CLASSICAL INFERENCE: CONFIDENCE INTERVALS

7.1.1 Confidence Intervals for a Mean, σ Known

Example 7.1 The Centers for Disease Control maintains growth charts for infants and children (http://cdc.gov/growthcharts/zscore.html). For 13-year-old girls, the mean weight is 101 pounds with a standard deviation of 24.6 pounds. We assume the weights are normally distributed. The public health officials in Sodor are interested in the weights of the teens in their town: they suspect that the mean weight of their girls might be different from the mean weight in the growth chart but are willing to assume that the variation is the same. If they survey a random sample of 150 thirteen-year-old girls and find their mean weight – an estimate of the population mean weight – is 95 pounds, how accurate this estimate be?

We assume the 150 sample values are from a normal distribution, $N(\mu, 24.6)$. Then the sampling distribution of mean weights is $N(\mu, 24.6/\sqrt{150})$. Let $\bar{X}$ denote the mean of the 150 weights, so standardizing gives $Z = (\bar{X}\breve{}\mu)/(24.6/\sqrt{150}) \sim N(0,1)$. For a standard normal random variable Z, we have

$$P(z_{\alpha/2} < Z < z_{1-\alpha/2})$$

The random interval $(\bar{X} - 3.937, \bar{X} + 3.937)$ has a probability of 0.95 of containing the mean μ. Now, once you drive your sample, the random variable $\bar{X}$ is replaced by the (observed) sample mean weight of $\bar{x} = 95$, and the interval (91.1, 98.9) is no longer a random interval. We interpret this interval by stating that we are 95% confident that the population mean weight of 13-year-old girls Sodor is between 91.9 in 98.9 pounds.

Remark

- The trick of doing algebra on equations that are inside of probability is often handy.
- Be careful when reading an equation such as $0.95 = P(\bar{X} - 3.937 < \mu < \bar{X} + 3.937)$. this does not mean that μ is random, with a 95% probability of falling between two values. The parameter μ is an unknown constant. Instead, it is the interval that is random, with a 95% probability of including μ. In the previous example, we computed a confidence interval of (91.1, 98.9). We should not attribute a probability to this interval: either the true meaning is in this interval or it is not! The statement "we are 95% confident" means that if we repeated the same process of drawing samples and computing intervals many times, then in the long run, 95% of the intervals would include μ.

More generally, for a sample of size n drawn from a normal distribution with unknown μ and known σ, a $(1-\alpha) \times 100\%$ confidence interval for the mean μ is

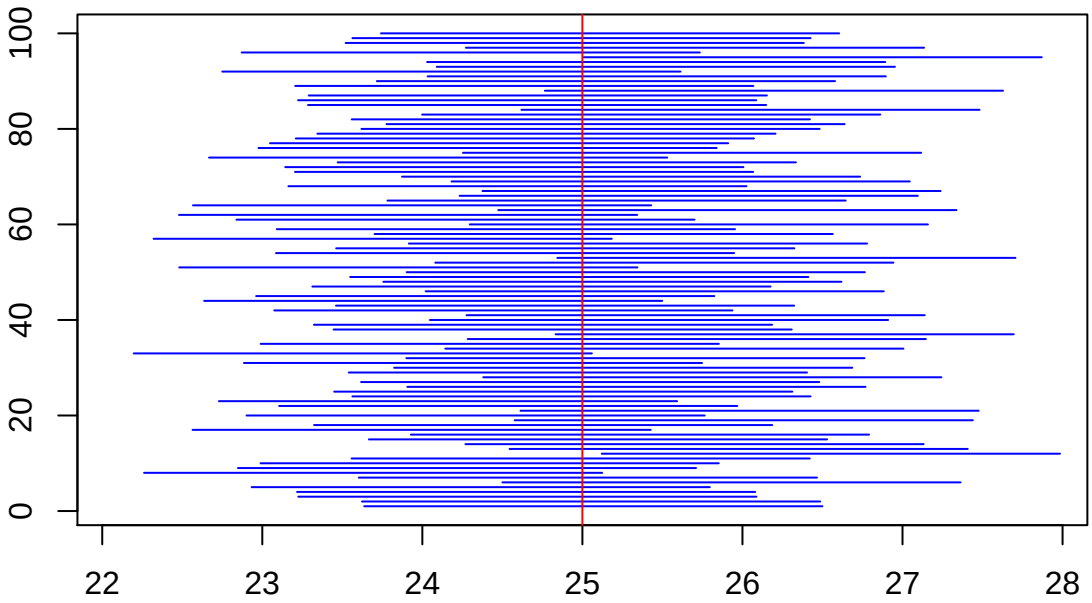
$$CI_{1-\alpha}(\mu) = \left(\bar{X} + z_{\alpha/2}\sigma/\sqrt{n}, \bar{X} + z_{1-\alpha/2}\sigma/\sqrt{n}\right)$$

If we draw thousands of random samples from a normal distribution with parameters μ, σ and compute the 95% confidence interval for each sample, then about 95% of the intervals would contain μ.

We illustrate this with the simulation by drawing random samples of size 30 from a $N(25, 4)$. For each sample, we construct a 95% confidence interval and check to see whether it contains $\mu = 25$. We do this 10,000 times and keep track of the number of times that the interval contains μ. For good measure, we will graph some of the random intervals.

```
set.seed(13)
counter <- 0 # set counter to 0
mu <- 25
sigma <- 4
n <- 30
sims <- 10^4
plot(x = c(mu - 4*sigma/sqrt(n), mu + 4*sigma/sqrt(n)), y = c(1, 100), type = "n", xlab = "", ylab = ""
for (i in 1:sims){
```

```
 x <- rnorm(n, mu, sigma)
 L <- mean(x) + qnorm(.025)*sigma/sqrt(n)
 U <- mean(x) + qnorm(0.975)*sigma/sqrt(n)
 if(L < mu && mu < U){counter <- counter + 1}
 if(i <= 100){
 segments(L, i, U, i, col = "blue")
 }
}
abline(v = mu, col = "red")
```




```
ACL <- counter/sims*100
ACL
```

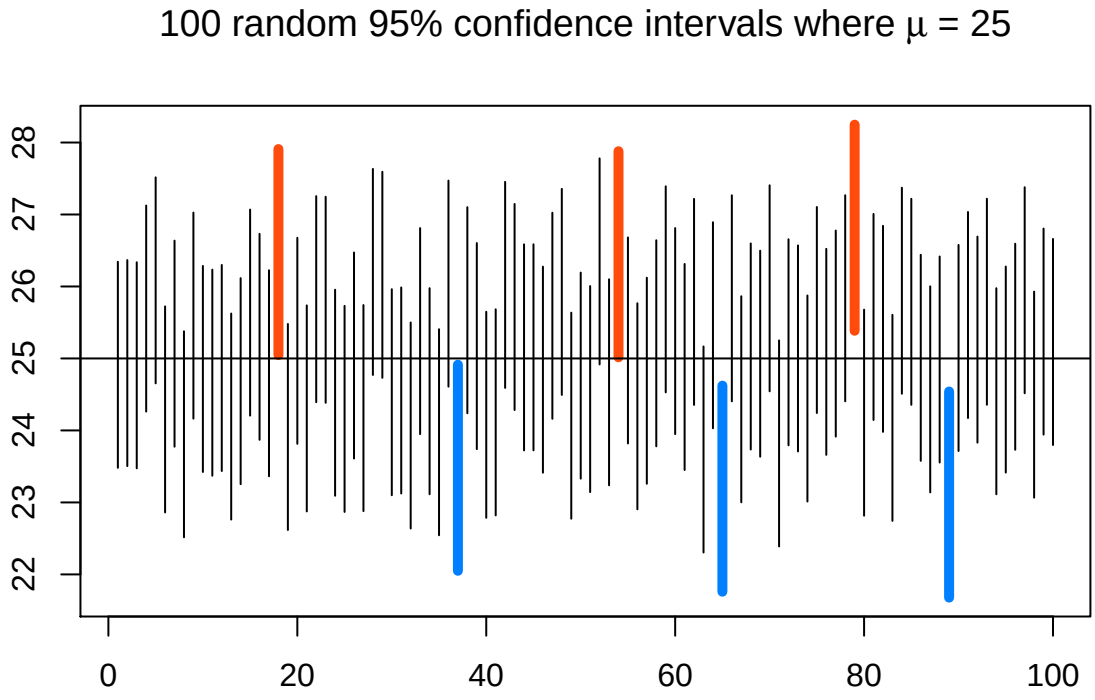
```
## [1] 95.08
```

The function `CIsim()` from the `PASWR` package is used to run a similar simulation below.

```
require(PASWR)
```

```
## Loading required package: PASWR
## Loading required package: e1071
## Loading required package: MASS
## Loading required package: lattice
```

```
set.seed(11)
CIsim(samples = 100, n = 30, parameter = 25, sigma = 4, type = "Mean")
```



Note: 6% of the random confidence intervals do not contain μ=25

```
## 6 % of the random confidence intervals do not contain Mu = 25 .
```

Example 7.2 An engineer test the gas mileage of a random sample of $n = 30$ of his company cars ready to be sold. The 95% confidence interval for the mean mileage of all the cars is (29.5, 33.4) miles per gallon. Evaluate the following statements:

1. We are 95% confident that the gas mileage for cars in this company is between 25.5 and 33.4 mpg.
2. 95% of all samples will give an average mileage between 29.5 and 33.4 mpg.
3. There is a 95% chance that the true mean is between 29.5 and 33.4 mpg.

Solution

1. This is not correct: a confidence interval is for a population parameter, and in this case the mean, not for individuals.
2. This is not correct: each sample will give rise to a different confidence interval and 95% of these intervals will contain the true mean.
3. This is not correct: μ is not random. The probability that it is between 29.5 and 33.4 is 0 or 1.

In our first example, we constructed a 95% confidence interval, but we can use other levels of confidence more generally let q denote the $(1 - \alpha/2)$ quantile that satisfies $P(Z < q) = 1 - \alpha/2$. Then, by symmetry, $P(-q < Z < q) = 1 - \alpha$. We will represent the quantile q in a standard normal distribution with the notation $z_{1-\alpha/2}$. For example, $z_{.975} = 1.959964$. Let $\bar{X}$ denote the mean of a random sample of size n from a normal distribution $N(\mu, \sigma)$. Then, since $\bar{X} \sim N(\mu, \sigma/\sqrt{n})$, by mimicking the previous algebra, we have

$$1 - \alpha = P(z_{\alpha/2} < Z < z_{1-\alpha/2}) \tag{1}$$

$$= P\left(\bar{X} - z_{1-\alpha/2}\sigma/\sqrt{n}, \bar{X} + z_{1-\alpha/2}\sigma/\sqrt{n}\right) \tag{2}$$

Example 7.3 Suppose the sample 3.4, 2.9, 2.8, 5.1, 6.3, 3.9 is drawn from the normal distribution with unknown μ and known $\sigma = 2.5$. Find a 90% confidence interval for μ.

Solution The mean of the six numbers is 4.0666667. Here, $1-\alpha = 0.90$, so $\alpha/2 = 0.05$. Thus, $z_{1-\alpha/2} = z_{0.95} = 1.6448536$. The 90% confidence interval is

$$\left(4.0667 - 1.6449 \times \frac{2.5}{\sqrt{6}}, 4.0667 + 1.6449 \times \frac{2.5}{\sqrt{6}}\right)$$

.

We are 90% confident that the population mean lies in the interval (2.387895, 5.7454384).

```
xs <- c(3.4, 2.9, 2.8, 5.1, 6.3, 3.9)
n <- length(xs)
SIGMA <- 2.5
alpha <- 0.10
LL <- mean(xs) - qnorm(1 - alpha/2)*SIGMA/sqrt(n)
UL <- mean(xs) + qnorm(1 - alpha/2)*SIGMA/sqrt(n)
CI <- c(LL, UL)
CI
```

```
## [1] 2.387895 5.745438
```

```
require(PASWR)  # or use z.test() from PASWR
z.test(x = xs, sigma.x = SIGMA, conf.level = 0.90)$conf
```

```
## [1] 2.387895 5.745438
## attr(,"conf.level")
## [1] 0.9
```

The term $z_{1-\alpha/2} \times \sigma/\sqrt{n}$ is called the *margin of error* (we abbreviate this as ME).

MARGIN OF ERROR: The margin of error for a symmetric confidence interval is the distance from the estimate to either end. The confidence interval is of the following form: estimate $\pm$ ME.

Example 7.4 Suppose researchers want to estimate the mean weight of girls in Sodor. They assume that the distribution of weights is normal with unknown mean μ, but with known $\sigma = 24.6$. How many girls should they sample if they want, with 95% confidence, their margin of error to be at most 5 pounds?

Solution Since $z_{1-\alpha/2} = z_{0.975} = 1.959964$, we set $1.96(24.6/\sqrt{n}) \leq 5$. This leads to $n \geq 92.9878888$, so there should be at least 93 girls in the sample.

```
n <- ceiling((qnorm(.975)*24.6/5)^2)
n
```

```
## [1] 93
```

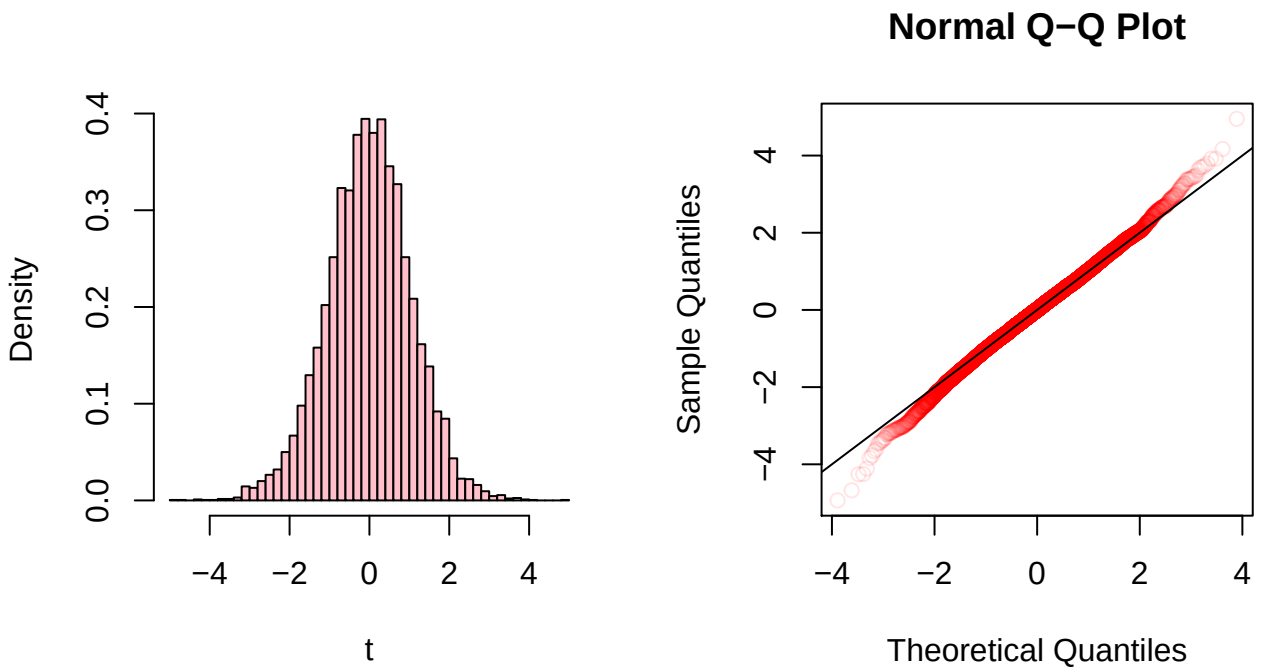
Remark Note that with the with of the interval, given by $z_{1-\alpha/2} \times (\sigma/\sqrt{n})$, depends on the level of confidence (which determines $z_{1-\alpha/2}$), the standard deviation, and the sample size. Analysts cannot control σ, but can adjust $z_{1-\alpha/2}$ or n. To make the confidence interval narrower, they can either increase the sample size n or decrease the size of the quantile $z_{1-\alpha/2}$, which amounts to decreasing the confidence level.

7.1.2 Confidence Intervals for a Mean, σ Unknown

In most real-life settings, a data analyst will not know the mean or the standard deviation of the population of interest. How then would we get an interval estimate of the mean μ? we have used the sample mean $\bar{X}$ as an estimate of μ, so it seems natural to consider the sample standard deviation S as an estimate of σ.

However, in deriving the confidence interval for μ, we used the fact that $(\bar{X} - \mu)/(\sigma/\sqrt{n})$ follows a standard normal distribution. Does changing σ to S, the sample standard deviation, change the distribution? Who uses simulation to investigate the distribution of $(\bar{X} - \mu)/(\sigma/\sqrt{n})$ for random samples drawn from a $N(\mu, \sigma)$.

```
set.seed(1)
N <- 10^4
TS <- numeric(N)
n <- 16
for(i in 1:N){
  x <- rnorm(n, 25, 7)
  xbar <- mean(x)
  s <- sd(x)
  TS[i] <- (xbar - 25)/(s/sqrt(n))
}
par(mfrow=c(1, 2))
hist(TS, breaks = "Scott", freq = FALSE, col = "pink", main = "", xlab = expression(t))
qqnorm(TS, col = rgb(1, 0, 0, .1))
abline(a = 0, b = 1)
```



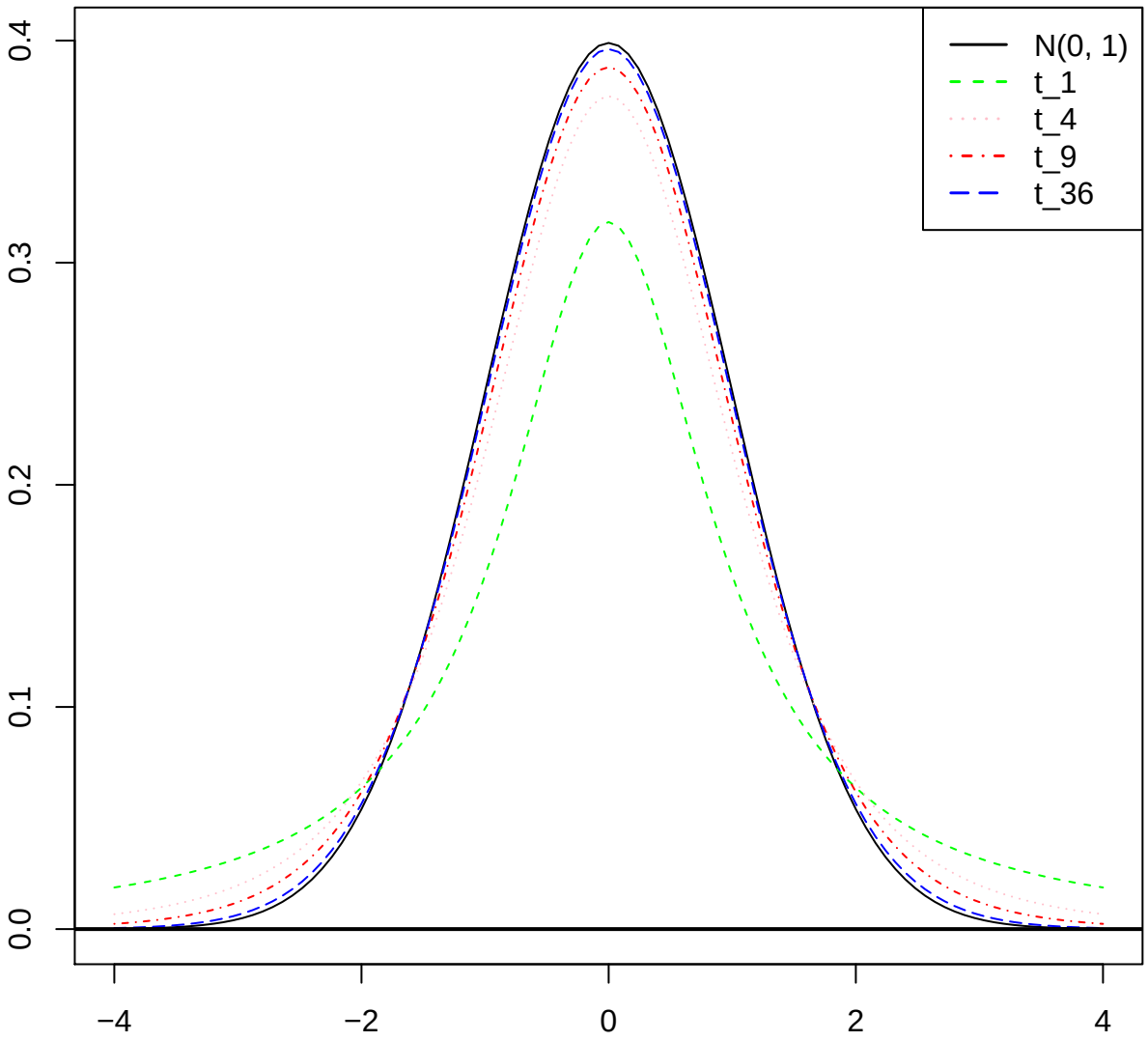

```
par(mfrow=c(1, 1))
```

This distribution does have slightly longer tails than the normal distribution; you could never tell this from a histogram, but it is apparent in the normal quantile plot. Sometimes S is smaller than σ, and when the denominator is small, the ratio is large. In effect, having to estimate σ using S adds variability.

