Package ‘bWGR’

July 16, 2022

Type  Package
Title  Bayesian Whole-Genome Regression
Version  2.1
Date  2022-07-14
Author  Alencar Xavier, William Muir, David Habier, Kyle Kocak, Shizhong Xu, Katy Rainey.
Maintainer  Alencar Xavier <alenxav@gmail.com>
Description  Whole-genome regression methods on Bayesian framework fitted via EM
or Gibbs sampling, single step (<doi:10.1534/g3.119.400728>), univariate and multivariate
(<doi:10.1186/s12711-022-00730-w>),
with optional kernel term and sampling techniques (<doi:10.1186/s12859-017-1582-3>).
License  GPL-3
Imports  Matrix, Rcpp, RcppEigen
LinkingTo  Rcpp, RcppEigen
Depends  R (>= 4.0)
NeedsCompilation  yes
Repository  CRAN
Date/Publication  2022-07-16 13:40:04 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>bWGR-package</td>
<td>2</td>
</tr>
<tr>
<td>Dataset</td>
<td>2</td>
</tr>
<tr>
<td>WGR1 (MC)</td>
<td>3</td>
</tr>
<tr>
<td>WGR2 (EM)</td>
<td>6</td>
</tr>
<tr>
<td>WGR3 (MV)</td>
<td>8</td>
</tr>
<tr>
<td>XTRA 1</td>
<td>9</td>
</tr>
<tr>
<td>XTRA 2</td>
<td>11</td>
</tr>
</tbody>
</table>

Index  13
### bWGR-package

**Bayesian Whole-Genome Regression**

**Description**


**Details**

- **Package:** bWGR
- **Type:** Package
- **Version:** 2.1
- **Date:** 2022-07-14
- **License:** GPL-3

**Author(s)**

Alencar Xavier, William Muir, Kyle Kocak, Shizhong Xu, Katy Rainey.

Maintainer: Alencar Xavier <alenxav@gmail.com>

**Examples**

```r
data(tpod)
Fit = wgr(y, gen)
cor(y, Fit$hat)
```

---

### Dataset

**Tetra-seed Pods**

**Description**

Two biparental crosses phenotyped for the percentage of pods containing four seeds

**Usage**

```r
data(tpod)
```
Details

Soybean nested association panel with 2 families (fam) containing 196 individuals. Genotypic matrix (gen) have 376 SNP across 20 chromosome (chr). Phenotypic information (y) regards the proportion of tetra-seed pods. Data provided by Rainey Lab for Soybean Breeding and Genetics, Purdue University.

Author(s)

Alencar Xavier and Katy Rainey

Description

Univariate model to find breeding values through regression with optional resampling techniques (Xavier et al. 2017) and polygenic term (Kernel). See “Details” for additional standalone functions written in C++.

Usage

wgr(y, X, it=1500, bi=500, th=1, bag=1, rp=FALSE, iv=FALSE, de=FALSE, pi=0, df=5, R2=0.5, eigK=NULL, VarK=0.95, verb=FALSE)

Arguments

y  Numeric vector of observations (n) describing the trait to be analyzed. NA is allowed.
X  Numeric matrix containing the genotypic data. A matrix with n rows of observations and (m) columns of molecular markers.
it  Integer. Number of iterations or samples to be generated.
bi  Integer. Burn-in, the number of iterations or samples to be discarded.
th  Integer. Thinning parameter, used to save memory by storing only one every 'th' samples.
bag  If different than one, it indicates the proportion of data to be subsampled in each Markov chain. For datasets with moderate number of observations, values of bag from 0.30 to 0.60 may speed up computation without losses in prediction properties. This argument enable users to enhance MCMC through subsampling (Xavier et al. 2017).
rp  Logical. Use replacement for bootstrap samples when bag is different than one.
iv  Logical. Assign markers independent variance, a T prior from a mixture of normals. If true, turns the default model BLUP into BayesA.
de  Logical. Assign markers independent variance through double-exponential prior. If true, turns the default model BLUP into Bayesian LASSO. This argument overrides iv.
pi: Value between 0 and 1. If greater than zero it activates variable selection, where markers have expected probability pi of having null effect.
df: Prior degrees of freedom of variance components.
R2: Expected R2, used to calculate the prior shape.
eigK: Output of function `eigen`. Spectral decomposition of the kernel used as a second random effect (e.g., pedigree matrix).
VarK: Numeric between 0 and 1. For reduction of dimensionality. Indicates the proportion of variance explained by Eigenpairs used to fit second random effect.
verb: Logical. If verbose is TRUE, function displays MCMC progress bar.

