_AP3(AtA,Atb)
 x  =

   3.5037431
   0.1342956

--> x=AtA\Atb
 x  =

   3.5037431
   0.1342956
```

Fazendo a aproximação exponencial, encontramos $ln(y) = 3.5037431 + 0.1342956x \Rightarrow y = e^{3.5037431+0.1342956x}$



Aproximação exponencial para os pontos dados

Apesar do gráfico não ser muito claro devido à proximidade dos pontos no eixo x e à grande distância no eixo y, é possível perceber que os pontos se encaixaram bem à aproximação feita.

```
--> Eb1=29.3-exp(3.5037431+(0.1342956*0))
 Eb1   =

  -3.9396387
```

```
--> Eb2=44.7-exp(3.5037431+(0.1342956*5))
 Eb2  =

  -20.354222

--> Eb3=143.8-exp(3.5037431+(0.1342956*10))
 Eb3  =

   16.480573

--> Eb4=371.6-exp(3.5037431+(0.1342956*15))
 Eb4  =

   122.41961

--> Eb5=597.5-exp(3.5037431+(0.1342956*20))
 Eb5  =

   109.82212

--> Eb6=1110.8-exp(3.5037431+(0.1342956*25))
 Eb6  =

   156.35206

--> Eb7=1895.6-exp(3.5037431+(0.1342956*30))
 Eb7  =

   27.623385

--> Eb8=2476.6-exp(3.5037431+(0.1342956*35))
 Eb8  =

  -1179.2690

--> eb=sqrt((Eb1)^2+(Eb2)^2+(Eb3)^2+(Eb4)^2+(Eb5)^2+(Eb6)^2+(Eb7)
    ^2+(Eb8)^2)
 eb  =

   1201.5129
```

3c) Qual equação dá uma melhor aproximação? Por quê?

Solução: Já que não podemos considerar os gráficos como uma resposta confiável para a melhor aproximação dentre as equações, podemos utilizar seu erro quadrático: $||e|| = \sqrt{(E_1^2 + E_2^2 + ... + E_n^2)}$. Assim, como os erros já foram calculados anteriormente (ea e eb), temos que a equação que dá a melhor aproximação é a quadrática, já que $ea = 174.19173 < eb = 1201.5129$.

3d) Qual a sua estimativa para a média salarial da liga adulta de beisebol em 2010 e em 2015?

Solução: Usando o método quadrático temos: $y = -395.95 + 70.267857x$. Para $t = 40$ e $t = 45$ temos que o salário será de respectivamente de 2414.7643 e

2766.1036 mil dólares, ou 2.414.7643 e 2.766.1036 milhões de dólares, respectivamente. Usando o método exponencial, para $t = 40$ e $t = 45$, o salário será de 7155.0029 e 14003.255 mil de dólares, ou 7.155.0029 e 14.003.255 milhões. Usando a estimativa quadrática, visto que seu erro é menor que o exponencial, temos um salário próximo ao que sabemos que foi pago em 2005. Além disso, é notável que, embora o salário estimado de 2010 seja menor que o salário ganho em 2005, que foi de 2.476.6 milhões de dólares, estimando o salário para 2005 com a equação temos como resposta um valor menor que o verdadeiro, de 2.063.4250 milhões, então não é impossível que o valor pago seja maior que o estimado.

5 Questão 4

Para essa questão, queremos obter o hiperplano $y = h(x) = \alpha_0 + \sum_{i=1}^{10} \alpha_i * x_i$ que melhor se ajuste aos dados passados usando o método dos mínimos quadrados, já que o sistema $h(x) = x * \alpha$ é um sistema impossível. Com isso, podemos encontrar a matriz de confusão criada, já que teremos quantos resultados foram estimados de forma correta ou incorreta comparando nosso modelo com os dados originais.

Indo para o Scilab, primeiramente, carregamos os arquivos *cancer_test.csv* e *cancer_train.csv*, e depois executamos os seguintes comandos:

```
--> A_tr = csvRead('cancer_train.csv');

--> A_te = csvRead('cancer_test.csv');

--> Y_tr = A_tr(:,11);

--> X_tr = [ones(Y_tr) A_tr(:,1:10)];

--> Y_te = A_te(:,11);

--> X_te = [ones(Y_te) A_te(:,1:10)];
```

Dando sequência ao código anterior para que possamos resolver as Equações Normais:

