37233 Linear Algebra
Tutorial Assignment 10 -Solutions

Question 1 (least squares - overdetermined system)

In[1]:=
```
A = {{3, 1}, {1, 1}, {1, 2}};
```

In[2]:=
```
A // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 3 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$$

In[3]:=
```
b = {1, 1, 1};
```

In[4]:=
```
b // MatrixForm
```

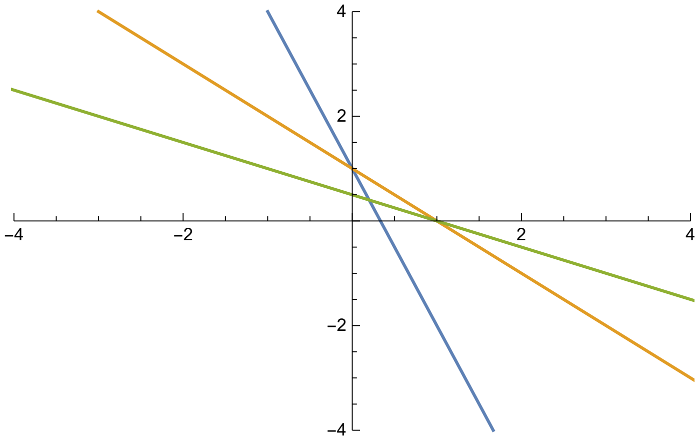
Out[4]//MatrixForm=

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Just for interest, plot the lines defined by the inconsistent equations

In[5]:=
```
p1 = Plot[{1/A[[1, 2]] * (-A[[1, 1]] * x + b[[1]]),
   1/A[[2, 2]] * (-A[[2, 1]] * x + b[[2]]), 1/A[[3, 2]] * (-A[[3, 1]] * x + b[[3]]) },
  {x, -10, 10}, PlotRange → {{-4, 4}, {-4, 4}}]
```

Out[5]=



Find the least squares solution

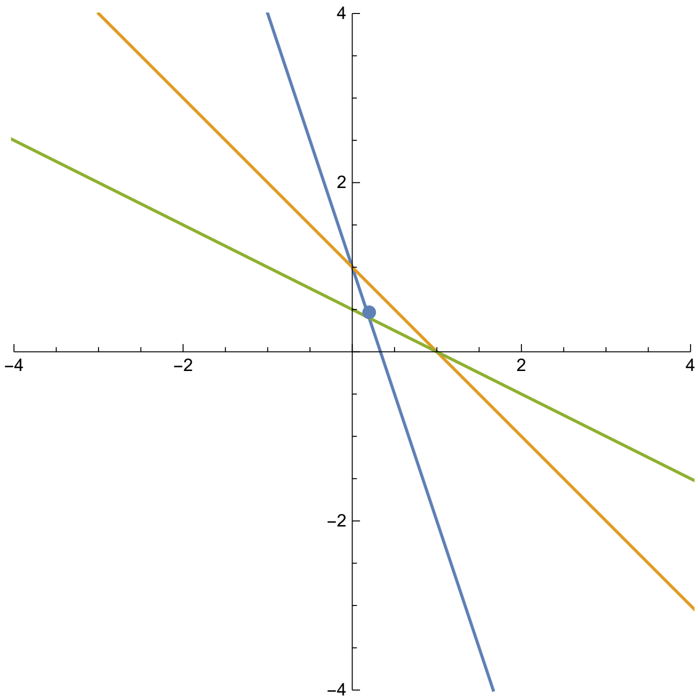
In[6]:=
```
sol = LinearSolve[Transpose[A].A, Transpose[A].b]
```

Out[6]= $\left\{\frac{1}{5}, \frac{7}{15}\right\}$

Superimpose the point representing the solution on the previous diagram

In[7]:=
```
Show[{p1, ListPlot[{sol}, {AspectRatio → 1, PlotStyle -> PointSize[0.02]}]},
  AspectRatio → 1]
```

Out[7]=



Question 2 (least squares - data fitting)

In[8]:=
```
data = {{2, 3}, {3, 2}, {5, 1}, {6, 0}}
```

Out[8]= {{2, 3}, {3, 2}, {5, 1}, {6, 0}}

In[9]:=
```
data // TableForm
```

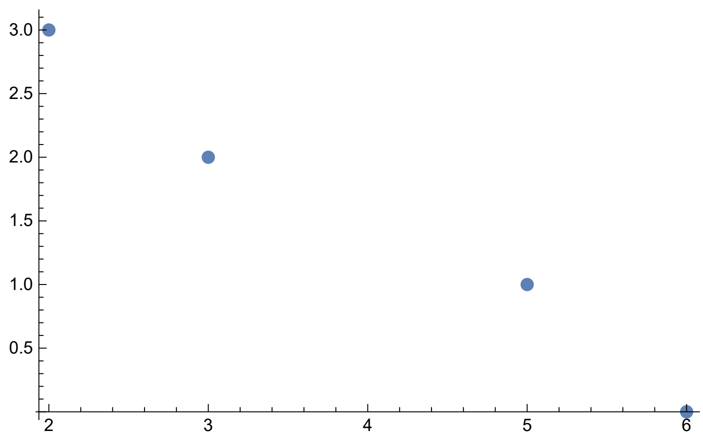
Out[9]//TableForm=

2	3
3	2
5	1
6	0

Plot the data points

In[10]:=
```
points = ListPlot[data, {Joined → False, PlotStyle -> PointSize[0.02]}]
```

Out[10]=



Set up the design matrix

```
In[11]:= xm = Transpose[{{1, 1, 1, 1}, Transpose[data][[1]]}]
Out[11]= {{1, 2}, {1, 3}, {1, 5}, {1, 6}}

In[12]:= xm // MatrixForm
Out[12]//MatrixForm=
```

$$\begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 5 \\ 1 & 6 \end{pmatrix}$$

Set up the observed dependent values

```
