Chapter 1: A Taste of Likelihood

S352: Data Modeling and Inference

Department of Statistics, Indiana University

What you hopefully learned about inference in STAT-S 350

- Plan to collect data random(ish)ly from some population
- Decide on one or more **parameters** of the population to estimate, e.g. the population mean
- Work out a **statistic** or **estimator** to estimate the parameter from the data, e.g. the sample mean
- Model the estimator as a **random variable**, e.g. by using the Central Limit Theorem to model the sample mean as having an approximately Normal distribution
- Get data and do the estimation
- Use probability to find standard errors and confidence intervals and/or do hypothesis tests, if you wish

But is your estimator any good?

What might we want out of an estimator?
Roughly, it seems like these things would be nice:

- On average, you get the right answer ("unbiased")
- If you get enough data, you get the right answer ("consistent")
- Your estimator gets at least as close to the right answer as any other estimator, on average ("efficient")

(Note that it may not necessarily be desirable, or even possible, for one estimator to achieve all these properties...)

What is “likelihood”?

Example: A survey

Question: "Do you like stinky tofu?"
Population (data): answers from US population
Parameter p: the proportion of the population who would answer "yes" if they were asked the question. The true value of p is denoted as p_0, which is a **fixed** number between 0 and 1, and is **unknown** to us.

To help estimate p_0, I give a survey asking this question to an **IID** sample of 100 people from the above population.

- "IID": Independent and identically distributed. Every time I sample a new person, I have the same chance p_0 of getting someone who'll answer "yes", regardless of what the people I've already sampled have answered.

  Note: This is literally true for a random sample with replacement, and approximately true for a sample without replacement from a large population.

Estimators

What are some good and bad estimators of p_0?

1. Just assume it's zero; no one likes stinky tofu
2. Take the **sample proportion**: the number of yeses over the sample size
3. Take the sample proportion, and double it
4. For each person in the survey, write down "0" if they said no and "1" if they said yes, and take the mean
5. For each person in the survey, write down "0" if they said no and "1" if they said yes, and take the median

The Binomial Distribution

- Let X be a discrete random variable representing the number of people in our survey (out of 100) who answer "yes."
- Under our assumptions, X will have a Binomial distribution, with parameters $n = 100$ and $p = p_0$.
- Recall that the **probability mass function (PMF)** of a **Binomial**(n, p) random variable is

$$f(x) \equiv P(X = x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

for $x = 0, 1, \ldots, n$.

- For our example, that becomes

$$f(x) = \frac{100!}{x!(100-x)!} p_0^x (1-p_0)^{100-x}$$

- What does $f(x)$ mean in this context? (Is it a probability? Compare to p_0.)
- Can we compute $f(x)$? (Probability vs Statistical Inference)

What would the probability have been?

Now suppose we do the survey, and find that 10 people out of the 100 answer "yes." We still don't know p_0... But we could guess that $p_0 = 0.1$. If so, the probability of getting 10 yeses would've been

$$f(10)_{p_0=0.1} = \frac{100!}{10!(100-10)!}\, 0.1^{10}\,(1-0.1)^{100-10}$$

You can get this in R:

```
> choose(100, 10) * .1^10 * .9^90
[1] 0.1318653
```

```
> dbinom(10, 100, 0.1)
[1] 0.1318653
```

(Is this high? Low? It's hard to tell without context...)

A slightly different p

What if we guessed that $p_0 = 0.09$?
Then the probability of getting 10 yeses would've been

$$f(10)_{p_0=0.09} = \frac{100!}{10!(100-10)!}\, 0.09^{10}\,(1-0.09)^{100-10}$$

In R:

```
> dbinom(10, 100, 0.09)
[1] 0.1242955
```

which is not too different.

So in some sense, $p_0 = 0.09$ is almost as believable as $p_0 = 0.1$.

A very different p

What if we guessed that $p_0 = 0.8$?
Then the probability of getting 10 yeses would've been

$$f(10)_{p_0=0.08} = \frac{100!}{10!(100-10)!}\, 0.8^{10}\,(1-0.8)^{100-10}$$

In R:

```
> dbinom(10, 100, 0.8)
[1] 2.300935e-51
```

which is a tiny number.

So $p_0 = 0.8$ is not really believable compared to $p_0 = 0.1$ or 0.09.

