Статистическое моделирование – введение

Джеймс Н. Стейгер

Отдел Психологии и Человеческое развития

Университет Вандербилт

Многоуровневое Регрессионное Моделирование, 2009

1. Введение

- Когда нам не нужно моделирование;

- Почему нам часто нужно моделирование;

- Основные пути использования моделирования.

2.Оценка доверительного интервала

- Понятие доверительного интервала;

- Простой интервал для пропорции;

- Интервал Вилсона для пропорции.

Поскольку мы уже видели, много ситуаций в статистическом выводе легко обработанные в соответствии с асимптотической нормальной теорией. Рассматриваемые параметры имеют оценки, которые являются или непредубежденными или очень близко к тому, чтобы быть так, и формулы для стандартных ошибок позволяют нам строить доверительные интервалы вокруг этих оценок параметра. Если оценка параметра имеет распространение, которое является разумно близко к его асимптотической нормальности в типовом размере, который мы используем, то доверительный интервал должен выполняться хорошо, в конечном счете.

Однако, к сожалению, много ситуаций, которые не такие простые. Для

примера:

- асимптотическое распространение могло бы быть известно, но конвергенция к нормальности будет крайне медленна;

- мы можем интересоваться некоторой сложной функцией параметров, и мы не имеем статистическую экспертизу получить даже асимптотическое приближение к распространению этой функции.

В ситуациях как эта, мы часто имеем разумного претендента на распространение исходных данных главного процесса, в то время как в то же самое время мы не можем понять распространение количества, мы интересуемся, потому что, то количество - очень сложная функция данных. В

таких случаях, мы можем извлечь существенную выгоду из использования статистического моделирования.

Есть несколько путей, в которых обычно используется статистическое моделирование:

Поколение доверительных интервалов с улучшением. В этом подходе, выборка распространения параметра оценки Θ моделируется, используя выборку, много раз, от текущих данных, и (заново) вычислительный параметр оценивает Θ от каждого 'улучшенного' образца. Изменчивость, показанная многими величинами Θ дает нам подсказку об изменчивости одной оценки Θ полученных от наших данных.

Расследования Монте - Карло показателя статистических процедур. В этом подходе, модель поколения данных и образцовые параметры определены, наряду с типовым размером. Данные произведены согласно модели. Статистическая процедура применена к данным. Этот процесс повторен много раз, и учет ведется, разрешая нам исследовать, как статистическая процедура выполняет при возвращении (известные) истинные оценки параметра.

Поколение приблизительного следующего распространения. В структуре Бэйсин, мы входим в процесс анализа с ‘предшествующим распространением’ параметра, и появляемся от процесса анализа с ‘следующим распространением’, которое отражает наше знание после рассматривания данных. Когда мы видим Θ, мы должны помнить, что это - оценка пункта. После наблюдения этого, мы были бы глупыми, чтобы предположить что Θ = Θ *.

Когда мы думаем об оценке доверительного интервала, это находится часто в контексте механической процедуры, которую мы используем, когда принадлежит к нормальной теории. Таким образом, мы берем оценку параметра и добавляем неподвижное расстояние вокруг нее, приблизительно ±2 стандартных ошибок.

Есть более общее мнение об оценке доверительного интервала, и то есть, доверительный интервал - диапазон оценок параметра, для которого данные не могут отклонить параметр.

Например, рассмотрим традиционный доверительный интервал для простого среднего, когда σ известен. Предположим, что мы знаем, что σ = 15 и N = 25 и мы наблюдаем образец, средний из X* = 105.

Предположим, что мы задаем вопрос, от чего величина μ достаточно далеко от 105 в положительном направлении так, что текущие данные только отклонили бы это? Мы находим эту величину μ - та, которая только производит Z-статистичекий показатель – 1,96.

Мы можем решить за эту величину μ, и это:

$$-1{,}96 = \frac{X* - \mu}{\sigma/\sqrt{N}} = \frac{105 - \mu}{3} \qquad (1)$$

Реконструкция, мы получаем μ = 110,88.

Конечно, мы приучены к получению 110,88 от немного различного и более механического подхода. Дело в том, что одно понятие доверительного интервала - то, что он является диапазоном пунктов, который включает все значения параметра, который не был бы отклонен по условию. Это понятие было продвинуто Е. Вилсоном в начале 1900 гг.

