

Chapter 3: Hypothesis Tests and Confidence Intervals

S352: Data Modeling and Inference

Department of Statistics, Indiana University

Statistical inference

Whenever we estimate a parameter (θ), there's always uncertainty—our estimate ($\hat{\theta}$) could be higher or lower than the true parameter value.

In frequentist statistics, ways of dealing with this uncertainty include:

- Estimating the bias/standard error/mean squared error etc. of our estimator
 - **Bias** = $E(\hat{\theta}) - \theta$.
 - **Variance** = $\text{Var}(\hat{\theta})$. (The standard error is the square root of this.)
 - **Mean squared error (MSE)** = $E[(\hat{\theta} - \theta)^2]$. This may be the best measure of “overall” error.
- Confidence intervals
- Hypothesis tests

) Review : 5350 , Ch 9

MLE and inference

There are two classical ways to do inference using maximum likelihood estimates.

- **Wald inference:** Approximate the distribution of the MLE with a Normal distribution.
- **Likelihood ratio inference:** Use something called the “likelihood ratio test statistic.”

large
 n !

In addition, there are more modern computationally-intensive methods such as the bootstrap.

Wald inference is the easiest, so that's what we'll start with.

Approximate Normality of MLEs

Wald inference for the mean (μ)

From intro stat (S350), you're already used to doing Wald inference for the mean:

- Estimate the population mean μ by taking a very big IID sample and finding the sample mean \bar{x} . (point est.)

- The **Central Limit Theorem** says that with big samples, the sample mean has an approximately Normal distribution with expected value μ , variance σ^2/n , and standard deviation σ/\sqrt{n} . $\bar{X} \approx N(\mu, \sigma^2/n)$

$$E(\bar{X}) = \mu, \text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

- The error $\bar{X} - \mu$ thus follows an approximately Normal distribution with expected value 0, variance σ^2/n , and standard deviation σ/\sqrt{n} . We can use this to construct confidence intervals and hypothesis tests. $\bar{X} - \mu \approx N(0, \sigma^2/n)$

- Oh wait, we usually don't know the true population σ . Well, as long as the sample size is very large, we can use the sample SD instead and probably nothing bad will happen.

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \approx N(0, 1), \quad \text{if } \sigma \text{ unknown, } \frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}} \approx N(0, 1)$$

When does this work?

The previous slide's inference for μ relied on the large sample assumption for two reasons:

- Using the Central Limit Theorem
- Using s instead of σ

In cases where the sample size is small or moderate, it may be necessary to make adjustments, e.g. using a t -distribution instead of a Normal.

Or the method might fail totally, for example if the population is heavily skewed (as well as in pathological cases like infinite variance.)

MLE Normality Theorem: One parameter case

We now want to extend this kind of inference for the mean to a more general form of inference for (almost) all MLEs. Suppose we wish to estimate a single parameter θ . Let the MLE of this parameter be $\hat{\theta}$.

Suppose we have a sufficiently ^{*n big!*} large sample and nothing extremely weird is going on.¹ Then (after doing a lot of math; see ch. 12 of Millar if you care or are applying to Ph.D. programs soon), we find that $\hat{\theta}$ has an approximately Normal distribution centered at the true parameter value θ_0 .

$$\hat{\theta} \approx N(\theta_0, \star)$$

This in turn implies the MLE is “asymptotically unbiased”: it might be biased in the finite sample case, but the bias goes to zero as n goes to infinity. $E(\hat{\theta}) - \theta_0 \rightarrow 0$.
 $E(\hat{\theta}) \rightarrow \theta_0, n \rightarrow \infty$

¹Fancy statisticians say “under appropriate regularity conditions,” but this basically means the same thing.

Complications

There are a couple of complications to sort through before we can use the MLE Normality Theorem to actually do inference.

- We need to find the variance/standard error of the MLE.

$$V(\hat{\theta}) = ?$$

$$se(\hat{\theta}) = ?$$

- Expression for the standard error often depend on the true parameter value θ_0 , and the whole point is that θ_0 is unknown.

Sometimes we can address these by math

Example: Bernoulli/Binomial distribution

Suppose we're trying to estimate the proportion p of a population that have some characteristic.

$$X_i \stackrel{\text{iid}}{\sim} \text{Ber}(p), i=1, \dots, n$$

We'll gather a sample of n IID Bernoulli(p) observations, with n large. Let Y be the number in the sample who have the characteristic.

$$Y = \sum_{i=1}^n X_i \sim \text{Bin}(n, p)$$

Our MLE is $\hat{p} = Y/n.$ $= \frac{\sum X_i}{n} = \bar{X}$

Then by the MLE Normality Theorem, \hat{p} has an approximately Normal distribution with expected value p .

(Did we know this already?)

$$\hat{p} \approx N(p, \text{Var}(\hat{p})) \leftarrow \begin{array}{l} \text{by CLT} \\ \text{or} \\ \text{MLE} \\ \text{normality} \end{array}$$

Example: Bernoulli/Binomial distribution

To do Wald inference on p , we need to know the variance/SE of \hat{p} .

Fortunately you already found this in Problem Set 1! The variance and standard error of \hat{p} is

$$V = \frac{p(1-p)}{n}, \quad \text{se} = \sqrt{\frac{p(1-p)}{n}}$$

If we don't know what the true p is, as long as n is very large, we can estimate the SE as

$$\hat{V} = \frac{\hat{p}(1-\hat{p})}{n}, \quad \hat{\text{se}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

used in CI !

What if there is one hypothesized or null value for p ?

Which standard error?

$$H_0: p = p_0$$

- When you're doing a hypothesis test, you have some null hypothesis value p_0 for the proportion. Then it's better to use this value to calculate the standard error for the purpose of finding a test statistic and eventually a P -value.

$$se = \sqrt{\frac{p_0(1 - p_0)}{n}}$$

use hypothesized parameter value for θ .

- When you're finding a confidence interval, you don't need to have a particular hypothesis in mind. Then you should use the \hat{se} , i.e. the formula with \hat{p} 's in it.

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

use $\hat{\theta}$

(Does this seem kind of inconsistent? If you think so, that might be a good reason to learn alternatives to Wald inference...)

Observed Fisher information

Recall that in the one-parameter case, the Hessian is the second-derivative of the log-likelihood function at its maximum, or the second-derivative of the negative log-likelihood function at its minimum, depend on which textbook you're reading.

