**Week 3 Writeup**

# Progress

**Week 1 – COMPLETED**

1. Import data from PyMongoDB into Pandas DataFrame format
2. Separate data by edge id
3. Gaussian models – Frequentist & Bayesian
4. Started Coursera course on Bayesian Inference in Python

**Week 2 – COMPLETED**

1. Multi-parameter Bayesian optimisation for Gaussian model
   1. What is the extra curve?
2. Goodness-of-fit (GOF) tests (KS, AD, CVM, MAE, MSE, AIC, BIC)
3. Compare data from different edges (e.g WayPoint70\_WayPoint71 or WayPoint69\_WayPoint70)
   1. Direction of the edge DOES matter
4. GOF tests against all distributions in Scipy for one edge (with most data)

**Week 3 – TO DO**

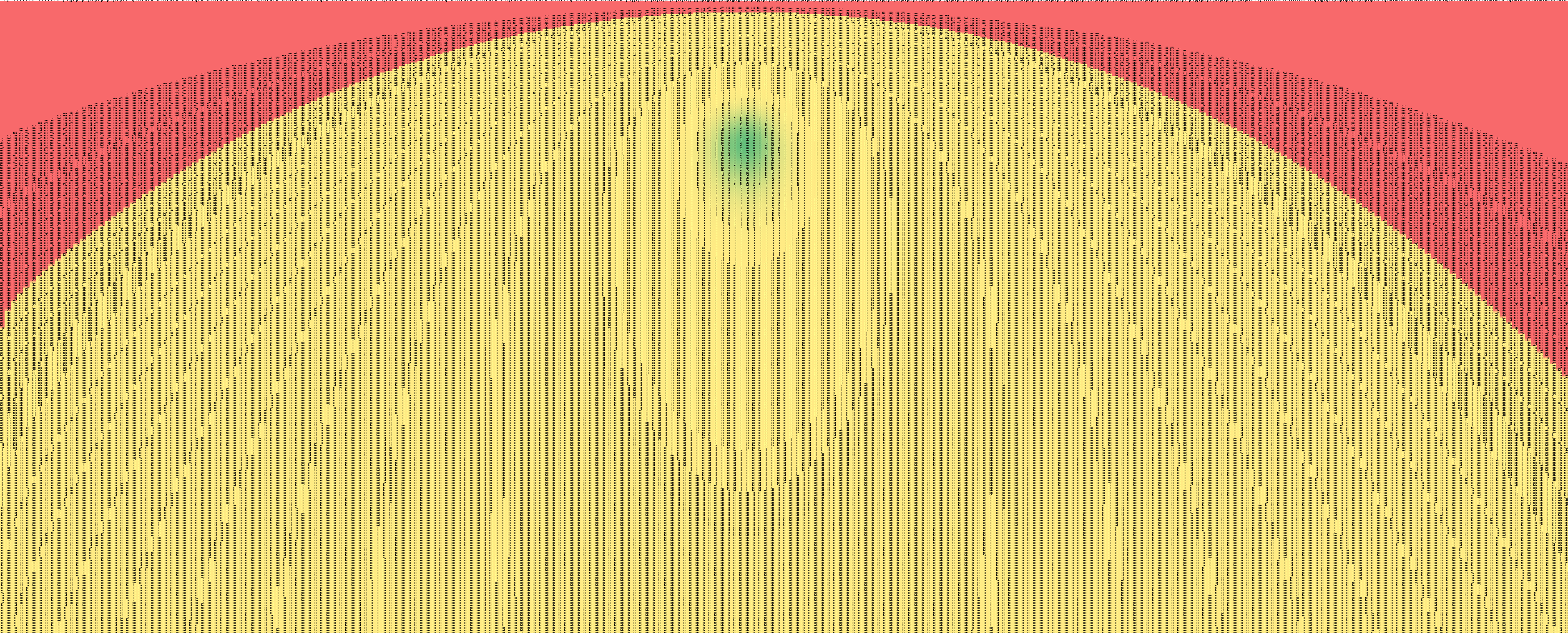
1. Filter out multi-robot data
2. GOF tests against all distributions in Scipy for multiple edges
3. Compare GOF tests
4. Choose distributions with high GOF and convenient properties
5. Multiparameter Bayesian updates for chosen distribution
   1. Online updates
   2. Probabilistic forecasting against test data for thresholding (i.e. determining when to stop subsequent updates)
6. Network visualisation

**NOTE:**

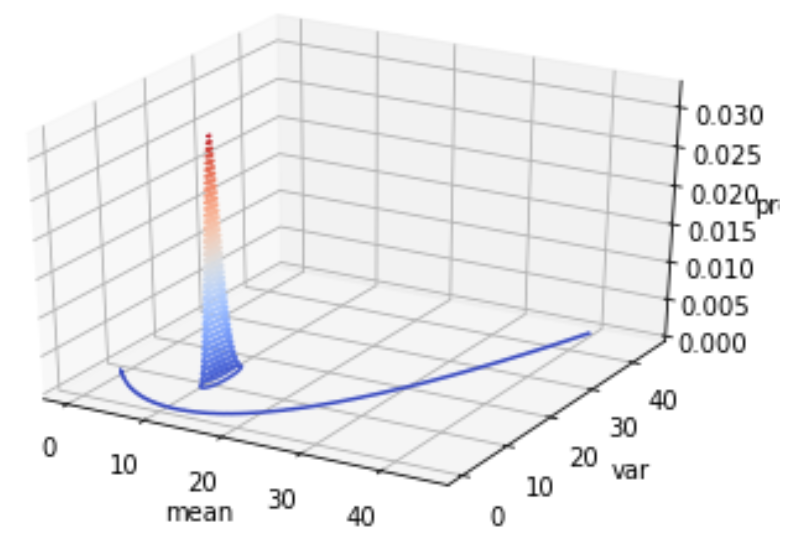
I chose to use the walmart\_random dataset in order to minimise the effect of congestion, by using the larger map and random origin-target pairings.

# Findings

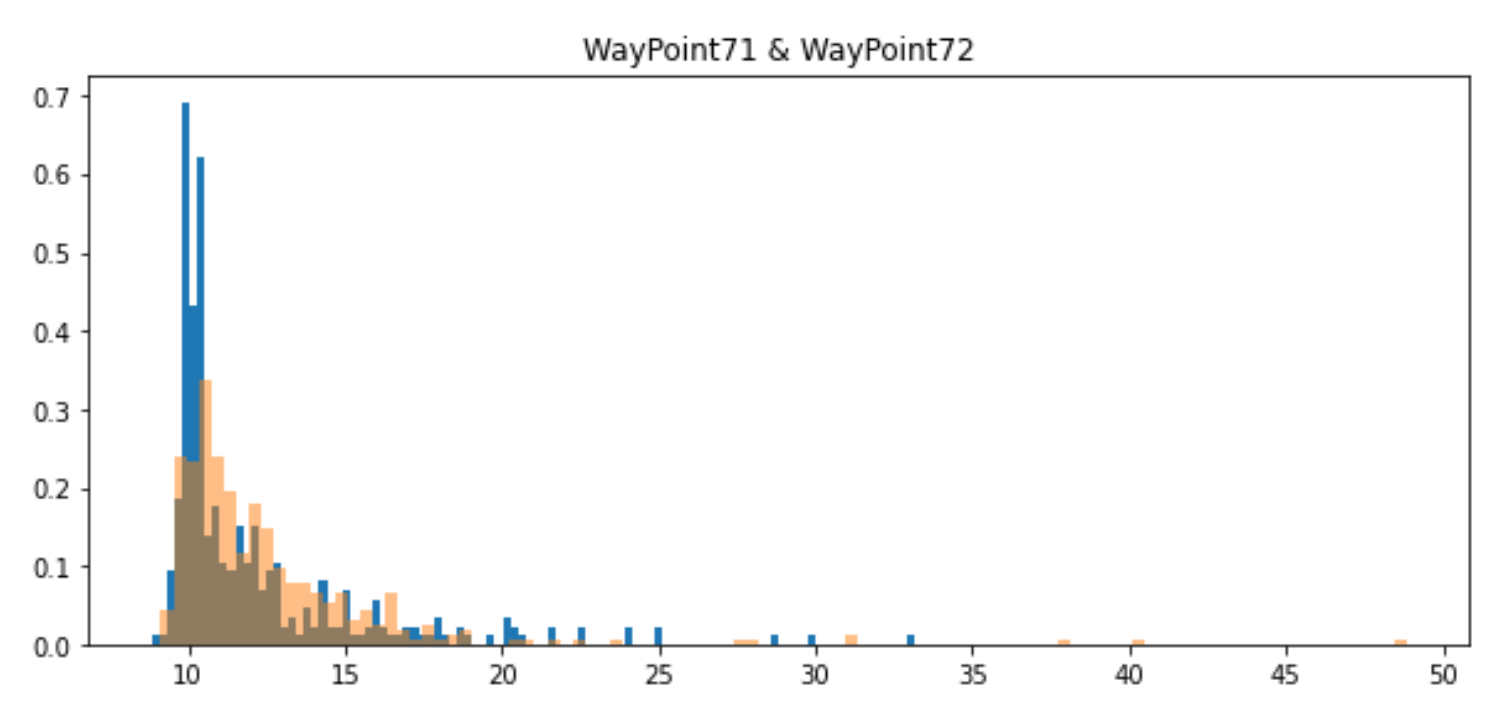
1. Was the multiparameter optimisation (from Wk2/MAIN2.ipynb) unimodal?
   1. YES
   2. The extra curve is a lowest-valued contour line
   3. Look at Wk2/3Dplot\_dec.xlsx (below). This shows the posterior data is unimodal



