Lab 09 - Sampling Distribution of Mean + Normality Testing with QQ Plots

Carl Gladish

Nov 4, 2025

This lab contains some instructional material along with some questions. You are required to submit your answers in a Microsoft Word (i.e., .docx) file produced from an R Notebook. This Word file will contain *all* your R code *and* your written answers and charts.

Use the R Markdown file Lab_09_Notebook.Rmd as a template to get started.

You will need to:

- adjust the author and date fields in the YAML metadata
- complete the missing R chunks
- type any written answers using R markdown formatting
- "knit" the result to .docx and submit your file to Learning Hub

Due date: 11:59pm, two school days from today (weekend days count as half)

Lab Objectives

The purpose of this lab is to:

- investigate how sample size n affects the sampling distribution of $\overline{X}$ when the underlying variable X is:
  - Normal, or
  - Exponential
- learn to create and interpret QQ plots for testing normality of a sample data set

Sampling Distribution of $\overline{X}$ for a Standard Normal Variable

Previously, we simulated sampling from a uniformly distributed population – e.g., dice rolls – and investigated the distribution of the sample mean, $\overline{X}$. This time, we will investigate the means of samples that are drawn from a *normally distributed* population.

Question 1

Create a function called `NormalMeans` that simulates taking `m.trials` samples of size `n.sample` from a normally distributed population with $\mu = 0$ and $\sigma = 1$. For each sample, compute and store the sample mean, $\overline{X}$. Your function should output:

- the grand mean of the `m.trials` sample means, which is an estimate of $\mu_{\overline{X}}$
- the standard deviation of the `m.trials` sample means, which is an estimate of $\sigma_{\overline{X}}$
- an appropriately labelled probability histogram of the `m.trials` sample means (use `breaks=100` to get 100 classes)

Run your function for `m.trials` $= 10^5$ and each of `n.sample` $= 1, 2, 10, 50, 100$.

For each `n.sample`, record the estimated value of $\mu_{\overline{X}}$ and $\sigma_{\overline{X}}$ and the histogram.

How do the means and standard deviations compare as `n.sample` increases?

How do the shapes of the probability histograms change as `n.sample` increases?

Sampling Distribution of the Mean for an Exponentially Distributed Variable

Now we will investigate the distributions of sample means of a very skewed population, one with an exponential distribution.

Question 2

Create a function called `ExponentialMeans` that simulates taking `m.trials` samples of size `n.sample` from an exponentially distributed population with mean $\beta = 1$. Your function should output the following:

- the grand mean of the `m.trials` sample means, which is an estimate of $\mu_{\overline{X}}$
- the standard deviation of the `m.trials` sample means, which is an estimate of $\sigma_{\overline{X}}$
- an appropriately labelled probability histogram of the `m.trials` sample means (use `breaks=100` to get 100 classes)

Run your function for `m.trials` $= 10^5$ and each of `n.sample` $= 1, 2, 10, 50, 100$.

For each `n.sample`, record the estimated value of $\mu_{\overline{X}}$ and $\sigma_{\overline{X}}$ and the histogram.

How do the means and standard deviations compare as `n.sample` increases?

How do the shapes of the probability histograms change as `n.sample` increases?

Question 3

If the sample mean $\overline{X}$ comes from samples of size n drawn from an exponential distribution, does $\overline{X}$ obey the Empirical Rule? To answer this question, write a function called `ExponentialMeansProb` that takes three arguments:

- `m.trials` (number of simulation trials)
- `beta` (the mean β)
- `n.sample` (sample size)
- `z` (Z – score)

The function should simulate and return the probability that

$$\mu - z \cdot \frac{\sigma}{\sqrt{n}} < \overline{X} < \mu + z \cdot \frac{\sigma}{\sqrt{n}}$$

(Recall that $\mu = \beta$ and $\sigma = \beta$ for an exponential distribution.)

