Many thanks to the anonymous reviewer and to the editor for identifying numerous issues with the submission (and one bug in R!) and for providing helpful suggestions. I have attempted to address each of the issues in my revised submission, as described below.

A new version of **xdvir** (0.1-3) has been submitted to CRAN to support some of the revisions.

Response to Anonymous Reviewer

```
It may be worth mentioning the origin of the name of the package - DVI files
may be well known to LaTeX users but these are not mentioned until well into
the article, and are not explained at that point. An introduction of the
flow of .tex (source) -> .dvi (binary) -> .ps (render) may serve both new and
experienced users well. For those not familiar with DVI files, the name may
not seem related.
```

I have slightly expanded the first mention of `xdvir` on page 2, but because the full explanation is more detailed, in order to avoid breaking the flow of the main text, the proper story of the package name is added in a footnote (on the same page).

```
Nitpick: the {gridtext} and {ggtext} references could be reversed; these
render as "gridtext (Wilke and Wiernik, 2022b) and ggtext (Wilke and Wiernik,
2022a)" ('b' then 'a'). I notice the same ordering appears in the vignette.
```

I have swapped the order of the packages so that the citations are now “a” followed by “b” [page 1].

```
"The gridtext and ggtext packages make it possible to change color within a
character value, but they do not allow a mixture of plain text and
mathematical expressions." - it may be worthwhile explaining that these do
not support ‘<math>‘ tags which would otherwise be available to markdown
rendering. Whether this is a technical limitation or a lack of implementation
may be outside of the scope of this article, but that difference potentially
impacts the strength of the argument that "this cannot be done".
```

I have clarified that “mathematical expressions” means “R mathematical expressions” and added a mention of the the lack of support for MathML (in `gridtext` and `ggtext`) [page 1].

```
Nitpick: "The R graphics system can draw a character value across multiple
lines, but only if explicit newlines are embedded in the character value
(i.e., the line breaks are manual)." - I would challenge the use of the word
"manual" here; ‘base::strwrap()‘ is able to achieve this, though as noted it
embeds the linebreaks in the string; is not justified; and does not hyphenate.

‘‘‘
txt <- "We ‘move’ the original population’s mean to a new
z_i and calculate the average fitness at that new mean
phenotype of the population to get the adaptive landscape,
W_i, then we combine the population mean and the average
fitness to get the fitness function."
plot(0:10, seq(0, 1, 0.1))
text(2.5, 0.8, paste(strwrap(txt, width = 45), collapse = "\n"))
‘‘‘
```

I have added a mention of the `strwrap()` function (and its limitations) [page 1].

```
"Figure 1" - it may be helpful to describe the contents of this figure in the
caption more clearly; viewing the article in monochrome, the significance of
colour is lost.
```

I have expanded the figure caption to explain where the changes in colour occur and the connection between the colours in the text and the colours in the plot lines.

```
"Figure 1: A plot with a text annotation that contains several typesetting
challenges: in-line mathematical equations;" - should be
"mathematical expressions"?
```

I have changed to using "mathematical expression" throughout (and removed any use of "mathematical equation").

```
The ’supplementary materials’ mentioned several times in the text appear to be
incomplete; the ‘TeX‘ directory was not supplied. I was hoping to better
understand the use of "local" LaTeX packages with the ’annotate-equations’
example which references

‘‘‘
LaTeXpackage(name="annotate",
             preamble="\\usepackage{TeX/annotate-equations}")
‘‘‘

and it was unclear as to where ’TeX’ refers. It seems this is a local,
relative directory, made clearer by a reference in the ‘purl‘ output

‘‘‘
schneiderLines <- readLines("TeX/schneider.tex")
‘‘‘

but this directory (and subsequently the .tex file) is omitted.
```

