Package 'envalysis'

May 8, 2026

Type Package

Title Miscellaneous Functions for Environmental Analyses

Version 0.7.0

Date 2024-03-20

Maintainer Zacharias Steinmetz <z.steinmetz@rptu.de>

Description Small toolbox for data analyses in environmental chemistry and ecotoxicology. Provides, for example, calibration() to calculate calibration curves and corresponding limits of detection (LODs) and limits of quantification (LOQs) according to German DIN 32645 (2008). texture() makes it easy to estimate soil particle size distributions from hydrometer measurements (ASTM D422-63, 2007).

URL https://github.com/zsteinmetz/envalysis, https://zsteinmetz.de/envalysis/

BugReports https://github.com/zsteinmetz/envalysis/issues

Encoding UTF-8

License GPL (>= 3)

LazyLoad yes

LazyData yes

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports drc, lmtest, ggplot2 (>= 3.4.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), MASS, investr, data.table, tibble, soiltexture

RoxygenNote 7.3.1

Config/testthat/edition 3

NeedsCompilation no

Author Zacharias Steinmetz [aut, cre] (ORCID: <https://orcid.org/0000-0001-6675-5033>), Julius Albert [ctb] (ORCID: <https://orcid.org/0000-0001-5468-6069>), Kilian Kenngott [ctb] (ORCID: <https://orcid.org/0000-0002-9600-4502>)

Repository CRAN

Date/Publication 2024-03-20 15:10:02 UTC

Contents

bisdom . . . 2
calibration . . . 3
clayloam . . . 6
din32645 . . . 7
icp . . . 8
matrix_effect . . . 8
neitzel2003 . . . 9
phenolics . . . 10
rmse . . . 11
se . . . 12
signifig . . . 13
sorption . . . 14
texture . . . 15
theme_publish . . . 17
weight_select . . . 18

Index **20**

`bisdom` *Categorize water drop penetration times*

Description

This wrapper function categorizes water drop penetration times (WDPT) in seconds according to the scale proposed by Bisdom et al. (1993).

Usage

```
bisdom(x, ...)
```

Arguments

`x`	a numeric vector containing WDPT measurement data [s].
`...`	arguments passed to `findInterval()`.

Author(s)

Zacharias Steinmetz

References

Bisdom, E., Dekker, L., & Schoute, J. (1993). Water Repellency of Sieve Fractions from Sandy Soils and Relationships with Organic Material and Soil Structure. *Geoderma* **56**, 105-118. doi:10.1016/00167061(93)90103R

See Also

`findInterval()` for the generic function.

Examples

```
bisdom(c(2,6,20,NA,3,385))
```

`calibration` *Analytical calibration functions*

Description

Defines a 'calibration' object for the calculation of concentrations from measurement signals including estimations for the limit of detection (LOD) and limit of quantification (LOQ) in accordance with DIN 32645 (2008).

Usage

```
calibration(
  formula,
  data = NULL,
  blanks = NULL,
  weights = NULL,
  model = "lm",
  check_assumptions = TRUE,
  ...
)

## S3 method for class 'calibration'
print(x, ...)

## S3 method for class 'calibration'
summary(object, ...)

## S3 method for class 'calibration'
plot(x, interval = "conf", level = 0.95, ...)

## S3 method for class 'calibration'
as.list(x, which = c("coef", "adj.r.squared", "lod", "loq", "blanks"), ...)
```

```
lod(x, ...)

## Default S3 method:
lod(x, ...)

## S3 method for class 'calibration'
lod(x, blanks = NULL, alpha = 0.01, level = 0.05, ...)

loq(x, ...)

## Default S3 method:
loq(x, ...)

## S3 method for class 'calibration'
loq(x, blanks = NULL, alpha = 0.01, k = 3, level = 0.05, maxiter = 10, ...)

## S3 method for class 'calibration'
predict(object, newdata = NULL, interval = "conf", ...)

inv_predict(x, ...)

## Default S3 method:
inv_predict(x, ...)

## S3 method for class 'calibration'
inv_predict(x, y, below_lod = NULL, method = "analytic", ...)
```