Use `m.trials` $= 10^5$ and `beta` $= 1$. Complete the table below for the given values of `n.sample` and `z`.

Do your results agree with the Empirical Rule? Why or why not?

	`z` $= 1$	`z` $= 2$	`z` $= 3$
`n.sample` $= 1$			
`n.sample` $= 5$			
`n.sample` $= 50$			

Normality Testing

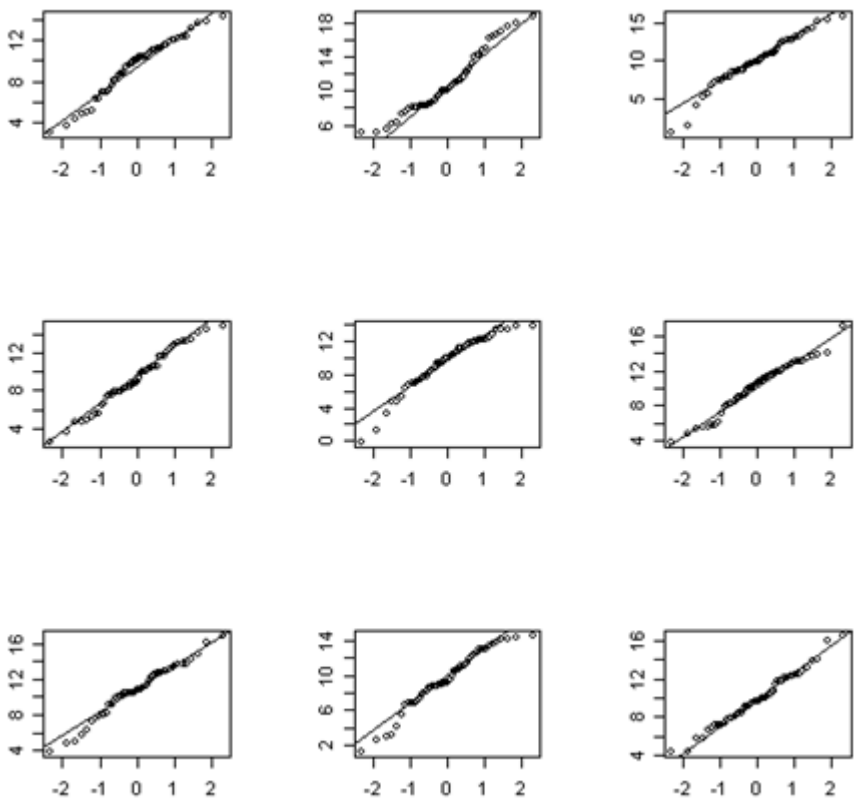
We have seen that the sample mean $\overline{X}$ is normally distributed if:

- the population follows a normal distribution, or
- the sample is *large enough* ($n \geq 30$)

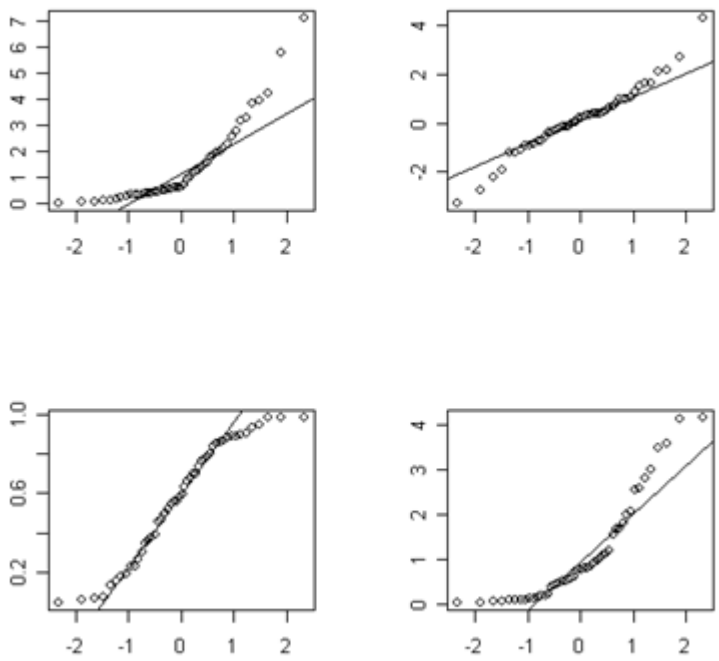
Soon you see that we can estimate population means based on sample means, but that some of our methods only work if the population is distributed normally. It follows that we need to be able to test whether a sample has come from a normally distributed population. One way to do that is by producing a *quantile-quantile plot* (QQ plot).

