I stumbled across [a very interesting recent Econometrica article](https://onlinelibrary.wiley.com/doi/full/10.3982/ECTA14943) by Joshua Miller and Adam Sanjuro. I was really surprised by the statistical result they discovered and guess the issue may even have fooled Nobel Prize winning behavioral economists. Before showing the statistical subtlety, let me briefly explain the Hot Hand Fallacy.

Consider a basketball player who makes 30 throws and whose chance to hit is always 50%, independent of previous hits or misses. The following R code simulates a possible sequence of results (I searched a bit for a nice random seed for the purpose of this post. So this outcome may not be “representative”):

set.seed(62)

x = sample(0:1,30,replace = TRUE)

x # 0=miss, 1=hit

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

The term Hot Hand Fallacy is used by psychologists and behavioral economists for the claim that people tend to systematically underestimate how often streaks of consecutive hits or misses can occur for such samples of i.i.d. sequences.

For example, consider the 5 subsequent hits from throws 9 to 13. In real life such streaks could be due to a hot hand, in the sense that the player had a larger hit probability during these throws than on average. Yet, the streak could also just be a random outcome given a constant hit probability. The Hot Hand Fallacy means that one considers such streaks as stronger statistical evidence against a constant hit probability than is statistically appropriate.

In their classical [article from 1985](https://www.sciencedirect.com/science/article/pii/0010028585900106) Gilovich, Vallone, and Tversky use data from real basketball throws. They compare the conditional probability of a hit given that either the previous 3 throws were a hit or the previous 3 throws were a miss.

Let us in several steps compute these probabilities for our vector x:

# Indexes of elements that come directly after

# a streak of k=3 subsequent hits

inds = find.after.run.inds(x,k=3, value=1)

inds

## [1] 12 13 14 19

The function find.after.run.inds is a custom function (see end of this blog for the code) that computes the indeces of the elements of a vector x that come directly after a streak of k=3 consecutive elements with the specified value. Here we have the 12th throw that comes after the 3 hits in 9,10,11, the 13th throw after the 3 hits in 10,11,12, and so on.

x[inds]

## [1] 1 1 0 0

Directly after all streaks of 3 hits, we find exactly 2 hits and 2 misses.

mean(x[inds])

## [1] 0.5

This means in our sample, we have a hit probability of 50% in throws that are directly preceeded by 3 hits.

We can also compute the a conditional hit probability after a streak of three misses in our sample:

# Look at results after a streak of k=3 subsequent misses

inds = find.after.run.inds(x,k=3, value=0)

mean(x[inds])

## [1] 0.5

Again 50%, i.e. there are no differences between the hit probabilities directly after 3 hits or 3 misses.

Looking at several samples of n throws, Gilovich, Vallone, and Tversky find also no large differences in the conditional hit probability after streaks of 3 hits or 3 misses. Neither do they find relevant differences for alternative streak lengths. They thus argue that in their data, there is no evidence for a hot hand. Believing in a hot hand in their data thus seems to be a fallacy. Sounds quite plausible to me.

Let us now slowly move towards the promised statistic subtlety by performing a systematic Monte-Carlo study:

sim.fun = function(n,k,pi=0.5, value=1) {

# Simulate n iid bernoulli draws

x = sample(0:1,n,replace = TRUE, prob=c(1-pi,pi))

# Find these indeces of x that come directly

# after a streak of k elements of specified value

inds = find.after.run.inds(x,k,value=value)

# If no run of at least k subsequent numbers of value exists

# return NULL (we will dismiss this observation)

if (length(inds)==0) return(NULL)

# Return the share of 1s in x[inds]

mean(x[inds])

}

# Draw 10000 samples of 30 throws and compute in each sample the

# conditional hit probability given 3 earlier hits

hitprob\_after\_3hits = unlist(replicate(10000, sim.fun(n=30,k=3,pi=0.5,value=1), simplify=FALSE))

head(hitprob\_after\_3hits)

## [1] 0.5000000 0.5000000 0.0000000 0.2500000 0.7142857 0.0000000

We have now simulated 10000 times 30 i.i.d. throws and computed for each of the 10000 samples the average probability of a hit in the throws directly after a streak of 3 hits.

Before showing you mean(hitprob\_after\_3hits), you can make a guess in the following quiz. Given that I already announced an interesting subtlety of statistics you can of course meta-guess, whether the subtlety already enters here, or whether at this point the obvious answer is still the correct one:

Loading…

OK let’s take a look at the result:

mean(hitprob\_after\_3hits)

## [1] 0.3822204

# Approximate 95% confidence interval

# (see function definition in Appendix)

ci(hitprob\_after\_3hits)

## lower upper

## 0.3792378 0.3852031

Wow! I find that result really, really surprising. I would have been pretty sure that given our constant hit probability of 50%, we also find across samples an average hit probability around 50% after streaks of 3 hits.