I am sorry that the `TeX` directory did not make it through. The `.zip` file that I submitted does contain that directory and the following files/paths were explicity mentioned in the R Journal submission:

```
data/chile.csv, data/auckland-flights.csv, data/youth-crime.csv, diagram/diag.tex,
Fonts/Economica-Regular.ttf, Fonts/Economica-BoldItalic.ttf,
Fonts/Economica-Italic.ttf, Fonts/Economica-Bold.ttf,
scripts/anzjs.R, scripts/schneider.R, scripts/rahlf-plot.R,
TeX/schneider.tex, TeX/longley.tex, TeX/rahlf.tex, TeX/annotate-equations.sty
```

Hopefully the R Journal Editor and/or technical team will be able to assist me with making sure that all of these files are included in the final submission.

```
Some discussion regarding system-wide packages versus "local" package may be
of benefit - can a seasoned LaTeX user leverage their suite of packages just
by adding the appropriate `\\usepackage{my-package}` preambles?
```

Yes, a simple `\usepackage{my-package}` will suffice for most cases. That approach was demonstrated in the previous example, but I have made the difference more explicit in the commentary on this local example.

```
"Figure 12" and "Figure 13" - very hard to read in monochrome.
```

I have significantly lightened the "blue" and darkened the "yellow" so that, in monochrome, the "Male" elements are easier to distinguish.

```
(Section 14: Discussion)
It may be worthwhile adding some brief notes on the availability of packages
for rendering LaTeX which don't have a stated goal of integrating into plots;
e.g. {texPreview} and {latexpdf}.

A link to the 'Literate Programming' section of the 'Reproducible Research'
CRAN Task View

https://cran.r-project.org/web/views/ReproducibleResearch.html

may be of value here.
```

This issue is mentioned briefly in Section 8 (Programmatic generation of LaTeX), but I have expanded that discussion and added references to those packages and the CRAN Task View there as well [page 11].

```
On Loading the package, user is presented with potentially cryptic details:

‘‘‘
            TeX:  /Library/TeX/texbin/latex
          xetex:  XeTeX 3.141592653-2.6-0.999996 (TeX Live 2024)
         luatex:  This is LuaTeX, Version 1.18.0 (TeX Live 2024)
luaotfload-tool:  3.28
‘‘‘

A header line explaining that this is the active configuration may be helpful,
if this information is really required. Otherwise, offering a mechanism to
purposefully expose it might be less confusing.
```

Most of this information is detail that the typical user does not need, so most of this messaging has been moved to a new `TeXstatus()` function that can be called as required. There will still be a startup message if TeX cannot be found at all, warning that typesetting features will not be available (and suggesting to look at installing TEX via **tinytex**).

```
The article text notes:

"The package start-up message reports on whether these are available."

but I do not consider this a report on availability; merely a listing. Users
unfamiliar with the TeX family of software may find these names very
confusing.

Given that this references a system installation of LaTeX, it raises the
question of the dependence on {tinytex} - is that merely a fallback? Are there
resolution strategies when packages are added to one or the other?
```

There is a reliance on **tinytex** to find (and use) a system install of LATEX. No attempt is made by this package to resolve issues with co-existence of **tinytex** and a system LATEX.

As mentioned in the previous response, the package startup message has been reduced to, at most, a message that TEX is not available with a pointer to **tinytex**.

In other words, **xdvir** leans as much as possible on **tinytex** for TEX support.

I have added a “trouble-shooting” vignette with the idea of building up some suggestions for getting around problems. This includes a link to the **tinytex** advice on trouble-shooting LATEX problems.

```
The article text notes:

"An implicit limitation is that xdvir requires a TEX installation, though that
is simplified through a dependency on the tinytex package (Xie, 2024)."

but I have not tested the package on a system lacking a TeX installation.
```

The package source is hosted on github (https://github.com/pmur002/xdvir) and that includes CI on the three major platforms both with and without TeX installations.

```
There does not appear to be a package-wide help file (e.g. xdvir-package)
which would enable ‘?xdvir‘. There _is_ a vignette which can be found with
‘??xdvir‘ and this is a helpful addition.
```

