ECE 510 Lecture 5
Reliability Plotting
T&T 6.1-6

Scott Johnson
Glenn Shirley

Reliability Plotting

Probit Plots in Excel

=NORMSINV(C6)

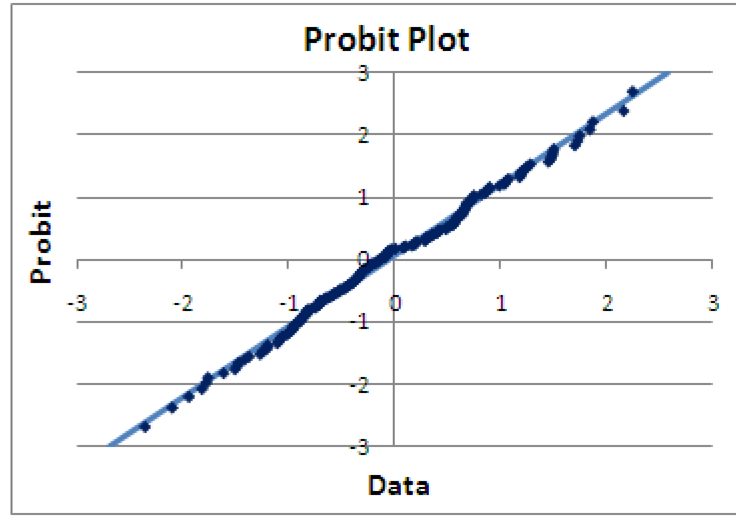
	200	
Data	CDF	Probit
2.92116	0.482535	-0.04379
4.69107	0.796906	0.830621
3.863768	0.622255	0.31141
0.556751	0.118263	-1.18371

mean (μ)	-0.05815
std dev (σ)	0.879253

probits data

= -INTERCEPT(D6:D205,B6:B205) / SLOPE(D6:D205,B6:B205)

=1/SLOPE(D6:D205,B6:B205)

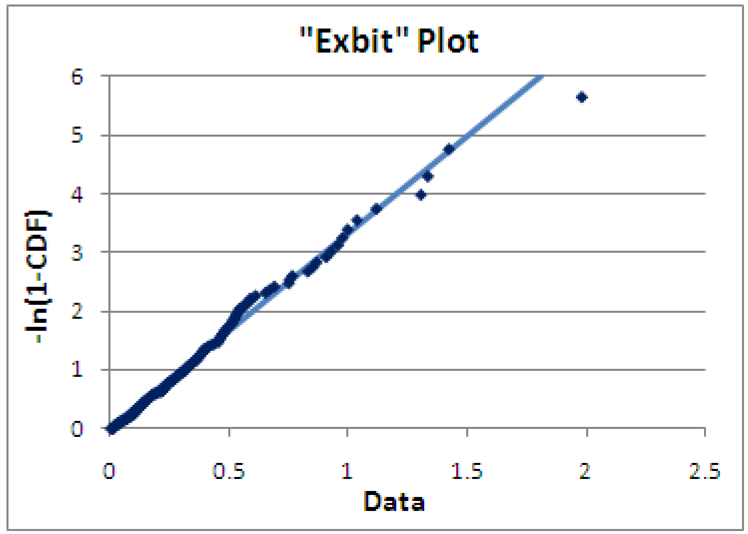


- Plot using:
  - y-axis = probit = NORMSINV(CDF)
  - x-axis = x
  - σ = 1/slope
  - μ = x-intercept = – (y-intercept) / slope

“Exbit” Plots

= -LN(1-H7)

Data	CDF	Probit	Exbit
0.257295	0.557385	0.144343	0.815055
0.04842	0.128244	-1.13473	0.137245
0.134112	0.347804	-0.39125	0.427411
0.032308	0.083333	-1.38299	0.087011



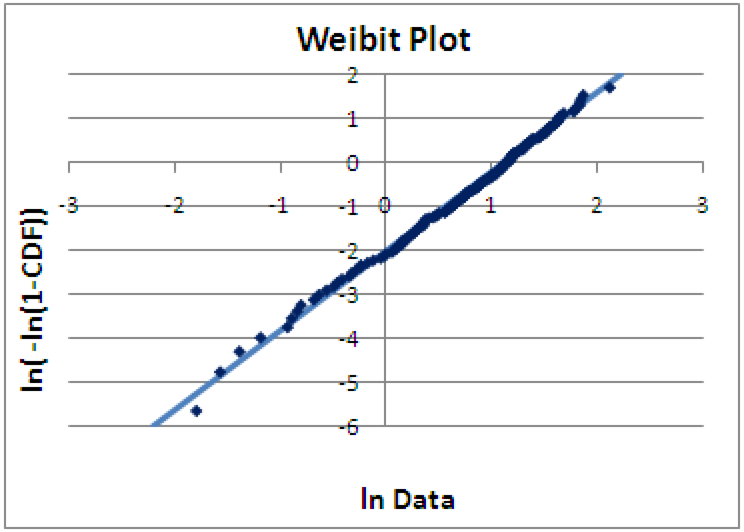
“exbits” data

=SLOPE(J7:J206,G7:G206)

lambda (λ)	3.29628329

- Plot using:
  - y-axis = “exbit” = -LN(1-CDF)
  - x-axis = x
  - λ = slope
- Note that “exbit” is not a standard name

Weibit Plots



=LN(-LN(1-H7))

Data	CDF	Probit	Exbit	ln Data	Weibit
3.857623	0.796906	0.830621	1.594087	1.350051	0.466301
3.044861	0.627246	0.324567	0.986835	1.113455	-0.01325
2.905862	0.582335	0.207871	0.873076	1.06673	-0.13573

