

# Package ‘PoisBinOrd’

October 12, 2022

**Type** Package

**Title** Data Generation with Poisson, Binary and Ordinal Components

**Version** 1.4.3

**Date** 2021-03-21

**Author** Gul Inan, Hakan Demirtas, Ran Gao

**Maintainer** Ran Gao <rgao8@ui.c.edu>

**Description** Generation of multiple count, binary and ordinal variables simultaneously given the marginal characteristics and association structure. Throughout the package, the word 'Poisson' is used to imply count data under the assumption of Poisson distribution. The details of the method are explained in Amatya, A. and Demirtas, H. (2015) <[DOI:10.1080/00949655.2014.953534](https://doi.org/10.1080/00949655.2014.953534)>.

**License** GPL-2 | GPL-3

**Depends** corpcor, GenOrd, Matrix, mvtnorm

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-03-22 05:30:02 UTC

## R topics documented:

PoisBinOrd-package . . . . .	2
correlation.bound.check . . . . .	3
correlation.limits . . . . .	4
gen.PoisBinOrd . . . . .	6
intermediate.corr.BO . . . . .	7
intermediate.corr.PBO . . . . .	9
intermediate.corr.PP . . . . .	10
overall.corr.mat . . . . .	11
validation.bin . . . . .	12
validation.corr . . . . .	13
validation.ord . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

## Description

Provides R functions for generation of multiple count, binary and ordinal variables simultaneously given the marginal characteristics and association structure.

## Details

Package: PoisBinOrd  
Type: Package  
Version: 1.4.3  
Date: 2021-03-21  
License: GPL-2 | GPL-3

PoisBinOrd package consists of ten functions. The functions `validation.bin`, `validation.ord`, and `validation.corr` validate the specified quantities to prevent users from committing obvious specification errors. `correlation.limits` returns the lower and upper bounds of the pairwise correlation of Poisson-Poisson, Poisson-binary, Poisson-ordinal, binary-binary, binary-ordinal, and ordinal-ordinal combinations given their marginal distributions, i.e. returns the range of feasible pairwise correlations. The function `correlation.bound.check` checks the validity of the values of pairwise correlations. The functions `intermediate.corr.PP`, `intermediate.corr.BO`, and `intermediate.corr.PBO` computes intermediate correlation matrix for Poisson-Poisson combinations, binary/ordinal and binary/ordinal combinations, and Poisson and binary/ordinal combinations, respectively. The function `overall.corr.mat` assembles the final correlation matrix. The engine function `gen.PoisBinOrd` generates mixed data in accordance with the specified marginal and correlational quantities. Throughout the package, variables are supposed to be inputted in a certain order, namely, first count variables, next binary variables, and then ordinal variables should be placed.

## Author(s)

Gul Inan, Hakan Demirtas, Ran Gao  
Maintainer: Ran Gao <rgao8@uic.edu>

## References

- Amatya, A. and Demirtas, H. (2015). Simultaneous generation of multivariate mixed data with Poisson and normal marginals. *Journal of Statistical Computation and Simulation*, (85)15, 3129-3139.
- Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, 65(2), 104-109.
- Demirtas, H., Hedeker, D., and Mermelstein, R.J. (2012). Simulation of massive public health data by power polynomials. *Statistics in Medicine*, 31(27), 3337-3346.

Ferrari, P.A. and Barbiero, A. (2012). Simulating ordinal data. *Multivariate Behavioral Research*, 47(4), 566-589.

---

correlation.bound.check

*Checks if the pairwise correlation among variables are within the feasible range*

---

### Description

This function checks if there are range violations among correlation of Poisson-Poisson, Poisson-binary, Poisson-ordinal, binary-binary, binary-ordinal, and ordinal-ordinal combinations.

### Usage

```
correlation.bound.check(n.P, n.B, n.O, lambda.vec = NULL, prop.vec = NULL,  
prop.list = NULL, corr.vec = NULL, corr.mat = NULL)
```

### Arguments

n.P	Number of Poisson variables.
n.B	Number of binary variables.
n.O	Number of ordinal variables.
lambda.vec	Rate vector for Poisson variables.
prop.vec	Probability vector for binary variables.
prop.list	A list of probability vectors for ordinal variables.
corr.vec	Vector of elements below the diagonal of correlation matrix ordered column-wise.
corr.mat	Specified correlation matrix.