It turns out that $T = (\bar{X} - \mu)/(S/\sqrt{n})$ has a Students t distribution with $n - 1$ degrees of freedom.

The density of a t distribution with k degrees of freedom is bell shaped and symmetric about 0, with heavier (longer) tails than that of the standard normal. As k tends toward infinity, the density of the t distribution tends toward the density of the standard normal.

```
curve(dnorm(x, 0, 1), -4, 4, col = "black", ylab = "", xlab = "")
curve(dt(x, 1), add = TRUE, lty = 2, col = "green")
curve(dt(x, 4), add = TRUE, lty = 3, col = "pink")
curve(dt(x, 9), add = TRUE, lty = 4, col = "red")
curve(dt(x, 36), add = TRUE, lty = 5, col = "blue")
abline(h = 0, lwd=2)
legend("topright", legend = c("N(0, 1)", "t_1", "t_4", "t_9", "t_36"), lty = c(1, 2, 3, 4, 5), col =c("
```



We derive the confidence interval for μ when σ is unknown in the same way as when σ is known. Let $t_{1-\alpha/2;n-1}$ denote the $(1-\alpha/2)$ quantile of the t distribution with $n-1$ degrees of freedom, $P(T_{n-1} < t_{1-\alpha/2;n-1}) = 1-\alpha/2, 0 < \alpha < 1$. Then using symmetry of the t distribution, we have

$$1-\alpha = P\left(-t_{1-\alpha/2;n-1} < \frac{\bar{X}-\mu}{S/\sqrt{n}} < t_{1-\alpha/2;n-1}\right) \tag{3}$$

$$= P\left(\bar{X} - t_{1-\alpha/2;n-1} \times \frac{S}{\sqrt{n}} < \mu < \bar{X} + t_{1-\alpha/2;n-1} \times \frac{S}{\sqrt{n}}\right) \tag{4}$$

T CONFIDENCE INTERVAL FOR NORMAL MEAN WITH UNKNOWN STANDARD DEVIATION If $X_i \sim N(\mu, \sigma), i = 1, \ldots, n$, with σ unknown, then a $(1-\alpha)\times 100\%$ confidence interval for μ is given by

$$CI_{1-\alpha}(\mu) = \left(\bar{X} - t_{1-\alpha/2;n-1} \times \frac{S}{\sqrt{n}}, \bar{X} + t_{1-\alpha/2;n-1} \times \frac{S}{\sqrt{n}}\right)$$

Example 7.5 The distribution of weights of boys in Sodor is normal with unknown mean μ. From a random sample of 28 boys, we find a sample mean of 110 pounds and a sample standard deviation of 7.5 pounds. To

compute a 90% confidence interval, find the 0.95 quantile of the t distribution with 27 degrees of freedom, which is 1.7032884 using the command `qt(0.95, 27)`. The interval is $(110-1.7033*7.5/\sqrt{28}, 110+1.7033*7.5/\sqrt{28})$; thus we are 90% confident that the true mean weight is between 107.6 and 112.4 pounds.

R Note:

The command `pt` or `qt` give probabilities or quantiles, respectively, for the t distribution. For instance, to find $P(T_{27} < 2.8)$ for the random variable T_{27} from a t distribution with 27 degrees of freedom,

```
pt(2.8, 27)
```

```
## [1] 0.9953376
```

To find the quantile $t_{.95;27}$ satisfying $P(T_{27} < t_{.95;27}) = 0.95$,

```
qt(.95, 27)
```

```
## [1] 1.703288
```

```
#
require(PASWR)
tsum.test(mean.x = 110, s.x = 7.5, n.x = 28, conf.level = 0.90)$conf
```

```
## [1] 107.5858 112.4142
## attr(,"conf.level")
## [1] 0.9
```

Compare the 0.95 quantile for a t distribution with 27 degrees of freedom with that of the standard normal: 1.7032884 versus 1.6448536. Thus, the t interval is slightly wider than the z interval, reflecting, as we noted previously, the extra uncertainty in not knowing the true σ.

Example 7.6 Find a 99% confidence interval for the mean weight of baby girls born in North Carolina in 2004.

Solution The mean and standard deviation of the weights of $n = 521$ girls is 3398.3166987 and 485.6911149g, respectively. A normal quantile plot shows that the weights are approximately normally distributed, so a t interval is reasonable.

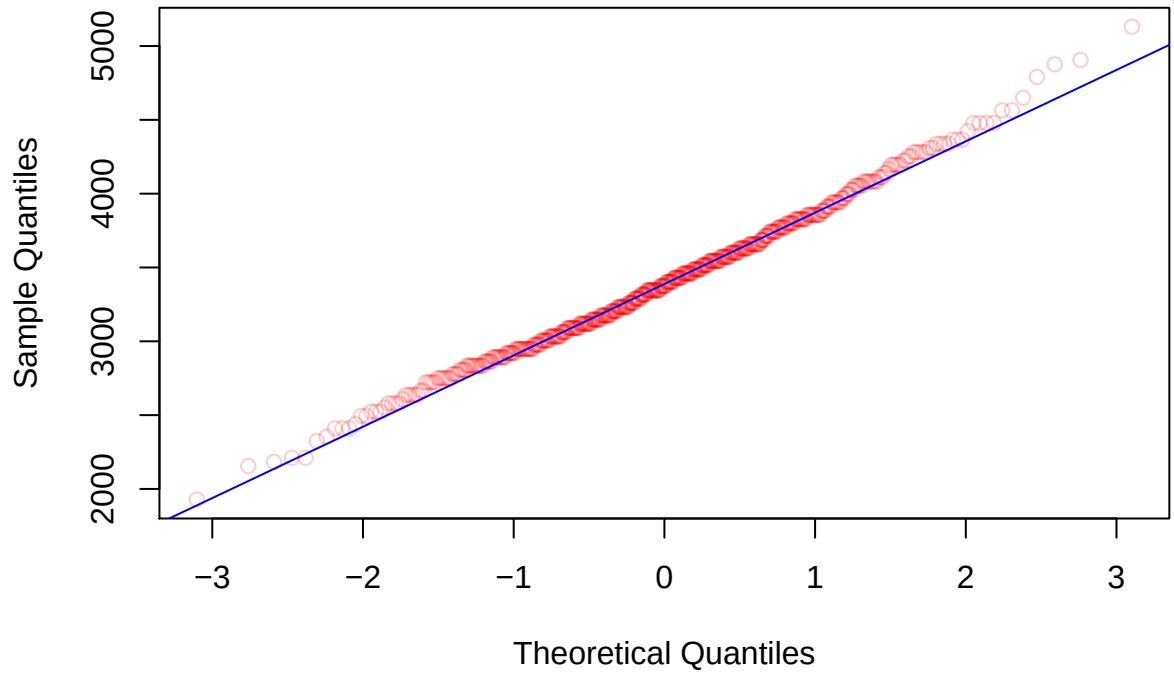
```
site <- "http://www1.appstate.edu/~arnholta/Data/NCBirths2004.csv"
NCBirths2004 <- read.csv(file=url(site))
MEANS <- tapply(NCBirths2004$Weight, NCBirths2004$Gender, mean)
SD <- tapply(NCBirths2004$Weight, NCBirths2004$Gender, sd)
MEANS
```

```
##   Female     Male
## 3398.317 3501.580
```

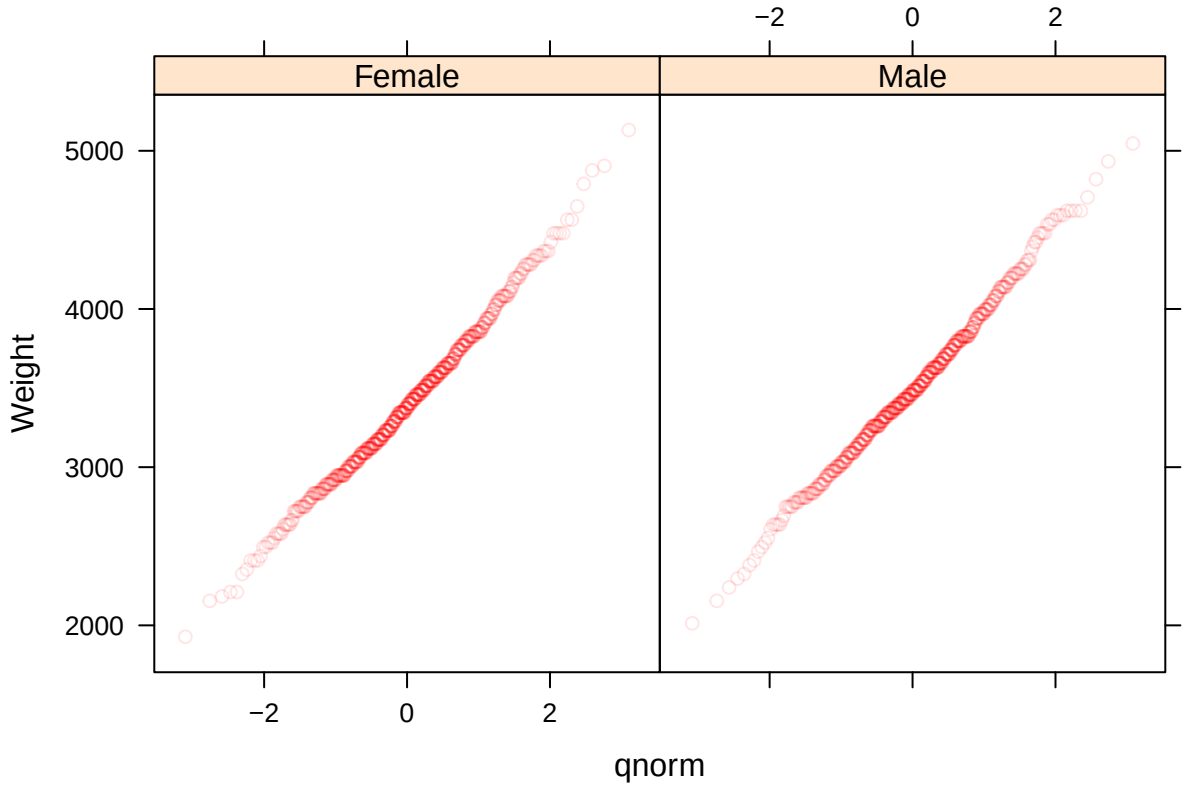
```
SD
```

```
##   Female     Male
## 485.6911 484.7505
```

```
qqnorm(NCBirths2004$Weight[NCBirths2004$Gender=="Female"], main = "", col = rgb(1,0,0,.2))
qqline(NCBirths2004$Weight[NCBirths2004$Gender=="Female"], col = "blue")
```



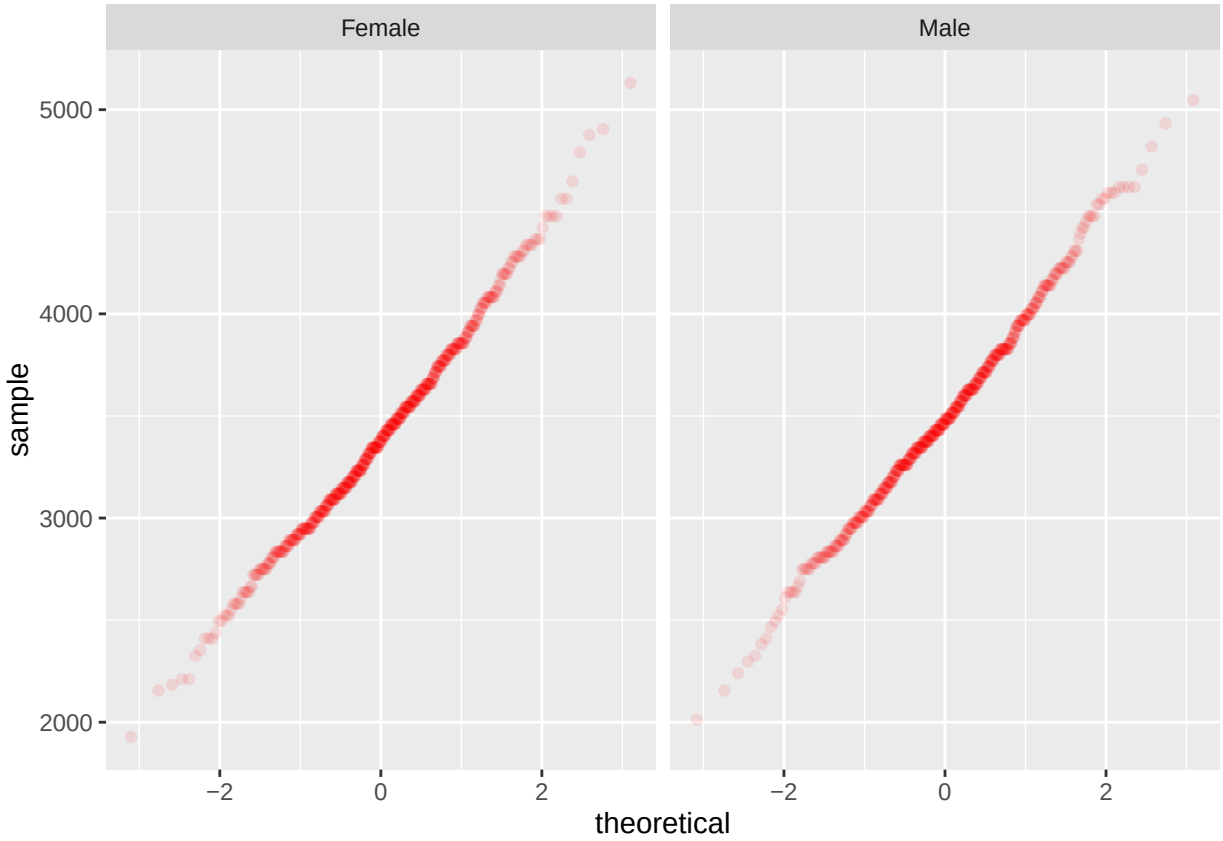

```
require(lattice)
qqmath(~Weight|Gender,data = NCBirths2004, col = rgb(1,0,0,.1))
```




```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
ggplot(data = NCBirths2004, aes(sample = Weight)) + stat_qq(color = rgb(1,0,0,.1)) + facet_grid(.~Gende
```



Since $1-\alpha = 0.99$, it follows that $\alpha/2 = 0.005$. The 0.995 quantile for the t distribution with 520 degrees of freedom is $t_{0.995;520} = 2.585317$. Thus, the 99% confidence interval is $3398.3166987 \pm 2.585317 \times 485.6911149/\sqrt{521} = (3343.3049953, 3453.328402)$ g.

R Note:

Use the command `t.test()` to find confidence intervals.

```
girls <- subset(NCBirths2004, select = Weight, subset = Gender =="Female", drop = TRUE)
t.test(girls, conf.level = 0.99)$conf
```

```
## [1] 3343.305 3453.328
## attr(,"conf.level")
## [1] 0.99
```

```
# Or
t.test(NCBirths2004$Weight[NCBirths2004$Gender=="Female"], conf = 0.99)$conf
```

```
## [1] 3343.305 3453.328
## attr(,"conf.level")
## [1] 0.99
```

Assumptions With any statistical procedure, one of the first questions to ask is, How robust is it? That is, what happens if the assumptions underlying the procedure are violated? The t confidence interval assumes that the underlying populations is normal, so what happens if that is not the case?

When the population has a normal distribution, the t interval is exact: a $(1-\alpha)\times 100\%)$ interval covers μ with probability $1-\alpha$ or, equivalently, misses μ on either side with probability $\alpha/2$; that is, the interval is completely above μ with probability $\alpha/2$ or is completely below with probability $\alpha/2$.

Let us check this for a nonnormal population by running a simulation.

Example 7.7 We draw random samples from the right-skewed gamma distribution with $\alpha = 5$ and $\lambda = 2$ and count the number of times the 95% confidence interval misses the mean $\mu = 5/2$ on each side.

```
set.seed(13)
tooLow <- 0       # set counter to 0
tooHigh <- 0      # set counter to 0
n <- 20           # sample size
q <- qt(0.975, n - 1)
N <- 10^5
for(i in 1:N){
  x <- rgamma(n, shape = 5, rate = 2)
  xbar <- mean(x)
  s <- sd(x)
  L <- xbar - q*s/sqrt(n)
  U <- xbar + q*s/sqrt(n)
  if(U < 5/2){tooLow <- tooLow + 1}
  if(L > 5/2){tooHigh <- tooHigh + 1}
}
TL <- tooLow/N*100
TH <- tooHigh/N*100
c(TL, TH)
```

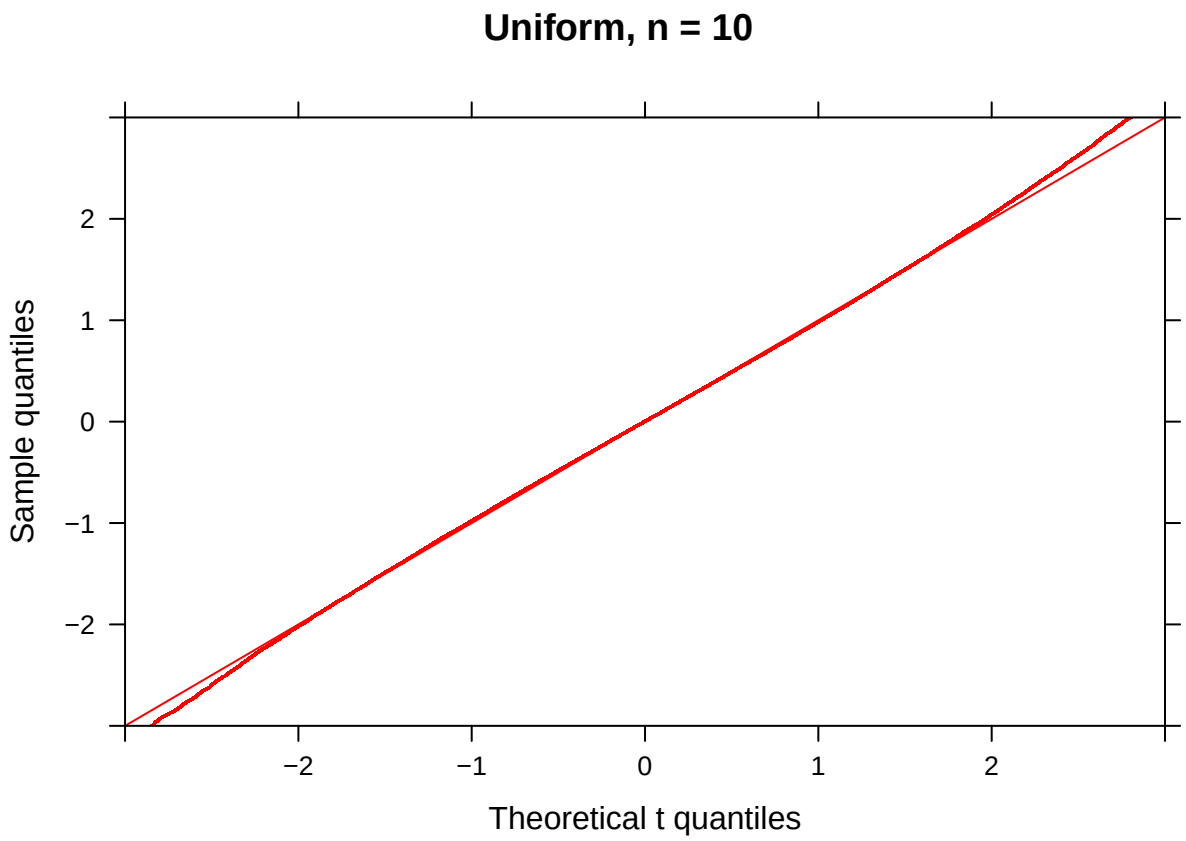
```
## [1] 4.340 1.328
```

What proportion of the times did the confidence intervals miss the true mean 5/2? In one run of this simulation, about 4.34% of the time, the interval was too low and was below 5/2, and about 1.328% of the time, the interval was too high and was above 5/2.