In[13]:= ym = Transpose[data][[2]]
Out[13]= {3, 2, 1, 0}
```

Solve for the model parameters

```
In[14]:= sol = LinearSolve[Transpose[xm].xm, Transpose[xm].ym]
```

Out[14]= $\left\{\frac{43}{10}, -\frac{7}{10}\right\}$

Plot the model equation

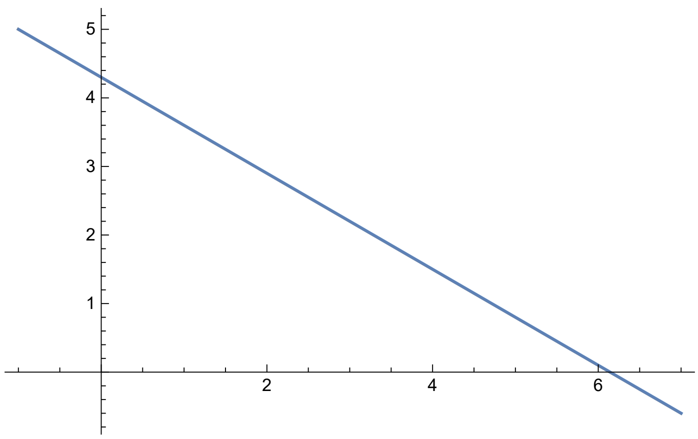
```
In[15]:= Clear[lbf];
         lbf[x_] := sol[[1]] + sol[[2]] * x;

In[17]:= lbf[2]
```

Out[17]= $\frac{29}{10}$

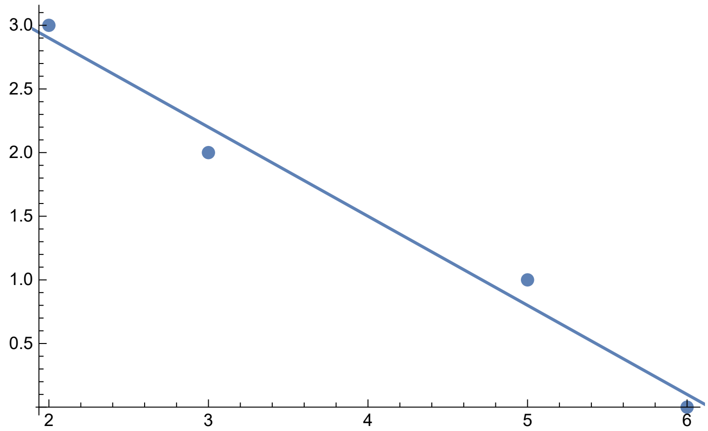
```
In[18]:= plbf = Plot[lbf[x], {x, -1, 7}]
```

Out[18]=



Superimpose the data on the model plot

In[19]:=
```
Show[{points, plbf}]
```

Out[19]=


Question 3 (least squares - data fitting)

In[20]:=
```
data = {{-0.31, 3.15}, {0.71, 9.12}, {2.11, 12.11}, {2.65, 14.01}}
```

Out[20]=
```
{{-0.31, 3.15}, {0.71, 9.12}, {2.11, 12.11}, {2.65, 14.01}}
```

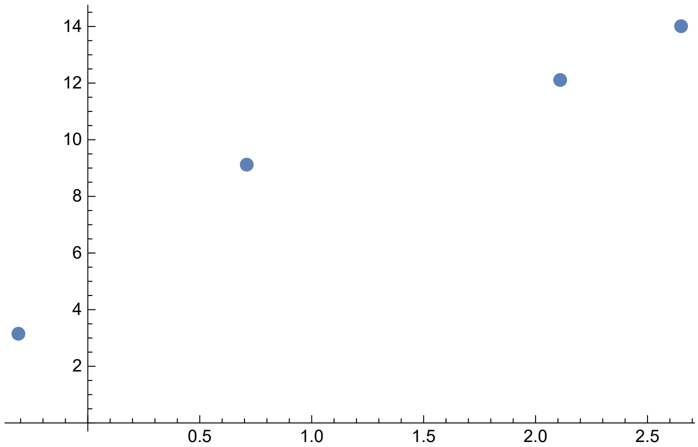
In[21]:=
```
data // TableForm
```

Out[21]//TableForm=

-0.31	3.15
0.71	9.12
2.11	12.11
2.65	14.01

Plot the data points

In[22]:=
```
points = ListPlot[data, {Joined → False, PlotStyle -> PointSize[0.02]}]
```

Out[22]=


Set up the design matrix

In[23]:=
```
xm = Transpose[{{1, 1, 1, 1}, Transpose[data][[1]]}]
```

Out[23]=
```
{{1, -0.31}, {1, 0.71}, {1, 2.11}, {1, 2.65}}
```

In[24]:=
```
xm // MatrixForm
```

Out[24]//MatrixForm=

$$\begin{pmatrix} 1 & -0.31 \\ 1 & 0.71 \\ 1 & 2.11 \\ 1 & 2.65 \end{pmatrix}$$

Set up the observed dependent values

In[25]:=
```
ym = Transpose[data][[2]]
```

Out[25]= {3.15, 9.12, 12.11, 14.01}

Solve for the model parameters

In[26]:=
```
sol = LinearSolve[Transpose[xm].xm, Transpose[xm].ym]
```

Out[26]= {5.15635, 3.44275}

Plot the model equation

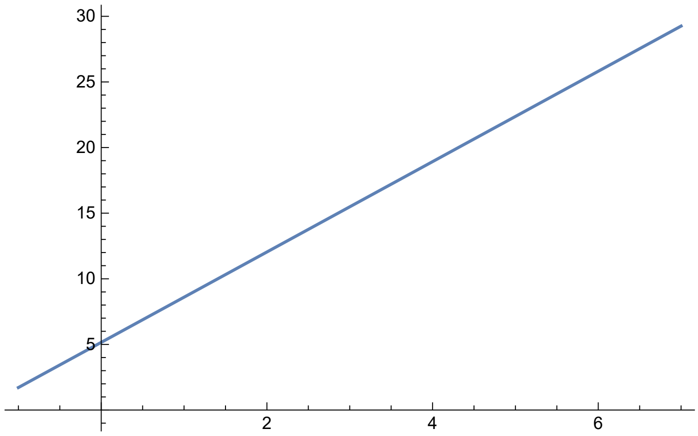
In[27]:=
```
Clear[lbf];
lbf[x_] := sol[[1]] + sol[[2]] * x;
```

In[29]:=
```
lbf[2]
```

Out[29]= 12.0419

In[30]:=
```
plbf = Plot[lbf[x], {x, -1, 7}]
