Angle arithmetic: **angleR**

A. Blejec

May 1, 2012

Abstract

A package to perform angle manipulations in degrees, minutes and seconds. Formatting, conversions and arithmetic functions are included. Might be useful for specialists in geodesy and geometry.

Contents

[cp1250]inputenc

```
> rm(list=ls(all=TRUE))
> newAngle <- function(...) NULL
> as.degs <- function(...) NULL
> as.mins <- function(...) NULL
> as.secs <- function(...) NULL
```

1 New angle

```
> normalize.angle <- function(x){
+ x <- unclass(x)
+ x <- x[1]+x[2]/60+x[3]/3600
+ st <- floor(x)
+ min <- floor((x-st)*60)
+ sek <- (x-st-min/60)*3600
+ return(c(st,min,sek))
+ }
> normalize.angle(c(10,10,10))
```

```
[1] 10 10 10
```

```
> normalize.angle(c(10,70,71))
```

```
[1] 11 11 11
```

```
> alfa <- c(10,-20,-30)
> normalize.angle(alfa)
```

```
[1]  9 39 30
```

```
> round(normalize.angle(c(0,20,30))-normalize.angle(c(0,-20,-30)))
```

```
[1]   1 -19   0
```

```
> newAngle <- function(d=0,m=0,s=0){
+ z <- sign(d)*(abs(d)+m/60+s/3600)
+ z <- structure(z,class=c("angle"))
+ return(z)
+ }
> str(newAngle(10,20,30))
```

```
Class 'angle'  num 10.3
```

```
> unclass(newAngle(10,70,70))
```

```
[1] 11.18611
```

```
> newAngle(-10,20,30)
```

```
[1] -10.34167
attr(,"class")
[1] "angle"
```

```
> newAngle(0,0,0)
```

```
[1] 0
attr(,"class")
[1] "angle"
```

2 Conversions

```
> as.degs <- function(x){
+ if(inherits(x,"numeric")) return(structure(x,class=c("angle"),units="degs"))
+ if(inherits(x,"angle")) return(structure(x,class="angle",units="degs"))
+
+ }
> alfa <- newAngle(30,30,10)
> alfa
```

```
[1] 30.50278
attr(,"class")
[1] "angle"
```

```
> as.degs(alfa)
```

```
[1] 30.50278
attr(,"class")
[1] "angle"
attr(,"units")
[1] "degs"
```

```
> as.degs(-alfa)
```

```
[1] -30.50278
attr(,"class")
[1] "angle"
attr(,"units")
[1] "degs"
```

```
> as.degs(30.5)
```

```
[1] 30.5
attr(,"class")
[1] "angle"
attr(,"units")
[1] "degs"
```

```
> as.degs(newAngle(0,0,0))
```

```
[1] 0
attr(,"class")
[1] "angle"
attr(,"units")
[1] "degs"
```

```
> as.mins <- function(x){
+ if(inherits(x,"numeric")) return(structure(x/60,class=c("angle"),units="mins
+ if(inherits(x,"angle")) return(structure(x,class="angle",units="mins"))
+ }
> alfa <- newAngle(30,30,10)
> alfa
```

```
[1] 30.50278
attr(,"class")
[1] "angle"
```

```
> as.mins(alfa)
```

```
[1] 30.50278
attr(,"class")
[1] "angle"
attr(,"units")
[1] "mins"
```

```
> as.secs <- function(x){
+ if(inherits(x,"numeric")) return(structure(x/3600,class="angle",units="secs"
+ if(inherits(x,"angle"))return(structure(x,class="angle",units="secs"))
+ }
> alfa <- newAngle(30,30,10.1)
> alfa
```

```
[1] 30.50281
attr(,"class")
[1] "angle"
```

```
> as.secs(alfa)
```

```
[1] 30.50281
attr(,"class")
[1] "angle"
attr(,"units")
[1] "secs"
```

```
> beta <- as.secs(1)
> beta
```

```
[1] 0.0002777778
attr(,"class")
[1] "angle"
attr(,"units")
[1] "secs"
```

```
> unclass(beta)
```

```
[1] 0.0002777778
attr(,"units")
[1] "secs"
```

```
> as.angle <- function(x){
+ if(inherits(x,"numeric")) class(x) <- "angle"
+ return(x)
+ }
> as.angle(10.5)
```

```
[1] 10.5
attr(,"class")
[1] "angle"
```

```
> is.degs <- function(x) attr(x,"units")=="degs"
> is.mins <- function(x) attr(x,"units")=="mins"
> is.secs <- function(x) attr(x,"units")=="secs"
>
```

```
> c.angle <- function (..., recursive = FALSE)
+ structure(c(unlist(lapply(list(...), unclass))), class = "angle")
> #
> alfa
```

```
[1] 30.50281
attr(,"class")
[1] "angle"
```

```
> beta
```

```
[1] 0.0002777778
attr(,"class")
[1] "angle"
attr(,"units")
[1] "secs"
```

```
> str(c(alfa,beta))
```

```
Class 'angle'  num [1:2] 3.05e+01 2.78e-04
```

```
> strsign.angle <- function(...) sign(unclass(...))
> #
> sign(c(as.angle(-10.5),as.angle(10)))
```

```
[1] -1  1
attr(,"class")
[1] "angle"
```