Arguments

`formula`	model formula providing the recorded signal intensities with respect to the nominal/specified analyte concentrations in the form of `signal ~ concentration` or `signal ~ concentration - 1`; model formulas are currently restricted to those forms.
`data`	an optional data frame containing the variables in the model.
`blanks`	a vector of numeric blank values overriding those automatically retrieved from calibration data.
`weights`	an optional character string containing one or more model variables, for example, in the form of `"1/concentration^0.5"` or `"1/signal"` which is internally converted to a numeric vector and passed to the fitting process of the selected model; see also `weight_select()`
`model`	model class to be used for fitting; currently, `lm()` and `rlm()` are supported.
`check_assumptions`	automatically check for normality and homoscedasticity of model residuals using `shapiro.test()` and `bptest()`, respectively; only executed if `weights == NULL`.
`...`	further arguments passed to submethods; for instance, the respective model environment such as `lm()`, `print()`, or `plot()`.

x, object	an object of class 'calibration' with a model formula as shown above.
interval	type of interval plotted (can be abbreviated); see predict() for details.
level	tolerance/confidence level; see predict() and confint() for details.
which	character vector indicating the parameters to export; defaults to c("coef", "adj.r.squared", "lod", "loq", "blanks").
alpha	numeric; error tolerance for the detection limit (critical value).
k	numeric; relative uncertainty for the limit of quantification (1/beta).
maxiter	a positive integer specifying the maximum number of iterations to calculate the LOQ.
newdata	a data frame in which to look for variables with which to predict. If NULL, values are guessed; predict.lm() for details.
y	numeric; the value to inverse predict.
below_lod	value to be assigned if inverse prediction is below LOD; defaults to "NULL" which keeps predicted values untouched. Other options may be NA or 0.
method	character indicating the method used for inverse prediction; defaults to "analytic".

Details

The LOD is defined as the lowest quantity of a substance that can be distinguished from the absence of that substance (blank value) within a given confidence level (alpha). The LOQ is defined as the lowest quantity of a substance that can be quantified/distinguished from another sample given with respect to a defined confidence level (k).

If the data supplied to calibration contain more than one blank value, namely measurements with a nominal/specified concentration of or close to zero, the LOD and LOQ are calculated from the deviation of the blank samples. This method is called "blank method" according to DIN 32645 (2008) and supposed to be more accurate than the so-called "calibration method" which will be used for the estimation of LOD and LOQ when data does not contain zero concentration measurements.

Value

calibration returns an object of class 'calibration'.

print() calls the function parameters together with the respective LOD and LOQ. summary() may be used to retrieve the summary of the underlying model. plot() plots the respective calibration curve together with the measurement values.

as.list() returns a named list.

lod() and loq() return a named vector with the LOD and LOQ together with lower and upper confidence limits.

predict() returns a data.frame of predictions.

inv_predict() predicts/calculates analyte concentrations from signal intensities.

Author(s)

Zacharias Steinmetz

References

Almeida, A.M.D., Castel-Branco, M.M., & Falcao, A.C. (2002). Linear regression for calibration lines revisited: weighting schemes for bioanalytical methods. *Journal of Chromatography B*, **774**(2), 215-222. doi:10.1016/S15700232(02)002441.

Currie, L.A. (1999). Nomenclature in evaluation of analytical methods including detection and quantification capabilities: (IUPAC Recommendations 1995). *Analytica Chimica Acta* **391**, 105-126.

DIN 32645 (2008). *Chemical analysis - Decision limit, detection limit and determination limit under repeatability conditions - Terms, methods, evaluation*. Technical standard. Deutsches Institut für Normung, Berlin.

Massart, D.L., Vandeginste, B.G., Buydens, L.M.C., Lewi, P.J., & Smeyers-Verbeke, J. (1997). *Handbook of chemometrics and qualimetrics: Part A*. Elsevier Science Inc.

See Also

`invest()` for alternative inverse prediction methods;

Other calibration: `din32645`, `icp`, `matrix_effect()`, `neitzel2003`, `phenolics`, `weight_select()`

Examples

```
data(din32645)
din <- calibration(Area ~ Conc, data = din32645)

print(din)
summary(din)
plot(din)

as.list(din)

lod(din)
loq(din)

predict(din)

inv_predict(din, 5000)
```

`clayloam` *Hydrometer readings for a clay loam*

Description

Data obtained with a 152H hydrometer in accordance with ASTM D422-63 (2007).