```
--> alfa_tr = Gaussian_Elimination_4_AP3(X_tr'*X_tr,X_tr'*Y_tr)
 alfa_tr  =

  -6.7579731
   29.311052
   2.0765803
  -18.730222
  -7.3665161
   1.2222756
   0.2283419
   0.0503253
   2.2385058
   0.0249405
   0.7704282
```

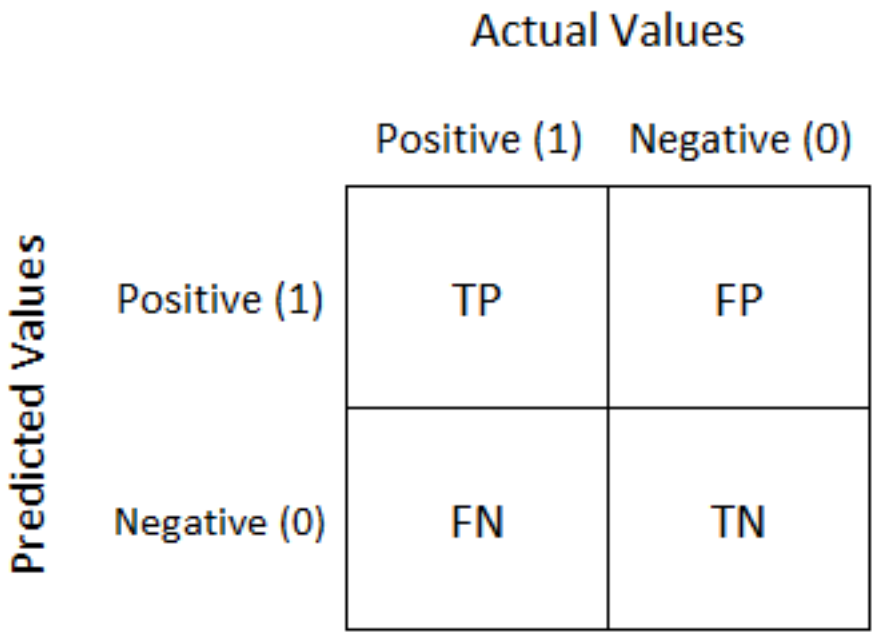
```
--> prev_tr = X_tr * alfa_tr;

--> conf_prev_tr = prev_tr .* Y_tr;

--> acertos_tr = sum(conf_prev_tr>0|conf_prev_tr==0)
 acertos_tr  =

   279.

--> erros_tr = sum(conf_prev_tr<0|conf_prev_tr==0)
 erros_tr  =

   21.

--> prev_te = X_te * alfa_tr;

--> conf_prev_te = prev_te .* Y_te;

--> acertos_te = sum(conf_prev_te>0|conf_prev_te==0)
 acertos_te  =

   185.

--> erros_te = sum(conf_prev_te<0|conf_prev_te==0)
 erros_te  =

   75.
```

Encontrando a quantidade acertos, é possível ver que o diagnóstico foi bem adequado para o arquivo de treino, tendo acertado 93% dos casos, não tão bem para o de teste, com 71.152% de acerto. Dessa forma, é possível ver que as predições com os dados de treino foram bem feitas, mas não exatamente a de teste, já que 30% das pacientes acabará recebendo um diagnóstico falso. Agora que o hiperplano está feito e adequado, podemos construir as matrizes de confusão desses dados.

```
--> h_tr = X_tr * alfa_tr;

--> h_tr_class = h_tr>=0;

--> h_tr_ones = ones(h_tr_class);

--> h_tr_class = 2*(h_tr_class)-h_tr_ones;

--> positivos_tr = sum(h_tr_class>0)
 positivos_tr  =

   145.

--> negativos_tr = sum(h_tr_class<0)
 negativos_tr  =

   155.

--> SomaPosTr = sum(Y_tr==1)
 SomaPosTr  =
```

```
 146.

--> SomaNegTr = sum(Y_tr==-1)
 SomaNegTr  =

 154.

--> h_te = X_te * alfa_tr;

--> h_te_class = h_te>=0;

--> h_te_ones = ones(h_te_class);

--> h_te_class = 2*(h_te_class)-h_te_ones;

--> positivos_te = sum(h_te_class>0)
 positivos_te  =

 135.