Во многих ситуациях, механический подход согласовывается с ‘зоной приемлемости’ подход, но в небольшом количестве простых ситуаций, методы не согласовываются. В качестве примера, Вилсон описал альтернативный подход к получению доверительного интервала на простой пропорции.

Мы можем иллюстрировать традиционный подход с доверительным интервалом для единственной двучленной типовой пропорции.

Предположим, что мы получаем типовую пропорцию p = 0,65 основанную на типовом размере N = 100.

Приблизительная p стандартная ошибка этой пропорции

$$\sqrt{\frac{0.65(1-0.65)}{100}} = 0.0477$$

Стандартному нормальному доверительному интервалу 95 % теории дали конечные точки

0.65 ± (1.96) (0.0477), таким образом наш доверительный интервал располагается от 0,5565 до 0,7435.

Функция R, чтобы вычислить этот интервал проводит только несколько линий:

```
> simple.interval   function(phat ,N, conf )
+ {
+ z   qnorm(1-(1 -conf )/2)
+ di s t   z * sqrt ( phat * (1 -phat )/N)
+ lower = phat - di s t
+ upper = phat + di s t
+ return( l i s t (lower=lower ,upper=upper))
+ }
```

```
> simple.interval (.65 ,100 , .95)
$lower
[1] 0.5565157
$upper
[1] 0.7434843
```

Подход в предыдущем примере игнорирует факт, что стандартная ошибка оценена от тех же самых данных, используемых, чтобы оценить типовую пропорцию. Подход Вилсона спрашивает, какие значения p являются достаточными далекими от ^p такими, чтобы ^p отклонил бы их. Эти пункты - конечные точки доверительного интервала.

Подход Вилсона требует, чтобы мы решили уравнения.

$$Z = \frac{\hat{p} - p}{\sqrt{p(1-p)/N}} \qquad (2)$$

и

$$-Z = \frac{\hat{p} - p}{\sqrt{p(1-p)/N}} \qquad (3)$$

Следует отметить, что для знаменателя, берется p, не ^p.

Если мы возведем в квадрат оба из вышеупомянутых уравнений, и упростим, определяя $\Theta = Z^2/N$, мы получим

$$(\hat{p} - p)^2 = \theta p(1-p) \qquad (4)$$

Это может быть перестроено в квадратное уравнение с p, которое мы учили как решать по алгебре средней школы по простой формуле. Решение может быть выражено как

$$p = \frac{1}{1+\theta}\left(\hat{p} + \frac{\theta}{2} \mp \sqrt{\hat{p}(1-\hat{p})\theta + \theta^2/4}\right) \qquad (5)$$

Мы можем легко написать функцию R, чтобы осуществить этот результат.

```
> wilson.interval   function(phat ,N, conf )
+ {
+ z   qnorm(1 - (1 -conf )/2)
+ theta   z^2 /N
+ mult   1/(1+ theta )
+ di s t   sqrt ( phat * (1 -phat )* theta + theta ^2 / 4)
+ upper = mult *( phat + theta/2 + di s t )
```

```
+ lower = mult *( phat + theta/2 - di s t )
+ return( l i s t (lower=lower ,upper=upper))
+ }
> wilson.interval (.65 ,100 , .95)
$lower
[1] 0.5525444
$upper
[1] 0.7363575
```

Методы, рассмотренные выше, оба предполагают, что типовое распространение пропорции нормально. В то время как распространение нормально под большим разнообразием обстоятельств, это может существенно отклониться от нормальности, когда N является маленьким или когда p или 1 - p подходы 1. Альтернативный подход к принятию, что распространение оценки является нормальным, состоит в том, чтобы моделировать распространение.

Этот непараметрический подход вовлекает:

- Решить ряд повторений

- Для каждого повторения

• Берут случайную выборку размера N, с заменой от данных

• Вычисляют статистику

• Сохраняют результаты

- когда все повторения закончены, вычисляют 0.975 и 0.025 квантиля в моделируемом распространении оценок

- эти значения - конечные точки 95%-ого доверительного интервала.

Когда данные являются бинарными, процедура моделирования рассматривается выше расчетного периода так, чтобы брать образцы от биномиального распределения с набором p, равным текущей типовой пропорции ^p.