Format

A data frame containing 7 rows and 4 columns with information on:

time the measurement time [min]

temperature the temperature of the soil suspension [°C]

reading the hydrometer reading at the bottom of the meniscus

blank a blank value obtained in 5 g/L sodium hexametaphosphate solution (composite correction)

Author(s)

ASTM International

References

ASTM D422-63 (2007). *Standard Test Method for Particle-Size Analysis of Soils*. Technical standard. ASTM International, West Conshohocken, PA. Available from https://www.astm.org/standards/d422.

See Also

Other texture: `texture()`

`din32645` *Calibration data from DIN 32645*

Description

Sample data for the calibration of carbon in water.

Format

A data frame containing 20 rows and 2 columns with information on:

Conc nominal concentration [mg/L]

Area measurement signal

Author(s)

Deutsches Institut für Normung

References

DIN 32645 (2008). *Chemical analysis - Decision limit, detection limit and determination limit under repeatability conditions - Terms, methods, evaluation*. Technical standard. Deutsches Institut für Normung, Berlin.

See Also

Other calibration: `calibration()`, `icp`, `matrix_effect()`, `neitzel2003`, `phenolics`, `weight_select()`

icp *ICP-AES calibration data*

Description

Sample data of the calibration of silver ions in four-fold replication using inductively coupled plasma atomic emission spectroscopy (ICP-AES).

Format

A data frame containing 16 rows and 6 columns with information on:

Label sample name

Element element name

Conc nominal concentration

Units concentration units

Signal signal intensity [cps]

Replicate number of the respective replicate

Author(s)

Zacharias Steinmetz

See Also

Other calibration: `calibration()`, `din32645`, `matrix_effect()`, `neitzel2003`, `phenolics`, `weight_select()`

matrix_effect *Assess matrix effects and matrix-matched calibrations*

Description

Calculate the matrix effect by comparing the slope of a solvent-based calibration curve with one or more matrix-matched calibration. The matrix effect is expressed as signal suppression/enhancement ratio.

Usage

```
matrix_effect(object, ...)

## S3 method for class 'calibration'
matrix_effect(object, ...)
```

Arguments

object	an object of class 'calibration' obtained from analyzing standard solutions of different concentration (solvent calibration data).
...	additional objects of the same type obtained from matrix-matched calibration data.

Details

Matrix effects or signal suppression/enhancement ratios should be evaluated during analytical method development to avoid over- or underestimation of sample concentrations. In addition, signal suppression/enhancement ratios may help to justify the validity of a regular solvent calibration as opposed to matrix-matched calibrations. This may be the case if matrix effects or signal suppression/enhancement ratios are close to measurement repeatability.

Value

The magnitude of a matrix effect is estimated by subtracting the slope of a matrix-matched calibration from that of the solvent-based calibration. The difference is divided by the slope of the solvent-based calibration.

Author(s)

Julius Albert, Zacharias Steinmetz

See Also

Other calibration: calibration(), din32645, icp, neitzel2003, phenolics, weight_select()

Examples

```
data(din32645)
din <- calibration(Area ~ Conc, data = din32645)

m32645 <- din32645
m32645$Area <- din32645$Area * 1.5
matrix <- calibration(Area ~ Conc, data = m32645)

matrix_effect(din, matrix)
```

---

neitzel2003 *Calibration data from Neitzel (2003)*

Description

Artificial sample data for the verification of quantification limits.

Format

A data frame containing 20 rows and 2 columns with information on:

Conc nominal concentration

Meas measurement signal

Author(s)

Volkmar Neitzel

References

Neitzel, V. (2003). Kalibrierung bei Analysenverfahren - Bestimmungsgrenze ist nicht gleich Bestimmungsgrenze. *CLB Chemie in Labor und Biotechnik*, **54**(7), 242-246

See Also

Other calibration: `calibration()`, `din32645`, `icp`, `matrix_effect()`, `phenolics`, `weight_select()`

`phenolics` *Degradation of phenolic compounds by Steinmetz et al. (2019)*

Description

The sample data is stored in a list consisting of two `data.frame`s: a sequence table (`seq`) and a sample table (`samples`).

The sequence table contains gas-chromatography/mass spectrometry measurement data of two phenolic compounds, these are tyrosol and vanillin. Besides the samples, standard mixtures and extraction blanks were acquired in three separate analysis batches. Each measurement resulted in an integrated peak area.

