APPLIED STATISTICS

Bootstrap

Dr Tao Zou

Research School of Finance, Actuarial Studies & Statistics
The Australian National University

Last Updated: Wed Oct 18 08:13:05 2017

Overview

- Simulation $\boxed{\text{Monte Carlo}}$

- Bootstrap

1. Bootstrap standard errors.

2. Bootstrap confidence intervals.

References

1. **B. Efron and R.J. Tibshirani** (1994)
   An Introduction to the Bootstrap
2. **P. Hall** (2013)
   The Bootstrap and Edgeworth Expansion
3. The slides are made by **R Markdown**.
   `http://rmarkdown.rstudio.com`

Review: Poisson Log-Linear Regression Model Assumptions

1. **Poisson distribution**: There is a Poisson distributed (sub)population of responses Z for given values of the explanatory variables $(X_1 = x_1, \cdots, X_k = x_k)$. That means if we let $X = (X_1, \cdots, X_k)$, the probability that $Z = z$ given X is

$$\mathrm{P}(Z = z) = \frac{e^{-\mu}\mu^z}{z(z-1)\cdots 1}, \text{ where } z = 0, 1, 2, \cdots.$$

   Based on the properties of the Poisson distribution, the mean of response Z is given by

$$\mu\{Z|X\} = \mu.$$

2. **Generalised Linearity**: The transformation of the mean of response μ falls on a linear function of the explanatory variables

$$g(\mu) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k, \text{ for } X = (X_1, \cdots, X_k),$$

   where $g(u) = \log(u)$, which is the log link function.

Poisson Log-Linear Regression Model Assumptions (Con'd)

Remark: $\mu = e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k}$.

3. **Independence**: Observations

$$(X_{1,1}, \cdots X_{k,1}, Z_1),$$

$$\vdots$$

$$(X_{1,m}, \cdots X_{k,m}, Z_m),$$

are independent, where m is the sample size.

Simulation

Lily wants to use R to generate random samples based on the Poisson log-linear regression model assumptions. That just means the random samples follow a Poisson log-linear regression model. She follows the steps below.

STEP 1: Specify $\beta_0 = 2$ and $\beta_1 = 1$.

```
rm(list=ls())
beta0=2;beta1=1
```

STEP 2: Suppose the observations $X_1, \cdots, X_m$ are $0.1, 0.2, 0.3, \cdots, 1.0$, so the number of observations $m = 10$ (**small sample size** m).

(1, 2, 3 ⋯ 10)

```
X=(1:10)/10
m=length(X)
```

STEP 3: Compute $\mu_i = e^{\beta_0+\beta_1 X_i}$ for $i = 1, \cdots, m$. m means

```
mu=exp(beta0+beta1*X)
```

→ vector with 10 values

→ generalised linearity assumption

Simulation (Con'd)

Step3

Step 4: Generate Z_i independently from the Poisson distribution with its mean μ_i. $i=1,\dots m$.

$X_i \to \mu_i \to Z_i \sim \text{Poisson}(\mu_i)$

10 values   10 values

```
#space to store the different realisations of response Z
Z=rep(0,m)
set.seed(2)
for (i in 1:m){
    Z[i]=rpois(1, mu[i])
}
Z
```

returns 1 value

mu ↓ a vector of 10 values

Z X

```
## [1]  6 10 10  7  8 13 12 15 26 19
```

Step 5: Repeat the above step 1 more time.

```
for (i in 1:m){
    Z[i]=rpois(1, mu[i])
}
Z
```

Z X

```
## [1]  9  8 13  7  8  9 18 16 22 22
```

Is the data generated this time the same?

Simulation (Con'd)

STEP 6: Repeat Step 4 1,000 times and obtain 1,000 different datasets (also called **repeated samples**) of $\{Z_i, X_i\}_{i=1}^{m}$.

```
R=1000
#space to store the different datasets
Zdata=matrix(0,ncol=m,nrow=R)
Xdata=matrix(0,ncol=m,nrow=R)
set.seed(2)
for (r in 1:R){
  for (i in 1:m){
      Z[i]=rpois(1, mu[i])
  }
  Zdata[r,]=Z
  Xdata[r,]=X
}
```

each row corresponds to one dataset of Z

for generating 1000 datasets

to generate one dataset

Poisson dist. assumption

indpt assumption

Response Z for the 1st dataset:

```
Zdata[1,]
```

```
##  [1]  6 10 10  7  8 13 12 15 26 19
```

10 values

Response Z for the 2nd dataset:

```
Zdata[2,]
```

```
##  [1]  9  8 13  7  8  9 18 16 22 22
```

...

/32

Simulation (Con'd)

For Lily, she has the 1,000 datasets now, and also she knows the true values of $\beta_0 = 2$ and $\beta_1 = 1$.

Lei Li is a friend of Lily. Lily hands over the above 1,000 datasets to him, but she does not tell him the true values of β_0 and β_1.

For Lei Li, he has the 1,000 datasets, but he does not know the true values of β_0 and β_1.

