Likelihood Ratio Test

Andrew Zieffler

Why use LRT?

- The LRT, like the AIC (which is preferred method), can be sued to select static predictors
  - Applied researchers must adhere to standards of the *p*-value culture (analyze, interpret results accordingly)
  - Researcher is satisfied only comparing nested models, and only two at a time
  - Researcher wants to use another criterion for balancing fit and parsimony aside from AIC (AICc)
  - Researcher is planning a study and estimate the number of subjects needed (power analysis)

LRT as Evidence and Effect

- In this course, LRT-*p* will be regarded as evidence for and against hypotheses in a single analysis
  - Alpha (α) will be treated as a single cutoff value with no long-run properties (i.e., not as type I error rate)
  - As such, there will be no concern about the number of multiple comparisons performed
  - Guarding against false-positives by adjusting α only is useful if there is a chance that *all* the null-hypotheses may be true
  - A null-hypothesis is never true in observational research

Evaluation of Two Nested Models

- Two major components of the LRT
  - χ^2 test statistic and χ^2 distribution from which LRT-*p* is derived
- χ^2 is empirical deviance of reduced model minus empirical deviance of full model

$$\chi^2 = deviance_R - deviance_F$$

  where reduced model is nested within full model

- Larger value of χ^2 is indicative of more separation between the deviances (better fit for the full model)
- LRT penalizes the χ^2 statistic by obtaining LRT-*p* from a distribution with *df* equal to the difference in the number of parameters between the two models (which affects shape of distribution)

Consider the following models:

Reduced model $$y_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})\text{grade}_{ij} + \epsilon_{ij}$$

Full model $$y_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})\text{grade}_{ij} + \beta_2(\text{eth2}_i)\epsilon_{ij}$$

- Setting $\beta_2 = 0$ in the full model gives the reduced model
- Nested LMER models are compared using the `anova()` function
- Full and reduced models must be fitted with the exact same sample (e.g., if there are missing values in the static predictor for `eth2`, then the `anova()` function will produce an error. In such cases the `na.omit()` function can be used prior to fitting the models.)

```
## Fit LMER models
> model.0 <- lmer( read ~ 1 + grade5 + ( 1 + grade5 | subid ),
                   data = mpls.l, REML = FALSE )
> model.1 <- lmer( read ~ 1 + grade5 + eth2 + ( 1 + grade5 | subid ),
                   data = mpls.l, REML = FALSE )

## LRT
> anova( model.0, model.1 )

Data: mpls.l
Models:
model.0: read ~ 1 + grade5 + (1 + grade5 | subid)
model.1: read ~ 1 + grade5 + eth2 + (1 + grade5 | subid)
        Df    AIC    BIC  logLik  Chisq Chi Df Pr(>Chisq)
model.0  6 583.70 597.99 -285.85
model.1  7 573.78 590.45 -279.89 11.918      1   0.000556 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If the reduced model is the true model, there is a very small probability (*p < 0.001*) that we would obtain a chi-square at least as large as the one we did

- Difference in empirical deviances (χ^2) is natural manner to assess relationship between hypotheses and sample evidence
- Difference in deviance can not be taken at face value (over-fitting)
- A better measure is ΔAIC

Consider the AIC for two models (reduced and full):

$$\begin{aligned}\Delta AIC &= AIC_R - AIC_F \\ &= (deviance_R + 2 \cdot K_R) - (deviance_F + 2 \cdot K_F) \\ &= (deviance_R - deviance_F) - 2 \cdot (K_F \cdot K_R) \\ &= \chi^2 - 2 \cdot (K_F \cdot K_R) \\ &= \chi^2 - 2 \cdot \Delta K\end{aligned}$$

- The ΔAIC penalizes the χ^2 statistic based on the difference in complexity of the models

- Based on ΔAIC the weight of evidence can be computed for the reduced model as

$$W_R = \frac{exp(-0.5 \cdot \Delta AIC)}{1 + exp(-0.5 \cdot \Delta AIC)}$$

- The weight of evidence for the full model is:

$$W_F = 1 - W_R$$

- These weights can then be used to create an evidence ratio

