统计咨询与实践第一次作业

张申铎 2176112379, 祖劭康 2173412124

目录

1 问题 1

非线性基本模型为

$$Y = f(X, \theta) + \varepsilon \quad \varepsilon \sim N(0, \sigma^2)\text{其中}\varepsilon\text{与}X\text{独立} \quad \theta \in \mathbb{R}^d \tag{1}$$

考虑非线性最小二乘回归，进行回归的非线性函数在本次实验中设定为

$$f(x_1, x_2, x_3) = 100sinx_1 + 200cosx_2 + \mathrm{e}x_3 \tag{2}$$

我们假设 x_1, x_2, x_3 都来自 $U(-5, 5)$，并由此生成 1000 个点上的数据，再采用非线性最小二乘模型去估计函数中每一项的参数。同时我们也进行了 1000 次重复实验，采用平均误差平方和作为我们的性能度量标准，并对参数估计值减真实值的渐近正态性进行了展示，做出了概率密度函数图像，及正态 QQ 图。

1.1 证明 $\hat{\theta}$ 的渐近正态性

$$l(\theta)=\frac{1}{n}\Sigma_{i=1}^{n}\{y_i-f(x_i,\theta)\}$$
$$\frac{\partial f}{\partial\theta}(x_i,\theta)=(\frac{\partial f}{\partial\theta_1}(x_i,\theta),\frac{\partial f}{\partial\theta_2,}(x_i,\theta),\cdots,\frac{\partial f}{\partial\theta_d}(x_i,\theta))^T$$
$$l^{'}(\theta)=-\frac{2}{n}\{y_i-f(x_i,\theta)\}\frac{\partial f}{\partial\theta}(x_i,\theta)$$
$$l^{''}(\theta)=\frac{2}{n}\Sigma_{i=1}^{n}\frac{\partial f}{\partial\theta}(x_i,\theta)\frac{\partial f}{\partial\theta}(x_i,\theta)^T-\frac{2}{n}\Sigma_{i=1}^{n}\{y_i-f(x_i,\theta)\}\frac{\partial^2 f}{\partial\theta^2}(x_i,\theta)$$
$$\widetilde{l^{''}}(\theta)\equiv\frac{1}{2}l(\theta)\quad\tilde{l}^{'}(\theta)\equiv-\frac{1}{2}l^{'}(\theta)$$
$$\hat{\theta}-\theta=[\widetilde{l^{''}}(\theta)]^{-1}\tilde{l}^{'}(\theta)+O(||\hat{\theta}-\theta||_2^2)$$
$$h_1(\theta)\equiv\frac{1}{n}\Sigma_{i=1}^{n}\{y_i-f(x_i,\theta)\}\frac{\partial^2 f}{\partial\theta^2}(x_i,\theta)$$
$$h_2(\theta)\equiv\frac{1}{n}\Sigma_{i=1}^{n}\frac{\partial f}{\partial\theta}(x_i,\theta)\frac{\partial f}{\partial\theta}(x_i,\theta)^T$$

于是$\widetilde{l^{''}}(\theta)=h_2(\theta)-h_1(\theta)$

$$E(\varepsilon|X)=0\Rightarrow E(\varepsilon|\frac{\partial^2 f}{\partial\theta^2}(X,\theta))=0\Rightarrow E(h_1(\theta))=0$$
$$E(h_2(\theta))=E[\frac{\partial f}{\partial\theta}(X,\theta)\frac{\partial f}{\partial\theta}(X,\theta)^T]$$
$$\widetilde{l^{''}}(\theta)\xrightarrow{P}E[\frac{\partial f}{\partial\theta}(X,\theta)\frac{\partial f}{\partial\theta}(X,\theta)^T]\equiv A$$
$$E(\tilde{l}^{'}(\theta))=E(\varepsilon\frac{\partial f}{\partial\theta}(X,\theta))=0\quad Cov(\sqrt{n}\tilde{l}^{'}(\theta))=Cov(\varepsilon\frac{\partial f}{\partial\theta}(X,\theta))=\sigma^2E[\frac{\partial f}{\partial\theta}(X,\theta)\frac{\partial f}{\partial\theta}(X,\theta)^T]=\sigma^2A$$

$\Longrightarrow\tilde{l}^{'}(\theta)\xrightarrow{d}N(0,\sigma^2A)$

综上$\sqrt{\hat{\theta}-\theta}\xrightarrow{d}N(0,\sigma^2A^{-1})$

1.2 准备工作数据

首先清理环境并安装所需软件包。

```
library(ggplot2)
```

设定我们重复实验的次数为 1000 次，

```
m <- 1000
beta <- c(100,200,exp(1))
```

1.3 小样本实验

设定样本数为 100，同时初始化程序结果与绘图数据。

```
mse <- 0
n <- 100
result <- list("sampleSize"=n, "number of experiments" = m,"mean of total MSE " = mse)
betaes <- c()
```

1.3.1 实验过程

我们首先随机生成 X，再根据 X 和 $f(X)$ 生成对应的观测数据，在使用 NLM 函数对参数 β 进行估计，记录均方误差值与估计得的参数值。重复实验 1000 次。

```
for(i in 1:m){
    set.seed(i)
    df <- data.frame(x1 <- runif(n,-5,5), x2 <- runif(n,-5,5), x3 <- runif(n,-5,5))
    y <- 100*sin(df[,1]) + 200*cos(df[,2]) + exp(1)*df[,3]^3 + rnorm(n,0,1)
    model <- nls(y ~ b1*sin(x1) + b2*cos(x2) + b3*x3^3,start = list(b1=1, b2 = 2, b3 = 3))
    z <- coef(model)-beta
    betaes <- cbind(betaes,z)
    mse <- mse + t(z) %*% z
}
mse <- mse/m
result$'Total MSE' <- mse
data1 <- data.frame(b1= t(betaes)[,1])
data2 <- data.frame(b2= t(betaes)[,2])
data3 <- data.frame(b3= t(betaes)[,3])
```

1.3.2 结果展示

对于我们的模型，我们选取了我们 1000 次重复实验中的一次进行展示。同时我们也给出了每一个模型参数的估计值的 1000 次实验的平均误差平方和，参数估计值的每一维度的分布，以及其正态 QQ 图。

1.3.2.1 模型总结

```
summary(model)
```

```
##
## Formula: y ~ b1 * sin(x1) + b2 * cos(x2) + b3 * x3^3
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## b1 1.001e+02  1.306e-01   766.3   <2e-16 ***
## b2 2.002e+02  1.441e-01  1389.3   <2e-16 ***
## b3 2.718e+00  2.354e-03  1154.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9365 on 97 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 8.519e-06
```

