Joint Distribution for Number of Crossings and Longest Run in Independent Bernoulli observations

by Tore Wentzel-Larsen and Jacob Anhøj

Abstract An abstract of less than 150 words.

Introduction

The setting is defined by a number of independent observations from a Bernoulli distribution with the same success probability. In statistical process control, our main intended application, this may be the useful observations in a runs plot, recording values above and below the median, disregarding any observations equal to the median (Anhøj, 2015). While the number of crossings follows a binomial distribution in the symmetric case (success probability 1/2), no closed form distribution is known for the longest run. The distribution of the longest run has been investigated in a number of articles, including Schilling (2012), and Fazekas et al. (2010) that in fact gives recursion formulas, and approximations have been given. However, what is needed in applications is the joint distribution of these two variables, for which the authors are not aware of exact results. Our primary aim is to present an iterative procedure for computing this distribution, in principle for an arbitrary number of observations.

The iterative procedure

In n independent Bernoulli observations with success probability p and failure probability q=1-p, values are denoted by 1 (success, with probability p) or 0. A crossing consists of two consecutive different values, and a run of length l consists of l successive observations, delimited by a crossing or the first or last observation. The possible values of the number C of crossings are $c = 0, \ldots, n-1$ and the possible values for the length L of the longest run are $l = 1, \ldots, n$. The joint probabilities of L and C for given n are denoted by $P_n(L = l, C = c)$.

The iterative procedure involves conditioning on the first observation denoted by S, with values 1 for success (probability p) and 0 for failure (probability q). What are computed in the iterative procedure are the conditional probabilities

$$P_n(L = l, C = c \mid S = 1), P_n(L = l, C = c \mid S = 0)$$

This conditioning on the first observation is an essential part of the procedure. One way to see that this is reasonable is to consider the case when p is close to 1. Then most observations are successes, most runs are success runs and the conditional joint distribution of runs and crossings is quite different dependent on the first observation. It is sufficient to be able to compute these conditional distributions, because the unconditional joint distribution is

$$P_n(L = l, C = c) = P_n(L = l, C = c \mid S = 1) \cdot p + P_n(L = l, C = c \mid S = 0) \cdot q$$

For the iterative procedure to work it is also necessary to take another variable into account, the first crossing. More precisely, we denote the end position of the first crossing by F, with values $f = 2, \ldots, n$. An additional value $f = 1$ denotes, by convention, the case of no crossing. The joint probabilities for C and L conditional on S are partioned by further conditioning on F as detailed below. First we present the starting point of the iterative procedure, the conditional probabilities in the rather redundant case with only one observation.

The conditional probabilities with one observation

If n=1, 0 is the only possible value of C and 1 the only possible value of L, therefore $P_1(C = 0, L = 1 \mid S = 1) = P_1(C = 0, L = 1 \mid S = 0) = 1$. In this case the joint distribution matrices of C and L conditional on S=1 or S=0 are simple 1 by 1 identity matrices. Moving to more than one observation, the next step is presenting the conditional distribution of the end position F of the first crossing, conditional on the starting position S.

The distribution of the first crossing conditional on the starting position

If the first value is 1 (success), no crossing means that all the remaining $n-1$ values are also 1, therefore $P_n(F=1 \mid S=1) = p^{n-1}$. Similarly, $P_n(F=1 \mid S=0) = q^{n-1}$. Next, if the first value is $f = 2, \cdots, n$ and the first value is a success, $F = f$ means that the sequence starts with a success, then $f-2$ more successes and then one failure. Therefore,

$$P_n(F=f \mid S=1) = p^{f-2} \cdot q, P_n(F=f \mid S=0) = q^{f-2} \cdot p, f = 2, \ldots, n$$

where the last formula is based on a similar argument conditional on S=0. In the following, arguments will in many cases be given for S=1 only, and similar results for S=0 will be stated with no explicit arguments. By symmetry, these results will simply involve replacing p by q. In the next step the formulas in this section will be used for partitioning the joint conditional probabilities for C and L given S, by the position F of the first crossing.