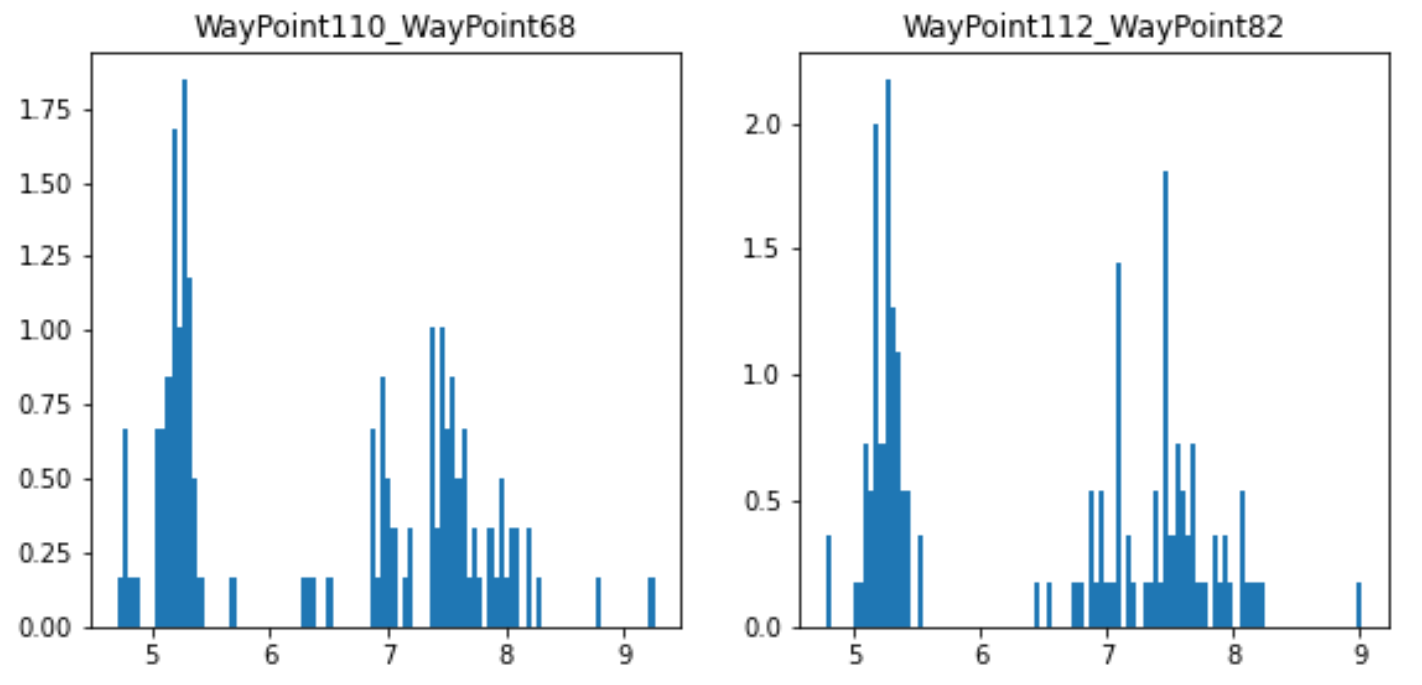
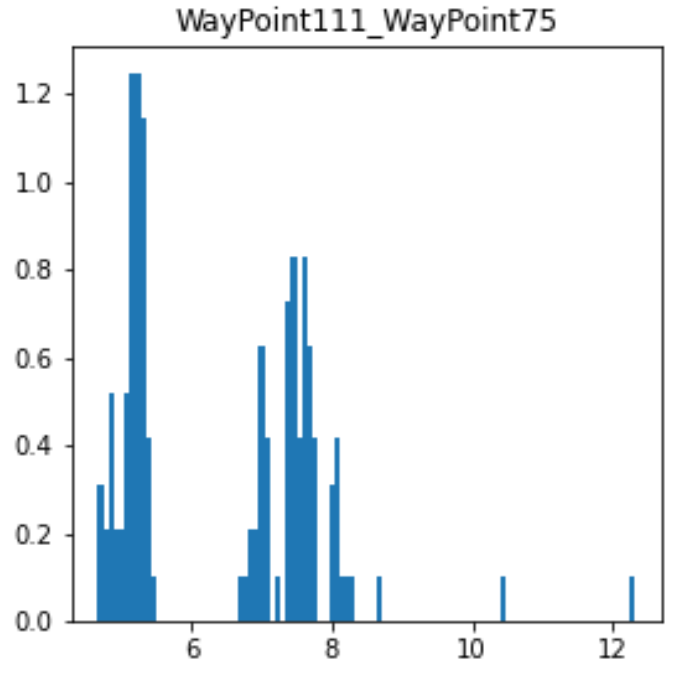
For clarity, the following was the 3D pyplot contour map for Bayesian optimisation of the mean and variance of the normal distribution:



1. The direction of edges DOES affect the duration distribution (Wk2/EDGES.ipynb)
   1. i.e. Waypoint71\_ Waypoint72 is different to Waypoint72\_ Waypoint71
   2. Only 20 out of 81 edges have a sample that is drawn from same duration distribution as the same edge in the opposite direction for (verified by 2-sample KS test with critical p-value of 5%)

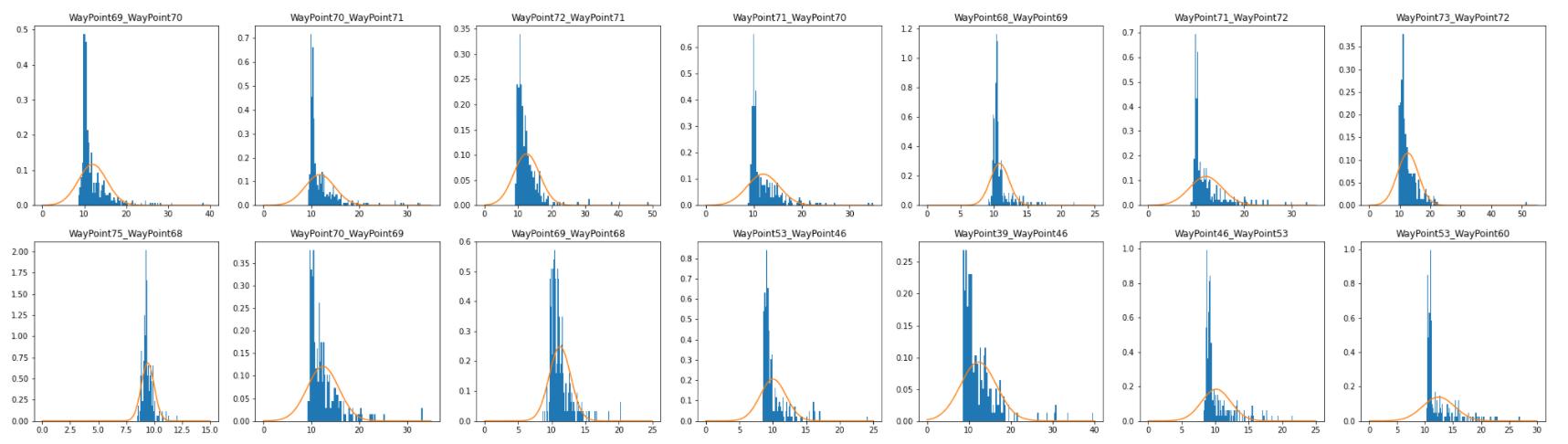


1. There are several edges that have multi-modal distributions (Wk2/EDGES.ipynb)
   1. WayPoint110\_WayPoint68, WayPoint112\_WayPoint82, WayPoint111\_WayPoint75
   2. More analysis is needed to determine whether removing multi-robot edges (i.e. transitions where more than one robot is traversing the same edge) will remove the multi-modal distributions

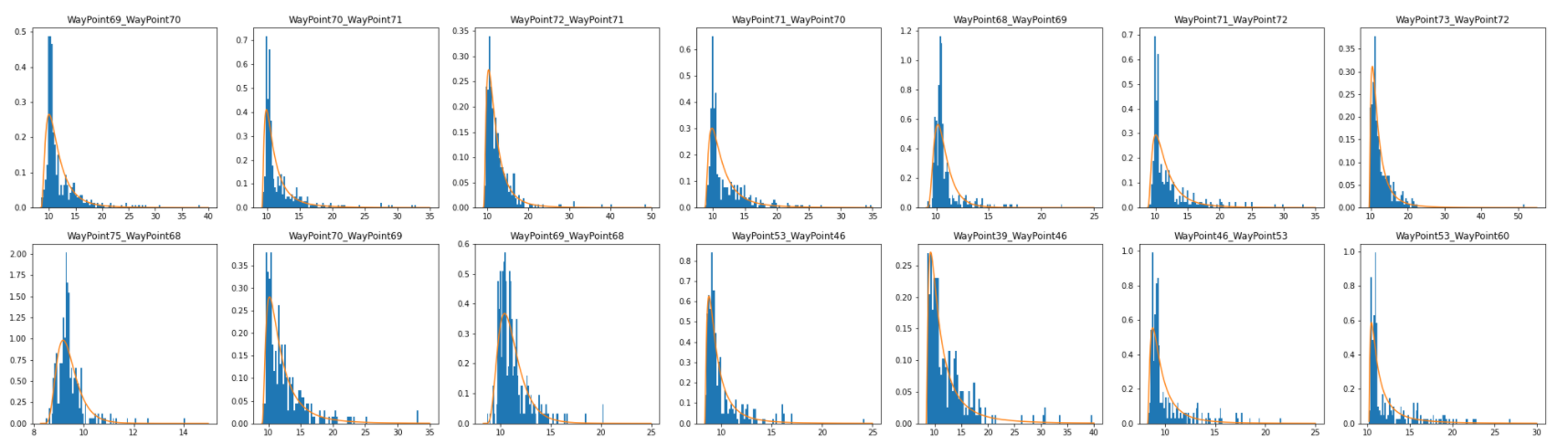
 

1. Gaussian is a poor fit. Lognormal is much better (Wk2/EDGES.ipynb)

The following is a Gaussian fit (next page):