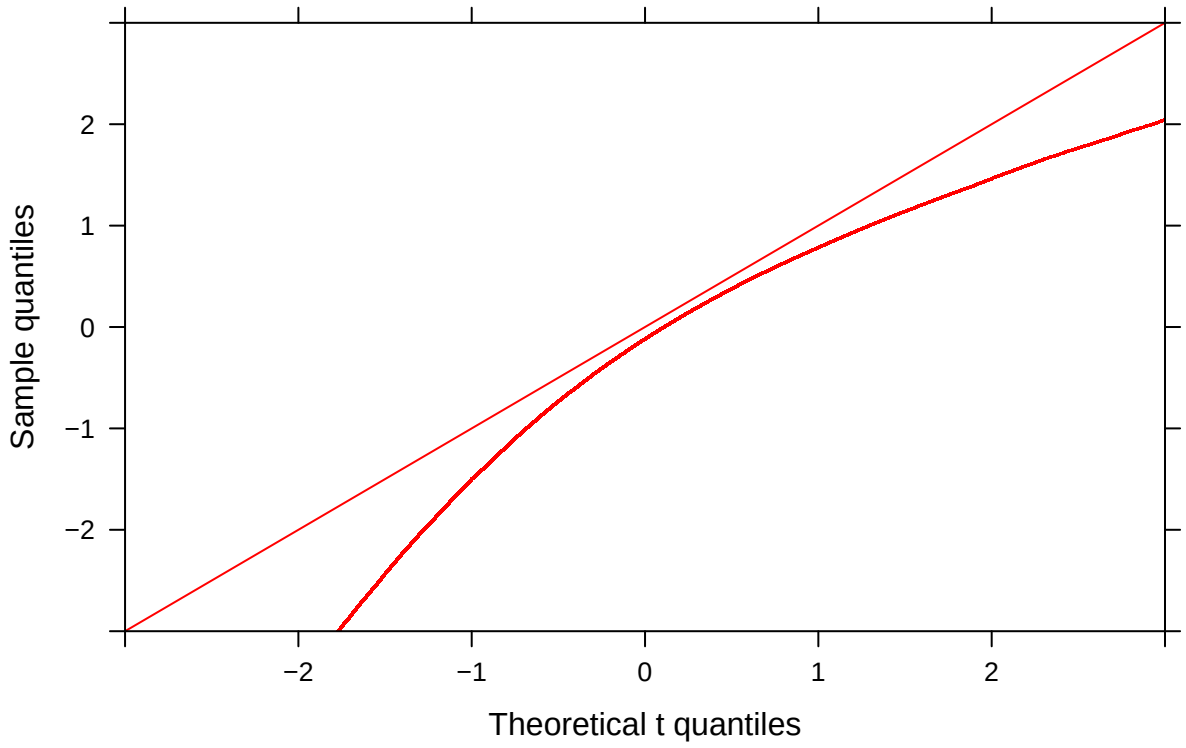
When the population is nonnormal but symmetric and the sample size is moderate or large, the t interval is very accurate. The main weakness of the t confidence interval occurs when the population is skewed. The simulation illustrated this problem. To see this from another point of view, we will look at the distributions of the t statistics, $T = (\bar{X} - \mu)/(S/\sqrt{n})$, since accuracy of t intervals depends on how close the t statistic is to having a t distribution.

```
set.seed(13)
n <- 10           # sample size
q <- qt(0.975, n - 1)
N <- 10^5
TSU <- numeric(N)
for(i in 1:N){
  x <- runif(n, 0, 1)
  xbar <- mean(x)
  s <- sd(x)
  TSU[i] <- (xbar - 0.5)/(s/sqrt(n))
}
TSE10 <- numeric(N)
for(i in 1:N){
  x <- rexp(n, 1)
  xbar <- mean(x)
  s <- sd(x)
  TSE10[i] <- (xbar - 1)/(s/sqrt(n))
}
n <- 100
TSE100 <- numeric(N)
for(i in 1:N){
```

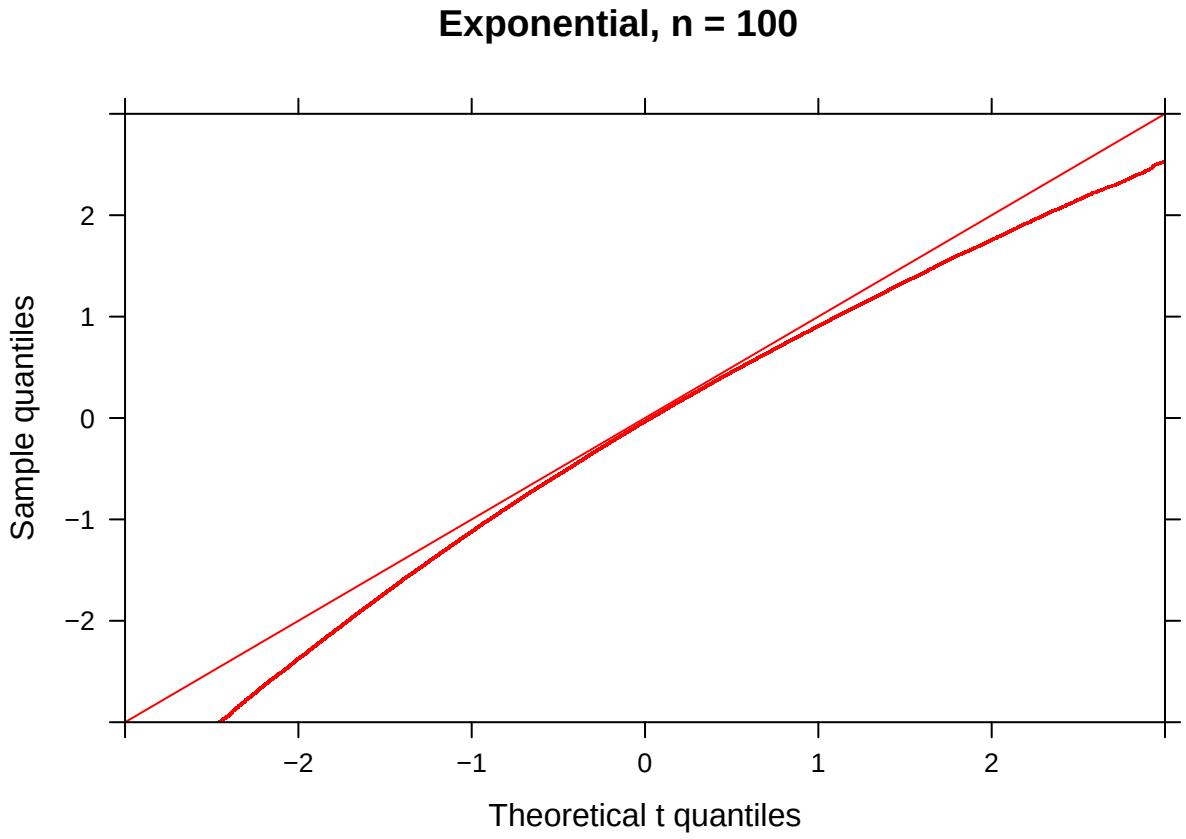
```
  x <- rexp(n, 1)
  xbar <- mean(x)
  s <- sd(x)
  TSE100[i] <- (xbar - 1)/(s/sqrt(n))
}
#
n <- 10
qqmath(~TSU, col = "red", xlim = c(-3,3), ylim = c(-3,3), distribution = function(p){qt(p, df = n - 1)}
  panel.qqmath(x, pch = ".", ...)
  panel.abline(a = 0, b =1, ...)
})
```




```
#
qqmath(~TSE10, col = "red", xlim = c(-3,3), ylim = c(-3,3), distribution = function(p){qt(p, df = n - 1
  panel.qqmath(x, pch = ".", ...)
  panel.abline(a = 0, b = 1, ...)
})
```



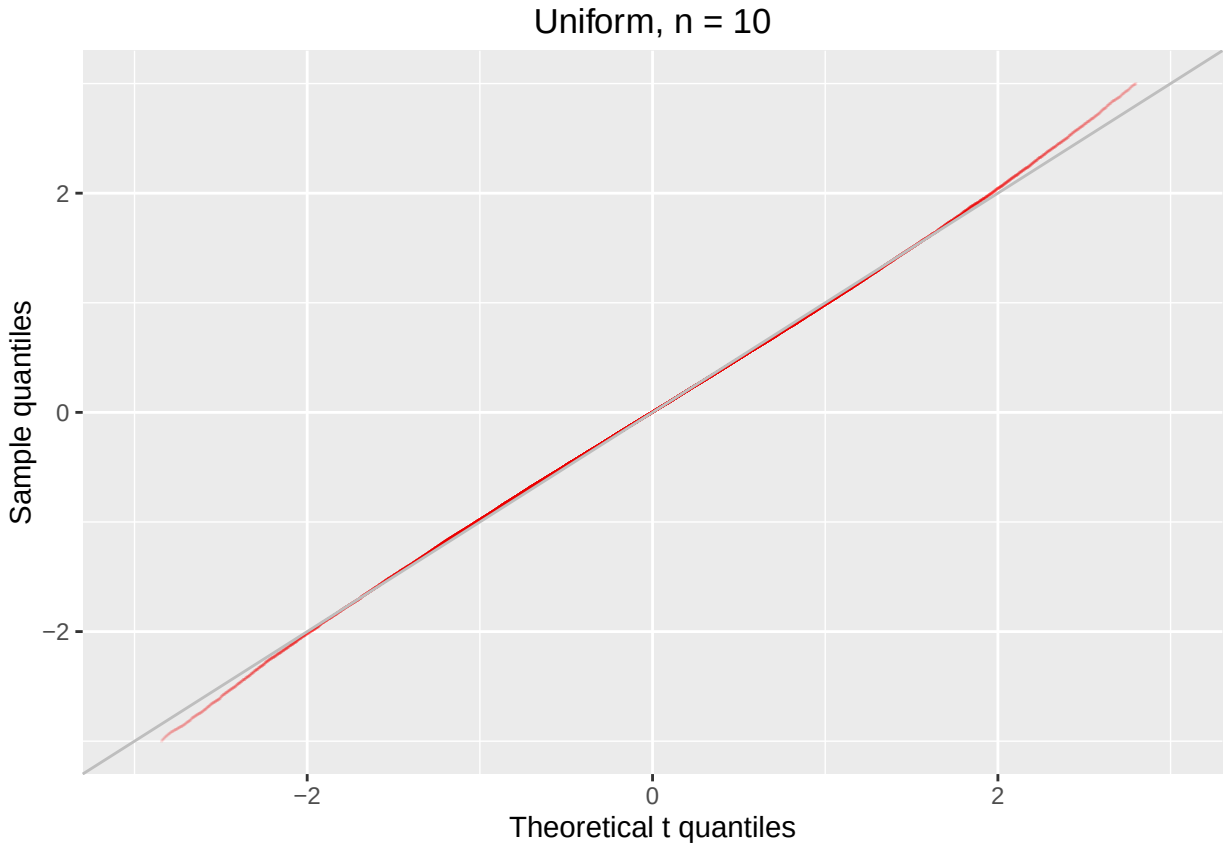

```
#
n <- 100
qqmath(~TSE100,col = "red", xlim = c(-3,3), ylim = c(-3,3), distribution = function(p){qt(p, df = n - 1
  panel.qqmath(x, pch = ".", ...)
  panel.abline(a = 0, b = 1, ...)
})
```




```
#
```

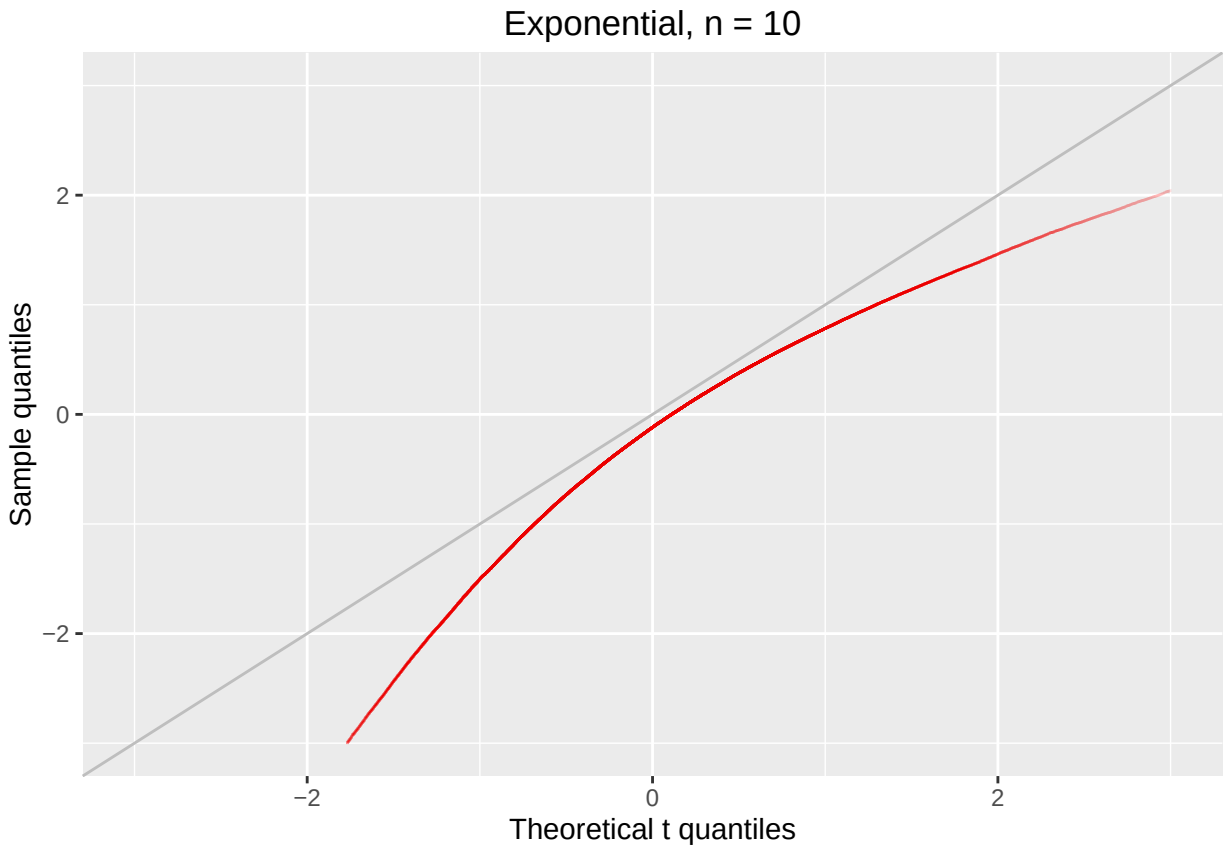
```
DF <- data.frame(TSU, TSE10, TSE100)
p <- ggplot(data=DF, aes(sample = TSU)) + stat_qq(distribution = qt, dparams = list(df = 9), pch =".", c
p + geom_abline(intercept = 0, slope = 1, color = "gray")
```

```
## Warning: Removed 1995 rows containing missing values (geom_point).
```



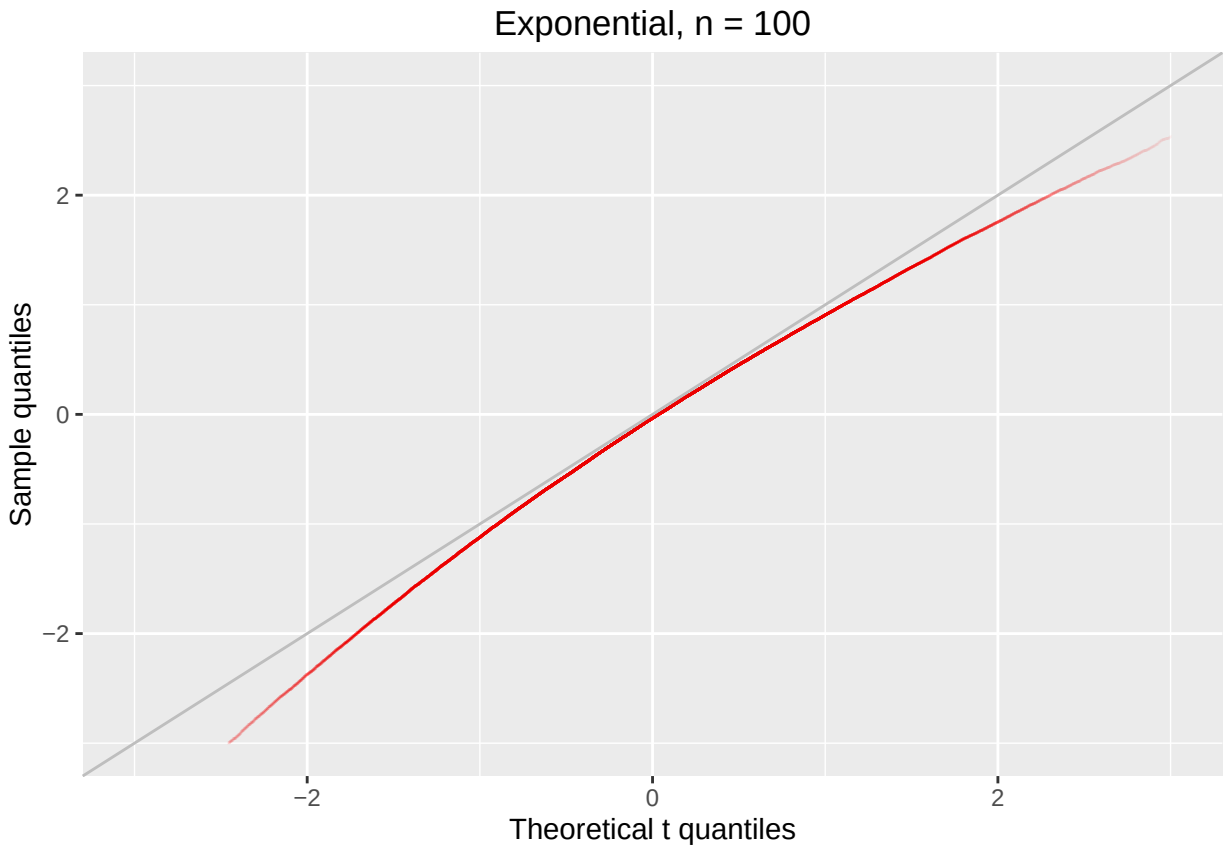

```
#
p <- ggplot(data=DF, aes(sample = TSE10)) + stat_qq(distribution = qt, dparams = list(df = 9), pch ="."
p + geom_abline(intercept = 0, slope = 1, color = "gray")
```

```
## Warning: Removed 6283 rows containing missing values (geom_point).
```




```
#
p <- ggplot(data=DF, aes(sample = TSE100)) + stat_qq(distribution = qt, dparams = list(df = 99), pch ="
p + geom_abline(intercept = 0, slope = 1, color = "gray")
```

```
## Warning: Removed 963 rows containing missing values (geom_point).
```



The previous graphs compare the distribution of t statistics for samples of size $n = 10$ from a uniform distribution, size $n = 10$ from an exponential distribution, and size $n = 100$ from an exponential distribution to the t distribution. The range on all plots is truncated so that we can focus on the range of values important for confidence intervals. Notice that for the uniform population, the distribution of the t statistic is close to the t distribution, except in the tails. For exponential populations, the discrepancy is much larger, and the discrepancy decreases only slowly as the sample size increases. To reduce the discrepancy (the difference between actual and nominal probabilities) by a factor of 10 requires a sample size 100 times larger. For an exponential population, we must have $n > 5000$ before the actual probabilities of a 95% t interval missing the true mean in either tail are within 10% of the desired probability of 2.5%; that is, the actual tail probabilities are between 2.25% and 2.75%.

