# Package ‘munsellinterpol’

July 20, 2019

**Type** Package

**Title** Interpolate Munsell Renotation Data from Hue/Chroma to CIE/RGB

**Version** 2.5-1

**Encoding** UTF-8

**Date** 2019-07-20

**Description**
Methods for interpolating data in the Munsell color system following the ASTM D-1535 standard. Hues and chromas with decimal values can be interpolated and converted to/from the Munsell color system and CIE xyY, CIE XYZ, CIE Lab, CIE Luv, or RGB. Includes ISCC-NBS color block lookup. Based on the work by Paul Centore, ``The Munsell and Kubelka-Munk Toolbox''.

**License** GPL (>= 3)

**LazyLoad** yes

**LazyData** yes

**Depends** R (>= 3.2.0)

**Imports** rootSolve, spacesRGB, spacesXYZ

**Suggests** microbenchmark, mgcv, knitr, rmarkdown, flextable

**Author** Jose Gama [aut, trl],
Paul Centore [aut, cph],
Glenn Davis [aut, cre]

**Maintainer** Glenn Davis `<gdavis@gluonics.com>`

**Repository** CRAN

**NeedsCompilation** no

**VignetteBuilder** knitr

**BuildVignettes** yes

**Date/Publication** 2019-07-20 04:30:05 UTC

# R topics documented:



---

`CentroidsISCCNBS` *Centroid Notations for the Revised ISCC-NBS Color-Name Blocks*

---

### Description

`CentroidsISCCNBS` is a table with the centroids of the revised ISCC-NBS Color-Name Blocks.

### Format

This `data.frame` has 267 rows and these columns:

**Number** ISCC-NBS number (an integer from 1 to 267)

**Name** ISCC-NBS name

**MunsellSpec** Munsell specification of the centroid of the block a (character string)

### Details

The earliest paper I am aware of is by *Nickerson, et. al.* in 1941. After the big Munsell renotation in 1943, the name blocks were revised in 1955. When the central colors were recomputed in *Kelly (1958)*, they were called the "Central Colors", though the text makes it clear that most are truly centroids, which were computed from the centroid of an "elementary shape", which is a "sector of a right cylindrical annulus". For the "peripheral blocks" of high chroma, the centroids were "estimated graphically by plotting the MacAdam limits".
In *Kelly (1965)* these were called "centroid colors", and that is the name we will use here.

### Contributor

Glenn Davis

### References

Nickerson, Dorothy and Sidney M. Newhall. **Central Notations for ISCC-NBS Color names**. J Opt. Soc. Am. Vol 31 Iss. 9. pp. 597-591. 1941.

Newhall, Sidney M., Dorothy Nickerson, Deane B. Judd. **Final Report of the O.S.A. Subcommitte on the Spacing of the Munsell Colors**. Journal of the Optical Society of America. Vol. 33. No. 7. pp. 385-418. July 1943.

Kelly, Kenneth L. and Deane B. Judd **The ISCC-NBS Method of Designating Colors and a Dictionary of Color Names**. National Bureau of Standards Circular 553. Washington DC: US Government Printing Office. November 1, 1955.

Kelly, Kenneth Low. **Central Notations for the Revised ISCC-NBS Color-Name Blocks**. Journal of Research of the National Bureau of Standards. Research Paper 2911. Vol. 61 No. 5. pp. 427-431. November 1958.

Kelly, Kenneth Low. **A Universal Color Language**. Color Engineering. Vol. 3 No. 2. pp. 2-7. March-April, 1965.

### Examples

```
print( CentroidsISCCNBS[ 1:5, ] )

##   Number          Name   MunsellSpec
## 1      1    vivid pink     1.5R 7/13
## 2      2   strong pink  1.5R 7.5/9.1
## 3      3     deep pink 1.9R 6.0/11.1
## 4      4    light pink  2.5R 8.6/5.2
## 5      5 moderate pink  2.5R 7.2/5.2
```

---

`ColorBlockFromMunsell` *Get ISCC-NBS Number and ISCC-NBS Name from Munsell Hue, Value, and Chroma*

---

## Description

Get ISCC-NBS Number and ISCC-NBS Name from Munsell Hue, Value, and Chroma.

## Usage

```
ColorBlockFromMunsell( MunsellSpec )
```

## Arguments

| | |
|---|---|
| `MunsellSpec` | a numeric Nx3 matrix or a vector that can be converted to such a matrix. Each row has Munsell HVC, where H is Hue Number, and V and C are the standard Munsell Value and Chroma. The Hue is automatically wrapped to the interval (0,100]. `MunsellSpec` can also be a character N-vector with standard Munsell notation; it is converted to an Nx3 matrix. |

## Details

The ISCC-NBS System is a partition of Munsell Color Solid into 267 color blocks. Each block is a disjoint union of *elementary blocks*, where an elementary block is defined by its minimum and maximum limits in Hue, Value, and Chroma. Some blocks are non-convex. The *peripheral blocks*, of which there are 120, have arbitrary large chroma and are considered semi-infinite for this function; there is no consideration of the MacAdam limits. For each query vector HVC, the function searches a private `data.frame` with 932 elementary blocks, for the one elementary block that contains it.

## Value

a `data.frame` with N rows and these columns:

| | |
|---|---|
| `HVC` | the input Nx3 matrix, or such a matrix converted from Munsell notation |
| `Number` | the corresponding ISCC-NBS color number - an integer from 1 to 267 |
| `Name` | the corresponding ISCC-NBS color name - a character string |
| `Centroid` | the centroid of the block in Munsell Notation - a character string; see `CentroidsISCCNBS` |

The `rownames` are set to the input `MunsellSpec`.

## History

The **Munsell Book of Color** was published in 1929. The first ISCC-NBS partition, in 1939, had 319 blocks and names (including 5 neutrals). There were no block numbers. The aimpoints of the Munsell samples were thoroughly revised in 1943. The ISCC-NBS partition was revised in 1955, and this is the version used here.

## Future Work

It might be useful to compute the distance from the query point to the boundary of the containing color block.

### Author(s)

Glenn Davis

### References

Munsell Color Company, A.H. Munsell, and A.E.O. Munsell. **Munsell book of color: defining, explaining, and illustrating the fundamental characteristics of color**. 1929.

Judd, Deane B. and Kenneth L. Kelly. **Method of Designating Colors**. Journal of Research of the National Bureau of Standards. Research Paper 1239. Volume 23 Issue 3. pp. 355-385. September 1939.

Newhall, Sidney M., Dorothy Nickerson, Deane B. Judd. **Final Report of the O.S.A. Subcommitte on the Spacing of the Munsell Colors**. Journal of the Optical Society of America. Vol. 33. No. 7. pp. 385-418. July 1943.

Kelly, Kenneth L. and Deane B. Judd **The ISCC-NBS Method of Designating Colors and a Dictionary of Color Names**. National Bureau of Standards Circular 553. Washington DC: US Government Printing Office. November 1, 1955.

### See Also

`CentroidsISCCNBS`

### Examples

```
ColorBlockFromMunsell( c( "3R 8/3", "7.4YR 3/4" ) )

##            HVC.H HVC.V HVC.C Number           Name      Centroid
##  3R 8/3      3.0   8.0   3.0      4     light pink  2.5R 8.6/5.2
##  7.4YR 3/4  17.4   3.0   4.0     58 moderate brown 5.5YR 3.5/3.9
```

---

## ColorlabFormatToMunsellSpec

*Convert Colorlab Munsell Format to Munsell HVC*

---

### Description

Convert Colorlab Munsell Format to Munsell HVC

### Usage

```
ColorlabFormatToMunsellSpec( HVCH )
```

### Arguments

| | |
|---|---|
| `HVCH` | a numeric Nx4 matrix, or a vector that can be converted to such a matrix, by row. Each row of the matrix contains an HVCH vector. |

### Details

Colorlab Munsell format uses 4 numbers.

1. Hue Step, in the interval (0,10], or 0 for neutrals
2. Munsell Value, in the interval (0,10]
3. Munsell Chroma, non-negative
4. Hue Index, an integer from 1 to 10, or 0 for neutrals

### Value

an Nx3 matrix, with each row an HVC vector. Value and Chroma are simply copied unchanged. The complex part is conversion of Colorlab Hue Step and Hue Index to Hue Number. For neutrals, both Hue Step and Hue Index are ignored. Invalid input values, such as a Hue Index that is not an integer from 0 to 10 (except for neutrals), are converted to `NA`s. The rownames of the input are copied to the output, but if these are `NULL`, the rownames are set to the Munsell notations.