For each dataset, Lei Li fits a Poisson log-linear model and computes the 1,000 different MLEs $\hat{\beta}_0$ and $\hat{\beta}_1$.

```
#space to store the different MLEs
hatbeta0=rep(0,R)
hatbeta1=rep(0,R)
for (r in 1:R){
  fit.pois=glm(Zdata[r,]~Xdata[r,],family=poisson(link=log))
  hatbeta0[r]=fit.pois$coef[1]
  hatbeta1[r]=fit.pois$coef[2]
}
```

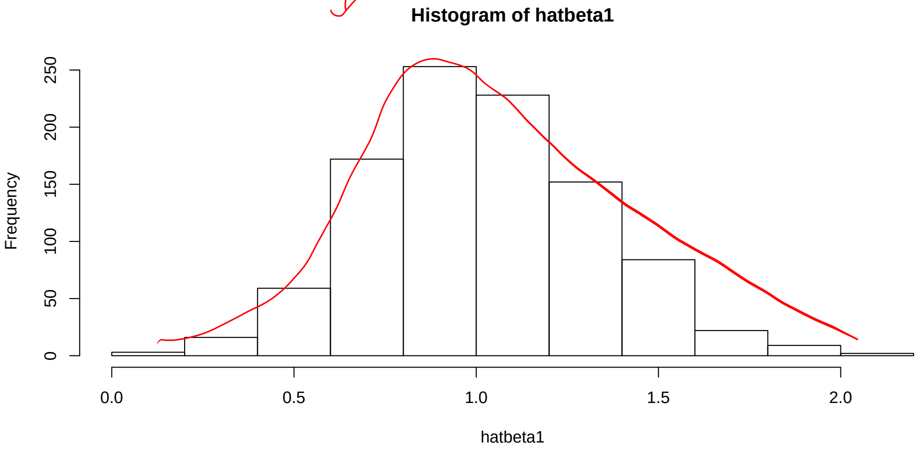
keep 1000 spaces to save $\hat{\beta}_0$ for 1000 datasets

r-th dataset

Simulation (Con'd)

The sampling distribution of $\hat{\beta}_1$ can be approximated via the $R = 1000$ different estimates of $\hat{\beta}_1$.

```r
hist(hatbeta1)
```



Mean

The mean of $\hat{\beta}_1$ is determined by the sampling distribution. Hence, the mean can also be approximated via the $R = 1000$ different estimates of $\hat{\beta}_1$.

```
mean(hatbeta1)
```

```
## [1] 1.016003
```

is an approximate of $\mathrm{E}\hat{\beta}_1$.

Lily knows the true values $\beta_1 = 1$. She looks at this result obtained by Lei Li and she can conclude that even though now we only have $m = 10$ observations, the MLE of $\hat{\beta}_1$ is still roughly unbiased if all the assumptions in the Poisson log-linear model are satisfied, i.e.

$$\mathrm{E}\hat{\beta}_1 \approx \beta_1.$$

Lei Li

Lily $\beta_1 = 1$

Standard Deviation

The standard deviation of $\hat{\beta}_1$ is also determined by the sampling dstribution. Hence, the standard deviation can also be approximated via the $R = 1000$ different estimates of $\hat{\beta}_1$.

```
sd(hatbeta1)
```

```
## [1] 0.3157341
```

is an approximate of $\boxed{\mathrm{SD}(\hat{\beta}_1)}$.

It is worth noting that the standard deviation measures how accurate the MLE $\hat{\beta}_1$ is.

Standard Deviation (Con'd)

Recall that for GLM, we never give the formula for the standard deviation of the MLE. We only give the approximate sampling distribution and the approximate standard error for MLE **when the sample size m is large enough.**

Here obviously the sample size $m = 10$ is not large enough. Even though Lily knows the true values of β_0 and β_1, she does not know the standard deviation since there is really no formula for it.

The best way to approximate the standard deviation is to use the above simulaiton and obtain

$$\mathrm{SD}(\hat{\beta}_1) \approx 0.3157.$$

Benefits of Simulation

Show sampling distributions of estimation by using the histogram.

Approximate the mean and the standard deviation of the estimation.

In the above benefits, no need of knowing the formulas for sampling distribution, mean, variance/standard deviation and etc. But $R = 1000$ **repeated samples** are required.

Other Benefits Mentioned in This Course

Use simulation to better understand statistical concepts:

- Never accepting H_0 in Lecture Notes 2.
- The meaning of 95% CI in Assignment 1.

Show heavy-tailed and skewed Q-Q plots and the Q-Q plot with outliers in Lecture Notes 3.

Show the violation of the constant variance assumption, and the usefulness of weighted regression in Lecture Notes 7.

Real Data and Simulation

James is a friend of Lei Li. Lei Li **only** hands over **the 1st dataset** to him.

has

```
Z=Zdata[1,];Z
```

```
##  [1]  6 10 10  7  8 13 12 15 26 19
```

```
X=Xdata[1,];X
```