As a function

Thanks to R, we can try a whole bunch of possible values of p_0:

1. Create a vector p of possible values of p_0.
2. For each p, find the probability of getting 10 yeses in 100 trials.
3. Plot this probability as a function of p (as a line graph.)

The likelihood function

Definition of the **likelihood function** (discrete case):

> *The likelihood function, denoted $L(\theta|\mathbf{x})$ or $L(\theta;\mathbf{x})$ or $L(\theta)$, gives the probability of obtaining the observed data $X = (x_1, x_2, \ldots, x_n)$, as a* **function** *of the parameter or parameters θ of the statistical model.*

For IID data where each observation has PMF $f(x|\theta)$ for parameter(s) θ, this becomes:

$$\begin{aligned} L(\theta|\mathbf{x}) &= f(x_1|\theta) \times f(x_2|\theta) \times \cdots \times f(x_n|\theta) \\ &= \prod_{i=1}^{n} f(x_i|\theta) \end{aligned}$$

We often just call this the **likelihood**. But it's vital to remember it's a function of the parameter(s)!

Things to know about the likelihood function

- It's not a PMF or PDF!
- It's not a Normal curve! (It's not even symmetric!)
- The magnitude of the numerical values of the likelihood function depend a *lot* on sample size.
  - For example, we previously saw that with 10 yeses in a survey of 100 people, the maximum value of the likelihood was `dbinom(10, 100, 0.1) = 0.13`.
  - If we got 1000 yeses in a survey of 10,000 people, the maximum value of the likelihood would be `dbinom(1000, 10000, 0.1) = 0.013`.

  This issue gets much worse with more complicated models. It means that it's very hard to interpret the numerical value of the likelihood in isolation (but that's almost never the goal...)

Some philosophy

- **The likelihood is a function of the parameter(s).**
- It treats the data as fixed. That is, other possible values of the data are not directly considered when calculating the likelihood.
- Frequentist likelihood methods do *not* put probabilities or distributions on possible values of the parameters. Rather, they aim to select one possible parameter value, or an interval of possible values, in a way that has "good properties."

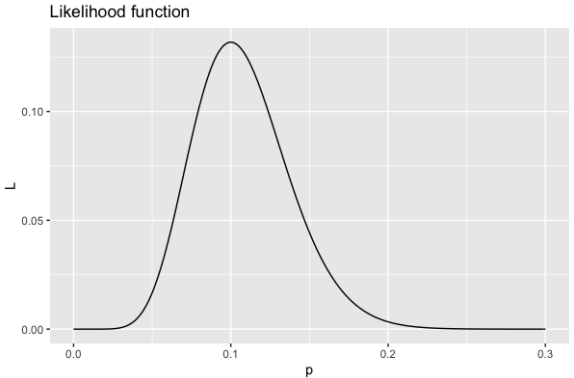
  (If you don't like this, you can become a Bayesian, but that requires a bit more math and computing. . .)

Maximum likelihood estimation

If you're trying to estimate the value for the parameter, it seems like a good guess would be the value that *maximizes* the likelihood function.

So if the likelihood function is maximized when $p = 0.1$, the **maximum likelihood estimate** (MLE) of the true value p_0 is 0.1.

(Is that it? Is all of statistics this simple?)



MLE and Binomial/Bernoulli Data

Yes/no data: Bernoulli and Binomial

There are two main ways of writing down the results of an IID yes/no survey (or coinflips or whatever) in terms of random variables:

- Consider the results as a sequence of zeroes and ones, e.g.: 0, 0, 1, 0, 0, 0, ... The survey results can be conceptualized as **a sequence of IID Bernoulli(p) random variables,** $X_1, X_2, \ldots, X_n$. Each X_i has probability p of being 1 and probability $1 - p$ of being 0.
- Count the number of yeses/heads/ones in the survey "We got 10 yeses out of 100 people". This can be conceptualized as **one random variable Y with a Binomial(n, p) distribution**.

(But does this throw away important information?)