To avoid ambiguity, let's work with the **observed Fisher information**. This is the second-derivative of the log-likelihood function at its maximum. This means that (when it exists) it's always negative.

$$\mathcal{I}(\hat{\theta}) = l''(\hat{\theta}) < 0$$

A good estimate of the variance of the MLE is minus one over the observed Fisher information:

$$\hat{V}(\hat{\theta}) = -\frac{1}{l''(\hat{\theta})}$$

Example: Observed information for the Binomial

The Binomial log-likelihood is

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

Take the first derivative with respect to p :

$$l'(p) = 0 + \frac{y}{p} - \frac{n - y}{1 - p}$$

Take the second derivative:

$$l''(p) = -\frac{y}{p^2} - \frac{n - y}{(1 - p)^2} < 0, \quad \hat{p} = \frac{y}{n} \text{ is MLE.}$$

Example: Observed information for the Binomial

After a bunch of algebra, this simplifies to

$$I''(p) = -\frac{y - 2yp + np^2}{p^2(1-p)^2}$$

To find the observed information, substitute \hat{p} for p . Since $\hat{p} = y/n$, also substitute $n\hat{p}$ for y .

$$\mathcal{I}(\hat{p}) = -\frac{n\hat{p} - 2n\hat{p}^2 + n\hat{p}^2}{\hat{p}^2(1-\hat{p})^2} = -\frac{n\hat{p} - n\hat{p}^2}{\hat{p}^2(1-\hat{p})^2} = -\frac{n\hat{p}(1-\hat{p})}{\hat{p}^2(1-\hat{p})^2} = -\frac{n}{\hat{p}(1-\hat{p})}$$

So the estimate of variance is the same as the one we found manually:

$$\hat{V}(\hat{p}) = -\frac{1}{\mathcal{I}(\hat{p})} = \frac{\hat{p}(1-\hat{p})}{n}$$

One Parameter Wald Inference

tests and intervals for one parameter

Using the Normal MLE Theorem

Suppose we have a large IID sample for size n . Then:

- We can find the MLE, $\hat{\theta}$ (e.g. from theoretical derivation or by using `optim()` to minimize the negative log-likelihood)
- We can estimate the variance of $\hat{\theta}$ by taking the *observed information* $\mathcal{I}(\hat{\theta})$ (again, from theoretical derivation or from the “Hessian” in the output of `optim()`) and finding

Theory: $\hat{V}(\hat{\theta}) = -\frac{1}{\mathcal{I}(\hat{\theta})} = -\frac{1}{l'''(\hat{\theta})}$ (or $\frac{1}{\text{Hessian}}$) *computation*

- We can combine this with the approximation that $\hat{\theta}$ follows a Normal distribution centered at the true θ_0 to do inference.

(*)

$$\hat{\theta} \approx N(\theta_0, \hat{V}(\hat{\theta}))$$

$$\hat{se}(\hat{\theta}) = \sqrt{\hat{V}(\hat{\theta})}$$

Wald CI: $\hat{\theta} \pm (z_{\frac{\alpha}{2}} \cdot \hat{se}(\hat{\theta}))$

Wald test: $W = \frac{(\hat{\theta} - \theta_0)^2}{\hat{V}(\hat{\theta})}$, use $\chi^2_{(1)}$

The Wald test statistic(s)

We could use a Z -statistic to do hypothesis tests. Suppose our null hypothesis is $H_0 : \theta = \theta_0$. Then the Z -statistic has an approximate standard Normal($0, 1^2$) distribution.

$(H_a : \theta \neq \theta_0)$

$$Z = \frac{\hat{\theta} - \theta_0}{\sqrt{\hat{V}(\hat{\theta})}} \approx N(0, 1^2)$$

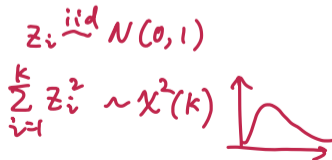
hypothesized value of θ .

For two-sided tests, it's common to use the square of Z as the test statistic (since then you don't have to worry about the sign.) The square of a standard Normal has a chi-squared distribution. Then the Wald statistic, W , has an approx. chi-squared distribution with one degree of freedom.

$$W = Z^2 = \frac{(\hat{\theta} - \theta_0)^2}{\hat{V}(\hat{\theta})} \approx \chi^2(1)$$

\otimes

We reject H_0 when W is big.



Toy example: Poisson

- Distribution: Poisson with rate parameter λ
- Goal: test the null hypothesis $H_0 : \lambda = 10$ ($H_1 : \lambda \neq 10$) $\lambda_0 = 10$ (hypothesized λ)
- Data: 100 observations X_1, \dots, X_{100} from this Poisson distribution $X_i \sim \text{Poi}(\lambda)$
 $i=1, \dots, n=100$
- Sample statistic: sample mean $\bar{x} = 9$.
- Previously we know:
 1. the ML estimator is $\hat{\lambda} = \bar{X}$. So the ML estimate is $\hat{\lambda} = \bar{x} = 9$. $\hat{\lambda} \approx N(\lambda, \text{Var}(\hat{\lambda}))$
 2. for a Poisson, $\text{Var}(X) = \lambda$.

Therefore, the variance of the MLE is (we don't need to find the Hessian/information here!)

$$\text{Var}(\hat{\lambda}) \stackrel{\text{Poi}}{=} \boxed{\text{Var}(\bar{X}) \stackrel{\text{iid}}{=} \frac{\text{Var}(X)}{n}} \stackrel{\text{Poi}}{=} \frac{\lambda}{n} \quad \text{always true} \quad \hat{\lambda} \approx N(\lambda, \frac{\lambda}{n})$$

We can just estimate the the variance of the MLE as: $\hat{V}(\hat{\lambda}) = \frac{\lambda_0}{n} = \frac{10}{100} = 0.1$.
(for test)

use hypothesized for test

Toy example: Poisson

Wald test

The Wald W statistic is

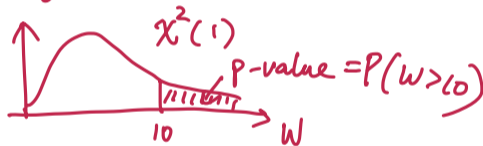
$$W = \frac{\hat{\lambda}^2 - \lambda_0^2}{V(\hat{\lambda})} = 10.$$

Using a $\chi^2(1)$ distribution, the P -value is:

$P(W \leq 10)$
> 1 - pchisq(10, df = 1)
[1] 0.001565402

If you instead wanted to do a (two-sided) Z -test:

```
> z <- (9 - 10) / sqrt(0.1)
> 2 * (1 - pnorm(abs(z)))
[1] 0.001565402
```



Real example: EPL soccer

Suppose we wish to test the hypothesis that the average number of goals scored in an English Premier League soccer game (by both teams combined) is 3.

$$H_0 : \mu = 3$$

$$H_1 : \mu \neq 3$$

$$n = 380$$

The spreadsheet ep1819.csv contains a bunch of information on the 380 games in the 2018-19 EPL season. Two of the variables are:

- FTHG: goals scored by the home team
- FTAG: goals scored by the away team

The games aren't IID, but they should be close enough.

