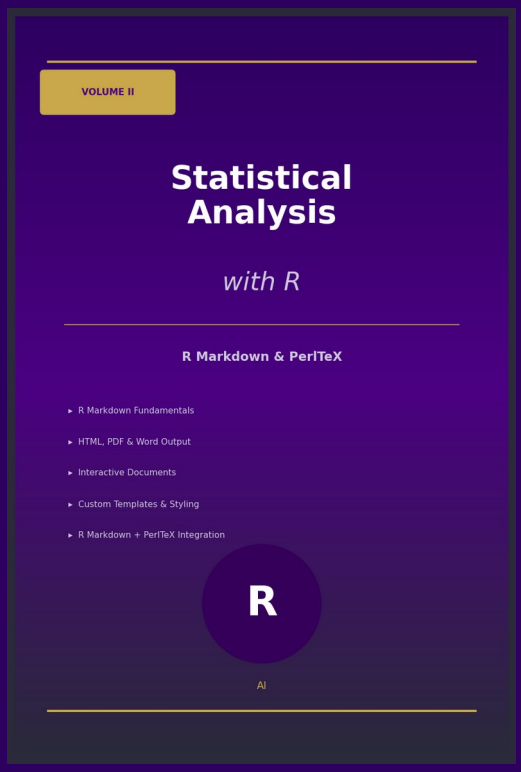
noteboxstylebackgroundcolor=brandbg, linecolor=indigo, linewidth=2pt, topline=false, bottomline=false, rightline=false, innerleftmargin=10pt, innerrightmargin=8pt, innertopmargin=6pt, innerbottommargin=6pt infoboxstylebackgroundcolor=indigo!10, linecolor=indigo, linewidth=1.2pt, roundcorner=4pt, innerleftmargin=10pt, innerrightmargin=8pt, innertopmargin=6pt, innerbottommargin=6pt warningboxstylebackgroundcolor=brandgold!12, linecolor=brandgold, linewidth=1.8pt, topline=false, bottomline=false, rightline=false, innerleftmargin=10pt, innerrightmargin=8pt, innertopmargin=6pt, innerbottommargin=6pt



Statistical Analysis with R

Volume II — R Markdown & PerlTeX

AI

March 13, 2026

Reproducible documents, dynamic reports, and advanced typesetting with PerlTeX.

Statistical Analysis with R — Volume II
R Markdown & PerlTeX

Author: AI (with guidance from publisher)

Typeset with LaTeX using the `pdflatex` engine.

Preface

Volume II of *Statistical Analysis with R* covers the R Markdown ecosystem in depth, from the simplest HTML notebooks to complex parameterised reports, interactive dashboards, and the advanced integration of PerlTeX macros within R Markdown documents compiled to PDF.

The companion Volume I covers the statistical methods and the R toolchain (SQL,`dplyr`, `tidyr`, `testthat`, Git, Podman) that underpin the analyses whose results are communicated via the documents described here.

The reader should be familiar with the R language at the level described in Volume I. No prior knowledge of LaTeX, Perl, or Markdown is assumed, though the chapters on PerlTeX will be most useful to those who have already encountered basic LaTeX.

AI

Contents

List of Figures

List of Tables

Part I

R Markdown Fundamentals

CHAPTER 1

Introduction to R Markdown

1.1 What Is R Markdown?

R Markdown is a plain-text file format (`.Rmd`) that weaves together narrative text written in Markdown, executable R code blocks, and optional LaTeX mathematics. When *rendered*, the file is processed by `knitr` (which executes the R code and inserts the results) and then by `pandoc` (which converts the resulting Markdown to the target output format).