### Author(s)

Jose Gama and Glenn Davis

### References

Color Processing Toolbox. Colorlab 1.0. `http://www.uv.es/vista/vistavalencia/software/colorlab.html`

### See Also

`MunsellSpecToColorlabFormat()`

### Examples

```
ColorlabFormatToMunsellSpec( c( 3.2,3,2,1,  2,5.1,0,0, 2,5.1,0.1,0 ) )
##                      H   V  C
##  3.20B 3.00/2.00 63.2 3.0  2
##  N 5.10/          0.0 5.1  0
##  <NA>              NA  NA NA
```

---

## `HVCfromMunsellName` *Convert Munsell Notation to numerical HVC*

---

### Description

Convert Munsell Notation to numerical HVC

### Usage

```
HVCfromMunsellName( MunsellName )
MunsellHVC( MunsellName )
HueNumberFromString( HueString )
```

### Arguments

`MunsellName` a character vector of length N > 0, where each string should be a valid Munsell notation, e.g. `'2.3P 5/2.3'`,`'9.2YR 3/6'`,`'N 2.3/'`, and `'N 4/0'`. Whitespace is optional and ignored. It is OK for a neutral to end in either `'/'` or `'/0'`.

`HueString` a character vector of length N > 0, where each string should be the initial hue part of a Munsell notation, e.g. '4.5GY', '2.5R', '10.3B', etc. Whitespace is optional and ignored. Neutrals, denoted by `'N'`, are invalid because the hue is undefined.

### Value

`HVCfromMunsellName()` returns a numeric Nx3 matrix with HVC in the rows. For neutral colors, both H and C are set to 0. If a string cannot be parsed, the entire row is set to `NA`s. The rownames are set to `MunsellName`.
`MunsellHVC()` returns a character Nx3 matrix with HVC in the rows, and is there for backward compatibility with older versions of the package. For neutral colors, H is set to `'N'` and C is set to `'0'`.
`HueNumberFromString()` returns the hue number H (in (0,100]). If the string cannot be parsed, or the color is neutral, the output is set to `NA`.
For all functions the Hue Number is wrapped to (0,100].

### Note

Ever since the *Munsell Book of Color (1929)*, the Munsell hue circle has been divided into 10 *principal hues* or arcs. And each principal hue has been assigned a 10-point scale, with 5 at the midpoint of the arc. Moreover, the hue "origin" has been at '10RP'. So a 100-point scale (with no letters) for the entire hue circle is obvious and trivial to construct, but I have been unable to determine the first explicit mention of such a scale. The earliest I have have found is from *Nimeroff (1968)*, Figures 20 and 21 on page 27.

There is a reference to *ASTM D 1535* in the **References** of *Nimeroff*, but it is not dated, and the 2 figures are not attributed to it. There was an *ASTM D 1535* in 1968 but I have not been able to locate it; it is possible that the 100-point scale first appeared in *ASTM D-1535 (1968)*, or even earlier in *ASTM D 1525-58T (1958)*.
Interestingly, in the *Atlas of the Munsell Color System (1915)* there were only 5 principal hues, and each arc was assigned a 10-point scale. If the entire hue circle were assigned a scale, it would have been a 50-point scale.

### Author(s)

Glenn Davis

### References

Nimeroff, I. **Colorimetry**. National Bureau of Standards Monograph 104. January 1968. 35 cents.

ASTM D 1535-80. Standard Practice for Specifying Color by the Munsell System. 1980.

Munsell Book of Color: defining, explaining, and illustrating the fundamental characteristics of color. Munsell Color Co. 1929.

Atlas of the Munsell Color System. Malden, Mass., Wadsworth, Howland & Co., inc., Printers. 1915.

### See Also

`MunsellNameFromHVC()`, `HueStringFromNumber()`

### Examples

```
HVCfromMunsellName( c( "4.2P 2.9/3.8", "N 2.3/", "N 8.9/0" ) )
##                 H   V   C
##  4.2P 2.9/3.8 84.2 2.9 3.8
##  N 2.3/        0.0 2.3 0.0
##  N 8.9/0       0.0 8.9 0.0

HueNumberFromString( c('4B','4.6GY','10RP','N') )
##  [1]  64.0  34.6 100.0    NA
```

---

`IsWithinMacAdamLimits` *Test xyY Coordinates for being Inside the MacAdam Limits*

---

### Description

Test xyY Coordinates for being Inside the MacAdam Limits for Illuminants C and D65

### Usage

```
IsWithinMacAdamLimits( xyY, Illuminant='C' )
```

### Arguments

| | |
|---|---|
| `xyY` | a numeric Nx3 matrix with CIE xyY coordinates in the rows, or a vector that can be converted to such a matrix, by row. The reference white is asssumed to have Y=100. |
| `Illuminant` | either `'C'` or `'D65'`. Partial matching is enabled, and is case-insensitive. |

### Details

The *MacAdam Limit* is the boundary of the *optimal color solid* (also called the *Rösch Farbkörper*), in XYZ coordinates. The optimal color solid is convex and depends on the illuminant. Points on the boundary of the solid are called *optimal colors*.

It is symmetric about the midpoint of the segment joining black and white (the 50% gray point). It can be expressed as a *zonohedron* - a convex polyhedron with a special form; for details on zonohedra, see Centore.

For each of the 2 illuminants, a zonohedron $Z$ is pre-computed (and stored in `sysdata.rda`). The wavelengths used are 380 to 780 nm with 5nm step (81 wavelengths). Each zonohedron has

81*80=6480 parallelogram faces, though some of them are coplanar. $Z$ is expressed as the intersection of 6480 halfspaces. The plane equation of each parallelogram is pre-computed, but redundant ones are not removed (in this version).

For testing a query point xyY, a pseudo-distance metric $\delta$ is used. Let the zonohedron $Z$ be the intersection of the halfspaces $\langle\, h_i, x \,\rangle \leq b_i \;\; i = 1, ..., n$, where each $h_i$ is a unit vector. The point xyY is converted to XYZ, and $\delta$(XYZ) is computed as: $\delta$(XYZ) := max( $\langle\, h_i$,XYZ $\rangle$ - $b_i$ ) where the maximum is taken over all $i = 1, ..., n$. This calculation can be optimized; because the zonohedron is centrally symmetric, only half of the planes actually have to be stored, and this cuts the memory and processing time in half. It is clear that XYZ is within the zonohedron iff $\delta$(XYZ) $\leq 0$, and that XYZ is on the boundary iff $\delta$(XYZ)=0. This pseudo-distance is part of the returned `data.frame`.

An interesting fact is that if $\delta$(XYZ)>0, then $\delta$(XYZ) $\leq$ dist(XYZ,$Z$), with equality iff the segment from $XYZ$ to the point $z$ on the boundary of $Z$ closest to XYZ is normal to one of the faces of $Z$ that contains $z$. This is why we call $\delta$ a *pseudo-distance*. Another interesting fact is that if $\delta$(XYZ) $\leq$ 0, then $\delta$(XYZ) = -min( $\Psi_Z(u)$ - $\langle\, u$,XYZ $\rangle$ ), where the minimum is taken over all unit vectors $u$ and where $\Psi_Z$ is the *support function* of $Z$.

## Value

A `data.frame` with N rows and these columns:

| | |
|---|---|
| `within` | a logical which is `TRUE` iff the corresponding row in `xyY` is inside the optimal color solid for the illuminant. If a point is exactly on the boundary (unlikely), `within=TRUE`. Explicitly, `within = (delta<=0)`. |
| `delta` | the pseudo-distance $\delta$ discussed in **Details** |

The row names of the output value are set equal to the row names of `xyY`.

## Author(s)

Glenn Davis and Jose Gama

## References


Centore, Paul. *A zonohedral approach to optimal colours*. **Color Research & Application**. Vol. 38. No. 2. pp. 110-119. April 2013.

Rösch, S. **Darstellung der Farbenlehre für die Zwecke des Mineralogen**. Fortschr. Mineral. Krist. Petrogr. Vol. 13 No. 143. 1929.

MacAdam, David L. **Maximum Visual Efficiency of Colored Materials**. Journal of the Optical Society of America. Vol 25, No. 11. pp. 361-367. November 1935.

