

1)

```
In [17]: 0.8 * 0.6 * 0.6 * 0.1
```

```
Out[17]: 0.0288
```

2) a)

$$\alpha(x_{\text{cand}} | x_{i-1}) = \min(1, (q(x_{i-1}|x_{\text{cand}}) f(x_{\text{cand}})) / (q(x_{\text{cand}} | x_{i-1}) f(x_{i-1})))$$

b)

```

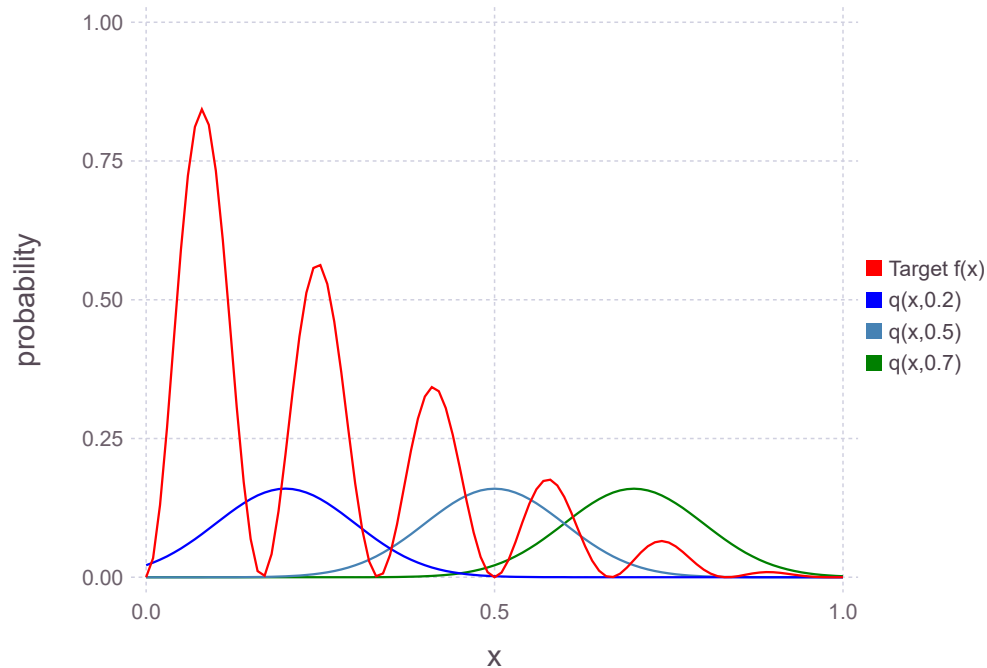
1) initialize  $x_0 \sim q$ 
2) for  $i = 1, 2, \dots, n$  do
    3) propose  $x_{\text{cand}} \sim q(x_i | x_{i-1})$ 
    4) Accept Probability:
         $\alpha(x_{\text{cand}} | x_{i-1}) = \min(1, (q(x_{i-1} | x_{\text{cand}}) * f(x_{\text{cand}})) / (q(x_{\text{cand}} | x_{i-1}) * f(x_{i-1})))$ 
    5)  $u \sim \text{unif}(0, 1)$ 
    6) if  $u < \alpha$ 
        7) Accept Proposal:  $x_i = x_{\text{cand}}$ 
    8) else
        9) Reject Proposal:  $x_i = x_{i-1}$ 
    10) end if
11) end
```

c) Yes. it's possible for the proposal distribution to generate a sample $x_{\text{cand}} = 0.3$ when $x_{i-1} = 0.5$

```
In [2]: using Distributions;
        using Gadfly;
        using StatsBase;
```

```
In [18]: x=collect(0:0.01:1);
f(x) = (1.-x).^2.*sin.(6.*pi.*x).^2;
q(x,y) = pdf.(Normal.(y,0.1),x);
plot(layer(x=x,y=f(x),Geom.line,Theme(default_color=colorant"red")),
layer(x=x,y=q(x,0.2)/25,Geom.line,Theme(default_color=colorant"blue")),
layer(x=x,y=q(x,0.5)/25,Geom.line,Theme(default_color=colorant"steelblue")),
layer(x=x,y=q(x,0.7)/25,Geom.line,Theme(default_color=colorant"green")),
Guide.ylabel("probability"),Guide.xlabel("x"), Guide.manual_color_key("",
["Target f(x)", "q(x,0.2)", "q(x,0.5)", "q(x,0.7)"], ["red","blue","steelblue","g
```

