Suppose that you have a sample of a variable of interest, e.g. the heights of men in certain population, and for some obscured reason you are interest not in the mean height μ but in its square μ². How would you inference on μ², e.g. test a hypothesis or calculate a confidnce interval? The delta method is the trick you need.

The delta method is mathematical assertion that can yield estimates for the varinance of functons of statistics under mild condition. Loosley speaking, let bₙ is an estimate of β, where n is the sample size, and the distribution of bₙ is approximately normal with mean β and variance σ²/n.

Then, if g is a function, then g(bₙ) is approximately normal with mean g(β) and and variance [g’(β)² ]⋅σ²/n, provided that the sample size is large. Don’t let the calculus scare you. R’s deltmethod function will handle the calculus for you.

If bₙ is a [Maximum Likelihood Estimate](https://en.wikipedia.org/wiki/Maximum_likelihood_estimation) (MLE) of β then under mild conditions it is approximately normal for a large sample size, and g(bₙ) and g’(bₙ) are MLEs for g(β) and g’(β) respectively. Add on top of this a MLE for σ, and you can implement statistical inference.

I will demonstrate the use of the delta method using the Titanic survival data. For the sake of the example I will use only 3 variables: Survived is a 0/1 variable indicating whther a passenger survived the sinking of the Titanic, Age is obviously the passenger’s age, and Parch is the passenger’s number of parents/children aboard. Lets take a look at some of the data:

> # install.packages("titanic")

> library(titanic)

> titanic=titanic\_train[, c(1, 2, 6,8)]

> head(titanic)

  PassengerId Survived Age Parch

1           1        0  22     0

2           2        1  38     0

3           3        1  26     0

4           4        1  35     0

5           5        0  35     0

6           6        0  NA     0

>

Let π be the probability of surviving. Then the odds of survival is π/(1-π). The sample size is n=891 is considered large, so we can apply the [Central Limit Theorem](https://en.wikipedia.org/wiki/Central_limit_theorem) to conclude that p is approximately normal with mean π and variance π/(1-π)/n. Then π can be estimated by p=0.384, the odds are estimated by p/(1-p)=0.623, and the variance of p is estimated by 0.000265:

> p\_survival=mean(titanic$Survived)

> print(p\_survival)

[1] 0.3838384

> survival\_odds=p\_survival/(1-p\_survival)

> print(survival\_odds)

[1] 0.6229508

> n=nrow(titanic)

> var\_p\_survival=(p\_survival\*(1-p\_survival))/n

> print(var\_p\_survival)

[1] 0.0002654394

>

Let g(*x*)=p/(1-*x*). Then g`(*x*)=1/(1-*x*)², (you can check this at [https://www.wolframalpha.com](https://www.wolframalpha.com/)) so the variane of the odds can be estimated by

Var(p/(1-p))=[g’(p)² ]⋅σ²/n=[1/(1-p)²]² ⋅ [p(1-p)/n].

This can be implemented in the following R code:

> dev\_g\_odds=1/(1-survival\_odds)^2

> var\_odds=(dev\_g\_odds^2)\*var\_p\_survival

> print(var\_odds)

[1] 0.01313328

> se\_odds=sqrt(var\_odds)

> print(se\_odds)

[1] 0.1146005

>

But of course, instead of doing all the calculus, you can use the deltamethod function of R’s msm package.

The function has three parameters:

* g is a formula object representating of the transformation g(*x*). The formula variables must be labeled x1, x2 and so on.
* mean is the estimate of g(β)
* cov is the estimate of var(β) which is σ²/n

The use of the delta function provides the same result:

> # install.packages("msm")

> library(msm)

> se\_odds\_delta=deltamethod(g=~x1/(1-x1), mean=survival\_odds, cov=var\_p\_survival)

> print(se\_odds\_delta)

[1] 0.1146005

>

The second example considers [logistic regression](https://en.wikipedia.org/wiki/Logistic_regression). We will model the (logit of the) probability of survival using Age nd Parch. Using R’s glm function we can obtain estimates to the logistic regression coefficients band their standard errors se\_b:

> model=glm(Survived ~ Age + Parch, family=binomial(link="logit")

> model\_beta=data.frame(summary(model)$coefficients[2:3, 1:2])

> names(model\_beta)=c("b", "se\_b")

> print(model\_beta)

                 b        se\_b

Age   -0.008792681 0.005421838

Parch  0.191531303 0.090643635

>

Since *exp*(β) is usually interpreted as the odd ratio (OR) of the response variable with regard to the regression independent variable, reseachers are interested in inference on *exp*(β). In order to perform such inference one nees to estimate the standard error of *exp*(β). Since b is a Maximum Likelihood Estimate for β, it is approximately normal and its variance is estimated by se\_b², and the delta method can be applied.

The calculus in this case is easy: g(*x*)=*exp*(*x*), so g’(*x*)=*exp*(*x*). Therefore, the standard error of *exp*(β) can be estimated by *exp*(b)⋅ se\_b:

> model\_beta$OR=exp(model\_beta$b)

> model\_beta$se\_OR=exp(model\_beta$b)\*model\_beta$se\_b

> print(model\_beta)

                 b        se\_b        OR       se\_OR

Age   -0.008792681 0.005421838 0.9912459 0.005374374

Parch  0.191531303 0.090643635 1.2111027 0.109778755

>

To use the deltamethod function, we will first use the vcov function to obtain the variance-covariance matrix of the resression coefficient estimates, and the variances will be the inputs of the deltamethod function:

> vc=vcov(model)[2:3, 2:3]

> print(vc)

               Age        Parch

Age   2.939632e-05 9.236876e-05

Parch 9.236876e-05 8.216269e-03

> model\_beta$se\_OR\_delta[1]=deltamethod(~exp(x1), mean=model\_beta$b[1], cov=vc[1,1])

> model\_beta$se\_OR\_delta[2]=deltamethod(~exp(x1), mean=model\_beta$b[2], cov=vc[2,2])

> print(model\_beta)

                 b        se\_b        OR       se\_OR se\_OR\_delta

Age   -0.008792681 0.005421838 0.9912459 0.005374374 0.005374374

Parch  0.191531303 0.090643635 1.2111027 0.109778755 0.109778755

>

Of course, we get the same results.