Chapter 16 Solutions

This chapter uses computationally demanding resampling procedures for which the use of a computer is critical. We used S-PLUS while writing this chapter, and give commands below for performing the analyses in S-PLUS. Our goal is to make it as easy as possible for you, and students, to focus on the statistical aspects of this chapter rather than the commands. You may use S-PLUS for this chapter even if you use other software for other chapters.

To use S-PLUS you need:

- A copy of S-PLUS; there is a free version for students at
  http://elms03.e-academy.com/splus
  Instructors should see:
  http://www.insightful.com/industry/academic
- The S+Resample package:
  http://www.insightful.com/downloads/libraries
- The data, as an S-PLUS package:
  http://www.insightful.com/Hesterberg/bootstrap/IPS6data.zip

Future versions of the packages might ship with S-PLUS or be found at:
http://csan.insightful.com

The Web site for resampling within *Introduction to the Practice of Statistics*, *6th edition*, is:
http://www.whfreeman.com/ips6e_resample
You may check that page for updated information.

In these solutions, we provide answers, example figures, and S-PLUS code used to obtain the answers and figures. The S-PLUS code is largely identical to example code included in the IPS6data package in file:
`IPS6data/doc/exercises16.ssc`
It differs from the example code in that:

- Here we specify random number seeds to make results reproducible; in the examples shown to students we do not, because we want them to experience randomness.
- The examples here include a few more details, and calculations that would give away the answers.

The solutions to exercises (3, 17, 19, 55) contain general comments that apply to many other exercises.

To use a script file such as `exercises16.ssc`, open it in S-PLUS (double-click it within Windows to open S-PLUS). To run code in a script window, select the lines you want to run, then hit F10 or click the triangle in the S-PLUS menu bar.

Here and in `exercises16.ssc` we use command-line code, but the menu-driven interface is easier to use. Documentation on using the menu interface is contained in *Student Guide to Use of Menus for Resampling*, stored as `studentResample.pdf` in the IPS6data package, and in the *S-PLUS Manual for Moore and McCabe's Introduction to the Practice of Statistics*, available from W. H. Freeman and Company.

16.1. Student answers in this problem will vary substantially due to using different random numbers.
(c) Here is a stemplot for 200 resamples (students do 20):

```
  3 : 999
  4 : 78
  5 : 03999
  6 : 578
  7 : 134499
  8 : 02255888
  9 : 133347
 10 : 0002233468999
 11 : 111123477777888888
 12 : 0012333456689999
 13 : 115557788888888
 14 : 00000033344445666668899
 15 : 023555688889
 16 : 112344666679
 17 : 002356668889
 18 : 011122444444777899
 19 : 0156667999
 20 : 225
 21 : 147
 22 : 022258
 23 : 07
 24 : 359
 25 : 1
 26 :
 27 :
 28 :
 29 : 2
```

(d) The standard error (SE) for 1000 resamples is 4.643 (students do only 20 so their numbers may be substantially higher or lower).

Additional comment – for an infinite number of resamples the answer would be 4.69, which is smaller than s/sqrt(n) by a factor sqrt(5/6). See the general comment after 16.17.

This is normally done by hand, but the following commands would work:

```
Exercise16.001 = data.frame(Time = c(26.47, 0.00, 5.32, 17.30, 29.78,
3.67))
boot1 = bootstrap(Exercise16.001$Time, mean, B=20, seed=0)
stem(boot1$replicates)
boot1  # The standard error is the printed "SE"

boot = bootstrap(Exercise16.001$Time, mean, B=200, seed=0)
stem(boot$replicates)
bootstrap(Exercise16.001$Time, mean, seed=0) # SE = 4.643
```

16.2. The standard deviation of a sample measures the spread of that sample. The SE estimates how much a sample mean would vary if you take the means of many samples from the same population. The SE is smaller by a factor of $\sqrt{\ }$.

16.3. (a) The bootstrap samples from the *data*, not the *population*. **(b)** The bootstrap samples *with replacement*. **(c)** The sample size should be *the same as* the original sample. **(d)** The bootstrap distribution is usually similar to the sampling distribution *in shape and spread, but not in center*.

General Comment—Normal Quantile Plots for Bootstrap Distributions
In the next few examples we'll be looking at bootstrap distributions, not data distributions. Our criteria for acceptable deviations from Normality are much stricter than when looking at data distributions. The bootstrap distributions are the result after the central limit theorem has had its one and only chance to work. This applies throughout this chapter.

For a quick assessment of how far from Normality this is, and what effect that might have on confidence intervals, do:

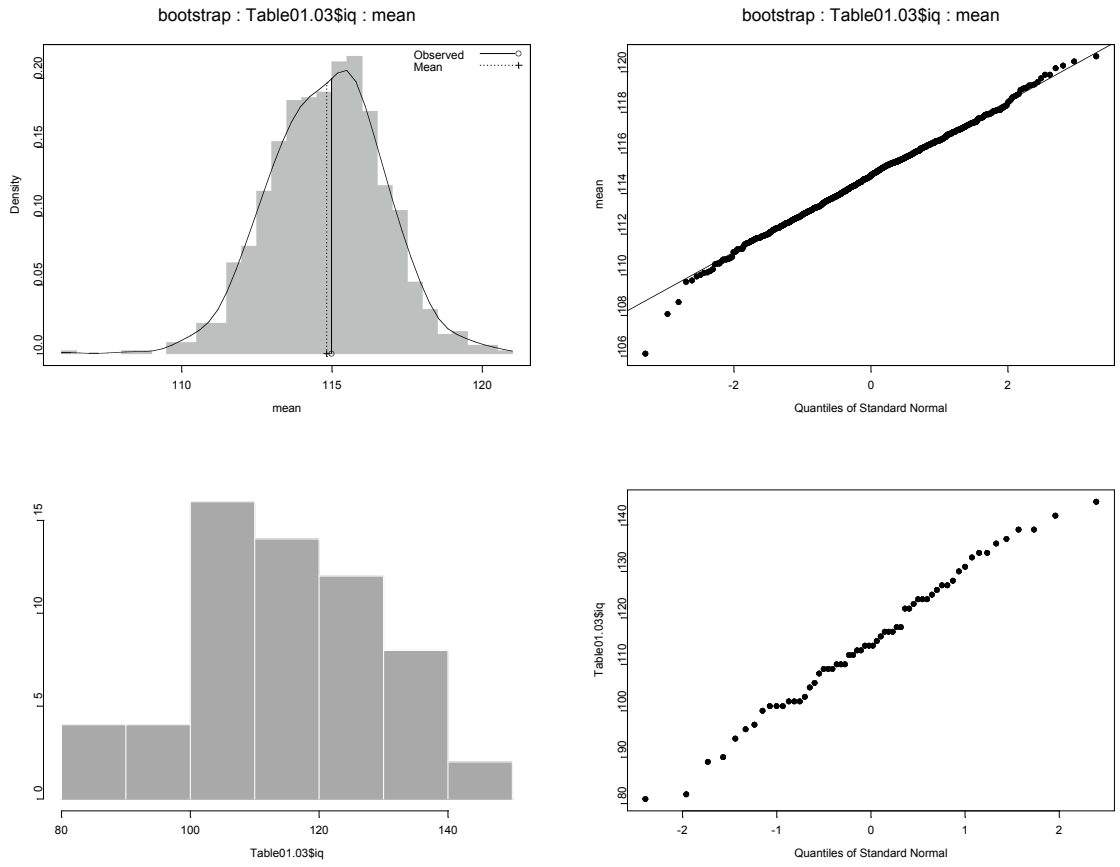
```
cdf(boot7$replicates, limits.t(boot7, z=T))
# Gives 0.012 0.029 0.944 0.957 - 1.2% on the left, 4.3% on the right
# for nominal .025, .05, .95, .975 values
```

This gives (.012, .957). In rough terms, this means that a confidence interval that should miss 2.5% of the time on either side would miss 1.2% on one side and 4.3% of the time on the other, for a sum of errors of 3.1%. (Actually this is an underestimate, based on the bootstrap percentile interval; the more accurate BCa and tilting intervals are affected even more by skewness. These intervals are discussed later in the chapter.)

The errors on the two sides combine to give a biased picture of where the parameter is likely to be.

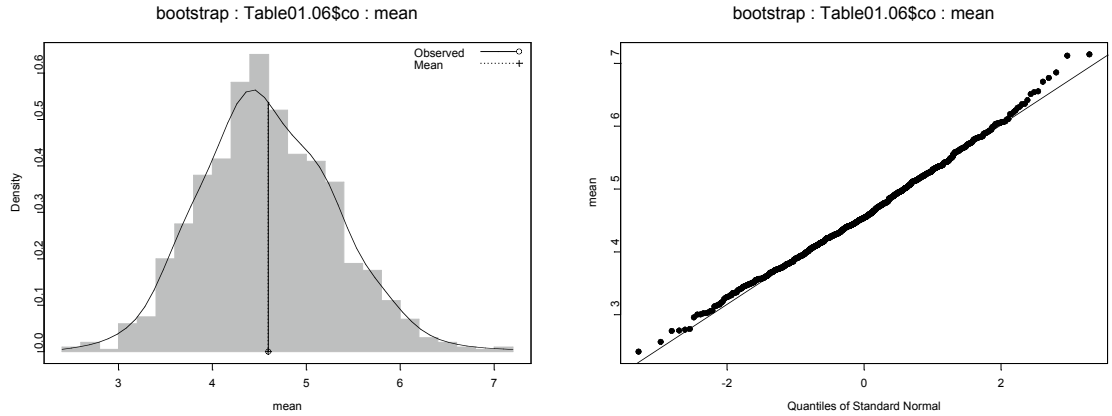
16.4. The bootstrap distribution is close to Normal (except in the left tail), so we expect the sampling distribution to also be close to Normal.

```
boot4 = bootstrap(Table01.03$iq, mean, seed=0)
plot(boot4)
qqnorm(boot4)
# Note - the book does not ask for this, but to view a histogram and
# Normal quantile plot of the original data, we do:
hist(Table01.03$iq)
qqnorm(Table01.03$iq)
```



16.5. The bootstrap distribution is close to Normal, with some positive skewness. We expect the sampling distribution to be close to Normal.

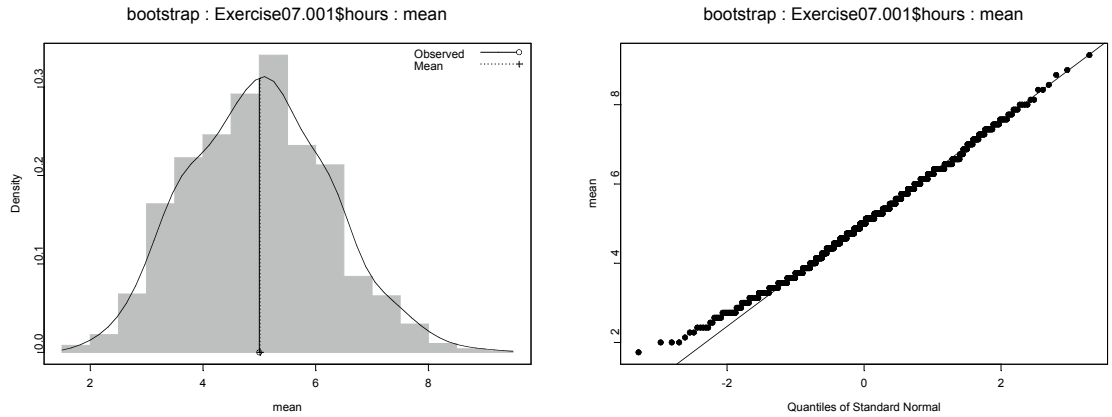
```
boot5 = bootstrap(Table01.06$co, mean, seed=0)
plot(boot5)
qqnorm(boot5)
```



16.6. The bootstrap distribution shows some positive skewness and mild discreteness. However, the discreteness is due to the small sample size, and the skewness may be too. With a sample size this small the sample skewness may vary substantially, so we cannot say if the sampling distribution is really skewed.

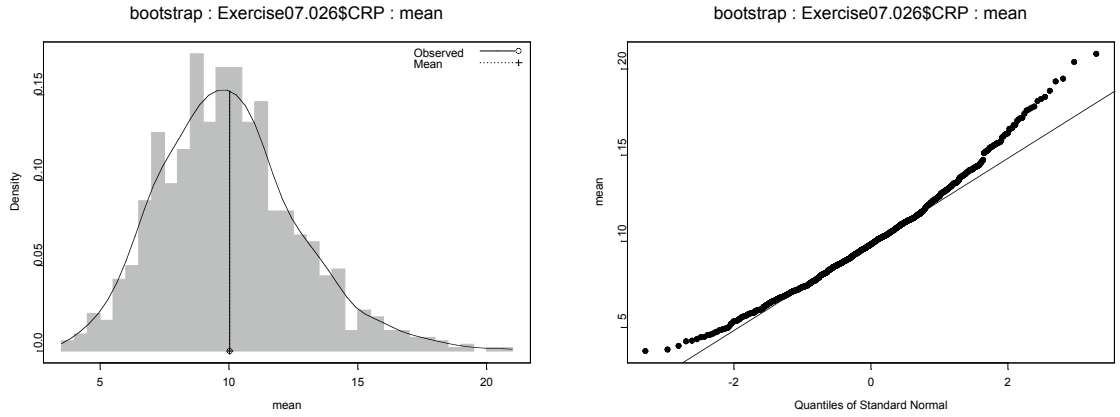
I recommend not assigning this exercise. The small-sample variability in skewness is not discussed until Section 16.3. The bootstrap is not well suited to datasets this small. —Tim Hesterberg

```
Exercise07.001 = data.frame(hours = c(5, 6, 0, 4, 11, 9, 2, 3))
boot6 = bootstrap(Exercise07.001$hours, mean, seed=0)
plot(boot6)
qqnorm(boot6)
```



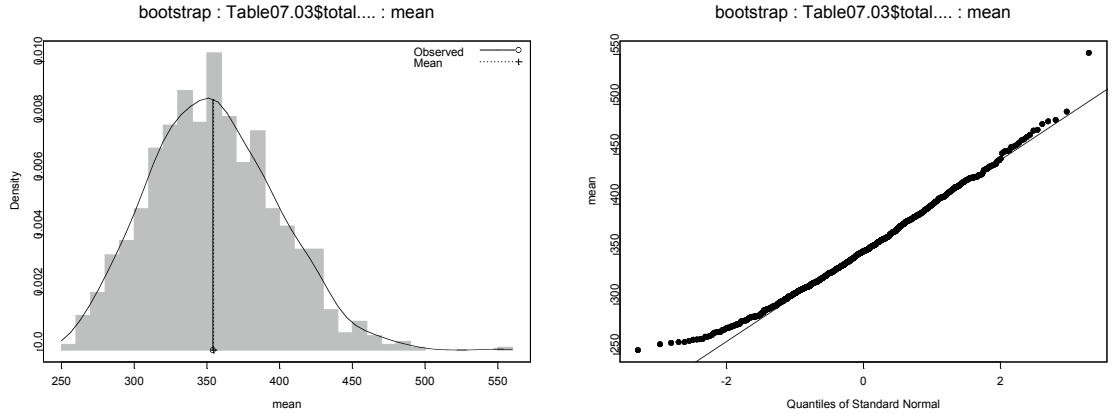
16.7. The bootstrap distribution is strongly non-Normal, so we expect the sampling distribution to also be strongly non-Normal.

```
boot7 = bootstrap(Exercise07.026$CRP, mean, seed=0)
plot(boot7)
qqnorm(boot7)
cdf(boot7$replicates, limits.t(boot7, z=T))
# 0.012 0.029 0.944 0.957 -- 1.2% on the left, 4.3% on the right
```



16.8. The bootstrap distribution is non-Normal, with positive skewness. We expect the sampling distribution to be skewed. This amount of skewness would not be a concern with data, but it is when working with bootstrap distributions.

```
Table07.03 # min sec total.secs
boot8 = bootstrap(Table07.03$total.secs, mean, seed=0)
plot(boot8)
qqnorm(boot8)
cdf(boot8$replicates, limits.t(boot8, z=T))
# 0.013 0.037 0.941 0.972 - 1.3% on the left, 2.8% on the right
```



16.9. (a) Formula 1.91, bootstrap 1.877; quite close. **(b)** Formula 0.696, bootstrap 0.71; quite close. **(c)** Formula 1.28, bootstrap 1.24; quite close. In part (c), student bootstrap answers may be substantially smaller–the bootstrap is biased downward for estimating standard error by a factor of about sqrt((*n* – 1)/*n*), which is substantial when *n* is only 8. See the general comment after 16.17.

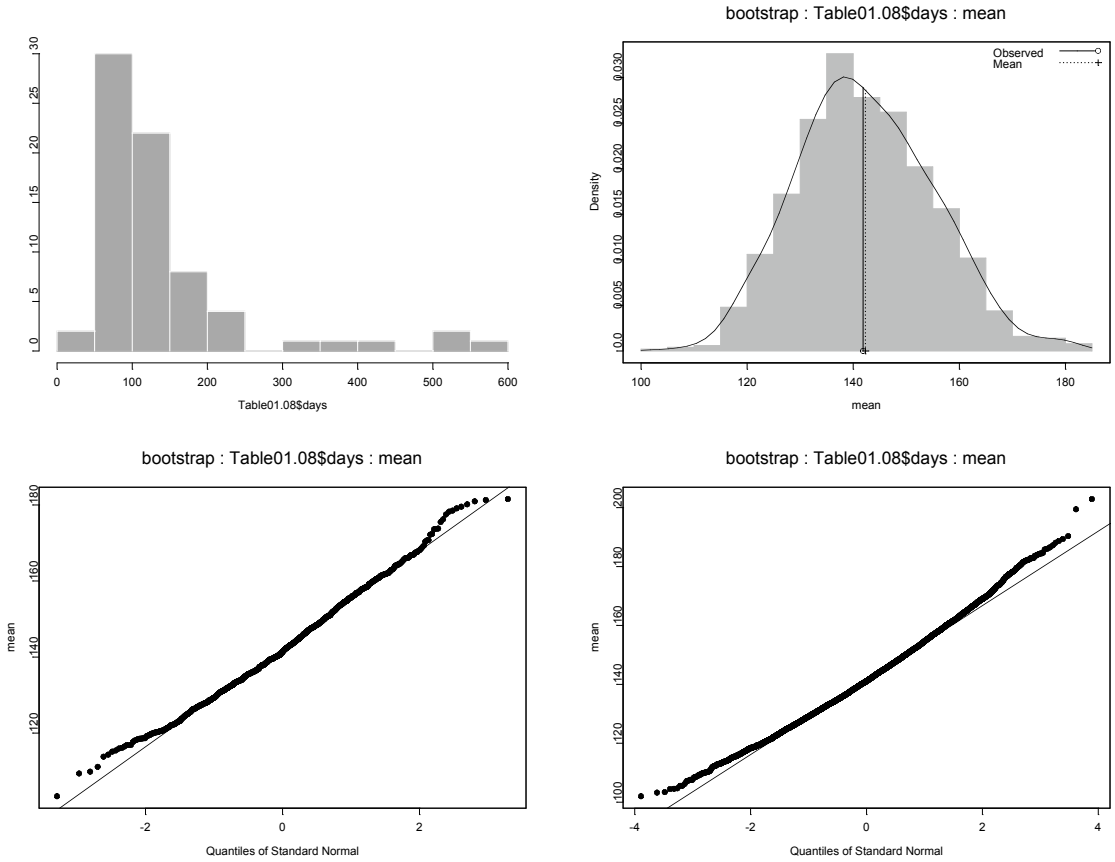
```
stdev(Table01.03$iq) / sqrt(nrow(Table01.03))
boot4

stdev(Table01.06$co) / sqrt(nrow(Table01.06))
boot5

stdev(Exercise07.001$hours) / sqrt(8)
boot6
```

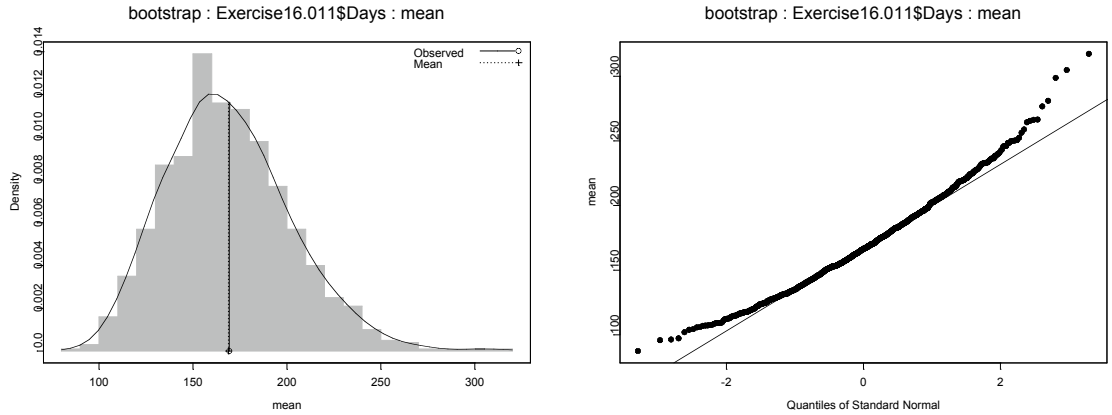
16.10. (a) The histogram is below; it does show strong skewness. **(b)** There is right skewness, but much less than for the data distribution. The right tail is longer and the left tail shorter than for a Normal distribution.

```
hist(Table01.08$days)
boot10 = bootstrap(Table01.08$days, mean, seed=0)
plot(boot10)
qqnorm(boot10)
qqnorm(update(boot10, B=10^4)) # higher accuracy
```



16.11. (a) It is roughly Normal, but with right skewness. It is less Normal, with greater skewness than for Exercise 14.6. **(b)** The two SEs are 12.8 and 33. The second one is larger for two reasons: a smaller sample size and a larger standard deviation in the data (153 vs. 109).

```
boot11 = bootstrap(Exercise16.011$Days, mean, seed=0)
plot(boot11)
qqnorm(boot11)
boot10                      # SE is 12.82
boot11                      # SE is 33.14
stdev(Table01.08$days)      # 109.2
stdev(Exercise16.011$Days) # 153.4
```

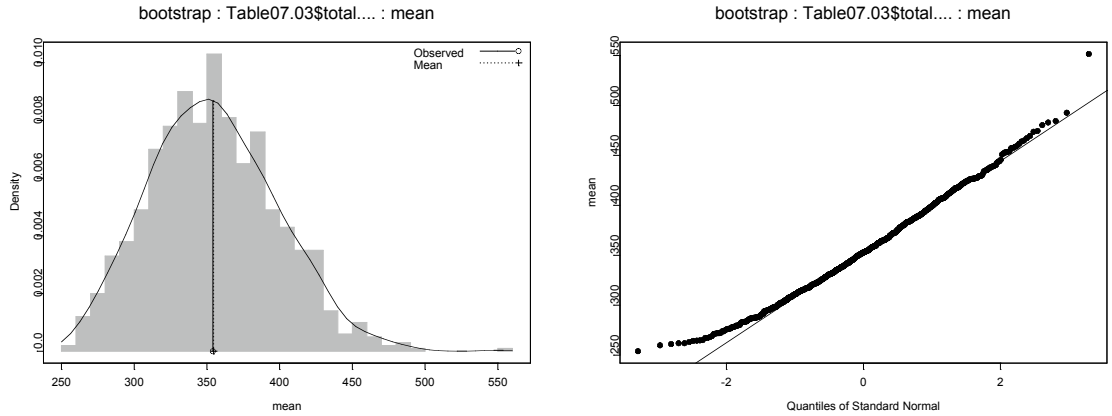


16.12. (a) 0.0053 **(b)** 4.725 (the standard deviation of the resample means). **(c)** (1.6, 25.9)

```
13.762 - mean(Exercise16.001$Time)
# 0.005333333
mean(Exercise16.001$Time) + c(-1,1) * qt(.975, length(Exercise16.001$Time)-
1) * 4.725
# 1.610667 25.902666
```

16.13. (a) There is substantial skewness; a *t* interval is appropriate only for quick-and-dirty calculations. **(b)** (266.0162, 442.1838) **(c)** (266.5823, 441.6177). In this case the bootstrap *t* interval is slightly wider. Students may get slightly narrower bootstrap *t* intervals.

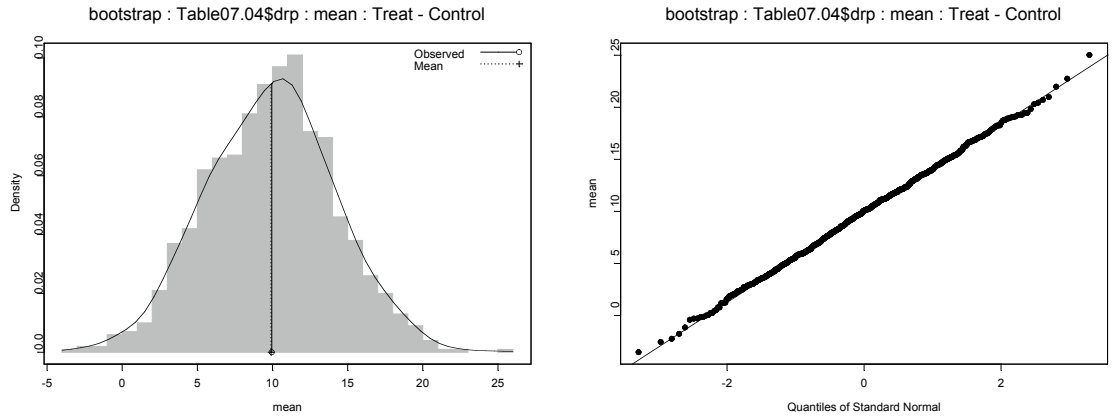
```
plot(boot8)
qqnorm(boot8)
limits.t(boot8)
t.test(Table07.03$total.secs)
```



16.14. (a) The bootstrap SE is 4.2. **(b)** Yes; the bootstrap distribution is very Normal, with no appreciable bias. The interval is (1.36, 18.55). **(c)** The *t* interval reported on page 7-63 is (1.2, 18.7). The bootstrap *t* interval is slightly shorter.

