

Package ‘optiRum’

July 3, 2018

Title Financial Functions & More

Description This fills the gaps credit analysts and loan modellers at Optimum Credit identify in the existing R code body. It allows for the production of documentation with less coding, replicates a number of Microsoft Excel functions useful for modelling loans (without rounding), and other helpful functions for producing charts and tables. It also has some additional scales for use, including a GBP scale.

Version 0.40.1

Encoding UTF-8

Depends R (>= 3.0.2)

Imports data.table (>= 1.9.6), ggplot2, AUC, grid, knitr, plyr, scales, stringr, XML

Suggests testthat, covr, rmarkdown

License GPL-3

LazyData true

URL <https://github.com/lockedata/optiRum>,
<https://itsalocke.com/optirum/>

BugReports <https://github.com/lockedata/optiRum/issues>

VignetteBuilder knitr

RoxygenNote 6.0.1.9000

NeedsCompilation no

Author Steph Locke [aut, cre],
Locke Data [fnd] (<https://itsalocke.com/>),
Optimum Credit Ltd's analysts [fnd] (<https://www.optimumcredit.co.uk/>),
Maëlle Salmon [ctb]

Maintainer Steph Locke <stephanie.g.locke@gmail.com>

Repository CRAN

Date/Publication 2018-07-03 19:30:06 UTC

R topics documented:

APR	2
calcNetIncome	3
CJ.dt	4
convertToXML	5
generatePDF	6
giniChart	7
giniCoef	8
logit.odd	8
logit.prob	9
multiplot	10
odd.logit	10
odd.prob	11
optiRum	12
PMT	12
pounds_format	13
prob.logit	14
prob.odd	14
PV	15
RATE	16
sanitise	17
scaledScore	17
taxYear	18
theme_optimum	19
thousands_format	19
wordwrap	20
Index	21

APR

*Calculates the compound interest rate for a loan***Description**

Based on period interest rate, number of periods, and loan amount, this function calculates the compound annual interest rate of the loan based on the monthly repayment. It calculates based on a fixed interest rate, FV=0, and charging is at the end of the period.

Usage

```
APR(nper, pmt, pv, fv = 0)
```

Arguments

nper	Number of periods - monthly
pmt	Instalment per period (should be negative)
pv	Present value i.e. loan advance (should be positive)
fV	Future value i.e. redemption amount

Value

rate The effective interest rate per year

See Also

[RATE](#)

Other finance: [PMT](#), [PV](#), [RATE](#)

Examples

```
# single set of values
APR(12,-10,110)

# vector of values
df<-data.frame(nper=c(12,24),pmt=c(-10,-10),pv=c(110,220))
APR(df$nper,df$pmt,df$pv)
```

calcNetIncome

Calculate income after tax and benefits

Description

Based on current UK taxation rules this function calculates components that subtract from gross income and provides net income.

Usage

```
calcNetIncome(persons = data.table(personID = 1:2, householdID = 1,
  employedIncome = c(15000, 40000), investmentIncome = c(0, 5000),
  nonTaxableIncome = 0, selfEmployedProfits = 0, taxCode = "1000L",
  numberOfChildren = 1, salarySacrificePercentage = c(0, 0.05), studentLoan =
  0:1), incomeGrain = "Month", financialYear = taxYear(Sys.Date()),
  modelArgs = list(model = FALSE, inflation = 1, years = 3, childBenefitChange =
  1, personalAllowanceChange = 500),
  thresholdsTable = fread(system.file("extdata", "annualthresholds.csv",
  package = "optiRum")), taxRateTable = fread(system.file("extdata",
  "annualtaxthresholds.csv", package = "optiRum")))
```

Arguments

persons	Provide the information required for calculating income, values should be provided as annual incomes
incomeGrain	Define the time period in which the income return should be expressed i.e. "Annual", "Month", "Week"
financialYear	What financial year the calculation should be performed for. Can't go back further than 2014, if you need to go back please submit a pull request on the CSVs in inst/extdata with them filled in.
modelArgs	Indicate whether a forward prediction with some changing values should be performed, and what scenario values should be used
thresholdsTable	The values needed for calculating various components
taxRateTable	The values needed for calculating Income Tax and NI (Class 1 and 4). Rate tables contain lower bound (LB), upper bound (UB) and the prevailing tax rates (Rate) at which portions of income are taxed at. $LB \geq \text{Income} < UB$

Details

Current, in the context of default values, is Tax Year 2014

Value

income Income components for each person at the relevant grain

See Also

Other tax: [taxYear](#)

CJ.dt

Cross join two data.tables

Description

The package `data.table` has a `CJ()` function which produces a `data.table` out of two vectors. This function does the Cartesian product of two `data.tables` instead.