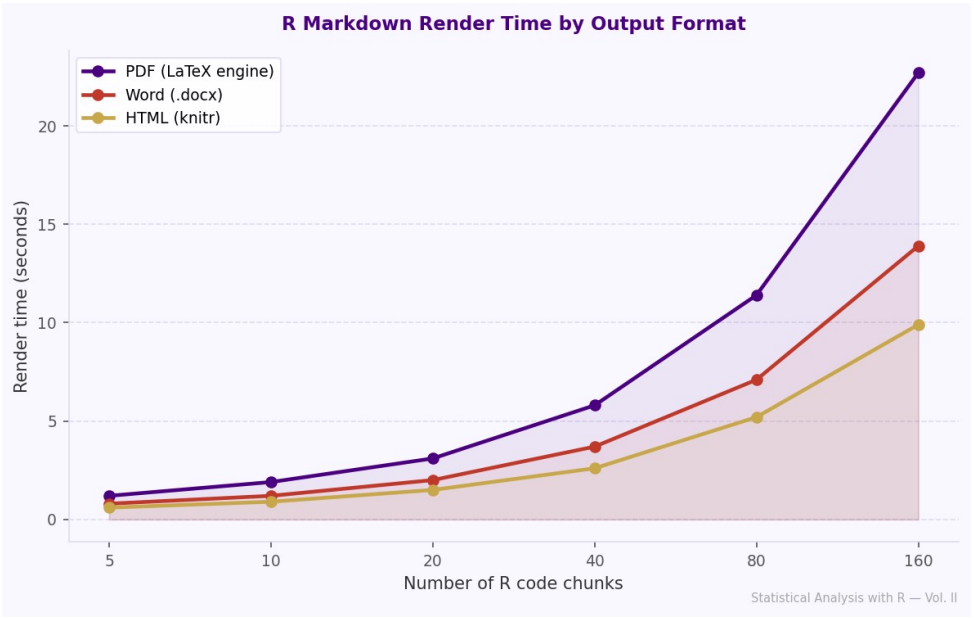


Figure 1.1. R Markdown render time by output format and number of code chunks.

1.2 Installing the Required Packages

```r
install.packages(c("rmarkdown","knitr","tinytex"),
                 repos = "https://cran.r-project.org")

# Install a minimal LaTeX distribution for PDF output
tinytex::install_tinytex()
```

Listing 1.1. Installing rmarkdown and knitr

Alternatively, for PDF output you may install the full TeX Live distribution via Fedora's repositories:

```
sudo dnf install texlive-scheme-full
```

Listing 1.2. Installing TeX Live on Fedora

1.3 A Minimal R Markdown Document

Save the following as `hello.Rmd` and render it:

```
---
title : " My First  R  Markdown   Document "
author :  " AI "
date : "` r Sys . Date () `"
output :  html _ document
---

##  Introduction

This  document   was  rendered   with  **R  Markdown  **.

` ``{r summary - stats }
x  <-  c (1 ,2 , 3 , 4,  5)
summary (x )
` ``

The  mean  is  `r  mean ( x ) `.
```

Listing 1.3. Minimal .Rmd document

```
library  (rmarkdown)
render ( " hello . Rmd " )

# Render  to  a  specific   format
render ( " hello . Rmd "output _ format  =  " pdf_ document ")
render ( " hello . Rmd "output _ format  =  " word_ document ")
```

Listing 1.4. Rendering from the R console

```
Rscript  -e  ' rmarkdown :: render (" hello . Rmd ") '
```

Listing 1.5. Rendering from the command line

CHAPTER 2

YAML Front Matter

2.1 Overview

Every R Markdown document begins with a YAML header delimited by three dashes (---). This header controls metadata and output options.

2.2 Core Fields

```
---
title :     " Statistical   Report "
subtitle :  " Monthly  Analysis "
author :    " AI "
date :      " ‘ rformat ( Sys . Date () ;%B  %d , %Y ’) ‘"
abstract :  |
  This  report  summarises   the  monthly
  statistical   findings .
output :
  html _ document :
    toc : true
    toc _ float :true
    number _ sections :  true
    theme :  cerulean
    highlight :  tango
    df _ print :paged
  pdf _ document :
    toc : true
    number _ sections :  true
    latex _ engine : pdflatex
    keep _ tex :true
---
```

Listing 2.1. Common YAML fields

2.3 PDF-Specific YAML Options

```
---
title :     " Statistical   Report "
author :    " AI "
date :      " ‘ rSys . Date () ‘"
output :
  pdf _ document :
    toc : true
    toc _ depth :3
    number _ sections :  true
    latex _ engine : pdflatex
    keep _ tex :false
    fig _ caption : true
```

```
    fig_width: 6
    fig_height: 4
geometry: "a4paper, margin=2.5cm"
fontsize: 11pt
documentclass: article
header-includes:
  - \usepackage{booktabs}
  - \usepackage{xcolor}
  - \definecolor{indigo}{HTML}{4B0082}
---
```

Listing 2.2. YAML for PDF output

2.4 Word Document YAML

```
---
title: "Statistical Report"
author: "AI"
output:
  word_document:
    toc: true
    reference_docx: "template.docx"
    fig_width: 6
    fig_height: 4
---
```

Listing 2.3. YAML for Word output

2.5 Parameterised Reports

Parameters are declared under the `params` key:

```
---
title: "Region Report: `r params$region`"
author: "AI"
params:
  region: "North"
  year: 2024
  alpha: 0.05
output: html_document
---
```

Listing 2.4. Parameterised YAML

Parameters are then accessed in code as `params$region`, `params$year`, etc.

