

# Package ‘munsell’

June 12, 2018

**Type** Package

**Title** Utilities for Using Munsell Colours

**Version** 0.5.0

**Author** Charlotte Wickham <cwickham@gmail.com>

**Maintainer** Charlotte Wickham <cwickham@gmail.com>

**Description** Provides easy access to, and manipulation of, the Munsell colours. Provides a mapping between Munsell's original notation (e.g. ``5R 5/10'') and hexadecimal strings suitable for use directly in R graphics. Also provides utilities to explore slices through the Munsell colour tree, to transform Munsell colours and display colour palettes.

**Suggests** ggplot2, testthat

**Imports** colorspace, methods

**License** MIT + file LICENSE

**URL** <https://cran.r-project.org/package=munsell>,  
<https://github.com/cwickham/munsell/>

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-06-12 04:29:06 UTC

## R topics documented:

chroma_slice . . . . .	2
complement . . . . .	3
complement_slice . . . . .	3
darker . . . . .	4
desaturate . . . . .	5
hue_slice . . . . .	5
hvc2mns1 . . . . .	6
lighter . . . . .	7

mns1 . . . . .	7
mns12hvc . . . . .	8
mns1_hues . . . . .	9
munsell . . . . .	9
pbgyr . . . . .	10
plot_closest . . . . .	10
plot_hex . . . . .	11
plot_mns1 . . . . .	12
rgb2mns1 . . . . .	12
rygbp . . . . .	13
saturate . . . . .	14
seq_mns1 . . . . .	14
value_slice . . . . .	15

<b>Index</b>	<b>16</b>
--------------	-----------

---

chroma_slice	<i>Plot all colours with the same chroma</i>
--------------	--

---

## Description

Plots slices of the Munsell colour system where chroma is constant.

## Usage

```
chroma_slice(chroma.name = seq(0, 26, by = 2), back.col = "white")
```

## Arguments

chroma.name	integer vector of the desired values.
back.col	colour for the background

## Value

ggplot object

## Examples

```
chroma_slice(2)
chroma_slice(18)
# Maybe want to delete text and add axis instead
p <- chroma_slice(18)
p$layers[[2]] <- NULL # remove text layer
p + ggplot2::theme(axis.text = ggplot2::element_text(),
  axis.text.x = ggplot2::element_text(angle = 90, hjust = 1))
# all values
## Not run: chroma_slice(seq(0, 38, by = 2))
```

---

complement	<i>Find the complement of a munsell colour</i>
------------	--

---

**Description**

Finds the munsell colour with the same chroma and value but on the opposite side of the hue circle. The complement is not defined for greys (hue == "N"), and the function returns the grey untransformed.

**Usage**

```
complement(col, ...)
```

**Arguments**

col	character vector of Munsell colours
...	deprecated

**Value**

character vector of Munsell colours

**Examples**

```
complement("5PB 2/4")
cols <- c("5PB 2/4", "5Y 7/8")
plot_mnsl(c(cols, complement(cols)))
```

---

complement_slice	<i>A vertical slice through the Munsell space</i>
------------------	---

---

**Description**

Plot a hue and its complement at all values and chromas

**Usage**

```
complement_slice(hue.name, back.col = "white")
```

**Arguments**

hue.name	character string of the desired hue.
back.col	colour for the background

**Value**

ggplot object

**Examples**

```
complement_slice("5PB")
complement_slice("5R")
complement_slice("10G")
```

---

darker

*Make a munsell colour darker*

---

**Description**

Decreases the value of the Munsell colour by 1.

**Usage**

```
darker(col, steps = 1)
```

**Arguments**

col	character vector of Munsell colours
steps	number of steps to take in decreasing value

**Value**

character vector of Munsell colours

**Examples**

```
darker("5PB 3/4")
cols <- c("5PB 3/4", "5Y 7/8")
p <- plot_mnsl(c(cols, darker(cols), darker(cols, 2)))
p + ggplot2::facet_wrap(~ names, ncol = 2)
```

---

desaturate	<i>Make a munsell colour less saturated</i>
------------	---

---

**Description**

Decreases the chroma of the Munsell colour by one step steps (multiples of 2).