Usage

`CJ.dt(X, Y)`

Arguments

X	A <code>data.table</code>
Y	A <code>data.table</code>

Value

dt A data.table

See Also

Other helper: [convertToXML](#), [generatePDF](#), [pounds_format](#), [sanitise](#), [thousands_format](#), [wordwrap](#)

Examples

```
library(data.table)
a <- data.table(a=1:2, b=letters[1:2])
b <- data.table(c=3:4, d=letters[3:4])
ab <- CJ.dt(a,b)
```

convertToXML

Produce an XML document of a table

Description

Produce a document containing all data.table or data.frame rows

Usage

```
convertToXML(data, name = "doc")
```

Arguments

data	The data to be converted
name	The toplevel node name

Details

Code was taken from <https://stat.ethz.ch/pipermail/r-help/2010-February/228025.html> and amended, basic tests applied

Value

xml An XML object

See Also

Other helper: [CJ.dt](#), [generatePDF](#), [pounds_format](#), [sanitise](#), [thousands_format](#), [wordwrap](#)

Examples

```
df<-data.frame(nper=c(12,24),pmt=c(-10,-10),pv=c(110,220))
xml<-convertToXML(df,name='examples')
```

 generatePDF

 Convert an *.Rnw* file to a PDF

Description

This function is designed to handle the production task of a 'standard' PDF process. It is designed to build using pdflatex (unless otherwise specified) an adequate number of times to enable full typesetting to occur after taking into account items like contents pages, glossaries, and figures.

Usage

```
generatePDF(srcpath = getwd(), srcname, destpath = getwd(),
  destname = srcname, DATED = FALSE, CLEANUP = TRUE, QUIET = FALSE,
  envir = new.env(parent = .GlobalEnv), ...)
```

Arguments

srcpath	Location of <i>.Rnw</i> file, default is current directory
srcname	<i>Rnw</i> file name without extension e.g. 'Style'
destpath	Location of PDF file to be sent to, default is current directory
destname	PDF file name without extension e.g. 'Style_output'
DATED	Boolean indicating whether PDF filename should include <i>yyyymmdd</i> added to it
CLEANUP	Boolean indicating whether ancilliary files should be removed after production
QUIET	Boolean indicating whether console output should be limited
envir	Set default environment for knitr to run in - prevents a <i>data.table</i> issue
...	Allows additional parameters to be passed to the <i>knit2pdf</i> function

See Also

`knit2pdf`

Other helper: [CJ.dt](#), [convertToXML](#), [pounds_format](#), [sanitise](#), [thousands_format](#), [wordwrap](#)

Examples

```
## Not run:
# simple call
generatePDF(srcname='basic')

# complex call
generatePDF(
  srcname='basic',
  destpath=getwd(),
  destname='basic',
  DATED=TRUE,
```

```
CLEANUP=FALSE,  
QUIET=TRUE,  
compiler='xelatex')  
  
## End(Not run)
```

giniChart

Produce a ROC curve with gini coefficient title

Description

This function uses ggplot to produce a themed Receiver Operator Curve and calculates a Gini coefficient based on it.

Usage

```
giniChart(pred, act)
```

Arguments

pred	Logit/scores/probabilities to be compared against actuals
act	This should be a column containing outcomes in a boolean form either as a factor or number

See Also

AUC roc [giniCoef](#)

Other creditrisk: [giniCoef](#), [logit.odd](#), [logit.prob](#), [odd.logit](#), [odd.prob](#), [prob.logit](#), [prob.odd](#), [scaledScore](#)

Examples

```
sampledata<- data.frame(val= rnorm(100) , outcome=rbinom(100,1,.8))  
giniChart(sampledata$val,sampledata$outcome)
```

giniCoef	<i>Produce a gini coefficient</i>
----------	-----------------------------------

Description

This function calculates a Gini coefficient based on a Receiver Operator Curve.