A package-wide help file has been added (which documents a couple of options that were otherwise invisible to the user).

```
Help files are minimal; ‘?element_latex‘, ‘?geom_latex‘, ‘?LaTeXpackage‘
provide no examples.
```

Examples and further explanations have been added to these help pages. The main user documentation is the introductory vignette, plus this article. More detailed information is provided in a technical report (https://stattech.wordpress.fos.auckland.ac.nz/2025/03/06/2025-01-latex-typesetting-in-r/); a link to that has been added to the DESCRIPTION file.

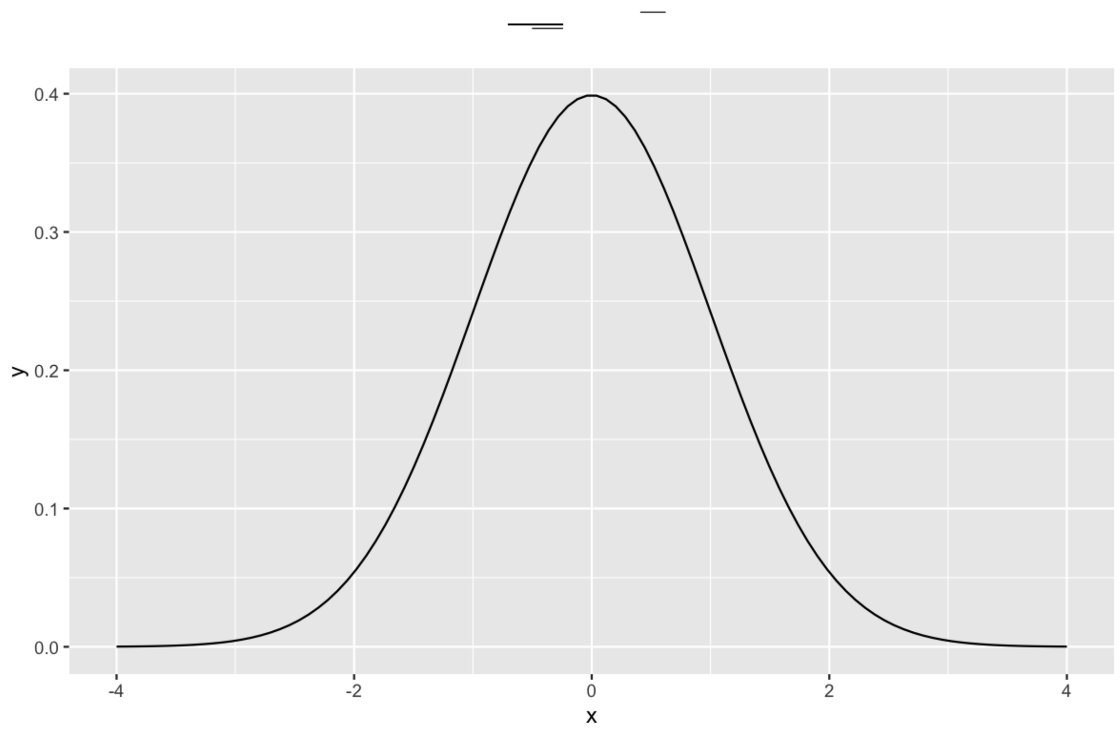
```
The mingling of R and LaTeX code is obviously a fine line to walk. I wonder if
it would make sense to add some R wrappers which limit the amount of LaTeX
which needs to be written, for those less familiar with the document
structure, but are familiar with the expression syntax. For example, in this
block

‘‘‘
annotateEquations <-
    LaTeXpackage(name="annotate",
                 preamble="\\usepackage{TeX/annotate-equations}")
registerPackage(annotateEquations)
‘‘‘

the ‘preamble‘ argument requires writing the LaTeX command ‘\\usepackage{}‘
with the remaining code all being R. A ‘source=‘ argument would
enable not interleaving LaTeX code, but would expand to the ‘\\usepackage{}‘
line. This need not prevent additional preamble from being added.
```

