Simulation error

Xueyuan Li

2022-09-13

Introduction

It is common for slight errors to arise during the simulation. There are several types of error. In this assignment, I will demonstrate the concepts of absolute error and relative error when estimating probabilities from simulation.

I will explain it from the following aspects:

1, Explain why the concept is important

2, Introduce key vocabulary terms

3, Demonstrate the concept in action

1,Why the concept is important:

Simulation generates approximate answers; there is some degree of error in a quantity estimated by Monte Carlo simulation. Calculating errors is very important.

Study the sources of errors and their regularity, reduce and eliminate them as much as possible in order to get accurate results and minimize unnecessary losses.

2,Introduce key vocabulary terms

Relative error Relative error is the ratio of the absolute error of a measurement to the true value being measured.

Let $\hat{p}$ denote the probability estimated from simulation, and let p denote the true underlying probability.

relative error = $|\hat{p}-p|/p$.

Generally speaking, the relative error can better reflect the credibility of the measurement.

Absolute error Absolute error is defined as the difference between the actual value and the measured value.

Let $\hat{p}$ denote the probability estimated from simulation, and let p denote the true underlying probability.
absolute error = $|\hat{p}-p|$

The importance of absolute error depends on the quantity that we are measuring. If the quantity is large such as road distance, a small error in centimetres is negligible. While measuring the length of a machine part an error in centimetre is considerable. Though the errors in both cases are in centimetres, the error in the second case is more important.

The absolute error is unavoidable, the relative error can be minimized.

3,Demonstrate the concept in action

We perform an experiment simulation that estimates the error for each combination of replicate number (22, 23, …, 215) and probability (0.01, 0.05, 0.10, 0.25, 0.50).

N is the number of replicates

```
require(tgsify)
```

```
## Loading required package: tgsify
```

```
## Loading required package: data.table
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: dtplyr
```

```
## Loading required package: showtext
```

```
## Loading required package: sysfonts
```

```
## Loading required package: showtextdb
```

```
library(tidyverse)
```

