Package ‘riskPredictClustData’

November 28, 2018

Type Package
Title Assessing Risk Predictions for Clustered Data
Version 0.2.6
Date 2018-11-18
Maintainer Weiliang Qiu <weiliang.Qiu@gmail.com>
Depends R (>= 3.4.0)
Imports MASS, stats, gee, Hmisc, mvtnorm, utils
License GPL (>= 2)
NeedsCompilation no
Author Bernard Rosner [aut, ctb],
   Weiliang Qiu [aut, cre],
   Meiling T. Lee [aut, ctb]
Repository CRAN
Date/Publication 2018-11-28 08:50:03 UTC

R topics documented:

    genSimDataGLMEM ................................................. 2
    getScore ......................................................... 4
    powerCal ......................................................... 5
    powerCalData ..................................................... 7
    riskPredict ...................................................... 9
    riskPredictDiff ............................................... 11

Index 14
Generate simulated data from logistic mixed effects model based on the AMD data

Description

Generate simulated data from logistic mixed effects model based on the AMD data.

Usage

gensimdataglmemH

nsubj ] QSQL
betaP ] MVL
sd.betaPi ] Q.58L
betaQ ] Q.58L
beta2 I MS.95L
betaS I S.Q5L
beta4 ] 2.PVL
beta5 ] P.5QL
betaV I Q.4WL
betaW ] S.QQL
p.smkcur ] P.P8L
p.inieyeSQ ] P.44L
p.inieyeS2 ] P.42L
p.inieye4Q ] P.Q2L
p.inieye42 ] P.QQL
sd.lncalorc ] P.SSI

Arguments

nSubj integer. Number of subjects. Each subject would have data for 2 eyes.
beta0 mean of intercept $\beta_0$, which is assumed random and follows normal distribution $N(\beta_0, \sigma^2_{\beta})$
sd.beta0i standard deviation $\sigma^2_{\beta}$ of the random intercept $\beta_0$.
beta1 slope for the binary covariate cursmk (current smoking status). $cursmk = 1$ indicates current smokers. $cursmk = 0$ indicates past smokers or never smokers.
beta2 slope for the continuous mean-centered covariate lncalorc.
beta3 slope for the binary covariate inieye3 indicating if an eye of a subject has initial grade equal to 3. $inieye3 = 1$ indicates the eye has initial grade equal to 3.
beta4 slope for the binary covariate inieye4 indicating if an eye of a subject has initial grade equal to 4. $inieye4 = 1$ indicates the eye has initial grade equal to 4.
beta5 slope for the binary covariate rtotfat indicating if the subject’s total fat intake is in the 2nd quartile of total fat intake. $rtotfat = 1$ indicates the subject is in the 2nd quartile.
We generate simulated data set from the following generalized linear mixed effects model:

$$\log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_0 + \beta_1 \text{smkcur}_i + \beta_2 \text{lncalor}_ci + \beta_3 \text{inieye3}_ij + \beta_4 \text{inieye4}_ij + \beta_5 \text{rtotfat3}_i + \beta_6 \text{rtotfat2}_i + \beta_7 \text{rtotfat}_3i,$$

where $i = 1, \ldots, N, j = 1, 2$, $\beta_{0i} \sim N\left(\beta_0, \sigma^2_0\right)$.

Value

A data frame with 8 columns: cid, subuid, prog, smkcur, lncalorc, inieye3, inieye4, and rtotfat, where cid is the subject id, subuid is the unit id, and prog is the progression status. $\text{prog} = 1$ indicates the eye is progressed. $\text{prog} = 0$ indicates the eye is not progressed. There are $nsubj \times 2$ rows. The first $nsubj$ rows are for the left eyes and the second $nsubj$ rows are for the right eyes.

Author(s)

Bernard Rosner <stbar@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Meiling Ting Lee <MLTLEE@umd.edu>

References


Examples

```r
set.seed(1234567)
datFrame = genSimDataGLMEM(nSubj = 30, beta0 = -6, sd.beta0i = 1.58,
                           beta1 = 1.58, beta2 = -3.95, beta3 = 3.15, beta4 = 2.06,
                           beta5 = 0.51, beta6 = 1.47, beta7 = 3.11,
                           p.smkcur = 0.08, p.inieye31 = 0.44, p.inieye32 = 0.42,
                           p.inieye41 = 0.12, p.inieye42 = 0.11, sd.lncalorc = 0.33)
print(dim(datFrame))
print(datFrame[1:2,])
```
getScore

Get data frame for the function riskPredict

Description

Get data frame for the function riskPredict.