A QQ plot requires a minimum of ≈ 15 data points to accurately reveal normality. If the data come from a normal distribution, then the points will form a line with positive slope.

Examples of QQ plots for Samples from Normal Populations



Examples of QQ plots for Samples from Non-Normal Populations



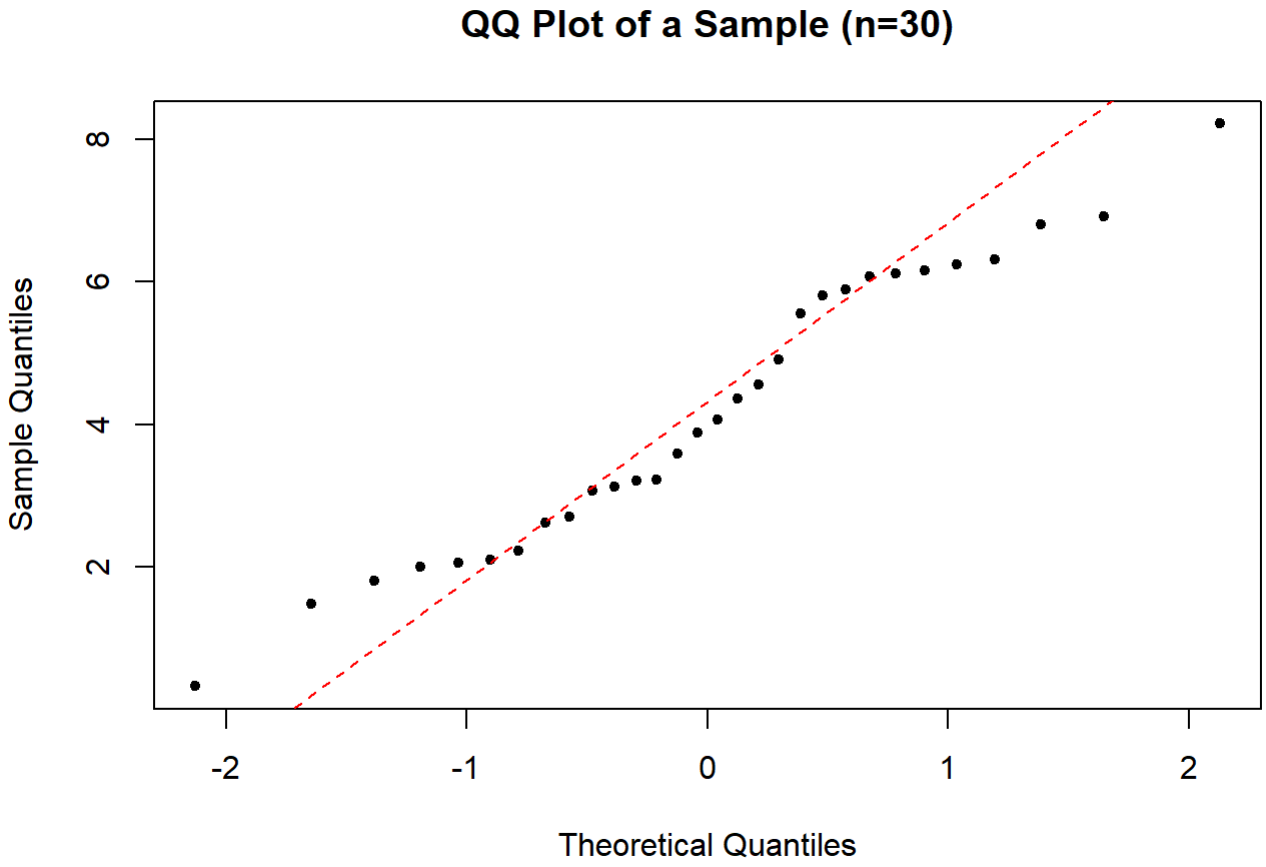
Quantile-Quantile Plots of Normal Data