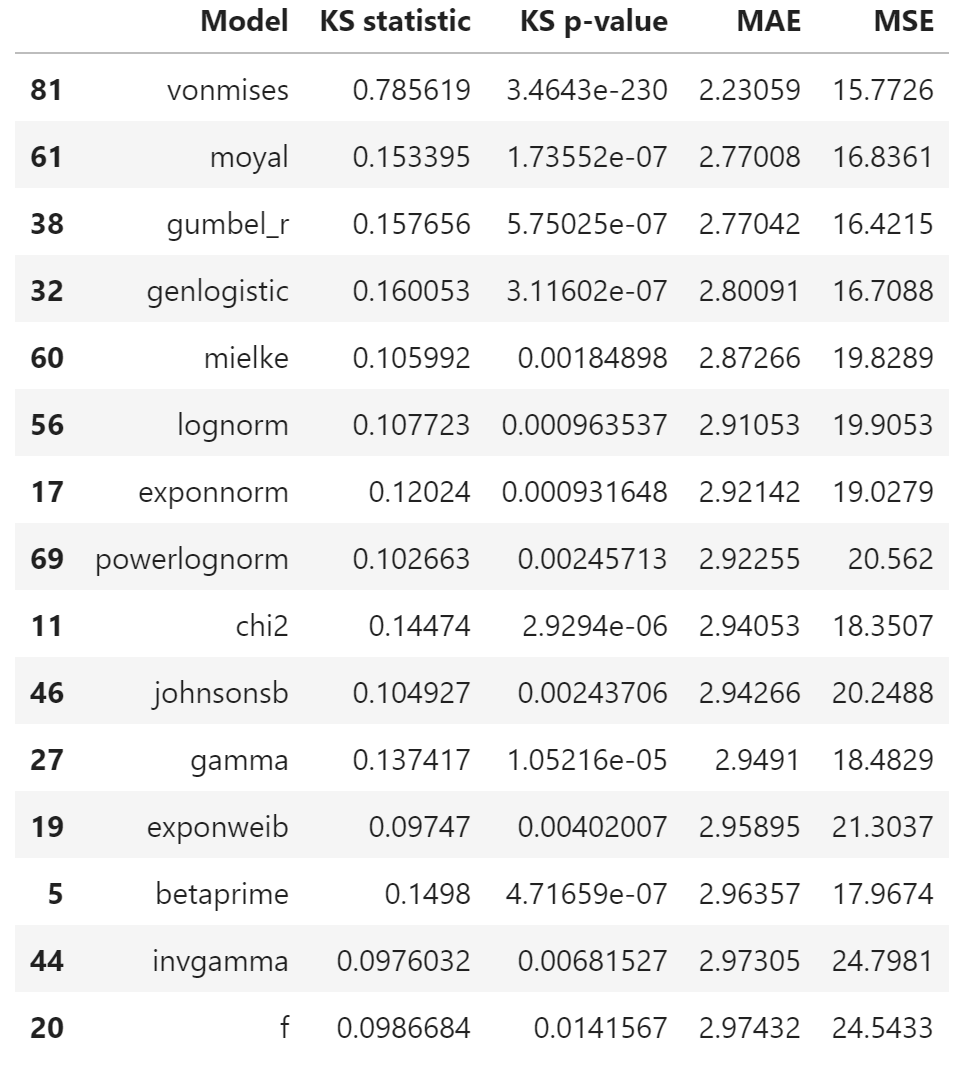
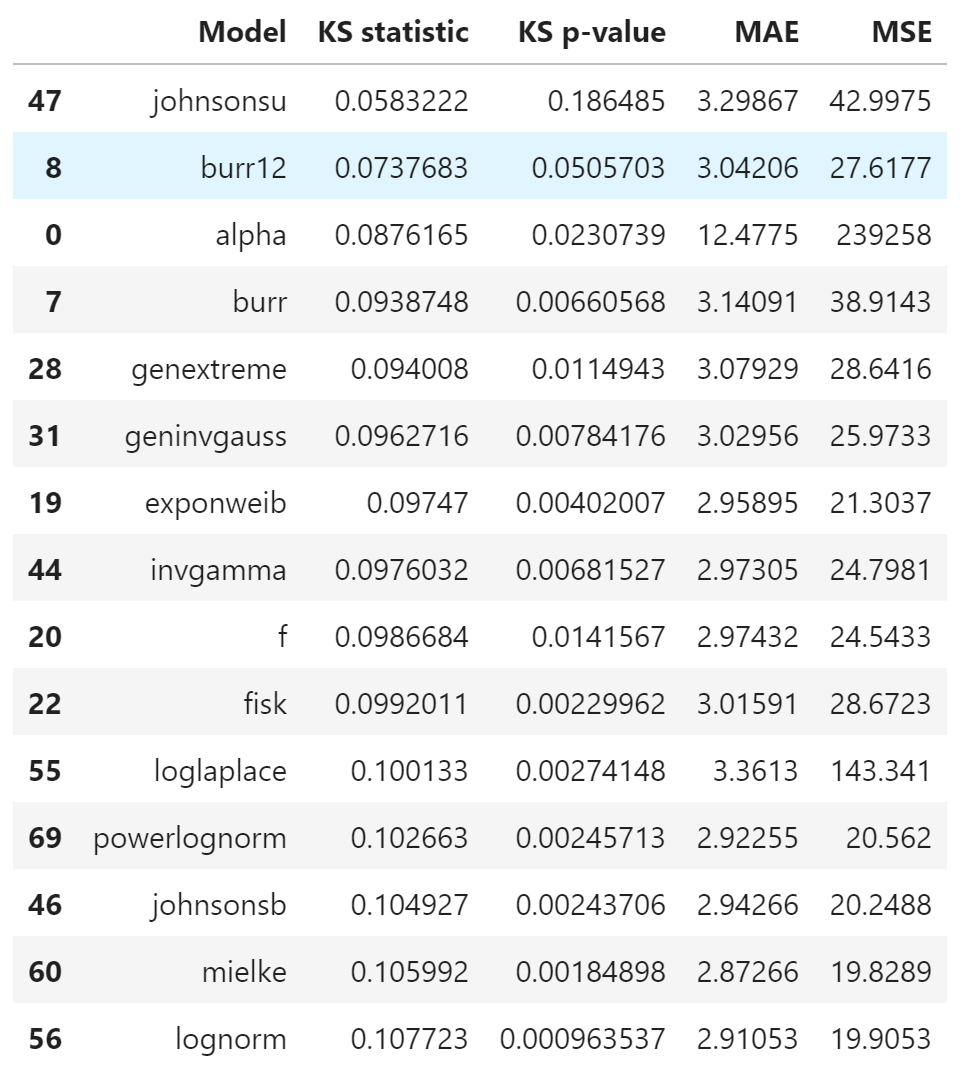


The following is a lognormal fit:



1. The following Scipy distributions have high GOF score (DISTRIBUTIONS.ipynb)
   1. exponweib (7,7)
   2. invgamma (14,8)
   3. f (15,9)
   4. powerlognorm (8,12)
   5. mielke (5,14)
   6. lognorm (6,15)

The left-hand table is ordered by KS statistic (low is good). The right-hand table is ordered by Mean Absolute Error MAE (low is good).

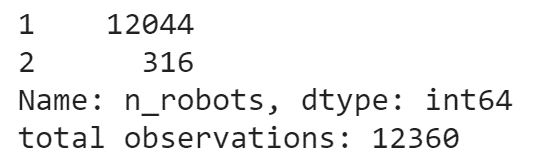


# Filtering out the multi-robot data

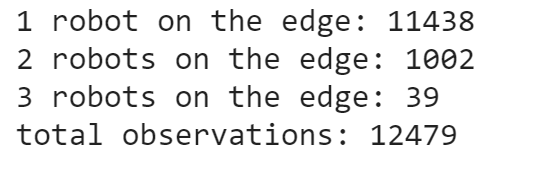
To make it easier to generalise, I have removed the following effects from the dataset:

* Having more than one robot traverse an edge during the same period can lead to problems with congestion.
* If the robot is moving towards its final destination node, it will slow down so that it can stop at that node.

Results from recording how many robots (in total) are present on the same edge during each observation:

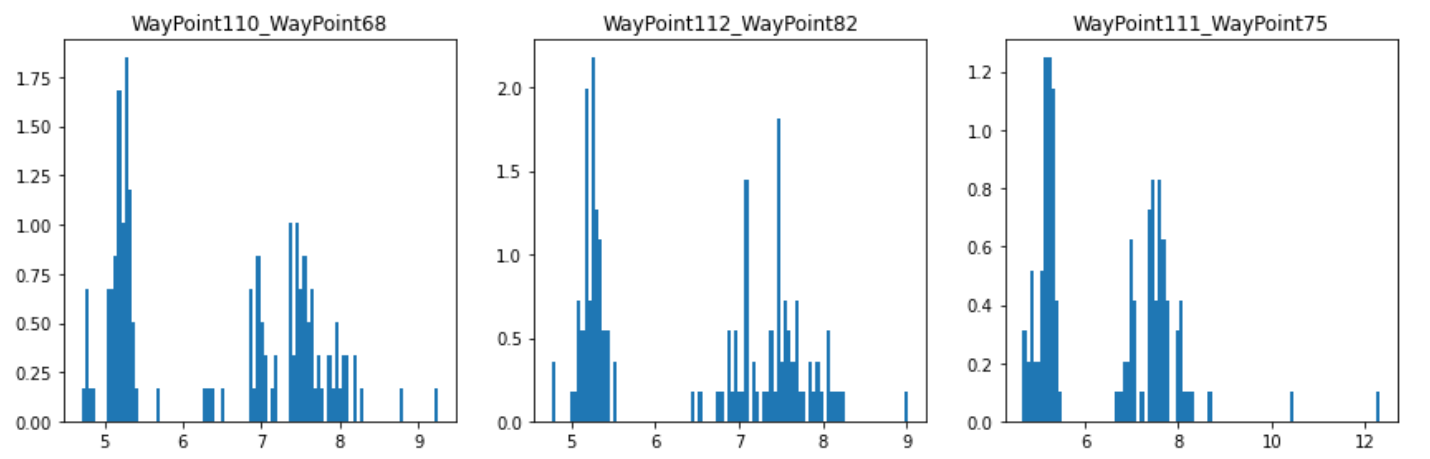


Compare with the YAML scalar context file (walmart\_random\_success.yaml):



The main difference is that there are a lot more instances in the YAML dataset that have 2 or more robots on the same edge at the same time. This may be because this dataset used more complex criteria for determining whether a robot will interfere with the operations of another robot.

