

# Package ‘qgen’

April 19, 2009

**Type** Package

**Title** Quantitative Genetics using R

**Version** 0.03-02

**Date** 2007-01-24

**Author** Thomas Fabbro <thomas.fabbro@unibas.ch>

**Maintainer** Thomas Fabbro <thomas.fabbro@unibas.ch>

**Depends** MASS, lme4, Matrix, methods

**Description** The package qgen is a collection of functions to analyse quantitative genetic data. It is especially helpful to perform parametric resampling of quantitative genetic data sets. Resampling allows first to determine a priori the expected variance of an estimator, second for a given empirical data set to calculate bootstrap confidence intervals, and third to evaluate different estimators and confidence intervals.

**License** GPL (>= 2)

**URL** <http://www.evolution.unibas.ch/people/TFabbro/qgen/qgen.htm>

**Repository** CRAN

**Date/Publication** 2009-02-04 09:57:48

## R topics documented:

qgen-package . . . . .	2
cal . . . . .	2
dis . . . . .	3
emp . . . . .	4
est . . . . .	5
qgen-classes . . . . .	7
sim . . . . .	10
sinapis . . . . .	11
sta . . . . .	12

stat1 . . . . .	13
stat2e . . . . .	14
the . . . . .	16

<b>Index</b>	<b>19</b>
--------------	-----------

---

qgen-package                      *Quantitative Genetics in R*

---

## Description

Functions to estimate parameters and compute statistics and their corresponding confidence intervals (e.g. heritability, genetic correlations...). All functions are written for sib analysis with up to 2 levels of nesting. Parametric simulations allow the calculation of bootstrap confidence intervals and the evaluation of dependence on different assumptions.

## Details

Further information is available in the following [vignette](#):

overview    An overview with examples (source, pdf)

For a complete list of functions with individual help pages, use `library(help="qgen")`.

## Author(s)

Thomas Fabbro <thomas.fabbro@unibas.ch>

---

cal                                      *Organising the resampling procedures.*

---

## Description

A function that generates a text file containing plain R code. This code has to be read and run by R (e.g. `source()`). Thus a `multi`-object is generated from which the statistics can be calculated using the `sta`-function.

## Usage

```
cal(filename, repetitions=2, splitting=1,
      file=TRUE, path=~/"qgen/")
```

**Arguments**

filename	the name of a file containing an object of class <code>paraDATA</code> or <code>multi</code>
repetitions	number of repetitions
splitting	not yet implemented; for splitting tasks among different computers
file	logical flag: Should the object be written to a file
path	the path for the files: <code>calfile.r</code> , and <code>multi...rda</code>

**Details**

The organisation of the resampling procedure with a separate file, which has to be run line by line, makes the organisation of the resampling very flexible. It allows to split up very computer intensive calculations among different machines. Therefore the file created by this function can simply be divided into different parts that are then run on different machines. The simple structure of the `multi-class`-object allows then easily to put the parts together again. Another advantage is that errors occurring during optimization are skipped and don't cause problems.

**Value**

a `multi` object, stored in a file: `path/multi...rda`

**Note**

*name*: from calling the different functions

**See Also**

The vignette and the functions `sta` and `dis`

---

dis *Bootstrap confidence intervals*

---

**Description**

Calculates different bootstrap confidence intervals.

**Usage**

```
dis(path=~"/qgen/", alpha=0.05)
```

**Arguments**

path	path searched for <code>statX.rda</code> -files.
alpha	number indicating the two sided error probability

**Details**

Depending on the available levels of resampling, percentile , basic (for simple resampling) and studentized (for nested resampling) confidence intervals are calculated

**Value**

the confidence intervals are printed

**Examples**

```
### dis()
```

---

emp

*Transforms empirical data sets into a conformable paraDATA object*

---

**Description**

Takes empirical data from a dataframe, defines new columns, and puts the data in an object of the [paraDATA-class](#).

