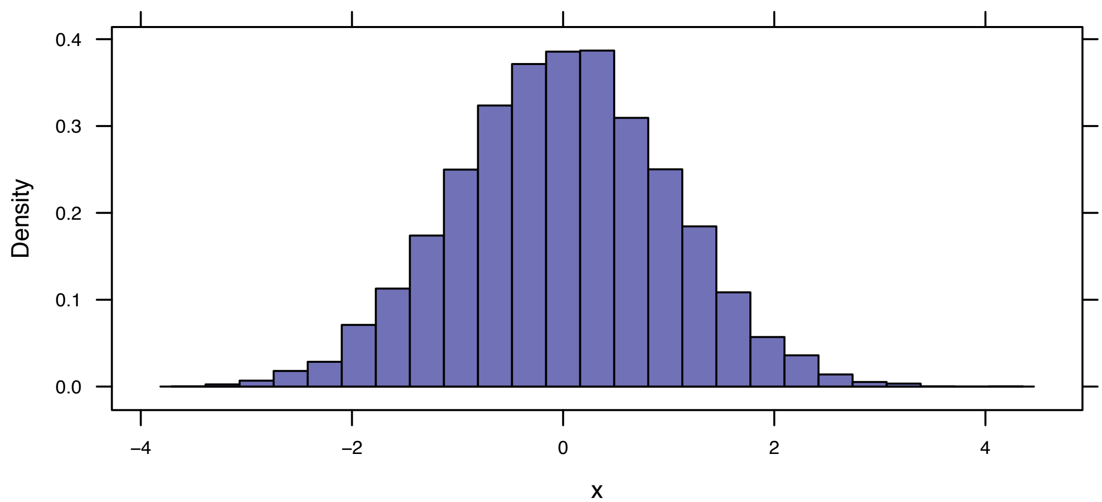
Quantile-Quantile Plots

An emerging question in the course: Whether data collected happens to be well-modeled by one of our standard statistical models. Since in many situations data seems normal-like, checking whether it this is really the case is where we begin this discussion.

Example 1: A large sample of values from a standard normal r.v.

Suppose we sample n (perhaps many thousands) of numbers from the standard normal distribution.

```
enn = 10000
x = rnorm(enn, mean=0, sd=1)
histogram(x, n=25)
```



If we count the number of values outside one, two, and three standard deviations from the mean, we should not be surprised we see the 68-95-99.7% rule in action.

```
sum( abs(x)<1 ) / enn
sum( abs(x)<2 ) / enn
sum( abs(x)<3 ) / enn
```

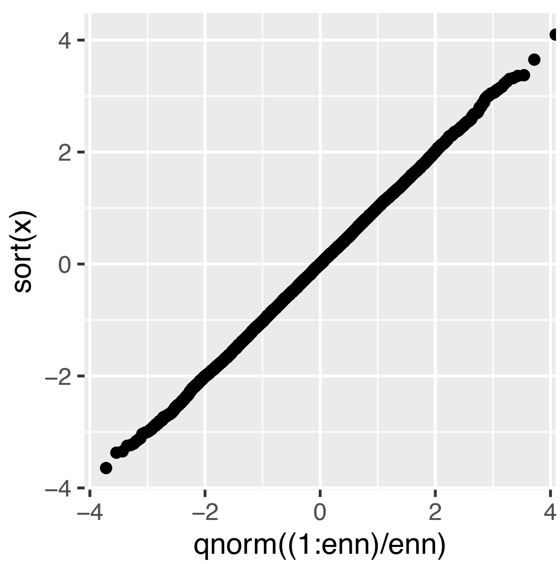
Likewise, if we were to sort the data and compare the j^{th} largest value with the (j/n)-quantile on a standard normal distribution, we would expect the two to be approximately the same.

```
sort(x)[50]     # 50th largest value

[1] -2.653773

qnorm(50/enn)     # the (50/10000)th quantile
```

```
[1] -2.575829

sort(x)[712]

[1] -1.50828

qnorm(712/enn)

[1] -1.466912
```

or

```
gf_point(sort(x) ~ qnorm((1:enn)/enn))
```



This scatterplot, taking the j^{th} largest sampled value as the ordinate with abscissa as the corresponding quantile on a standard normal distribution where the j^{th}, is known as a **normal quantile plot**. Not surprisingly, the points fall close to the line $y = x$.