```
## Delta AIC
> delta.aic <- 11.918 - 2 * (7 - 6)
> delta.aic
[1] 9.918

## Weight for reduced model
> weight.r <- exp( -0.5 * delta.aic ) / ( 1 + exp( -0.5 * delta.aic ) )
> weight.r
[1] 0.006971008

## Weight for full model
> weight.f <- 1 - weight.r
> weight.f
[1] 0.993029

## Evidence ratio
> weight.f / weight.r
[1] 142.4513
```

Give the data, and the two models, and that neither model is the true model, the full model is 142 times more likely to be the best approximating model

- Both the AIC and LRT approaches provide a trade-off between fit and complexity
- LRT approach has assumption that the reduced model is the true model–the model assumed in the null hypothesis
- If multiple LRTs are carried out, the true model is often changing, since the full model suggested by one LRT is often the reduced model in a subsequent test–this seems to tread on thin philosophical ice
- If the LRT is used, it is perhaps better to move to a weight of evidence interpretation rather than an interpretation of the *p*-value

α	**Strength of Evidence**
0.100	Borderline
0.050	Moderate
0.025	Substantial
0.010	Strong
0.005	Very strong
0.001	Overwhelming

Strength of Evidence via Predictive Accuracy

- Need χ^2 common to both LRT and ΔAIC (as basis to compare weight of evidence and *p*-values)
- Consider case in which full model and reduced model differ by one parameter (*df* = ΔK = 1)

ΔAIC < 0 Reduced model has better fit

ΔAIC > 0 Full model has better fit

ΔAIC > 0 when $\chi^2 > 2 \cdot \Delta K = 2$

Full model has superior fit when test statistic is greater than 2

$\chi^2 = 2$ results in $\Delta AIC = 2 - 2 = 0$

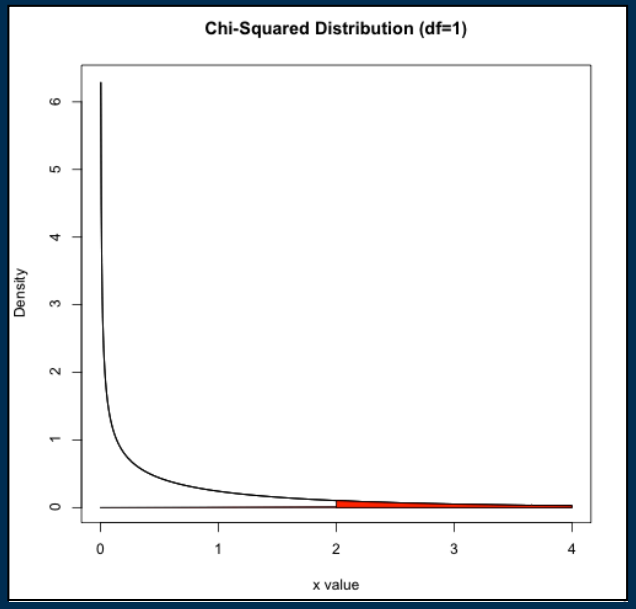
$$W_R = \frac{1}{1+2} = 0.5 \quad \text{and} \quad W_F = 1 - 0.5 = 0.5$$

Reduced model and full model are equally plausible in terms of being the best approximating model

Rule of parsimony may favor reduced model, but this is not based on reasoning from the weight of evidence

- Since both LRT and ΔAIC share χ^2 it is interesting to see which α criterion is associated with $\chi^2 = 2$ (with $df = 1$)
- We use the `pchisq()` function, providing the χ^2 value to the argument `q=` and the *df* to the `df=` argument. The `lower.tail=FALSE` argument computes the probability in the upper tail

