

```

In [54]: function acc_mc(S, X, H, r, sigma, settlement, T, M, ngrid)
    dt = 1 / T
    npath = 1000
    nsample = M ÷ npath
    price = zeros(nsample)
    strikepath = fill(X, T)

    for i in 1:nsample
        p = 0
        for j in 1:npath
            settle = vcat(0, settlement)
            accumulation = ones(T)
            spath, knockout, discount = stockpath(S, H, r, sigma, T, ngrid)
            if knockout > 0
                accumulation[knockout:T] .= 0
                for ix in 1:length(settle)-1
                    if settle[ix] >= knockout
                        settle[ix] = knockout
                        settle = settle[1:ix]
                        break
                    end
                end
            end
            accumulation = (1*(spath .> strikepath) + 2*(spath .<= strikepath)) .*
            for k in 1:length(settle)-1
                p += sum(accumulation[settle[k]+1:settle[k+1]]) * (spath[settle[k+1]
            end
            price[i] = p / npath
        end
    end
    value = mean(price)
    se = sqrt(var(price) / nsample)
    return value, se
end

function stockpath(S, q, H, r, sigma, T, ngrid)
    N = T * ngrid
    dt = 1 / N
    s = zeros(N + 1)
    discount = zeros(N)
    s[1] = S

    eps = exp.((r - q - 0.5 * sigma^2) * dt .+ sigma * sqrt(dt) .* randn(N))

    knockout = 0
    stayin = 1
    for i in 2:(N+1)
        s[i] = s[i - 1] * eps[i - 1]
        if ((s[i] > H) && (stayin == 1))
            knockout = floor(Int, (i - 1) / ngrid)
            stayin = 0
        end
    end
    path = s[1 + ngrid:ngrid:N + 1]
    discount = exp.(-r * dt * collect(1:N))
    return path, knockout, discount
end

```

Out[54]: stockpath (generic function with 2 methods)

```

In [55]: # Inputs
S = 46.65349

```

```
X = 42.1746
H = 48.3963
r = 0.037
sigma = 0.169
settlement = [10,19,29,38,48,58,68,78,88,98,107,117,126,136,146,156,164,173,183,193]
T = 250
M = 1000000
ngrid = 1

# Call the function
path, knockout, discount = stockpath(S, H, r, sigma, T, ngrid)

# Now `path`, `knockout`, and `discount` contain the output of the `stockpath` func
println("Path: ", path)
println("Knockout: ", knockout)
println("Discount: ", discount)
```