**Usage**

```
emp(data.use=NULL,
     fixedblock.use, character.use,
     environment.use="environment",
     randomblock.use,
     sire.use="sire",
     dam.use="dam",
     individual.use="individual",
     without="", partitioning="REML",
     file=TRUE, path="~/qgen/")
```

**Arguments**

`data.use`            the name of a dataframe

`fixedblock.use`        the name of the column in `data.use` containing the levels of a fixed blocking factor (as factor)

`character.use`        a vector of characters strings indicating the column names to be chosen for analysis (numeric values only)

`environment.use`       the name of the column in `data.use` containing the levels of the environments (as factor)

`randomblock.use`       the name of the column in `data.use` indicating the levels of a random blocking factor (as factor)

<code>sire.use</code>	the name of the column in <code>data.use</code> containing the levels of the sire effect (as factor)
<code>dam.use</code>	the name of the column in <code>data.use</code> containing the levels of the dam effect (as factor)
<code>individual.use</code>	the name of the column in <code>data.use</code> containing the levels of the individual effect (as factor)
<code>without</code>	a vector of characters indicating the names of the rows to be excluded (for outliers)
<code>partitioning</code>	a character string indicating the variance partitioning method (used by the function <code>est</code> ); see <code>the</code> for details
<code>file</code>	logical flag: Should the object be written to <code>emp.rda</code>
<code>path</code>	indicating the path where the file <code>emp.rda</code> should be written

**Value**

an object of class `paraDATA` with

`orig`            **hist** "emp"

**Note**

*name*: empirical investigations

**See Also**

`paraDATA`

**Examples**

```
emp(file=FALSE) #creates a paraDATA object with the data from "sinapis"
```

---

est

*estimating parameters from data sets*

---

**Description**

estimates variance components for random effects and coefficients of the fixed effects for a given `paraDATA`-object.

**Usage**

```
est(paraDATA, file=TRUE, path="~/qgen/")
```

## Arguments

<code>paraDATA</code>	list with a defined structure; see <code>paraDATA</code>
<code>file</code>	Logical flag: Should the <code>paraDATA</code> object be written to <code>path/"est.rda"</code> ?
<code>path</code>	indicating the path where the file <code>est.rda</code> should be written

## Details

The parameters of the `the` and the `emp` function determine the estimations that are performed and the exact structure of the output

## Value

a `paraDATA`-object:

<code>orig</code>	full
<code>supl</code>	full
<code>para</code>	full, with <code>phS=NULL</code>
<code>spec</code>	depending on the <code>supl</code> -slot of the <code>paraDATA</code> and on the <code>history</code> of the object itself:

**unbal** information on the unbalancedness of the data; different degrees of freedom:

**xxDF** normal degrees of freedom

**xxDFappREML** Satterthwaite approximated degrees of freedom using the REML variance components

**xxDFappANOVA** Satterthwaite approximated degrees of freedom using the ANOVA estimators for variance components

**xxDFappANOVAuw** Satterthwaite approximated degrees of freedom using the ANOVA estimators with unweighted sums of squares for variance components

weights used to calculate the expected mean squares

**w1u** for dam variance in calculation of sire mean square

**w2u** for sire variance

**w3u** for dam variance in calculation of dam mean square

**modelsummary** additional information:

**sBLUP** the best linear unbiased predictors for the sire effect

**modelsumry** the `summary(lme4-object)`

**secondcontrast** the same model with a different contrast matrix to estimate the difference in on character between environments:

**FIX** an array with the means of all fixed effects

**modelsumry2** the `summary(lme4-object)` for a second model estimated with contrast for testing whether a character differs between two environments

**secondpartitioning** variance components additionally (to REML) estimated by:

**ANOVA** estimators (normal):

**rbS** variance component for factor sire  
**siS** variance component for factor sire  
**daS** variance component for factor dam  
**idS** variance component for residuals  
**phS** NULL  
**error** 0; the model residuals are included in the idS

**ANOVAuw** estimators based on unweighted sums of squares (for balanced data equal to ANOVA):

**rbS** variance component for factor sire  
**siS** variance component for factor sire  
**daS** variance component for factor dam  
**idS** variance component for residuals  
**phS** NULL  
**error** 0; the model residuals are included in the idS

#### Note

*name*:est from estimation

#### References

Burdick, R. K. and Graybill, F. A. (1992) *Confidence intervals on variance components*. Marcel Dekker.  
 Sen, B., Graybill, F. A. and Ting, N. (1992) Confidence intervals on variance components. *Biometrical Journal* **3**, 259–274.

