Solutions to computer class 2

Richard Wilkinson

2 March 2018

Solution 1

Suppose $X_1, \dots, X_{10} \sim N(\mu, \sigma^2)$, and that we observe data

$$\{2.561, -0.328, 2.607, 3.466, 2.012, 1.293, -2.301, 1.914, 5.779, 1.369\}$$

- Assume that $\sigma = 2$. Use a Monte Carlo test to test the null hypothesis

  $$H_0 : \mu = 1$$

  against the alternative

  $$H_1 : \mu \neq 1.$$

- Now assume that $\mu = 1$. Use a Monte Carlo test to test the null hypothesis

  $$H_0 : \sigma^2 = 1$$

  against the alternative

  $$H_1 : \sigma^2 > 1.$$

Solution

Let's enter the data

```
X <- c(2.561, -0.328,  2.607,  3.466,  2.012,  1.293, -2.301,  1.914,  5.779,  1.369)
```

The basic proceedure is to simulate values of $X_1, \dots, X_{10}$ under the null distribution, and to look at the simulated distribution of the test statistics, and to see where the observed value lies in this distribution. For the first problem, it would make sense to use a test statistic of the form

$$T(X) = \frac{\bar{X} - 1}{2}$$

```
simH0 <- function(){
  X <- rnorm(10, 1, 2) # simulate under the null
  Tx <- (mean(X) - 1)/2
  return(Tx)
}
```

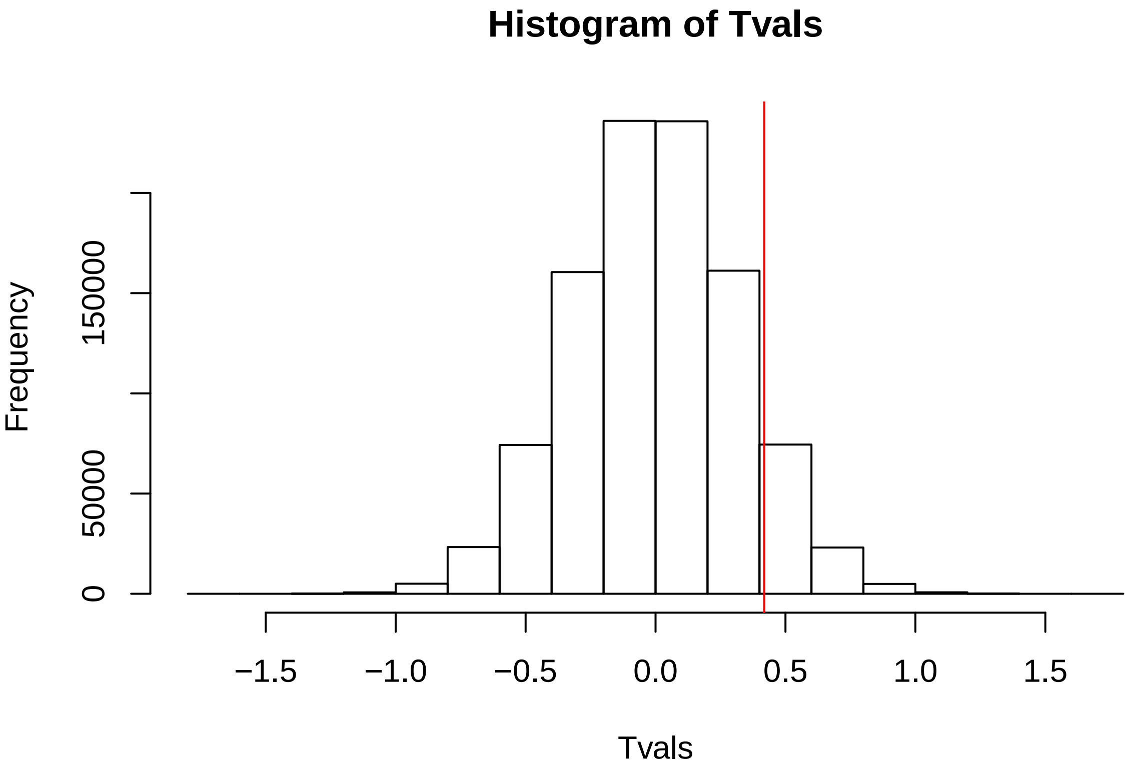
The observed value of the test statistic is

```
(Tobs <- (mean(X) - 1)/2)
```

```
## [1] 0.4186
```

so let's see how this compares to the simulated distribution.

```
Tvals <- replicate(10^6, simH0())
hist(Tvals)
abline(v=Tobs, col=2)
```



I've drawn a histogram and put a vertical line where the observed value of the test statistic is. By eye, this doesn't look particularly extreme, but we can check by calculating the p-value

```
mean(abs(Tvals) > Tobs)
```

```
## [1] 0.185799
```