Usage

```
giniCoef(pred, act)
```

Arguments

pred	Logit/scores/probabilities to be compared against actuals
act	This should be a column containing outcomes in a boolean form either as a factor or number

Value

gini The coefficient

See Also

AUC roc [giniChart](#)

Other creditrisk: [giniChart](#), [logit.odd](#), [logit.prob](#), [odd.logit](#), [odd.prob](#), [prob.logit](#), [prob.odd](#), [scaledScore](#)

Examples

```
sampladata<- data.frame(val= rnorm(100) , outcome=rbinom(100,1,.8))
giniCoef(sampladata$val,sampladata$outcome)
```

logit.odd	<i>Convert a logit to odds</i>
-----------	--------------------------------

Description

Transform a logit response from a glm into odds

Usage

```
logit.odd(logit)
```


Arguments

logit The log(odds)

Value

odds Odds

See Also

[logit.prob](#)

Other creditrisk: [giniChart](#), [giniCoef](#), [logit.prob](#), [odd.logit](#), [odd.prob](#), [prob.logit](#), [prob.odd](#), [scaledScore](#)

Examples

```
logit.odd(0) # equals 1
```

logit.prob	<i>Convert a logit to probability</i>
------------	---------------------------------------

Description

Transform a logit response from a glm into probability

Usage

```
logit.prob(logit)
```

Arguments

logit The log(odds)

Value

prob Probability

See Also

[logit.odd](#) [odd.prob](#)

Other creditrisk: [giniChart](#), [giniCoef](#), [logit.odd](#), [odd.logit](#), [odd.prob](#), [prob.logit](#), [prob.odd](#), [scaledScore](#)

Examples

```
logit.prob(0) # equals 0.5
```

multiplot *Multiple plot function*

Description

Multiplot allows the laying out of multiple charts in a custom layout

Usage

```
multiplot(..., plotlist = NULL, cols = 1, layout = NULL)
```

Arguments

...	ggplot objects can be passed in ...
plotlist	a list of ggplot objects
cols	Number of columns in layout
layout	A matrix specifying the layout. If present, 'cols' is ignored

Details

If the layout is something like `matrix(c(1,2,3,3), nrow=2, byrow=TRUE)`, then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom.

Code is taken as-is from [http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_\(ggplot2\)/](http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/) and no tests are maintained for it at present

See Also

Other visualisation: [theme_optimum](#)

odd.logit *Convert an odd into a logit*

Description

Transforming odds into logits (the response from binomial glms)

Usage

```
odd.logit(odds)
```

Arguments

odds	Odds
------	------

Value

logit Log(odds)

See Also

[logit.odd](#) [logit.prob](#)

Other creditrisk: [giniChart](#), [giniCoef](#), [logit.odd](#), [logit.prob](#), [odd.prob](#), [prob.logit](#), [prob.odd](#), [scaledScore](#)

Examples

```
odd.logit(1) # equals 0
```

odd.prob	<i>Convert an odds to probability</i>
----------	---------------------------------------

Description

Transform odds into a probability

Usage

```
odd.prob(odds)
```

Arguments

odds Odds

Value

prob Probability

See Also

[logit.odd](#) [logit.prob](#)

Other creditrisk: [giniChart](#), [giniCoef](#), [logit.odd](#), [logit.prob](#), [odd.logit](#), [prob.logit](#), [prob.odd](#), [scaledScore](#)

Examples

```
odd.prob(1) # equals 0.5
```

optiRum *optiRum is a helper package*

Description

optiRum is a growing package of utilities created by Optimum Credit Ltd's analysts. It is designed to provide convenience functions, standards, and useful snippets. Optimum Credit derives significant value from the R platform and associated community, so non-commercially sensitive functionality is made available in the spirit of reciprocity.