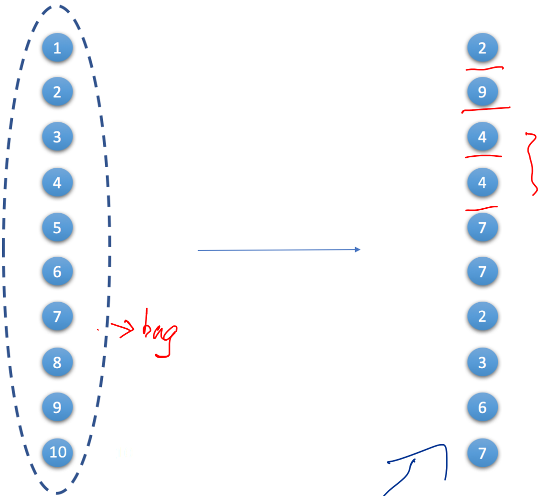
```
##  [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

When we deal with the real data, we play the role of James.

1. James does not know the true values of the parameters β_0 and β_1. Only Lily knows.
2. James only has one sample $\{Z_i, X_i\}_{i=1}^m$ observed, instead of 1,000 datasets analysed by Lei Li.

Based on the data (one sample) $\{Z_i, X_i\}_{i=1}^m$, if he can find a way to randomly generate $R = 1000$ repeated "pseudo samples" without knowing the true values of the parameters β_0 and β_1, can he utilize the similar idea of simulation to approximate the standard deviation of estimation?

Drawing Balls with Replacement (1st Time)

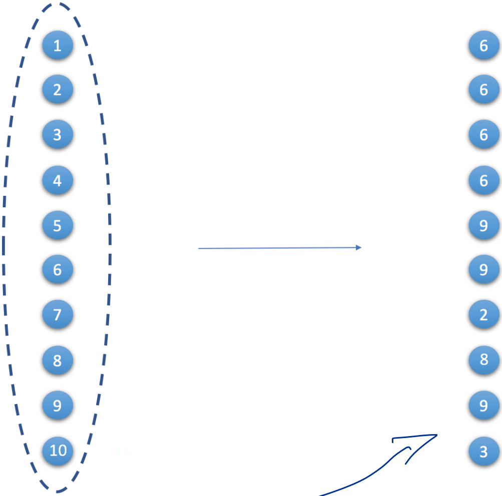


m=10

```
set.seed(3)
sample(1:m,size=m,replace=T)
```

```
## [1] 2 9 4 4 7 7 2 3 6 7
```

Drawing Balls with Replacement (2nd Time)




```
sample(1:m,size=m,replace=T)
```

```
## [1] 6 6 6 6 9 9 2 8 9 3
```

We can keep doing the above procedure.

Bootstrap Sample

The bootstrap is a computationally intensive method based on the idea of randomly drawing **bootstrap samples** with replacement from $\{Z_i, X_i\}_{i=1}^m$ (equivalently to randomly drawing balls with replacement from $\{1, 2, \cdots, m\}$ and here $m = 10$).

Example of 1st Bootstrap Sample

$$
\begin{array}{l}
Z_1, X_1 \\
Z_2, X_2 \\
Z_3, X_3 \\
Z_4, X_4 \\
Z_5, X_5 \\
Z_6, X_6 \\
Z_7, X_7 \\
\dots \\
Z_{10}, X_{10}
\end{array}
\qquad\qquad
\begin{array}{l}
Z_2, X_2 \\
Z_9, X_9 \\
Z_4, X_4 \\
Z_4, X_4 \\
Z_7, X_7 \\
Z_7, X_7 \\
Z_2, X_2 \\
\dots \\
Z_7, X_7
\end{array}
\quad \xrightarrow{\text{written as}} \quad
\begin{array}{l}
Z_1^{*(1)}, X_1^{*(1)} \\
Z_2^{*(1)}, X_2^{*(1)} \\
Z_3^{*(1)}, X_3^{*(1)} \\
Z_4^{*(1)}, X_4^{*(1)} \\
Z_5^{*(1)}, X_5^{*(1)} \\
Z_6^{*(1)}, X_6^{*(1)} \\
Z_7^{*(1)}, X_7^{*(1)} \\
\dots \\
Z_{10}^{*(1)}, X_{10}^{*(1)}
\end{array}
$$

```
set.seed(3)
bootind=sample(1:m,size=m,replace=T);Z[bootind]
```

```
## [1] 10 26  7  7 12 12 10 10 13 12
```

```
X[bootind]
```

$X = 0.1, 0.2 \dots 1.0$

```
## [1] 0.2 0.9 0.4 0.4 0.7 0.7 0.2 0.3 0.6 0.7
```

Example of 2nd Bootstrap Sample

$$
\begin{array}{lll}
Z_1, X_1 & Z_6, X_6 & & Z_1^{*(2)}, X_1^{*(2)} \\
Z_2, X_2 & Z_6, X_6 & & Z_2^{*(2)}, X_2^{*(2)} \\
Z_3, X_3 & Z_6, X_6 & & Z_3^{*(2)}, X_3^{*(2)} \\
Z_4, X_4 & Z_6, X_6 & & Z_4^{*(2)}, X_4^{*(2)} \\
Z_5, X_5 & Z_9, X_9 & \xrightarrow{\text{written as}} & Z_5^{*(2)}, X_5^{*(2)} \\
Z_6, X_6 & Z_9, X_9 & & Z_6^{*(2)}, X_6^{*(2)} \\
Z_7, X_7 & Z_2, X_2 & & Z_7^{*(2)}, X_7^{*(2)} \\
\dots & \dots & & \dots \\
Z_{10}, X_{10} & Z_3, X_3 & & Z_{10}^{*(2)}, X_{10}^{*(2)}
\end{array}
$$

