**Documentation for R-markdown report**

When reading through this guide it is important to remember that it is not a final solution to constructing automatic reports. The output should always be kept in mind and each step should be evaluated before implemented in the report as each report is different.

**Installations**

1. Make sure to open the project you can do this by double clicking the “\*.Rproj” file.

2. Ensure that Rstudio and Rmarkdown look for files in the same way. We do this by setting the following option in Rstudio: *Tools -> Global Options -> Rmarkdown -> Evauluate chunks in directory: "Project"*

3. Try to create the pdf document by pressing Knit

4. It will most likely error the first time, we need to install a LateX engine to create the pdf-documents. The easiest way to do this is to run the code: tinytex::install\_tinytex() in the console. This will take some time, so be patient.

5. If the error persist try downloading Miktex here: [Getting MiKTeX](https://miktex.org/download) (Just a different way for R to read LateX)

**Structuring the report – what did we do?**

* Tidy data in file “0. Tidy\_data.r”
* Calculations, visualization and text in “01. Template.Rmd”
* Automatic generation of reports in “02. Generate reports.R”
* Layout: PrettyPDF.tex

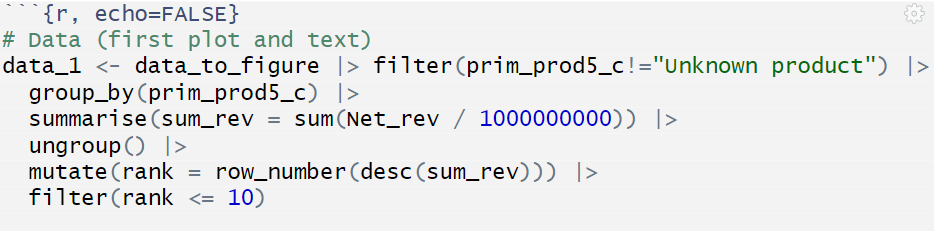
1. **Create tidy data**
   1. **Tidy data is ensuring that data is cleaned and fulfills that:**
      1. Each variable is a column
      2. Each observation is a row
      3. Each type of observational unit is a table

In this template tidy data fulfills i), ii) and iii). Furthermore data was also manipulated so that:

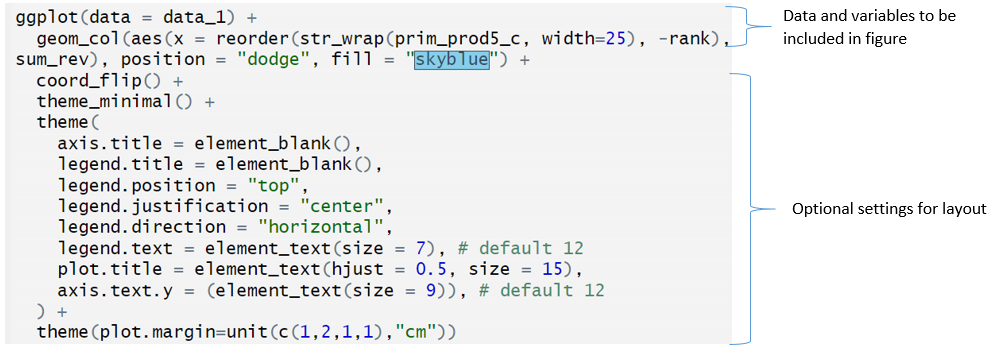
1. Missing values was replace by 0 (This makes calculations easier)
2. Converting variables to text

Examples functions to manipulate your data:

* + 1. mutate → adding new columns
    2. select → selecting columns
    3. filter →filtering rows
    4. summarise → reducing values into summaries (for example average of variable)
    5. arrange → reordering rows
    6. case\_when/ifelse → make new conditional variable
    7. Use the pivot\_longer() to make long data[[1]](#footnote-1)

1. **Create you report template**The easiest (and most intuitive) way to make the report is by including each element in chronological order.
   1. **Do calculations as needed**
      1. For instance using group\_by and mutate, filter etc.  
         

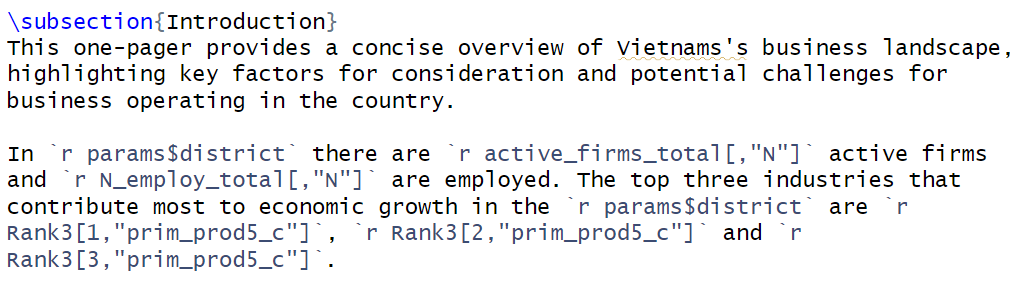
In this example, data is filtered, grouped by primary production code and new variables are defined.

* 1. **Visualize  
     i) we prefer the ggplot-package**Link to documentation [here](https://ggplot2.tidyverse.org/)  
     Link to ggplot-inspiration [here](https://r-graph-gallery.com/ggplot2-package.html)  
     
  2. **Add text** 
     1. **Chapters and sections**

Easy to make. We use LateX-code for formatting these.   
Examples:

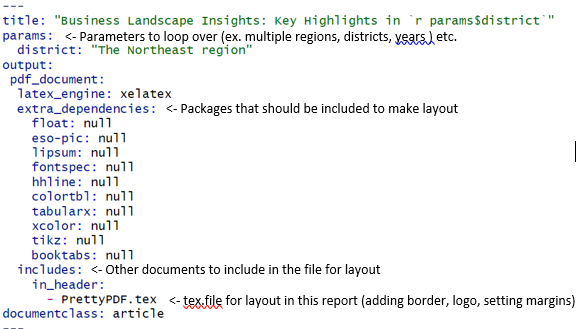
* + - 1. \section{text}
      2. \subsection{text}
      3. \subsubsection{text}
      4. \paragraph{text}

Here a headline “Introduction” is included and text written as ‘normal’.



Calculations can be included in text writing `r data[row,column]`

1. **Layout**
   1. **Settings in YAML**



Be aware that YAML is sensitive to indentation.

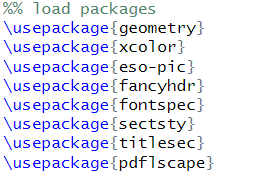
**R-markdown options for including figures and code**

include = FALSE prevents code and results from appearing in the finished file. R Markdown still runs the code in the chunk, and the results can be used by other chunks.  
echo = FALSE prevents code, but not the results from appearing in the finished file. This is a useful way to embed figures.  
results=’asis’ output is printed “as-is,”. Alternative results=’hide’.

Options as fig.height=, fig.width=, fig.cap= can also be included to format figures.

* 1. **Layout using \*.tex-files**

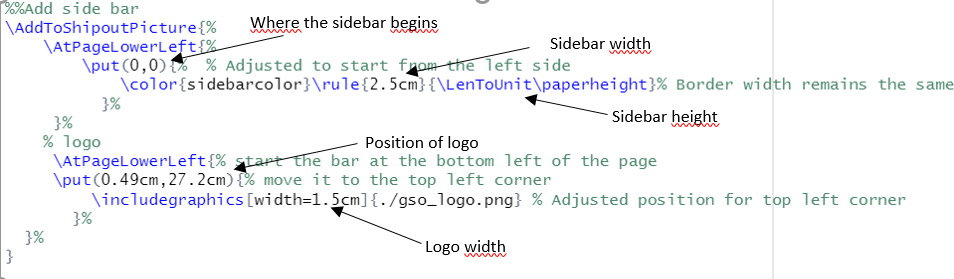
Load packages using latex-function “\usepackage”



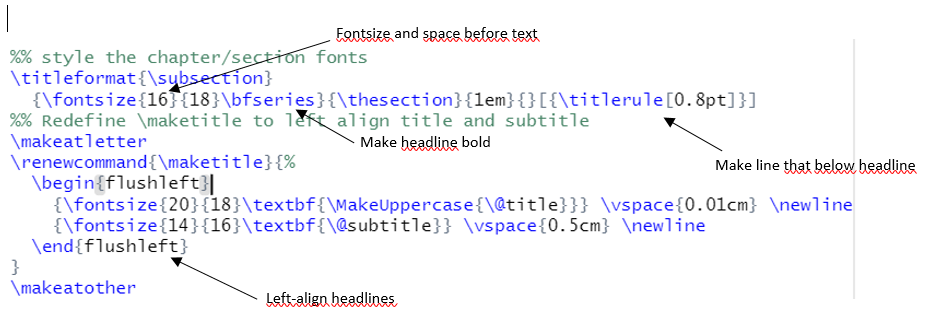
How to adjust fonttype

How to adjust page-size (margins)

One suggesting for a nice layout: Adding a colored sidebar. Just delete this part if you want a different layout



How to adjust headlines



Edit pagenumber  


1. **Generating multiple reports**
   1. This is done using the “02. Generate report”-file. The file iterates through districts and filters data accordingly, ensuring that one report per district is printed.
2. **If the reports cannot print…**

Run through each step manually to locate the error. Most often R gives a good error-message to locate the error. Remember the difference between knitting the document and running the code line by line. A knit is always done in a clean environment. If an error happens when you knit but not when you run the code try to clear your environment and run from the beginning.

Errors in loops can be tricky because the error might only happen in a single iteration of the loop. If you think an error is happening in the loop try to iterate through it manually and run it line by line.

When error is located, check the input data and ask:  
“is it as you expected it to be?” – Most often the error comes from the data not being as we expected.

Comments are your best friend, you might remember what a code section does tomorrow, but maybe not in a week or a year, so adding even a short explanation of each code piece will always be helpful.

1. This is the format often needed to plot and visualize data. [↑](#footnote-ref-1)