1. (a) One way to produce a data frame directly in R:

```
treeSpecies <- data.frame(
   plotType = c( rep("Unlogged", 12), rep("Logged", 9) ),
   speciesCount = c( 22, 18, 22, 20, 15, 21, 13, 13, 19, 13, 19, 15,
                     17, 4, 18, 14, 18, 15, 15, 10, 12 )
 )
head(treeSpecies)
```

```
  plotType speciesCount
1 Unlogged           22
2 Unlogged           18
3 Unlogged           22
4 Unlogged           20
5 Unlogged           15
6 Unlogged           21
```

We get the observed difference in group sample means with command

```
diff( mean(speciesCount ~ plotType, data=treeSpecies) )
```

```
Unlogged
3.833333
```

To get one bootstrap statistic like it:

```
diff( mean(speciesCount ~ plotType, data=resample( treeSpecies, group = plotType )))
```

```
Unlogged
4.444444
```

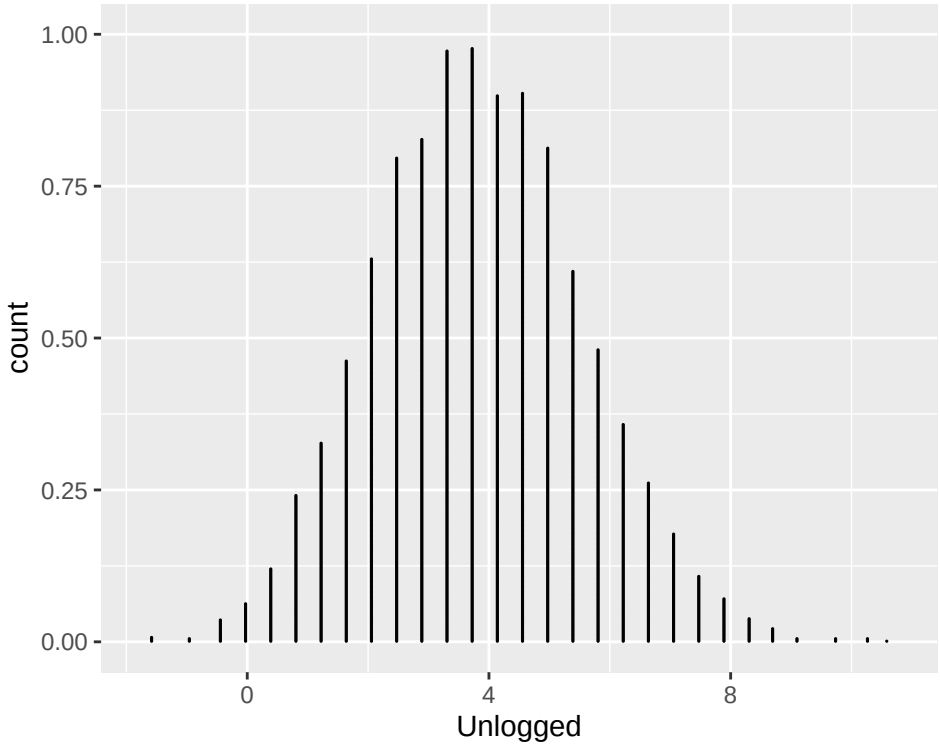
The "`group = plotType`" switch ensures the number of "Unlogged" and "Logged" plots in the randomization sample is always 12 and 9, respectively. This command we repeat many times. I plot the results as a dotplot so you can see how to reduce the size of dots.

```
manyBstrapDiffs <- do(5000) * diff( mean(speciesCount ~ plotType,
                                    data=resample( treeSpecies, group = plotType )))
head(manyBstrapDiffs)
```

```
  Unlogged
1 2.361111
2 7.611111
3 4.166667
4 3.722222
5 7.611111
6 4.083333
```

```
gf_dotplot(~ Unlogged, data=manyBstrapDiffs, dotsize = 0.05 )
```

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



(b)
```
sd <- sd(~ Unlogged, data=manyBstrapDiffs)
sd

[1] 1.711551
```

Bootstrap distributions change from one run to the next, and this standard error changes as a result. Yours should be in the same ballpark: 1.712. The resulting 95% confidence, using the point estimate $\bar{x}_U - \bar{x}_L = 3$ found above, is

```
 3.833 + c(-1,1) * 1.96 * sd

[1] 0.4783603 7.1876397
```

or $(0.478, 7.188)$.

(c) Taking μ_U to be the mean count of tree species on unlogged plots of rain forest, and μ_L the mean count of tree species on plots of rain forest that had been logged 8 years earlier, ours is a confidence interval for $\mu_U - \mu_L$.

