

REPRODUCIBLE ANALYSIS PDF REPORT

PLACEHOLDER COVER PICTURE

First page sub-title

Tur, sequis sunt abor rempos molupiet lab ipsa natque con niminve llesequos derro
restes serionet autasperit ut estrum, ut quia excero et vit voloreic torum quae erias
earchil explique pe officabo

Contents

Tables	4
Figures	5
Acronyms	6
Template	8
R Markdown	8
Preview paged HTML report	8
Print to PDF	9
YAML metadata	9
Typography	10
Inline text element syntax	10
Headings	10
Lead	11
Quotes	12
Abbreviations	12
Footnotes	12
Lists	13
Images, plots and tables	15
Images	15
Plots	17
Tables	18
Helper class	21
Text color	21
Background color	22
Columns	22
Floats	23
Page break	24

Tables

Figures

1 Total number of refugees - UNHCR Refugee Statistics	17
2 Total number of refugees and IDPs - UNHCR Refugee Statistics	18

Acronyms

UNHCR

The United Nations High Commissioner for Refugees

Template

This R Markdown document includes all the necessary features and styles to create an UNHCR branded report.

This specific template is using principally the Pagedown package, for all the information about the package see <https://pagedown.rbind.io/>. The template targets mainly the creation of paginated HTML reports that can be easily printed as PDF, but from one R Markdown document we can have many different outputs (see below).

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   : 2.00
## 1st Qu.:12.0    1st Qu.:26.00
## Median :15.0    Median :36.00
## Mean   :15.4    Mean   :42.98
## 3rd Qu.:19.0    3rd Qu.:56.00
## Max.   :25.0    Max.   :120.00
```

Preview paged HTML report

When you compile an R Markdown document to HTML in RStudio, RStudio will display the HTML page through a local web server, so paged.js will work in RStudio Viewer. However, when you view such pages in a real web browser, you will need a separate web server. The easiest way to preview these HTML pages in a web browser may be through the RStudio **servr** package, just type the following code in your R console and head to <http://127.0.0.1:4321/> in your browser to access the files.

```
# Load webserver to preview paged HTML
servr::httd()
```


Print to PDF

The easiest way to print to PDF is with Google Chrome or Chromium using the menu “Print” or by pressing Ctrl + P (or Command + P on macOS). Remember to allow background graphics to be printed.

There are also 2 other built-in ways:

1. using the function `pagedown::chrome_print()`. Its first argument (input) accepts a path to a local Rmd or HTML file or an URL. Google Chrome or Chromium must be installed on your system.
2. in RStudio, adding the following line in the YAML header of your Rmd file:

```
knit: pagedown::chrome_print
```

With this metadata parameter, the behavior of the “Knit” button of RStudio is modified: it produces both the HTML document and the PDF with Chrome. This functionality is suitable for any R Markdown HTML output format and is mainly convenient for small documents like presentations or notes. If `chrome_print()` cannot find Chrome or Chromium automatically, set the `PAGEDOWN_CHROME` environment variable to the path to Chrome or Chromium executable.

YAML metadata

A lot of components such as the table of contents, list of figures, etc. can be activated, deactivated or tweaked directly in the YAML at the top of the R Markdown document. Please refer at the guidance directly added there.

See <https://pagedown.rbind.io/#miscellaneous-features> for extra information on the YAML variables of the package.

Typography

Documentation and examples for typography, including global settings, headings, body text, lists, and more.

The template uses **Lato**, one of the brand recommended font paired with **Arial** as a fallback (`font-family: 'Lato', Arial, sans-serif;`). **Lato** has been selected as it can easily be downloaded and added to the report automatically.

For all the coding part we needed a monospace font, so we used **Roboto mono** as it is easily accessible too.

Inline text element syntax

Styling for common inline elements.

Example:

This line rendered as italicized text.

This line rendered as bold text.

This line rendered as italicized and bold text.

~~This line of text will be strikethrough.~~

You can use the `<u>` tag to underline text.

You can use the `<mark>` tag to **highlight text**.

This ^{^text or number^} rendered as ^{superscript}.

```
*This line rendered as italicized text.*
**This line rendered as bold text.**
***This line rendered as italicized and bold text.***
~~This line of text will be strikethrough.~~
# There is no R Markdown code for underline but HTML can be entered directly.
You can use the '<u>' tag to <u>underline text</u>.
# There is no R Markdown code for mark but HTML can be entered directly.
You can use the '<mark>' tag to <mark>highlight text</mark>.
This '^text or number^' rendered as ^superscript^.
```

Headings

All HTML headings, <h1> through <h6>, are available. To create a heading use # sign and include as many as the heading level (see example below).

If the table of content is active tags from <h1> to <h3> will appear in it with their respective hierarchy.

Example:

h1 Heading

h2 Heading

h3 Heading

H4 HEADING

h5 Heading

h6 Heading

```
# h1 Heading
## h2 Heading
### h3 Heading
#### h4 Heading
##### h5 Heading
##### h6 Heading
```

Lead

To create subheading type of paragraph with bigger font.

Example:

Example of a paragraph with the `.lead` class applied

```
# To apply a specific class to an element use the below R Markdown code
::: {.lead}
Example of a paragraph with the .lead class applied
:::
```

Quotes

In R Markdown, quotes in a certain part of the text are indicated using a blockquote element. The greater-than sign is used for this purpose (>). The structure showed in the below examples should be maintained for better results. If other format are needed please open an issue.

Example:

Complete quotes including source, open and closing quote symbols.

"It is further proof that refugee movements and the broader issue of migration of populations ... is a global challenge that cannot be confined to a few countries."
— Filippo Grandi

Simple quote with no source nor quote symbols.

The situation in South Sudan is far too fragile and dangerous for people to go back.

```
# Complete
> Quote text
>
> First and Last name

# Simple
> Quote text
```

Abbreviations

A list of abbreviations is automatically built if the document contains at least one HTML abbr element.

For instance, if the R Markdown document contains `<abbr title="United Nations High Commissioner for Refugees">UNHCR</abbr>`, a list of abbreviations is built with the CSS definition.

Example:

UNHCR

```
# There is no R Markdown code for abbreviations but HTML can be entered directly.
<abbr title="The United Nations High Commissioner for Refugees">UNHCR</abbr>
```

Footnotes

Footnotes are added using the `^[This is a footnote]` R Markdown notation. The number will be automatically generated and the text between brackets will be added at the bottom of the footnote page.

Example:

This is a footnote¹, it can include anything and can also be stylize UNHCR²

```
This is a footnote^[Text of the footnote]
UNHCR^[The **United Nations** High Commissioner for Refugees
<https://www.unhcr.org/>]
```

Lists

Unordered

To create an unsorted list in Markdown, you can use either the **plus sign**, the **hyphen** or an **asterisk**. All three ways produce the same result. To create sub-level list simply use **4 spaces indentation**.

Example:

- item 1
- item 2
 - sub-item 2.1
 - sub-item 2.2
- item 3
 - sub-item 3.1
 - sub-item 3.2

```
* item 1
* item 2
  + sub-item 2.1
  + sub-item 2.2
* item 3
  + sub-item 3.1
  + sub-item 3.2
```

Ordered

A sorted list is generated by a **number directly followed by a period**. To create sub-level list simply use **4 spaces indentation**.

¹Text of the footnote

²The **United Nations** High Commissioner for Refugees <https://www.unhcr.org/>

Example:

1. item 1
2. item 2
 1. item 2.1
 2. item 2.2
3. item 3
 1. item 3.1
 2. item 3.2

```
1. item 1
2. item 2
  1. item 2.1
  2. item 2.2
3. item 3
  1. item 3.1
  2. item 3.2
```

Mixed

A mixed list is generated by using both the unordered and ordered symbols. To create sub-level list simply use **4 spaces indentation**.

Example:

1. item 1
- item 2
 1. item 2.1
 2. item 2.2
1. item 3
 - item 3.1
 - item 3.2

```
1. item 1
* item 2
  1. item 2.1
  2. item 2.2
3. item 3
  * item 3.1
  * item 3.2
```

Images, plots and tables

This section will provide with all the guidance to insert images, charts and tables.

Images

By default the images will take the **whole width** of the page area (max-width = 100%, 18cm inside the margins). It is important to adapt and fix the picture ratio before inserting them. There are some helper **.class** that can be used to have some flexibility in the layout. If none of the **.class** cover your needs you can always add some `style` directly to the element.

To add an image just use `![image-name](image-link)`. A caption can be added on the line right after the image (no extra line break). For proper styling please include the caption between `*` symbol (sign to make text *italic*).

Example:

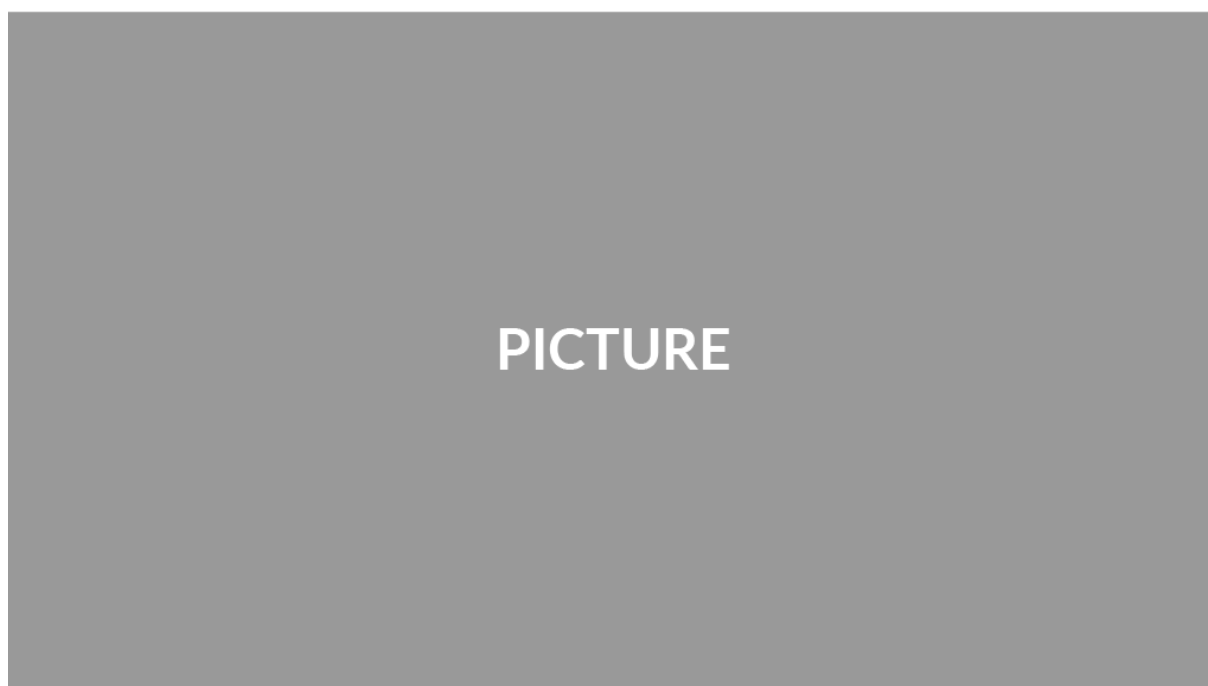


Image caption text. © UNHCR Photographer name

```
# Important NOT to include an extra line break after the image
# to make sure the caption is stylized correctly
![Placeholder](img/169_pics_landscape.png)
*Image caption text. &copy; UNHCR Photographer name*
```

Helper classes and images

Whole width images might not always be the best option for your layout, especially with *portrait orientation* pictures. Use the `.img50` (max-width: 50%) or `.img75` (max-width: 75%) classes around the image and caption to reduce their sizes.

Example:

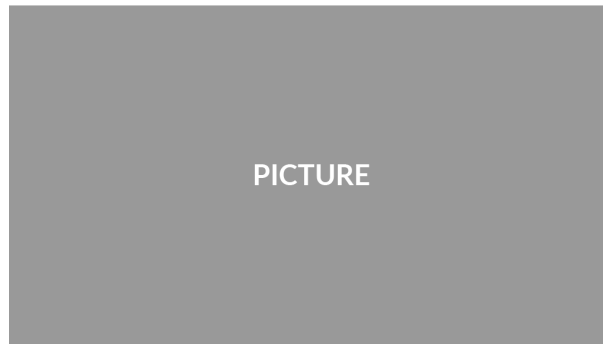


Image caption text. © UNHCR Photographer name

```

::: {.img50}
! [Placeholder](img/169_pics_landscape.png)
*Image caption text. &copy; UNHCR Photographer name*
:::

```

Combine it with the `.float-start` or `.float-end` classes to align the image on a specific side of the page and even have text around it. **Be careful with the use of floats** as the image and caption aren't part of the text flow anymore so all the content that follows will be placed next to it and not anymore below. Wrap both the image (+ caption) and the content you want to be next to it with the class `.clearfix` to cancel this. The content after the `<div class="clearfix">` will be back within the normal flow.

Example:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum finibus dui eu mi tincidunt, blandit posuere tellus aliquet. Vivamus eget neque felis. Nam vitae est quis lacus aliquet maximus ut id magna. Nulla quis massa felis. Pellentesque sit amet varius felis, non lobortis orci. Aliquam sit amet arcu suscipit, porttitor sapien eget, ultricies ipsum.

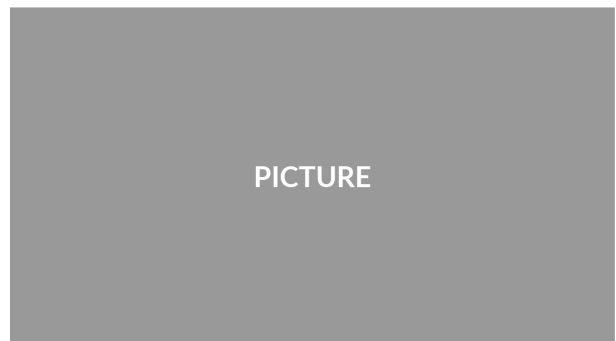


Image caption text. © UNHCR Photographer name


```

:::{.clearfix}
:::{.img50 .float-end}
![Placeholder](img/169_pics_landscape.png)
*Image caption text. &copy; UNHCR Photographer name*
:::
Lorem ipsum dolor sit amet, text that will be placed next to the image
:::

```

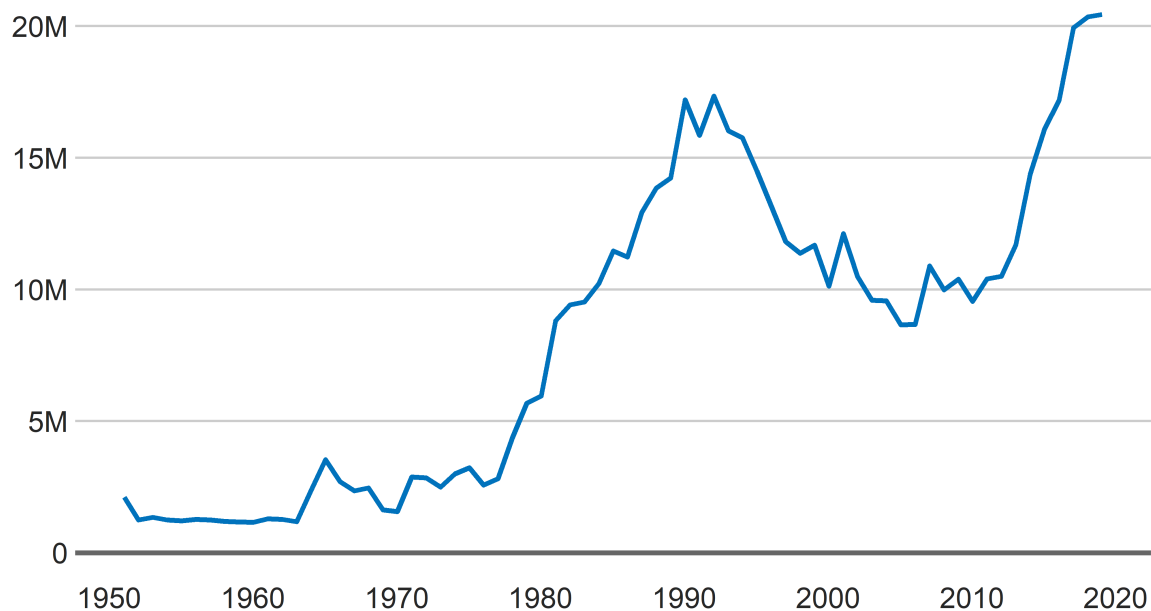
Plots

You can also embed plots created directly in R. Please review the vignette on the `unhcr_theme` for `ggplot2` for more details on how to create chart following UNHCR brand.

If you have the List of figure activated you will need to include `fig.cap` in code chunk parameter. The text will be used for the caption itself as well as the name in the List of figure.

Total number of refugees

Evolution of refugee population 1951-2020



Source: UNHCR Refugee Statistics

Figure 1: Total number of refugees - UNHCR Refugee Statistics

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot and only show the plot itself in the report.

Unlike the images the size of the plots will be defined directly in the code chunk using the `out.width` parameter. We recommend that you use relative measure like `out.width="75%"`. You can also set the alignment of the figure in the code chunk parameter using the `fig.align` tag, where you set it to 'left', 'center' or 'right'.

Total number of refugees and IDPs

Evolution of refugee and IDP population from 1951 to 2020

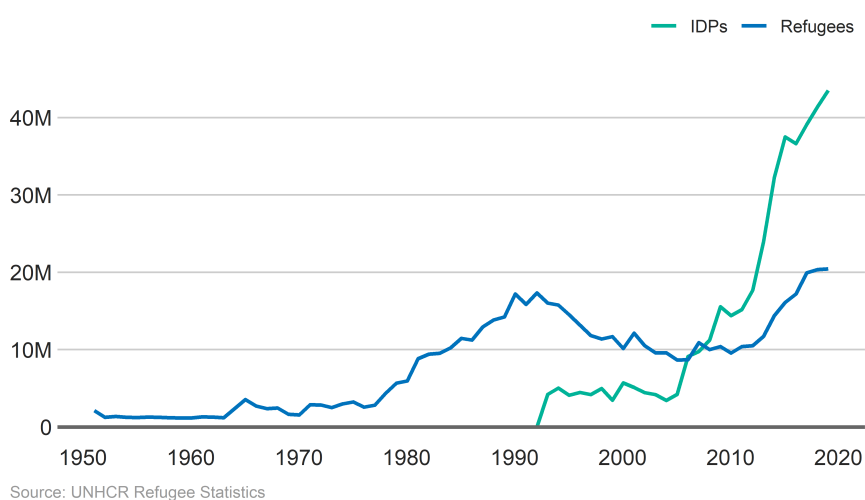


Figure 2: Total number of refugees and IDPs - UNHCR Refugee Statistics

Tables

You can also embed tables. The easiest way is to use the `kable` function from the `knitr` package, that is included in the code chunks of your Rmd file. To include the UNHCR styles a class `.table` needs to be added to the `kable` declaration.

For example see the below table added simply with the `knitr::kable(data_frame_name, table.attr = "class=\"table\"", format = "html")` function, including some basic UNHCR styles. Be careful of the escape character to make sure it includes the extra `""` as well as including the `format="html"` parameter.

Example:

Year	REF	ASY	IDP	VDA	STA	OOC
2015	16110280	3223460	37494172	0	2687759	870688
2016	17184291	2729521	36627127	0	3242206	803084
2017	19940568	3089503	39118516	0	3853982	1596125
2018	20359556	3501627	41425168	2592947	3851981	1182756
2019	20444819	4148194	43503362	3582202	3561980	5740622

There are 2 ways to edit the table: using some parameters of the `kable` function or by adding CSS classes to the `table.attr` parameter.

The `kable` function is mainly to affect the content of the table itself, like formatting numbers, changing column names or add figure caption. You can find some information on how to use it on the bookdown page³.

Example:

Years	Refugees	Asylum-seekers	IDPs	Venezuelans displaced abroad	Stateless persons	Others of concern
2015	16,110,280	3,223,460	37,494,172	0	2,687,759	870,688
2016	17,184,291	2,729,521	36,627,127	0	3,242,206	803,084
2017	19,940,568	3,089,503	39,118,516	0	3,853,982	1,596,125
2018	20,359,556	3,501,627	41,425,168	2,592,947	3,851,981	1,182,756
2019	20,444,819	4,148,194	43,503,362	3,582,202	3,561,980	5,740,622

Table 1: Population of concern to UNHCR for the last 5 years

```
knitr::kable(ref_idp_table,
  table.attr = "class=\"table\"",
  format = "html",
  col.names = c('Years', 'Refugees', 'Asylum-seekers', 'IDPs',
    'Venezuelans displaced abroad', 'Stateless persons',
    'Others of concern'),
  format.args = list(big.mark = ",", scientific = FALSE),
  caption = "Population of concern to UNHCR for the last 5 years")
```

Adding CSS classes to the `table.attr` parameter will only affect the style of the table.

Here is the list of the available classes:

- `.table`: Apply base UNHCR style. **Mandatory!!!**
- `.table-striped`: Apply light-grey background even odd rows.
- `.table-striped-blue`: Apply light-blue background even odd rows.
- `.table-width`: To remove the default width of 100% and apply the width required by the data inside the table.
- `.table-bordered`: To apply borders all around the table and cells.
- `.table-borderless`: To remove all borders.

³<https://bookdown.org/yihui/rmarkdown-cookbook/kable.html#customize-html-tables>

Example:

Year	REF	ASY	IDP	VDA	STA	OOC
2015	16,110,280	3,223,460	37,494,172	0	2,687,759	870,688
2016	17,184,291	2,729,521	36,627,127	0	3,242,206	803,084
2017	19,940,568	3,089,503	39,118,516	0	3,853,982	1,596,125
2018	20,359,556	3,501,627	41,425,168	2,592,947	3,851,981	1,182,756
2019	20,444,819	4,148,194	43,503,362	3,582,202	3,561,980	5,740,622

Table 2: Population of concern to UNHCR for the last 5 years

```
knitr::kable(ref_idp_table,
  table.attr = "class=\"table table-striped-blue table-width\"",
  format = "html",
  format.args = list(big.mark = ",", scientific = FALSE),
  caption = "Population of concern to UNHCR for the last 5 years")
```

Helper class

A serie of CSS `.class` to give extra flexibility to the report layout, as well as simplifying recurrent styling need.

A class can be easily applied to a block of content, using the fenced divs approach⁴. For more granular need you can also apply a class directly to inline content with bracketed spans⁵.

Example:

A ut perore, ut lam simodi sinvel magnis doluptatem que est.

A ut perore, ut lam simodi sinvel **magnis doluptatem** que est.

```
# Fenced divs
:::{.unhcr-blue}
A ut perore, ut lam simodi sinvel magnis doluptatem que est.
:::

# Bracketed spans
A ut perore, ut lam simodi sinvel [magnis doluptatem]{.unhcr-blue} que est.
```

Text color

Simply add the `.unhcr-blue` or `.unhcr-grey` classes to apply UNHCR branded colors to text element. As mentioned before this can be applied both to fenced divs and inline elements.

Example:

This will be colored **blue** and this will be grey. It can even be mixed with other inline styling like **bold**.

```
This will be colored [blue]{.unhcr-blue} and this will be [grey]{.unhcr-grey}.
It can even be mixed with other inline styling like [bold]{.unhcr-blue}.
```

⁴<https://pandoc.org/MANUAL.html#extension-fenced-divs>

⁵<https://pandoc.org/MANUAL.html#extension-bracketed-spans>

Background color

Easily create call out boxes by adding the background helper classes. They come in light-blue (.bg-blue) and light-grey (.bg-grey). Both come with padding to allow spacing around the elements inside. These classes need to be applied **only to fenced divs element**.

Example:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin ultrices mauris eget ligula imperdiet tincidunt. Proin eget odio laoreet, sagittis metus a, rutrum risus. Duis in ante sed enim pharetra pulvinar. Phasellus sit amet vestibulum lorem. Aliquam tincidunt mauris sit amet pharetra viverra.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin ultrices mauris eget ligula imperdiet tincidunt. Proin eget odio laoreet, sagittis metus a, rutrum risus. Duis in ante sed enim pharetra pulvinar. Phasellus sit amet vestibulum lorem. Aliquam tincidunt mauris sit amet pharetra viverra.

```
# Blue example
:::{.bg-blue}
Lorem ipsum dolor sit amet,...
:::

# Grey example
:::{.bg-grey}
Lorem ipsum dolor sit amet,...
:::
```

Columns

The `.col2` class creates a 2 columns layout for the content inside the section comprised between the opening `:::{.col2}` and closing `:::` tags. The use is recommended **only for text**. For more complex layout including images or plots please refer to the [Floats \(page 23\)](#) section for documentation or to the [Helper classes and images \(page 16\)](#) part for examples.

Example:

<p>A ut perore, ut lam simodi sinvel magnis doluptatem que est, tem quosantis is cone ea dolessus anditas pereper sperum que sim assundebis eum fuga. Et</p>	<p>eate litia seruntotas ideni untorro mollatis recest anisqui ilicab ipsanitio.</p>
--	--

```
...{.col2}
```

A ut perore, ut lam simodi sinvel magnis doluptatem que est, tem quosantis is cone ea dolessus anditas pereper sperum que sim assundebis eum fuga. Et eate litia seruntotas ideni untorro mollatis recest anisqui ilicab ipsanitia.

```
...
```

Floats

These utility classes float an element to the left or right using the CSS float property. Important is included to avoid specificity issues.

We highly encourage you to use the `.clearfix` class to quickly and easily clear floated content within a container by adding a clearfix utility. Apply `.clearfix` to the **parent element** this will ensure the content flow is back to normal after the floating elements, see [Helper classes and images \(page 16\)](#) for examples.

Example:

This will float start (left of the page).

This will float end (right of the page).

```
# Clearfix is needed to make sure it recreates the normal content flow
```

```
...{.clearfix}
```

```
...{.float-start}
```

This will float **start** (left of the page).

```
...
```

```
...
```

```
...{.clearfix}
```

```
...{.float-end}
```

This will float **end** (right of the page).

```
...
```

```
...
```

Page break

The template automatically creates a page break when your content doesn't fit into a page. However, you may need to control this fragmentation to avoid solitary element. Simply create a fenced div with the class `.page-break` and it will push all the following content to the next page. Avoid using the `.page-break` class alongside floating element.

Example:

```
# Create an empty div that will push all following content to the next page.  
:::{.page-break}  
:::
```


REPRODUCIBLE ANALYSIS PDF REPORT



For more information please contact:

Contact1 Name

Contact1 Title

email1@unhcr.org

Contact2 Name

Contact2 Title

email2@unhcr.org