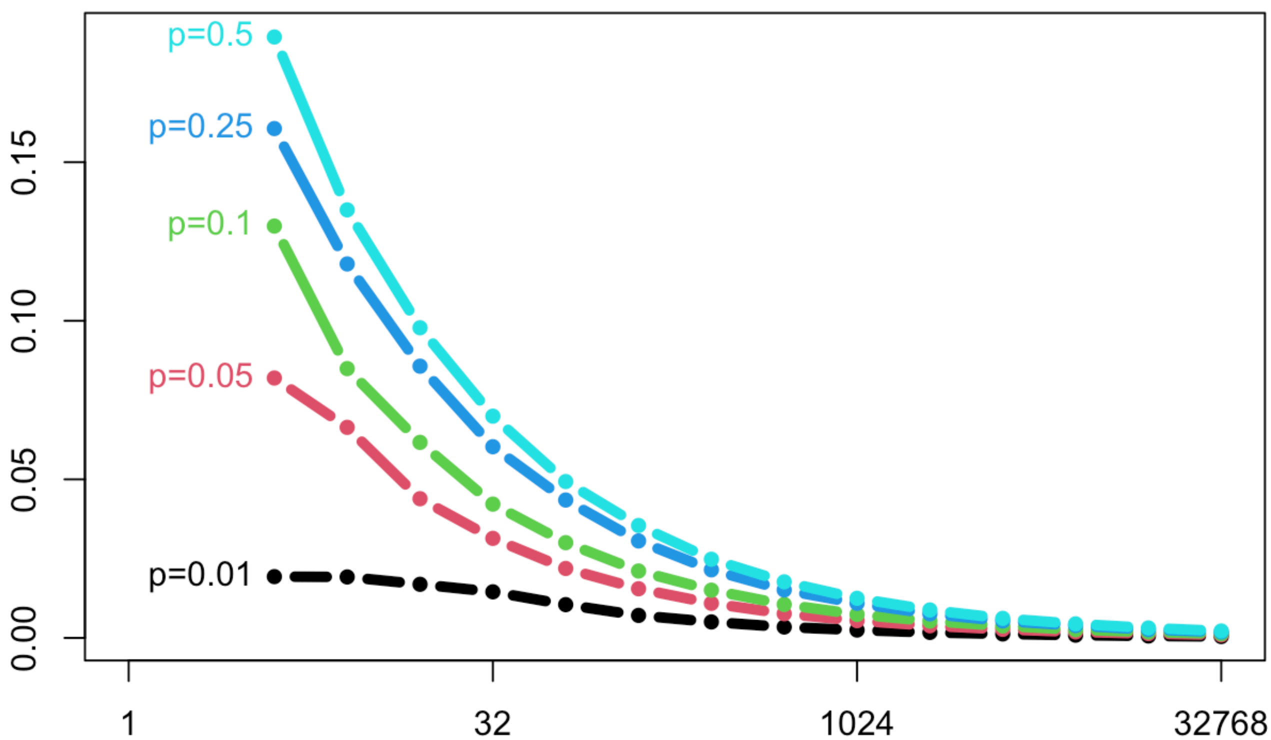
```
## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.8     ✔ stringr 1.4.0
## ✔ tidyr   1.2.0     ✔ forcats 0.5.1
## ✔ readr   2.1.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::between()   masks data.table::between()
## ✖ dplyr::filter()    masks stats::filter()
## ✖ dplyr::first()     masks data.table::first()
## ✖ dplyr::lag()       masks stats::lag()
## ✖ dplyr::last()      masks data.table::last()
## ✖ purrr::transpose() masks data.table::transpose()
```

```
output<-expand.grid(
     N=2^c(2:15)
   , P=c(0.01, 0.05, 0.10, 0.25, 0.50)
   , abs_error=NA
   , rel_error=NA
   , KEEEP.OUT.ATTRS= FALSE
)

r<- 10000
for (i in 1:nrow(output)){
  p<-output$P[i]
  n<-output$N[i]
  phat<- rbinom(r,n,p)/n
  output[i,"abs_error"]<-mean(abs(phat -p))
  output[i,"rel_error"]<-mean(abs(phat -p)/p)
}


#ggplot(output,aes(x = N, y = abs_error,color=P,group=P)) +
#  geom_line()+
#    geom_point()+scale_x_continuous(trans="log2")

#ggplot(output,aes(x = N, y = rel_error,color=P,group=P)) +
#  geom_line()+
#    geom_point()+scale_x_continuous(trans="log2")


#Absolute error
output %>%
  mutate(x=log2(N)) %>%
 # mutate(abs_error=log10(abs_error)) %>%
  mutate(col =as.factor(P) %>% as.numeric) %>%
  #plotstyle(upright) %>%
  plot_setup(abs_error ~ x,c(0,15)) %>%
  split(.$P) %>%
  lwith({
    lines(x, abs_error,col=col[1], lwd=5, type="b", pch=16)
    text(x[1],abs_error[1],"p="%|%P[1], pos=2,col=col[1])
  })
```

```
## $`0.01`
## NULL
##
## $`0.05`
## NULL
##
## $`0.1`
## NULL
##
## $`0.25`
## NULL
##
## $`0.5`
## NULL
```