CHAPTER 3

Markdown Syntax in R Markdown

3.1 Text Formatting

```
**bold text**
*italic text*
***bold and italic***
`inline code`
~~strikethrough~~
> Blockquote
```

Listing 3.1. Markdown text formatting

3.2 Headings

```
# Heading 1
## Heading 2
### Heading 3
#### Heading 4
```

Listing 3.2. Markdown headings

3.3 Lists

```
- Item 1
- Item 2
  - Nested item
  - Another nested
- Item 3

1. First
2. Second
3. Third
```

Listing 3.3. Unordered and ordered lists

3.4 Links and Images

```
[Link text](https://www.example.com)
[Link with title](https://example.com "Title")

![Alt text](path/to/image.png)
![Alt text](path/to/image.png){width=50%}
```

Listing 3.4. Links and images

3.5 Mathematics

Inline maths uses single dollar signs: `$E = mc^2$` renders as $E = mc^2$. Display maths uses double dollar signs or the equation environment.

```
Inline : $\ bar { x } = \ frac {1}{ n }\ sum{ i =1}^ nx_ i$

Display :
$$
s ^2 = \ frac {1}{ n -1}\ sum{i =1}^{ n }( xi - \ bar {x }) ^2
$$
```

Listing 3.5. Mathematics in R Markdown

CHAPTER 4

Code Chunks

4.1 Chunk Syntax

Code chunks begin with ```` ```{r} ```` and end with ```` ``` ````. A named chunk is preferred for cross-referencing:

````
```{r scatter-plot, fig.cap="Scatter plot of x vs y"}
plot(x, y, col = "#4B0082", pch = 16)
```
````

Listing 4.1. Named code chunk

4.2 Chunk Options

Option	Default	Effect
`eval`	`TRUE`	Execute the chunk
`echo`	`TRUE`	Show source code
`include`	`TRUE`	Include output in document
`message`	`TRUE`	Show messages
`warning`	`TRUE`	Show warnings
`error`	`FALSE`	Continue on error
`cache`	`FALSE`	Cache results
`fig.width`	`7`	Figure width (inches)
`fig.height`	`5`	Figure height (inches)
`fig.cap`	`NULL`	Figure caption
`fig.align`	`"default"`	Figure alignment
`out.width`	`NULL`	Output width (e.g. `"80%"`)

Table 4.1. Common knitr chunk options

4.3 Global Chunk Options

Set default options for all chunks at the top of the document:

```
```{r setup, include = FALSE}
knitr::opts_chunk$set(
  echo    = TRUE,
  message = FALSE,
  warning = FALSE,
  fig.width  = 7,
  fig.height = 5,
  fig.align  = "center",
  cache   = FALSE,
  comment = "#>"
)
```
```

```
```
```

**Listing 4.2.** Setting global chunk options

## 4.4 Inline Code

Values computed in R can be inserted directly into text:

```
The sample mean is `r round(mean(x), 2)` and the
standard deviation is `r round(sd(x), 2)`.
```

Listing 4.3. Inline R evaluation

4.5 Child Documents

Large reports can be split into multiple `.Rmd` files:

```
```{r child = "sections/methods.Rmd"}
```

```{r child = "sections/results.Rmd"}
```
```

**Listing 4.4.** Including child documents
```

CHAPTER 5

Output Formats

5.1 HTML Documents

HTML is the most flexible R Markdown output format and supports interactive elements.

```
output :
  html _ document :
    toc : true
    toc _ depth :3
    toc _ float :
      collapsed :  false
      smooth _ scroll : true
    number _ sections :  true
    theme :  flatly          # or  cerulean ,  cosmo , journal ,  lumen ,
                             #     paper , readable ,  sandstone ,
                             #     simplex ,  spacelab ,  united ,  yeti
    highlight :  zenburn
    code _ folding : hide    # show  | hide  | none
    df _ print :paged
    self _ contained : true
    css : custom . css
```