We will begin by generating a sample that comes from a normal distribution using `rnorm`.

```
normal.data = rnorm(30, mean=4, sd=2)
```

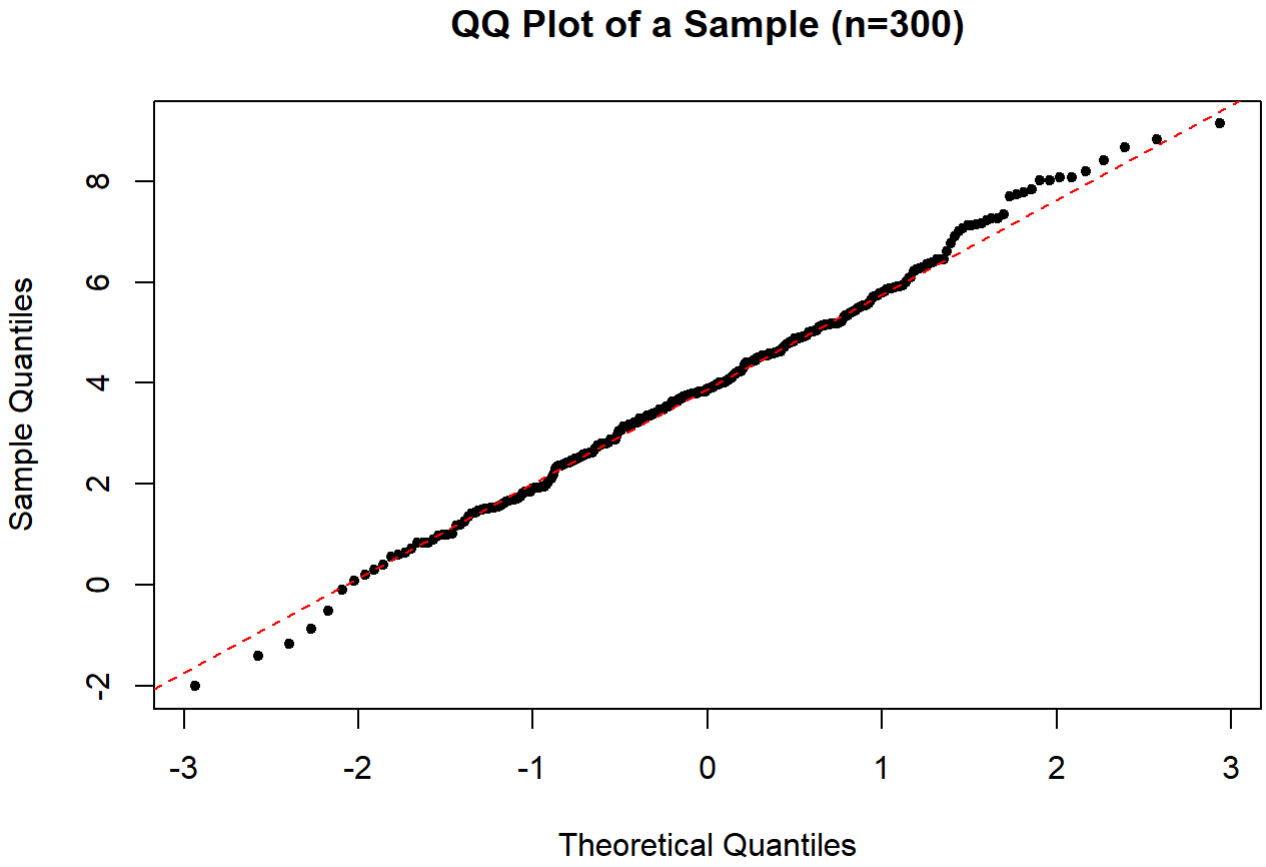
We then produce the QQ plot with the function `qqnorm`:

```
qqnorm(normal.data, pch=20, main="QQ Plot of a Sample (n=30)")
qqline(normal.data, col="red", lty="dashed")
```