Before using a t confidence interval, you should create a normal quantile plot to see whether the data are skewed. The larger the sample size, the more skew can be tolerated. There are skewness adjusted versions of t intervals including bootstrap t intervals that we cover later. However, be particularly careful with outliers: since $\bar{x}$ is sensitive to extreme values, outliers can have a big impact on confidence intervals – a bigger impact than skewness. If you have outliers in your data, you should investigate: are these recording errors or observations that are not representative of the population? If the former, correct them; and if the latter, remove them. If the outliers cannot be removed, then advanced, more robust techniques may be required.

7.1.3 Confidence Intervals for a Difference in Means

Let X and Y be random variables with $X \sim N(\mu_1, \sigma_1)$ and $Y \sim N(\mu_2, \sigma_2)$. Then $X - Y \sim N\left(\mu_1 - \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2}\right)$ For samples sizes n_1 and n_2,

$$\bar{X} - \bar{Y} \sim N\left(\mu_1 - \mu_2, \sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}\right)$$

Of course, in practice we usually do not know the population variances, so we will plug in the sample variances. As in the single-sample case, we call this a t statistics:

$$T = \frac{(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)}{\sqrt{S_X^2/n_X + S_Y^2/n_Y}}$$

The exact distribution of this statistic is a n unsolved problem. It does, however, have approximately a t distribution if the populations are normal. The difficult part is the degrees of freedom. The degrees of freedom are given with Welch's approximation:

$$\nu = \frac{(s_X^2/n_x + s_Y^2/n_Y)^2}{(s_X^2/n_X)^2/(n_X - 1) + (S_Y^2/n_Y)^2/(n_Y - 1)}.$$

T CONFIDENCE INTERVAL FOR DIFFERENCE IN MEANS If $X_i \sim N(\mu_X, \sigma_X), i = 1, \ldots, n_X$, and $Y_j \sim N(\mu_Y, \sigma_Y), j = 1, \ldots, n_Y$, then an approximate $(1 - \alpha) \times 100\%$ confidence interval for $\mu_X - \mu_Y$ is given by

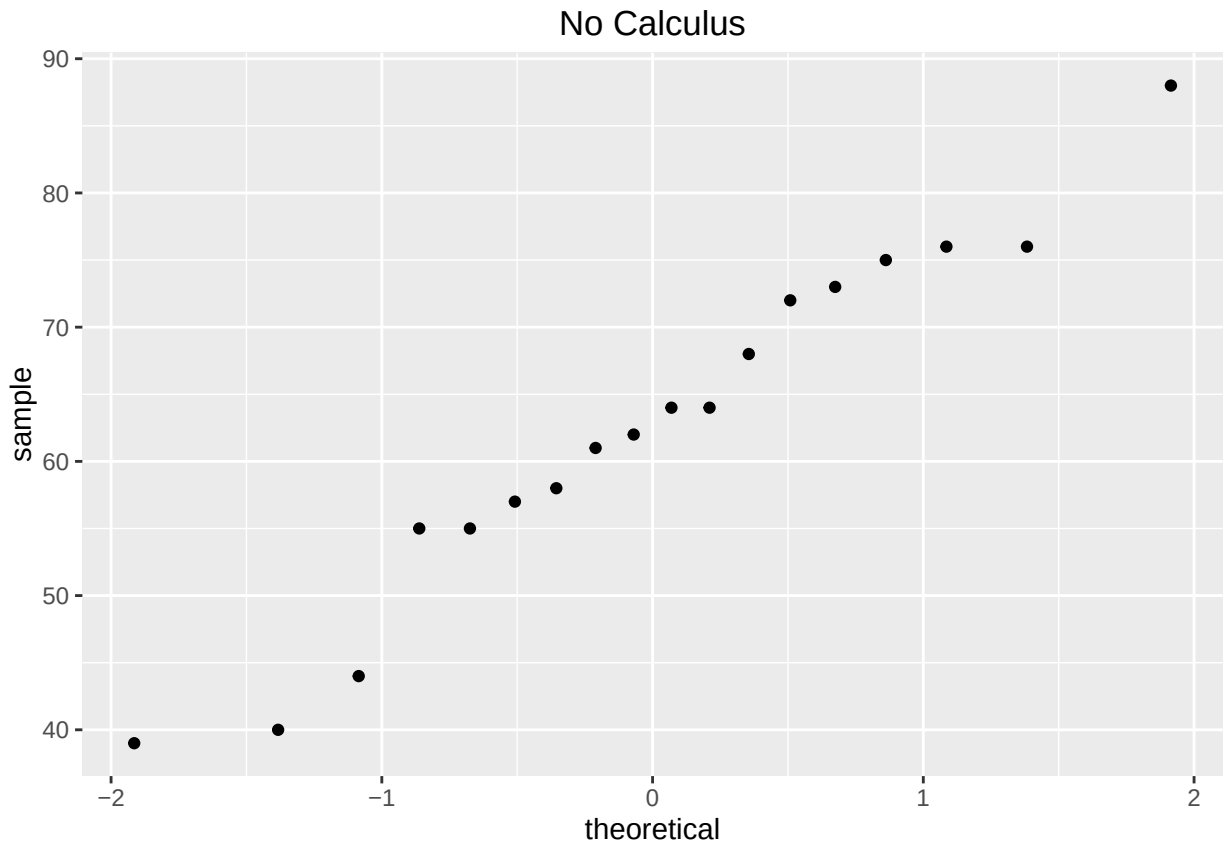
$$CI_{1-\alpha}(\mu_X - \mu_Y) = \left(\bar{X} - \bar{Y} - t_{1-\alpha/2;\nu} \times \sqrt{\frac{S_X^2 + S_Y^2}{n_X + n_Y}}, \bar{X} - \bar{Y} + t_{1-\alpha/2;\nu} \times \sqrt{\frac{S_X^2 + S_Y^2}{n_X + n_Y}} \right)$$

Example 7.8

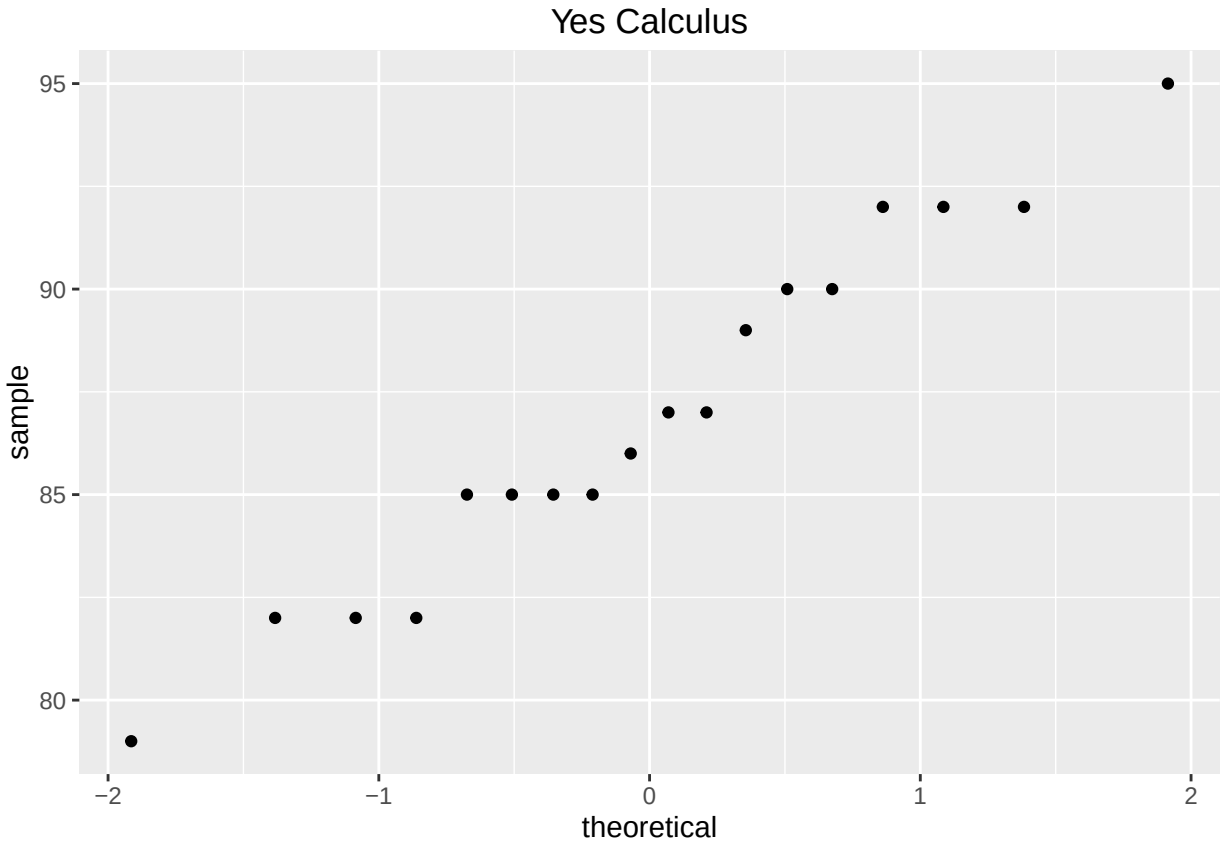
```
library(PASWR)
library(ggplot2)
```

Construct a 90% confidence interval for $\mu_X - \mu_Y$ using the information in `Calculus` which provides the assessment scores for students enrolled in a biostatistics course according to whether they had completed a calculus course prior to enrolling in the biostatistics course. Before constructing a confidence interval, one should verify the assumptions needed to have a valid confidence interval. In this case we need to check for normality of both samples.

```
ggplot(data = Calculus, aes(sample = Calculus$No.Calculus)) + stat_qq() + ggtitle("No Calculus")
```




```
ggplot(data = Calculus, aes(sample = Calculus$Yes.Calculus)) + stat_qq() + ggtitle("Yes Calculus")
```



For the $n_X = 18$ students who had no calculus, the mean and standard deviation of their course scores are $\bar{x} = 62.6111111$, and $s_X = 13.2227028$. For the $n_Y = 18$ students who had calculus, the mean and standard deviation of their course scores are $\bar{y} = 86.9444444$, and $s_Y = 4.3178456$. The mean difference is $\bar{x} - \bar{y} =$ -24.3333333 with the standard error of the difference of 3.2785808 and degrees of freedom are 20.5847782. The 0.95 quantile of the t distribution with 20.5847782 degrees of freedom is 1.7223436. Thus, the 90% confidence interval is $-24.333 \pm 1.7233 \times 3.2786 = (-29.9802, -18.6865)$.

R Note:

```
t.test(Calculus$No.Calculus, Calculus$Yes.Calculus, conf.level = 0.90)
```

```
##
##  Welch Two Sample t-test
##
## data:  Calculus$No.Calculus and Calculus$Yes.Calculus
## t = -7.4219, df = 20.585, p-value = 3.04e-07
## alternative hypothesis: true difference in means is not equal to 0
## 90 percent confidence interval:
##  -29.98018 -18.68649
## sample estimates:
## mean of x mean of y
##  62.61111  86.94444
```

```
t.test(Calculus$No.Calculus, Calculus$Yes.Calculus, conf.level = 0.90)$conf.int
```

```
## [1] -29.98018 -18.68649
## attr(,"conf.level")
## [1] 0.9
```

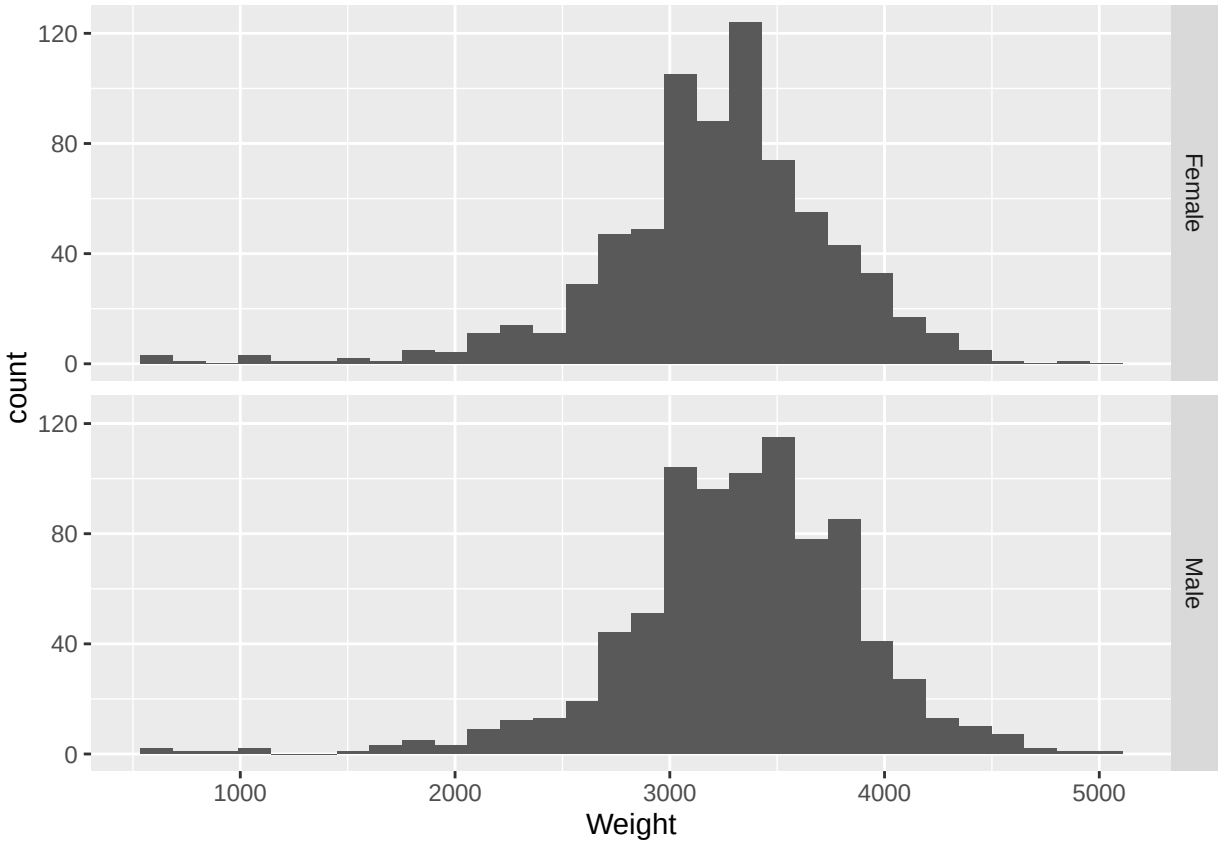
Remark

- If the confidence interval for the difference in means contains 0, then we cannot rule out the possibility that the means might be the same, $\mu_X - \mu_Y = 0$ or, equivalently, $\mu_X = \mu_Y$.
- Skewness is less of an issue for the two-sample t confidence intervals than for one-sample intervals, because the skewness from the two samples tends to cancel out. In particular, if the populations have the same skewness and variance and the sample sizes are equal, then the skewness cancels out exactly, and the distribution of t statistics can be very close to a t distribution even for quite small samples.

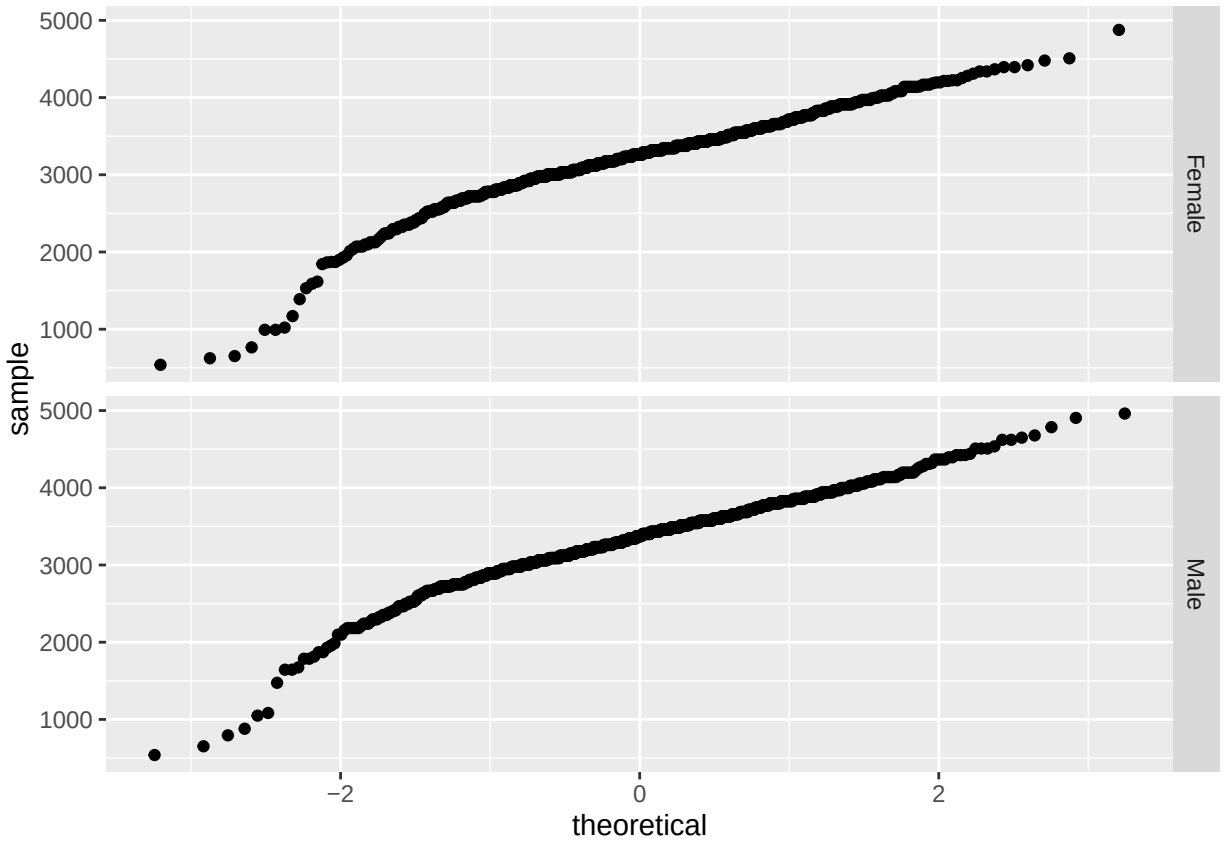
Example 7.9 Consider the weights of boy and girl babies born in Texas in 2004. Construct a 95% t confidence interval for the mean difference in weights (boys - girls).

```
site <- "http://www1.appstate.edu/~arnholta/Data/TXBirths2004.csv"
Texas <- read.csv(file=url(site))
ggplot(data = Texas, aes(x = Weight)) + geom_histogram() + facet_grid(Gender~.)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(data = Texas, aes(sample = Weight)) + stat_qq() + facet_grid(Gender~.)
```




```
t.test(Texas$Weight~Texas$Gender)
```

```
##
##  Welch Two Sample t-test
##
## data:  Texas$Weight by Texas$Gender
## t = -4.193, df = 1552.9, p-value = 2.908e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -170.12395  -61.68415
## sample estimates:
## mean in group Female   mean in group Male
##             3220.939             3336.843
```

```
str(Texas)
```

```
## 'data.frame':    1587 obs. of  8 variables:
##  $ ID        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MothersAge: Factor w/ 7 levels "15-19","20-24",..: 2 2 3 3 1 4 4 3 6 3 ...
##  $ Smoker    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender    : Factor w/ 2 levels "Female","Male": 2 2 1 1 1 1 2 1 2 1 ...
##  $ Weight    : int  3033 3232 3317 2560 2126 2948 3884 2665 3714 2977 ...
##  $ Gestation : int  39 40 37 36 37 38 39 38 40 37 ...
##  $ Number    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Multiple  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
Texas <- within(data = Texas, expr={Gender <- factor(Gender, levels =c("Male", "Female"))})
str(Texas)
```

```
## 'data.frame':    1587 obs. of  8 variables:
##  $ ID        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MothersAge: Factor w/ 7 levels "15-19","20-24",..: 2 2 3 3 1 4 4 3 6 3 ...
##  $ Smoker    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender    : Factor w/ 2 levels "Male","Female": 1 1 2 2 2 2 1 2 1 2 ...
##  $ Weight    : int  3033 3232 3317 2560 2126 2948 3884 2665 3714 2977 ...
##  $ Gestation : int  39 40 37 36 37 38 39 38 40 37 ...
##  $ Number    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Multiple  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
t.test(Texas$Weight~Texas$Gender)
```

```
##
##  Welch Two Sample t-test
##
## data:  Texas$Weight by Texas$Gender
## t = 4.193, df = 1552.9, p-value = 2.908e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   61.68415 170.12395
## sample estimates:
##   mean in group Male mean in group Female
##             3336.843             3220.939
```