**Usage**

```
desaturate(col, steps = 1)
```

**Arguments**

col	character vector of Munsell colours
steps	number of steps to take in decreasing chroma

**Value**

character vector of Munsell colours

**Examples**

```
desaturate("5PB 2/4")
cols <- c("5PB 2/6", "5Y 7/8")
p <- plot_mnsl(c(cols, desaturate(cols), desaturate(cols, 2)))
p + ggplot2::facet_wrap(~ names, ncol = 2)
```

---

hue_slice	<i>Plot all colours with the same hue</i>
-----------	---

---

**Description**

Plots slices of the Munsell colour system where hue is constant.

**Usage**

```
hue_slice(hue.name = "all", back.col = "white")
```

**Arguments**

hue.name	character vector of the desired hues. Or "all" for all hues.
back.col	colour for the background

**Value**

ggplot object

## Examples

```
hue_slice("5R")
hue_slice(c("5R", "5P"))
## Not run: hue_slice("all")
```

---

hvc2mns1

*Converts a hue, chroma and value to a Munsell colour*

---

## Description

Takes separate specifications of hue, value and chroma and returns the text specification of that colour.

## Usage

```
hvc2mns1(hue, value = NULL, chroma = NULL, ...)
```

## Arguments

hue	a character vector of Munsell hues, or a 3 column data frame containing the hue value and chroma levels
value	a numeric vector of values
chroma	a numeric vector of chromas
...	passed on to <a href="#">check_mns1</a> . Use <code>fix = TRUE</code> to fix "bad" colours

## Details

Munsell colours are specified by hue, value and chroma. They take a form like "5PB 5/10" where the first characters represent the hue, followed by a space then the value and chroma separated by a "/". In this package value should be an integer in 0:10 and chroma an even number at most 24. Note that not all possible specifications result in representable colours. Regular recycling rules apply.

## Value

a character string specification of a hex colour

## See Also

[check\\_mns1](#), [mns12hex](#)

## Examples

```
hvc2mns1("5PB", 5, 10)
# All values of 5PB with chroma 10
hvc2mns1("5PB", 1:9, 10) # note some are undefined
plot_mns1(hvc2mns1("5PB", 1:9, 10))
```

---

lighter	<i>Make a munsell colour lighter</i>
---------	--------------------------------------

---

**Description**

Increases the value of the Munsell colour.

**Usage**

```
lighter(col, steps = 1)
```

**Arguments**

col	character vector of Munsell colours
steps	number of steps to take in increasing value

**Value**

character vector of Munsell colours

**Examples**

```
lighter("5PB 2/4")
cols <- c("5PB 2/4", "5Y 6/8")
p <- plot_mnsl(c(cols, lighter(cols), lighter(cols, 2)))
p + ggplot2::facet_wrap(~ names, ncol = 2)
# lighter and darker are usually reversible
lighter(darker("5PB 2/4"))
# unless you try to pass though white or black
lighter(darker("5PB 1/4"))
```

---

mns1	<i>Converts a Munsell colour to hex</i>
------	---

---

**Description**

Take a character string representation of a Munsell colour and returns the hex specification of that colour

**Usage**

```
mns1(col, ...)
```

**Arguments**

col	a character string representing a Munsell colour.
...	passed on to <a href="#">in_gamut</a> . Use <code>fix = TRUE</code> to fix "bad" colours

**Details**

Munsell colours are specified by hue, value and chroma. They take a form like "5PB 5/10" where the first characters represent the hue, followed by a space then the value and chroma separated by a "/". In this package value should be an integer in 0:10 and chroma an even number at most 24. Note that not all possible specifications result in representable colours.

**Value**

a character string specification of a hex colour

**See Also**

[check\\_mns1,in\\_gamut](#), [hvc2mns1](#)

**Examples**

```
mns12hex("5PB 5/10")
# use a munsell colour in a plot
plot.new()
rect(0, 0, 1 ,1 , col = mns1("5R 5/10"))
```

---

mns12hvc

---

*Converts a Munsell colour to a hue, chroma and value triplet*


---

**Description**

Takes a text specification of a Munsell colour and returns the hue, chroma and value triplet.

**Usage**

```
mns12hvc(col, ...)
```

**Arguments**

`col` a character vector of Munsell colours  
`...` passed on to [check\\_mns1](#). Use `fix = TRUE` to fix "bad" colours

**Details**

Munsell colours are specified by hue, value and chroma. They take a form like "5PB 5/10" where the first characters represent the hue, followed by a space then the value and chroma separated by a "/". In this package value should be an integer in 0:10 and chroma an even number at most 24. Note that not all possible specifications result in representable colours.