Details

The model for the whole-genome regression is as follows:

\[ y = mu + Xb + u + e \]

where \( y \) is the response variable, \( mu \) is the intercept, \( X \) is the genotypic matrix, \( b \) is the regression coefficient or effect of an allele substitution, with \( d \) probability of being included into the model, \( u \) is the polygenic term if a kernel is used, and \( e \) is the residual term.

Users can obtain four WGR methods out of this function: BRR (pi=0, iv=F), BayesA (pi=0, iv=T), BayesB (pi=0.95, iv=T), BayesC (pi=0.95, iv=F) and Bayesian LASSO or BayesL (pi=0, de=T). Theoretical basis of each model is described by de los Campos et al. (2013).

Gibbs sampler that updates regression coefficients is adapted from GSRU algorithm (Legarra and Misztal 2008). The variable selection of functions `wgr`, `BayesB` and `BayesC` works through the unconditional prior algorithm proposed by Kuo and Mallick (1998), whereas `BayesCpi` and `BayesDpi` are based on Metropolis-Hastings. Prior shape estimates are computed as \( S_b = R^2 * df * \text{var}(y) / \text{MS}_x \) and \( S_e = (1-R^2) * df * \text{var}(y) \), with an exception for `BayesC` and `BayesCpi` where the prior shape is \( S_b = R^2 * df * \text{var}(y) / \text{MS}_x / (1-pi) \). The polygenic term is solved by Bayesian algorithm of reproducing kernel Hilbert Spaces proposed by de los Campos et al. (2010).

In addition to `wgr`, standalone C++ functions available include:

01) `BayesA(y,X,it=1500,bi=500,df=5,R2=0.5)`
02) `BayesB(y,X,it=1500,bi=500,pi=0.95,df=5,R2=0.5)`
03) `BayesC(y,X,it=1500,bi=500,pi=0.95,df=5,R2=0.5)`
04) `BayesCpi(y,X,it=1500,bi=500,df=5,R2=0.5)`
05) `BayesDpi(y,X,it=1500,bi=500,df=5,R2=0.5)`
06) `BayesL(y,X,it=1500,bi=500,df=5,R2=0.5)`
07) `BayesRR(y,X,it=1500,bi=500,df=5,R2=0.5)`

The implementations that support two random effects include:

08) `BayesA2(y,X1,X2,it=1500,bi=500,df=5,R2=0.5)`
09) `BayesB2(y,X1,X2,it=1500,bi=500,pi=0.95,df=5,R2=0.5)`
10) `BayesRR2(y,X1,X2,it=1500,bi=500,df=5,R2=0.5)`

And the cross-validation for the C++ implementations, with arguments analogous to `emCV`.

`mcmcCV(y,gen,k=5,n=5,it=1500,bi=500,pi=0.95,df=5,R2=0.5,avg=T,llo=NULL,tbv=NULL,ReturnGebv=FALSE)`
Value

The function wgr returns a list with expected value from the marker effect ($b$), probability of marker being in the model ($d$), regression coefficient ($g$), variance of each marker ($V_b$), the intercept ($\mu$), the polygene ($u$) and polygenic variance ($V_k$), residual variance ($V_e$) and the fitted value ($\hat{y}$).