Real example: EPL soccer

Find the mean goals per game that season:

```
TotalGoals <- epl1819$FTHG + epl1819$FTAG  
mean(TotalGoals)
```

I got an average of 2.82 goals per game that season. ($\bar{x} = 2.82$)

Now if you want, you can just do a test for the mean like you did in intro stat.



BUT suppose your soccer-loving friend tells you that the number of goals in a EPL game follows a Poisson distribution.

$$X_i \sim Poi(\lambda)$$

IF she's right, then it would be better to use that knowledge to construct your test.

Is it Poisson?

To do a rough check of whether the Poisson model is reasonable:

- Find the MLE $\hat{\lambda}$
- Simulate data from a $\text{Poisson}(\hat{\lambda})$ distribution
- See if it looks like the observed data

(There are more formal ways to check the Poisson assumption; we'll look at those later...)



Wald test: theory approach

As before, the variance of $\hat{\lambda}$ is λ/n .

If we have a null hypothesis value λ_0 , it's better to use that for testing purposes. So under the null, $\text{Var}(\hat{\lambda}) = 3/380 \approx 0.00789$.

Now we can do the test:

```
W <- (mean(TotalGoals) - 3)^2 / (3/380)
1 - pchisq(W, df = 1)
```

I get a W statistic of 4.06 and a P -value of 0.044. There's some evidence the average isn't 3. (So what is it? We probably want to follow up with a confidence interval...)



Wald test: computational approach

Let's use good old `optim()`:

```
goals.nll <- function(lambda){  
  lik <- dpois(TotalGoals, lambda)  
  return(-sum(log(lik)))  
}  
lambda.mle <- optim(par = 3, goals.nll, method = "Brent",  
                    lower = 0, upper = 999, hessian = TRUE)  
  
lambda.mle$par
```

This gave an MLE of 2.82 (as it should.)



Wald test: computational approach (cont'd)

```
> lambda.mle$hessian  
[,1]  
[1,] 134.7015
```

We can get a variance estimate from the Hessian:

```
> var.mle <- 1 / lambda.mle$hessian  
> var.mle  
[,1]  
[1,] 0.007423821
```

For tests, it's a bit better to use the theoretical value, but it doesn't matter much...



Wald test: computational approach (cont'd)

```
> W.comp <- (lambda.mle$par - 3)^2 / var.mle  
> 1 - pchisq(W.comp, df = 1)  
[,1]  
[1,] 0.03781257
```

The theory-based P -value is a bit more reliable, but we reach the same basic conclusion—there's some evidence that the mean isn't 3.

The benefit is that we now have the right variance/SE to use in a confidence interval:

```
> lambda.mle$par - qnorm(.975) * sqrt(var.mle)  
> lambda.mle$par + qnorm(.975) * sqrt(var.mle)
```

This gives (2.652179, 2.989926). Compare with results from `t.test`....



Summary of one parameter Wald inference

Suppose we're willing to assume a probability distribution for X and we can estimate the parameter θ using MLE.

- If you're doing a hypothesis test, it's better to calculate the variance of $\hat{\theta}$ using the null hypothesis value θ_0 and some theory (where possible.)
- If you're finding a confidence interval, it's better to get a variance estimate based on data (e.g. by inverting the Hessian.)

BUT! if you're just estimating the population mean, then unless you're pretty sure about what distribution the data comes from, it may be safer to just use STAT-S 350 stuff (e.g. your old buddy `t.test()`.)

MLE Normality Theorem for Two Parameters

Multiple parameters

Warning: Things are about to get heavy for a bit. . .

In many statistical models, we estimate multiple parameters at once.
e.g. Recall that with a gamma distribution, we need to estimate both a shape parameter and a rate parameter simultaneously.

If we then want to do inference, using Wald-type methods becomes messier (and there are more ways things can go horribly wrong. . .)

MLE Normality Theorem for two parameters

Now suppose we wish to estimate two parameters, θ_1 and θ_2 . Let the MLEs of these parameters be $\hat{\theta}_1$ and $\hat{\theta}_2$. Again suppose we have a sufficiently large sample and nothing extremely weird is going on.

Then the vector $(\hat{\theta}_1, \hat{\theta}_2)$ has an approximately **bivariate Normal distribution**, centered at (θ_1, θ_2) .

$$\begin{pmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{pmatrix} \approx N \left(\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \Sigma_{2 \times 2} \right)$$

Recall from STAT-S 350 (Trosset ch. 14) that this means:

- The distribution of $\hat{\theta}_1$ can be approximated by a Normal with expected value θ_1 ;
- The distribution of $\hat{\theta}_2$ can be approximated by a Normal with expected value θ_2 ;
- The distribution of any linear combination of $\hat{\theta}_1$ and $\hat{\theta}_2$ (e.g. $\hat{\theta}_1 + \hat{\theta}_2$ or $\hat{\theta}_1 - \hat{\theta}_2$) is approximately Normal.

A similar result holds for any finite number of parameters; we just need to use a **multivariate Normal distribution** instead.

Example: Normal population (Theoretical Results)

from ch2

Let Y_1, \dots, Y_n be IID Normal(μ_0, σ_0^2). Recall that the MLEs are \bar{Y} and $\hat{\sigma}^2$.

