

This demonstration illustrates using the PP-plot to check normality of data. We use the canned dataset “Nile” in `R` of measurements of the annual flow of the river Nile at Ashwan 1871–1970.

We will illustrate the construction of the PP-plot “by hand” and using the instructions available in `R`. We shall use the random number generators to illustrate how non-normal PP-Plots look.

---

**R Session:**

---

```
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

```
Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

```
[R.app GUI 1.31 (5538) powerpc-apple-darwin8.11.1]
```

```
[Workspace restored from /Users/andrejstreibergs/.RData]
```

```
> ##### TO CHECK NORMALITY WE COMPARE OBSERVED QUANTILES WITH THEORETICAL
> #
> ##### OBSERVED DATA
> # The canned data set of annual flow of the Nile River.
> N<-Nile;N
Time Series:
Start = 1871
End = 1970
Frequency = 1
 [1] 1120 1160 963 1210 1160
 [6] 1160 813 1230 1370 1140
[11] 995 935 1110 994 1020
[16] 960 1180 799 958 1140
[21] 1100 1210 1150 1250 1260
[26] 1220 1030 1100 774 840
```

```

[31] 874 694 940 833 701
[36] 916 692 1020 1050 969
[41] 831 726 456 824 702
[46] 1120 1100 832 764 821
[51] 768 845 864 862 698
[56] 845 744 796 1040 759
[61] 781 865 845 944 984
[66] 897 822 1010 771 676
[71] 649 846 812 742 801
[76] 1040 860 874 848 890
[81] 744 749 838 1050 918
[86] 986 797 923 975 815
[91] 1020 906 901 1170 912
[96] 746 919 718 714 740
> # To see more information about the data:
> help(Nile)
> # The number of entries in the data:
> n <- length(Nile); n
[1] 100
> # Sort the readings
> N <- sort(N)
> xbar<-mean(N); s<-sd(N); xbar; s
[1] 919.35
[1] 169.2275
> # Standardize: subtract xbar from each entry and divide by s
> sN <-(N-xbar)/s; sN
[1] -2.738030156
[2] -1.597553583
[3] -1.438005047
[4] -1.343457766
[5] -1.331639356
[6] -1.308002536
[7] -1.290274921
[8] -1.284365716
[9] -1.213455255
[10] -1.189818435
[11] -1.142544795
[12] -1.059815924
[13] -1.047997514
[14] -1.036179104
[15] -1.036179104
[16] -1.024360694
[17] -1.006633079
[18] -0.947541029
[19] -0.917995003
[20] -0.894358183
[21] -0.876630568
[22] -0.858902953
[23] -0.817538518
[24] -0.728900442
[25] -0.722991237
[26] -0.711172827

```

[27] -0.699354417  
[28] -0.634353161  
[29] -0.628443956  
[30] -0.616625546  
[31] -0.581170316  
[32] -0.575261111  
[33] -0.563442701  
[34] -0.522078265  
[35] -0.516169060  
[36] -0.510259855  
[37] -0.480713830  
[38] -0.468895420  
[39] -0.439349395  
[40] -0.439349395  
[41] -0.439349395  
[42] -0.433440190  
[43] -0.421621780  
[44] -0.350711319  
[45] -0.338892909  
[46] -0.327074499  
[47] -0.321165294  
[48] -0.267982449  
[49] -0.267982449  
[50] -0.173435168  
[51] -0.132070733  
[52] -0.108433913  
[53] -0.078887887  
[54] -0.043432657  
[55] -0.019795837  
[56] -0.007977427  
[57] -0.002068222  
[58] 0.021568598  
[59] 0.092479059  
[60] 0.122025084  
[61] 0.145661904  
[62] 0.228390775  
[63] 0.240209185  
[64] 0.257936800  
[65] 0.293392030  
[66] 0.328847261  
[67] 0.382030106  
[68] 0.393848516  
[69] 0.441122156  
[70] 0.447031361  
[71] 0.535669437  
[72] 0.594761487  
[73] 0.594761487  
[74] 0.594761487  
[75] 0.653853538  
[76] 0.712945588  
[77] 0.712945588  
[78] 0.772037639

```

[79] 0.772037639
[80] 1.067497891
[81] 1.067497891
[82] 1.067497891
[83] 1.126589941
[84] 1.185681992
[85] 1.185681992
[86] 1.303866093
[87] 1.303866093
[88] 1.362958143
[89] 1.422050193
[90] 1.422050193
[91] 1.422050193
[92] 1.481142244
[93] 1.540234294
[94] 1.717510446
[95] 1.717510446
[96] 1.776602496
[97] 1.835694546
[98] 1.953878647
[99] 2.012970698
[100] 2.662983252

