```
function x = market_portfolio(f, r, Sig)

% Define function func such that
%   func(sig) = 0   when   risk_free_rate(sig, r, Sig) = f.
func = @(sig) risk_free_rate(sig,r,Sig) - f;

% Compute the minimum value of sig
cvx_begin quiet
    variable x(size(r))
    minimize (x' * Sig * x)
    subject to
        sum(x) == 1;
        x >= 0;
cvx_end

% Lower and upper bounds for sig
sig1 = sqrt( x' * Sig * x);
sig2 = sqrt(max(diag(Sig)));

% Use BinarySearch to solve func(sig) = 0
sig = BinarySearch(func, sig1, sig2);

% The market portfolio is the portfolio on the efficient frontier with
 risk
% equal to the sig satisfying   risk_free_rate(sig, r, Sig) = f.
cvx_begin quiet
    variable x(size(r))
    maximize (r'* x)
    subject to
        sum(x) == 1;
        x >= 0;
        norm(sqrtm(Sig)*x) <= sig;
cvx_end  ;
end


function rate = risk_free_rate(sig, r, Sig)
n = length(r);
[sqrtSig, resnorm] = sqrtm(Sig);

% Dual multiplier lambda gives slope of efficient frontier at the
 point
% (r'*x, sqrt(x'*Sig*x)), where x is the portfolio with maximum
 expected
% rate of return with risk at most sig.
cvx_begin quiet
    variable x(n)
    dual variable lambda
    maximize( r'*x )
    subject to
        norm(sqrtSig*x) <= sig : lambda
        sum(x) == 1
        x >= 0
cvx_end

rmax = r'*x;
% The risk-free rate is the y-intercept of the line tangent to the
% efficient frontier at the point (r'*x, sqrt(x'*Sig*x)).
rate = lambda*(-sig) + rmax;
end

function x = BinarySearch(func, x1, x2)
% This is a generic binary search routine.
% Given x1 and x2, with
%   func(x1) < 0  and  func(x2) > 0, or
%   func(x1) > 0  and  func(x2) < 0,
% returns x with abs(func(x)) < 1e-6.

disp('Binary search:');
if func(x1) > 0 && func(x2) < 0
    % Swap x1 and x2
    tmp = x1;  x1 = x2;  x2 = x1;
end
ii = 0; y = Inf;
fprintf('%4s%10s%10s\n', 'Iter', 'x', '|f(x)|');
while abs(y) > 1e-6
    ii = ii + 1;  x = (x1 + x2)/2;  y = func(x);
    fprintf('%4d%10.2e%10.2e\n', ii, x, abs(y));
    if y < 0
        x1 = x;
    else
        x2 = x;
    end
end
disp('Done.');

end
```

Published with MATLAB® R2018a