Author(s)

Alencar Xavier

References


Examples

```r
## Not run:

# Load data
data(tpod)

# BLUP
fit_BRR = wgr(y,gen,iv=FALSE,pi=0)
cor(y,fit_BRR$hat)

# BayesA
fit_BayesA = wgr(y,gen,iv=TRUE,pi=0)
cor(y,fit_BayesA$hat)

# BayesB
fit_BayesB = wgr(y,gen,iv=TRUE,pi=.95)
cor(y,fit_BayesB$hat)

# BayesC
fit_BayesC = wgr(y,gen,iv=FALSE,pi=.95)
cor(y,fit_BayesC$hat)

# BayesCpi
fit_BayesCpi = BayesCpi(y,gen)
```
WGR2 (EM)  

**Description**

Univariate models to find breeding values through regression fitted via expectation-maximization implemented in C++.

**Usage**

```r
emRR(y, gen, df = 10, R2 = 0.5)
emBA(y, gen, df = 10, R2 = 0.5)
emBB(y, gen, df = 10, R2 = 0.5, Pi = 0.75)
emBC(y, gen, df = 10, R2 = 0.5, Pi = 0.75)
emBL(y, gen, R2 = 0.5, alpha = 0.02)
emEN(y, gen, R2 = 0.5, alpha = 0.02)
emDE(y, gen, R2 = 0.5)
emML(y, gen, D = NULL)
emCV(y, gen, k = 5, n = 5, Pi = 0.75, alpha = 0.02,
    df = 10, R2 = 0.5, avg=TRUE, llo=NULL, tbv=NULL, ReturnGebv = FALSE)
```

**Arguments**

- `y`  
  Numeric vector of response variable \(n\). NA is not allowed.
- `gen`  
  Numeric matrix containing the genotypic data. A matrix with \(n\) rows of observations and \(m\) columns of molecular markers.
- `df`  
  Hyperprior degrees of freedom of variance components.
- `R2`  
  Expected R2, used to calculate the prior shape (de los Campos et al. 2013).
- `Pi`  
  Value between 0 and 1. Expected probability \(p_i\) of having null effect (or 1-\(p_i\) if \(p_i>0.5\)).
alpha
Value between 0 and 1. Intensity of L1 variable selection.

D
NULL or numeric vector with length p. Vector of weights for markers.

k
Integer. Folding of a k-fold cross-validation.

n
Integer. Number of cross-validation to perform.

avg
Logical. Return average across CV, or correlations within CV.

llo
NULL or a vector (numeric or factor) with the same length as y. If provided, the cross-validations are performed as Leave a Level Out (LLO). This argument allows the user to redefine the splits. This argument overrides k and n.

tbv
NULL or numeric vector of ‘true breeding values’ (n) to use to compare cross-validations to. If NULL, the cross-validations will have the phenotypes as prediction target.

ReturnGebv
Logical. If TRUE, it returns a list with the average marker values and fitted values across all cross-validations, in addition to the regular output.

Details
The model for the whole-genome regression is as follows:

\[ y = \mu + Xb + e \]

where \( y \) is the response variable, \( \mu \) is the intercept, \( X \) is the genotypic matrix, \( b \) is the effect of an allele substitution (or regression coefficient) and \( e \) is the residual term. A k-fold cross-validation for model evaluation is provided by \( \text{emCV} \).

Value
The EM functions returns a list with the intercept (\( \mu \)), the regression coefficient (\( b \)), the fitted value (\( \hat{y} \)), and the estimated intraclass-correlation (\( h^2 \)).

The function \( \text{emCV} \) returns the predictive ability of each model, that is, the correlation between the predicted and observed values from k-fold cross-validations repeated \( n \) times.

Author(s)
Alencar Xavier

Examples
```R
# Not run:
data(tpod)
emCV(y, gen, 3, 3)
```

# End(Not run)
Multivariate Regression

Description

Multivariate model to find breeding values.

Usage

\[
\text{mrr}(Y, X) \\
\text{mkr}(Y, K)
\]

Arguments

- \( Y \): Numeric matrix of observations x trait. NA is allowed.
- \( K \): Numeric matrix containing the relationship matrix.
- \( X \): Numeric matrix containing the genotyping matrix.