The likelihood function for IID Bernoulli data

$$L(p|x) = f(x_1|p) \times f(x_2|p) \times \cdots \times f(x_n|p)$$

Now, $f(x_i|p)$ is p if the ith observation is a yes, and $(1-p)$ if it's no.
So the likelihood will just be the product of a bunch of p's and $(1-p)$'s.
If there are y yeses and $(n-y)$ nos, the likelihood will be:

$$L(p|x) = p^y \cdot (1-p)^{n-y}, \qquad \text{where } y = \sum x_i$$

To maximize this, differentiate with respect to p, set to 0, and solve (and check it gives a maximum.) Still remember the Chain Rule and Product Rule?

The likelihood function for the Binomial data

$$L(p|y) = \binom{n}{y} p^y \cdot (1-p)^{n-y}$$

This is almost the same—the only difference is a multiplicative constant (i.e. it doesn't have p in it) in front. (Does this matter?)

The log-likelihood function

- The likelihood function is awkward to work with because it can vary across many orders of magnitude.
- Instead, we usually use the (base e) *log* of the likelihood function, which we unimaginatively call the **log-likelihood** function.
- A crucial point is the parameter value that maximizes the likelihood function also maximizes the log-likelihood. Further, the log-likelihood is usually much easier to deal with numerically. (There are also a bunch of theoretical reasons to prefer it.)

The log-likelihood function for IID data

The likelihood function once again:

$$L(\theta|x) = \prod_{i=1}^{n} f(x_i|\theta)$$

Recall that **the log of a product is the sum of the logs**. So the log-likelihood function $l(\theta|x)$ is:

$$l(\theta|x) = \log L(\theta|x) = \sum_{i=1}^{n} \log f(x_i|\theta)$$

This is great news! Sums are much easier to do calculus on than products!

The log-likelihood function for IID Bernoulli data

$$L(p) = p^y \cdot (1-p)^{n-y}$$

Remembering **properties of logs**, in particular that $\log(a^b) = b \log a$, the log-likelihood is

$$\begin{aligned} l(p) = \log L(p) &= \log\left(p^y \cdot (1-p)^{n-y}\right) \\ &= \log\left(p^y\right) + \log\left((1-p)^{n-y}\right) \\ &= y \log p + (n-y) \log(1-p) \end{aligned}$$

Derivative of the log-likelihood for IID Bernoulli data

Now we have to remember how to take derivatives of logs:

$$\frac{d}{dp}\log p = \frac{1}{p}, \quad \frac{d}{dp}\log(1-p) = -\frac{1}{1-p}$$

So

$$\begin{aligned} l(p) &= y\log p + (n-y)\log(1-p) \\ \frac{d}{dp}l(p) = l'(p) &= y\frac{1}{p} + (n-y)\frac{-1}{1-p} \\ &= \frac{y(1-p) - (n-y)p}{p(1-p)} \\ &= \frac{y - yp - np + yp}{p(1-p)} \\ &= \frac{y-np}{p(1-p)} \end{aligned}$$

Maximizing the log-likelihood for IID Bernoulli data

To find the value $\hat{p}$ that (possibly) gives the maximum, set $l'(p)$ to zero and solve.

$$\frac{y - n\hat{p}}{\hat{p}(1-\hat{p})} = 0$$

For the LHS to be zero, we require

$$\begin{aligned} y &= n\hat{p} \\ \hat{p} &= \frac{y}{n} \end{aligned}$$

So the maximum likelihood estimator of p appears to be the number of yeses divided by the sample size: i.e. just the usual sample proportion. That's reassuring!

Checking that we got a maximum

But wait! A point where the derivative is zero could be a maximum, but it also could be a minimum, or an inflection point.

We can do a formal check to verify that we got a maximum, i.e., check $l''(p)$.

Or we can just plot the log-likelihood in R and look at it...

The log-likelihood function for the Binomial

$$
\begin{aligned}
L(p) &= \binom{n}{y} p^y \cdot (1-p)^{n-y} \\
l(p) &= \log\left(\binom{n}{y} p^y \cdot (1-p)^{n-y}\right) \\
&= \log\binom{n}{y} + \log\left(p^y\right) + \log\left((1-p)^{n-y}\right) \\
&= \log\binom{n}{y} + y\log p + (n-y)\log(1-p)
\end{aligned}
$$

(Spot the difference between this and the Bernoulli case...)

Maximizing the log-likelihood for the Binomial

$$l(p) = \log \binom{n}{y} + y \log p + (n - y) \log (1 - p)$$

The only difference between this and the Bernoulli log-likelihood is the constant $\log \binom{n}{y}$. But the derivative of a constant is zero...