--> negativos_te = sum(h_te_class<0)
 negativos_te  =

 125.

--> SomaPosTe = sum(Y_te==1)
 SomaPosTe  =

 60.

--> SomaNegTe = sum(Y_te==-1)
 SomaNegTe  =

 200.
```

Observando os valores obtidos na predição, é possível ver que o método teve números bem próximos ao resultados verdadeiros, como pode ser visto comparando *positivos*, que são valores positivos indicados pela predição ($TP+FP$) e *negativos*, que são valores negativos indicados pela predição ($TN+FN$), com $SomaPos$ e $SomaNeg$, que são a soma de valores positivos ($TP+FN$) e negativos ($TN+FP$) medida nos arquivos, em ambos. Fazemos a matriz:



Exemplo de Matriz de Confusão utilizada, no caso substituindo valores negativos por -1 ao invés de 0

Matriz de Treino:

$$\begin{cases} TP + FP = 145 \\ TP + FN = 146 \\ TN + FP = 154 \\ TN + FN = 155 \end{cases}$$

$$
\begin{cases}
TP + FP = 145 \\
TP + FN = 146 \\
FN - FP = 1 \Rightarrow FN = FP + 1 \\
\\
TP + FP = 145 \\
TN + FP = 154 \\
TN - TP = 9 \Rightarrow TN = TP + 9 \\
\\
TP + TN = 279 \Rightarrow TP + TP + 9 = 279 \Rightarrow 2TP = 270 \Rightarrow TP = 135 \\
TP + FP = 145 \Rightarrow FP = 10 \\
TN = TP + 9 \Rightarrow TN = 144 \\
FN = FP + 1 \Rightarrow FN = 11
\end{cases}
$$

TP	FP
FN	TN

$\Downarrow$

135	10
11	144

Matriz de Teste:

$$
\begin{cases}
TP + FP = 135 \\
TP + FN = 60 \\
TN + FP = 200 \\
TN + FN = 125
\end{cases}
$$

$$
\begin{cases}
TP + FP = 135 \\
TP + FN = 60 \\
FN - FP = -75 \Rightarrow FN = FP - 75 \\
\\
TP + FP = 135 \\
TN + FP = 200 \\
TN - TP = 65 \Rightarrow TN = TP + 65 \\
\\
TP + TN = 185 \Rightarrow TP + TP + 65 = 185 \Rightarrow 2TP = 120 \Rightarrow TP = 60 \\
TP + FP = 135 \Rightarrow FP = 75 \\
TN = TP + 65 \Rightarrow TN = 125 \\
FN = FP - 75 \Rightarrow FN = 0
\end{cases}
$$

TP	FP
FN	TN

$\Downarrow$

60	75
0	125

Fazendo as matrizes de confusão, foi possível encontrar a quantidade de positivos e negativos verdadeiros assim como falsos negativos e positivos. Com isso, é possível ver mais uma vez como a predição foi correta na grande maioria dos testes. É possível também calcular as medidas decorrentes dessas matrizes:

- Acurácia: $(TP + TN)/(TP + TN + FP + FN)$, Valores Corretos/Todos os Valores
- Erro de Classificação: $(FP + FN)/(TP + TN + FP + FN)$, Valores Incorretos/Todos os Valores
- Precisão: $(TP)/(TP + FP)$, Positivos Verdadeiros/Positivos Estimados
- Recall/Sensitividade: $(TP)/(TP + FN)$, Positivos Verdadeiros/Positivos Corretos
- Especificidade: $(TN)/(TN + FP)$, Negativos Verdadeiros/Negativos Corretos
- Probabilidade de Falso Alarme: $(FP)/(TP + FP)$, Falsos Positivos/Positivos Estimados

- Probabilidade de Falsa Omissão de Alarme: $(FN)/(TN + FN)$, Falsos Negativos/Negativos Estimados

Aqui, a acurácia representa a porcentagem de acertos da predição em comparação com os valores previamente medidos, enquanto o erro de classificação representa a porcentagem de erros da predição em comparação com os valores previamente medidos. A predição mostra quantos dos valores preditos positivos eram verdadeiramente positivos, enquanto o recall nos diz quantos positivos verdadeiros conseguimos predizer corretamente com nosso modelo. A especificidade é o contrário do recall, mostrando quantos negativos verdadeiros conseguimos predizer corretamente com nosso modelo.

Matriz de Treino:

- Acurácia: $(135 + 144)/(135 + 144 + 10 + 11) = 279/300 = 0.93 = 93\%$ (Note que essa probabilidade já havia sido calculada anteriormente para achar a probabilidade de valores acertados)
- Erro de Classificação: $(10+11)/(135+144+10+11) = 21/300 = 0.07 = 7\%$