Partitioning by the position of the first crossing

Partitioning on F we have

$$P_n(L=l, C=c \mid S=1) = \sum_{f=1}^{n} P_n(L=l, C=c \mid S=1, F=f) \cdot P_n(F=f \mid S=1)$$

where, as shown in the previous section, $P_n(F=f \mid S=1) = p^{f-2}q$ if $f \geq 2$ and $P_n(F=1 \mid S=1) = p^{n-1}$. The formula for $P_n(L=l, C=c \mid S=0)$ is the same, just interchanging p and q. This implies that the joint probabilities of C and L conditional on S may be computed if it is possible to compute all the joint probabilities of C and L conditional on S and F. This is the next step.

Joint distribution conditional on both S and F

First, if there is no crossing ($F=1$) the entire series constitutes one single run, therefore

$$P_n(C=0, L=1 \mid S=1, F=1) = P_n(C=0, L=1 \mid S=0, F=1) = 1$$

and all other conditional probabilities are 0. Thus, the matrices of joint probabilities of C and L conditional on $F=1$ together with each value of S, are matrices with all components equal to 0, except for a 1 in the upper right corner.

If crossings do occur ($f = 2, \ldots, n$), the conditional probabilities

$$P_n(C=c, L=l \mid S=1, F=f), P_n(C=c, L=l \mid S=0, F=f)$$

are more complicated. The key to computing these probabilities is to recognize that, except for the initial run of $f-1$ observations, the remaining observations constitute $n-(f-1) = n+1-f$ identical and independent Bernoulli observations with success probability p, they represent the same setting as for all n observations, just a shorter sequence. Further, these $n+1-f$ observations are also conditional on a fixed value of their first observation, only that this fixed value is the opposite as in the entire series. This is because the last $n+1-f$ observations start with the observation after the first crossing.

We now have to distinguish between two cases. In case 1, the initial run of $f-1$ observations before the first crossing are at least as many as the last $n+1-f$ ones. In case 2, the initial run is shorter:

Case 1: $f-1 \geq n+1-f$

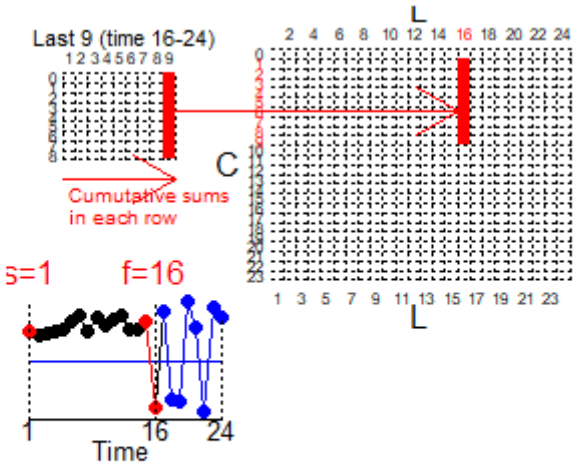
Case 2: $f-1 < n+1-f$

Case 1, at least as many observations before the first crossing as thereafter

This is the simplest case. Here, the first $f-1$ observations constitute a run of length $f-1$, and no run in the last $n+f-1$ observations may be longer than that. Therefore, the longest run is $f-1$, and the non-zero probabilities $P_n(C=c, L=l \mid S=1, F=f)$ are confined to the vertical strip $l = f-1$. And, in fact, to only a part of this strip. First, there is at least one crossing, from time $f-1$ to f. Also, any further crossings are within the last $n+1-f$ observations, and may be any number between 0 and $(n+1-f)-1$. The total number of crossings may therefore be any number between 1 and $n+1-f$,

which means that the non-zero probabilities $P_n(C = c, L = l \mid S = 1, F = f)$ are confined to the strip $l = f - 1, 1 \leq c \leq n + 1 - f$. The following figure illustrates this in the case $n = 24$ and $f = 16$.

Figure 1: A runs chart with n=24 time points, starting above the midline (S=1), and with the first crossing ending at time 16 (F=16). The last 9 observations, from time 16 on, start below the midline (S=0). The figure also shows (upper right) the joint distribution of crossings and runs for the whole time series conditional on (S=1, F=16), and (upper left) the joint distribution among the last 9 observations conditional on (S=0).