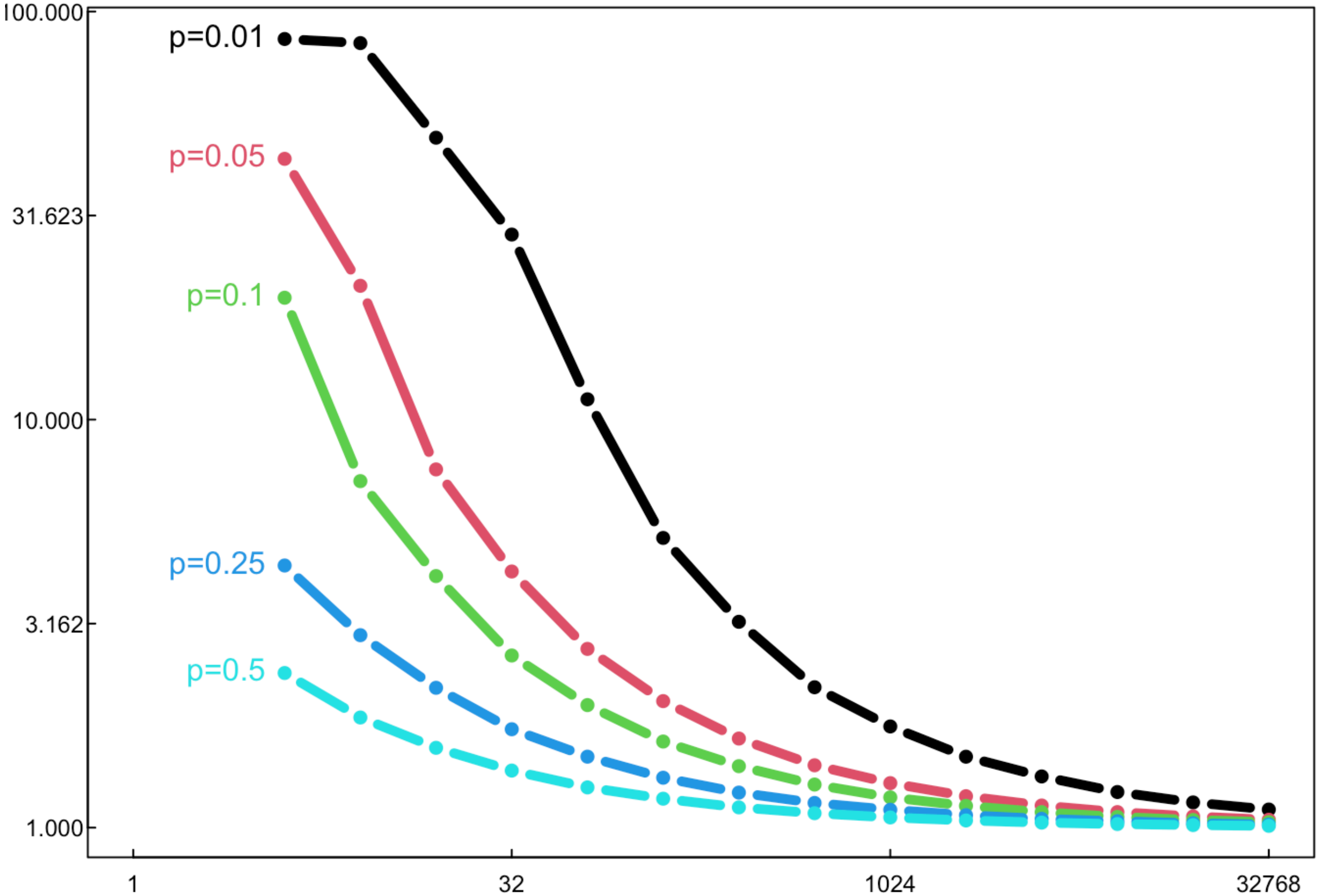
```
axis(2)
axis(1,at=axTicks(1),labels=2^axTicks(1))
box()
```



```
#Relative error
 output %>%
  mutate(x=log2(N)) %>%
  mutate(rel_errorr=log10(rel_error)) %>%
  mutate(col =as.factor(P) %>% as.numeric) %>%
  plotstyle(upright) %>%
  plot_setup(rel_error ~ x,c(0,15)) %>%
  split(.$P) %>%
  lwith({
    lines(x, rel_error,col=col[1], lwd=5, type="b", pch=16)
    text(x[1],rel_error[1],"p="%|%P[1], pos=2,col=col[1])
  })
```

```
## $`0.01`
## NULL
##
## $`0.05`
## NULL
##
## $`0.1`
## NULL
##
## $`0.25`
## NULL
##
## $`0.5`
## NULL
```

```
 axis(2,at=axTicks(2),labels=sprintf("%4.3f",10^axTicks(2)))
 axis(1,at=axTicks(1),labels=2^axTicks(1))
 box()
```



Explanation

I define the functions of relative and absolute error separately,using the for() loop statement, the N,P values in output are brought in sequentially and calculated using rbinom() function to obtain the absolute errors and relative errors.Then I plot test results by the graphs. Here's the observation of the graphs:

We can see from the first graph that the absolute error gradually decreases as the number of tests increases. The probability of setting is low, the absolute error is small, and the absolute error does not vary much, then the smoother the graph line is. However, the greater the probability of setting, the greater the change in absolute error and the greater the decline.

We can also see from the second graph that the relative error gradually decreases as the number of tests increases. Different from the absolute error, the larger the probability, the smaller the relative error and the smaller the variation.

Conclusion

In summary, we have discussed the concept, meaning, and experimental simulation of the two types of errors, absolute and relative. We find the relationship between the number of replicates and simulation error: the degree of error should get smaller as the number of simulation replicates increases.