Out[18]:



d)

```
In [19]: function metropolis_hastings(n)
    x = zeros(n);
    count = 1;
    x[1] = abs(rand(Normal(0,0.1)));
    while(count < n)
        x_cand = rand(Normal(x[count],0.1));
        if((x_cand < 0) | (x_cand > 1))
            continue;
        end
        rho = (q(x[count],x_cand)/
            q(x_cand,x[count]))*(f(x_cand)/f(x[count]));
        alpha = minimum([1,rho]);
        u = rand();
        count = count + 1
        if (u < alpha)
            x[count] = x_cand;
        else
            x[count] = x[count-1];
        end
    end
    return x;
end
```

Out[19]: metropolis_hastings (generic function with 1 method)

```
In [20]: samples = metropolis_hastings(10000);
```

e)

```
In [22]: hist = [fit(Histogram,samples,collect(0:0.01:1)).weights; 0]./800;
plot(layer(x=x,y=f(x),Geom.line,Theme(default_color=colorant"red")),
layer(x=collect(0:0.01:1),y=hist, Geom.bar,
Theme(default_color=colorant"SteelBlue")),
Guide.manual_color_key("", ["Target f(x)", "Samples"],
["red", "SteelBlue"]),
Guide.ylabel("probability"),Guide.xlabel("x"),
Coord.Cartesian(xmin=0,xmax=1))
```

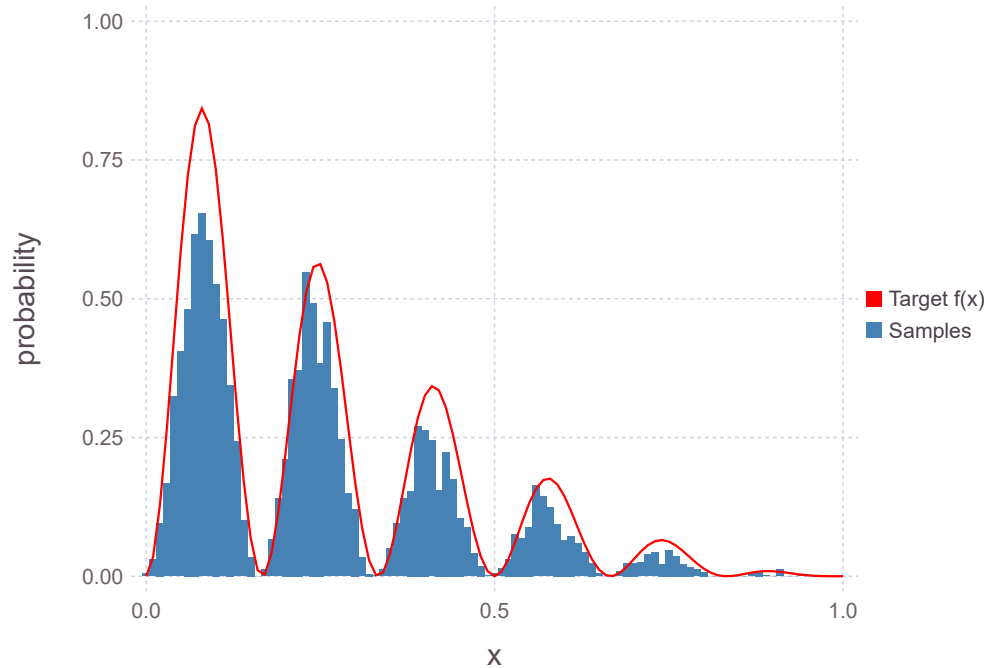
WARNING: Default for keyword argument "closed" has changed from :right to :left.

To avoid this warning, specify closed=:right or closed=:left as appropriate.