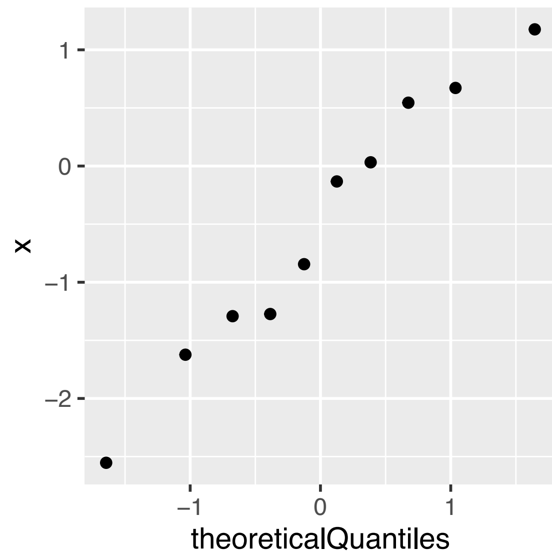
■

Example 2: A small sample of values from a standard normal r.v.

As before, assume we sample n values from a standard normal r.v., but with n small, say $n = 10$. It is not reasonable to expect the sorted list of sampled values to coincide with the 10^{th}, 20^{th}, ..., 100^{th} percentiles of a standard normal distribution; rather, we might expect them to be approximately the same as the 5^{th}, 15^{th}, 25^{th}, ..., 95^{th} theoretical percentiles.

```
x = sort( rnorm(10) ); x

[1] -2.55324655 -1.62342893 -1.29187435 -1.27341548 -0.84464012 -0.13276301  0.03189044
[8]  0.54474662  0.67178867  1.17586284
```

```
p.list = seq(0.05, 0.95, 0.1)
theoreticalQuantiles = qnorm(p.list)
gf_point(x ~ theoreticalQuantiles)
```



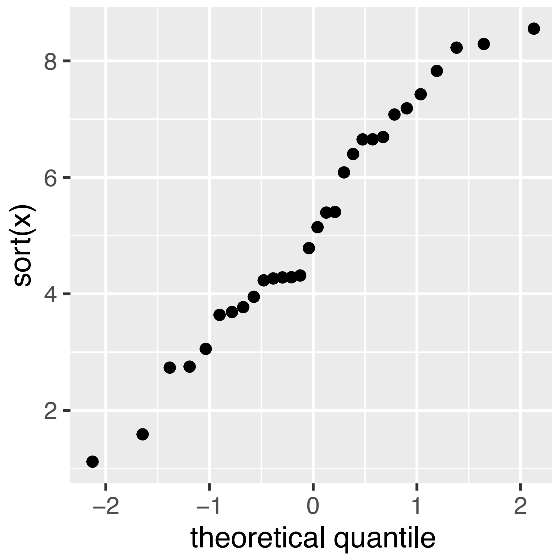
The match is not as close as when the sample was large (the points in the normal quantile plot are not as close to the line $y = x$), but that is consistent with the understanding that random behavior is unpredictable in the short term, patterned in the long term.

■

Example 3: Values sampled from $\text{Norm}(\mu, \sigma)$

Now suppose we sample our values from $X \sim \text{Norm}(\mu, \sigma)$. Since $X = \sigma Z + \mu$, where $Z \sim \text{Norm}(0, 1)$, it seems the same procedure—comparing the j^{th} largest sampled value with the (j/n)-quantile of a standard normal distribution—should be usable. The resulting plot, again called a **normal quantile plot**, should result in points that fall reasonably close to the line with slope σ and intercept μ.

```
enn = 30
x = rnorm(enn, 5, 2)
p.list = ppoints(enn)      # an improvement on ((1:enn)-.5)/enn
gf_point(sort(x) ~ qnorm(p.list, mean=0)) %>% gf_labs(x="theoretical quantile")
```