Listing 5.1. HTML document options

5.2 PDF Documents

PDF output uses LaTeX as an intermediate format.

```
output :
  pdf _ document :
    toc : true
    toc _ depth :3
    number _ sections :  true
    latex _ engine : pdflatex
    fig _ caption : true
    fig _ width :  6
    fig _ height : 4
    keep _ tex :false
    extra _ dependencies :
      - booktabs
      - longtable
      - array
```

Listing 5.2. Full PDF document options

5.2.1 Embedding Raw LaTeX

```
The  coefficient   of  variation   is  defined   as :
```

```
$$CV = \frac{s}{\bar{x}} \times 100\% $$

\begin{equation}
  \hat{\beta} = (X^{\top} X)^{-1} X^{\top} y
  \label{eq:ols}
\end{equation}

Equation \ref{eq:ols} is the OLS estimator.
```

Listing 5.3. Embedding LaTeX in Rmd

5.3 Word Documents

```
output:
  word_document:
    toc: true
    toc_depth: 2
    fig_width: 6
    fig_height: 4
    reference_docx: "my_template.docx"
```

Listing 5.4. Word document configuration

5.3.1 Creating a Reference Document

```
library(rmarkdown)
draft("template.Rmd", template = "word_document")
render("template.Rmd", output_format = "word_document")
# Then open the .docx in LibreOffice, modify styles,
# save as reference_docx
```

Listing 5.5. Generate a reference .docx

Part II

Advanced R Markdown

CHAPTER 6

Tables

6.1 knitr::kable

```
library (knitr)
df <- data.frame(
  Method     = c("t - test ","Wilcoxon ","ANOVA "),
  Statistic  = c(2.34, 12.1, 5.67),
  p_value    = c(0.021, 0.034, 0.006)
)
kable(df, digits = 3,
      caption = "Statistical Test Results",
      col.names = c("Method","Statistic","p - value"),
      align = c("l","r","r"))
```

Listing 6.1. Basic kable table

6.2 kableExtra

```
library (kableExtra)
df |>
  kable(digits = 3,
        caption = "Regression Coefficients") |>
  kable_styling(
    bootstrap_options = c("striped","hover","condensed"),
    full_width = FALSE,
    position = "center"
  ) |>
  row_spec(0, bold = TRUE,
           background = "#4B0082", color = "white") |>
  column_spec(3, bold = TRUE,
              color = ifelse(df$p_value < 0.05, "#4B0082","black"))
```

Listing 6.2. Styled table with kableExtra

6.3 gt Tables

```
# install.packages("gt")
library (gt)
mtcars[1:6, c("mpg","cyl","hp","wt")]
  gt() |>
  tab_header(title = "Motor Trend Car Road Tests",
             subtitle = "First six rows") |>
  cols_label(mpg = "MPG", cyl = "Cylinders",
             hp = "Horsepower", wt = "Weight (1000 lbs)") |>
  tab_style(
    style = cell_fill(color = "#4B0082") |>
```

```
                cell_text(color = "white"),
    locations   = cells_column_labels()
  ) |>
  fmt_number(columns = c(mpg, wt), decimals = 1)
```

Listing 6.3. Publication-quality tables with gt

CHAPTER 7

Figures

7.1 Base R Graphics

````
```{rbase-hist, fig.cap="Histogram of sample data",
    fig.width=6, fig.height=4, out.width="80%"}
set.seed(42)
x <- rnorm(500)
hist(x, col = "#4B0082", border = "white",
     main = "Sample Histogram", xlab = "Value")
abline(v = mean(x), col = "#C8A84B", lwd = 2)
```
````

Listing 7.1. Chunk producing a base R figure

7.2 Figure Cross-References

Cross-references work when using `bookdown` output formats:

```
output: bookdown::html_document2
```

Listing 7.2. YAML for bookdown output

````
See Figure \@ref(fig:scatter-plot) for the scatter plot.