7.4 CONFIDENCE ITERVALS FOR PROPORTIONS

In 2010, according to an AP-Gfk Poll conducted on October 13-18, 59% of 846 likely voters responded that they felt things in this country were heading in the wrong direction (http:www.ap-gfkpoll.com/poll.archive.html). Let X denote the number of likely voters in a sample of size n who think the country is headed in the wrong direction. We assume X is binomial, $X \sim Bin(n, p)$. We know that the proportion of likely voters, $\hat{p} = X/n$ is an unbiased estimator of p and for large n, $Z = (\hat{p} - p)/\sqrt{p(1-p)}$ is approximately standard normal. Thus,

$$P\left(-z_{1-\alpha/2} < \frac{\hat{p} - p}{\sqrt{p(1-p)/n}} < z_{1-\alpha/2}\right) \approx 1 - \alpha$$

Isolating the p in this expression requires a bit more of algebra than the earlier problems. We set

$$-z_{1-\alpha/2} = \frac{\hat{p} - p}{\sqrt{p(1-p)/n}}$$

and solve for p (we get the same answer if we had set the right-hand side of the above to $z_{1-\alpha/2}$). This leads to the quadratic equation

$$\left(z_{1-\alpha/2}^2 + n\right) p^2 - \left(2n\hat{p} + z_{1-\alpha/2}\right) p + n\hat{p}^2$$

Using the quadratic formula to solve for p gives a $(1-\alpha)$ 100% confidence interval (L, U), where

$$L = \frac{\hat{p} + z_{1-\alpha/2}^2/(2n) - z_{1-\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n + z_{1-\alpha/2}^2/(4n^2)}}{1 - z_{1-\alpha/2}^2/n},$$

$$U = \frac{\hat{p} + z_{1-\alpha/2}^2/(2n) + z_{1-\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n + z_{1-\alpha/2}^2/(4n^2)}}{1 - z_{1-\alpha/2}^2/n}.$$

Thus for a 90% confidence interval using $\hat{p} = 0.59$ and $n = 846$ we have

$$\frac{0.59 + (1.645^2/(2 \times 846)) \pm 1.645\sqrt{0.59 \times 0.41/846 + 1.645^2/(4 \times 846^2)}}{1 - 1.645^2/846} = 0.589 \pm 0.0278 = (0.5618, 0.6173)$$

R Note:

```
prop.test(x = 499, n = 846, conf.level = 0.90, correct = FALSE)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  499 out of 846, null probability 0.5
## X-squared = 27.31, df = 1, p-value = 1.733e-07
## alternative hypothesis: true p is not equal to 0.5
## 90 percent confidence interval:
##  0.5617755 0.6173207
## sample estimates:
##         p
## 0.5898345
```

```
prop.test(x = 499, n = 846, conf.level = 0.90, correct = FALSE)$conf
```

```
## [1] 0.5617755 0.6173207
## attr(,"conf.level")
## [1] 0.9
```

Remark

- This interval is called a Wilson or Wilson score interval (Wilson (1927)).
- The center of the score interval is $(\hat{p} + z^2_{1-\alpha/2})/(2n + z^2_{1-\alpha/2}/n)$. If we set $\kappa = z^2_{1-\alpha/2}$, then the center can be written as $\hat{p}(n/(n+\kappa)) + (1/2)(\kappa/(n+\kappa))$, a weighted average of the observed proportions and 1/2. As n increases, more weight is given to $\hat{p}$.

7.4.1 The Agresti-Coull Interval for a Proportion

Now, the limits of the interval given for the score interval are pretty messy, so in general, we would want software to do the calculations. However, in the event that we must resort to hand calculations, it would be nice to find a simpler expression for the confidence interval. Agresti and Coull (1998) considered the 95% confidence interval for which the 0.975 quantile is $z_{0.975} \approx 1.96$, and hence $z^2_{0.975} \approx 4$.

The AGRESTI-COULL 95% CONFIDENCE INTERVAL FOR A PROPORTION

If X denotes the number of successes in a sample of size n, let $\tilde{X} = X = 2, \tilde{n} = n + 4$, and $\tilde{p} = \tilde{X}/\tilde{n}$. Then an approximate 95% confidence interval for p is

$$\left(\tilde{p} - 1.96\sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}}, \tilde{p} + 1.96\sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}}\right)$$

Example 7.18 Suppose the sample size is $n = 210$ with $x = 130$. Then $\tilde{x} = 130 + 2 = 132$, $\tilde{n} = 210 + 4 = 214$, and $\tilde{p} = 132/214 = 0.6168224$. Thus an approximate 95% confidence interval is given by

```
xtilde <- 132
ntilde <- 214
ptilde <- xtilde/ntilde
ptilde + c(-1, 1)*qnorm(.975)*sqrt(ptilde*(1 - ptilde)/ntilde)
```

```
## [1] 0.5516864 0.6819585
```

```
# Compare to
prop.test(x = 130, n = 210, correct = FALSE)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  130 out of 210, null probability 0.5
## X-squared = 11.905, df = 1, p-value = 0.0005599
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.5517861 0.6820320
## sample estimates:
##         p
## 0.6190476
```

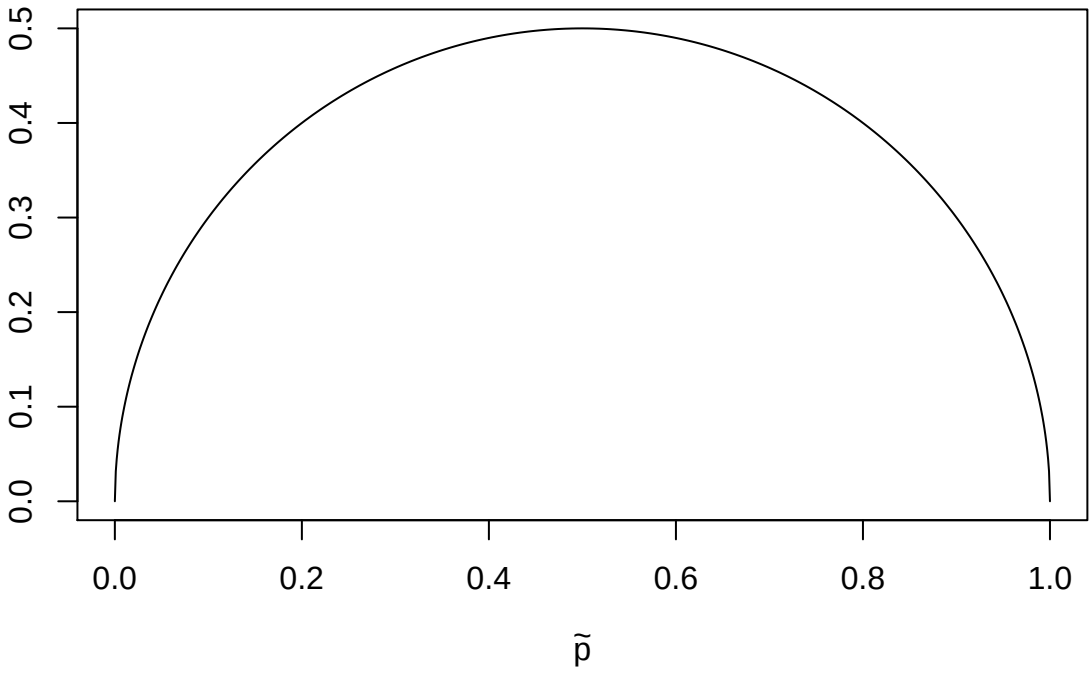
Example 7.19 A political candidate prepares to conduct a survey to gauge voter support for his candidacy for senator. He would like a confidence interval with an error of at most 4%, with 95% confidence. How large should the sample size be for the survey?

Solution Since the Agresti-Coull interval is symmetric, the margin of error is $1.96\sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}}$. Thus, we want to solve for $\tilde{n}$ in

$$1.96\sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}} \leq 0.04.$$

Unfortunately, we do not know $\tilde{p}$ — if we did, the candidate would not need to conduct the survey! We will use $\tilde{p} = 0.5$ since this will maximize the expression under the radical sign. Use calculus to prove this on your own.

```
ptilde <- seq(0, 1, length= 1000)
fptilde <- sqrt(ptilde*(1 - ptilde))
plot(ptilde, fptilde, type = "l", ylab = "", xlab =expression(tilde(p)))
```



$$1.96\sqrt{\frac{0.5(1-0.5)}{\tilde{n}}} \leq 0.04$$

Solving for $\tilde{n}$ yields

$$\left(\frac{1.96(0.5)}{0.04}\right)^2 \leq \tilde{n} \rightarrow \tilde{n} \geq 600.25 \rightarrow n \geq 596.25.$$

```
ntilde <- (1.96*(0.5)/0.04)^2
n <- ntilde - 4
n <- ceiling(n)
n
```

```
## [1] 597
```

Thus, he should survey at least 597 people. In some instances, based on prior knowledge, the analyst may substitute another estimate for $\tilde{p}$ (e.g., the proportion form a previous poll).

Coverage Probabilities of Binomial Confidence Intervals

Suppose a new process for making a prescription drug is in development. Of $n = 30$ trial batches made with the current version of the process, $X = 24$ batches give satisfactory results. Then $\hat{p} = 24/30 = 0.8$ estimates the population proportion $p = P(Success)$ of satisfactory batches with the current version of the process. Wondering how near $\hat{p}$ might be to p, the investigators use

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p} \times (1-\hat{p})}{n}} \qquad (1)$$

to obtain the approximate 95% confidence interval 0.8 ± 0.1431355 or (0.6568645, 0.9431355). The question is whether a 95% level of confidence in the resulting interval is warranted. If (1) is used repeatedly, in what proportion of instances does it yield an interval that covers the true value p? If (1) is valid here, then the simple answer ought to be 95%. Unfortunately, there is no simple answer to this question. It turns out that the coverage probability depends on the value of p.

In our situation, there are 31 possible values $0, 1, , \ldots, 30$ of X, and thus of $\hat{p}$. From (1) we can compute the confidence interval corresponding to each of these 31 possible outcomes, just as we computed the confidence interval (0.6568645, 0.9431355) corresponding to the outcome $X = 24$ above.

Now choose a particular value of p, say $p = 0.8$, so that X has a binomial distribution with $n = 30$ trials and probability of success ($p = 0.8$). We write this as $X \sim Bin(n = 30, p = 0.8)$. The vector of `prob` of the 31 probabilities $P(X = x)$ in this distribution is found with the `R` function `dbinom(0:30, 30, 0.8)`. In the `R` code, we use `pp` and `sp` for population proportion and sample proportion respectively. Next, we determine which of the 31 confidence intervals cover the value $p = 0.8$. Finally, the coverage probability is computed: It is the sum of the probabilities corresponding to values of x that yield intervals covering p.

```
alpha <- 0.05
n <- 30   # number of trials
x <- 0:n
sp <- x/n # sample proportion
m.err <- qnorm(1 - alpha/2)*sqrt(sp*(1 - sp)/n)
lcl <- sp - m.err
ucl <- sp + m.err
pp <- 0.8   # pp = P(Success)
prob <- dbinom(x, n, pp)
cover <- (pp >= lcl) & (pp <= ucl)  # vector of Os and 1s
RES <- round(cbind(x, sp, lcl, ucl, prob, cover), 4)
RES[18:31, ]
```

```
##        x     sp    lcl    ucl   prob cover
##  [1,] 17 0.5667 0.3893 0.7440 0.0022     0
##  [2,] 18 0.6000 0.4247 0.7753 0.0064     0
##  [3,] 19 0.6333 0.4609 0.8058 0.0161     1
##  [4,] 20 0.6667 0.4980 0.8354 0.0355     1
##  [5,] 21 0.7000 0.5360 0.8640 0.0676     1
##  [6,] 22 0.7333 0.5751 0.8916 0.1106     1
##  [7,] 23 0.7667 0.6153 0.9180 0.1538     1
##  [8,] 24 0.8000 0.6569 0.9431 0.1795     1
##  [9,] 25 0.8333 0.7000 0.9667 0.1723     1
## [10,] 26 0.8667 0.7450 0.9883 0.1325     1
## [11,] 27 0.9000 0.7926 1.0074 0.0785     1
## [12,] 28 0.9333 0.8441 1.0226 0.0337     0
## [13,] 29 0.9667 0.9024 1.0309 0.0093     0
## [14,] 30 1.0000 1.0000 1.0000 0.0012     0
```

```
sum(dbinom(x[cover], n, pp))  # total coverage prob at pp
```

```
## [1] 0.9463279
```

Thus the total coverage probability for $p = 0.8$ is $P(Cover) = P(X = 19) + P(X = 20) + \cdots + P(X = 27) = 0.9463279$. This is only a little smaller than the claimed value of 95%. In contrast, a similar computation of $n = 30$ and $p = 0.79$ gives a coverage probability of 0.8876. This is very far below the claimed coverage of 95%. The individual binomial probabilities do not change much when p changes from 0.80 to 0.79. The main reason for the large change in the coverage probability is that, for $p = 0.79$, the confidence interval corresponding to $x = 27$ no longer covers p.

```
alpha <- 0.05
n <- 30   # number of trials
x <- 0:n
sp <- x/n # sample proportion
m.err <- qnorm(1 - alpha/2)*sqrt(sp*(1 - sp)/n)
lcl <- sp - m.err
ucl <- sp + m.err
pp <- 0.79   # pp = P(Success)
prob <- dbinom(x, n, pp)
cover <- (pp >= lcl) & (pp <= ucl)  # vector of 0s and 1s
RES <- round(cbind(x, sp, lcl, ucl, prob, cover), 4)
RES[18:31, ]
```

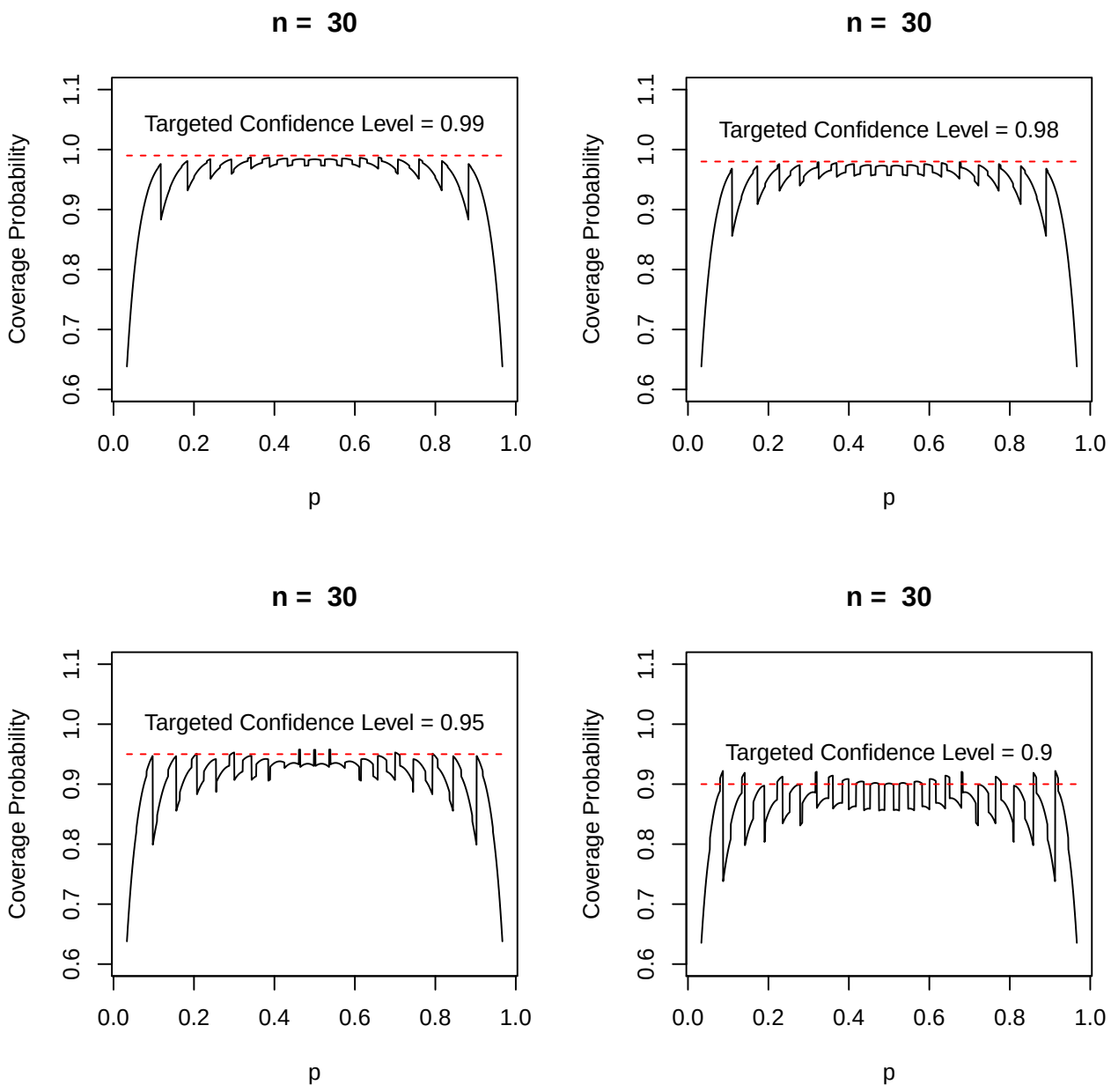
```
##        x     sp    lcl    ucl   prob cover
##  [1,] 17 0.5667 0.3893 0.7440 0.0034     0
##  [2,] 18 0.6000 0.4247 0.7753 0.0091     0
##  [3,] 19 0.6333 0.4609 0.8058 0.0217     1
##  [4,] 20 0.6667 0.4980 0.8354 0.0449     1
##  [5,] 21 0.7000 0.5360 0.8640 0.0805     1
##  [6,] 22 0.7333 0.5751 0.8916 0.1239     1
##  [7,] 23 0.7667 0.6153 0.9180 0.1621     1
##  [8,] 24 0.8000 0.6569 0.9431 0.1778     1
##  [9,] 25 0.8333 0.7000 0.9667 0.1605     1
## [10,] 26 0.8667 0.7450 0.9883 0.1161     1
## [11,] 27 0.9000 0.7926 1.0074 0.0647     0
## [12,] 28 0.9333 0.8441 1.0226 0.0261     0
## [13,] 29 0.9667 0.9024 1.0309 0.0068     0
## [14,] 30 1.0000 1.0000 1.0000 0.0008     0
```

```r
sum(dbinom(x[cover], n, pp))  # total coverage prob at pp
```

```
## [1] 0.8875662
```