Both the Bernoulli log-likelihood and the Binomial log-likelihood will be maximized at exactly the same value of p. That is, the MLE is $\hat{p} = y/n$ in both cases.

- Sometimes constant terms in the log-likelihood aren't important, and sometimes they are. Until you get a good sense for them, it's best not to ignore them...

So. . . Bernoulli or Binomial?

- It doesn't matter whether you treat the data as IID Bernoulli or Binomial: you come to the same conclusion.
- Either way, all the useful information about p is contained in Y, the number of yeses.
  – If there are two yeses, it makes no difference whether they were the 1st and 3rd observations or the 29th and 82nd observations.
- We say that Y is a **sufficient statistic**.
  (See Millar Definition 14.1 for a formal definition of sufficiency. Take STAT-S 420 or STAT-S 621 to understand what it means.)

Likelihood and Confidence Intervals

Uncertainty

We know that any point estimate is unlikely to be exactly right. That's why we spent most of S350 doing inference that took account of uncertainty:

- standard errors
- confidence intervals
- hypothesis testing

We can still do all that stuff, right?

Standard errors

- The **error** of an estimator is its value minus the true parameter value.
  For the MLE of p:
  $$\text{Error} = \hat{p} - p$$
- If we consider $\hat{p}$ as a random variable, then the error is also a random variable and so has a probability distribution, expected value, and standard deviation.
- The **standard error** of an estimator is the standard deviation of the distribution of the error:
  $$se(\hat{p}) = SD(\hat{p} - p)$$
  But in frequentist statistics, p is taken to be fixed. So
  $$se(\hat{p}) = SD(\hat{p})$$

Example: Binomial data

For Binomial data, the MLE for p is

$$\hat{p} = Y/n$$

where Y is the number of successes and n is the number of trials.

(What is the difference between Y and y?)

You found (or will find) that the standard error of $\hat{p}$ is

$$se(\hat{p}) = \sqrt{\frac{p(1-p)}{n}}$$

Since p is unknown, the estimated standard error of $\hat{p}$ is

$$\widehat{se}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Basic 95% confidence intervals

The basic kind of confidence interval you've previously seen, the **Wald interval**, is based on the assumption of a Normal distribution for the estimator.

For Binomial data, the MLE is $\hat{p}$ and a 95% confidence interval for p is

$$\hat{p} \pm 1.96 \times \widehat{se}(\hat{p})$$

or

$$\hat{p} \pm 1.96 \times \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Justification: Central Limit Theorem. For large n ....

Example: Binomial

Suppose we observe 10 successes in 100 trials.
MLE:

$$\hat{p} = \frac{10}{100} = 0.1$$

(estimated) Standard error:

$$\begin{aligned} \widehat{se}(\hat{p}) &= \sqrt{\frac{0.1(1-0.1)}{100}} \\ &= 0.03 \end{aligned}$$

A 95% Wald confidence interval is $0.1 \pm 1.96 \times 0.03$, or $(0.0412, 0.1588)$.
(What about a different confidence level?)

Is Normality justified?

We could try out the following simulation.

1. Suppose the true value of p really is 0.1.
2. Run 100 IID trials, each with probability 0.1 of success, and count up y, the number of success;
3. Repeat a few thousand times;
4. Assess the Normality of the y's, e.g. via a QQ plot.

Caveat: The true p might be somewhat off from 0.1, so it would be even better to try out a few different values...

Some possible issues

But wait! There are so many questions ...

1. What if the likelihood is difficult to write? (new distribution, complex model, ...)
2. What if it is hard to find the MLE(s)?
3. What if I do not know the standard error?
4. What if the Normality cannot be justified?

⋮

Likelihood and Optimization

How to solve for MLE numerically

Solving for the MLE numerically

Recall that the likelihood function (from the point of view of the binomial) was

$$L(p|y) = \binom{n}{y} p^y \cdot (1-p)^{n-y}$$

The general-purpose optimization functions are `optimize()`, for one-dimensional functions, and `optim()`, for more general functions.

- For numerical reasons, it's better to work with the log-likelihood than the likelihood itself.
- By default, these functions find the value that *minimizes* the function, so it's better to use the *negative* log-likelihood.

