Hypothesis testing for parameters of a Tukey one-df interaction robust regression model

WENXING(Blue) WANG

Dec 3, 2020

Introduction

In genomic studies, we consider associations across genes we are studying and also associations between genes and environments. A model proposed by Chatterjee (2006) and his team adopted a Tukey 1-df association model to study the estimated effect that two groups of genes contribute to a certain disease. We consider the model

$$Y_i = \alpha + S_{i1}^T\beta_1 + S_{i2}^T\beta_2 + \gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2)$$

where S_{i1} and S_{i2} are expression level of two groups of genes as covariates, and $\gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2)$ term denotes the association. We study the application of this model in Logistic Regression and robust regression case

We are interested in testing whether S_{ii} is needed in the model or not, that is,

$$H_0 : \beta_1 = 0.$$

To finish the test, there are two main difficulties: 1. It is hard to estimate β_1. 2. The interaction term $\gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2)$ has been another obstacle. In order to finish the testing, we need to determine the parameter γ. Methods to overcome these two difficulties have been proposed in the paper by Chatterjee's team (2006).

Methods & Material

Logistic Regression Case

The application of Tukey 1-df model for logistic regression can be denoted as:

$$\log\mathrm{it}\left[\Pr\left(D = 1 \mid \mathbf{S}_1, \mathbf{S}_2\right)\right] = \alpha + S_{i1}^T\beta_1 + S_{i2}^T\beta_2 + \gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2)$$

The methods for estimation of γ and test statistics for hypothesis testing of β_1 has been proposed in the paper by Chatterjee's team (2006). 1. Obtain the maximum likelihood estimators for β_2 and β_0 under null hypothesis that $\beta_1 = 0$ and denote the estimator as $\hat{\psi} = \left(\hat{\alpha}, \hat{\beta}_2\right)$. We denote the likelihood function as

$$L = \sum_{i=1}^{N} D_i \log P_{\alpha,\beta_1,\beta_2;\theta}\left(\mathbf{S_{1i}}, \mathbf{S_{2i}}\right) + \left(1 - \mathbf{D_i}\right)\log\left[1 - \mathbf{P}_{\alpha,\beta_1,\beta_2;\theta}\left(\mathbf{S_1}_i\mathbf{S_{2i}}\right)\right]$$

2. For a fixed value γ_0, derive the score function and plug in the value obtained before.

$$S_{\beta_1}(\theta) = \sum_{l=1}^{N}\left(1 + \gamma_0\mathbf{S}_{21}^T\hat{\beta}_2\right)\mathbf{S}_{11}\left[D_i - \hat{P}_{H_0(1)}\left(\mathbf{S}_{2i}\right)\right]$$

3. derive the variance-covariance matrix for the estimated score function and take the its inverse

$$I^{\beta_1\beta_1}(\theta) = \left[I_{\beta_1\beta_1}(\theta) - I_{\beta_1\psi}(\theta)I_{\psi\psi}^{-1}I_{\psi\beta_1}(\theta)\right]^{-1}$$

where

$$I_{\beta_1\beta_1}(\theta) = \sum_{i=1}^{N}\left[1+\gamma_0\hat{\beta}_2^T\mathbf{S}_{2i}\right]^2 \hat{P}_{\text{NULL}}\left(\mathbf{S}_{1i},\mathbf{S}_{2i}\right)\times\left[1-\hat{\mathbf{P}}_{\text{NULL}}\left(\mathbf{S}_{1i},\mathbf{S}_{2i}\right)\right]\mathbf{S}_{1i}\mathbf{S}_{1i}^{\mathbf{T}}$$

$$I_{\beta_1,\psi}(\theta) = \sum_{i=1}^{N}\left[1+\gamma_0\hat{\beta}_2^T\mathbf{S}_{2i} \mid \hat{P}_{\text{NULL}}\left(\mathbf{S}_{1i}\mathbf{S}_{2i}\right)\times\left[1-\hat{\mathbf{P}}_{\text{NULL}}\left(\mathbf{S}_{1i}\mathbf{S}_{2i}\right)\right]\mathbf{S}_{1i}\mathbf{Z}_{2i}^{\mathrm{T}}\right.$$

$$I_{\psi,\psi} = \sum_{i=1}^{N}\hat{P}_{\text{NULL}}\left(\mathbf{S}_{1i}\mathbf{S}_{2i}\right)\left[1-\hat{\mathbf{P}}_{\text{NULL}}\left(\mathbf{S}_{1i},\mathbf{S}_{2i}\right)\right]\mathbf{Z}_{2i}\mathbf{Z}_{2i}^{\mathbf{T}}$$

where $Z=[1,S]$ and N is the sample size.

4. Calculate the test statistic with γ_0

$$T_1(\gamma_0) = S_{\beta_1}(\gamma_0)^T I^{\beta_1\beta_1}(\gamma_0)S_{\beta_1}(\gamma_0)$$

   Compute the final test statistics as $T_1^* = \max_{L\leqslant\gamma\leqslant U} T(\gamma)$ where L and U denote some prespecified values for lower and upper limits of γ, respectively.

5. The maximized $T_1(\gamma)$ follows a chi-square distribution and based on this, the hypothesis testing can be conducted.

Linear Regression Case

We tried to apply the same mathod for linear regression case. In linear regression, the model is denoted as:

$$Y_i = \alpha + S_{i1}^T\beta_1 + S_{i2}^T\beta_2 + \gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2) + e_i,$$

and e_i are i.i.d $N(0,\sigma^2)$ errors. In the case of linear regression, we applied the methods for logistics regression, the first step is also to determine the likelihood function.

$$\ell = -\frac{n}{2}log(2\pi\sigma^2) - \sum_{i=1}^{n}[Y_i - \alpha - S_{i1}^T\beta_1 - S_{i2}^T\beta_2 - \gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2)]^2/(2\sigma^2)$$

And then the score function for β_1 is

$$\Psi_{\beta_2}(\beta_0,\beta_1,\beta_2,\sigma^2) = \frac{\delta\ell}{\delta\beta_2} = \sum_{i=1}^{n} S_{i2}\{1+\gamma S_{i1}^T\beta_1\}[Y_i - \beta_0 - S_{i1}^T\beta_1 - S_{i2}^T\beta_2 - \gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2)]/\sigma^2$$

The variance-covariance matrix is a little bit more complex since we have 4 unknown parameters in this case while there are only 3 in logistic regression case.

$$I = \begin{bmatrix} I_{\beta_1\beta_1} & I_{\beta_1\alpha} & I_{\beta_1\beta_2} & I_{\beta_1\sigma^2} \\ I_{\beta_1\alpha} & I_{\alpha\alpha} & I_{\alpha\beta_2} & I_{\alpha\sigma^2} \\ I_{\beta_2\beta_1} & I_{\beta_2\alpha} & I_{\beta_2\beta_2} & I_{\beta_2\sigma^2} \\ I_{\sigma^2\beta_1} & I_{\sigma^2\alpha} & I_{\sigma^2\beta_2} & I_{\sigma^2\sigma^2} \end{bmatrix}$$

Denote L as the likelihood function and $I_{\beta_1\beta_1} = \dfrac{dL}{d\beta_1 d\beta_1^T}$ and also the same for other components of I.

$$I_{\beta_0\beta_1} = E[\Psi_{\beta_0}\Psi_{\beta_1}^T] = E[\sum_{i=1}^{n} S_{i1}\{1+\gamma S_{i2}^T\beta_2\}e_i^2/\sigma^4] = \sum_{i=1}^{n} S_{i1}\{1+\gamma S_{i2}^T\beta_2\}/\sigma^2$$

$$I_{\beta_0\beta_2} = E[\Psi_{\beta_0}\Psi_{\beta_2}^T] = E[\sum_{i=1}^{n} S_{i2}\{1+\gamma S_{i1}^T\beta_1\}e_i^2/\sigma^4] = \sum_{i=1}^{n} S_{i2}\{1+\gamma S_{i1}^T\beta_1\}/\sigma^2$$

$$I_{\beta_1\beta_1} = E[\Psi_{\beta_1}\Psi_{\beta_1}^T] = E[\sum_{i=1}^{n} S_{i1}S_{i1}^T\{1+\gamma S_{i2}^T\beta_2\}^2 e_i^2/\sigma^4] = \sum_{i=1}^{n} S_{i1}S_{i1}^T\{1+\gamma S_{i2}^T\beta_2\}^2/\sigma^2$$

$$I_{\beta_1\beta_2} = E[\Psi_{\beta_1}\Psi_{\beta_2}^T] = E[\sum_{i=1}^{n} S_{i1}S_{i2}^T\{1+\gamma S_{i1}^T\beta_1\}\{1+\gamma S_{i2}^T\beta_2\} e_i^2/\sigma^4] = \sum_{i=1}^{n} S_{i1}S_{i2}^T\{1+\gamma S_{i1}^T\beta_1\}\{1+\gamma S_{i2}^T\beta_2\}/\sigma^2$$

$$I_{\beta_2\beta_2} = E[\Psi_{\beta_2}\Psi_{\beta_2}^T] = E[\sum_{i=1}^{n} S_{i2}S_{i2}^T\{1+\gamma S_{i1}^T\beta_1\}^2 e_i^2/\sigma^4] = \sum_{i=1}^{n} S_{i2}S_{i2}^T\{1+\gamma S_{i1}^T\beta_1\}^2/\sigma^2$$

$$I_{\beta_0\sigma^2} = E[\Psi_{\beta_0}\Psi_{\sigma^2}] = 0,\ I_{\beta_1\sigma^2} = E[\Psi_{\beta_1}\Psi_{\sigma^2}] = 0,\ I_{\beta_2\sigma^2} = E[\Psi_{\beta_2}\Psi_{\sigma^2}] = 0$$

$$I_{\sigma^2\sigma^2} = E[\Psi_{\sigma^2}\Psi_{\sigma^2}] = E[\{-\frac{n}{2\sigma^2} + \sum_{i=1}^{n} e_i^2/(2\sigma^4)\}^2] = var[\sum_{i=1}^{n} e_i^2/(2\sigma^4)] = \frac{n}{2\sigma^4}$$

and also

$$I_{\beta_1\psi} = \begin{bmatrix} I_{\beta_1\alpha} & I_{\beta_1\beta_2} & I_{\beta_1\sigma^2} \end{bmatrix}$$

$$I_{\psi\psi} = \begin{bmatrix} I_{\alpha\alpha} & I_{\alpha\beta_2} & I_{\alpha\sigma^2} \\ I_{\beta_2\alpha} & I_{\beta_2\beta_2} & I_{\beta_2\sigma^2} \\ I_{\sigma^2\alpha} & I_{\sigma^2\beta_2} & I_{\sigma^2\sigma^2} \end{bmatrix}$$

The same method for logistic regression is also applied for the linear regression case to compute the test statistic and our simulation results indicated that the same method also works well for linear regression with Tukey 1-df interaction model.