```

```

> ##### THEORETICAL DATA
> # Midpoints of n equal subintervals of [0,1]
> pp<-ppoints(n); pp
[1] 0.005 0.015 0.025 0.035
[5] 0.045 0.055 0.065 0.075
[9] 0.085 0.095 0.105 0.115
[13] 0.125 0.135 0.145 0.155
[17] 0.165 0.175 0.185 0.195
[21] 0.205 0.215 0.225 0.235
[25] 0.245 0.255 0.265 0.275
[29] 0.285 0.295 0.305 0.315
[33] 0.325 0.335 0.345 0.355
[37] 0.365 0.375 0.385 0.395
[41] 0.405 0.415 0.425 0.435
[45] 0.445 0.455 0.465 0.475
[49] 0.485 0.495 0.505 0.515
[53] 0.525 0.535 0.545 0.555
[57] 0.565 0.575 0.585 0.595
[61] 0.605 0.615 0.625 0.635
[65] 0.645 0.655 0.665 0.675
[69] 0.685 0.695 0.705 0.715
[73] 0.725 0.735 0.745 0.755
[77] 0.765 0.775 0.785 0.795
[81] 0.805 0.815 0.825 0.835
[85] 0.845 0.855 0.865 0.875
[89] 0.885 0.895 0.905 0.915
[93] 0.925 0.935 0.945 0.955
[97] 0.965 0.975 0.985 0.995
>