The sample table describes the samples' origin from a 29-day degradation experiment, in which the phenolic compounds were either degraded in the dark by the native soil microbial community or photooxidized under UV irradiation. The samples were processed in threefold replication. Their weight [g], the volume [mL] of extract solution, and the dilution factor were recorded.

Format

A list containing two `data.frame`s.

`seq` is a data frame with 160 rows and 6 columns with information on:

Compound name of the phenolic compound

Type sample type

Batch number of the extraction batch

Name sample name

Area integrated peak area

Spec Conc specified/nominal concentration [mg/L] of standards

`samples` is a data frame with 42 rows and 9 columns with information on:

Name sample name; same as in `seq`

Day day of the incubation experiment

Lighting lighting conditions of the sample (dark/UV)

Sterilization if the sample was sterilized prior incubation

Treatment treatment name (Biogradation/Photooxidation)

Replicate replicate number

Weight sample weight [g]

Extract extract volume [mL]

Dilution dilution factor

Author(s)

Zacharias Steinmetz

References

Steinmetz, Z., Kurtz, M.P., Zubrod, J.P., Meyer, A.H., Elsner, M., & Schaumann, G.E. (2019) Biodegradation and photooxidation of phenolic compounds in soil—A compound-specific stable isotope approach. *Chemosphere* **230**, 210-218. DOI: doi:10.1016/j.chemosphere.2019.05.030.

See Also

Other calibration: `calibration()`, `din32645`, `icp`, `matrix_effect()`, `neitzel2003`, `weight_select()`

`rmse` *Root mean square error*

Description

This function computes the root mean square error (RMSE) between a vector of observed values `x` and simulated values `y`. `rel` = `FALSE` returns the absolute RMSE, `rel` = `TRUE` the relative one. If `na.rm` is `TRUE`, missing values are omitted before the computation proceeds.

Usage

```
rmse(x, y, rel = F, na.rm = T)
```

Arguments

x	a numeric vector containing observed values.
y	a numeric vector containing simulated values.
rel	logical. If TRUE, the relative RMSE is calculated, if FALSE the absolute RMSE is returned.
na.rm	logical. Should missing values be removed?

Author(s)

Zacharias Steinmetz

Examples

```
rmse(c(0.12,0.59,NA), c(0.15,0.63,1.2))
```

---

se *Standard error and confidence interval*

Description

These wrapper functions compute the standard error (SE) or the confidence interval (CI) of the values in x. If na.rm is TRUE, missing values are removed before the computation proceeds.

Usage

```
se(x, na.rm = FALSE)

CI(x, level = 0.95, na.rm = FALSE)
```

Arguments

x	a numeric vector or an R object which is coercible to one by as.vector(x, "numeric").
na.rm	logical. Should missing values be removed?
level	the confidence level required.

Author(s)

Zacharias Steinmetz

See Also

sd() for the standard deviation.

Examples

```
se(1:5)
CI(1:5)
```

`signifig` *Print significant figures*

Description

This function reports the significant figures of a given `mean` together with its respective `error` term (for instance confidence interval or standard deviation).

Usage

```
signifig(x, error, data, style = "pm", na.digit = 2, ...)
```

Arguments

`x`	a numeric vector or data frame object containing the averaged values.
`error`	a numeric vector or data frame object containing the respective error terms.
`data`	a data frame containing the specified columns. If empty, `x` and `error` need to be given as numeric vectors.
`style`	a string specifying the output style to be used. The default style `"pm"` reports the results as "3 ± 6", while `"par"` results in outputs like "0.26 (0.02)". "siunitx" returns "0.26 (2)" which might be used together with xtable for automated LaTeX table outputs.
`na.digit`	an integer controlling to which significant digit the mean value should be rounded if the error is zero or no error data was provided.
`...`	arguments passed to `prettyNum()`.

Author(s)

Zacharias Steinmetz

References

Taylor, J.R. (1997). *Error analysis: the study of uncertainties in physical measurements*. University Science Books, Sausalito, CA.

Examples

```
signifig(c(0.28, 5, -31.6, 2.6, 2, NA, 27.1),
         c(0.688, 0.8, 11.6, 9.6, NA, 1.6, 0))
```

`sorption` *Sorption isotherms*

Description

This function returns the concentration of a substance sorbed to a surface boundary after an equilibrium has established at constant temperature given the concentration(s) `x` of the dissolved substance.