```
bootind=sample(1:m,size=m,replace=T);Z[bootind]
```

```
## [1] 13 13 13 13 26 26 10 15 26 10
```

```
X[bootind]
```

```
## [1] 0.6 0.6 0.6 0.6 0.9 0.9 0.2 0.8 0.9 0.3
```

We can keep doing the above procedure, and we obtain

R **repeated bootstrap samples** (bootstrap datasets):

$$\{Z_i^{*(1)}, X_i^{*(1)}\}_{i=1}^m, \ldots, \{Z_i^{*(R)}, X_i^{*(R)}\}_{i=1}^m,$$

R=1000

m=10

```
R=1000
#space to store the different datasets
Zstardata=matrix(0,ncol=m,nrow=R)
Xstardata=matrix(0,ncol=m,nrow=R)
set.seed(3)
for (r in 1:R){
  bootind=sample(1:m,size=m,replace=T)
  Zstardata[r,]=Z[bootind]
  Xstardata[r,]=X[bootind]
}
```

Have a look at 1st and 2nd bootstrap samples:

```
Zstardata[1,]
```

```
##  [1] 10 26  7  7 12 12 10 10 13 12
```

```
Xstardata[1,]
```

```
##  [1] 0.2 0.9 0.4 0.4 0.7 0.7 0.2 0.3 0.6 0.7
```

```
Zstardata[2,]
```

```
##  [1] 13 13 13 13 26 26 10 15 26 10
```

```
Xstardata[2,]
```

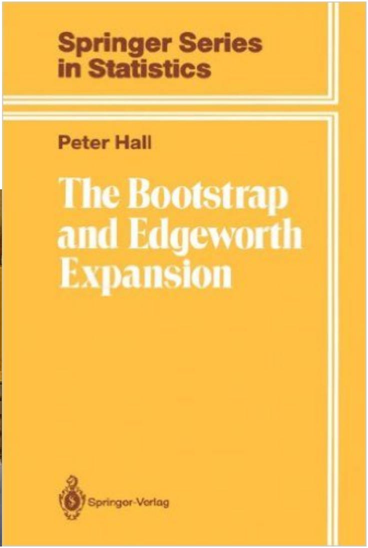
```
##  [1] 0.6 0.6 0.6 0.6 0.9 0.9 0.2 0.8 0.9 0.3
```

Bootstrap Sample (Con'd)

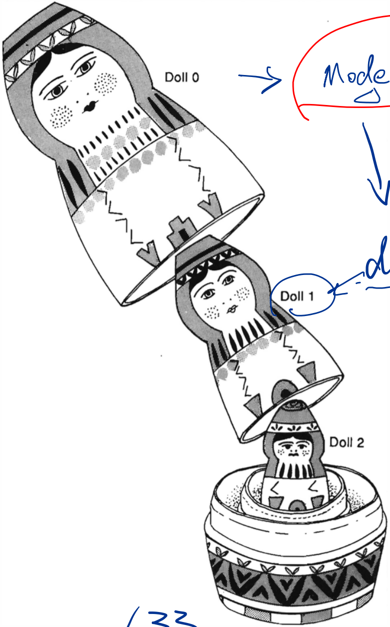
Based on the data (one sample) $\{Z_i, X_i\}_{i=1}^{m}$, James uses this way to randomly generate $R = 1000$ repeated bootstrap samples without knowing the true values of the parameters β_0 and β_1.

Now he also has 1,000 bootstrap datasets from this bootstrap method. However, obviously these datasets are different from the ones that Lei Li has.

Prof Peter Hall (1951-2016, Australian National University 1978-2005)



Prof Peter Hall's Comments on Bootstrap



Model

β_0, β_1

simulation

data / sample $\{z_i, x_i\}_{i=1}^n$

bootstrap

bootstrap sample $\{z_i^*, x_i^*\}$

/33

Bootstrap Standard Error

Question: can James utilize the similar idea of simulation to approximate the standard deviation of estimation based on his 1,000 bootstrap datasets from this bootstrap method? The answer is yes!

For $R = 1000$ bootstrap datasets,

$$\{Z_i^{*(1)}, X_i^{*(1)}\}_{i=1}^m, \cdots, \{Z_i^{*(R)}, X_i^{*(R)}\}_{i=1}^m,$$

James fits Poisson log-linear models and computes the 1,000 different MLEs

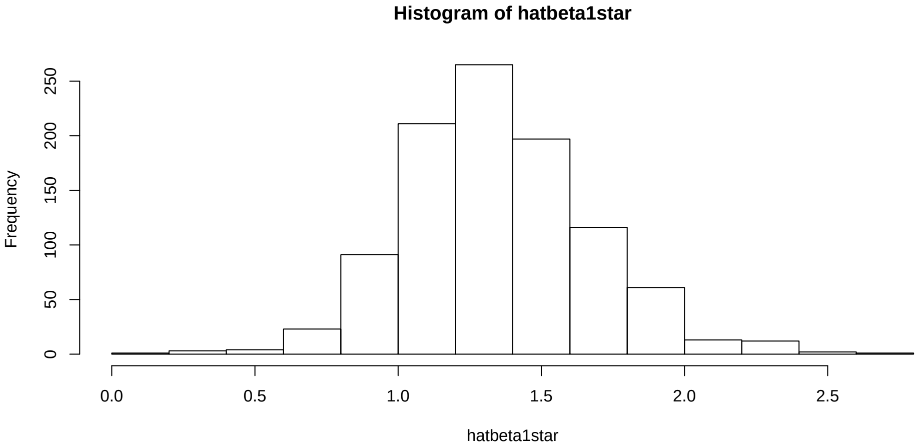
$$(\hat{\beta}_0^{*(1)}, \hat{\beta}_1^{*(1)}), \cdots, (\hat{\beta}_0^{*(R)}, \hat{\beta}_1^{*(R)}), \text{respectively.}$$