#### See Also

[paraDATA](#)

#### Examples

```
parameters <- est(sim(the(file=FALSE)), file=FALSE)
# takes the default causal parameters of the()
# simulates a data set
# estimates the parameters
```

---

qgen-classes

*qgen*: Class definitions

---

#### Description

Describing classes associated with qgen

**Usage**

```
###
"paraDATA"# is a virtual class, extended by:
"orig"
"supl"
"para"
"DATA"
"spec"
###
"multi"
###
"stat"
```

**paraDATA**

**orig hist** A list with the name of all functions that have handled the object; in chronological order

**warn** A list with warnings (character strings) from the functions that have handled the object; in chronological order

**time** A list with the time (character strings) when the functions have handled the object; in chronological order

**part** A character string; defines which variance partitioning method(s) should be used:  
**"a11"** REML for the resampling; additionally ANOVA and ANOVAuw  
**"REML"** REML  
**"ANOVA"** ANOVA with traditional sums of squares  
**"ANOVAuw"** ANOVA with unweighted sums of squares

**supl chN** the number of characters

**enN** the number of environments

**fbN** the number of fixed blocks (crossed with environments and characters)

**rbN** the number of random blocks (crossed with sire)

**siN** the number of sires

**daN** the number of dams within sires

**idN** the number of individuals

**miss** proportion of randomly missing values per environment–character–fixedblock combination in a array;  $\text{dim} = c(\text{fb}, \text{en}, \text{ch})$ ; if scalar all are assumed to be equal

**para** all parameters needed to describe the distribution of the data:

**rbS** matrix, random block effect

**siS** matrix, sire effect

**daS** matrix, dam effect

**idS** matrix, individual effect

**phs** matrix, phenotypic effect (not observable!)

**error** scalar, variance of model residuals

**fixe** array ( $\text{dim} = c(\text{fb}, \text{en}, \text{ch})$ ), cell means of every environment–characer-fixedblock combination

**DATA dat** data-frame with columns

**ch** a factor with one level for every character  
**en** a factor with one level for every environment  
**fb** a factor with one level for every fixed block (crossed with the characters and environment)  
**rb** a factor with one level for every random block (crossed with the characters and environment)  
**si** a factor with one level for every sire  
**da** a factor with one level for every dam  
**id** a factor with one level for every individual  
**y** measurement (numeric)

**spec** the place for specific information on the model used to estimate the parameter etc.

**additional.partitioning** A list with qgen `para-class` objects containing parameters variance partitioning by ANOVA and/or ANOVAuw (with unweighted sums of squares)

**unbalanced** A list; information on the unbalancedness of the data (where "xx" is "si", "da", or "id")

**xxDF** normal degrees of freedom

**xxDFappREML** Satterthwaite approximated degrees of freedom using the REML variance components

**xxDFappANOVA** Satterthwaite approximated degrees of freedom using the ANOVA estimators for variance components

**xxDFappANOVAuw** Satterthwaite approximated degrees of freedom using the ANOVA estimators with unweighted sums of squares for variance components

weights used to calculate the expected mean squares

**w1u** for dam variance in calculation of sire mean square

**w2u** for sire variance

**w3u** for dam variance in calculation of dam mean square

**modelsummary** A list with additional information on the REML model that was used to estimate the parameters

**sBLUP** A matrix with the best linear unbiased predictors for the sire effect

**modelsummary** An object of `summary.lmer-class` (package Matrix)

**secondcontrast** A list with additional modelsummaries if different model contrasts were used

## multi

**list.paraDATA** a list of `paraDATA`-objects

**level** a character indicating the level of resampling (T,S,R,Q)

**x** still empty

**y** still empty

## stat

**orig** Object of `orig-class`

**stat** A numeric value of the statistic

**lower.ci** A numeric value, the lower confidence limit

**upper.ci** A numeric value, the upper confidence limit

**lower.limes** A numeric value, the lowest possible value of that statistic (for setting limits in plots)

**upper.limes** A numeric value, the highest possible value of that statistic (for setting limits in plots)

### See Also

Functions that construct `paraDATA`-objects: `emp` (from empirical data), `the` (from chosen biological parameters). Functions that handle and manipulate `paraDATA`-objects: `sim`, `est`, `stat1`.

The function that constructs `multi`-objects: `cal` (exactly: running the `calfile.rda` constructed by `cal`).

The function that handles and manipulates `multi`-objects: `sta`.

The function that constructs `stat`-objects: `sta`.

---

sim

*simulation of quantitative genetic data sets*

---

### Description

This function simulates data sets according to the sampling design and the parameters given by a `paraDATA`-object.