(Примечание: образец Гелман и Хил от нормального распределения в одном из их примеров, но это не необходимо с R). Это вовлекает намного больше вычислительных попыток чем методы, обсужденные предварительно.

```
> bootstrap.interval   function(phat ,N,conf , reps )
+ {
+ lower.p   (1 -conf )/2
+ upper.p   1 - lower.p
+ lower   rep(NA , length( phat ))
+ upper   rep(NA , length( phat ))
+ for (i in 1: length( phat ))
+ {
+ x   rbinom(reps ,N, phat [i])
+ lower[i]   quanti le (x, lower.p ,names=F)/N
+ upper[i]   quanti le (x, upper.p ,names=F)/N
+ }
+ return( l i s t (lower=lower ,upper=upper))
+ }
> bootstrap.interval (.95 ,30 , .95 ,1000)
$lower
[1] 0.8666667
$upper
[1] 1
```

Только что иллюстрированный подход называют 'улучшенный метод процентили’. Следует отметить, что он произведет различные результаты, начиная с различного семени, пока вовлечена случайная ничья.

Во многих ситуациях, интервалы выдадут результаты очень близко друг к другу.

Однако, предположите ^p =:95 и N = 30. Тогда

```
> simple.interval (.95 ,30 , .95)
$lower
[1] 0.8720108
$upper
[1] 1.027989
> bootstrap.interval (.95 ,30 , .95 ,1000)
$lower
[1] 0.8666667
$upper
[1] 1
```

On the other hand,

```
> wilson.interval (.95 ,30 , .95)
$lower
[1] 0.8094698
$upper
```

```
[1] 0.9883682
```

Теперь мы видим, что есть существенная разница между результатами. Вопрос, какой доверительный интервал фактически выполняется лучше?

Есть множество способов характеризовать показатель доверительных интервалов. Например, мы можем исследовать, как близко фактическая вероятность границ к номинальной величине. В этом случае, мы можем легко вычислить точные вероятности границ для каждого интервала, потому что R позволяет нам вычислять точные вероятности от биномиального распределения и N является маленьким. Поэтому, мы можем

- Вычислить каждую возможную величину ^p
- Определить доверительный интервал для этой величины
- Увидеть содержит ли доверительный интервал действительное значение p
- Сложить вероятности для интервалов, которые действительно покрывают p

В функции R ниже мы вычисляем эти вероятности границ для данного N, p, и уровня доверия. (Мы игнорируем факт, что улучшенный интервал может измениться согласно числу повторений и случайной величины семени.)

```
> actual.coverage.probability   function(N,p, conf )
+ {
+ x   0:N
+ phat   x/N
+ probs   dbinom(x,N,p)
+ wilson   wilson.interval (phat ,N, conf )
+ simple   simple.interval (phat ,N, conf )
+ bootstrap   bootstrap.interval (phat ,N,conf ,1000)
+ s 0
+ w 0
+ b 0
+ results   new.env()
+ for (i in 1:N+1) i f (( simple$lower[i] < p)&( simple$upper[i] >p)) s s+ probs [i]
+ for (i in 1:N+1) i f (( wilson$lower[i] < p)&( wilson$upper[i] >p)) w w+ probs [i]
+ for (i in 1:N+1) i f (( bootstrap$lower[i] < p)&( bootstrap$upper[i] >p)) b b+ probs [i]
+ return( l i s t ( simple.coverage =s, wilson.coverage =w, bootstrap.coverage =b))
```

```
+ }
> actual.coverage.probability (30 , .95 ,.95)
$simple.coverage
[1] 0.7820788
$wilson.coverage
[1] 0.9392284
$bootstrap.coverage
[1] 0.7820788
```

Следует отметить, что интервал Вилсона близок к номинальному уровню границ, в то время как традиционные интервалы и улучшенные интервалы выполняются плохо.

Предположим, что мы не поняли, что точные вероятности были доступны для нас. Мы могли все еще получать превосходное приближение точных вероятностей с помощью моделирования Монте Карло.

Моделирование Монте Карло работает следующим образом:

- Выбрать ваши параметры

- Выбрать множество повторений

- Для каждого ответа:

• Выдают данные согласно модели и параметров

• Вычисляют статистический тест или доверительный интервал

• Следят за изменением показателя, например, отклоняет ли статистический тест, или включает ли доверительный интервал истинный параметр

- Показать результаты.