```
#space to store the different MLEs
hatbeta0star=rep(0,R)
hatbeta1star=rep(0,R)
for (r in 1:R){
  fit.pois=glm(Zstardata[r,]~Xstardata[r,],family=poisson(link=log))
  hatbeta0star[r]=fit.pois$coef[1]
  hatbeta1star[r]=fit.pois$coef[2]
}
```

Bootstrap Standard Error (Con'd)

The sampling distribution of $\hat{\beta}_1$ can be approximated via the $R = 1000$ different $\hat{\beta}_1^{*(1)}, \hat{\beta}_1^{*(2)}, \cdots, \hat{\beta}_1^{*(R)}$. $\rightarrow$ James has one dataset

```
hist(hatbeta1star)
```



Bootstrap Standard Error (Con'd)

The standard deviation of $\hat{\beta}_1$ is determined by the sampling dstribution. Hence, the standard deviation can also be approximated via the $R = 1000$ different $\hat{\beta}_1^{*(1)}, \hat{\beta}_1^{*(2)}, \cdots, \hat{\beta}_1^{*(R)}$:

one dataset

```
sd(hatbeta1star)
```

James

1000 bootstrap datasets

```
## [1] 0.3262645
```

is an approximate of $\mathrm{SD}(\hat{\beta}_1)$. It is worth noting that this result based on $R = 1000$ **repeated bootstrap samples** from James, is very close to

```
sd(hatbeta1)
```

1000 datasets

```
## [1] 0.3157341
```

Lei Li

based on $R = 1000$ **repeated samples** ~~analsed~~ by Lei Li.

analysed

Bootstrap Standard Error (Con'd)

We call

```r
sd(hatbeta1star)
```

```
## [1] 0.3262645
```

the bootstrap standard error of $\hat{\beta}_1$, denoted by $\mathrm{SE}_b(\hat{\beta}_1)$, which provides a good estimate of $\mathrm{SD}(\hat{\beta}_1)$ when the **the sample size m is very small.**

$m=10$

Comparison between $\mathrm{SE}_a(\hat{\beta}_1)$ and $\mathrm{SE}_b(\hat{\beta}_1)$

Recall that the summary(glm()) output of R provides the approximate standard error $\mathrm{SE}_a(\hat{\beta}_1)$, which is determined by a formula from complicated mathematical inductions for GLM.

$\mathrm{SE}_a(\hat{\beta}_1)$ is a good estimate of $\mathrm{SD}(\hat{\beta}_1)$ **when the sample size m is large enough**.

In this case, since $m = 10$ is small, $\mathrm{SE}_a(\hat{\beta}_1)$ cannot be used.

The bootstrap standard error $\mathrm{SE}_b(\hat{\beta}_1)$ provides an accurate estimation of $\mathrm{SD}(\hat{\beta}_1)$ **even if the sample size m is small**.

One can also show that $\mathrm{SE}_b(\hat{\beta}_1)$ still provides an accurate estimation of $\mathrm{SD}(\hat{\beta}_1)$ **when the sample size m is large**.

Sample Size	Standard Error for Parameter Estimation of GLM
m is small	SE_b is better.
m is large	Both SE_a and SE_b can be used.

Benefits of Bootstrap I

Bootstrap inherits the benefits from simulation.

1. Estimate sampling distributions of paramter estimation.

2. Estimate the standard deviation of the estimation.

In the above benefits, ~~no need of~~ knowing the formulas for sampling distribution and standard deviation/standard error (the appoximate standard error in GLM requires a formula from complicated mathematical inductions).

$\rightarrow$ glm()

The sample size can be small or large. It doesn't matter.

But $R = 1000$ **repeated bootstrap samples** are required, and hence bootstrap is computationally intensive.

$m = 100\,000\,000$

$\rightarrow SE_a(\hat{\beta}_1) \rightarrow$ R output

Review: Confidence Intervals (CI) for β_j in GLM

$\frac{X-\mu}{\sigma} \sim N(0,1)$

$\Downarrow$

$X - \mu \sim N(0, \sigma^2)$

Recall the practical sampling distributions of $\hat{\beta}_j$:

$$\frac{\hat{\beta}_j - \beta_j}{\mathrm{SE}_a(\hat{\beta}_j)} \overset{a}{\sim} N(0,1), \text{ for } j = 0, \cdots, k,$$

when **the sample size** m **is large** for a GLM with k explanatory variables.

Using this information, a $(1-\alpha)$ CI for β_j is

$$\hat{\beta}_j \mp z_{\alpha/2} \times \mathrm{SE}_a(\hat{\beta}_j)$$

where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of $N(0,1)$, namely

$$\mathrm{P}(Z \le z_{\alpha/2}) = 1 - \alpha/2 \text{ or } \mathrm{P}(Z > z_{\alpha/2}) = \alpha/2$$

for $Z \sim N(0,1)$.

Interpretation of Confidence Intervals

m is large $\hat{\beta}_j - \beta_j$ follows

The above indicates that dist.

We want ~~to know~~ dist. of $\hat{\beta}_j - \beta_j$

$$\boxed{\hat{\beta}_j - \beta_j} \overset{a}{\sim} N\left(0, \mathrm{SE}_a^2(\hat{\beta}_j)\right), \text{ for } j = 0, \cdots, k,$$

m is small
We do not know the dist. of $\hat{\beta}_j - \beta_j$

when **the sample size** m is large for a GLM with k explanatory variables.

$P(\hat{\beta}_j - \beta_j \le z_{\alpha/2} SE_a)$ dist. of $\hat{\beta}_j - \beta_j$

One can verify that $z_{\alpha/2}\mathrm{SE}_a(\hat{\beta}_j)$ is the $1 - \alpha/2$ quantile of $N(0, \mathrm{SE}_a^2(\hat{\beta}_j))$, and $-z_{\alpha/2}\mathrm{SE}_a(\hat{\beta}_j)$ is the $\alpha/2$ quantile of $N(0, \mathrm{SE}_a^2(\hat{\beta}_j))$.

dist. of $\hat{\beta}_j - \beta_j$

As a consequence, $P(\hat{\beta}_j - \beta_j \le -z_{\alpha/2} SE_a)$

$$\mathrm{P}(-z_{\alpha/2}\mathrm{SE}_a(\hat{\beta}_j) \le \boxed{\hat{\beta}_j - \beta_j} \le z_{\alpha/2}\mathrm{SE}_a(\hat{\beta}_j)) \approx 1 - \alpha,$$

which leads to the $(1 - \alpha)$ CI for β_j

$1-\alpha/2$ quantile

$$\hat{\beta}_j \mp z_{\alpha/2}\mathrm{SE}_a(\hat{\beta}_j).$$

$\hat{\beta}_j - (-z_{\alpha/2} SE_a)$

$\alpha/2$ quantile

Efron's Bootstrap Percentile CI

Question: can James construct **a different CI** based on his 1,000 bootstrap datasets? The answer is yes!

For the original dataset that Lei Li hands over to James, James fits a Poisson log-linear model and computes $\hat{\beta}_1$.