1.3.2.1.1 1000 次实验后参数估计的平均均方误差值

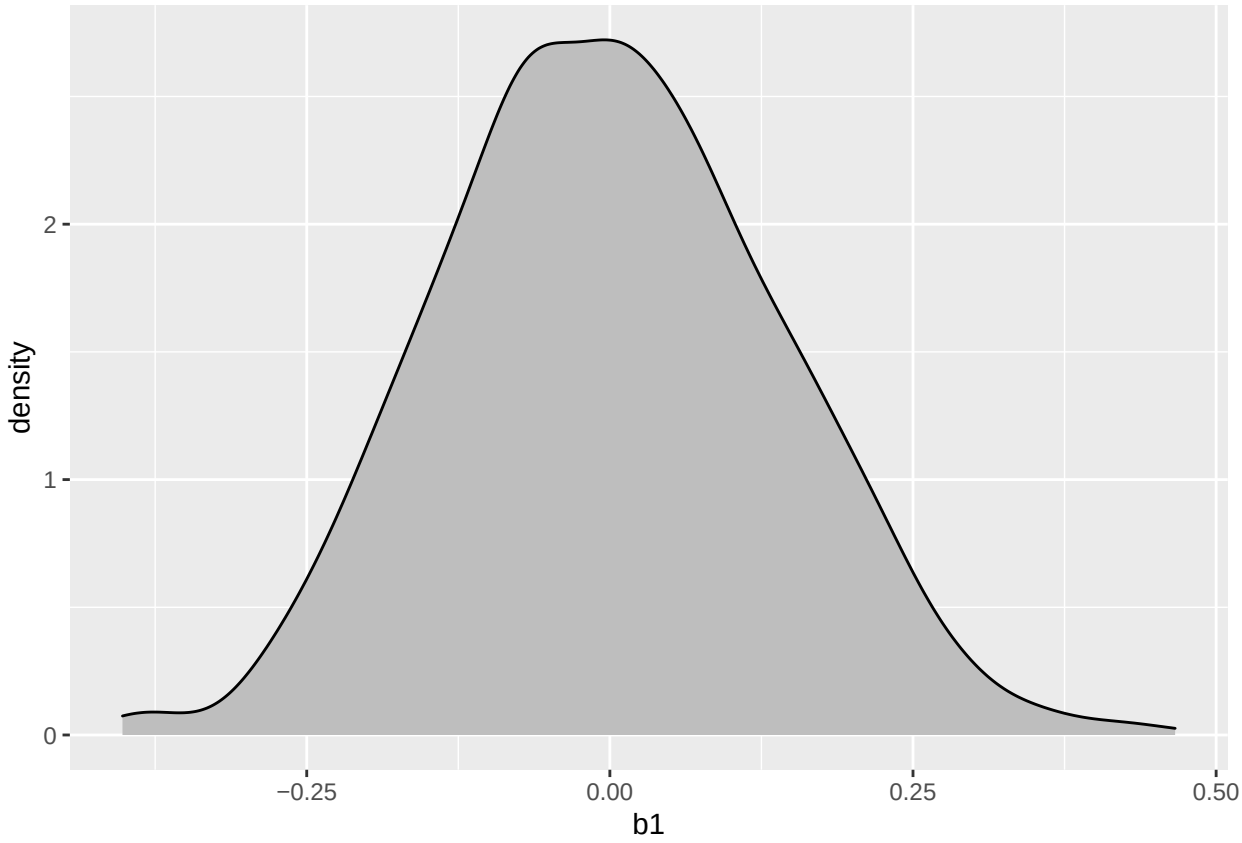
```
result
```

```
## $sampleSize
## [1] 100
##
## $`number of experiments`
## [1] 1000
##
```

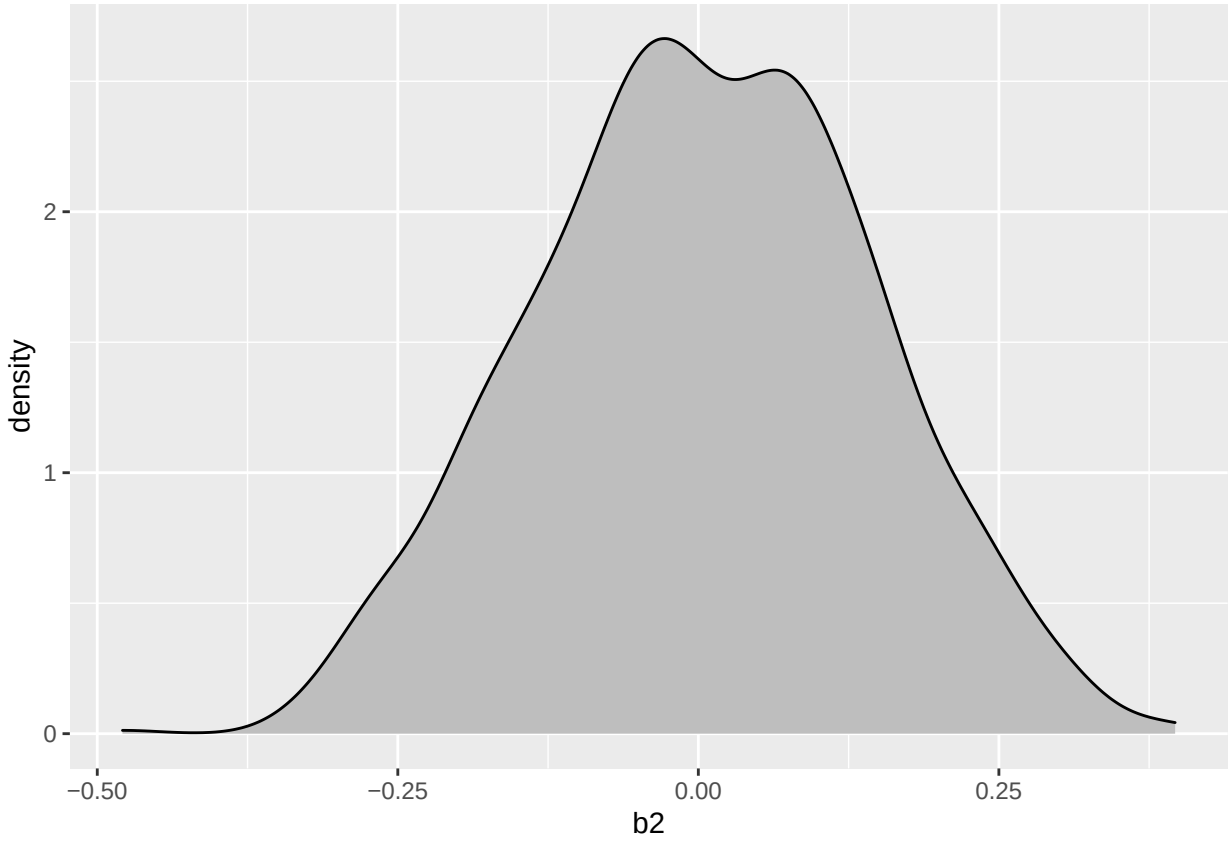
```
## $`mean of total MSE `
## [1] 0
##
## $`Total MSE`
##             [,1]
## [1,] 0.03856038
```

1.3.2.1.2 1000 次实验后每一维度参数估计的分布

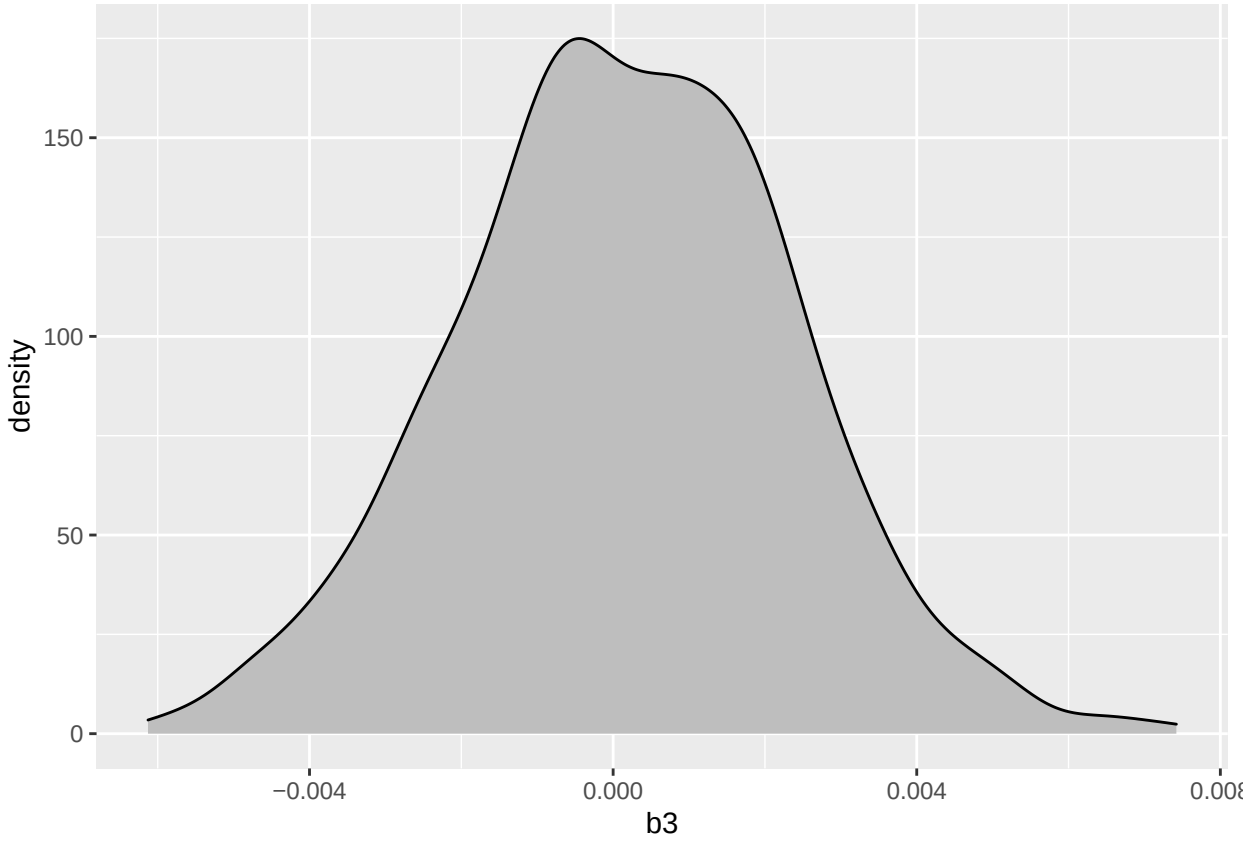
```
p1 <- ggplot(data1,aes(x = b1))
p1 + geom_density(color = 'black', fill = 'gray')
```




```
p2 <- ggplot(data2,aes(x = b2))
p2 + geom_density(color = 'black', fill = 'gray')
```

