Problem Set 10: t-test

ADS2

Semester 1 2023/24

We expect this problem set to take around an hour to complete. But professors are sometimes wrong![citation missing]. If this or future problem sets are too long, please let us know, so we can adjust and plan accordingly.

Guinness Quality Control

In the lecture, we heard that the Student's *t*-distribution was devised to provide a statistical framework for assessing the quality of Guinness from taking small samples during the brewing process. The dark colour and characteristic taste of Guinness comes from roasting a portion of the barley, but each pint needs to contain at least 50 g barley. The file 'barley.txt' contains the weight of barley in 50 pints out of the total 2,000 pints brewed in one day.

1. Is the brewery adding enough barley?

SOLUTION

```r
barley<-scan("barley.txt")
t.test(barley, mu = 50)
```

```
##
##      One Sample t-test
##
## data:  barley
## t = -5.7889, df = 49, p-value = 4.941e-07
## alternative hypothesis: true mean is not equal to 50
## 95 percent confidence interval:
##  44.36975 47.27145
## sample estimates:
## mean of x
##   45.8206
```

This can be tested by performing a simple one-sample t-test as described in the lecture. The reported *p*-value = 4.9x 10^{-7} which is below *p*-value = 0.05. We therefore reject the null hypothesis that the sample mean is equal to 50. As the sample mean is 45.8 g, we therefore conclude that the factory is **NOT** adding enough barley.

2. Is the *t*-distribution an appropriate test to answer this question? Do these data meet the assumptions required?

a. The first assumption is that the volumes recorded are continuous and randomly selected. A quick look at the data will reveal that the values are continuous.

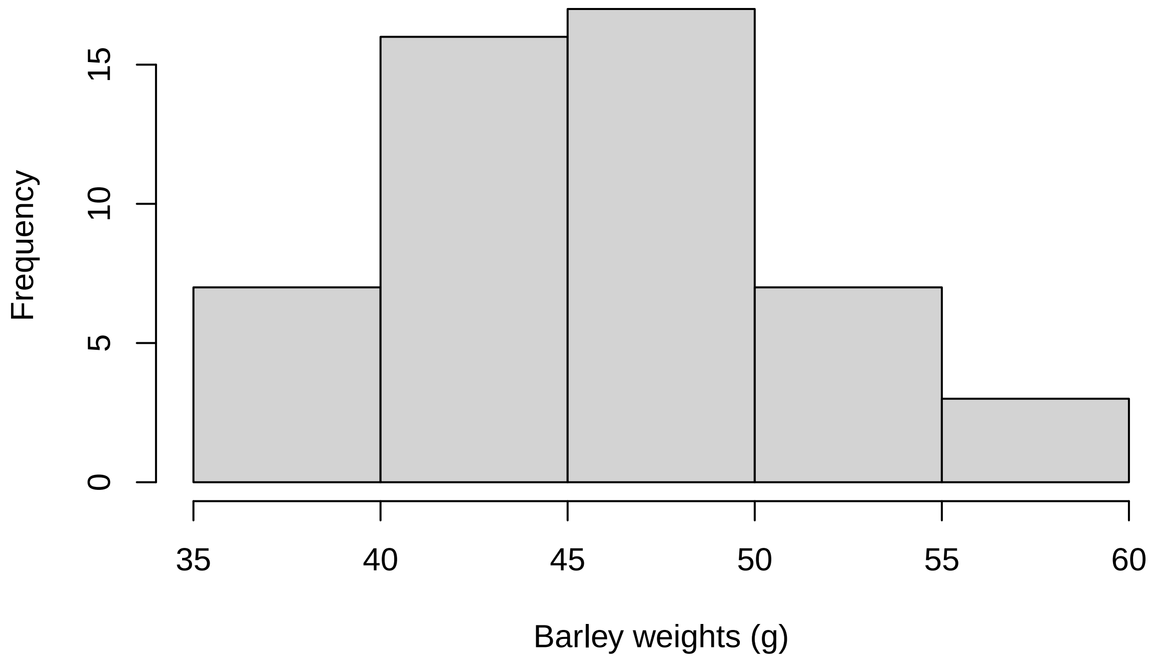
```r
head(barley)
```

```
## [1] 41.03 45.99 50.01 51.44 49.53 47.47
```

We don't know whether the bottles whose volumes were recorded were selected at random. In the real world you would likely know much more about where the data came from and could determine whether it was randomly-sampled. For the purposes of the rest of the problem sheet, we are going to assume that this the data is a random sample from the population of volumes.

b. The second assumption is that the sample is normally distributed. This can be done by plotting a histogram to determine whether the data looks normal enough to me - note here you have to make your own judgement.

```r
hist(barley, xlab = "Barley weights (g)", main = "")
```



This assumption could also be tested formally using the Shapiro-Wilks test, which is a statistical test of normality. In this test, the null hypothesis is that the data under consideration **is** normally distributed.