Absolute Regression Case

The main difficulty for absolute regression is that there is no closed for estimators like logistic regression and linear regression. Denote $\mu = \alpha + S_{i1}^T\beta_1 + S_{i2}^T\beta_2 + \gamma(S_{i1}^T\beta_1)(S_{i2}^T\beta_2)$, the likelihood function for absolute regression case is

$$l = -\ln\sigma - \frac{\sum_{i=1}^{N}|Y_i - \mu_i|}{2\sigma^2}$$

and the score function is

$$\frac{-\sum_{i=1}^{N} sign\left(Y_i - \mu_i\right) S_{1i}\left(1+\gamma S_{2i}\beta_{2i}\right)}{2\sigma^2}$$

The second problem is that due to the sign function, the derivative of the score function will be 0, which means the method we applied for computing variance-covariance matrix is not suitable in this situation. We applied Weak Law of Large Numbers to solve this problem. When sample size is large enough, the estimated value for β_1, β_2, α and σ can be treated as the real values of these parameters. Thus, we derived the variance-covariance matrix directly from the score function. (modification needed)

$$Var\left(S\left(\beta_{1_n}\right)\right) = \frac{\sum_{i=1}^{N} S_{1_n i}^2\left(1+\gamma S_{2i}\beta_{2i}\right)^2}{4\sigma^4}$$

$$Cov\left(S\left(\beta_{1_p}\right), S(\beta_{1_q})\right) = \frac{\sum_{i=1}^{N} S_{1_p i}S_{1_q i}\left(1+\gamma S_{2i}\beta_{2i}\right)^2}{4\sigma^4}$$

for all n, p, q smaller than the length of S_1 vector and $p \neq q$. The rest of the steps are the same as the logistic regression to determine the test statistic. The simulation has also given a good result under absolute regression case.

Permutation test

Because of the dependence of WLLN and CLT, the distribution of the maximized test statistic for logistic regression is not chi-square anymore. It appears like a normal distribution and it is hard to determine its mean and variance. Thus we consider the permutation test to finish the hypothesis testing.
The method is proposed by Freedman and Lane (1983), which involves the following steps:

1. estimate the value of β_2 under the null hypothesis that $\beta_1 = 0$
2. plug in the estimator in the reduced model and get the residule for each sample
3. permuate the residues randomly and produce $R^*_{Y|X}$
4. New values of Y^* are calculated by adding $R^*_{Y|X}$ to the fitted values, that is $Y^* = \hat{\alpha} + S_2\hat{\beta}_2 + R^*_{Y|X}$
5. compute the test statistic with Y^* obtained before and covariates.

Results

Performance of test statistics

We designed a "Tukey" function for the cases of linear regression and logistic regression.

```
library(carData)
library(sp)
library(raster)
library(fastmatrix)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following objects are masked from 'package:raster':
##
##     area, select
```

```
library(L1pack)
tukey=function(Y,S1,S2,lower,upper,loss)
{
  S=cbind(S1,S2)
  N=length(Y)
  gamma_vec=seq(lower,upper,by=0.05)
  pscore=gamma_vec
  trial=length(pscore)
  if (loss=="absolute")
  {
    sd=rlm(Y ~ S2, psi = psi.huber, scale.est="proposal 2")$s
    MLEline=l1fit(S2,Y)
    fi=MLEline$coefficients
    varr=sd^2
    betahat=fi[2:6]
    #---------get the score function------------
    for (i in 1:trial)
    {
      gamma=gamma_vec[i]
      scorevector=c(0,0,0,0,0)
      for (k in 1:5)
      {
```

```
      vec=-(S1[,k]+gamma*S1[,k]*(S2%*%betahat))*sign(Y-fi[1]-S2%*%betahat)
      scorevector[k]=colSums(vec)
    }
    scorevector=scorevector/(2*sd^2)
    intermediate=cbind(1+gamma*(S2%*%betahat),1+gamma*(S2%*%betahat),
                       1+gamma*(S2%*%betahat),1+gamma*(S2%*%betahat),
                       1+gamma*(S2%*%betahat))
    cov=t(S1*intermediate)%*%(S1*intermediate)
    cov=cov/(4*sd^4)
    pscore[i]=t(scorevector)%*%solve(cov)%*%scorevector
  }
  return (list(TG=which.max(pscore)*0.05-5,Tscore=max(pscore),Pscore=pscore))
}
if (loss=="logistic")
{
  MLEline=glm(Y~S2,family=binomial(link="logit"))
  MLEtable=summary(MLEline)
  fi=c(MLEtable$coefficients[,1])
  betahat=fi[2:6]
  ###-------------get the score function----------------###
  prob=exp(cbind(1,S2)%*%fi)/(1+exp(cbind(1,S2)%*%fi))
  for (l in 1:trial)
  {
    ###-------------get the score function vector-------------###
    otherpart=(1+gamma_vec[l]*(S2%*%betahat))*(Y-prob)
    otherparts=cbind(otherpart,otherpart,otherpart,otherpart,otherpart)
    scorevec=c(colSums(otherparts*S1))
    ###-------------get the fisher information----------------###
    vector_part_1=(1+gamma_vec[l]*(S2%*%betahat))^2*prob*(1-prob)
    vector_part_2=prob*(1-prob)
    vector_part_3=(1+gamma_vec[l]*(S2%*%betahat))*prob*(1-prob)
    I_beta1_beta1=matrix(data=0,nrow=5,ncol=5)
    I_beta1_fi=matrix(data=0,nrow=5,ncol=6)
    I_fi_fi=matrix(data=0,nrow=6,ncol=6)
    for (q in 1:100)
    {
      SKvec=c(S1[q,])
      I_beta1_beta1=I_beta1_beta1+(SKvec%*%t(SKvec))*vector_part_1[q]
    }
    for (q in 1:100)
    {
      Svec=c(S1[q,])
      Zvec=c(1,S2[q,])
      I_beta1_fi=I_beta1_fi+(Svec%*%t(Zvec))*vector_part_3[q]
    }
    for (q in 1:100)
    {
      Z2=c(1,S2[q,])
      I_fi_fi=I_fi_fi+(Z2%*%t(Z2))*vector_part_2[q]
    }
    fisher=solve(I_beta1_beta1-I_beta1_fi%*%solve(I_fi_fi)%*%t(I_beta1_fi))
    pscore[l]=t(scorevec)%*%fisher%*%scorevec
  }
```

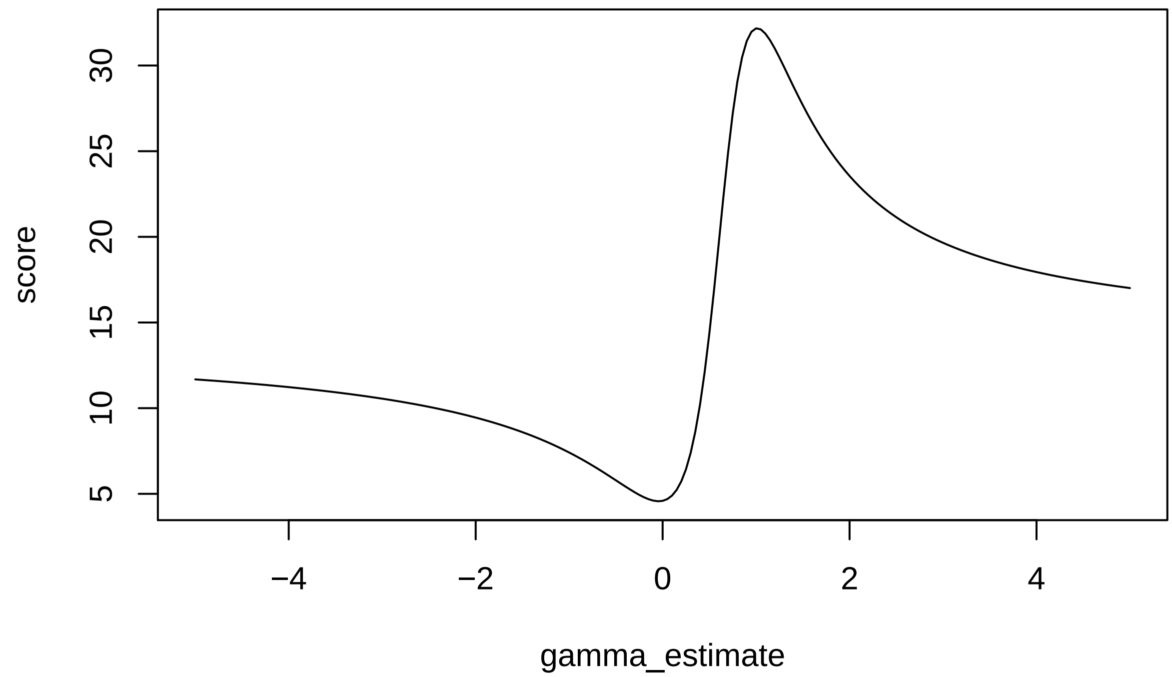
```r
  return (list(TG=which.max(pscore)*0.05-5,Tscore=max(pscore),Pscore=pscore))
}
if (loss=="linear")
{
  MLEline=lm(Y~S2)
  MLEtable=summary(MLEline)
  fi=c(MLEtable$coefficients[,1])
  betahat=fi[2:6]
  #---------get the score function------------
  error_estimates=MLEtable$sigma
  for (l in 1:trial)
  {
    #score function
    otherpart=(1+gamma_vec[l]*(S2%*%betahat))*(Y-cbind(1,S2)%*%fi)
    otherparts=cbind(otherpart,otherpart,otherpart,otherpart,otherpart)
    score=as.vector(colSums(otherparts*S1))/(error_estimates^2)
    #---------get the I_beta1_beta1------------
    utlpart1=(1+gamma_vec[l]*(S2%*%betahat))
    I_beta1_beta1=matrix(data=0,ncol=5,nrow=5)
    for(i in 1:100)
    {
      I_beta1_beta1=I_beta1_beta1+
        (utlpart1[i])^2*(S1[i,]%*%t(S1[i,]))
    }
    I_beta1_beta1=I_beta1_beta1/(error_estimates^2)
    #---------get the I_beta1_fi------------
    I_beta1_sigma=0
    I_beta1_beta2=matrix(data=0,ncol=5,nrow=5)
    for(i in 1:100)
    {
      I_beta1_beta2=I_beta1_beta2+
        utlpart1[i]*(S1[i,]%*%t(S2[i,]))
    }
    I_beta1_beta2=I_beta1_beta2/(error_estimates^2)
    I_beta1_alpha=c(rep(0,5))
    for(i in 1:100)
    {
      I_beta1_alpha=I_beta1_alpha+S1[i,]*utlpart1[i]
    }
    I_beta1_alpha=I_beta1_alpha/(error_estimates^2)
    I_beta_fi=cbind(I_beta1_alpha,I_beta1_beta2,I_beta1_sigma)
    #---------get the I_fi_fi------------
    I_fi_fi=matrix(data=0,ncol=7,nrow=7)
    I_alpha_alpha=100/(error_estimates^2)
    I_alpha_sigma=0
    I_sigma_sigma=100/2*(error_estimates^4)
    I_beta2_beta2=matrix(data=0,ncol=5,nrow=5)
    for(i in 1:100)
    {
      I_beta2_beta2=I_beta2_beta2+
        S2[i,]%*%t(S2[i,])
    }
    I_beta2_beta2=I_beta2_beta2/(error_estimates^2)
```

```r
      I_beta2_sigma=0
      I_beta2_alpha=colSums(S2)/(error_estimates^2)

      I_fi_fi[2:6,2:6]=I_beta2_beta2
      I_fi_fi[1,1]=I_alpha_alpha
      I_fi_fi[7,7]=I_sigma_sigma
      I_fi_fi[1,2:6]=I_beta2_alpha
      I_fi_fi[2:6,1]=I_beta2_alpha
      I_fi_fi[7,2:6]=I_beta2_sigma
      I_fi_fi[2:6,7]=I_beta2_sigma
      #---------get the information-------
      info=I_beta1_beta1-I_beta_fi%*%solve(I_fi_fi)%*%t(I_beta_fi)
      #---------get the value------------
      pscore[l]=t(score)%*%solve(info)%*%score
    }
    return (list(TG=which.max(pscore)*0.05-5,Tscore=max(pscore),Pscore=pscore))
  }
}
```

