
Базовая информация

Основная информация в YAML пустого `.Rmd` комплект документов для `.pdf` вывод включает в себя следующее

```
title: "A catchy title"
author: "First Last"
date: "19 February 2021"
output: pdf_document
```

Вы также можете добавить субтитры с помощью `subtitle` аргумент и установите автоматическое обновление данных с помощью встроенного кода **R** 2021-02-21. Вы также можете создать собственный формат даты. Например, следующее `date` опция будет напечатана как `day full_month_name four_digit_year`. Ознакомьтесь с вариантами `format()` чтобы узнать, как изменить формат даты.

```
title: "A catchy title"
subtitle: "A pithy subtitle"
author: "First Last"
date: "`r format(Sys.time(), '%d %B %Y')`"
output: pdf_document
```

Макет документа

Макет документа по умолчанию для `.pdf` файлы в **R Markdown** довольно хороши, но есть несколько вариантов его улучшения. Эти параметры указаны в первую очередь в документе `YAML`, но некоторые из них будут помещены в «параметры фрагментов» внутри **R**. блоков кода

Появление

Мы можем изменить общий вид или *класс* документа с помощью `documentclass` аргумент, который вы установили в `YAML`. Вот некоторые из вариантов.

Статья

The `articleclass` — это базовый макет с относительно большими полями.

```
title: "Example template for exploring formatting options"
author: "First Last"
```

```
date: "2/19/2021"
output: pdf_document
documentclass: article
```

Отчет

Класс `report` создаст отдельную титульную страницу и вставит разрывы страниц между заголовками разделов первого уровня. Он также имеет относительно большую маржу.

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: report
```

Две колонки

Вы можете создать документ с двумя столбцами текста с помощью `classoption` аргумент. Например, вот статья в две колонки.

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: report
classoption: twocolumn
```

Поля

Вы можете установить одинаковое расстояние между полями для всех четырех сторон или установить каждую из них отдельно с помощью кнопки `geometry` аргумент. Вот пример установки всех полей в `article` до 1,5 дюймов. (Обратите внимание, что для полей также можно использовать метрические единицы.)

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: article
geometry: margin=1.5in
```

Вот пример установки верхнего и нижнего полей в `article` до 1 дюйма, а левое и правое поля — до 1,25 дюйма. Обратите внимание: при использовании нескольких настроек в аргументах нам нужно установить их в отдельных строках, которым предшествует тире. —.

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: article
geometry:
- top=1in
- bottom=1in
- left=1.25in
- right=1.25in
```

Межстрочный интервал

Вы можете увеличить межстрочный интервал с помощью `linestretch` аргумент. Например, вот `article` полями шириной 1 дюйм и двойным межстрочным интервалом.

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: article
geometry: margin=1in
linestretch: 2
```

Номера строк

Вы можете добавить номера строк в `.pdf` документы, но это предполагает использование дополнительного **LaTeX** вызова пакета `lineno`. Чтобы использовать дополнительные пакеты **LaTeX**, вам необходимо использовать команду `header-includes` аргумент в `YAML`. В частности, это выглядит так для двойного интервала `article` полями в 1 дюйм:

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: article
geometry: margin=1in
linestretch: 2
header-includes:
- \usepackage{lineno}
- \linenumbers
```