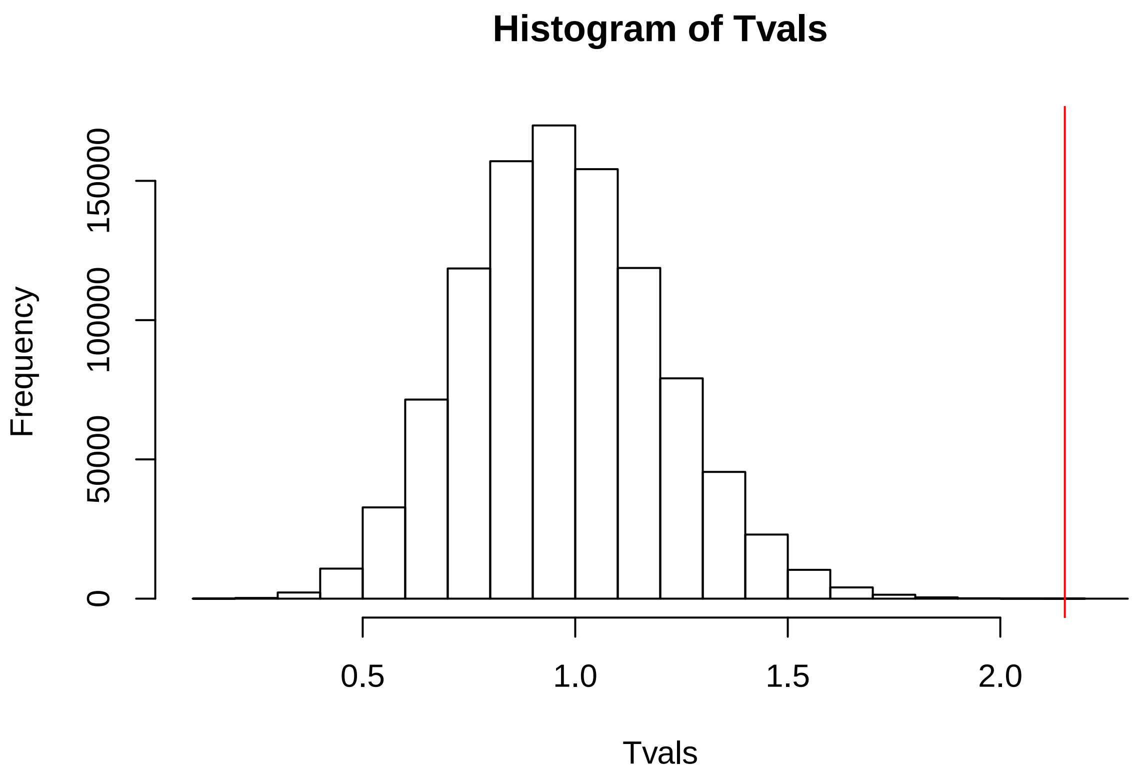
Note that I've taken the absolute value of the T values.

For the second problem where we want to test a hypothesis about the variance, we can use the test statistic

$$T(X) = s^2 = \frac{1}{n-1}\sum(X_i - \bar{X})^2$$

The classical test tells us that $(n-1)s$ has a χ^2 distribution under H_0, but we don't need to use that here. We can just argue that T is an estimator of σ^2, and so if the observed value of T is extreme compared to the distribution of T under H_0, this would be grounds to reject H_0.

```
simT <- function(){
  X <- rnorm(10, 1, 1) # simulate under the null
  return(sd(X))
}

Tvals <- replicate(10^6, simT())
hist(Tvals)
abline(v=sd(X), col=2)
```



This time T_{obs} does look extreme compared to the distribution of T. The p-value it gives us in this case is

```
(sum(Tvals>sd(X))+1)/(10^6+1)
```