В функции ниже, мы моделируем 10 000 повторений Монте Карло

```
> estimate.coverage.probability function(N,p,conf ,reps , seed.value =12345)
+ {
+ ## Set seed , create empty matrices to hold results
+ set . seed ( seed.value )
+ results   new.env()
+ coverage.wilson 0
+ coverage.simple 0
+ coverage.bootstrap 0
+ ## Loop through the Monte Carlo replications
+ for (i in 1: reps )
```

```
+ {
+ ## create the simulated proportion
+ phat   rbinom(1,N,p)/N
+ ## calculate the intervals
+ wilson   wilson.interval (phat ,N, conf )
+ simple   simple.interval (phat ,N, conf )
+ bootstrap   bootstrap.interval (phat ,N,conf ,1000)
+ ## test for coverage , and update the count if successful
+ i f (( simple$lower < p)&( simple$upper >p))
+ coverage.simple   coverage.simple + 1
+ i f (( wilson$lower < p)&( wilson$upper >p))
+ coverage.wilson   coverage.wilson + 1
+ i f (( bootstrap$lower < p)&( bootstrap$upper >p))
+ coverage.bootstrap   coverage.bootstrap + 1
+
+ }
+ ## convert results from count to probability
+ results$simple   coverage.simple/reps
+ results$wilson   coverage.wilson/reps
+ results$bootstrap  coverage.bootstrap/reps
+ ## return as a named list
+ return( a s . l i s t ( results ))
+ }
> estimate.coverage.probability (30 , .95 ,.95 ,10000)
$bootstrap
[1] 0.7853
$wilson
[1] 0.9381
$simple
[1] 0.788
```

Получить лучшую идею относительно полного показателя двух методов оценки интервала, когда N = 30, мы могли бы, исследуя нормы границ функции p. С нашими написанными функциями, мы все готовы. Мы просто настраиваем вектор значений p, и храним результаты, поскольку мы готовы.

Вот - некоторый код:

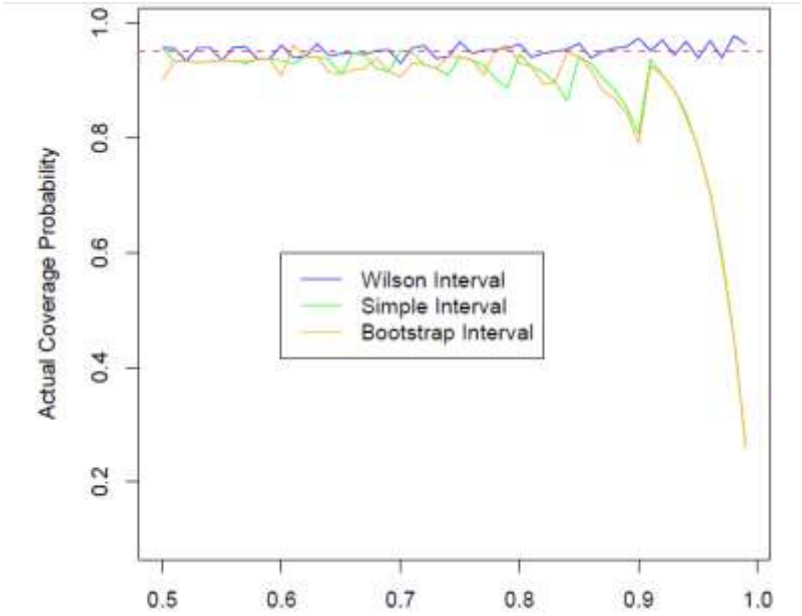
```
> ## set up empty vectors to hold 50 cases
> p   matrix(NA ,50 ,1)
> wilson   matrix(NA ,50 ,1)
> simple   matrix(NA ,50 ,1)
```

```
> bootstrap   matrix(NA ,50 ,1)
> ## step from .50 to .99 , saving results as we go
> for (i in 1:50)
+ {
+ p[i] (49+ i)/100
+ res   actual.coverage.probability (30 ,p[i],.95)
+ wilson [i]   res$wilson.coverage
+ simple [i]   res$simple.coverage
+ bootstrap [i]   res$bootstrap.coverage
+ }
```