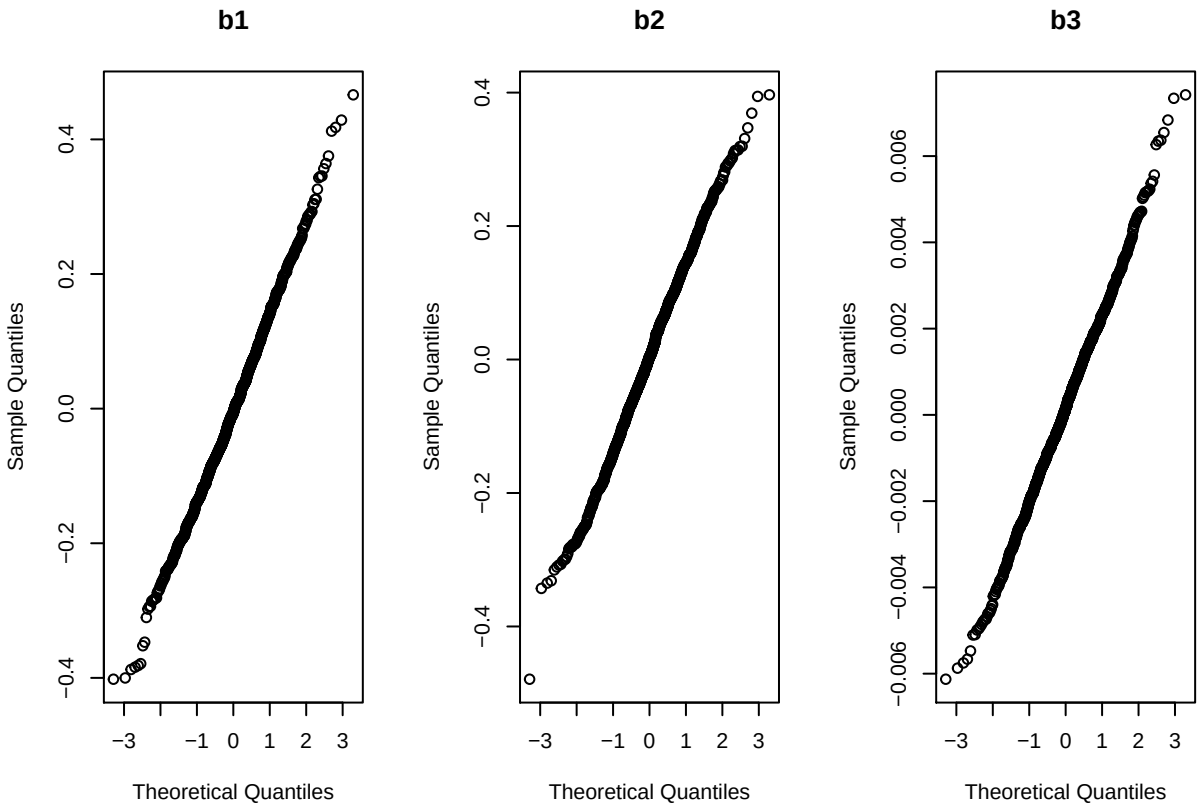



```
p3 <- ggplot(data3,aes(x = b3))
p3 + geom_density(color = 'black', fill = 'gray')
```



1.3.2.1.3 1000 次实验后模型每一维度参数估计的正态 QQ 图

```
par(mfrow = c(1,3))
qqnorm(t(betaes)[,1], main = "b1")
qqnorm(t(betaes)[,2], main = "b2")
qqnorm(t(betaes)[,3], main = "b3")
```



1.4 大样本实验

设定样本数为 10000，同时初始化程序结果与绘图数据。

```
mse <- 0
n <- 10000
result <- list("sampleSize"=n, "number of experiments" = m,"Total MSE " = mse)
betaes <- c()
```

1.4.1 实验过程

过程同小样本实验，我们首先随机生成 X，再根据 X 和 $f(X)$ 生成对应的观测数据，在使用 NLM 函数对参数 β 进行估计，记录均方误差值与估计得的参数值。重复实验 1000 次。

```
for(i in 1:m){
    set.seed(i)
    df <- data.frame(x1 <- runif(n,-5,5), x2 <- runif(n,-5,5), x3 <- runif(n,-5,5))
```

```
    y <- 100*sin(df[,1]) + 200*cos(df[,2]) + exp(1)*df[,3]^3 + rnorm(n,0,1)
    model <- nls(y ~ b1*sin(x1) + b2*cos(x2) + b3*x3^3,start = list(b1=1, b2 = 2, b3 = 3))
    z <- coef(model)-beta
    betaes <- cbind(betaes,z)
    mse <- mse + t(z) %*% z
}
mse <- mse/m
result$'Total MSE' <- mse
result
```

```
## $sampleSize
## [1] 10000
##
## $`number of experiments`
## [1] 1000
##
## $`Total MSE `
## [1] 0
##
## $`Total MSE`
##               [,1]
## [1,] 0.0003952815
```

```
data1 <- data.frame(b1= t(betaes)[,1])
data2 <- data.frame(b2= t(betaes)[,2])
data3 <- data.frame(b3= t(betaes)[,3])
```

1.4.2 结果展示

同小样本情形。

1.4.2.1 模型总结

```
summary(model)
```

```
##
## Formula: y ~ b1 * sin(x1) + b2 * cos(x2) + b3 * x3^3
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## b1 1.000e+02  1.360e-02    7352   <2e-16 ***
## b2 2.000e+02  1.429e-02   13990   <2e-16 ***
## b3 2.718e+00  2.096e-04   12969   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.987 on 9997 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 1.715e-06
```