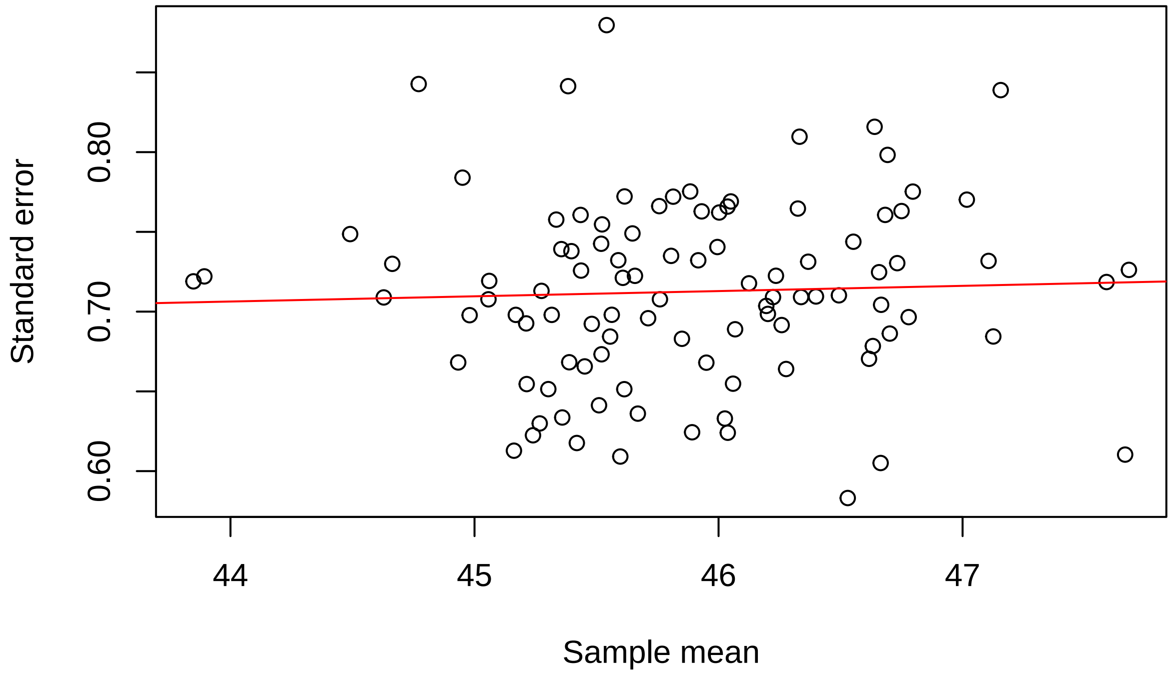
```r
shapiro.test(barley)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  barley
## W = 0.98349, p-value = 0.7051
```

As the reported p-value from this test is greater than 0.05 (the exact value will change for each round of sampling) we can confirm that this distribution of sampling means is normal.

c. The third assumption is that the mean and standard error are independent. I test this by performing random sampling, where I record the standard error as well as the mean. A plot of these two values against each other suggests that there is little relationship between them, and this can be confirmed by performing a linear regression. The estimated value of the coefficient describing the relationship between the sampled standard errors and sample means is small (exactly how small will change during each round of sampling) and its associated p-value is large (often > 0.05) suggesting that there is little statistical support for this relationship

```
sampling_errors<-vector()
sampling_means<-vector()
for (replicate in 1:100){
    barley_sample<-sample(barley, size = length(barley), replace = TRUE)
    standard_error<-sd(barley_sample)/sqrt(length(barley_sample))
    sampling_errors<-c(sampling_errors, standard_error)
    sampling_means<-c(sampling_means, mean(barley_sample))
}
plot(sampling_means, sampling_errors, xlab = "Sample mean", ylab = "Standard error")
lmfit<-lm(sampling_errors~sampling_means)
abline(lmfit, col = 'red')
```



```
summary(lmfit)
```

```
##
## Call:
## lm(formula = sampling_errors ~ sampling_means)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max 
## -0.131477 -0.036914  0.000484  0.038139  0.168240 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.562074   0.361548   1.555    0.123
## sampling_means 0.003278   0.007880   0.416    0.678
## 
## Residual standard error: 0.05749 on 98 degrees of freedom
## Multiple R-squared:  0.001763,   Adjusted R-squared:  -0.008423 
## F-statistic: 0.1731 on 1 and 98 DF,  p-value: 0.6783
```