To get a more comprehensive view of the performance of confidence intervals based on formula (1), we step through two thousand values of p from near 0 to near 1. For each value of p we go through a coverage probability and subsequently plot coverage probability versus p.

```r
opar <- par(no.readonly = TRUE)
par(mfrow=c(2, 2))
for(alpha in c(0.01, 0.02, 0.05, 0.10)){
n <- 30     # number of trials
CL <- 1 - alpha
x <- 0:n
adj <- 0    #(2 for Agresti-Coull)
k <- qnorm(1 - alpha/2)
sp <- (x + adj)/(n + 2*adj)
m.err <- k * sqrt(sp*(1 - sp)/(n + 2*adj))
lcl <- sp - m.err
ucl <- sp + m.err
m <- 2000 # number of values of pp
pp <- seq(1/n, 1 - 1/n, length = m)
p.cov <- numeric(m)
for(i in 1:m){
  cover <- (pp[i] >= lcl) & (pp[i] <= ucl)  # vector of 0s and 1s
  p.rel <- dbinom(x[cover], n, pp[i])
  p.cov[i] <- sum(p.rel)
}
plot(pp, p.cov, type = "l", ylim =c(0.60, 1.1), main = paste("n = ", n), xlab = "p", ylab = "Coverage P
lines(c(1/n, 1- 1/n), c(1 - alpha, 1- alpha), col = "red", lty = "dashed")
      text(0.5, CL + 0.05, paste("Targeted Confidence Level =", CL))
}
```

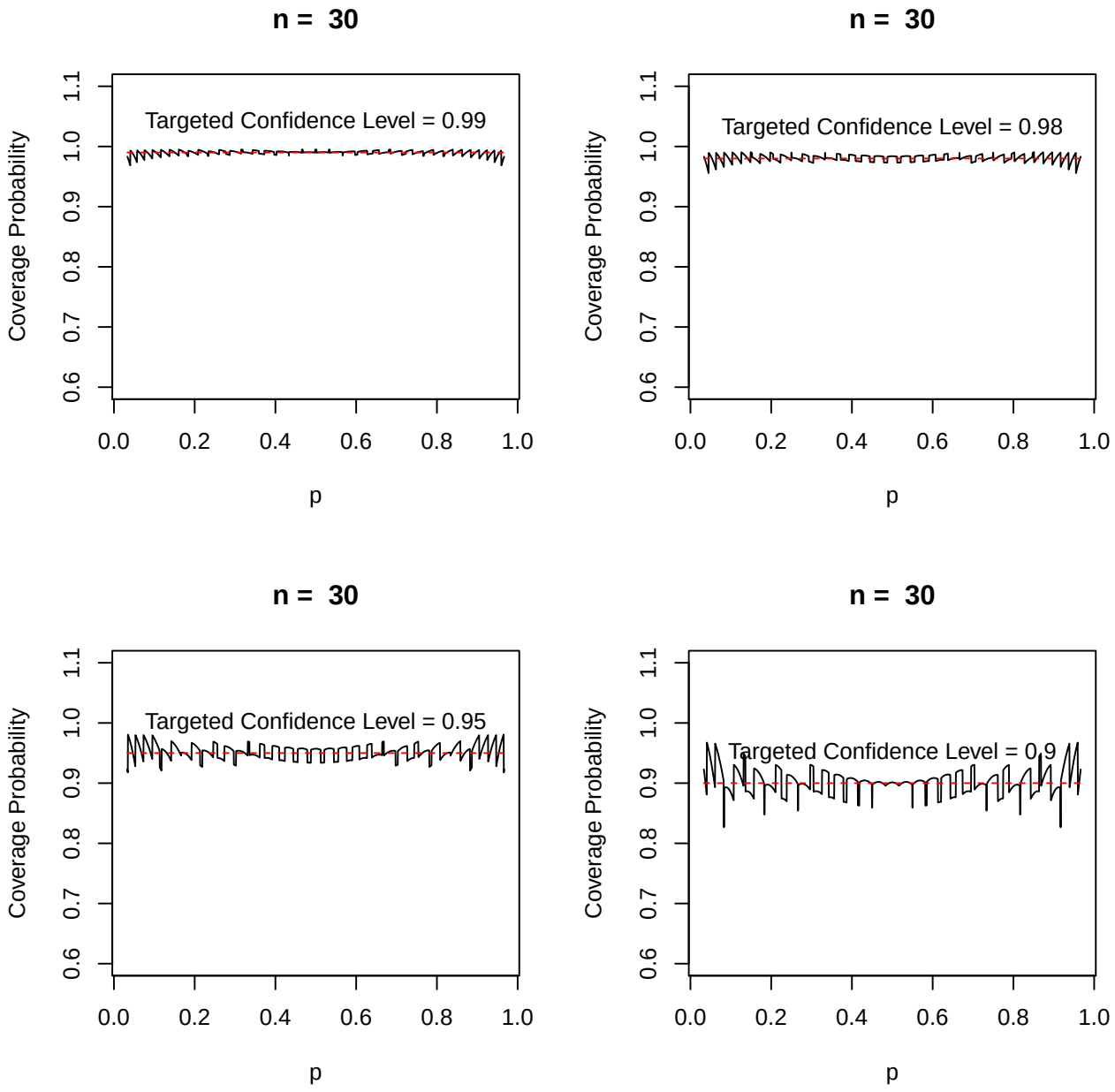



```
par(opar)
```

Next we use Wilson score intervals to do the same thing.

```
opar <- par(no.readonly = TRUE)
par(mfrow=c(2, 2))
for(alpha in c(0.01, 0.02, 0.05, 0.10)){
n <- 30     # number of trials
CL <- 1 - alpha
x <- 0:n
z <- qnorm(1 - alpha/2)
sp <- x/n
sptilda <- (x + z^2/2)/(n + z^2)
m.err <- (z/(n + z^2))*sqrt(n*sp*(1 - sp) + z^2/4)
lcl <- sptilda - m.err
ucl <- sptilda + m.err
```

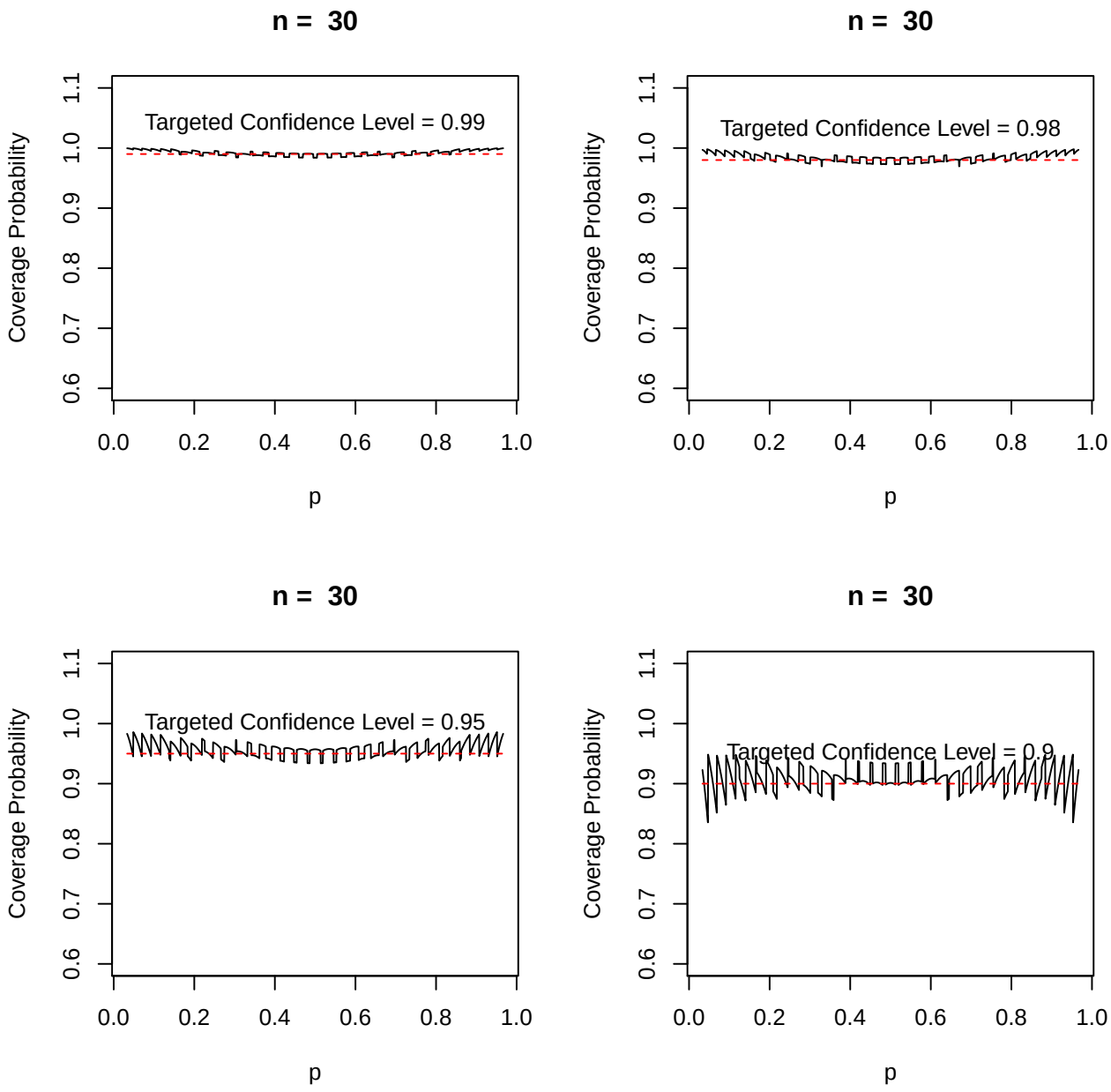
```r
m <- 2000 # number of values of pp
pp <- seq(1/n, 1 - 1/n, length = m)
p.cov <- numeric(m)
for(i in 1:m){
  cover <- (pp[i] >= lcl) & (pp[i] <= ucl)  # vector of 0s and 1s
  p.rel <- dbinom(x[cover], n, pp[i])
  p.cov[i] <- sum(p.rel)
}
plot(pp, p.cov, type = "l", ylim =c(0.60, 1.1), main = paste("n = ", n), xlab = "p", ylab = "Coverage Pr
lines(c(1/n, 1- 1/n), c(1 - alpha, 1- alpha), col = "red", lty = "dashed")
      text(0.5, CL + 0.05, paste("Targeted Confidence Level =", CL))
}
```



```
par(opar)
```

Next consider the Agresti-Coull confidence intervals.

```
opar <- par(no.readonly = TRUE)
par(mfrow=c(2, 2))
for(alpha in c(0.01, 0.02, 0.05, 0.10)){
n <- 30     # number of trials
CL <- 1 - alpha
x <- 0:n
adj <- 2  # 0 for large sample 2 for Agresti Coull
z <- qnorm(1 - alpha/2)
sp <- (x + adj)/(n + 2*adj)
m.err <- z*sqrt(sp*(1 - sp)/(n + 2*adj))
lcl <- sp - m.err
ucl <- sp + m.err
m <- 2000 # number of values of pp
pp <- seq(1/n, 1 - 1/n, length = m)
p.cov <- numeric(m)
for(i in 1:m){
  cover <- (pp[i] >= lcl) & (pp[i] <= ucl)  # vector of 0s and 1s
  p.rel <- dbinom(x[cover], n, pp[i])
  p.cov[i] <- sum(p.rel)
}
plot(pp, p.cov, type = "l", ylim =c(0.60, 1.1), main = paste("n = ", n), xlab = "p", ylab = "Coverage P
lines(c(1/n, 1- 1/n), c(1 - alpha, 1- alpha), col = "red", lty = "dashed")
      text(0.5, CL + 0.05, paste("Targeted Confidence Level =", CL))
}
```




```
par(opar)
```

7.5 BOOTSTRAP t CONFIDENCE INTERVALS

The bootstrap percentile confidence interval for giving a range of plausible values for a parameter was introduced earlier. Another confidence interval is the bootstrap t interval that is based on estimating the actual distribution of the t statistic from the data, rather than just assuming that the t statistic has a Student's t distribution.

Recall the Bangladesh arsenic levels data. The distribution of arsenic levels was skewed right.

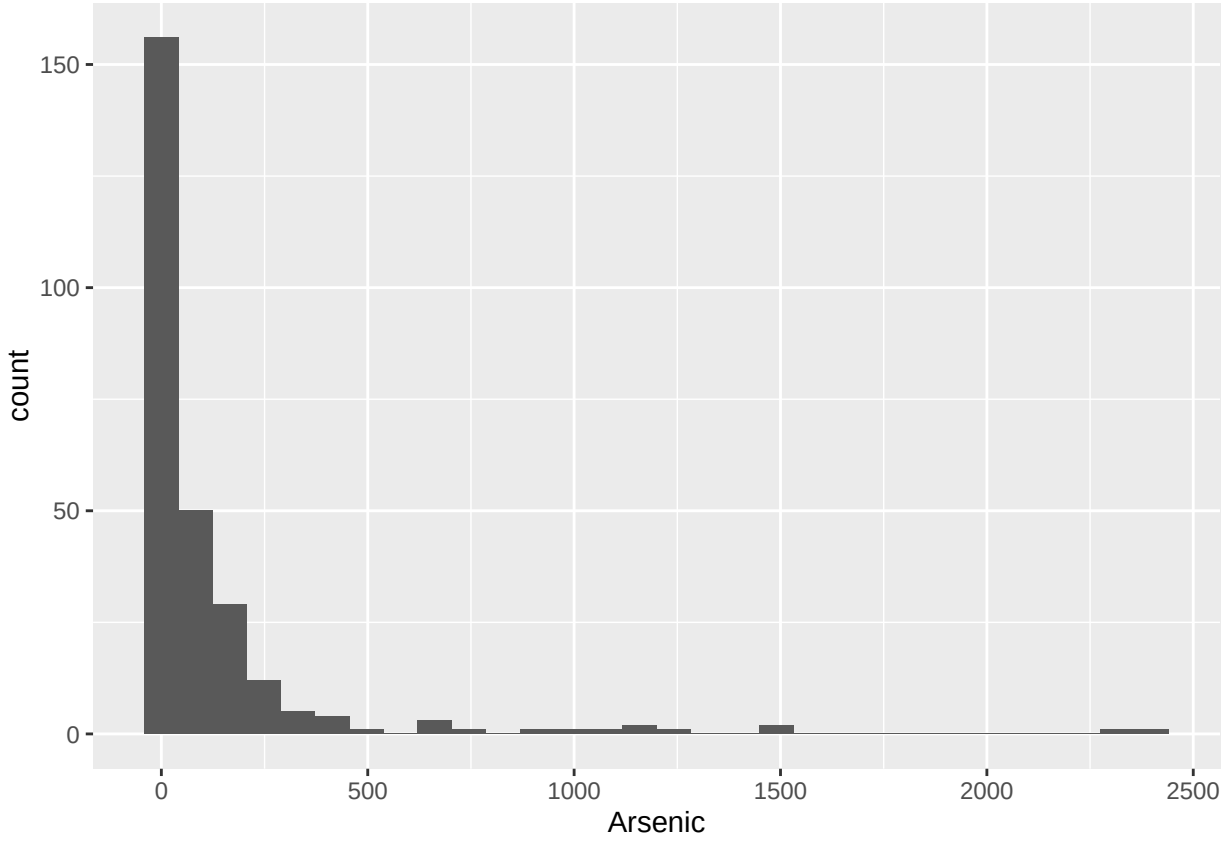
```
site <- "http://www1.appstate.edu/~arnholta/Data/Bangladesh.csv"
Bangladesh <- read.csv(file=url(site))
head(Bangladesh)
```

```
##   Arsenic Chlorine Cobalt
## 1    2400      6.2   0.42
```

```
## 2       6    116.0   0.45
## 3     904     14.8   0.63
## 4     321     35.9   0.68
## 5    1280     18.9   0.58
## 6     151      7.8   0.35
```

```
ggplot(data = Bangladesh, aes(x = Arsenic)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



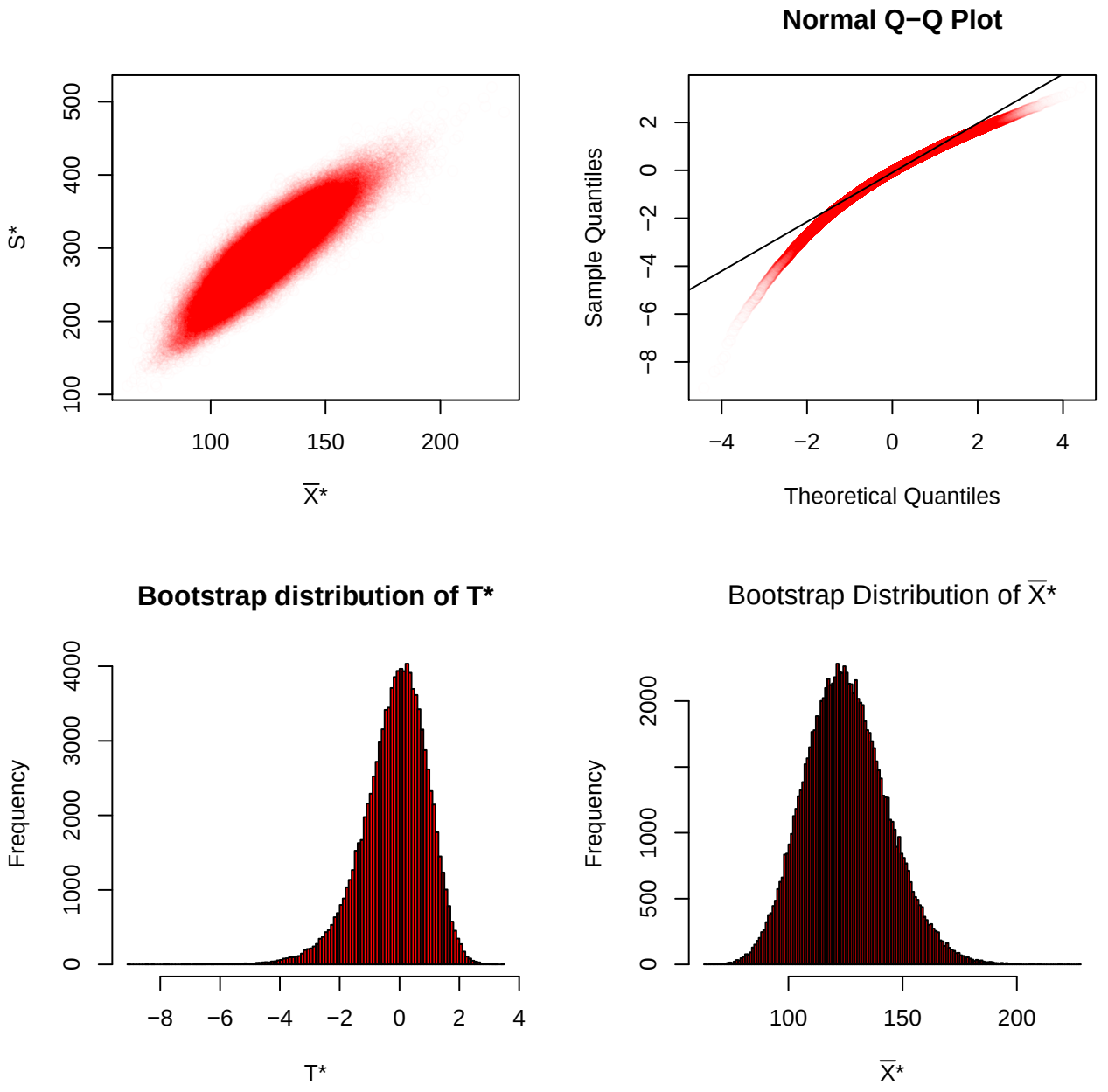
To use the t interval formula for the mean μ requires that the statistic $T = (\bar{x} - \mu)/(S/\sqrt{n})$ to follow a t_ν. That seems unlikely for data this skewed. Instead, we bootstrap the t statistic: for each of the 10^5 resamples, we compute the resample mean $\bar{X}^*$, resample standard deviation S^*, and then compute the resample T statistic $T^* = (\bar{X}^* - \bar{x})/(S^*/\sqrt{n})$