Ниже, мы изображаем результаты в виде графика, представляя вероятность границ как функцию p. Преимущество показателя интервала Вилсона очевидно.

```
> plot (p,wilson , type ="l", col =" blue ",
+ ylim =c(.1 ,.99), xlab =" Population Proportion p",
+ ylab =" Actual Coverage Probability ",main =" Confidence Interval Performance (N = 30)")
> l ine s (p,simple , col =" green ")
> l ine s (p, bootstrap , col =" orange ")
> abl ine (.95 ,0, lty =2, col ="red")
> legend(.6 ,.6 ,c(" Wilson Interval "," Simple Interval "," Bootstrap Interval "),
+ col =c(" blue "," green "," orange "),lty=c (1 ,1 ,1))
0.5 0.6 0.7 0.8 0.9 1.0
0.2 0.4 0.6 0.8 1.0
```

Изображение доверительного интервала (N = 30)



Предыдущие примеры демонстрируют, как мы можем использовать моделирование в очень простой ситуации, в двух чрезвычайно различных целях:

- Помогать строить доверительные интервалы имея очевидные данные
- Исследовать показатель статистического теста или процедуры оценки интервала в ситуациях, где параметры 'известны'.

Гелман и Хил именуют первую ситуацию как прогнозирующее моделирование, и второе как поддельное моделирование данных. Ситуации, которые мы исследовали, мы фактически не нуждались в моделировании - лучшие процедуры были доступны.

Моделирование широко используется, потому что, во многих ситуациях, мы не имеем качественной процедуры как интервал Вилсона. Даже когда процедуры могли бы существовать где-нибудь в статистической литературе, мы не могли бы знать о них, или быть в состоянии сделать соответствующую пересадку. В таких ситуациях, моделирование может сэкономить огромное время, все еще давая очень точные ответы на наши вопросы.

Гелман и Хилл представляют библиотеку функций, sim, для быстрого и эффективного моделирования, следующее распределение параметров величин от lm или glm пригодный объект, полученный от предсказания y от k предсказателей в X. Шаги в этой процедуре описаны на странице 143.

- Вычислить β и приблизительную остаточную разницу σ^2 используя стандартное регрессионное приближение.

- Создают n.sims случайное моделирование коэффициента, векторное и остаточное стандартное отклонение, основанное на нормальной теории. Таким образом, для каждого моделирования, создать

1) $\sigma^2 = \hat{\sigma}^2/(\chi^2_{N-k}/(N-k))$

2)Данную случайную ничью σ^2, моделируйте β от многомерного нормального распределения со средним β и матрицей ковариации $\sigma^2 V_\sigma$.

Это распределение представляет следующее распределение для параметров, представляя нашу неуверенность о них. Предположение - то, что предшествующее распределение является неинформативным, то есть, вы не имеете по существу никаких знаний о параметрах по собранным данным.

В Гелман и Хилл глава 5, страницы 86- 88, пример был представлен, вовлекая хорошо-переключающее поведение в Бангладеше. Первая модель предсказала двоичную хорошо-переключающую переменную от единственного предсказателя, расстояние от самого близкого колодца. Фигуры потенциально смешивают, так как один завертывает коэффициенты, полученные от пригонки расстояния в единицах 100 измерителя, другой изображает припадок как функцию расстояния в измерителях. Мы начинаем присоединять данные колодцев.

```
> wells   read. table (" wells.dat ", header = TRUE )
> attach( wells )
> dist100   di s t
```

Затем, мы приспосабливаем логистическую регрессию, используя только расстояние в метрах к ближайшему известному безопасному колодцу. Мы ожидаем, конечно, что вероятность переключения будет обратно пропорционально связана с расстоянием.

```
> fit.1   glm( switch ~ dist , family = binomial ( l ink = " logit "))
> display ( fit.1 , di g i t s = 4)
glm(formula = switch ~ dist, family = binomial(link = "logit"))
coef.est coef.se
(Intercept) 0.6060 0.0603
dist -0.0062 0.0010
---
n = 3020, k = 2
residual deviance = 4076.2, null deviance = 4118.1 (difference = 41.9)
```

Затем, мы моделируем следующее распределение β0 и β1:

```
> sim.1   sim(fit.1 , n.sims =1000)
```

Мы можем подготовить следующее двумерное распределение коэффициентов:

```
> plot ( sim.1$coef [,1], sim.1$coef [,2], xlab = expression(beta [0]) ,
+ ylab = expression(beta [1]))
```

