Coding Practice Guides suggested by RUM:

[https://google.github.io/styleguide/Rguide.xml](https://outlook.manchester.ac.uk/owa/redir.aspx?C=xStQclSWDhcZM1Ke5pObZeIcb6O-4poVz7whx2-P1UD-FoSNBOjWCA..&URL=https%3a%2f%2fgoogle.github.io%2fstyleguide%2fRguide.xml)

[http://adv-r.had.co.nz/Style.html](https://outlook.manchester.ac.uk/owa/redir.aspx?C=6igXe5DDUI7-LjIbnNCo4U4gIcyHlNBhhbVjFkX-TT3-FoSNBOjWCA..&URL=http%3a%2f%2fadv-r.had.co.nz%2fStyle.html)

[https://style.tidyverse.org/](https://outlook.manchester.ac.uk/owa/redir.aspx?C=Q9L0vf8CutoLMsdeyd6TtCLHZnO_fratFZS7BTrKT4j-FoSNBOjWCA..&URL=https%3a%2f%2fstyle.tidyverse.org%2f)

[https://journal.r-project.org/archive/2012-2/RJournal\_2012-2\_Baaaath.pdf](https://outlook.manchester.ac.uk/owa/redir.aspx?C=-d-qT0Zg-WNeEPe1kEcG9MePGfbUIVPLrwYnppT0bGkzO4uNBOjWCA..&URL=https%3a%2f%2fjournal.r-project.org%2farchive%2f2012-2%2fRJournal_2012-2_Baaaath.pdf)

<ftp://ftp.math.ethz.ch/sfs/pub/Software/CRAN/web/packages/rockchalk/vignettes/Rstyle.pdf>

“To reach some sort of consistency, it might be helpful to  
(a) create a project file (.Rproj) where some style is suggested  
(b) in case you use Rstudio turn some of the diagnostics (Global  
Options -> Code -> Diagnostics) that remind you of coding practices and  
other things (sadly not portable across devices as far as I know)  
(c) use a linter, such as [https://github.com/jimhester/lintr](https://outlook.manchester.ac.uk/owa/redir.aspx?C=F74uMGi_-2h4A2tTA5-akHgUCdFwZfcQONzsdBdk_aAzO4uNBOjWCA..&URL=https%3a%2f%2fgithub.com%2fjimhester%2flintr) which  
allows you to check the code written against a style guide (does not  
seem to be well integrated within Rstudio though)”

[https://github.com/jennybc/here\_here](https://outlook.manchester.ac.uk/owa/redir.aspx?C=9Y2MqFiOg_nNy55LVXYc7eJPYpMLro50_ItZg04oHpYE5ZuNBOjWCA..&URL=https%3a%2f%2fgithub.com%2fjennybc%2fhere_here)

[https://www.britishecologicalsociety.org/wp-content/uploads/2017/12/guide-to-reproducible-code.pdf](https://outlook.manchester.ac.uk/owa/redir.aspx?C=AZxRmlACdqCRiJh5-P_YqvibG4ehPCOBvMHg3pHpfYLVjqyNBOjWCA..&URL=https%3a%2f%2fwww.britishecologicalsociety.org%2fwp-content%2fuploads%2f2017%2f12%2fguide-to-reproducible-code.pdf)

[http://doi.org/gbkbwp](https://outlook.manchester.ac.uk/owa/redir.aspx?C=QGT0qQCZfd2tgpyHjpoFKSH3sHmkG3oa3Yj4ZrkmVf7VjqyNBOjWCA..&URL=http%3a%2f%2fdoi.org%2fgbkbwp)

“I find it’s usually best to use the style guidelines published by a programming languages developers. With that said… I cannot find an official one for R. Others have linked you to Google’s recommendations and they seem reasonable.

Another option could be to use a development tool called a linter that will check your code complies with some style standard and report nonconformities. It won’t fix them (usually). <https://github.com/jimhester/lintr>. I think this is particularly important for R as R Studio is weird with how it manages indentation, brace style etc and will let the user get away with anything.

On the more technical side of things you can use software design patterns to ensure everyone is developing code in a standardised way. Object orientation is really good for organising code and having a scalable architecture (not so easily achieved in R, though). <https://community.rstudio.com/t/design-patterns-for-r/1830>

If possible you and your team should do regular code reviews to make sure the code being checked in to master is of high quality and understandable to someone other than just the person who authored it. Look out for poorly named variables and functions, unnecessary comments, very complicated code and nested conditional blocks.

You could refer to: Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin)

And: Code Complete by Steve McConnell”