CHEME 5660 Actual Prelim 2 Equity Question - Lukas Wenzl

You are a Quant at Olin Financial, an up-and-coming hedge fund. You have been tasked with computing the distribution of possible future share price values for firm XYZ. Suppose the share price of firm XYZ at time t, denoted by $S(t)$, is governed by a geometric Brownian motion with the solution:

$$S(t) = S_{\circ}\exp\left(\left(\mu - \frac{\sigma^2}{2}\right)(t - t_{\circ}) + \sigma\sqrt{t - t_{\circ}}\cdot Z(0,1)\right)$$

where $S_{\circ}$ denotes the initial share price at $t_{\circ} < t$, μ denotes the rate of return parameter, $\sigma > 0$ denotes the volatility parameter and $Z(0,1)$ denotes a standard-normal random variable.

Assumptions: (i) the implied volatility of an At the Money (ATM) `01/20/2023` option on XYZ is 52.27\%; (ii) there are 78 days to 01/20/2023 (from today); (iii) the current share price of XYZ is 270.89 USD/share.

Use the Jupyter notebook `CHEME-5660-PP2-Equity.ipynb`, and any associated data sets, or other course materials to answer the following questions:

- a) Estimate the rate of return parameter μ from historical OHLC data for firm XYZ.
- b) Estimate the volatility parameter σ from historical OHLC data for firm XYZ.
- c) Using your estimates of the μ and σ parameters, along with the analytical GBM solution, estimate a distribution of possible future share price values of XYZ on `01/20/2023`. Generate N = 10,000 sample paths and let $t_{\circ} = 0$ (now).