```

```

> # The normal quantiles: pp[i] = P(Z ≤ qq[i]) where Z ~ N(0,1)
> qq <- qnorm(pp); qq
[1] -2.57582930 -2.17009038
[3] -1.95996398 -1.81191067
[5] -1.69539771 -1.59819314
[7] -1.51410189 -1.43953147
[9] -1.37220381 -1.31057911
[11] -1.25356544 -1.20035886
[13] -1.15034938 -1.10306256
[15] -1.05812162 -1.01522203
[17] -0.97411388 -0.93458929
[19] -0.89647336 -0.85961736
[21] -0.82389363 -0.78919165
[23] -0.75541503 -0.72247905
[25] -0.69030882 -0.65883769
[27] -0.62800601 -0.59776013
[29] -0.56805150 -0.53883603
[31] -0.51007346 -0.48172685
[33] -0.45376219 -0.42614801
[35] -0.39885507 -0.37185609
[37] -0.34512553 -0.31863936
[39] -0.29237490 -0.26631061
[41] -0.24042603 -0.21470157
[43] -0.18911843 -0.16365849
[45] -0.13830421 -0.11303854
[47] -0.08784484 -0.06270678
[49] -0.03760829 -0.01253347
[51] 0.01253347 0.03760829
[53] 0.06270678 0.08784484
[55] 0.11303854 0.13830421
[57] 0.16365849 0.18911843
[59] 0.21470157 0.24042603
[61] 0.26631061 0.29237490
[63] 0.31863936 0.34512553
[65] 0.37185609 0.39885507
[67] 0.42614801 0.45376219
[69] 0.48172685 0.51007346
[71] 0.53883603 0.56805150
[73] 0.59776013 0.62800601
[75] 0.65883769 0.69030882
[77] 0.72247905 0.75541503
[79] 0.78919165 0.82389363
[81] 0.85961736 0.89647336
[83] 0.93458929 0.97411388
[85] 1.01522203 1.05812162
[87] 1.10306256 1.15034938
[89] 1.20035886 1.25356544
[91] 1.31057911 1.37220381
[93] 1.43953147 1.51410189
[95] 1.59819314 1.69539771
[97] 1.81191067 1.95996398
[99] 2.17009038 2.57582930

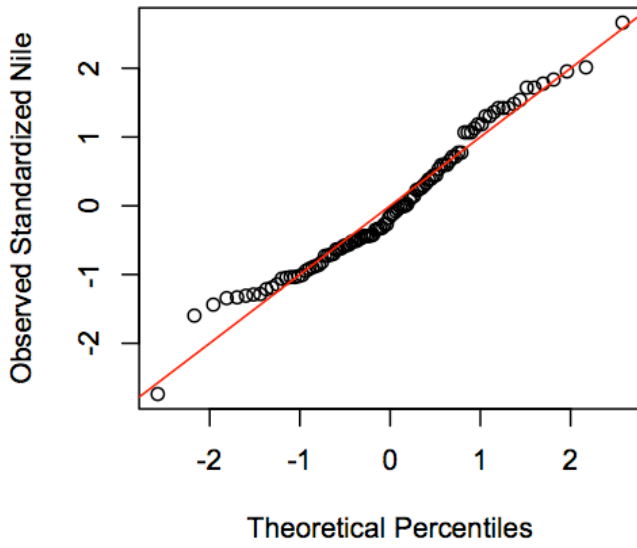
```

```

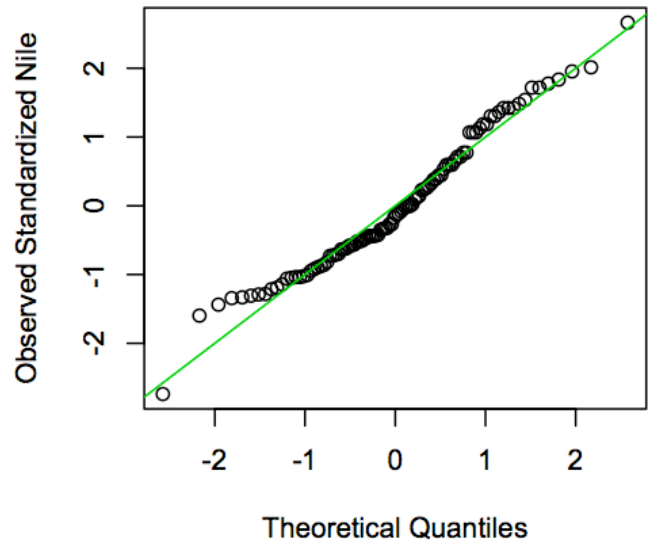
> ##### COMPARE OBSERVED TO THEORETICAL
>
> # Set layout to make four plots per page.