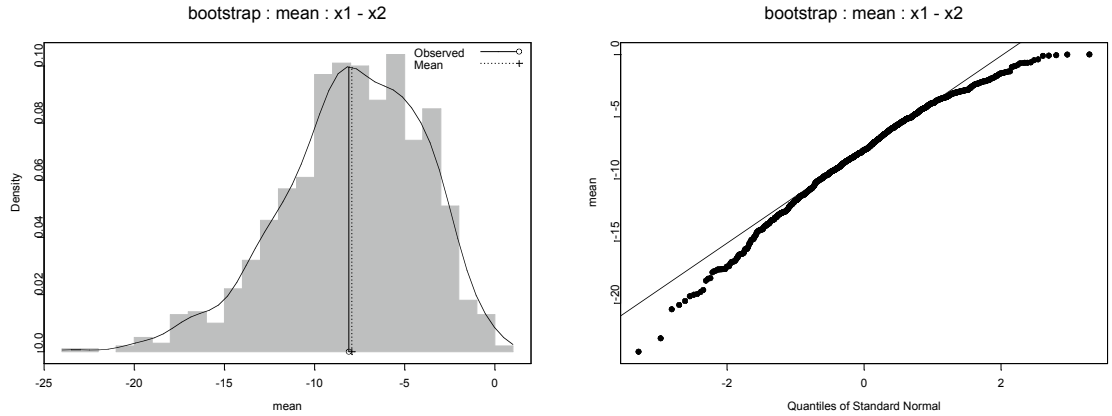
```
Table07.04  # has columns group, g, drp
boot14 = bootstrap2(Table07.04$drp, mean,
                    treatment = Table07.04$group, seed=0)
boot14
plot(boot14)
qqnorm(boot14)
limits.t(boot14)
```

```
t.test(Table07.04$drp, treatment = Table07.04$group, var.equal=F)
```



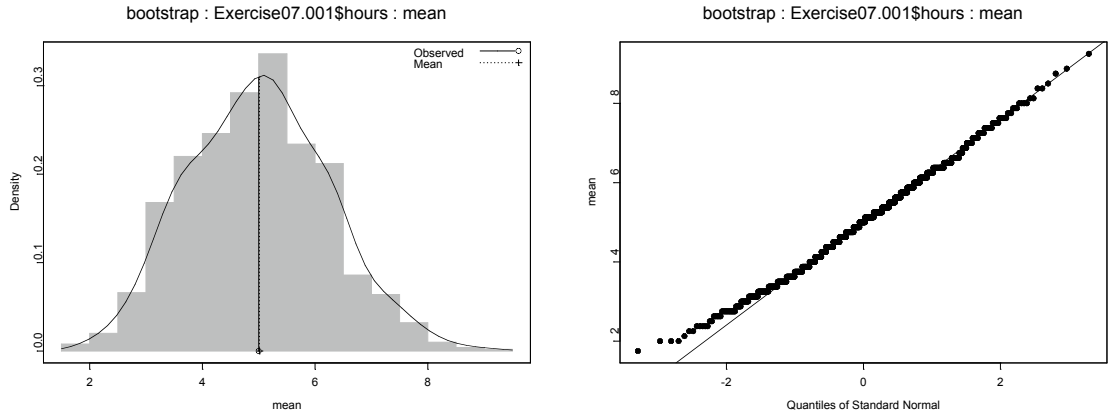
16.15. The formula SE is 4.08; the bootstrap SE is 3.89. The bootstrap SE is slightly smaller.

```
# First, split into two data sets
x1 = Example16.06$Time[ Example16.06$Group == "ILEC" ]
x2 = Example16.06$Time[ Example16.06$Group == "CLEC" ]
sqrt( var(x1) / length(x1) + var(x2) / length(x2) )
boot15 = bootstrap2(x1, data2=x2, mean, seed=0)
# another way, if we didn't have x1 and x2:
# boot15 = bootstrap2(Example16.06$Time, mean, treatment = Group)
boot15
plot(boot15)
qqnorm(boot15)
```



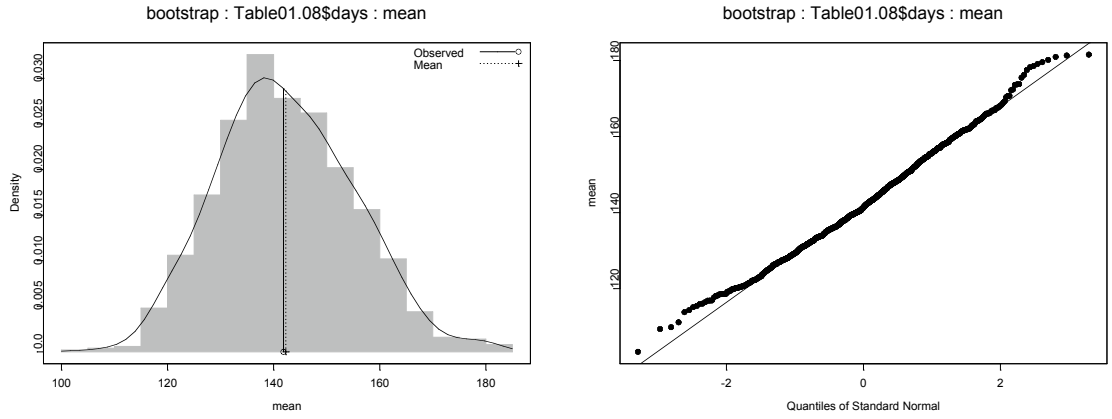
16.16. (a) The bootstrap distribution shows noticeable skewness; if this were a high-accuracy situation, we would hesitate to use *t* intervals. However, with this small data set, the sample skewness and skewness of bootstrap distributions can vary substantially (see Section 16.3), so we do not have a strong reason to avoid a *t* interval. The bias is 0.012 (small compared to SE of 1.24). **(b)** bootstrap *t*: 2.066959 7.933041 **(c)** formula *t*: 1.969167 8.030833. The formula *t* is slightly wider.

```
plot(boot6)
qqnorm(boot6)
boot6
limits.t(boot6)
t.test(Exercise07.001$hours)
```



16.17. (a) The estimated bias is 0.45 (this answer will vary randomly above or below zero, but should be close to zero). This is small compared to the SE of 12.8. This indicates that the bias is small; in this case, only 3% of a standard error. **(b)** The bootstrap *t* interval is (116.3, 167.4). **(c)** The bootstrap SE is 12.82; the formula SE is 12.87. The formula *t* interval is (116.2, 167.5), slightly wider than the bootstrap *t*.

```
boot10  # See Exercise16.010 above to create this
plot(boot10)
qqnorm(boot10)
qqnorm(update(boot10, B=10^4, trace=F))
limits.t(boot10)
boot10
stdev(Table01.08$days) / sqrt(nrow(Table01.08))
t.test(Table01.08$days)
```



General Comment: Bootstrap Standard Errors Are Too Small

Aside from randomness, the bootstrap estimate of SE for a single mean is $\sqrt{\quad}$ rather than $\sqrt{\quad}$, where $\sqrt{\quad\quad}$. This is because the bootstrap plugs in the empirical distribution as an estimate of the population, and the standard deviation of the empirical distribution is . So bootstrap SEs are too small by a factor of about $(1 - 1/(2n))$. Throughout this document you'll find that bootstrap SEs are slightly smaller than formula SEs (or substantially smaller, for example when $n = 8$ in Exercise14.18). There are ways around this, but we don't tell students about them because we want to keep things simple.

One remedy is "bootknife sampling"—to draw a (bootknife) bootstrap sample, first omit one random observation, then draw a sample of size n with replacement from the remaining $n - 1$ observations. In S+Resample, you can select this option from the resampling menus or from the command line by specifying `sampler=samp.bootknife` when calling `bootstrap`.

Another is to draw bootstrap samples of size n-1 rather than n; from the menu or command line, use `sampler=samp.bootstrap(reduceSize=1)`.

16.18. (a) The new bootstrap distribution should be similar to Figure 16.7. The histogram may have superficial differences—heights of individual bars, for example. The Normal quantile plot appears a bit different, in part because the comparison line is different. But the general pattern—Normal except for some right skewness—is the same. **(b)** The results now are:

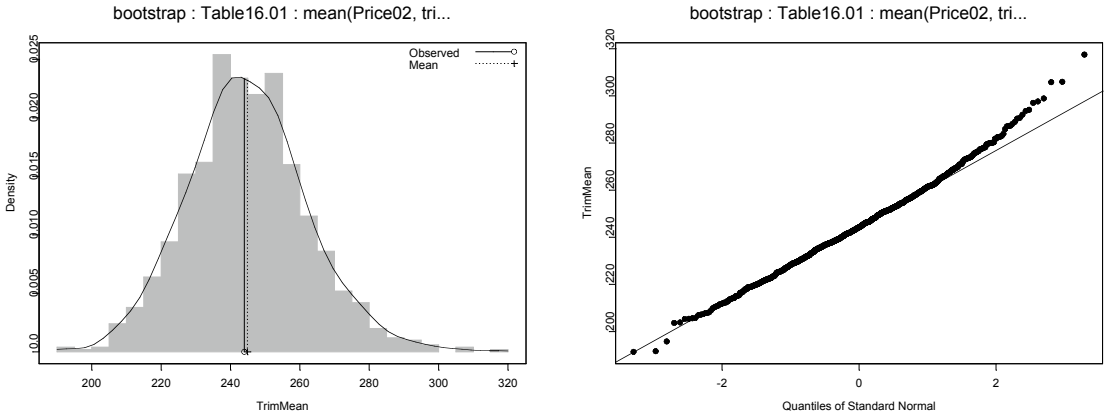
```
           Observed  Mean   Bias    SE
TrimMean      244 244.9 0.8987 17.29
```

In contrast to before:

```
           Observed   Mean    Bias     SE
TrimMean      244  244.7  0.7171  16.83
```

The bias estimates are similar. The SEs differ by more than we like to see. For a few additional runs of 1000 bootstrap samples the SEs are 16.9, 18.7, 17.5, and 18.1. **(c)** The bootstrap t interval is (209.2, 278.8). This is slightly wider, reflecting the larger SE.

```
boot18 = bootstrap(Table16.01, mean(Price02, trim=.25),
  statisticName = "TrimMean", seed=0)
plot(boot18)
qqnorm(boot18)
boot18
limits.t(boot18)
```



16.19. The shape is roughly Normal, but with some odd departures at the lower end, and a shorter tail at the upper end. (These still occur for $B = 10^4$, so they are probably not random artifacts.) The estimated bias is –10.7, which is substantial, and a large fraction of the SE (83) and the observed s (317). That downward bias may be surprising; under simple random sampling we expect that s^2 is an unbiased estimate of σ^2, and s is nearly unbiased for σ. We comment on this below; first we answer the rest of the question. There does appear to be moderate bias, so the bootstrap t interval should only be used in applications where high accuracy is not necessary. The bootstrap t interval is (149, 484). This is a large fraction above and below the estimated standard deviation of 316, indicating that the estimated standard deviation is imprecise.

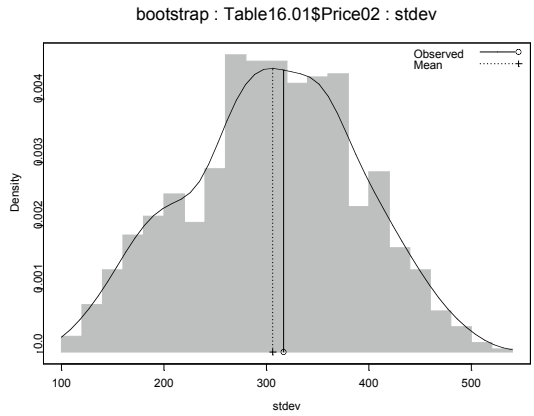
```
boot19 = bootstrap(Table16.01$Price02, stdev, seed=0)
```

```
plot(boot19)
qqnorm(boot19)
boot19
limits.t(boot19)
```

Printouts from the last two commands are:

```
      Observed  Mean   Bias    SE
stdev    316.8 306.1 -10.72 83.2

          2.5%       5%      95%    97.5%
stdev 149.4278 177.2108 456.4493 484.2323
```





Comment About the Bias in the Bootstrap Distribution of *s*

The reason the bootstrap finds bias here is somewhat technical. The bootstrap assumes that the statistic being bootstrapped is a plug-in statistic that works in exactly the same way for the sample as for the population. This is true for the mean, but not for *s*. The usual "unbiased" estimate of sample variance, with a divisor of $(n-1)$, is not a plug-in statistic. We calculate variance and standard deviation differently for a sample than for the equivalent discrete population, using a divisor of $(n-1)$ for one and n for the other.

More formally, the bootstrap is designed for functional statistics, whose value depends solely on the empirical distribution, not on other factors such as the sample size. If you get a different answer by repeating each observation twice before calculating the statistic, the statistic is not functional.

$\sqrt{\quad\quad}$ is a functional statistic—it is the standard deviation of a population where each value x_i has probability $1/n$. s is a multiple of the functional statistic, $\sqrt{\quad\quad}$. The bootstrap implicitly assumes that s is an estimate of $\sqrt{\quad\quad}$, and indicates that s is negatively biased as an estimate of that parameter.

There is a second factor, that while s^2 is unbiased for σ^2, s is not unbiased for σ. The bias is approximately $-\sigma/(4n)$ if the underlying data are Normally distributed, or larger (more negative) if the distribution has longer tails. In other words, using a divisor of $(n-1)$ in place of n still leaves about 1/3 or more of the bias for estimating σ.

16.20. (a) *t* procedure seems reasonable; no outliers, no strong skewness. There is some left skewness in the "*s*" group and right skewness in the "*n*" group. **(b)** Yes, the bootstrap distribution has little bias, and is very Normal. **(c)** bootstrap SE = 4.143, *t* interval (–19.15, –2.52) **(d)** formula *t* interval (–19.09, –2.58), slightly narrower than bootstrap *t*.

```
Exercise07.081 # ns dbh
boxplot(split(Exercise07.081$dbh, Exercise07.081$ns))
```

```
       stem(Exercise07.081$dbh[Exercise07.081$ns == "n"])
       stem(Exercise07.081$dbh[Exercise07.081$ns == "s"])
       qqnorm(Exercise07.081$dbh[Exercise07.081$ns == "n"])
       qqnorm(Exercise07.081$dbh[Exercise07.081$ns == "s"])
       boot20 = bootstrap2(Exercise07.081, mean(dbh), treatment = ns, seed=0)
       plot(boot20)
       qqnorm(boot20)
       boot20
       limits.t(boot20)
       t.test(Exercise07.081$dbh, treatment = Exercise07.081$ns)

> stem(Exercise07.081$dbh[Exercise07.081$ns == "n"])

N = 30   Median = 17.05
Quartiles = 10.2, 39.1

Decimal point is 1 place to the right of the colon

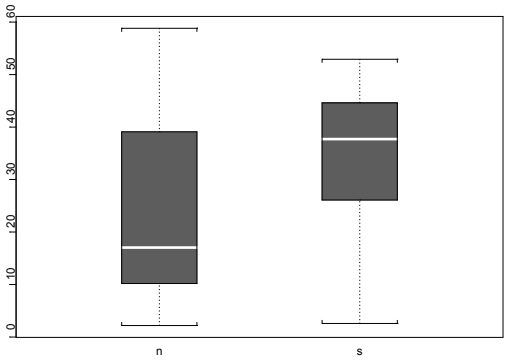
   0 : 2234556
   1 : 014444559
   2 : 5688
   3 : 179
   4 : 0347
   5 : 459

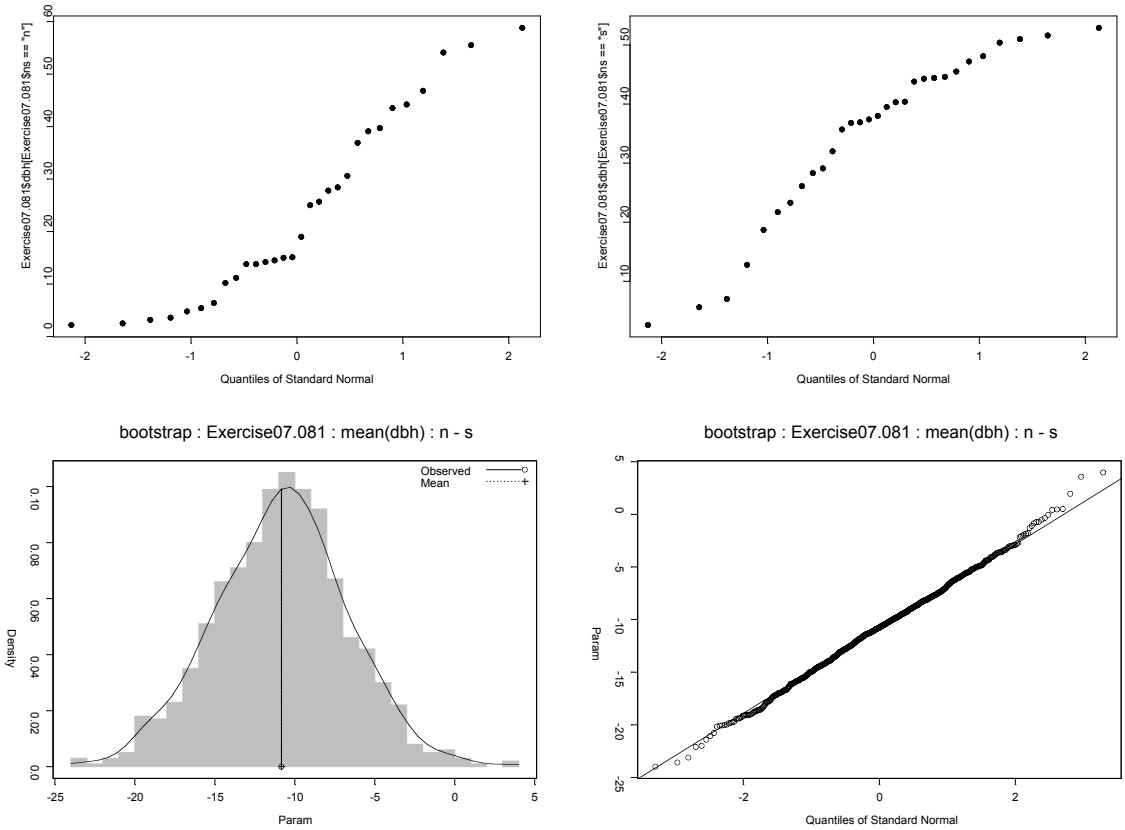

> stem(Exercise07.081$dbh[Exercise07.081$ns == "s"])

N = 30   Median = 37.7
Quartiles = 26.1, 44.6

Decimal point is 1 place to the right of the colon

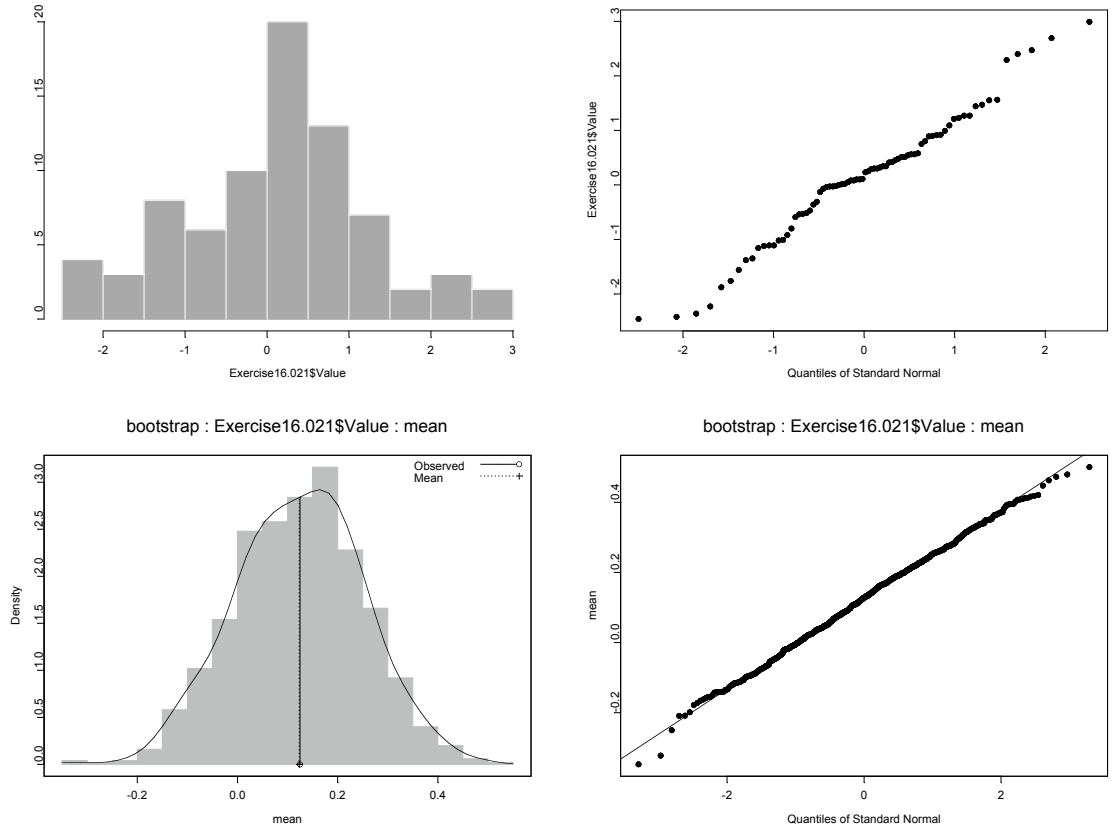
   0 : 367
   1 : 39
   2 : 23689
   3 : 2677789
   4 : 004445578
   5 : 0123
```





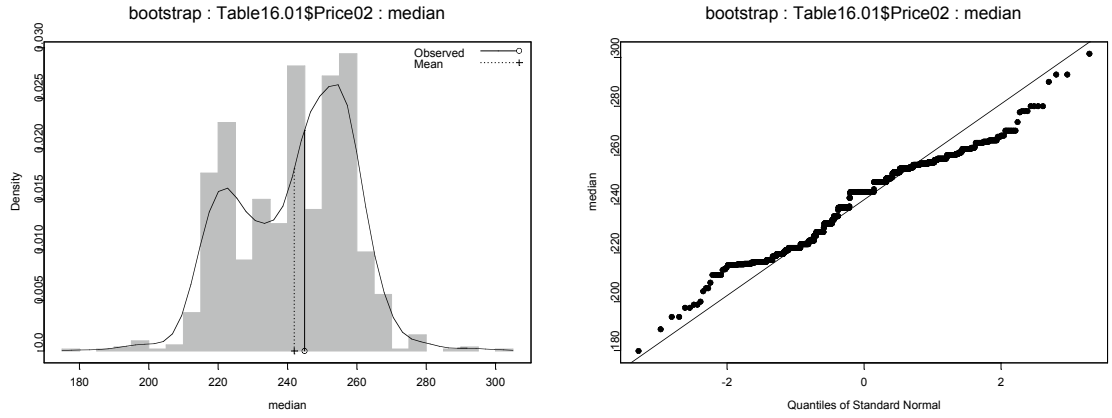
16.21. (a) The data appear roughly Normal, though with the typical random gaps and bunches that usually occur with random data. This is more apparent from the Normal quantile plot than from the histogram. It appears from both the histogram and Normal quantile plot that the mean is slightly larger than zero, but the difference is not large enough to rule out an $N(0,1)$ distribution. **(b)** Yes; the bootstrap distribution is extremely close to Normal with no appreciable bias. **(c)** The bootstrap t interval is (–0.13, 0.38); the formula t interval is (–0.14, 0.39).

```
hist(Exercise16.021$Value)
qqnorm(Exercise16.021$Value)
boot21 = bootstrap(Exercise16.021$Value, mean, seed=0)
plot(boot21)
qqnorm(boot21)
boot21
stdev(Exercise16.021$Value) / sqrt(length(Exercise16.021$Value))
limits.t(boot21)
t.test(Exercise16.021$Value)
```



16.22. The bootstrap distribution is quite non-Normal. It appears to have multiple modes. The Normal quantile plot shows that it is also discrete.

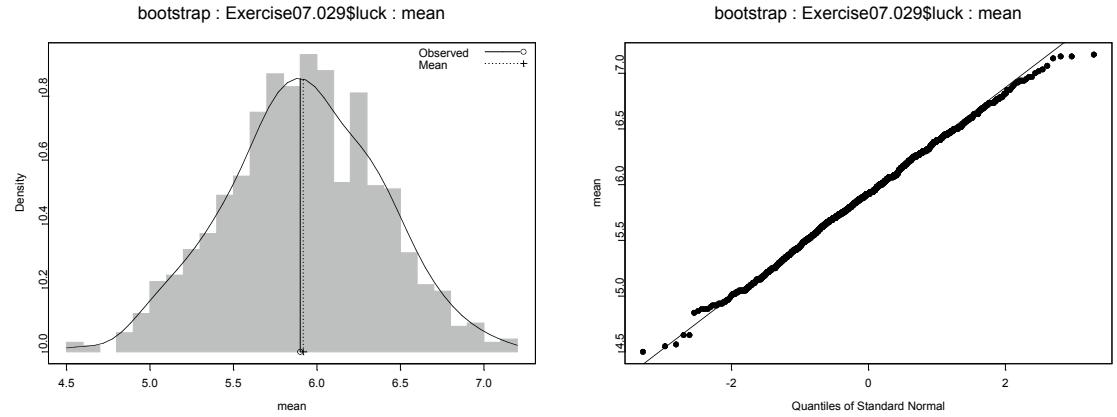
```
boot22 = bootstrap(Table16.01$Price02, median, seed=0)
plot(boot22)
qqnorm(boot22)
boot22
```



16.23. Yes, it is justified—the bootstrap distribution is very Normal, with no bias. The bootstrap *t* interval is (5.00, 6.82); the usual *t* interval is (4.93, 6.87). The formula *t* interval is slightly longer.