PMT *Calculates the repayment for a loan*

Description

Based on period interest rate, number of periods, and loan amount, this function calculates the repayment of the loan such that it would be paid off fully at the end of the loan. This function is designed to be equivalent to the Excel function PMT. It calculates based on a fixed interest rate, FV=0, and charging is at the end of the period. Response is rounded to 2dp

Usage

```
PMT(rate, nper, pv)
```

Arguments

rate	The nominal interest rate per period (should be positive)
nper	Number of periods
pv	Present value i.e. loan advance (should be positive)

Value

pmt Instalment per period (should be negative)

See Also

[PV RATE](#)

Other finance: [APR](#), [PV](#), [RATE](#)

Examples

```
PMT(0.1,12,3000) # ==-440.29 taken from excel
```

```
df<-data.frame(rate=c(.1, .2),nper=c(12,24),pv=c(3000,1000))
PMT(df$rate,df$nper,df$pv) # ==-440.29,-202.55 taken from excel
```

pounds_format	<i>Currency formatter: round to nearest penny and display pounds sign.</i>
---------------	--

Description

The returned function will format a vector of values as currency. Values are rounded to the nearest penny, and pennies are displayed if any of the values has a non-zero pennies and the largest value is less than `largest_with_penny` which by default is 100000.

Usage

```
pounds_format(x, largest_with_penny = 1e+05)
```

```
pounds(x)
```

Arguments

`x` a numeric vector to format
`largest_with_penny` the value that all values of `x` must be less than in order for the cents to be displayed

Details

Based heavily on the scales work by Hadley

Value

a function with single parameter `x`, a numeric vector, that returns a character vector

See Also

Other helper: [CJ.dt](#), [convertToXML](#), [generatePDF](#), [sanitise](#), [thousands_format](#), [wordwrap](#)

Examples

```
pounds_format()(c(100, 0.23, 1.456565, 2e3))  
pounds_format()(c(1:10 * 10))  
pounds(c(100, 0.23, 1.456565, 2e3))  
pounds(c(1:10 * 10))  
pounds(10^(1:8))
```

prob.logit	<i>Convert a probability into a logit</i>
------------	---

Description

Transforming probabilities into logits (the response from binomial glms)

Usage

```
prob.logit(prob)
```

Arguments

prob	Probability
------	-------------

Value

logit Log(odds)

See Also

[prob.odd](#) [odd.logit](#)

Other creditrisk: [giniChart](#), [giniCoef](#), [logit.odd](#), [logit.prob](#), [odd.logit](#), [odd.prob](#), [prob.odd](#), [scaledScore](#)

Examples

```
prob.logit(0.5) # equals 0
```

prob.odd	<i>Convert a probability into odds probability</i>
----------	--

Description

Transform probabilities into odds

Usage

```
prob.odd(prob)
```

Arguments

prob	Probability
------	-------------

Value

odds Odds

See Also

[prob.logit](#) [odd.logit](#)

Other creditrisk: [giniChart](#), [giniCoef](#), [logit.odd](#), [logit.prob](#), [odd.logit](#), [odd.prob](#), [prob.logit](#), [scaledScore](#)

Examples

```
prob.odd(0.5) # equals 1
```

PV	<i>Calculates the present value</i>
----	-------------------------------------

Description

Based on period interest rate, number of periods, and instalment, this function calculates the present value of the loan such that it would be paid off fully at the end of the loan. This function is designed to be equivalent to the Excel function PV. It calculates based on a fixed interest rate, FV=0 and charging is at the end of the period. Response is rounded to 2dp

Usage

```
PV(rate, nper, pmt, fv = 0)
```

Arguments

rate	The nominal interest rate per period (should be positive)
nper	Number of periods
pmt	Instalment per period (should be negative)
fv	Future value i.e. redemption amount

Value

pv Present value i.e. loan advance (should be positive)

See Also

[PMT RATE](#)

Other finance: [APR](#), [PMT](#), [RATE](#)

Examples

```
PV(0.1,12,-10) # 68.14 Taken from excel
```

```
df<-data.frame(rate=c(.1,.1),nper=c(12,24),pmt=c(-10,-15))
PV(df$rate,df$nper,df$pmt) # c(68.14,134.77) Taken from excel
```

RATE	<i>Calculates compounded interest rate</i>
------	--

Description

Based on loan term, instalment, and the loan amount, this function calculates the associated compound interest rate. This function is designed to be equivalent to the Excel function RATE. It calculates a fixed interest rate.