- Precisão: $(135)/(135 + 10) = 135/145 \approx 0.93103 = 93.103\%$
- Recall/Sensitividade: $(135)/(135 + 11) = 135/146 \approx 0.92465 = 92.465\%$
- Especificidade: $(144)/(144 + 10) = 144/154 \approx 0.93506 = 93.506\%$
- Probabilidade de Falso Alarme: $(10)/(135 + 10) = 10/145 \approx 0.06896 = 6.896\%$
- Probabilidade de Falsa Omissão de Alarme: $(11)/(144 + 11) = 11/155 \approx 0.07096 = 7.096\%$

Matriz de Teste:

- Acurácia: $(60 + 125)/(60 + 125 + 75 + 0) = 185/260 \approx 0.71153 = 71.153\%$ (Note que essa probabilidade já havia sido calculada anteriormente para achar a probabilidade de valores acertados)
- Erro de Classificação: $(75 + 0)/(60 + 125 + 75 + 0) = 75/260 \approx 0.28846 = 28.846\%$
- Precisão: $(60)/(60 + 75) = 60/135 \approx 0.44444 = 44.444\%$
- Recall/Sensitividade: $(60)/(60 + 0) = 60/60 = 1 = 100\%$
- Especificidade: $(125)/(125 + 75) = 125/200 = 0.625 = 62.5\%$
- Probabilidade de Falso Alarme: $(75)/(60 + 75) = 75/135 \approx 0.55555 = 55.555\%$
- Probabilidade de Falsa Omissão de Alarme: $(0)/(125 + 0) = 0/125 = 0\%$

Note que a acurácia e o erro de classificação, assim como a precisão e a probabilidade de falso alarme são complementares.

Analisando as medições pelas matrizes de confusão, podemos ver que a previsão dos dados foi bem feita para a matriz de treino, mas não tão adequada para a de teste. Os dados de treino tiveram uma acurácia bem alta, de 93%, enquanto os de teste tiveram uma acurácia mais mediana, de 71.153%, então a quantidade de valores estimados corretamente foi pelo menos maior que a média em ambas, e maior que 90% nos dados de treino.

A precisão de treino foi novamente acima de 90%, porém a precisão de teste foi bem ruim, por volta de 44.44%, ou seja, mais valores estimados positivos foram verdadeiramente positivos na matriz de treino que na de teste.

O recall, positivos verdadeiros que conseguimos medir com o nosso modelo, foi muito bom nas duas, sendo de 100% para os dados de teste já que não achamos nenhum falso negativo.

A especificidade novamente teve uma grande variação entre eles, visto que no arquivo de teste nosso modelo previu um número bem alto de falsos positivos, maior até do que positivos verdadeiros. A previsão de treino, por outro lado foi bem específica, já que de 145 resultados medidos positivos, apenas 10 eram falsos.

Analisando as probabilidades de alarme falso, é visível que o modelo de treino foi bem adequado, com uma probabilidade menor que 10% de devolver tanto um alarme falso quanto omitir um verdadeiro. O de teste, por outro lado, tem uma probabilidade bem alta de dar um falso positivo, o que é ruim para as pacientes, embora tenha uma probabilidade 0 de omitir um verdadeiro positivo. Ainda analisando as probabilidades de alarme falso, perceba que, no modelo de treino, dado que um resultado indicado seja falso, é mais provável que seja um falso negativo do que um falso positivo, o que é perigoso para as pacientes, sendo apenas 47.619% dos resultados falsos, falsos positivos ($(FP)/(FP+FN)$),

Como bônus, também fiz pelo caminho inverso, treinando o modelo com o arquivo de teste e testando no de treino:

```
--> alfa_te = Gaussian_Elimination_4_AP3(X_te'*X_te,X_te'*Y_te)
 alfa_te  =

  -5.0229554
   27.931616
   0.9214507
  -23.555718
  -2.7523961
   0.5863272
```

```
  -1.0575316
   2.3988973
   1.4747597
   0.8646323
   0.7477812

--> prev_te = X_te * alfa_te;

--> conf_prev_te = prev_te .* Y_te;

--> acertos_te = sum(conf_prev_te>0|conf_prev_te==0)
 acertos_te  =

   250.