Path: [46.41891710806904, 46.28269489359133, 46.747896462863864, 46.39877100017320  
5, 45.71266163095988, 45.94849237042383, 46.156231312595445, 45.23995595728177, 4  
5.260353406497856, 45.562635266808535, 45.57863375346337, 45.30125020014804, 45.81  
917664619633, 46.14630986835438, 46.73788941294184, 46.17144620483753, 46.26819483  
2608465, 47.15704857463433, 46.34757230135657, 46.423233100879784, 45.852628123583  
93, 45.75504653897442, 45.44922797158729, 45.65082797863982, 45.067149345436015, 4  
4.516759965083146, 44.04084569277266, 43.99378570642838, 44.35234721533518, 44.555  
93595634509, 44.400813599625735, 43.51388936407075, 44.41279214088941, 44.08429987  
861695, 44.54510528693955, 43.77253664078503, 44.114929351128325, 44.7091701886496  
6, 44.96994113351859, 45.20872303706463, 45.379937569799246, 45.12615258178695, 4  
5.28352686905306, 45.88111435452315, 45.64617931720383, 45.95027179208633, 45.9093  
781230004, 46.41742563116933, 45.8370095187314, 45.45330537229548, 45.555714061470  
17, 44.85922713699061, 44.72409629322351, 45.41968841694925, 46.319745279851766, 4  
6.3428484501764, 46.73224268671783, 46.32836442995752, 47.4390637795211, 47.830760  
518740675, 47.394631286881754, 46.621456011279236, 46.314440895368776, 47.12154314  
12419, 47.32314631346165, 47.67348010287254, 47.880013013273974, 47.03139618327609  
4, 46.83913747901427, 46.37872662529021, 46.11683076805978, 46.035546933838226, 4  
5.66741291949201, 45.2684081392002, 45.44872030280496, 45.37061718510691, 45.40141  
9463681954, 45.99714851847895, 45.77331705035809, 46.05327145772193, 46.4404220696  
8988, 46.6302000634643, 46.54803489153091, 46.58396872991522, 45.552323140360784,  
46.25168475981004, 45.768953553078234, 45.43566471142222, 45.62960208393134, 45.75  
5601262377176, 46.095501365234554, 46.61326483433445, 46.35172752808094, 45.882491  
74459412, 46.09262771624785, 45.482008686282825, 45.44461075287708, 47.18299470452  
717, 46.520965983162675, 46.93717788636648, 47.43098940284105, 47.20740968408825,  
47.4033891823806, 49.433343673812914, 50.25724078910781, 50.42141360212086, 50.214  
331803709136, 50.41084363136805, 50.63116657137152, 49.935933588376855, 49.5814185  
1711888, 49.22540940843894, 48.514972969612174, 48.081842947737826, 48.46746552753  
4976, 48.82788472934373, 48.16754266983056, 48.5905872211898, 48.108940286374285,  
48.89182218552834, 48.96611423835189, 48.64804219299482, 48.731349283167525, 49.92  
107761321322, 49.898873974878626, 49.94768753805968, 50.006409071137604, 50.258919  
03781999, 50.39098427647686, 51.019950754686434, 51.91603535178277, 52.10382322848  
806, 51.57027414006636, 50.92519025595189, 50.09427702040658, 50.17522641793043, 4  
9.52064160083224, 50.61562121392289, 50.43030161004245, 50.14219173230662, 50.5934  
4145525525, 50.527597106776874, 50.40773688698788, 50.23816796008409, 50.967053498  
746424, 50.83199130387681, 51.48033596194146, 51.26525862765909, 51.8628494029231  
8, 51.56142219491611, 52.263612098871356, 51.6587860446642, 51.5424863249799, 51.2  
8294965428479, 51.76358452913292, 52.28546965919719, 52.59276667098936, 52.5396493  
15619236, 53.27812948890933, 52.68027430523745, 51.91446860540395, 53.108650344725  
31, 53.0638565042891, 52.253567898431044, 52.68303291547608, 51.9640292524778, 52.  
15484944921853, 51.812503760921366, 51.77166348251096, 51.95840969240701, 51.94882  
817671918, 52.79154382589707, 53.06061081363481, 53.50106595595009, 52.64843395567  
554, 51.907113654559055, 52.350892916385284, 52.55839122640851, 52.54617599702127,  
51.921228041619806, 51.49853647148679, 51.544028585113885, 51.286271575839166, 50.  
4901654910901, 50.974079636145945, 51.49502403693179, 52.316935431365515, 52.30053  
957592831, 51.940570823696056, 52.207903157500034, 52.21340062721039, 52.135456817  
819716, 51.659831169110156, 51.788072244656036, 52.15426322367225, 52.807263581918  
015, 52.67486509263275, 52.815932484842264, 52.21951466650931, 50.892975496142, 5  
0.85310720240742, 49.79463552358017, 48.89281601929356, 48.03192740883896, 47.4949  
5698535318, 47.78750691431128, 47.73950172396025, 47.05746742566028, 46.7538389272  
565, 46.78275273539018, 45.961391014846924, 46.21766029622718, 46.57184228514311,  
46.89877659950565, 46.25650920313257, 45.92290248206543, 46.474802109325424, 46.51  
5350521245864, 45.851333461172175, 45.3993044403480066, 44.925468106365265, 44.9451  
5521667715, 44.570368688584864, 45.28391836241227, 45.010458451223975, 44.53300609  
9132564, 44.0238857983241, 43.75404582644065, 43.85417368879285, 44.0998918395842  
4, 44.27490894827438, 44.51240194644761, 44.13865929807145, 43.5625176829194, 42.5  
9134993338139, 42.92109668391608, 43.37315661580449, 43.58770339746345, 43.2511901  
7861222, 43.28988438404365, 43.801408415157304, 43.54622312007182, 43.647140737586  
57, 43.304707600921354, 43.18959661204335, 42.88557552209886, 42.44665305263516, 4  
2.862620677956144, 42.79039311599958, 42.4577985511034]

Knockout: 104

Discount: [0.9998520109514597, 0.9997040438036779, 0.9995560985534135, 0.999408175  
197426, 0.9992602737324752, 0.9991123941553214, 0.9989645364627255, 0.998816700651  
4489, 0.9986688867182534, 0.9985210946599011, 0.9983733244731551, 0.99822557615477  
82, 0.9980778497015346, 0.9979301451101881, 0.9977824623775036, 0.997634801500246