Шрифты

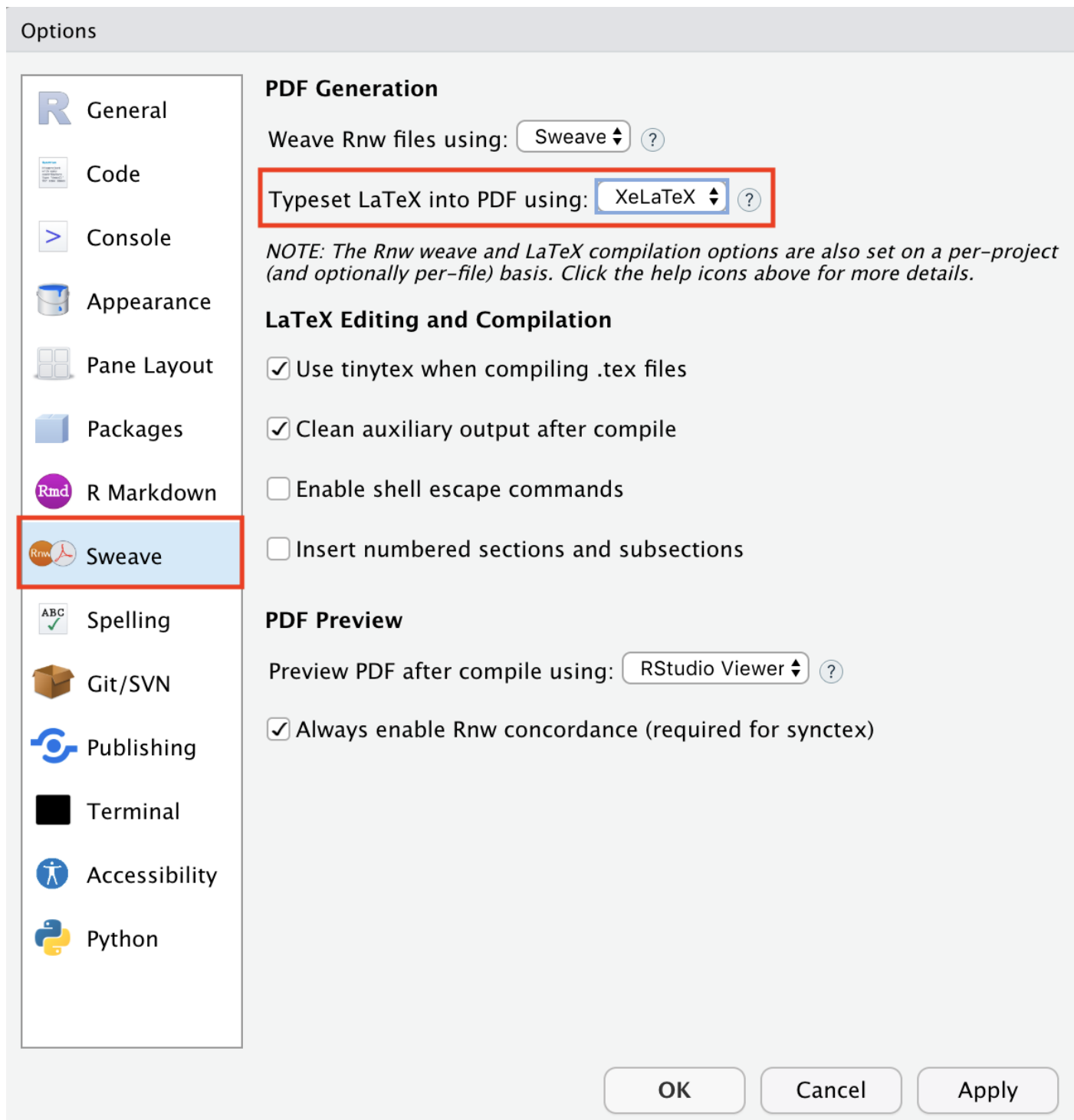
Размер

Вы можете установить размер шрифта в документе с помощью `fontsize` аргумент. Например, вот двойной интервал `article` с полями шириной 1 дюйм и шрифтом размером 12 пунктов.

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: article
geometry: margin=1in
linestretch: 2
fontsize: 12pt
```

Семья

Вы можете изменить семейство шрифтов, но это зависит от **LaTeX** используемого вами механизма. Вы можете проверить это, изучив **глобальные параметры в RStudio**. Для этого в главном меню выберите пункт `Tools > Global Options...` В диалоговом окне выберите пункт `Sweave` вариант слева и посмотрите на `Typeset LaTeX into PDF using:` параметр.



Если вы используете `pdflatex`, вы можете использовать `fontfamily` аргумент для выбора пакета шрифтов **LaTeX** для загрузки в документ для изменения шрифта. Также обратите внимание, что для этого необходимо использовать спецификацию многострочной опции для `output` как мы видели для `geometry` выше.

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output:
  pdf_document:
    latex_engine: pdflatex
```

```
fontfamily: Arial
documentclass: article
geometry: margin=1in
linestretch: 2
fontsize: 12pt
```

Тогда в документе будет использоваться шрифт Arial. Если ваш **LaTeX** дистрибутив — **TinyTeX** и необходимые пакеты шрифтов не установлены, они должны быть установлены автоматически при компиляции документа.