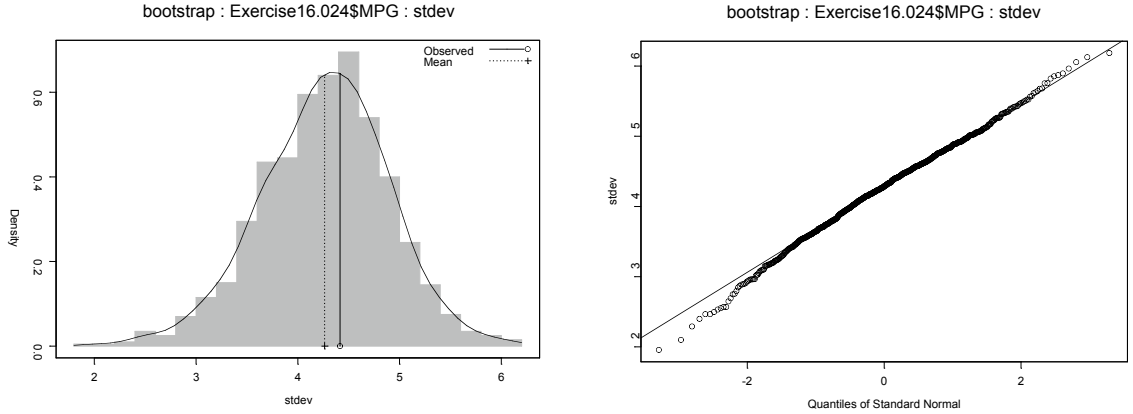
```
Exercise07.029  # variable: luck
boot23 = bootstrap(Exercise07.029$luck, mean, seed=0)
```

```
plot(boot23)
qqnorm(boot23)
limits.t(boot23)
t.test(Exercise07.029$luck)
```



16.24. (a) $s = 4.41$. **(b)** bootstrap SE = 0.627. **(c)** The SE is quite large relative to s, suggesting that s is not a reliable estimate. **(d)** No, there is substantial bias and skewness, so a t interval would not be appropriate.

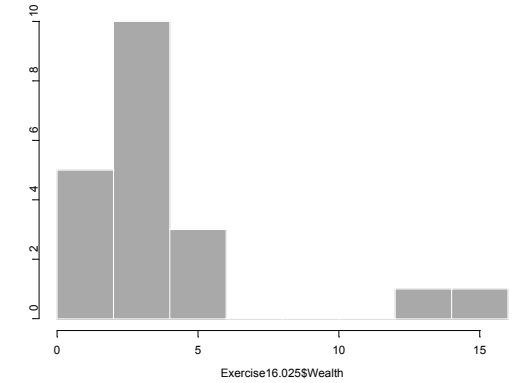
```
stdev(Exercise16.024$MPG)
boot24 = bootstrap(Exercise16.024$MPG, stdev, seed=0)
boot24
plot(boot24)
qqnorm(boot24)
```

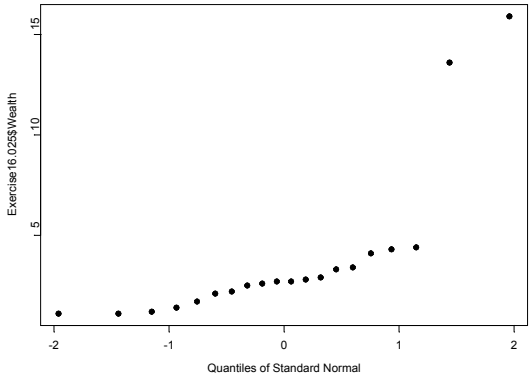


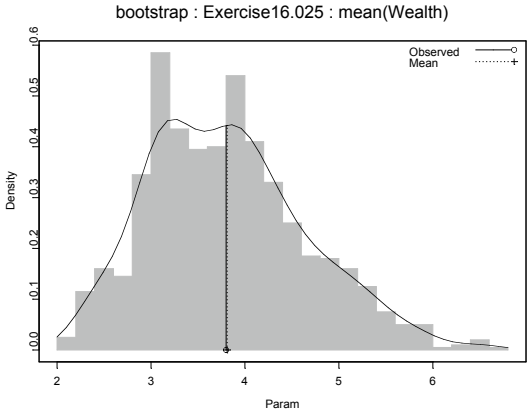
16.25. The distribution is right skewed, so we will use a trimmed mean rather than an ordinary mean. The trimmed mean is 2.72, SE is 0.36, and estimated bias is 0.02—note that the bias is a substantial fraction of the SE. The bootstrap distribution is quite skewed to the right, so a bootstrap t interval would be questionable if accuracy is important. The 95% bootstrap t confidence interval is (1.97, 3.47) for the average wealth of the middle 50% of billionaires; the percentile interval is (2.09, 3.45). For comparison, the ordinary mean is 3.8 with a SE of 0.87.

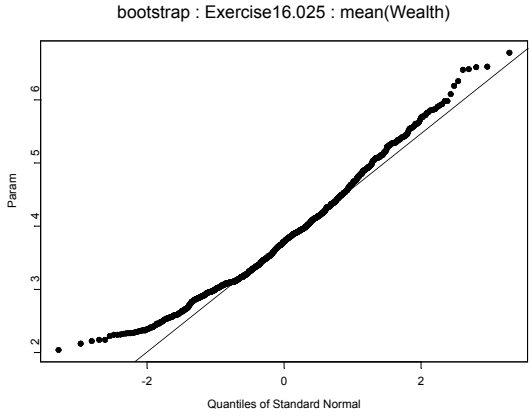
```
hist(Exercise16.025$Wealth)
qqnorm(Exercise16.025$Wealth)
boot25a = bootstrap(Exercise16.025, mean(Wealth), seed=0)
boot25b = bootstrap(Exercise16.025, mean(Wealth, trim=.25), seed=0)
plot(boot25a)
qqnorm(boot25a)
```

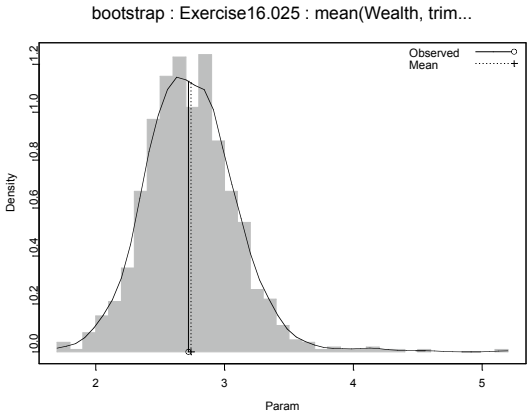
```
boot25a
limits.t(boot25a)
limits.percentile(boot25a)
plot(boot25b)
qqnorm(boot25b)
boot25b
limits.t(boot25b)
limits.percentile(boot25b)
mean(Exercise16.025$Wealth)
stdev(Exercise16.025$Wealth)/sqrt(length(Exercise16.025$Wealth))
```

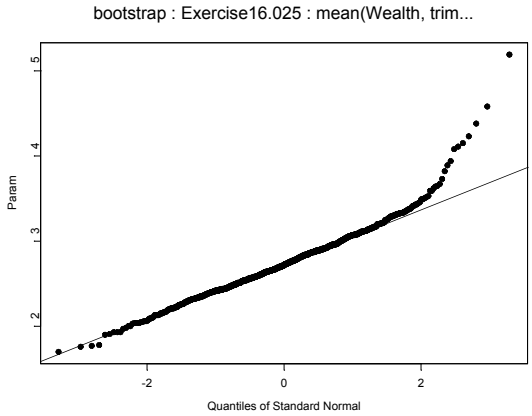








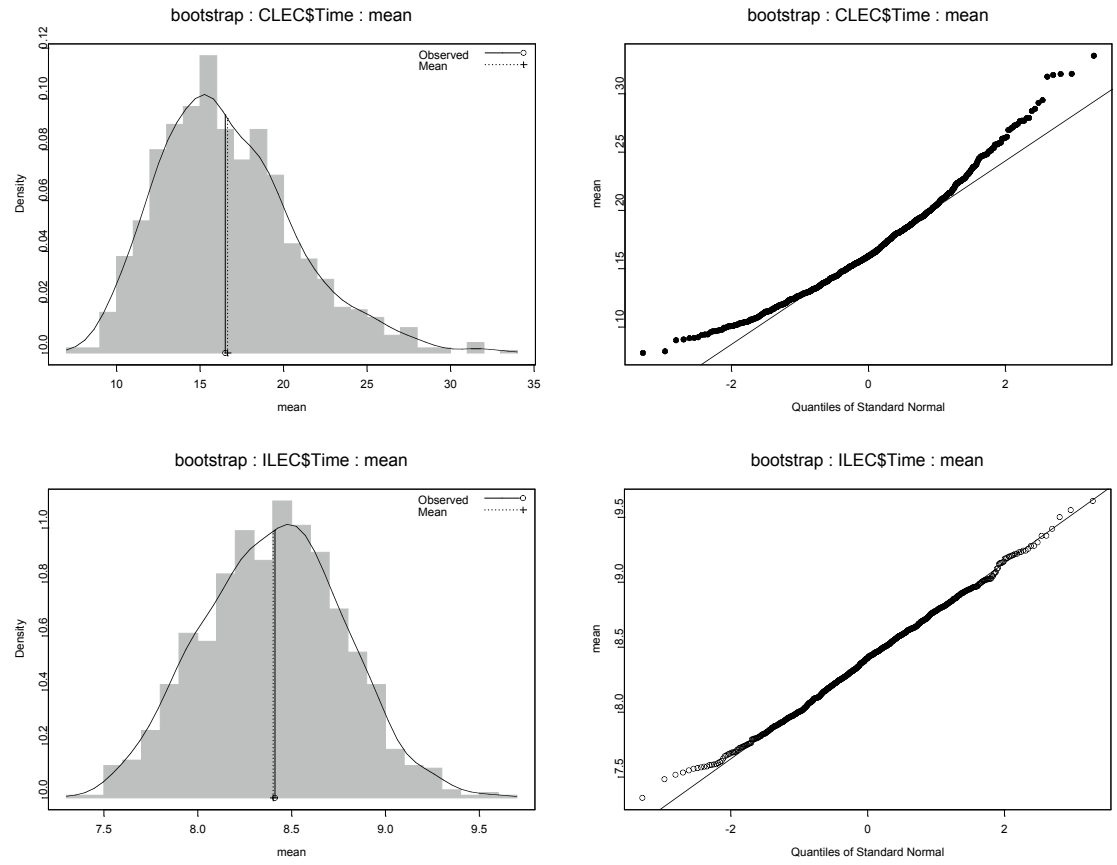




16.26. (a) The bootstrap distribution for CLEC mean is strongly right–skewed, with a mean of 16.5 and SE of 4.1. For comparison, the bootstrap distribution for ILEC mean is barely skewed, with a mean of 8.4 and SE of 0.37. **(b)** Because the CLEC bootstrap means vary so little, what really

matters when you take (ILEC mean) – (CLEC mean) is the CLEC mean. Because of the minus sign, the right skewness of the CLEC means makes the difference have left skewness.

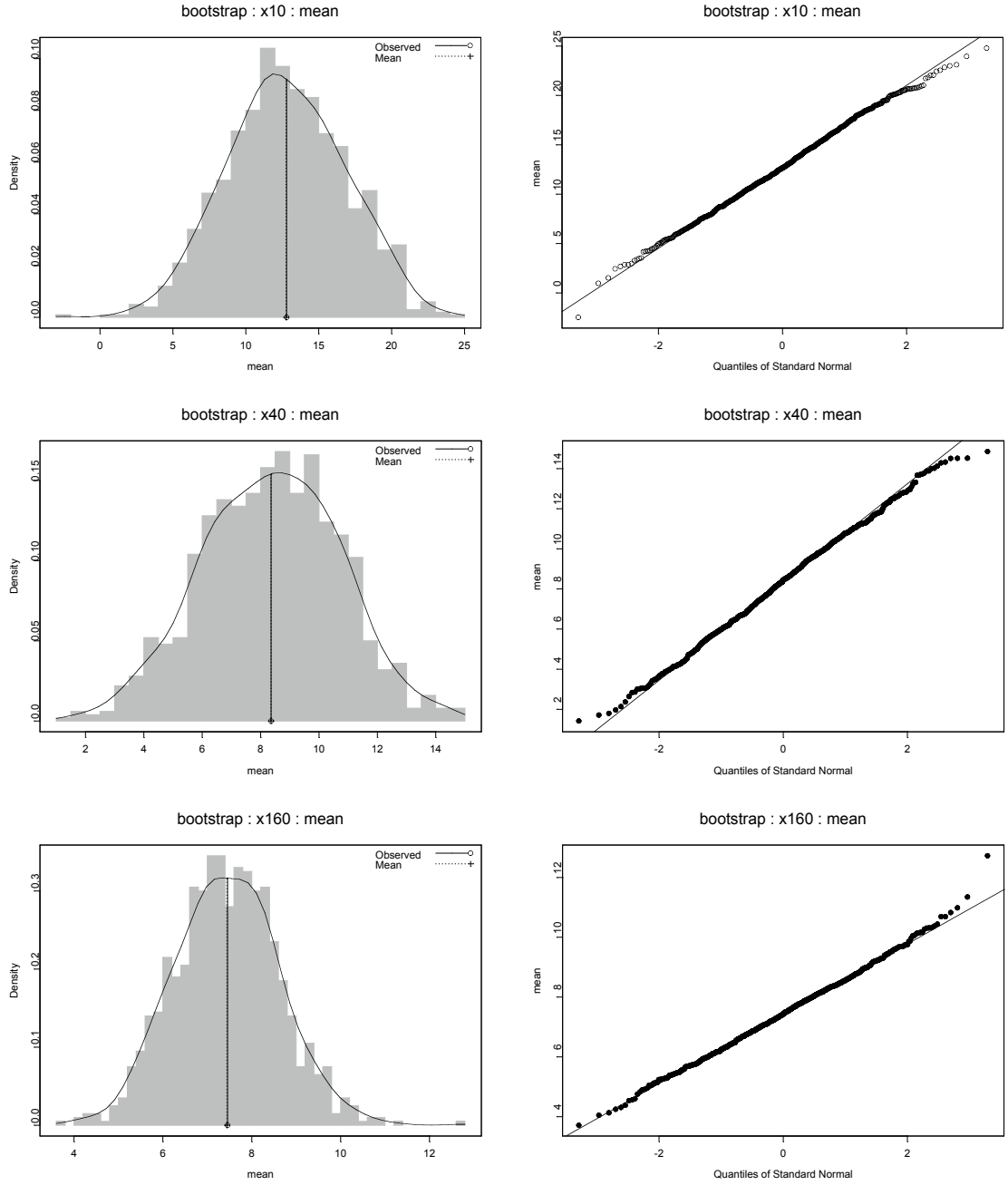
```
boot26 = bootstrap(CLEC$Time, mean, seed=0)
plot(boot26)
qqnorm(boot26)
boot26
boot26b = bootstrap(ILEC$Time, mean, seed=0)
boot26b
plot(boot26b)
qqnorm(boot26b)
```



16.27. (a) Normal with mean 8.4 and standard deviation $14.7/\sqrt{\ }$. **(b)** and **(c)** See figures.
(d) Student answers may vary a lot, depending on their samples. There may be some skewness (right or left) for smaller samples. There should be almost no bias, and SE should decrease by approximately a factor of 2 each time the sample size increases by a factor of 4, except that the SE for small samples may be higher or lower than the geometric progression would indicate.

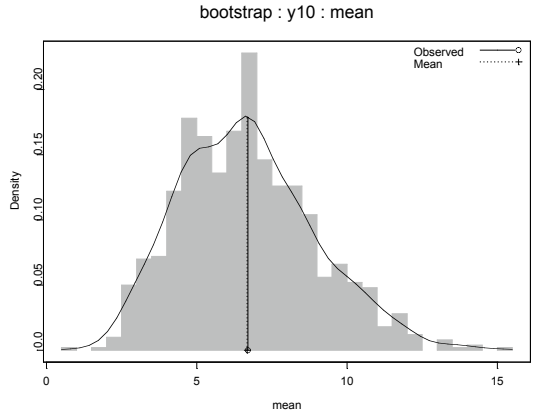
```
set.seed(1)  # So can reproduce results (reserve seed 0 for bootstrap
calls)
x10 = rnorm(10, mean = 8.4, sd = 14.7)
boot27a = bootstrap(x10, mean, seed=0)
plot(boot27a)
qqnorm(boot27a)
boot27a
set.seed(1)
x40 = rnorm(40, mean = 8.4, sd = 14.7)
boot27b = bootstrap(x40, mean, seed=0)
plot(boot27b)
qqnorm(boot27b)
```

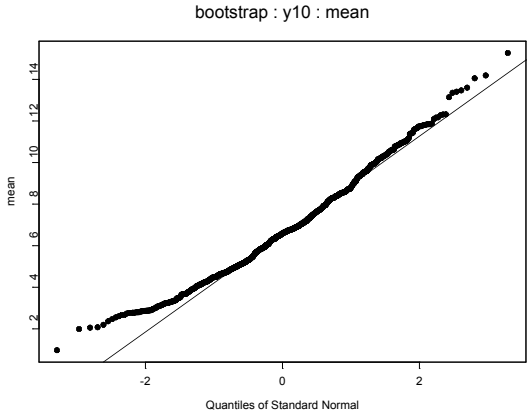
```
boot27b
set.seed(1)
x160 = rnorm(160, mean = 8.4, sd = 14.7)
boot27c = bootstrap(x160, mean, seed=0)
plot(boot27c)
qqnorm(boot27c)
boot27c
```

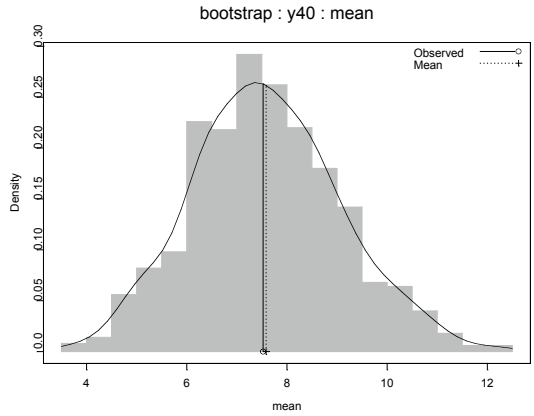


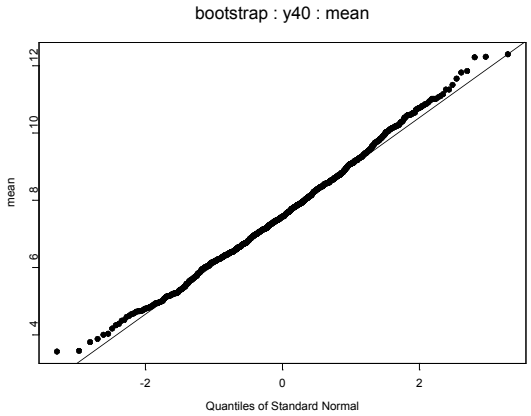
16.28. (a) The mean is 8.4, and the standard deviation is $14.7/\sqrt{\quad}$. **(b)** and **(c)** See figures. **(d)** Student answers may vary, depending on their samples. There should be substantial right skewness for smaller samples, closer to Normal for large samples. There should be almost no bias, and SE should decrease by approximately a factor of 2 each time the sample size increases by a factor of 4, except that the SE for small samples may be higher or lower than indicated by the geometric progression.

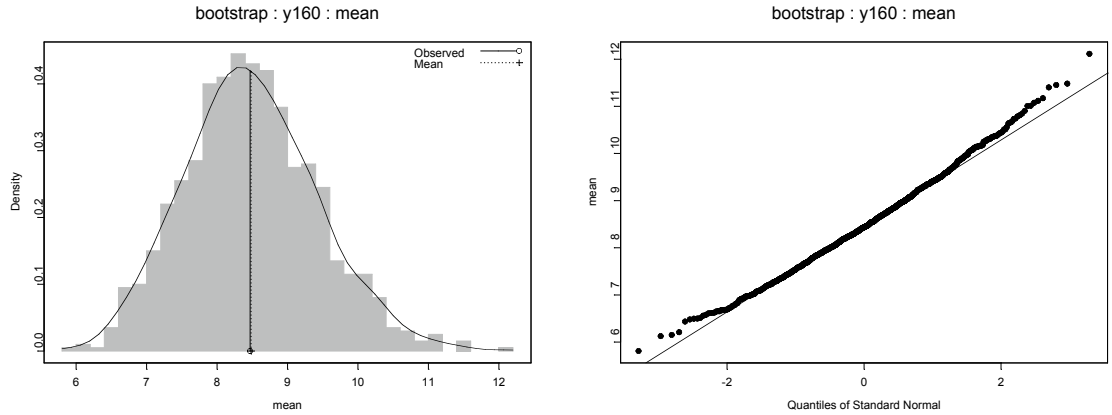
```
set.seed(1)
y10 = sample(ILEC$Time, size=10)
boot28a = bootstrap(y10, mean, seed=0)
plot(boot28a)
qqnorm(boot28a)
boot28a
set.seed(1)
y40 = sample(ILEC$Time, size=40)
boot28b = bootstrap(y40, mean, seed=0)
plot(boot28b)
qqnorm(boot28b)
boot28b
set.seed(1)
y160 = sample(ILEC$Time, size=160)
boot28c = bootstrap(y160, mean, seed=0)
plot(boot28c)
qqnorm(boot28c)
boot28c
```









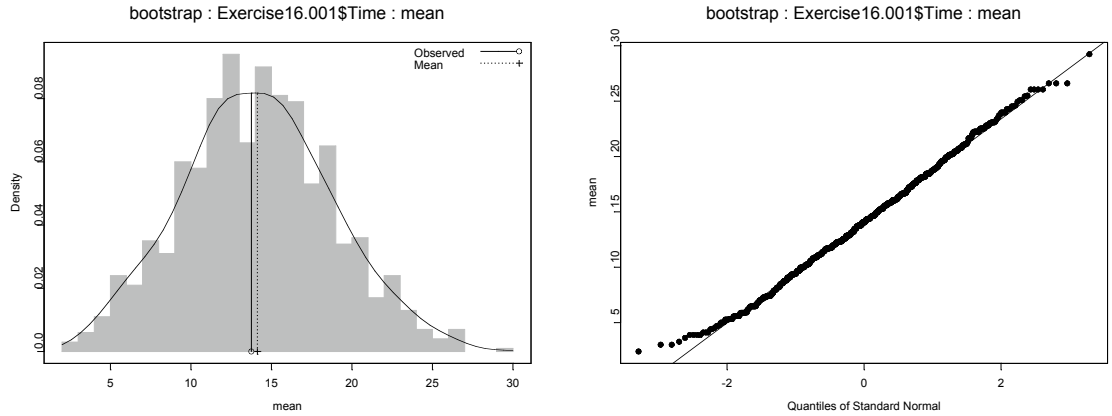


16.29. Student answers should vary depending on their samples. They should notice that the bootstrap distributions are approximately Normal for larger sample sizes. For small samples, the sample could be skewed one way or the other in Exercise 16.27, and most should be right-skewed for Exercise 16.28. Some of that skewness should come through into the bootstrap distribution.

16.30. 5% and 95%. 1% and 99%.

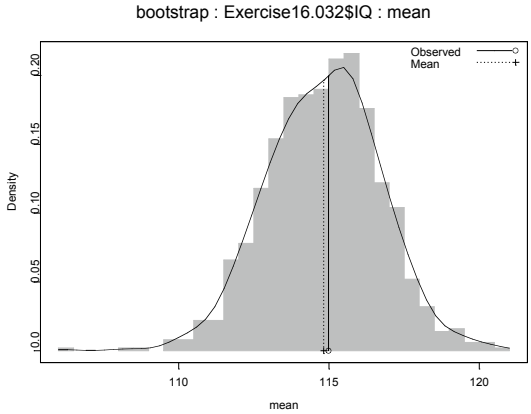
16.31. (a) It looks close to Normal (though that doesn't mean much with this small sample). The bias is small. **(b)** The bootstrap *t* interval is (1.8, 25.7). **(c)** The bootstrap percentile interval is (5.3, 23.7). The *t* interval is much wider because the 97.5% *t* quantile is 1.3 times larger than the corresponding Normal quantile, when the *t* has only 5 degrees of freedom.

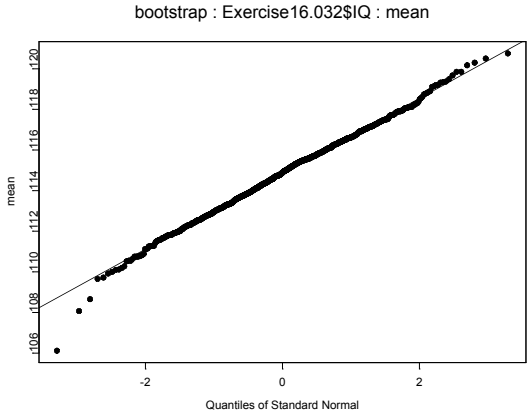
```
boot31 = bootstrap(Exercise16.001$Time, mean, seed=0)
plot(boot31)
qqnorm(boot31)
boot31
limits.t(boot31)
limits.percentile(boot31)
qt(.975,5) / qnorm(.975) # ratio is 1.31
```

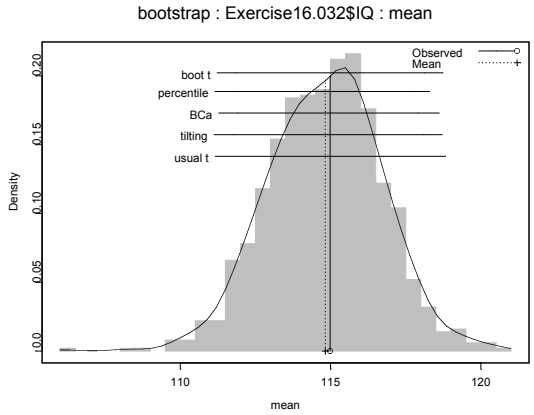


16.32. (a) The formula SE is 1.91, and formula *t* confidence interval is (111.2, 118.8). **(b)** The bootstrap distribution appears Normal, except for a long tail on the far left. The bootstrap SE is 1.88, and bootstrap *t* confidence interval is (111.2, 118.7). **(c)** The intervals agree fairly well, except that the percentile interval is shorter on the right side. Here the formula interval would be fine.

```
stdev(Exercise16.032$IQ) / length(Exercise16.032$IQ)
t.test(Exercise16.032$IQ)
boot32 = bootstrap(Exercise16.032$IQ, mean, seed=0)
plot(boot32)
qqnorm(boot32)
boot32
limits.t(boot32)
limits.percentile(boot32)
limits.bca(boot32)
limits.tilt(boot32)
# Create a function to automate the process of showing intervals
graphically
plotWithIntervals = function(boot, usualT=NULL, ..., Par = NULL){
  # Do bootstrap plot, and add all four bootstrap intervals,
  # and optionally the usual t interval.
  # ... may be used for the subset.statistic argument
  # Par = list of arguments to plot.resamp
  if(!is.null(list(...)$subset.statistic))
    Par <- c(Par, list(subset.statistic = list(...)$subset.statistic))
  if(!is.null(Par))
    do.call("plot", c(list(boot), Par))
  else
    plot(boot, ...)
  ymax = par("usr")[4]
  f = function(x, y, label=""){
    # Draw a horizontal line for one interval
    lines(x=x, y=rep(y, length=length(x)))
    text(min(x), y, paste(label," "), adj=1)
  }
  f(limits.t(boot, ...), ymax*.9, "boot t")
  f(limits.percentile(boot, ...), ymax*.84, "percentile")
  f(limits.bca(boot, ...), ymax*.77, "BCa")
  f(limits.tilt(boot, ...), ymax*.70, "tilting")
  if(!is.null(usualT))
    f(usualT, ymax*.63, "usual t")
}
plotWithIntervals(boot32, t.test(Exercise16.032$IQ)$conf.int)
```





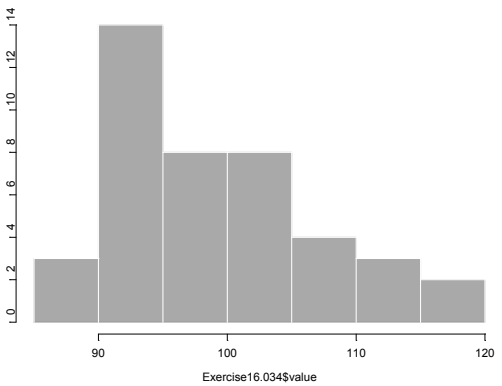


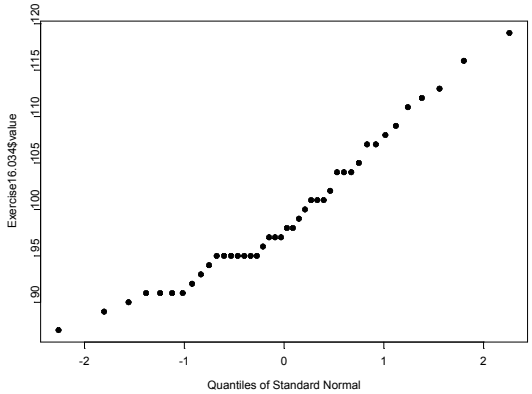
16.33. (a) The bootstrap percentile interval is (–0.13, 0.37). For comparison, the bootstrap *t* interval is (–0.13, 0.38). These are very close, suggesting that the *t* intervals are acceptable.
(b) Yes.