Usage

```
RATE(nper, pmt, pv, fv = 0)
```

Arguments

nper	Number of periods
pmt	Instalment per period (should be negative)
pv	Present value i.e. loan advance (should be positive)
fv	Future value i.e. redemption amount

Value

rate The corresponding compound interest rate required to arrive at an FV of 0

See Also

[PMT](#) [PV](#)

Other finance: [APR](#), [PMT](#), [PV](#)

Examples

```
RATE(12, -500, 3000) # 0.126947 Taken from excel
```

```
df<-data.frame(nper=c(12,12),pmt=c(-500,-400),pv=c(3000,3000))  
RATE(df$nper,df$pmt,df$pv) # c(0.126947,0.080927) Taken from excel
```

sanitise	<i>A cleaning function for special characters</i>
----------	---

Description

This function is a helper for cleaning xtable outputs in preparation for PDF production

Usage

```
sanitise(str)
```

Arguments

str	The text to be sanitised
-----	--------------------------

See Also

Other helper: [CJ.dt](#), [convertToXML](#), [generatePDF](#), [pounds_format](#), [thousands_format](#), [wordwrap](#)

Examples

```
sanitise('&##<>\\')
```

scaledScore	<i>Produce a scaled score based on a logit</i>
-------------	--

Description

This function takes a logit and scales based on an intercept and doubling of odds ratio

Usage

```
scaledScore(logit, offset = 300, scale = 50)
```

Arguments

logit	Logit to be scaled
offset	Midrange, default is 300
scale	Value in which odds are double, default is 50

See Also

[glm](#)

Other creditrisk: [giniChart](#), [giniCoef](#), [logit.odd](#), [logit.prob](#), [odd.logit](#), [odd.prob](#), [prob.logit](#), [prob.odd](#)

Examples

```
scaledScore(0) # 300
scaledScore(0,offset=600) # 600
```

taxYear	<i>Returns the UK financial tax year for a given date</i>
---------	---

Description

Based on UK tax year April 6 - April 5, this returns the year (YYYY) the tax period covers. Tax Year start date can be overridden.

Usage

```
taxYear(date = Sys.Date(), start = "04-06")
```

Arguments

date	Date to be checked
start	Provide the month & day that will be used as the first tax day (mm-dd)

Value

year The financial year

See Also

Other tax: [calcNetIncome](#)

Examples

```
# single set of values
taxYear(Sys.Date())

# vector of values
taxYear(seq(Sys.Date(),by=1,length=500))
```

theme_optimum	<i>Produce an Optimum-standard base chart</i>
---------------	---

Description

This theme no longer builds on the Stephen Few theme from ggthemes, but now produces a chart without an enclosing box, to produce a good baseline for charting in R. Gets called as would any typical theme.

Usage

```
theme_optimum(base_size = 14, base_family = "")
```

Arguments

base_size	Anchor font size
base_family	Font family to use

See Also

Other visualisation: [multiplot](#)

Examples

```
library(ggplot2)
ggplot(data.frame(x=1:10,y=1:10),aes(x,y))+theme_optimum()+geom_line()
```

thousands_format	<i>Thousands formatter: format number with commas separating the number thousands and suffixed with a k.</i>
------------------	--

Description

Based heavily on the scales work by Hadley

Usage

```
thousands_format()
```

```
thousands(x)
```

Arguments

x	a numeric vector to format
---	----------------------------

Value

a function with single parameter `x`, a numeric vector, that returns a character vector

See Also

Other helper: [CJ.dt](#), [convertToXML](#), [generatePDF](#), [pounds_format](#), [sanitise](#), [wordwrap](#)

Examples

```
thousands_format()(c(1, 1e3, 2000, 1e6))
thousands_format()(c(1, 1e3, 2000, 1e6))
thousands(c(1, 1e3, 2000, 1e6))
```

wordwrap

Produce a string with one word per line

Description

Designed for splitting strings to fit better on axis on charts

Usage

```
wordwrap(x, ...)
```

Arguments

<code>x</code>	string
<code>...</code>	Allows additional parameters to be passed to <code>gsub</code>

See Also

Other helper: [CJ.dt](#), [convertToXML](#), [generatePDF](#), [pounds_format](#), [sanitise](#), [thousands_format](#)

Examples

```
library('ggplot2')
names <- wordwrap(c("This is a name", "Single"))
ggplot(data.frame(x=names, y=1:10), aes(x, y))+theme_optimum()+geom_line()
```