```{rscatter-plot, fig.cap="Scatter plot of x vs y"}
plot(x, y, col = "#4B0082", pch = 16)
```
````

Listing 7.3. Cross-referencing a figure

CHAPTER 8

Citations and Bibliographies

8.1 BibTeX Integration

```
---
title : " Analysis  Report "
bibliography :  references . bib
csl : apa . csl
link - citations : true
output :  pdf _ document
---
```

Listing 8.1. YAML for bibliography

```
@book { wickham2023 ,
  author      = { Hadley  Wickham  and  Mine  \c { C} etinkaya - Rundel
                  and  Garret  Grolemund } ,
  title       = {R  for  Data  Science },
  edition     = {2 nd },
  year        = {2023} ,
  publisher   = {O ' Reilly }
}

@article { breiman2001 ,
  author    = { Leo Breiman } ,
  title     = { Random  Forests } ,
  journal   = { Machine  Learning },
  year      = {2001} ,
  volume    = {45} ,
  pages     = {5 - -32}
}
```

Listing 8.2. Sample references.bib file

```
According  to  @wickham2023 ,  the  pipe  operator ...

Random  forests  were  introduced  by  [- @breiman2001 ].

Multiple  citations :  [ @wickham2023 ;  @breiman2001 ].
```

Listing 8.3. Citing in R Markdown text

CHAPTER 9

Interactive Documents

9.1 htmlwidgets

`htmlwidgets` allows R to generate JavaScript-powered interactive visualisations.

```
# install . packages (" DT ")
library ( DT )
datatable (
  mtcars [, c( " mpg " ," cyl " ," hp " ," wt ") ] ,
  options = list (
    pageLength = 10 ,
    dom = " Bfrtip " ,
    buttons = c( " copy " ," csv " ," excel " ," pdf " )
  ) ,
  filter = " top " ,
  rownames = TRUE
)
```

Listing 9.1. Interactive table with DT

```
# install . packages (" leaflet ")
library ( leaflet )
leaflet () | >
  addTiles () | >
  addMarkers ( lng = -0.1278 , lat = 51.5074 ,
               popup = " London ")
```

Listing 9.2. Interactive maps with leaflet

```
# install . packages (" plotly ")
library ( plotly )
p <- ggplot ( mtcars , aes (wt, mpg ,
              colour = factor ( cyl ) ,
              text = rownames( mtcars ) ) )+
  geom _ point ( size = 3) +
  theme _ minimal ()
ggplotly (p )
```

Listing 9.3. Interactive charts with plotly

9.2 Shiny Integration

Adding `runtime: shiny` to the YAML converts the document into an interactive application.

```
---
title : " Interactive Report "
output : html _ document
runtime : shiny
---
```

Listing 9.4. Shiny-enabled R Markdown

```
‘ ‘‘{r shiny -hist , echo = FALSE }
sliderInput (" n " ," Sample size : " ,min = 10 , max = 1000 , value = 100)

renderPlot ({
  x <- rnorm( input$n )
  hist (x , col = " #4 B0082 " ,border = " white " ,
        main = paste ( " n= " ,input $n ))
})
‘‘‘
```

Listing 9.5. Shiny input and output in the Rmd body

CHAPTER 10

Custom Templates and Styling

10.1 Custom CSS for HTML Output

```
output :
  html _ document :
    css : custom . css
```

Listing 10.1. Applying custom CSS

```
body  {
  font - family : sans - serif ;
  background :  linear - gradient ( to bottom ,
    rgba (255 ,255 ,255 ,0) ,
    rgba (75 ,0 ,130 ,0.15) ,
    rgba (75 ,0 ,130 ,0.35) ) ;
}

h1 , h2 , h3  {  color :  #4 B0082 ; }

. code - block {
  background :  # F3F0FB ;
  border - left : 3 px solid  #4 B0082 ;
  padding :  8 px 12 px ;
}

table  th  {
  background - color :  #4 B0082 ;
  color :  white ;
}
```

Listing 10.2. custom.css matching the brand palette

10.2 Custom LaTeX Template

For complete control over PDF output, supply a custom `.tex` template:

```
output :
  pdf _ document :
    template :  custom _ template . tex
    latex _ engine : pdflatex
```

Listing 10.3. Using a custom LaTeX template

The template file must contain the Pandoc variables `$body$`, `$title$`, `$author$`, etc. Pandoc replaces these with the document content at render time.