This function is used to compute the maximized test statistic, Users only need to input Y, S_1, S_2 and type of regression. Simulation studies are conducted based on each case of regression. ### Logistic regression

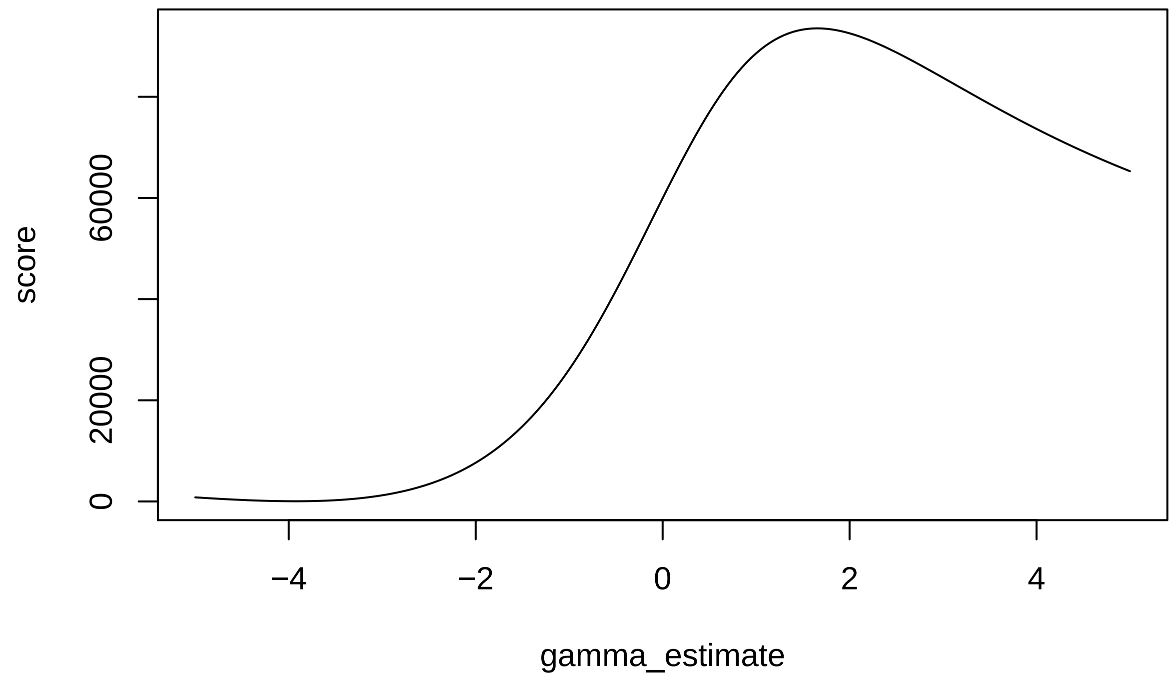
```r
N=100;k1=5;k2=5
beta1=c(1,-1,2,-1,-2);beta2=beta1;beta=c(beta1,beta2)
sigma=100;alpha=3;gamma_true=1
tmp=sample(x = c(0,1), size = 100*10, replace = TRUE)
S=matrix(tmp, nrow = 100, ncol = 10)
Sreduced=S[,6:10];Sk1=S[,1:5]
Y=c(rep(0,100))
miu=S %*% beta + alpha + gamma_true*((Sk1%*%beta1)*(Sreduced%*%beta2))
rate=exp(miu)/(1+exp(miu))
Y=rbinom(100,1,rate)
score=tukey(Y,Sk1,Sreduced,-5,5,"logistic")$Pscore
gamma_estimate=c(seq(-5,5,by=0.05))
plot(gamma_estimate,score,type="l")
```



The maximizer of the test statistic is roughly around the true γ, which is 1.

Linear regression with quadratic loss

```
num_of_dataset=1000
N=5000
k1=5
k2=5
beta1=c(rep(0.7,5))
beta2=c(rep(0.2,5))
beta=c(beta1,beta2)
sigma=3
alpha=3
gamma_true=1
tmp=rnorm(N*10)
Y=c(rep(0,N))
S=matrix(tmp, nrow = N, ncol = 10)
Sk2=S[,6:10]
Sk1=S[,1:5]
#----------generate dataset---------
Y=rnorm(N,S %*% beta + alpha + gamma_true*((Sk1%*%beta1)*(Sk2%*%beta2)),sigma)
score=tukey(Y,Sk1,Sk2,-5,5,"linear")$Pscore
gamma_estimate=c(seq(-5,5,by=0.05))
plot(gamma_estimate,score,type="l")
```

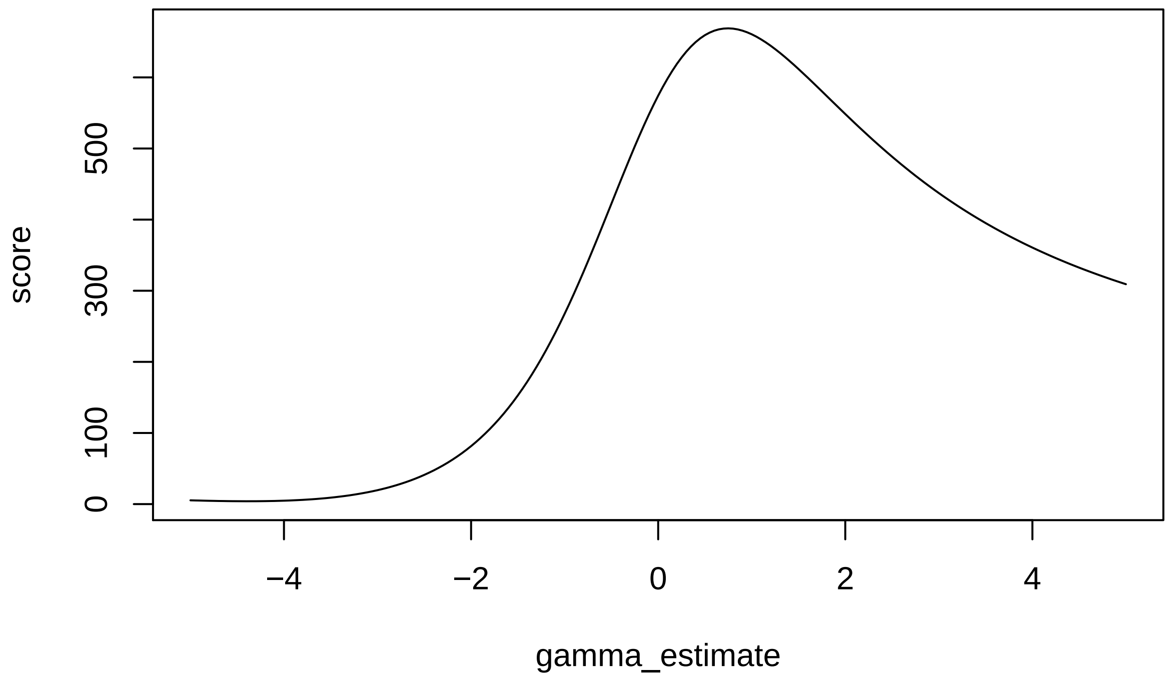


The maximizer of the test statistic is also roughly around the true γ, which is 1.

Linear regression with absolute loss

In this case, we apply the dataset used in linear regression with quadratic loss.

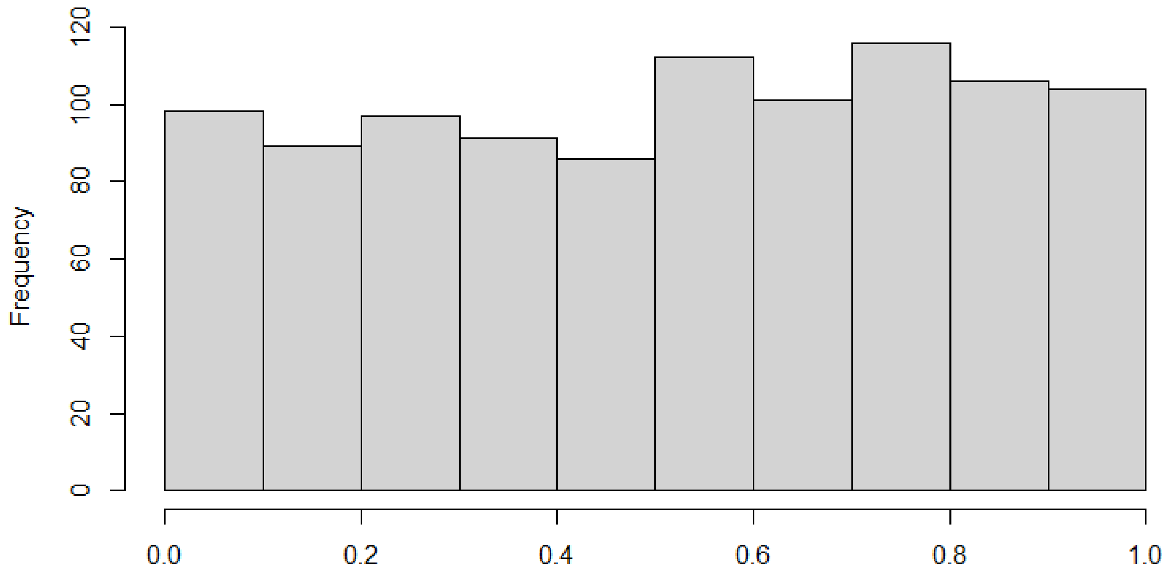
```
score=tukey(Y,Sk1,Sk2,-5,5,"absolute")$Pscore
gamma_estimate=c(seq(-5,5,by=0.05))
plot(gamma_estimate,score,type="l")
```