```
fit.pois=glm(Z~X,family=poisson(link=log))
fit.pois$coefficients
```

```
## (Intercept)           X
##    1.711635    1.358360
```

```
hatbeta1=fit.pois$coefficients[2]
hatbeta1
```

```
##       X
## 1.35836
```

Efron's Bootstrap Percentile CI (Con'd)

For $R = 1000$ bootstrap datasets,

What is the dist. of

$$\{Z_i^{*(1)}, X_i^{*(1)}\}_{i=1}^m, \cdots, \{Z_i^{*(R)}, X_i^{*(R)}\}_{i=1}^m, \quad \boxed{\hat{\beta}_1 - \beta_1}$$

James fits Poisson log-linear models and computes the 1,000 different

plug-in idea

$$\hat{\beta}_1^{*(1)} - \hat{\beta}_1, \cdots, \hat{\beta}_1^{*(R)} - \hat{\beta}_1, \text{ respectively.}$$

$\hat{\beta}_1^* - \hat{\beta}_1$

```
#space to store the different MLEs
hatbeta0star=rep(0,R)
hatbeta1star=rep(0,R)
for (r in 1:R){
  fit.pois=glm(Zstardata[r,]~Xstardata[r,],family=poisson(link=log))
  hatbeta0star[r]=fit.pois$coef[1]
  hatbeta1star[r]=fit.pois$coef[2]
}
Sstar=hatbeta1star-hatbeta1
```

1000 values

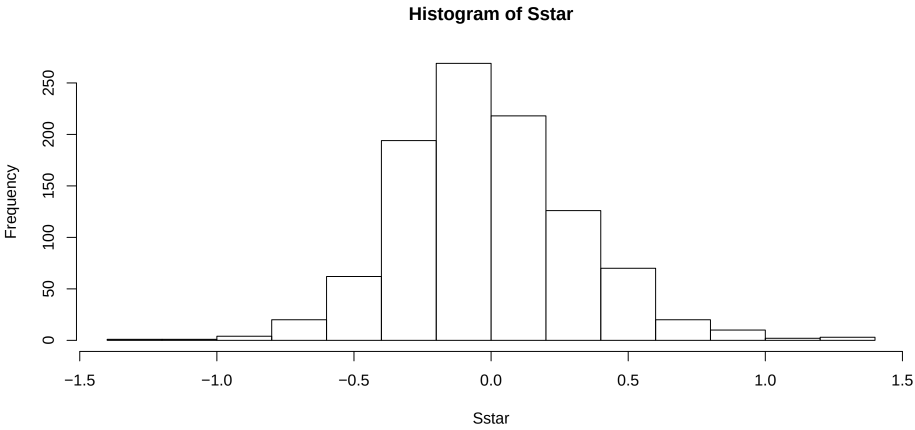
Efron's Bootstrap Percentile CI (Con'd)

$\{z, x\}$

The distribution of $\hat{\beta}_1 - \beta_1$ can be well approximated via the $R = 1000$ different $\hat{\beta}_1^{*(1)} - \hat{\beta}_1, \hat{\beta}_1^{*(2)} - \hat{\beta}_1, \cdots, \hat{\beta}_1^{*(R)} - \hat{\beta}_1$, **no matter the sample size** m **is small or large**.

$\{z^*, x^*\}$