10.3 Branding a Report with a Custom Function

```
brand_table <- function(df, caption = NULL) {
  library(kableExtra)
  df |>
    knitr::kable(caption = caption) |>
    kable_styling(full_width = FALSE) |>
    row_spec(0, background = "#4B0082", color = "white",
             bold = TRUE)
}

brand_plot <- function(p) {
  p + ggplot2::theme(
    plot.title  = ggplot2::element_text(colour = "#4B0082",
                                        face = "bold"),
    axis.title  = ggplot2::element_text(colour = "#4B0082"),
    axis.line   = ggplot2::element_line(colour = "#4B0082")
  )
}
```

Listing 10.4. Brand helper functions for consistent styling

CHAPTER 11

Parameterised Reports

11.1 Declaring and Using Parameters

```
---
title : "‘ r paste (’ Report  for ’, params $region ) ‘"
params :
  region :  " North "
  year :    2024
  alpha :   0.05
output :  pdf _ document
---

# Analysis :  ‘r  params $region ‘ , ‘r  params $year ‘

Significance    level : $\ alpha = ‘r  params $alpha ‘$.
```

Listing 11.1. Parameter declaration and use in text

11.2 Rendering Multiple Reports Programmatically

```
library  (rmarkdown)
library  ( purrr )

regions  <- c( " North " ," South " ," East " ," West ")

walk ( regions , function  ( r) {
  render (
    input          = " regional _ report . Rmd " ,
    output _ file  = paste0 ( " report_" , r , " . pdf ") ,
    params         = list  ( region  = r , year  = 2024)
  )
})
```

Listing 11.2. Batch rendering with purrr

Part III

PerlTeX Integration

CHAPTER 12

Introduction to PerlTeX

12.1 What Is PerlTeX?

PerlTeX is a LaTeX package that allows LaTeX macro definitions to be written in Perl. The package works by keeping a Perl interpreter running as a co-process alongside TeX; when TeX encounters a `\perlnewcommand` definition, it sends the Perl code to the interpreter and receives back the result as TeX tokens.

This makes it possible to perform arbitrary computation—string manipulation, file I/O, regular expressions, numerical processing—inside a LaTeX document, and to feed the results directly into the typeset output.

12.2 Installing PerlTeX on Fedora

```
# PerlTeX ships with TeX Live
sudo dnf install texlive-perltex perl

# Verify
perltex --version
```

Listing 12.1. Installing PerlTeX on Fedora

12.3 Compiling with PerlTeX

Documents that use PerlTeX **must** be compiled with the `perltex` wrapper rather than `pdflatex` directly:

```
perltex --latex=pdflatex mydoc.tex

# Allow arbitrary Perl (use with caution)
perltex --nosafe --latex=pdflatex mydoc.tex
```

Listing 12.2. Compiling a PerlTeX document

[style=warningboxstyle]**Warning:** The `-nosafe` flag disables the Safe compartment that restricts Perl code. Use it only with documents whose Perl code you have written or fully trust, as it allows unrestricted file and system access.

12.4 A Minimal PerlTeX Example

```
\documentclass{article}
\usepackage{perltex}

\perlnewcommand{\today@long}{
  use POSIX qw(strftime);
  return strftime("%A, %B %d, %Y",localtime);
}
```

```
8
9  \ begin { document }
10 Today  is  \ today@long .
11 \ end { document }
```

Listing 12.3. Minimal PerlTeX document

CHAPTER 13

PerlTeX Macro Programming

13.1 Defining Commands with \perlnewcommand

The syntax mirrors LaTeX's \newcommand:

```
\perlnewcommand{\factorial}[1]{
  my $n = $_[0];
  my $result = 1;
  $result *= $_ for 2..$n;
  return $result;
}

$10! = \factorial{10}$   % produces 3628800
```

Listing 13.1. perlnewcommand with arguments

13.2 String Processing

```
\perlnewcommand{\titlecase}[1]{
  my $str = $_[0];
  $str =~ s/\b(\w)/uc($1)/ge;
  return $str;
}

\titlecase{the quick brown fox}
% produces: The Quick Brown Fox
```

Listing 13.2. String manipulation macro

13.3 Reading External Data

```
\perlnewcommand{\csvlookup}[2]{
  # Args: filename, field name
  my ($file, $field) = @_;
  open my $fh, '<', $file or return "ERROR";
  my $header = <$fh>;
  chomp $header;
  my @cols = split /,/, $header;
  my %idx;
  @idx{@cols} = (0..$#cols);
  while (<$fh>) {
    chomp;
    my @vals = split /,/;
    return $vals[$idx{$field}] if defined $idx{$field};
  }
  return "N/A";
}
```

```
The sample mean was \csvlookup{results.csv}{mean}.
```

Listing 13.3. Reading a CSV value into a LaTeX macro

13.4 Numerical Computations

```
\perlnewcommand{\binomcoeff}[2]{
  my ($n, $k) = @_;
  use List::Util qw();
  my $result = 1;
  for my $i (0..$k-1) {
    $result = $result * ($n - $i) / ($i + 1);
  }
  return int($result + 0.5);
}

$\binom{10}{3} = \binomcoeff{10}{3}$
```

Listing 13.4. Numerical computation in PerlTeX

CHAPTER 14

R Markdown with PerlTeX

14.1 Motivation

Combining R Markdown with PerlTeX enables a powerful three-tier pipeline:

1. **R** executes statistical computations and writes results to intermediate files (CSV, JSON, plain text).
2. **PerlTeX macros** read those files and make the values available as LaTeX commands.
3. **pdflatex** (via the `perltex` wrapper) typesets the final document with the numerical results embedded.