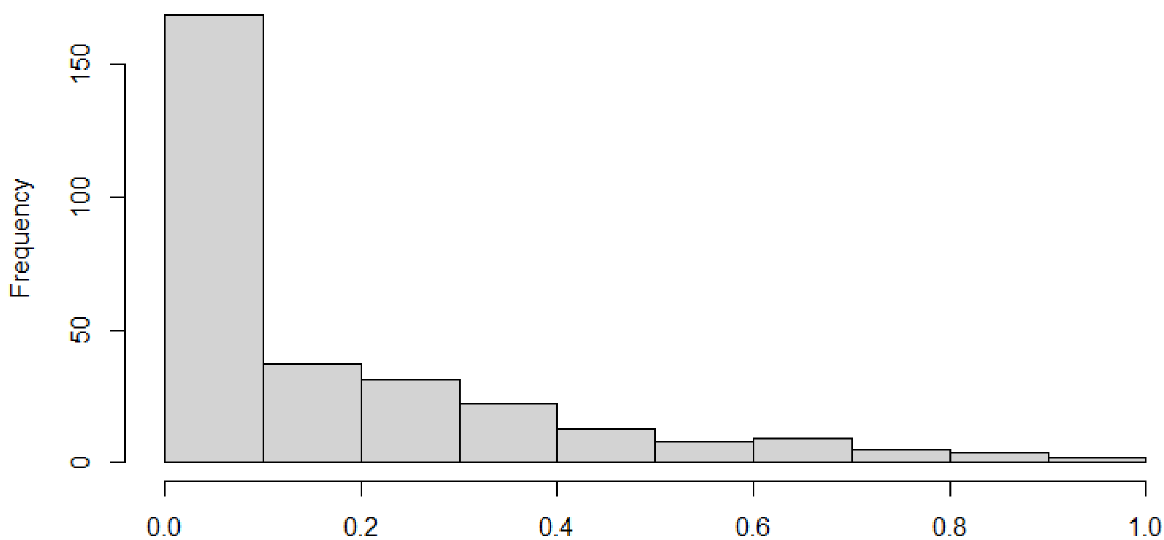
The image is roughly the same as the case with quadratic loss.

Permutation test

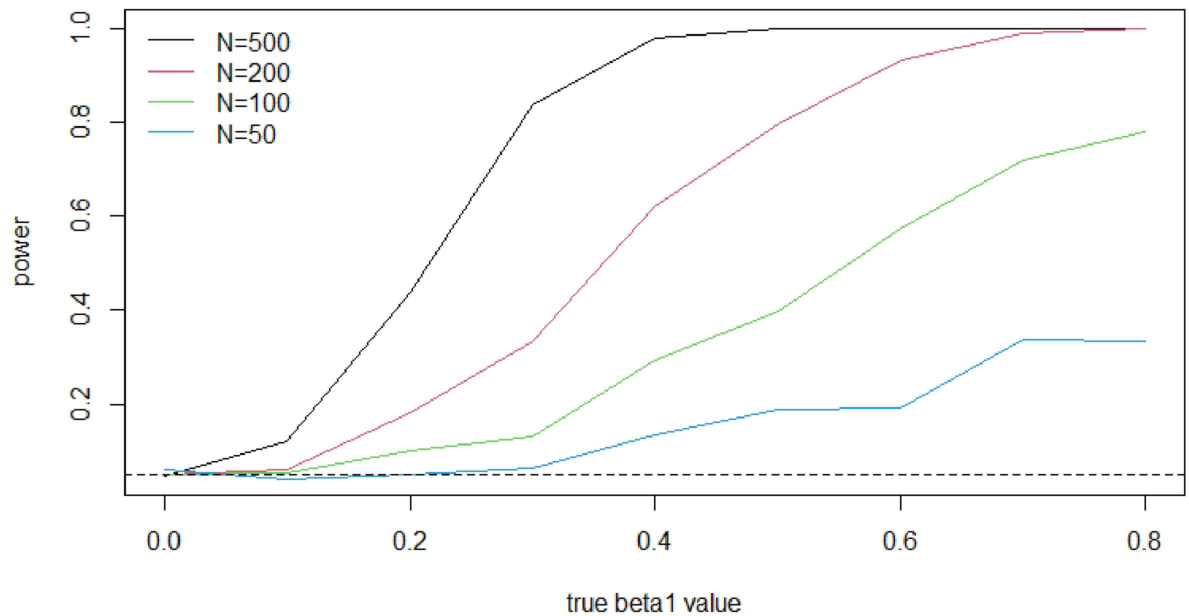
To test the rate of type 1 error, 1000 tests were conducted for the situation that null hypothesis is true (i.e. $\beta_1 = 0$). When sample size is 500 for each time of test, the histogram plot for α values is displayed, which is roughly a uniform distribution.



If we increase β_1, numbers will be gathered near 0. For instance, if we set $\beta_1 = 0.2$, the result looks like:



If we set $\alpha = 0.05$, the power curves under sample size N=50,100,200 and 500 are like



Simulation studies

To test if the regression analysis with absolute (Huber) loss can better resist the effect of outliers in this case, simulation studies are conducted with simulated datasets out of other several distributions, with each containing some outliers. In these datasets, S values (independent variables) are generated by standard normal distribution of the variance of 1 while Y (dependent variables) are generated based on S part of datasets with the following formula:

$$Y = \alpha + S_1^T\beta_1 + S_2^T\beta_2 + \gamma(S_1^T\beta_1)(S_2^T\beta_2) + \varepsilon$$

There are in total 10 covariates in S, with 5 controlled by the vector β_1 and the other 5 controlled by the vector β_2. β_2 was set to be 0.2 for all 5 values in the vector, γ was set to be 1 and α was 3. ε were generated with the following pre-set distributions and parameters.

	mean	the other paramater
Logistic distribution	0	$\frac{3\sqrt{3}}{\pi}$
Laplace distribution	0	$\frac{3}{\sqrt{2}}$
Cauchy distribution	0	3
Mixed distribution	0	$3, 10, 20, 70$

Logistic distribution, Laplace distribution, Cauchy distribution and Mixed distributions were applied to generate the value of ε during the simulations. The former 3 distributions contain 2 parameters, with one determining the mean of the distribution and the other determining the variance of the distribution. Since The method using mixed distributions is elaborated in the next section of this part. The simulation studies are conducted to test the power of hypothesis testing on β_1 with null hypothesis to be

$$H_0 : \beta_1 = 0$$

The type 1 error is estimated with 1000 times of simulation for each set of fixed parameter. In each time of simulation to find out the type 1 error, all 5 values in β_1 is set to be 0. The power of this hypothesis testing method is estimated with 300 times of simulation for each set of fixed parameter. In each time of simulation, β_1 is set from 0.1 to 0.8 with a gradient of increase of 0.1. Apart from different parameter sets, there are in total 4 different sample sizes, which are 50, 100, 200, 500 applied for each simulation study.

Type 1 error

The type 1 errors are generated with the sample size of 1000. The following table records the type 1 error of the hypothesis testing method with absolute loss to deal with data generated from different distributions with outliers.

	cauchy	laplace	logistic
50	0.052	0.055	0.051
100	0.053	0.048	0.06
200	0.064	0.058	0.047
500	0.041	0.052	0.049

The following table records the type 1 error of the hypothesis testing method with absolute loss to deal with data generated from mixed distributions with different parameters.

	0.1,3,10	0.2,3,20	0.2,3,70
50	0.062	0.056	0.05
100	0.058	0.063	0.059
200	0.054	0.056	0.066
500	0.054	0.063	0.049

There were no very extreme values in two tables above, indicating that the test statistics generated by absolute regression method performs well to resist outliers in terms of predicting type 1 error. Therefore, all datasets will be tested for their powers.

The following table records the type 1 error of the hypothesis testing method with quadratic loss to deal with data generated from different distributions with outliers.

	cauchy	laplace	logistic
50	0.186	0.063	0.068
100	0.159	0.088	0.064
200	0.133	0.04	0.051
500	0.141	0.162	0.047

The following table records the type 1 error of the hypothesis testing method with quadratic loss to deal with data generated from mixed distributions with different parameters.

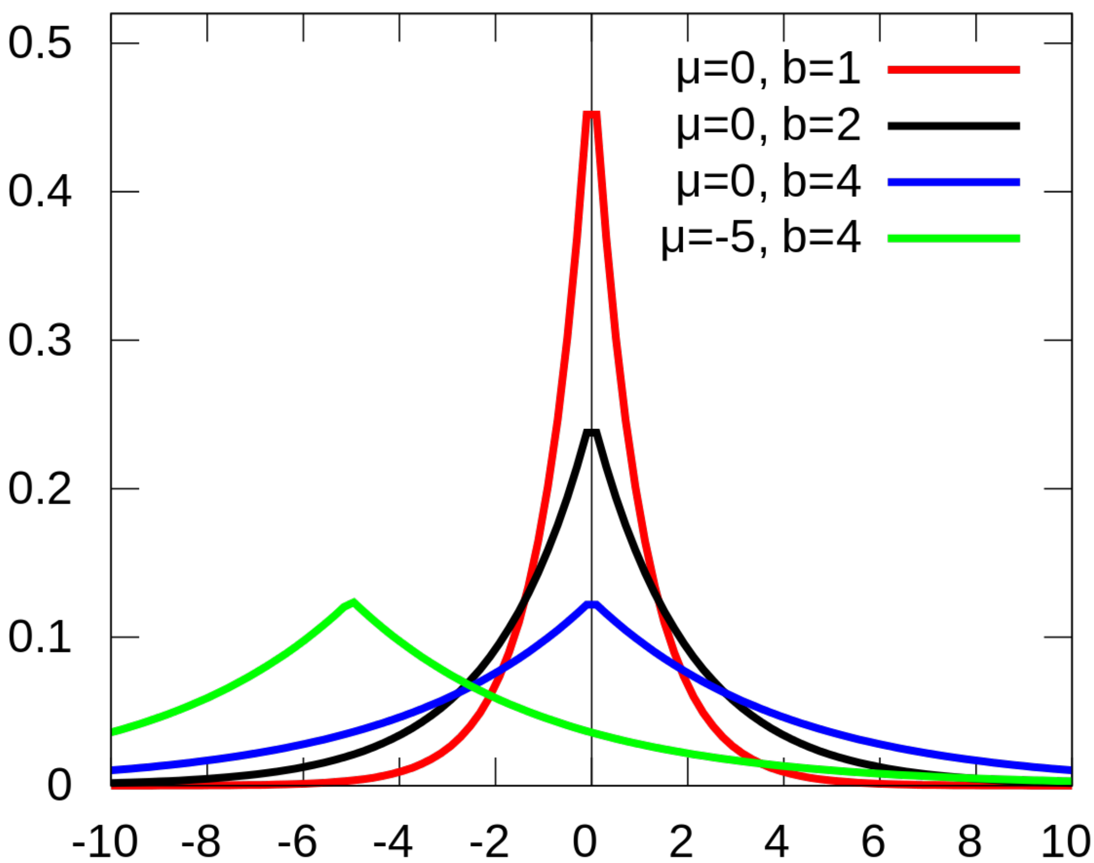
	0.1,3,10	0.2,3,20	0.2,3,70
50	0.064	0.083	0.158
100	0.068	0.078	0.09
200	0.052	0.065	0.097
500	0.063	0.045	0.065

There were a few extreme values in two tables above, indicating that the test statistics generated by quadratic regression method performs not as well as the test statistics generated by absolute regression method to resist outliers in terms of predicting type 1 error. All type 1 errors with datasets generated by cauchy distribution were too big so they will not be tested for powers. Other datasets will be tested for powers.