After filtering our (non-YAML) dataset to remove transitions, we find that the multimodal distributions are unchanged:

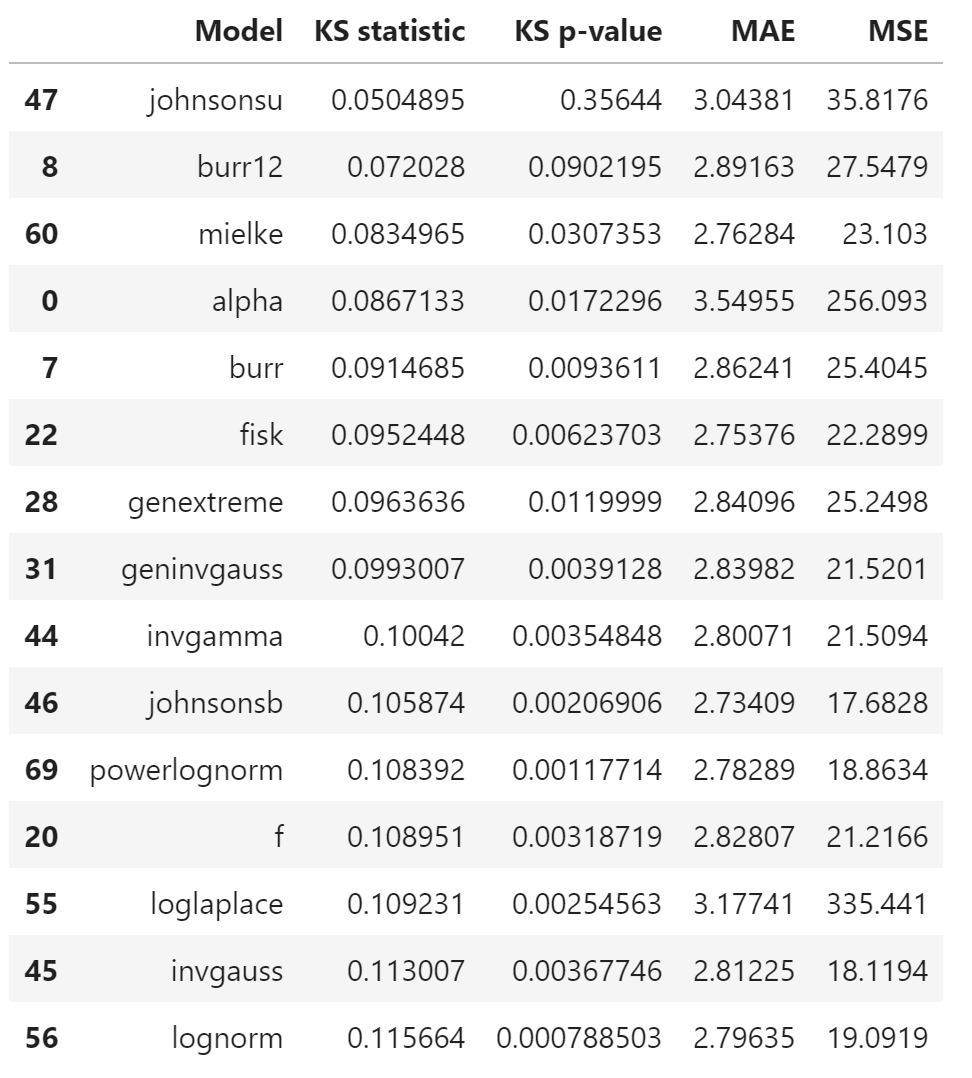
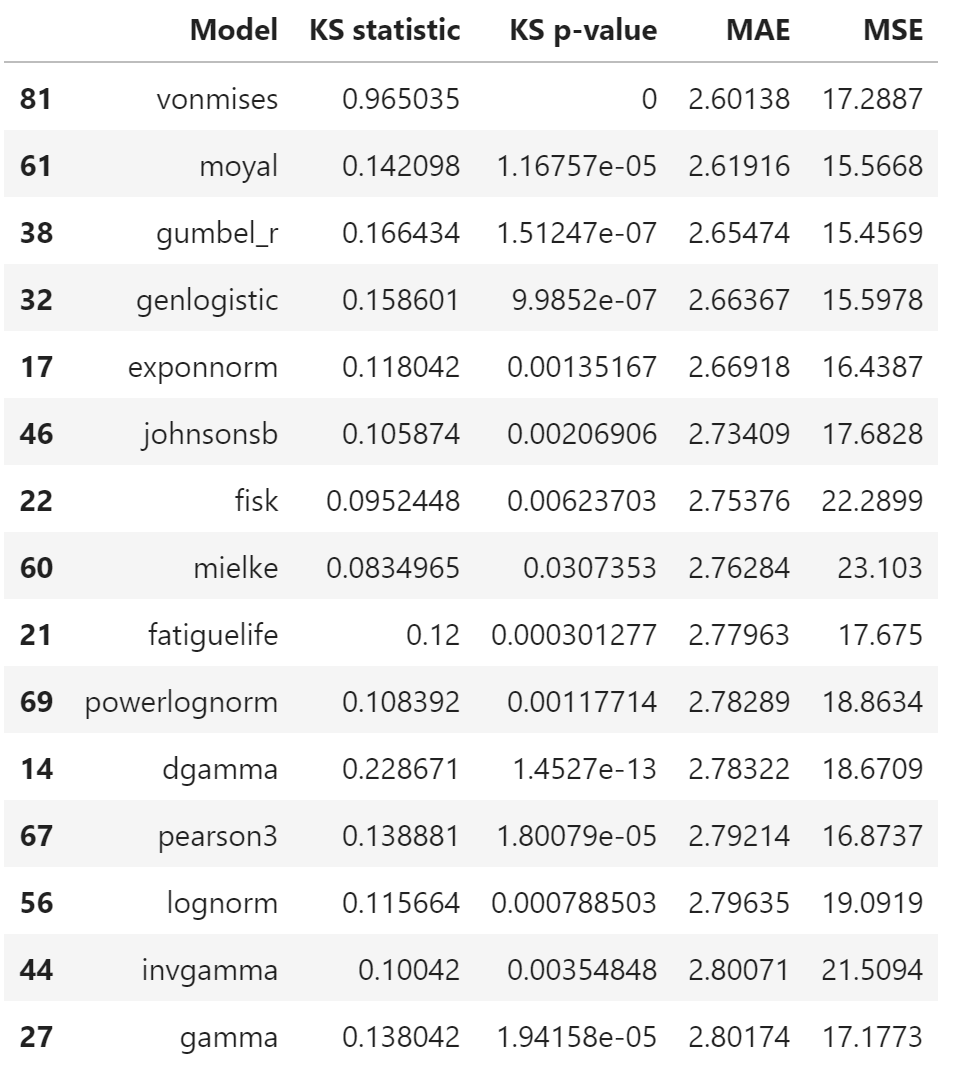


This could be due to the structure of these nodes, meaning that there is a chance to fail when traversing these edges on the first try. We can check this later with network visualisation (or simply reading how many connections each node has).

The filtering also made changes to the best-fitting distributions. Distributions that appear on both lists include:

* Mielke (3,8)
* Fisk (6,7)
* Invgamma (8,14)
* Johnsonsb (9,6)
* Powerlognorm (10,9)
* Lognorm (15,13)

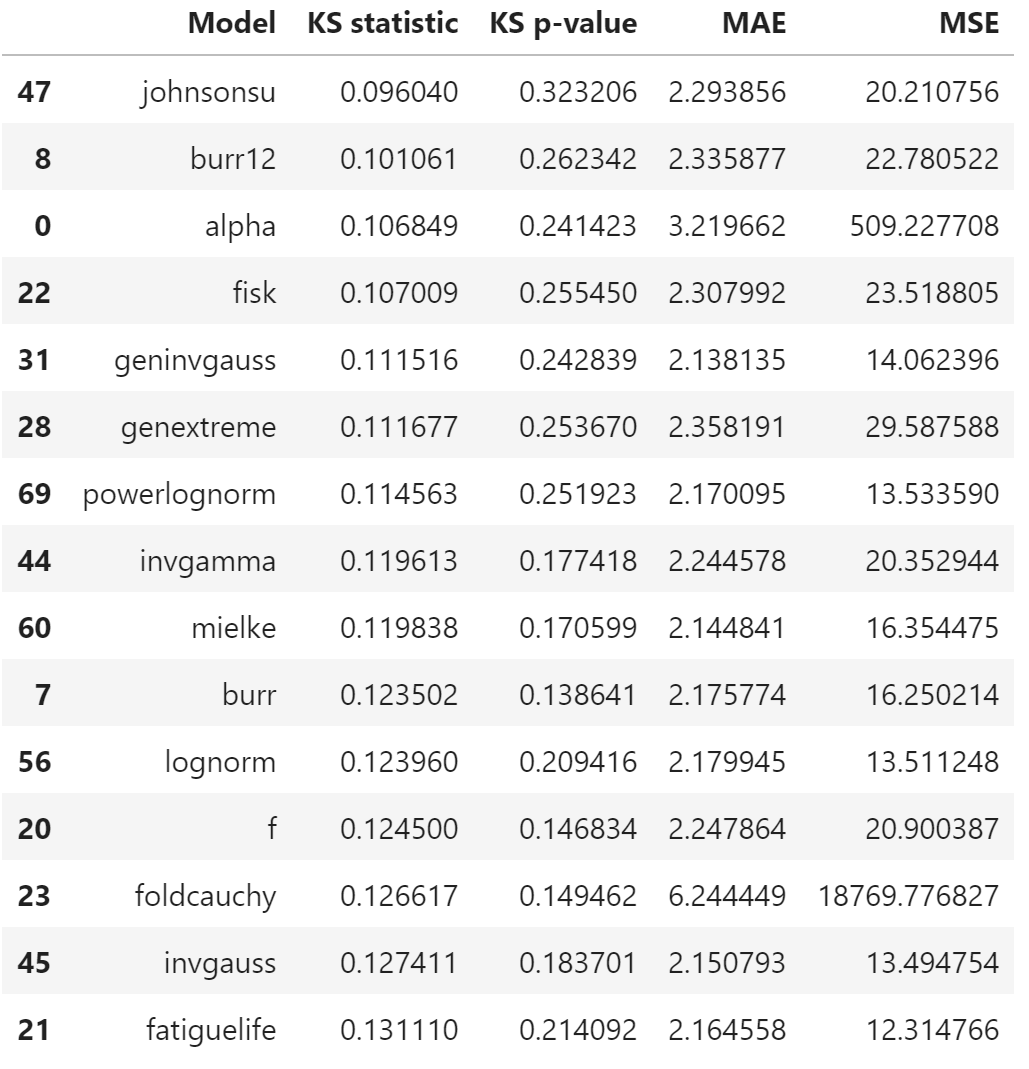
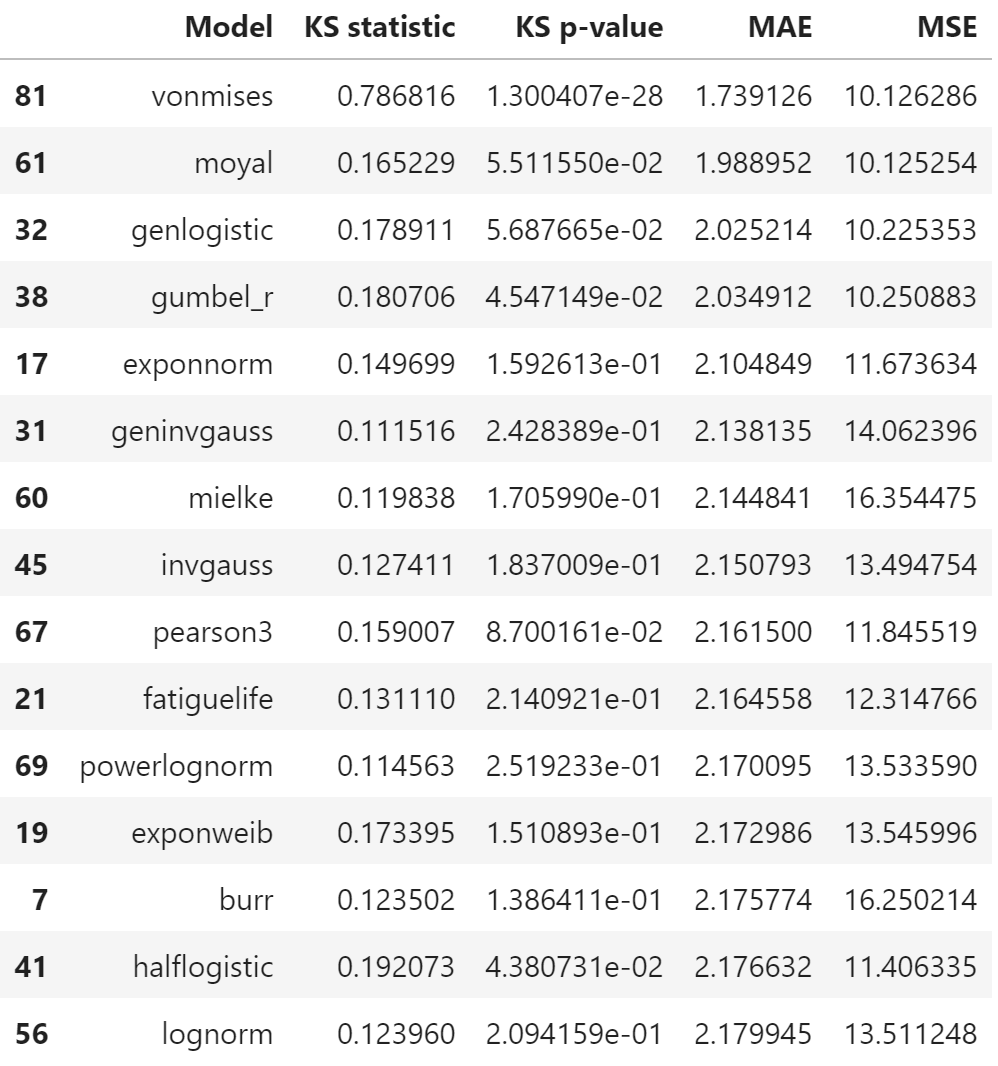
Note, this is all for one edge (“WayPoint69\_WayPoint70”)