### Value

The function returns TRUE if no specification problem is encountered. Otherwise, it returns an error message.

### References

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, 65(2), 104-109.

Demirtas, H., Hedeker, D., and Mermelstein, R.J. (2012). Simulation of massive public health data by power polynomials. *Statistics in Medicine*, 31(27), 3337-3346.

### See Also

[validation.corr](#), [correlation.limits](#)

**Examples**

```
## Not run:
n.P<-1
n.B<-1
n.O<-1
lambda.vec<-c(1)
prop.vec<-c(0.3)
prop.list<-list(c(0.3,0.6))
corr.mat=matrix(c(1,0.2,0.1,0.2,1,0.5,0.1,0.5,1),3,3)
correlation.bound.check(n.P,n.B,n.O,lambda.vec,prop.vec,prop.list,corr.vec=NULL,
corr.mat)

n.P<-2
n.B<-2
n.O<-2
lambda.vec<-c(1,2)
prop.vec<-c(0.3,0.5)
prop.list<-list(c(0.3,0.6),c(0.5,0.6))
corr.mat=matrix(0.8,6,6)
diag(corr.mat)=1
correlation.bound.check(n.P,n.B,n.O,lambda.vec,prop.vec,prop.list,corr.vec=NULL,
corr.mat)

## End(Not run)
```

---

correlation.limits	<i>Computes lower and upper correlation bounds for each pair of variables</i>
--------------------	---

---

**Description**

This function computes lower and upper limits for pairwise correlations of Poisson-Poisson, Poisson-binary, Poisson-ordinal, binary-binary, binary-ordinal, and ordinal-ordinal combinations.

**Usage**

```
correlation.limits(n.P, n.B, n.O, lambda.vec = NULL, prop.vec = NULL,
prop.list = NULL)
```

**Arguments**

n.P	Number of Poisson variables.
n.B	Number of binary variables.
n.O	Number of ordinal variables.
lambda.vec	Rate vector for Poisson variables.
prop.vec	Probability vector for binary variables.
prop.list	A list of probability vectors for ordinal variables.

**Details**

While the function computes the exact lower and upper bounds for pairwise correlations among binary-binary variables as formulated in Demirtas et al. (2012), it computes approximate lower and upper bounds for pairwise correlations among Poisson-Poisson, Poisson-binary, Poisson-ordinal, binary-ordinal, and ordinal-ordinal variables through the method suggested by Demirtas and Hedeker (2011).

**Value**

The function returns a matrix of size  $(n.P + n.B + n.O) * (n.P + n.B + n.O)$ , where the lower triangular part of the matrix contains the lower bounds and the upper triangular part of the matrix contains the upper bounds of the feasible correlations.

**References**

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, 65(2), 104-109.

Demirtas, H., Hedeker, D., and Mermelstein, R.J. (2012). Simulation of massive public health data by power polynomials. *Statistics in Medicine*, 31(27), 3337-3346.

**See Also**

[validation.corr](#), [correlation.bound.check](#)

**Examples**

```
## Not run:
n.P<-3
n.B<-2
n.O<-3
lambda.vec<-c(1,2,3)
prop.vec<-c(0.3,0.5)
prop.list<-list(c(0.3,0.6),c(0.25,0.5,0.75),c(0.1,0.2,0.8,0.9))

#Correlation limits among Poisson variables
correlation.limits(n.P,n.B=0,n.O=0,lambda.vec,prop.vec=NULL,prop.list=NULL)

#See also Cor.PP.Limit in R package PoisNor

#Correlation limits among binary variables
correlation.limits(n.P=0,n.B,n.O=0,lambda.vec=NULL,prop.vec,prop.list=NULL)

#See also correlation.limits in R package BinNonNor

#Correlation limits among ordinal variables
correlation.limits(n.P=0,n.B=0,n.O,lambda.vec=NULL,prop.vec=NULL,prop.list)

#See also Limit_forOO in R package OrdNor

#Correlation limits among Poisson and binary variables and within themselves.
correlation.limits(n.P,n.B,n.O=0,lambda.vec,prop.vec,prop.list=NULL)
```

```

#Correlation limits among Poisson and ordinal variables and within themselves.
correlation.limits(n.P,n.B=0,n.O,lambd.vec,prop.vec=NULL,prop.list)

#Correlation limits among binary and ordinal variables and within themselves.
correlation.limits(n.P=0,n.B,n.O,lambd.vec=NULL,prop.vec,prop.list)

#Correlation limits among Poisson, binary, and ordinal variables and within themselves.
correlation.limits(n.P,n.B,n.O,lambd.vec,prop.vec,prop.list)

n.P<-2
lambd.vec=c(-1,1)
correlation.limits(n.P,n.B=0,n.O=0,lambd.vec,prop.vec=NULL,prop.list=NULL)

## End(Not run)