We already know that \bar{Y} has a Normal distribution:

$$\bar{Y} = \mu_{MLE}$$

$$\bar{Y} \sim N(\mu_0, \sigma_0^2/n)$$

$$E(\bar{Y}) = \mu_0$$

$$\text{Var}(\bar{Y}) = \frac{\sigma_0^2}{n}, \quad \text{se}(\bar{Y}) = \sqrt{\frac{\sigma_0^2}{n}}$$

Now we know that $\hat{\sigma}^2$ also has an approximately Normal distribution. After [math], it turns out that

$$\hat{\sigma}^2 = \sigma_{MLE}^2$$

$$\hat{\sigma}^2 \approx N(\sigma_0^2, 2\sigma_0^4/n)$$

$$E(\hat{\sigma}^2) = \sigma_0^2$$

$$\text{Var}(\hat{\sigma}^2) = \frac{2\sigma_0^4}{n}$$

$$\text{se}(\hat{\sigma}) = \sqrt{\frac{2\sigma_0^4}{n}}$$

Fisher information: multi-parameter case

If there are two parameters, then instead of being a single number, the Fisher information is a 2×2 matrix of second derivatives and partial derivatives of the log-likelihood.

e.g. If the parameters are a and b , the observed Fisher information $\mathcal{I}(\hat{\theta})$ is the matrix

$$\theta = (a, b), \quad \mathcal{I}(\theta) = \begin{pmatrix} \frac{\partial^2 l}{\partial a^2} & \frac{\partial^2 l}{\partial a \partial b} \\ \frac{\partial^2 l}{\partial b \partial a} & \frac{\partial^2 l}{\partial b^2} \end{pmatrix}$$

evaluated at the MLE $\hat{\theta} = (\hat{a}, \hat{b})$. ($\frac{\partial^2 l}{\partial b \partial a}$ means “take the derivative of the likelihood with respect to a , then take the derivative of the result with respect to b .”)

Fortunately, as we've seen, we can get software and R functions like `optim()` to calculate this stuff for us.

The variance matrix

We now need to find the **inverse** of the information matrix.

Suppose \mathbf{X} is a square matrix. Its inverse \mathbf{X}^{-1} is a square matrix such that

$$\mathbf{X}^{-1}\mathbf{X} = \mathbf{I}$$

where \mathbf{I} is an *identity* matrix: a square matrix that's all 1's along the main diagonal and 0 everywhere else.

The matrix

$$\hat{\mathbf{V}}(\hat{\boldsymbol{\theta}}) = -\mathcal{I}(\hat{\boldsymbol{\theta}})^{-1}$$

is an estimate of what's called the **variance matrix** or the **variance-covariance matrix**. (Let's not worry about "covariance" for now.)

Getting the variance estimates

Suppose what we want to know are the variance estimates of the MLEs.
Then take the variance matrix

$$\hat{\mathbf{V}} = -\mathcal{I}(\hat{\boldsymbol{\theta}})^{-1} = \begin{pmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \end{pmatrix}$$

and look at the main diagonal. These will be the variance estimates of our MLEs.

e.g. If we're estimating a and b from the observed information from a couple of slides ago, then the estimated variance of \hat{a} will be the entry in the first row and first column of $\hat{\mathbf{V}}$. The estimated variance of \hat{b} will be the entry in the second row and second column of $\hat{\mathbf{V}}$.

$$\hat{V}(\hat{a}) = v_{1,1}, \quad \hat{V}(\hat{b}) = v_{2,2}$$

Two parameter case: Separate intervals

Suppose we have two parameters to estimate, θ_1 and θ_2 .

Quite often we want to find a confidence interval for θ_1 , or an interval for θ_2 , or both.

This is similar to doing two one-parameter problems.

The main complication is that the Hessian is now a matrix (not a single number) that we have to invert to get the variance matrix.

Two parameters: Negative binomial example

Let's go back to the `schooldays.txt` data. Previously we fitted a negative binomial to the `absent` variable:

```
schooldays <- read.table("schooldays.txt", header = TRUE)
absent <- schooldays$absent
absent.nll <- function(params){
  size <- params[1]
  prob <- params[2]
  lik <- dnbinom(absent, size, prob)
  return(-sum(log(lik)))
}
absent.mle <- optim(par = c(1, 0.5), fn = absent.nll, hessian = TRUE)
```



Two parameters: Negative binomial example

```
> absent.mle$par  
[1] 1.10989583 0.06434699  
> absent.mle$hessian  
[,1]      [,2]  
[1,] 187.7174 -2393.467  
[2,] -2393.4669 44139.152
```

To get our variance estimates, we need to find the inverse of the Hessian. We get this in R using `solve()`:

```
absent.var <- solve(absent.mle$hessian)
```



Two parameters: Negative binomial example

```
> absent.var  
[,1]      [,2]  
[1,] 0.0172620690 9.360441e-04  
[2,] 0.0009360441 7.341307e-05
```

So the estimated variance of our size MLE is 0.0173 and the estimated variance of our prob MLE is 0.0000734.

If you prefer standard errors (and who doesn't):

```
> sqrt(diag(absent.var))  
[1] 0.131385193 0.008568143
```



Two parameters: Negative binomial example

Find 95% confidence intervals for the size and prob parameters:

```
> se <- sqrt(diag(absent.var))  
> absent.mle$par - qnorm(.975) * se  
[1] 0.85238558 0.04755374  
> absent.mle$par + qnorm(.975) * se  
[1] 1.36740607 0.08114024
```



A 95% CI for the size parameter is 0.85 to 1.37.

A 95% CI for the prob parameter is 0.048 to 0.081.

Interpreting the intervals

If we simulated a large sample from a Negative binomial distribution and found intervals for size and prob using this method, then repeated this lots of times, we'd find:

- About 95% of our intervals for size would contain the true size parameter;
- About 95% of our intervals for prob would contain the true prob parameter.

On the other hand, if our distribution isn't actually Negative binomial, we might do much worse. . .

Additional warning: The two intervals aren't independent. If one interval missed its true parameter, there's a good chance the other does as well.

Fancier Wald tests and intervals

Functions of the parameters

We can also do Wald tests on (smooth) functions of the parameters.

Let $\xi = g(\boldsymbol{\theta})$. Suppose we wish to test $H_0 : \xi = \xi_0$. We find the MLE $\hat{\boldsymbol{\theta}}$ and stick it into the function to get $\hat{\xi} = g(\hat{\boldsymbol{\theta}})$. Then the Wald statistic

$$W = \frac{(\hat{\xi} - \xi_0)^2}{\text{Var}(\hat{\xi})}$$

has a large-sample chi-squared distribution with one degree of freedom.

The tricky parts are: (a) setting up the problem in terms of a ξ ; (b) estimating $\text{Var}(\hat{\xi})$.

Exponential or Gamma?