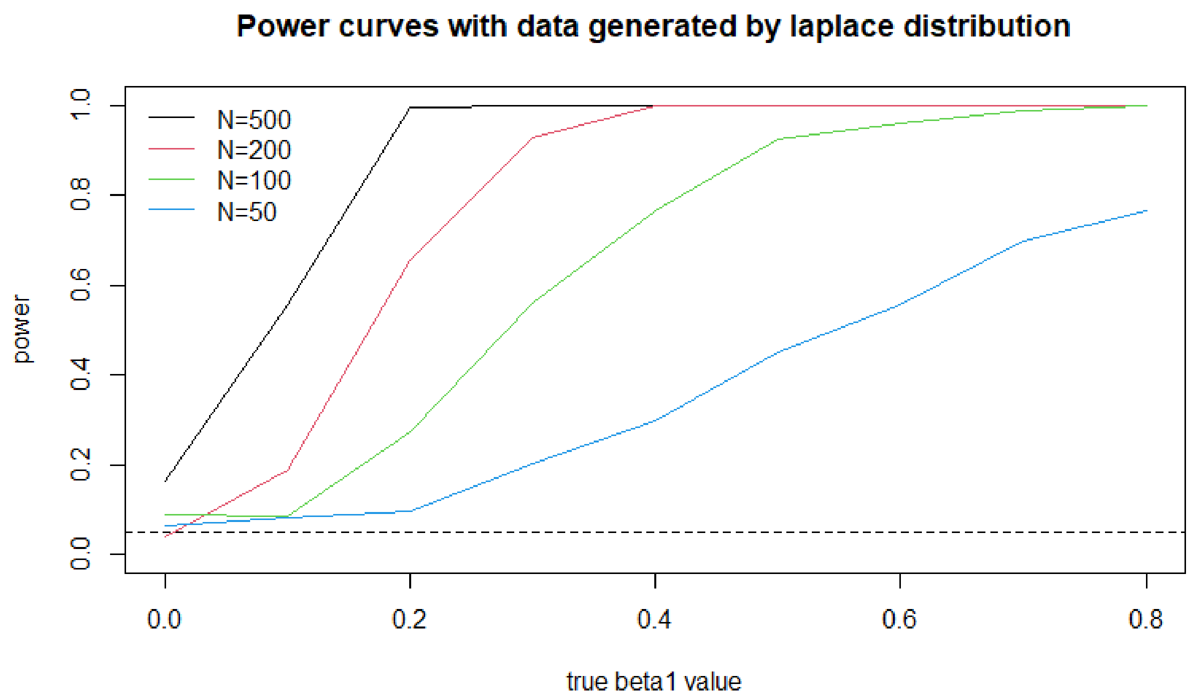
Laplace Distribution

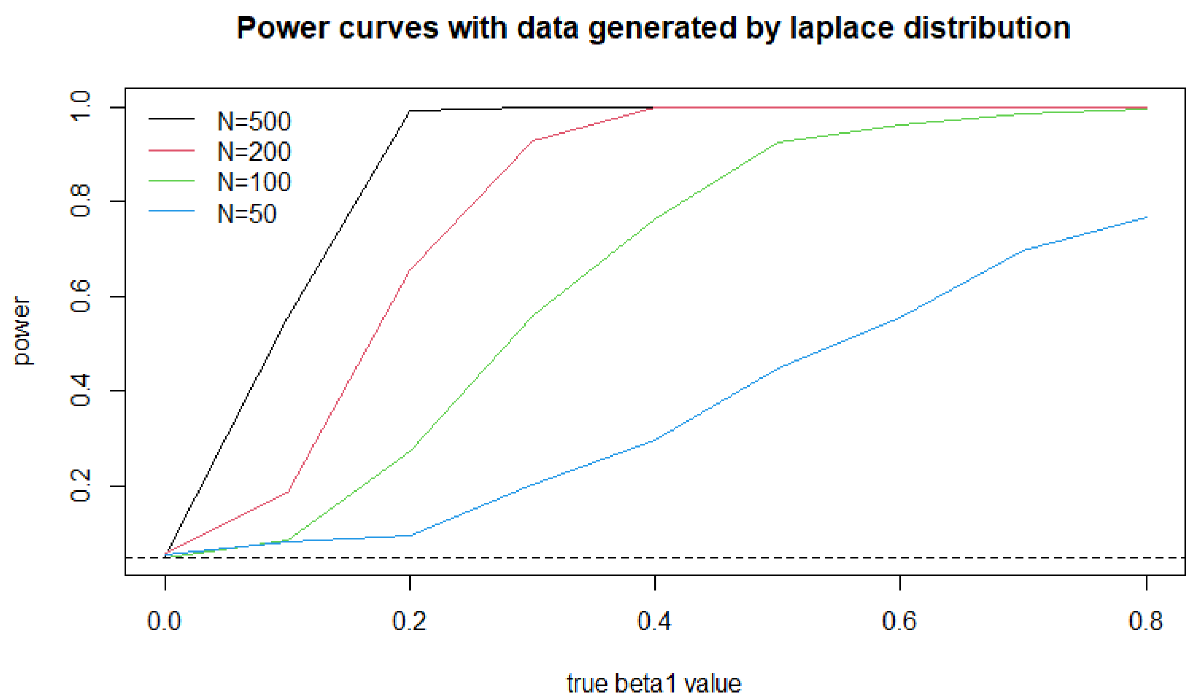
Laplace distribution is a good way to generate outliers since it is heavily tailed. It is also called double exponential distribution. Its probability density function and shapes are shown below:

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$$



The following two figures display the power curve of test statistics generated by regression methods with quadratic and absolute loss to deal with data generated from laplace distribution, correspondingly.