Stacktrace:

```
[1] depwarn(::String, ::Symbol) at ./deprecated.jl:70
[2] _check_closed_arg at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:12 [inlined]
[3] #fit#108(::Symbol, ::Function, ::Type{StatsBase.Histogram{Int64,N,E} where E where N}, ::Tuple{Array{Float64,1}}, ::Tuple{Array{Float64,1}}) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:282
[4] (::StatsBase.#kw##fit)(::Array{Any,1}, ::StatsBase.#fit, ::Type{StatsBase.Histogram{Int64,N,E} where E where N}, ::Tuple{Array{Float64,1}}, ::Tuple{Array{Float64,1}}) at ./<missing>:0
[5] fit(::Type{StatsBase.Histogram{Int64,N,E} where E where N}, ::Array{Float64,1}, ::Array{Float64,1}) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:225
[6] #fit#112(::Array{Any,1}, ::Function, ::Type{StatsBase.Histogram}, ::Array{Float64,1}, ::Vararg{Array{Float64,1},N} where N) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:340
[7] fit(::Type{StatsBase.Histogram}, ::Array{Float64,1}, ::Array{Float64,1}) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:340
[8] include_string(::String, ::String) at ./loading.jl:522
[9] include_string(::Module, ::String, ::String) at /users/PES0801/nifaullah/.julia/v0.6/Compat/src/Compat.jl:84
[10] execute_request(::ZMQ.Socket, ::IJulia.Msg) at /usr/local/julia/0.6.4/site/v0.6/IJulia/src/execute_request.jl:180
[11] (::Compat.#inner#6{Array{Any,1},IJulia.#execute_request,Tuple{ZMQ.Socket,IJulia.Msg}})() at /users/PES0801/nifaullah/.julia/v0.6/Compat/src/Compat.jl:125
[12] eventloop(::ZMQ.Socket) at /usr/local/julia/0.6.4/site/v0.6/IJulia/src/eventloop.jl:8
[13] (::IJulia.##15#18)() at ./task.jl:335
while loading In[22], in expression starting on line 1
```

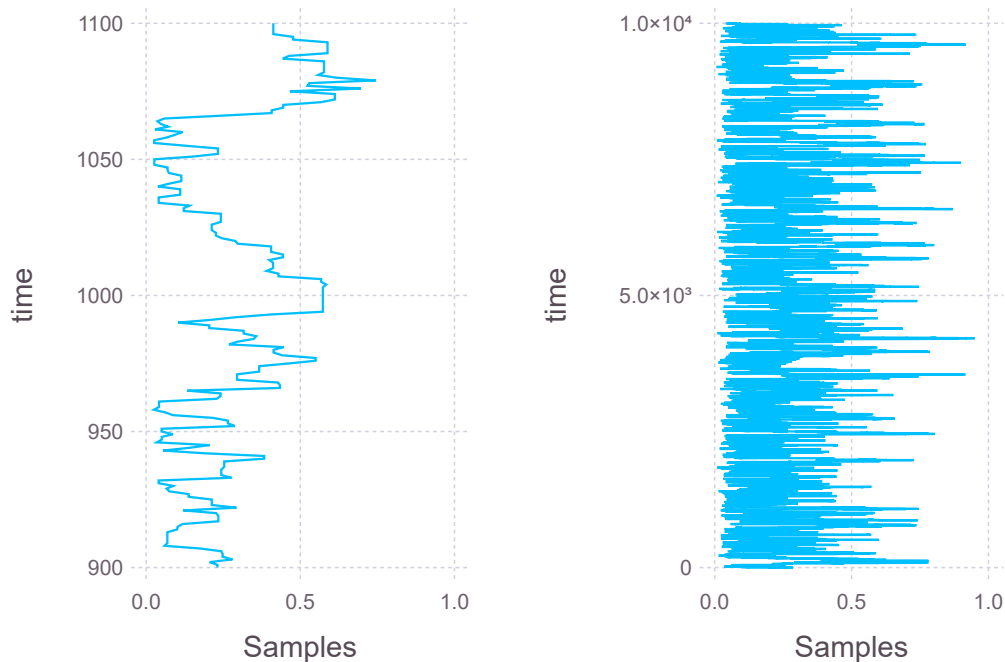
Out[22]:



f)

```
In [23]: myplot1 = plot(y=900:1100,x=samples[900:1100], Geom.path, Guide.ylabel("time"),
Guide.xlabel("Samples"),Coord.Cartesian(xmin=0,xmax=1));
myplot2 = plot(y=1:10000,x=samples[1:10000], Geom.path, Guide.ylabel("time"),
Guide.xlabel("Samples"),Coord.Cartesian(xmin=0,xmax=1));
myplot= hstack(myplot1,myplot2)
```