Solution

In [1]:
```
import Pkg; Pkg.activate("."); Pkg.resolve(); Pkg.instantiate();
```

```
  Activating project at `~/OtherCodes/CHEME-5660-Markets-Mayhem-Example-Notebooks/prelims/P2/actual`
    Updating `~/OtherCodes/CHEME-5660-Markets-Mayhem-Example-Notebooks/prelims/P2/actual/Project.toml`
  [336ed68f] + CSV v0.10.4
  [5ae59095] + Colors v0.12.8
  [a93c6f00] + DataFrames v1.3.4
  [31c24e10] + Distributions v0.25.68
  [033835bb] + JLD2 v0.4.22
  [6e0fd8a8] + PQEcolaPoint v0.1.2
  [91a5bcdd] + Plots v1.31.7
  [f3b207a7] + StatsPlots v0.15.1
    Updating `~/OtherCodes/CHEME-5660-Markets-Mayhem-Example-Notebooks/prelims/P2/actual/Manifest.toml`
  [621f4979] + AbstractFFTs v1.2.1
  [79e6a3ab] + Adapt v3.4.0
  [7d9fca2a] + Arpack v0.5.3
  [13072b0f] + AxisAlgorithms v1.0.1
  [fbb218c0] + BSON v0.3.5
  [336ed68f] + CSV v0.10.4
  [49dc2e85] + Calculus v0.5.1
  [d360d2e6] + ChainRulesCore v1.15.3
  [9e997f8a] + ChangesOfVariables v0.1.4
  [aaaa29a8] + Clustering v0.14.2
  [944b1d66] + CodecZlib v0.7.0
  [35d6a980] + ColorSchemes v3.19.0
  [3da002f7] + ColorTypes v0.11.4
  [c3611d14] + ColorVectorSpace v0.9.9
  [5ae59095] + Colors v0.12.8
⌅ [34da2185] + Compat v3.46.0
  [d38c429a] + Contour v0.6.2
  [a8cc5b0e] + Crayons v4.1.1
  [9a962f9c] + DataAPI v1.10.0
  [a93c6f00] + DataFrames v1.3.4
  [864edb3b] + DataStructures v0.18.13
  [e2d170a0] + DataValueInterfaces v1.0.0
  [e7dc6d0d] + DataValues v0.4.13
  [b429d917] + DensityInterface v0.4.0
  [b4f34e82] + Distances v0.10.7
  [31c24e10] + Distributions v0.25.68
  [ffbed154] + DocStringExtensions v0.9.1
  [fa6b7ba4] + DualNumbers v0.6.8
  [411431e0] + Extents v0.1.1
  [c87230d0] + FFMPEG v0.4.1
  [7a1cc6ca] + FFTW v1.5.0
  [5789e2e9] + FileIO v1.15.0
  [48062228] + FilePathsBase v0.9.18
  [1a297f60] + FillArrays v0.13.2
  [53c48c17] + FixedPointNumbers v0.8.4
  [59287772] + Formatting v0.4.2
  [28b8d3ca] + GR v0.66.2
  [cf35fbd7] + GeoInterface v1.0.1
  [5c1252a2] + GeometryBasics v0.4.3
  [42e2da0e] + Grisu v1.0.2
  [cd3eb016] + HTTP v1.3.1
  [34004b35] + HypergeometricFunctions v0.3.11
  [83e8ac13] + IniFile v0.5.1
  [842dd82b] + InlineStrings v1.1.4
  [a98d9a8b] + Interpolations v0.14.4
  [3587e190] + InverseFunctions v0.1.7
  [41ab1584] + InvertedIndices v1.1.0
  [92d709cd] + IrrationalConstants v0.1.1
  [c8e1da08] + IterTools v1.4.0
  [82899510] + IteratorInterfaceExtensions v1.0.0
  [033835bb] + JLD2 v0.4.22
  [692b3bcd] + JLLWrappers v1.4.1
  [682c06a0] + JSON v0.21.3
  [5ab0869b] + KernelDensity v0.6.5
  [b964fa9f] + LaTeXStrings v1.3.0
  [23fbe1c1] + Latexify v0.15.16
  [2ab3a3ac] + LogExpFunctions v0.3.18
  [e6f89c97] + LoggingExtras v0.4.9
  [1914dd2f] + MacroTools v0.5.9
  [739be429] + MbedTLS v1.1.3
  [442fdcdd] + Measures v0.3.1
  [e1d29d7a] + Missings v1.0.2
⌅ [6f286f6a] + MultivariateStats v0.9.1
  [77ba4419] + NaNMath v1.0.1
  [b8a86587] + NearestNeighbors v0.4.11