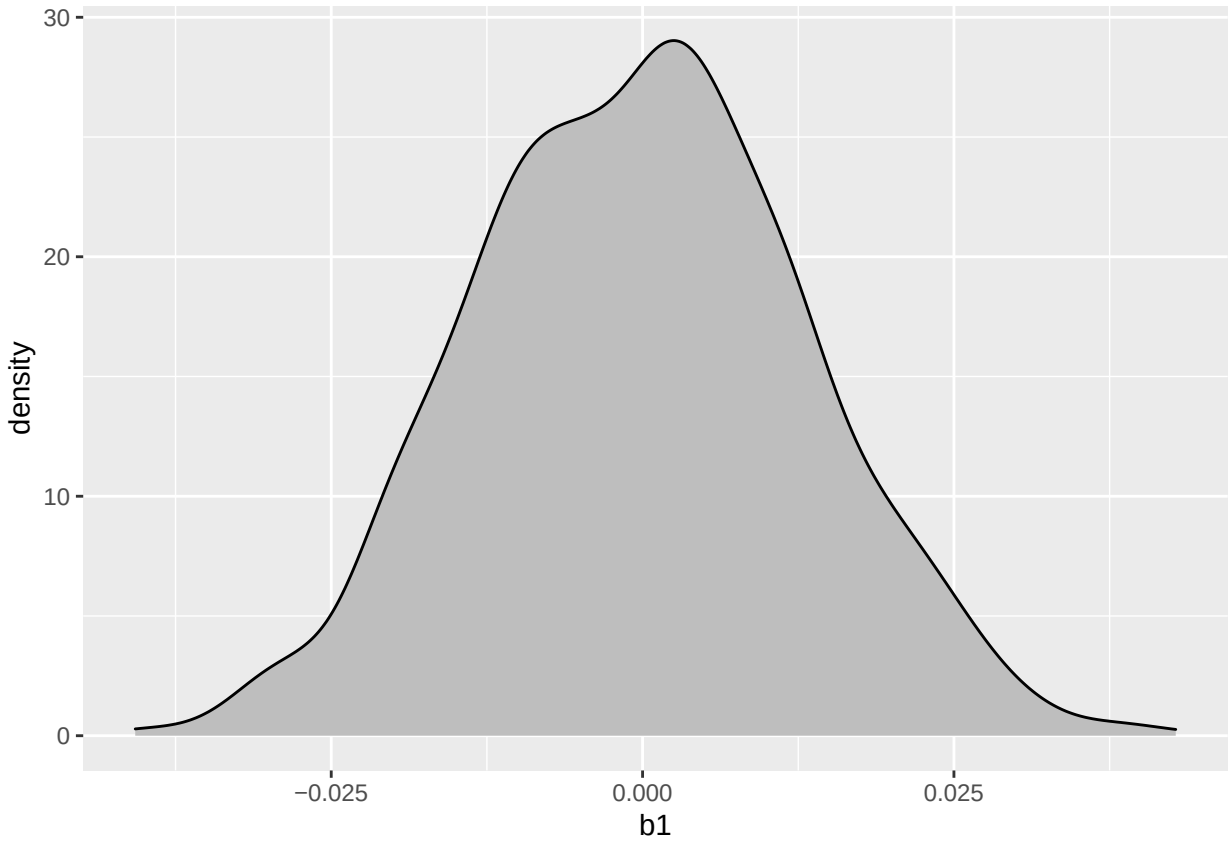
1.4.2.1.1 1000 次实验后参数估计的平均均方误差值

```
result
```

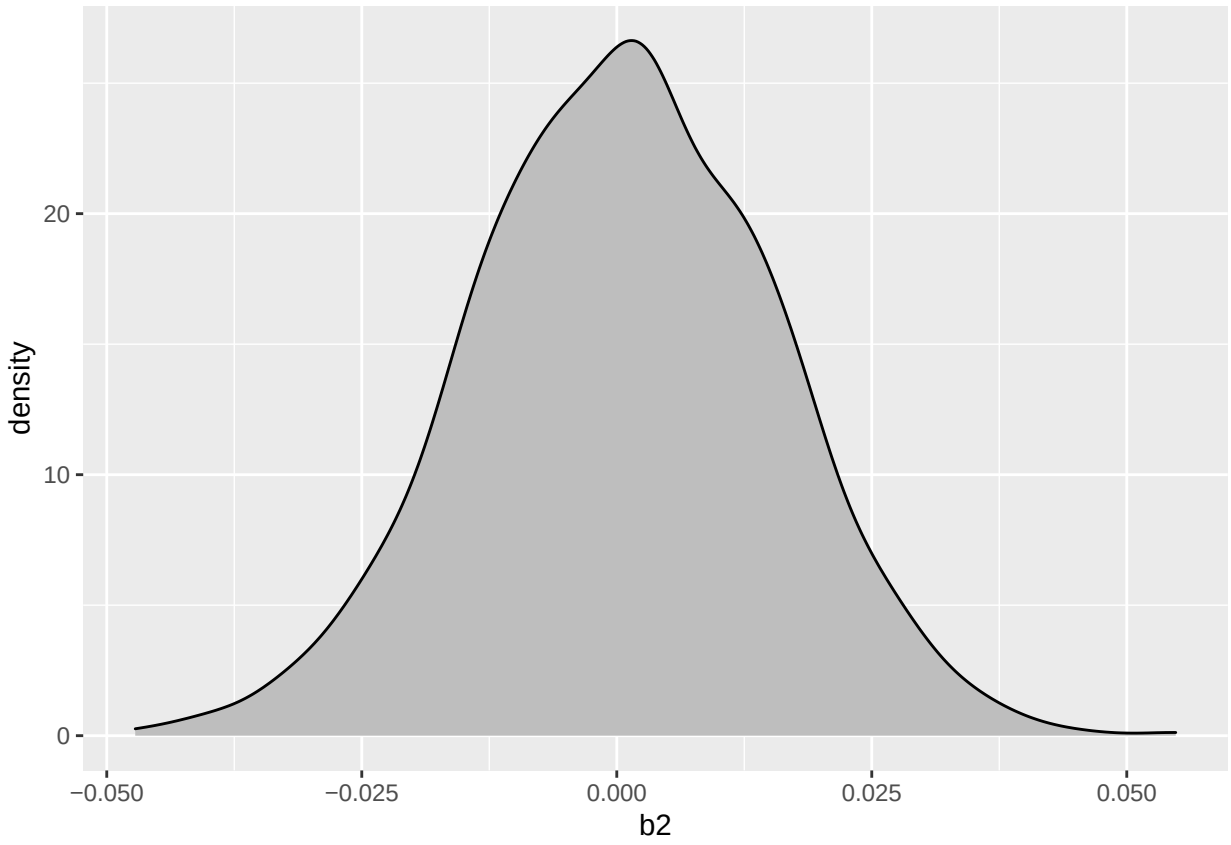
```
## $sampleSize
## [1] 10000
##
## $`number of experiments`
## [1] 1000
##
## $`Total MSE `
## [1] 0
##
## $`Total MSE`
##              [,1]
## [1,] 0.0003952815
```