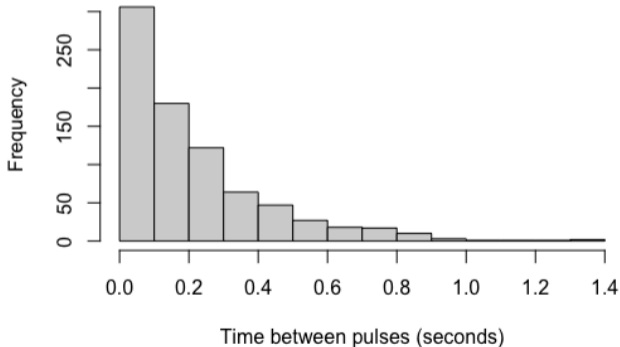
Recall that an $\text{Exponential}(\lambda)$ distribution is a special case of a $\text{Gamma}(k, \lambda)$ distribution with $k = 1$.

If we need to see if an Exponential distribution is sufficient or if we need a gamma, we can fit a gamma distribution and test the hypothesis that $k = 1$.

Example: Nerve firing times

The data set `nerve.txt` contains 799 waiting times between successive pulses along a nerve fiber. Do the times follow an Exponential distribution?

Distribution of times between nerve pulses



Fit a gamma

```
nerve.nll <- function(pars){  
  k <- pars[1]  
  lambda <- pars[2]  
  lik <- dgamma(nerve, shape = k, rate = lambda)  
  return(-sum(log(lik)))  
}  
nerve.mle <- optim(par = c(1, 1), fn = nerve.nll, hessian = TRUE)
```

I get an estimate $\hat{k} = 1.174$. Is this close enough to 1?



Find the variance matrix

Ideally we'd use the theoretical variance of \hat{k} to do the hypothesis test. However, it's not clear (without doing a bunch more math) what this theoretical variance is. Instead, let's invert the Hessian:

```
> nerve.var <- solve(nerve.mle$hessian)
> nerve.var
[,1]      [,2]
[1,] 0.00274911 0.01257745
[2,] 0.01257745 0.08829494
```

Our estimated variance of \hat{k} is 0.00275.



Test the hypothesis

The Wald test statistic from previous slides was

$$W = \frac{(\hat{\theta} - \theta_0)^2}{\hat{V}}$$

```
w <- (nerve.mle$par[1] - 1)^2 / nerve.var[1, 1]  
1 - pchisq(w, df = 1)
```

I get a Wald statistic of 11.0 and a P -value of 0.0009.



There's enough evidence to reject the hypothesis that the distribution really truly is Exponential.

Confidence intervals

We might as well find confidence intervals for our parameters:

```
se <- sqrt(diag(nerve.var))  
nerve.mle$par - qnorm(.975) * se  
nerve.mle$par + qnorm(.975) * se
```

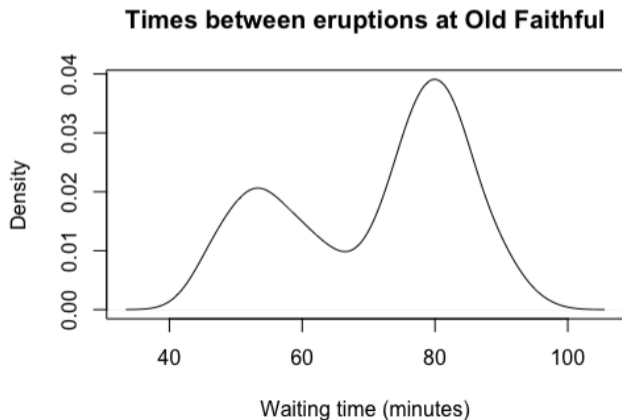
I get intervals of $[1.07, 1.28]$ for k and $[4.79, 5.96]$ for λ .



Example from the textbook: Old Faithful



How would you model this?



Textbook model

You could model this as a *mixture* of two Normal distributions.

$$pN(\mu, \sigma^2) + (1 - p)N(\nu, \tau^2)$$

You'd need to estimate five parameters:

- p , the probability of being in the first Normal;
- μ and σ , the expected value and SD of the first Normal;
- ν and τ , the expected value and SD of the second Normal.

The code to fit this model is on Millar p. 46.

Results

```
> round(Wald.table, 4)
```

	MLE	Std.Errors	LowerBound	UpperBound
--	-----	------------	------------	------------

p	0.3609	0.0312	0.2998	0.4220
---	--------	--------	--------	--------

mu	54.6145	0.6995	53.2435	55.9856
----	---------	--------	---------	---------

sigma	5.8698	0.5370	4.8173	6.9224
-------	--------	--------	--------	--------

nu	80.0908	0.5046	79.1017	81.0798
----	---------	--------	---------	---------

tau	5.8682	0.4010	5.0822	6.6542
-----	--------	--------	--------	--------

Implications

It seems like σ and τ might be the same.

Could we test the null hypothesis $H_0 : \sigma = \tau$?

It's possible to do this using Wald tests, but it's a bit tricky, and in any case Wald tests might not be the most accurate approach (because the multivariate Normal assumption is iffy.)

We'll need a different way to do testing...

Likelihood ratio tests and intervals

Another kind of hypothesis test

Suppose we're fitting a model where the parameter vector is θ .
Suppose we have a null hypothesis about the first r parameters:

$$H_0 : \theta_1 = \theta_{01}, \theta_2 = \theta_{02}, \dots, \theta_r = \theta_{0r}$$

We might be able to do a Wald test or tests, but it would better to do a test based directly on the likelihood.

Likelihood ratio tests: The idea

1. Find the maximum value of the likelihood without any restrictions.
2. Find the maximum value of the likelihood under the restrictions

$$H_0 : \theta_1 = \theta_{01}, \theta_2 = \theta_{02}, \dots, \theta_r = \theta_{0r}$$

Because of the restrictions, this will usually be a bit smaller.

3. Find the ratio of the first likelihood to the second likelihood. If the ratio is too big, reject the null hypothesis.

$$\frac{\max L(\theta)}{\max L(\theta_0)}$$

Example: Binomial test

Suppose we wish to test the null hypothesis $H_0 : p = 0.5$. We toss a coin 100 times and get 60 heads.

- Unrestricted maximum likelihood: The MLE is $\hat{p} = 0.6$. The likelihood under the MLE is

$$\binom{100}{60} \cdot 0.6^{60} \cdot 0.4^{40} \approx 0.0812.$$

- Restricted likelihood: Under H_0 , we have to use $p_0 = 0.5$. The likelihood is then

$$\binom{100}{60} \cdot 0.5^{60} \cdot 0.5^{40} \approx 0.0108.$$

The likelihood ratio is $0.0812/0.0108 \approx 7.5$.