Usage

getScore(fmla, cidVar, subuidVar, statusVar, datFrame, mycorstr = "exchangeable", verbose = FALSE)

Arguments

fmla          A formula object for the function gee
_cidVar       character. Phenotype variable name for cluster id
_subuidVar    character. Phenotype variable name for unit id
_statusVar    character. Phenotype variable name for progression status
_datFrame     A data frame with at least 3 columns corresponding to _cid (indicated by _cidVar),
               _subuid (indicated by _subuidVar), _status (indicated by _statusID). _cid indicates cluster id; _subuid indicates unit ID within a cluster; _status=1 indicates an eye is progressed; _status=0 indicates an eye is not progressed.
_mycorstr     character. indicates correlation structure. see the manual for the function gee in the R library gee
_verbose      logical. indicating if summary of gee results should be printed out.

Value

A list with two elements: frame and gee.obj. frame is a data frame with at least 4 columns: _cid, _subuid, _status, and _score. _cid indicates cluster id; _subuid indicates unit ID within a cluster; _status=1 indicates an eye is progressed; _status=0 indicates an eye is not progressed; _score represents the risk score.

gee.obj is the object returned by gee function.

Author(s)

Bernard Rosner <stbar@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Meiling Ting Lee <MLTLEE@umd.edu>

References

Examples

```r
set.seed(1234567)
datFrame = genSimDataGLMEM(nSubj = 30, beta0 = -6, sd.beta0i = 1.58,
beta1 = 1.58, beta2 = -3.95, beta3 = 3.15, beta4 = 2.06,
beta5 = 0.51, beta6 = 1.47, beta7 = 3.11,
p.smkcur = 0.08, p.inieye31 = 0.44, p.inieye32 = 0.42,
p.inieye41 = 0.12, p.inieye42 = 0.11, sd.lncalorc = 0.33)

print(dim(datFrame))
print(datFrame[1:2,])

ttl = getScore(fmla = prog~smkcur+lncalorc+inieye3+inieye4+factor(rtotfat),
   cidVar = "cid", subuidVar = "subuid", statusVar = "prog",
   datFrame = datFrame, mycorstr = "exchangeable",
   verbose = FALSE)
myframe1=ttl$frame

gee.obj=ttl$gee.obj
print(summary(gee.obj))

print(dim(myframe1))
print(myframe1[1:3,])
```

powerCal  

*Calculate the power for testing $\delta = 0$*

Description

Calculate the power for testing $\delta = 0$.

Usage

```r
powerCal(
   nSubj, 
   mu1, 
   triangle, 
   rho, 
   rho11, 
   rho22, 
   rho12, 
   p11, 
   p10, 
   p01, 
   alpha = 0.05)
```
Arguments

nSubj integer. number of subjects to be generated. Assume each subject has two observations.

mu1 $\mu_1 = H(Y) - H(Y_c)$ is the difference between probit transformation $H(Y)$ and probit-shift alternative $H(Y_c)$, where $Y$ is the prediction score of a randomly selected progressing subunit, and $Y_c$ is the counterfactual random variable obtained if each subunit that had progressed actually had not progressed.

triangle the difference of the expected value the the extended Mann-Whitney U statistics between two prediction rules, i.e., $\Delta = \eta^{(1)}_c - \eta^{(2)}_c$

rho $\rho = corr \left( H(Z_{ij}), H(Z_{k\ell}) \right)$, where $H = \Phi^{-1}$ is the probit transformation.

rho11 $\rho_{11} = corr \left( H^{(1)}_{ij}, H^{(1)}_{i\ell} \right)$, where $H = \Phi^{-1}$ is the probit transformation.

rho22 $\rho_{22} = corr \left( H^{(2)}_{ij}, H^{(2)}_{i\ell} \right)$, where $H = \Phi^{-1}$ is the probit transformation.

rho12 $\rho_{12} = corr \left( H^{(1)}_{ij}, H^{(2)}_{i\ell} \right)$, where $H = \Phi^{-1}$ is the probit transformation.

p11 $p_{11} = Pr(\delta_{i1} = 1 \& \delta_{i2} = 1)$, where $\delta_{ij} = 1$ if the $j$-th subunit of the $i$-th cluster has progressed.