Index

- *Topic **AUROC**
 - [giniChart](#), 7
 - [giniCoef](#), 8
- *Topic **CJ**
 - [CJ.dt](#), 4
- *Topic **PDF**
 - [generatePDF](#), 6
 - [sanitise](#), 17
- *Topic **Rnw**
 - [generatePDF](#), 6
 - [sanitise](#), 17
- *Topic **XML**
 - [convertToXML](#), 5
- *Topic **apr**
 - [APR](#), 2
- *Topic **charts**
 - [multiplot](#), 10
- *Topic **data.frame**
 - [convertToXML](#), 5
- *Topic **data.table**
 - [CJ.dt](#), 4
 - [convertToXML](#), 5
- *Topic **financial**
 - [APR](#), 2
 - [calcNetIncome](#), 3
 - [PMT](#), 12
 - [PV](#), 15
 - [RATE](#), 16
 - [taxYear](#), 18
- *Topic **generate**
 - [generatePDF](#), 6
 - [sanitise](#), 17
- *Topic **ggplot2**
 - [multiplot](#), 10
 - [theme_optimum](#), 19
- *Topic **ggplot**
 - [wordwrap](#), 20
- *Topic **gini**
 - [giniChart](#), 7
 - [giniCoef](#), 8
- *Topic **glm**
 - [logit.odd](#), 8
 - [logit.prob](#), 9
 - [odd.logit](#), 10
 - [odd.prob](#), 11
 - [prob.logit](#), 14
 - [prob.odd](#), 14
 - [scaledScore](#), 17
- *Topic **income**
 - [calcNetIncome](#), 3
- *Topic **knitr**
 - [generatePDF](#), 6
 - [sanitise](#), 17
- *Topic **logit**
 - [logit.odd](#), 8
 - [logit.prob](#), 9
 - [odd.logit](#), 10
 - [odd.prob](#), 11
 - [prob.logit](#), 14
 - [prob.odd](#), 14
 - [scaledScore](#), 17
- *Topic **multiple**
 - [multiplot](#), 10
- *Topic **odds**
 - [logit.odd](#), 8
 - [logit.prob](#), 9
 - [odd.logit](#), 10
 - [odd.prob](#), 11
 - [prob.logit](#), 14
 - [prob.odd](#), 14
- *Topic **pdflatex**
 - [generatePDF](#), 6
 - [sanitise](#), 17
- *Topic **pmt**
 - [APR](#), 2
 - [PMT](#), 12
 - [PV](#), 15
 - [RATE](#), 16

*Topic **probability**

logit.prob, 9
 odd.logit, 10
 odd.prob, 11
 prob.logit, 14
 prob.odd, 14

*Topic **pv**

APR, 2
 PMT, 12
 PV, 15
 RATE, 16

*Topic **rate**

RATE, 16

*Topic **roc**

giniChart, 7
 giniCoef, 8

*Topic **score**

scaledScore, 17

*Topic **tax**

calcNetIncome, 3
 taxYear, 18

*Topic **theme**

theme_optimum, 19

*Topic **xml**

convertToXML, 5

APR, 2, 12, 15, 16

calcNetIncome, 3, 18

CJ.dt, 4, 5, 6, 13, 17, 20

convertToXML, 5, 5, 6, 13, 17, 20

generatePDF, 5, 6, 13, 17, 20

giniChart, 7, 8, 9, 11, 14, 15, 17

giniCoef, 7, 8, 9, 11, 14, 15, 17

glm, 17

logit.odd, 7, 8, 8, 9, 11, 14, 15, 17

logit.prob, 7-9, 9, 11, 14, 15, 17

multiplot, 10, 19

odd.logit, 7-9, 10, 11, 14, 15, 17

odd.prob, 7-9, 11, 11, 14, 15, 17

optiRum, 12

optiRum-package (optiRum), 12

PMT, 3, 12, 15, 16

pounds (pounds_format), 13

pounds_format, 5, 6, 13, 17, 20

prob.logit, 7-9, 11, 14, 15, 17

prob.odd, 7-9, 11, 14, 14, 17

PV, 3, 12, 15, 16

RATE, 3, 12, 15, 16

sanitise, 5, 6, 13, 17, 20

scaledScore, 7-9, 11, 14, 15, 17

taxYear, 4, 18

theme_optimum, 10, 19

thousands (thousands_format), 19

thousands_format, 5, 6, 13, 17, 19, 20

wordwrap, 5, 6, 13, 17, 20, 20