  [510215fc] + Observables v0.5.1
  [6fe1bfb0] + OffsetArrays v1.12.7
  [bac558e1] + OrderedCollections v1.4.1
  [90014a1f] + PDMats v0.11.16
  [6e0fd8a8] + PQEcolaPoint v0.1.2
  [69de0a69] + Parsers v2.3.2
  [ccf2f8ad] + PlotThemes v3.0.0
  [995b91a9] + PlotUtils v1.3.0
  [91a5bcdd] + Plots v1.31.7
  [2dfb63ee] + PooledArrays v1.4.2
  [21216c6a] + Preferences v1.3.0
  [08abe8d2] + PrettyTables v1.3.1
  [1fd47b50] + QuadGK v2.4.2
  [c84ed2f1] + Ratios v0.4.3
  [3cdcf5f2] + RecipesBase v1.2.1
  [01d81517] + RecipesPipeline v0.6.3
  [189a3867] + Reexport v1.2.2
  [05181044] + RelocatableFolders v0.3.0
  [ae029012] + Requires v1.3.0
  [79098fc4] + Rmath v0.7.0
  [6c6a2e73] + Scratch v1.1.1
  [91c51154] + SentinelArrays v1.3.13
  [992d4aef] + Showoff v1.0.3
  [777ac1f9] + SimpleBufferStream v1.1.0
  [a2af1166] + SortingAlgorithms v1.0.1
  [276daf66] + SpecialFunctions v2.1.7
  [90137ffa] + StaticArrays v1.5.6
  [1e83bf80] + StaticArraysCore v1.3.0
⌅ [82ae8749] + StatsAPI v1.2.2
  [2913bbd2] + StatsBase v0.33.21
  [4c63d2b9] + StatsFuns v1.0.1
  [f3b207a7] + StatsPlots v0.15.1
  [09ab397b] + StructArrays v0.6.12
  [ab02a1b2] + TableOperations v1.2.0
  [3783bdb8] + TableTraits v1.0.1
  [bd369af6] + Tables v1.7.0
  [62fd8b95] + TensorCore v0.1.1
  [3bb67fe8] + TranscodingStreams v0.9.8
  [5c2747f8] + URIs v1.4.0
  [1cfade01] + UnicodeFun v0.4.1
  [41fe7b60] + Unzip v0.1.2
  [ea10d353] + WeakRefStrings v1.4.2
  [cc8bc4a8] + Widgets v0.6.6
  [efce3f68] + WoodburyMatrices v0.5.5
  [68821587] + Arpack_jll v3.5.1+1
  [6e34b625] + Bzip2_jll v1.0.8+0
  [83423d85] + Cairo_jll v1.16.1+1
  [5ae413db] + EarCut_jll v2.2.3+0
  [2e619515] + Expat_jll v2.4.8+0
  [b22a6f82] + FFMPEG_jll v4.4.2+0
  [f5851436] + FFTW_jll v3.3.10+0
  [a3f928ae] + Fontconfig_jll v2.13.93+0
  [d7e528f0] + FreeType2_jll v2.10.4+0
  [559328eb] + FriBidi_jll v1.0.10+0
  [0656b61e] + GLFW_jll v3.3.8+0
  [d2c73de3] + GR_jll v0.66.2+0
  [78b55507] + Gettext_jll v0.21.0+0
  [7746bdde] + Glib_jll v2.68.3+2
  [3b182d85] + Graphite2_jll v1.3.14+0
  [2e76f6c2] + HarfBuzz_jll v2.8.1+1
  [1d5cc7b8] + IntelOpenMP_jll v2018.0.3+2
  [aacddb02] + JpegTurbo_jll v2.1.2+0
  [c1c5ebd0] + LAME_jll v3.100.1+0
  [88015f11] + LERC_jll v3.0.0+1
  [dd4b983a] + LZO_jll v2.10.1+0
  [e9f186c6] + Libffi_jll v3.2.2+1
  [d4300ac3] + Libgcrypt_jll v1.8.7+0
  [7e76a0d4] + Libglvnd_jll v1.3.0+3
  [7add5ba3] + Libgpg_error_jll v1.42.0+0
  [94ce4f54] + Libiconv_jll v1.16.1+1
  [4b2f31a3] + Libmount_jll v2.35.0+0
  [89763e89] + Libtiff_jll v4.4.0+0
  [38a345b3] + Libuuid_jll v2.36.0+0
  [856f044c] + MKL_jll v2022.1.0+0
  [e7412a2a] + Ogg_jll v1.3.5+1
  [458c3c95] + OpenSSL_jll v1.1.17+0
  [efe28fd5] + OpenSpecFun_jll v0.5.5+0
  [91d4177d] + Opus_jll v1.3.2+0
  [2f80f16e] + PCRE_jll v8.44.0+0
  [30392449] + Pixman_jll v0.40.1+0
  [ea2cea3b] + Qt5Base_jll v5.15.3+1
  [f50d1b31] + Rmath_jll v0.3.0+0
