

Dear Reviewers,

To facilitate peer-review and reproducibility of our research, we are sharing the R code underlying our analyses. This document is a brief instruction on how to use the code.

All the codes is provided in the Github repository, under the paperCode folder, accessible from <https://github.com/Shoodood/mROC/tree/master/paperCode>

Please note that the code requires the mROC R package to be installed. To do so, you can do one of the following:

- If you have installed devtools package, you can directly install the mROC package by running the following command:

```
devtools::install_github("Shoodood/mROC")
```

Note that because the package include C++ files, this will require the availability of RCpp compiler on your system.

- If you are working with a MS Window-based PC, you should be able to download the package binaries and install the package from the downloaded (compiled) file. The binary can be downloaded from <https://gofile.io/?c=AED4wk> . Once downloaded, you can run the following command:

```
install.packages("[PATH]/mROC_0.0.0.9000.zip", repos = NULL, type  
= "win.binary")
```

with [PATH] replaced with the folder that contains the downloaded file.

Once the package was installed, please load the library (`library("mROC")`) source the mROC.paperCode.R file located in the paperCode folder. Note that mROC depends on the pROC package, and some of the analysis functions will require other dependency packages (gsubfn, proto, RSQLite, sqldf).

1. R code for the stylized example for Supplementary Material – Section 1 (marked as section 1 in the code):

```
stylized_example()
```

2. R code for the stylized example under subtitle “The connection between mROC curve, case-mix, and model calibration”, and Figure 1 (marked as section 2 – stylized simulation in the code):

```
stylized_simulation()
```

3. R code for the section titled “Simulation study” (marked as section 3 – detailed simulations in the code)

- The first set of simulations (reported in Supplementary Material – section 3) can be run by calling the following function:

```
detailed_sim_linear()
```

\*Note that the above function takes several arguments, but the results that are reported are based on default argument values. Importantly, the coefficients a and b, as described in the text, are represented by, respectively, b0 and b1 in the input arguments. Also, the population calibration and ROC plots can be generated by calling the function with `draw_plots="pp"` and `draw_plots="roc"`, respectively.

- The second set of simulations (reported in the main text and Figure 2, 3, and 4) can be run by calling the following function:

```
detailed_sim_power()
```

\*Note that the above function takes several arguments, but the results that are reported are based on default argument values. Importantly, the coefficients a and b, as described in the text, are represented by, respectively, b0 and b1 in the input arguments. Also, the population calibration and ROC plots can be generated by calling the function with `draw_plots="pp"` and `draw_plots="roc"`, respectively.

4. R code for the case study (section titled ‘Application’ in the main text, and Figure 5 – marked as section 4: case study in the code).

- When all exacerbations are the outcome:

```
case_study()
```

- When severe exacerbations are the outcome:

```
case_study(only_severe=TRUE)
```