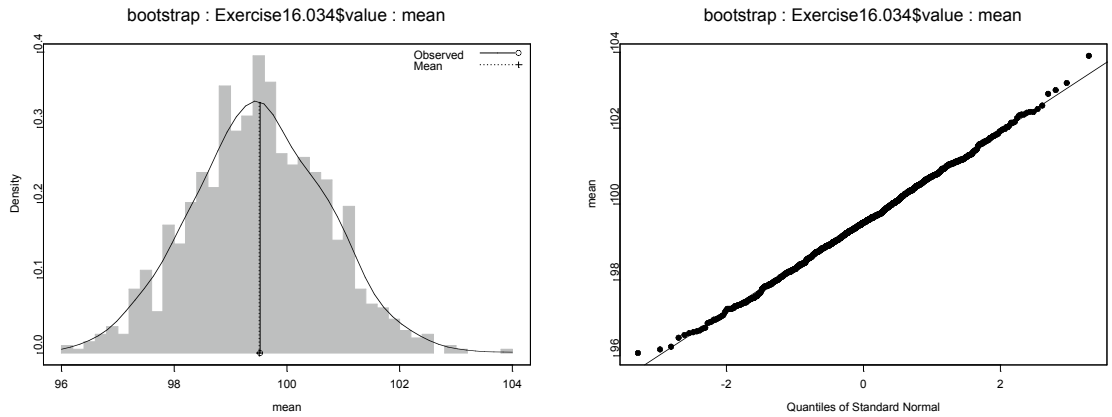
```
#boot21 = bootstrap(Exercise16.021$value, mean, seed=0)
limits.percentile(boot21)
```

16.34. (a) There is a small amount of right skewness, probably not enough to matter except for situations where high accuracy is needed. **(b)** The usual *t* interval is (97.1, 101.9). **(c)** The bootstrap distribution is very close to Normal with no appreciable bias; a *t* interval should be accurate. The bootstrap *t* interval is (97.1, 101.9). **(d)** The bootstrap percentile interval is (97.2, 101.9). This is similar to the *t* intervals. We conclude that the usual *t* interval is accurate.

```
Exercise16.034 <- data.frame(value=
c(108, 107, 113, 104,  94, 100, 107,  98, 112,  97,  98,  95, 95,  97,
   99,  95,  97,  90, 109, 102,  89, 101,  93,  95, 105,  91, 96, 104,
   95,  87,  91, 101, 119, 116,  91,  95,  95, 104, 111, 101, 92,  91))
hist(Exercise16.034$value)
qqnorm(Exercise16.034$value)
t.test(Exercise16.034$value)
boot34 = bootstrap(Exercise16.034$value, mean, seed=0)
plot(boot34)
qqnorm(boot34)
limits.percentile(boot34)
limits.t(boot34)
```







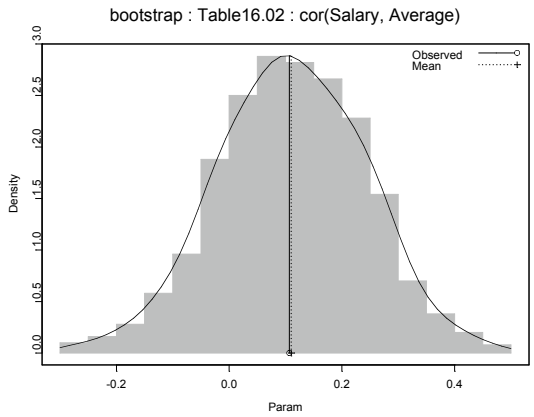
16.35. The bootstrap *t* interval is (–0.144, 0.358), and the percentile interval is (–0.128, 0.356). The differences are relatively small relative to the width of the intervals, so they do not indicate appreciable skewness.

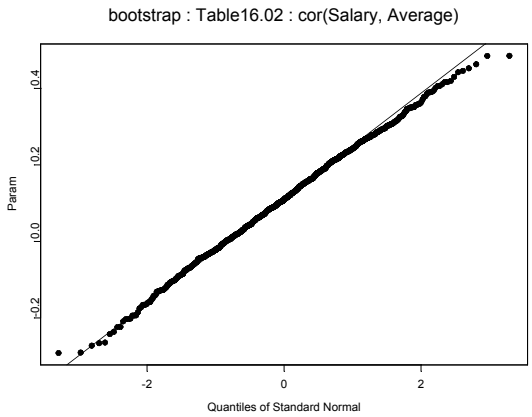
16.36. (a) The BCa interval is (97.3, 102.0), and the tilting interval is (97.3, 102.1). **(b)** The usual *t* interval is (97.1, 101.9). The tilting interval is slightly farther to the right on both ends than the *t* interval, suggesting that there is some skewness that should be taken into account in high-accuracy situations. **(c)** For a quick check, we use the percentile interval. For a more accurate check, we use a BCa or tilting interval.

```
limits.bca(boot34)
limits.tilt(boot34)
t.test(Exercise16.034$value)
```

16.37. The BCa interval is (–0.165, 0.356), and tilting interval is (–0.136, 0.362). For comparison, the bootstrap *t* interval is (–0.158, 0.371), and percentile interval is (–0.159, 0.360). The BCa and percentile are in close agreement, and the *t* interval is not far off. This suggests that the simpler intervals are adequate. The tilting interval is substantially shorter on the left side. This is probably due to a weakness in the tilting intervals. There are different variations of tilting intervals. At the time these exercises were written the version implemented in S-PLUS gives intervals that are too short for small samples for some nonlinear statistics.

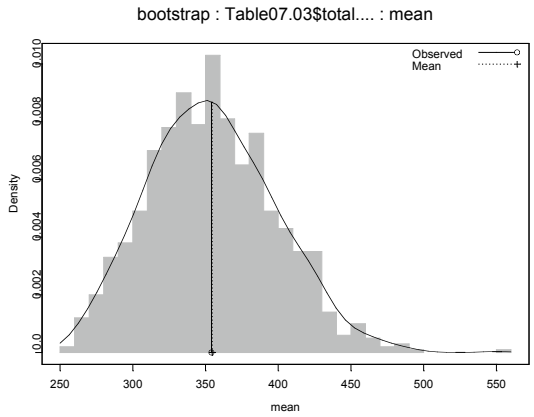
```
boot37 = bootstrap(Table16.02, cor(Salary,Average), seed=0)
plot(boot37)
qqnorm(boot37)
limits.t(boot37)
limits.percentile(boot37)
limits.bca(boot37)
limits.tilt(boot37)
```

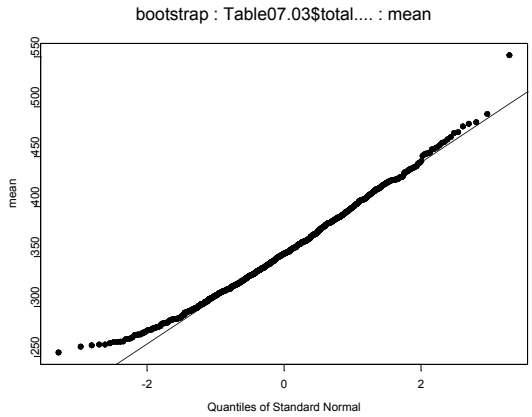


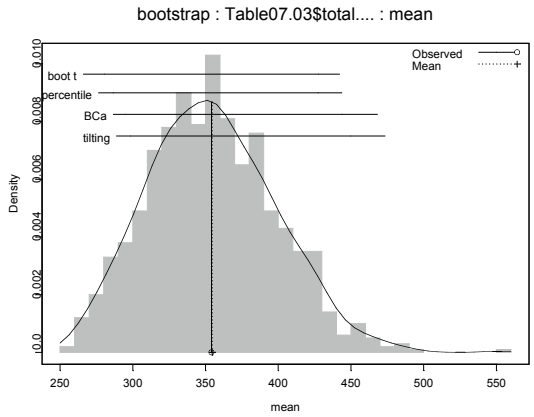


16.38. Yes, the skewness is a cause for concern. The percentile interval is to the right of the *t* interval on the lower end. The BCa and tilting intervals are substantially to the right of the *t* interval and percentile interval on both ends. Hence, the *t* and percentile intervals are inaccurate here.

```
plot(boot8)
qqnorm(boot8)
limits.t(boot8)
limits.percentile(boot8)
limits.bca(boot8)
limits.tilt(boot8)
plotWithIntervals(boot8)
```

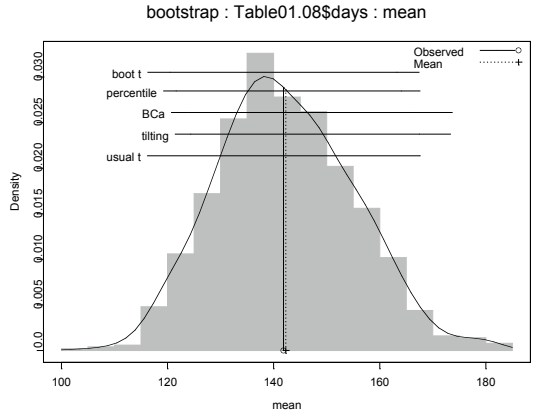






16.39. We previously found that the bootstrap *t* interval is (116.3, 167.4), and the formula *t* interval is (116.2, 167.5). The BCa interval is (120.7, 173.7), and the tilting interval is (121.4, 173.4). The bootstrap *t* and percentile intervals do not agree; the percentile interval is farther to the right, especially on the left side. The BCa and tilting intervals reach even farther to the right, on both sides. This is a case where the percentile interval makes part of the correction for skewness that the BCa and tilting intervals make. (This is what we expect from statistical theory.)

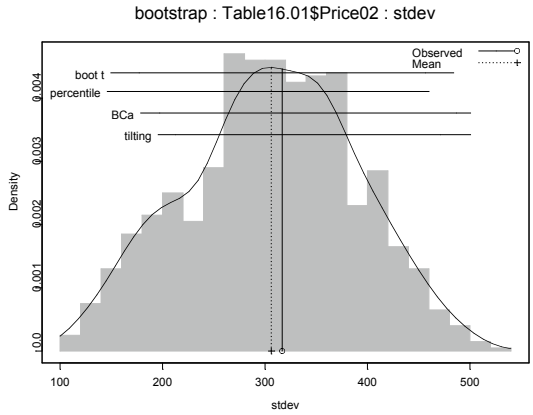
```
#boot10  # See Exercise16.010 above to create this
limits.t(boot10)
limits.percentile(boot10)
limits.bca(boot10)
limits.tilt(boot10)
plotWithIntervals(boot10, t.test(Table01.08$days)$conf.int)
```

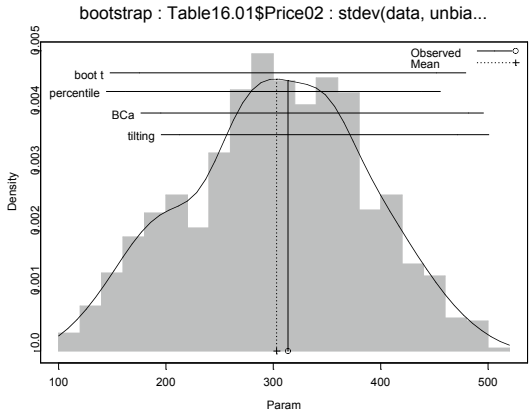


16.40. The percentile interval is shorter on the right side than is the bootstrap *t* interval. The BCa and tilting intervals are farther to the right than either of the simpler intervals. One issue is that the bootstrap is designed for a functional statistic; this is discussed in Exercise 16.19. Hence, the BCa interval is actually an interval for a different parameter, $\sqrt{\quad}$, rather than for σ. This would be true for tilting as well, except that the `limits.tilt` function in S-PLUS checks whether the statistic appears to be functional and makes a correction when it is not. Students should notice the difference between intervals, and recommend using BCa or tilting. In practice, we recommend being conservative and using the left endpoint from the percentile interval and right endpoint from tilting (i.e., on each side, use the most extreme of the three bootstrap endpoints).

```
# boot19 = bootstrap(Table16.01$Price02, stdev, seed=0)
plotWithIntervals(boot19)

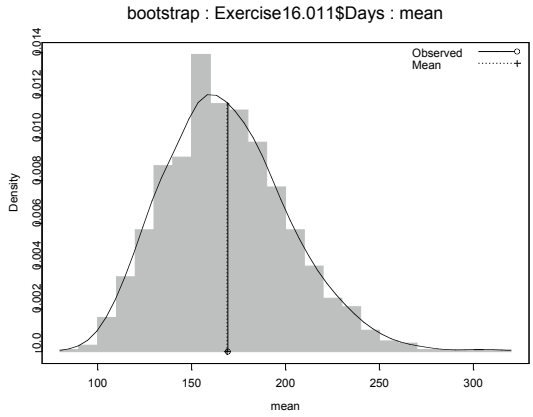
# Now bootstrap the functional versions of standard deviation.
boot19b = bootstrap(Table16.01$Price02, stdev(data, unbiased=F), seed=0)
qqnorm(boot19b)
plotWithIntervals(boot19b)
# The tilting intervals are the same for boot19 and boot19b.
```

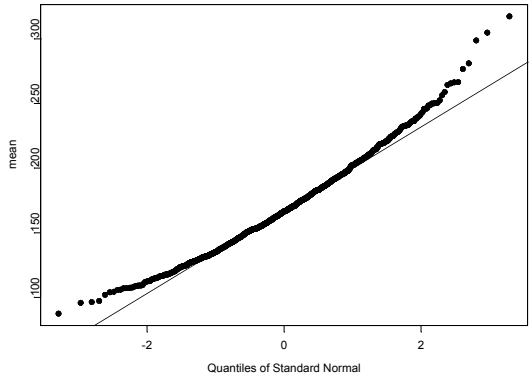


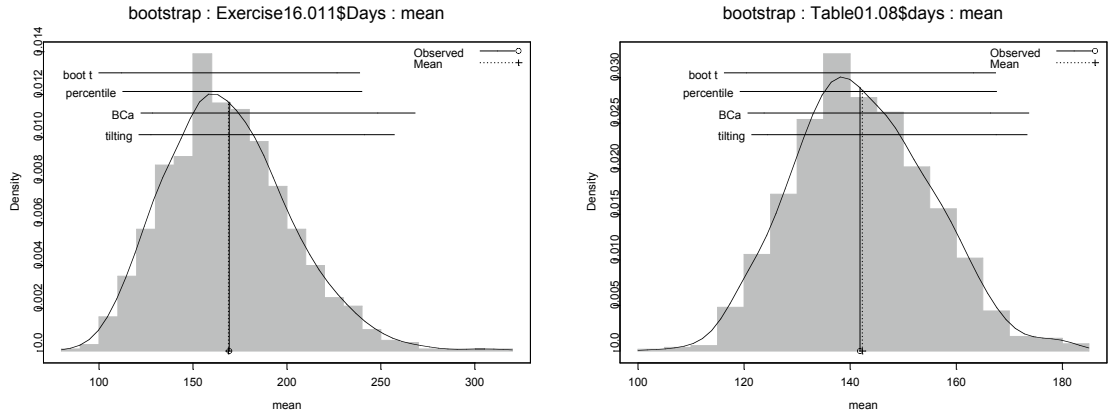


16.41. The bootstrap distribution is very skewed, indicating that the sampling distribution probably is too, so we should not use *t* intervals. The formula *t* interval is (97.5, 241.1); bootstrap *t* is (99.8, 238.8), percentile is (112.6, 239.9), BCa is (122.3, 268.3), and tilting is (121.2, 257.2). These intervals are much wider than for the full dataset. The BCa and tilting intervals are also more asymmetrical than for the full data set. For the full dataset the intervals are somewhat less dissimilar.

```
# boot11 = bootstrap(Exercise16.011$Days, mean, seed=0)
plot(boot11)
qqnorm(boot11)
t.test(Exercise16.011$Days)
limits.t(boot11)
limits.percentile(boot11)
limits.bca(boot11)
limits.tilt(boot11)
plotWithIntervals(boot11)
plotWithIntervals(boot10) # full dataset, for comparison
```

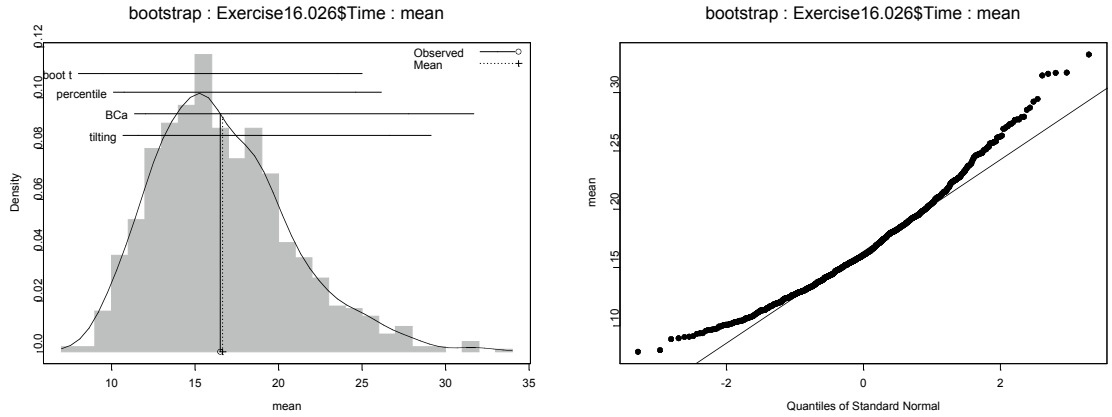






16.42. (a) The bootstrap distribution shows strong right skewness. The formula *t*, bootstrap *t*, and (to a lesser extent) the percentile intervals would not be accurate. The BCa and tilting intervals adjust for skewness, so they should be accurate. **(b)** The bootstrap *t* interval is (8.0, 25.0), percentile interval is (10.1, 26.2), BCa interval is (11.4, 31.7), and tilting interval is (10.7, 29.1). The BCa and tilting intervals are much more asymmetrical than the other intervals because they take into account the skewness in the data; the bootstrap *t* ignores the skewness, and the percentile interval only catches part of the skewness. In practical terms, a *t* interval would underestimate the true value. A *t* interval would not stretch far enough to the right, and would have a much higher than 2.5% probability of missing the population mean because the interval is too low. This is true to a lesser extent for the percentile interval.

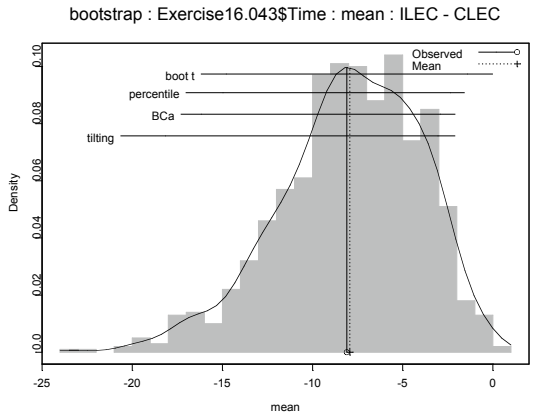
```
boot42 = bootstrap(Exercise16.026$Time, mean, seed=0)
plot(boot42)
qqnorm(boot42)
limits.t(boot42)
limits.percentile(boot42)
limits.bca(boot42)
limits.tilt(boot42)
plotWithIntervals(boot42)
```



16.43. The observed difference is –8.1 (mean of ILEC – mean of CLEC). The BCa interval for the population difference is (–17.3, –2.1). A *t* interval does not reach far enough to the left and reaches too far to the right. In practical terms, the effect of using a *t* interval would be that the interval would be too high too often, effectively overestimating where the true difference lies.

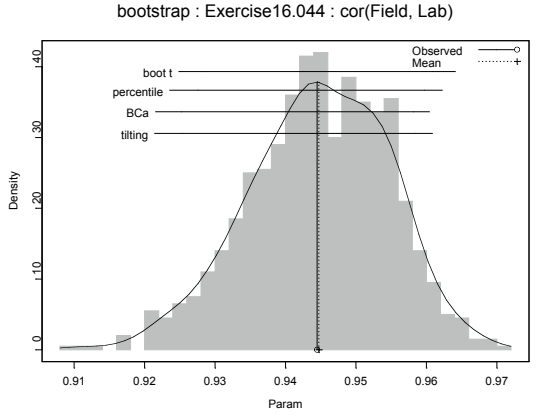
```
boot43 = bootstrap2(Exercise16.043$Time, mean,
                    treatment = Exercise16.043$Group, seed=0)
limits.bca(boot43)
```

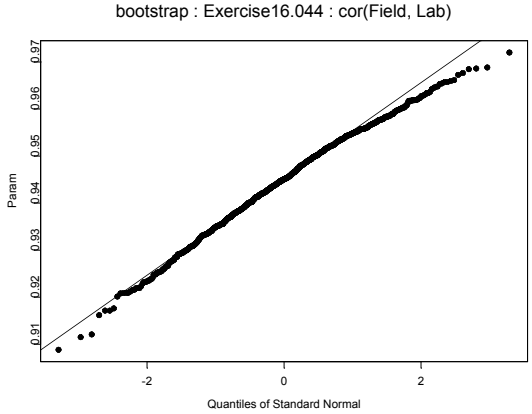
```
plotWithIntervals(boot43)
```



16.44. (a) The bootstrap distribution is approximately Normal, with some left skewness and no appreciable bias. The simpler intervals would be appropriate, except in situations where high accuracy is needed. **(b)** The bootstrap *t* interval is (0.925, 0.964), percentile interval is (0.924, 0.962), BCa interval is (0.922, 0.960), and tilting interval is (0.921, 0.961). The BCa and tilting intervals reach farther to the left than the other intervals, rather more than expected from the amount of skewness in the bootstrap distribution. It appears that the simpler intervals are not particularly accurate; those intervals are biased, and they would be too high too often. A BCa or tilting interval should be used if accuracy is needed.

```
boot44 = bootstrap(Exercise16.044, cor(Field, Lab), seed=0)
plot(boot44)
qqnorm(boot44)
limits.t(boot44)
limits.percentile(boot44)
limits.bca(boot44)
limits.tilt(boot44)
plotWithIntervals(boot44)
```

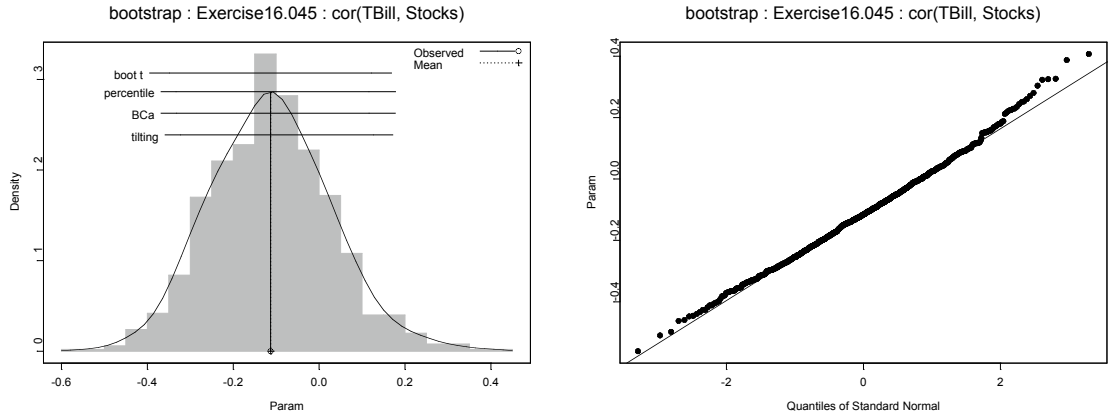




16.45. (a) The bootstrap distribution appears approximately Normal, with a small amount of right skewness, and no appreciable bias. Hence, the assumptions underlying the *t* interval are approximately met. **(b)** The BCa interval is (–0.37, 0.18), and the tilting interval is (–0.36, 0.17). These agree well, with the tilting interval slightly narrower. These intervals do not provide significant evidence that the population correlation is not zero; in fact, they indicate that zero is well within the range of likely values of the population correlation.

```
boot45 = bootstrap(Exercise16.045, cor(TBill, Stocks), seed=0)
plot(boot45)
```

```
qqnorm(boot45)
limits.t(boot45)
limits.percentile(boot45)
limits.bca(boot45)
limits.tilt(boot45)
plotWithIntervals(boot45)
```



16.46. We should resample whole observations. If the data are stored in a spreadsheet with observations in rows and the *x* and *y* variables in two columns, then we should pick a random sample of rows with replacement. When a row is picked, we put the whole row into a bootstrap data set. By doing so we maintain the relationship between *x* and *y*.