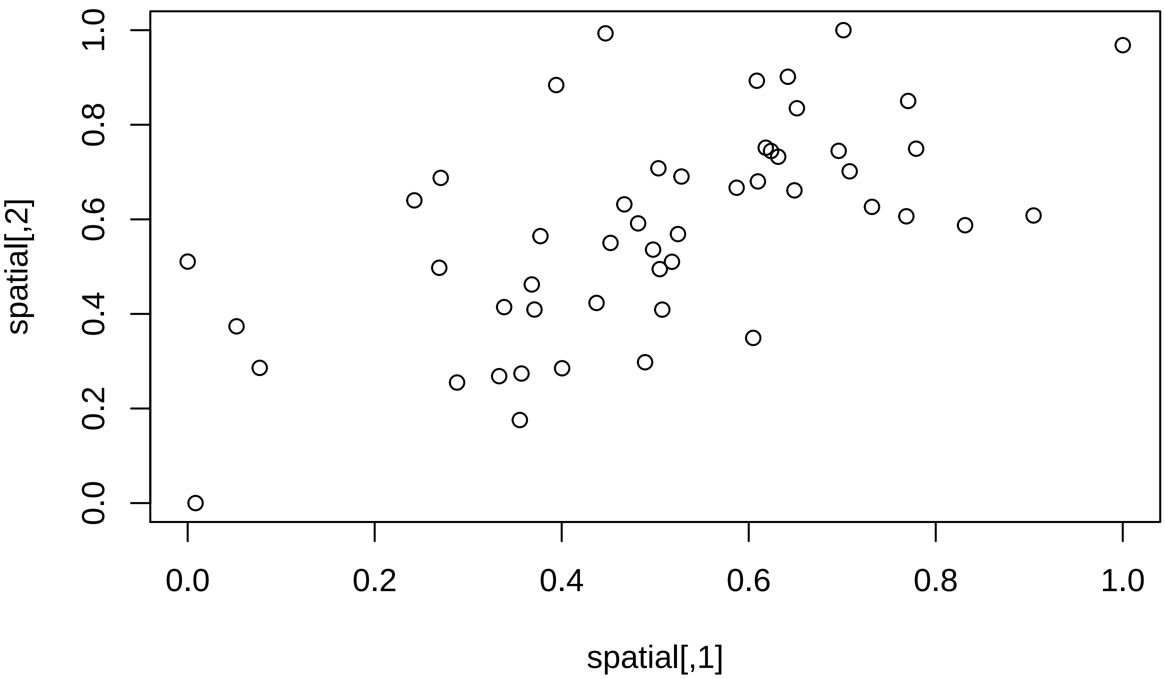
```
## [1] 3.999996e-06
```

So there is strong evidence to reject H_0.

Solution 2

Locations of 50 points are stored in the vector `spatial`. Check that you can plot the data with the command

```
load(file = 'Class1data.Rdata')
plot(spatial)
```

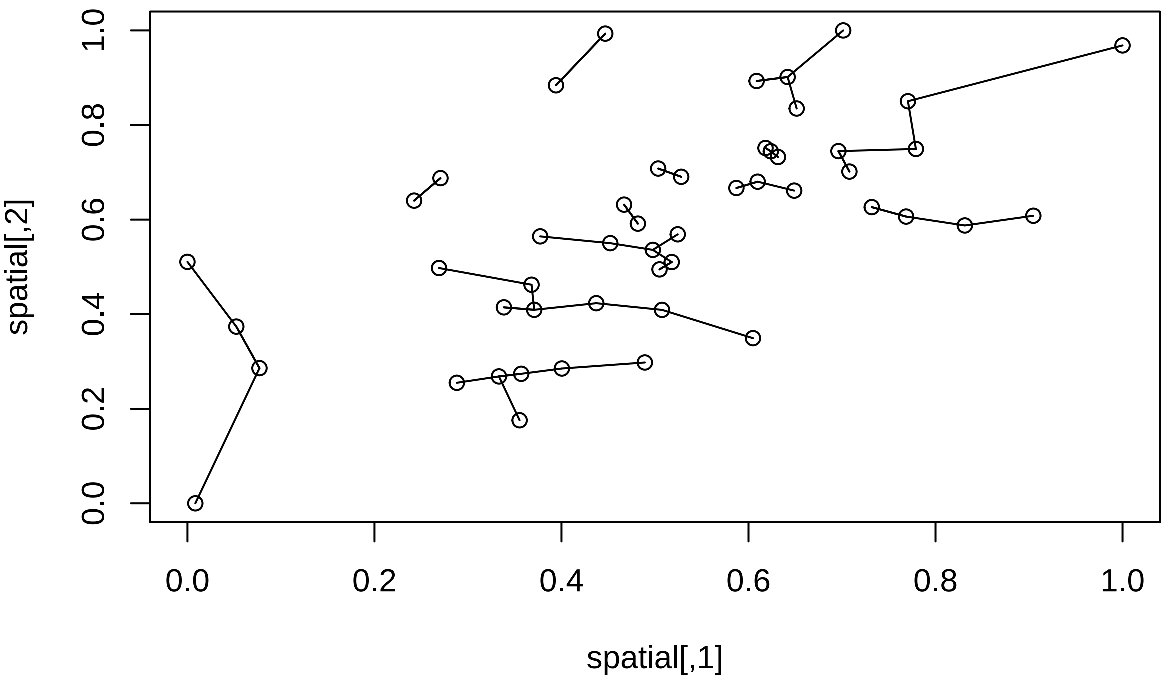


Use a Monte Carlo test to test the null hypothesis that the distribution of points is uniform over the unit square.

Hint: The hardest part to this problem is computing the nearest neighbour distances for any particular pattern of points. You should think about how to do this, but in case you are stuck, one approach is given below.

We can start by giving a visual illustration of our test statistic:

```
n<-length(spatial[,1])
mx1<-matrix(spatial[,1],n,n,byrow=F)
mx2<-matrix(spatial[,2],n,n,byrow=F)
distances<-((mx1-t(mx1))^2+(mx2-t(mx2))^2)^0.5
distances<-distances+diag(100,n,n)

plot(spatial)
for(i in 1:50){
  index<-which(distances[i,]==min(distances[i,]))
  lines(c(spatial[i,1],spatial[index,1]),c(spatial[i,2],spatial[index,2]))
}
```



```
1/mean(apply(distances,2,min))
```

```
## [1] 15.1431
```