The data points *roughly* follow a line. However, if we take a larger sample, the result is more clearly a straight line.

```
normal.data = rnorm(300, mean=4, sd=2)
qqnorm(normal.data, pch=20, main="QQ Plot of a Sample (n=300)")
qqline(normal.data, col="red", lty="dashed")
```



Note that the graph gets a bit irregular at the edges. This is typical, and it does not mean the sample is from a non-normal distribution (as we know in the present example).

QQ Plot Details

This section is meant to explain what is happening in a QQ plot.

If X is any normally distributed variable, then the various percentiles of X correspond to specific Z scores. For example, the 30th percentile value P_{30} must have a Z-score of:

```
qnorm(0.300)
```

```
## [1] -0.5244005
```

If we randomly sample 99 values of X (which we assume is normally distributed) and then sort those values in increasing order as $X_1, X_2, \ldots, X_{99}$, then we expect:

$X_1 \approx P_1$ which has a Z-score $\texttt{qnorm(0.01)} = -2.3263479$.

$X_2 \approx P_2$ which has a Z-score $\texttt{qnorm(0.02)} = -2.0537489$.

$X_{99} \approx P_{99}$ which has a Z-score $\texttt{qnorm(0.99)} = 2.3263479$.

Plotting the "Sample Quantile" $X_1, X_2, \ldots$ against the "Theoretical Quantiles" $Z_1, Z_2, \ldots$ would then give a straight line.

In general, if there are n sample values, which when sorted are $X_1, X_2, \ldots, X_n$, then we expect X_i to approximately equal the $\frac{i}{(n+1)} \times 100\%$ percentile value of X.

Therefore, if we plot the points:

$(X_1,$ `qnorm(1/(n+1))` $)$

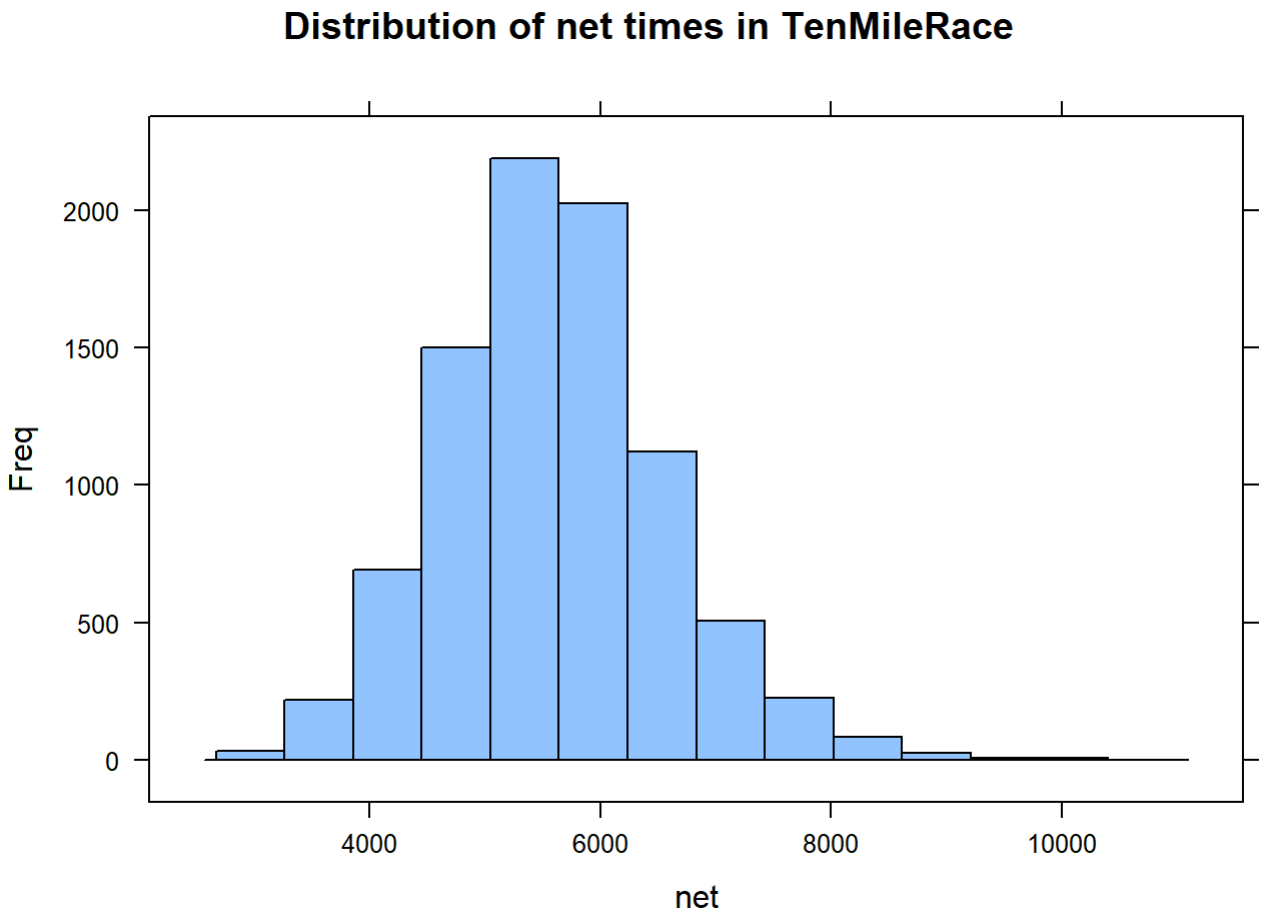
$(X_2,$ `qnorm(2/(n+1))` $)$

$\dots$

$(X_n,$ `qnorm(n/(n+1))` $)$

we should get a straight line (more-or-less), as long as X is a normally distributed variable.