Usage

```
sorption(x, par, type = "freundlich")
```

Arguments

`x`	a numeric vector containing the concentration(s) of the dissolved substance.
`par`	a numeric vector specifying the function parameters, see examples for details and correct order.
`type`	a character string indicating the type of sorption isotherm to be used: `"linear"` for the linear type, `"freundlich"` for the Freundlich isotherm, `"langmuir"` for the Langmuir isotherm, `"BET"` for the BET model according to Brunauer, Emmet, and Teller `"redlich"` for the Redlich-Peterson isotherm.

Author(s)

Zacharias Steinmetz

References

Atkins, P.W. (2001). *Physical chemistry*, Oxford University Press, Oxford.

Examples

```
sorption(1:5, par = c(Kd = 2.5), type = "linear")
sorption(1:5, par = c(K = 4, n = 0.6), type = "freundlich")
sorption(1:5, par = c(KL = 2, qmax = 10), type = "langmuir")
sorption(1:5, par = c(K = 50, qmax = 10, Csat = 10), type = "BET")
sorption(1:5, par = c(A = 30, B = 0.8), type = "redlich")
```

texture *ASTM soil texture analysis*

Description

Calculates the particle size distribution and both DIN and USDA texture classes from a series of hydrometer readings in accordance with ASTM D422-63 (2007).

Usage

```
texture(reading, ...)

## S3 method for class 'formula'
texture(formula, data = NULL, ...)

## Default S3 method:
texture(
  reading,
  blank,
  time,
  temp,
  conc = 50,
  Gs = 2.65,
  hydrometer = "auto",
  model = "auto",
  plot = F,
  ...
)

## S3 method for class 'texture'
print(x, ...)

## S3 method for class 'texture'
plot(x, ...)

as_tridata(x, ...)

## Default S3 method:
as_tridata(x, ...)

## S3 method for class 'texture'
as_tridata(x, which = NULL, ...)
```

Arguments

reading	a numeric vector of data values providing the hydrometer readings at the bottom of the meniscus.

...	further arguments to be passed to texture() (currently not used), print(), or plot().
formula	an object of class 'formula' of the form reading ~ blank + time + temp.
data	a data frame containing the variables in formula.
blank	a numeric vector containing the blank readings taken in 5 g/L sodium hexametaphosphate solution (composite correction).
time	a numeric vector containing the time passed since the beginning of the measurement in minutes.
temp	an integer vector containing the measured temperature.
conc	the concentration of the soil solution, default is 50 g/L as proposed in the ASTM guideline.
Gs	specific gravity of the suspension.
hydrometer	a character string specifying the hydrometer used; accepted values are "auto" for auto-detection (default), "151H", and "152H".
model	string is passed to drm(), "auto" chooses the best fitting model automatically.
plot	logical; if TRUE the particle size distribution is plotted.
x	an object of class 'texture'.
which	character value indicating the soil texture classification system to export; accepts "din" or "usda".

Value

texture returns an object of class 'texture'. The functions print() and plot() are available to retrieve the soil texture classes and the particle size distribution, respectively.

An object of class 'texture' is a list containing the following components:

meta Measurement meta data

distribution data frame providing the particle size distribution

model information on the fitted drm model

din Main DIN texture classes

usda Main USDA texture classes

as_tridata converts 'texture' to data.frames of a specific structure require for soiltexture-package.

Author(s)

Zacharias Steinmetz

References

ASTM D422-63 (2007). *Standard Test Method for Particle-Size Analysis of Soils*. Technical standard. ASTM International, West Conshohocken, PA. Available from https://www.astm.org/standards/d422.

See Also

Other texture: clayloam

Examples

```
data(clayloam)
texture(reading ~ blank + time + temperature, data = clayloam)
```

---

theme_publish *ggplot2 theme for scientific publications*

Description

Themes set the general aspect of the plot such as the color of the background, grid lines, the size and color of fonts. This particular theme is based on the classic dark-on-light ggplot2 theme_bw and has been used for scientific publications.