3 Format and print

3.1 zfill()

Formatiranje celih števil na nekaj mest z vodilnimi ničlami

```
> zfill <- function(x,digits=3){
+ #nd <- 0
+ #if(x>1) nd <- floor(log10(x))
+ #z <- paste(c(rep("0",digits-nd-1),x),collapse="")
+ z <- sprintf(paste("%0",digits,"d",sep=""),x)
+ return(z)
+ }
> sprintf("%04d",15)
```

```
[1] "0015"
```

```
> #
> zfill(9)
```

```
[1] "009"
```

```
> zfill(0)
```

```
[1] "000"
```

```
> zfill(30,2)
```

```
[1] "30"
```

```
> format.angle <- function(x,units=NULL,m.small=1,s.dec=0,sep="",collapse=NULL
+ if(!is.null(units)) attr(x,"units") <- units
+ if(is.null(attr(x,"units"))) {
+ sign <- c("-"," ","+")[2+sign(x)]
+ x <- abs(x)
+ d <- floor(x)
+ m <- floor((x-d)*60)
+ s <- (x-d-m/60)*3600
+ return(paste(sign,zfill(d,3),dd,zfill(m,2),mm,
+ round(s,m.small),ss,sep=sep,collapse=collapse))
+ }
+ if(attr(x,"units")=="dms") {
+ sign <- c("-"," ","+")[2+sign(x)]
+ x <- abs(x)
+ d <- floor(x)
+ m <- floor((x-d)*60)
+ s <- round((x-d-m/60)*36000)
+ return(paste(sign,zfill(d,3),zfill(m,2),
+ zfill(s,3),sep=sep,collapse=collapse))
+ }
+ if(attr(x,"units")=="degs") return(
+ paste(round(x,m.small),"°",sep=sep,collapse=collapse))
+ if(attr(x,"units")=="mins") return(
+ paste(round(x*60,m.small),"'",sep=sep,collapse=collapse))
+ if(attr(x,"units")=="secs") return(
+ paste(round(x*3600,m.small),"''",sep=sep,collapse=collapse))
+ }
> alfa <- newAngle(30,20,10.1)
> beta <- newAngle(50,40,1.1)
> format(alfa)
[1] "+030° 20' 10.1''"
> format(structure(c(alfa,-beta),class="angle",units="dms"))
[1] "+03020101" "-05040011"
> format(alfa,"mins")
[1] "1820.2'"
> format(as.degs(alfa))
[1] "30.3°"
> format(as.mins(alfa))
[1] "1820.2'"
> format(as.secs(alfa))
[1] "109210.1''"
> format(alfa,"dms")
[1] "+03020101"
```

```
> print.angle <- function(x){
+ if(inherits(x,"angle")) print(noquote(format(x)))
+ }
> alfa <- newAngle(30,20,10)
> print(alfa)
[1] +030° 20' 10''
> alfa
[1] +030° 20' 10''
> as.degs(alfa)
[1] 30.3°
> as.mins(alfa)
[1] 1820.2'
> as.secs(alfa)
[1] 109210''
> beta <- newAngle(50,40,1.1)
> print(c(alfa,-beta))
[1] +030° 20' 10''  -050° 40' 1.1''
```

4 Arithmetic

Angles are internally saved as decimal degrees, so usual arithmetic functions apply!