Out[23]:

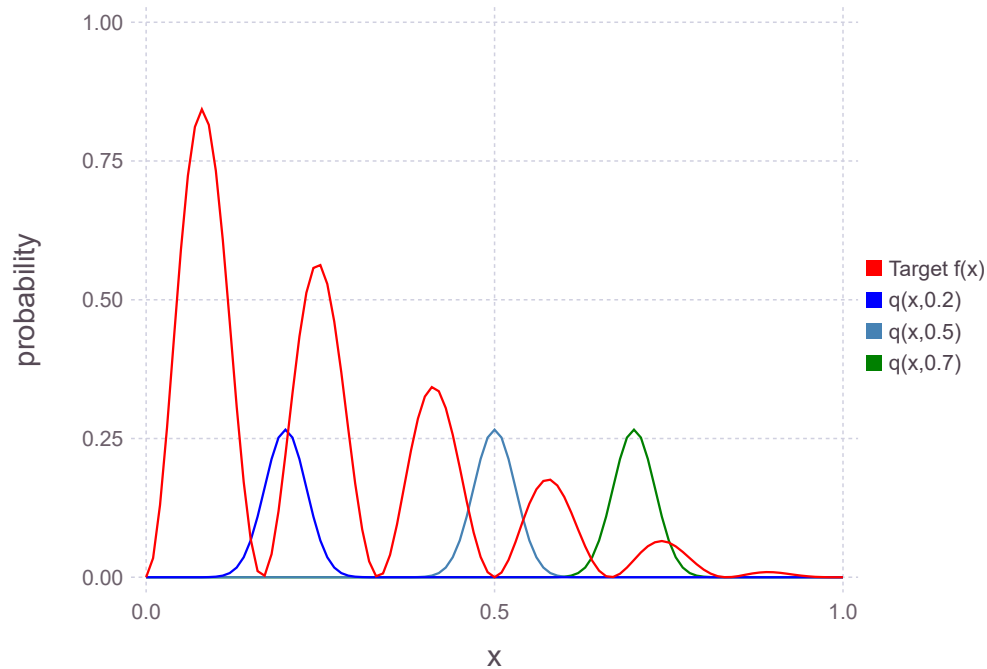


3) a) Yes. it is possible for the proposal distribution to generate a sample $x_{\text{cand}} = 0.3$ when $x_{i-1} =$

0.5

```
In [24]: q(x,y) = pdf.(Normal.(y,0.03),x);
plot(layer(x=x,y=f(x),Geom.line,Theme(default_color=colorant"red")),
layer(x=x,y=q(x,0.2)/50,Geom.line,Theme(default_color=colorant"blue")),
layer(x=x,y=q(x,0.5)/50,Geom.line,Theme(default_color=colorant"steelblue")),
layer(x=x,y=q(x,0.7)/50,Geom.line,Theme(default_color=colorant"green")),
Guide.ylabel("probability"),Guide.xlabel("x"), Guide.manual_color_key("",
["Target f(x)", "q(x,0.2)", "q(x,0.5)", "q(x,0.7)"], ["red","blue","steelblue","g
```

Out[24]:



b)

```
In [25]: function metropolis_hastings_1(n)
        x = zeros(n);
        count = 1;
        x[1] = abs(rand(Normal(0,0.3)));
        while(count < n)
            x_cand = rand(Normal(x[count],0.3));
            if((x_cand < 0) | (x_cand > 1))
                continue;
            end
            rho = (q(x[count],x_cand)/
                q(x_cand,x[count]))*(f(x_cand)/f(x[count]));
            alpha = minimum([1,rho]);
            u = rand();
            count = count + 1
            if (u < alpha)
                x[count] = x_cand;
            else
                x[count] = x[count-1];
            end
        end
        return x;
    end
```