The hardest part of this question is writing a function which, given observations x find the test statistic. This can be done with the following code:

```
nnsum<-function(x){
    n<-length(x[,1])
    mx1<-matrix(x[,1],n,n,byrow=F)
    mx2<-matrix(x[,2],n,n,byrow=F)
    distances<-((mx1-t(mx1))^2+(mx2-t(mx2))^2)^0.5
    distances<-distances+diag(100,n,n) # so that we don't pick the distance
    # to ourselves
    return(mean(apply(distances,2,min)))
}

1/nnsum(spatial)
```

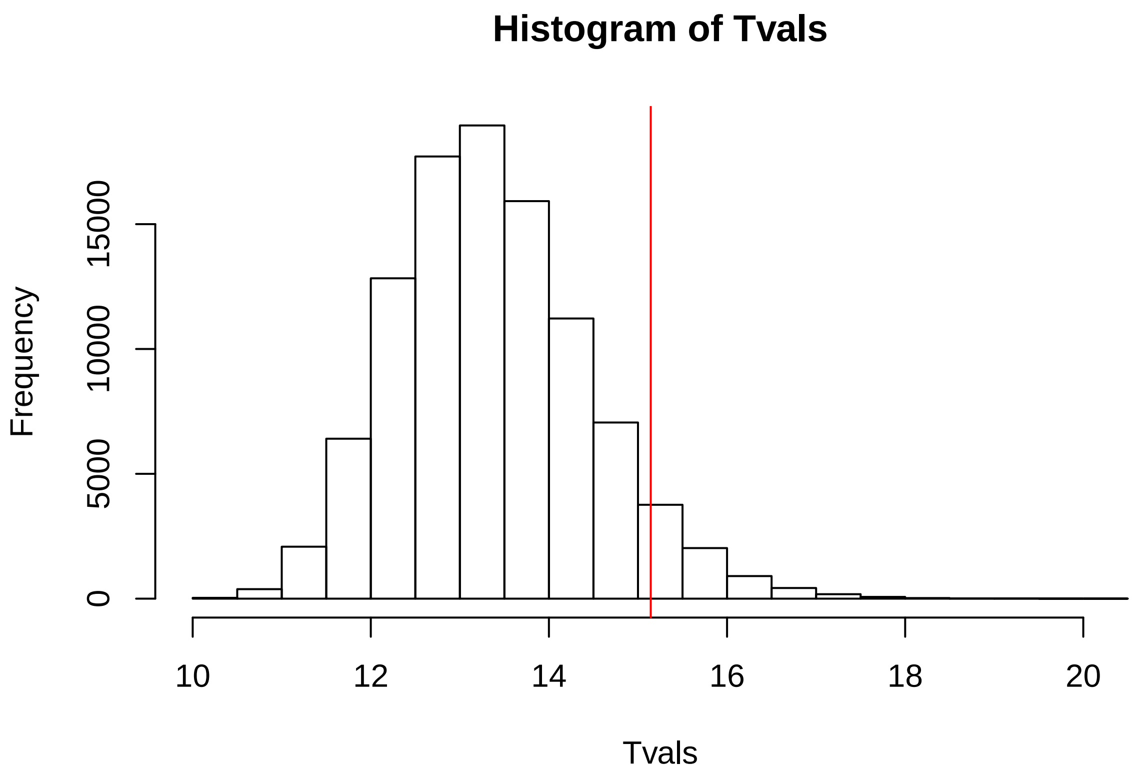
```
## [1] 15.1431
```

We now need to write code to simulate from the distribution of the test statistic under the null hypothesis.

```
simT <- function(){
  rx<-matrix(runif(100),nrow=50,ncol=2)
  1/nnsum(rx)
}
```

and we can do the Monte Carlo test with the commands

```
Tvals <- replicate(10^5, simT())
hist(Tvals)
abline(v=1/nnsum(spatial), col=2)
```