```
> pchisq( q = 2, df = 1, lower.tail = FALSE )
[1] 0.1572992
```



α = 0.1573 when $W_R = W_F = 0.5$, which implies that LRT-*p* = 0.1573 is evaluated as less than "borderline" or "negligible"

- Now consider case in which full model and reduced model still differ by one parameter (*df* = ΔK = 1), but this time $\chi^2 = 8$

$\chi^2 = 8$ results in $\Delta AIC = 8 - 2 = 6$

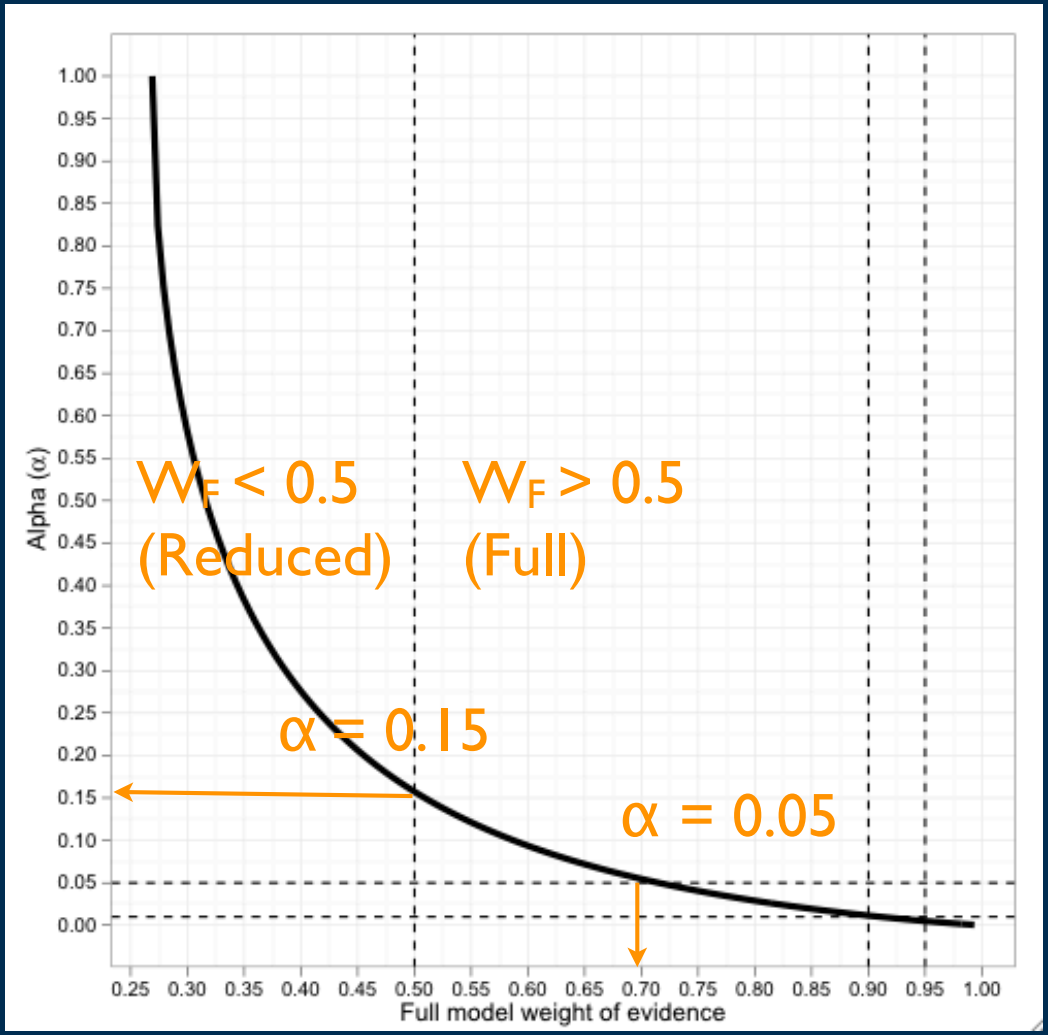
$W_R = 0.0474$ and $W_F = 0.9526$

Indicating the full model is more likely to be the best approximating model