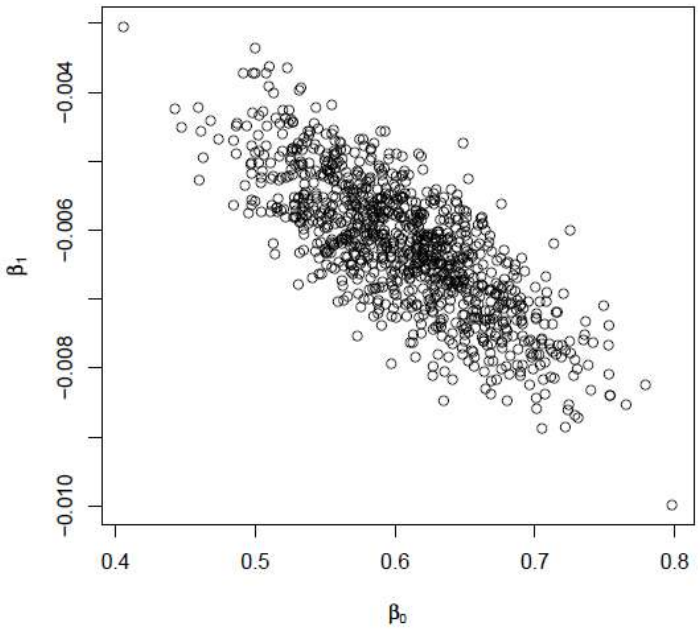
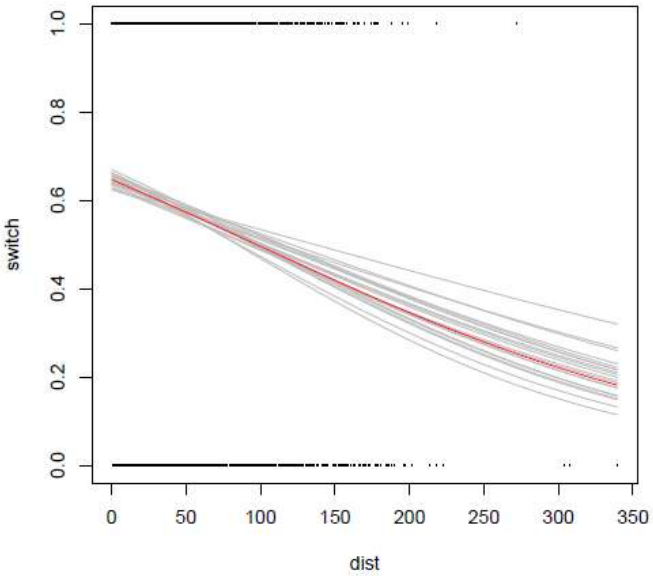


Рисунок 7.6b демонстрирует, как вы можете подготовить неуверенность в уравнении предсказания, готовя кривые, соответствующие значениям от моделируемого следующего распределения. Каждая пара значений соответствует линии заговора. Гелман и Хилл приготовили 20 линий. Я отметил линию от оригинальных данных красным.

```
> plot ( dist ,switch ,pch=".")
> for (s in 1:20)
+ {
+ curve( invlogi t ( sim.1$coef [s ,1] + sim.1$coef [s ,2] *x),
+ col =" gray ",add= TRUE )
+ }
> curve( invlogi t ( fit.1$coef [1] + fit.1$coef [2] *x), col ="red",add= TRUE )
```



На странице 149, Гелман и Хилл обсуждают моделирование неуверенности, которое происходит, предсказывая новые результаты. В этом

примере, они начинают с гипотезы, что есть n*2 матрица X представление величин n новых видов на переменной предсказателя dist. Это - то, что они делают:

- Для каждого моделирования, они предсказывают вероятность переключения использования значений предсказателя в X и величин β от моделирования.

- Тогда, для каждого моделирования, они берут набор образцов из двух предметов (0,1) случайная переменная с вероятностью, равной вероятности переключения от шага (1).

- Так, после 1000 моделирований, каждое новое семейство имеет 1000 (0,1) результатов, каждый основанный на одной величине от (моделируемого) следующего распространения значений β.

- Я предполагаю, что пропорция 1's в заканчивающихся колонках взята как оценка переключающейся вероятности, которая отражает нашу следующую неуверенность в фактическом наклоне и значениях точки пересечения от оригинальных данных.

- Эта заключительная матрица также отражает виды (совсем других) фактических образцов результата, которые могли бы появиться!