Weibit ln data

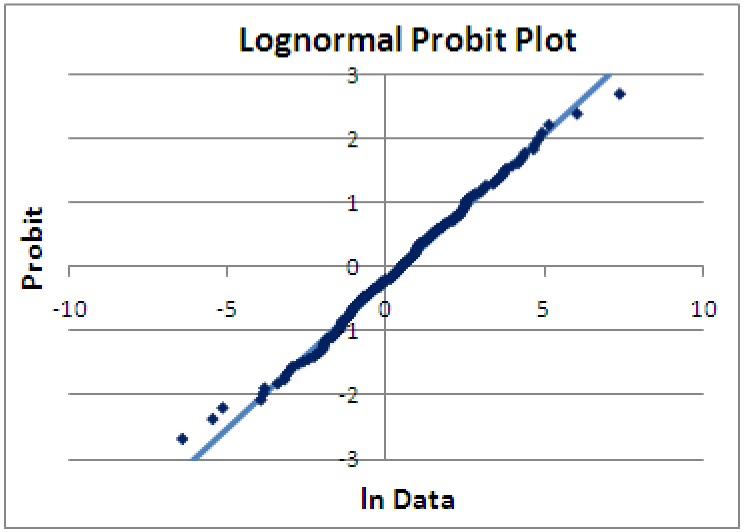
=SLOPE(L7:L206,K7:K206)

shape (β)	1.80701926
scale (α)	3.05820444

=EXP(-INTERCEPT(L7:L206,K7:K206)/T38)

- Plot using:
  - y-axis = Weibit = ln(-ln(1-CDF))
  - x-axis = ln(x)
  - β = slope
  - α = exp(-intercept/slope)
- Note that "Weibit" *is* a standard name

Lognormal Probit Plot



Data	CDF	Probit	Exbit	ln Data	Weibit
0.072804	0.068363	-1.48809	0.070812	-2.61998	-2.64772
5.155989	0.722056	0.58896	1.280335	1.640159	0.247122
171.1415	0.986527	2.212298	4.307064	5.142491	1.460256

probits ln data

= -INTERCEPT(I7:I206,K7:K206) / SLOPE(I7:I206,K7:K206)

mean (μ)	0.46846058
std dev (σ)	2.21137733

=1/SLOPE(I7:I206,K7:K206)

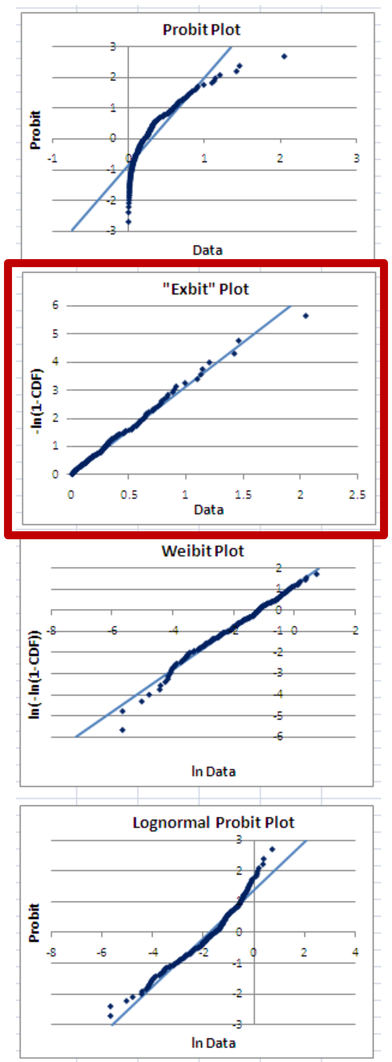
- Plot using:
  - y-axis = probit = NORMSINV(CDF)
  - x-axis = ln(t)
  - σ = 1/slope
  - ln(t50) = x-intercept

Exercise 4.3

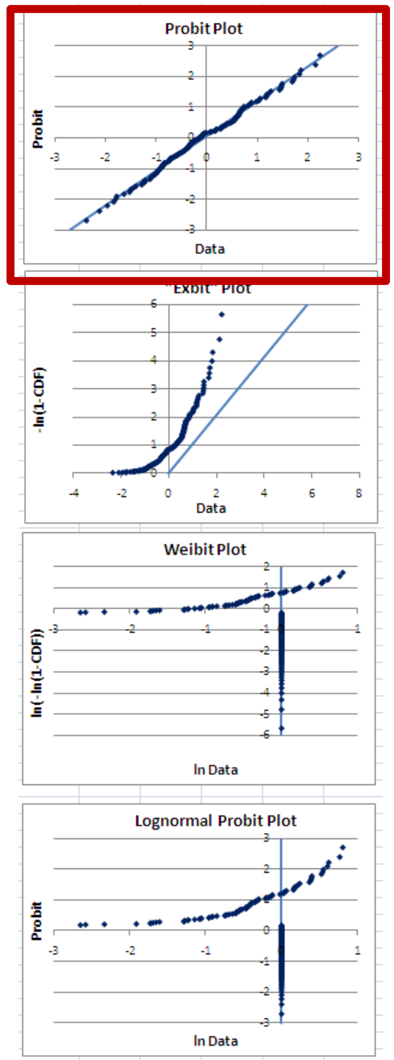
- Make probit, “exbit”, Weibit, and lognormal probit plots
- Determine parameters for each plot
- Look at all 4 data sets (0 – 3)
- Determine which type each distribution is
  - Give the parameters for each correct distribution