Wikipedia. **Support Function**. `http://en.wikipedia.org/wiki/Support_function`


## Examples

```
IsWithinMacAdamLimits( c(0.6,0.3,10, 0.6,0.3,20, 0.6,0.3,30, 0.6,0.3,40  ), 'C' )

##    within  delta
## 1   TRUE -1.941841
## 2   TRUE -1.332442
## 3  FALSE  3.513491
```

```
## 4 FALSE 12.826172
```

---

LabtoMunsell *Convert CIE Lab coordinates into a Munsell specification*

---

### Description

`LabtoMunsell` Converts CIE Lab coordinates to a Munsell specification, by interpolating over the extrapolated Munsell renotation data

### Usage

```
LabtoMunsell( Lab, white=c(95.047,100,108.883), adapt='Bradford', ... )
```

### Arguments

| | |
|---|---|
| `Lab` | CIE Lab coordinates An Nx3 matrix, or a vector that can be converted to such a matrix. Each row of the matrix has Lab. |
| `white` | XYZ for the source white. The default is Illuminant=D65, 2 observer |
| `adapt` | method for chromatic adaptation, see `CAT()` for valid values. Also see **Details**. |
| `...` | other parameters passed to `XYZtoMunsell()` |

### Details

The conversion is done in 3 steps.

- Lab → XYZ using `XYZfromLab()` with the given `white`.
- XYZ is then adapted from the given `white` to Illuminant C using the given `adapt` method.
- XYZ → HVC using `XYZtoMunsell()`.

### Value

An Nx3 matrix with the Munsell HVC coordinates in each row. The rownames are set to those of Lab.

### Author(s)

Jose Gama and Glenn Davis

### References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

### See Also

`CAT()`, `XYZfromLab()`, `XYZtoMunsell()`

### Examples

```
LabtoMunsell( c(74.613450, -20.4, 10.1) )
```

---

## LuvtoMunsell — *Convert CIE Luv coordinates into a Munsell specification*

---

### Description

`LuvtoMunsell` Converts CIE Luv coordinates into a Munsell specification, by interpolating over the extrapolated Munsell renotation data

### Usage

```
LuvtoMunsell( Luv, white=c(95.047,100,108.883), adapt='Bradford', ... )
```

### Arguments

| | |
|---|---|
| `Luv` | CIE Luv coordinates An Nx3 matrix, or a vector that can be converted to such a matrix. Each row of the matrix has Luv. |
| `white` | XYZ for the reference white. The default is Illuminant=D65, 2 observer |
| `adapt` | method for chromatic adaptation, see **Details** |
| `...` | other parameters passed to `XYZtoMunsell()` |

### Details

The conversion is done in 3 steps.

- Luv $\rightarrow$ XYZ using `XYZfromLuv()` with the given `white`.
- XYZ is then adapted from the given `white` to Illuminant C using the given chromatic adaptation method, see `CAT()`.
- XYZ $\rightarrow$ HVC using `XYZtoMunsell()`

### Value

An Nx3 matrix with the Munsell HVC coordinates in each row. The rownames are set to those of Luv.

### Author(s)

Jose Gama and Glenn Davis

### References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

### See Also

`XYZtoMunsell()`, `XYZfromLuv()`, `CAT()`

### Examples

```
LuvtoMunsell( c( 74.613450, -5.3108, 10.6 ) )
```

---

## `Munsell2xy` *The Munsell HVC to xy 3D Lookup Table*

---

### Description

This is the discrete data for the Munsell Renotation System, which is often considered to be the most perceptually uniform color atlas. It was created by the NBS and OSA from "3,000,000 color judgments" by 40 observers.

### Format

A data frame with 4995 observations of the following 6 variables.

- `H` the Munsell Hue. Each `H` is a multiple of 2.5 and in the interval (0,100].
- `V` the Munsell Value. Each `V` is an integer from 1 to 10, or one of 0.2, 0.4, 0.6, 0.8
- `C` the Munsell Chroma. Each `C` is a positive even integer.
- `x` the x chromaticity coordinate, for Illuminant C.
- `y` the y chromaticity coordinate, for Illuminant C.
- `real` a logical value. If `TRUE` then x,y were published, otherwise they have been extrapolated.

Note that the luminance factor Y is *not* here, since Y is a simple function of V, see `YfromV()`.

### Details

All the (x,y) data here comes from the file `all.dat` downloaded from Rochester Institute of Technology, see **Source**. The file `real.dat` is a subset, and contains the (x,y) published in *Newhall, et. al. (1943)*. These rows have `real=TRUE` and are only for Value $\geq 1$. There are 2734 of these. Similarly, for Value<1 (very dark colors), (x,y) data from the paper *Judd et. al. (1956)* also have `real=TRUE`. There are 355 of these.
So `all.dat` has 4995 colors, of which 2734+355=3089 are "real" colors, and the remaining 1906 are extrapolated. I am confident that the extrapolation was done by *Schleter et. al. (1958)* at the NBS, and put online by the Rochester Institute of Technology. For more details, and the abstract of the 1958 article, see the **munsellinterpol User Guide**.

### Note

For the purpose of this package, I have found that the extrapolated (x,y) for V$\geq$1 work well. But for V<1 they did not work so well, and I was able to get better results with my own extrapolation. Moreover, to get reliable results in this package for high Chroma, it was necessary to extrapolate past the data in `all.dat`.

### Author(s)

Glenn Davis

### Source

Rochester Institute of Technology. Program of Color Science. Munsell Renotation Data. `http://www.rit.edu/science/pocs/renotation`

### References

Newhall, Sidney M., Dorothy Nickerson, Deane B. Judd. **Final Report of the O.S.A. Subcommitte on the Spacing of the Munsell Colors**. Journal of the Optical Society of America. Vol. 33. No. 7. pp. 385-418. July 1943.

Judd, Deane B. and Gunter Wyszecki. **Extension of the Munsell Renotation System to Very Dark Colors**. Journal of the Optical Society of America. Vol. 46. No. 4. pp. 281-284. April 1956.

Schleter, J. C, D. B. Judd, D. B., H. J. Keegan. **Extension of the Munsell Renotation System (Abstract)**. J. Opt. Soc. Am. Vol 48. Num. 11. pp. 863-864. presented at the Forty-Third Annual Meeting of the Optical Society of America. Statler Hilton Hotel, Detroit, Michigan. October 9, 10, and 11, 1958.

### See Also

`YfromV()`

### Examples

```
str(Munsell2xy)

##  'data.frame': 4995 obs. of  6 variables:
##   $ H   : num  32.5 35 37.5 37.5 40 40 42.5 42.5 45 45 ...
##   $ V   : num  0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 ...
##   $ C   : num  2 2 2 4 2 4 2 4 2 4 ...
##   $ x   : num  0.713 0.449 0.262 -0.078 0.185 -0.257 0.144 -0.235 0.117 -0.209 ...
##   $ y   : num  1.414 1.145 0.837 2.16 0.676 ...
##   $ real: logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
```

---

`MunsellNameFromHVC` *Convert Munsell Numeric to Munsell String Notation*

---

### Description

Convert Munsell Numeric to Munsell String Notation

### Usage

```
MunsellNameFromHVC( HVC, format='g', digits=2 )

HueStringFromNumber( Hue, format='g', digits=2 )
```

### Arguments

| | |
|---|---|
| `HVC` | a numeric Nx3 matrix or a vector that can be converted to such a matrix. Each row has an HVC vector, where H is Hue Number, and V and C are the standard Munsell Value and Chroma. The Hue is automatically wrapped to the interval (0,100]. |
| `Hue` | a numeric vector of Hue Numbers, which are automatically wrapped to the interval (0,100]. |
| `format` | determines the meaning of the argument `digits`, and usually equal to `'g'` or `'f'` though other options are available, see `formatC()` for details. When `format='f'` trailing 0s might be displayed. |
| `digits` | when `format='g'` the number of significant digits, and when `format='f'` the number of digits displayed after the decimal point. Both `format` and `digits` are passed as arguments to `formatC()`. |

### Value

Both functions return a character vector of length N. `MunsellNameFromHVC()` returns the full notation. `HueStringFromNumber()` returns just initial the hue part; which is useful for labeling plots.

### Note

If `format='f'`, then Chroma is first *rounded* to to the given `digits`. Chromas close to 0 may then become 0 and be displayed as a neutral, see **Examples**.
The `width` argument of `formatC()` is always set to 1, to suppress leading spaces.