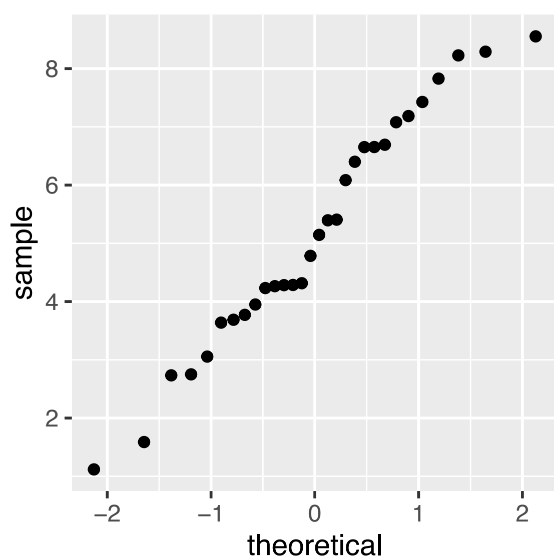


■

The procedure outlined in the previous examples can be used

- to check if it is reasonable to consider data from a process/source not known to be normal can reasonably be considered *normal*. The procedure was made easier with the introduction of the `ppoints()` command, but can be easier still using `gf_qq()`.

```
gf_qq(~x)      # only the data is required, and no sorting is necessary
```



- to check if data might be consistent with other distributional assumptions (statistical models). The plots one produces in such a context are called **quantile-quantile plots**. The next example explores this.

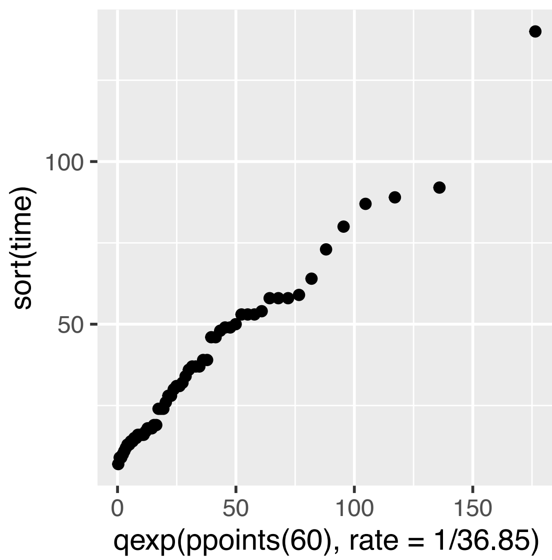
Example 4:

The file `http://scofield.site/teaching/data/csv/stob/scores.csv` contains time (in seconds) between scores in a basketball game played between Kalamazoo College and Calvin College on Feb. 7, 2003. Does it seem to follow an exponential distribution?

```
bball = read.csv("http://scofield.site/teaching/data/csv/stob/scores.csv")
favstats(~time, data=bball)

min Q1 median    Q3 max  mean       sd  n missing
  7 16     31 50.75 140 36.85 25.65734 60       0
```

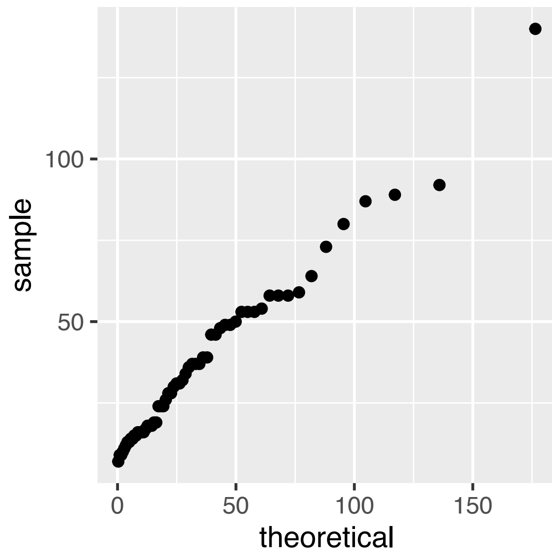
It is natural to assume the best comparison is between the measured times and the quantiles of $X \sim \text{Exp}(1/36.85)$, since this is the mean time between scores in the data set. By Example 3.3.9, p. 155, given any two exponential r.v.s X, Y, there is a scalar k such that $Y = kX$, which means that a quantile-quantile plot of values will look linear for all exponential distributions, or it will look linear for none of them. We illustrate this below:

```
gf_point(sort(time) ~ qexp(ppoints(60), rate=1/36.85), data=bball)
```



We may use `gf_qq()` for this sort of comparison, too, employing the `distribution` and `dparams` switches.

```
gf_qq(~time, data=bball, distribution = qexp, dparams=1/36.85)
```


100
50
sample
0
50
100
150
theoretical

■