14.2 Configuring the Knit Engine

To compile the `.tex` file produced by knitr with `perltex` instead of `pdflatex`, set the `knit` option in the YAML:

```
---
title :  " Statistical  Report  with  PerlTeX "
author :  " AI "
output :
  pdf _ document :
    latex _ engine : pdflatex
    keep _ tex :true
knit : |
  ( function ( input , ...)  {
    rmarkdown :: render (
      input ,
      output _ format  = " pdf _ document " ,
      ...
    )
  })
header - includes :
  - \ usepackage { perltex }
---
```

Listing 14.1. YAML for Rmd+PerlTeX pipeline

[style=noteboxstyle]**Note:** Because `perltex` wraps `pdflatex`, the `latex_engine` in the YAML should remain `pdflatex`. The PerlTeX compilation step is applied to the `.tex` file produced by knitr using a custom post-processing script.

14.3 A Complete Rmd + PerlTeX Workflow

The recommended approach is a two-stage pipeline:

1. Use `knitr::knit()` to convert `report.Rmd` to `report.tex`, running all R code chunks and writing interim results to files.

2. Call `perltex` on the resulting `.tex` file.

```r
library(knitr)

# Stage 1: run R chunks, produce .tex
knit("report.Rmd", output = "report.tex")

# Stage 2: compile with PerlTeX
system("perltex --nosafe --latex=pdflatex report.tex")
system("perltex --nosafe --latex=pdflatex report.tex")  # twice for TOC
```

Listing 14.2. Two-stage knit + perltex pipeline in R

```
#!/usr/bin/env bash
Rscript -e 'knitr::knit("report.Rmd", output = "report.tex")'
perltex --nosafe --latex=pdflatex report.tex
perltex --nosafe --latex=pdflatex report.tex
```

Listing 14.3. Same pipeline as a shell script

14.4 Sharing Results Between R and PerlTeX

The cleanest approach is to write R results to a simple key–value text file and read it from PerlTeX macros.

```r
```{r write-results, include=FALSE}
results <- list(
  n          = nrow(mtcars),
  mean_mpg   = round(mean(mtcars$mpg), 2),
  sd_mpg     = round(sd(mtcars$mpg), 2),
  p_value    = round(t.test(mtcars$mpg, mu = 20)$p.value, 4)
)

lines <- paste0(names(results), "=", unlist(results))
writeLines(lines, "results.txt")
```
```

Listing 14.4. Writing results from an R chunk

```
\usepackage{perltex}

\perlnewcommand{\Rresult}[1]{
  my $key = $_[0];
  open my $fh, '<', 'results.txt' or return '??';
  while (<$fh>) {
    chomp;
    my ($k, $v) = split /=/, $_, 2;
    return $v if $k eq $key;
  }
  return '??';
}
```

Listing 14.5. PerlTeX macro to load the results file

```
The dataset contains \Rresult{n} observations.
The mean fuel efficiency was \Rresult{mean_mpg}~mpg
(SD~$=$ \Rresult{sd_mpg}$), which differed significantly
from 20~mpg ($p = \Rresult{p_value}$).
```

Listing 14.6. Using the macro in LaTeX body text

CHAPTER 15

Advanced PerlTeX Patterns

15.1 Generating LaTeX Tables from R Data

```
```{r make-table, include = FALSE}
library(dplyr)
tab <- mtcars |>
  group_by(cyl) |>
  summarise(n = n(), mean_mpg = round(mean(mpg),1),
            sd_mpg = round(sd(mpg),1))

# Write as booktabs LaTeX table body
rows <- apply(tab,1, function(r)
  paste0(r["cyl"], "&", r["n"], "&",
         r["mean_mpg"], "&", r["sd_mpg"], "\\\\"))
writeLines(rows, "table_body.tex")
```
```