# Multiple Edges + All Scipy Models

When using data for the 25 edges with most observations, the best fitting distributions are:

* Geninvgauss
* Powerlognorm
* Invgauss
* Mielke
* Burr
* Lognorm
* Fatiguelife

Reasons for KS test:

1. Visualisation (compared to CVM / AD)
2. Quick to calculate

Reasons for MAE/MSE:

1. Approximation to CVM

# Promising Distributions

1. Lognorm / Powerlognorm
2. Invgauss / Geninvgauss
3. Mielke
4. Burr
5. Fatiguelife
6. Invgamma

Look for conjugate priors that allow for optimisation over all parameters

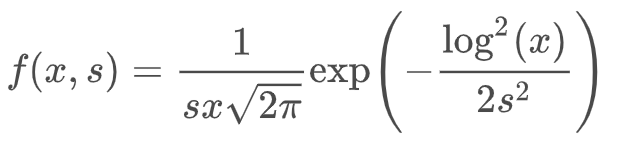
Mixture models?

# Lognormal

#### Common Uses

If X is lognormal, then lnX is normal

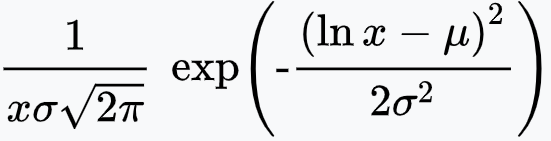
#### Scipy parameterisation:



Parameters are **s, loc, scale**

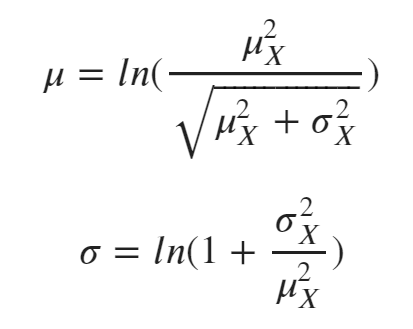
* S = sigma(X = lnY)
* Scale = exp(mu(X = lnY)
* **lognorm.pdf(x, s, loc, scale)** is identically equivalent to **lognorm.pdf(y, s) / scale** with **y = (x - loc) / scale**

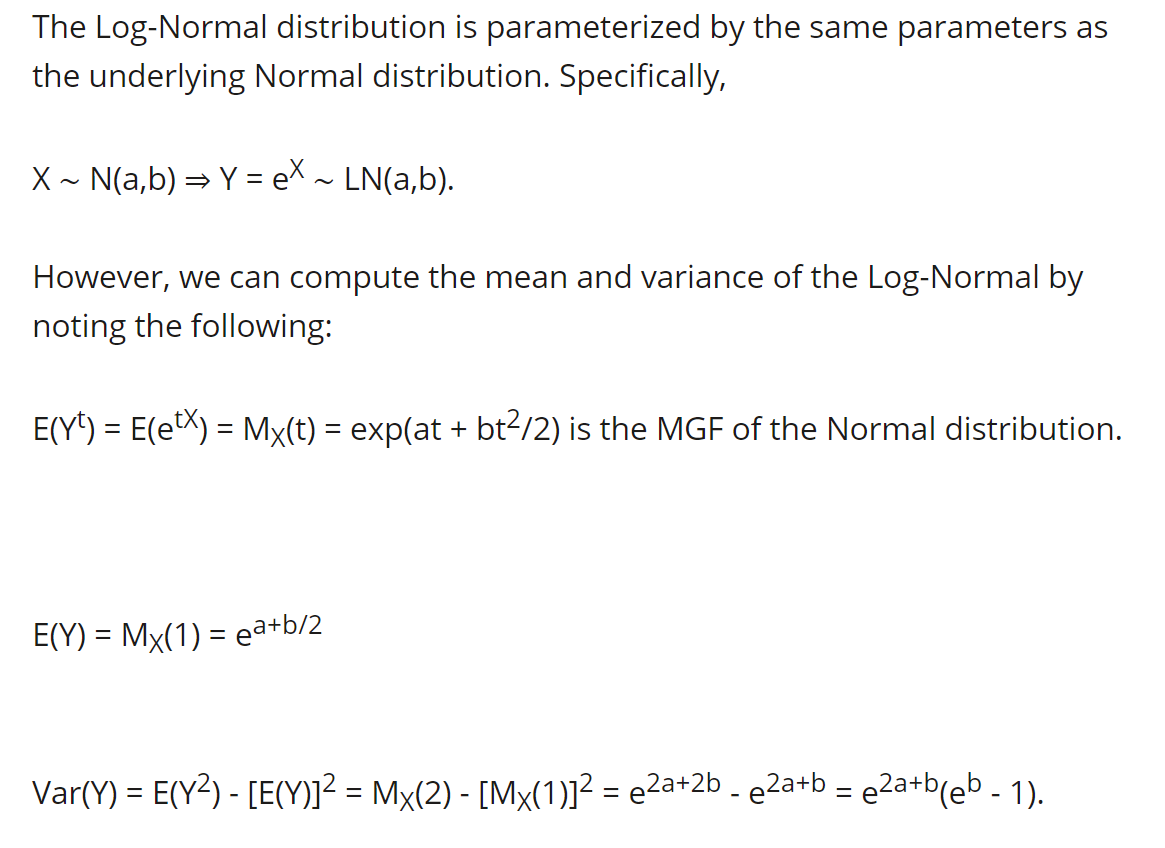
#### Wikipedia/Coursera parameterisation:



Parameters are **mu, sigma**

* Mu, sigma are NOT the actual mean/std of the lognormal distribution

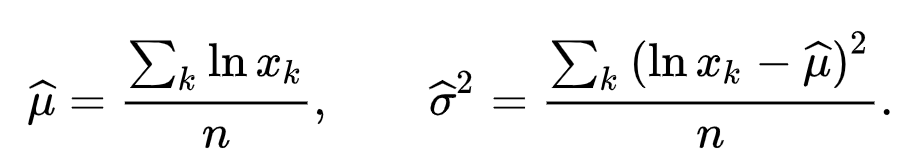
 where mu\_x, sigma\_x are the mean and std of the lognormal distribution



From [Converting Normal distribution to Lognormal | Wyzant Ask An Expert](https://www.wyzant.com/resources/answers/231349/converting_normal_distribution_to_lognormal)

