Goodness of Fit Analysis

Viren Halaharivi

2023-03-17

After the last game night with your friends and Hank, you realize that you need to collect some data on Hank's dice rolls. Throughout the night, you record his rolls of a 6-sided die. Once you get home, you look at the data you've collected, ready to answer once and for all if Hank is a cheater.

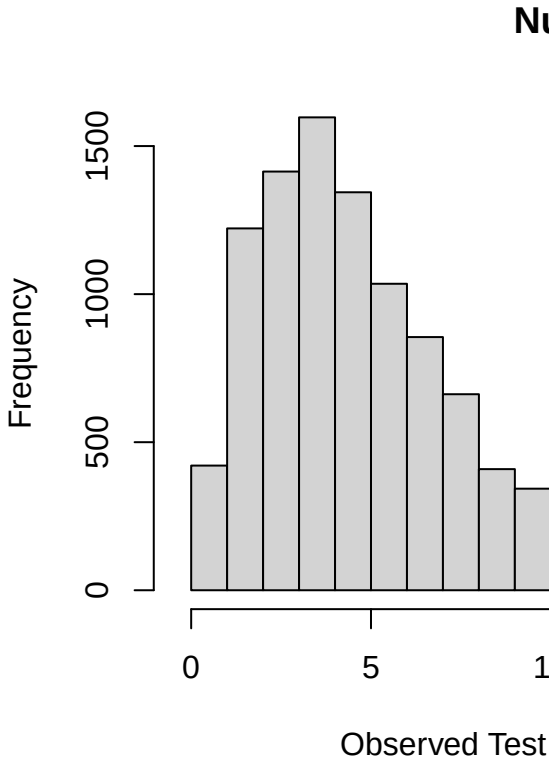
Die Face	1	2	3	4	5	6
Roll Frequency	10	11	13	11	12	27

(i) We will conduct a goodness-of-fit test to determine if Hank is using a fair 6-sided die (i.e., that all sides should show up with equiprobability). State your null hypothesis and compute your test statistic. The null hypothesis is that all sides show up with equiprobability and Hank is using a fair six sided die. The test statistic is 14.857.

```
numrolls = 84
prob = 1/6
exp6 = numrolls*prob
test_stat = sum((rollfreq-exp6)^2/exp6)
test_stat
```

```
## [1] 14.85714
```

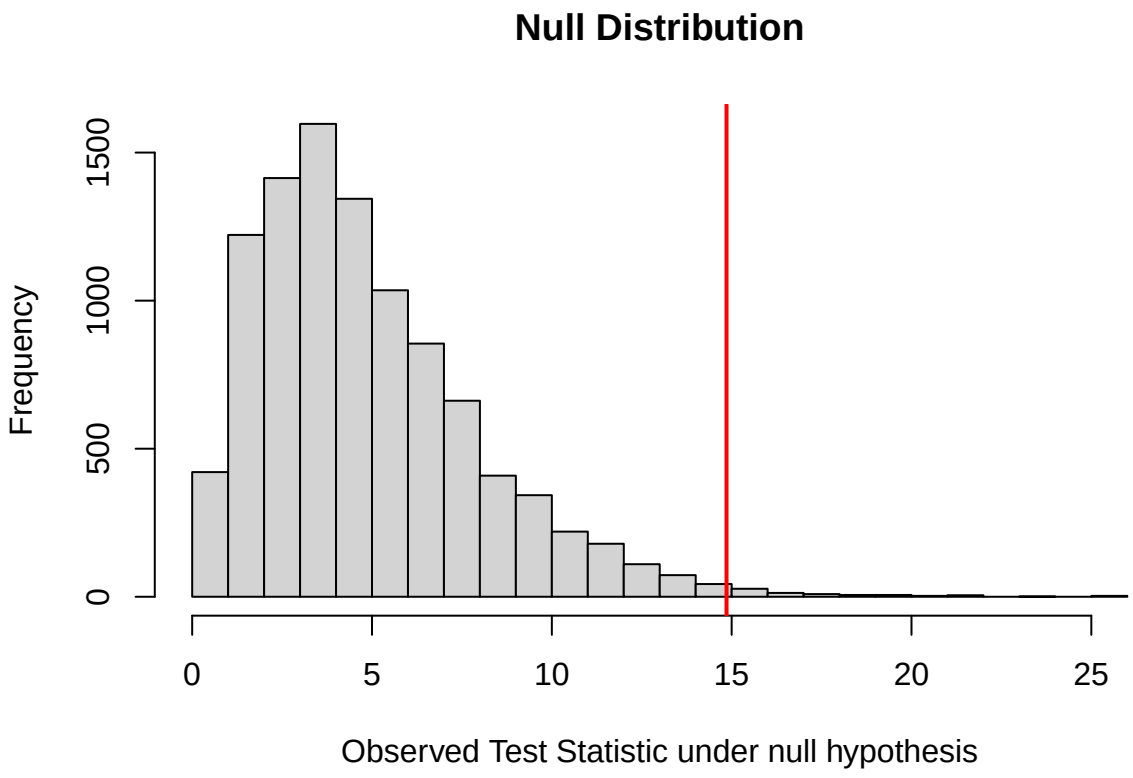
```

n_sims= 10000
p = rep(1/6, 6)
sim_test_stat = rep(NA, n_sims) # a vector indicating whether we see at least one heads each simulation
for(ii in 1:n_sims)
  {
    sim_dice_rolls = tabulate(sample(c(1:6), size=numrolls, prob = p, replace=TRUE), nbins=6)
    sim_test_stat[ii] = sum((sim_dice_rolls - exp6)^2/exp6)
}

hist(sim_test_stat, breaks=20, main="Null Distribution", xlab = "Observed Test Statistic under null hyp
```