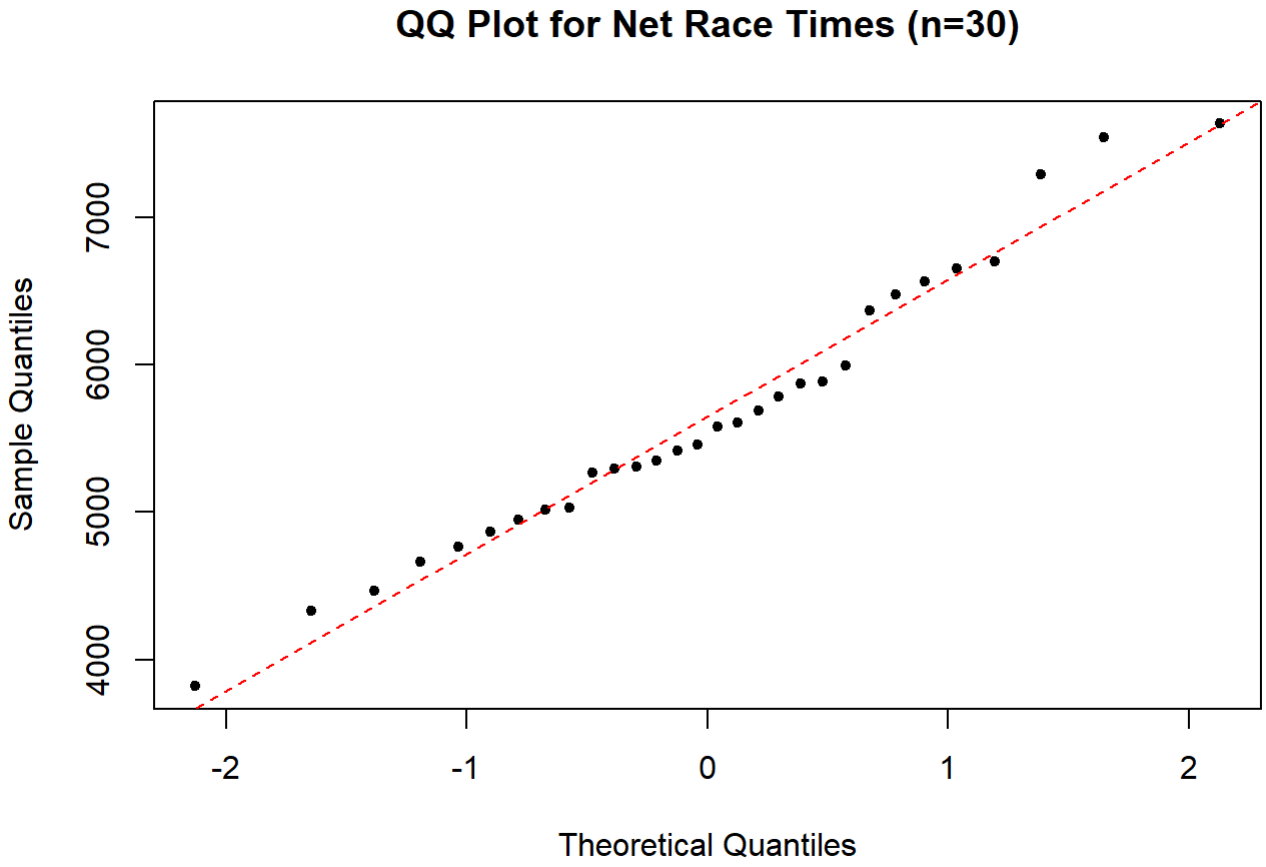
Example Let's try this with some real data. Recall the dataset `TenMileRace` from Lab 1. Load the `mosaicData` library and then load `TenMileRace`. Here is a histogram of net race times, which appears to follow a normal distribution:



In real life, we often don't have access to entire populations or even large samples. When our samples are small, a QQ plot gives us a clearer picture of the distribution than a histogram. Since the population of net times is normally distributed, a sample of times will probably be close to normally distributed as well.

To test this, generate a sample of $n = 30$ randomly selected net times and make a QQ plot.

```
race.times.sample <- TenMileRace$net[sample(1:nrow(TenMileRace),30)]
qqnorm(race.times.sample, main="QQ Plot for Net Race Times (n=30)", pch=20)
qqline(race.times.sample, lty="dashed",col="red")
```



The fact that this is close to a straight line means that the sample (and therefore the population) of race times is normally distributed.

