



DEC 20, 2023

## Protocol for Medication Possession Ratio (MPR) Calculation

Leonardo Teodoro de

Priscila Rodrigues Moreira<sup>1</sup>, Farias<sup>1</sup>,

Ana Carolina Figueiredo Modesto<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Atenção e Avaliação em Saúde/Faculdade de Farmácia. Universidade Federal de Goiás (UFG), Goiânia, GO, Brasil.



Priscila Rodrigues Moreira

OPEN ACCESS



DOI:

[dx.doi.org/10.17504/protocols.io.j8nlkoy2wv5r/v1](https://dx.doi.org/10.17504/protocols.io.j8nlkoy2wv5r/v1)

**Protocol Citation:** Priscila Rodrigues Moreira, Leonardo Teodoro de Farias, Ana Carolina Figueiredo Modesto 2023. Protocol for Medication Possession Ratio (MPR) Calculation. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.j8nlkoy2wv5r/v1>

**MANUSCRIPT CITATION:** MOREIRA; FARIAS; MODESTO, 2023.

**License:** This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

### ABSTRACT

The Medication Possession Ratio (MPR) provides an estimate of patient adherence based on the actual possession of medications. It is a metric that offers crucial information to assist in the development of strategies for the proper management of chronic diseases, such as Chronic Myeloid Leukemia (CML). RStudio is a free, open-source software maintained by an active community of developers, ensuring frequent updates and specialized packages in the healthcare domain. This software can be easily integrated into the practices of healthcare professionals for MPR calculations, optimizing their activities and allowing more time for direct patient care. In this context, a protocol was developed using the R programming language for MPR calculation, utilizing a spreadsheet with Tyrosine Kinase Inhibitor dispensation data. This script provides flexibility for MPR calculations to be applied to any chronic disease.

### BEFORE START INSTRUCTIONS

The RStudio program must be installed on the computer. If it is not installed, the link below provides the tutorial for download.

<https://posit.co/download/rstudio-desktop/>

**Protocol status:** Working  
We use this protocol and it's working

**Created:** Dec 19, 2023

**Last Modified:** Dec 20, 2023

**PROTOCOL integer ID:**  
92501

**Keywords:** Patient  
Compliance, Chronic Disease,  
Leukemia Myeloid Chronic-  
Phase

## Installing the packages

- 1 In the specific context of this script, the section "Installing the packages" is dedicated to installing additional packages that are necessary to execute the subsequent analyses. Each line `install.packages("package_name")` installs a specific package. Press Ctrl + Enter to run the code. The code is as follows:

### Command

```
install.packages("tidyverse")  
install.packages("openxlsx")  
install.packages("psych")  
install.packages("gtsummary")  
install.packages("flextable")  
install.packages("magrittr")
```

## Loading the packages

- 2 Every time you open the software, you should load the following packages:

## Command

```
library(tidyverse) # Loads the Tidyverse, a collection of packages for data manipulation and visualization, including popular packages like dplyr and ggplot2.
library(openxlsx) # Loads the openxlsx library, which is used for spreadsheet manipulation and working with Excel files.
library(psych) # Loads the psych library, providing functions for statistical analysis and psychometrics.
library(gtsummary) # Loads the gtsummary library, used for creating summary tables in a concise and elegant manner.
library(flextable) # Loads the flextable library, providing a flexible approach to creating tables with various formatting options.
library(magrittr) # Loads the magrittr library, facilitating data manipulation using the pipe operator or %>%. The pipe operator is used to chain operations together in a readable and efficient way.
```



## Setting the working directory to a specific path

- 3** Setting the working directory is important because it determines the default location for reading and writing files.
- 1- In RStudio, go to the "Session" menu.
  - 2- Select "Set Working Directory" and choose "Choose Directory..."
  - 3- Navigate to the directory
- # Alternatively, you can use the command below:

## Command

```
Setwd ("C:/Path/To/Your/Directory") #put the path of your directory inside the parenthesis
```

## Reading CSV file

- 4** # Here we read the data file called "file.mpr." and store it in a variable called MPR.  
# The read.csv2 function is used to read data from a CSV file where the column separator is a semicolon

(;).

# The argument "file.mpr" represents the name of the file to be read. You should download the spreadsheet with the medication dispensation data from your institution's dispensation system. In this example, the obtained spreadsheet contains: Medication name, number of dispensed units, and patient medical record number. We chose to use the medical record number to avoid homonyms.

#### Command

```
MPR <- read.csv2("file.mpr.csv") #Add .csv after the file name
```

# Now, the variable MPR contains the data from the CSV file, and you can manipulate or perform additional analyses as needed.

## Selecting the columns of interest

- 5 Now, we select the columns that are relevant to our analysis and store them in a new variable called MPR1.

#### Command

```
MPR1 <- MPR %>%  
  select(Drug, Dispensed, Medical.record)
```

## Inspecting the Database

- 6 We use the "glimpse" command to visualize the structure of our MPR1 dataset.

#### Command

```
glimpse(MPR1)
```

## Transforming the medical.record column into character

- 7 In this step, we convert the by "Medical.record" column into characters to ensure it is handled correctly.

#### Command

```
MPR1$Medical.record <- as.character(MPR1$Medical.record)
```

## Calculating the percentage

- 8 We perform calculations in the "Dispensed" column based on the conditions specified for each "drug".  
We also group the data by "Medical.record" and "drug" and calculate the percentage.

## Command

```
MPR1 <- MPR1 %>%
  mutate(
    Dispensed = Dispensed / 365, #calculation for a period of 1 year)
    Dispensed = case_when(
      Drug == "Imatinibe 100mg" ~ Dispensed / 2, #referring to two tablets per day. In the case of 3
tablets, divide by 3, and so on.
      Drug == "Dasatinibe 20mg" ~ Dispensed / 2,
      Drug == "Nilotinibe 200mg" ~ Dispensed / 4,
      Drug == "Imatinibe 400mg" ~ Dispensed,
      Drug == "Dasatinibe 100mg" ~ Dispensed
    )
  ) %>%
group_by(Medical.record, Drug) %>%
summarise(
  Porcentagem = ifelse(sum(Dispensed) > 100, 100, round(sum(Dispensed), 2) * 100),
  .groups = "drop"
) %>%
mutate(Porcentagem = ifelse(Porcentagem > 100, 100, Porcentagem),
  Porcentagem = format(Porcentagem, digits = 2, nsmall = 2, scientific = FALSE) %>%
  paste0("%"))
```

## Sorting the table alphabetically by Drug

- 9 Now we sort the data alphabetically based on the "Drug" column.

## Command

```
MPR1 <- MPR1 %>%
  arrange(Drug)
```

## Saving the result to a .docx file

- 10 Here we create a flexible table (ft) and adjust its properties such as width and alignment.

### Command

```
ft <- flextable(MPR1)
set_table_properties(ft,
  width = 0.99,
  align = "center") #Aligns the table contents to the center.
```

## Saving the .docx document

- 11 Finally, we save the document with the name "MPR\_Percentagem.docx" in the specified location.

### Command

```
save_as_docx(ft, path = "C:/Path/To/Your/Directory/MPR_Percentagem.docx") #put the path of your
directory inside the parenthesis
```