The non-zero probabilities $P_n(C = c, L = l \mid S = 1, F = f), l = f - 1, 1 \leq c \leq n + 1 - f$ are somehow determined by what happens within the last n+1-f observations. The precise procedure for computing these probabilities is illustrated by the red arrows in Figure 1 above. The clue is to see that the last $n + 1 - f$ observations constitute a situation identical to the original n observations, only shorter, and with the starting observation fixed on the opposite side of the central line. To put it into a formula,

$$P_n(C = c, L = f - 1 \mid S = 1, F = f) = P_{n+1-f}(C = c - 1 \mid S = 0)$$

where $C = c - 1$ is because the crossing from $f - 1$ to f is just before the last $n + 1 - f$ observations. Similarly,

$$P_n(C = c, L = f - 1 \mid S = 0, F = f) = P_{n+1-f}(C = c - 1 \mid S = 1)$$

The probabilities on the right hand side of these formulas are for a lower number of observations and are therefore already computed in the iterative procedure.

Note that the $n + 1 - f$ observations after the initial run start on opposite side of the middle line. Therefore it is necessary to compute conditional probabilities conditional on starting values both above and below the middle line in the iterative procedure, they cannot be computed separately. The computations are a bit moe complicated in the second case, when the initial run is the shorter part, but the main idea is the same.

Case 2, fewer observations before the first crossing than thereafter

As in case 1, the total number of crossings is between 1 and $n - f + 1$. As to the longest run L, it cannot be shorter than $f - 1$ or longer than $n - f + 1$, and it is necessary to distinguish between values $l = f - 1$ and $l \geq f$. A longest run $f - 1$ in the entire series means that all runs in the last $n - f + 1$ observations have length $l \leq f - 1$. Therefore

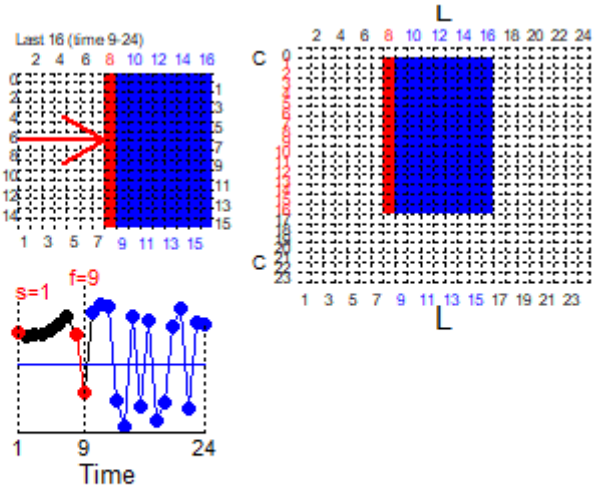
$$P_n(C = c, L = f - 1 \mid S = 1, F = f) = P_{n+1-f}(C = c - 1, L \leq f - 1 \mid S = 0)$$

(and similarly conditional on S=0). For longer runs, $f \leq l \leq n + 1 - f$ the longest run has to be within the last n+1-f observations and we have

$$P_n(C = c, L = l \mid S = 1, F = f) = P_{n+1-f}(C = c - 1, L = l \mid S = 0)$$

(and similarly conditional on S=0). All these conditional probabilies, based on a shorter sequence, have already been computed in an iterative computation procedure. This procedure is illustrated in Figure 2:

Figure 2: A runs chart with n=24 time points, starting above the midline ($S = 1$), and with the first crossing ending at time 9 ($F = 9$). Also (top right) the joint distribution of C and L conditional on (S=1, F=9), and (top left) the joint distribution among the last 16 observations conditional on (S=0), including a red strip representing the cumulative sum of the first $f - 1 = 8$ columns, and a blue rectangle representing joint probabilities from L=9 on. The corresponding contributions to the joint probabilities for n=24 are shown in the same colors (top right).