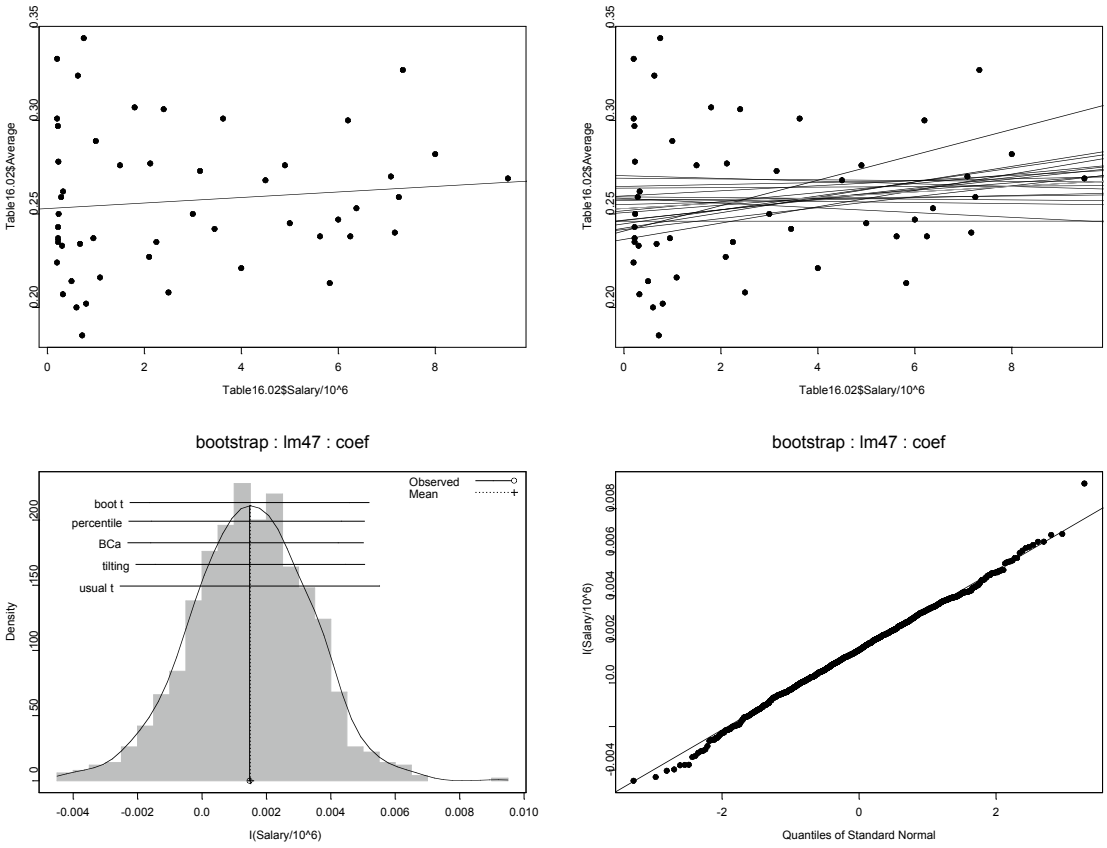
Regression problems follow. Here is command-line code, but it would be more natural to do this using menus, under: Statistics:Resample:Linear Regression.

16.47. (a) The line is (predicted BA) = 0.253 + 0.00148(salary). Each $1 million in salary increases the predicted batting average by 0.00148. **(b)** The bootstrap distribution suggests that any of the intervals would be reasonably accurate. The 95% intervals are bootstrap *t* (–0.0022, 0.0052), percentile (–0.0023, 0.0050), BCa (–0.0023, 0.0050), and tilting (–0.0021, 0.0050). **(c)** This agrees; all intervals include zero, which corresponds to no (linear) relationship between batting average and salary.

As a side note (the exercise does not ask about this): the formula *t* interval is wider than the others. This is the opposite of the situation from Exercise 16.48. Here it appears that the residual variance is smaller for large values of the predictor variable, a violation of the usual regression assumptions, causing the formula SE to be an overestimate.

```
Table16.02 # variables Salary, Average
plot(Table16.02$Salary/10^6, Table16.02$Average)
lm47 = lm(Average ~ I(Salary/10^6), data=Table16.02)
lm47
abline(lm47)
bootstrap(lm47, abline, B=20, seed=0)
abline(lm47, col=5)
boot47 = bootstrap(lm47, coef, seed=0)
plot(boot47, subset.statistic = 2)
qqnorm(boot47, subset.statistic = 2)
limits.t(boot47)
limits.percentile(boot47)
limits.bca(boot47)
limits.tilt(boot47)
summary(lm47)  # This prints the std. error for slope, 0.0020
plotWithIntervals(boot47, coef(lm47)[2] +
```

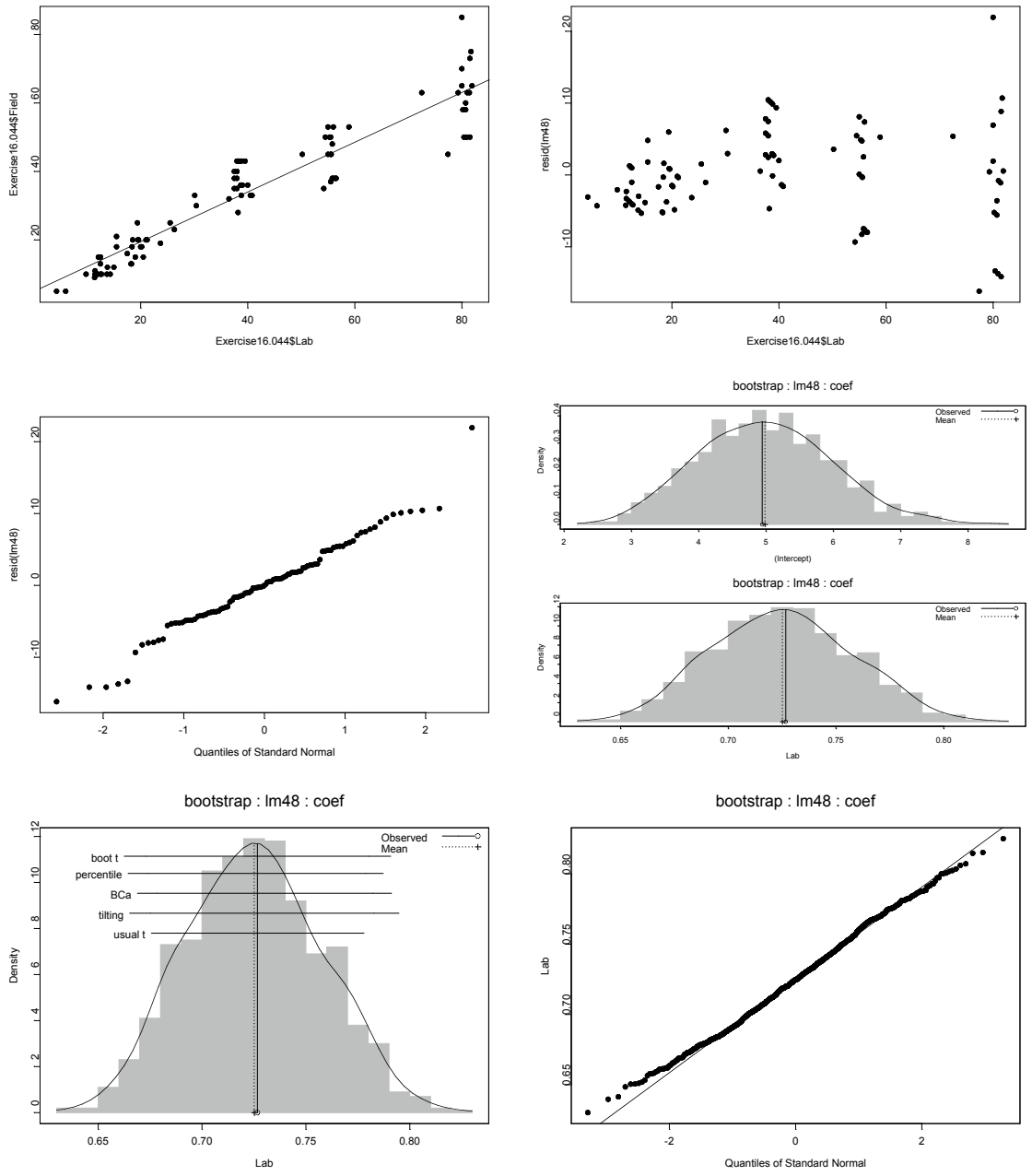
```
qt(c(.025,.975), nrow(Table16.02)-2) * 0.0020,
subset.statistic = 2)
```



16.48. (a) There is heteroskedasticity—the residual variance is much larger when the predicted values are larger (or equivalently, when the explanatory variable is larger). There is one outlier in the residuals. **(b)** The bootstrap distribution looks quite Normal. The bootstrap estimate of bias for b_1 is –0.0016, about 0.05 SEs. This amount of bias would cause a nominal 95% interval to have miss rates of 2.8% and 2.2%, not far from the desired 2.5%. Bootstrap *t* and percentile intervals do seem appropriate. **(c)** Short answer: the bootstrap *t* and percentile should be fine, while the formula *t* is too short. Details: The formula SE for b_1 is 0.0255; the bootstrap SE is 0.0323. This difference is probably due to the heteroskedasticity—the assumptions underlying the formula *t* interval are violated. The formula *t* interval for b_1 is (0.676, 0.777), the bootstrap *t* interval is (0.662, 0.791), the percentile interval is (0.664, 0.787), BCa interval is (0.669, 0.791), and tilting interval is (0.665, 0.795). The formula *t* is much shorter than the other intervals. The left endpoint of the BCa differs from the other bootstrap intervals; this may be due to Monte Carlo variability with only 1000 resamples (the BCa really needs more).

```
plot(Exercise16.044$Lab, Exercise16.044$Field)
lm48 = lm(Field ~ Lab, data=Exercise16.044)
abline(lm48)
plot(lm48) # This gives a variety of plots
plot(Exercise16.044$Lab, resid(lm48))
qqnorm(resid(lm48))
boot48 = bootstrap(lm48, coef, seed=0)
plot(boot48)
qqnorm(boot48)
plot(boot48, subset.statistic = 2)
qqnorm(boot48, subset.statistic = 2)
boot48
```

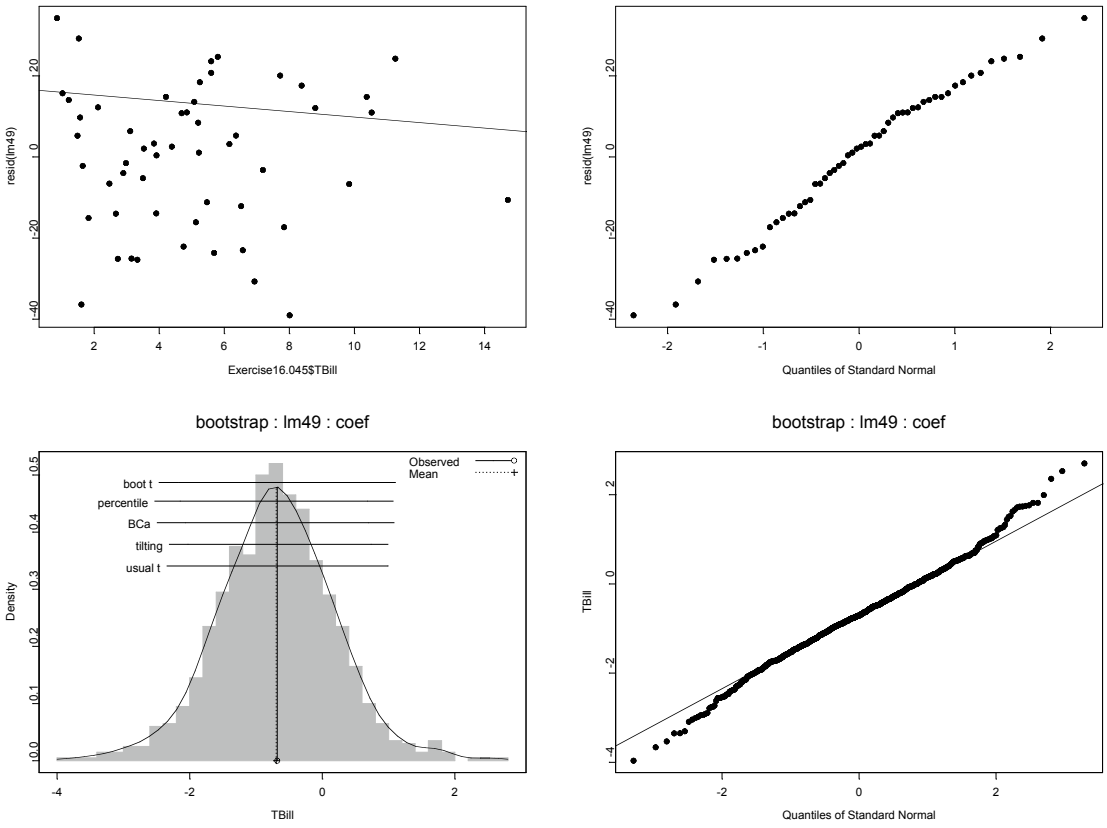
```
lm48
summary(lm48)  # This prints the std. error for slope
coef(lm48)[2] +
     qt(c(.025,.975), nrow(Exercise16.044)-2) * 0.0255
limits.t(boot48, subset.statistic = 2)
limits.percentile(boot48, subset.statistic = 2)
limits.bca(boot48, subset.statistic = 2)
plotWithIntervals(boot48, coef(lm48)[2] +
                qt(c(.025,.975), nrow(Exercise16.032)-2) * 0.0255,
                subset.statistic = 2)
```



16.49. (a) The residuals appear quite Normal. **(b)** The bootstrap distribution has no appreciable bias but does have somewhat longer tails than a Normal distribution. But there is no skewness, and most of the long tails happen outside the range of ±2 standard deviations, so the *t* interval should be accurate. **(c)** The regression slope is –0.68, and SE is 0.83. The formula *t* interval is (–2.34, 0.98).

The bootstrap *t* interval is (–2.46, 1.10), the percentile interval is (–2.53, 1.07), and the BCa interval is (–2.49, 1.08). The intervals are similar, except the formula *t* interval is shorter than the others. All of the intervals include 0, well within the interval. They do not provide evidence that the population slope is nonzero.

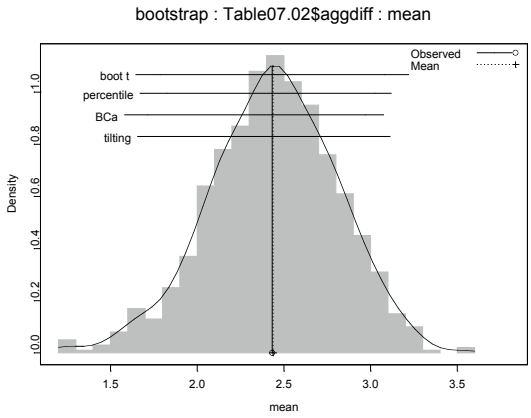
```
lm49 = lm(Stocks ~ TBill, data=Exercise16.045)
plot(lm49) # This gives a variety of plots
plot(Exercise16.045$TBill, resid(lm49))
abline(lm49)
qqnorm(resid(lm49))
boot49 = bootstrap(lm49, coef, seed=0)
plot(boot49)
qqnorm(boot49)
qqnorm(boot49, subset.statistic = 2)
lm49
summary(lm49)  # This prints the std. error for slope
coef(lm49)[2] +
     qt(c(.025,.975), nrow(Exercise16.045)-2) * 0.8292
limits.t(boot49, subset.statistic = 2)
limits.percentile(boot49, subset.statistic = 2)
limits.bca(boot49, subset.statistic = 2)
plotWithIntervals(boot49, coef(lm49)[2] +
               qt(c(.025,.975), nrow(Exercise16.045)-2) * 0.8292,
               subset.statistic = 2)
```

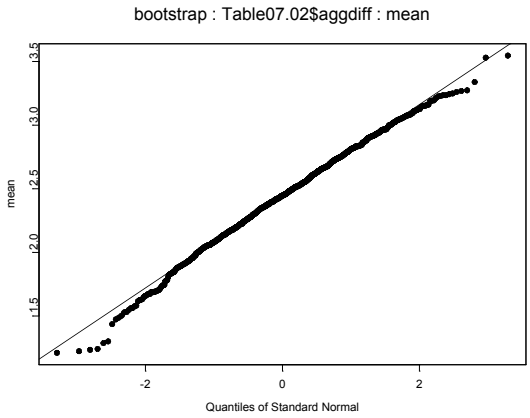


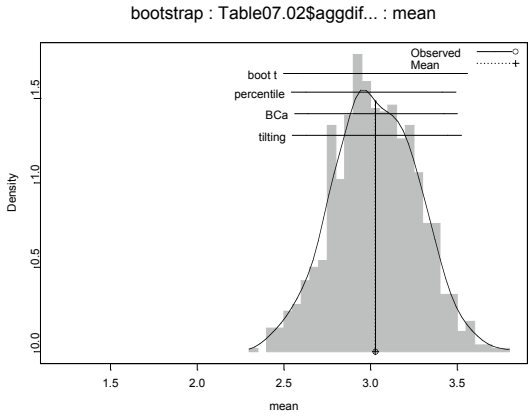
16.50. (a) The bootstrap distribution has slight left skewness for the full data set, and is close to Normal when the outliers are removed. The bias is close to zero in each case. The spread differs substantially—the two SEs are 0.37 and 0.24, with outliers included and excluded. **(b)** The BCa intervals are (1.58, 3.08) and (2.56, 3.50); the interval is higher and much narrower when low

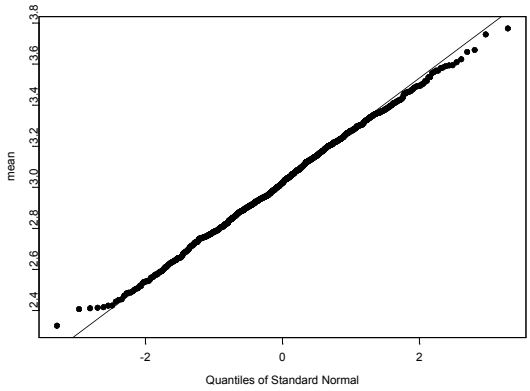
outliers are excluded. The tilting intervals are (1.66, 3.11) and (2.55, 352); again, the intervals are higher and narrower when the lower outliers are excluded.

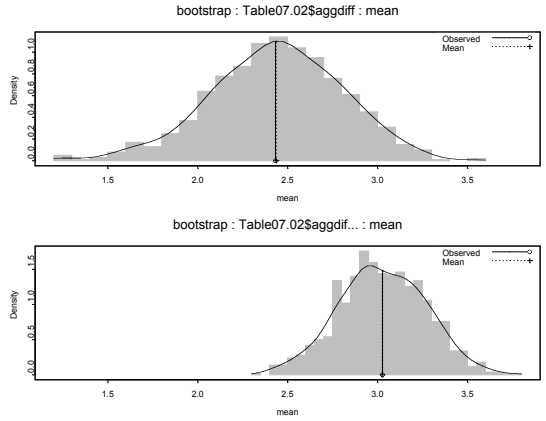
```
hist(Table07.02$aggdiff)
Table07.02$aggdiff
boot50a = bootstrap(Table07.02$aggdiff, mean, seed=0)
boot50b = bootstrap(Table07.02$aggdiff[c(1:11,15)], mean, seed=0)
plot(boot50a)
qqnorm(boot50a)
plot(boot50b)
qqnorm(boot50b)
boot50a
boot50b
limits.bca(boot50a)
limits.bca(boot50b)
limits.tilt(boot50a)
limits.tilt(boot50b)
par(mfrow=c(2,1))  # plot both distributions with common axes
plot(boot50a, xlim=c(1.2, 3.8))
plot(boot50b, xlim=c(1.2, 3.8))
par(mfrow=c(1,1))
plotWithIntervals(boot50a, Par=list(xlim=c(1.2, 3.8)))
plotWithIntervals(boot50b, Par=list(xlim=c(1.2, 3.8)))
```











16.51. No, because in order to pool the data as required for a permutation test, the two populations must be the same when the null hypothesis is true. Here one population probably has a smaller standard deviation.

16.52. The standard error is about 0.014. No, it would not feel comfortable; the real *P*-value could easily be above 0.05, even above 0.067.

```
sqrt(.04 * .96 / 200)  # 0.01385641
.04 + 2 * sqrt(.04 * .96 / 200) # .0677
```

16.53. (a) 20.25. **(b)**, **(c)**, and **(d)** Answers will vary substantially. (e) There is a typo, there are 6! permutations and 15 combinations. In this example there are only 15 possible combinations, which give differences in means equal to (–20.25, –17.25, –16.50, –8.25, –5.25, –3.75, –3.00, 0.00, 5.25, 8.25, 8.25, 9.00, 11.25, 12.00, 20.25), and the exact *P*-value is 1/15.

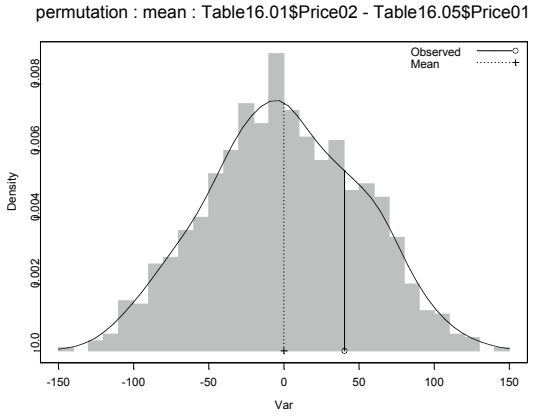
This is normally done by hand, but the following commands would work:

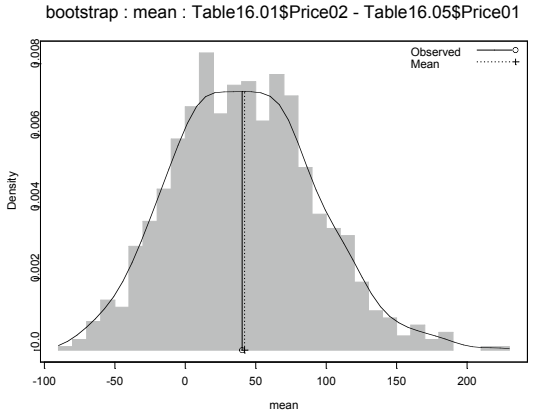
```
mean( c(57, 53)) - mean( c(19, 37, 41, 42))
perm53 = permutationTestMeans(c(57,53), data2=c(19,37,41,42),
                          alternative="greater", seed=0)
perm53
indices = samp.combinations(n=6, B=15, k=2)
values = (indexMeans(c(57,53,19,37,41,42), indices[1:2,]) -
          indexMeans(c(57,53,19,37,41,42), indices[3:6,]))
sort(values)
mean(values >= perm53$observed)
# 0.06666667
# Another approach, that only works if all choose(6,2) values are unique
# and perm53 doesn't miss any:
values = sort(unique(perm53$replicates))
mean(values >= perm53$observed)
```

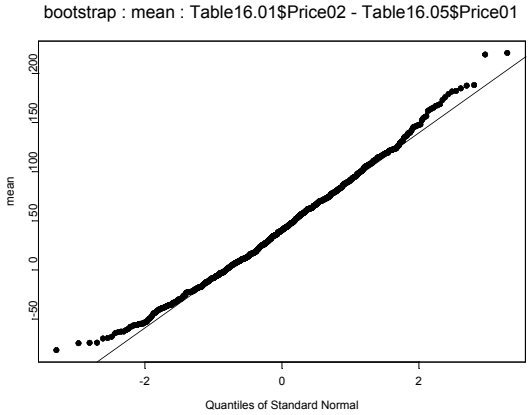
16.54. (a) Let μ_1 denote the mean selling price for all Seattle real estate transactions in 2001, and μ_2 the mean selling price for all Seattle real estate transactions in 2002. We test H_0: $\mu_1 = \mu_2$ vs. H_a: $\mu_1 \neq \mu_2$. **(b)** 0.423 (not pooling variances). **(c)** The *P*-value is 0.462. This is consistent with the *P*-value we computed in (b). We conclude that there is little evidence that the population means μ_1 and μ_2 differ. **(d)** A BCa 95% confidence interval for the change from 2001 to 2002 is (–40.1, 164.8). This interval includes 0 and suggests that the two means are not significantly different at the 0.05 level. This is consistent with the conclusions in (c).

```
t.test(Table16.01$Price02, Table16.05$Price01, var.equal = F)
```

```
perm54 = permutationTestMeans(Table16.01$Price02,
                              data2 = Table16.05$Price01, seed=0)
perm54
plot(perm54)
boot54 = bootstrap2(Table16.01$Price02,
                    data2=Table16.05$Price01, mean, seed=0)
plot(boot54)
qqnorm(boot54)
limits.bca(boot54)
```



