

# Package ‘NVIpretty’

June 13, 2025

**Title** Tools for making R-output pretty in accord with NVI's graphical profile

**Version** 0.5.0

**Date** 2025-06-13

**Description** Provides tools for styling output from R in accord with NVI's graphical profile. Included are colour palettes and a NVI-theme for ggplot2 as well as tools to format and style output in Excel.

**URL** <https://github.com/NorwegianVeterinaryInstitute/NVIpretty>

**BugReports** <https://github.com/NorwegianVeterinaryInstitute/NVIpretty/issues>

**Depends** R (>= 4.1.0)

**License** BSD\_3\_clause + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** checkmate (>= 2.1.0),  
ggplot2,  
grid,  
openxlsx,  
NVIcheckmate (>= 0.5.0),  
NVIdb (>= 0.13.1)

**Suggests** covr,  
datasets,  
desc,  
devtools,  
knitr,  
remotes,  
rmarkdown,  
roxygen2,  
testthat (>= 3.0.0),  
NVIpackager,  
NVIrpackages,  
R.rsp

**Remotes** NorwegianVeterinaryInstitute/NVIcheckmate,  
NorwegianVeterinaryInstitute/NVIdb,  
NorwegianVeterinaryInstitute/NVIpackager,  
NorwegianVeterinaryInstitute/NVIrpackages

**VignetteBuilder** knitr, R.rsp

**Language** en-GB  
**Roxygen** list(markdown = FALSE)  
**RoxygenNote** 7.3.2  
**Config/testthat/edition** 3

Contents

add_formatted_worksheet . . . . .	2
append_text_line . . . . .	5
palette_gradient_green . . . . .	6
palette_graph . . . . .	9
palette_graph_2020 . . . . .	10
palette_NVI . . . . .	10
palette_NVI_profile_2020 . . . . .	11
palette_table_blue_2020 . . . . .	12
palette_table_grey_2020 . . . . .	12
style_background_per_column . . . . .	13
style_text_line . . . . .	15
theme_NVI . . . . .	16
<b>Index</b>	<b>18</b>

---

add_formatted_worksheet
<i>Add excel sheet with a formatted header</i>

---

Description

Add excel sheet with a header formatted with bold font, column labels and column width. The function is a wrapper for `openxlsx::addWorksheet` and are using `NVIdb::standardize_columns` to standardize columns.

Usage

```
add_formatted_worksheet(  
  data,  
  workbook,  
  sheet,  
  wrapHeadlineText = FALSE,  
  collabels = TRUE,  
  colwidths = TRUE,  
  standards = NULL,  
  dbsource = deparse(substitute(data)),  
  FUN = NULL,  
  ...  
)
```

**Arguments**

data	[data.frame] The data to export to the Excel sheet.
workbook	[work book object] The workbook object used to create the Excel workbook.
sheet	[character(1)] The Excel sheet name.
wrapHeadlineText	[logical(1)] Should headline allow wrapping of text. Defaults to FALSE.
collabels	[logical(1)] Should headline be changed to standard labels. Defaults to TRUE.
colwidths	[logical(1)] or "auto" Should defined standard column widths be used. Defaults to TRUE.
standards	[data.frame   list   character(1)] The column standards to be used as input for <code>NVIdb::standardize_columns</code> when formatting the Excel sheet columns, see details. Defaults to NULL.
dbsource	[character(1)] Database source of data in column standards table. Defaults to name of input data.
FUN	[function] Function for additional formatting of the worksheet. Either predefined functions in this package or self made. Defaults to NULL.
...	Other arguments to be passed to FUN.

**Details**

Add excel sheet with a formatted header. The header will be formatted with bold text.

`collabels = TRUE` the column names are translated to column labels in accord with the column standards table, see `NVIdb::standardize_columns`.

`colwidths = TRUE` the column widths are given in accord with the column standards table, see `NVIdb::standardize_columns`.