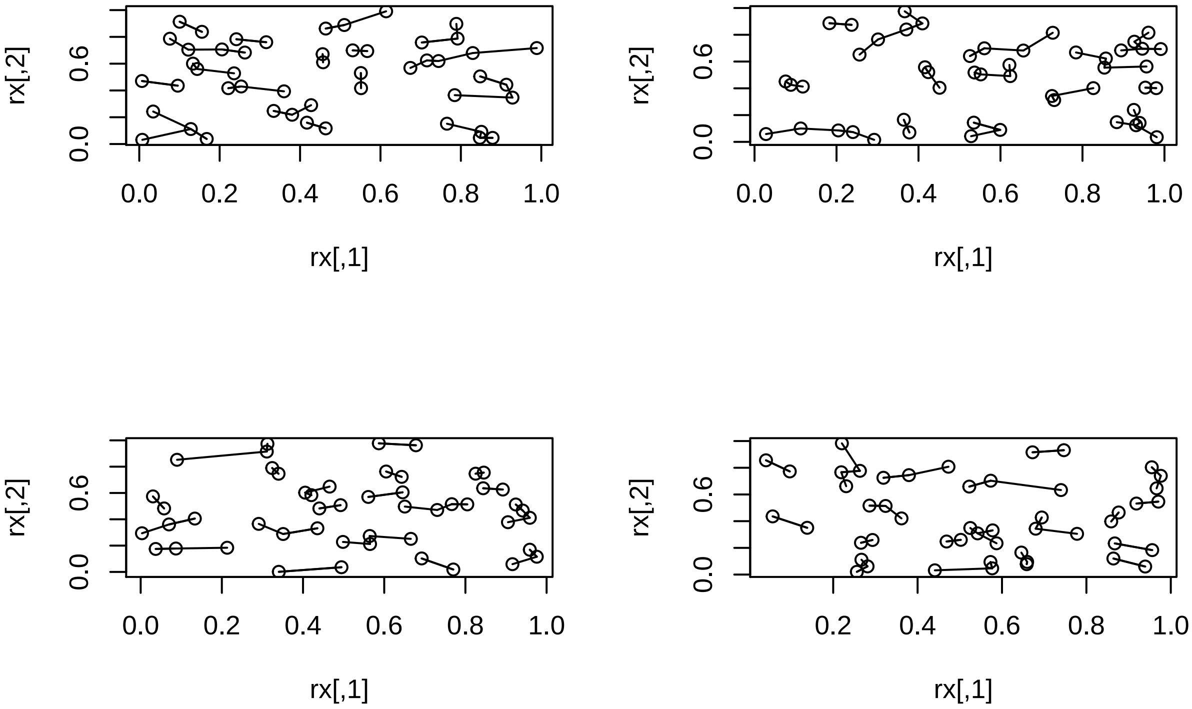
```
mean(Tvals>(1/nnsum(spatial)))
```

```
## [1] 0.06059
```

I've used a smaller number of replicates here, as it takes about 10 minutes to do 10^6 replicates on my machine. The p-value obtained suggests there is not evidence to reject H_0 at the 5% level.

We can visualise the test statistics for a few samples using

```
par(mfrow=c(2,2))
for(j in 1:4){

    rx<-matrix(runif(100),nrow=50,ncol=2)

    n<-length(rx[,1])
    mx1<-matrix(rx[,1],n,n,byrow=F)
    mx2<-matrix(rx[,2],n,n,byrow=F)
        distances<-((mx1-t(mx1))^2+(mx2-t(mx2))^2)^0.5
        distances<-distances+diag(100,n,n)

    plot(rx)


    for(i in 1:50){
        index<-which(distances[i,]==min(distances[i,]))
    lines(c(rx[i,1],rx[index,1]),c(rx[i,2],rx[index,2]))
    }
}
```



Solution 3

```
library(MASS)
attach(birthwt)
bwt.smoke <- bwt[smoke==1]
bwt.nonsmoke <- bwt[smoke==0]
```

To perform a permutation/randomization test we should randomly reallocate each data point into a simulate smoker or non-smoker group. We then calculate the test statistic.

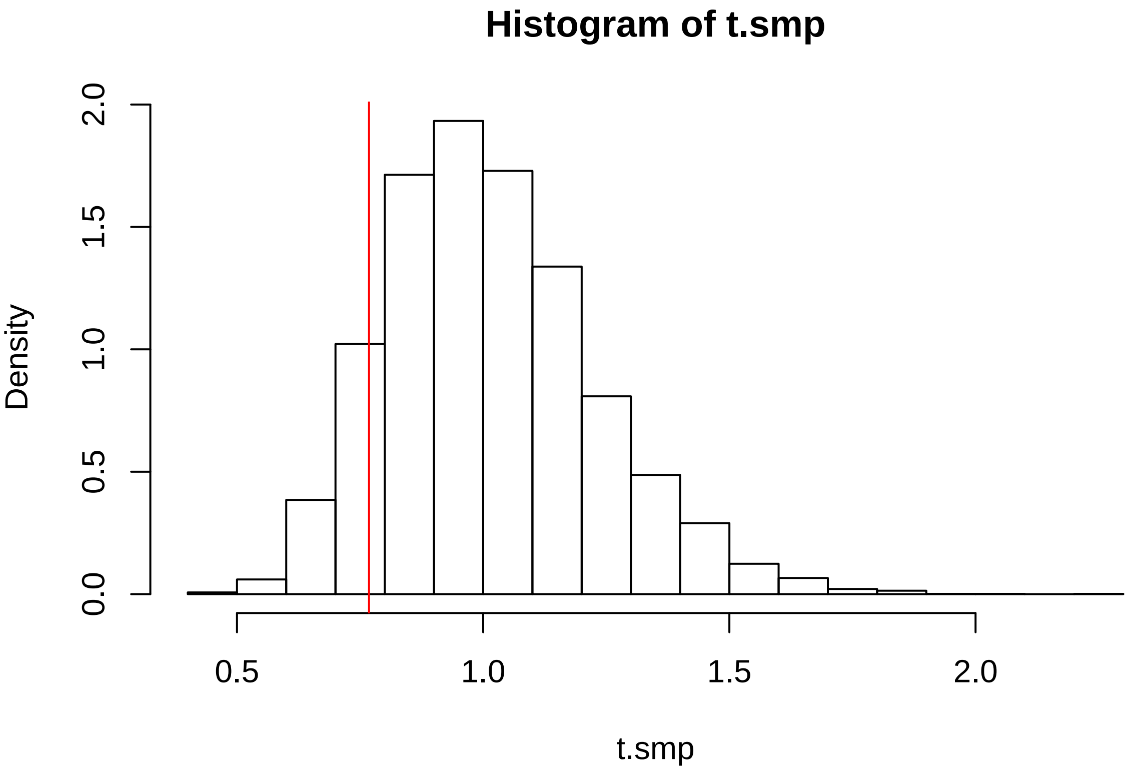
```
permute <- function(){
  n <- length(bwt)
  bwt.smokeindices <- sample(1:n, length(bwt.smoke), replace=FALSE)
  bwt.smoke.smp <- bwt[bwt.smokeindices]
  bwt.nonsmoke.smp <- bwt[-bwt.smokeindices]
  t.smp <- var(bwt.smoke.smp)/var(bwt.nonsmoke.smp)
  return(t.smp)
}

t.smp <- replicate(10^4, permute())
t.obs <- var(bwt.smoke)/var(bwt.nonsmoke)
```