Out[25]: metropolis_hastings_1 (generic function with 1 method)

```
In [10]: samples_1 = metropolis_hastings_1(10000);
```

c)

```
In [26]: hist = [fit(Histogram,samples_1,collect(0:0.01:1)).weights; 0]./800;
plot(layer(x=x,y=f(x),Geom.line,Theme(default_color=colorant"red")),
layer(x=collect(0:0.01:1),y=hist, Geom.bar,
Theme(default_color=colorant"SteelBlue")),
Guide.manual_color_key("", ["Target f(x)", "Samples"],
["red", "SteelBlue"]),
Guide.ylabel("probability"),Guide.xlabel("x"),
Coord.Cartesian(xmin=0,xmax=1))
```

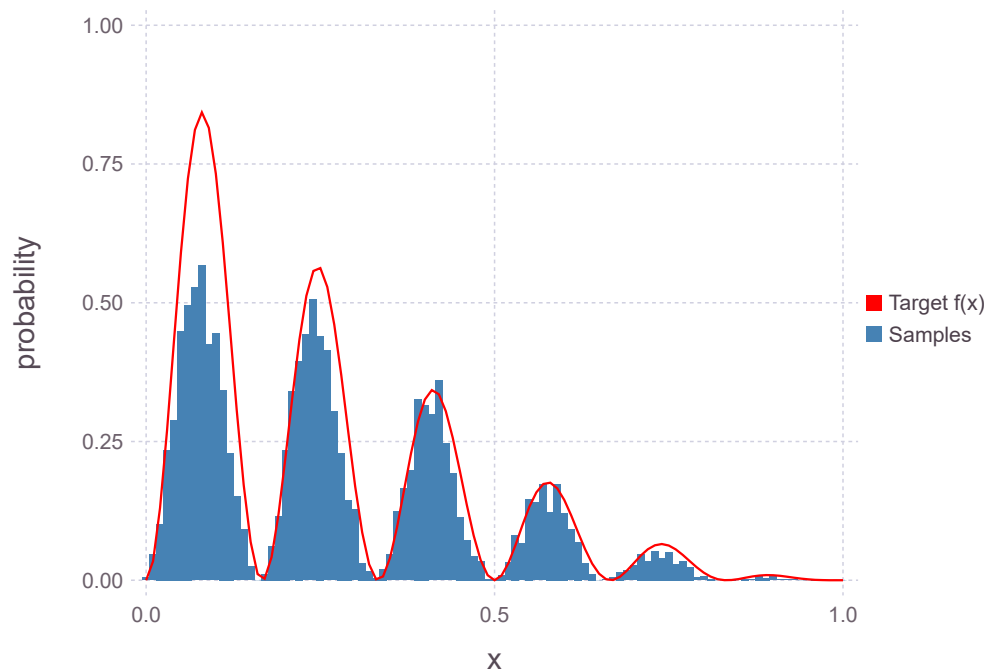
WARNING: Default for keyword argument "closed" has changed from :right to :left.

To avoid this warning, specify closed=:right or closed=:left as appropriate.