Solution 4.3

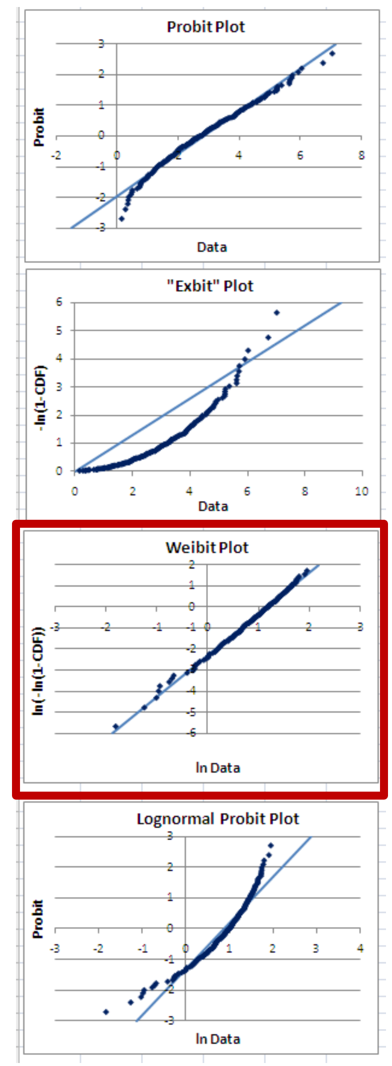
Data0 – exponential
λ = 3.12



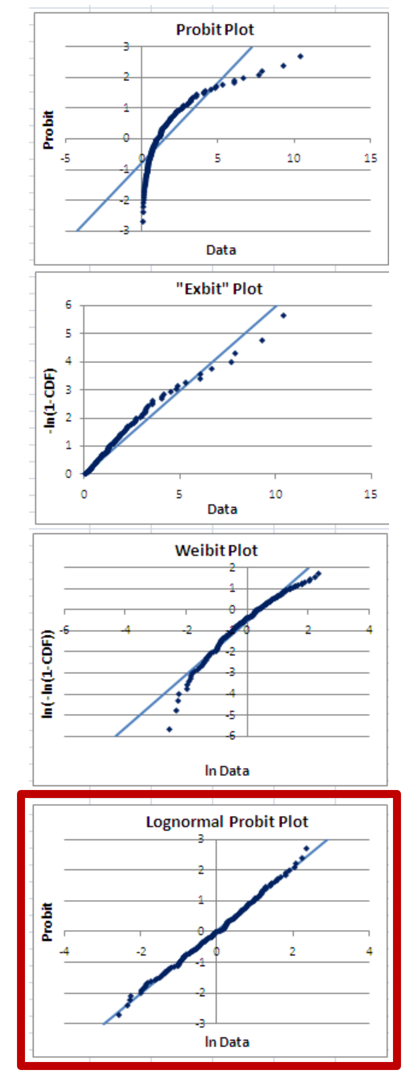
Data1 – normal
μ = –0.06 σ = 0.88



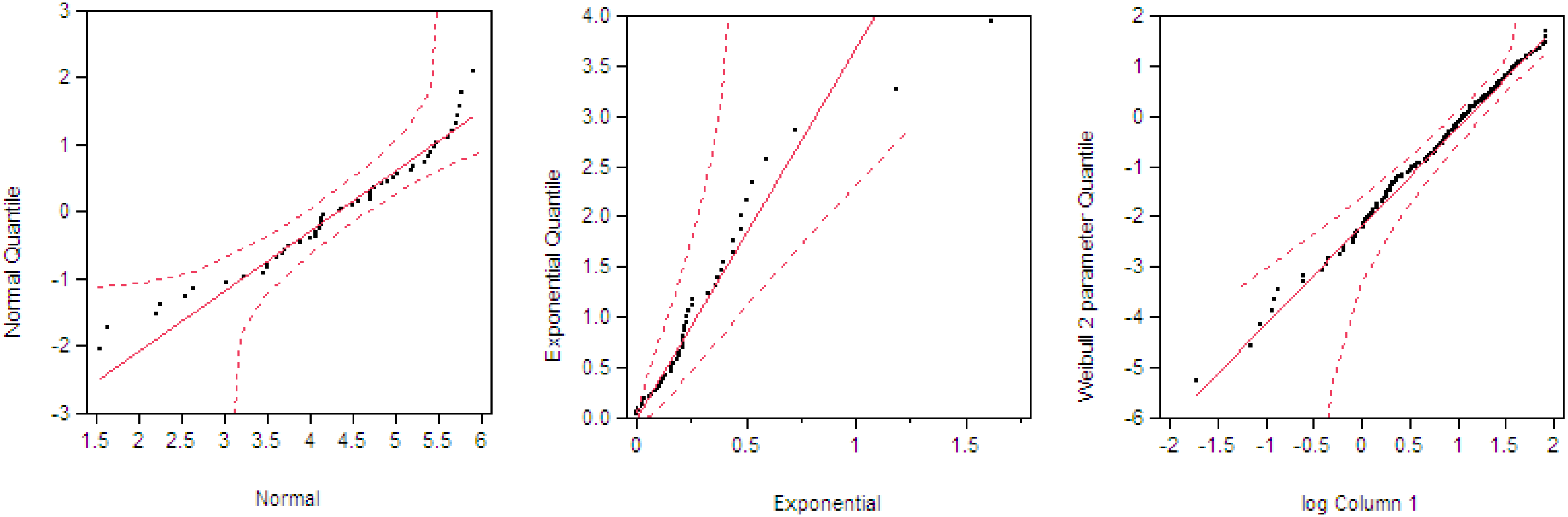
Data2 – Weibull
α = 3.22 β = 1.97



Data3 – lognormal
μ =0.88 σ = 0.67

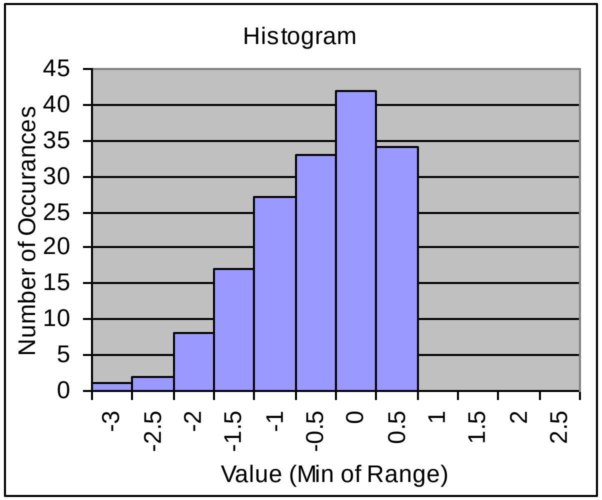


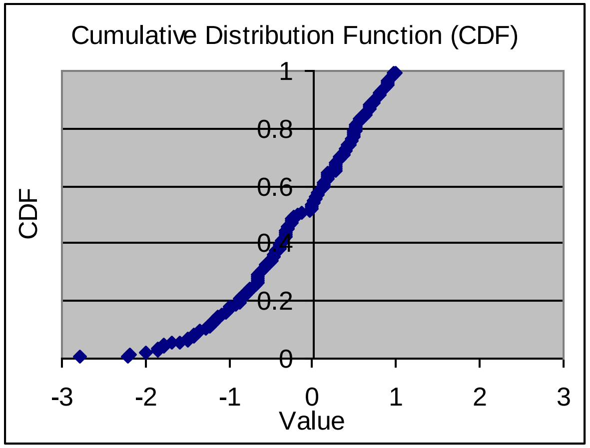
JMP Plots

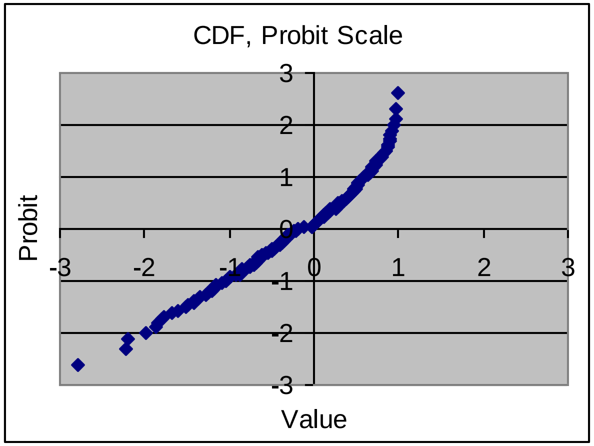


- JMP versions of probit, “exbit”, and Weibit plots

Truncated Distributions







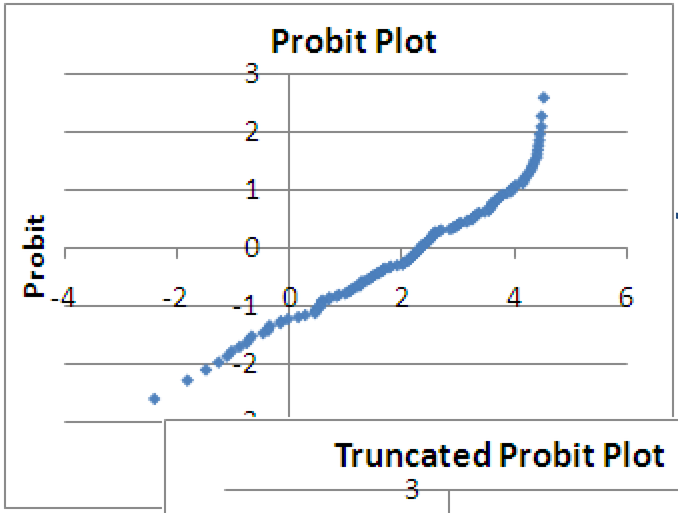
Top-Truncated Distributions

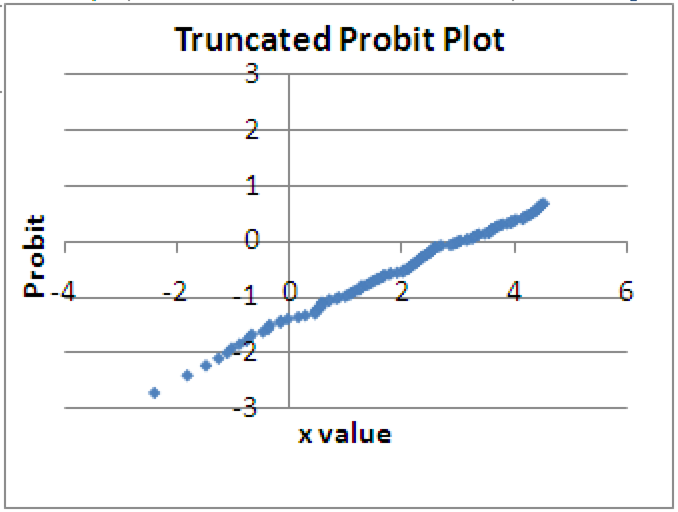
$$\frac{Rank - 0.3}{Count + 0.4}$$