> layout(matrix(c(1,3,2,4),ncol=2))
>
> # Plot Standardized Observed Quantiles vs. Theoretical Normal Quantiles
> qqplot(qq,sN,xlab="Theoretical Percentiles",ylab="Observed Standardized Nile",main="PP Plot of Nile")
> # Add the line $y=x$ for visual appeal.
> abline(0,1,col=2)
>
> # Same. This time "qqnorm" automatically generates standard normal quantiles
> qqnorm(sN,ylab="Observed Standardized Nile",main="PP Plot of Nile 2")
> abline(0,1,col=3)
>
> # This time plot non-standard observed quantiles vs. standard normal quantiles
> # Because it is a linear change of observed data, the graph has same shape.
> # "lm" computes the intercept and slope of the best fit line through the points.
> qqnorm(N,ylab="Observed Standardized Nile",main="PP Plot of Nile 3")
> abline(lm(N ~ qq),col=4)
>
> # Same. This time "qqline" plots the line through the quartile points.
> qqnorm(N,ylab="Observed Standardized Nile",main="PP Plot of Nile")
> qqline(N,col=5)
> #
> #
> # Notes on the four graphs:
> #
> # 1.) Except for the scales on the left, ALL FOUR HAVE THE SAME SHAPE.
> # This is because all use the same x-coordinates and the y-coordinates
> # for the lower graphs are linearly transformed versions of the upper.
> # The diagonals have been computed differently: x=y in the first two,
> # best fit line in the third and through the quartiles in the fourth.
> #
> # 2.) The data lines up pretty well with the diagonal. 100 observations is
> # a fairly small sample and one would expect a little wobble. There
> # is a slight "S" shape to the data, showing that the data may have
> # come from a distribution that is slightly "light tailed."