`colwidths = "auto"` the column widths are given automatic column width. This is not recommend for large tables.

`wrapHeadlineText = TRUE` the headline text is allowed to wrap on two or more lines. The parameter should be chosen in accord with what looks nice depending on column labels and column widths.

`standards` is the name of the table or file with column standards. If `standards = NULL` the file "columns\_standards.csv" is used. Column names are translated to column labels in accord with the column standards table, see `NVIdb::standardize_columns`.

It is also possible to give the standards as a `data.frame` or as a `list`. The list input to `column_standards` must follow a specific format. It is a list with at least three named vectors:

- `colname`: a vector of all columns in in the source file that should be included in the Excel report with the selection list.
- `collabel`: A vector with the labels that should be used in the Excel report.
- `colwidth`: A vector with the column width that should be used in the Excel report.

In addition one may input:

- `colorder`: the order of the columns to be used in the Excel report. The default is to use the same order as they are entered in the vectors.
- `column_db`: input added as a possibility to keep on the same format as `column_standards`. Not necessary to input.
- `table_db`: input added as a possibility to keep on the same format as `column_standards`. Must be the same as `dbsource`. Not necessary to input.

All vectors must have the same order and the same length. #' `dbsource` is the `dbsource` in the column standards table making it possible to tailor the column labels and column widths per table.

## Value

None. A new sheet with formatted headline is added to the workbook object.

## Author(s)

Petter Hopp Petter.Hopp@vetinst.no

## Examples

```
## Not run:
# Attach packages and set up with temporary directory
td <- tempdir()
library(NVipretty)
library(openxlsx)

# Generate Excel-sheet
workbook <- createWorkbook()

# Add a sheet to the workbook
add_formatted_worksheet(iris,
                        workbook,
                        sheet = "iris",
                        wrapHeadlineText = TRUE,
                        collabels = TRUE,
                        colwidths = FALSE,
                        standards = NULL)

# Save the workbook
saveWorkbook(wb = workbook,
             file = paste0(td, "/iris.xlsx"),
             overwrite = TRUE)

print(paste("One may examine the Excel workbook at", normalizepath(td, )))

## End(Not run)
```

---

append_text_line	<i>Append new last row with text</i>
------------------	--------------------------------------

---

### Description

Function to add new row with text in a data frame. This is intended for appending explanation text (footnotes etc.) to data frames that should be published in Excel.

### Usage

```
append_text_line(data, text, empty_rows = 0)
```

### Arguments

data	[data.frame] The data that should get a new row with the text.
text	[character(1)] The text that should be appended.
empty_rows	[integer(1)] Number of empty rows between the data and the text. Defaults to 0.

### Details

Two rows are appended to the data frame, the first is empty, the second has the generated date in the first column. The

### Value

A data frame with two more rows, one empty and one with generated data in the first column.

### Author(s)

Petter Hopp Petter.Hopp@vetinst.no

### Examples

```
## Not run:  
# Add row with generated date using standard values  
gris_virus_slaktegris_utvalg <- append_date_generated_line(gris_virus_slaktegris_utvalg)  
  
## End(Not run)
```

---

`palette_gradient_green`*Palette with colours for graphs for 2025 - today's date*

---

**Description**

A vector with the hex values for colours to be used in graphs. The order of the colours are as they should be used in the graphs.

Colour gradients with six levels for six different base colours. The colours in the NVI profile are used as base colours and the gradients are developed from these.