Если вы используете **LaTeX** движок **xelatex** или **lualatex**, вы сможете выбирать шрифты, доступные на вашем локальном компьютере, и вам не придется устанавливать дополнительные пакеты LaTeX. Для этого используйте `mainfont` аргумент такой:

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output:
  pdf_document:
    latex_engine: xelatex
mainfont: Arial
documentclass: article
geometry: margin=1in
linestretch: 2
fontsize: 12pt
```

Передняя часть

Вы можете добавить к документу так называемую *вступительную часть*, например оглавление.

Оглавление

Чтобы автоматически создать оглавление, установите `toc` аргумент для `true`. Вы можете контролировать, сколько уровней заголовков вы хотите отображать в оглавлении, с помощью `toc-depth` аргумент. Например, вот двойной интервал `report` с полями шириной 1 дюйм, шрифтом размером 12 пунктов и оглавлением, показывающим первые два уровня заголовков.

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: report
geometry: margin=1in
linestretch: 2
```

```
fontsize: 12pt
toc: TRUE
toc-depth: 2
```

Версия документа

Для некоторых документов я предпочитаю устанавливать собственный *номер версии* на основе даты, а не устанавливать `date` аргумент в явном виде. Для этого просто добавьте в *основной документ* строку , подобную этой:

```
This is version `r paste0('0.',format(Sys.time(), '%y.%m.%d'))`.
```

который будет рендериться

«Это версия 0.21.02.21».

Размещение фигуры

В отличие от Microsoft Word, в котором рисунки размещаются непосредственно там, где указывает пользователь, **LaTeX** попытается разместить рисунок в положении, не нарушающем определенные типографские правила. Таким образом, рисунки могут «уплывать» из того места, где они упоминаются в тексте. Эти так называемые *поплавки* используются в качестве контейнеров для рисунков (и таблиц), которые нельзя разбить на страницы.

В *R Markdown **фигуры фактически генерируются с помощью среды фигур в LaTeX****. Например, Pandoc преобразует следующий код Markdown для указания изображения.

```
![This is an image.](images/my_picture.jpg)
```

к следующему коду **LaTeX** :

```
\begin{figure}
  \includegraphics{images/cool.jpg}
  \caption{This is an image.}
\end{figure}
```

Если рисунок или таблица не могут поместиться в свободное место на текущей странице, **LaTeX** попытается разместить их вверху следующей страницы. Если рисунок достаточно высокий, он может занять всю следующую страницу, даже если еще останется место для нескольких строк текста. Поведением поплавок можно управлять, используя элементы управления размещением в квадратных скобках, например `\begin{figure}[]`. Вот некоторые возможные элементы управления:

- `h`: поместите поплавок сюда (или рядом с ним).
- `t`: Позиция вверху страницы.

- b: Позиция внизу страницы.
- p: разместить на специальной странице только для плавающих объектов.

Вы можете переопределить размещение по умолчанию, поставив перед элементом управления восклицательный знак (например, `[!t]`).

Если вы хотите разместить число с плавающей запятой *именно в том месте* кода **LaTeX**, вам нужно будет включить вызов метода `float` пакет в YAMLa затем установите `fig.pos = " !H"` опция `chunk` в блоке кода. Например,

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: article
geometry: margin=1in
linestretch: 2
fontsize: 12pt
toc: TRUE
toc-depth: 2
header-includes:
  - \usepackage{float}
```

Подфигуры

Иногда вам может потребоваться включить несколько изображений в одну среду фигур.

Подфигуры позволяют нам добиться этого, размещая несколько изображений в одной среде и снабжая каждое собственным подзаголовком. Для дополнительных рисунков требуется **LaTeX**. пакет `subfig`, который вы можете вызвать в YAMLa, такой как

```
title: "Example template for exploring formatting options"
author: "First Last"
date: "2/19/2021"
output: pdf_document
documentclass: article
geometry: margin=1in
linestretch: 2
fontsize: 12pt
toc: TRUE
toc-depth: 2
header-includes:
  - \usepackage{subfig}
```


Чтобы упорядочить все графики из фрагмента кода в подфигуры, вам необходимо установить следующие параметры фрагмента.

- `fig.cap`: подпись для всего окружения фигуры
- `fig.subcap`: `a character` вектор подписей для каждого из подрисунков

Кроме того, вы, вероятно, захотите установить следующие параметры

- `fig.ncol`: количество столбцов с подфигурами (по умолчанию все графики расположены в одну строку)
- `out.width`: выходная ширина отдельных графиков, которая обычно равна 100 %, разделенным на количество столбцов.

Например, этот фрагмент кода создаст 3 подграфика:

```
```{r subplot, fig.cap='A figure with 3 subplots.', fig.subcap=c('The
numbers 1 through 10.', 'Speed versus distance.', 'Time series of white
noise.'), fig.ncol = 2, out.width = "50%", fig.align = "center", echo =
FALSE}
plot(1:10, pch = 16)
plot(cars)
plot.ts(rnorm(30))
```
```