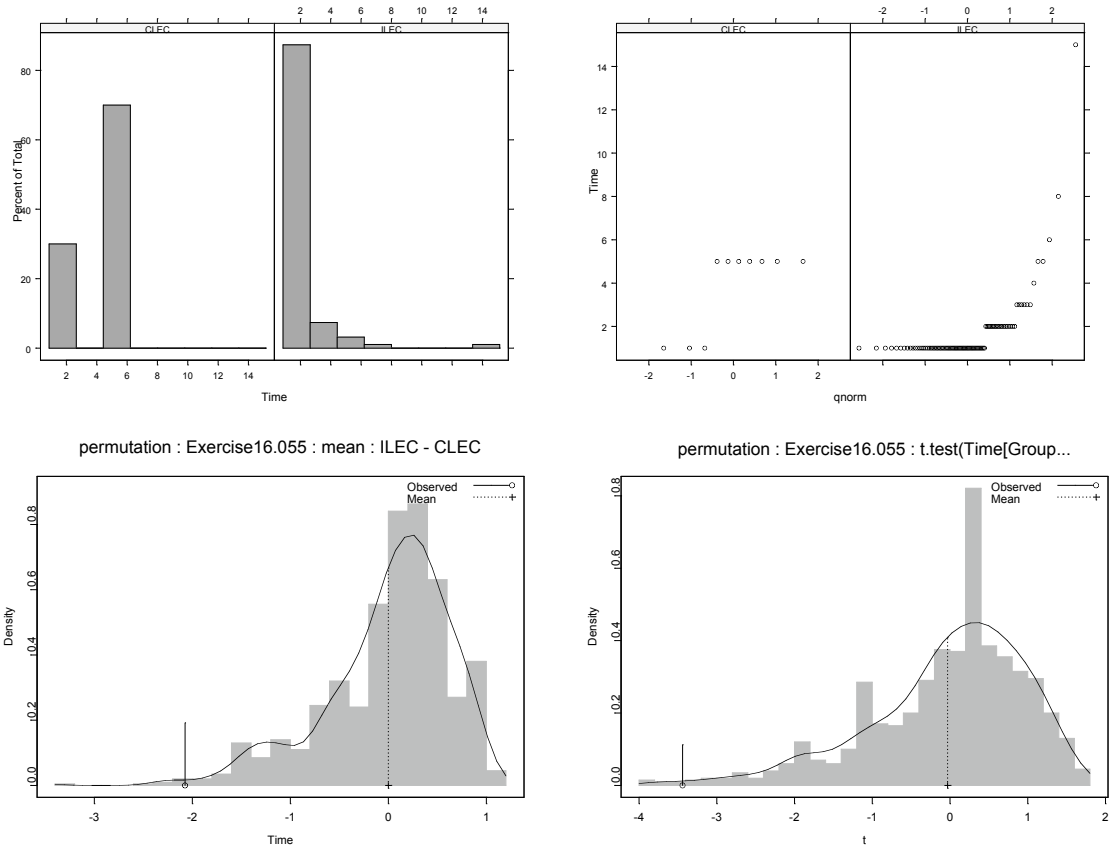




16.55. (a) The ILEC distribution is clearly skewed to the right. The CLEC distribution is skewed slightly to the left. The means and spreads appear to be similar. **(b)** The *P*-value is 0.004 for a one-sided test against the alternative that the mean for the CLEC distribution is greater. **(c)** The *P*-value is 0.0073. The permutation test does not require Normal distributions and gives more accurate answers in the case of skewness. A plot of the permutation distributions shows there is substantial skewness. **(d)** The difference is significant at both the 5% and 1% levels.

```
Exercise16.055 # variables Time and Group (ILEC or CLEC)
histogram(~Time | Group, data=Exercise16.055)
qqmath(~Time | Group, data=Exercise16.055)
t.test(Exercise16.055$Time, treatment = Exercise16.055$Group,
       alternative="less", var.equal=F)
# It is easy to do perm test for difference in means:
perm55a = permutationTestMeans(Exercise16.055, treatment=Group,
                           alternative="less", seed=0)
perm55a
permutationTestMeans(Exercise16.055, treatment=Group, B=99999, # higher
accuracy
                           alternative="less", seed=0)
```

```
# It is harder to do a perm test where the test statistics is a
# pooled-variance t; need to ensure that only one variable is permuted.
# As a side note, if you use the same random numbers, then a permutation
# test using the difference in means gives exactly the same P-value as
# one based on the pooled-variance t-statistics.  So we rarely bother
# with the latter.
perm55b = permutationTest(Exercise16.055, resampleColumns = "Time",
                          alternative="less",
          t.test(Time[Group == "ILEC"],
                 Time[Group == "CLEC"],
                 alternative="less", var.equal=T)$statistic, seed=0)
perm55b
plot(perm55a)
plot(perm55b)
# permutation distributions are very skewed
```



General Comment—Permutation Tests Using Difference in Means or *t* Statistic

For permutation testing comparing two groups, the test statistic can be the difference in means or a *t* statistic. It is easier to do the former. And what is particularly interesting is that the *P*-value is exactly the same if you use the difference in means or the *t* statistic with pooled variance, as long as you use the same random numbers. That is because in permutation tests, there is a 1-1 monotone relationship between the difference in means and the pooled-variance *t* statistic. For that matter, we could use just one of the means as the test statistic. As a check of this in S-PLUS, do:

```
perm55c = update(perm55b,
  statistic = mean(Time[Group == "ILEC"]) - mean(Time[Group == "CLEC"]))
perm55c  # Note the same p-value
plot(perm55b$replicates, perm55c$replicates)
points(perm55b$observed, perm55c$observed, pch=2)
```

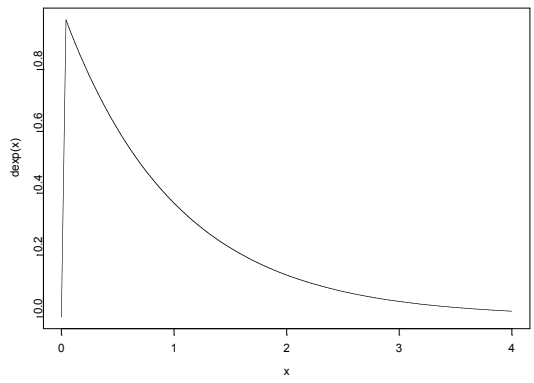
That creates `perm55c`, using the same inputs as `perm55b`, except for a different statistic. We have to use `permutationTest`, rather than the more convenient `permutationTest2` or `permutationTestMeans`, in order to use the same random numbers.

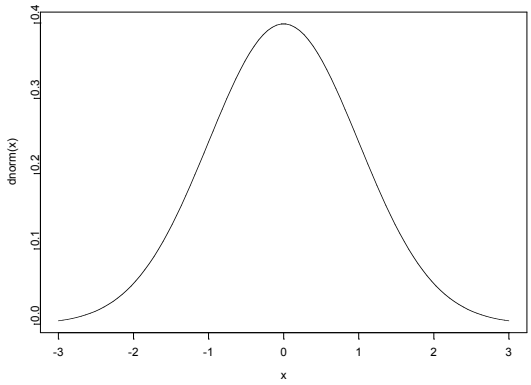
16.56. The SEs are $\sqrt{\qquad\qquad}$, or approximately $\sqrt{\qquad\qquad}$. The SEs are 0.0038 and 0.00019 for DRP and Verizon, respectively.

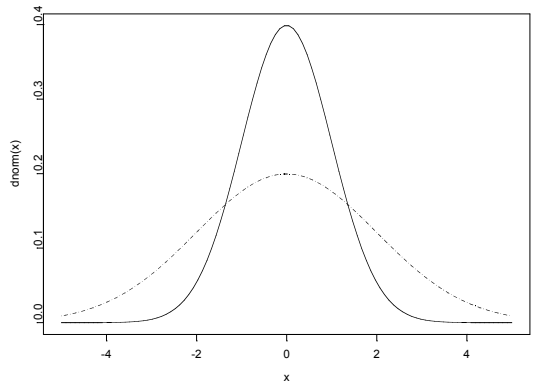
```
sqrt(.015*(1-.015)/1000)      # 0.003843826
sqrt(.0183*(1-.0183)/500000) # 0.0001895527
```

16.57. (a) When the two populations are the same and skewed. **(b)** When the two populations are the same and Normal. **(c)** When the two populations have the same mean but different standard deviations (the non-pooled *t* test would be valid in this case; the pooled *t* test would only be valid if the sample sizes are the same).

```
x = seq(0, 4, length=100); plot(x, dexp(x), type="l") # exponential
x = seq(-3, 3, length=100); plot(x, dnorm(x), type="l") # Normal
x = seq(-5, 5, length=100); plot(x, dnorm(x), type="l")
lines(x, dnorm(x, sd=2), lty=3)
```

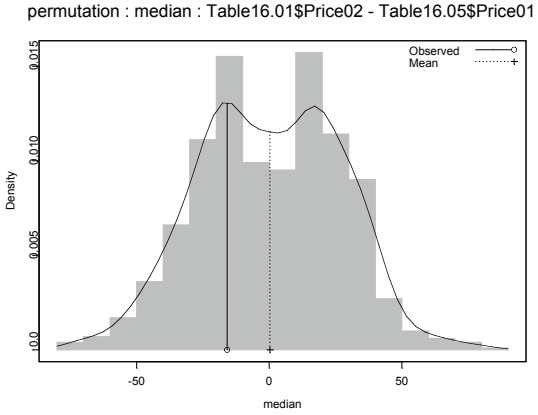






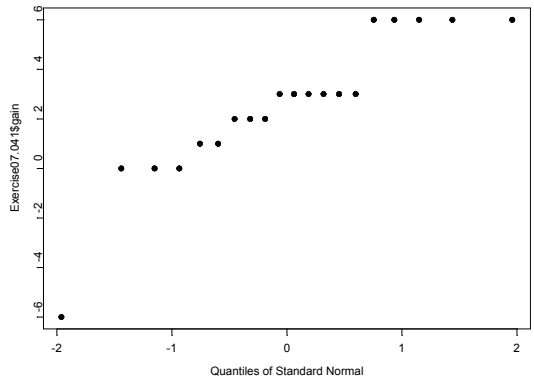
16.58. The *P*-value is 0.608 for a two-sided test. Pure random chance would produce a difference in medians as large as that in the actual data about 61% of the time; we conclude that the difference is easily explained by chance and is therefore not statistically significant. The permutation distribution is bimodal; that does not matter.

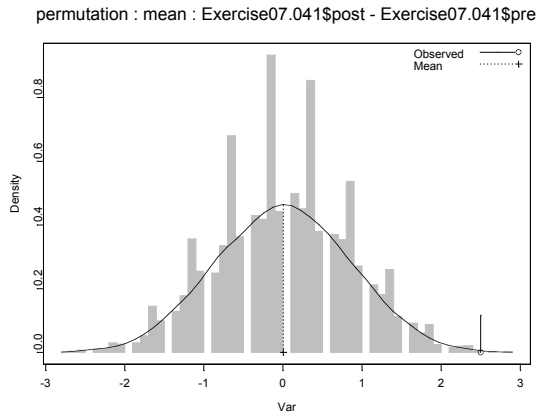
```
perm58 = permutationTest2(Table16.01$Price02,
                          data2=Table16.05$Price01, median, seed=0)
perm58
plot(perm58)
```

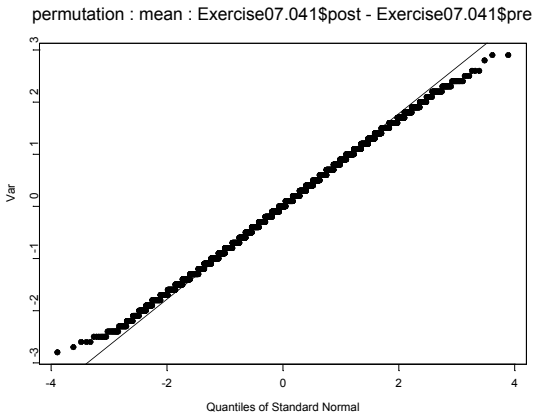


16.59. (a) H_0: $\underline{\mu} = 0$, and H_a: $\mu > 0$, where μ is the population mean difference before and after the summer language institute. The matched-pairs t-value is 3.86 on 19 degrees of freedom, with a P-value of .0005. **(b)** See figure. **(c)** The odd appearance is because the mean difference can only be a multiple of 0.1, so the permutation distribution has many evenly spaced modes. The histogram bars include different numbers of these modes. The P-value is 0.001. Both tests lead to the same conclusion: that the difference is statistically significant. As a side note, the phenomenon of histogram bars including different numbers of modes is common. But this is a rare case where the mode counts differ by as much as 2, rather than 1. Here, both the histogram bars and the modes have a spacing of 0.1. Algebraically, the modes and bar edges are multiples of 0.1. But decimal numbers are rounded on a computer that uses binary arithmetic, and the rounding occurs differently for bar edges and modes, which are differences of two means. After floating-point arithmetic comparisons, some histogram bars end up including the modes at both edges, while other bars include the mode at just one edge, and still others none.

```
t.test(Exercise07.041$gain, alternative = "two.sided")
# You may change the alternative to "less" or "greater"
qqnorm(Exercise07.041$gain)
perm59 = permutationTestMeans(Exercise07.041$post,
  data2=Exercise07.041$pre, paired = T, B=9999,
  alternative="two.sided", seed=0)
plot(perm59)
qqnorm(perm59)
perm59
#Instructor:
t.test(Exercise07.041$gain, alternative = "greater")
perm59b = permutationTestMeans(Exercise07.041$post,
  data2=Exercise07.041$pre, paired = T, B=9999,
  alternative="greater", seed=0)
perm59b
```

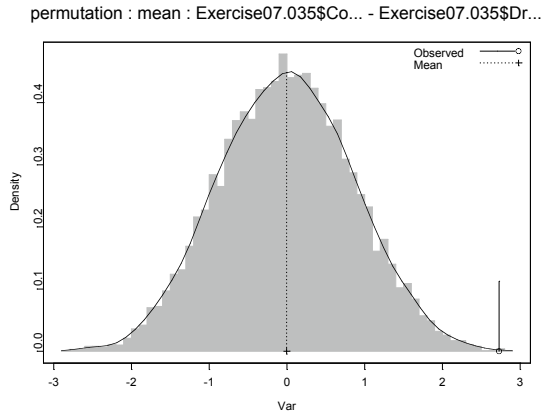






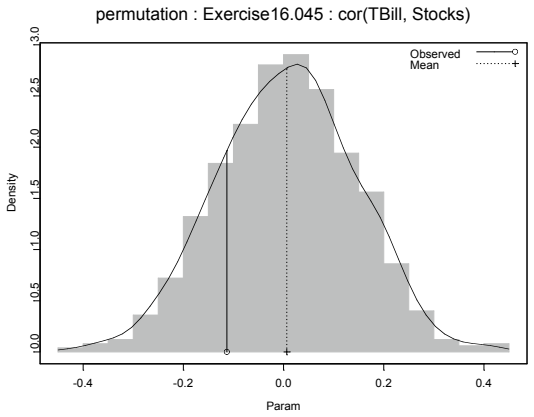
16.60. (a) H_0: $\mu_1 - \mu_2 = 0$, where ρ is the population correlation. H_a: $\mu_1 - \mu_2 _ 0$. Note that the conjecture here is made only after looking at the data (see Exercise 7.35) so a two-sided alternative is appropriate. The *t* value is 4.358, and the *P*-value is 0.0003. This provides significant evidence against the null hypothesis. **(b)** The permutation test *P*-value is 0.0006. Answers will vary due to randomness. This is reasonably close to the value from the *t* test.

```
t.test(Exercise07.035$Computer, Exercise07.035$Driver, paired=T)
perm60 = permutationTestMeans(Exercise07.035$Computer,
  data2=Exercise07.035$Driver, paired=T, B=9999, seed=0)
perm60
plot(perm60)
```



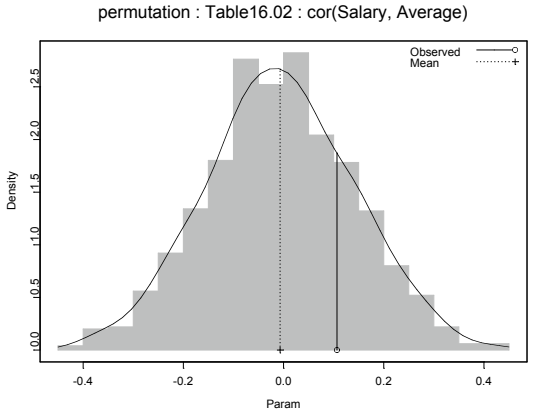
16.61. H_0: $\rho = 0$, H_a: $\rho _ 0$. The two-sided P-value is 0.338. We conclude that there is not significant evidence against the null hypothesis; the true correlation could well be zero. This is consistent with the earlier conclusion.

```
perm61 = permutationTest(Exercise16.045, resampleColumns = "TBill",
                cor(TBill, Stocks), seed=0)
plot(perm61)
perm61
```



16.62. (a) H_0: $\rho = 0$, where ρ is the population correlation. H_a: $\rho > 0$. **(b)** The P-value is 0.226. This does not provide significant evidence against the null hypothesis.

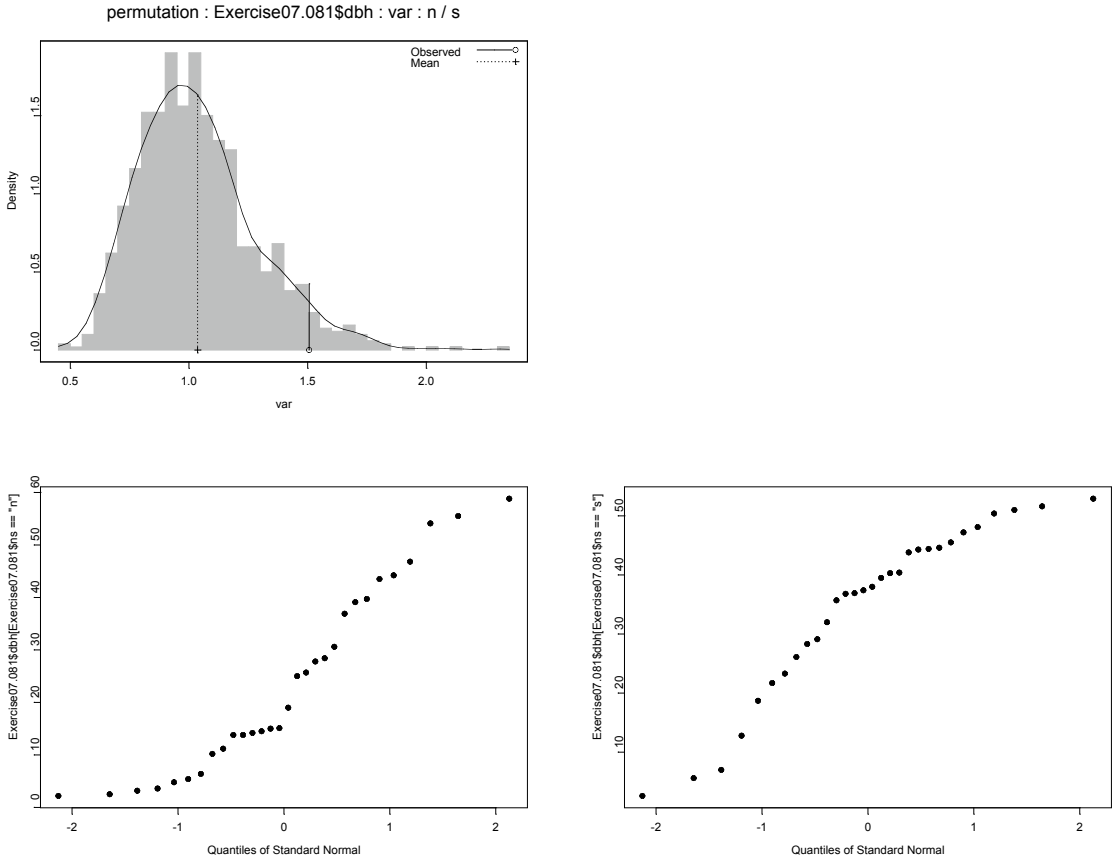
```
Table16.02 # variables Salary and Average
# must permute only one of the variables
perm62 = permutationTest(Table16.02, resampleColumns = "Average",
                cor(Salary, Average), alternative="greater", seed=0)
plot(perm62)
perm62
```



16.63. The permutation test P-value is 0.094. The F statistic is 1.50 on (29,29) df, and the P-value is 0.2757. The results are very different, by a factor of three. In this case the permutation test P-value is smaller, which is typical of short-tailed distributions.

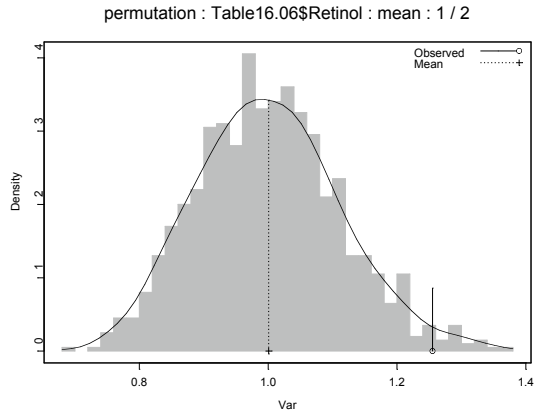
```
perm63 = permutationTest2(Exercise07.081$dbh, var,
  treatment = Exercise07.081$ns, ratio = T, seed=0)
perm63
plot(perm63)
var.test(Exercise07.081$dbh[Exercise07.081$ns == "n"],
         Exercise07.081$dbh[Exercise07.081$ns == "s"])
# Check for long- or short-tailed distributions
```

```
qqnorm(Exercise07.081$dbh[Exercise07.081$ns == "n"])
qqnorm(Exercise07.081$dbh[Exercise07.081$ns == "s"])
```



14.64. (a) The means for the two groups are 0.620 and 0.778. **(b)** H_0: $r = 1$, where r is the ratio of means between the two groups. H_a: $r > 1$. The one-sided P-value is 0.019. The permutation distribution is centered at about 1, standard deviation 0.11, and approximately Normal with some right skewness. We expect the permutation distribution to be centered at about 1 because that is the null hypothesis value for the ratio. When we do permutation sampling, we sample in a way that is consistent with the null hypothesis.

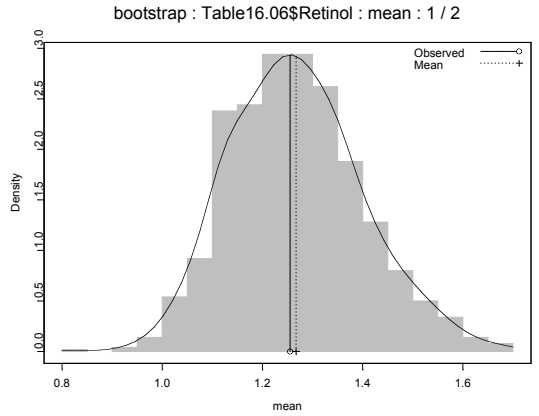
```
Table16.06 # variables Retinol and Group
groupMeans(Table16.06$Retinol, Table16.06$Group)
perm64 = permutationTestMeans(Table16.06$Retinol,
  treatment = Table16.06$Group, ratio=T, alternative="greater", seed=0)
plot(perm64)
perm64
```

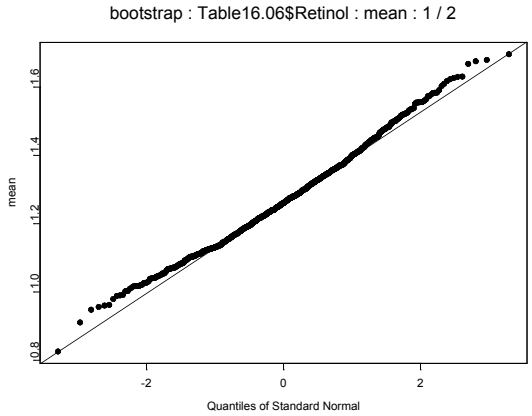


16.65. For the permutation test, we must resample in a way that is consistent with the null hypothesis. Hence, we pool the data (assuming that the two populations are the same), and draw data for each group from the pooled data. For the bootstrap, we do not assume the null hypothesis is true, so we do not assume that the two populations are the same. We sample from each of the two datasets separately, rather than pooling the data first.

14.66. (a) The BCa interval is (1.018, 1.532), so we would finish the sentence: "is (1.02, 1.53); the ratio is marginally statistically significantly different from 1.0." **(b)** The bootstrap distribution is right skewed, with moderate bias, hence a BCa or tilting interval would be preferred and a *t* interval would be inaccurate. The bootstrap percentile interval is (1.031, 1.555), which is a bit different than the BCa interval. The most important difference is that it is about 1.5 times as far on the left side from the null hypothesis value of 1.0, so it would give an impression of stronger statistical significance.

```
boot66 = bootstrap2(Table16.06$Retinol, mean,
  treatment = Table16.06$Group, ratio=T, seed=0)
plot(boot66)
qqnorm(boot66)
boot66
limits.bca(boot66)
limits.percentile(boot66)
```

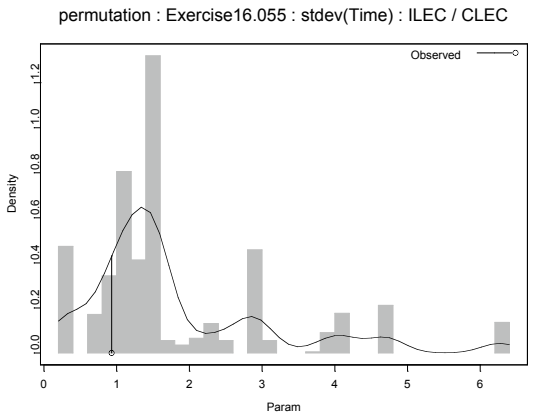




16.67. (a) Some of the ratios were infinite because the permutation test produced a standard deviation of 0 in the denominator. The mean and SE of the permutation distribution are undefined, but we can still obtain the *P*-value of 0.366. This *P*-value indicates that there is not strong evidence that the variability in the repair times for ILEC and CLEC customers differ.

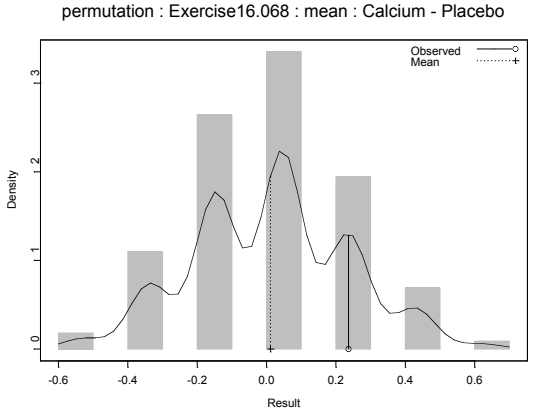
(b) These *P*-values are very different (0.67 vs. 0.366), indicating that the *F* test is inaccurate. The *F* test depends on Normality; here the distributions are very non-Normal.

```
perm67 = permutationTest2(Exercise16.055, stdev(Time), treatment=Group,
                          ratio = T, seed=0)
perm67
plot(perm67)
table(Exercise16.055$Group)  # 10 CLEC, 95 ILEC
var67 = groupVars(Exercise16.055$Time, Exercise16.055$Group) # 3.73, 3.24
(1 - pf( var67[1] / var67[2], 9, 94 )) * 2
# 0.67
```



16.68. The permutation distribution is roughly Normal, except for discreteness, with about seven equally spaced values. The two-sided *P*-value is 0.546; we conclude that there is not significant evidence that the population proportions differ.