Yet, we find in our 10000 samples of 30 throws on average a substantially lower hit probability of 38%, with a very tight confidence interval.

To get an intuition for why we estimate a conditional hit probability after 3 hits below 50%, consider samples of only n=5 throws. The following table shows all 6 possible such samples that have a throw after a streak of 3 hits.

|  |  |  |
| --- | --- | --- |
| **Row** | **Throws** | **Share of hits after streak of 3 hits** |
| 1 | 11100 | 0% |
| 2 | 11101 | 0% |
| 3 | 11110 | 50% |
| 4 | 11111 | 100% |
| 5 | 01110 | 0% |
| 6 | 01111 | 100% |
| Mean | : | 41.7% |

Assume we have hits in the first 3 throws (rows 1-4). If then throw 4 is a miss (rows 1-2) then throw 5 is irrelevant because it is not directly preceeded by a streak of 3 hits. So in both rows the share of hits in throws directly after 3 hits is 0%.

If instead throw 4 is a hit (rows 3-4) then also throw 5, which is equally likely a hit or miss, is directly preceeded by 3 hits. This means the average share of hits in throws after 3 hits in rows 3-4 is only 75%, while it was 0% in rows 1-2. In total over all 6 rows this leads to a mean of only 41.7%

Of course, the true probability of the player making a hit in a throw directly after 3 hits is still 50% given our i.i.d. data generating process. Our procedure just systematically underestimates this probability. Miller and Sanjuro call this effect a *streak selection bias*. It is actually a small sample bias that vanishes as n goes to infinity. Yet the bias can be quite substantial for small n as the simulations show.

We get a mirroring result if we use our procedure to estimate the mean hit probability in throws that come directly after 3 misses.

hitprob\_after\_3misses = unlist(replicate(10000, sim.fun(n=30,k=3,pi=0.5,value=0), simplify=FALSE))

mean(hitprob\_after\_3misses)

## [1] 0.6200019

ci(hitprob\_after\_3misses)

## lower upper

## 0.6170310 0.6229728

We now have an upward bias and estimate that in throws after 3 misses, we find on average a 62% hit probability instead of only 50%.

What if for some real life samples we would estimate with this procedure that the conditional probabilities of a hit after 3 hits and also after 3 misses are both roughly 50%? Our simulation studies have shown that if there was indeed a fixed probability of a hit of 50%, we should rather estimate a conditional hit probability of 38% after 3 hits and of 62% after 3 misses. This means 50% vs 50% instead of 38% vs 62% is rather statistical evidence for a hot hand!

Indeed, Miller and Sanjuro re-estimate the seminal articles on the hot hand effect using an unbiased estimator for the conditional hit probabilities. While the original studies did not find a hot hand effect and thus concluded that there is a Hot Hand Fallacy, Miller and Sanjuro find substantial hot hand effects. This means, at least in those studies, there was a “Hot Hand Fallacy” Fallacy.

Of course, just by showing that in some data sets there is a previously unrecognized hot hand effect, does not mean that people never fall for the Hot Hand Fallacy. Also, for the case of basketball, it has already be shown before with different data sets and more control variables that there is a hot hand effect. Still, it is kind of a cool story: scientists tell statistical layman that they interpret a data set wrongly, and more than 30 years later one finds out that with the correct statistical methods the layman were actually right.

**Appendix: Custom R functions used above**

# Simply function to compute approximate 95%

# confidence interval for a sample mean

ci = function(x) {

n = length(x)

m = mean(x)

sd = sd(x)

c(lower=m-sd/sqrt(n), upper=m + sd/sqrt(n))

}

find.after.run.inds = function(x,k,value=1) {

runs = find.runs(x)

# Keep only runs of specified value

# that have at least length k

runs = runs[runs$len>=k & runs$val==value,,drop=FALSE]

if (NROW(runs)==0)

return(NULL)

# Index directly after runs of length k

inds = runs$start+k

# Runs of length m>k contain m-k+1 runs

# of length k. Add also all indices of these

# subruns

# The following code is vectorized over rows

# in run

max.len = max(runs$len)

len = k+1

while (len <= max.len) {

runs = runs[runs$len >= len,,drop=FALSE]

inds = c(inds,runs$start+len)

len = len+1

}

# ignore indices above n and sort for convenience

inds = sort(inds[inds<=length(x)])

inds

}

find.runs = function(x) {

rle\_x = rle(x)

# Compute endpoints of run

len = rle\_x$lengths

end = cumsum(len)

start = c(1, end[-length(end)]+1)

data.frame(val=rle\_x$values, len=len,start=start, end=end)

}