--> erros_te = sum(conf_prev_te<0|conf_prev_te==0)
 erros_te  =

   10.

--> prev_tr = X_tr * alfa_te;

--> conf_prev_tr = prev_tr .* Y_tr;

--> acertos_tr = sum(conf_prev_tr>0|conf_prev_tr==0)
 acertos_tr  =

   199.

--> erros_tr = sum(conf_prev_tr<0|conf_prev_tr==0)
 erros_tr  =

   101.

--> h_te = X_te * alfa_te;

--> h_te_class = h_te>=0;

--> h_te_ones = ones(h_te_class);

--> h_te_class = 2*(h_te_class)-h_te_ones;

--> positivos_te = sum(h_te_class>0)
 positivos_te  =

   52.

--> negativos_te = sum(h_te_class<0)
 negativos_te  =

   208.

--> SomaPosTe = sum(Y_te==1)
 SomaPosTe  =

   60.

```

```
--> SomaNegTe = sum(Y_te==-1)
 SomaNegTe  =

   200.

--> h_tr = X_tr * alfa_te;

--> h_tr_class = h_tr>=0;

--> h_tr_ones = ones(h_tr_class);

--> h_tr_class = 2*(h_tr_class)-h_tr_ones;

--> positivos_tr = sum(h_tr_class>0)
 positivos_tr  =

   49.

--> negativos_tr = sum(h_tr_class<0)
 negativos_tr  =

   251.

--> SomaPosTr = sum(Y_tr==1)
 SomaPosTr  =

   146.

