DSC 152: Applied Statistical Data Analysis and Inference

Lecture #3
One-sample Nonparametric tests
and Type I Error rates

Tuesday, April 7
Spring Quarter 2026
Peter Chi

Last time

- We saw that running a t-test on a small sample from a non-normal distribution may suffer in terms of its Type I Error rate

- We learned how to obtain simulated estimates of Type I Error rates

And we said:

Things still to come:

- Non-parametric tests (e.g., but not limited to, what you've learned in DSC 10 and 80 where you get a p-value via simulating the null distribution) are less dependent on conditions to be valid.

- What about statistical power?

Non-parametric one-sample tests

Recall. . .

The t-Test relies on $\overline{x}$ following a normal distribution in order for its test statistic t_s to follow the t-Distribution under H_0.

Question: how might we perform a non-parametric test for $H_0 : \mu = \mu_0$?

The main thing is that it needs to require no conditions on the distribution of values (e.g., in the UCSD students' sleep example from last time, a non-parametric test would presume that the distribution on the number of hours of sleep per night could be anything)

Non-parametric one-sample tests

Consider:

- In some sense, the most true that H_0 could possibly be is if every data value equaled μ_0 (in this case, 6 hours)

- But, how else might the data look and still be in line with H_0?

pollev.com/chi

The Sign Test

One common example of a one-sample non-parametric test is known as the sign test. How does this work?

Sign test

- Take each data value and subtract μ_0^{***}
  - If the difference is positive, assign a "+" value
  - If the difference is negative, assign a "-" value
  - If the difference is 0, ignore that value
- Under H_0, there should be 50% "+" and 50% "-". We then calculate the probability of observing at least as many "+" or "-" as what was actually observed, under a Binomial distribution with $p = 0.5$.

***Almost. See next slide for correction.

The Sign Test

Question: Why is the sign test not actually a test for μ?

Answer:

It works by assuming that, under H_0:

- 50% of all values are above the null value
- and 50% of all values are below the null value.

But, a mean does not always actually have this property. What is the name of the thing that does??

The Sign Test

Question: Why is the sign test <u>not</u> actually a test for μ?

Answer:

It works by assuming that, under H_0:

- 50% of all values are above the null value
- and 50% of all values are below the null value.

But, a mean does not always actually have this property. What is the name of the thing that does??

So instead of μ_0, we will designate the null hypothesis value as $\tilde{\mu}_0$ to represent the null hypothesis value of the <u>median</u>.

The Sign Test

Example: recall the UCSD sleep data:

$$3, 7, 1, 2, 2$$

and $\tilde{\mu}_0 = 6$. So, our data for the sign test become:

$$-, +, -, -, -$$

pollev.com/chi

(Don't look at the next slide yet!)

The Sign Test

The p-value for a sign test is thus:

$$P(X = 4) + P(X = 5) + P(X = 0) + P(X = 1)$$

where $X \sim Binomial(n = 5, p = 0.5)$. In R, this is:

```
dbinom(x=4, size=5, prob=0.5) + dbinom(x=5, size=5, prob=0.5) +
  dbinom(x=0, size=5, prob=0.5) + dbinom(x=1, size=5, prob=0.5)
```

```
## [1] 0.375
```

or equivalently,

```
pbinom(1, size=5, prob=0.5)*2 # Wait why?
```

```
## [1] 0.375
```

So now we would fail to reject H_0. And furthermore, note that this p-value is way bigger than what we found via the t-Test last time!

The Sign Test

Question: what is the Type I Error rate of the sign test under a variety of situations?

Last time, with the t-test at $\alpha = 0.05$ and $n = 5$, we saw that...

- If the data follow a Gamma distribution, the Type I Error rate was approximately 0.10 - 0.11
- If the data follow a uniform distribution, the Type I Error rate was approximately 0.06 - 0.07
- If the data follow a normal distribution, the Type I Error rate was approximately the nominal 0.05 level

How does the sign test do in these scenarios?

The Sign Test

One problem: with $n = 5$, the sign test cannot actually do a 0.05-level test.

Daily Check Question (answer to be written in your Rmd file):

Why can't it?

Your Turn #1

First, in your R Markdown file, write a function to perform the sign test for a sample of size $n = 5$ and $\alpha = 0.0625 + \epsilon$.

- The function should take a vector of 5 values as its input, and return the (two-sided) p-value according to the sign test.
- You may hardcode the null hypothesis value of 6.
- Your function may ignore ties (since we will be simulating from continuous distributions, there is a probability of 0 that any simulated value will exactly equal 6).
- Using the `pbinom` function in some manner will likely be the easiest way to do it.