(ii) Run 10,000 simulations of $N = 84$ rolls under the null hypothesis.

(iii) Compute the p-value and state your conclusion. The p-value is .01 which is less than a significance value of .05. Thus we reject the null hypothesis and conclude that Hank was using an unfair die.

```
hist(sim_test_stat, breaks=20, main="Null Distribution", xlab = "Observed Test Statistic under null hypo
abline(v=test_stat,col="red",lwd=2)
```



```
p_val = length(which(sim_test_stat>test_stat))/n_sims
p_val
```

```
## [1] 0.0076
```

Caeser's Cipher: "There are 26 possible shifts. If I can compute χ^2 test statistics for each possible shift, using the expected letter frequencies, then I'll be able to find the most likely Caesar cipher shift."

Letters	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Letter Frequency	0.08	0.01	0.02	0.04	0.12	0.02	0.02	0.06	0.07	0.00	0.00	0.04	0.02	0.06	0.07	0.01	0.00	0.06	0.06	0.09	0.02	0.01	0.02	0.00	0.02	0.00

(i) For each of the 26 possible shifts, compute the χ^2 test statistic. The most likely shift is the lowest chi square statistic in the code. In this case it is 23 letters forward or 3 letters backward.

```
text = read.delim("encrypted (1).txt", header = FALSE, sep = "\n")
length = str_count(text, "")

text1 = str_split(text, '')


txt_tablefreq = table(text1)/length
comparison = as.vector(txt_tablefreq)


#26 possible shift for loop
#for loop through each of the 26 shifts
#calculate chi squares for each shift
#then we choose the lowest value as our statistic/shift.
chi_test = rep(0,26)
for(i in 0:25){
  possible_shift = c(letterfreq[(i+1):26], letterfreq[1:i])
  chi_test[i] = sum((possible_shift-comparison)^2/comparison)


}
```

```
## Warning in possible_shift - comparison: longer object length is not a multiple
## of shorter object length
```

```
## Warning in (possible_shift - comparison)^2/comparison: longer object length is
## not a multiple of shorter object length
```

```
chi_test[0:25]
```

```
##  [1] 11.863681372 53.198411068  4.785536795 12.730573918 16.347872684
##  [6] 21.046734944 18.356346003  5.143934297  6.526536670  5.319926124
## [11] 29.382709046 18.204047035  5.604144873 15.125677534 16.765788058
## [16] 13.371308919 29.405330972 24.415841169  7.536272349  7.934617278
## [21] 11.508397232  8.514788767  0.009897153 23.258638561  7.540603026
```

(ii) Using your computed χ^2 statistics in (i), which shift should we use to decrypt the message? According to my chi statistics we should use 23 spaces forward or 3 letters backward to decrypt the message.

```r
most_likely_shift <- which.min(chi_test[0:25])
most_likely_shift
```

```
## [1] 23
```

(iii) What is your decrypted message? Do you believe that you cracked the code? Could not figure out the code to decrypt the message however shifting 23 letters give us the message:

ONANEXCEPTIONALLYHOTEVENINGEARLYINJULYAYOUNGMANCAMEOUTOFTHEGARRETINWHICHHELOD

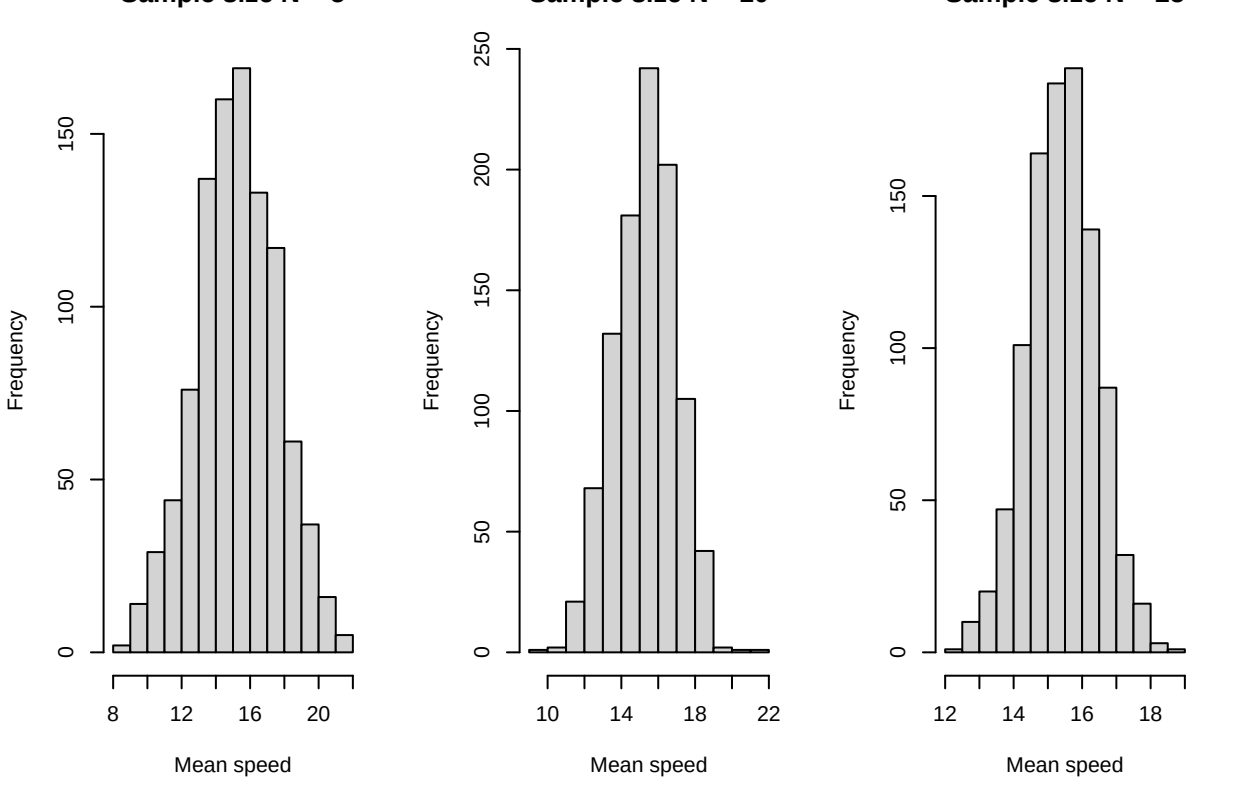
I do believe I cracked the code.

Question 3 - Need for Speed

In this question, we will visualize the central limit theorem through an exploration of the `cars` R package.

```r
N <- c(5, 10, 25)
M <- 1000

par(mfrow = c(1, 3))

for (i in 1:3) {
  means <- replicate(M, mean(sample(cars$speed, size = N[i], replace = TRUE)))
  hist(means, main = paste0("Sample size N = ", N[i]), xlab = "Mean speed")
}
```

(i) For $M = 1000$ simulations, compute the mean of the speed of a random sample of N cars. Then, plot a histogram of the simulated means. Do this for $N = 5, 10, 25$.



(ii) What do you notice as N increases? Why does this happen? AS N increases the sampling variability decreases. If you use the equation of the central limit theorem you'll see as N gets larger the standard error gets smaller sigma/n = standard error. The graph becomes more bell shaped and there is less variability.