We are asked to do a one-sided test. Under H_0 we expect $T \approx 1$, but under H_1 we expect to observe small values of T.

```
mean(t.obs>t.smp)
```

```
## [1] 0.1095
```

```
hist(t.smp, probability=T)
abline(v= t.obs, col=2)
```



So there is very weak evidence against H_0, but nothing to cause us to reject the null. Note we can perform the classical F-test using

```
var.test(bwt.smoke,bwt.nonsmoke, alternative = 'less')
```

```
##
##  F test to compare two variances
##
## data:  bwt.smoke and bwt.nonsmoke
## F = 0.76809, num df = 73, denom df = 114, p-value = 0.1127
## alternative hypothesis: true ratio of variances is less than 1
## 95 percent confidence interval:
##  0.000000 1.099214
## sample estimates:
## ratio of variances
##          0.7680924
```

Part ii) Now assume that both distributions are Gaussian

We have to start by fitting a Gaussian distribution to both datasets. Under H_0 the variance of both distributions are the same, but each distribution is allowed a different mean. Let's estimate these quantities, using a pooled estimator for the variance, and the individual samples for the means.

```
(mu.bwt.smoke <- mean(bwt.smoke))
```

```
## [1] 2771.919
```

```
(mu.bwt.nonsmoke <- mean(bwt.nonsmoke))
```

```
## [1] 3055.696
```

```
(sd.shared <- sd(c(bwt.smoke-mu.bwt.smoke, bwt.nonsmoke - mu.bwt.nonsmoke)))
```