Details

Algorithm is described in Xavier and Habier (2022). The model for the ridge regression (mrr) is as follows:

\[
Y = Mu + XB + E
\]

where \( Y \) is a matrix of response variables, \( Mu \) represents the intercepts, \( X \) is the matrix of genotypic information, \( B \) is the matrix of marker effects, and \( E \) is the residual matrix.

The model for the kernel regression (mkr) is as follows:

\[
Y = Mu + UB + E
\]

where \( Y \) is a matrix of response variables, \( Mu \) represents the intercepts, \( U \) is the matrix of Eigen-vector of \( K \), \( b \) is a vector of regression coefficients and \( E \) is the residual matrix.

Algorithm: Residuals are assumed to be independent among traits. Regression coefficients are solved via a multivariate adaptation of Gauss-Seidel Residual Update. Since version 2.0, the solver of mrr is based on the Randomized Gauss-Seidel algorithm. Variance and covariance components are solved with an EM-REML like approach proposed by Schaeffer called Pseudo-Expectation. S

Other related implementations:

- 01) mkr2X\((Y, K1, K2)\): Solves multi-trait kernel regressions with two random effects.
- 02) mrr2X\((Y, X1, X2)\): Solves multi-trait ridge regressions with two random effects.
- 03) MRR3\((Y, X, \ldots)\): Similar to mrr, with a couple additional parameters.
Value

Returns a list with the random effect covariances (Vb), residual variances (Ve), genetic correlations (GC), matrix with marker effects (b) or eigenvector effects (if mkr), intercepts (mu), heritabilities (h2), and a matrix with fitted values (hat).

Author(s)

Alencar Xavier, David Habier

References


Examples

```r
# Load genomic data
data(tpod)
X = CNT(gen)

# Simulate phenotyp
sim = SimY(X)
Y = sim$Y
TBV = sim$tbv

# Fit regression model
test = mrr(Y,X)

# Genetic correlation
test$GC

# Heritability

test$h2

# Accuracy
diag(cor(TBV,test$hat))
```

Description

Function to solve univariate mixed models with or without the usage of omic information. This function allows single-step modeling of replicated observations with marker information available through the usage of a linkage function to connect to a whole-genome regression method. Genomic estimated values can be optionally deregressed (no shrinkage) while fitting the model.
Usage

```r
mixed(y, random=NULL, fixed=NULL, data=NULL, X=list(),
      alg=emML, maxit=10, Deregress=FALSE, ...)
```

Arguments

- `y`: Response variable from the data frame containing the dataset.
- `random`: Formula. Right-hand side formula of random effects.
- `fixed`: Formula. Right-hand side formula of fixed effects.
- `data`: Data frame containing the response variable, random and fixed terms.
- `X`: List of omic incidence matrix. Row names of these matrices connect the omic information to the levels of the indicated random terms (e.g., `X=list("ID"=gen)`).
- `alg`: Function. Whole-genome regression algorithm utilized to solve link functions. These include MCMC (wgr, BayesB, etc) and EM (emEN, emDE, etc) algorithms. By default, it runs maximum likelihood emML.
- `maxit`: Integer. Maximum number of iterations.
- `Deregress`: Logical. Deregress (unshrink) coefficients while fitting the model?
- `...`: Additional arguments to be passed to the whole-genome regression algorithms specified on `alg`.

Details

The model for the whole-genome regression is as follows:

\[ y = Xb + Zu + Wa + e \]

where `y` is the response variable, \( Xb \) corresponds to the fixed effect term, \( Zu \) corresponds to one or more random effect terms, \( W \) is the incidence matrix of terms with omic information and \( a \) is omic values by \( a = Mg \), where \( M \) is the genotypic matrix and \( g \) are marker effects. Here, \( e \) is the residual term. An example is provided using the data from the NAM package with: `demo(mixedmodel)`.