Вот - некоторый код:

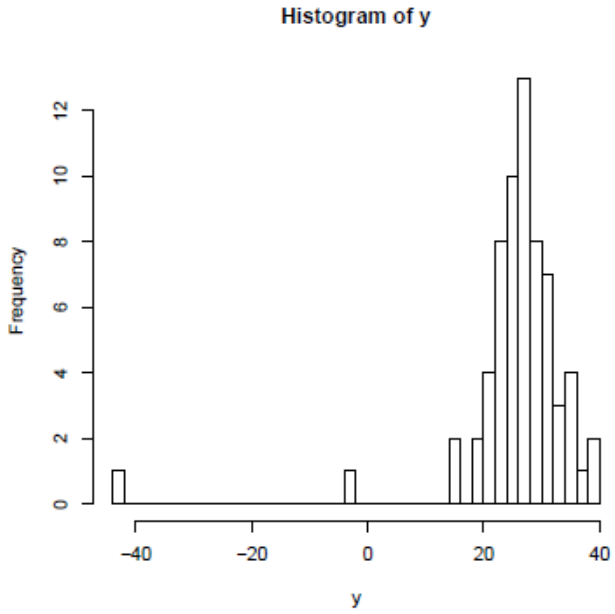
```
> n.sims 1000
> X.tilde   matrix(c (1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,120 ,45 ,109 ,54 > n.tilde   nrow( X.tilde )
> y.tilde   array(NA ,c(n.sims , n.tilde ))
> for (s in 1: n.sims ){
+ p.tilde   invlogi t ( X.tilde %*% sim.1$coef [s ,])
+ y.tilde [s ,]   rbinom( n.tilde ,1, p.tilde )
+ }
> y.tilde [1:20 ,]
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1		1	1	0	1	0	0	1	0	0
[2,]	1		1	1	0	1	0	1	1	0	1
[3,]	0		0	0	1	1	0	1	0	0	1
[4,]	1		1	1	0	1	0	1	0	0	1
[5,]	1		1	0	1	0	0	1	1	1	0
[6,]	1		1	0	1	1	0	0	0	1	0
[7,]	0		1	1	0	1	0	1	0	1	1
[8,]	1		1	0	0	1	0	0	1	0	0
[9,]	0		1	0	1	0	0	1	1	0	1
[10,]	1		0	1	0	1	0	1	1	1	0
[11,]	1		0	0	1	1	1	1	1	1	1
[12,]	1		1	0	0	1	0	1	1	0	1
[13,]	0		1	0	1	0	1	1	0	1	1
[14,]	0		0	0	0	1	0	0	0	1	1

```
[15,] 0    1    1    0    1    0    1    1    1    1
[16,] 0    0    1    0    0    0    1    0    0    1
[17,] 0    1    0    1    0    0    1    1    0    0
[18,] 0    0    0    0    0    0    1    0    0    0
[19,] 0    0    1    1    1    0    0    0    1    1
[20,] 0    0    1    0    1    0    0    0    0    1
```

Поскольку Гелман и Хилл указывают на странице 159, самый фундаментальный способ проверить, что припадок всех аспектов модели должен сравнить копируемые наборы данных с фактическими данными. Этот пример вовлекает копируемые размеры Нюкомб приблизительной скорости света.

```
> y   scan (" lightspeed.dat ", skip =4)
> # plot the data
          > hi s t (y, breaks =40)
```




```
> # fit the normal model
> #(i.e. , regression with no predictors )
> lm.light   lm (y ~ 1)
> display ( lm.light )
lm(formula = y ~ 1)
coef.est coef.se
(Intercept) 26.21 1.32
---
n = 66, k = 1
residual sd = 10.75, R-Squared = 0.00
> n   length (y)
> n.sims   1000
> sim.light   sim ( lm.light , n.sims )
> y.rep   array (NA , c(n.sims , n))
> for (s in 1: n.sims ){
+ y.rep [s ,]   rnorm (1, sim.light$coef [s], sim.light+ }
> # gather the minimum values from each sample
>
> test   function (y){
+ min (y)
+ }
```

```
> test.rep   rep (NA , n.sims )
> for (s in 1: n.sims ){
+ test.rep [s]   test ( y.rep [s ,])
+ }
> # plot the histogram of test statistics of replications and of actual data
>
> hi s t ( test.rep , xlim =range ( test (y), test.rep ))
> l ine s (rep ( test (y), 2), c(0,n), col ="red")
```