Stacktrace:

```
[1] depwarn(::String, ::Symbol) at ./deprecated.jl:70
[2] _check_closed_arg at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:12 [inlined]
[3] #fit#108(::Symbol, ::Function, ::Type{StatsBase.Histogram{Int64,N,E} where E where N}, ::Tuple{Array{Float64,1}}, ::Tuple{Array{Float64,1}}) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:282
[4] (::StatsBase.#kw##fit)(::Array{Any,1}, ::StatsBase.#fit, ::Type{StatsBase.Histogram{Int64,N,E} where E where N}, ::Tuple{Array{Float64,1}}, ::Tuple{Array{Float64,1}}) at ./<missing>:0
[5] fit(::Type{StatsBase.Histogram{Int64,N,E} where E where N}, ::Array{Float64,1}, ::Array{Float64,1}) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:225
[6] #fit#112(::Array{Any,1}, ::Function, ::Type{StatsBase.Histogram}, ::Array{Float64,1}, ::Vararg{Array{Float64,1},N} where N) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:340
[7] fit(::Type{StatsBase.Histogram}, ::Array{Float64,1}, ::Array{Float64,1}) at /users/PES0801/nifaullah/.julia/v0.6/StatsBase/src/hist.jl:340
[8] include_string(::String, ::String) at ./loading.jl:522
[9] include_string(::Module, ::String, ::String) at /users/PES0801/nifaullah/.julia/v0.6/Compat/src/Compat.jl:84
[10] execute_request(::ZMQ.Socket, ::IJulia.Msg) at /usr/local/julia/0.6.4/site/v0.6/IJulia/src/execute_request.jl:180
[11] (::Compat.#inner#6{Array{Any,1},IJulia.#execute_request,Tuple{ZMQ.Socket,IJulia.Msg}})() at /users/PES0801/nifaullah/.julia/v0.6/Compat/src/Compat.jl:125
[12] eventloop(::ZMQ.Socket) at /usr/local/julia/0.6.4/site/v0.6/IJulia/src/eventloop.jl:8
[13] (::IJulia.##15#18)() at ./task.jl:335
while loading In[26], in expression starting on line 1
```

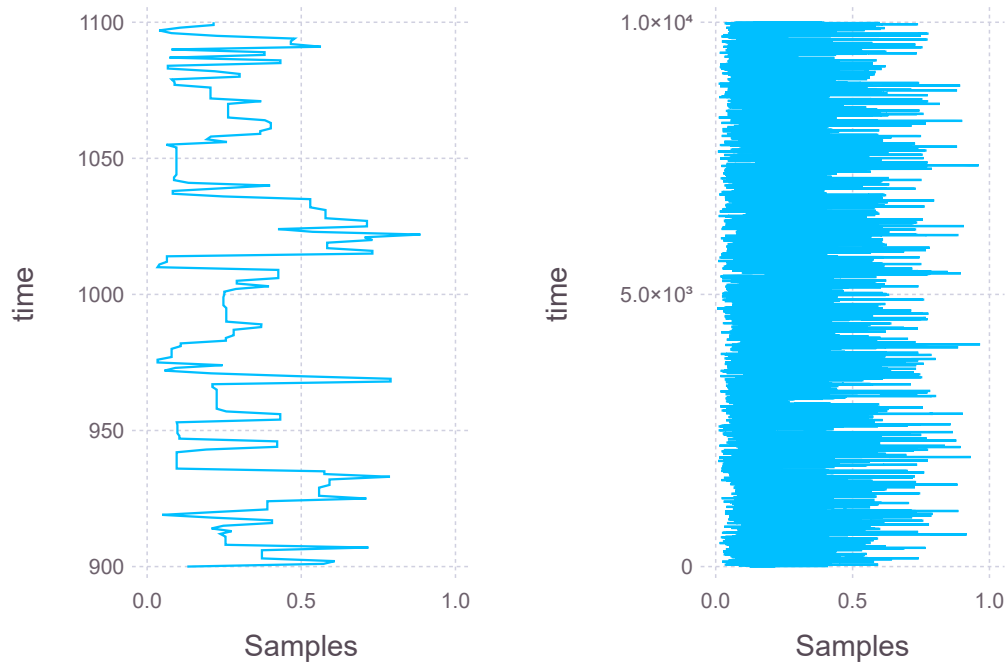
Out[26]:



d) With a narrower distribution, the number of samples generated in the upper end of the probability spectrum is less compared to the previous candidate and the samples generated are thickly concentrated between 0.0 to 0.5

```
In [27]: myplot1 = plot(y=900:1100,x=samples_1[900:1100], Geom.path, Guide.ylabel("time"),
Guide.xlabel("Samples"),Coord.Cartesian(xmin=0,xmax=1));
myplot2 = plot(y=1:10000,x=samples_1[1:10000], Geom.path, Guide.ylabel("time"),
Guide.xlabel("Samples"),Coord.Cartesian(xmin=0,xmax=1));
myplot= hstack(myplot1,myplot2)
```

Out[27]:



e) The main reason is the variance of the candidate distribution, less the variance more quickly it reaches a equilibrium and then revolves around that equilibrium but higher variance it sort of checks around all values in the space.