$$\frac{Rank - 0.3}{Count + Missing + 0.4}$$

Missing	50			
Count	150			
Data	**CDF**	**Probit**	**Adj CDF**	**Adj Probit**
4.510582	0.995346	2.600519	0.747006	0.665098
4.473302	0.988697	2.280022	0.742016	0.649573
4.469389	0.982048	2.09801	0.737026	0.634203
4.438034	0.975399	1.966836	0.732036	0.618982
4.42984	0.96875	1.862732	0.727046	0.603903

Note Adj CDF doesn't reach 1

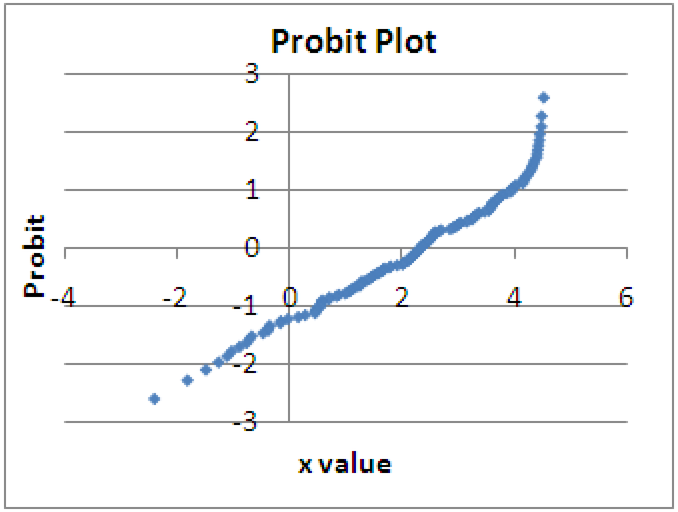




Exercise 5.1

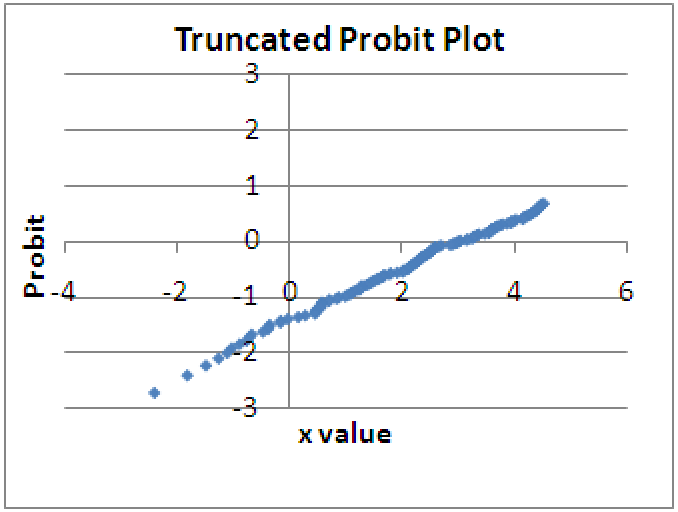
- Make a truncated probit plot of the data on tab Ex 5.1.
- Find the mean and standard deviation of the original distribution as best you can.