Simplifications in the symmetric case

For $p = \frac{1}{2}$ there is a symmetry between crossings up or down, and between success and failure runs. Therefore conditioning on the first observation is not necessary, although it is still necessary to partition on the first crossing F. Also, by an induction argument following the iterative procedure, all these probabilities are integer multiples of $\left(\frac{1}{2}\right)^{n-1}$ and, in fact, represent a partition of the binomial coefficients in the distribution of C, by the values $l = 1, \ldots, n$ of L.

Precision considerations

To enhance precision, computations have been performed in the R package Rmpfr (Maechler, 2018), an R interface to the GNU MPFR library (Fousse et al., 2007). Preliminary investigations pointed to precision problems above values about 50 for series length n without this increased precision, but no such problems up to $n = 100$ when using Rmpfr. To further enhance precision, probabilities have been multiplied by m^{n-1} where m is a multiplier with default value 2. Thereby very small numbers are avoided, at least to some extent, and the numbers computed are integers in the symmetric case. The joint probabilities for a low $n = 15$ are shown below in this representation:

n=15	l=1	l=2	l=3	l=4	l=5	l=6	l=7	l=8	l=9	l=10	l=11	l=12	l=13	l=14	l=15
c= 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
c= 1	0	0	0	0	0	0	0	2	2	2	2	2	2	2	0
c= 2	0	0	0	0	1	9	18	18	15	12	9	6	3	0	0
c= 3	0	0	0	4	48	88	84	60	40	24	12	4	0	0	0
c= 4	0	0	1	100	280	270	175	100	50	20	5	0	0	0	0
c= 5	0	0	50	530	666	420	210	90	30	6	0	0	0	0	0
c= 6	0	0	357	1197	861	392	147	42	7	0	0	0	0	0	0
c= 7	0	8	1008	1456	672	224	56	8	0	0	0	0	0	0	0
c= 8	0	84	1470	1044	324	72	9	0	0	0	0	0	0	0	0
c= 9	0	252	1200	450	90	10	0	0	0	0	0	0	0	0	0
c=10	0	330	550	110	11	0	0	0	0	0	0	0	0	0	0
c=11	0	220	132	12	0	0	0	0	0	0	0	0	0	0	0
c=12	0	78	13	0	0	0	0	0	0	0	0	0	0	0	0
c=13	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0
c=14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The corresponding joint probabilities are obtained by dividing these integers by $2^{n-1} =$

$2^{14} = 16384$, for instance $P(C = 5, L = 6) = 420/16384 = 0.026$. The highest joint probability is $P(C = 8, L = 3) = 1470/16384 = 0.090$. It is also seen that a hight proportion of the joint probabilities consists of zeroes, and except for some very small numbers the joint probabilities are concentrated within a narrow band sloping band. These are fairly general phenomena. For comparison the joint distribution for n=15 is also shown below for p=0.6, a case where observations tend to stay above the midline. These probabilities are still shown in the "times" representation, they are multiplied by $2^{n-1} = 16384$, and are shown with one decimal digit:

p=0.7	l=1	l=2	l=3	l=4	l=5	l=6	l=7	l=8	l=9	l=10	l=11	l=12	l=13	l=14	l=15
c=0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.7
c=1	0	0	0	0	0	0	0	1.5	1.8	2.3	3.2	4.7	6.9	10.3	0
c=2	0	0	0	0	1.1	11.4	29.3	33.7	31.8	29.7	26.8	21.9	13.8	0	0
c=3	0	0	0	3.0	41.7	91.7	105.9	87.8	70.3	51.9	32.6	13.9	0	0	0
c=4	0	0	0.9	99.2	318.3	354.3	260.6	172.1	101.9	49.2	15.1	0	0	0	0
c=5	0	0	39.3	468.8	670.6	483.1	278.4	141.5	57.1	14.1	0	0	0	0	0
c=6	0	0	319.2	1180.3	948.8	485.7	210.1	70.9	14.2	0	0	0	0	0	0
c=7	0	6.0	836.1	1324.0	679.9	257.7	75.4	12.9	0	0	0	0	0	0	0
c=8	0	68.5	1289.2	994.7	341.5	86.5	12.6	0	0	0	0	0	0	0	0
c=9	0	196.4	991.5	402.9	89.8	11.5	0	0	0	0	0	0	0	0	0
c=10	0	267.1	466.8	100.8	11.3	0	0	0	0	0	0	0	0	0	0
c=11	0	170.3	106.7	10.5	0	0	0	0	0	0	0	0	0	0	0
c=12	0	61.2	10.6	0	0	0	0	0	0	0	0	0	0	0	0
c=13	0	10.5	0	0	0	0	0	0	0	0	0	0	0	0	0
c=14	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0