### Usage

```
sim(paraDATA, file=FALSE, path="~/qgen/")
```

### Arguments

<code>paraDATA</code>	an object of class <code>paraDATA</code>
<code>file</code>	Logical flag: Should the <code>paraDATA</code> object be written to <code>path/"sim.rda"</code> ?
<code>path</code>	indicating the path where the file <code>sim.rda</code> should be written

### Value

a `paraDATA`-object, with the slots:

<code>orig</code>	full
<code>supl</code>	full
<code>para</code>	full, the realized sampling variance-covariance components; <code>phS=NULL</code>
<code>DATA</code>	full; the simulated data set
<code>spec</code>	NULL

### Note

*Name:* simulation

**Examples**

```

my.parameters <- the(file=FALSE)           # defines a paraDATA object
my.paraDATA <- sim(my.parameters)        # simulates the corresponding data set
data.set <- my.paraDATA@DATA@dat        # extracts the simulated data
                                         # from the paraDATA--object

```

---

sinapis

*Paternal Half-sib Data from Sinapis arvensis*


---

**Description**

Data from an experiment with paternal half-sibs. It allows to study the inheritance of leaf area and plant weight and the genetic correlation among this two characters.

**Usage**

```
data(sinapis)
```

**Format**

A data frame with 1100 observations on the following 7 variables.

**fixedblock** a blocking factor with 3 levels

**randomblock** a blocking factor without any levels

**sire** a factor with 31 levels

**dam** a factor with 3 levels (to be nested within sire)

**individual** a factor with 1100 levels

**environment** a factor with 2 levels ("cont" for control, "herb" for herbivory)

**leafarea** numeric: the area of one leaf in (unit: square centimetre)

**plantweight** numeric: the fresh weight of the above-ground plant parts (unit: gram)

**Source**

my phd thesis

**Examples**

```

data(sinapis)
## str(sinapis)

```

---

sta	<i>Calculates the specified statistic on all replications of one resampling level</i>
-----	---

---

### Description

This function takes the name of one file and applies the selected function to all replications of the same resampling level.

### Usage

```
sta(filename= "",
     statistic.name,
     alpha = 0.05,
     transformation = "none",
     file=TRUE,
     path = "~/qgen/")
```

### Arguments

filename	character string indicating the name of the file (all files of this resampling level will then be chosen!)
alpha	number indicating the two sided error probability
statistic.name	indicate a function that calculates the statistic you are interested in (e.g. <a href="#">stat1</a> )
transformation	character string indicating the transformation that should be applied on the statistic to calculate bootstrap confidence intervals ("logodds", "asin", "fisher")
file	logical flag: Should the object be written to a file
path	indicating the path where the files should be read and the written

### Details

This function is used to calculate a selected statistic from all [paraDATA](#)-objects within the indicated file. It automatically applies the same statistic to all [paraDATA](#)-objects on the same replication level, even if they are stored in different files.

### Value

multi            an object of class [multi](#).  
 stored in a file within the selected path called "statT.rda", "statS.rda", "statR.rda", "statQ.rda", depending on the resampling level.

### Examples

```
##
```

---

 stat1 *Statistics for one trait*


---

**Description**

Calculates the narrow-sense heritability and its confidence intervals for one trait.

**Usage**

```
stat1(paraDATA,
      alpha=0.05,
      frommethod="REML",
      Satterthwaite=TRUE,
      tex.table=FALSE,
      file=FALSE,
      path="~/qgen/")
```

**Arguments**

paraDATA	object of class <code>paraDATA</code> created by the function <code>est</code> (or <code>emp</code> , see below)
alpha	two sided error probability
frommethod	If the paraDATA-object provides results form different variance-partioning methods you can choose which to take otherwise the REML solutions taken without warning: "ANOVAuw" ANOVA estimators with unweighted sums of squares "ANOVA" normal ANOVA estimators "REML" REML solutions
Satterthwaite	Logical flag: Should the approximated degree of freedoms for sire and dam be used (only possible if they are available)?
file	Logical flag: Should the object be written to a file called "stat1.rda"?
tex.table	Logical flag: should a LaTeX file be created with a table of all estimates
path	the path for the output files

**Details**

This function calculates the heritability of individual traits. If the paraDATA provides parameters for several characters or several environments or both, then for every character-trait combination heritability is calculated individually. If the paraDATA has a `unbal` slot with data about the balancedness of the data set then also the confidence intervals are calculated. If the paraDATA provides parameters estimated with diffent variance partitioning methods then it is possible to choose among them.