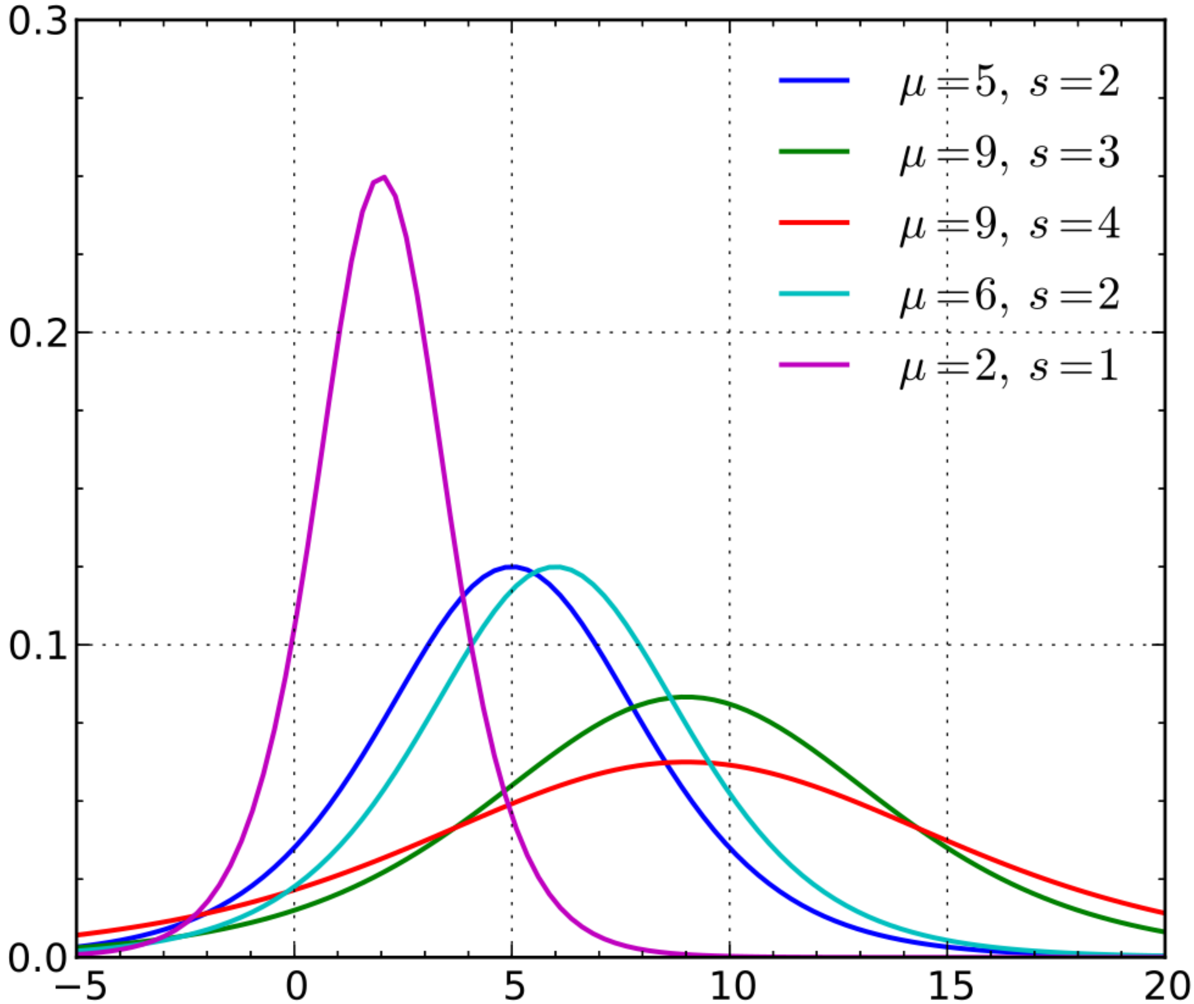




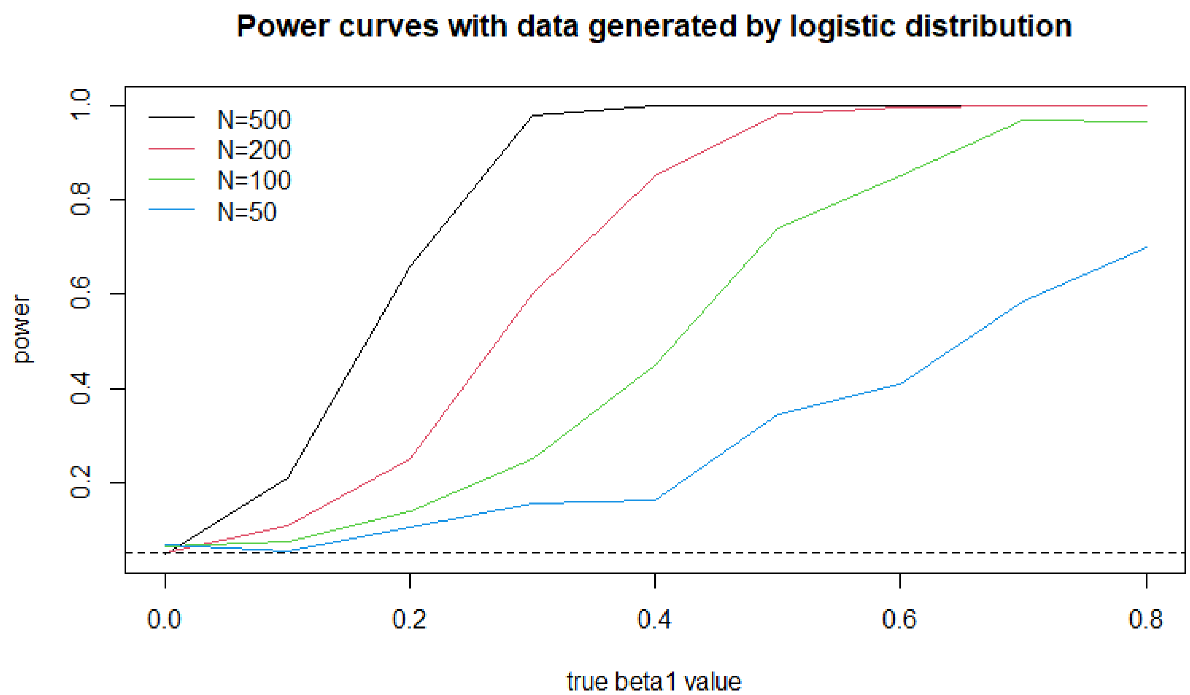
Logistic Distribution

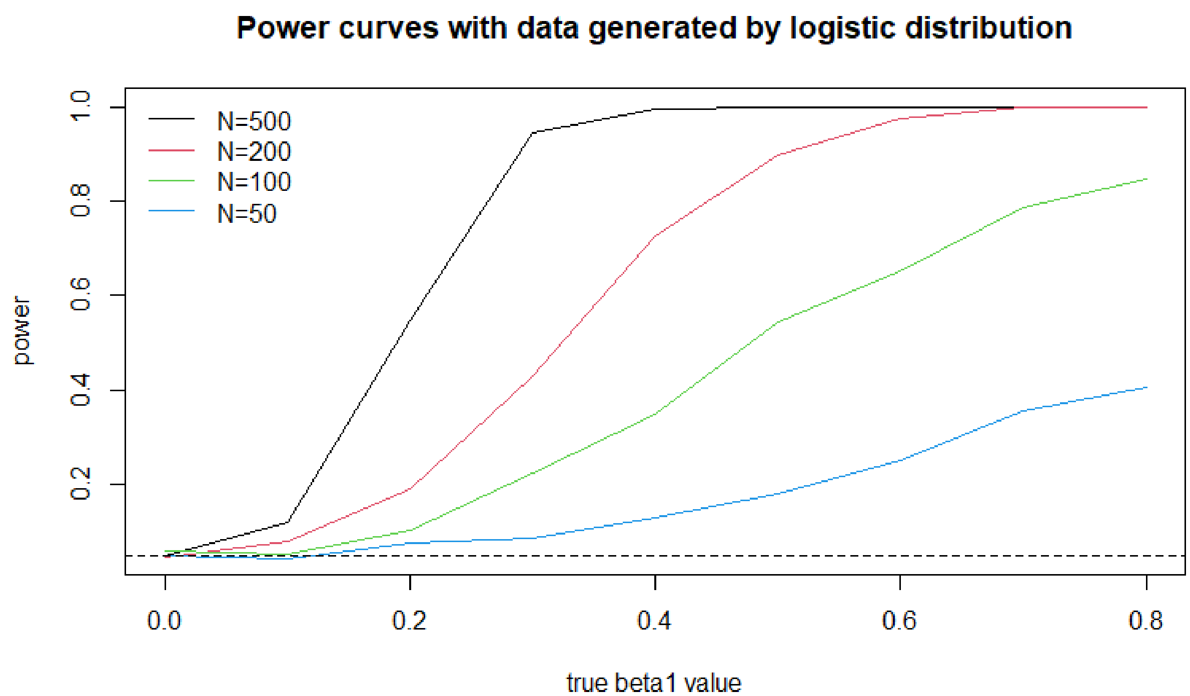
Logistic distribution is another good way to generate outliers since it is also heavily tailed. It has a shape similar to normal distribution. It is a special case of the Tukey lambda distribution. Its probability density function and shapes are shown below:

$$f(x) = \frac{e^{-(x-\mu)/s}}{s\left(1+e^{-(x-\mu)/s}\right)^2}$$



The following two figures display the power curve of test statistics generated by regression methods with quadratic and absolute loss to deal with data generated from logistic distribution, correspondingly.