```
> pchisq( q = 8, df = 1, lower.tail = FALSE )
[1] 0.004677735
```

The cutoff for the LRT-p is α = 0.0047. Based on the table of strength of evidence, this would be "very strong" evidence for the full model

- Below we examine the relationship between the LRT-*p* cutoff and W_F for several values of χ^2



α is a nonlinear function of W_F

Larger values of α are associated with smaller values of W_F

LRT-*p* < 0.15 offers support for reduced model (perhaps a reasonable cutoff value?)

LRT-*p* ≥ 0.05 corresponds to evidence for the full model at a weight of evidence of at least 2.33 (0.7/0.3)

LRT-*p*	**W_F**	**Evidence for Full Model**
LRT-$p > 0.15$	$W_F < 0.50$	None
$0.10 <$ LRT-$p \leq 0.15$	$0.50 \leq W_F < 0.70$	Weak
$0.01 <$ LRT-$p \leq 0.10$	$0.70 \leq W_F < 0.90$	Moderate
$0.001 <$ LRT-$p \leq 0.01$	$0.90 \leq W_F < 0.99$	Strong
LRT-$p \leq 0.001$	$W_F \geq 0.99$	Very strong

- These 'cutoffs' are valid for *df* = 1–4
- With *df* > 4 the LRT-*p* can no longer be used to judge weight of evidence
- In these cases LRT-*p* tends to be too small, overestimating the strength of evidence
- LRT is not recommended for testing models with large number of fixed effects
- It is always possible to replace large *df* comparisons with a series of smaller ones

Comparing Multiple Models

- We will revisit the RQ from last set of notes using LRT
- First we examine the muddy waters of an exploratory analysis dressed up as a confirmatory analysis
- Important since examination of many LRT analyses may yield spurious results
- Two disciplined approaches considered: Step-up method and the Top-down method

Step-Up Method

- Begin with simple model
- LRT is used to examine if model can be made more complex (e.g., adding predictors)
- When this is data-driven, method known as *forward selection*
- Here we have a priori specified the models (each differing by 1 *df*)
- We also a priori set the cutoff value (e.g., LRT-*p* to < 0.01)
- A graded series of models is formulated based on RQ

To what extent does risk act as a proxy for ethnicity? Do achievement gaps based on risk and/or ethnicity persist over time?

Model	Effects
0	grade5
1	grade5, eth2
2	grade5, eth2, dadv
3	grade5, eth2, dadv, (grade5*eth2)
4	grade5, eth2, dadv, (grade5*eth2), (grade5*dadv)

```
## Fit models
> model.0 <- lmer(read ~ grade5...)
            ⋮
> model.4 <- lmer(read ~ grade5...)

## LRT results
> myout <- anova(model.0, model.1, model.2, model.3, model.4)

## Effect size
> myout$delta.aic <- c(myout$Chisq - 2 * myout$"Chi Df")
> myout$weight.f <- (1 - exp(-0.5 * myout$delta.aic) / (1 +
    exp(-0.5 * myout$delta.aic)))
> myout$eratio.f <- myout$weight.f / (1 - myout$weight.f)

## Print
> myout[ ,-c(3:4)]

> print(model.2, cor = FALSE)
```

Weight of evidence for full model

```
## Print results

         Df    AIC   Chisq Chi Df Pr(>Chisq) delta.aic weight.f eratio.f
model.0   6 583.70
model.1   7 573.78 11.9178      1    0.00056    9.9178  0.99303  142.436
model.2   8 573.54  2.2372      1    0.13473    0.2372  0.52961    1.126
model.3   9 574.62  0.9270      1    0.33564   -1.0730  0.36900    0.585
model.4  10 576.59  0.0232      1    0.87886   -1.9768  0.27123    0.372
```

Each subsequent model has one additional fixed effect

LRT-*p*

How many times better the full model fits better than the reduced model

***IMPORTANT: Cannot rank order these models. Weight of evidence is for comparing two nested models

- Recommended the evidence for all model comparisons be presented
- Model 2 is selected
    - Weak evidence that the risk intercept needs to be included when ethnicity is already in the model