Also see MIT brief notes in Lognormal\_Normal\_MeanVariance.pdf

#### MLE estimators:



##### Conjugate priors

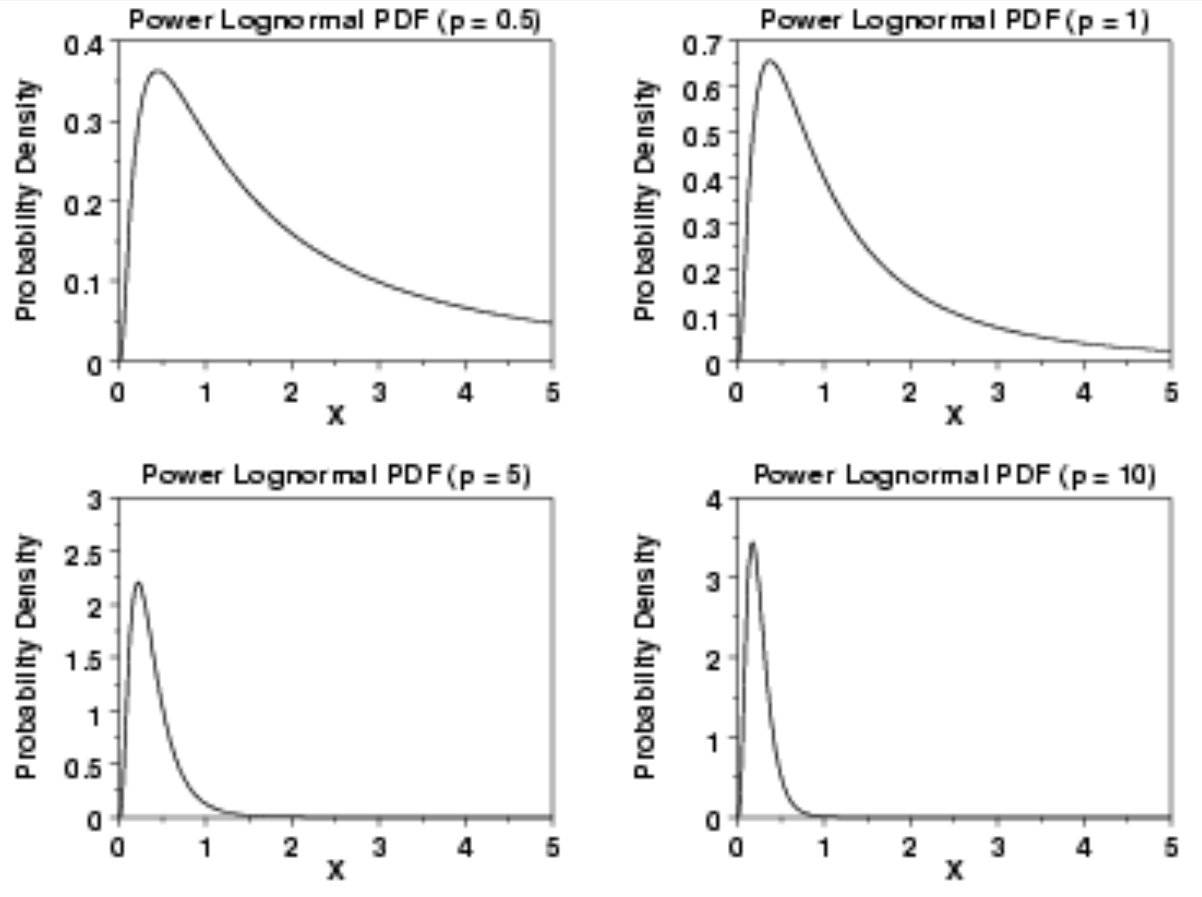
Same as for the normal distribution after applying the natural logarithm to the data for the posterior hyperparameters.

Norma distribution has normal-gamma conjugate prior for optimising both mean & variance

# Powerlognorm – not so useful

#### Common uses

#### Scipy parameterisation



[1.3.6.6.14. Power Lognormal Distribution (nist.gov)](https://www.itl.nist.gov/div898/handbook/eda/section3/eda366e.htm)

#### MLE

Software for maximum likelihood estimation of the parameters of the power lognormal distribution is not as readily available as for other reliability distributions such as the exponential, Weibull, and lognormal. [1.3.6.6.14. Power Lognormal Distribution (nist.gov)](https://www.itl.nist.gov/div898/handbook/eda/section3/eda366e.htm)

#### Conjugate priors

???

Couldn’t find any

# Invgauss (Wald)

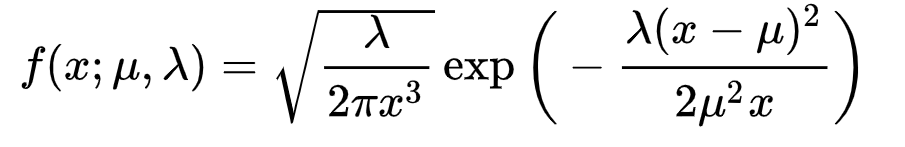
#### Common uses

Gaussian describes a Brownian motion's level at a fixed time.

Inverse Gaussian describes the distribution of the time a Brownian motion with positive drift takes to reach a fixed positive level. Also describes

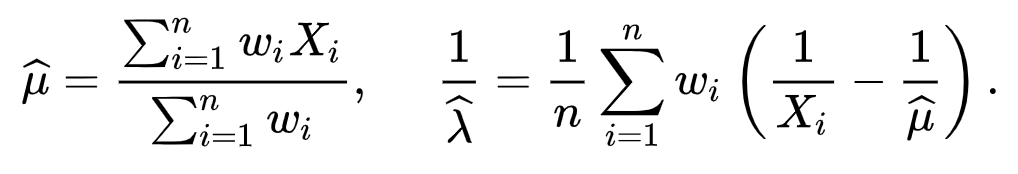
* the time a stock reaches a certain price for the first time

#### Wikipedia/Scipy parameterisation



Parameters are mu (mean), lambda (shape)

#### MLE

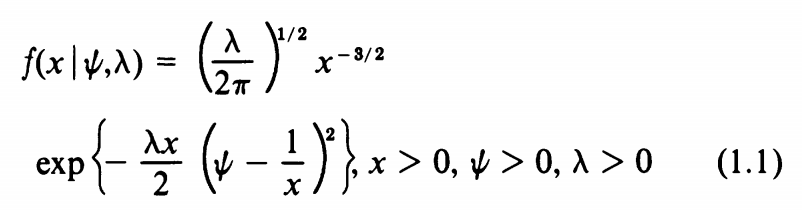


Set wi = 1

#### Conjugate priors

Uses Banerjee\_InverseGaussian paper

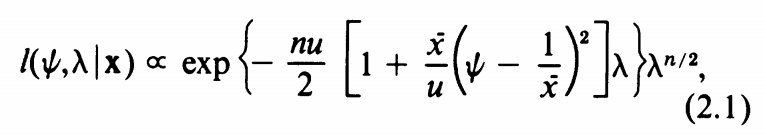
**PDF:**

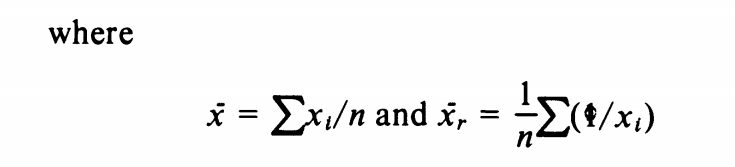


**Alternative parameterisation**

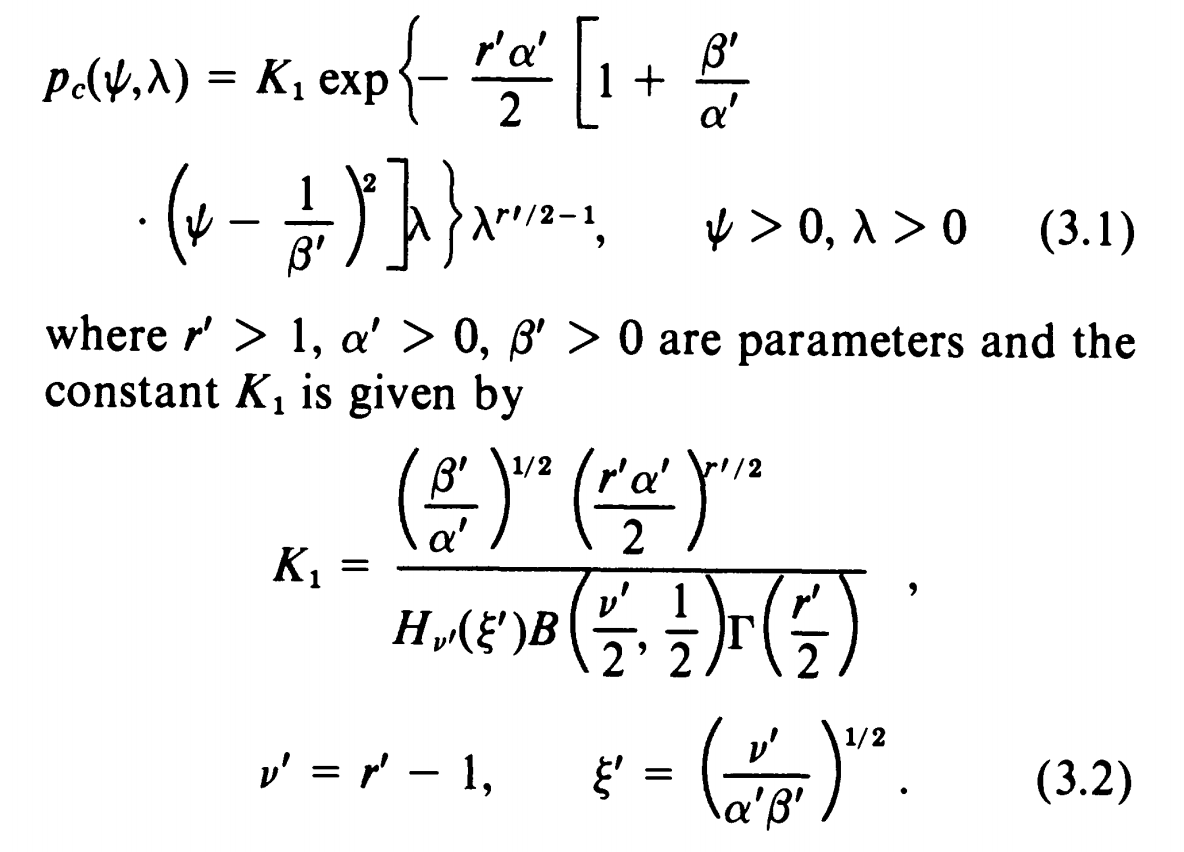
* Ψ = 1/μ
* λ = δ2/ψ = δ2μ = (σ/μ)2μ = var/mu