This is a tricky call. One argument for the current arrangement is that the `\usepackage` command may include options (within square brackets). The approach taken is to reduce the burden on the user to “just” LaTeX fragments, but always with the fallback of the user writing complete LaTeX code if that is what it takes. I am not convinced that this particular additional argument is worth adding.

```
It is not clear why ‘LaTeXpackage()‘ should not perform the
‘registerPackage()‘ step itself; the article does not reference the
‘annotateEquations‘ object after registration, merely the name "annotate"
provided there as an argument. Is that object useful outside of the
registration step?
```

The value returned by `LaTeXpackage()` is an R object of class `"LaTeXpackage"`. The value specified for the `packages` argument in functions like `grid.latex()` and `author()` can be any combination of character values and `"LaTeXpackage"` objects (in a list if necessary). The `registerPackage()` function allows the user to just specify the name of the package as a character value, but there are cases where it may be more convenient just to create a `"LaTeXpackage"` object without registering (e.g., it is only possible to register the same name once, but the user might want to generate multiple variations of the same package, with different package options, but the same name, which they can do if they just create the `"LaTeXpackage"` objects without registering them).

```
Attempting to run one of the examples results, some issues are encountered

‘‘‘
library(xdvir)
#>             TeX:  /Library/TeX/texbin/latex
#>           xetex:  XeTeX 3.141592653-2.6-0.999996 (TeX Live 2024)
#>          luatex:  This is LuaTeX, Version 1.18.0 (TeX Live 2024)
#> luaotfload-tool:  3.28
packageVersion("xdvir")
#> [1] ’0.1.2’
library(ggplot2)
packageVersion("ggplot2")
#> [1] ’3.5.1’

tex <- r"(\huge $\Phi(z) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{z^2}{2}}$)"
x <- seq(-4, 4, length.out=100)
df <- data.frame(x=x, y=dnorm(x))

gg <- ggplot(df) + geom_line(aes(x, y))

gg +
  labs(title=paste("The Normal Distribution:", tex)) +
  theme(plot.title=element_latex())

Created on 2025-04-06 with [reprex v2.1.1](https://reprex.tidyverse.org)
‘‘‘

(plot export attached). Test was performed on a Mac.

I _am_ able to generate the rendered TeX with ‘plot.new(); grid.latex(tex)‘.
```



This was a bug in the glyph-rendering code for the `quartz()` graphics device. The bug has been fixed in the development version of R.

```
The package contains a ‘tests‘ directory, but it appears to not involve a
testing framework (e.g. {testthat}). Perhaps these tests are run manually?
The test code appears to only check that it runs without error, not that it
produces an expected result. I do not see any tests for {ggplot2} generated
figures, and no validation that this produces an expected result. Such a test
may have caught the above issue.
```

I have a large regression test suite (including **ggplot2** tests) that makes use of **gdiff** to check for visual differences. These tests are not part of the CRAN submission, partly because of the size of the control images and partly because of the time required to run these tests.

Response to Editor

```
The LaTeX \color{} command does not take a second argument. It simply changes
the color from that point onwards. I think you mean \textcolor{}{}.
```

Inappropriate uses of `\color{}` have been changed to `\textcolor{}`.

```
In Figure 6, please use em-dashes (---) rather than hyphens (-).
```

Hyphens have been replaced with em-dashes in Figure 6.

> You mix plain TeX commands within LaTeX environments. I suggest you replace {\bf ..} with \textbf{..} and {\it ..} with \textit{..}. The results are not identical.

Instances of `{\bf ...}` have been replaced with `\textbf{...}`.

> In the discussion, it is perhaps worth including the ggtikz package.

The **ggtikz** package is now mentioned in the Discussion, within the paragraph on the **tikzDevice** package [page 22].