### Author(s)

Glenn Davis

### References

ASTM D 1535-97. Standard Practice for Specifying Color by the Munsell System. 1997

### See Also

`formatC()`, `HVCfromMunsellName()`, `HueNumberFromString()`

### Examples

```
MunsellNameFromHVC( c(39,5.1,7.3,  0,5.1234,0.003 )  )
##  [1] "9GY 5.1/7.3"    "10RP 5.1/0.003"

MunsellNameFromHVC( c(39,5.1,7.34,  0,5.1234,0.003 ), format='f' )
##  [1] "9.00GY 5.10/7.34" "N 5.10/"

HueStringFromNumber( seq( 2.5, 100, by=2.5 ) )   # make nice labels for a plot
##   [1] "2.5R"  "5R"    "7.5R"  "10R"   "2.5YR" "5YR"   "7.5YR" "10YR"  "2.5Y"
##  [10] "5Y"    "7.5Y"  "10Y"   "2.5GY" "5GY"   "7.5GY" "10GY"  "2.5G"  "5G"
```

```
## [19] "7.5G"  "10G"   "2.5BG" "5BG"   "7.5BG" "10BG"  "2.5B"  "5B"    "7.5B"
## [28] "10B"   "2.5PB" "5PB"   "7.5PB" "10PB"  "2.5P"  "5P"    "7.5P"  "10P"
## [37] "2.5RP" "5RP"   "7.5RP" "10RP"
```

---

MunsellSpecToColorlabFormat
*Convert Munsell Specification to Colorlab Format*

---

### Description

Convert Munsell Specification to Colorlab Format

### Usage

```
MunsellSpecToColorlabFormat( MunsellSpec )
```

### Arguments

| | |
|---|---|
| `MunsellSpec` | a numeric Nx3 matrix, or a vector that can be converted to such a matrix, by row. Each row of the matrix contains an HVC vector. H is automatically wrapped to the interval (0,100].<br>`MunsellSpec` can also be a character vector with Munsell Notation; which is converted to an Nx3 matrix using `HVCfromMunsellName()`. |

### Details

Colorlab Munsell format uses 4 numbers.

1. Hue Step, in the interval (0,10], or 0 for neutrals. In Colorlab documentation it is called the *hue shade*. It is also the Hue Number H mod 10 (unless H is an exact multiple of 10).
2. Munsell Value, in the interval [0,10]
3. Munsell Chroma, non-negative
4. Hue Index, an integer from 1 to 10, or 0 for neutrals. This index defines the *principal hue*, see **Details**.

### Value

an Nx4 matrix, with rows as described in **Details**. Value and Chroma are simply copied unchanged. The complex part is conversion of Hue Number to Colorlab Hue Step and Hue Index. If Chroma is 0, both the Hue Step and Hue Index are set to 0. Invalid input values are converted to `NA`s.
If the input is a character vector, the rownames of the returned matrix are set to that vector.

### Note

The Colorlab format is closer to the *Munsell Book of Color (1929)* than HVC. In the book the hue circle is divided into 10 *principal hues* - 5 simple and 5 compound. The 10 hue labels are: **R, YR, Y, GY G, BG, B, PB, P** (simple are 1 letter and compound are 2 letters). In Colorlab these labels are replaced by the Hue Index. WARNING: In the Munsell System, see *Cleland*, there is a different Hue Index - **R** is 1, **YR** is 2, ..., **P** is 10. The Colorlab index has a different origin, and goes around the circle in a different direction !
Each one of these major hues corresponds to an arc on the circle, with a 10-point hue scale. The midpoint of the arc has hue step = 5. Fortunately this 10-point hue scale is exactly the same as the Colorlab Hue Step.

### Author(s)

Jose Gama and Glenn Davis

### References

Color Processing Toolbox. Colorlab 1.0. `http://www.uv.es/vista/vistavalencia/software/colorlab.html`

Cleland, T. M. **A Practical description of the Munsell Color System with Suggestions for its Use**. (1921)

### See Also

`HVCfromMunsellName()`, `ColorlabFormatToMunsellSpec()`

### Examples

```
MunsellSpecToColorlabFormat( c(100,5,5, 10,3,4, 90,4,3, 77,1,2, 66,2,0, 0,1,2 ) )
##                    HN V C HI
##  10.00RP 5.00/5.00 10 5 5  8
##  10.00R 3.00/4.00  10 3 4  7
##  10.00P 4.00/3.00  10 4 3  9
##  7.00PB 1.00/2.00   7 1 2 10
##  N 2.00/            0 2 0  0
##  10.00RP 1.00/2.00 10 1 2  8
```

---

`MunsellToLab` *Convert a Munsell specification to CIE Lab coordinates*

---

### Description

`MunsellToLab` Converts a Munsell specification to CIE Lab coordinates, by interpolating over the extrapolated Munsell renotation data

### Usage

```
MunsellToLab( MunsellSpec, white=c(95.047,100,108.883), adapt='Bradford', ... )
```

### Arguments

| | |
|---|---|
| `MunsellSpec` | a numeric Nx3 matrix with HVC values in the rows, or a vector that can be converted to such a matrix, by row.<br>`MunsellSpec` can also be a character vector with Munsell Notations, which is converted to an Nx3 matrix using `HVCfromMunsellName()` |
| `white` | XYZ for the destination white. The default is Illuminant=D65, 2 observer |
| `adapt` | method for chromatic adaptation, see `CAT()` for valid values. Also see **Details**. |
| `...` | other parameters passed to `MunsellToXYZ()` |

### Details

The conversion is done in 3 steps.

- HVC $\rightarrow$ XYZ using `MunsellToXYZ()`
- XYZ is adapted from Illuminant C to the given `white` using `adaptXYZ()` and the given chromatic adaptation method
- XYZ $\rightarrow$ Lab using `LabfromXYZ()` with the given `white`

### Value

An Nx3 matrix with the Lab coordinates in each row. The rownames are copied from input to output.

### Author(s)

Jose Gama and Glenn Davis

### References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

### See Also

`LabtoMunsell()`, `MunsellToXYZ()`, `LabfromXYZ()`, `adaptXYZ()`, `CAT()`

### Examples

```
MunsellToLab('7.6P 8.9/2.2')
```

MunsellToLuv *Convert a Munsell specification to CIE Luv coordinates*

## Description

MunsellToLuv Converts a Munsell specification to CIE Luv coordinates, by interpolating over the extrapolated Munsell renotation data

## Usage

```
MunsellToLuv( MunsellSpec,  white=c(95.047,100,108.883), adapt='Bradford', ... )
```

## Arguments

| | |
|---|---|
| MunsellSpec | a numeric Nx3 matrix with HVC values in the rows, or a vector that can be converted to such a matrix, by row.<br>MunsellSpec can also be a character vector with Munsell Notations, which is converted to an Nx3 matrix using HVCfromMunsellName() |
| white | XYZ for the destination white. The default is Illuminant=D65, 2 observer |
| adapt | method for chromatic adaptation, see CAT() for valid values. Also see **Details**. |
| ... | other parameters passed to MunsellToXYZ() |

## Details

The conversion is done in 3 steps.

- HVC → XYZ using MunsellToXYZ()
- XYZ is adapted from Illuminant C to the given white using the given chromatic adaptation method
- XYZ → Luv using LuvfromXYZ() with the given white

## Value

An Nx3 matrix with the Luv coordinates in each row. The rownames are copied from input to output.

## Author(s)

Jose Gama and Glenn Davis

## References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html

### See Also

`LuvtoMunsell()`, `MunsellToXYZ()`, `CAT()`, `LuvfromXYZ()`

### Examples

```
MunsellToLuv('7.6P 8.9/2.2')
```

---

## `MunsellToRGB` *Convert a Munsell specification to RGB coordinates*

---

### Description

`MunsellToRGB` Converts a Munsell specification to RGB coordinates, by interpolating over the extrapolated Munsell renotation data

### Usage

```
MunsellToRGB( MunsellSpec, space='sRGB', maxSignal=255, adapt='Bradford', ... )
```

### Arguments

| | |
|---|---|
| `MunsellSpec` | a numeric Nx3 matrix with HVC values in the rows, or a vector that can be converted to such a matrix, by row.<br>`MunsellSpec` can also be a character vector with Munsell Notations, which is converted to an Nx3 matrix using `HVCfromMunsellName()`. |
| `space` | the name of an installed RGB space. Spaces `'sRGB'` and `'AdobeRGB'` are pre-installed, and others can be installed with `installRGB()`. |
| `maxSignal` | maximum of the non-linear signal RGB; Other popular values are 1, 1023, and 65535 |
| `adapt` | method for chromatic adaptation, see `CAT()` for valid values. Also see **Details**. |
| `...` | other parameters passed to `MunsellToxyY()` |

### Details

The conversion is done in these steps.