**Likelihood function:**

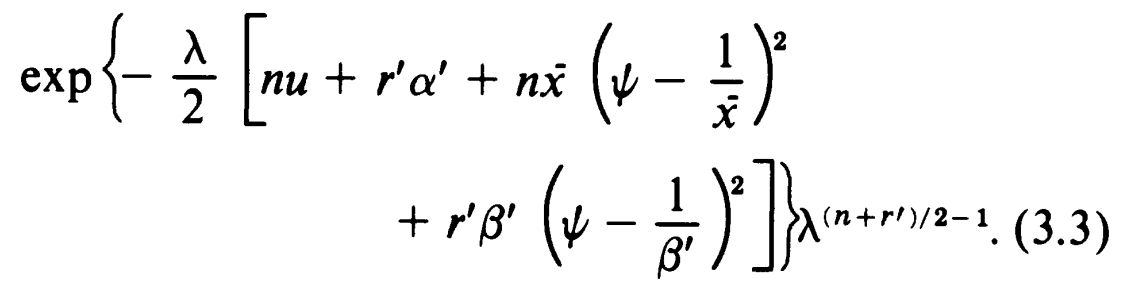


 are the mean of sample & mean of reciprocals of sample

**Conjugate prior**



Conjugate posterior is proportional to

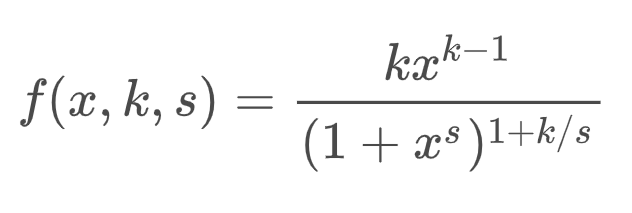


# Mielke (Mielke Beta-Kappa / Dagum / Burr type III) - unsuitable

#### Common uses

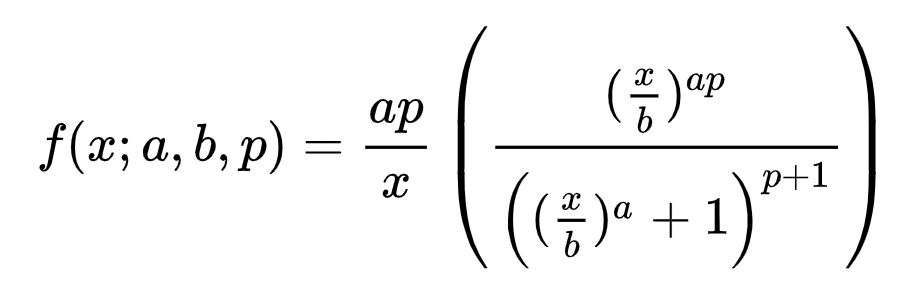
Models income & wealth distributions

#### Scipy Parameterisation



Parameters are **k** & **s** > 0

#### Wikipedia parameterisation

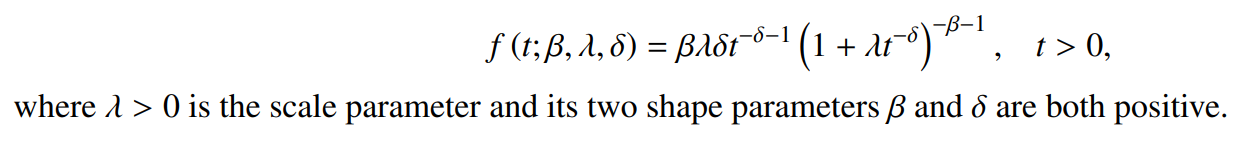


Parameters are a, b, p

* k 🡺 ap = sp
* s = a
* x 🡺 x/b

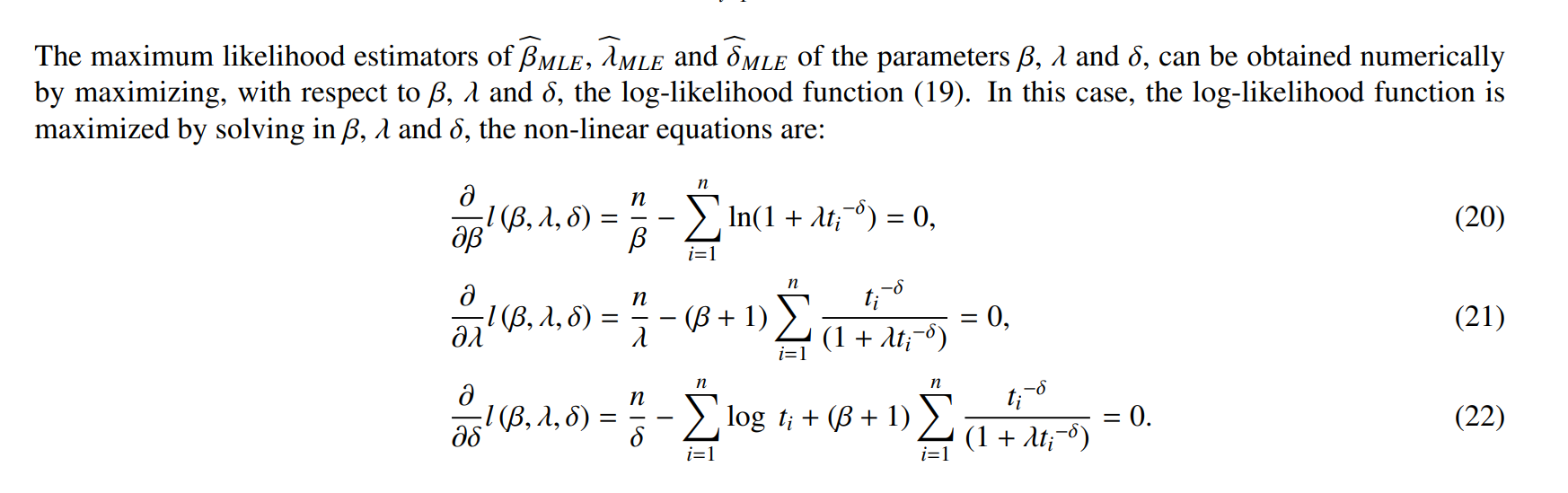
#### Paper Parameterisation

Uses results from paper Dey\_Dagum\_Properties



#### MLE

Uses results from paper Dey\_Dagum\_Properties



#### Conjugate Prior

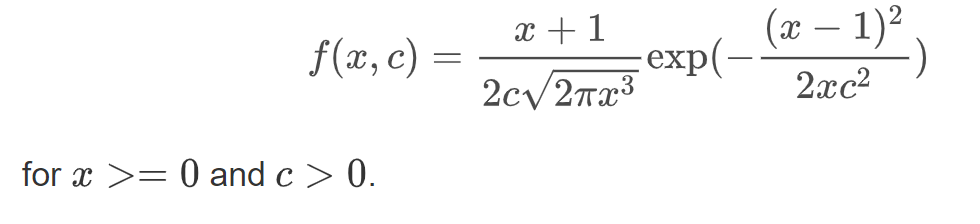
From Alotaibi\_Dagum: When all the parameters are unknown, no joint conjugate prior is available

# Fatiguelife (Birnbaum-Saunders)

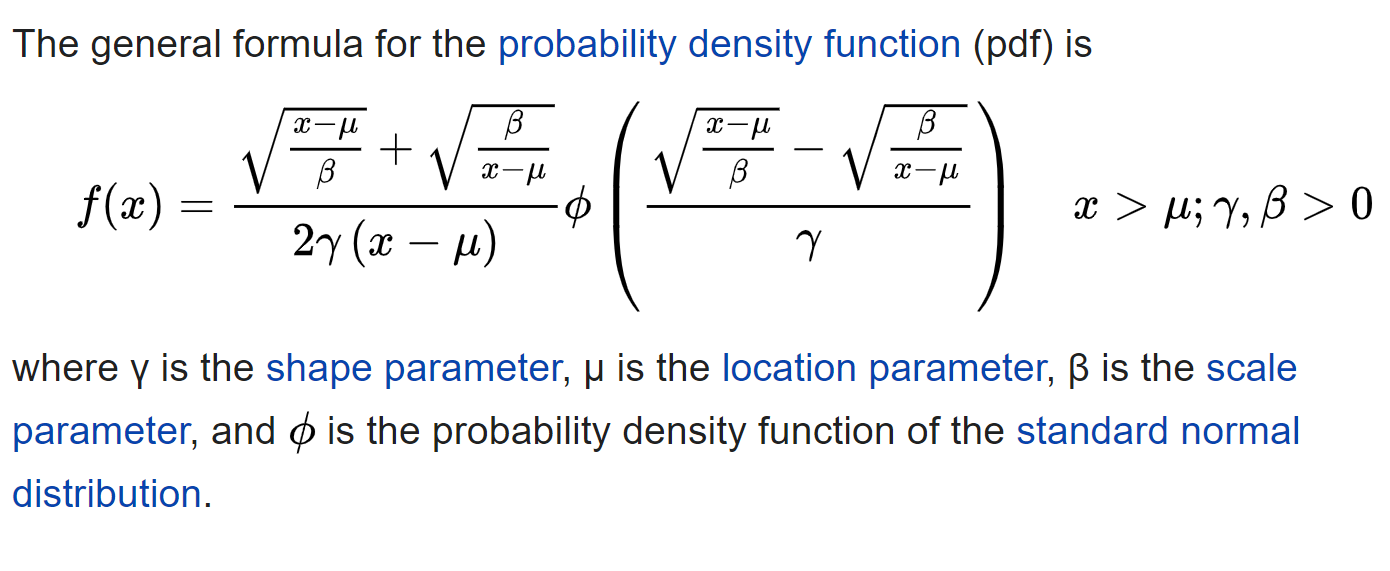
#### Common uses

Used in reliability applications to model failure times due to cracks

#### Scipy parameterisation



#### Wikipedia parameterisation



#### MLE

Horribly complex

See Balakrishnan\_Fatiguelife paper

Conjugate prior

Continuous conjugate joint prior distribution does not exist

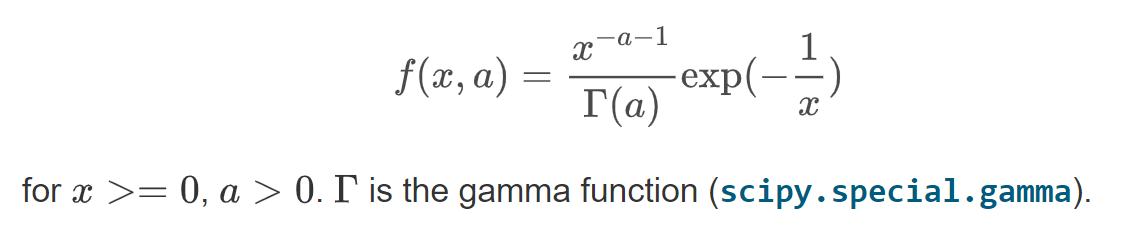
See Wang\_Fatiguelife\_BayesianAnalysis

# Invgamma

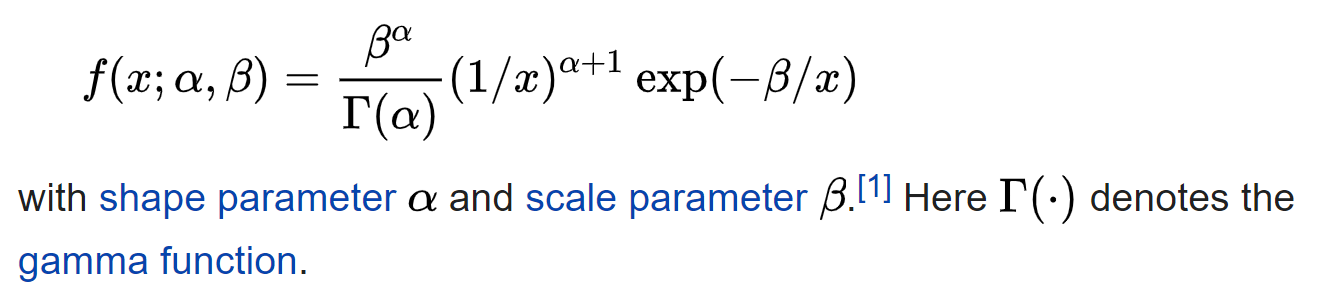
#### Common uses

* marginal posterior distribution for the unknown variance of a normal distribution, if an uninformative prior is used
* conjugate prior for unknown variance of a normal distribution (i.e. known mean)