p10 $p_{10} = Pr(\delta_{i1} = 1 \& \delta_{i2} = 0)$, where $\delta_{ij} = 1$ if the $j$-th subunit of the $i$-th cluster has progressed.

p01 $p_{01} = Pr(\delta_{i1} = 0 \& \delta_{i2} = 1)$, where $\delta_{ij} = 1$ if the $j$-th subunit of the $i$-th cluster has progressed.

alpha type I error rate

Value

the power

Author(s)

Bernard Rosner <stbar@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Meiling Ting Lee <MLTLEE@umd.edu>

References


Examples

```r
set.seed(1234567)
mu1 = 0.8

power = powerCal(nSubj = 30, mu1 = mu1,
```
powerCalData

\[
\text{triangle} = 0.05, \ rho = 0.93, \ rho11 = 0.59, \ rho22 = 0.56, \ rho12 = 0.52,
\ p11 = 0.115, \ p10 = 0.142, \ p01 = 0.130, \ alpha = 0.05
\]

print(power)

---

**powerCalData**

*Calculate the power for testing $\delta = 0$ based on a dataset*

**Description**

Calculate the power for testing $\delta = 0$ based on a dataset.

**Usage**

```r
powerCalData(
  nSubj,  
  triangle,  
  frame,  
  alpha = 0.05)
```

**Arguments**

- `nSubj` integer. number of subjects to be generated. Assume each subject has two observations.
- `triangle` the difference of the expected value the the extended Mann-Whitney U statistics between two prediction rules, i.e., $\Delta = \eta^{(1)} - \eta^{(2)}$.
- `frame` A data frame with 5 columns: cid, subuid, status, score1, and score2. cid indicates cluster id; subuid indicates unit ID within a cluster; status=1 indicates an eye is progressed; status=0 indicates an eye is not progressed; score1 represents the score based on prediction rule 1. score2 represents the score based on prediction rule 2.
- `alpha` type I error rate

**Value**

A list with 11 elements.

- `power` the esstimated power
- `rho` $\rho = \text{corr}(H(Z_{ij}), H(Z_{k\ell}))$, where $H = \Phi^{-1}$ is the probit transformation.
- `rho11` $\rho_{11} = \text{corr}(H^{(1)}_{ij}, H^{(1)}_{i\ell})$, where $H = \Phi^{-1}$ is the probit transformation.
- `rho22` $\rho_{22} = \text{corr}(H^{(2)}_{ij}, H^{(2)}_{i\ell})$, where $H = \Phi^{-1}$ is the probit transformation.
- `rho12` $\rho_{12} = \text{corr}(H^{(1)}_{ij}, H^{(2)}_{i\ell})$, where $H = \Phi^{-1}$ is the probit transformation.
\[ p_{11} = Pr(\delta_{i1} = 1 & \delta_{i2} = 1), \] where \( \delta_{ij} = 1 \) if the \( j \)-th subunit of the \( i \)-th cluster has progressed.

\[ p_{10} = Pr(\delta_{i1} = 1 & \delta_{i2} = 0), \] where \( \delta_{ij} = 1 \) if the \( j \)-th subunit of the \( i \)-th cluster has progressed.

\[ p_{01} = Pr(\delta_{i1} = 0 & \delta_{i2} = 1), \] where \( \delta_{ij} = 1 \) if the \( j \)-th subunit of the \( i \)-th cluster has progressed.

\[ p_{00} = Pr(\delta_{i1} = 0 & \delta_{i2} = 0), \] where \( \delta_{ij} = 1 \) if the \( j \)-th subunit of the \( i \)-th cluster has progressed.

\[ \mu_1 = H(Y) - H(Y_c) \] is the difference between probit transformation \( H(Y) \) and probit-shift alternative \( H(Y_c) \) for the first prediction score, where \( Y \) is the prediction score of a randomly selected progressing subunit, and \( Y_c \) is the counterfactual random variable obtained if each subunit that had progressed actually had not progressed.

\[ \mu_2 = H(Y) - H(Y_c) \] is the difference between probit transformation \( H(Y) \) and probit-shift alternative \( H(Y_c) \) for the second prediction score, where \( Y \) is the prediction score of a randomly selected progressing subunit, and \( Y_c \) is the counterfactual random variable obtained if each subunit that had progressed actually had not progressed.