```
## Print model 2 output

> print(model.2, cor = FALSE)
Linear mixed model fit by maximum likelihood
Formula: read ~ grade5 + eth2 + dadv + (grade5 | subid)
   Data: mpls.l
   AIC   BIC logLik deviance REMLdev
 573.5 592.6 -278.8    557.5   541.2
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subid    (Intercept) 190.3723 13.7975
          grade5        7.2398  2.6907  -0.287
 Residual              18.1078  4.2553
Number of obs: 80, groups: subid, 22

Fixed effects:
            Estimate Std. Error t value
(Intercept) 203.6478     6.5114  31.276
grade5        4.8767     0.7491   6.510
eth2W        19.0958     6.9074   2.765
dadvDADV    -10.4699     6.8212  -1.535
```

- Initial analysis included model.0 which is not of interest, only use to test model.1
- In step-up approach, intercept effects are evaluated prior to any slope effects (main effects before interactions)
- Data driven analysis can be used in step-up approach
  - Model with no predictors is reduced model
  - Model with one static predictor having largest effect is full model
  - Single predictors are added one at a time

Top-Down Method

- Begin with most complex model
- LRT is used to examine if model can be made more simple (e.g., removing predictors)
- When this is data-driven, method known as *backward selection*
- In this approach, interactions are always tested prior to main effects
- First step is to fit most complex model and examine effects of predictors to make list of candidates for removal (candidates for removal must only be of the highest order terms)

```
## most complex model

> model.1 <- lmer(read ~ grade5 + eth2 + dadv + grade5 * eth2 +
      grade5 * dadv + (grade5 | subid), mpls.l, REML = FALSE)

> round(summary(model.1)@coefs, 2)

                Estimate Std. Error t value
(Intercept)       202.49       6.82   29.70
grade5              5.68       1.61    3.53
eth2W              21.33       7.32    2.91
dadvDADV          -10.04       7.23   -1.39
grade5:eth2W       -1.59       1.71   -0.92
grade5:dadvDADV    -0.26       1.70   -0.15
```

Candidate for removal

```
## most complex model vs. reduced model

> model.2 <- lmer(read ~ grade5 + eth2 + dadv + grade5 * eth2 +
      (grade5 | subid), mpls.l, REML = FALSE)
> myout <- anova(model.2, model.1)
> myout$delta.aic <- c(myout$Chisq - 2 * myout$"Chi Df")
> myout$weight.f <- (1 - exp(-0.5 * myout$delta.aic) /
      (1 + exp(-0.5 * myout$delta.aic)))
> myout$eratio.f <- myout$weight.f / (1 - myout$weight.f)
> myout[ ,-c(3:4)]

        Df    AIC  Chisq Chi Df Pr(>Chisq) delta.aic weight.f eratio.f
model.2  9 574.62
model.1 10 576.59 0.0232      1    0.87886   -1.9768  0.27123  0.37218
```

Not a lot of evidence for the full model

Reduced model accepted, becomes full model in next analysis. New reduced model drops remaining slope term (only higher order term remaining)

```
## new full model vs. new reduced model

> model.3 <- lmer(read ~ grade5 + eth2 + dadv + (grade5 | subid),
      mpls.l, REML = FALSE)
> myout <- anova(model.3, model.2)
> myout$delta.aic <- c(myout$Chisq - 2 * myout$"Chi Df")
> myout$weight.f <- (1 - exp(-0.5 * myout$delta.aic) /
      (1 + exp(-0.5 * myout$delta.aic)))
> myout$eratio.f <- myout$weight.f / (1 - myout$weight.f)
> myout[ ,-c(3:4)]

         Df    AIC Chisq Chi Df Pr(>Chisq) delta.aic weight.f eratio.f
model.3   8 573.54
model.2   9 574.62 0.927      1    0.33564    -1.073    0.369  0.58479
```

Not a lot of evidence for the full model

Reduced model accepted, becomes full model in next analysis. New reduced model is determined by examining coefficients.