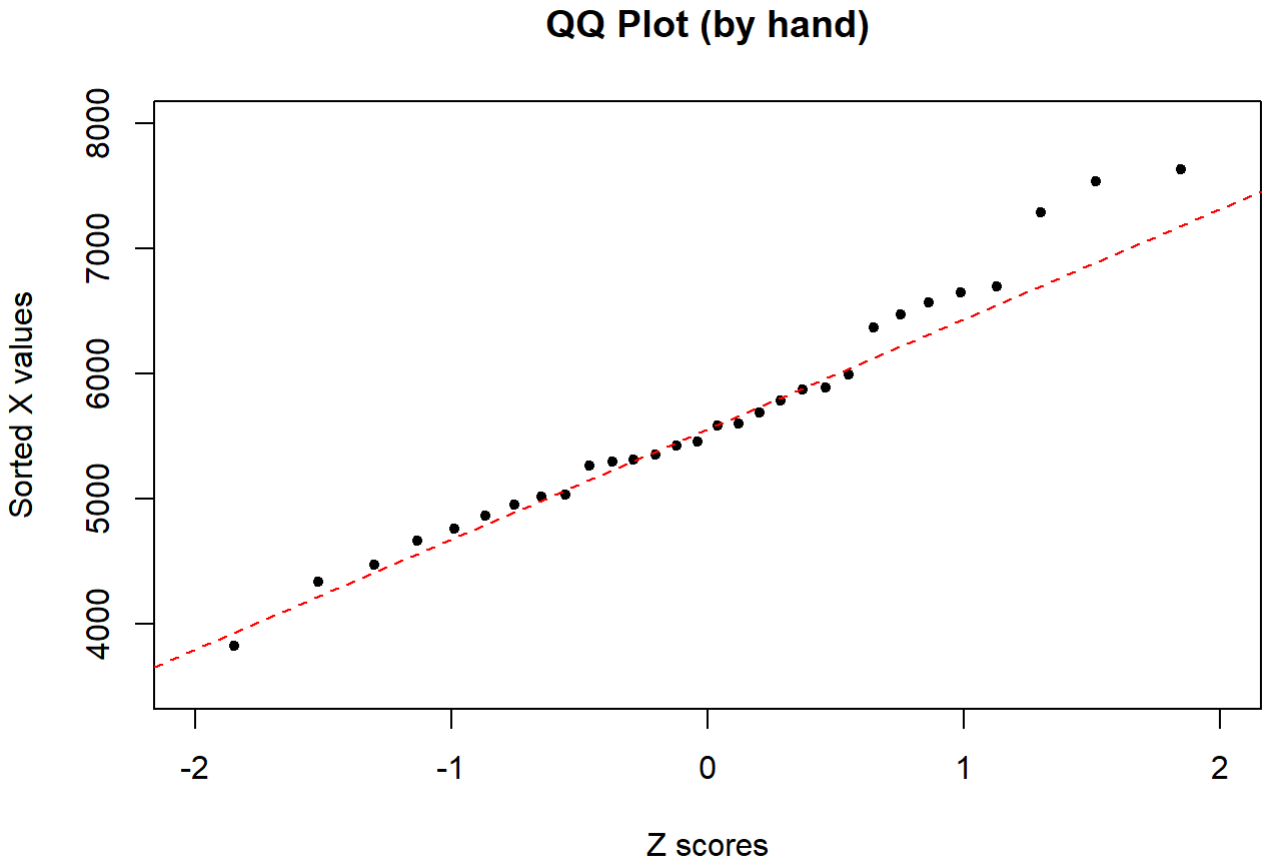
(NOTE: If you run this code, your result will differ. It may even appear that your sample is non-normal. In statistics, we must live with limited confidence in our results.)

Side Note We can produce the above QQ plot ourselves using

```
X.vals <- race.times.sample
n <- length(X.vals)
X.quantiles <- sort(X.vals)

Z.vals <- qnorm( (1:n)/(n+1) )
plot(Z.vals, X.quantiles, main="QQ Plot (by hand)", xlab="Z scores",
     ylab="Sorted X values", pch=20,
     xlim=c(-2.0, 2.0), ylim=c(3500, 8000))

use.points <- pnorm(Z.vals) >= 0.25 &  pnorm(Z.vals) <= 0.75
model <- lm(X.quantiles[use.points]~Z.vals[use.points])
abline(model, col="red", lty="dashed")
```



Question 4

Load the `genotype` dataframe, which is part of the `MASS` library. Take a minute to read over its help file.

Filter your data to obtain four data sets, grouped by `Litter` (genotype). Note that these data sets are small, so we require them to be normally distributed if we wish to find a confidence interval. Create QQ plots for `weight` for each genotype.

Do the four samples appear to be normal?