```
Arsenic <- subset(Bangladesh, select = Arsenic, drop = T)
xbar <- mean(Arsenic)
S <- sd(Arsenic)
N <- 10^5
n <- length(Arsenic)
Tstar <- numeric(N)
Sstar <- numeric(N)
Xbarstar <- numeric(N)
set.seed(13)
for (i in 1:N)
{
  x <-sample(Arsenic, size = n, replace = T)
  Xbarstar[i] <- mean(x)
  Sstar[i] <- sd(x)
```

```
}
Tstar <- (Xbarstar - xbar)/(Sstar / sqrt(n))
CIt <- quantile(Tstar, c(0.025, 0.975))
names(CIt) <- NULL
CIt
```

```
## [1] -2.647581  1.656184
```

```
opar <- par(no.readonly = TRUE)
par(mfrow= c(2, 2))
plot(Xbarstar, Sstar, ylab = "S*", xlab = substitute(paste(bar(X),"*")), col = rgb(1,0,0,0.01))
qqnorm(Tstar, col = rgb(1,0,0,0.01))
qqline(Tstar)
hist(Tstar, xlab = "T*", main = "Bootstrap distribution of T*", col = "red", breaks = "Scott")
hist(Xbarstar, xlab = substitute(paste(bar(X),"*")), main = substitute(paste("Bootstrap Distribution of
```



```
par(opar)
```

Note that the bootstrap distribution for the t statistic is left skewed; in fact, it is more left skewed than the the bootstrap distribution of the mean is right skewed! The reason for this is the strong positive relationship between $\bar{X}^*$ and S^*. If a bootstrap resample contains a large number of the big values from the right tail of the original data, then $\bar{X}^*$ is large and hence S^* is especially large (standard deviations are computed by squaring distances from the mean, so they are affected even more by large observations than a mean is). The large denominator thus keeps T^* from being particularly large. Conversely, when there are relatively few of the big observations in the resample, then $\bar{X}^* - \bar{x}$ is negative and the denominator can be especially small, thus resulting in a T ratio that is large negative.

The 2.5% and 97.5% percentiles of the bootstrap t distribution are -2.6475811 and 1.656184, compared to $\pm$ 1.968789 for the Student's t distribution. This is a reflection of the skewed nature of the bootstrap t distribution compared to the symmetric Student's t distribution.

Before proceeding with the bootstrap t, let us think about what skewness implies for the accuracy of the formula-based t confidence intervals ($\bar{X} \pm t_{1-\alpha/2,\nu} \times S/\sqrt{n}$). For right-skewed data, when $\bar{X} < \mu$, typically $S < \sigma$, so the confidence interval tends to be narrow, and the interval falls below μ more often than $\alpha/2 \times 100\%$ of the time. This is bad. Conversely, when $\bar{X} > \mu$, typically $S > \sigma$, so the intervals tend to be to wide and do not miss μ often enough; this is also bad. Overall, the intervals tend to be to the left of where they should be and give a biased picture of where the mean is likely to be.

We return to the bootstrap t and consider samples from nonnormal populations. Let $T = (\bar{X} - \mu)/(S/\sqrt{n})$ and F be the cdf for the T statistic (the cdf of the sampling distribution). Let Q_1 and Q_2 denote the $\alpha/2$ and $(1 - \alpha/2)$ quantiles of this distribution; that is, $Q_1 = F^{-1}(\alpha/2)$ and $Q_2 = F^{-1}(1 - \alpha/2)$. Then

$$1 - \alpha = P(Q_1 < T < Q_2) = P\left(Q_1 < \frac{\bar{X} - \mu}{S/\sqrt{n}} < Q_2\right).$$

This suggest the confidence interval:

$$CI_{1-\alpha}(\mu) = \left(\bar{X} - Q_2\frac{S}{\sqrt{n}}, \bar{X} - Q_1\frac{S}{\sqrt{n}}\right).$$

The quantiles Q_1 and Q_2 are unknown, but they can be estimated using quantiles of the bootstrap distribution of the t statistic:

$$T^* = \frac{\bar{X}^* - \bar{x}}{S^*/\sqrt{n}},$$

where $\bar{X}^*$ and S^* are the mean and standard deviation of a bootstrap resample. We use the standard error formula for every bootstrap sample because the bootstrap statistic should mimic $T = (\bar{X} - \mu)/(S/\sqrt{n})$.

Thus Q_1 = -2.6475811 and Q_2 = 1.656184, so we compute

```
LL <- xbar - CIt[2]*S/sqrt(n)
UL <- xbar - CIt[1]*S/sqrt(n)
c(LL, UL)
```

```
## [1]  95.3418 173.2431
```

The 95% bootstrap percentile interval is (92.3224354, 163.2506181) while the formula t confidence interval is (89.6834234, 160.956429). The bootstrap t interval is stretched further to the right, reflecting the right-skewed distribution of the data. Because of the large sample size, we report the 95% bootstrap t confidence interval (95.3417995, 173.2430563) μ g/dL.

Using the package `boot`

Next we consider computing the bootstrap percentile and t confidence intervals using the functions `boot()` and `boot.ci()` functions from the package `boot`. We write the function `mean.boot()` and use `mean.boot()` inside the `boot()` function storing the results in the object `boot.out`. Finally, the function `boot.ci()` is applied to `boot.out` which results in the creation of the percentile and confidence intervals.

```
require(boot)
```

```
## Loading required package: boot
```

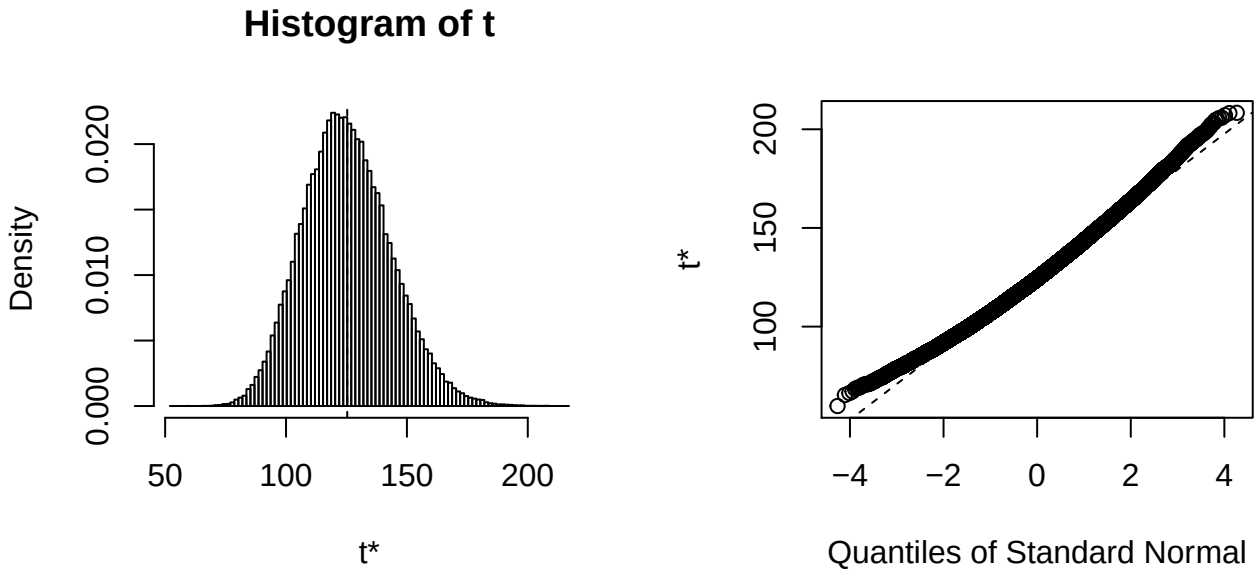
```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

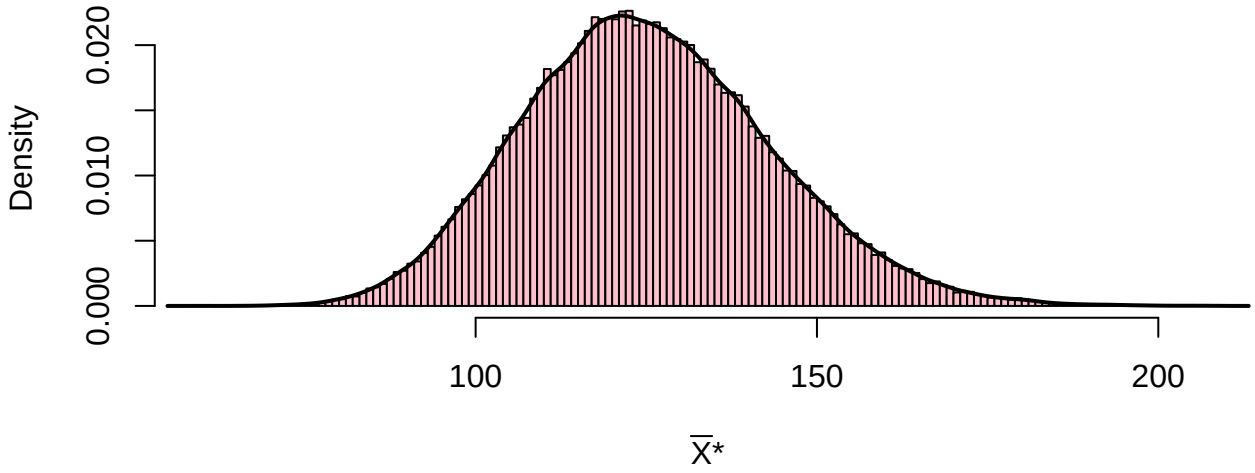
```
mean.boot <- function(data, i){
  d <- data[i]
  M <- mean(d)
  V <- var(d)/length(i)
  return(c(M, V))
}
boot.out <- boot(Arsenic, mean.boot, R=10^5)
boot.ci(boot.out, conf = 0.95, type = c("perc", "stud"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 100000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.out, conf = 0.95, type = c("perc", "stud"))
##
## Intervals :
## Level    Studentized          Percentile
## 95%   ( 95.4, 173.3 )   ( 92.4, 163.1 )
## Calculations and Intervals on Original Scale
```

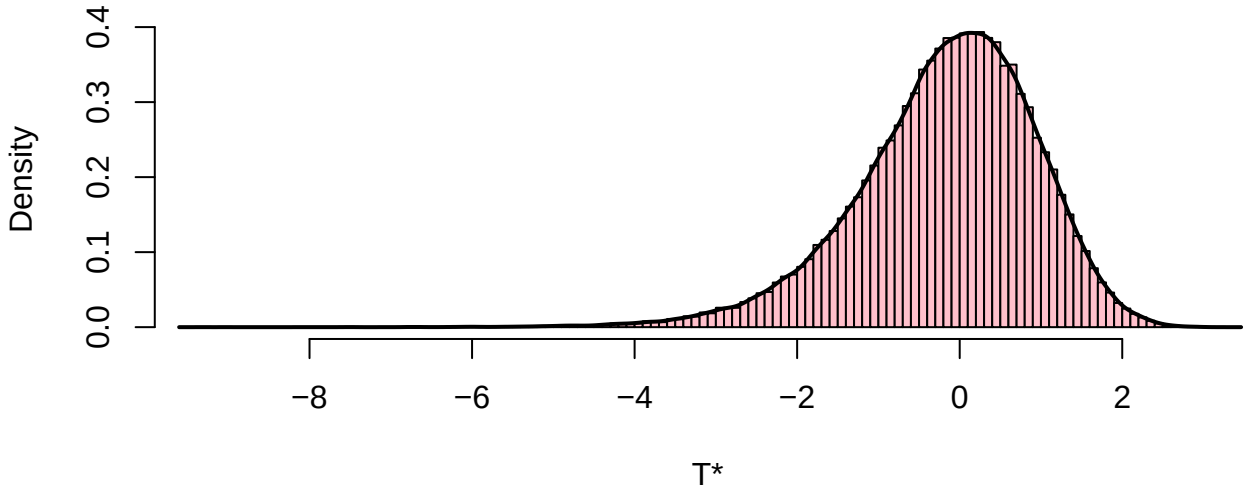
```
plot(boot.out)
```



```r
hist(boot.out$t[,1], col = "pink", breaks = "Scott", main = "", xlab = substitute(paste(bar(X),"*")), fr
lines(density(boot.out$t[,1]), lwd = 2)
```




```r
hist((boot.out$t[,1] - boot.out$t0[1])/(boot.out$t[,2])^.5, col = "pink", breaks = "Scott", main = "", x
lines(density((boot.out$t[,1] - boot.out$t0[1])/(boot.out$t[,2])^.5), lwd = 2)
```



The bootstrap intervals for a difference in means follows the same idea.

BOOTSTRAP t CONFIDENCE INTERVAL FOR $\mu_1 - \mu_2$ For each of many resamples, calculate the bootstrap t statistic

$$T^* = \frac{\bar{X_1}^* - \bar{X_2}^* - (\bar{x_1} - \bar{x_2})}{\sqrt{S_1^{2*}/n_1 + S_2^{2*}/n_2}},$$

Let Q_1^* and Q_2^* be the empirical $\alpha/2$ and $(1-\alpha/2)$ quantiles of the bootstrap t distribution, respectively. The bootstrap t confidence interval is

$$\left((\bar{x_1} - \bar{x_2}) - Q_2^* \times \sqrt{s_1^2/n_1 + s_2^2/n_2}, (\bar{x_1} - \bar{x_2}) - Q_1^* \times \sqrt{s_1^2/n_1 + s_2^2/n_2} \right).$$

Recall the Verizon example, where we considered the difference in means of two very skewed distributions of repair times for two very unbalanced samples ($n_1 = 23$ versus $n_2 = 1664$). Let us look at the data again.

```
site <- "http://www1.appstate.edu/~arnholta/Data/Verizon.csv"
Verizon <- read.csv(file=url(site))
Time.ILEC <- subset(Verizon, select=Time, Group == "ILEC", drop=TRUE)
Time.CLEC <- subset(Verizon, select=Time, Group == "CLEC", drop=TRUE)
thetahat <- mean(Time.ILEC) - mean(Time.CLEC)
nx <- length(Time.ILEC)  #nx=1664
ny <- length(Time.CLEC)  #ny=23
SE <- sqrt(var(Time.ILEC)/nx + var(Time.CLEC)/ny)
N <- 10^4
Tstar <- numeric(N)
DM <- numeric(N)
set.seed(1)
for(i in 1:N)
{
  bootx <- sample(Time.ILEC, nx, replace=TRUE)
  booty <- sample(Time.CLEC, ny, replace=TRUE)
  Tstar[i] <- (mean(bootx) - mean(booty) - thetahat) /
    sqrt(var(bootx)/nx + var(booty)/ny)
  DM[i] <- mean(bootx) - mean(booty)
}
CItboot <- thetahat - quantile(Tstar, c(.975, .025)) * SE
names(CItboot) <- NULL
CItboot
```

```
## [1] -22.375169  -2.221755
```

```
CIperct <- quantile(DM, c(0.025, 0.975))
CIperct
```

```
##       2.5%      97.5%
## -16.644491  -1.595858
```

```
t.test(Time.ILEC, Time.CLEC)$conf
```

```
## [1] -16.5568985   0.3618588
## attr(,"conf.level")
## [1] 0.95
```

The 95% bootstrap t interval for the difference in means is (-22.3751689, -2.2217546). For comparison, the formula t interval is (-16.5568985, 0.3618588) and the bootstrap percentile interval is (-16.6444915, -1.5958584). The more accurate bootstrap t interval stretches farther in the negative direction, even more than the bootstrap percentile interval.

The same basic procedure can also be used for confidence intervals for statistics other than one or two means—to compute a t statistic for each of many resamples, using the appropriate standard error for $\hat{\theta}$, find the quantiles of that bootstrap t distribution and create an estimate of the form $(\hat{\theta} - Q_2^* \times SE, \hat{\theta} - Q_1^* \times SE)$.

Using the package `boot` for the difference in means

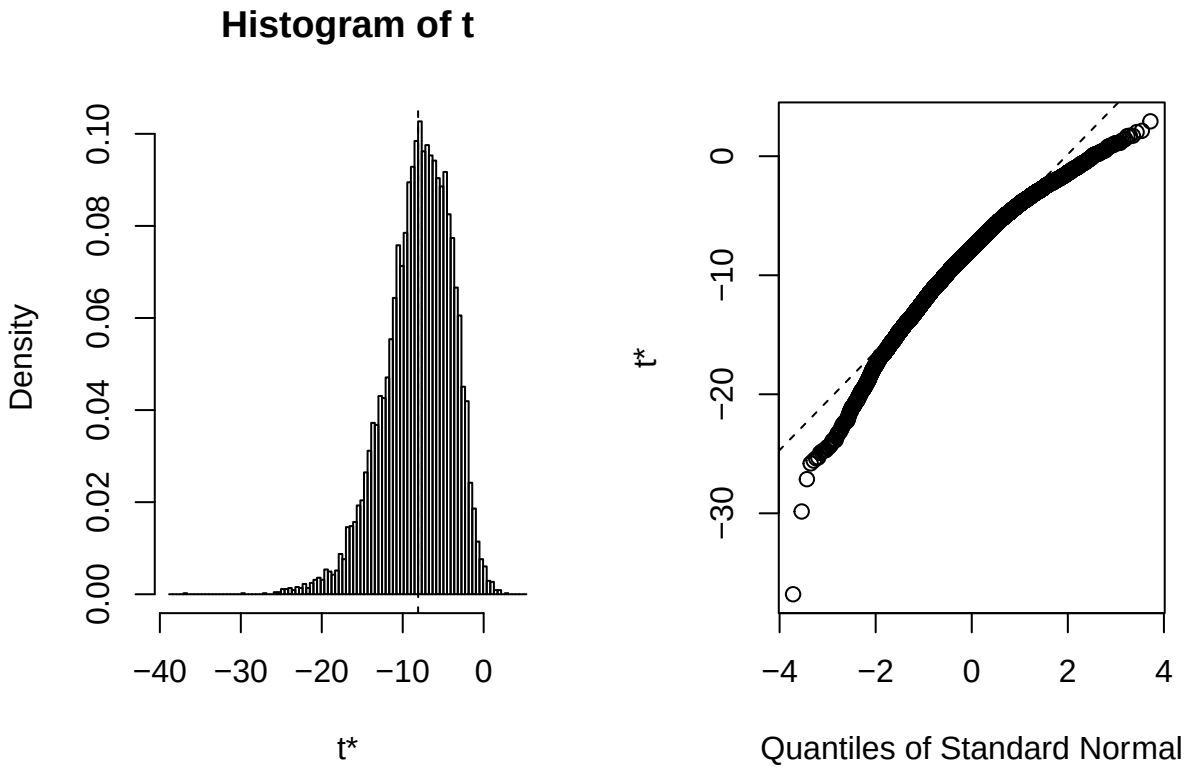
Bootstrap percentile and t confidence intervals using the functions `boot()` and `boot.ci()` functions from the package `boot` are constructed using the user created `mean2.boot()` function.

```
require(boot)
mean2.boot <- function(data, i){
  d <- data[i, ]
  M <- tapply(d$Time, d$Group, mean)
  V <- tapply(d$Time, d$Group, var)/tapply(d$Time, d$Group, length)
  return(c(M[2] - M[1], V[2] + V[1]))
```