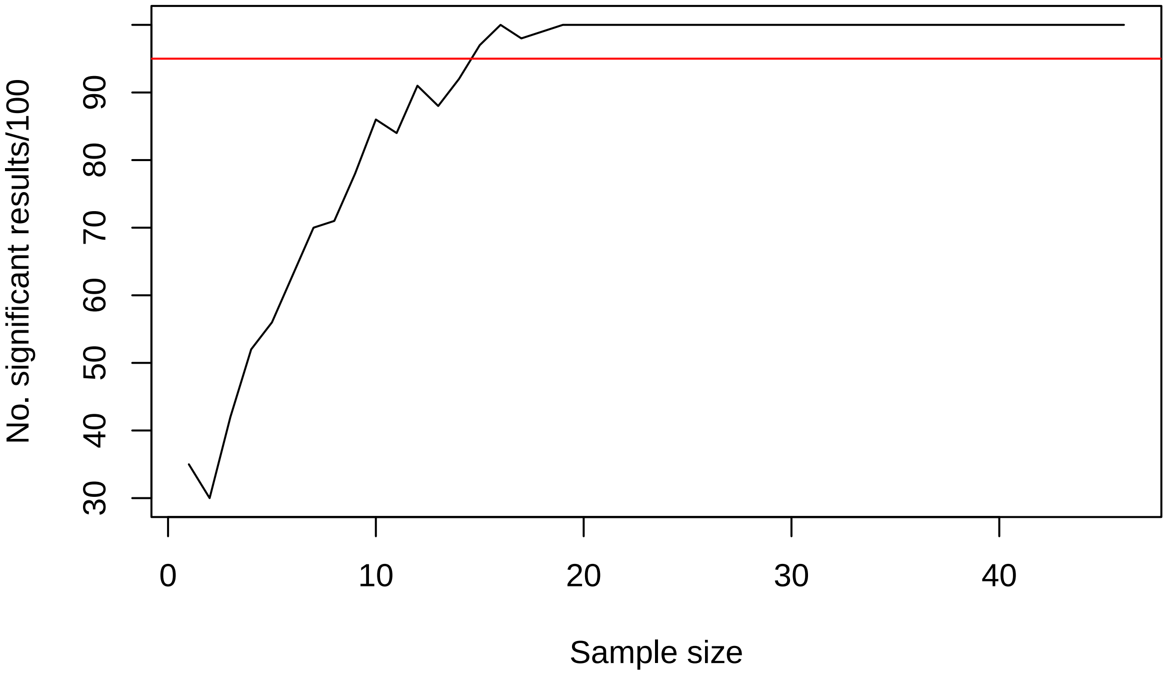
3. The t-distribution is most useful when there are small sample sizes, but how small is small? Can you run a simulation to determine the power of the t-distribution as the number of pints sampled decreases? What is the minimum number of pints we need to sample to have 95% confidence that we could detect any real difference from the required 50 g?

As above, the power of the t-distribution can be explored through sampling. I run 100 replicates for sample sizes between 5 and 50. For each individual sample size, I have recorded the number of replicates for which I received a significant result (as simply judged by $p <= 0.05$) and plotted this against the sample size.

```r
sig_results<-vector()

for (sample_size in 5:50){
    found_sigs<-0
    for (replicate in 1:100){
        barley_sample<-sample(barley, size = sample_size, replace = FALSE)
        p_value<-t.test(barley_sample, mu = 50)$p.value
        if (p_value <= 0.05){
            found_sigs = found_sigs + 1
        }
    }
    sig_results<-c(sig_results, found_sigs)
}

plot(sig_results, type = 'l', ylab = "No. significant results/100", xlab = "Sample size")
abline(h = 95, col = 'red')
```



The minimum sample size at which at least 95 of the 100 replicates returned a p-value <= 0.05 is determined as shown below:

```
min(which(sig_results >= 95))
```

```
## [1] 15
```