**Value**

an object of the class `stat`.

**Note**

*Attributes:* limes

**References**

Burdick, R. K. and Graybill, F. A. (1992) *Confidence intervals on variance components*. Marcel Dekker, New York.

Sen, B., Graybill, F. A. and Ting, N. (1992) Confidence intervals on variance components. *Biometrical Journal* **3**, 259–274.

**See Also**

(not yet implemented) `stat2e`, for the same statistics of an index of two environments; `statXt`, for several traits within one environment.

**Examples**

```
## calculates narrow-sense heritability and its confidence intervals
myparameters <- est(sim(the(file=FALSE)), file=FALSE) # takes the default causal parameters
# simulates a data set
# estimates the variance components
stat1(myparameters) # calculates heritability and
# confidence invals from estimated parameters
```

---

stat2e

*Statistics for characters in two environments*

---

**Description**

Calculates the narrow-sense heritability of plasticity and its confidence intervals.

**Usage**

```
stat2e(paraDATA,
       alpha=0.05,
       v=list( c(-1,1), c(0,1), c(1,0), c(1,1), c(1,-1) ),
       Satterthwaite=TRUE,
       file=FALSE,
       path="~/qgen/")
```

## Arguments

<code>paraDATA</code>	object of class <code>paraDATA</code> created by the function <code>est</code> (or <code>emp</code> , see below)
<code>alpha</code>	two sided error probability
<code>v</code>	list with weights for the two environments; the indices
<code>Satterthwaite</code>	Logical flag: Should the approximated degree of freedoms for sire and dam be used (only possible if they are available)?
<code>file</code>	Logical flag: Should the object be written to a file called "stat2e.rda"?
<code>path</code>	the path for the output files

## Details

This function calculates the heritability of individual traits. If the feeder provides parameters for several characters or several environments or both, then for every character–trait combination heritability is calculated individually. If the feeder has a `unbal` slot with data about the balancedness of the data set then also the confidence intervals are calculated. If the feeder provides parameters estimated with different variance partitioning methods then it is possible to choose among them.

## Value

an object of the class `stat`

## Note

*Attributes:* limes

## References

Burdick, R. K. and Graybill, F. A. (1992) *Confidence intervals on variance components*. Marcel Dekker, New York.

Sen, B., Graybill, F. A. and Ting, N. (1992) Confidence intervals on variance components. *Biometrical Journal* **3**, 259–274.

## See Also

`stat1`, for the same statistics of one trait; `statXt`, for several traits within one environment.

## Examples

```
## calculates narrow-sense heritability and its confidence intervals
```

---

the *Transformation of causal genetic parameters to observational parameters*

---

## Description

For theoretical investigations we need a function that takes the chosen causal genetic parameters and transforms them into observational parameters. Additionally the form of the object needs to be adjusted to fit the form of an object of class `paraDATA`. This function handles the following causal variance components: additive; dominance, maternal, micro-environmental; and the following sources of observable variance components: sire, dam, and individuals.

## Usage

```
the(partitioning = "all",
     chN = 1, # number of characters
     enN = 1, # number of environments
     fbN = 1, # number of fixed blocks
     rbN = 1, # number of random blocks
     siN = 100, # number sires (within each block)
     daN = 6, # number of dams (within each sire)
     idN = 3, # number of individuals (within each dam)
     randomblockCor = matrix(0.5, chN*enN, chN*enN),
     randomblockVar = rep(100, chN*enN),
     additiveCor = matrix(0.5, chN*enN, chN*enN),
     additiveVar = rep(100, chN*enN),
     dominanceCor = matrix(0.5, chN*enN, chN*enN),
     dominanceVar = rep(100, chN*enN),
     maternalCor = matrix(0.5, chN*enN, chN*enN),
     maternalVar = rep(100, chN*enN),
     environmentalCor = matrix(0, chN*enN, chN*enN),
     environmentalVar = rep(100, chN*enN),
     ch.names = paste("ch", 1:chN, sep=""),
     en.names = paste("en", 1:enN, sep=""),
     fb.names = paste("fb", 1:fbN, sep=""),
     fixe = array(0, dim=c(enN, chN, fbN)),
     miss = array(0, dim=c(enN, chN, fbN)),
     file=TRUE,
     path="~/qgen/")
```