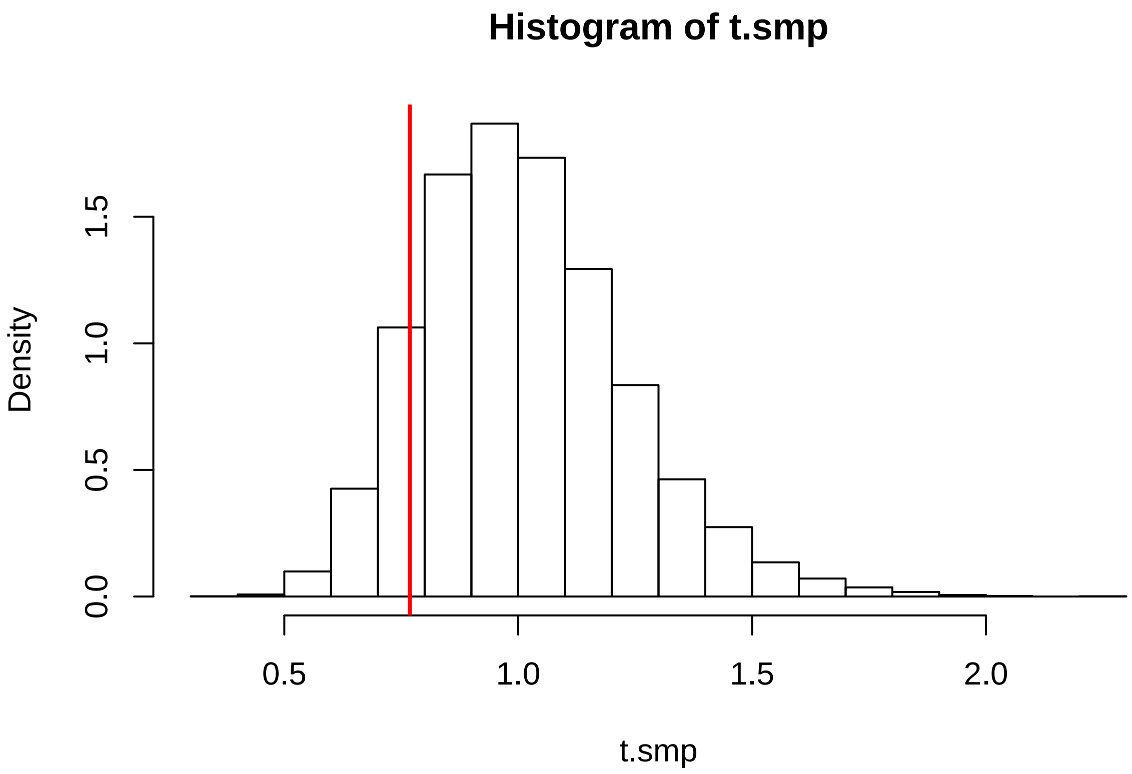
```
## [1] 715.8677
```

The Monte Carlo test then simulates data from the Gaussian distributions with these parameters, and recalculates the test statistic. Let's create a function to sample a dataset and calculate the test statistic.

```
MCtest <- function(){
  bwt.smoke.smp <- rnorm(length(bwt.smoke), mean=mu.bwt.smoke, sd=sd.shared)
  bwt.nonsmoke.smp <- rnorm(length(bwt.nonsmoke), mean=mu.bwt.nonsmoke, sd=sd.shared)
  t.smp <- var(bwt.smoke.smp)/var(bwt.nonsmoke.smp)
  return(t.smp)
}
```

The Monte Carlo test consists of running this command a large numnber of times to create the sampling distribution of the test statistic under H_0.

```
t.smp <- replicate(10^4, MCtest())
hist(t.smp, probability = TRUE)
abline(v = t.obs, col=2, lwd=2)
```



```
mean(t.smp<t.obs)
```

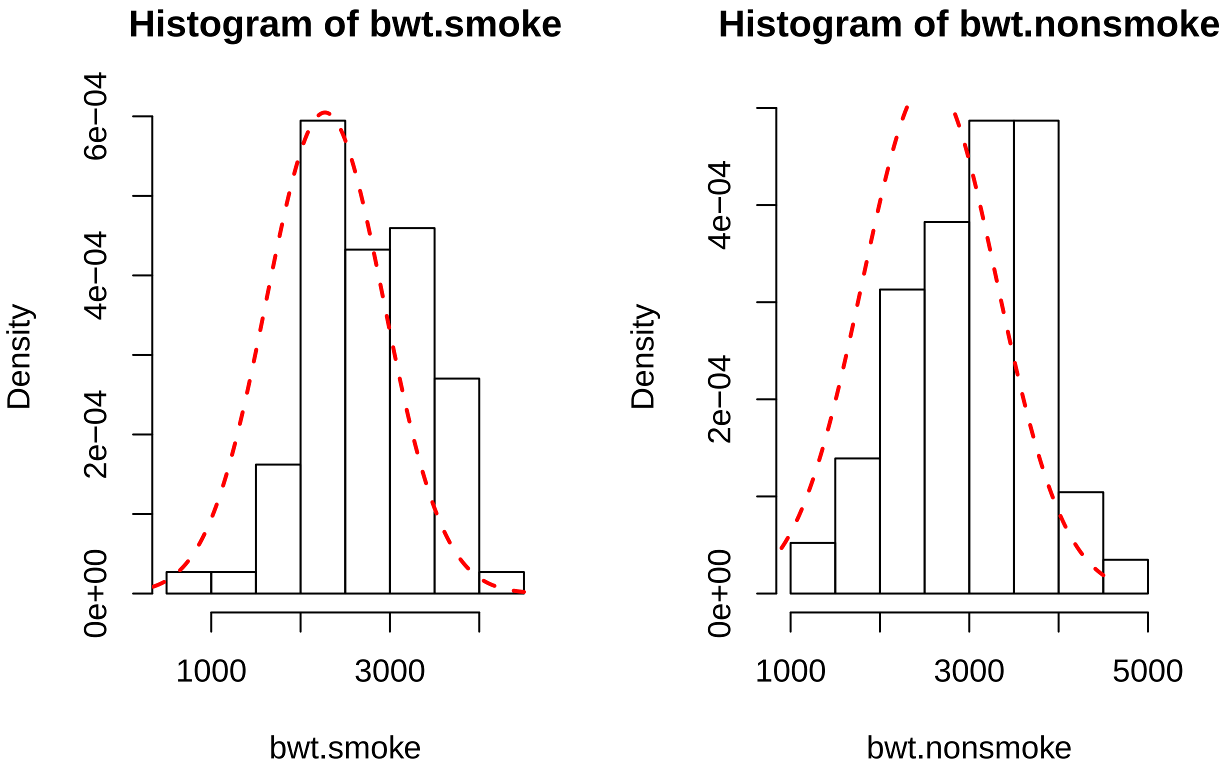
```
## [1] 0.1172
```

This gives us very similar results to the previous question (as we'd expect). Note that the Monte Carlo test can be seen as an implementation of the classical test using simulation.

The easiest way to check the Gaussian assumption is to plot histograms for both datasets.

```
par(mfrow=c(1,2))
x <- seq(500,5000)
hist(bwt.smoke, probability = T)
```

```r
lines(dnorm(x, mean(bwt.smoke), sd(bwt.smoke)), lty=2, col=2, lwd=2)
hist(bwt.nonsmoke, probability = T)
lines(dnorm(x, mean(bwt.nonsmoke), sd(bwt.nonsmoke)), lty=2, col=2, lwd=2)
```



So in both cases there is some evidence that the distributions are skewed.