But we usually use log-likelihoods

Because likelihoods are numerically annoying, we usually do an equivalent test based on the log-likelihood instead:

1. Find the maximum value of the log-likelihood without any restrictions.
2. Find the maximum value of the log-likelihood under H_0 .
3. If the first log-likelihood is significantly bigger than the second log-likelihood, reject H_0 .

$$\log \frac{\max L(\theta)}{\max L(\theta_0)} = \log \max L(\theta) - \log \max L(\theta_0) = \max l(\theta) - \max l(\theta_0)$$

Usually, this turns out to be a slightly more reliable test statistic than the Wald statistic, especially in problems with more than one parameter.

Example: Binomial test

The log-likelihood for the Binomial(100, p) is

$$\log \binom{100}{60} + 60 \log p + 40 \log(1 - p)$$

The unrestricted maximum log-likelihood is

$$\log \binom{100}{60} + 60 \log 0.6 + 40 \log 0.4 = -2.5106$$

The restricted log-likelihood is

$$\log \binom{100}{60} + 60 \log 0.5 + 40 \log 0.5 = -4.5242$$

The difference is $-2.5106 - (-4.5242) = 2.0136$. Is that big? Small?

The likelihood ratio test

$$H_0 : \theta_1 = \theta_{01}, \theta_2 = \theta_{02}, \dots, \theta_r = \theta_{0r}$$

Let $l(\hat{\theta})$ be the unrestricted maximum log-likelihood.

Let $l(\hat{\theta}_0)$ be the restricted (maximum) log-likelihood.

The **likelihood ratio statistic** is

$$X = 2 \left(l(\hat{\theta}) - l(\hat{\theta}_0) \right)$$

Under the null and with a large sample size (and if nothing crazy is happening), X has an approximate *chi-squared distribution with r degrees of freedom* (where r is the number parameters that we fixed under the null.)

Example: Binomial test

For our Binomial example, the difference in log-likelihoods was

$$-2.5106 - (-4.5242) = 2.0136.$$

The LR test statistic is $2 \times 2.0136 = 4.027$.

The P -value is

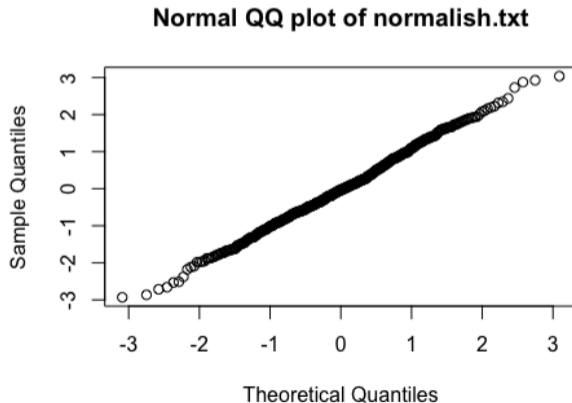
```
1 - pchisq(4.027103, df = 1)
```

which is 0.045. There's a bit of evidence against the null, but only a bit.
(Also note that 100 isn't a huge sample size for a Binomial...)

Example: Normalish

The file `normalish.txt` contains 500 observations I generated from a Normal distribution.

However, I don't know if the data is *standard* Normal (i.e. $\text{Normal}(0, 1^2)$) or not.



Example: Normalish

```
normalish <- scan("normalish.txt")  
x.bar <- mean(normalish)  
sigma2.hat <- mean(normalish^2) - x.bar^2
```

I get MLEs of -0.006 for μ and 1.098 for σ^2 .



Are these close enough to 0 and 1^2 that I can attribute the differences to random variation?

Example: Normalish

Maximum likelihood:

```
max.lik <- dnorm(normalish, x.bar, sqrt(sigma2.hat))
```

Null hypothesis likelihood:

```
null.lik <- dnorm(normalish, 0, 1)
```

The test statistic is twice the difference in log-likelihoods:

```
max.ll <- sum(log(max.lik))  
null.ll <- sum(log(null.lik))  
X <- 2 * (max.ll - null.ll)
```



Example: Normalish

I got a X statistic of 2.28.

To do the LR test, compare this statistic to a chi-squared distribution.

Because the null fixes *two* parameters, μ and σ^2 , we compare to a chi-squared with 2 degrees of freedom:

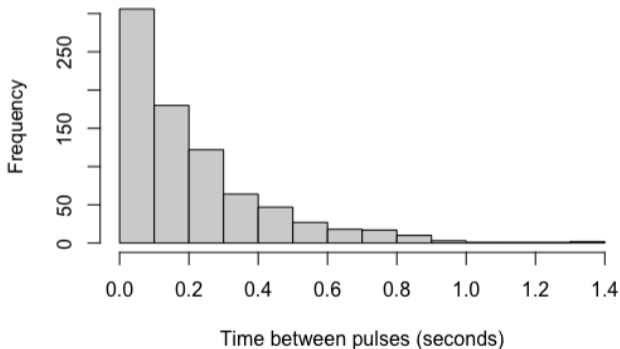
```
1 - pchisq(X, df = 2)
```

I get a P -value of 0.32. There's no evidence to reject the null; the data set is consistent with a standard Normal.

Example: Nerves revisited

Previously, there was a Wald test of the null hypothesis about the model for 799 waiting times between successive pulses along a nerve fiber.

Distribution of times between nerve pulses



Test the hypothesis

H_0 : Data come from $\text{Exp}(\lambda)$ or $\text{Gamma}(k = 1, \lambda)$

The Wald test statistic was

$$W = \frac{(\hat{\theta} - \theta_0)^2}{\hat{V}} = \frac{(\hat{k} - 1)^2}{\hat{V}}$$

```
w <- (nerve.mle$par[1] - 1)^2 / nerve.var[1, 1]  
1 - pchisq(w, df = 1)
```

I got a Wald statistic of 11.0 and a P -value of 0.0009.

Does the LR test give a similar result? In order to do the test, we need to find the maximum log-likelihood under $\text{Gamma}(k, \lambda)$ and the maximum log-likelihood when we fix the shape parameter $k = 1$, i.e., $\text{Gamma}(1, \lambda)$.

Maximum log-likelihood: Unrestricted

```
nerve <- scan("nerve.txt")
nerve.nll <- function(pars){
  k <- pars[1]
  lambda <- pars[2]
  lik <- dgamma(nerve, shape = k, rate = lambda)
  return(-sum(log(lik)))
}
nerve.mle <- optim(par = c(1, 1), fn = nerve.nll, hessian = TRUE)
```

Then

```
> -nerve.mle$value
[1] 422.1499
```

gives the maximum log-likelihood (the negative of the minimum log-likelihood.)