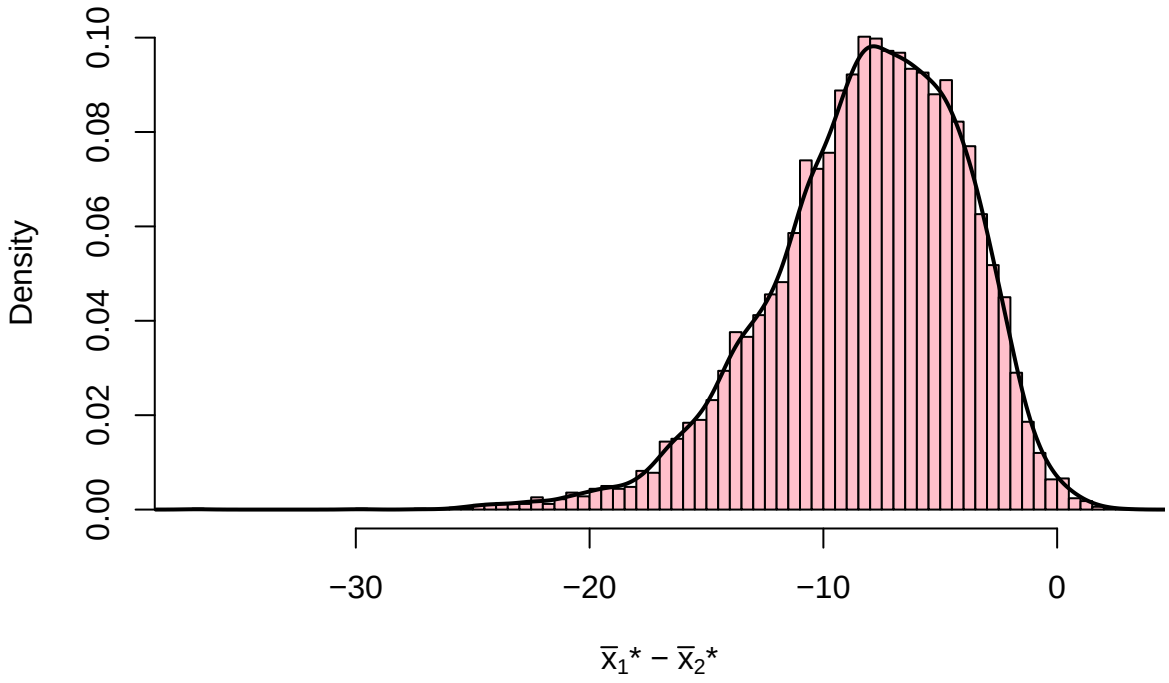
```
}
set.seed(1)
boot.out <- boot(Verizon, mean2.boot, R=10^4)
boot.ci(boot.out, conf = 0.95, type = c("perc", "stud"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.out, conf = 0.95, type = c("perc", "stud"))
##
## Intervals :
## Level    Studentized          Percentile
## 95%   (-22.686,  -2.130 )   (-17.260,  -1.536 )
## Calculations and Intervals on Original Scale
```

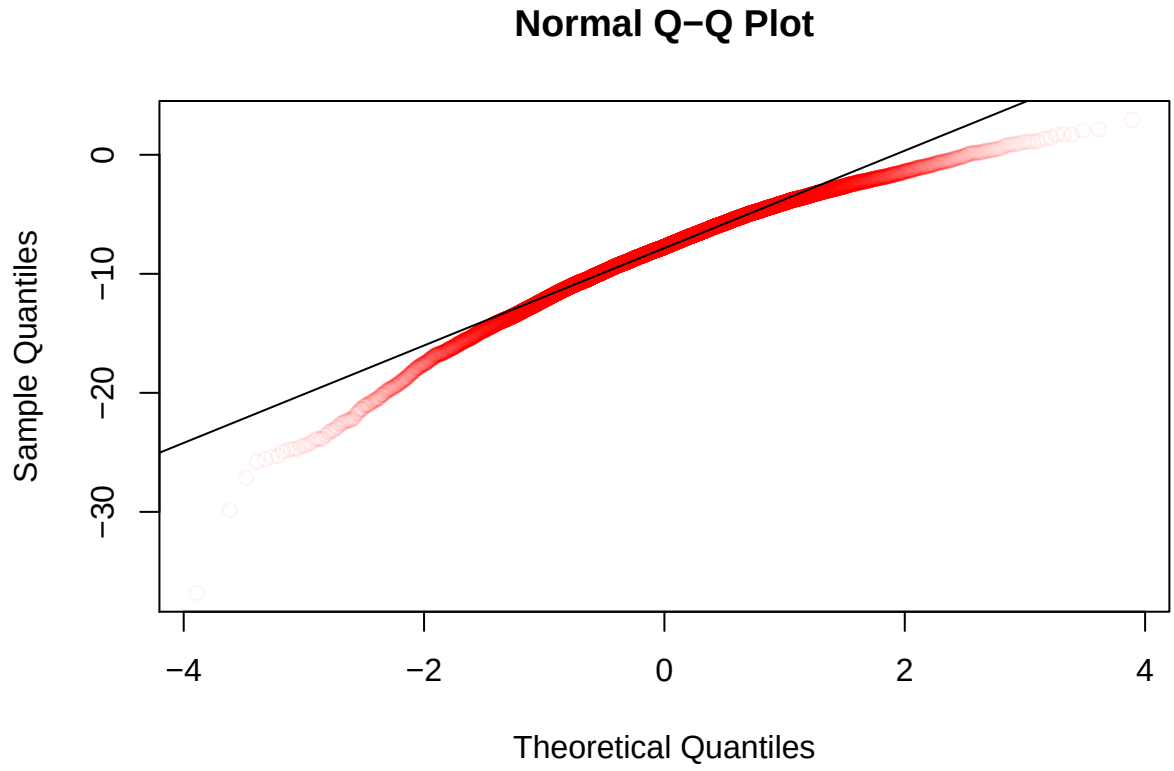
```
plot(boot.out)
```




```
hist(boot.out$t[,1], col = "pink", breaks = "Scott", main = "", freq= FALSE, xlab = substitute(paste(ba
lines(density(boot.out$t[,1]), lwd = 2)
```

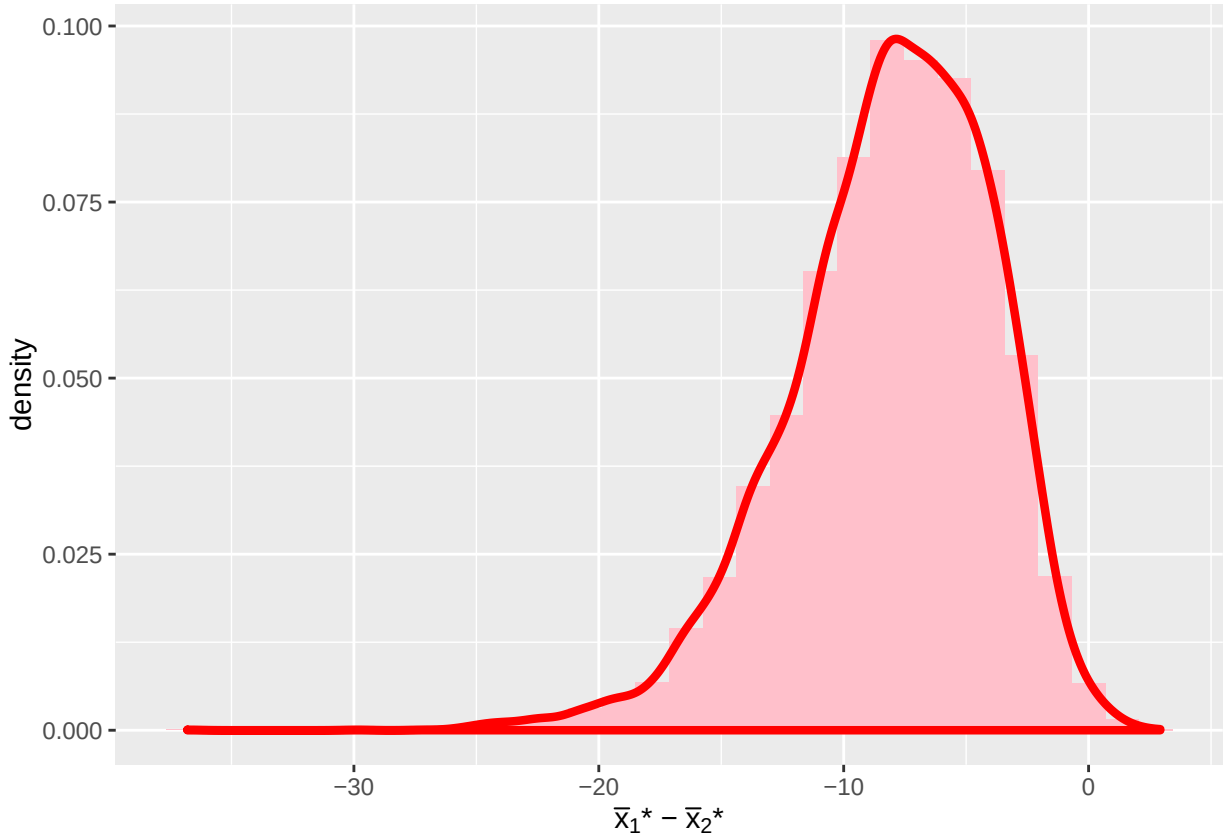



```
qqnorm(boot.out$t[,1], col =rgb(1,0,0,.05))
qqline(boot.out$t[,1])
```

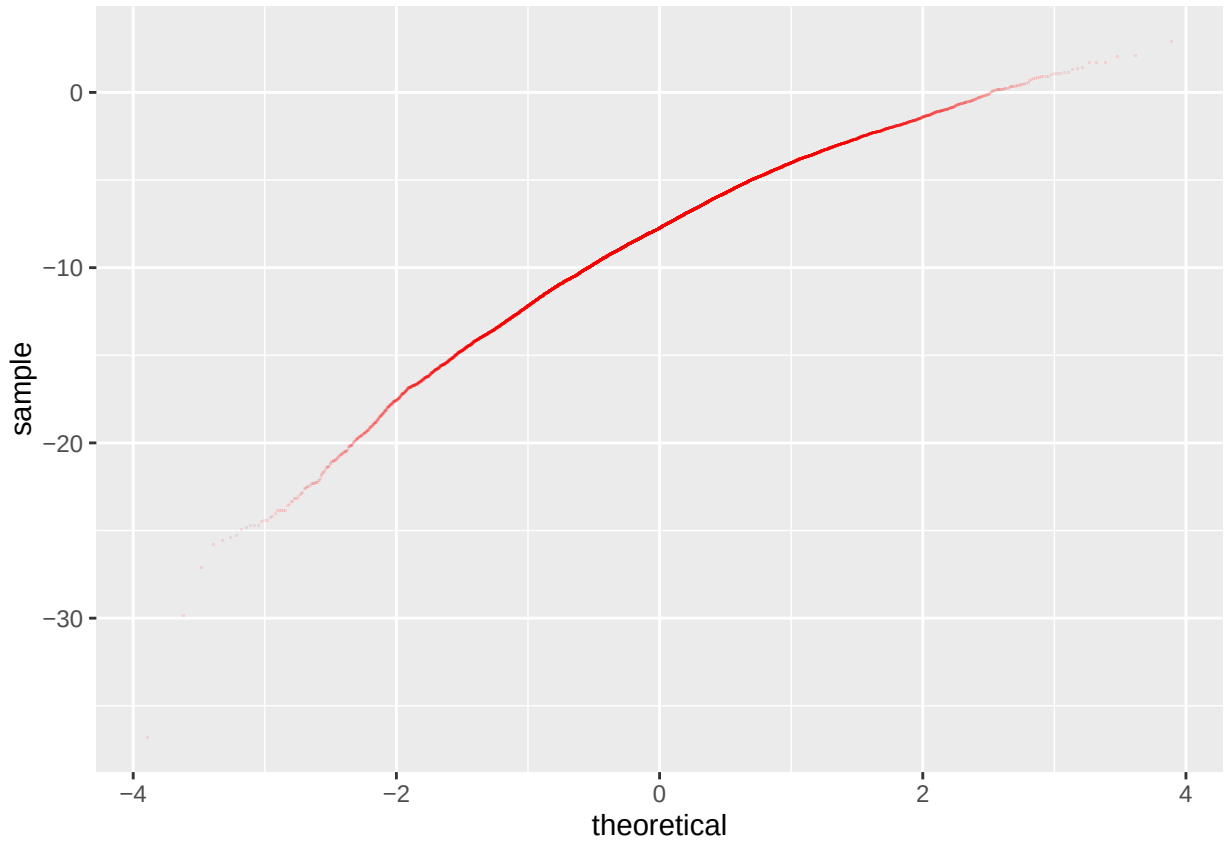



```
# ggplot2 Now
require(ggplot2)
BO <- as.data.frame(boot.out$t)
ggplot(data = BO, aes(x = V1, y = ..density..)) + geom_histogram(fill = "pink") + xlab(substitute(paste
```

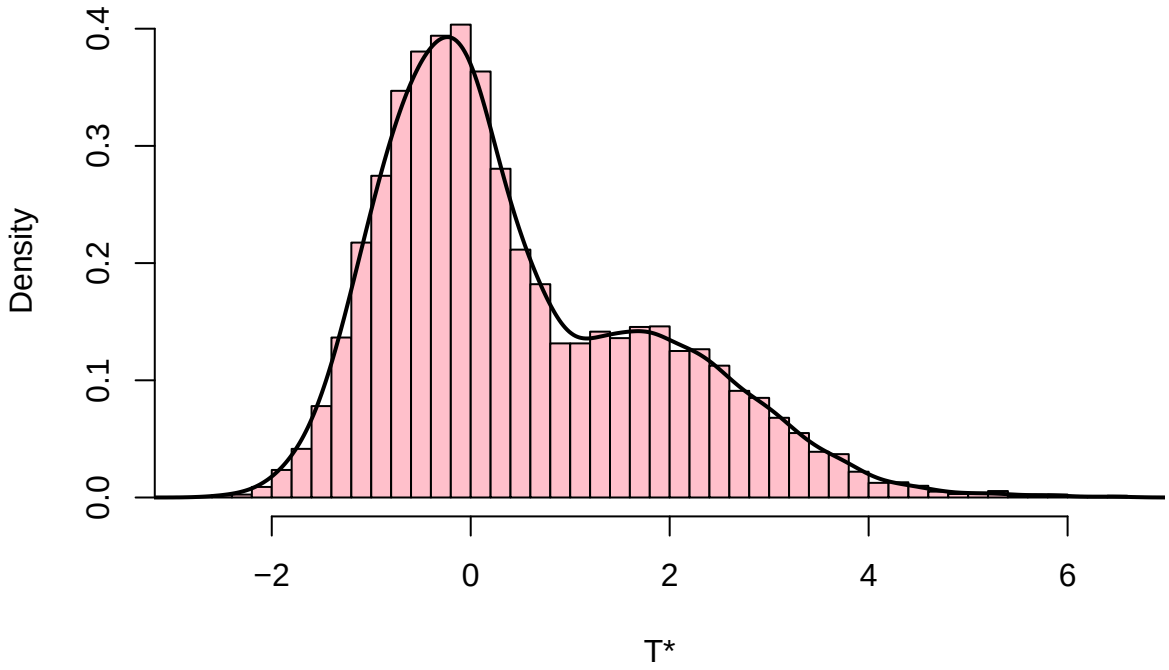
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



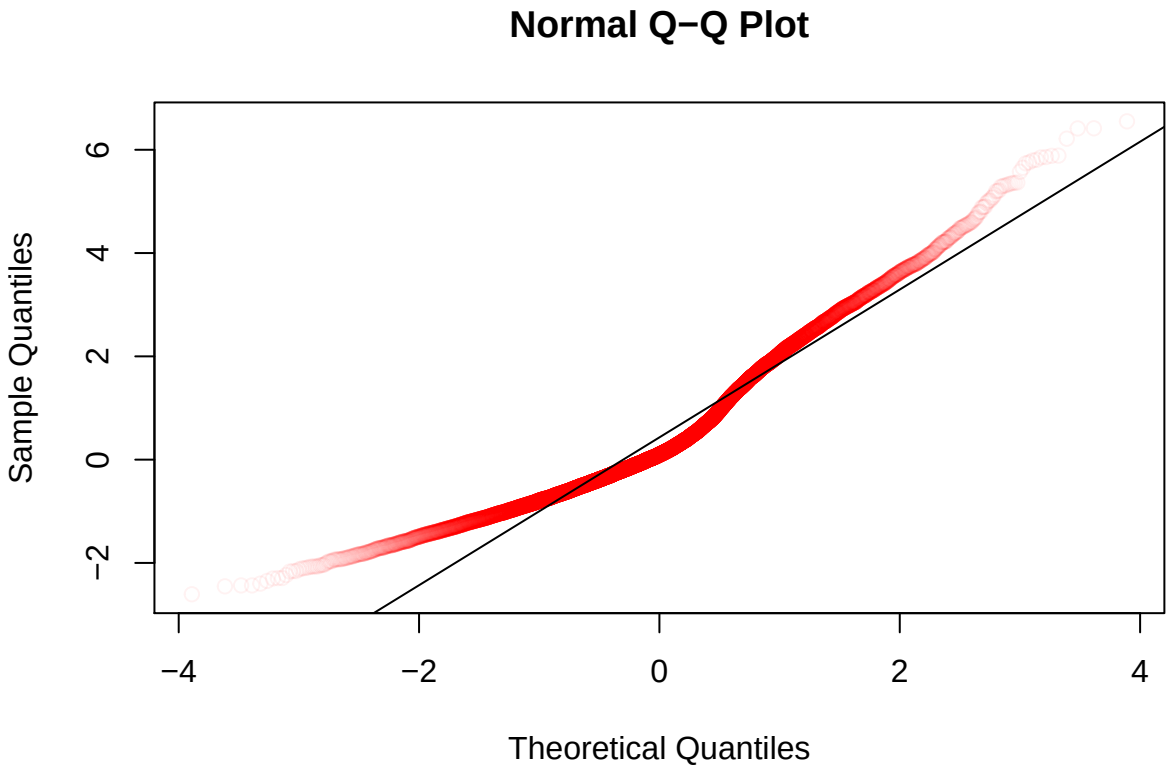

```
p <- ggplot(data = BO, aes(sample = V1)) + stat_qq(pch = ".", color = rgb(1, 0, 0, 0.1))
p
```




```
hist((boot.out$t[,1] - boot.out$t0[1])/(boot.out$t[,2])^.5, col = "pink", breaks = "Scott", main = "", 
lines(density((boot.out$t[,1] - boot.out$t0[1])/(boot.out$t[,2])^.5), lwd = 2)
```




```
qqnorm((boot.out$t[,1] - boot.out$t0[1])/(boot.out$t[,2])^.5, col = rgb(1,0,0,.05))
qqline((boot.out$t[,1] - boot.out$t0[1])/(boot.out$t[,2])^.5)
```



7.5.1 Comparing Bootstrap t and Formula t Confidence Intervals It is useful to compare the bootstrap distribution to classical statistical inferences. With classical t intervals of the form $\bar{x} \pm t \times s/\sqrt{n}$, the confidence interval width varies substantially in small samples as the sample standard deviation s varies. Correspondingly, the classical standard error $s/\sqrt{n}$ varies as s varies. The bootstrap is no different in this regard —bootstrap standard errors and widths of confidence intervals for the mean are proportional to s.

Where the bootstrap does differ from classical inference is in how it handles skewness. The bootstrap percentile interval and bootstrap t interval are in general asymmetrical, with asymmetry depending on the sample. These intervals estimate the skewness of a population from the skewness of the sample. In contrast, classical t intervals assume that the population has no underlying skewness (skewness is 0).

Which is preferred? Frankly, neither, but rather something in between. This is an area that needs attention from statistical researchers. Until then, we will recommend the formula t if $n \leq 10$, and the bootstrap t otherwise; the reason being in large samples, we should put more trust in the data—in this case, the bootstrap t is preferred. In small samples, the classical procedure is probably better—if the sample size is small then skewness cannot be estimated accurately from the sample, and it may be better to assume that there is no skewness (skewness is 0) in spite of the bias, rather than to use an estimate that has high variability. Something between classical intervals and the bootstrap procedures would be best—something that makes a trade-off between bias and variance and transitions smoothly from being like the formula t for small n and bootstrap t for large n.

The bootstrap percentile makes less of a skewness correction than does the bootstrap t. Hence, for smaller samples, it is less variable than the bootstrap t. For larger samples, the bootstrap t is preferred. In the long run, increasing the sample size by a factor of 10 reduces the coverage errors of the bootstrap t intervals by a factor of 10 but reduces the errors of symmetric formula intervals and the bootstrap percentile interval only by a factor of $\sqrt{10}$.

For comparing two samples, if the sample sizes are equal or nearly equal, you may use the formula t for every sample size unless there is reason to believe that the skewness differs between the two populations. However, it is good to also do the bootstrap t interval except for small samples.

Skewness

You could write your own function to compute skewness or use one that has already been written. The function `skewness()` in the `e1071` package will work just fine for your homework.

```
require(e1071)
skewness(Verizon$Time)
```

```
## [1] 4.52238
```