```
## examine coefficients

> round(summary(model.3)@coefs, 2)

            Estimate Std. Error t value
(Intercept)   203.65       6.51   31.28
grade5          4.88       0.75    6.51
eth2W          19.10       6.91    2.76
dadvDADV      -10.47       6.82   -1.53
```

Candidate for removal

```
> model.4 <- lmer(read ~ grade5 + eth2 + (grade5 | subid), mpls.l,
      REML = FALSE)
> myout <- anova(model.4, model.3)
> myout$delta.aic <- c(myout$Chisq - 2 * myout$"Chi Df")
> myout$weight.f <- (1 - exp(-0.5 * myout$delta.aic) /
      (1 + exp(-0.5 * myout$delta.aic)))
> myout$eratio.f <- myout$weight.f / (1 - myout$weight.f)
> myout[ ,-c(3:4)]

        Df    AIC  Chisq Chi Df Pr(>Chisq) delta.aic weight.f eratio.f
model.4  7 573.78
model.3  8 573.54 2.2372      1    0.13473   0.23719  0.52961   1.1259
```

```
> model.5 <- lmer(read ~ grade5 + (grade5 | subid), mpls.l,
      REML = FALSE)
> myout <- anova(model.5, model.4)
> myout$delta.aic <- c(myout$Chisq - 2 * myout$"Chi Df")
> myout$weight.f <- (1 - exp(-0.5 * myout$delta.aic) /
      (1 + exp(-0.5 * myout$delta.aic)))
> myout$eratio.f <- myout$weight.f / (1 - myout$weight.f)
> myout[ ,-c(3:4)]

        Df    AIC  Chisq Chi Df Pr(>Chisq) delta.aic weight.f eratio.f
model.5  6 583.70
model.4  7 573.78 11.918      1 0.00055601    9.9178  0.99303   142.44
```

Model 4 is adopted

Parametric Bootstrap

- Sampling distribution of χ^2 is inaccurate with large *df* and/or sample size is small
- To improve accuracy, parametric bootstrap can be used
- Provides approximation of the sampling distribution for χ^2 based on models fit to simulated data
- Can use observed value of χ^2 to compute empirical *p*-value

Consider Model 0 (grade5) and Model 1 (grade5, eth2) from Step-Up Procedure

```
## Estimate models

> model.0 <- lmer(read ~ grade5 + (grade5 | subid), mpls.1,
      REML = FALSE)
> model.1 <- lmer(read ~ grade5 + eth2 + (grade5 | subid), mpls.1,
      REML = FALSE)
```

Observed χ^2 value = 15.105

- Use reduced model to estimate sampling distribution of χ^2
- Evaluate 15.105 in this bootstrap distribution

To help understand the simulation process we will carry out a single bootstrap

```
## simulate data from the reduced model

> simDV <- simulate( model.0 )
```

```
## fit full and reduced models to simulated data

> b.full <- refit( model.1, simDV[ ,1] )
> b.reduced <- refit( model.0, simDV[ ,1] )
```

model object

new response data

```
## compute chi-squared from these newly estimated models

> chisq.star <- deviance(b.reduced) - deviance(b.full)
```

To help understand the simulation process we will carry out a single bootstrap

```
## simulate data from the reduced model

> simDV <- simulate( model.0 )
```

```
## fit full and reduced models to simulated data

> b.full <- refit( model.1, simDV[ ,1] )
> b.reduced <- refit( model.0, simDV[ ,1] )
```

model object

new response data

```
## compute chi-squared from these newly estimated models

> chisq.star <- deviance(b.reduced) - deviance(b.full)
```

We will combine all of these single-line expressions into a function called `LRT()`