R code for the iteration procedure

The iterative procedure is coded in the function crossrunbin.

```
crossrunbin <- function(nmax=100, prob=.5, mult=2, prec=120, printn=FALSE) {
  nill <- mpfr(0,prec)
  one <- mpfr(1,prec)
  multm <- mpfr(mult,prec)
  pm <- mpfr(prob,prec)
  qm <- one - pm
  pmultm <- pm*multm
  qmultm <- qm*multm
  # conditioning of S= first value, pat: above 0, pbt: below 0
  # suffix t: probabilities times multm^(n-1).
  # n=1:
  pat <- list(pt1=mpfr2array(one, dim=c(1,1)))
  pbt <- list(pt1=mpfr2array(one, dim=c(1,1)))
  pt <- list(pt1=mpfr2array(one, dim=c(1,1)))
  qat <- list(pt1=mpfr2array(one, dim=c(1,1)))
  qbt <- list(pt1=mpfr2array(one, dim=c(1,1)))
  qt <- list(pt1=mpfr2array(one, dim=c(1,1)))
  for (nn in 2:nmax) {
    pat[[nn]] <- mpfr2array(rep(nill,nn*nn), dim=c(nn,nn))
    pbt[[nn]] <- mpfr2array(rep(nill,nn*nn), dim=c(nn,nn))
    rownames(pat[[nn]]) <- c(0:(nn-1))
    rownames(pbt[[nn]]) <- c(0:(nn-1))
    colnames(pat[[nn]]) <- c(1:nn)
    colnames(pbt[[nn]]) <- c(1:nn)
    pat[[nn]][1,nn] <- (pmultm^(nn-1)) # from cond on no crossing
    pbt[[nn]][1,nn] <- (qmultm^(nn-1)) # from cond on no crossing
    for (ff in 2:nn) { # from cond on first crossing at ff
      if (nn-ff+1<=ff-1) { # if last part shortest:
        f1 <- ff # unnecessary, but makes code checking easier
        pat[[nn]][2:(nn-f1+2),f1-1] <-
          pat[[nn]][2:(nn-f1+2),f1-1] +
          (pmultm^(f1-2)) * qmultm * qbt[[nn-f1+1]][1:(nn-f1+1),nn-f1+1]
        pbt[[nn]][2:(nn-f1+2),f1-1] <-
          pbt[[nn]][2:(nn-f1+2),f1-1] +
          (qmultm^(f1-2)) * pmultm * qat[[nn-f1+1]][1:(nn-f1+1),nn-f1+1]
      } # end if last part shortest
      if (nn-ff+1>ff-1) {# if last part longest
```

```
        f2 <- ff # unnecessary, but makes code checking easier
        pat[[nn]][2:(nn-f2+2),f2-1] <-
          pat[[nn]][2:(nn-f2+2),f2-1] +
          (pmultm^(f2-2)) * qmultm * qbt[[nn-f2+1]][1:(nn-f2+1),f2-1]
        pat[[nn]][2:(nn-f2+2),f2:(nn-f2+1)] <-
          pat[[nn]][2:(nn-f2+2),f2:(nn-f2+1)] +
          (pmultm^(f2-2)) * qmultm * pbt[[nn-f2+1]][1:(nn-f2+1),f2:(nn-f2+1)]
        pbt[[nn]][2:(nn-f2+2),f2-1] <-
          pbt[[nn]][2:(nn-f2+2),f2-1] +
          (qmultm^(f2-2)) * pmultm * qat[[nn-f2+1]][1:(nn-f2+1),f2-1]
        pbt[[nn]][2:(nn-f2+2),f2:(nn-f2+1)] <-
          pbt[[nn]][2:(nn-f2+2),f2:(nn-f2+1)] +
          (qmultm^(f2-2)) * pmultm * pat[[nn-f2+1]][1:(nn-f2+1),f2:(nn-f2+1)]
      } # end if last part longest
    } # end for ff
    pt[[nn]] <- pm*pat[[nn]] + qm*pbt[[nn]]
    qat[[nn]] <- cumsumm(pat[[nn]])
    qbt[[nn]] <- cumsumm(pbt[[nn]])
    qt[[nn]] <- pm*qat[[nn]] + qm*qbt[[nn]]
    rownames(pt[[nn]]) <- c(0:(nn-1))
    colnames(pt[[nn]]) <- c(1:nn)
    rownames(qat[[nn]]) <- c(0:(nn-1))
    colnames(qat[[nn]]) <- c(1:nn)
    rownames(qbt[[nn]]) <- c(0:(nn-1))
    rownames(qat[[nn]]) <- c(0:(nn-1))
    colnames(qt[[nn]]) <- c(1:nn)
    colnames(qt[[nn]]) <- c(1:nn)
    if (printn) {
      print(nn)
      print(Sys.time())
    } # end optional timing information
  } # end for nn
  names(pat) <-paste("pat", 1:nmax, sep="")
  names(pbt) <-paste("pbt", 1:nmax, sep="")
  names(pt) <-paste("pt", 1:nmax, sep="")
  names(qat) <-paste("qat", 1:nmax, sep="")
  names(qbt) <-paste("qbt", 1:nmax, sep="")
  names(qt) <-paste("qt", 1:nmax, sep="")
  return(list(pat=pat,pbt=pbt,pt=pt,qat=qat,qbt=qbt,qt=qt))
} # end function crossrunbin
```

