Wald-Wolfowitz Runs Test in R

Ryan Honea

9/22/2017

Introduction to the Wald-Wolfowitz Runs Test

Suppose one wants to test the randomness of a sequence of 2 different symbols. For example, you suspect that your professor has a pattern that is being used when creating their True-False test, and that they aren't randomly deciding when they will have a true value, and when they will have a false value. For example, say you observe on a test the following pattern

$$T \quad T \quad F \quad F \quad T \quad T \quad F \quad F \quad T \quad T$$

which appears to be far from randomly distributed.

Abraham Wald and Jacob Wolfowitz devised a method to test whether or not a series of sequences (or runs) are truly randomly distributed. The test uses basic counting principles to calculate the probability of such a series occurring. Where r is the number of runs, the below formulas determine those probabalities for an odd number of runs and an even number of runs.

$$P(r = 2k+1) = \frac{\binom{n-1}{k-1}\binom{m-1}{k} + \binom{n-1}{k}\binom{m-1}{k-1}}{\binom{m+n}{n}} \qquad P(r = 2k) = \frac{2\binom{n-1}{k-1}\binom{m-1}{k-1}}{\binom{m+n}{n}}$$

Crafting the Test in R

The first step in running this test given any data set then is to determine the number of runs which is done by the function below. Note that the sum of runs starts at 1 because a sequence that is all the same is still a single run.

```
num_runs <- function(seq) {
  sum = 1
  for (i in 1:(length(seq) - 1)) {
    if (seq[i] != seq[i+1]) sum = sum + 1
  }
  return(sum)
}
```

Next, one must determine whether or not there is an even or odd number of runs which can be done with a simple if statement One also needs to know n and m, i.e. the number of occurrences of each variable. Using R, this is easy. This allows us to calculate the p-value for even or odd numbers of runs.

```
even_calc <- function(runs, n, m) {
  k = runs/2
  numerator = choose(n-1, k-1)*choose(m-1,k-1)
  denominator = choose(m + n, n)
  return(k*numerator/denominator)
}
odd_calc <- function(runs, n, m) {
  k = runs%/%2
  numerator = choose(n-1,k)*choose(m-1,k-1) + choose(n-1,k-1)*choose(m-1,k)
  denominator = choose(m + n, n)
```

```
  return(numerator/denominator)
}
```

Below is the full function that utilizes all information from above and calculates a p-value.

```
ww.test <- function(seq) {
  seq = factor(seq)
  levels(seq) <- c(0,1)
  runs = num_runs(seq)
  n = length(seq[seq == 0])
  m = length(seq[seq == 1])
  if (runs %% 2 == 0) {
    p.value = even_calc(runs, n, m)
  } else {
    p.value = odd_calc(runs, n, m)
  }
  return(p.value)
}
```

Testing the function

To test this function, we will consider thirty observations from the standard normal distribution and test their randomness. We will do this with the Wald-Wolfowitz test by simulating thirty observations and marking them as a - if they are less than the mean and a + if they are greater than the mean. If they are equal to the mean, they will be removed. This will be repeated one-hundred times,

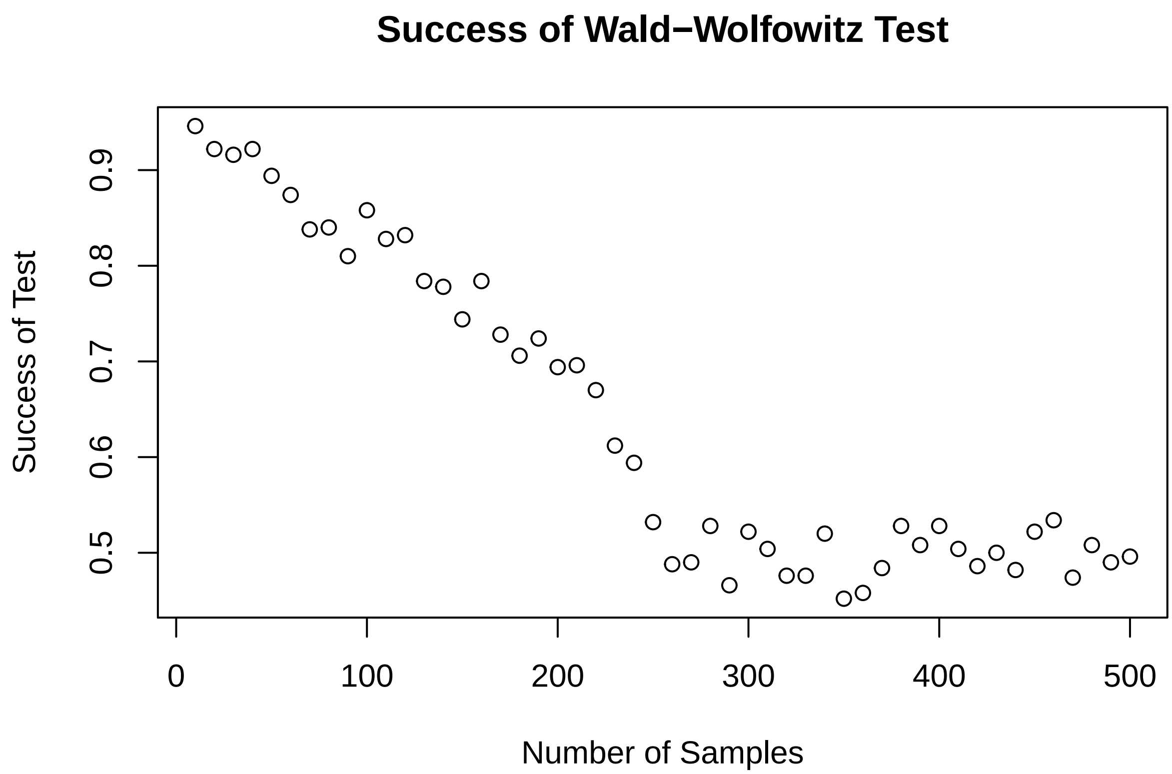
```
rejected = 0
sims = 100
for (i in 1:sims) {
  sim <- rnorm(30)
  sim[sim < 0] = '-'
  sim[sim > 0] = '+'
  sim = sim[sim != 0]
  if (ww.test(sim) <= 0.05) rejected = rejected + 1
}
cat("This test was successful",(sims-rejected)/sims,"percent of the time")
```

```
## This test was successful 0.91 percent of the time
```

We might also be interested to see how this test performs with increasing numbers of observations from the standard normal. Below, we create a plot of that information.

```
n = 10
proportions <- c()
while (n <= 500) {
  rejected = 0
  sims = 500
  for (i in 1:sims) {
    sim <- rnorm(n)
    sim[sim < 0] = '-'
    sim[sim > 0] = '+'
    sim = sim[sim != 0]
    if (ww.test(sim) <= 0.05) rejected = rejected + 1
  }
  proportions <- c(proportions, (sims - rejected)/sims)
```

```r
  n = n + 10
}
plot(seq(10,500,by = 10), proportions,
     xlab = "Number of Samples", ylab = "Success of Test",
     main = "Success of Wald-Wolfowitz Test")
```



And so contrary to what we are used to, this test starts to become less effective as the sample size increases, and appears to approach a fifty percent success rate.