**Author(s)**

Bernard Rosner <stbar@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Meiling Ting Lee <MLTLEE@umd.edu>

**References**


**Examples**

```r
set.seed(1234567)

datFrame = genSimDataGLMEM(nSubj = 30, beta0 = -6, sd.beta0i = 1.58,
                          beta1 = 1.58, beta2 = -3.95, beta3 = 3.15, beta4 = 2.06,
                          beta5 = 0.51, beta6 = 1.47, beta7 = 3.11,
                          p.smkcur = 0.08, p.inieye31 = 0.44, p.inieye32 = 0.42,
                          p.inieye41 = 0.12, p.inieye42 = 0.11, sd.lncalorc = 0.33)

print(dim(datFrame))
print(datFrame[1:2,])

# prediction rule 1
ttl1 = getScore(fmla = prog~smkcur+lncalorc+inieye3+inieye4+factor(rtotfat),
                 cidVar = "cid", subuidVar = "subuid", statusVar = "prog",
                 datframe = datFrame, mycorstr = "exchangeable",
                 verbose = FALSE)
```
**riskPredict**

Assessing risk prediction performance for clustered data.

### Description

Assessing risk prediction performance for clustered data.

### Usage

```r
riskPredict(frame, alpha=0.05)
```

### Arguments

- **frame**: A data frame with 4 columns: cid, subuid, status, and score. cid indicates cluster id; subuid indicates unit ID within a cluster; status=1 indicates an eye is progressed; status=0 indicates an eye is not progressed; score represents the risk score.
- **alpha**: numeric. confidence level for $\eta_c$. 

```r
going to frame
myframe1=tt$frame
print(dim(myframe1))
print(myframe1[1:3,])

###
# prediction rule 2
if2 = getScore(formula = prog~smkcur+lnalorc+inieye3+inieye4,
cidVar = "cid", subuidVar = "subuid", statusVar = "prog",
datFrame = datFrame, mycorstr = "exchangeable",
verbose = FALSE)
myframe2=tt2$frame
print(dim(myframe2))
print(myframe2[1:3,])

# combine scores from two prediction rules
myframe12=myframe1[, c("cid", "subuid", "status")]
myframe12$score1=myframe1$score
myframe12$score2=myframe2$score
print(dim(myframe12))
print(myframe12[1:3,])

res = powerCalData(nSubj = 30, triangle = 0.05, frame=myframe12, alpha = 0.05)
print(res)
```
Details

To obtain 95% confidence interval of \( \eta_c \), we first obtain 95% confidence interval \([c_1, c_2]\) for \( \Phi^{-1}(\eta_c) \), then transform back: \( [\Phi(c_1), \Phi(c_2)] \).

Value

A list of 6 elements:

- **stat**: the test statistics \( \hat{\eta}_c^{(1)} \) based on the prediction rule.
- **se.stat**: standard error of the test statistic under the null hypothesis.
- **z**: z score \( z=\text{stat} - 0.5)/\text{se.stat} \)
- **pval**: p-value of the test
- **rho**: correlation between \( H(Z_{ij}) \) and \( H(Z_{i\ell}) \)
- **mu.hat**: estimated \( \mu \).
- **theta.hat**: estimated \( \theta \).
- **theta.c.hat**: estimated \( \theta_c \).
- **E.stat.Ha**: expectation of \( \hat{\eta}_c \) under the alternative hypothesis.
- **se.stat.Ha**: standard error for \( \hat{\eta}_c \) under the alternative hypothesis.
- **CIlow**: lower confidence limit for \( \eta_c \).
- **CIupp**: upper confidence limit for \( \eta_c \).
- **datHk**: A \( n \text{Subj} \times 2 \) matrix of probit transformed risk scores by using only the first 2 observations of each subject.
- **ci**: the vector of \( c_i \), the number of progressing subunits for the \( i \)-th subject.
- **di**: the vector of \( d_i \), the number of non-progressing subunits for the \( i \)-th subject.

Author(s)

Bernard Rosner <stbar@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Meiling Ting Lee <MLTLEE@umd.edu>

References


Examples

```r
set.seed(1234567)
datFrame = genSimDataGLMEM(nSubj = 30, beta0 = -6, sd.beta0i = 1.58, beta1 = 1.58, beta2 = -3.95, beta3 = 3.15, beta4 = 2.06, beta5 = 0.51, beta6 = 1.47, beta7 = 3.11, p.smkcur = 0.08, p.inieye31 = 0.44, p.inieye32 = 0.42, p.inieye41 = 0.12, p.inieye42 = 0.11, sd.lncalorc = 0.33)
```
```
print(dim(datFrame))
print(dim(myframe1))
print(dim(myframe1[1:3,]))

res1 = riskPredict(myframe1)
print(names(res1))
print(res1)
```

---

**riskPredictDiff**

*Difference of two risk prediction rules for clustered data*

**Description**

Difference of two risk prediction rules for clustered data.

