linear algebra & image processing matlab case 5
Least Squares

Before taking up the LS case that comes wth the book by D. Lay
we look at possible house pricing formulas first

House pricing formulas:
Sometimes the work of estate agents is said to be as simple as
"take the amount of cubic meters of air walled in plus the area in square meters
of the plot of land the house is standing on and multiply this by a unit price
and this will give you the price the lot is worth"
We can check such a proposed formula (or pricing model) by comparing model prices
with prices asked and realised in a certain region by using data from
housing agancies and the legal administration (kadaster) of house purchases.

First lets see how far price formula
price_asked = x1*volume+x2*area
is away from those asked in Oegstgeest:
The estate agents near Leiden usually mention square metres of living space
instead of the volume of the house, but this doesn't make much difference since
the volume of the house and its area of living space is related by the average
height of a floor (say 2.25-2.5 metres).
On the Internet housing site www.funda.nl we find that in Oegstgeeest
for all houses exept appartments living space and plot area are both given
in square metres; for houses between 350 and 400 KEuros:

```
house1: price1=375 KEuro livingspace1=   135 m2 plotarea=156 m2
house2:        399                  190             82
house3:        349                  145             85
house4:        390                  190             77
house5:        399                  150            141
house6:        349                  160            132
```

this list can be extended at will.

The formula pa = x1*la + x2*pa applied to this would lead to an overdetermined
set of 6 equations with 2 unknowns:

```
A = [135   156;
     190    82;
     145    85;
     190    77;
     150   141;
     160   132]
```

and

```
p=[ 375;399;349;390;399;349]
```

Using our original reduction approach by using augmented matrix [A:p]
doesn't give us solutions:

```
Ap=[A p]
rref(Ap)
```

The Least Squares approach however does:
first we create the transformed version of A:

```
AT=A'
```

and use AT to create ATA and ATp:

```
ATA=AT*A
ATp=AT*p
```

we subsequently solve x using augmented matrix [ATA|ATp]:

```
ATAp=[ATA ATp]
rref(ATAp)
```

The last column of solution [I|x] gives one the weights,
x1 to be multiplied by la (living space)
and x2 to be multiplied by pa (plot area),

of how heavily la and pa count in setting the house price
according to this model.

The model prices Pest are now given by multiplying A with x:
Pest=A*x

Question: what according to this model has more weight?
Living space or area of the plot of land the house is standing on?

Given these estimated model prices one can calculate the difference
between realised prices (kadaster lookup) and prices asked;
this difference is called the least squares error of the particular model.

Which parameters could be added to the first pricing model to reduce
the least squares error?

The great advantage of matrix ATA is the fact that it is always a square matrix
with precisely the same amount of dimensions as there are parameters in the model.
The original matrix A can be lengthened considerably to get better statistics.

By comparing several pricing models with real data one can make a well-funded
choice for a simpler/more elaborate model.

The next Least Squares cases are from the Lay LS case:
First the one with the Olympic 400 metres record times.
Read the text in the LS case statement.

Make script line400.m that constructs the design matrix X for a line
approximation of record times as a function of olympic year and
observation vector y for the realised (record)times.

In matlab the transposed of a matrix A is A',
Construct XTX=XT*X and XTy=XT*y and form augmented matrix XTXy = [XTX|XTy]

Try 3 ways to determine a LS solution for beta:
1) reduction of XTXy
2) using the inverse
3) using the matlab backslash operator for a LS solution:beta=X\y

Compare these results.

Make a function rt=rt400(year,beta) that given a year (minus 1972)
and a parameter vector produces a LS result value for that year.

Make a plotfunction (using plotax from the computer graphics case
and remember to use "hold" to keep the graphic result) for the data
and the LS line approximation for the period 1972-2008.
Maak nu een plotfunctie voor de data en de LS lijnbenadering voor
de periode 1972 t/m 2008
(gebruik plotax uit plotfunctie CGcase voor afbakenen afbeeldingsruimte en
gebruik hold voor vasthouden grafisch resultaat)

Next, apply a quadratic model to fit the 400 metres recordtimes.
Construct function kwa400.m to construct design matrix K
(use a loop to fill in the 1, x and x*x values for each row).
Also form y,KTK,KTy and KTKaug.

Determine the LS solution using the matlab backslash operator: K\y
Check whether the result is like rref(KTKaug).

Plot using an adapted plotfunction the graphical representation of data, line and quadratic curve approximation between 1972 and 2008.

You can continue by following the same approach to the yearly max and min temperature fluctuations (see matlab LS case description).