```

---

gen.PoisBinOrd	<i>Simulates a sample of size n from a set of multivariate Poisson, binary, and ordinal variables</i>
----------------	---

---

## Description

This function simulates a sample of size  $n$  from a set of multivariate Poisson, binary, and ordinal data with pre-specified marginals and a correlation matrix.

## Usage

```
gen.PoisBinOrd(n, n.P, n.B, n.O, lambda.vec = NULL, prop.vec = NULL,
prop.list = NULL, final.corr.mat)
```

## Arguments

n	Number of variates.
n.P	Number of Poisson variables.
n.B	Number of binary variables.
n.O	Number of ordinal variables.
lambda.vec	Rate vector for Poisson variables.
prop.vec	Probability vector for binary variables.
prop.list	A list of probability vectors for ordinal variables.
final.corr.mat	Final correlation matrix produced from <a href="#">overall.corr.mat</a>

## Value

A matrix of size  $n \times (n.P + n.B + n.O)$ , of which the first  $n.P$  columns are Poisson variables, the next  $n.B$  columns are binary variables, and the last  $n.O$  columns are ordinal variables.

**Examples**

```
## Not run:
n=10000
n.P<-2
n.B<-2
n.O<-2
lambda.vec<-sample(10,2)
prop.vec<-runif(2)
prop.list<-list(c(0.3,0.6,0.7),c(0.2,0.3,0.5))
M=c(-0.05, 0.26, 0.14, 0.09, 0.14, 0.12, 0.13, -0.02, 0.17, 0.11,
-0.04, 0.19, 0.10, 0.35, 0.39)
N=diag(6)
N[lower.tri(N)]=M
corr.mat=N+t(N)
diag(corr.mat)<-1
final.corr.mat=overall.corr.mat(n.P,n.B,n.O,lambda.vec,prop.vec,
prop.list,corr.vec=NULL,corr.mat)
mymixdata=gen.PoisBinOrd(n,n.P,n.B,n.O,lambda.vec,prop.vec,prop.list,
final.corr.mat)

## End(Not run)
```

---

`intermediate.corr.BO` *Computes an intermediate normal correlation matrix for any combination of binary and ordinal variables given the specified correlation matrix*

---

**Description**

Computes an intermediate normal correlation matrix for any combination of binary and ordinal variables before dichotomization/ordinalization given the specified correlation matrix as formulated in Ferrari and Barbiero (2012).

**Usage**

```
intermediate.corr.BO(n.B, n.O, prop.vec = NULL, prop.list = NULL, corr.vec = NULL,
corr.mat = NULL)
```

**Arguments**

<code>n.B</code>	Number of binary variables.
<code>n.O</code>	Number of ordinal variables.
<code>prop.vec</code>	Probability vector for binary variables.
<code>prop.list</code>	A list of probability vectors for ordinal variables.
<code>corr.vec</code>	Vector of elements below the diagonal of correlation matrix ordered column-wise.
<code>corr.mat</code>	Specified correlation matrix.

**Value**

A correlation matrix of size  $(n.B+n.O)*(n.B+n.O)$ .

**References**

Ferrari, P.A. and Barbiero, A. (2012). Simulating ordinal data. *Multivariate Behavioral Research*, 47(4), 566-589.

Barbiero, A. and Ferrari, P.A. (2015). GenOrd: Simulation of ordinal and discrete variables with given correlation matrix and marginal distributions. <http://www.cran.r-project.org/web/packages/GenOrd>.