- HVC $\rightarrow$ xyY using `MunsellToxyY()`. This xyY is for Illuminant C.
- xyY is adapted from Illuminant C to the white-point of the RGB `space` using `adaptxyY()` and the given chromatic adaptation method
- xyY $\rightarrow$ XYZ using `XYZfromxyY()`
- XYZ $\rightarrow$ RGB using `RGBfromXYZ()` with the given `space` and `maxSignal`

### Value

a `data.frame` with these columns

| | |
|---|---|
| `xyY` | an Nx3 matrix with xyY values in the rows, that are adapted to Illuminant C. This is an intermediate result that is sometimes useful, e.g. it can be passed to `IsWithinMacAdamLimits()`. |
| `RGB` | an Nx3 matrix with non-linear RGB signal values in the rows. All values are clamped to the appropriate cube, e.g. $[0, 255]^3$ |
| `OutOfGamut` | logical vector, TRUE means the result was out of gamut (the cube) before clamping it |

In case of error, it returns `NULL`.

### Author(s)

Jose Gama and Glenn Davis

### References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

### See Also

`installRGB()`,`MunsellToXYZ()`,`RGBfromXYZ()`,`XYZfromxyY()`,`CAT()`,`IsWithinMacAdamLimits()`

### Examples

```
MunsellToRGB('7.6P 8.9/2.2')
```

---

## `MunsellTosRGB` *Convert a Munsell specification to sRGB coordinates*

---

### Description

`MunsellTosRGB` Converts a Munsell specification to non-linear sRGB coordinates, by interpolating over the extrapolated Munsell renotation data

### Usage

```
MunsellTosRGB( MunsellSpec, maxSignal=255, ... )
```

### Arguments

| | |
|---|---|
| `MunsellSpec` | a numeric Nx3 matrix with HVC values in the rows, or a vector that can be converted to such a matrix, by row. `MunsellSpec` can also be a character vector with Munsell Notations, which is converted to an Nx3 matrix using `HVCfromMunsellName()`. |
| `maxSignal` | maximum of signal sRGB; Other popular values are 1, 1023, and 65535 |
| `...` | other parameters passed to `MunsellToxyY()` |

### Details

The conversion is done in these steps.

- HVC → xyY using `MunsellToxyY()`. This xyY is for Illuminant C.
- xyY is adapted from Illuminant C to Illuminant D65 (from the sRGB standard) using `adaptxyY()` and the *Bradford* chromatic adaptation method (CAT)
- xyY → XYZ using `XYZfromxyY()`
- XYZ → sRGB using `RGBfromXYZ()` with the given `maxSignal`

### Value

a `data.frame` with these columns

| | |
|---|---|
| `xyY` | an Nx3 matrix with xyY values in the rows, that are adapted to Illuminant C. This is an intermediate result that is sometimes useful, e.g. it can be passed to `IsWithinMacAdamLimits()`. |
| `RGB` | an Nx3 matrix with non-linear signal sRGB values in the rows. All values are clamped to the appropriate cube, e.g. $[0, 255]^3$ |
| `OutOfGamut` | logical vector, TRUE means the result was out of gamut (the cube) before clamping it |

### Note

The function `MunsellToRGB()` also performs this conversion. The only reason to use this one is that it takes a little less time, since the CAT (using the *Bradford* method) is precomputed.

### Author(s)

Jose Gama and Glenn Davis

### References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

### See Also

`MunsellToXYZ()`, `MunsellToRGB()`, `RGBfromXYZ()`, `XYZfromxyY()`, `CAT()`, `IsWithinMacAdamLimits(`

## Examples

```
MunsellTosRGB('7.6P 8.9/2.2')
##                SAMPLE_NAME      xyY.x      xyY.y      xyY.Y    RGB.R    RGB.G    RGB.B Ou
##  7.6P 8.9/2.2 7.6P 8.9/2.2  0.3109520  0.3068719 74.6134498 231.3575 221.1421 230.3501
```

---

`MunsellToxyY` *Convert a Munsell specification into xyY coordinates*

---

## Description

`MunsellToxyY` Converts a Munsell specification into xyY coordinates, by interpolating over the extrapolated Munsell renotation data

## Usage

```
MunsellToxyY( MunsellSpec, xyC='NBS', hcinterp='bicubic', vinterp='cubic',
                         YfromV='ASTM', warn=TRUE )
```

## Arguments

`MunsellSpec` a numeric Nx3 matrix or a vector that can be converted to such a matrix. Each row has Munsell HVC, where H is Hue Number, and V and C are the standard Munsell Value and Chroma. The Hue is automatically wrapped to the interval (0,100]. `MunsellSpec` can also be a character N-vector with standard Munsell notation; it is converted to an Nx3 matrix.

`xyC` a numeric 2-vector with xy chromaticity of Illuminant C. It can also be one of the strings in the first column of this table; it is then replaced by the corresponding xy in the second column.

| | xy white point | reference |
|---|---|---|
| 'NBS' | `c(0.3101,0.3163)` | Kelly, et. al. [RP1549] (1943). Rheinboldt et al. (1960) |
| 'JOSA' | `c(0.31012,0.31631)` | Judd, Deane B. (1933) |
| 'NTSC' | `c(0.310,0.316)` | NTSC (1953) |
| 'CIE' | `c(0.31006,0.31616)` | CIE:15 2004 |

The default `'NBS'` is probably what is intended by *Newhall et. al.* although no xy for C appears in that paper. This is the C used in the first computer program for conversion: *Rheinboldt et al. (1960)*. The other options are provided so that a neutral Munsell chip has the xy that the user expects. Alternative values of `xyC` should not be *too far* from the above. If `hcinterp` is `'bicubic'`, this parameter only affects chips with Chroma $< 4$ (except Chroma=2). If `hcinterp` is `'bilinear'`, this parameter only affects chips with Chroma $< 2$.

`hcinterp` either `'bicubic'` or `'bilinear'` (partial matching enabled). In the bicubic case, for a general input point, the output value is interpolated using a 4x4 subgrid of the lookup table, and the interpolation function is class $C^1$ (except at

| | |
|---|---|
| | the neutrals). In the bilinear case, the interpolation uses a 2x2 subgrid, and the function is class $C^0$. |
| vinterp | either 'cubic' or 'linear' (partial matching enabled). In the cubic case, for a general input point, the output value is interpolated using 4 planes of constant Value, and the interpolation function is class $C^1$. In the linear case, the interpolation uses 2 planes and the function is class $C^0$. |
| YfromV | passed as the parameter which to the function YfromV(). See YfromV() for details. Option 'MGO' is not allowed because then Y>100 when V=10. |
| warn | if a chip cannot be mapped (usually because the Chroma is too large), its x and y are set to NA in the returned data.frame. Just before returning, if any rows have NA, and this argument is TRUE, then a warning is logged. |

## Details

In case hcinterp='bicubic' or vinterp='cubic' a Catmull-Rom spline is used; see the article *Cubic Hermite spline*. This spline has the nice property that it is *local* and requires at most 4 points. And if the knot spacing is uniform: 1) the resulting spline is $C^1$, 2) if the knots are on a line, the interpolated points are on the line too.

## Value

a data.frame with these columns:

| | |
|---|---|
| SAMPLE_NAME | the original MunsellSpec if that was a character vector. Or the Munsell notation string converted from HVC. |
| HVC | the input Nx3 matrix |
| xyY | the computed output matrix, with CIE xyY coordinates of MunsellSpec illuminated by Illuminant C. In case of error, x and y are set to NA. |

## Warning

Even when vinterp='cubic' the function HVC $\to$ xyY is not $C^1$ on the plane V=1. This is because of a change in Value spacing: when V$\geq$1 the Value spacing is 1, but when V$\leq$1 the Value spacing is 0.2.

## Note

When making plots in planes of constant Value, option hcinterp='bicubic' makes fairly smooth ovals, and hcinterp='bilinear' makes polygons. The ovals are smooth even when vinterp='linear', but the function is not class $C^1$ at the planes of integer Value. To get a fully $C^1$ function (except at the neutrals and on the plane V=1), hcinterp and vinterp must be set to the defaults.

## Author(s)

Jose Gama and Glenn Davis

## Source

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html
http://www.rit.edu/science/pocs/renotation
http://www.rit-mcsl.org/MunsellRenotation/all.dat
http://www.rit-mcsl.org/MunsellRenotation/real.dat

## References

Judd, Deane B. **The 1931 I.C.I. Standard Observer and Coordinate System for Colorimetry**. Journal of the Optical Society of America. Vol. 23. pp. 359-374. October 1933.