Your Turn #1

Now, here is the code from last time to estimate the Type I Error rate of the t-Test with Gamma-distributed data:

```
count <- 0

for(i in 1:10000){
  gam_data <- rgamma(n=5, shape=1.2, scale=(6/1.2))
  p.val <- t.test(gam_data, mu=6)$p.value
  if(p.val < 0.05){
    count <- count + 1
  }
}

TypeI <- count / 10000
TypeI
```

Your Turn #1

- Edit the code on the previous slide to produce a simulated estimate of the Type I Error rate of the sign test for Gamma-distributed data with $n = 5$ at $\alpha = 0.0625 + \epsilon$.
  - Note: to obtain a median of 6, we actually need to edit the `scale` value to be equal to 6.757 (the details of why are way beyond the scope of this course)

- Then, repeat with normal distribution with $\mu = 6$ and $\sigma = 1$ like last time.

- Repeat again with the uniform distribution from 2 to 10 (also from last time)

Comment briefly on what you observe, specifically on whether each estimate of the Type I Error rate is as expected or not.

Bootstrapped Confidence Intervals

In DSC 10, we learned how to construct bootstrap confidence intervals, with Python code looking like this:

```
boot_means = np.array([])

sleep = [3, 7, 1, 2, 2]

for i in range(10000):

    # Resample from my_sample WITH REPLACEMENT and compute the mean.
    mean = np.random.choice(sleep, size=5, replace=True).mean()

    # Store it in our array of means
    boot_means = np.append(boot_means, mean)
```

Bootstrapped Confidence Intervals

and then the 95% bootstrap confidence interval is:

```
left = np.percentile(boot_means, 2.5)
right = np.percentile(boot_means, 97.5)
[left, right]
```

```
## [1.6, 5.0]
```

Bootstrapped Confidence Intervals

Here is R code to do the same thing:

```
sleep <- c(3, 7, 1, 2, 2)
boot_means <- NA

for(i in 1:1000){
  boot_means[i] <- mean(sample(sleep, replace=TRUE))
}

quantile(boot_means, probs=c(0.025, 0.975))
```

```
##  2.5% 97.5%
## 1.600 5.005
```

(there is randomness, so the confidence intervals from R and Python may or may not match, but they are in principle from exactly the same procedure)

Bootstrapped Confidence Intervals

We also learned in DSC 10 that a confidence interval can be used to perform a hypothesis test:

Inverting the confidence interval

- CI does not contain $\mu_0 \Leftrightarrow$ reject H_0
- CI contains $\mu_0 \Leftrightarrow$ fail to reject H_0

While we don't exactly get a p-value from this procedure, we do get the decision of whether to reject or fail to reject H_0 at any α level (e.g. a 95% confidence interval corresponds to $\alpha = 0.05$).

Now, what is the Type I Error rate of this procedure?

Like before, let us estimate the Type I Error rate under three scenarios:

- Gamma-distributed data (the original one that had a mean of 6)
- Normally distributed data with $\mu = 6$ and $\sigma = 1$.
- Uniformly distributed data from 2 to 10 (also from last time)

In each case, we will use $n = 5$ again. And here, we can actually go back to doing a test of the mean μ (instead of the median like the sign test does).

Your Turn #2

Let's do it for the Gamma-distributed data together. Here's some partial code:

```
count <- 0

# The outer loop is so that it simulates 1000 random sets of data
for(j in 1:1000){
  gam_data <- _______
  boot_means <- NA

  # This is the bootstrap on each dataset
  for(i in 1:1000){
    boot_means[i] <- ______
  }

  ci <- _______
  if(______){
    count <- count + 1
  }
}

TypeI <- _______
```

Your Turn #2

Then, on your own, modify the code to estimate the Type I Error rate under the normal and uniform cases.

Recap

- We evaluated the Type I Error rates of the sign test and the bootstrap hypothesis test via simulation
  - The sign test always gives a proper α-level test without any distributional conditions necessary, but:
    - It is a test of the median (not the mean)
    - Not every desired α-level is possible due to the discrete nature of the rejection regions
  - The bootstrap hypothesis test is (surprisingly) not good!
    - We will investigate this further in Lab 2.

Still to come

- Statistical power!
- More complex statistical models

Daily check for today

Upload your R Markdown pdf output file to Gradescope, consisting of:

1. The answer to the question on Slide 10: Why can't a sign test do a 0.05-level test with $n = 5$?

2. Your Turn #1 from Slides 11 - 13 consisting of the simulations to estimate Type I Error rate from the sign test for the three cases (Gamma, normal, uniform).

3. Your Turn #2 from Slides 19 - 20 consisting of the simulations to estimate Type I Error rate from the bootstrap hypothesis test for the three cases (Gamma, normal, uniform).