```

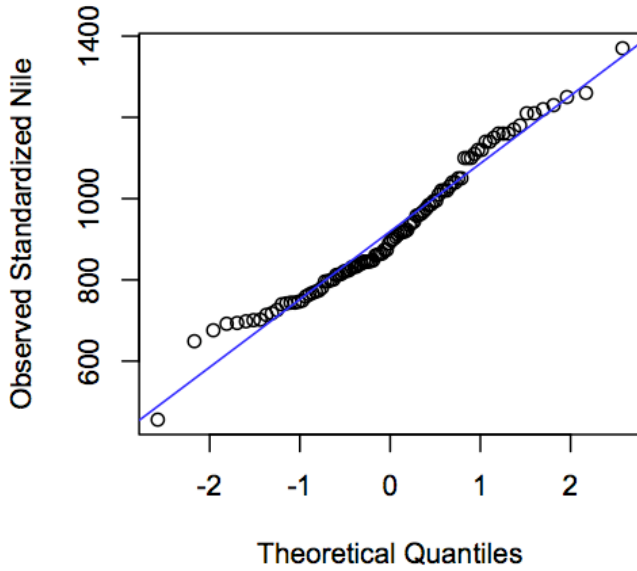
**PP Plot of Nile 1**



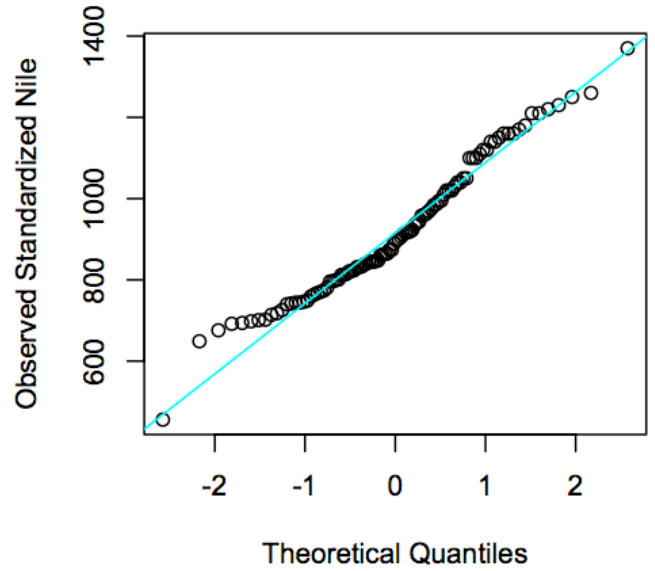
**PP Plot of Nile 2**



**PP Plot of Nile 3**



**PP Plot of Nile**



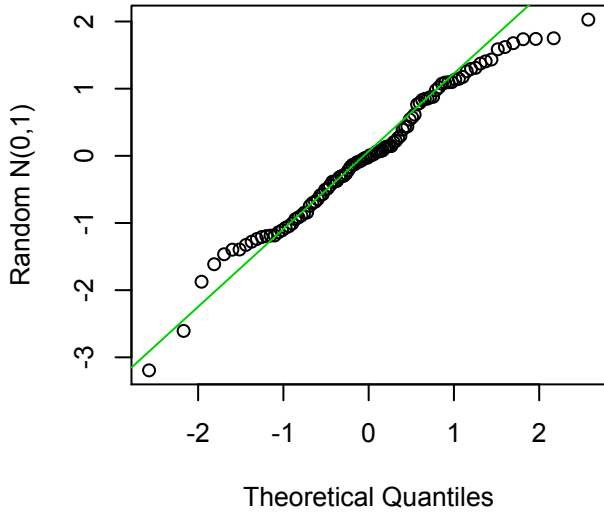
```

>
> ##### VARIABILITY IN RANDOM SAMPLES
> #
> # Since there is some variability, we don't expect the qqplots to line up exactly.
> # To see this, we generate four different samples of 100 random N(0,1) pts and qqplot.
> x<-rnorm(100);qqnorm(x,ylab="Random N(0,1)",main="Sample 1"); qqline(x,col=3)
> x<-rnorm(100);qqnorm(x,ylab="Random N(0,1)",main="Sample 2"); qqline(x,col=4)
> x<-rnorm(100);qqnorm(x,ylab="Random N(0,1)",main="Sample 3"); qqline(x,col=5)
> x<-rnorm(100);qqnorm(x,ylab="Random N(0,1)",main="Sample 4"); qqline(x,col=6)
> #
> #
> #
> # Notes on the four graphs:
> #
> # 1.) In this case, we have taken a different random sample for each graph.
> #      Each sample is 100 random numbers, coming from the same N(0,1)
> #      distribution. The variability is inherent in sampling: just by chance
> #      you get different observations so that the samples have different
> #      means and variances, as well as different QQ-plots.
> #
> # 2.) Each sample lines up pretty well with the diagonal. 100 observations is
> #      a fairly small sample and one would expect a little wobble. One might
> #      see a slight "r" shape in Sample 1, is a slight "N" shape to Sample 3,
> #      and a slight "S" shape to Sample 4. One may be misled by the quartile
> #      lines that tend to exaggerate the tail departure from linear. But
> #      the point is that one should not rule out normality from small trends
> #      since they may be due to random variation in the sample and not
> #      due to non-normality of the data.

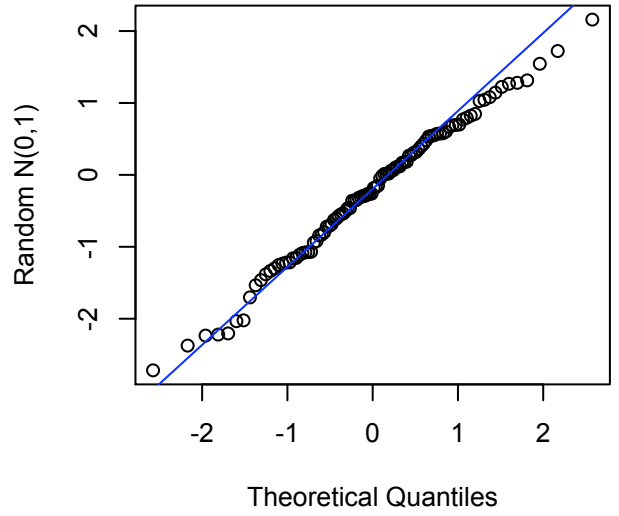
```