**Usage**`palette_gradient_green``palette_gradient_red``palette_gradient_air``palette_gradient_darkblue``palette_gradient_earth``palette_gradient_food``palette_gradient_graa``palette_gradient_grey``palette_gradient_gul``palette_gradient_jord``palette_gradient_lightblue``palette_gradient_luft``palette_gradient_mat``palette_gradient_red``palette_gradient_vann``palette_gradient_water``palette_gradient_yellow`**Format**

A vector with 15 unnamed colours:

**Jord** #59CD8B  
**Luft** #D7F4FF  
**Vann** #1C4FB9  
**Svart** #091A3E  
**Lys jord** #BCEED1  
**Mørk luft** #95D9F3  
**Lyst vann** #C7D9FF  
**Mat** #FF5447  
**Lys mat** #FFDAD4  
**Gul** #F5D34A  
**Lys gul** #FBECB2  
**Lilla** #9A55D2  
**Lys lilla** #DEC2F5  
**Orange** #FCA446  
**Lys orange** #F9D3A9

A vector with unnamed colours:

**palette\_gradient\_green** Green colour gradient based on the NVI colour "Jord".

**palette\_gradient\_jord** Alternative name for the green colour gradient based on the NVI colour "Jord".

**palette\_gradient\_earth** Alternative name for the green colour gradient based on the NVI colour "Jord".

**palette\_gradient\_darkblue** Blue colour gradient based on the NVI colour "Vann".

**palette\_gradient\_vann** Alternative name for the blue colour gradient based on the NVI colour "Vann".

**palette\_gradient\_water** Alternative name for the blue colour gradient based on the NVI colour "Vann".

**palette\_gradient\_lightblue** Blue colour gradient based on the NVI colour "Luft".

**palette\_gradient\_luft** Alternative name for the blue colour gradient based on the NVI colour "Luft".

**palette\_gradient\_air** Alternative name for the blue colour gradient based on the NVI colour "Luft".

**palette\_gradient\_red** Red colour gradient based on the NVI colour "Mat".

**palette\_gradient\_mat** Alternative name for the red colour gradient based on the NVI colour "Mat".

**palette\_gradient\_food** Alternative name for the red colour gradient based on the NVI colour "Mat".

**palette\_gradient\_yellow** Yellow colour gradient based on the NVI colour "Gul".

**palette\_gradient\_gul** Alternative name for the yellow colour gradient based on the NVI colour "Gul".

**palette\_gradient\_grey** Grey colour gradient based on the NVI colour "Svart".

**palette\_gradient\_graa** Alternative name for the grey colour gradient based on the NVI colour "Svart".

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes.R"`.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

An object of class character of length 6.

## Details

The colour codes are based on the hex values in the Graphical profile handbook (available at NVI's intranet). The colours are displayed in the vignette "NVI colour palettes".

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes.R"`.

The colour gradients are intended for choropleth maps, graphs and other figures where different gradients of one colour is wanted.

There are six different gradients, each with six colours. Each palette is given a name of the form `"palette_gradient_xxxxxx"` where `xxxxxx` is the name of the base colour. To make it easier to remember the gradient name, each gradient is represented by several palettes, for example the green gradient based on the NVI colour "Jord" is represented by the palettes: `palette_gradient_jord`, `palette_gradient_earth`, and `palette_gradient_green`.

The colours are displayed in the vignette "NVI colour palettes".

## Source

Grafisk profil for Veterinærinstituttet version 2024 (available on NVI's intranet).

Grafisk profil for Veterinærinstituttet version 2024 (available on NVI's intranet).



---

palette\_graph*Palette with NVI profile colours for 2025 - today's date*

---

### Description

A vector with the hex values for NVI profile colours.

### Usage

```
palette_graph
```

### Format

A vector with 10 named colours:

**Jord** #59CD8B

**Luft** #D7F4FF

**Vann** #1C4FB9

**Hvit** #F7FDFF

**Svart** #091A3E

**Lys\_jord** #BCEED1

**Mørk\_luft** #95D9F3

**Lyst\_vann** #C7D9FF

**Mat** #FF5447

**Lys\_mat** #FFDAD4

### Details

The colour codes are based on the hex values in the Graphical profile handbook (available at NVI's intranet). The colours are displayed in the vignette "NVI colour palettes".

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes.R"`.

### Source

Grafisk profil for Veterinærinstituttet version 2024 (available on NVI's intranet).

---

palette_graph_2020	<i>Old palette with colours for graphs for 2020-2024</i>
--------------------	--

---

**Description**

A vector with the hex values for colours to be used in graphs.

**Usage**

```
palette_graph_2020
```

**Format**

A vector with named colours:

**colours** 13 different colours

**Details**

The colour codes are based on the hex values in the Graphical profile handbook except for the body text where the cmyk value is used. There is not agreement between the hex and cmyk values, but hex seem to give the most correct colour.

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes_2020.R"`.

**Source**

Grafisk profil for Veterinærinstituttet versjon 2020.

---

palette_NVI	<i>Palette with NVI's profile colours for 2025 - today's date</i>
-------------	---

---

**Description**

A vector with the hex values for NVI's profile colours.

**Usage**

```
palette_NVI
```

**Format**

A vector with named colours:

**Jord** The green colour that signifies NVI's core area "Terrestrial animals".

**Luft** A light blue colour".

**Vann** The blue colour that signifies NVI's core area "Aquatic animals".

**Hvit** Whitish. It is used in headings with a coloured or black background.

**Svart** Blackish for use in body text in reports etc. It can also be used in headings.

**Lys jord** Used in graphs.

**Mørk luft** Used in graphs.

**Lyst vann** Used in graphs.

**Mat** The red colour that signifies NVI's core area "Food and feed safety".

**Lys mat** Used in graphs.

## Details

The colour codes are based on the hex values in the official colour profile.

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes.R"`.

## Source

Grafisk profil for Veterinærinstituttet version 2024 (available on NVI's intranet).

---

palette\_NVI\_profile\_2020

*Old palette with NVI's profile colours for 2020-2024*

---

## Description

A vector with the hex values for NVI's profile colours.

## Usage

```
palette_NVI_profile_2020
```

## Format

A vector with named colours:

**blue** The blue colour that signifies NVI's core area "Aquatic animals".

**green** The green colour that signifies NVI's core area "Terrestrial animals".

**red** The red colour that signifies NVI's core area "Food and feed safety".

**bodytext** Black 80% for use in body text in reports etc. It can also be used in headings.

**headingblue** Blue alternative for use in headings.

**background** Background colour for large areas as on stands.

## Details

The colour codes are based on the hex values except for the body text where the cmyk value is used. There is not agreement between the hex and cmyk values, but hex seem to give the most correct colour.

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes_2020.R"`.

## Source

Grafisk profil for Veterinærinstituttet versjon 2020.

---

palette\_table\_blue\_2020

*Old palette with colours for table with blue background for 2020-2024*

---

### Description

A vector with the hex values for a table with blue background.

### Usage

```
palette_table_blue_2020
```

### Format

A vector with named colours:

**blue1** Background colour for heading.

**blue2** Background colour for sub-heading.

**blue3** Background colour for sub-heading.

**blue4** Background colour for table text.

**blue5** Preferred background colour for table text.

**bodytext** Black 70% font colour for use in body text.

**heading** White font colour for headings.

### Details

The colour codes are based on the hex values in the Graphical profile handbook except for the body text where the cmyk value is used. There is not agreement between the hex and cmyk values, but hex seem to give the most correct colour.

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes_2020.R"`.

### Source

Grafisk profil for Veterinærinstituttet versjon 2020.

---

palette\_table\_grey\_2020

*Old palette with colours for table with grey background for 2020-2024*

---

### Description

A vector with the hex values for a table with grey background.

### Usage

```
palette_table_grey_2020
```

## Format

A vector with named colours:

**black1** Background colour for heading.

**black2** Background colour for sub-heading.

**black3** Background colour for sub-heading.

**black4** Background colour for table text.

**black5** Preferred background colour for table text.

**bodytext** Black 85% font colour for use in body text.

**heading** White font colour for headings.

## Details

The colour codes are based on the cmyk values in the Graphical profile handbook.

The script for preparing the palette is given in `"/data-raw/generate_NVI_palettes_2020.R"`.

## Source

Grafisk profil for Veterinærinstituttet versjon 2020.

---

`style_background_per_column`

*Style cell background in Excel sheet based on column name*

---

## Description

Style the cell background with different colours. The background colours are chosen based on the column names, for example "paavist" or "tvilsom". In each column, all cells that are following the given rule, will get the same background colour.

## Usage

```
style_background_per_column(  
  workbook = workbook,  
  sheet = sheet,  
  data = NULL,  
  colnames_in_data = NULL,  
  nrows_in_data = NULL,  
  rule,  
  colname,  
  palette  
)
```

**Arguments**

workbook	[work book object] The workbook object used to create the Excel workbook. Defaults to 'work-book'.
sheet	[character(1)] The Excel sheet name. Defaults to 'sheet'.
data	[data.frame] The data that have been exported to the Excel sheet. Used to find column number and number of rows in the workbook that should be styled. Can be NULL if both colnames_in_data and nrows_in_data are different from NULL. Defaults to NULL.
colnames_in_data	[character] The column names of the data frame that has been exported to Excel, i.e. colnames(data). Can be NULL if data has a non-NULL argument. Values in colnames_in_data have preference for values in data. Defaults to NULL.
nrows_in_data	[integer(1)] The number of rows of the data frame that has been exported to Excel, i.e. nrow(data). Can be NULL if data has a non-NULL argument. Values in nrows_in_data have preference for values in data. Defaults to NULL.
rule	[integer(1)] Rule that should be sourced to openxlsx::conditionalFormatting.
colname	[character] Vector with colour names from the palette that should be used for styling the columns. If the column name is not the same as the colour name in the palette, the colour name should be named with the column name in the data frame.
palette	[character] Palette with named colours.

**Details**

The columns will be styled with the colour in the palette that have the same name as the column. If the column name is not the same as the name of the colour in the palette, the vector of column names can be named with the corresponding colour name, see example.

**Value**

None. One or more columns in the workbook object is styled.

**Author(s)**

Petter Hopp Petter.Hopp@vetinst.no

**Examples**

```
## Not run:
# Attach packages and set up with temporary directory
td <- tempdir()
library(NVipretty)
library(openxlsx)
```

```

# Make workbook object
workbook <- createWorkbook()

# Add a sheet to the workbook
add_formatted_worksheet(iris,
                        workbook,
                        sheet = "iris",
                        wrapHeadlineText = TRUE,
                        collabels = TRUE,
                        colwidths = FALSE,
                        standards = NULL)

style_background_per_column(workbook = workbook,
                           sheet = sheet,
                           data = iris,
                           rule = "> 3",
                           colname = c("bluegrey" = "Sepal.Length",
                                         "bluegrey" = "Sepal.Width",
                                         "green" = "Petal.Length"),
                           palette = c("bluegrey" = "#6fb2d3", "green" = "#579e65"))

# Save the workbook
saveWorkbook(wb = workbook, file = paste0(td, "/iris.xlsx"), overwrite = TRUE)

print(paste("You may examine the Excel workbook at", normalizePath(td, winslash = "/")))

## End(Not run)

```

---

style\_text\_line

*Style a row in an Excel sheet based on the text in a cell*


---

## Description

Style the row if at least one cell in the row includes the specified text. This can be used to style rows with footnotes, explanatory text, summary rows or similar.

## Usage

```

style_text_line(
  workbook = workbook,
  sheet = sheet,
  data,
  text,
  text_decoration = NULL,
  wrap_text = FALSE,
  merge_cells = FALSE,
  heights = NULL,
  ...
)

```

**Arguments**

workbook	[work book object] The workbook object used to create the Excel workbook. Defaults to 'work-book'.
sheet	[character(1)] The Excel sheet name. Defaults to 'sheet'.
data	[data.frame] The data that have been exported to the Excel sheet. Used to find column number and row number for the pretext for which the row should be styled.
text	[character(1)] The text in the cell for which the row should be styled.
text_decoration	[character(1)] The text decoration style that should be used. Should be one of c("bold", "strike-out", "italic", "underline", "underline2"), see <code>openxlsx::createStyle</code> . Defaults to NULL.
wrap_text	[logical(1)] Should the text in the cell be wrapped to fit in the column. Defaults to FALSE.
merge_cells	[logical(1)] Should the cells in the row spanning the table be merged? Defaults to FALSE.
heights	[integer] The row height for the formatted row. Defaults to NULL.
...	Other arguments to be passed to <code>openxlsx::createStyle</code> .

**Details**

The whole line will be styled.

`openxlsx::createStyle` is used for formatting the row. Use `text_decoration = "bold"` to format the row with bold font. Use `wrap_text = TRUE` and `merge_cells = TRUE` if you want a footnote to span all cells under the table and potentially span more lines. If the text spans more than one line, the height of the row can be adjusted by setting `heights = .`

**Value**

None. One row in the workbook object is styled.

**Author(s)**

Petter Hopp [Petter.Hopp@vetinst.no](mailto:Petter.Hopp@vetinst.no)

---

 theme\_NVI

*NVI theme for ggplot2*


---

**Description**

A complete ggplot2 theme in accord with NVI's graphical profile. The theme control all non-data display.



## Usage

```
theme_NVI(  
  base_size = 11,  
  base_family = "",  
  base_line_size = base_size/22,  
  base_rect_size = base_size/22  
)
```

## Arguments

base_size	[numeric(1)] Base font size, given in pts. Defaults to 11.
base_family	[character(1)] Base font family. Defaults to "".
base_line_size	[numeric(1)] Base size for line elements. Defaults to 'base_size' / 22.
base_rect_size	[numeric(1)] Base size for rectangular elements. Defaults to 'base_size' / 22.

## Details

The NVI theme without background for plot and legend, light grey horizontal support lines and dark grey axis' marks.

## Examples

```
library(ggplot2)  
library(NVIppretty)  
  
# Scatter plot  
ggplot(iris, aes(Sepal.Length, Sepal.Width)) +  
  geom_point(aes(color = Species)) +  
  theme(legend.position = "right") +  
  theme_NVI() +  
  scale_colour_manual(values = palette_graph)  
  
# Changing default theme for the whole session  
theme_set(theme_NVI())  
  
# The default them is now "theme_NVI" and there is no need to specify the theme within "ggplot".  
# Box plot in accord with NVI's graphical profile  
ggplot(iris, aes(Species, Sepal.Length)) +  
  geom_boxplot(aes(fill = Species)) +  
  theme(legend.position = "right") +  
  scale_fill_manual(values = palette_graph)
```

# Index

## \* datasets

- palette\_gradient\_green, [6](#)
- palette\_graph, [9](#)
- palette\_graph\_2020, [10](#)
- palette\_NVI, [10](#)
- palette\_NVI\_profile\_2020, [11](#)
- palette\_table\_blue\_2020, [12](#)
- palette\_table\_grey\_2020, [12](#)

add\_formatted\_worksheet, [2](#)

append\_text\_line, [5](#)

palette\_gradient\_air  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_darkblue  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_earth  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_food  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_graa  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_green, [6](#)

palette\_gradient\_grey  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_gul  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_jord  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_lightblue  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_luft  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_mat  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_red  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_vann  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_water  
    (palette\_gradient\_green), [6](#)

palette\_gradient\_yellow  
    (palette\_gradient\_green), [6](#)

palette\_graph, [9](#)

palette\_graph\_2020, [10](#)

palette\_NVI, [10](#)

palette\_NVI\_profile\_2020, [11](#)

palette\_table\_blue\_2020, [12](#)

palette\_table\_grey\_2020, [12](#)

style\_background\_per\_column, [13](#)

style\_text\_line, [15](#)

theme\_NVI, [16](#)