**See Also**

[intermediate.corr.PBO](#)

**Examples**

```
## Not run:
n.B=1
n.O=2
prop.vec=0.7
prop.list=list(cumsum(c(0.30, 0.40)), cumsum(c(0.4, 0.2, 0.3)))
corr.mat=matrix ( c(
1.0000000, 0.1767231, 0.3006186,
0.1767231, 1.0000000, -0.139923,
0.3006186, -0.1399230, 1.0000000),3,3)
intmatB0=intermediate.corr.BO(n.B,n.O,prop.vec,prop.list,corr.vec=NULL,
corr.mat)

n.B=1
n.O=1
prop.vec=-c(0.3)
prop.list=-list(c(0.3,0.6))
corr.mat=matrix(c(1,0.2,0.1,0.2,1,0.5,0.1,0.5,1),3,3)
intmatB0=intermediate.corr.BO(n.B,n.O,prop.vec,prop.list,corr.vec=NULL,
corr.mat)

n.B=2
prop.vec=c(0.4,0.7)
corr.mat=matrix(c(1,-0.3,-0.3,1),2,2)
intmatBB=intermediate.corr.BO(n.B,n.O=0,prop.vec,prop.list=NULL,corr.vec=NULL,
corr.mat)

#See Tetra.Corr.BB in R package BinNonNor
#Tetra.Corr.BB(n.BB=2,prop.vec=c(0.4,0.7),corr.vec=NULL,corr.mat=corr.mat)

n.B=0
n.O=2
prop.list=list(cumsum(c(0.30, 0.40)), cumsum(c(0.4,0.2,0.3)))
corr.mat=matrix(c(1.0000000, -0.139923,-0.139923,1.0000000),2,2)
intmatO0=intermediate.corr.BO(n.B,n.O,prop.vec=NULL,prop.list,corr.vec=NULL,
corr.mat)
```



```
#See Intermediate00(plist, OOCorrMat) in R package OrdNor
#Intermediate00(plist=list(cumsum(c(0.30,0.40)),cumsum(c(0.4,0.2,0.3))),
OOCCorrMat=corr.mat)

## End(Not run)
```

---

`intermediate.corr.PBO` *Computes the pairwise entries of the intermediate normal correlation matrix for all Poisson-binary and Poisson-ordinal variable combinations given the specified correlation matrix.*

---

### Description

This function computes the pairwise entries of the intermediate normal correlation matrix for all Poisson-binary and Poisson-ordinal variable combinations given the specified correlation matrix as formulated in Amatya and Demirtas (2015).

### Usage

```
intermediate.corr.PBO(n.P, n.B, n.O, lambda.vec = NULL, prop.vec = NULL,
prop.list = NULL, corr.vec = NULL, corr.mat = NULL)
```

### Arguments

<code>n.P</code>	Number of Poisson variables.
<code>n.B</code>	Number of binary variables.
<code>n.O</code>	Number of ordinal variables.
<code>lambda.vec</code>	Rate vector for Poisson variables.
<code>prop.vec</code>	Probability vector for binary variables.
<code>prop.list</code>	A list of probability vectors for ordinal variables.
<code>corr.vec</code>	Vector of elements below the diagonal of correlation matrix ordered column-wise.
<code>corr.mat</code>	Specified correlation matrix.

### Value

A matrix of  $n.P \times (n.B + n.O)$

### References

Amatya, A. and Demirtas, H. (2015). Simultaneous generation of multivariate mixed data with Poisson and normal marginals. *Journal of Statistical Computation and Simulation*, 85(15), 3129-3139.

**See Also**

[intermediate.corr.PP](#), [intermediate.corr.B0](#)

**Examples**

```
## Not run:
n.P<-1
n.B<-1
n.0<-1
lambda.vec<-c(1)
prop.vec<-c(0.3)
prop.list<-list(c(0.3,0.6))
corr.mat=matrix(c(1,0.2,0.1,0.2,1,0.5,0.1,0.5,1),3,3)

intmatPB0=intermediate.corr.PB0(n.P,n.B,n.0,lambda.vec,prop.vec,prop.list,
corr.vec=NULL,corr.mat)

## End(Not run)
```

---

`intermediate.corr.PP` *Computes an intermediate normal correlation matrix for Poisson variables given the specified correlation matrix*

---

**Description**

This function computes the intermediate normal correlation matrix for Poisson-Poisson combinations before inverse cdf matching as formulated in Amatya and Demirtas (2015).