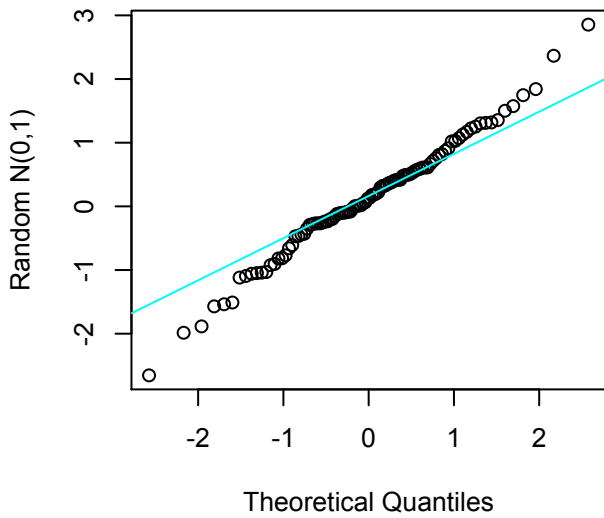
**Sample 1**



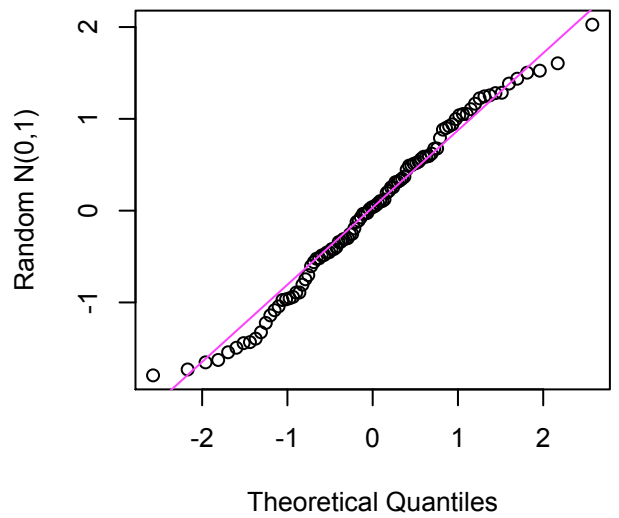
**Sample 2**



**Sample 3**



**Sample 4**

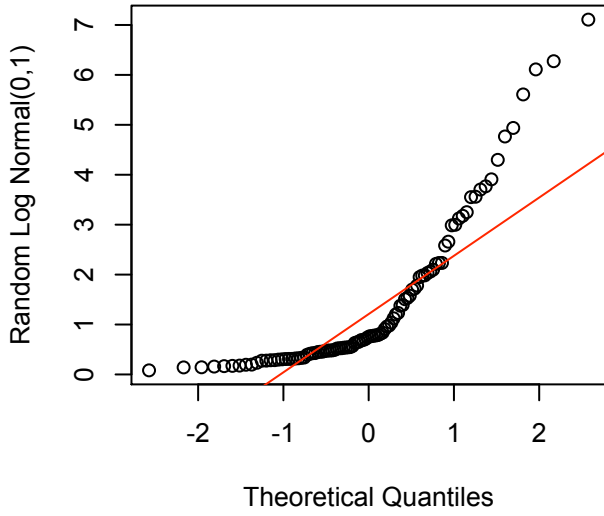


```

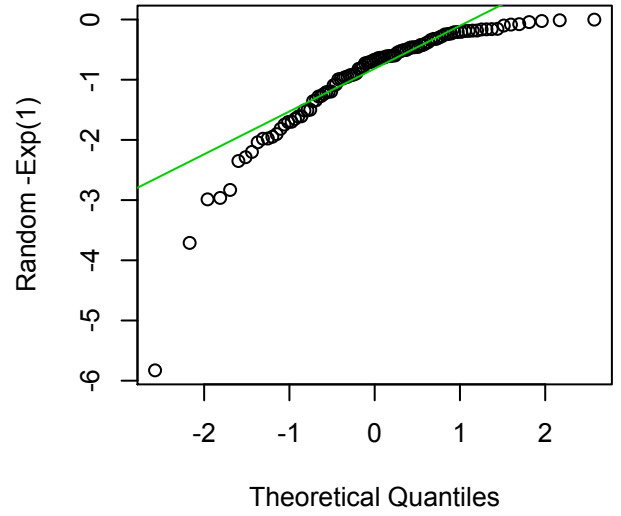
> ##### HOW NON-NORMAL SAMPLE WILL FAIL TO LINE UP.
> # Skewed right
> # Take standard lognormal variables. pdf is zero for negative X
> # but is greater than normal for X positive
> # "light left tail & heavy right tail."
> # The QQ-plot is "J" shaped.
> x<-rlnorm(100)
> qqnorm(x,ylab="Random Log Normal(0,1)",main="Skewed Right");qqline(x,col=2)
>
> #skewed left
> # Take  $X \sim -\text{Exp}(1)$ . pdf is zero for positive X but is greater than normal
> # for X negative "heavy left tail & light right tail."
> # The QQ-plot is "r" shaped.
> x<--rexp(100)
> qqnorm(x,ylab="Random -Exp(1)",main="Skewed Left");qqline(x,col=3)
>
>
> # light-tailed
> # Take  $X \sim \text{Beta}(.5,2)$ . pdf is zero for  $|X-.5|>.5$ . "light tailed."
> # The QQ-plot is "S" shaped.
> x<-rbeta(100,.5,2)
> qqnorm(x,ylab="Random Beta(.5,2)",main="Light Tailed");qqline(x,col=4)
>
> # heavy tailed
> x<-rt(100,df=3)
> # Take  $X \sim t(3)$ . pdf is greater than normal in tails "heavy tailed."