Limitations

One main limitation is that the iterative procedure cannot in any obvious way be generalized to autocorrelated time series. It may, however be generalized to time series with independent, but not necessarily identical, observations, this is implemented in a function crossrunchange. Another limitation is that the code has so far only been checked for $n \leq 100$. In that range it seems to work well. It has been checked with manual computations for $n \leq 6$, and with 100,000 simulations for $n = 100$. Specifically, the mean and standard deviations for both C and L have been computed, as well as the mean of $C \cdot L$ and also the cumulative distribution functions of C and L separately, with no substantial deviations. It has also been checked for $n \leq 100$ that the marginal distribution of C computed from the joint distribion agrees with the correct binomial distribution in the symmetric case. These last investigations werew, in fact, what pointed to the necessity to base the code on Rmpfr. The performance of the procedure has not, however, so far been checked for n>100. For applications to statistical process control $n \leq 100$ should be sufficient in most cases, but other applications may require higher n.

Bibliography

J. Anhøj. Diagnostic value of run chart analysis: Using likelihood ratios to compare run chart rules on simulated data series. *PLoS ONE*, 10(3):e0121349, 2015. URL https://doi.org/10.1371/journal.pone.0121349. [p1]

I. Fazekas, Z. Karácsony, and Z. Libor. Diagnostic value of run chart analysis: Using likelihood ratios to compare run chart rules on simulated data series. *Acta Univ. Sapientiae, Mathematica*, 2(2):215 – 228, 2010. [p1]

L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. Mpfr: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw.*, 33(2):13, 2007. ISSN 0098-3500. doi: http://doi.acm.org/10.1145/1236463.1236468. [p4]

M. Maechler. *Rmpfr: R MPFR - Multiple Precision Floating-Point Reliable*, 2018. URL https://CRAN.R-project.org/package=Rmpfr. R package version 0.7-0. [p4]

M. F. Schilling. The surprising predictability of long runs. *Mathematics Magazine*, 85(2):141 – 149, 2012. URL https://www.jstor.org/stable/10.4169/math.mag.85.2.141. [p1]

Tore Wentzel-Larsen
Centre for Child and Adolescent Mental Health, Eastern and Southern Norway
Norwegian Centre of Violence and Traumatic Stress Studies
Norway
tore.wentzellarsen@gmail.com

Jacob Anhøj
Rigshospitalet, University of Copenhagen
Denmark
(ORCiD https://orcid.org/0000-0002-7701-1774)
jacob@anhoej.net