```

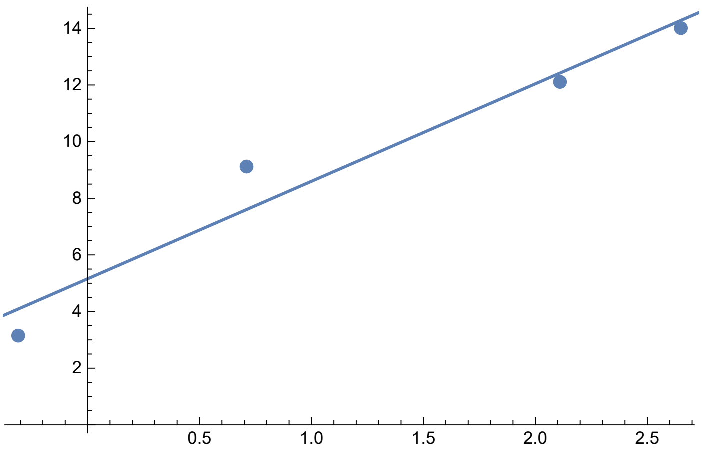
Out[30]=



Superimpose the data on the model plot

In[31]:=
```
Show[{points, plbf}]
```

Out[31]=



Question 4

Part (a)

In[32]:=
```
A = {{1, 2}, {4, 3}}
```
Out[32]=
```
{{1, 2}, {4, 3}}
```

Mathematica has some useful built-in commands for finding eigenvalues (the Eigenvalues command) and eigenvectors (the Eigenvectors command), or both (the Eigensystem command):

In[33]:=
```
e = Eigensystem[A]
```
Out[33]=
```
{{5, -1}, {{1, 2}, {-1, 1}}}
```

The first element (a list of two numbers) is the list of eigenvalues that can be used to construct a diagonal matrix. The second element (a list of two vectors) is a list of eigenvectors, each associated with the corresponding eigenvalue in the first list. If these are linearly indpendent they can be used to costruct an invertible matrix P (the change of basis matrix from the standard basis to an eigenvector basis)

In[34]:=
```
d = DiagonalMatrix[e[[1]]]
```
Out[34]=
```
{{5, 0}, {0, -1}}
```

In[35]:=
```
p = Transpose[e[[2]]]
```
Out[35]=
```
{{1, -1}, {2, 1}}
```

In[37]:=
```
A == p.d.Inverse[p]
```
Out[37]=
```
True
```

To do the same calculations by hand is a bit more tedious, but straightforward. The outline is below:

In[38]:=
```
eigenvals = Solve[Det[A - lam IdentityMatrix[2]] == 0, lam]
```
Out[38]=
```
{{lam → -1}, {lam → 5}}
```

In[40]:=
```
{lam1, lam2} = Table[eigenvals[[i, 1, 2]], {i, 1, Length[eigenvals]}]
```
Out[40]=
```
{-1, 5}
```

In[41]:=
```
d = DiagonalMatrix[{lam1, lam2}]
```
Out[41]=
```
{{-1, 0}, {0, 5}}
```

In[43]:=
```
A - lam1 IdentityMatrix[2]
```
Out[43]=
```
{{2, 2}, {4, 4}}
```

In[50]:=
```
NullSpace[A - lam1 IdentityMatrix[2]]
```
Out[50]=
```
{{-1, 1}}
```

In[51]:=
```
p1 = NullSpace[A - lam1 IdentityMatrix[2]][[1]]
```
Out[51]=
```
{-1, 1}
```

In[52]:=
```