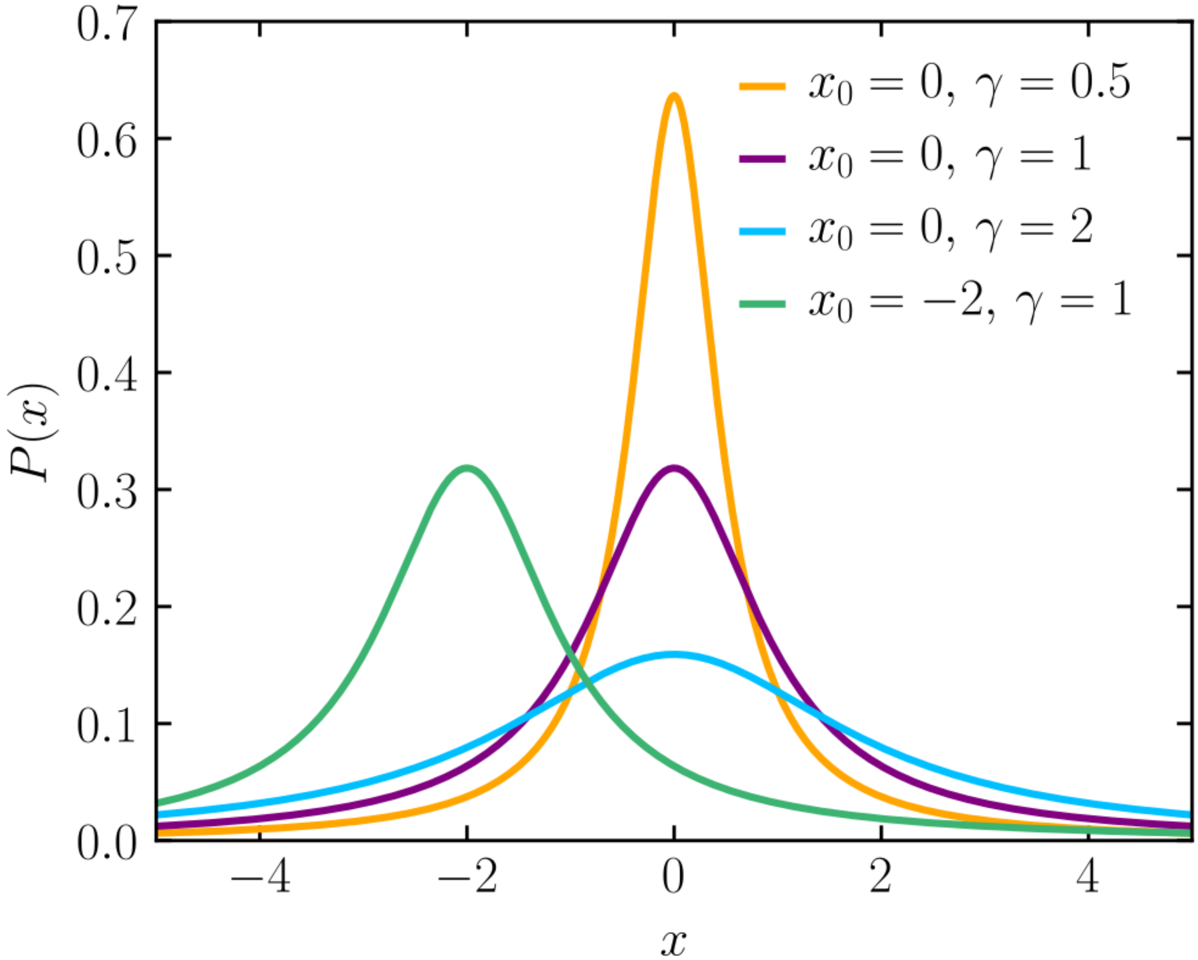




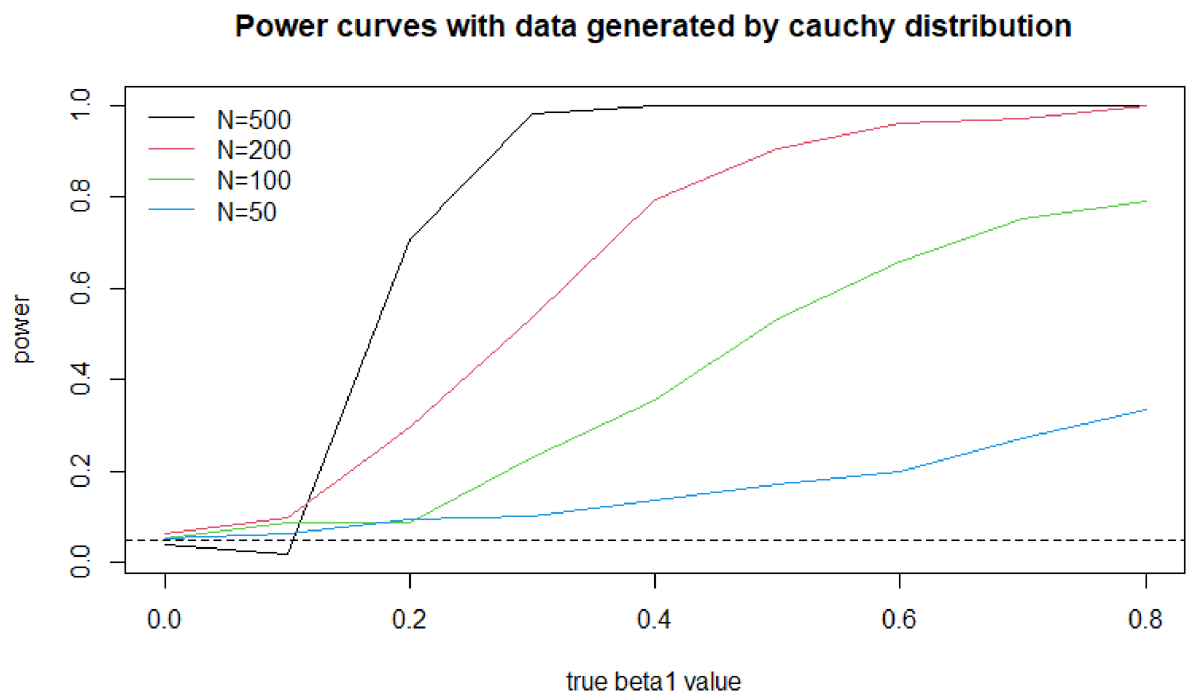
Cauchy distribution

Cauchy distribution is a good way to generate very extreme outliers since it has no moments. Its variance is infinity and it is applied in this project to assess the ability of two different losses to resist very extreme outliers. Its probability density function and shapes are shown below:

$$f(x) = \frac{1}{\pi\gamma\left[1 + \left(\frac{x - x_0}{\gamma}\right)^2\right]}$$



The following figure displays the power curve of test statistics generated by regression methods with absolute loss to deal with data generated from cauchy distribution. Since the test statistics generated by regression methods with quadratic loss did not pass the test of type 1 error, its power is not estimated.

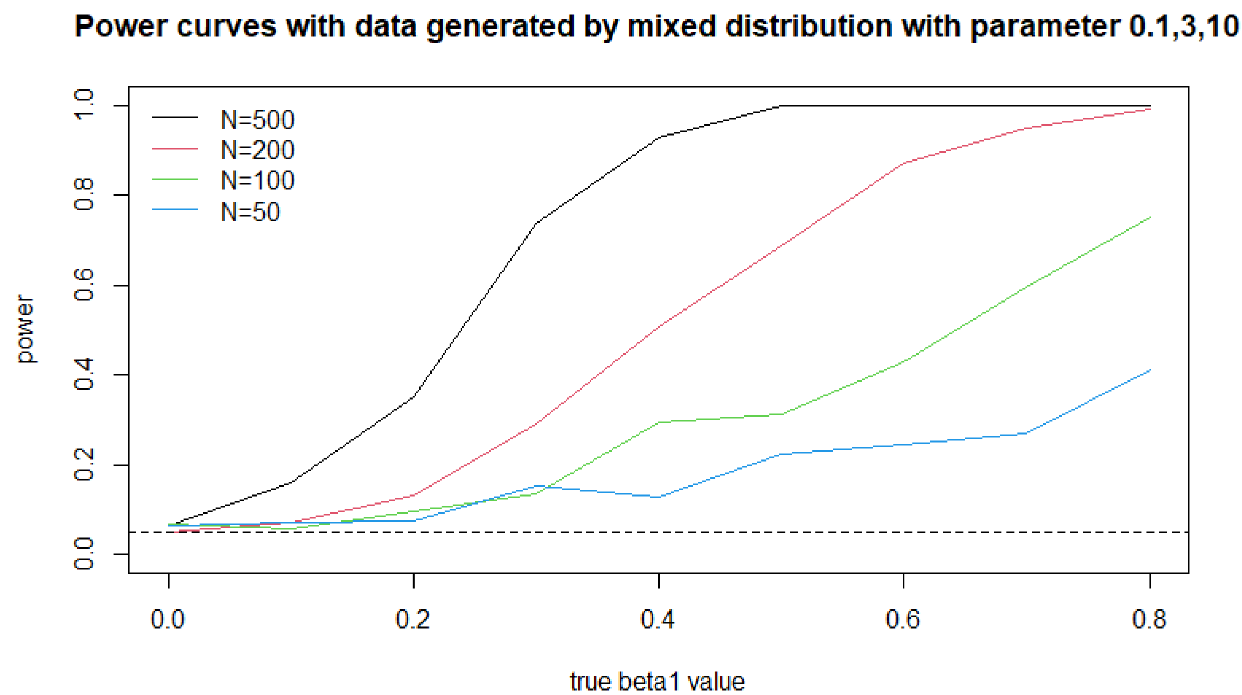


Mixed distribution

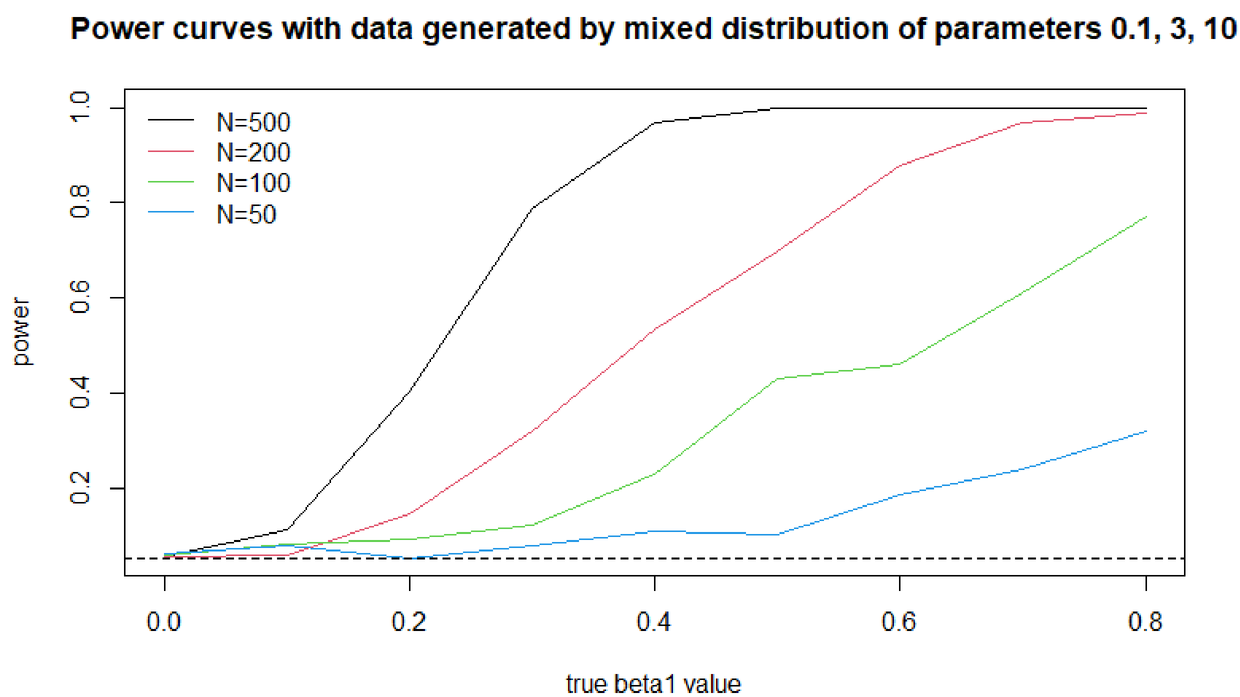
A mixed distribution refers to the distribution of a random variable that is derived from other several distributions each with a certain fixed probability. For example, in one of the simulation studies, data are generated from the distribution $rnorm(0, 3)$ with the probability of 0.9 and data are generated from the distribution $rnorm(0, 10)$ with the probability of 0.1. The purpose of this simulation is to test the performance of the two test statistics generated by two loss functions with the controlled percentage of outliers in the whole dataset. There are in total 3 datasets generated.

	mean	sd 1	sd 2	percentage of sd 2
Dataset 1	0	3	10	0.1
Dataset 2	0	3	20	0.2
Dataset 3	0	3	70	0.2

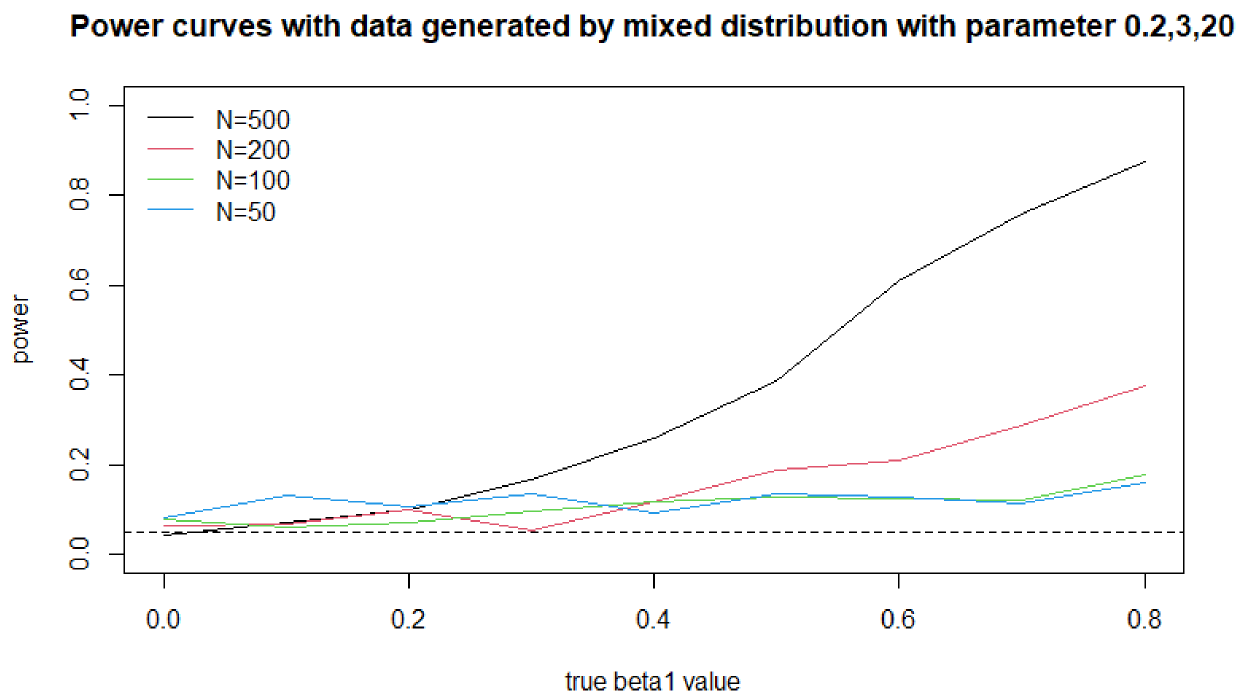
The following two figures display the power curve of test statistics generated by regression methods with quadratic and absolute loss to deal with data generated from mixed distribution with the parameter 0.1,3,10,