```
hist(Sstar)
```



Efron's Bootstrap Percentile CI (Con'd)

The quantiles are determined by the distribution. Hence, the quantiles can also be approximated via the $R = 1000$ different

$\hat{\beta}_1^{*(1)} - \hat{\beta}_1, \hat{\beta}_1^{*(2)} - \hat{\beta}_1, \cdots, \hat{\beta}_1^{*(R)} - \hat{\beta}_1.$

The $1 - \alpha/2$ quantile of the distribution of $\hat{\beta}_1 - \beta_1$ can be approximated by

1000 values

```
alpha=0.05
quantile(Sstar,1-alpha/2)
```

```
##     97.5%
## 0.6721028
```

97.5% sample quantile of

The $\alpha/2$ quantile of the distribution of $\hat{\beta}_1 - \beta_1$ can be approximated by

```
quantile(Sstar,alpha/2)
```

```
##       2.5%
## -0.615079
```

the quantiles of dist. of $\hat{\beta}_1 - \beta_1$

Efron's Bootstrap Percentile CI (Con'd)

As a consequence

$\alpha/2$ quantile dist. $1-\alpha/2$ quantile

$$\mathrm{P}(-0.6151 \le \hat{\beta}_1 - \beta_1 \le 0.6721) \approx 1-\alpha,$$

which leads to the $(1-\alpha)$ CI for β_1 is

$$[\hat{\beta}_1 - 0.6721, \hat{\beta}_1 + 0.6151], \text{ namely}$$

```
c(hatbeta1-quantile(Sstar,1-alpha/2),
  hatbeta1-quantile(Sstar,alpha/2))
```

```
##          X         X
## 0.686257 1.973439
```

CI of β_1

$P(L \le \hat{\beta}_1 - \beta_1 \le U) = 1-\alpha$

β_1 is unknow

dist. of $\hat{\beta}_1 - \beta_1$

1000 values of $\hat{\beta}_1^{*(1)} - \hat{\beta}_1$

... $\hat{\beta}_1^{*(1000)} - \hat{\beta}_1$

We call it Efron's bootstrap percentile CI for β_1 with confidence $1-\alpha = 95\%$.

$\alpha/2$ quantile of 1000 values

$1-\alpha/2$ quantile of 1000 values

/L34

CI for Mean of Response

Recall that for Poisson log-linear model, we have the mean of response

$$\mu\{Z|X\} = e^{\beta_0+\beta_1 X_1+\cdots+\beta_k X_k}$$

CI of $\mu\{Z|X\}$

given some specific values of $X = (X_1, \cdots, X_k)$.

And also based on MLE we can obtain the estimated mean of response

$$\hat{\mu}\{Z|X\} = e^{\hat{\beta}_0+\hat{\beta}_1 X_1+\cdots+\hat{\beta}_k X_k}$$

However, we never mention the sampling distribution of

$$\hat{\mu}\{Z|X\} - \mu\{Z|X\},$$

even when **the sample size** m is large, since obtaining it requires complicated mathematical induction.

Question: can James utilize the similar idea to construct CI for mean of response based on his 1,000 bootstrap datasets? The answer is yes!

CI for Mean of Response (Con'd)

For the original dataset that Lei Li hands over to James, James fits a Poisson log-linear model and computes $\hat{\mu}\{Z|X\} = \hat{\mu}$ given $X = 0.25$.

```
fit.pois=glm(Z~X,family=poisson(link=log))
Xnew=data.frame(X=0.25)
hatmu=predict(fit.pois,Xnew,type='response')
hatmu
```