NullSpace[A - lam2 IdentityMatrix[2]]
```
Out[52]=
```
{{1, 2}}
```

In[53]:=
```
p2 = NullSpace[A - lam2 IdentityMatrix[2]][[1]]
```
Out[53]=
```
{1, 2}
```

In[54]:=
```
p = Transpose[{p1, p2}]
```
Out[54]=
```
{{-1, 1}, {1, 2}}
```

In[55]:=
```
A == p.d.Inverse[p]
```
Out[55]=
```
True
```

In[56]:=
```
p // MatrixForm
```
Out[56]//MatrixForm=

$$\begin{pmatrix} -1 & 1 \\ 1 & 2 \end{pmatrix}$$

In[57]:=
```
d // MatrixForm
```
Out[57]//MatrixForm=

$$\begin{pmatrix} -1 & 0 \\ 0 & 5 \end{pmatrix}$$

Question 5

In[58]:=
```
a = {{1, 3}, {3, 1}}
```
Out[58]=
```
{{1, 3}, {3, 1}}
```

Mathematica has some useful built-in commands for finding eigenvalues (the Eigenvalues command) and eigenvectors (the Eigenvectors command), or both (the Eigensystem command):

In[59]:=
```
e = Eigensystem[a]
```
Out[59]=
```
{{4, -2}, {{1, 1}, {-1, 1}}}
```

The first element (a list of two numbers) is the list of eigenvalues that can be used to construct a diagonal matrix. The second element (a list of two vectors) is a list of eigenvectors, each associated with the corresponding eigenvalue in the first list. If these are linearly indpendent they can be used to costruct an invertible matrix P (the change of basis matrix from the standard basis to an eigenvector basis)

In[61]:=
```
d = DiagonalMatrix[e[[1]]]
```
Out[61]=
```
{{4, 0}, {0, -2}}
```

In[62]:=
```
p = Transpose[e[[2]]]
```
Out[62]=
```
{{1, -1}, {1, 1}}
```

In[63]:=
```
a == p.d.Inverse[p]
```
Out[63]=
```
True
```

To do the same calculations by hand is a bit more tedious, but straightforward. The outline is below:

In[64]:=
```
eigenvals = Solve[Det[a - lam IdentityMatrix[2]] == 0, lam]
```
Out[64]=
```
{{lam → -2}, {lam → 4}}
```
In[65]:=
```
{lam1, lam2} = Table[eigenvals[[i, 1, 2]], {i, 1, Length[eigenvals]}]
```
Out[65]=
```
{-2, 4}
```
In[66]:=
```
