1. Find the equation of the straight line that best fits the data

\[
\begin{array}{c|cccc}
  x & 1 & 2 & 4 & 5 \\
  y & 6 & -3 & 11 & 14 \\
\end{array}
\]

2. Find the quadratic

\[ y = ax^2 + bx + c \]

that best fits the data

\[
\begin{array}{c|cccc}
  x & 1 & 2 & 3 & 4 \\
  y & 1 & 6 & 3 & 12 \\
\end{array}
\]

3. The CPU times required to compute the LU factorisation of various \( n \times n \) matrices on my laptop were

<table>
<thead>
<tr>
<th>( n )</th>
<th>CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.1</td>
</tr>
<tr>
<td>707</td>
<td>0.2</td>
</tr>
<tr>
<td>1000</td>
<td>0.6</td>
</tr>
<tr>
<td>1414</td>
<td>1.6</td>
</tr>
<tr>
<td>2000</td>
<td>4.6</td>
</tr>
<tr>
<td>2828</td>
<td>12.6</td>
</tr>
<tr>
<td>4000</td>
<td>35.1</td>
</tr>
</tbody>
</table>

We expect

\[ \text{CPU time} = cn^\alpha. \]

Find \( c \) and \( \alpha \) using least squares techniques and the QR factorisation. (Hint: use logarithms to turn this into a linear equation.) Use your \( c \) and \( \alpha \) to estimate the CPU time required for \( n = 8000 \).

4. Consider the approximation

\[ y = ax + b \]

for the data

\[ (x_i, y_i) \quad i = 1, \ldots, n. \]
(a) We define the least squares error

\[ \phi(a, b) = \sum_{i=1}^{n} [y_i - (ax_i + b)]^2. \]

Use calculus techniques to show that \( \phi(a, b) \) is minimised when \( a \) and \( b \) satisfy

\[ a \sum_{i=1}^{n} x_i^2 + b \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i, \quad a \sum_{i=1}^{n} x_i + bn = \sum_{i=1}^{n} y_i. \]

(b) Show that these equations are equivalent to the normal equations.