Solution 5.1



mean (μ)	2.177593
std dev (σ)	1.630164

Original:	
mean (μ)	3.00
std dev (σ)	2.03



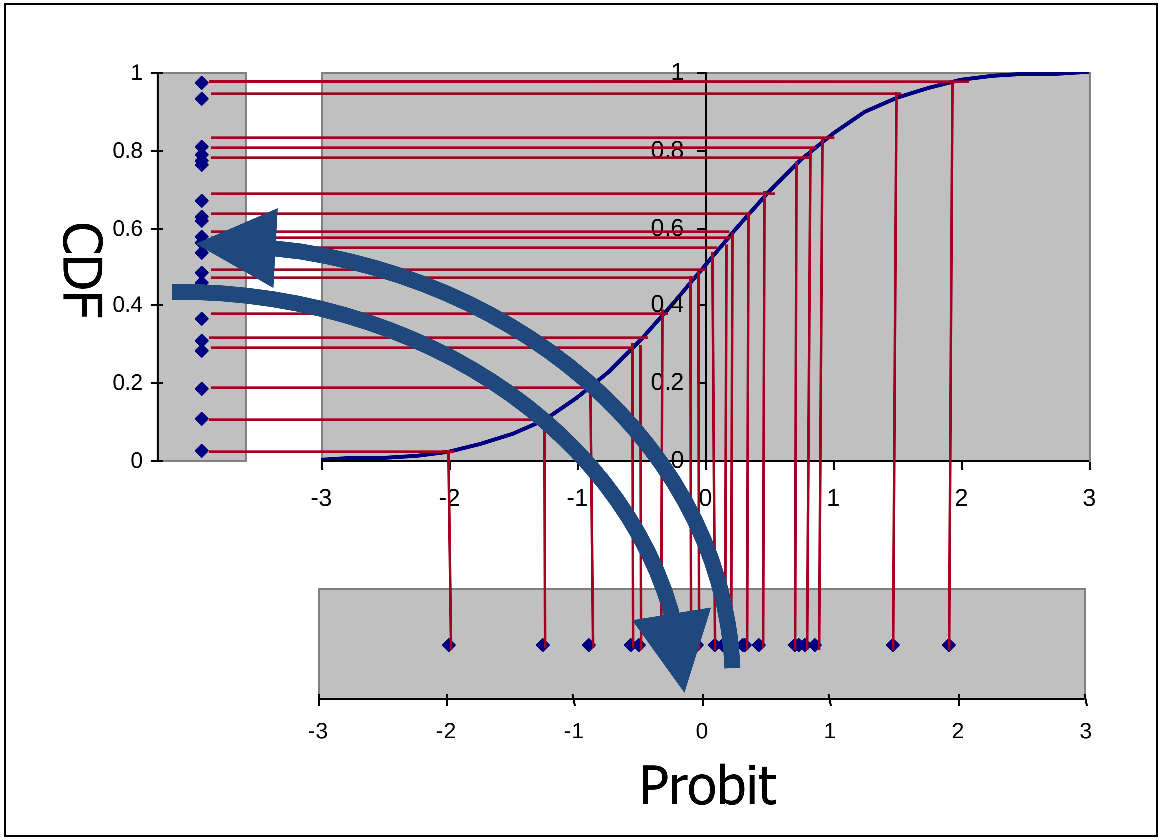
mean (μ)	3.097412
std dev (σ)	2.188841

Data Censoring

- Missing data is called "censored"
  - Type I, time censored
    - Exact times to fail up to time t; no data after
  - Type II, fail count censored
    - Exact times to fail for the first r units to fail; no data after
  - Multicensored or readout
    - Have a time interval within which each unit failed up to tmax; no data after

Generating Random Distributions

CDF as Translator

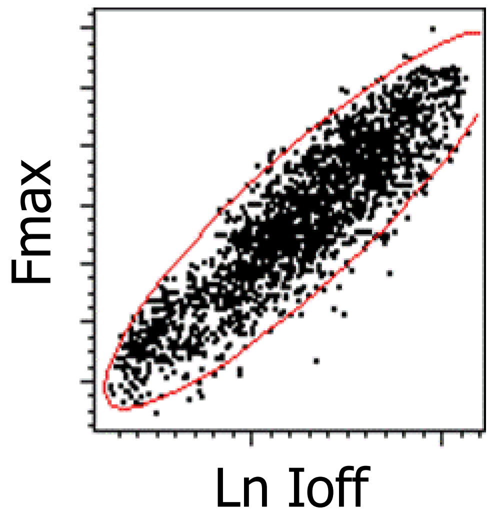


- Probit = NORMSINV(CDF)
- CDF = NORMSDIST(Probit)

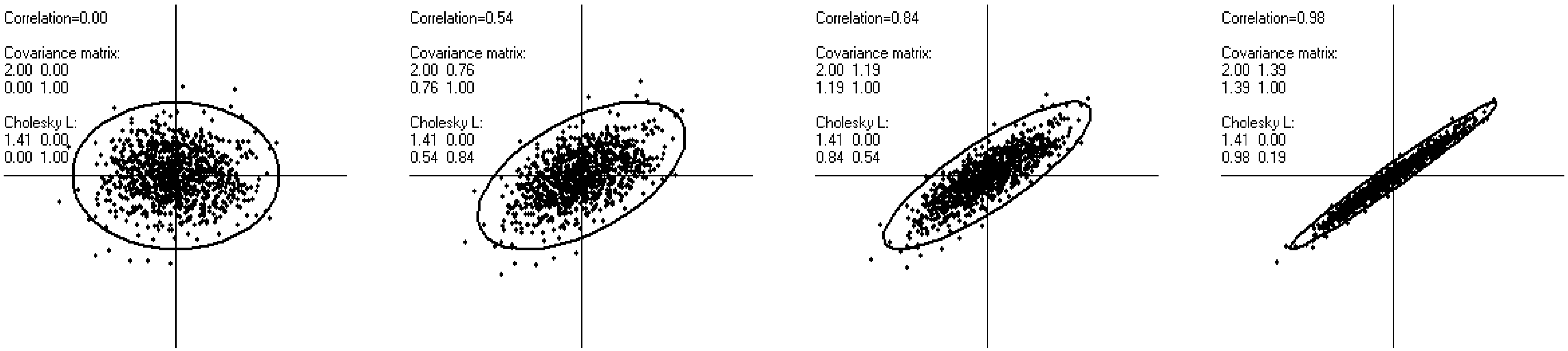
Correlations

- Quantities like Isb and Fmax are correlated
- Correlations are combined with variances ($=$stdev^2$=\sigma^2$) in a covariance matrix

$$\begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} \mathrm{STDEV}(list_1)^2 & \mathrm{COVAR}(list_1, list_2) \\ \mathrm{COVAR}(list_1, list_2) & \mathrm{STDEV}(list_2)^2 \end{bmatrix}$$