d = DiagonalMatrix[{lam1, lam2}]
```
Out[66]=
```
{{-2, 0}, {0, 4}}
```
In[67]:=
```
a - lam1 IdentityMatrix[2]
```
Out[67]=
```
{{3, 3}, {3, 3}}
```
In[68]:=
```
NullSpace[a - lam1 IdentityMatrix[2]][[1]]
```
Out[68]=
```
{-1, 1}
```
In[69]:=
```
p1 = NullSpace[a - lam1 IdentityMatrix[2]][[1]]
```
Out[69]=
```
{-1, 1}
```
In[70]:=
```
NullSpace[a - lam2 IdentityMatrix[2]]
```
Out[70]=
```
{{1, 1}}
```
In[71]:=
```
p2 = NullSpace[a - lam2 IdentityMatrix[2]][[1]]
```
Out[71]=
```
{1, 1}
```
In[72]:=
```
p = Transpose[{p1, p2}]
```
Out[72]=
```
{{-1, 1}, {1, 1}}
```
In[73]:=
```
a == p.d.Inverse[p]
```
Out[73]=
```
True
```

To construct the spectral decomposition: first represent p1 as a 2x1 matrix and make it unit norm:

In[74]:=
```
MatrixForm[p1m = (Transpose[{p1}]) / Norm[p1]]
```
Out[74]//MatrixForm=

$$\begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Now form $P_1 \, P_1{}^T$

In[75]:=
```
MatrixForm[p1mat = p1m.Transpose[p1m]]
```
Out[75]//MatrixForm=

$$\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Do the same for P_2:

In[76]:=
```
MatrixForm[p2m = (Transpose[{p2}]) / Norm[p2]]
```
Out[76]//MatrixForm=

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

In[77]:=
```
MatrixForm[p2mat = p2m.Transpose[p2m]]
```

Out[77]//MatrixForm=

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Finally, check that the decomposition is correct:

In[78]:=
```
MatrixForm[decomposition = lam1 * p1mat + lam2 * p2mat]
```

Out[78]//MatrixForm=

$$\begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

In[79]:=
```
a == decomposition
```

Out[79]= True

In[96]:=
```
P = {Flatten[p1m ], Flatten[p2m ]} // MatrixForm
```

Out[96]//MatrixForm=

$$\begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

In[97]:=
```
d // MatrixForm
```

Out[97]//MatrixForm=

$$\begin{pmatrix} -2 & 0 \\ 0 & 4 \end{pmatrix}$$