**Value**

a data frame with named columns hue, value and chroma containing the hue, value and chroma levels.



**See Also**

[check\\_mns1](#), [hvc2mns1](#)

**Examples**

```
mns12hvc("5PB 5/10")
hvc2mns1(mns12hvc("5PB 5/10"))
```

---

mns1\_hues

*Munsell hues*


---

**Description**

Returns a character vector of the Munsell hues in hue order starting at 2.5R and excluding grey ("N").

**Usage**

```
mns1_hues()
```

**Value**

a character vector containing the hue values.

**Examples**

```
mns1_hues()
```

---

munsell

*Munsell colour system.*


---

**Description**

This package makes it easy to access and manipulate the colours in the munsell colour system. The conversion from munsell specifications to sRGB based on the renotation data from <http://www.cis.rit.edu/mcsl/online/munsell.php> which is a digitization of Table 1 in Newhall, Nickerson & Judd (1943). The code for conversion can be found in the package directory in `inst/raw/getmunsellmap.r`

**References**

S. M. Newhall, D. Nickerson, and D. B. Judd. Final report of the O.S.A. subcommittee on the spacing of the munsell colors. *J. Opt. Soc. Am.*, 33(7):385-411, 07 1943.

Munsell Renotation Data, RIT Munsell Color Science Laboratory. <http://www.cis.rit.edu/mcsl/online/munsell.php>

---

pbgyr

*Change the hue of a munsell colour*

---

### Description

Moves the hue of a munsell colour in the direction purple->blue->green->yellow->red->purple

### Usage

```
pbgyr(col, steps = 1)
```

### Arguments

col                    character vector of Munsell colours  
 steps                 number of hue steps to take

### Value

character vector of Munsell colours

### Examples

```
my_red <- "2.5R 4/8"
pbgyr(my_red)
plot_mnsl(c(my_red, pbgyr(my_red, 2), pbgyr(my_red, 4)))
```

---

plot\_closest

*Plot closest Munsell colour to an sRGB colour*

---

### Description

Take an sRGB colour and plots it along with the closest Munsell colour (using [rgb2mnsl](#) to find it)

### Usage

```
plot_closest(R, G = NULL, B = NULL, back.col = "white")
```

### Arguments

R                     a numeric vector of red values or a 3 column matrix with the proportions R, G, B in the columns.  
 G                     numeric vector of green values  
 B                     numeric vector of blue values  
 back.col             colour for the background

**Value**

ggplot object

**See Also**

[rgb2mns1](#)

**Examples**

```
plot_closest(0.1, 0.1, 0.3)
plot_closest(matrix(c(.1, .2, .4, .5, .6, .8), ncol = 3))
```

---

plot_hex	<i>Plot hex colours</i>
----------	-------------------------

---

**Description**

Quick way to look at a set of hex colours.

**Usage**

```
plot_hex(hex.colour, back.col = "white")
```

**Arguments**

hex.colour	character vector specifying colours in hex form
back.col	specification of background colour of display

**Value**

A ggplot object

**Examples**

```
plot_hex("#000000")
plot_hex(c("#000000", "#FFFFFF"))
```

---

plot\_mns1 *Plot a munsell colour*

---

### Description

Takes munsell text specifications and plots colour squares of them.

### Usage

```
plot_mns1(cols, back.col = "white", ...)
```

### Arguments

cols	character vector specifying colours in Munsell form
back.col	specification of background colour of display
...	passed to <code>check_mns1</code> . Add <code>fix = TRUE</code> to fix "bad" colours()

### Value

A ggplot object

### Examples

```
plot_mns1("5R 5/6")
plot_mns1("5R 5/6", back.col = "grey40")
p <- plot_mns1(c("5R 6/6", "5Y 6/6", "5G 6/6", "5B 6/6", "5P 6/6"),
  back.col = "grey40")
p
# returned object is a ggplot object so we can alter the layout
summary(p)
p + ggplot2::facet_wrap(~ num, nrow = 1)
```

---

rgb2mns1 *Converts an sRGB colour to Munsell*

---

### Description

Finds the closest Munsell colour (in LUV space) to the specified sRGB colour

### Usage

```
rgb2mns1(R, G = NULL, B = NULL)
```

**Arguments**

R	a numeric vector of red values or a 3 column matrix with the proportions R, G, B in the columns.
G	numeric vector of green values
B	numeric vector of blue values