Alternative (and updated) implementations have similar syntax:

1. `mm(y, random=NULL, fixed=NULL, data=NULL, M=NULL, bin=FALSE, AM=NULL, it=10, verb=TRUE, FLM=TRUE, wgtM=TRUE, cntM=TRUE, nPc=3)`
2. `mtmixed = function(resp, random=NULL, fixed=NULL, data, X=list(), maxit=10, init=10, regVC=FALSE)`

Value

The function wgr returns a list with Fitness values (Fitness) containing observation obs, fitted values hat, residuals res, and fitted values by model term fits; Estimated variance components (VarComp) containing the variance components per se (VarComponents) and variance explained by each model term (VarExplained), regression coefficients by model term (Coefficients), and the effects of structured terms (Structure) containing the marker effects of each model term where markers were provided.
Author(s)

Alencar Xavier

References


Examples

```r
## Not run:
demo(mixedmodel)
## End(Not run)
```

Additional tools

Description

Complementary functions that may help with handling parameters and routine operations.

Details

- `emGWA(y, gen)` # Simple MLM for association analysis
- `markov(gen, chr=NULL)` # Markovian imputation of genotypes coded as 012
- `IMP(X)` # Imputes genotypes with SNP expectation (column average)
- `CNT(X)` # Recodes SNPs by centralizing columns in a matrix
- `GAU(X)` # Creates Gaussian kernel as exp(-Dist2/mean(Dist2))
- `GRM(X, Code012=FALSE)` # Creates additive kinship matrix VanRaden (2008)
- `SPC(y, blk, row, col, rN=3, cN=1)` # Spatial covariate
- `SPM(blk, row, col, rN=3, cN=1)` # Spatial design matrix
- `SibZ(id, p1, p2)` # Pedigree design matrix compatible to regression methods
- `Hmat(ped, gen=NULL)` # Kinship combining pedigree and genomics
- `EigenGRM(X, centralizeZ = TRUE, cores = 2)` # GRM using Eigen library
- `EigenARC(X, centralizeZ = TRUE, cores = 2)` # ArcCosine kernel
- `EigenGAU(X, phi = 1.0, cores = 2)` # Gaussian kernel using Eigen library
- `EigenCNT(X, cores = 2)` # Center SNPs without missing Eigen library
- `EigenEVD(A, cores = 2)` # Eigendecomposition from Eigen library
- `EigenBDCSVD(X, cores = 2)` # BDC single value composition from Eigen
- `EigenJacobiSVD(X, cores = 2)` # Jacobi single value composition from Eigen
- `EigenAcc(X1, X2, h2 = 0.5, cores = 2)` # Deterministic accuracy X1 -> X2
Author(s)

Alencar Xavier
### Index

<table>
<thead>
<tr>
<th>Package/Function</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesA (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesA2 (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesB (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesB2 (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesC (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesCpi (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesDpi (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesL (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesRR (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BayesRR2 (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>bWGR (bWGR-package)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>chr (Dataset)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CNT (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Dataset</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EigenAcc (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>EigenARC (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>EigenBCSVD (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>EigenCNT (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>EigenEVD (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>EigenGAU (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>EigenGRM (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>EigenJacobiSVD (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>emBA (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emBB (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emBC (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emBL (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emCV (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emDE (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emEN (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emGWA (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>emML (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emML2 (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>emRR (WGR2 (EM))</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>fam (Dataset)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>GAU (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>gen (Dataset)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>GRM (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>GS2EIGEN (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>GSEN (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>GSFLM (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>GSRR (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Hmat (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>IMP (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>KMUP (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>KMUP2 (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>markov (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>mcmcCV (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>mixed (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>mkr (WGR3 (MV))</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>mkr2X (WGR3 (MV))</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>mm (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>mrr (WGR3 (MV))</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>mrr2X (WGR3 (MV))</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>MRR3 (WGR3 (MV))</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>mtgmsru (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>mtmixed (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>NNS (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>NNSEARCH (XTRA 1)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>SibZ (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>SimY (WGR3 (MV))</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>SPC (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>SPM (XTRA 2)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>tpod (Dataset)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>wgr (WGR1 (MC))</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>WGR1 (MC)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>WGR2 (EM)</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>WGR3 (MV)</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>XTRA 1</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>
XTRA 2, 11

y (Dataset), 2