

Case	Program
<b>Integer parameters: Number of objects</b>	
Total number of objects in the population	TagAndRelease.py
Number of marked or tagged objects in a population of known size.	MarkedPopulation.py
Total number in a population that is serially labelled (i.e. 1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> etc)	FamilySize.py
<b>Proportion/Fraction type parameter</b>	
One parameter	ProportionParameter.py
Difference between two proportions/fractions	DifferenceProportionParameter.py
Compare multiple proportions/fractions	MultiProportionParameter.py
Multiple fractions vs. dose - logistic response model	ProportionDoseResponse.py
<b>Rate constant type parameters, e.g. Poisson process</b>	
Rare events, no background	RareCounts.py
Rare events with background	RareCountsBackgnd.py
Multiple observations of rare events	MultiRareCounts.py
<b>Measures of central tendency and spread: The Mean and Standard Deviation of a population</b>	
Standard model: Gaussian (or normally distributed) noise	MeanStdDev.py
Large values of noise or errors occurs more frequently than normal	MeanStdDevFatTailNoise.py
Compare two means and/or two variances (Bayesian replacement for 'T'-test and 'F'-test)	DifferenceInMeans.py
Compare two means when the raw data is not available: only have summary data	DiffMeansFromStats.py
Compare multiple means. Hierarchical model with hyper-parameters. (Bayesian replacement for ANOVA)	MultiMeanHierarchy.py
<b>Non-parametric comparison of populations</b>	
Rank test (Bayesian replacement for Wilcoxon rank test)	RankTest.py+DifferenceInMeans.py
Population ID (Classification)	PopulationID.xls
<b>Survival/Decay type data</b>	
Exponential decay in time or space	DecayTimeLength.py
General decay in time or space (Survival analysis)	SurvivalWeibull.py
<b>Curve Fitting</b>	
Straight line	LinearRegression.py
Polynomial	CurveFitBIC.py
Sinusoidal	PeriodicSeries.py
<b>General</b>	
Change/difference in a parameter given two posterior pdfs	DiffPdf.py