**See Also**

[plot\\_closest](#)

**Examples**

```
rgb2mns1(0.1, 0.1, 0.3)
rgb2mns1(matrix(c(.1, .2, .4, .5, .6, .8), ncol = 3))
plot_closest(matrix(c(.1, .2, .4, .5, .6, .8), ncol = 3))
```

---

rygbp

*Change the hue of a munsell colour*

---

**Description**

Moves the hue of a munsell colour in the direction red->yellow->green->blue->purple->red

**Usage**

```
rygbp(col, steps = 1)
```

**Arguments**

col	character vector of Munsell colours
steps	number of hue steps to take

**Value**

character vector of Munsell colours

**Examples**

```
my_red <- "10R 4/8"
rygbp(my_red)
plot_mns1(c(my_red, rygbp(my_red, 2), rygbp(my_red, 4)))
```

saturate *Make a munsell colour more saturated*

---

**Description**

Increases the chroma of the Munsell colour by step steps (multiples of 2).

**Usage**

```
saturate(col, steps = 1)
```

**Arguments**

col                    character vector of Munsell colours  
steps                  number of steps to take in increasing chroma

**Value**

character vector of Munsell colours

**Examples**

```
saturate("5PB 2/4")  
cols <- c("5PB 2/2", "5Y 7/6")  
p <- plot_mnsl(c(cols, saturate(cols), saturate(cols, 2)))  
p + ggplot2::facet_wrap(~ names, ncol = 2)
```

---

seq\_mnsl *Generate a sequence of Munsell colours*

---

**Description**

Generates a sequence of Munsell colours. The sequence is generated by finding the closest munsell colours to a equidistant sequence of colours in # LUV space.

**Usage**

```
seq_mnsl(from, to, n, fix = FALSE)
```

**Arguments**

from                  character string of first Munsell colour  
to                    character string of last Munsell colour  
n                      number of colours in sequence  
fix                    Should colours outside of the gamut be fixed? Passed on to [fix\\_mnsl](#)

**Value**

character vector of Munsell colours

**Examples**

```
seq_mnsl("5R 2/4", "5R 5/16", 4)
plot_mnsl(seq_mnsl("5R 2/4", "5R 5/16", 4))
plot_mnsl(seq_mnsl("5R 5/6",
  complement("5R 5/6"), 5))
```

---

value\_slice

*Plot all colours with the same value*

---

**Description**

Plots slices of the Munsell colour system where value is constant.

**Usage**

```
value_slice(value.name = 1:10, back.col = "white")
```

**Arguments**

value.name	integer vector of the desired values.
back.col	colour for the background

**Value**

ggplot object

**Examples**

```
value_slice(2)
value_slice(c(2, 4))
# all values
## Not run: value_slice(1:10)
```

# Index

[check\\_mnsl](#), [6](#), [8](#), [9](#), [12](#)  
[chroma\\_slice](#), [2](#)  
[complement](#), [3](#)  
[complement\\_slice](#), [3](#)  
  
[darker](#), [4](#)  
[desaturate](#), [5](#)  
  
[fix\\_mnsl](#), [14](#)  
  
[hue\\_slice](#), [5](#)  
[hvc2mnsl](#), [6](#), [8](#), [9](#)  
  
[in\\_gamut](#), [7](#), [8](#)  
  
[lighter](#), [7](#)  
  
[mnsl](#), [7](#)  
[mnsl2hex](#), [6](#)  
[mnsl2hex \(mnsl\)](#), [7](#)  
[mnsl2hvc](#), [8](#)  
[mnsl\\_hues](#), [9](#)  
[munsell](#), [9](#)  
[munsell-package \(munsell\)](#), [9](#)  
  
[package-munsell \(munsell\)](#), [9](#)  
[pbgvr](#), [10](#)  
[plot\\_closest](#), [10](#), [13](#)  
[plot\\_hex](#), [11](#)  
[plot\\_mnsl](#), [12](#)  
  
[rgb2mnsl](#), [10](#), [11](#), [12](#)  
[rygbp](#), [13](#)  
  
[saturate](#), [14](#)  
[seq\\_mnsl](#), [14](#)  
  
[value\\_slice](#), [15](#)