Using `optimize()` to find the MLE

- Write a function to evaluate the negative log-likelihood at p.
- The main arguments of `optimize()` are:
  - `f`: the name of the function you've optimizing
  - `interval`: a vector containing the minimum and maximum parameter values you're willing to consider, e.g. `c(0, 1)`

Writing a function

Suppose we wish to numerically find the value of x that minimizes $y = x^2$.
We can write a function to evaluate y at any given x:

```
square <- function(x){
    y <- x^2
    return(y)
}
```

Here `square` is the name of the function and `x` is its argument.

Using optimize

Let's suppose we're willing to consider values of x from -1000 to $+1000$.
Use `optimize()`:

```
> optimize(square, c(-1000, 1000))
$minimum
[1] 2.842171e-14

$objective
[1] 8.077936e-28
```

The minimum occurs at basically zero. The minimum value ("objective") is also basically zero.

R Exercise

1. Write an R function called `binom.nloglik()` with argument p to evaluate the negative log-likelihood at `p` when you observe 10 heads in 100 trials.
2. Use `optimize()` to find the value of `p` for which the negative log-likelihood is minimized.

Likelihood and Optimization

Let's try the idea on a "new" distribution

Example: Geometric distribution

Experiment: Toss a coin until you get a head. Suppose the probability of getting a head is p.
Random variable: Let X be the number of tails you get before the first head.
Distribution: Then X has a Geometric distribution with parameter p.

The PMF of X is:

$$f(x) = \begin{cases} (1-p)^x \, p & x = 0, 1, 2, \ldots \\ 0 & \text{otherwise.} \end{cases}$$

Geometric distribution: Log-likelihood

Suppose we toss the coin and get 3 tails, then a head.
The likelihood is

$$L(p) = (1-p)^3 p$$

The log-likelihood is

$$\begin{aligned} l(p) &= \log\left[(1-p)^3 p\right] \\ &= \log\left[(1-p)^3\right] + \log p \\ &= 3\log(1-p) + \log p \end{aligned}$$

Negative log-likelihood

Write a function to find the negative log-likelihood when $x = 3$.

```
geom.nloglik <- function(p){
    nloglik <- -3 * log(1 - p) - log(p)
    return(nloglik)
}
```

Draw a picture:

```
curve(geom.nloglik, from = 0, to = 1)
```

Geometric MLE

Use `optimize()` to find the MLE:

```
> optimize(geom.nloglik, c(0, 1))
$minimum
[1] 0.2500143

$objective
[1] 2.249341
```

Seems like the MLE is 0.25. (Does that make sense?)

Multiple observations

Suppose we had two independent observations, $x_1 = 3$ and $x_2 = 2$.
Add the log-likelihoods:

$$l(p) = 3\log(1-p) + \log p + 2\log(1-p) + \log p$$

or

$$l(p) = 5\log(1-p) + 2\log p$$

MLE with multiple observations

```
geom.nloglik2 <- function(p){
    nloglik1 <- -3 * log(1 - p) - log(p)
    nloglik2 <- -2 * log(1 - p) - log(p)
    nloglik <- nloglik1 + nloglik2
    return(nloglik)
}
optimize(geom.nloglik2, c(0, 1))
```

Likelihood and Optimization

optim(), hessian, and standard error

Using `optim()` to find the MLE

`optim()` is a more flexible function that can find minima with respect to multiple variables. It's a bit more annoying to use, however.

- Write a function to evaluate the negative log-likelihood at p.
- The main arguments of `optim()` are:
  - `par`: initial values for the parameters
  - `fn`: the name of the function you've optimizing
  - `method`: the name of the algorithm used to find the optimum. For one parameter problems, use `"Brent"`.
  - `lower` and `upper`: the minimum and maximum parameter values you're willing to consider
  - `hessian`: Do you want to return the estimated Hessian matrix (`TRUE` or `FALSE`)? In statistics, the Hessian turns out to be extremely useful, so set this to `TRUE`.