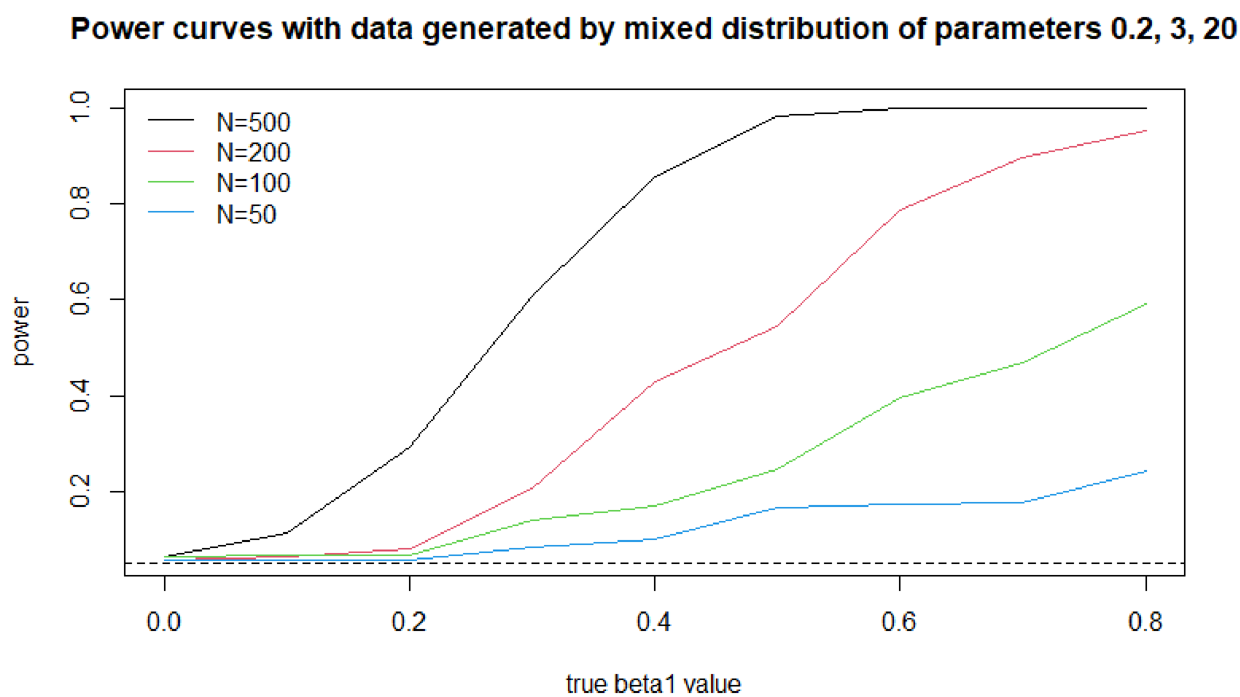
correspondingly.



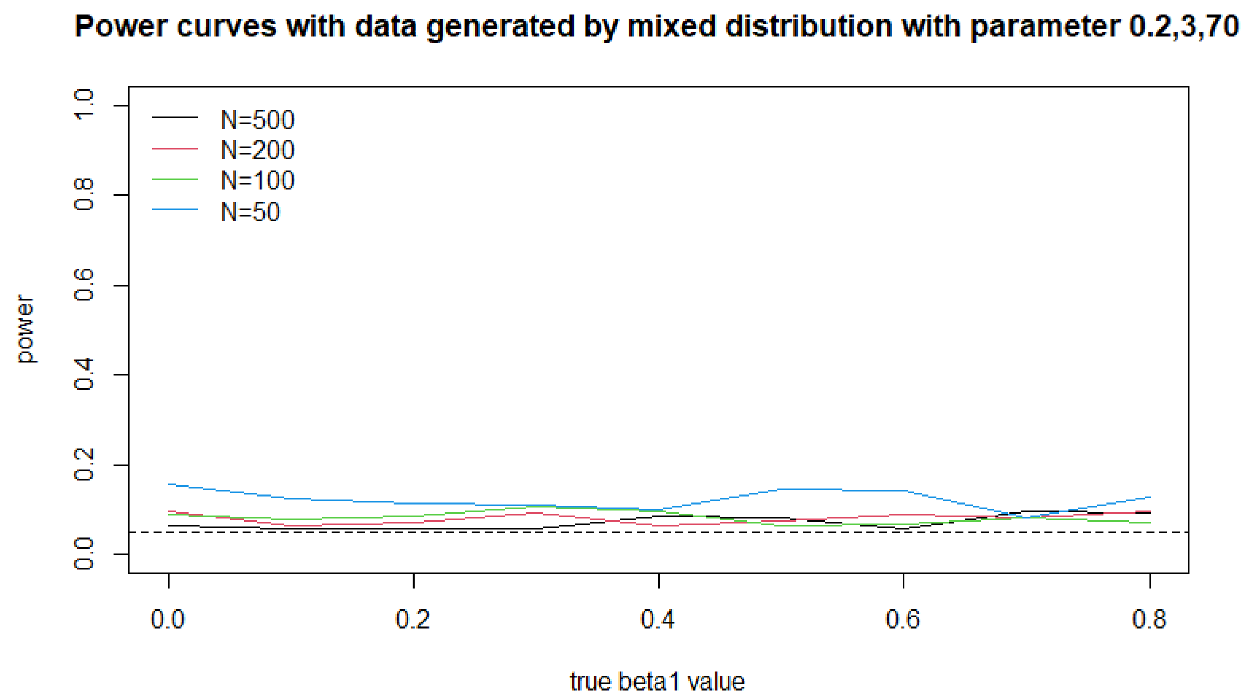
The following two figures display the power curve of test statistics generated by regression methods with quadratic and absolute loss to deal with data generated from mixed distribution with the parameter 0.2,3,20,



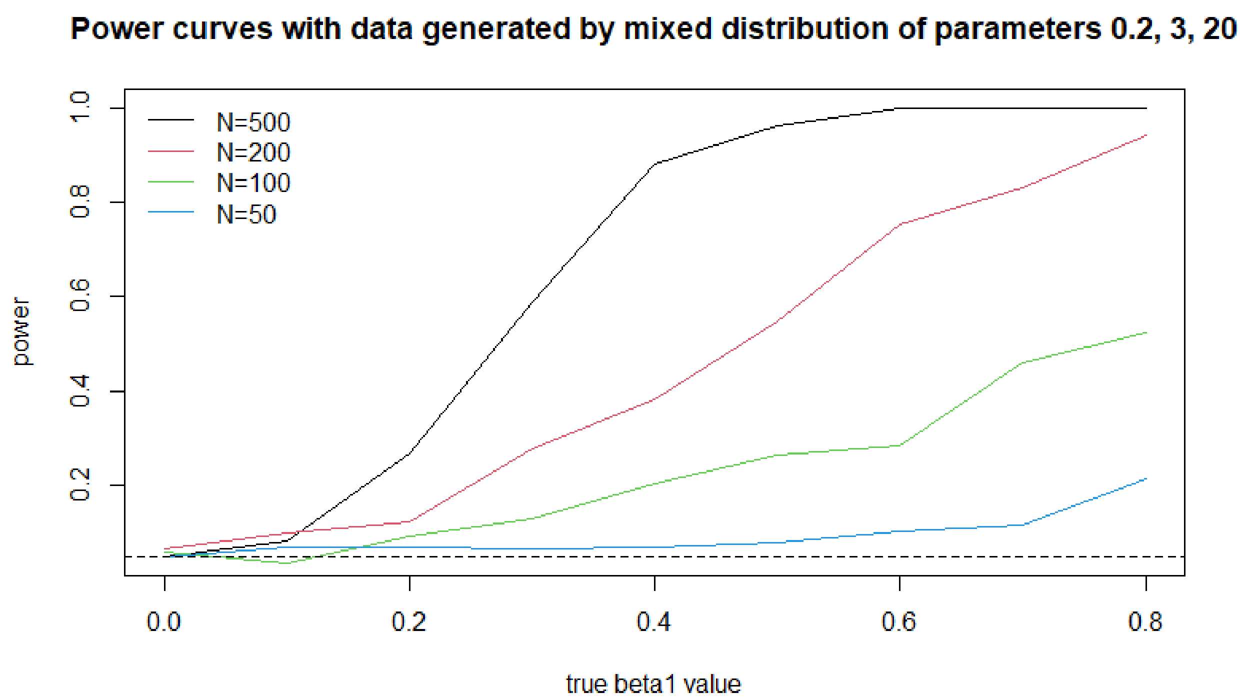
correspondingly.



The following two figures display the power curve of test statistics generated by regression methods with quadratic and absolute loss to deal with data generated from mixed distribution with the parameter 0.2,3,70,



correspondingly.



It can be seen from figures above that the test statistics generated by regression methods with quadratic loss loses its power gradually as the number and extend of ouliers increase while the test statistics generated by regression methods with absolute loss keeps its power with the increasing number and extent of outliers.

Discussion

In summary, we have verified the viability of the proposed test statistic for hypothesis testing of parameters in a Tukey 1-df interaction absolute regression model with a Huber loss.

The performance of the test statistic with the maximizer γ has shown great consistency across logistic regression, simple linear regression and absolute regression. In multiple simulation studies, γ that maximizes the test statistics has been proved close to true γ. The power of permutation of residues under reduced model has verified our original conjecture. It works better for large sample size such as N=200 or N=500 and works relatively poorly for smaller sample size such as N=50 or N=100.

In conclusion, the proposed method for logistic regression has shown a potential in the application of robust regression model, providing new horizons for testing genetic association in the presence of gene-gene and gene-environment interactions. Future work is needed to develop a more precise estimation of the test statistic for smaller sample size.

References

Chatterjee, N., Kalaylioglu, Z., Moslehi, R., Peters, U., & Wacholder, S. (2006). Powerful multilocus tests of genetic association in the presence of gene-gene and gene-environment interactions. The American Journal of Human Genetics, 79(6), 1002-1016.

Freedman, D., & Lane, D. (1983). A nonstochastic interpretation of reported significance levels. Journal of Business & Economic Statistics, 1(4), 292-298.