(d) There is a nice command in R for doing this:

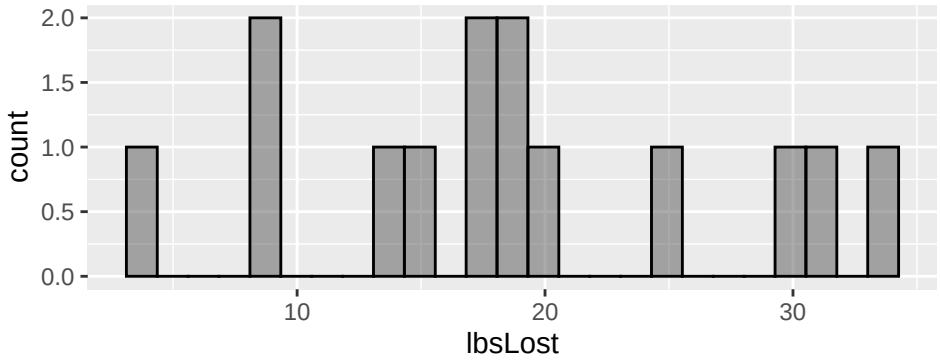
```
qdata(~Unlogged, data=manyBstrapDiffs, c(0.025, 0.975))

       quantile     p
2.5%  0.6388889 0.025
97.5% 7.2777778 0.975
```

Without learning about this command, there are various trial-and-error ways to find these quantiles, or one can turn to StatKey. At any rate, the interval extending from the 0.025-quantile to the 0.975-quantile (which also varies rrom one bootstrap distribution to the next) is $(0.639, 7.278)$. Both of these numbers are higher than the corresponding endpoints of the confidence interval found in part (b). This is possibly explainable by the non-pure normality of the bootstrap distribution found in part (a).

2. (a) The sample size is on the small side ($n = 14$), not in the comfort range of $n \geq 30$. But if the data comes from a *normal* population, any sample size would be large enough. An histogram of the data does little to suggest much of a bell shape but, again, perhaps 14 data points is not enough to bring out any distinct shape. The safe bet is to **not** assume normality.

```
weightLossDat = read.csv("http://scofield.site/teaching/data/csv/navMonk/wtLost.csv")
gf_histogram(~ lbsLost, data = weightLossDat, color="black")
```

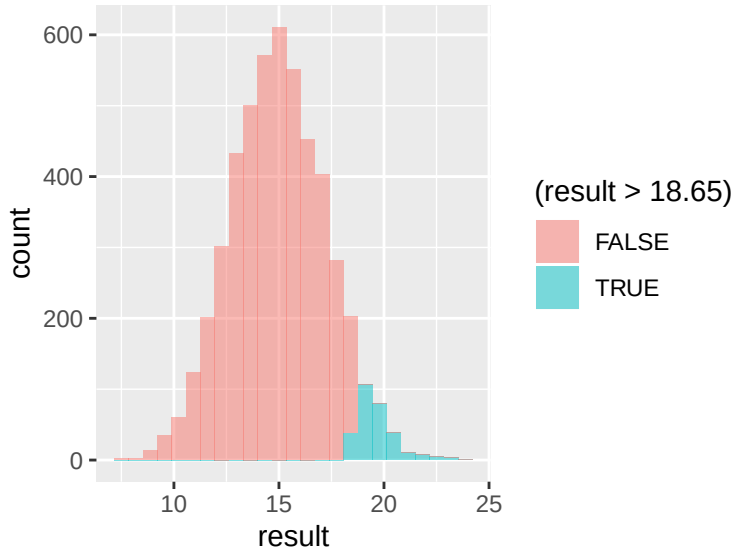


(b) The test statistic is $\bar{x}$, computed below using the `mean()` command to be 18.65.

```
mean(~lbsLost, data=weightLossDat)
```

(c) We do this in a manner similar to bootstrapping, taking care to center the distribution at 15 instead of 18.65:

```
manyXBars = do(5000) * (mean(~lbsLost, data=resample(weightLossDat)) - 3.65)
gf_histogram(~result, data=manyXBars, fill=~(result>18.65), bins=25)
```



(d) One command (different, but no better, than using `subset()` to select out the results which are as extreme as ours, counting the number we selected, and then dividing by the total number of rows) which gives us the approximate P-value

```
prop(~(result >= 18.65), data=manyXBars)

prop_TRUE
   0.0584
```

The evidence supports the alternate hypothesis, but is not strictly less than 0.05, so is not statistically significant at that level.