  [a2964d1f] + Wayland_jll v1.19.0+0
  [2381bf8a] + Wayland_protocols_jll v1.25.0+0
  [02c8fc9c] + XML2_jll v2.9.14+0
  [aed1982a] + XSLT_jll v1.1.34+0
  [4f6342f7] + Xorg_libX11_jll v1.6.9+4
  [0c0b7dd1] + Xorg_libXau_jll v1.0.9+4
  [935fb764] + Xorg_libXcursor_jll v1.2.0+4
  [a3789734] + Xorg_libXdmcp_jll v1.1.3+4
  [1082639a] + Xorg_libXext_jll v1.3.4+4
  [d091e8ba] + Xorg_libXfixes_jll v5.0.3+4
  [a51aa0fd] + Xorg_libXi_jll v1.7.10+4
  [d1454406] + Xorg_libXinerama_jll v1.1.4+4
  [ec84b674] + Xorg_libXrandr_jll v1.5.2+4
  [ea2f1a96] + Xorg_libXrender_jll v0.9.10+4
  [14d82f49] + Xorg_libpthread_stubs_jll v0.1.0+3
  [c7cfdc94] + Xorg_libxcb_jll v1.13.0+3
  [cc61e674] + Xorg_libxkbfile_jll v1.1.0+4
  [12413925] + Xorg_xcb_util_image_jll v0.4.0+1
  [2def613f] + Xorg_xcb_util_jll v0.4.0+1
  [975044d2] + Xorg_xcb_util_keysyms_jll v0.4.0+1
  [0d47668e] + Xorg_xcb_util_renderutil_jll v0.3.9+1
  [c22f9ab0] + Xorg_xcb_util_wm_jll v0.4.1+1
  [35661453] + Xorg_xkbcomp_jll v1.4.2+4
  [33bec58e] + Xorg_xkeyboard_config_jll v2.27.0+4
  [c5fb5394] + Xorg_xtrans_jll v1.4.0+3
  [3161d3a3] + Zstd_jll v1.5.2+0
  [a4ae2306] + libaom_jll v3.4.0+0
  [0ac62f75] + libass_jll v0.15.1+0
  [f638f0a6] + libfdk_aac_jll v2.0.2+0
  [b53b4c65] + libpng_jll v1.6.38+0
  [f27f6e37] + libvorbis_jll v1.3.7+1
  [1270edf5] + x264_jll v2021.5.5+0
  [dfaa095f] + x265_jll v3.5.0+0
  [d8fb68d0] + xkbcommon_jll v1.4.1+0
  [0dad84c5] + ArgTools v1.1.1
  [56f22d72] + Artifacts
  [2a0f44e3] + Base64
  [ade2ca70] + Dates
  [8bb1440f] + DelimitedFiles
  [8ba89e20] + Distributed
  [f43a241f] + Downloads v1.6.0
  [7b1f6079] + FileWatching
  [9fa8497b] + Future
  [b77e0a4c] + InteractiveUtils
  [4af54fe1] + LazyArtifacts
  [b27032c2] + LibCURL v0.6.3
  [76f85450] + LibGit2
  [8f399da3] + Libdl
  [37e2e46d] + LinearAlgebra
  [56ddb016] + Logging
  [d6f4376e] + Markdown
  [a63ad114] + Mmap
  [ca575930] + NetworkOptions v1.2.0
  [44cfe95a] + Pkg v1.8.0
  [de0858da] + Printf
  [3fa0cd96] + REPL
  [9a3f8284] + Random
  [ea8e919c] + SHA v0.7.0
  [9e88b42a] + Serialization
  [1a1011a3] + SharedArrays
  [6462fe0b] + Sockets
  [2f01184e] + SparseArrays
  [10745b16] + Statistics
  [4607b0f0] + SuiteSparse
  [fa267f1f] + TOML v1.0.0
  [a4e569a6] + Tar v1.10.0
  [8dfed614] + Test
  [cf7118a7] + UUIDs
  [4ec0a83e] + Unicode
  [e66e0078] + CompilerSupportLibraries_jll v0.5.2+0
  [deac9b47] + LibCURL_jll v7.84.0+0
  [29816b5a] + LibSSH2_jll v1.10.2+0
  [c8ffd9c3] + MbedTLS_jll v2.28.0+0
  [14a3606d] + MozillaCACerts_jll v2022.2.1
  [4536629a] + OpenBLAS_jll v0.3.20+0
  [05823500] + OpenLibm_jll v0.8.1+0
  [83775a58] + Zlib_jll v1.2.12+3
  [8e850b90] + libblastrampoline_jll v5.1.1+0
  [8e850ede] + nghttp2_jll v1.48.0+0
  [3f19e933] + p7zip_jll v17.4.0+0
        Info Packages marked with ⌅ have new versions available but cannot be upgraded. To see why use `status --outd
ated -m`
```