1.4.2.1.2 1000 次实验后每一维度参数估计的分布

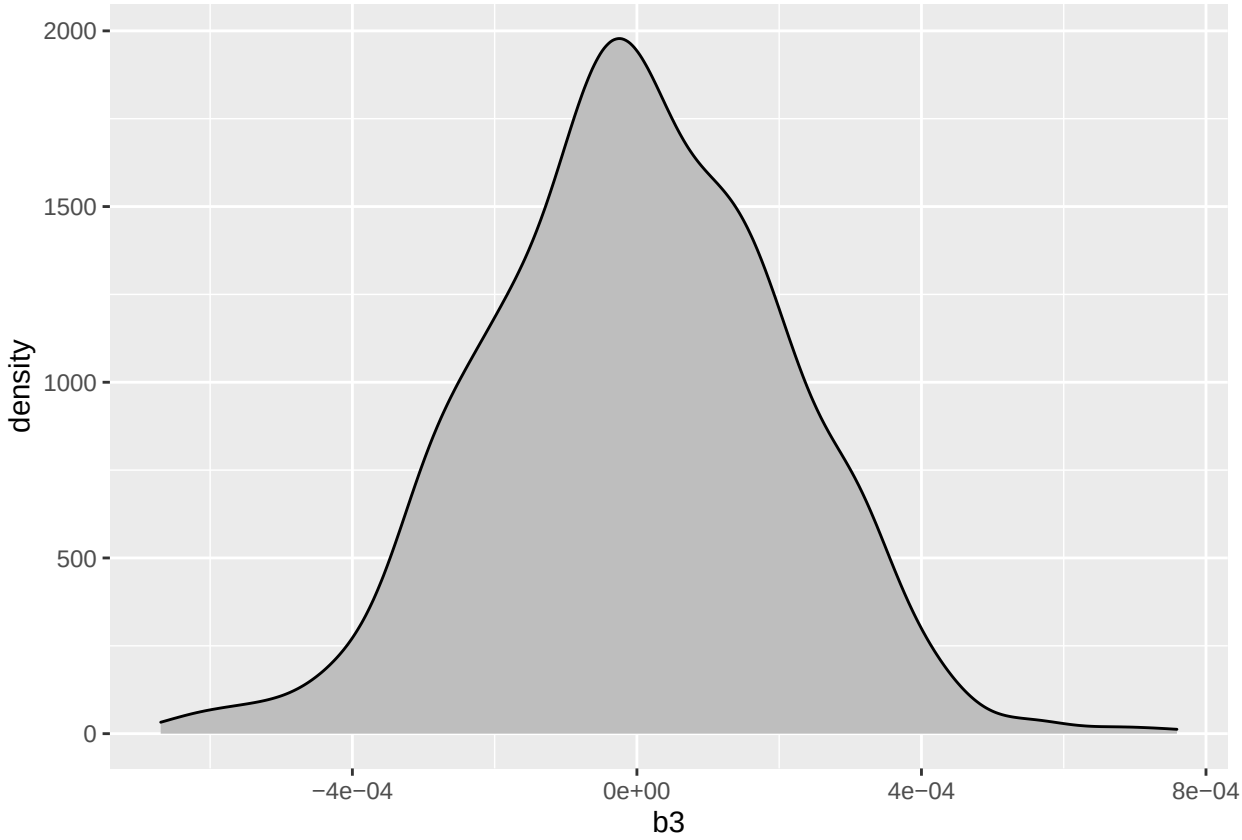
```
p1 <- ggplot(data1,aes(x = b1))
p1 + geom_density(color = 'black', fill = 'gray')
```




```
p2 <- ggplot(data2,aes(x = b2))
p2 + geom_density(color = 'black', fill = 'gray')
```

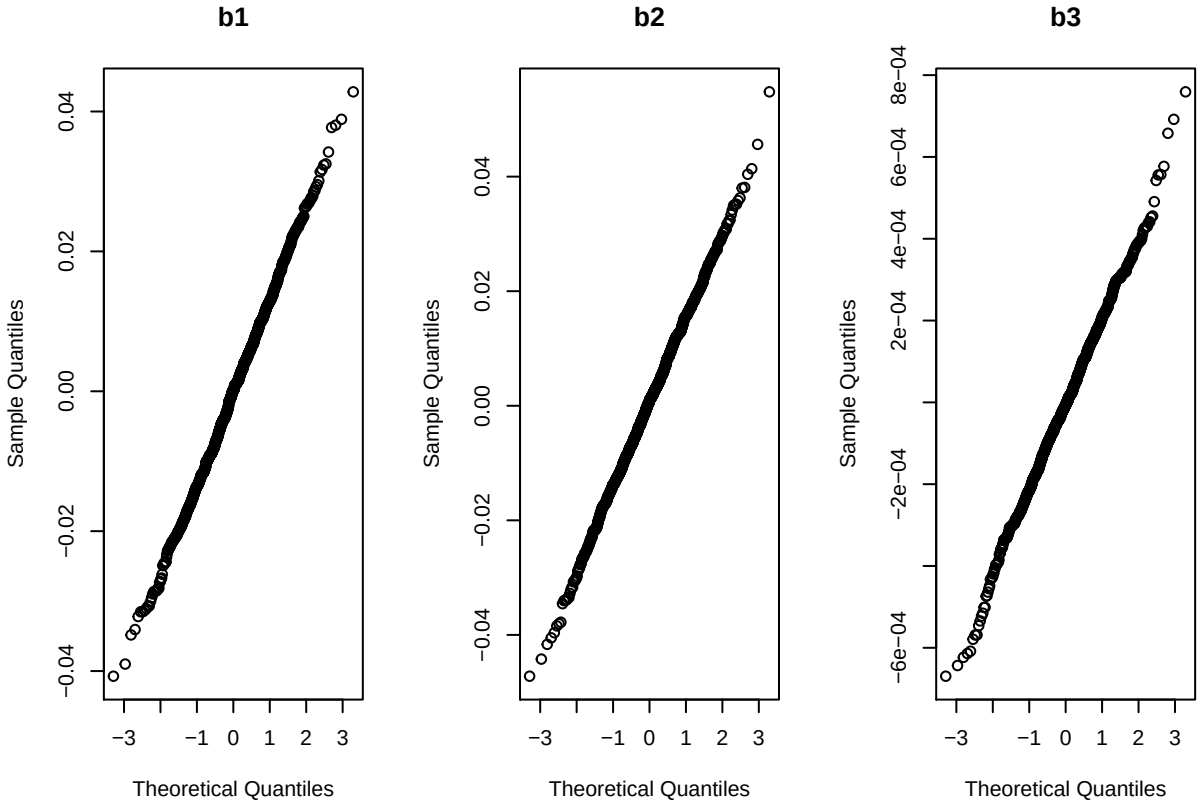



```
p3 <- ggplot(data3,aes(x = b3))
p3 + geom_density(color = 'black', fill = 'gray')
```



1.4.2.1.3 1000 次实验后模型每一维度参数估计的正态 QQ 图

```
par(mfrow = c(1,3))
qqnorm(t(betaes)[,1], main = "b1")
qqnorm(t(betaes)[,2], main = "b2")
qqnorm(t(betaes)[,3], main = "b3")
```



2 问题 2

考虑三种广义线性模型，首先讨论高斯分布。

我们考虑的是在自然链接函数下的广义线性模型的正态情形。数据生成公式如下，

$$Y|X = x \sim N(\mu, 1), \quad \mu = X^T\beta \tag{3}$$

对于对数回归模型（泊松分布），我们考虑的是在自然链接函数下的广义线性模型。数据生成公式如下，

$$Y|X = x \sim P(\lambda), \quad \lambda = e^{-\theta} = e^{-X^T\beta} \tag{4}$$

Logestic 回归模型（二项分布模型）

$$Y|X = x \sim b(p), \quad p = \frac{e^{\theta}}{1+e^{\theta}}, \quad \theta = X^T\beta \tag{5}$$

我们对三种模型选取三维的 $\beta = (3,2,1)^T$ 生成数据，然后采取相应的广义线性模型去估计参数。采用平均误差平方和作为我们的性能度量标准。同时我们也进行了多次重复实验，完成了三个参数在不同模型下的多次估计，给出了三个不同模型的每个参数估计值的分布图，以及其正态 QQ 图。我们同时也对比了不同样本数量下其估计表现的差异以及估计值的分布。

2.1 准备工作数据

首先清理环境并安装所需软件包。

```
rm(list = ls())
library(ggplot2)
```

设定我们重复实验的次数为 100 次，

```
n=100 #number of experiments
beta <- c(3,2,1)
```

2.2 小样本实验

设定样本数为 100，同时初始化程序结果与绘图数据。

```
sampleSize = 100
result <- list("sampleSize"=sampleSize, "number of experiments" = n,
               "Total MSE for Gaussian" = 0,"Total MSE for Poisson"=0,"Total MSE for Logestic"=0)
PlotData <- data.frame(beta=numeric(),index=numeric(),type=numeric())
```