Usage

```
theme_publish(base_size = 12, base_family = "", base_linewidth = 0.25, ...)
```

Arguments

base_size	base font size
base_family	base font family
base_linewidth	base line width for ticks and axes
...	further arguments to be passed to theme_bw

Author(s)

Zacharias Steinmetz

See Also

ggtheme theme_bw

Examples

```
library(ggplot2)
p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg,
     colour = factor(gear))) + facet_wrap( ~ am)
p
p + theme_publish()
```

`weight_select` *Tools for weighted calibrations*

Description

Selecting optimum model weights by comparing sum relative errors, this is `relerr()`, of weighted `calibration()` models as suggested by Almeida et al. (2002).

Usage

```
weight_select(x, weights, ...)

## S3 method for class 'calibration'
weight_select(x, weights = NULL, ...)

relerr(x)

## S3 method for class 'calibration'
relerr(x)
```

Arguments

`x`	an object of class '`calibration`'.
`weights`	a list of weights to be added to the default weights to be checked. These are `1/concentration^0.5`, `1/concentration^1`, `1/concentration^2`, `1/signal^0.5`, `1/signal^1`, and `1/signal^2`.
`...`	further arguments passed to `calibration()`.

Details

If calibration data is not homoscedastic, a weighted least squares linear calibration model may be applied to counteract the influence of high concentrations on the regression model. This, in turn, typically improves the accuracy at the lower end of the calibration curve (Almeida et al., 2002). `weight_select` uses sum relative errors (`relerr`) to find the best weight as suggested by Almeida et al. (2002). Predefined weights include `1/concentration^0.5`, `1/concentration^1`, `1/concentration^2`, `1/signal^0.5`, `1/signal^1`, and `1/signal^2` (see `calibration()` for details).

Value

`weight_select()` produces a matrix with differently weighted '`calibration`' models ordered by sum relative errors. `relerr()` compares the nominal concentrations with those predicted by the `calibration` model.

Author(s)

Julius Albert, Kilian Kenngott, Zacharias Steinmetz

References

Almeida, A.M.D., Castel-Branco, M.M., & Falcao, A.C. (2002). Linear regression for calibration lines revisited: weighting schemes for bioanalytical methods. *Journal of Chromatography B*, **774**(2), 215-222. doi:10.1016/S15700232(02)002441.

See Also

Other calibration: `calibration()`, `din32645`, `icp`, `matrix_effect()`, `neitzel2003`, `phenolics`

Examples

```
data(din32645)
din <- calibration(Area ~ Conc, data = din32645)

weight_select(din)

relerr(din)
```

Index

∗ **calibration**
  calibration, 3
  din32645, 7
  icp, 8
  matrix_effect, 8
  neitzel2003, 9
  phenolics, 10
  weight_select, 18
∗ **data**
  clayloam, 6
  din32645, 7
  icp, 8
  neitzel2003, 9
  phenolics, 10
∗ **texture**
  clayloam, 6
  texture, 15

as.list.calibration *(*calibration*)*, 3
as_tridata *(*texture*)*, 15

bisdom, 2
bptest, *4*

calibration, 3, *7–11*, *18, 19*
CI *(*se*)*, 12
class, *5*, *16*
clayloam, 6, *17*
confint, *5*

data.frame, *10*
din32645, *6*, 7, *8–11*, *19*
drm, *16*

findInterval, *2, 3*
formula, *16*

ggtheme, *17*

icp, *6, 7*, 8, *9–11*, *19*
inv_predict *(*calibration*)*, 3
invest, *6*

lm, *4*
lod *(*calibration*)*, 3
loq *(*calibration*)*, 3

matrix_effect, *6–8*, 8, *10, 11*, *19*

neitzel2003, *6–9*, 9, *11*, *19*

phenolics, *6–10*, 10, *19*
plot, *4*
plot.calibration *(*calibration*)*, 3
plot.texture *(*texture*)*, 15
predict, *5*
predict.calibration *(*calibration*)*, 3
predict.lm, *5*
print, *4*
print.calibration *(*calibration*)*, 3
print.texture *(*texture*)*, 15

relerr *(*weight_select*)*, 18
rlm, *4*
rmse, 11

sd, *12*
se, 12
shapiro.test, *4*
signifig, 13
sorption, 14
summary.calibration *(*calibration*)*, 3

texture, *7*, 15
theme_bw, *17*
theme_publish, 17

weight_select, *4*, *6–11*, 18