**Usage**

```
intermediate.corr.PP(n.P, lambda.vec, corr.vec = NULL, corr.mat = NULL)
```

**Arguments**

<code>n.P</code>	Number of Poisson variables.
<code>lambda.vec</code>	Rate vector for Poisson variables.
<code>corr.vec</code>	Vector of elements below the diagonal of correlation matrix ordered column-wise.
<code>corr.mat</code>	Specified correlation matrix.

**Value**

A correlation matrix of size  $n.P*n.P$

**References**

Amatya, A. and Demirtas, H. (2015). Simultaneous generation of multivariate mixed data with Poisson and normal marginals. *Journal of Statistical Computation and Simulation*, 85(15), 3129-3139.

**See Also**

[intermediate.corr.PBO](#)

**Examples**

```
n.P<-3
lambda.vec<-c(1,2,3)
corr.mat<-matrix(c(1,0.352,0.265,0.352,1,0.121,0.265,0.121,1),n.P,n.P)
intmatPP=intermediate.corr.PP(n.P,lambda.vec,corr.vec=NULL,corr.mat)

## Not run:
#See also cmat.star in R package PoisNor
#cmat.star(no.pois=3,no.norm=0,corMat=corr.mat,lamvec=lambda.vec)

## End(Not run)
```

---

overall.corr.mat

*Computes the final intermediate correlation matrix*

---

**Description**

This function computes the final correlation matrix by combining pairwise intermediate correlation matrix entries for Poisson-Poisson, Poisson-binary, Poisson-ordinal, binary-binary, binary-ordinal, and ordinal-ordinal combinations. If the resulting correlation matrix is not positive definite, a nearest positive matrix will be used.

**Usage**

```
overall.corr.mat(n.P, n.B, n.O, lambda.vec = NULL, prop.vec = NULL, prop.list = NULL,
corr.vec = NULL, corr.mat = NULL)
```

**Arguments**

n.P	Number of Poisson variables.
n.B	Number of binary variables.
n.O	Number of ordinal variables.
lambda.vec	Rate vector for Poisson variables.
prop.vec	Probability vector for binary variables.
prop.list	A list of probability vectors for ordinal variables.
corr.vec	Vector of elements below the diagonal of correlation matrix ordered column-wise.
corr.mat	Specified correlation matrix.

**Value**

A matrix of size  $(n.P+n.B+n.O)*(n.P+n.B+n.O)$

**See Also**

[intermediate.corr.PP](#), [intermediate.corr.B0](#), [intermediate.corr.PB0](#)

**Examples**

```
## Not run:
n.P<-1
n.B<-1
n.O<-1
lambda.vec<-c(1)
prop.vec<-c(0.3)
prop.list<-list(c(0.3,0.6))
corr.vec=NULL
corr.mat=matrix(c(1,0.2,0.1,0.2,1,0.5,0.1,0.5,1),3,3)
finalmat=overall.corr.mat(n.P,n.B,n.O,lambda.vec,prop.vec,prop.list,corr.vec=NULL,
corr.mat)

## End(Not run)
```

---

validation.bin

*Validates the marginal specification of the binary variables*

---

**Description**

Checks whether the marginal specification of the binary part is valid and consistent.

**Usage**

```
validation.bin(n.B, prop.vec = NULL)
```

**Arguments**

n.B	Number of binary variables.
prop.vec	Probability vector for binary variables.

**Value**

The function returns TRUE if no specification problem is encountered. Otherwise, it returns an error message.

**Examples**

```
n.B<-3
prop.vec<-c(0.25,0.5,0.75)
validation.bin(n.B,prop.vec)

## Not run:
n.B<-3
validation.bin(n.B)
```

```
n.B<--3
prop.vec<-c(0.25,0.5,0.75)
validation.bin(n.B,prop.vec)

n.B<-0
prop.vec<-c(0.25,0.5,0.75)
validation.bin(n.B,prop.vec)

n.B<-5
prop.vec<-c(0.25,0.5,0.75)
validation.bin(n.B,prop.vec)

n.B<-3
prop.vec<-c(0.25,0.5,-0.75)
validation.bin(n.B,prop.vec)

## End(Not run)
```

---

validation.corr	<i>Validates the specified correlation matrix</i>
-----------------	---

---

## Description

This function validates the specified correlation vector and/or matrix for appropriate dimension, symmetry, range, and positive definiteness. If both correlation matrix and correlation vector are supplied, it checks whether the matrix and vector are conformable.