Somewhat a matter of taste and expert judgement. The parametric Monte Carlo test assumes Gaussian data, which as we've seen, is not entirely accurate in this case. However, the departure from Gaussianity is not far from being valid, and so the test is probably saying something sensible. The randomization test assumes the data points are selected at random from the population and that the results can be generalised to the entire population. On balance, I think I'd prefer the randomization test here, but using either test would be defensible.

```r
mean(t.smp>t.obs)
```

```
## [1] 0.01
```

```r
t.test(bwt.smoke, bwt.nonsmoke)
```

```
##
##  Welch Two Sample t-test
##
## data:  bwt.smoke and bwt.nonsmoke
## t = -2.7299, df = 170.1, p-value = 0.007003
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -488.97860  -78.57486
## sample estimates:
## mean of x mean of y
##  2771.919  3055.696
```

–>

Part iii) Use a randomization test…

First subtract the hypothesised mean to get a dataset that has mean 0 under H_0

```
bwt.smoke_zeromean <- bwt.smoke -2600 # under H_0 these should have mean 0.
t.obs <- abs(mean(bwt.smoke_zeromean))
```

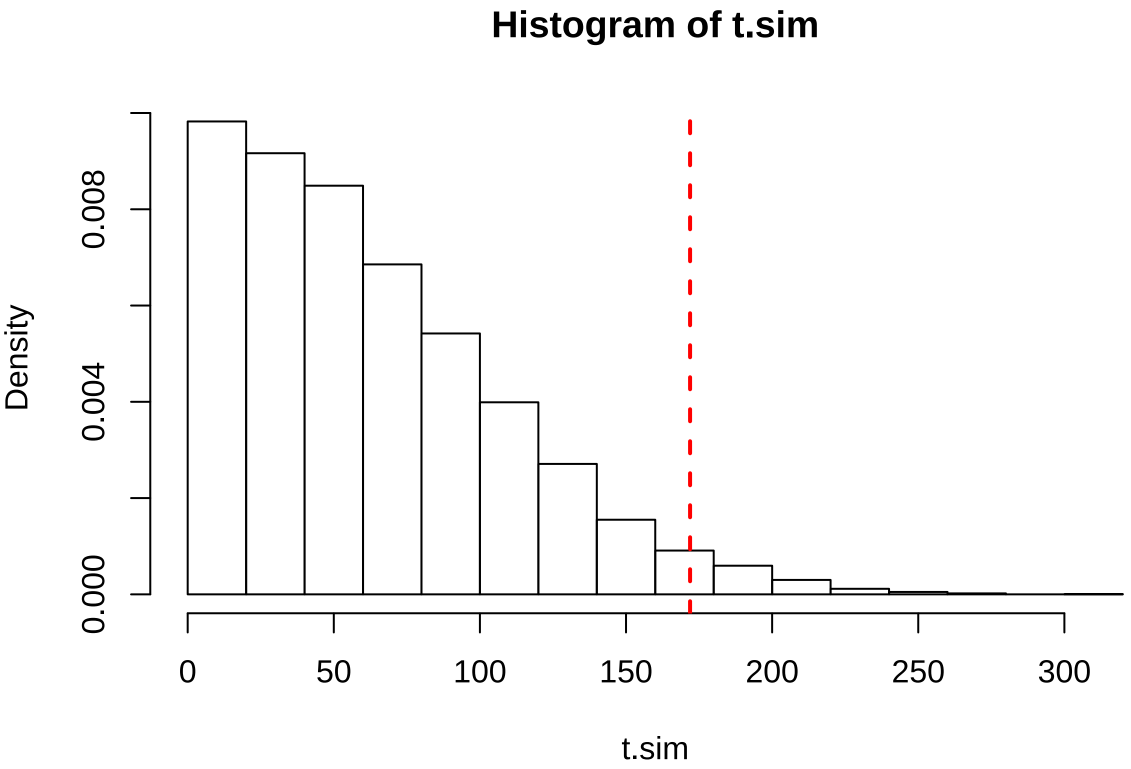
Then the one-sample randomisation test simply flips the sign (or not) of the zero-mean data at random.

```
OneSampleRandomization <- function(){
  abs(mean(bwt.smoke_zeromean*sample(c(1,-1), length(bwt.smoke_zeromean), replace=TRUE)))
}
```

```
t.sim <- replicate(10^4, OneSampleRandomization())
mean(t.sim>t.obs)
```

```
## [1] 0.0281
```

```
hist(t.sim, probability = T)
abline(v=t.obs, col=2, lwd=2, lty=2)
```



```
t.test(bwt.smoke, mu = 2600)
```

```
##
##  One Sample t-test
##
## data:  bwt.smoke
## t = 2.242, df = 73, p-value = 0.028
## alternative hypothesis: true mean is not equal to 2600
## 95 percent confidence interval:
##  2619.094 2924.744
## sample estimates:
## mean of x
## 2771.919
```

So there is reasonable evidence against H_0.