2.2.1 实验过程

我们首先随机生成 X，然后根据生成的 X 来根据三个不同的模型生成所对应的观测值数据。再使用 GLM 函数用生成数据对模型参数进行估计，记录均方误差值与估计得的参数值。重复实验 100 次。

```
for (i in 1:n) {
  #
  set.seed(i)
  #
  df <- data.frame(x1 = runif(sampleSize,-2,0),
                   x2 = runif(sampleSize,0,2),
```

```
                    x3 = runif(sampleSize,-1,1))
  df1 <- df
  df1$y <- beta[1]*df1$x1+beta[2]*df1$x2+beta[3]*df1$x3+rnorm(sampleSize,0,1)
  ex1<-glm(formula=y~0+x1+x2+x3,data = df1,family = gaussian())
  result$`Total MSE for Gaussian` <- (result$`Total MSE for Gaussian` +
                                        sum((beta-ex1$coefficients)^2)/3)
  for (j in 1:length(ex1$coefficients)) {
   PlotData[nrow(PlotData)+1,]<- list(ex1$coefficients[j],j,"Gaussian")
  }
  #
  df2 <- df
  df2$y <- rpois(sampleSize,exp(beta[1]*df2$x1+beta[2]*df2$x2+beta[3]*df2$x3))
  ex2<-glm(formula=y~0+x1+x2+x3,data = df2,family = poisson())
  result$`Total MSE for Poisson` <- (result$`Total MSE for Poisson` +
                                       sum((beta-ex1$coefficients)^2)/3)
  for (j in 1:length(ex2$coefficients)) {
    PlotData[nrow(PlotData)+1,]<- list(ex2$coefficients[j],j,"Poisson")
  }
  #
  df3<-df
  df3$y <- rbinom(sampleSize,10,1/(1+exp(-(beta[1]*df3$x1+beta[2]*df3$x2+beta[3]*df3$x3))))/10
  ex3 <- glm(formula=y~0+x1+x2+x3,data = df3, family = binomial())
  result$`Total MSE for Logestic` <- (result$`Total MSE for Logestic` +
                                        sum((beta-ex3$coefficients)^2)/3)
  for (j in 1:length(ex3$coefficients)) {
    PlotData[nrow(PlotData)+1,]<- list(ex3$coefficients[j],j,"Logestic")
  }
}
```

2.2.2 结果展示

对于我们的模型总结，我们选取了我们 100 次重复实验中的一次进行展示。同时我们也给出了每一个模型参数的估计值的 100 次实验中每一次均方误差的综合，以及参数估计值的分布，以及其正态 QQ 图。

2.2.2.1 模型总结

```
summary(ex1)#Gaussian
```

```
##
## Call:
## glm(formula = y ~ 0 + x1 + x2 + x3, family = gaussian(), data = df1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.98873  -0.54011  -0.06484   0.52605   2.62997
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## x1   3.0034     0.1258  23.868  < 2e-16 ***
## x2   2.0279     0.1176  17.250  < 2e-16 ***
## x3   0.8703     0.1833   4.748  7.1e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9913992)
##
##     Null deviance: 666.827  on 100  degrees of freedom
## Residual deviance:  96.166  on  97  degrees of freedom
## AIC: 287.88
##
## Number of Fisher Scoring iterations: 2
```

```
summary(ex2)#Poisson
```

```
##
## Call:
## glm(formula = y ~ 0 + x1 + x2 + x3, family = poisson(), data = df2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5231  -0.5542  -0.2623   0.2487   1.7948
```

```
##
## Coefficients:
##    Estimate Std. Error z value Pr(>|z|)
## x1  2.84882    0.19876  14.333   <2e-16 ***
## x2  2.00889    0.05391  37.266   <2e-16 ***
## x3  0.99864    0.10387   9.614   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1078.305  on 100  degrees of freedom
## Residual deviance:   69.307  on  97  degrees of freedom
## AIC: 231.87
##
## Number of Fisher Scoring iterations: 5
```

```
summary(ex3)#Logestic
```

```
##
## Call:
## glm(formula = y ~ 0 + x1 + x2 + x3, family = binomial(), data = df3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8597  -0.2516  -0.1049   0.2666   0.7809
##
## Coefficients:
##    Estimate Std. Error z value Pr(>|z|)
## x1   3.2472     0.6252   5.194 2.06e-07 ***
## x2   2.1393     0.4585   4.665 3.08e-06 ***
## x3   1.3394     0.6119   2.189   0.0286 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 76.027  on 100  degrees of freedom
## Residual deviance: 11.330  on  97  degrees of freedom
## AIC: 58.705
##
## Number of Fisher Scoring iterations: 6
```

2.2.2.1.1 100 次实验后参数估计的均方误差值总和

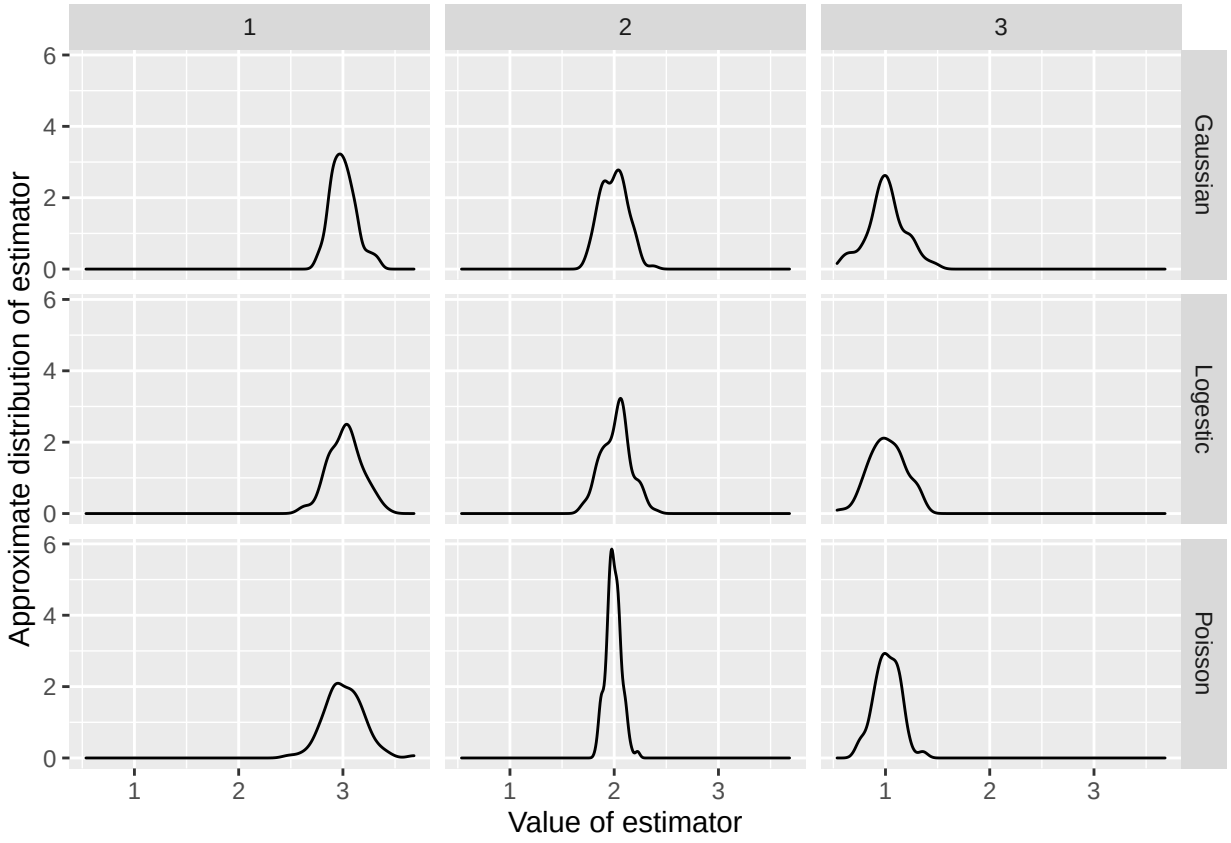
```
result
```

```
## $sampleSize
## [1] 100
##
## $`number of experiments`
## [1] 100
##
## $`Total MSE for Gaussian`
## [1] 2.157786
##
## $`Total MSE for Poisson`
## [1] 2.157786
##
## $`Total MSE for Logestic`
## [1] 2.486469
```

```
#The actual error in each experienments shall be divided by number of experiments which is 100 her
```