- Correlated random variables are synthesized for simulations (as below)



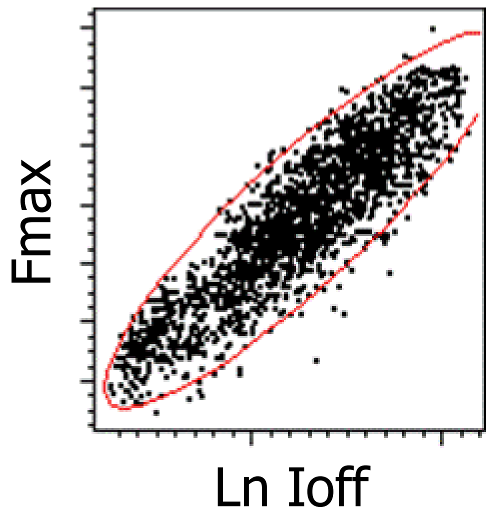
Correlation Matrix

scalars: $x = \mu + \sqrt{v} \times n$

vectors: $\mathbf{x} = \boldsymbol{\mu} + \sqrt{\mathbf{v}} \times \mathbf{n}$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + \sqrt{\begin{bmatrix} c_{11} & c_{21} \\ c_{12} & c_{22} \end{bmatrix}} \times \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

Correlated result vector | mean vector | Cholesky root of covariance matrix | vector of normals



- Correlations among parameters are handled by a matrix version of the variance, the covariance matrix
- Using this matrix formalism, random, *correlated*, normal distributions can be generated

Exercise 5.1a

	A	B	C	D	E	F	G	H	I
1	corr 1	corr 2							
2	2.043164	1.348018		Input matrix			Cholesky matrix		
3	1.067295	0.648187		variance 1	covariance		(Don't edit these cell formulas -- they take the Cholesky root of the green matrix.)		
				covariance	variance 2				
6	[illegible]	[illegible]		1.737355	1.123414		1.318088	0	
7	1.035236	1.264849		1.123414	0.869569		0.852306	0.378343	
8	0.510458	0.843535							
				corr =	0.913995				
11	[illegible]	[illegible]		1.737355	1.123414				
12	-1.40594	-0.40367		1.123414	0.869569				
13	-1.7017	-1.0635							
14	0.824828	0.408285		rand 1	rand 2	norm 1	norm 2	corr 1	corr 2
15	0.905433	1.159727		0.443094	0.614765	-0.14313	0.291759	-0.18866	-0.0116
16	1.282392	0.638498		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
17	-2.66669	-1.6663		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
18	0.758024	-0.17095		[illegible]	0.599276	[illegible]	0.251474	[illegible]	-0.61838
19	-0.51645	-0.27492		[illegible]	0.867503	[illegible]	1.114663	[illegible]	1.551428
20	2.452766	[illegible]		[illegible]	0.576667	[illegible]	[illegible]	[illegible]	0.753014
21	0.244298	[illegible]		[illegible]	0.822614	[illegible]	[illegible]	[illegible]	-0.55577
22	0.110783	[illegible]		[illegible]	0.072829	-0.17061	[illegible]	[illegible]	-0.69592
23	-0.87227	[illegible]		[illegible]	[illegible]	1.095113	[illegible]	[illegible]	0.623961
24	3.20297	[illegible]		[illegible]	[illegible]	-0.48173	[illegible]	-0.63496	-0.2145
25	1.795466	[illegible]		[illegible]	[illegible]	-0.66016	[illegible]	-0.87015	-1.1776
26	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	2.089083	1.018589
							[illegible]	-1.87533	-1.23956
							[illegible]	-0.67553	-0.318
							[illegible]	0.175101	0.13847
							[illegible]	0.792125	1.048567
31	[illegible]	[illegible]		0.121678	0.210588	-1.16664	-0.80438	-1.53774	-1.29867
32	1.5261	0.82225		0.681311	0.652081	0.471368	0.390946	0.621304	0.549661
33	-2.4084	-1.48554		0.341648	0.553597	-0.40797	0.134754	-0.53774	-0.29673
34	-2.29329	-1.06052		0.376075	0.924159	-0.31581	1.433613	-0.41626	0.273234
35	-0.66483	-0.29574		0.787645	0.669069	0.798276	0.437344	1.052197	0.845841
36	-1.41346	-0.96915		0.007106	0.334248	-2.45188	-0.42821	-3.23179	-2.25176
37	0.251672	0.582907		0.002789	0.319443	-2.77159	-0.46926	-3.6532	-2.53978
38	-0.80988	-0.7955		0.560619	0.603851	0.152539	0.263327	0.20106	0.229638
39	-1.68399	-0.48199		0.899624	0.04032	1.279412	-1.74698	1.686377	0.429494
40	0.561073	0.118965		0.926835	0.88807	1.45262	1.216326	1.914681	1.698265
41	-0.7502	-0.80612		0.816592	0.034234	0.902455	-1.82191	1.189515	0.079861
42	-1.26139	-0.83376		0.10097	0.253855	-1.27604	-0.66241	-1.68193	-1.33819
43	0.028836	-0.04276		0.266979	0.501327	-0.62198	0.003326	-0.81982	-0.52885

=STDEV(A2:A101)^2

=COVAR(A2:A101,B2:B101)