Using optim()

```
> square <- function(x){
+     y <- x^2
+     return(y)
+ }
> optim(par = 1, fn = square, method = "Brent",
lower = -1000, upper = 1000, hessian = "TRUE")
$par
[1] 2.842171e-14

$value
[1] 8.077936e-28

$hessian
[,1]
[1,]    2
```

Understanding the Hessian

In one dimension:

- The Hessian is (an estimate of) the second derivative at the supposed minimum. Recall that a *positive* second derivative at an optimum indicates it's a minimum point.
- Quite magically, when working with the negative log-likelihood, the Hessian is the reciprocal of the variance of the MLE.
  So the estimated standard error is

$$\widehat{se}(\hat{\theta}) = \frac{1}{\sqrt{\text{Hessian}}}$$

Likelihood and Confidence Intervals

using R packages

Confidence intervals

There are a *lot* of alternative ways to make a confidence interval for p.

Two useful R packages:

- `binom`: lots and lots of methods for binomial data
  - `binom.profile()`
- `Bhat`: likelihood-based methods for a variety of distributions
  - `plkhci()`

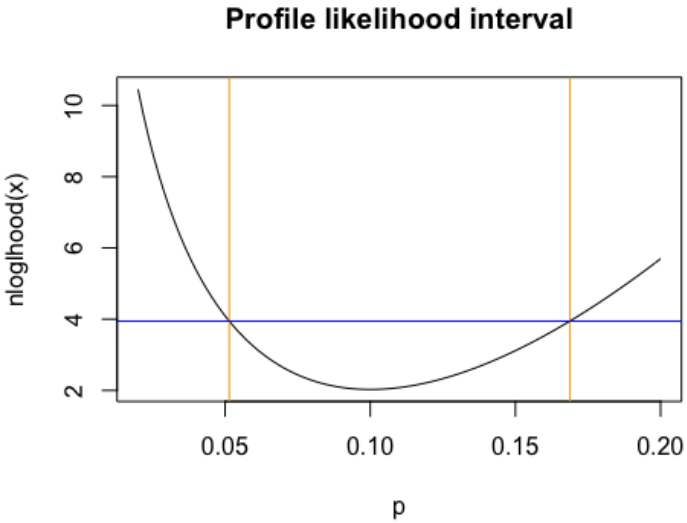
Let's try `binom` first...

binom.profile()

We want to create an interval in the following way: Find all values of p for which the log-likelihood exceeds some threshold, where the threshold is chosen so that the interval will have approximately 95% coverage. (More technical details in chapter 3.)

```
> library(binom)
> binom.profile(10, 100)
method  x   n mean      lower    upper
1 profile 10 100  0.1 0.05142278 0.168782
> binom.profile(10, 100)["lower"]
lower
1 0.05142278
> binom.profile(10, 100)["upper"]
upper
1 0.168782
```

Profile likelihood



The `plkhci()` function

This is a bit trickier to use. The arguments are:

- `x`: a **list** with the following components
  - `label`: the name of the parameter
  - `est`: the initial parameter estimate
    - **use the MLE**
  - `low`: a lower bound for the parameter
  - `upp`: an upper bound for the parameter
- `nlogf`: the negative log-likelihood function
- `label`: the parameter to find the CI for (in case there's more than one)

Using the `plkhci()` function

Let's write a function `binom.nloglik(p)` to evaluate the binomial negative log-likelihood at p when you get 10 heads in 100 tosses.:

```
binom.nloglik <- function(p){
    nll <- -log(choose(100, 10)) - 10 * log(p) - 90 * log(1 - p)
    return(nll)
}
```

Using the `plkhci()` function

We use the MLE $\hat{p} = 0.1$ as our initial estimate of p.

```
> library(Bhat)
> control.list = list(label = "p", est = 0.1, low = 0, upp = 1)
> plkhci(control.list, binom.nloglik, "p")

...

[1] 0.05141279 0.16877909
```

Using the `plkhci()` function

The main advantage of `plkhci()` is that you can use it for other distributions.

For our geometric example with $x_1 = 3$ and $x_2 = 2$:

```
control.list = list(label = "p", est = 2/7, low = 0, upp = 1)
plkhci(control.list, geom.nloglik2, "p")
```

I get an interval from 0.054 to 0.65 for p.

Inference redux

- Large-sample inference, like Wald confidence intervals, works well when you have large samples (and the population distribution isn't too skewed and doesn't have horrendous outliers.)
- If that's not the case, there are plenty of alternatives. In particular, if you can write down a (credible) likelihood, likelihood methods are hard to beat.
- If you can't write down a likelihood, there are always methods like the bootstrap or nonparametric tests. Take S425/625 to learn more...