## Usage

```
validation.corr(n.P, n.B, n.O, corr.vec = NULL, corr.mat = NULL)
```

## Arguments

n.P	Number of Poisson variables.
n.B	Number of binary variables.
n.O	Number of ordinal variables.
corr.vec	Vector of elements below the diagonal of correlation matrix ordered column-wise.
corr.mat	Specified correlation matrix.

## Value

The function returns TRUE if no specification problem is encountered. Otherwise, it returns an error message.

## See Also

[correlation.limits](#), [correlation.bound.check](#)

**Examples**

```
n.P<-1
n.B<-1
n.O<-1
corr.vec=c(0.2,0.1,0.5)
validation.corr(n.P,n.B,n.O,corr.vec,corr.mat=NULL)
```

```
n.P<-2
n.B<-2
n.O<-2
corr.mat=matrix(0.5,6,6)
diag(corr.mat)=1
validation.corr(n.P,n.B,n.O,corr.vec=NULL,corr.mat)
```

---

validation.ord

*Validates the marginal specification of the ordinal variables*


---

**Description**

Checks whether the marginal specification of the ordinal part is valid and consistent.

**Usage**

```
validation.ord(n.O, prop.list = NULL)
```

**Arguments**

n.O	Number of ordinal variables.
prop.list	A list of probability vectors corresponding to each ordinal variable. The i-th element of prop.list is a vector of the cumulative probabilities defining the marginal distribution of the i-th ordinal component of the multivariate variables. If the i-th ordinal variable has k categories, the i-th vector of the prop.list will contain k-1 probability values. The k-th element is implicitly 1.

**Value**

The function returns TRUE if no specification problem is encountered. Otherwise, it returns an error message.

**Examples**

```
n.O<-3
prop.list<-list(c(0.3,0.6),c(0.25,0.5,0.75),c(0.1,0.2,0.8,0.9))
validation.ord(n.O,prop.list)
```

```
## Not run:
n.O<-3
validation.ord(n.O)
```

```
n.0<-NULL
prop.list<-list(c(0.3,0.6),c(0.25,0.5,0.75),c(0.1,0.2,0.8,0.9))
validation.ord(prop.list=prop.list)

n.0<--3
prop.list<-list(c(0.3,0.6),c(0.25,0.5,0.75),c(0.1,0.2,0.8,0.9))
validation.ord(-3,prop.list)

n.0<--0
prop.list<-list(c(0.3,0.6),c(0.25,0.5,0.75),c(0.1,0.2,0.8,0.9))
validation.ord(n.0,prop.list)

n.0<-5
prop.list<-list(c(0.3,0.6),c(0.25,0.5,0.75),c(0.1,0.2,0.8,0.9))
validation.ord(n.0,prop.list)

n.0<-3
prop.list<-list(c(0.3,0.6),c(0.25,0.5,-0.75),c(0.1,0.2,0.8,1.5))
validation.ord(n.0,prop.list)

n.0<-3
prop.list<-list(0.3,c(0.3,0.4),c(0.4,0.2,0.3))
validation.ord(n.0,prop.list)

## End(Not run)
```

# Index

correlation.bound.check, [2](#), [3](#), [5](#), [13](#)  
correlation.limits, [2](#), [3](#), [4](#), [13](#)

gen.PoisBinOrd, [2](#), [6](#)

intermediate.corr.BO, [2](#), [7](#), [10](#), [12](#)  
intermediate.corr.PBO, [2](#), [8](#), [9](#), [11](#), [12](#)  
intermediate.corr.PP, [2](#), [10](#), [10](#), [12](#)

overall.corr.mat, [2](#), [6](#), [11](#)

PoisBinOrd (PoisBinOrd-package), [2](#)  
PoisBinOrd-package, [2](#)

validation.bin, [2](#), [12](#)  
validation.corr, [2](#), [3](#), [5](#), [13](#)  
validation.ord, [2](#), [14](#)