Newhall, Sidney M., Dorothy Nickerson, Deane B. Judd. **Final Report of the O.S.A. Subcommitte on the Spacing of the Munsell Colors**. Journal of the Optical Society of America. Vol. 33. No. 7. pp. 385-418. July 1943.

Kelly, Kenneth L. Kasson S. Gibson. Dorothy Nickerson. **Tristimulus Specification of the Munsell Book of Color from Spectrophometric Measurements** National Bureau of Standards RP1549 Volume 31. August 1943.

Judd, Deane B. and Günther Wyszecki. **Extension of the Munsell Renotation System to Very Dark Colors**. Journal of the Optical Society of America. Vol. 46. No. 4. pp. 281-284. April 1956.

National Television System Committee. [Report and Reports of Panel No. 11, 11-A, 12-19, with Some supplementary references cited in the Reports, and the Petition for adoption of transmission standards for color television before the Federal Communications Commission] (1953)

Rheinboldt, Werner C. and John P. Menard. **Mechanized Conversion of Colorimetric Data to Munsell Renotations**. Journal of the Optical Society of America. Vol. 50, Issue 8, pp. 802-807. August 1960.

Wikipedia. Cubic Hermite spline. https://en.wikipedia.org/wiki/Cubic_Hermite_spline

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html

## See Also

xyYtoMunsell()

## Examples

```
MunsellToxyY( '7.6P 8.9/2.2' )
```

`MunsellToXYZ` *Convert a Munsell specification to CIE XYZ coordinates*

## Description

`MunsellToXYZ` Converts a Munsell specification to XYZ coordinates, by interpolating over the extrapolated Munsell renotation data

## Usage

```
MunsellToXYZ( MunsellSpec, ... )
```

## Arguments

| | |
|---|---|
| `MunsellSpec` | a numeric Nx3 matrix with HVC values in the rows, or a vector that can be converted to such a matrix, by row.<br>`MunsellSpec` can also be a character vector with Munsell Notations, which is converted to an Nx3 matrix using `HVCfromMunsellName()`. |
| `...` | other parameters passed to `MunsellToxyY()` |

## Details

This function calls `MunsellToxyY()` and `XYZfromxyY()`.

## Value

an Nx3 matrix with XYZ values in the rows. The rownames are copied from the input HVC matrix. Exception: If the input matrix rownames are `NULL`, then the output rownames are the Munsell notation. Note that these XYZ values are for viewing under Illuminant C, with Y=100. There is no chromatic adaptation.

## Author(s)

Jose Gama and Glenn Davis

## References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

## See Also

`MunsellToxyY()`, `XYZfromxyY()`

## Examples

```
MunsellToXYZ('7.6P 8.9/2.2')
```

`NickersonColorDifference`
*Calculate the Nickerson Color Difference between two Colors*

## Description

Calculate the Nickerson Color Difference between two colors, given in Munsell coordinates; see *Nickerson*.

## Usage

```
NickersonColorDifference( HVC0, HVC1, symmetric=TRUE )
```

## Arguments

| | |
|---|---|
| `HVC0` | a numeric Nx3 matrix with HVC values in the rows, or a vector that can be converted to such a matrix, by row. `HVC0` can also be a numeric 3-vector with a single HVC, and it is then replicated to match the size of `HVC1`.<br>`HVC0` can also be a character N-vector with Munsell Notations, which is converted to an Nx3 matrix using `HVCfromMunsellName()`. |
| `HVC1` | a numeric Nx3 matrix with HVC values in the rows, or a vector that can be converted to such a matrix, by row. `HVC1` can also be a numeric 3-vector with a single HVC, and it is then replicated to match the size of `HVC0`.<br>`HVC1` can also be a character N-vector with Munsell Notations, which is converted to an Nx3 matrix using `HVCfromMunsellName()`. |
| `symmetric` | if `FALSE` then use the original Nickerson difference formula, and if `TRUE` then use a symmetrized version; see **Details**. |

## Details

If `HVC0`=$H_0, V_0, C_0$ and If `HVC1`=$H_1, V_1, C_1$ then the original Nickerson formula is:

$$NCD(HVC0, HVC1) = 0.4C_0 \Delta H \ + \ 6\Delta V \ + \ 3\Delta C$$

where $\Delta H = |H_0 - H_1|$ (on the circle), $\Delta V = |V_0 - V_1|$ and $\Delta C = |C_0 - C_1|$. Unfortunately, if `HVC0` and `HVC1` are swapped, the color difference is different. The first color is considered to be the *reference color* and the second one is the *test color*. The difference is not symmetric.
Another problem is that the difference is not continuous when the second color is a neutral gray, for rectangular coordinates on a plane of constant V.

Both of these problems are fixed with a slightly modified formula:

$$NCD(HVC0, HVC1) = 0.4 \min(C_0, C_1) \Delta H \ + \ 6\Delta V \ + \ 3\Delta C$$

For the first formula set `symmetric=FALSE` and for the second formula set `symmetric=TRUE`.

## Value

A numeric N-vector with the pairwise differences, i.e. between row i of `HVC0` and row i of `HVC1`.

### Author(s)

Jose Gama and Glenn Davis

### References

Nickerson, Dorothy. The Specification of Color Tolerances. **Textile Research**. Vol 6. pp. 505-514. 1936.

### See Also

`LabtoMunsell()`, `MunsellToXYZ()`, `LabfromXYZ()`, `adaptXYZ()`, `CAT()`

### Examples

```
NickersonColorDifference( '7.6P 8.9/2.2', '8P 8.2/3'  )
```

---

## plotLociHC *Plot Curves of Constant Munsell Hue and Chroma*

---

### Description

Plot Curves of Constant Munsell Hue and Chroma

### Usage

```
plotLociHC( value=5, hue=seq(2.5,100,by=2.5), chroma='auto', coords='xy',
                          main="Value %g/", est=FALSE, ... )
```

### Arguments

| | |
|---|---|
| `value` | a Munsell value for which the plot is created. It must be in the interval (0,10]. `value` can also be a numeric vector of such numbers, and then a separate plot is made for each element of the vector. |
| `hue` | a numeric vector for which curves of contant Hue are plotted. Each of these radial curves starts at Munsell Chroma = `min(chroma,1)` and extends to `max(chroma)`. `hue` can also be a character vector, which is then converted to a numeric vector using `HueNumberFromString()`. |
| `chroma` | a numeric vector for which ovoids of constant Chroma are plotted. Each of these ovoids is closed; i.e. goes full circle from Hue=0 to Hue=100. If `chroma='auto'` then a vector is chosen appropriate for the current `value`. |
| `coords` | either `'xy'` or `'ab'`. If `coords='xy'` then the plots are in the standard xy chromaticity plane. If `coords='ab'` then the plots are in the ab chrominance plane, from Lab. Even when it is `'ab'`, the interpolation of loci takes place in `'xy'` *before* transformation to Lab. |
| `main` | a string used to set the main title of the plot. The optional placeholder `'%g'` is replaced by the current `value`. |

`est` if `TRUE`, initial estimates for the iteration used in `xyYtoMunsell()` are plotted

`...` other arguments passed to the function `MunsellToxyY()`. This includes `hcinterp`, `vinterp`, and `xyC`. However `warn=FALSE` is forced.

### Details

The plot limits (`xlim` and `ylim`) are set to include all points where the Hue radials intersect the Chroma ovoids, plus the white point.
If `value` is one of 0.2,0.4,0.6,0.8,1,2,3,4,5,6,7,8,9,10 then published points from `real.dat` are plotted with filled black points (real points), and extrapolated points from `all.dat` are drawn with open circles (unreal points).

### Value

`TRUE` for success and `FALSE` for failure.

### Note

The option `hcinterp='bicubic'` makes fairly smooth ovoids, and `hcinterp='bilinear'` makes 40-sided polygons (when `coords='xy'`). Compare with the plots in *Newhall et. al. (1943)*, *Judd, et. al. (1956)*, and *Judd, et. al. (1975)* p. 263.

### Author(s)

Glenn Davis

### References

Newhall, Sidney M., Dorothy Nickerson, Deane B. Judd. **Final Report of the O.S.A. Subcommitte on the Spacing of the Munsell Colors**. Journal of the Optical Society of America. Vol. 33. No. 7. pp. 385-418. July 1943.

Judd, Deane B. and Günther Wyszecki. **Extension of the Munsell Renotation System to Very Dark Colors**. Journal of the Optical Society of America. Vol. 46. No. 4. pp. 281-284. April 1956.

Judd, Deane B. and Günther Wyszecki. **Color in Business, Science, and Industry**. 3rd edition. John Wiley & Sons. 1975.