4.1 `*.angle()`

```
> (alfa <- newAngle(30,10,20))
[1] +030° 10' 20''
> (beta <- newAngle(20,20,50))
[1] +020° 20' 50''
> beta*2
[1] +040° 41' 40''
> beta+beta
[1] +040° 41' 40''
> 2*beta
[1] +040° 41' 40''
> beta*0.5
[1] +010° 10' 25''
```

4.2 +.angle()

```
> (alfa <- newAngle(30,10,20))
[1] +030° 10' 20''
> (beta <- newAngle(20,20,50))
[1] +020° 20' 50''
> alfa+beta
[1] +050° 31' 10''
> as.degs(alfa)
[1] 30.2°
```

4.3 -.angle()

```
> (alfa <- newAngle(30,10,20))
[1] +030° 10' 20''
> (beta <- newAngle(20,20,50))
[1] +020° 20' 50''
> -beta
[1] -020° 20' 50''
> alfa-beta
[1] +009° 49' 30''
> alfa-alfa
[1]  000° 00' 0''
> beta+beta
[1] +040° 41' 40''
> 2*beta
[1] +040° 41' 40''
```

4.4 /.angle()

```
> (alfa <- newAngle(30,15,25))
[1] +030° 15' 25''
> (beta <- newAngle(20,25,55))
[1] +020° 25' 55''
> beta/2
[1] +010° 12' 57.5''
> beta/beta
[1] +001° 00' 0''
> alfa/beta
[1] +001° 28' 51.1''
> beta/0.5
[1] +040° 51' 50''
> newAngle(180,0,0)/newAngle(45,0,0)
[1] +004° 00' 0''
> newAngle(180,0,0)-3*newAngle(45,0,0)
[1] +045° 00' 0''
> newAngle(180,0,0)-2*newAngle(45,0,0)
[1] +090° 00' 0''
> newAngle(180,0,0)-2*newAngle(445,0,0)
[1] -710° 00' 0''
> newAngle(180,0,0)+2*newAngle(445,0,0)
[1] +1070° 00' 0''
```

5 Calculations

```
> alfa
[1] +030° 15' 25''
> beta
[1] +020° 25' 55''
> alfa-beta+2*(beta/2)-alfa
[1]  000° 00' 0''
> beta-2*(beta/2)
[1]  000° 00' 0''
> alfa-alfa
[1]  000° 00' 0''
> -alfa
[1] -030° 15' 25''
> -alfa+newAngle(0,0,0)
[1] -030° 15' 25''
> -alfa+alfa
[1]  000° 00' 0''
> -alfa+beta-2*(beta/2)
[1] -030° 15' 25''
> -alfa+beta-2*(beta/2)+alfa
[1]  000° 00' 0''
> alfa+beta
[1] +050° 41' 20''
> alfa-beta
[1] +009° 49' 30''
> beta+alfa
[1] +050° 41' 20''
> 7*alfa
[1] +211° 47' 55''
```

SessionInfo

Windows 7 x64 (build 7601) Service Pack 1

- R version 2.14.1 (2011-12-22), `x86_64-pc-mingw32`
- Locale: `LC_COLLATE=Slovenian_Slovenia.1250`, `LC_CTYPE=Slovenian_Slovenia.1250`, `LC_MONETARY=Slovenian_Slovenia.1250`, `LC_NUMERIC=C`, `LC_TIME=Slovenian_Slovenia.1250`
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, utils
- Other packages: Hmisc 3.9-2, patchDVI 1.8.1584, survival 2.36-10
- Loaded via a namespace (and not attached): cluster 1.14.1, grid 2.14.1, lattice 0.20-0, tools 2.14.1