#### Scipy parameterisation



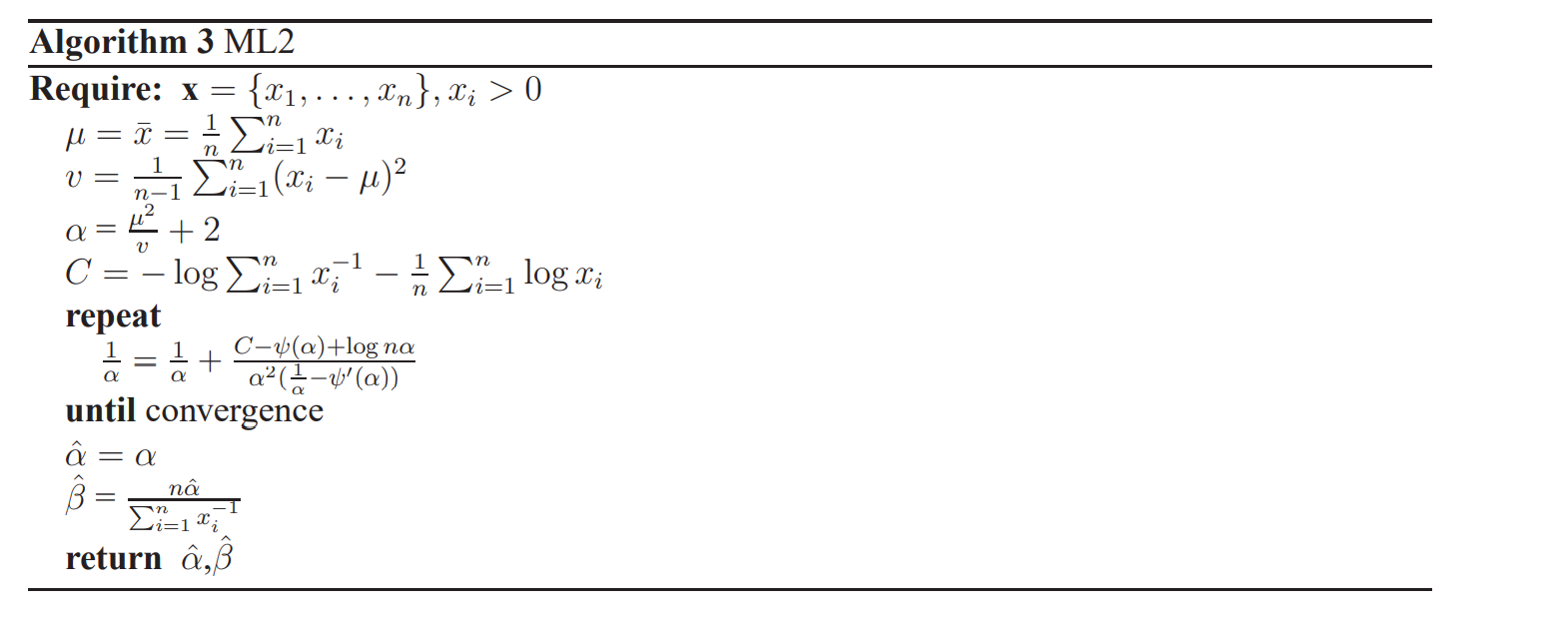
#### Wikipedia parameterisation



* Beta is a scale parameter

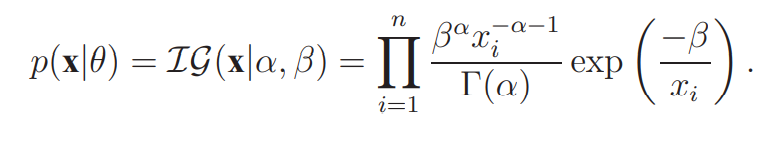
#### MLE

Uses LLera\_InverseGamma

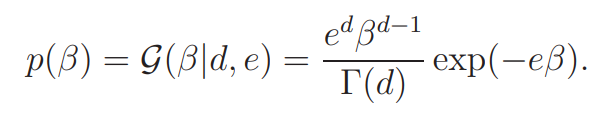


#### Conjugate priors

**Likelihood**

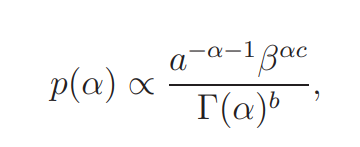


**Gamma conjugate prior for scale parameter, beta**

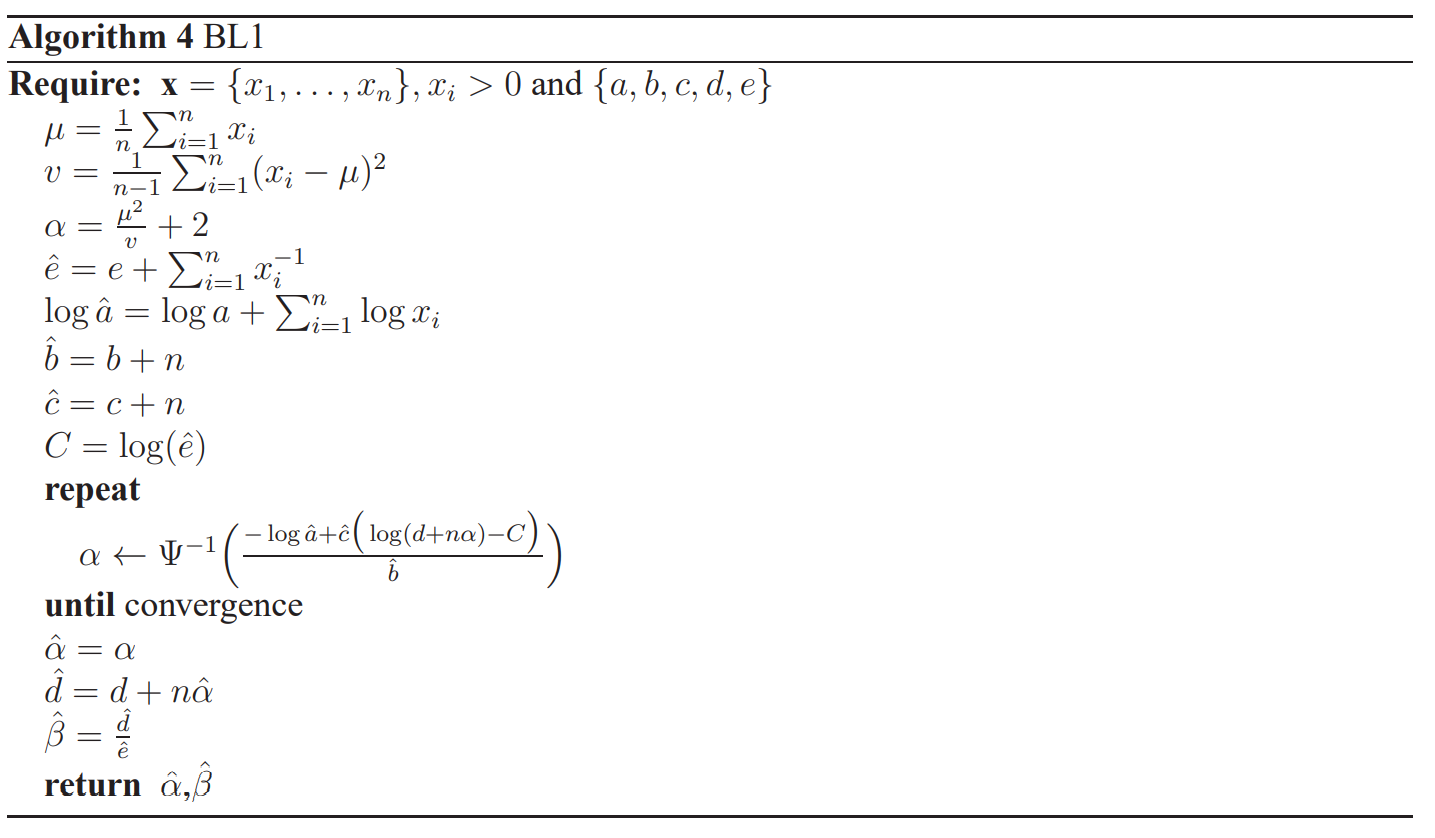


**Conjugate prior for shape parameter, alpha**

From Llera\_InverseGamma



An approximation algorithm exists for both parameters



# Summary of Likelihood distributions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Distribution** | **Parameters** | **MLE** | **Conjugate priors** | **Common use** | **Difficulty** |
| Lognorm | Mu, sigma | Simple | Transform lognormal to normal | If X is lognormal, then lnX is normal | Easiest |
| Powerlognorm |  | Hard | None that I could find |  | Hard & unsuitable |
| Invgauss (Wald) | Mu, lambda | Simple | Exists, but complicated | Time for Brownian motion to reach a certain level | Medium |
| Mielke  (Dagum / burr) | a, b, p | Hard | None for optimising both parameters | Modelling income & wealth distributions | Hard & unsuitable |
| Fatiguelife | Mu, gamma, beta | Hard | None for optimising both parameters | Models failure times due to crack growth | Hard & unsuitable |
| Invgamma | Alpha, beta | Algorithm exists | Algorithm exists | Used as a prior | Medium |

Also try Gaussian Mixture Models & Kernel estimators