```
> LRT <- function( r, f ){
    simDV <- simulate( r );
    b.full <- refit( f, simDV[ ,1] );
    b.reduced <- refit( r, simDV[ ,1] );
    chisq.star <- deviance( b.reduced ) - deviance( b.full );
    return( chisq.star );
    }
```

The `function()` operator is used to write a new function.

- Each expression is included (and terminated with a semi-colon) between a pair of curly braces.
- The 'r' and 'f' in the parentheses are the arguments for our function ('r' will be the reduced model and 'f' the full model)
- The `return()` function in the last line returns the value of `chisq.star`

Use the LRT() function

```
## Allows reader to replicate results

> set.seed( 101 )



## Use function

> LRT( r = model.0, f = model.1 )
[1] 3.332222
```

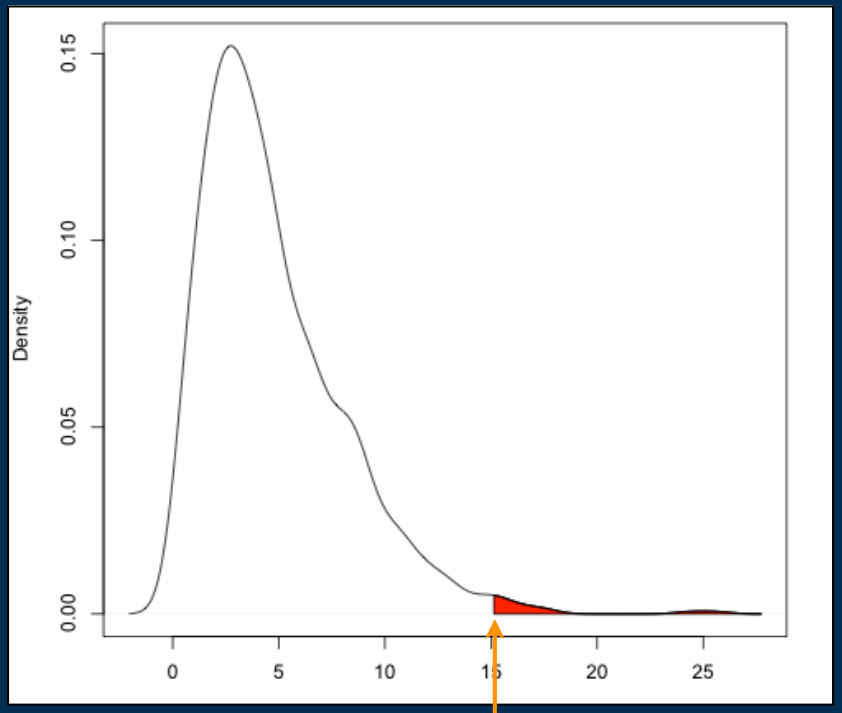
Carry out LRT function a great many times

The `rdply()` function from the **plyr** library evaluates an expression/ function a given number of times and then stores he result in a data frame.

```
## load plyr library

> library( plyr )


## implement LRT function 999 times

> my.boot <- rdply(
    .n = 999,
    .expr = boot.func(r = model.0, f = model.1),
    .progress = "text" )
```

Examine results

```
> head( myboot )

  .n       V1
1  1 9.271173
2  2 7.922311
3  3 3.524140
4  4 3.018975
5  5 2.109137
6  6 6.814550
```



Observed χ^2 (15.105)

```
> plot( density( myboot$V1 ))
```

```
> length( myboot$V1 [myboot$V1 >= 15.105] )
[1] 9
```

Compute bootstrapped *p*-value

Nine (9) of the 999 bootstrapped results were as extreme, or more extreme than the observed χ^2 of 15.105. To compute a bootstrapped *p*-value we compute

$$p_{\text{boot}} = \frac{q+1}{r+1}$$

where *r* is the number of bootstrap replicates in the simulation (999) and *q* are the number of bootstrap replicates at least as extreme as the observed statistic (9)

```
> 10 / 1000
[1] 0.01
```