Maximum log-likelihood: Under the null

Now change our function so that k is fixed at 1:

```
null.nll <- function(lambda){  
  lik <- dgamma(nerve, shape = 1, rate = lambda)  
  return(-sum(log(lik)))  
}  
null.mle <- optim(2, null.nll, method = "Brent",  
  lower = 0, upper = 99, hessian = TRUE)
```

Then

```
> -null.mle$value  
[1] 415.9868
```

gives the maximum log-likelihood under the null. (Was there an easier way of getting this?)

Nerves: LR test

The test statistic is twice the difference in log-likelihoods:

```
X <- 2 * (- nerve.mle$value + null.mle$value)
```

We only fixed one parameter under the null, so compare to a chi-squared distribution with one degree of freedom:

```
1 - pchisq(X, df = 1)
[1] 0.0004466359
```

We can clearly reject the null hypothesis that the data literally comes from an Exponential distribution.

Important notes

- The LR test tends to do a bit better than the Wald test, and avoids some of Wald's arbitrariness (e.g. deciding whether to use p_0 or \hat{p} to calculate standard errors.)
- It's a little less intuitive than a Wald test. Most people who have taken some statistics know what a Z -statistic is; not everyone knows what a likelihood is.
- The LR test (like the Wald test) is still best for large samples. For small samples, more exact methods may be preferable.

Profile likelihood intervals

Inversion

In general, a 95% confidence interval for a parameter can be constructed out of all the parameter values that would *not* be rejected in a level $\alpha = 0.05$ hypothesis test.

To see if a value θ_0 is in the 95% confidence interval for some single parameter θ :

1. Find the maximum value of the unrestricted log-likelihood;
2. Find the maximum value of the log-likelihood under the restriction $\theta = \theta_0$;
3. See if the LR test statistic is less than the critical value $qchisq(0.95, df = 1)$.
If so, θ_0 is in the interval.

(As usual, this works best for large samples.)

Profile likelihood intervals

```
> qchisq(.95, df = 1)
[1] 3.841459
```

θ_0 will be in the interval if and only if double the difference in log-likelihoods is less than 3.84.

Put another way, θ_0 will be in the interval if the maximum log-likelihood under the restriction $\theta = \theta_0$ is within $3.84/2 = 1.92$ of the overall maximum log-likelihood.

This means we can find a confidence interval just by glancing at a plot of the log-likelihood and seeing what values give a log-likelihood within 1.92 of the peak. This is what we've called the **profile likelihood confidence interval**.

Old example: Binomial profile likelihood

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

The maximum log-likelihood is

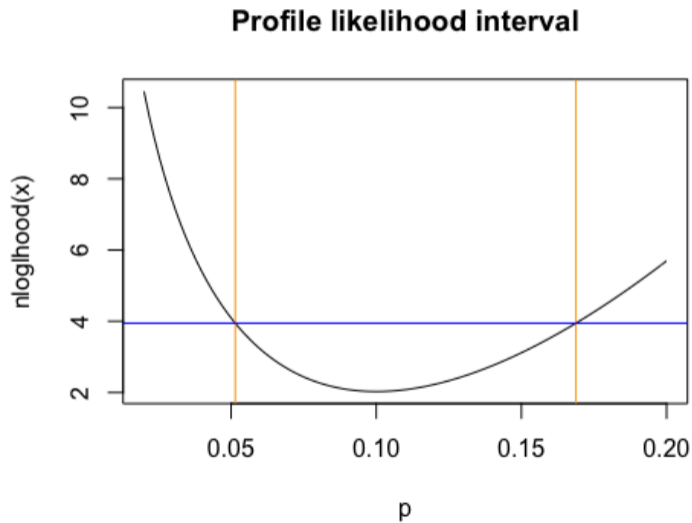
```
> log(dbinom(10, 100, 0.1))
[1] -2.025974
```

The log-likelihood at the lower and upper bounds is

```
> log(dbinom(10, 100, c(0.05142278, 0.168782)))
[1] -3.945704 -3.946837
```

What's the difference? The magical 1.92.

Old example: Binomial profile likelihood



Old example: Geometric profile likelihood

Toss a coin until you get a head. Count the number of tails until the first head.
Recall that the PMF of a $\text{Geometric}(p)$ distribution is

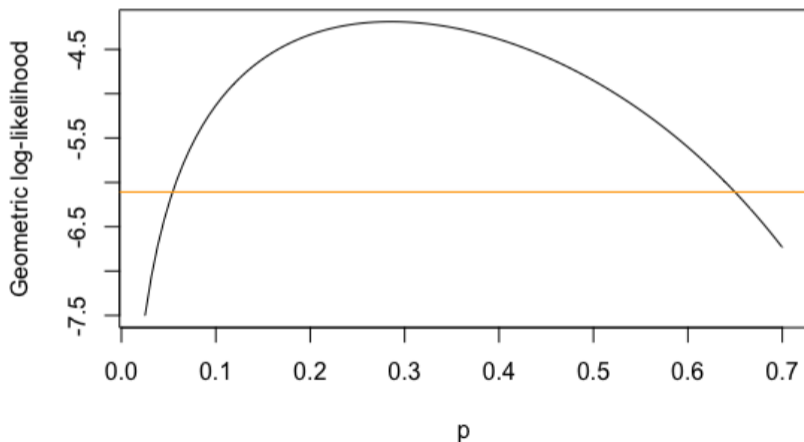
$$f(x) = \begin{cases} (1-p)^x p & x = 0, 1, 2, \dots \\ 0 & \text{otherwise.} \end{cases}$$

Suppose we observe $X_1 = 3, X_2 = 2$ from this distribution. We got 2 heads in 7 tosses, so the MLE is $\hat{p} = 2/7$.

What's a confidence interval for p ?

Geometric profile likelihood

Profile likelihood confidence interval



Old example: Geometric profile likelihood

Maximum log-likelihood:

```
> sum(log(dgeom(c(3, 2), 2/7)))  
[1] -4.187887
```

The cut-offs for a 95% interval will be where the log-likelihood is $\text{sum}(\log(\text{dgeom}(c(3, 2), 2/7))) - \text{qchisq}(.95, \text{df} = 1)/2$ or about -6.11 .

