```
[muK,qf] = countvmf2(mu,C/m,tolh);  % number of red margin failures
fprintf('pf =  %d',pf)
fprintf('    qf =  %d \n',qf)
[~,pm] = countmlu2(lamb,tols);  % number of  points such that lambda_i > tols
[~,qm] = countmlv2(mu,tols);   % number of  points such that mu_i > 0
fprintf('pm =  %d',pm)
fprintf('    qm =  %d \n',qm)
%  lambda_i <= tols
[lmz,nz] = countLzero(lamb,mu,tols);
pm2 = numsvl1 + pf; qm2 = numsvm1 + qf;
fprintf('pm2 =  %d',pm2)
fprintf('   qm2 =  %d \n',qm2)
lnu = max(2*pf/m,2*qf/m); unu = min(2*pm/m,2*qm/m);
fprintf('lnu =  %d',lnu)
fprintf('  unu =  %d \n',unu)
fprintf('nz =  %d \n',nz)
if nu < lnu
      fprintf('** Warning; nu is too small ** \n')
else
      if nu > unu
           fprintf('** Warning; nu is too big ** \n')
      end
end

fprintf('C/m  =  %.15f ',C/m)
fprintf('    (C nu)/2 =  %.15f \n',(C*nu)/2)
fprintf('sum(lambda) =  %.15f ',sum(lamb))
fprintf('    sum(mu) =  %.15f \n',sum(mu))
lamsv = 0; musv = 0; xi = 0; xip = 0;
if numsvl1 > 0 && numsvm1 > 0
   sx1 = zeros(n,1); sy1 = 0; num1 = 0;
   sx2 = zeros(n,1); sy2 = 0; num2 = 0;
   for i = 1:m
       if lambnz(i) > 0
          sx1 = sx1 + X(i,:)'; sy1 = sy1 + y(i);
          num1 = num1 + 1;
       end
       if munz(i) > 0
          sx2 = sx2 + X(i,:)'; sy2 = sy2 + y(i);
          num2 = num2 + 1;
       end
   end
```