Listing 15.1. R chunk: write table as LaTeX

```
\begin{table}[H]
  \centering
  \begin{tabular}{rrrr}
    \toprule
    Cylinders & $n$ & Mean MPG & SD \\
    \midrule
    \input{table_body.tex}
    \bottomrule
  \end{tabular}
  \caption{Fuel efficiency by number of cylinders.}
\end{table}
```

Listing 15.2. LaTeX table shell that reads R output

15.2 Conditional Formatting with PerlTeX

```
\perlnewcommand{\pval}[1]{
  my $p = $_[0];
  if ($p < 0.001) {
    return "\\textcolor{indigo}{\\textbf{<0.001}}";
  } elsif ($p < 0.05) {
    return "\\textcolor{indigo}{$p}";
  } else {
    return $p;
  }
}

The p-value was \pval{0.024}.
```

Listing 15.3. Colour significant p-values automatically

15.3 Automating Report Sections

```
\ perlnewcommand {\ groupsections }{
   open  my  $fh , '<',  ' groups . txt 'or  return   ' ';
   my  $out  =  ' ';
   while  ( my $line  = <$fh >) {
      chomp  $line ;
      $out  .=  "\\ subsection { Group : $line }\ n ";
      $out  .=  " Results  for  group  $line  are  shown  above .\ n ";
   }
   return  $out ;
}

\ groupsections
```

Listing 15.4. Generate one section per group from a file

CHAPTER 16

Workflows and Best Practices

16.1 Project Structure

```
myproject/
├── data/
│   ├── raw/            # Original, unmodified data
│   └── processed/      # Cleaned/transformed data
├── R/
│   ├── 01_clean.R
│   ├── 02_analyse.R
│   └── helpers.R
├── reports/
│   ├── report.Rmd
│   ├── custom.css
│   └── refs.bib
├── output/
│   ├── report.pdf
│   └── figures/
├── tests/
│   └── testthat/
│       └── test-helpers.R
├── Dockerfile
├── .gitignore
└── README.md
```

Listing 16.1. Recommended project directory structure

16.2 Version Control for Reports

- Commit `.Rmd` source files, never the compiled outputs.
- Store large data files with Git LFS or exclude them with `.gitignore`.
- Use a `_freeze/` or `cache/` directory for knitr cache, which may be version-controlled to speed up CI/CD pipelines.

APPENDIX A

YAML Quick Reference

Key	Applies to	Values
`toc`	html, pdf, word	`true`/`false`
`toc_depth`	html, pdf, word	integer (default 3)
`toc_float`	html	`true`/`false`
`number_sections`	html, pdf	`true`/`false`
`theme`	html	Bootstrap theme name
`highlight`	html, pdf	Syntax highlight style
`fig_width`	html, pdf, word	numeric (inches)
`fig_height`	html, pdf, word	numeric (inches)
`keep_tex`	pdf	`true`/`false`
`latex_engine`	pdf	`pdflatex`, `xelatex` . . .
`css`	html	path to CSS file
`reference_docx`	word	path to .docx template
`runtime`	html	`shiny`

Table A.1. Common YAML output options

APPENDIX B

knitr Chunk Option Quick Reference

Option	Type	Default
`eval`	logical	`TRUE`
`echo`	logical	`TRUE`
`include`	logical	`TRUE`
`results`	character	`"markup"`
`message`	logical	`TRUE`
`warning`	logical	`TRUE`
`error`	logical	`FALSE`
`cache`	logical	`FALSE`
`cache.path`	character	`"cache/"`
`fig.width`	numeric	`7`
`fig.height`	numeric	`5`
`fig.align`	character	`"default"`
`fig.cap`	character	`NULL`
`out.width`	character	`NULL`
`dev`	character	`"pdf"` (PDF), `"png"` (HTML)

Table B.1. knitr chunk option reference

Bibliography

[1] Xie, Y., Allaire, J.J. and Grolemund, G. (2018). *R Markdown: The Definitive Guide*. Chapman & Hall/CRC Press.

[2] Xie, Y., Dervieux, C. and Riederer, E. (2020). *R Markdown Cookbook*. Chapman & Hall/CRC Press.

[3] Pakin, S. (2007). *The PerlTeX Package*. CTAN:`texmf/tex/latex/perltex/`.

[4] Wickham, H., Çetinkaya-Rundel, M. and Grolemund, G. (2023). *R for Data Science*, 2nd ed. O'Reilly.