> # The QQ-plot is "N" shaped.
> qqnorm(x,ylab="Random t(3)",main="Heavy Tailed");qqline(x,col=7)
> #
> # Notes on the four graphs:
> #
> # 1.) This time, the departure from linearity is far more severe. Indeed,
> # the samples were chosen from non-normal distributions.
> #
> # 2.) The properties of the distributions reflect in the normal PP-plots.
> #
> # a.) Sample 1 is  $X \sim \text{lognormal}(0,1)$ . There are no negative X but
> # the pdf is greater than normal for X very positive so
> # "light left tail & heavy right tail." The qq-plot is "J" shaped:
> #
> # b.) Sample 2 is  $X \sim -\text{Exp}(1)$  (exponential for negative X and no
> # positive X.) The QQ-plot is "r" shaped. The pdf is zero for
> # positive X but is greater than normal for X negative.
> # "heavy left tail & light right tail." The QQ-plot is "r" shaped.
> #
> # c.) Sample 3 is  $X \sim \text{Beta}(.5,2)$ . pdf is zero for  $|X-.5|>.5$ .
> # "light tailed." The QQ-plot is "S" shaped.
> #
> # d.) Sample 4 is  $X \sim t(3)$ . pdf is greater than normal in tails
> # "heavy tailed." The QQ-plot is "N" shaped.

```

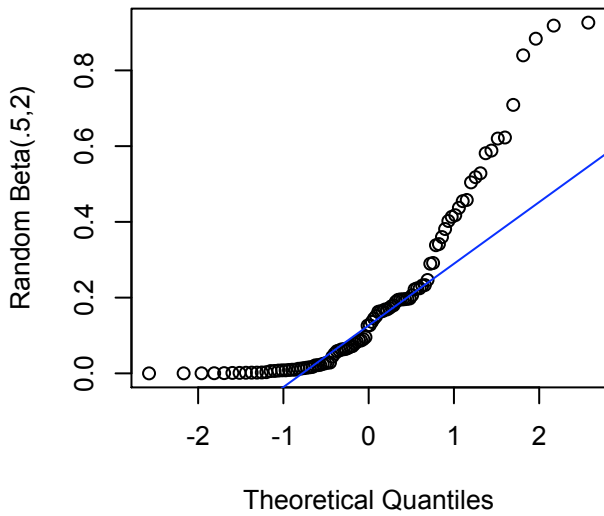
**Skewed Right**



**Skewed Left**



**Light Tailed**



**Heavy Tailed**

