

R code for inference (confidence interval, hypothesis testing, power) about a single proportion.

Hypothesis testing and P-values:

Suppose our data are such that out of a sample of $n=180$ trials (=students), 120 resulted in successes (=indicated that they are in favor of lowering the drinking age to below 18 years). Our goal is to test hypotheses and construct confidence intervals about the true (unknown) proportion of successes (proportion of students who are in favor of lowering the drinking age) in the entire population. Let's look at a two-sided test of the (alternative) hypothesis that the true proportion is different from 0.6 first. If you want to get the **P-value** for a **two-sided** test of the null hypothesis $H_0: \pi = 0.6$ versus the alternative hypothesis $H_A: \pi \neq 0.6$, use the following R command. (The P-value and relevant part of the output is highlighted below)

```
> prop.test(x=120, n=180, p=0.6, correct=FALSE)
```

```
1-sample proportions test without continuity correction
data: 120 out of 180, null probability 0.6
X-squared = 3.3333, df = 1, p-value = 0.06789
alternative hypothesis: true p is not equal to 0.6
95 percent confidence interval:
 0.5949523 0.7314158
sample estimates:
      p
0.6666667
```

I.e., the P-value for the two-sided test is 0.06789. At a significance level of (alpha=) 5%, we have insufficient evidence (P-value > 0.05) to reject the null hypothesis. There is insufficient evidence to conclude that the true proportion of students who are in favor of lowering the drinking age is different from 0.6.

For this problem, it's probably more natural to ask: "Is the proportion of students who want to lower the drinking age significantly larger than 60%?" This requires a one-sided test. For getting the (one-sided) P-value for the one-sided test when the alternative hypothesis has the form $H_A: \pi > 0.6$, type

```
> prop.test(x=120,n=180, p=0.6, alternative="greater", correct=FALSE)
```

```
1-sample proportions test without continuity correction
data: 120 out of 180, null probability 0.6
X-squared = 3.3333, df = 1, p-value = 0.03394
alternative hypothesis: true p is greater than 0.6
95 percent confidence interval:
 0.6067808 1.0000000
sample estimates:
      p
0.6666667
```

Conclusion: At a 5% significance level, there is sufficient evidence ($P\text{-value} < 5\%$) to reject the null hypothesis and conclude that the true proportion of students in favor of lowering the drinking age is significantly larger than 60%. (Note: The conclusion looks different at a 2.5% significance level.)

To get the $P\text{-value}$ when the alternative hypothesis has the form $H_A: \pi < 0.6$, use `alternative="less"`. (Wouldn't make much sense to test for our example, as the sample proportion is already larger than 0.6!)

Confidence Intervals:

The following code gives the confidence interval for a single proportion for the data on the proportion of students favoring lowering the drinking age ($n=180$, # of students in favor: 120). The confidence interval is highlighted.

```
> prop.test(x=120, n=180, correct=FALSE)

      1-sample proportions test without continuity correction

data:  120 out of 180, null probability 0.5
X-squared = 20, df = 1, p-value = 7.744e-06
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.5949523 0.7314158
sample estimates:
      p
0.6666667
```

Conclusion: We are 95% confident that the interval from 59.5% to 73.1% captures the true proportion of students who are in favor of lowering the drinking age. You can also say: We are 95% confident that the true proportion of students in favor of lowering the drinking age is at least 59.5% and at most 73.3%. **Note** that 60% is included in the confidence interval, that's why we could not reject the hypothesis $H_0: p = 0.6$. In this regard, the confidence interval is much more informative than the hypothesis test. It not only tells you whether you can reject or not reject the null hypothesis, but it also gives you information about the range (size) of plausible values of the unknown proportion.

If you want the 90% confidence interval, type (I just show the relevant output):

```
> prop.test(x=120, n=180, conf.level=0.90, correct=FALSE)
...
90 percent confidence interval:
 0.6067808 0.7216165
...
```

If you want to get a **one-sided** confidence interval, i.e., just a lower bound or just an upper bound, use the option `alternative=...`:

```
> prop.test(x=120,n=180, alternative="greater", correct=FALSE) #this
gives LOWER bound
```

```
...
95 percent confidence interval:
0.6067808 1.0000000
```

Conclusion: We are 95% confident that the true proportion of students favoring lowering the drinking age is at least 60.7%. (So, slightly larger than 60%, which agrees exactly with our one-sided test above!)

Often, for one-sided confidence intervals, we use a level of 97.5%, which you can easily get by typing:

```
> prop.test(x=120,n=180, alternative="greater", conf.level=0.975,
correct=FALSE)
```

```
...
97.5 percent confidence interval:
0.5949523 1.0000000
```

Conclusion: We are 97.5% confident that the true proportion of students favoring lowering the drinking age is at least 59.5%. (So, slightly less than 60%, which is why we couldn't reject the hypothesis $H_0: \pi = 0.6$ in favor of the alternative hypothesis $H_A: \pi > 0.6$ at the 2.5% significance level!)

For an upper bound (not really relevant for the example here), use:

```
> prop.test(x=120,n=180, alternative="less", correct=FALSE)
```

```
...
95 percent confidence interval:
0.0000000 0.7216165
```

Note: The underlying formula (for the two-sided interval) that R is using to compute this confidence interval (called the **Wilson score interval** for a single proportion) is given by this:

$$\hat{\pi} \left(\frac{n}{n + z_{\alpha/2}^2} \right) + \frac{1}{2} \left(\frac{z_{\alpha/2}^2}{n + z_{\alpha/2}^2} \right) \pm z_{\alpha/2} \sqrt{\frac{1}{n + z_{\alpha/2}^2} \left[\hat{\pi}(1 - \hat{\pi}) \left(\frac{n}{n + z_{\alpha/2}^2} \right) + \left(\frac{1}{2} \right) \left(\frac{1}{2} \right) \left(\frac{z_{\alpha/2}^2}{n + z_{\alpha/2}^2} \right) \right]}$$

where $\hat{\pi}$ is the sample proportion and $z_{\alpha/2}$ is the $1-\alpha/2$ quantile from the standard normal distribution (i.e., 1.96 when $\alpha=5\%$) However, you do not need to manually compute the confidence interval using this formula, it is automatically done in the `prop.test()` function in R!