2, 0.9974871624751814, 0.9973395452990756, 0.9971919499686952, 0.9970443764808073, 0.9968968248321796, 0.9967492950195798, 0.9966017870397766, 0.9964543008895391, 0.9963068365656366, 0.996159394064839, 0.9960119733839169, 0.9958645745196412, 0.995717197468783, 0.9955698422281144, 0.9954225087944076, 0.9952751971644356, 0.9951279073349715, 0.9949806393027891, 0.9948333930646627, 0.9946861686173669, 0.9945389659576771, 0.9943917850823688, 0.9942446259882182, 0.9940974886720019, 0.9939503731304971, 0.9938032793604812, 0.9936562073587324, 0.9935091571220293, 0.9933621286471507, 0.9932151219308762, 0.993068136969986, 0.9929211737612601, 0.9927742323014797, 0.9926273125874261, 0.9924804146158811, 0.9923335383836273, 0.9921866838874472, 0.9920398511241244, 0.9918930400904425, 0.9917462507831858, 0.991599483199139, 0.9914527373350873, 0.9913060131878165, 0.9911593107541126, 0.9910126300307622, 0.9908659710145525, 0.9907193337022712, 0.9905727180907061, 0.9904261241766459, 0.9902795519568796, 0.9901330014281966, 0.9899864725873869, 0.989839965431241, 0.9896934799565497, 0.9895470161601043, 0.989400574038697, 0.9892541535891197, 0.9891077548081656, 0.9889613776926277, 0.9888150222393, 0.9886686884449764, 0.9885223763064518, 0.9883760858205215, 0.9882298169839809, 0.9880835697936263, 0.9879373442462543, 0.9877911403386619, 0.9876449580676466, 0.9874987974300066, 0.9873526584225403, 0.9872065410420466, 0.9870604452853251, 0.9869143711491756, 0.9867683186303985, 0.986622877257946, 0.9864762784321655, 0.9863302907463127, 0.9861843246650387, 0.9860383801851462, 0.9858924573034384, 0.985746556016719, 0.9856006763217923, 0.9854548182154628, 0.9853089816945356, 0.9851631667558165, 0.9850173733961113, 0.9848716016122268, 0.9847258514009699, 0.984580122759148, 0.9844344156835693, 0.9842887301710419, 0.984143066218375, 0.9839974238223778, 0.9838518029798602, 0.9837062036876326, 0.9835606259425057, 0.9834150697412908, 0.9832695350807996, 0.9831240219578443, 0.9829785303692378, 0.9828330603117928, 0.9826876117823234, 0.9825421847776434, 0.9823967792945675, 0.9822513953299107, 0.9821060328804884, 0.9819606919431167, 0.9818153725146122, 0.9816700745917915, 0.9815247981714722, 0.981379543250472, 0.9812343098256096, 0.9810890978937034, 0.9809439074515729, 0.9807987384960377, 0.9806535910239182, 0.9805084650320349, 0.9803633605172092, 0.9802182774762624, 0.9800732159060169, 0.9799281758032953, 0.9797831571649203, 0.9796381599877156, 0.9794931842685053, 0.9793482300041138, 0.9792032971913658, 0.9790583858270869, 0.9789134959081031, 0.9787686274312404, 0.9786237803933258, 0.9784789547911864, 0.9783341506216502, 0.9781893678815453, 0.9780446065677002, 0.9778998666769444, 0.9777551482061072, 0.9776104511520187, 0.9774657755115097, 0.9773211212814111, 0.9771764884585544, 0.9770318770397715, 0.9768872870218949, 0.9767427184017573, 0.9765981711761924, 0.9764536453420339, 0.9763091408961161, 0.9761646578352736, 0.976020196156342, 0.9758757558561567, 0.975731336931554, 0.9755869393793706, 0.9754425631964435, 0.9752982083796103, 0.9751538749257093, 0.9750095628315786, 0.9748652720940575, 0.9747210027099853, 0.9745767546762021, 0.974432527989548, 0.9742883226468642, 0.9741441386449917, 0.9739999759807726, 0.973855834651049, 0.9737117146526636, 0.9735676159824596, 0.9734235386372807, 0.973279482613971, 0.9731354479093752, 0.9729914345203383, 0.9728474424437058, 0.9727034716763238, 0.9725595222150385, 0.9724155940566971, 0.9722716871981469, 0.9721278016362358, 0.9719839373678122, 0.9718400943897246, 0.9716962726988226, 0.9715524722919557, 0.9714086931659743, 0.971264935317729, 0.9711211987440708, 0.9709774834418513, 0.9708337894079228, 0.9706901166391375, 0.9705464651323487, 0.9704028348844096, 0.9702592258921742, 0.970115638152497, 0.9699720716622328, 0.9698285264182368, 0.969685002417365, 0.9695414996564734, 0.969398018132419, 0.9692545578420587, 0.9691111187822503, 0.968967700949852, 0.9688243043417222, 0.9686809289547199, 0.9685375747857048, 0.9683942418315368, 0.9682509300890761, 0.9681076395551841, 0.9679643702267218, 0.9678211221005509, 0.9676778951735342, 0.967534689442534, 0.9673915049044136, 0.9672483415560369, 0.9671051993942679, 0.9669620784159711, 0.9668189786180118, 0.9666758999972555, 0.966532842550568, 0.9663898062748161, 0.9662467911668664, 0.9661037972235865, 0.9659608244418443, 0.965817872818508, 0.9656749423504464, 0.9655320330345287, 0.9653891448676247, 0.9652462778466047, 0.9651034319683391, 0.9649606072296991, 0.9648178036275564, 0.9646750211587828, 0.9645322598202509, 0.9643895196088338, 0.9642468005214047, 0.9641041025548375, 0.9639614257060066, 0.9638187699717868, 0.9636761353490535]

```
In [57]: using Statistics
# Call the acc_mc function
price, standard_error = acc_mc(S, X, H, r, sigma, settlement, T, M, ngrid)

# Print the output
println("Price: ", price)
```

```
println("Standard Error: ", standard_error)
# Open a file
f = open("output.txt", "w")

# Write to the file
println(f, "Price: ", price)
println(f, "Stderr: ", stderr)

# Close the file
close(f)
```

Price: 5.4446975751855184  
Standard Error: 0.5248537867969437

In [ ]: