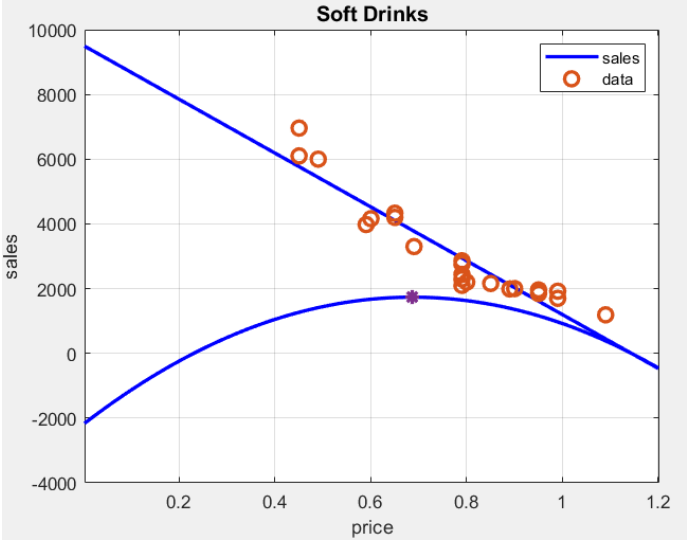
1(d)

```
a1 = ones(22,1);
pc = [0.59;0.8;0.95;0.45;0.79;0.99;0.9;0.65;0.79;0.69;0.79;...
0.49;1.09;0.95;0.79;0.65;0.45;0.6;0.89;0.79;0.99;0.85];
sc = [3980;2200;1850;6100;2100;1700;2000;4200;2440;3300;2300;...
6000;1190;1960;2760;4330;6960;4160;1990;2860;1920;2160];

a = [a1,pc]; A = a' * a; b = a' * sc; c = A^(-1) * b;

r = sc - a*c;
RMSE = 1/sqrt(22).*norm(r,2); fprintf('RMSE = %f',RMSE);

x = 0:.01:1.2; sce = c(1)+c(2).*x;
p = @(x) (c(1) + c(2).*x)*(x-0.23);
px = @(x) c(1) + 2*c(2).*x - c(2)*0.23;

max_price = fzero(px,0);
max_profit = p(max_price);

x = 0:.01:1.2;
p = (c(1) + c(2).*x).*(x - 0.23);

plot(x,sce,'b-',pc,sc,'o',x,p,'b-',max_price,max_profit,'*','LineWidth',2,...
    'MarkerSize',8);
xlabel('price'); ylabel('sales');
title('Soft Drinks','FontSize',12);
legend('sales','data');
grid on;
```



RMSE = 518, optimal price = 0.69, profit = 1736

3(d)

Here is the function code:

```
function [Q,R] = mgs( A )
    [m, n] = size(A);
    Q = zeros(m,n);
    R = zeros(n,n);
    I = 1:m;
    for i = 1:n
        y = A(I, i);
        for j = 1:i-1
            R(j, i) = dot(Q(I, j), y);
            y = y - R(j, i) * Q(I, j);
        end
    R(i, i)= norm(y);
    Q(I, i)= y / R(i, i);
    end
end
```

There are two cases: both cases will have a result close enough to 0.

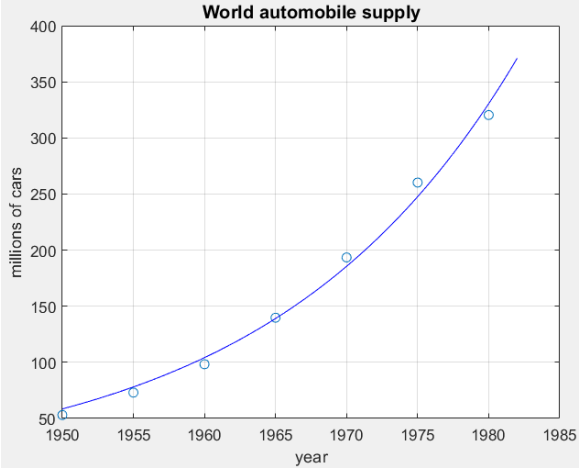
```
A =

     1     2
     2     2


Q =

    0.4472    0.8944
    0.8944   -0.4472


R =

    2.2361    2.6833
         0    0.8944

norm(A-Q*R)=0.000e+00
norm(Q^T Q-I)=1.634e-16
```

```
A =

     4     8     1
     0     2    -2
     3     6     7


Q =

    0.8000         0   -0.6000
         0    1.0000         0
    0.6000         0    0.8000


R =

     5    10     5
     0     2    -2
     0     0     5

norm(A-Q*R)=0.000e+00
norm(Q^T Q-I)=2.665e-17
```

4.

```
function xc = gaussNewton( r,rx, x0,tol,maxIterations )
    n = length(x0);
    xc = x0;
    rc = r(xc);
    k = 1;
    while k < maxIterations
        [Q, R] = qr(rx(xc),0);
        v = -R \ (Q'*rc);
        xc = xc + v;
        rc = r(xc);

        fprintf('GaussNewton: k=%4d: x=[%13.6e',k,xc(1));
        for( j=2:n ) fprintf(',%13.6e',xc(j)); end
        fprintf('] || correction ||_2=%8.2e\n',norm(v));

        if( norm(v) < tol )
            break;
        end
    end
return
```

```
(a)
GaussNewton: k=   1: x=[ 4.745744e+01, 7.900008e-02] || correction ||_2=2.54e+00
GaussNewton: k=   2: x=[ 5.586202e+01, 6.119939e-02] || correction ||_2=8.40e+00
GaussNewton: k=   3: x=[ 5.860621e+01, 5.764194e-02] || correction ||_2=2.74e+00
GaussNewton: k=   4: x=[ 5.850297e+01, 5.771937e-02] || correction ||_2=1.03e-01
GaussNewton: k=   5: x=[ 5.850774e+01, 5.771607e-02] || correction ||_2=4.76e-03
GaussNewton: k=   6: x=[ 5.850754e+01, 5.771621e-02] || correction ||_2=2.01e-04
GaussNewton: k=   7: x=[ 5.850754e+01, 5.771620e-02] || correction ||_2=8.62e-06
RMSE = 7.676587
```



(b)

```
GaussNewton: k=   1: x=[ 8.792056e+00,-2.258896e-01] || correction ||_2=3.79e+00
GaussNewton: k=   2: x=[ 9.782482e+00,-2.131377e-01] || correction ||_2=9.91e-01
GaussNewton: k=   3: x=[ 9.797125e+00,-2.150833e-01] || correction ||_2=1.48e-02
GaussNewton: k=   4: x=[ 9.796929e+00,-2.150872e-01] || correction ||_2=1.96e-04
GaussNewton: k=   5: x=[ 9.796928e+00,-2.150872e-01] || correction ||_2=1.10e-06
RMSE = 0.263228
```