Power:

Let's pretend that the true proportion $\pi = 0.7$. We are interested in the power of our test to detect this, i.e., what is the power (=probability of rejecting the null when it is false (or when the alternative is true)) of our test to reject $H_0: \pi = 0.6$ in favor of the alternative hypothesis $H_A: \pi > 0.6$ when the true proportion $\pi = 0.7$.

Compute right-hand side of formula (see class notes):

```
> null.pi=0.6
> true.pi=0.7
> n=180
> alpha=0.05
> z.q=qnorm(1-alpha)
> z.q
[1] 1.644854
> luis = (z.q*sqrt(null.pi*(1-null.pi)/n) + null.pi - true.pi) /
sqrt(true.pi*(1-true.pi)/n)
> luis
[1] -1.169278
> 1-pnorm(luis)
[1] 0.8788541
```

So, the probability to reject H_0 is equal to 87.9% when the true proportion is equal to 0.7.

Now, find the power for a bunch of values for the true proportion, let's say from 0.6 to 0.8. You can repeat the above calculation, but it is best to do this with a function in R where you just pass the arguments. Just copy and paste this function into R:

```
power <- function(null.pi, true.pi, n, alpha=0.05, alternative="not equal") {
  z.q = switch(alternative,
    "less" = qnorm(alpha),
    "greater" = qnorm(1-alpha),
    qnorm(1-alpha/2)
  )
  luis = (z.q*sqrt(null.pi*(1-null.pi)/n) + null.pi - true.pi) / sqrt(true.pi*(1-true.pi)/n)
  if (alternative=="not equal") {
    z.q = qnorm(alpha/2)
    luis1 = (z.q*sqrt(null.pi*(1-null.pi)/n) + null.pi - true.pi) / sqrt(true.pi*(1-true.pi)/n)
  }
  pow = switch(alternative,
    "less" = pnorm(luis),
    "greater" = 1-pnorm(luis),
    pnorm(luis1) + (1-pnorm(luis))
  )
  return(pow)
}
```

For example, if we call this function with the same setup as before, we get the same answer:

```
> power(null.pi=0.6, true.pi=0.7, n=180, alternative="greater")
[1] 0.8788541
```

But now we can call it for different true.pi values:

```
> power(null.pi=0.6, true.pi=0.6, n=180, alternative="greater")
[1] 0.05 (Why?)
power(null.pi=0.6, true.pi=0.65, n=180, alternative="greater")
[1] 0.3885823
> power(null.pi=0.6, true.pi=0.68, n=180, alternative="greater")
[1] 0.716831
```

```
> power(null.pi=0.6, true.pi=0.72, n=180, alternative="greater")
[1] 0.9633536
```

In fact, we can get more values by typing

```
> power.seq = power(null.pi=0.6, true.pi=seq(0.6,0.75,0.005), n=180,
alternative="greater")
> power.seq
 [1] 0.05000000 0.06537469 0.08425147 0.10703758 0.13407596
 [6] 0.16561123 0.20175590 0.24246009 0.28748818 0.33640617
[11] 0.38858229 0.44320297 0.49930442 0.55581851 0.61162996
[16] 0.66564004 0.71683104 0.76432552 0.80743429 0.84568867
[21] 0.87885412 0.90692464 0.93009951 0.94874609 0.96335362
[26] 0.97448362 0.98272238 0.98863992 0.99275858 0.99553238
[31] 0.99733712
> plot(power.seq~seq(0.6,0.75,0.005), type="l", main="Power for HA:
pi>true.pi", xlab="true.pi", ylab="Power")
```

One can also plot power for fixed true.pi, but varying n, to see the effect of the sample size.

```
> n.seq = power(null.pi=0.6, true.pi=0.7, n=seq(50,200,10),
alternative="greater")
> n.seq
 [1] 0.4147320 0.4728474 0.5268363 0.5766684 0.6223939
 [6] 0.6641283 0.7020363 0.7363179 0.7671962 0.7949070
[11] 0.8196915 0.8417895 0.8614353 0.8788541 0.8942595
[16] 0.9078522
> plot(n.seq~seq(50,200,10), type="l", main="Power for HA: pi>0.7",
xlab="sample size n", ylab="Power")
```

Finally, one can do the power plot for various values of n:

```
> power.seq.n1 = power(null.pi=0.6, true.pi=seq(0.6,0.75,0.005), n=80,
alternative="greater")

> power.seq.n2 = power(null.pi=0.6, true.pi=seq(0.6,0.75,0.005),
n=130, alternative="greater")

> power.seq.n3 = power(null.pi=0.6, true.pi=seq(0.6,0.75,0.005),
n=180, alternative="greater")

> plot(power.seq.n1~seq(0.6,0.75,0.005), type="l", main="Power for HA:
pi>true.pi", xlab="true.pi", ylab="Power", col="red", ylim=c(0,1))

> lines(power.seq.n2~seq(0.6,0.75,0.005), col="blue")

> lines(power.seq.n3~seq(0.6,0.75,0.005), col="green")

> legend("bottomright", c("n = 80", "n = 130", "n = 180"), col =
c("red","blue","green"), text.col = c("red","blue","green"),
lty=c(1,1,1))
```