## Arguments

`partitioning` character string: The method of variance partitioning used throughout the analysis

**"all"** (default) resampling is based on REML solutions but both ANOVA estimators are provided additionally

**"REML"** uses the package `lme4` to calculate REML solutions

	" <b>ANOVA</b> " calculates traditional ANOVA estimators; can only be used for investigating exactly one character in one environment
	" <b>ANOVAuw</b> " calculates ANOVA estimators using unweighted sums of squares; can only be used for investigating exactly one character in one environment
chN	number of characters
enN	number of environments
fbN	number of fixed blocks
rbN	number of random blocks, a random factor crossed with sire
siN	number of sires
daN	number of dams within sires
idN	number of individuals within dams
randomblockCor	for the random block effects
additiveCor	for the additive genetic effects
dominanceCor	for the dominance genetic effects
maternalCor	for the maternal effects
environmentalCor	for the microenvironmental effects
randomblockVar	for the random blocks
additiveVar	for the additive genetic effects
dominanceVar	for the dominance genetic effects
maternalVar	for the maternal effects
environmentalVar	for the microenvironmental effects
fixe	array with the mean of every fixedblock–environment–character combinations (sequence: character in environments in fixedblock, ex. fb1: en1ch1, en1ch2...en2ch1...enxchx fb2: en1ch1,...)
miss	vector with the proportion of randomly missing values per environment–character combination; if scalar all are assumed to be equal
en.names	vector of character strings: names for the environments
ch.names	vector of character strings: names for the traits
fb.names	vector of character strings: names for the fixed blocks
file	logical flag: Should the object be written to a file called "the.rda"?
path	character vector: containing a single path name

### Details

This function transforms the causal sources of variance (additive genetic effects, dominance genetic effects, common (maternal) environmental effects, and individual (special) environmental effects) into the three observable variance components from a nested full–sib, half–sib mating design (North-Carolina Design I). Assumptions: no sources of epistatic variance; see Lynch & Walsh 1997, p.572.

**Value**

a [paraDATA](#) object: with the individual slots

orig	fully specified
supl	fully specified
para	fully specified
data	empty
spec	empty

**Note**

*name*: theoretical investigations on chosen biological parameters

**References**

Lynch, M. and Walsh, B. (1997) *Genetics and analysis of quantitative traits*. Sinauer.

**See Also**

[emp](#), [paraDATA](#)

**Examples**

```
## create a paraDATA-object
## (with a full para section and an empty DATA section)
the(file=FALSE)
```

# Index

- \*Topic **datasets**
  - sinapis, 11
- \*Topic **file**
  - cal, 2
  - dis, 3
  - emp, 4
  - est, 5
  - sim, 10
  - sta, 12
  - stat1, 13
  - stat2e, 14
  - the, 16
- \*Topic **package**
  - qgen-package, 2
- cal, 2, 9
- DATA (*qgen-classes*), 7
- DATA-class (*qgen-classes*), 7
- dis, 3, 3
- emp, 4, 5, 9, 13, 15, 18
- est, 5, 5, 9, 13, 15
- multi, 2, 3, 12
- multi (*qgen-classes*), 7
- multi-class, 3
- multi-class (*qgen-classes*), 7
- orig, 5
- orig (*qgen-classes*), 7
- orig-class, 9
- orig-class (*qgen-classes*), 7
- para (*qgen-classes*), 7
- para-class (*qgen-classes*), 7
- paraDATA, 2, 5–7, 9, 10, 12, 13, 15, 16, 18
- paraDATA (*qgen-classes*), 7
- paraDATA-class, 4
- paraDATA-class (*qgen-classes*), 7
- qgen (*qgen-package*), 2
- qgen-classes, 7
- qgen-package, 2
- sim, 9, 10
- sinapis, 11
- spec (*qgen-classes*), 7
- spec-class (*qgen-classes*), 7
- sta, 2, 3, 9, 12
- stat, 14, 15
- stat (*qgen-classes*), 7
- stat-class (*qgen-classes*), 7
- stat1, 9, 12, 13, 15
- stat2e, 14, 14
- supl (*qgen-classes*), 7
- supl-class (*qgen-classes*), 7
- the, 5, 9, 16
- vignette, 2