In [2]:
```
# load external packages that are required for the calculations -
using DataFrames
using CSV
using Dates
using Statistics
using LinearAlgebra
using Plots
using Colors
using Distributions
using StatsPlots

# setup paths to load XYZ OHLC data set -
const _NOTEBOOK_ROOT = pwd();
const _PATH_TO_DATA = joinpath(_NOTEBOOK_ROOT, "data");
```

In [3]:
```
include("CHEME-5560-AP2-CodeLib.jl"); # Look inside me to find out what I have!
```

In [4]:
```
# load the OHLC data set -
df = CSV.read(joinpath(_PATH_TO_DATA, "CHEME-5660-OHLC-XYZ-AP2-F22.csv"), DataFrame);
```

In [5]:
```
# setup constants -
ΔT = 78.0;              # days to expiration (units: days)
B = 365.0;              # number of days per year
IV = 52.27;             # implied volatility
S₀ = 270.89;            # initial share price given in the problem

# plot and sim constants stuff for later -
number_of_bins = 80;
number_of_sample_paths = 10000;
```

Split the historical data into training and prediction sets

In [6]:
```
α = 0.80; # fraction of data for training (you get to choose this)
```

In [40]:
```
# instead of using all the data from 1 - let's specify a start index -
start_index = 280;#280; # you also get to choose this! (1 -> all data) #default: 280
df_local = df[start_index:end, :]

N = nrow(df_local); # this is the number of rows in the total data set -
ℒ = Int64(round(α*N));

# split the data into to two chunks, training and validation
all_range = range(1,stop=N,step=1) |> collect
T_all = all_range*(1.0/365.0) .- (1.0/365.0)

# time ranges for the training, and prediction sets
training_range = range(1,stop=ℒ, step=1);
prediction_range = range(ℒ+1,stop=N, step=1);

# data sets -
df_training = df_local[training_range,:];
df_prediction = df_local[prediction_range,:];
```

In [41]:
```
# build an empty model, add stuff to it -
training_model = GeometricBrownianMotionModel()
training_model.T₁ = 0.0
training_model.T₂ = 0.0;
training_model.h = (1.0/365.0)
training_model.X₀ = df_local[1,:close]; # we can change where we start

# parameter values -
training_model.μ = 0.0; # we don't know this value yet, 0 for now
training_model.σ = 0.0; # we don't know this value yet, 0 for now
```

a) Estimate the rate of return parameter μ

Strategy

Let A denote the $S\times 2$ matrix holding the time values; the first column of A is all 1's while the second column holds the $(t-t_{\circ})$ values. Further, let Y denote the QQQ close price values (in the same order as the A matrix). Then, the y-intercept and slope can be estimated by solving the `overdetermined` system of equations:

$$A\theta = Y$$

where θ denotes a 2×1 vector of unknown parameters; the first element is the y-intercept $b = \ln S_{\circ}$ while the second element is $\hat{\mu}$, an estimate of the growth-rate parameter. This system can be solved as:

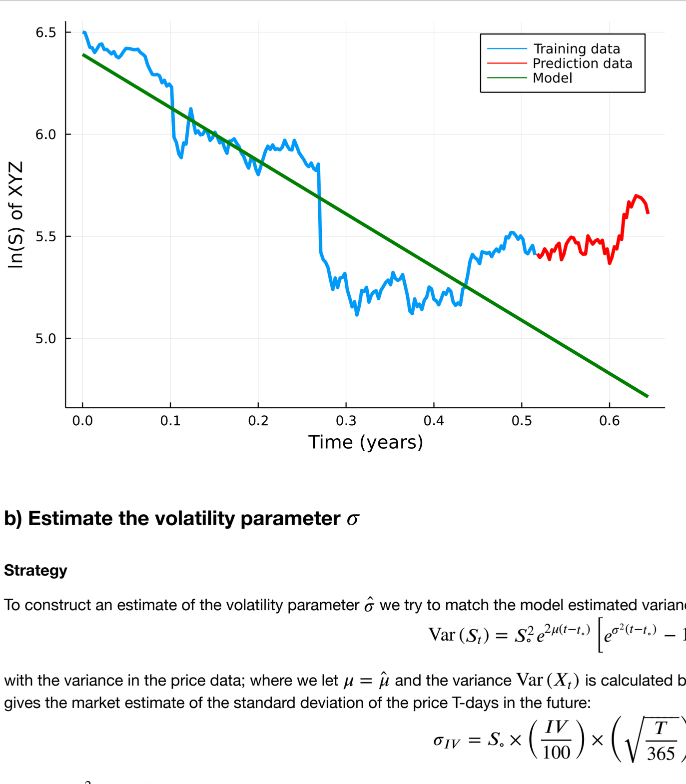
$$\theta = (A^{T}A)^{-1}A^{T}Y$$

where A^{T} denotes the transpose of the matrix A.

In [42]:
```
# Setup the normal equations -
XD = [ones(Int64(round(α*N))) T_all[training_range]];
[illegible]
```

[illegible]

Out[44]:



b) Estimate the volatility parameter σ

Strategy

[illegible]

where:

$$T' = \frac{1}{365}(T - T_{\circ})$$

In [47]:
```
# estimate the volatility σ̂ -
S₀ = df_local[1,:close]; # initial share price
N_local = length(df_local[1,:close]);
Var_market = (S₀*(IV/100.0)*sqrt(N_local/365))^2 # variance of price using the IV -
T_market = (N_local/365.0)
a = Var_market/((S₀^2)*exp(2*μ̂*T_market)) + 1
σ̂ = sqrt((1/T_market)*log(a))
```