**Usage**

```r
riskPredictDiff(frame, alpha = 0.05)
```

**Arguments**

- `frame`: A data frame with 5 columns: `cid`, `subuid`, `status`, `score1`, and `score2`. `cid` indicates cluster id; `subuid` indicates unit ID within a cluster; `status=1` indicates an eye is progressed; `status=0` indicates an eye is not progressed; `score1` represents the score based on prediction rule 1. `score2` represents the score based on prediction rule 2.
- `alpha`: numeric. The confidence level.

**Value**

A list of 7 elements:

- `diff`: the difference of test statistics \( \hat{\eta}_c^{(1)} - \hat{\eta}_c^{(2)} \) based on the 2 prediction rules.
- `se.diff`: standard error of the difference under the null hypothesis.
- `z`: z score \( z = \text{diff} / \text{se.diff} \)
- `pval`: p-value of the test
res1  output object of the function riskPredict for prediction rule 1.
res2  output object of the function riskPredict for prediction rule 2.
rhocov  A vector of 4 correlations: $\rho = \text{cov}(H^{(1)}_{ij}, H^{(2)}_{ij}), \rho_{11} = \text{cov}(H^{(1)}_{ij}, H^{(1)}_{it}), \rho_{22} = \text{cov}(H^{(2)}_{ij}, H^{(2)}_{it}),$ and $\rho_{12} = \text{cov}(H^{(1)}_{ij}, H^{(2)}_{it})$
E.diff.Ha  expectation of the difference under the alternative hypothesis.
se.diff.Ha  standard error of the difference under the alternative hypothesis.
CIlow.diff  Lower confidence limit.
CIup.diff  Upper confidence limit.

Author(s)
Bernard Rosner <stbar@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Meiling Ting Lee <MLTLEE@umd.edu>

References

Examples

```r
set.seed(1234567)
datFrame = genSimDataGLMEM(nSubj = 30, beta0 = -6, sd.beta0i = 1.58,  
  beta1 = 1.58, beta2 = -3.95, beta3 = 3.15, beta4 = 2.06,  
  beta5 = 0.51, beta6 = 1.47, beta7 = 3.11,  
  p.smkcur = 0.08, p.inieye31 = 0.44, p.inieye32 = 0.42,  
  p.inieye41 = 0.12, p.inieye42 = 0.11, sd.lncalorc = 0.33)

print(dim(datFrame))
print(datFrame[1:2,])

# prediction rule 1
tt1 = getScore(formula = prog~smkcur+lncalorc+inieye3+inieye4+factor(rtotfat),  
  cidVar = "cid", subuidVar = "subuid", statusVar = "prog",  
  datFrame = datFrame, mycorstr = "exchangeable",  
  verbose = FALSE)
myframe1=tt1$frame

print(dim(myframe1))
print(myframe1[1:3,])

###
# prediction rule 2
tt2 = getScore(formula = prog~smkcur+lncalorc+inieye3+inieye4,  
  cidVar = "cid", subuidVar = "subuid", statusVar = "prog",  
  datFrame = datFrame, mycorstr = "exchangeable",
```
```r
riskPredictDiff

verbose = FALSE
myframe2 = tt2$frame

print(dim(myframe2))
print(myframe2[1:3,])

# combine scores from two prediction rules
myframe12 = myframe1[, c("cid", "subuid", "status")]
myframe12$score1 = myframe1$score
myframe12$score2 = myframe2$score
print(dim(myframe12))
print(myframe12[1:3,])

####
resDiff = riskPredictDiff(frame = myframe12)
print(names(resDiff))
print(resDiff)
```
Index

*Topic **method**
   gensimDataGLMEM, 2
   getScore, 4
   powerCal, 5
   powerCalData, 7
   riskPredict, 9
   riskPredictDiff, 11

genSimDataGLMEM, 2
getScore, 4

powerCal, 5
powerCalData, 7

riskPredict, 9
riskPredictDiff, 11