--> SomaNegTr = sum(Y_tr==-1)
 SomaNegTr  =

   154.
```

Fazendo as matrizes de confusão:

Matriz de Teste:

$$\begin{cases} TP + FP = 52 \\ TP + FN = 60 \\ TN + FP = 200 \\ TN + FN = 208 \end{cases}$$

$$
\begin{cases}
TP + FP = 52 \\
TP + FN = 60 \\
FN - FP = 8 \Rightarrow FN = FP + 8 \\
\\
TP + FP = 52 \\
TN + FP = 200 \\
TN - TP = 148 \Rightarrow TN = TP + 148 \\
\\
TP + TN = 250 \Rightarrow TP + TP + 148 = 250 \Rightarrow 2TP = 102 \Rightarrow TP = 51 \\
TP + FP = 52 \Rightarrow FP = 1 \\
TN = TP + 148 \Rightarrow TN = 199 \\
FN = FP + 8 \Rightarrow FN = 9
\end{cases}
$$

TP	FP
FN	TN

$\Downarrow$

51	1
9	199

Matriz de Treino:

$$
\begin{cases}
TP + FP = 49 \\
TP + FN = 146 \\
TN + FP = 154 \\
TN + FN = 251
\end{cases}
$$

$$
\begin{cases}
TP + FP = 49 \\
TP + FN = 146 \\
FN - FP = 97 \Rightarrow FN = FP + 97 \\
\\
TP + FP = 49 \\
TN + FP = 154 \\
TN - TP = 105 \Rightarrow TN = TP + 105 \\
\\
TP + TN = 199 \Rightarrow TP + TP + 105 = 199 \Rightarrow 2TP = 94 \Rightarrow TP = 47 \\
TP + FP = 49 \Rightarrow FP = 2 \\
TN = TP + 105 \Rightarrow TN = 152 \\
FN = FP + 97 \Rightarrow FN = 99
\end{cases}
$$

TP	FP
FN	TN

$\Downarrow$

47	2
99	152

Fazendo a análise das matrizes:

Matriz de Teste:

- Acurácia: $(51 + 199)/(51 + 199 + 1 + 9) = 250/260 \approx 0.96154 = 96.154\%$
- Erro de Classificação: $(1 + 9)/(51 + 199 + 1 + 9) = 10/260 \approx 0.03846 = 3.846\%$
- Precisão: $(51)/(51 + 1) = 51/52 \approx 0.98077 = 98.077\%$
- Recall/Sensitividade: $(51)/(51 + 9) = 51/60 = 0.85 = 85\%$
- Especificidade: $(199)/(199 + 1) = 199/200 = 0.955 = 95.5\%$
- Probabilidade de Falso Alarme: $(1)/(51 + 1) = 1/52 \approx 0.01923 = 1.923\%$
- Probabilidade de Falsa Omissão de Alarme: $(9)/(199 + 9) = 9/208 \approx 0.04327 = 4.327\%$

Matriz de Treino:

- Acurácia: $(47 + 152)/(47 + 152 + 2 + 99) = 199/300 \approx 0.66333 = 66.333\%$

- Erro de Classificação: $(2+99)/(47+152+2+99) = 101/300 \approx 0.33666 = 33.666\%$
- Precisão: $(47)/(47+2) = 47/49 \approx 0.95918 = 95.918\%$
- Recall/Sensitividade: $(47)/(47+99) = 47/146 \approx 0.32192 = 32.192\%$
- Especificidade: $(152)/(152+2) = 152/154 \approx 0.98701 = 98.701\%$
- Probabilidade de Falso Alarme: $(2)/(47+2) = 2/49 \approx 0.04081 = 4.081\%$
- Probabilidade de Falsa Omissão de Alarme: $(99)/(152+99) = 99/251 \approx 0.39442 = 39.442\%$

Usando agora o arquivo de teste como treino e vice-versa, é possível notar que, agora, a matriz de teste é extremamente acurada e precisa, com uma probabilidade muito baixa de ter erros de classificação. Ela também é bem específica, com quase 100% de especificidade, visto que dos 200 valores negativos verdadeiros, 199 foram preditos corretamente. Seu recall, ou seja, positivos verdadeiros que conseguimos medir corretamente com nosso modelo também teve um valor bem alto. As probabilidades de alarme falso são notavelmente baixas, embora seja perceptível que, dado que um resultado seja falso, apenas 10% são falsos positivos, o que é arriscado para as clientes, já que 9 a cada 10 resultados falsos são falsos negativos, ou seja, têm câncer de mama e o exame diz o contrário.

A matriz de treino por outro lado, não é tão boa assim. Tem uma acurácia mediana, abaixo de 70%, embora tenha uma precisão bem alta. Seu recall também é baixo, já que dos 300 valores, apenas 16.333% são preditos como positivos, e apenas 32.192% dos valores realmente positivos são preditos como tal. Assim como a matriz de teste, essa também é bem específica, e tem uma baixa probabilidade de alarme falso, embora tenha uma probabilidade de falsa omissão de alarme de quase 50%, já que 33% dos resultados encontrados são falsos negativos.

Tendo usado ambas as matrizes como treino e teste de maneira alternada, é possível perceber que, geralmente, os dados de treinamento geram uma matriz de confusão com propriedades bem adequadas, ou seja, bem precisas, acuradas e específicas, com baixas probabilidades de omitir um alarme falso. As de teste, por outro lado, como tratam-se de valores usados para previsão do modelo, podem não ser tão compatíveis com ele, como pode ser visto em ambos os casos apresentados, tendo números altos de resultados falsos e não tão acurados e precisos quanto os dados de treino criados exatamente para isso.

6 Fontes de consulta

- Aulas Gravadas
- Álgebra Linear, David Poole (2^a edição)
- Geogebra (para construção dos gráficos das aproximações lineares)
- Censo estadunidense de 2010 (http://g1.globo.com/mundo/noticia/2010/12/eua-populacao-cresceu-97-entre-2000-e-2010-a-taxa-mais-baixa-desde-1930-1.html)
- Aproximação por Mínimos Quadrados (https://www1.univap.br/spilling/CN/ExRes_MMQ.pdf)
- Simple guide to confusion matrix terminology (https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/)
- Taking the Confusion Out of Confusion Matrices (https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e)
- Confusion matrix (https://en.wikipedia.org/wiki/Confusion_matrix)
- Confusion Matrix for Your Multi-Class Machine Learning Model

  (https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826)
- Everything you Should Know about Confusion Matrix for Machine Learning (https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/)
- Dados de Treino e Teste (https://didatica.tech/dados-de-treino-e-teste/)