2.2.2.1.2 100 次实验后各个模型参数估计的分布

```
p11<-ggplot(data = PlotData,aes(x=PlotData[,1]))+
  geom_density()+
  facet_grid(PlotData[,3]~PlotData[,2])+
  xlab("Value of estimator")+
  ylab("Approximate distribution of estimator")
p11
```

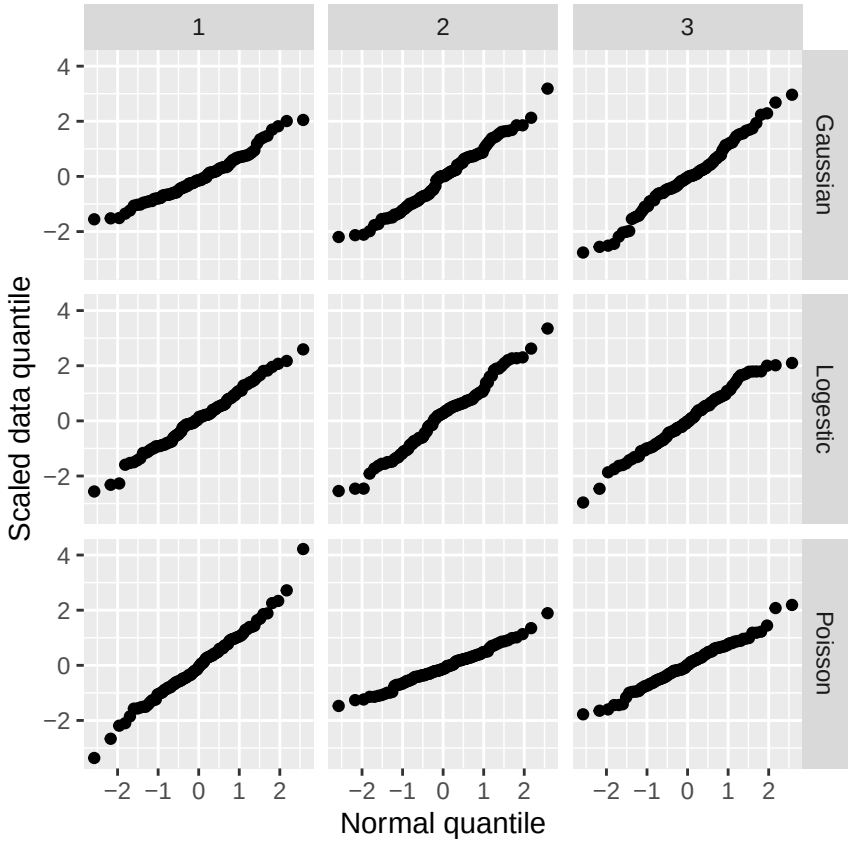


我们可以看到我们的参数近似地在各真实值附近呈单峰分布，其较均值偏离明显。该曲线为 ggplot 自带 density 函数根据真值插值而来。

2.2.2.1.3 100 次实验后各个模型参数估计的正态 QQ 图

为了画 QQ 图，我们将我们的估计值标准化。

```
PlotData[PlotData$index==1,1]<-scale(PlotData[PlotData$index==1,1])
PlotData[PlotData$index==2,1]<-scale(PlotData[PlotData$index==2,1])
PlotData[PlotData$index==3,1]<-scale(PlotData[PlotData$index==3,1])
p12<-ggplot(data = PlotData,aes(sample=PlotData[,1]))+
  geom_qq(distribution = stats::qnorm)+
  facet_grid(PlotData[,3]~PlotData[,2])+
  xlab("Normal quantile")+ylab("Scaled data quantile")+
  theme(aspect.ratio = 1)
p12
```



从正态 QQ 图来看，我们认为此时估计出来的参数具有渐进正态性。

2.3 大样本实验

设定样本数为 10000，同时初始化程序结果与绘图数据。

```
sampleSize = 10000
result <- list("sampleSize"=sampleSize, "number of experiments" = n,"Total MSE for Gaussian" = 0,
               "Total MSE for Poisson"=0,"Total MSE for Logestic"=0)
PlotData <- data.frame(beta=numeric(),index=numeric(),type=numeric())
```

2.3.1 实验过程

过程同小样本实验，我们首先随机生成 X，然后根据生成的 X 来根据三个不同的模型生成所对应的观测值数据。再使用 GLM 函数用生成数据对模型参数进行估计，记录均方误差值与估计得的参数值。重复实验 100 次。

```
for (i in 1:n) {
  #
  set.seed(i)
  #
  df <- data.frame(x1 = runif(sampleSize,-2,0),
                   x2 = runif(sampleSize,0,2),
                   x3 = runif(sampleSize,-1,1))
  df1 <- df
  df1$y <- beta[1]*df1$x1+beta[2]*df1$x2+beta[3]*df1$x3+rnorm(sampleSize,0,1)
  ex1<-glm(formula=y~0+x1+x2+x3,data = df1,family = gaussian())
  result$`Total MSE for Gaussian` <- result$`Total MSE for Gaussian` +
    sum((beta-ex1$coefficients)^2)/3
  for (j in 1:length(ex1$coefficients)) {
    PlotData[nrow(PlotData)+1,]<- list(ex1$coefficients[j],j,"Gaussian")
  }
  #
  df2 <- df
  df2$y <- rpois(sampleSize,exp(beta[1]*df2$x1+beta[2]*df2$x2+beta[3]*df2$x3))
  ex2<-glm(formula=y~0+x1+x2+x3,data = df2,family = poisson())
  result$`Total MSE for Poisson` <- result$`Total MSE for Poisson` +
    sum((beta-ex1$coefficients)^2)/3
  for (j in 1:length(ex2$coefficients)) {
    PlotData[nrow(PlotData)+1,]<- list(ex2$coefficients[j],j,"Poisson")
  }
  #
  df3<-df
  df3$y <- rbinom(sampleSize,10,1/(1+exp(-(beta[1]*df3$x1+beta[2]*df3$x2+beta[3]*df3$x3))))/10
  ex3 <- glm(formula=y~0+x1+x2+x3,data = df3, family = binomial())
  result$`Total MSE for Logestic` <- result$`Total MSE for Logestic` +
    sum((beta-ex3$coefficients)^2)/3
  for (j in 1:length(ex3$coefficients)) {
    PlotData[nrow(PlotData)+1,]<- list(ex3$coefficients[j],j,"Logestic")
  }
}
```

2.3.2 结果展示

同小样本情形。

2.3.2.1 模型总结

```
summary(ex1)#Gaussian
```

```
##
## Call:
## glm(formula = y ~ 0 + x1 + x2 + x3, family = gaussian(), data = df1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4838  -0.6572   0.0054   0.6699   3.6177
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## x1  2.98141    0.01293  230.53   <2e-16 ***
## x2  1.97028    0.01298  151.84   <2e-16 ***
## x3  0.98062    0.01713   57.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9791221)
##
##     Null deviance: 65885.2  on 10000  degrees of freedom
## Residual deviance:  9788.3  on  9997  degrees of freedom
## AIC: 28173
##
## Number of Fisher Scoring iterations: 2
```

```
summary(ex2)#Poisson
```

```
##
## Call:
```

```
## glm(formula = y ~ 0 + x1 + x2 + x3, family = poisson(), data = df2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6394  -0.6633  -0.3043   0.1539   4.2518
##
## Coefficients:
##    Estimate Std. Error z value Pr(>|z|)
## x1 2.968140   0.018883  157.19   <2e-16 ***
## x2 1.995571   0.005833  342.10   <2e-16 ***
## x3 1.007053   0.011759   85.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 98284.6  on 10000  degrees of freedom
## Residual deviance:  7681.4  on  9997  degrees of freedom
## AIC: 20710
##
## Number of Fisher Scoring iterations: 5
```

```
summary(ex3)#Logestic
```

```
##
## Call:
## glm(formula = y ~ 0 + x1 + x2 + x3, family = binomial(), data = df3)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.16806  -0.24817  -0.06537   0.21089   1.19048
##
## Coefficients:
##    Estimate Std. Error z value Pr(>|z|)
## x1  3.03742    0.06003   50.60   <2e-16 ***
## x2  2.03298    0.04767   42.65   <2e-16 ***
## x3  1.01378    0.05061   20.03   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6797.2  on 10000  degrees of freedom
## Residual deviance: 1005.3  on  9997  degrees of freedom
## AIC: 5102.7
##
## Number of Fisher Scoring iterations: 6
```