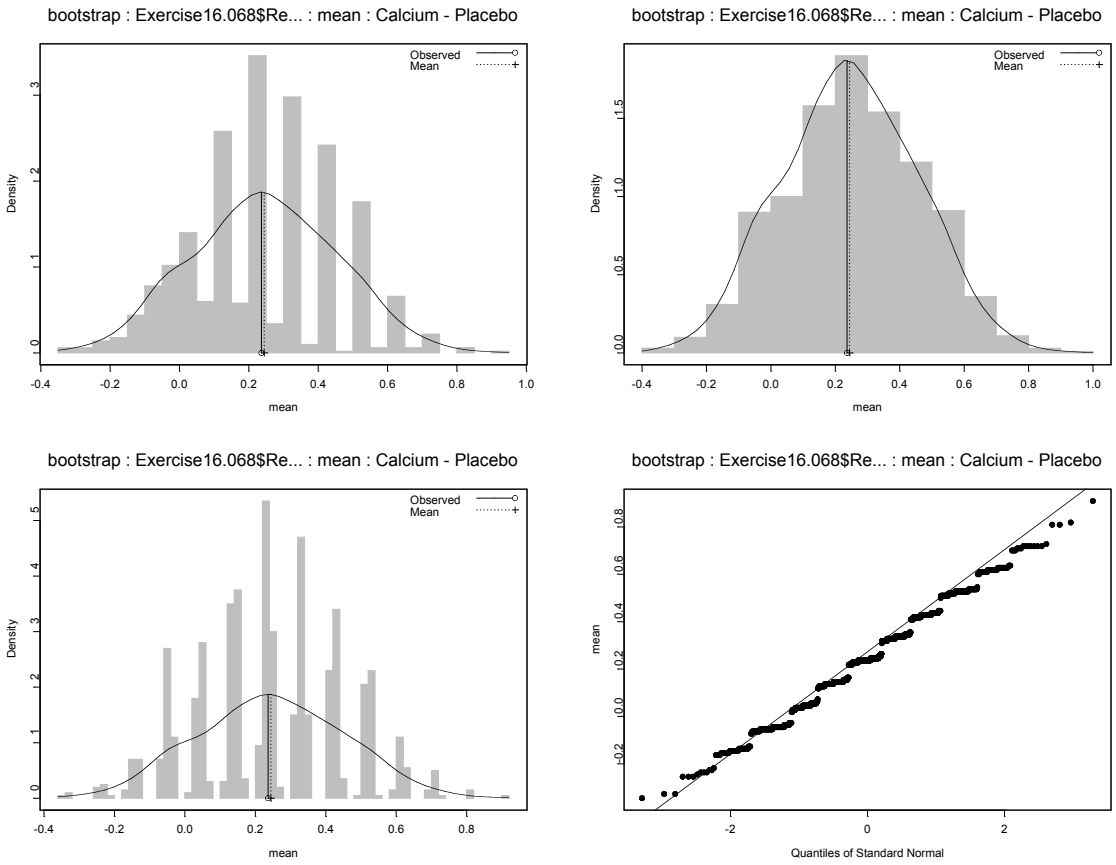
```
# From the graphical interface we can enter the numbers in the table.
# Here we have to expand them into a dataset of 31 observations.
# We'll let 1 = success, 0 = fail, so the proportion of successes
# is the same as the mean.
Exercise16.068 = data.frame(
  Group = rep(c("Calcium","Placebo"), c(10,11)),
  Result = rep(c(1,0,1,0), c(6,4,4,7)))
groupMeans(Exercise16.068$Result, Exercise16.068$Group) # check of data
perm68 = permutationTestMeans(Exercise16.068, treatment=Group, seed=0)
plot(perm68)
perm68
```



16.69. (a) The plus four interval is (–.186, 0.583). **(b)** The bootstrap *z* interval is (–0.17, 0.64), the percentile interval is (–0.15, 0.63), the BCa interval is (–0.16, 0.62), and tilting interval is (–0.21,

0.62). The tilting interval reaches farther to the left than do the other bootstrap intervals; the left endpoint of the plus four interval falls between that of the tilting and the other bootstrap intervals. The tilting interval is wider than the plus four interval. As a side note, the tilting interval is preferred here because it handles discrete or semi-discrete bootstrap distributions for proportions better than do the percentile or BCa intervals. The latter two could give confidence interval endpoints only at one of the discrete values in the bootstrap distribution, while the tilting interval can give endpoints that fall between those values.

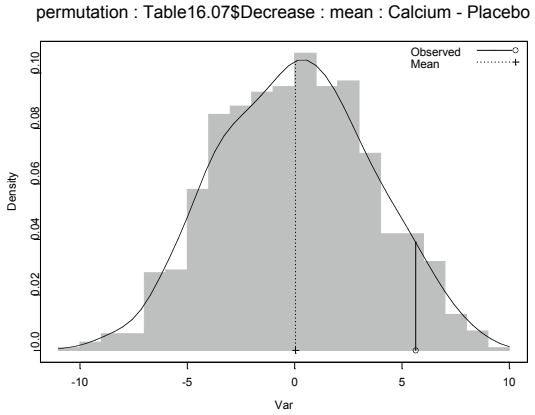
```
p1 = (6+1)/(10+2)
p2 = (4+1)/(11+2)
p1 - p2 + c(-1,1) * qnorm(.975) * sqrt(p1*(1-p1)/(10+2) + p2*(1-p2)/(11+2))
boot69 = bootstrap2(Exercise16.068$Result, mean,
  treatment=Exercise16.068$Group, seed=0)
plot(boot69)  # Unusual plot -- high-low pattern.
# Histograms are sensitive to just where the bin edges are.
plot(boot69, nclass=10)  # more reasonable
plot(boot69, nclass=40)  # shows gaps
qqnorm(boot69)# note the bunching & gaps, explains the high-low pattern
boot69
limits.t(boot69, z=T)  # Do a z rather than t interval
limits.percentile(boot69)
limits.bca(boot69)
limits.tilt(boot69)
```



16.70. The one-sided *P*-value is 0.065. The difference is not statistically significant (though it is borderline).

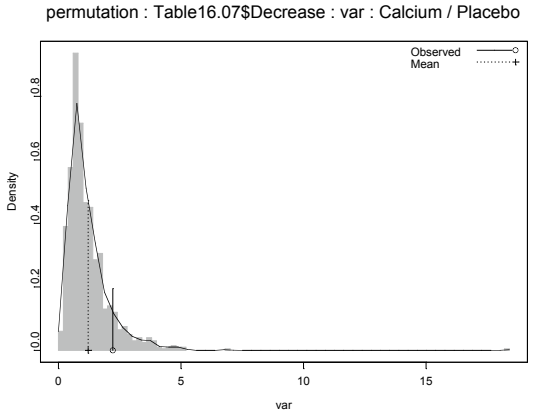
```
Table16.07 # variables:  Group GroupID Begin End Decrease
perm70 = permutationTestMeans(Table16.07$Decrease,
  treatment = Table16.07$Group, alternative="greater", seed=0)
```

```
plot(perm70)
perm70
```



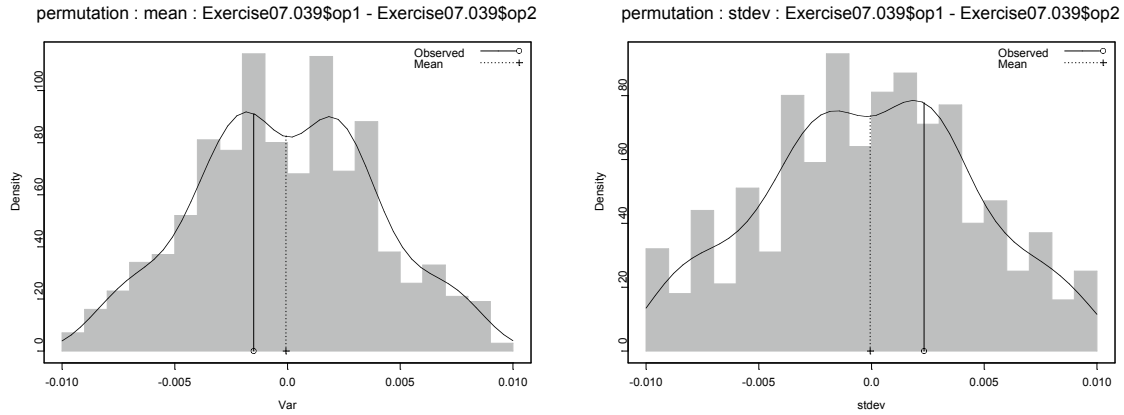
16.71. **(a)** H_0 is that $\sigma_1 = \sigma_2$; H_a is $\sigma_1 _ \sigma_2$, where σ_1 and σ_2 are the standard deviations of the two populations. **(b)** The *P*-value is 0.212; there is not significant evidence that the population standard deviations differ. **(c)** The *P*-value from the *F* test is 0.236. The data are approximately Normal, so the *F* test is reasonably accurate.

```
perm71 = permutationTest2(Table16.07$Decrease, var, ratio=T,
  treatment = Table16.07$Group, seed=0)
plot(perm71)
perm71
```



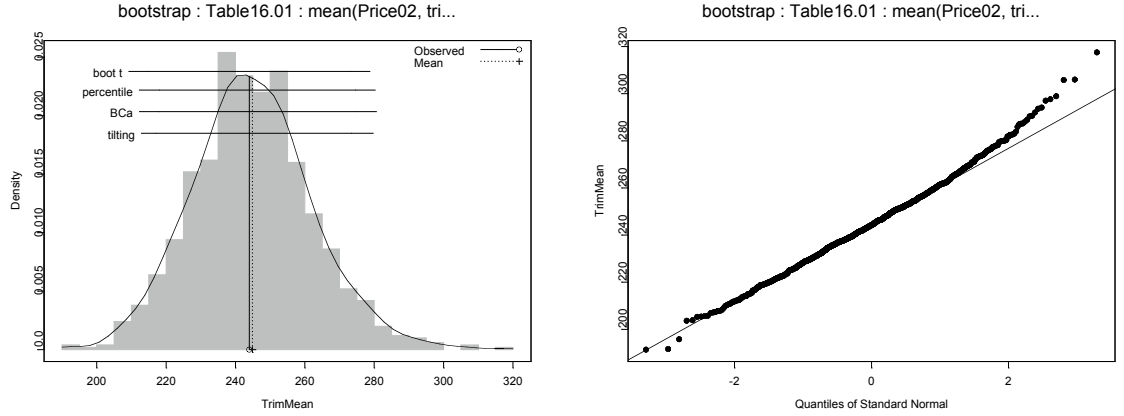
16.72. A permutation test comparing the means gives a *P*-value of 0.80; a permutation test comparing the standard deviations gives a *P*-value of 0.67. In both cases, there is not statistically significant evidence of a difference between the operators. The differences could easily arise by chance—even larger differences would occur by chance more than half the time.

```
perm72a = permutationTestMeans(Exercise07.039$op1,
  data2=Exercise07.039$op2, paired=T, seed=0)
plot(perm72a)
perm72a
perm72b = permutationTest2(Exercise07.039$op1,
  data2=Exercise07.039$op2, stdev, paired=T, seed=0)
plot(perm72b)
perm72b
```



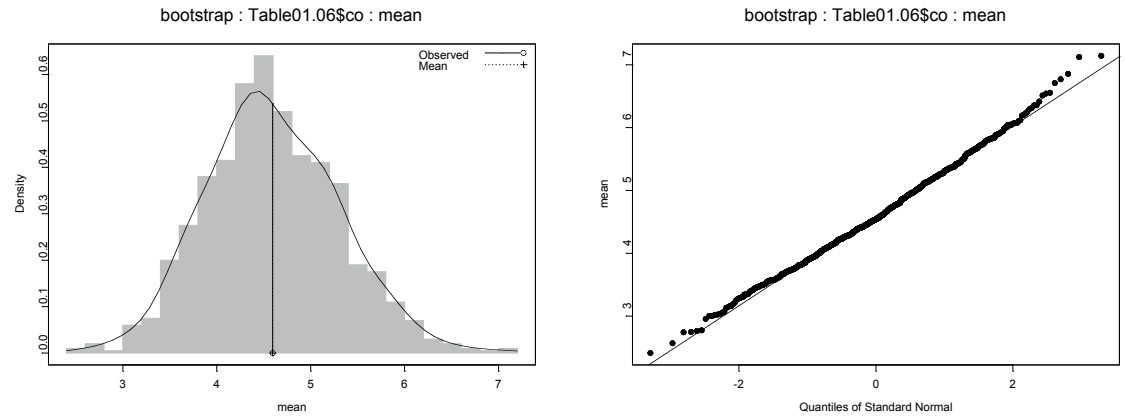
16.73. The bootstrap *t* interval is (209, 279), percentile interval is (212, 280), BCa interval is (212, 281), and tilting interval is (213, 280). The right endpoints are similar, but the left endpoint for the bootstrap *t* is different from the other intervals. The bootstrap *t* ignores the skewness in the bootstrap distribution; here it appears that any of the other intervals (including the percentile interval) would be preferable. The percentile and *t* intervals would be adequate if high accuracy is not needed.

```
# boot18 = bootstrap(Table16.01, mean(Price02, trim=.25), seed=0)
limits.t(boot18)
limits.percentile(boot18)
limits.bca(boot18)
limits.tilt(boot18)
qqnorm(boot18)
plotWithIntervals(boot18)
```



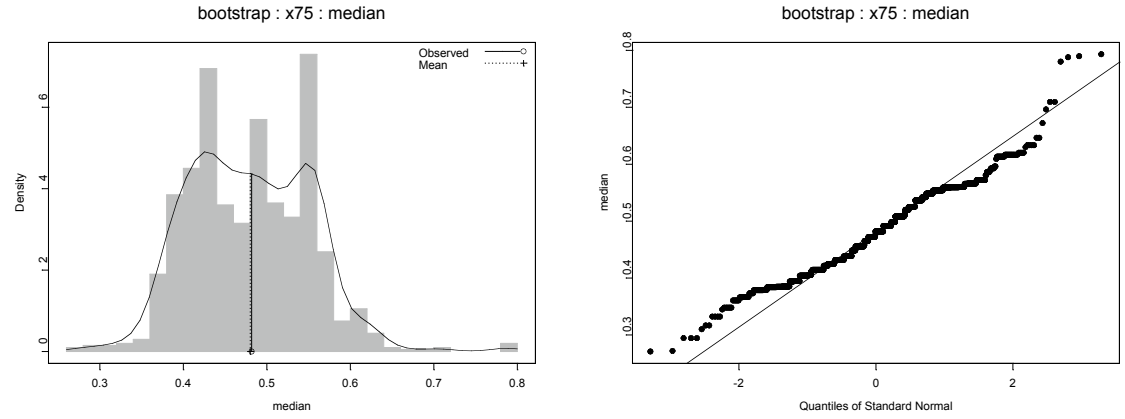
16.74. (a) The bootstrap distribution is approximately Normal, with small positive skewness. A *t* interval should be okay unless high accuracy is needed. (b) The *t* limits are (3.17, 6.03), and the BCa limits are (3.47, 6.31). (c) The BCa limits are to the right of the *t* limits, as expected. Hence, it is possible that the *t* limits are not very accurate. But the BCa is highly variable with only 1000 replications, so does not provide a good diagnostic for the *t* intervals here.

```
# boot5 = bootstrap(Table01.06$co, mean, seed=0)
plot(boot5)
qqnorm(boot5)
limits.t(boot5)
limits.bca(boot5)
```



16.75. In this problem, the answers will depend strongly on the random sample drawn. **(a)** For a uniform distribution on 0 to 1, the population median is 0.5. The shape of the bootstrap distribution will depend on the sample drawn. **(b)** The bootstrap SE is 0.054. A 95% bootstrap *t* interval is (0.45, 0.67). (Both depend heavily on the sample drawn.) **(c)** The intervals are (0.43, 0.63) and (0.45, 0.63). (Both depend heavily on the sample drawn.) The *t* interval is not reliable here. Students will probably say so and give an explanation that the bootstrap distribution does not appear Normal. The actual reason is that the bootstrap SE is unreliable, depending strongly on the sizes of the gaps between the observations near the middle. As a side note, even though the bootstrap distribution shapes and spreads are unreliable estimates for sample medians, the bootstrap percentile, BCa, and tilting intervals are not bad. They give answers similar to the exact rank-based confidence intervals obtained by inverting hypothesis tests. One variation of tilting intervals matches the exact intervals.

```
set.seed(1)
x75 = runif(50)
median(x75)
boot75 = bootstrap(x75, median, seed=0)
plot(boot75)
qqnorm(boot75)
boot75
limits.t(boot75)
limits.bca(boot75)
limits.tilt(boot75)
```



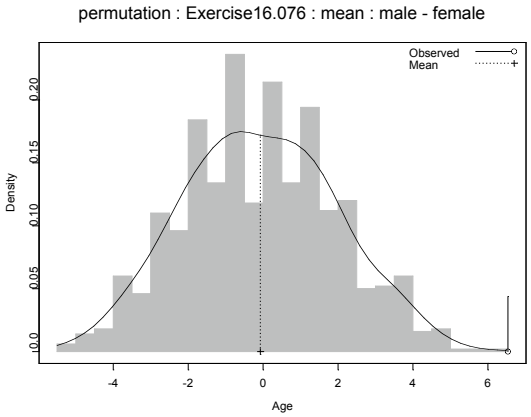
16.76. (a) The difference appears quite large; it should be significant.

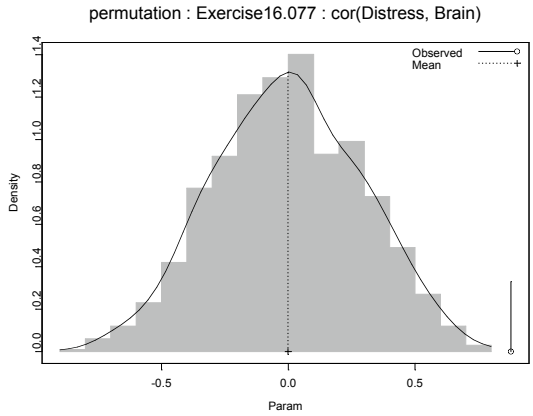
```
: 1 : 9
: 2 : 01
```

```
  3 : 2 : 2233
  5 : 2 : 4
  6 : 2 :
899 : 2 :
 01 : 3 :
 22 : 3 :
  5 : 3 :
```

(b) The *P*-value is about 0.00056 (based on 100,000 resamples). For 1000 resamples, the answer will usually be either 0.002 or 0.004 (round up, then double). **(c)** We conclude that there is significant evidence that the mean ages differ. Note: With enough resamples we would see that the permutation test *P*-value is consistent with the answer from a *t* test, but students doing only 1000 resamples will get a larger answer. They may then answer that the *t* test *P*-value is too small.

```
Exercise16.076 # variables Age and Sex (male or female)
stem(Exercise16.076$Age[ Exercise16.076$Sex == "male" ],
     scale = -1, nl=2)
stem(Exercise16.076$Age[ Exercise16.076$Sex == "female" ],
     scale = -1, nl=2)
# do back-to-back stemplot by hand
perm76 = permutationTestMeans(Exercise16.076, treatment=Sex, seed=0)
plot(perm76)
perm76
permutationTestMeans(Exercise16.076, treatment=Sex, B=99999, seed=0)
```





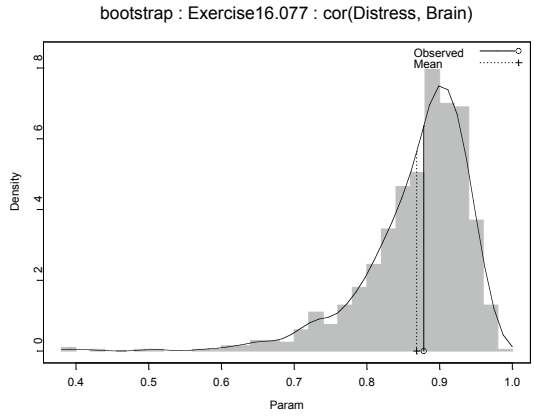
16.77. The one-sided permutation test using 1000 samples gives a *P*-value of 0.001, the smallest possible value. We conclude there is a significant positive relationship. (The figure is above.)

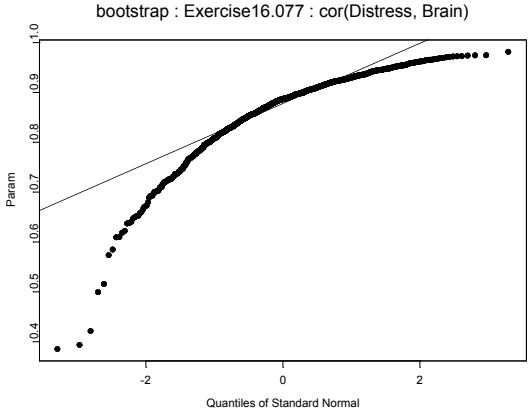
```
plot(Exercise16.077$Distress, Exercise16.077$Brain)
perm77 = permutationTest(Exercise16.077,
  cor(Distress, Brain), resampleColumns = "Brain",
  alternative = "greater", seed=0)
plot(perm77)
perm77
```

16.78. (a) The permutation distribution is centered about 0.0, because for a hypothesis test we resample in a way that is consistent with the null hypothesis of no correlation. In contrast, the bootstrap is centered approximately at the observed correlation of 0.878. The right tail is bounded above by 1.0, whereas the left tail can be much longer. **(b)** The bootstrap BCa confidence interval is (0.652, 0.955); the percentile interval is (0.680, 0.961). Neither is even close to including the value of 0; we conclude that there is a significant positive relationship. The bootstrap *t* interval is (0.717, 1.039), the tilting interval is (0.675, 0.948).

```
boot78 = bootstrap(Exercise16.077, cor(Distress, Brain), seed=0)
plot(boot78)
```

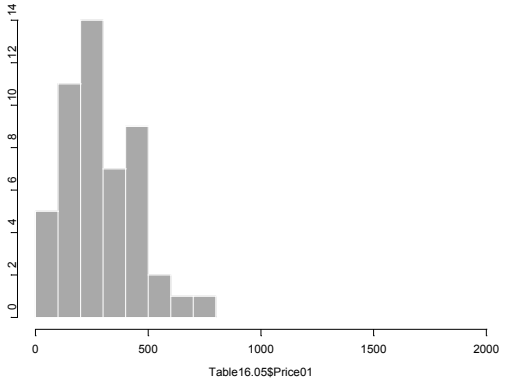
```
qqnorm(boot78)
limits.t(boot78)
limits.percentile(boot78)
limits.bca(boot78)
limits.tilt(boot78)
```

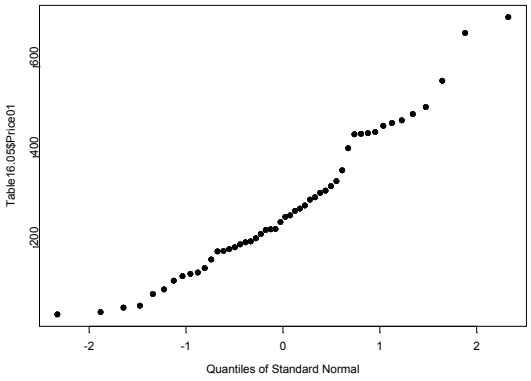


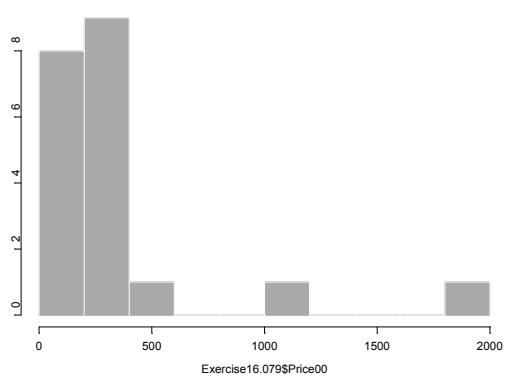


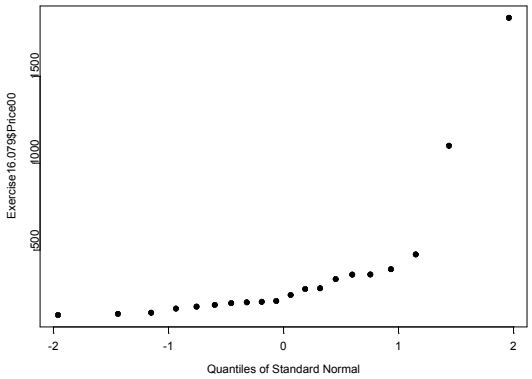
16.79. (a) The 2000 data is strongly right-skewed with two outliers. This violates the guideline for using the *t* procedure given in Chapter 7, namely for a sample size of at least 15, that *t* procedures can be used except in the presence of outliers or strong skewness. The 2001 data is approximately Normal. **(b)** The *P*-value for the permutation test for the difference in means is 0.292. We conclude that there is not strong evidence that the mean selling prices are different for all Seattle real estate in 2000 and in 2001.

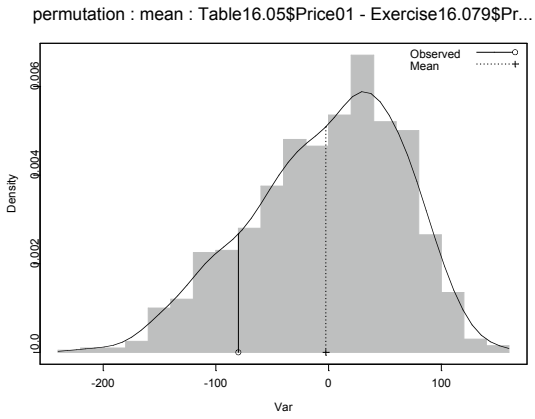
```
Exercise16.079 # variable Price00
par(mfrow=c(2,2))  # do plots on a 2x2 grid
hist(Table16.05$Price01, xlim=c(0,2000))
qqnorm(Table16.05$Price01)
hist(Exercise16.079$Price00, xlim=c(0,2000))
qqnorm(Exercise16.079$Price00)
par(mfrow=c(1,1))  # back to Normal
perm79 = permutationTestMeans(Table16.05$Price01,
  data2 = Exercise16.079$Price00, seed=0)
plot(perm79)
perm79
```





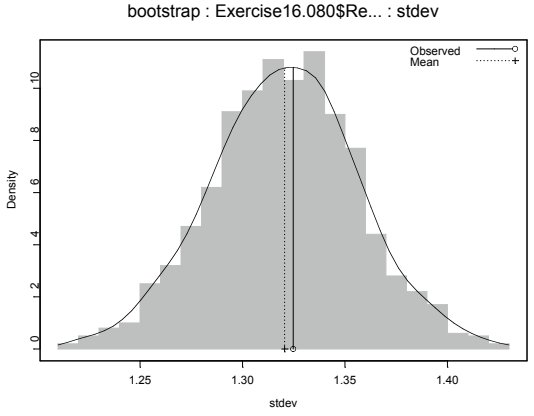


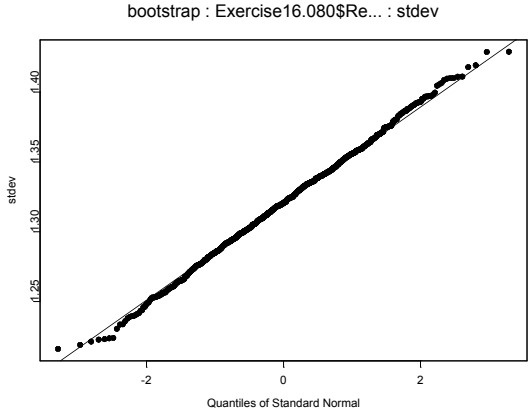




16.80. (a) $s = 1.32$. **(b)** The bootstrap SE for s is 0.036. **(c)** The bootstrap t interval is (1.25, 1.40). **(d)** The BCa interval is (1.26, 1.41); the tilting interval is (1.26, 1.41). The bootstrap t interval is a bit low on both ends.