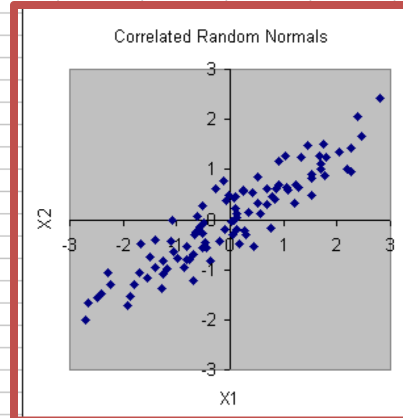
=RAND()

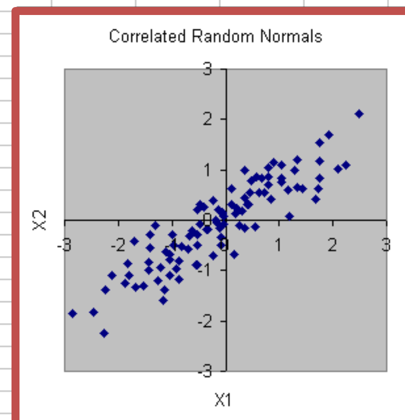
=NORMSINV(D15)

Matrix x Vector

=$G$6*F15+$H$6*G15

=$G$7*F15+$H$7*G15





Exercise 5.2b

- What fraction of this population has both X1<1 and X2<2? Use a Monte Carlo simulation. Give both your answer and an indication of how accurate it is.

Generating Random Numbers

$$\text{rand exponential} = -\frac{\ln(1-CDF)}{\lambda}$$

`= -LN(RAND())/$B$5`

$$\text{rand normal} = NORMSINV(CDF)$$

`=NORMSINV(RAND())*$C$5+$C$3`

$$\text{rand Weibull} = \alpha[-\ln(1-CDF)]^{1/\beta}$$

`= $D$3*(-LN(1-RAND()))^(1/$D$5)`

$$\text{rand normal} = \exp(NORMSINV(CDF))$$

`=EXP(NORMSINV(RAND())*$E$5+$E$3)`

	mean (μ)	scale (α)	mean (μ)
	4	3	0.5
lambda (λ)	std dev (σ)	shape (β)	std dev (σ)
3	1	2	2
Exponential	Normal	Weibull	Lognormal
0.25943959	4.2940054	2.4556021	1.82753248
0.15294363	2.0571855	2.6443629	0.22927996
0.36137749	2.7199053	3.6447156	0.82474469

Exercise 5.3

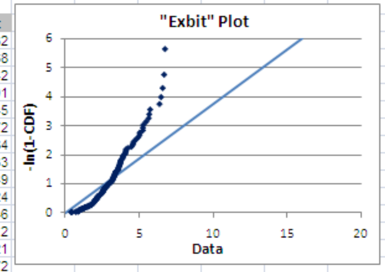
- Replace the 4 columns Data0 through Data3 from exercise 4.3 with random number generators for each of the 4 functions.

Solution 5.3

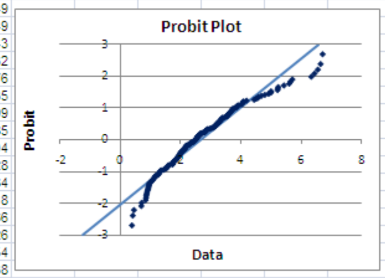
	mean (μ)	scale (α)	mean (μ)
	4	3	0.5
lambda (λ)	std dev (σ	shape (β)	std dev (σ)
3	1	2	2