### See Also

`MunsellToxyY()`, `HueNumberFromString()`

---

`plotPatchesH` *Plot Colored Patches for a fixed Munsell Hue*

---

## Description

This plot simulates a page from the Munsell Book of Color. The colors are best viewed on a display calibrated for the RGB space given as the second argument.

## Usage

```
plotPatchesH( hue, space='sRGB', adapt='Bradford', background='gray50',
                            main="Hue %s  (H=%g)      [%s   adapt=%s]", ... )
```

## Arguments

| | |
|---|---|
| `hue` | a Munsell hue for which the plot is created. It is automatically wrapped to the interval (0,100]. It does not have to be a multiple of 2.5. `hue` can also be a numeric vector of such numbers, and then a separate plot is made for each element of the vector. `hue` can also be a character vector of Hue Names, which is then converted to a numeric vector using `HueNumberFromString()`. |
| `space` | the name of an installed RGB space. Spaces `'sRGB'` and `'AdobeRGB'` are pre-installed, and others can be installed with `installRGB()`. |
| `adapt` | method used to adapt xyY for Illuminant C to xyY for Illuminant D65. It is passed to `MunsellToRGB()`. |
| `background` | background color for the plot. It is passed to `par()` as argument `bg`. |
| `main` | a string used to set the main title of the plot. The optional placeholder `'%s'` is replaced by the Hue Name, `'%g'` is replaced by the Hue Number, the next `'%s'` is replaced by `space`, and the last `'%s'` is replaced by `adapt`. |
| `...` | other arguments passed to the function `MunsellToRGB()`. This includes `hcinterp`, `vinterp`, and `xyC`. |

## Details

The discrete Values are always the same: integers from 0 to 10. And so the plotting parameter `ylim=c(0,11)`.
The discrete Chromas are contiguous even integers depending on the Hue, and determined as follows. For the closest discrete Hue in `real.dat`, the patches in `real.dat` are transformed to xyY using simple lookup. These are then tested against the MacAdam Limits for Illuminant C using `IsWithinMacAdamLimits()`. The patches outside the limits are discarded, and the maximum Chroma of the remaining patches, which is always an even integer, determines `xlim`.
Patches inside the MacAdam Limits can still be outside the RGB cube. Patches inside the cube are drawn in the usual way, and those outside are drawn in outline only, and with the clamped RGB coordinates printed inside.

## Value

`TRUE` for success and `FALSE` for failure.

### Author(s)

Glenn Davis

### See Also

`MunsellToRGB()`,`HueNumberFromString()`,`IsWithinMacAdamLimits()`,`installRGB()`

---

`RGBtoMunsell` *Convert sRGB coordinates to a Munsell specification*

---

### Description

`RGBtoMunsell` Converts RGB coordinates to a Munsell specification, by interpolating over the extrapolated Munsell renotation data

### Usage

```
RGBtoMunsell( RGB, space='sRGB', maxSignal=255, adapt='Bradford', ... )
```

### Arguments

| | |
|---|---|
| `RGB` | a numeric Nx3 matrix with RGB coordinates in the rows, or a vector that can be converted to such a matrix, by row. These are non-linear display values, but they are not required to be integers. |
| `space` | the name of an installed RGB space. Spaces `'sRGB'` and `'AdobeRGB'` are pre-installed, and others can be installed with `installRGB()`. |
| `maxSignal` | maximum value of RGB for display. Other popular values are 1, 1023, and 65535. Even when 1, they are still taken to be non-linear display values. |
| `adapt` | method for chromatic adaptation, see `CAT()` for valid values. Also see **Details**. |
| `...` | other parameters passed to `XYZtoMunsell()` |

### Details

The conversion is done in 3 steps.

- RGB → XYZ using `XYZfromRGB()` with the given `space` and `maxSignal`
- XYZ is adapted from the white-point of `space` to Illuminant C using the given chromatic adaptation method
- XYZ → HVC using `XYZtoMunsell()`

### Value

a numeric Nx3 matrix with HVC coordinates in the rows. The rownames are copied from input to output.
In case of error, it returns `NULL`.

### Author(s)

Jose Gama and Glenn Davis

### References

Wikipedia. **sRGB**. https://en.wikipedia.org/wiki/SRGB.

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html

### See Also

XYZfromRGB(), XYZtoMunsell(), CAT()

### Examples

```
RGBtoMunsell( c(255,45,67) )
##                   H        V        C
##  5.4R 5.5/18 5.401135 5.477315 18.01984

RGBtoMunsell( c(255,45,67), space='Adobe' )
##                   H        V        C
##  5.9R 6.2/22 5.924749 6.214155 21.83907
```

---

## sRGBtoMunsell *Convert sRGB coordinates to a Munsell specification*

---

### Description

Converts non-linear sRGB coordinates to a Munsell specification, by interpolating over the extrapolated Munsell renotation data

### Usage

```
sRGBtoMunsell( sRGB, maxSignal=255, ... )
```

### Arguments

| | |
|---|---|
| sRGB | a numeric Nx3 matrix with signal sRGB coordinates in the rows, or a vector that can be converted to such a matrix, by row. These are non-linear signal values, but they are not required to be integers. |
| maxSignal | maximum value of signal sRGB. Other popular values are 1, 1023, and 65535. Even when 1, they are still taken to be non-linear signal values. |
| ... | other parameters passed to XYZtoMunsell() |

### Details

The conversion is done in 3 steps.

- sRGB → XYZ using `XYZfromRGB()` with the given `maxSignal`
- XYZ is adapted from Illuminant D65 (from the sRGB standard) to Illuminant C using the *Bradford* chromatic adaptation method (CAT)
- XYZ → HVC using `XYZtoMunsell()`

### Value

a numeric Nx3 matrix with HVC coordinates in the rows. The rownames are copied from input to output.

### Note

The function `RGBtoMunsell()` also performs this conversion. The only reason to use this one is that it takes a little less time, since the CAT (using the *Bradford* method) is precomputed.

### Author(s)

Jose Gama and Glenn Davis

### References

Wikipedia. **sRGB**. `https://en.wikipedia.org/wiki/SRGB`.

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

### See Also

`XYZfromRGB()`, `CAT()`, `XYZtoMunsell()`, `RGBtoMunsell()`

### Examples

```
sRGBtoMunsell( c(255,45,67) )
##                      H        V        C
##  5.4R 5.5/18 5.401135 5.477315 18.01984

sRGBtoMunsell( c(1,0,1), maxSignal=1 )
##                  H        V        C
##  8P 6/26 87.98251 5.981297 25.64534
```

## VandY — *Convert Munsell Value V to Luminance Factor Y, and back again*

### Description

Convert non-linear Munsell Value V to linear Luminance Factor Y, and back again

### Usage

```
YfromV( V, which='ASTM' )
VfromY( Y, which='ASTM' )
```

### Arguments

| | |
|---|---|
| `V` | a numeric vector with elements in the interval [0,10] |
| `Y` | a numeric vector with elements in the interval [0,100] |
| `which` | one of the strings in the first column of this table |

| | reference |
|---|---|
| 'ASTM' | ASTM D-1535 (2008) |
| 'OSA' | Newhall, et. al. (1943) |
| 'MgO' | Newhall, et. al. (1943) |
| 'Munsell' | Munsell, et. al. (1933) |
| 'Priest' | Priest, et. al. (1920) |

Partial matching is enabled, and it is case insensitive.

### Details

`'Priest'` is the earliest (1920) transfer function in this package. It is implemented as:

$$V = sqrt(Y) \quad and \quad Y = V^2$$

One readily checks that when V=10, Y=100, and vice-versa. This transfer function has been implemented in colorimeters, using analog electric circuits. It is used in Hunter Lab - the precursor of CIE Lab.

`'Munsell'` is the next (1933) transfer function, and was proposed by Munsell's son (Alexander Ector Orr Munsell) and co-workers. It is implemented as:

$$V = sqrt(1.474 * Y - 0.00474 * Y^2)$$

$$Y = 50 * ((1474 - sqrt(1474^2 - 4 * 4740 * V^2))/474)$$

One readily checks that when V=10, Y=100, and vice-versa. The luminance factor Y is *absolute*, AKA *relative to the perfect reflecting diffuser*.

`'Priest'` and `'Munsell'` are included in this package for historical interest only.

The remaining three define Y as a quintic polynomial in V.