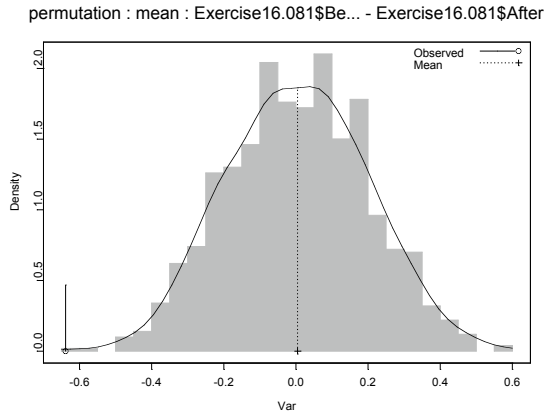
```
stdev(Exercise16.080$Return)
boot80 = bootstrap(Exercise16.080$Return, stdev, seed=0)
boot80
plot(boot80)
qqnorm(boot80)
limits.t(boot80)
limits.bca(boot80)
limits.tilt(boot80)
```





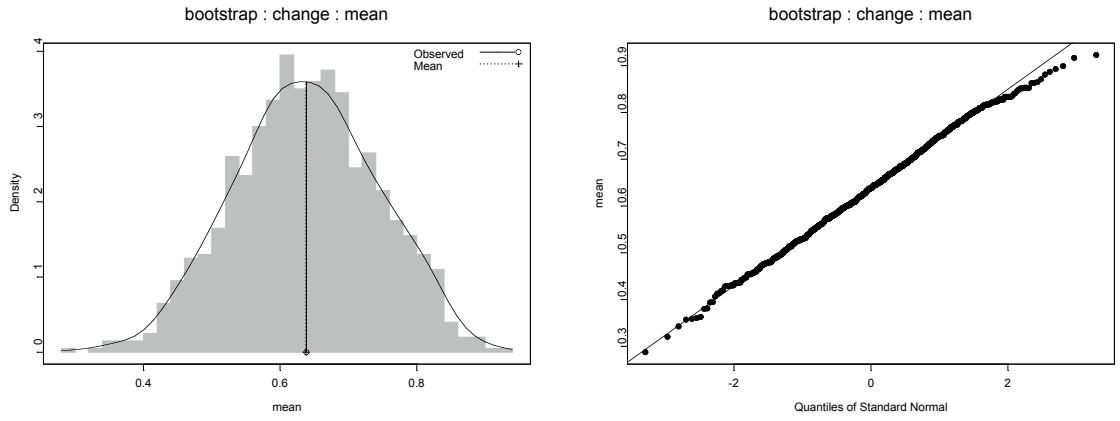
16.81. (a) We are interested in whether the presentation causes an improvement in glove use. We must use a matched pairs method because the same nurses were recorded both times. **(b)** The *P*-value is 0.002. This provides significant evidence that the presentation was helpful.

```
Exercise16.081 # variables Before and After
perm81 = permutationTestMeans(Exercise16.081$Before,
                              data2 = Exercise16.081$After, paired=T,
                              alternative = "less", seed=0)
plot(perm81)
perm81
```



16.82. A 95% bootstrap *t* confidence interval for the mean change (after–before) is (0.41, 0.87). Zero is outside this interval, so the result is significant at the 0.05 level. We conclude that there is strong evidence that the mean change is different from 0, that is, positive. The other bootstrap intervals differ somewhat, but the differences are not nearly enough to change this conclusion.

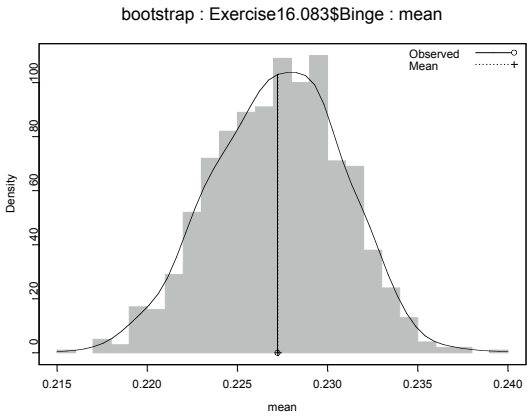
```
change = Exercise16.081$After - Exercise16.081$Before
boot82 = bootstrap(change, mean, seed=0)
plot(boot82)
qqnorm(boot82)
limits.t(boot82)
limits.percentile(boot82)
limits.bca(boot82)
limits.tilt(boot82)
```

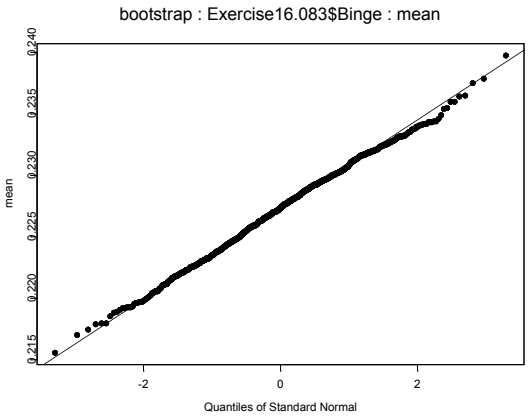


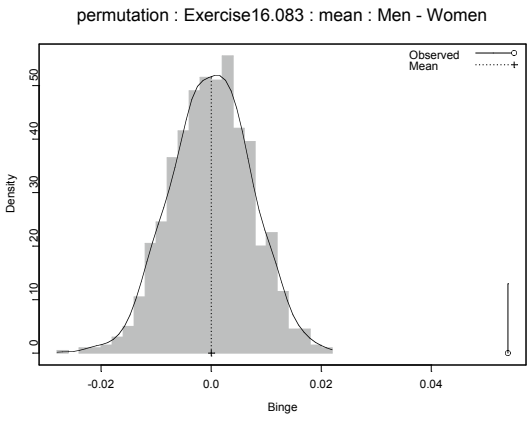
16.83. (a) The bootstrap distribution is very Normal, with no appreciable bias, so a bootstrap *t* interval is adequate; the interval is (0.220, 0.234). A bootstrap percentile interval is nearly identical. **(b)** H_0: $\mu_1-\mu_2=0$, H_a: $\mu_1-\mu_2>0$. The one-sided permutation test *P*-value is 0.001—the smallest possible value with 999 resamples. The plot of the permutation distribution is even more striking,

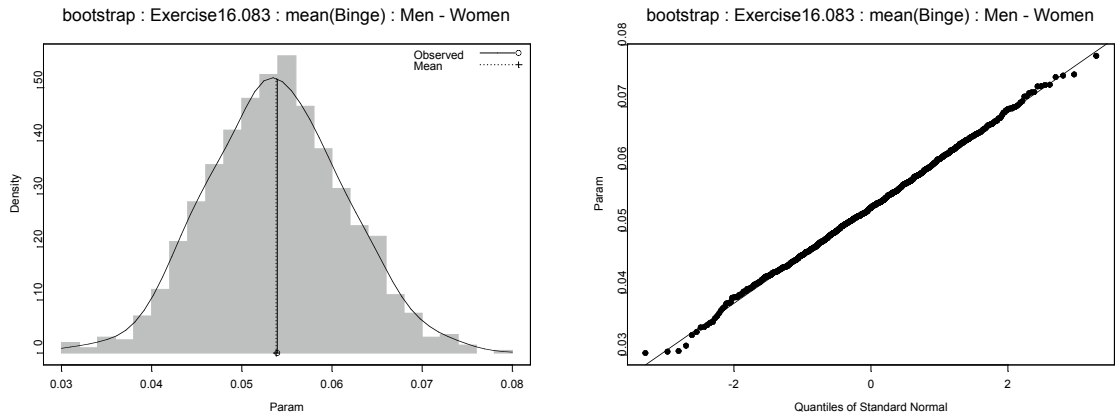
suggesting how exceedingly rare it would be to observe a difference this large by chance. **(c)** The bootstrap distribution for the difference is very Normal, with no appreciable bias. The bootstrap *t* interval for the difference in proportions is (0.039, 0.069); the bootstrap percentile interval is nearly identical.

```
# From the graphical interface we can enter the numbers in the table.
# Here we have to expand them into a dataset of many observations.
# We'll let 1 = binge drinker, 0 = other, so the proportion of binge
# drinkers is the same as the mean.
Exercise16.083 = data.frame(
  Gender = rep(c("Men","Women"), c(5348,8471)),
  Binge = rep(c(1,0,1,0), c(1392,5348-1392,1748,8471-1748)))
boot83a = bootstrap(Exercise16.083$Binge, mean, seed=0)
plot(boot83a)
qqnorm(boot83a)
limits.t(boot83a)
limits.percentile(boot83a)
perm83 = permutationTestMeans(Exercise16.083, treatment=Gender,
  alternative = "greater", seed=0)
plot(perm83)
perm83
boot83b = bootstrap2(Exercise16.083, mean(Binge), treatment=Gender, seed=0)
plot(boot83b)
qqnorm(boot83b)
limits.t(boot83b)
limits.percentile(boot83b)
```



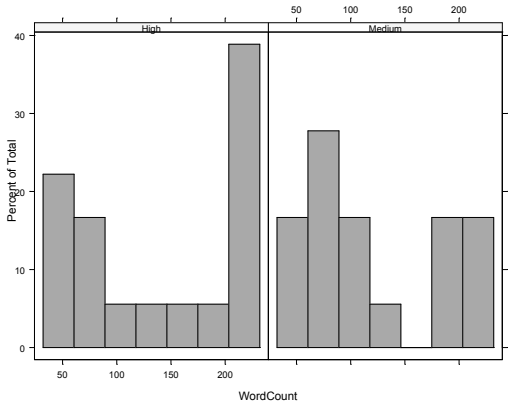


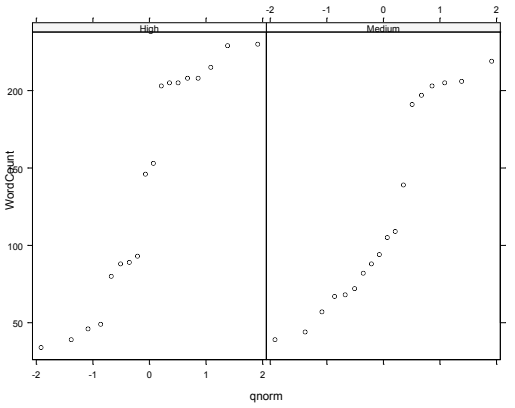




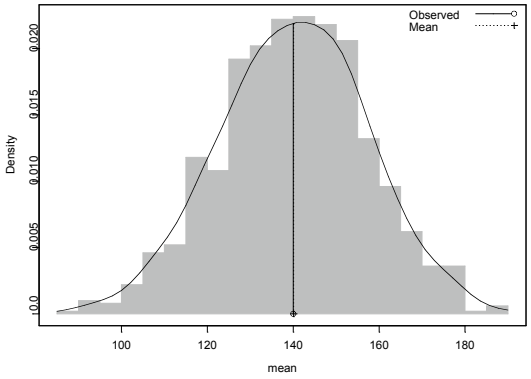
16.84. (a) The histograms do not give the impression of Normality. However, the Normal quantile plots suggest that the underlying distributions could well be Normal; there are no outliers or systematic deviations from Normality. The mean word counts are 140 and 121.4 for the High and Medium education levels, respectively. **(b)** The bootstrap SEs are 17.0 and 14.7 for the High and Medium groups, respectively. **(c)** The bootstrap distributions appear very close to Normal; they are centered at the respective sample means, with about the same amount of spread. **(d)** The percentile intervals are (106.4, 173.7) and (93.3, 146.6), and the tilting intervals are (106.8, 171.1) and (94.6, 150.9). Using either confidence interval method, the intervals for the two groups overlap. This suggests that there may not be a statistically significant difference between the two groups. However, we should do a confidence interval for the difference to answer that question. **(e)** The 95% bootstrap percentile confidence interval for the difference is (–26.3, 62.4). This interval includes zero. The observed data do not provide strong evidence of a difference in mean word counts.

```
Exercise16.084 # variables WordCount and Education (High, Medium)
histogram(~WordCount | Education, data=Exercise16.084)
qqmath(~WordCount | Education, data=Exercise16.084)
groupMeans(Exercise16.084$WordCount, group=Exercise16.084$Education)
x84 = split(Exercise16.084$WordCount, Exercise16.084$Education)
boot84a = bootstrap(x84$High, mean, seed=0)
boot84b = bootstrap(x84$Medium, mean, seed=0)
boot84a
boot84b
par(mfrow=c(2,2)) # 2x2 grid for plots
plot(boot84a);qqnorm(boot84a)
plot(boot84b);qqnorm(boot84b)
par(mfrow=c(1,1)) # back to normal plots
limits.percentile(boot84a)
limits.tilt(boot84a)
limits.percentile(boot84b)
limits.tilt(boot84b)
boot84c = bootstrap2(Exercise16.084$WordCount, mean,
                     treatment = Exercise16.084$Education, seed=0)
plot(boot84c)
qqnorm(boot84c)
limits.percentile(boot84c)
```

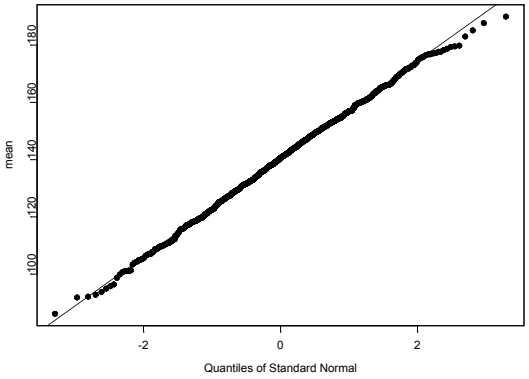




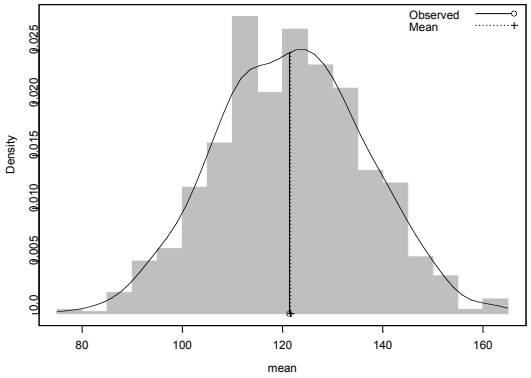
bootstrap : x84$High : mean



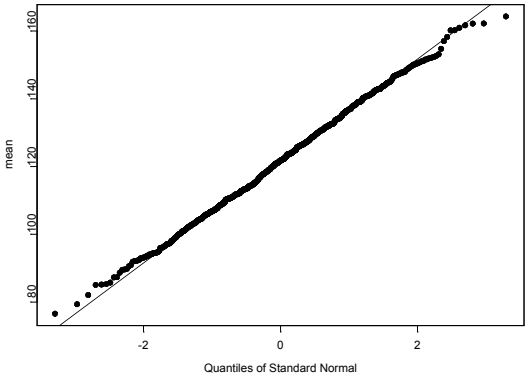
bootstrap : x84$High : mean



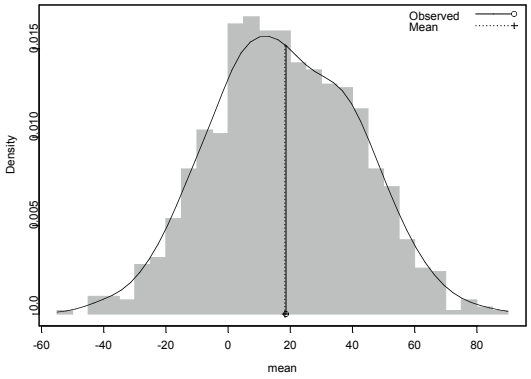
bootstrap : x84$Medium : mean



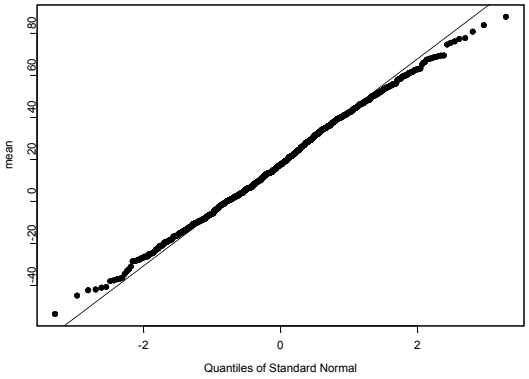
bootstrap : x84$Medium : mean



bootstrap : Exercise16.084$Wo... : mean : High - Medium

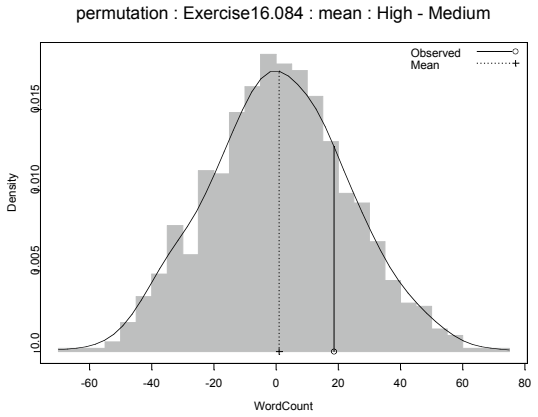


bootstrap : Exercise16.084$Wo... : mean : High - Medium



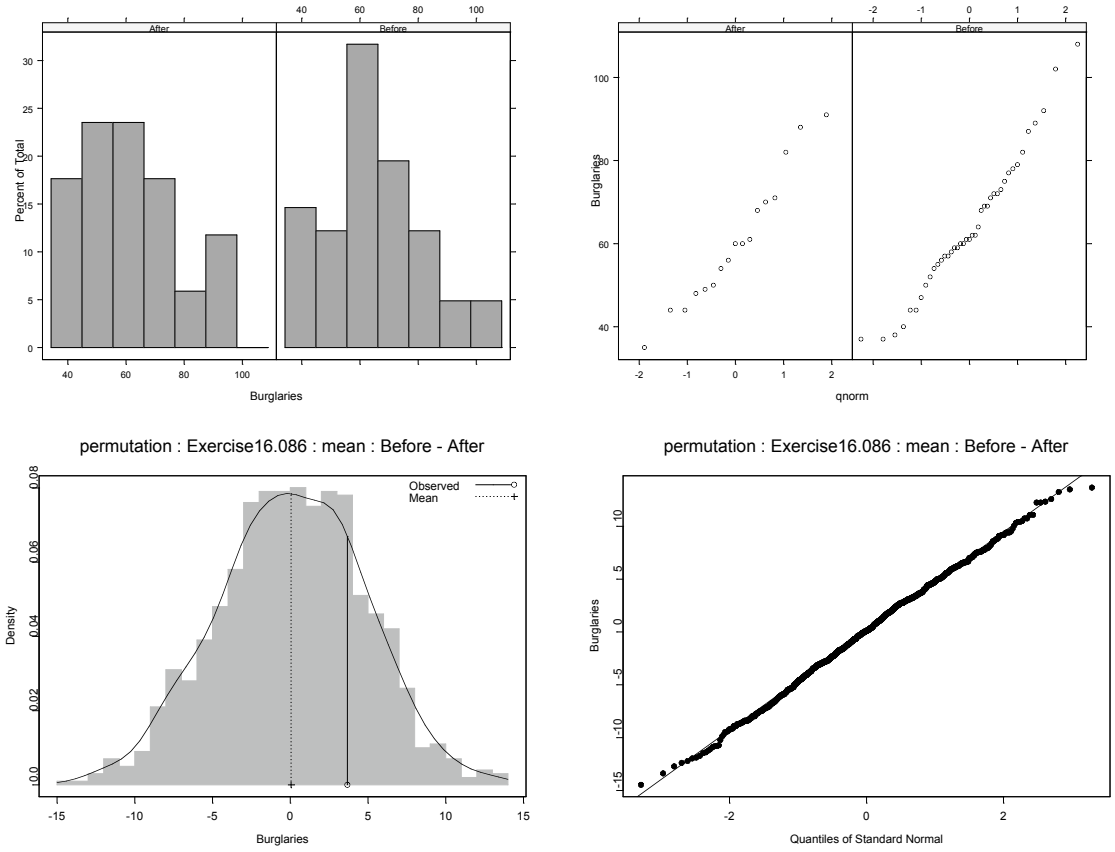
16.85. We test the hypotheses H_0: $\mu_1 = \mu_2$ vs. H_a: $\mu_1 > \mu_2$. The *P*-value is 0.212. Thus, there is not strong evidence that the mean word count is higher for ads placed in magazines aimed at people with high education levels, as opposed to ads placed in magazines aimed at people with medium education levels. The 95% confidence interval in Exercise 14.78 for the difference in means contained 0. This suggests that there is not strong evidence of a difference in mean word counts. We conclude that there is not strong evidence that the mean word count is higher for ads placed in magazines aimed at people with high education levels, as opposed to ads placed in magazines aimed at people with medium education levels.

```
perm85 = permutationTestMeans(Exercise16.084, treatment = Education,
                              alternative = "greater", seed=0)
plot(perm85)
perm85
```



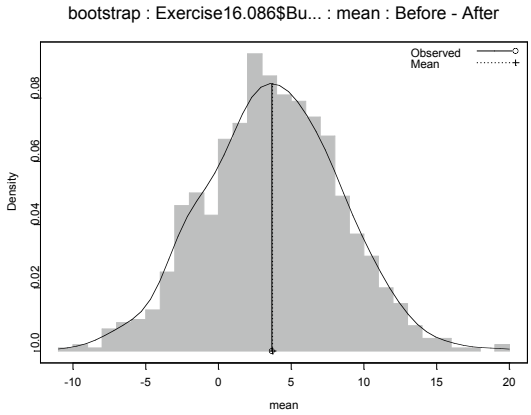
16.86. (a) Roughly Normal. **(b)** The average number of burglaries was 64.3 before and 60.6 after the program began. This is a one-sided test because we wish to test whether or not the program made an improvement. The *P*-value for the two-sample *t* test is 0.22. **(c)** The *P*-value for the permutation test is 0.22. This is the same (after rounding) as for the formula *t* test. **(d)** The *P*-value is 0.788. This is testing whether the data provide strong evidence that burglaries have increased. That is the opposite of our goal, to determine whether the program has reduced burglaries. The *P*-value is more than 50% because the actual change was negative rather than positive; there is greater than a 50% chance that random chance would yield an increase greater than –3.7.

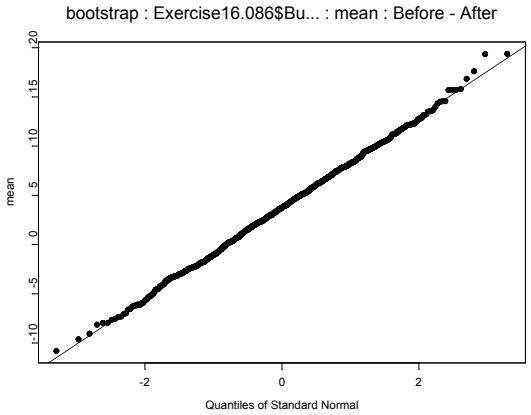
```
Exercise16.086 # variables Burglaries, When (Before and After)
groupMeans(Exercise16.086$Burglaries, Exercise16.086$When)
histogram(~Burglaries | When, data=Exercise16.086)
qqmath(~Burglaries | When, data=Exercise16.086)
x86 = split(Exercise16.086$Burglaries, Exercise16.086$When)
t.test(x86$Before, x86$After, alternative="greater")
perm86a = permutationTestMeans(Exercise16.086, treatment = When,
                              alternative = "greater", seed=0)
perm86a
plot(perm86a)
qqnorm(perm86a)
perm86b = permutationTestMeans(Exercise16.086, treatment = When,
                              alternative = "less", seed=0)
perm86b
```



16.87. (a) The bootstrap distribution appears to be approximately Normal. **(b)** The bootstrap SE is 4.6. A 95% bootstrap *t* confidence interval using the conservative method for the degrees of freedom is (–13.4, 6.1). Using $(n_1 + n_2 - 2)$ degrees of freedom gives a narrower interval (–12.9, 5.6). **(c)** A 95% bootstrap percentile interval is (–12.5, 5.6). This agrees closely with the second interval found in (b), so we conclude that the intervals are reasonably accurate. These intervals include 0, so we conclude that there is not strong evidence (at the 0.05 level) of a difference in the mean monthly burglary counts. The tests in the previous exercise were one-sided tests and showed no strong evidence of a decrease in mean monthly burglaries, so the two sets of conclusions are consistent.

```
boot87 = bootstrap2(Exercise16.086$Burglaries, mean,
                    treatment=Exercise16.086$When, seed=0)
plot(boot87)
qqnorm(boot87)
boot87
limits.t(boot87, df = "smaller")
limits.t(boot87, df = "pooled")
limits.t(boot87) # estimate the degrees of freedom
t.test(x86$Before, x86$After)
limits.percentile(boot87)
```


bootstrap : Exercise16.086$Bu... : mean : Before - After
Observed
Mean
Density
0.0
0.02
0.04
0.06
0.08
-10
-5
0
5
10
15
20
mean


bootstrap : Exercise16.086$Bu... : mean : Before - After
mean
-10
-5
0
5
10
15
20
-2
0
2
Quantiles of Standard Normal