2.3.2.1.1 100 次实验后参数估计的均方误差值总和

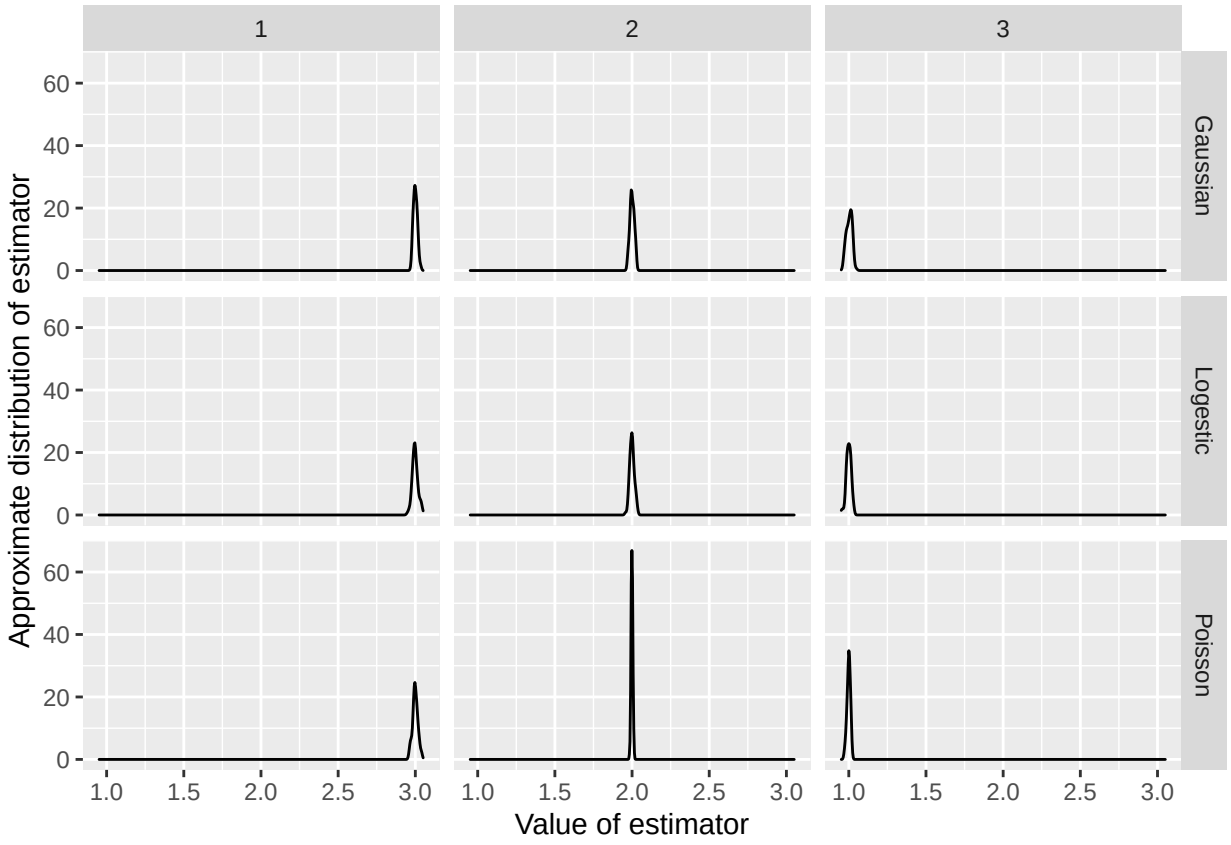
```
result#The actual error in each experienments shall be divided by number of experiments which is 1
```

```
## $sampleSize
## [1] 10000
##
## $`number of experiments`
## [1] 100
##
## $`Total MSE for Gaussian`
## [1] 0.02337456
##
## $`Total MSE for Poisson`
## [1] 0.02337456
##
## $`Total MSE for Logestic`
## [1] 0.02763664
```

2.3.2.1.2 100 次实验后各个模型参数估计的分布

```
p21<-ggplot(data = PlotData,aes(x=PlotData[,1]))+
  geom_density()+
  facet_grid(PlotData[,3]~PlotData[,2])+
```

```
  xlab("Value of estimator")+ylab("Approximate distribution of estimator")
p21
```

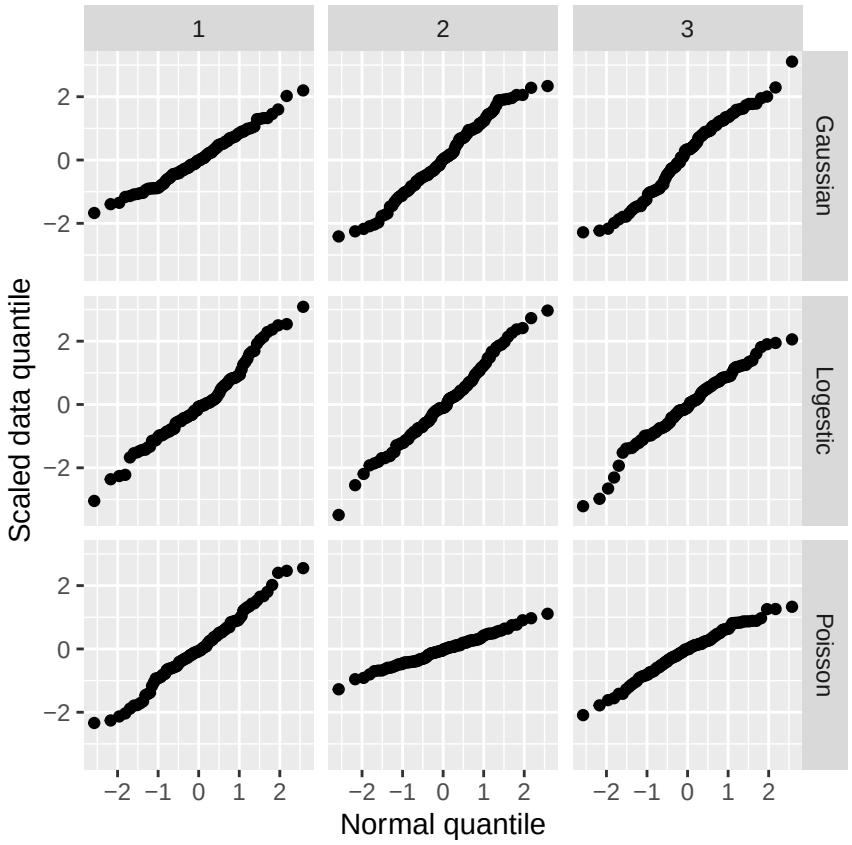


我们可以看到，10000 个大样本条件下，与小样本相比，估计值依旧呈单峰分布，集中在真值附近，同时各样本较均值的偏差非常之小，估计更为稳定。

2.3.2.1.3 100 次实验后各个模型参数估计的正态 QQ 图 ma 为了画 QQ 图，我们将我们的估计值标准化。

```
PlotData[PlotData$index==1,1]<-scale(PlotData[PlotData$index==1,1])
PlotData[PlotData$index==2,1]<-scale(PlotData[PlotData$index==2,1])
PlotData[PlotData$index==3,1]<-scale(PlotData[PlotData$index==3,1])
p22<-ggplot(data = PlotData,aes(sample=PlotData[,1]))+
  geom_qq(distribution = stats::qnorm)+
  facet_grid(PlotData[,3]~PlotData[,2])+
  xlab("Normal quantile")+ylab("Scaled data quantile")+
```

```
  theme(aspect.ratio = 1)
p22
```



从正态 QQ 图来看，我们认为此时估计出来的参数具有渐进正态性。