We could proceed by trial and error:

```
> sum(log(dgeom(c(3, 2), .05)))  
[1] -6.247931  
> sum(log(dgeom(c(3, 2), .65)))  
[1] -6.110676
```

Remember `plkhci()`?

```
geom.nll <- function(p){  
  lik <- dgeom(c(3, 2), p)  
  return(-sum(log(lik)))  
}  
library(Bhat)  
control.list = list(label = "p", est = 2/7, low = 0, upp = 1)  
plkhci(control.list, geom.nll, "p", prob = 0.95)
```

Example: Men's heights

Suppose we want to find *separate* confidence intervals for μ and σ^2 for adult American men's heights.

We have the data file `mensheights.txt`.

Let's try the profile likelihood method.

Profile CIs for the mean and variance

```
mensheights <- scan("mensheights.txt")
x.bar <- mean(mensheights)
sigma2.hat <- mean(mensheights^2) - x.bar^2
height.nll <- function(pars){
  mu <- pars[1]
  sigma2 <- pars[2]
  lik <- dnorm(mensheights, mu, sqrt(sigma2))
  return(-sum(log(lik)))
}
control.list = list(label = c("mu", "sigma2"), est = c(x.bar, sigma2.hat),
                    low = c(0, 0), upp = c(999, 999))
```



Profile CIs for the mean and variance

```
plkhci(control.list, height.nll, "mu")  
plkhci(control.list, height.nll, "sigma2")
```



I get intervals of 175.7 cm to 176.0 cm for the mean and 53.5 to 58.6 cm² for the variance (and hence 7.31 to 7.66 cm for the SD.)

A nice thing about this method is that if we parameterized the Normal in terms of mean and SD rather than mean and variance, we'd get the same result!

The G -test

Multinomial distributions

Suppose we have observations that fall into one of m categories.

Example (Trosset Chapter 13, exercise 13.4.2): Mars, the manufacturers of M&M's, claimed that M&M's were mixed in the following percentages:

- Brown: 13%
- Yellow: 14%
- Red: 13%
- Blue: 24%
- Orange: 20%
- Green 16%

This is called a **multinomial distribution**.

Multinomial data

To test the claim, Prof. Trosset got his students to open bags of M&M's and count the colors.

Color	Brown	Yellow	Red	Blue	Orange	Green
Mars's claim	0.13	0.14	0.13	0.24	0.20	0.16
Observed M&M's	121	84	118	226	226	123

Is the observed data consistent with Mars's claim, or can we reject that hypothesis?

Two tests

Given a decent sample size, there are two somewhat similar tests we can carry out to test a multinomial hypothesis:

- The traditional **Pearson's chi-squared test**;

$$K = \sum_{i=1}^m \frac{(o_i - e_i)^2}{e_i}$$

- A likelihood ratio test, in this case called a **G-test**.

You may have seen Pearson's chi-squared test elsewhere. It's usually fine, but the likelihood ratio test is a bit more fundamentally sound (plus the point of this course is likelihood.)

The G -test for a fully-specified multinomial hypothesis

1. Find the maximum log-likelihood with no restrictions other than that the probabilities have to add up to 1. If there are m categories, this will have $m - 1$ free parameters.
2. Find the (maximum) log-likelihood under the null hypothesis. For a **fully-specified null hypothesis**, i.e. one where all probabilities are given by the null, this has no free parameters. $H_0 : \mathbf{p} = \mathbf{p}_0$.
3. Find G , twice the difference in log-likelihoods, and compare this to a chi-squared distribution with $m - 1$ degrees of freedom.

The G -statistic is

$$G = 2 (l(\hat{\mathbf{p}}) - l(\mathbf{p}_0)) \approx \chi^2(m - 1),$$

where $\mathbf{p} = (p_1, p_2, \dots, p_m)$ and $\mathbf{p}_0 = (p_{01}, p_{02}, \dots, p_{0m})$.

Multinomial distribution

Suppose we draw a ball from an urn which has balls with m different colors labeled "color 1, color 2, ..., color m ." Let $\mathbf{p} = (p_1, \dots, p_m)$, $p_i \geq 0$ and $\sum_{i=1}^m p_i = 1$ and suppose that p_i is the probability of drawing a ball of color i . Draw n times (independent draws with replacement) and let $\mathbf{X} = (X_1, \dots, X_m)$ where X_i is the number of times that color i appears. Then $n = \sum_{i=1}^m X_i$. We say \mathbf{X} has a Multinomial(n, \mathbf{p}) distribution. The probability function is

$$f(\mathbf{x}) = \binom{n}{x_1, \dots, x_m} p_1^{x_1} \cdots p_m^{x_m} = \frac{n!}{x_1! \cdots x_m!} p_1^{x_1} \cdots p_m^{x_m}$$

- Multivariate version of Binomial distribution.
- $E(\mathbf{X}) = n\mathbf{p} = n(p_1, \dots, p_m) = (np_1, \dots, np_m)$
- The MLE of \mathbf{p} is $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_m) = \left(\frac{X_1}{n}, \dots, \frac{X_m}{n}\right)$.
- The marginal distribution of X_i is Binomial(n, p_i).
- R functions: `dmultinom(x, size, prob)`, `rmultinom(n, size, prob)`

The G -test in practice

Suppose we have a (fully-specified) null that the probabilities of each category are $p_{01}, p_{02}, \dots, p_{0m}$.

We collect a sample of overall size n .

1. Count up your data. Let o_i be the observed count in category i .
2. Find each e_i , the expected count in category i under the null, as

$$e_i = np_{0i}$$

3. The G -statistic reduces to

$$G = 2 \sum_{i=1}^m o_i \cdot \log \left(\frac{o_i}{e_i} \right)$$

4. The P -value is $1 - \text{pchisq}(G, \text{df} = m - 1)$.

Example: M&M's

```
observed <- c(121, 84, 118, 226, 226, 123)
n <- sum(observed)
p0 <- c(.13, .14, .13, .24, .20, .16)
expected <- n * p0
G <- 2 * sum(observed * log(observed / expected))
1 - pchisq(G, df = 5)
```



Looks like Mars was incorrect: there are too many orange M&M's and not enough yellows and greens.

More about G -tests

- For the G -test to be accurate, none of the expected values can be too small (weird things can happen when you divide by a small number.) The rule of thumb is that all expected counts must be at least five.
- Under these conditions, Pearson's chi-squared test is usually okay, so it's not worth making too much fuss if someone prefers that test.
- There are lots more chi-squared tests/ G -tests for data that's categorical or that can be forced into categories (e.g. tabular data, count data.) The basic idea is all the same, but determine the expected counts can be tricky; so can working out the correct degrees of freedom.