```
========= maxautovalore.m =====================================
==========
function [a,v,r]=maxautovalore(A,it)
% Applica it iterazioni del metodo delle potenze
%  alla matrice A
% Restituisce l'ultimo quoziente di Railegh
%  (approx dell'autovalore max) e la relativa
%  approssimazione dell'autovettore
% r = successione quozienti Railegh

if nargin<2
    it=20;
end

if nargout>2
  r=zeros(1,it);
end

N=size(A);
if (N(1)==N(2))
    N=N(1);
else
    error
end
%Posso iniziare da un vettore qualsiasi, ad esempio "ones", oppure "rand", ...
y=ones(N,1);%rand(N,1);

for k=1:it
    y = y / sqrt(y'*y);% sqrt(y'*y) = norm(y,2);
    v = A*y;     % "v" è variabile in output
    a = (v'*y) ; % quoziente di Railegh ("a" è variabile in output)
    if nargout>2
      r(k) = a;  % solo se ho richiesto la "storia di convergenza" delle a
    end
    y = v;
end
===============================================================
=================
>> A=ferrovielombarde();
>> [a,v]=maxautovalore(A,10);
```

```
>> a
a =
   3.4108
>> [a2,v2]=maxautovalore(A,20);
>> a2
a2 =
   3.4160
>> a2-maxautovalore(A,21)
ans =
  -1.4664e-05
%Alla 20-esima iterazione, 4 cifre decimali sono corrette e solo la
% quinta sta ancora cambiando

================ classifica.m =========
[A,S]=ferrovielombarde();
N=size(A,1);

n_link = sum(A,1);
%vettore riga con la somma lungo le colonne

% for k=1:N
%     B(:,k)=A(:,k)/n_link(k);
% end
B = A ./ (ones(N,1)*n_link);

[a,v]=maxautovalore(B,100);
[s,ind]=sort(v);%v ordinato e ordine degli indici
for k=N:-1:1
   disp(sprintf('%15s %f',S{ind(k)},s(k)/s(end)));
end
==================================================
>> classifica
         Milano 1.000000
        Bergamo 0.571429
          Lecco 0.571429
        Brescia 0.428572
           Como 0.428572
           Lodi 0.285715
        Cremona 0.285714
           Asso 0.142857
```

```
      Varese 0.142857
       Pavia 0.142857
     Mantova 0.142857
     Sondrio 0.142857

% Vogliamo studiare la convergenza del metodo delle potenze
>> [a,v,r]=maxautovalore(B,100);
>> a-1
ans =
  -4.8171e-07
% Stiamo convergendo a 1
>> plot(r,'.-')
>> plot(abs(r-1),'.-')
% Infatti la distanza da 1 va a zero
% Guardiamola nelle diverse scale degli assi:
>> loglog(abs(r-1),'.-')
>> semilogx(abs(r-1),'.-')
>> semilogy(abs(r-1),'.-')
% Nell'ultimo grafico l'errore si comporta "come una retta"
% Se chiamo M e Q i coefficienti della "retta", deduco che
% log(ERR(k)) ~ Mk+Q
% ERR(k) ~ exp(Mk+Q) = exp(Q) * (exp(M))^k = C * rho^k
% dove C=exp(Q) e rho=exp(M)
% Ad ogni iterazione (incremento k di 1), l'errore viene
% moltiplicato per "rho"
% Dal grafico vediamo che M<0, quindi rho<1 e ERR(K)-->0
% (Con la teoria dell'approssimazione, impareremo a stimare
% M e Q dal grafico e quindi dedurre "rho")
```