Out[47]: `0.5263454316968543`

c) Predict share price distribution of `XYZ` T = 78 days into the future

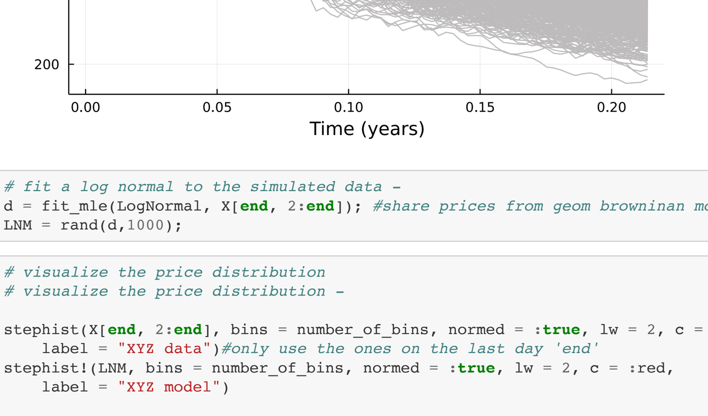
In [56]:
```
# build new prediction model -
future_prediction_model = GeometricBrownianMotionModel()
future_prediction_model.T₁ = 0.0
future_prediction_model.T₂ = (ΔT/B);
future_prediction_model.h = (1.0/365.0)
future_prediction_model.X₀ = S₀; # share price from the problem

# parameter values -
future_prediction_model.μ = μ̂; # use the estimated value -
future_prediction_model.σ = σ̂; # use the estimated value -
```

In [57]:
```
# run simulation using the solve function in the code library
future_prediction = solve(future_prediction_model; 𝒫 = 10000);
```

In [58]:
```
T = future_prediction[:,1];
X = future_prediction[:,2:end];
```

In [59]:
```
# check: it would awesome to see the what those simulation paths look like!
skip_factor = 10
plot(T, X[:,1:skip_factor:end], label="", c=colorant"#BDBBBB")
xlabel!("Time (years)", fontsize=18);
ylabel!("Share price of XYZ", fontsize=18)
```

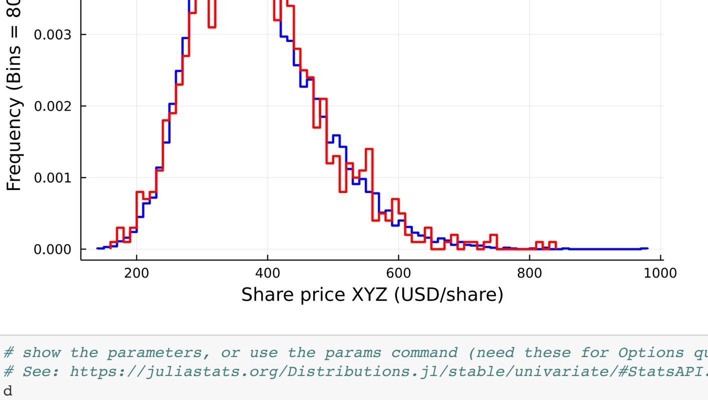
Out[59]:



In [60]:
```
# fit a log normal to the simulated data -
d = fit_mle(LogNormal, X[end, 2:end]); #share prices from geom browninan motion are lognormal distrubuted!
LNM = rand(d,1000);
```

In [61]:
```
# visualize the price distribution
# visualize the price distribution -

stephist(X[end, 2:end], bins = number_of_bins, normed = :true, lw = 2, c = :blue,
    label = "XYZ data")#only use the ones on the last day 'end'
stephist!(LNM, bins = number_of_bins, normed = :true, lw = 2, c = :red,
    label = "XYZ model")

xlabel!("Share price XYZ (USD/share)", fontsize=18)
ylabel!("Frequency (Bins = $(number_of_bins)) (AU)", fontsize=18)
```

Out[61]:



In [63]:
```
# show the parameters, or use the params command (need these for Options question)
# See: https://juliastats.org/Distributions.jl/stable/univariate/#StatsAPI.params-Tuple{UnivariateDistribution}
d
```

Out[63]: `LogNormal{Float64}(μ=5.912192363959601, σ=0.24192452652479932)`

In [ ]:

In [ ]: