runidue package

It is a package in latex used for creating lecture slides. The package contains all codes necessary to knit slides. It has predefined environments for writing lemma, corollary, math formulas, etc. By specifying on the respective YAML header, at the start of the slides, some basic aspects such as font size, aspect ratio, etc, of the slides can be changed.

This readme documentation gives a detailed description, using code snippets and screenshots, on what can be achieved using the unidue package.

Installing and loading the package

The package is contained in the github and can be installed in the local computer. To do so first devtools library needs to be installed and loaded in the local computer. The follwing code checks if the devtools package is installed. If it's not installed, it installs it; otherwise, it prints a message saying that it's already present:

```
# Check if devtools package is installed
if (!requireNamespace("devtools", quietly = TRUE)) {
  # If not installed, install devtools
  install.packages("devtools")
  # Load devtools
  library(devtools)
  print("devtools package has been installed and loaded.")
} else {
  # If already installed, print a message
  print("devtools package is already installed.")
}
# Load devtools
library(devtools)
```

Once devtools library is installed, the package *unidue* can be installed from github using the code below:

```
install_github("https://github.com/jens-klenke/runidue.git")
```

Now it can be loaded using:

```
library(runidue)
```

Now the package is loaded and is ready to be used.

YAML header

A YAML header, also known as YAML front matter, is a section at the beginning of a file written in YAML (YAML Ain't Markup Language) format. It's commonly used in Markdown files (.md) and other types of files to provide metadata or configuration information about the content of the file.

In Markdown files, the YAML header is typically enclosed between triple dashes `---` at the beginning and end. Within the YAML header, you can specify key-value pairs separated by colons `:`. These key-value pairs provide information such as title, author, date, tags, etc., depending on the purpose of the file.

A typical YAML header which uses this package looks like:

```
---
title: "Title"
author: "Author"
date: "Term X"
output: runidue::lectureslides
---
```

Several modifications to the lecture slides, can be done using the YAML header. To get more information about the specifications, the follwoing code can be used:

```
?runidue::lectureslides
```

An example of a more elaborate YAML header is as follows:

```
---
title: "Title of Your Document"
author: "Author Name"
date: "February 27, 2024"
output:
  runidue::lectureslides:
    lang: "en"
    theme: "default"
    highlight: "zenburn"
    slide_level: 2
    incremental: true
    colorlinks: true
natbib: true
natbiboptions:
    - "round"
    - "semicolon"
biblio-style: "plainnat"
csl-refs: true
urlcolor: "red"
citecolor: "green"
mainfont: "Times New Roman"
papersize: "a4paper"
fontenc: "T1"
linestretch: 1.2
fontfamily: "times"
toc: true
toc_depth: 2
fontsize: 8pt
classoption: aspectratio=169
---
```

This leads to the following output:

Output

Environments

Several environments which are predefined in the package, can be called during knitting the files. These environments are used for highlighting Lemma, corollary, proofs, examples, math functions, etc. A comprehensive lists of all such environments which the package offers is as follow:

Command	Use
`thmn`	Theorem
`lem`	Lemma
`cor`	Corollary
`defn`	Definition
`ass`	Annahme or assumption
`rem`	Anmerkung or Remark
`prop`	Proposition
`xmpl`	Beispiel
`que`	Frage or Question
`exe`	Aufgabe or Exercise
`maththm`	Mathematical Theorem

An example of implementation of an environment is as follows:

```
\xmpl[Heading of the example]
The example itself..
\endxmpl
```