```
##        1
## 7.777425
```

CI for Mean of Response (Con'd)

$\hat{\mu} - \mu$ dist.

For $R = 1000$ bootstrap datasets,

$$\{Z_i^{*(1)}, X_i^{*(1)}\}_{i=1}^m, \cdots, \{Z_i^{*(R)}, X_i^{*(R)}\}_{i=1}^m,$$

1000 values

James fits Poisson log-linear models and computes the 1,000 different

$$\hat{\mu}^{*(1)} - \hat{\mu}, \cdots, \hat{\mu}^{*(R)} - \hat{\mu}, \text{respectively.}$$

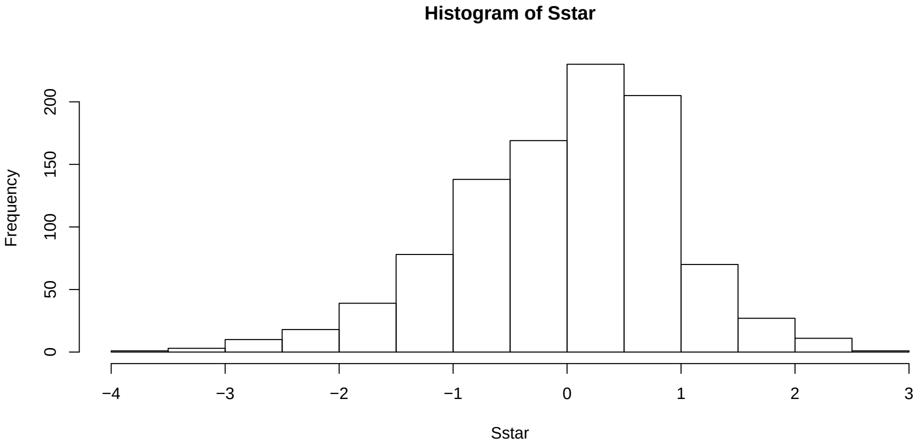
```
#space to store the different mus
hatmustar=rep(0,R)
for (r in 1:R){
 Xstar=Xstardata[r,]
 fit.pois=glm(Zstardata[r,]~Xstar,family=poisson(link=log))
 Xnew=data.frame(Xstar=0.25)
 hatmustar[r]=predict(fit.pois,Xnew,type='response')
}
Sstar=hatmustar-hatmu
```

1000 values

CI for Mean of Response (Con'd)

The distribution of $\hat{\mu} - \mu$ can be well approximated via the $R = 1000$ different $\hat{\mu}^{*(1)} - \hat{\mu}, \hat{\mu}^{*(2)} - \hat{\mu}, \cdots, \hat{\mu}^{*(R)} - \hat{\mu}$, **no matter the sample size** m **is small or large**.

```r
hist(Sstar)
```



CI for Mean of Response (Con'd)

The quantiles are determined by the distribution. Hence, the quantiles can also be approximated via the $R = 1000$ different $\hat{\mu}^{*(1)} - \hat{\mu}, \hat{\mu}^{*(2)} - \hat{\mu}, \cdots, \hat{\mu}^{*(R)} - \hat{\mu}$.

The $1 - \alpha/2$ quantile of the distribution of $\hat{\mu} - \mu$ can be approximated by

1000 - values

```
alpha=0.05
quantile(Sstar,1-alpha/2)
```

```
##     97.5%
## 1.710726
```

$P(-2.2135 \leq \hat{\mu} - \mu \leq 1.7107) = 1-\alpha$

The $\alpha/2$ quantile of the distribution of $\hat{\mu} - \mu$ can be approximated by

1000 - values

```
quantile(Sstar,alpha/2)
```

```
##      2.5%
## -2.213538
```

CI for Mean of Response (Con'd)

As a consequence

$$\mathrm{P}(-2.2135 \leq \hat{\mu} - \mu \leq 1.7107) \approx 1 - \alpha,$$

which leads to Efron's bootstrap percentile CI for mean of response with confidence $1 - \alpha = 95\%$ is

$1-\alpha/2$ quantile $\alpha/2$ quantile

$$[\hat{\mu} - 1.7107, \hat{\mu} + 2.2135], \text{ namely}$$

```
c(hatmu-quantile(Sstar,1-alpha/2),
  hatmu-quantile(Sstar,alpha/2))
```

```
##        1        1
## 6.066699 9.990963
```

CI of μ

estimation of $\hat{\mu}$

$$P(L \leq \hat{\mu} - \mu \leq U) = 1-\alpha$$

dist. of $\hat{\mu} - \mu$

a

$\hat{\mu}^{*(1)} - \hat{\mu}$ ... $\hat{\mu}^{*(1000)} - \hat{\mu}$

$\alpha/2$ quantile of 1000 values

$1-\alpha/2$ quantile of 1000 values

Bootstrap CI Idea

CI of e^{β_1} $\qquad [e^{\hat{\beta}_1} - U, \; e^{\hat{\beta}_1} - L]$

$\downarrow$

estimation $e^{\hat{\beta}_1}$

$\downarrow$

$$P(L \le e^{\hat{\beta}_1} - e^{\beta_1} \le U) = 1-\alpha$$

$\downarrow$

dist of $e^{\hat{\beta}_1} - e^{\beta_1}$ (m is small)

$\downarrow$

$e^{\hat{\beta}_1^{*(1)}} - e^{\hat{\beta}_1}, \; \ldots \; e^{\hat{\beta}_1^{*(1000)}} - e^{\hat{\beta}_1}$

$\downarrow$

$\alpha/2$ quantile of 1000 values

$1-\alpha/2$ quantile of 1000 values

Benefits of Bootstrap II

Obtain the CIs for regression paramters and the mean of response.

No need of knowing the formulas for sampling distributions and standard errors (the sampling distribution and the appoximate standard error in GLM requires a formula from complicated mathematical inductions).

The sample size can be small or large. It doesn't matter.

Sample Size	Confidence Interval (CI)
m is small	Bootstrap CI is better.
m is large	Both bootstrap CI and the classical CI can be used.

But $R = 1000$ **repeated bootstrap samples** are required, and hence bootstrap is computationally intensive.

Final note: the bootstrap method is not limited to small or large sample size, is robust to normal or non-normal assumption, and can be applied in all the MLR and GLM models for estimating the sampling distribution, computing standard errors and obtaining CIs. The bootstrap method has much more applications which will not be introduced in this course.