Input	
2	<-- 0-3 for exponential - lognormal

Exponentia	Normal	Weibull	Lognormal	Data	CDF	Probit	Exbit	ln Data	Weibit
0.2594396	4.294005	2.455602	1.8275325	2.4556	0.48253	-0.04379	0.65881	0.89837	-0.41732
0.1529436	2.057185	2.644363	0.22928	2.64436	0.56238	0.15699	0.82639	0.97243	-0.19068
0.3613775	2.719905	3.644716	0.8247447	3.64472	0.80689	0.86648	1.64448	1.29328	0.49742
0.1166976	2.997349	3.202084	0.1114537	3.20208	0.68713	0.48772	1.16195	1.1638	0.1501
0.1339676	4.451015	2.630789	1.2692602	2.63079	0.55739	0.14434	0.81506	0.96728	-0.2045
0.077476	4.069708	2.488495	0.159152	2.48849	0.4975	-0.00625	0.68817	0.91168	-0.37372
0.1053087	3.383248	3.20448	4.721108	3.20448	0.69212	0.50186	1.17803	1.16455	0.16384
0.7363992	4.085132	1.874399	0.8194747	1.8744	0.27295	-0.6039	0.31877	0.62829	-1.1433
0.5928205	3.077203	0.886599	0.4706787	0.8866	0.0484	-1.66054	0.04961	-0.12036	-3.00349
0.7156102	4.125102	0.390412	5.1165543	0.39041	0.00349	-2.69751	0.0035	-0.94055	-5.65524
0.5394854	4.627135	0.842362	15.785497	0.84236	0.02844	-1.90418	0.02886	-0.17155	-3.54546
0.0985859	4.136847	3.304352	0.2013189	3.30435	0.72206	0.58896	1.28034	1.19524	0.24712
0.6552569	5.014581	1.036466	0.9128894	1.03647	0.09331	-1.32062	0.09796	0.03582	-2.32321
0.4931708	3.466973	1.58976	2.058762	1.58976	0.20309	-0.83062	0.22702	0.46358	-1.48272
1.1435902	4.119858	6.33339	1.8723406	6.33339	0.97655	1.98715	3.75275	1.84584	1.32249
0.4094484	3.477385	2.528186	0.7169048	2.52819	0.51747	0.04379	0.7287	0.9275	-0.31649
0.1143007	4.641488	1.264781	13.01411	1.26478	0.13822	-1.08834	0.14876	0.2349	-1.90543
0.2532303	3.116112	2.500235	0.0678124	2.50024	0.5025	0.00625	0.69815	0.91638	-0.35932
0.0867345	3.351616	0.90508	1.497345	0.90508	0.05339	-1.61281	0.05487	-0.09973	-2.90276
0.3826959	5.95218	1.919054	0.4728648	1.91905	0.2979	-0.53044	0.35369	0.65183	-1.03935
0.3703065	5.254077	1.858797	1.6172887	1.8588	0.26796	-0.61898	0.31193	0.61993	-1.16499
0.3672311	2.448294	1.743947	1.9914196	1.74395	0.23303	-0.72889	0.26531	0.55615	-1.32685
0.4554272	3.720878	1.295141	114.9978	1.29514	0.1482	-1.04417	0.16041	0.25862	-1.83004
0.5486102	3.260955	3.327886	1.9728172	3.32789	0.73204	0.61898	1.3169	1.20234	0.27528
0.2139738	3.422584	3.554968	1.747406	3.55497	0.77196	0.7453	1.47822	1.26835	0.39084
0.1360725	4.733957	2.476432	1.2124052	2.47643	0.49251	-0.01876	0.67829	0.90682	-0.38818
0.1633239	3.213611	1.759555	0.9473734	1.75956	0.24301	-0.69664	0.27841	0.56506	-1.27866
0.7442227	5.439803	1.530915	0.447459	1.53091	0.18812	-0.88483	0.20841	0.42587	-1.56826
1.3896049	4.762249	1.755819	0.3533413	1.75582	0.23802	-0.71267	0.27184	0.56294	-1.30254
0.0046929	3.166331	4.079416	1.427722	4.07942	0.88673	1.2093	2.17795	1.40595	0.77838
0.0556413	3.714354	1.902523	0.0707193	1.90252	0.28792	-0.55946	0.33957	0.64318	-1.08007
0.2826799	3.999868	0.958469	0.1438538	0.95847	0.07834	-1.4163	0.08158	-0.04242	-2.50614
0.0482062	2.662349	5.217069	19.597014	5.21707	0.94162	1.56849	2.84073	1.65194	1.04406
0.1481542	3.398414	3.086362	4.0956621	3.08636	0.6522	0.39125	1.05612	1.12699	0.0546
0.0338686	4.450813	2.564436	0.2463566	2.56444	0.54242	0.10652	0.78179	0.94174	-0.24617
0.1000715	2.345938	1.604572	229.17626	1.60457	0.21307	-0.7958	0.23962	0.47286	-1.4287
0.1062232	2.147338	2.117984	1.3568776	2.11798	0.37774	-0.31141	0.4744	0.75046	-0.74569
0.711743	3.060636	0.921237	0.379926	0.92124	0.05838	-1.56849	0.06016	-0.08204	-2.8108
0.9766721	4.956015	3.80462	4.2924811	3.80462	0.84182	1.00195	1.844	1.33622	0.61194
0.3205808	3.47058	0.952024	10.102159	0.95202	0.07335	-1.45126	0.07618	-0.04917	-2.57462
0.2346252	4.535575	2.674959	10.809712	2.67496	0.58234	0.20787	0.87308	0.98393	-0.13573
0.1609403	3.476108	2.01523	1.9517497	2.01523	0.34281	-0.40479	0.41979	0.70073	-0.868
0.5455537	4.53415	1.555717	4.072061	1.55572	0.19311	-0.86648	0.21457	0.44194	-1.53911
0.9438724	3.474965	3.360762	2.8814505	3.36076	0.74202	0.64957	1.35486	1.21217	0.3037
0.0073427	3.777688	5.705027	44.926681	5.70503	0.97156	1.90418	3.55985	1.74135	1.26972
0.3249322	4.864663	0.701584	5.0439482	0.70158	0.01846	-2.08658	0.01864	-0.35442	-3.98268
0.0654376	6.060194	1.740606	3.7983472	1.74061	0.22305	-0.76192	0.25238	0.55423	-1.3768
0.0742414	5.493994	1.777578	1.4592887	1.77758	0.248	-0.68078	0.28502	0.57525	-1.25518
0.0055238	4.272769	6.708215	1.7989675	6.70821	0.99651	2.69751	5.65699	1.90333	1.73289
0.2319599	4.96945	5.000358	0.1832161	5.00036	0.93663	1.52706	2.75871	1.60951	1.01476
0.3982416	4.226371	1.600306	1.8897729	1.60031	0.20808	-0.81309	0.2333	0.4702	-1.45543
0.0182421	3.441178	1.313032	0.1544259	1.31303	0.15818	-1.00195	0.17219	0.27234	-1.75914
0.8028374	4.650147	3.68984	4.9904259	3.68984	0.82186	0.92246	1.72516	1.30558	0.54532
0.2497396	3.339075	3.16696	2.7770437	3.16696	0.67216	0.44587	1.11522	1.15277	0.10905
0.0700481	6.924415	0.875142	6.7803044	0.87514	0.04341	-1.71238	0.04438	-0.13337	-3.11488
0.0090973	4.582525	2.889295	0.7402587	2.88929	0.62725	0.32457	0.98684	1.06101	-0.01325
0.0386866	4.611129	4.574807	1.6421342	4.57481	0.91168	1.35115	2.42675	1.52056	0.88655
1.0077968	4.872402	2.464205	3.4691037	2.4642	0.48752	-0.03128	0.6685	0.90187	-0.40271
0.0012842	3.822115	2.649075	2.744802	2.64907	0.56737	0.16967	0.83786	0.97421	-0.1769
0.0551778	2.850232	1.466602	0.6205933	1.4666	0.17814	-0.92246	0.19619	0.38295	-1.62867
0.2404713	4.528872	2.222792	0.8408888	2.22279	0.43263	-0.16967	0.56675	0.79876	-0.56783

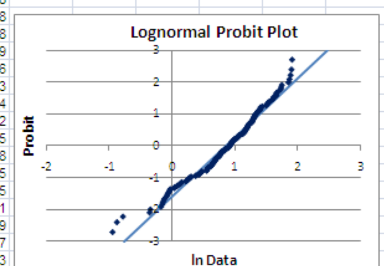


Parameter	Graphical	Best
lambda (λ)	0.7227231	0.37491
fit:		
16.0036288	6	
0	0	



mean (μ)	2.6672715	2.66727
std dev (σ)	1.3522755	1.30878
fit:		
6.59361461	3	
-1.2590717	-3	

shape (β)	2.324313	–
scale (α)	2.9962209	–
fit:		
1.9578211	2	
-1.4839957	-6	



mean (μ)	0.8512807	0.85128
std dev (σ)	0.5505603	0.53799
fit:		
2.4652428	3	
-0.7626814	-3	

The End