The next one historically - `'MgO'` - is implemented as:

$$Y = (((((8404 * V - 210090) * V + 2395100) * V - 2311100) * V + 10000000) * V)/10000000$$

One readily checks that when V=10, Y=102.568. This Y is larger than 100, because the authors decided to make Y relative to a clean surface of MgO, instead of the perfect reflecting diffuser. In their words:

- It should be noted that the reflectances indicated are not absolute but relative to magnesium oxide; whereas the maximum at value 10/ was formerly 100 percent, it is now 102.57. Use of this relation facilitates results and also avoids the somewhat dubious conversion to absolute scale, by permitting Y determinations with a MgO standard to be converted directly to Munsell value.

Nowadays, the perfect reflecting diffuser is preferred over MgO. For users who would like to modify this quintic as little as possible, with the perfect reflecting diffuser in mind (going back to `'MUNSELL'`), I offer `'OSA'`, which is given by this quintic of my own design

$$Y = (((((8404 * V - 210090) * V + 2395100) * V - 2311100) * V + 10000000) * V)/10256800$$

ASTM had a similar modification in mind, but did it a little differently by scaling each coefficient. `'ASTM'` is given by this quintic:

$$Y = (((((81939 * V - 2048400) * V + 23352000) * V - 22533000) * V + 119140000) * V)/1.e8$$

One readily checks that when V=10, Y=100 exactly (for both `'OSA'` and `'ASTM'`).

The inverses - from Y to V - of all 3 quintics are implemented as 3 `splinefun()`s at a large number (about 300) of points. These inverses are both fast and accurate. The round-trip Y $\rightarrow$ V $\rightarrow$ Y is accurate to 7 digits after the decimal. The round-trip V $\rightarrow$ Y $\rightarrow$ V is accurate to 8 digits after the decimal.

## Value

a numeric vector the same length as the input

## Note

The quintic functions `'ASTM'` and `'OSA'` are very close. They agree at the endpoints 0 and 10 exactly, and the largest difference is near V=6.767 where they differ by about 0.0007.

## Author(s)

Glenn Davis

### References

Priest, I. G. Gibson, K. S. and McNicholas, H. J. **An Examination of the Munsell Color System. I. Spectral and and Total Reflection and the Munsell Scale of Value**. Technologic Papers of the Bureau of Standards, No. 167. pp. 1-33. Washington D.C. 1920.

Munsell, A. E. O., L. L. Sloan, and I. H. Godlove. **Neutral Value Scales. I. Munsell Neutral Value Scale**. Journal of the Optical Society of America. Vol. 23. Issue 11. pp. 394-411. November 1933.

Newhall, Sidney M., Dorothy Nickerson, Deane B. Judd. **Final Report of the O.S.A. Subcommitte on the Spacing of the Munsell Colors**. Journal of the Optical Society of America. Vol. 33. No. 7. pp. 385-418. July 1943.

ASTM D 1535-08. Standard Practice for Specifying Color by the Munsell System. 2008

### See Also

`MunsellToxyY()`, `xyYtoMunsell()`,

### Examples

```
VfromY( c(0,50,100) )
```

---

`xyYtoMunsell` *Convert xyY coordinates into a Munsell specification*

---

### Description

`xyYtoMunsell` Converts xyY coordinates into a Munsell specification, by interpolating over the extrapolated Munsell renotation data

### Usage

```
xyYtoMunsell( xyY, xyC='NBS', hcinterp='bicubic', vinterp='cubic',
                     VfromY='ASTM', warn=TRUE, perf=FALSE )
```

### Arguments

| | |
|---|---|
| `xyY` | a numeric Nx3 matrix with CIE xyY coordinates in the rows, or a vector that can be converted to such a matrix, by row. These are for viewing in an environment with Illuminant C, with Y=100. |
| `xyC` | a numeric 2-vector with xy chromaticity of Illuminant C. It can also be one of the strings given in `MunsellToxyY()`. |
| `hcinterp` | either `'bicubic'` or `'bilinear'` (partial matching enabled). See `MunsellToxyY()` for details. |
| `vinterp` | either `'cubic'` or `'linear'` (partial matching enabled). See `MunsellToxyY()` for details. |

| | |
|---|---|
| `VfromY` | passed as the parameter `which` to the function `VfromY()`. See `VfromY()` for details. Option `'MGO'` is not allowed because then Y>100 when V=10. |
| `warn` | if an xyY cannot be mapped (usually because the root finder has wandered afar), its H and V are set to `NA` in the returned `data.frame`. Just before returning, if any rows have `NA`, and this argument is `TRUE`, then a warning is logged. |
| `perf` | if `perf` is `TRUE`, then extra performance related metrics are appended to the returned `data.frame`, see **Value**. |

## Details

See `MunsellToxyY()` and the **User Guide - Appendix C**.

## Value

a `data.frame` with N rows and these columns:

| | |
|---|---|
| `xyY` | The input xyY |
| `HVC` | the computed HVC. H is automatically wrapped to (0,100]. In case of failure, H and C are `NA`. |
| `SAMPLE_NAME` | a character vector - the Munsell notation for HVC |

If `perf` is `TRUE` then there are these additional columns:

| | |
|---|---|
| `time.elapsed` | elapsed time in seconds. If available, the function `microbenchmark::get_nanotime()` is used. |
| `iterations` | the number of iterations of `rootSolve::multiroot()` |
| `evalations` | the number of forward (HVC $\rightarrow$ xyY) function evaluations |
| `estim.precis` | the estimated precision from `rootSolve::multiroot()`. This is in the HC plane for the Munsell Value easily computed from Y. |

If the rownames of xyY are not `NULL` and have no duplicates, they are copied to the returned data frame.

## Warning

Even when `vinterp='cubic'` the function xyY $\rightarrow$ HVC is not $C^1$ on the plane V=1. This is because of a change in Value spacing: when V$\geq$1 the Value spacing is 1, but when V$\leq$1 the Value spacing is 0.2.

## Author(s)

Jose Gama and Glenn Davis

## Source

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`
`http://www.rit.edu/science/pocs/renotation`
`http://www.rit-mcsl.org/MunsellRenotation/all.dat`
`http://www.rit-mcsl.org/MunsellRenotation/real.dat`

### References

Judd, Deane B. **The 1931 I.C.I. Standard Observer and Coordinate System for Colorimetry**. Journal of the Optical Society of America. Vol. 23. pp. 359-374. October 1933.

Newhall, Sidney M., Dorothy Nickerson, Deane B. Judd. **Final Report of the O.S.A. Subcommitte on the Spacing of the Munsell Colors**. Journal of the Optical Society of America. Vol. 33. No. 7. pp. 385-418. July 1943.

Kelly, Kenneth L. Kasson S. Gibson. Dorothy Nickerson. **Tristimulus Specification of the Munsell Book of Color from Spectrophometric Measurements** National Bureau of Standards RP1549 Volume 31. August 1943.

Judd, Deane B. and Günther Wyszecki. **Extension of the Munsell Renotation System to Very Dark Colors**. Journal of the Optical Society of America. Vol. 46. No. 4. pp. 281-284. April 1956.

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox `http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html`

### See Also

`MunsellToxyY()`, `multiroot()`, `get_nanotime()`

### Examples

```
xyYtoMunsell(c(0.310897, 0.306510, 74.613450))
##        xyY.1     xyY.2     xyY.3     HVC.H     HVC.V     HVC.C    SAMPLE_NAME
## 1  0.310897  0.306510 74.613450 87.541720  8.900000  2.247428 7.5P 8.9/2.2
```

---

`XYZtoMunsell` *Convert XYZ coordinates to a Munsell specification*

---

### Description

`XYZtoMunsell` Convert XYZ coordinates to a Munsell specification, by interpolating over the extrapolated Munsell renotation data

### Usage

```
XYZtoMunsell( XYZ, ... )
```

### Arguments

| | |
|---|---|
| `XYZ` | a numeric Nx3 matrix with CIE XYZ coordinates in the rows, or a vector that can be converted to such a matrix, by row. The XYZ are for viewing in an environment with Illuminant C. |
| `...` | other parameters passed to `xyYtoMunsell()` |

### Details

the function calls `XYZ2xyY()` and `xyYtoMunsell()`.

### Value

an Nx3 matrix with Munsell HVC in the rows. The rownames are copied from input to output.

### Author(s)

Jose Gama and Glenn Davis

### References

Paul Centore 2014 The Munsell and Kubelka-Munk Toolbox http://centore.isletech.net/~centore/MunsellAndKubelkaMunkToolbox/MunsellAndKubelkaMunkToolbox.html

### See Also

`MunsellToXYZ()`

### Examples

```
XYZtoMunsell(c(0.310897, 0.306510, 74.613450))
```